

Universidad ORT Uruguay

Facultad de Ingeniería

**Servicio de consultoría para la implantación de
CRM (Customer Relationship Management) y
productos satélites**

Entregado como requisito para la obtención del título

Ingeniero en Sistemas

Marcos Bazzano - 220591

Juan Andrés Nervi - 223393

Romina Rodríguez - 209234

Tutor: Darío Macchi

2023

Declaratoria de Autoría

Nosotros, Marcos Bazzano, Juan Andrés Nervi y Romina Rodríguez, declaramos que el trabajo que se presenta en esta obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el proyecto final de carrera.
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad.
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra.
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros.
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



Marcos Bazzano

28 de septiembre de 2023



Juan Andrés Nervi

28 de septiembre de 2023



Romina Rodríguez

28 de septiembre de 2023

Entorno Conceptual de Software Factory y sus objetivos

El Laboratorio de Ingeniería de Software de la Universidad ORT Uruguay, denominado ORT Software Factory (ORTsf) se dedica a la enseñanza de Ingeniería de Software y a la producción de software en forma industrial [1].

ORTsf está abocada fundamentalmente a desarrollar en los alumnos las habilidades que un profesional de las Tecnologías de la Información debe dominar y aplicar. Para esto se ha diseñado un método de enseñanza para estudiantes de fin de carrera, que apoyados por tutores especializados, trabajan en equipos de desarrollo aplicando prácticas avanzadas de Ingeniería de Software en proyectos reales.

Estos proyectos surgen en colaboración con la industria o como apoyo a las líneas de investigación del departamento. Buscan construir productos que satisfagan a sus clientes, promover el aprendizaje de prácticas reales de ingeniería de software y proveer tecnología probada al mercado.

Agradecimientos

Queremos agradecer a nuestras familias, amigos y a todas las personas que nos apoyaron en la carrera y este proyecto, alentándonos a seguir adelante y dar lo mejor de nosotros.

El siguiente agradecimiento se lo dedicamos a nuestro tutor, Darío Macchi, quien primeramente confió en nosotros para llevar adelante el proyecto y nos apoyó en el transcurso de forma dedicada y comprometida, guiando al equipo en momentos complicados y dando consejos a lo largo del camino. Agradecer también por sus enseñanzas, que nos ayudaron a mejorar tanto a nivel personal como profesional. Fue un actor clave en cuanto a las decisiones tomadas en cada etapa del proyecto.

Además, agradecemos a nuestro cliente, Evimed, en especial a Antonio López y Agustín Gómez, por darnos la posibilidad de trabajar en este proyecto, por la confianza depositada en nosotros, brindando retroalimentaciones valiosas, dando sugerencias y aprendiendo con ellos.

Por otro lado, queremos agradecer a la Universidad ORT Uruguay y a todos los que nos acompañaron en este camino, especialmente a los docentes, coordinadores, ayudantes y bedeles que contribuyeron a formarnos a nivel académico, profesional y personal a lo largo de estos años.

Finalmente, agradecemos a los docentes que actuaron como revisores, Álvaro Ortas, Mariel Feder y Gastón Mousqués, dado que desde su experiencia aportaron al éxito del proyecto y nos brindaron retroalimentación clave que nos ayudó a mejorar nuestro proceso y proyecto.

A todos ellos, nuestros más sinceros agradecimientos, nada de esto podría haber sido posible sin su apoyo.

¡Muchas gracias!

Abstract

La empresa Evimed sufrió un cambio de modelo de negocio, lo cual implicó nuevos desafíos. En sus inicios, migró su modelo de negocio de B2B al B2C y B2B. Su arquitectura y sistemas actuales no permitieron escalar con facilidad. Es por esto que Evimed presenta varios sistemas que no se encuentran debidamente comunicados, dando lugar a problemas como el envío de información contradictoria a clientes, no poder segmentarlos fácilmente para campañas de marketing, haciendo que se necesite obtener dicha información a pedido y de forma manual. Es por esto que Evimed presenta su proyecto ante la Feria de Proyectos de la Universidad ORT Uruguay, y de donde nuestro equipo lo toma como propio.

El proyecto fue llevado a cabo con una duración de un año, donde el equipo profundizó en la problemática y la solución. La solución que el cliente propuso fue investigar diferentes CRMs que existen en el mercado, que cumplieran con determinadas características y restricciones que el equipo evaluó junto a la empresa Evimed, para luego implantar dicho CRM a la operativa diaria y de esa manera solucionar los problemas antes descritos.

Para lograr dicho objetivo fue que se aplicaron los conocimientos impartidos en la carrera, en cuanto al análisis de requerimientos, diseño, gestión del proyecto y aseguramiento de la calidad para lograr un proyecto exitoso.

Como resultado final, se optó por implantar Odo CRM a la operativa diaria de la empresa, mejorando dicho CRM por los productos satélites. Dichos productos fueron construidos especialmente para potenciar ciertos aspectos, como la comunicación entre sistemas de Evimed y Odo CRM mediante colas de mensajes *Amazon SQS*, la posibilidad de generación de templates de correos electrónicos para envío de campañas de marketing mediante la construcción de una extensión de Google Chrome implementada con Javascript, el framework *Vue.js* y la *API chat completions* de GPT introduciendo la utilización de la inteligencia artificial.

Para concluir, en base a comentarios y evidencia escrita recibidos, se constató que la solución logró satisfacer con creces las necesidades y expectativas del cliente.

Palabras Clave

CRM, Odoo, Javascript, Vue.js framework, Python, Adapter, Google Chrome Extension, AWS, Amazon SQS, colas de mensaje, inteligencia artificial, ChatGPT, OpenAI, análisis de sentimiento, segmentación de clientes, campañas de marketing, PopcornFlow, entorno ágil

Glosario

- **API (*Application Programming Interface*):** es un intermediario entre dos sistemas, que permite que una aplicación se comunique con otra y pida datos o acciones específicas.
- **AWS (*Amazon Web Services*):** es una plataforma de servicios en la nube proporcionada por Amazon, líder a nivel mundial, que ofrece una amplia gama de servicios de cómputo, almacenamiento, bases de datos, redes, análisis, aprendizaje automático y más, permitiendo a las organizaciones ejecutar aplicaciones y cargas de trabajo de manera escalable, segura y rentable en la nube.
- **Backlog:** lista priorizada de tareas a realizar.
- **B2B (*Business to Business*):** modelo de negocio donde una empresa vende sus productos o servicios directamente a otras empresas en lugar de a consumidores individuales.
- **B2C (*Business to Consumer*):** modelo de negocio donde una empresa vende sus productos o servicios a consumidores individuales.
- **Backend:** refiere a la parte de un sistema informático o una aplicación web que se encuentra en el lado del servidor.
- **Brainstorming:** *Brainstorming* o “tormenta de ideas” es un proceso de ideación basado en generar muchas ideas en un ambiente creativo con el equipo.
- **Brainwriting:** *Brainwriting* o “escritura de ideas” es una técnica de generación de ideas en grupo parecida al *Brainstorming*, en la que los participantes escriben individualmente sus ideas en una hoja y después construyen sobre las de los demás.
- **Bug:** se le considera bug a todo aquel error o problema que causa un comportamiento inesperado o incorrecto en un producto de software.
- **CRM (*Customer Relationship Management*):** es una aplicación o software que ayuda a las empresas a gestionar sus relaciones con los clientes, almacenar información relevante y mejorar la interacción con ellos para impulsar las ventas y la fidelización.

- **CSV (*Comma Separated Values*):** es un formato de archivo de texto utilizado para almacenar datos tabulares, donde cada fila representa un registro y los valores de cada columna se separan por comas.
- **Deployment:** proceso por el cual un sistema se proporciona a los usuarios para su uso.
- **DOM (*Document Object Model*):** es una representación en memoria de la estructura de un documento HTML o XML en una página web, permitiendo a los programadores acceder y manipular los elementos de la página de manera dinámica mediante código JavaScript.
- **Elastic Beanstalk:** es un servicio de plataforma como servicio (PaaS) de AWS que simplifica la implementación y administración de aplicaciones web, permitiendo a los desarrolladores cargar su código y dejar que AWS se encargue de la infraestructura subyacente, escalabilidad y balanceo de carga.
- **Elastic Computing Cloud (EC2):** es un servicio de cómputo en la nube de AWS que ofrece instancias de máquinas virtuales escalables y personalizables, permitiendo a los usuarios ejecutar aplicaciones y cargas de trabajo en una infraestructura informática altamente flexible y configurable.
- **Endpoint:** es un punto en el que un sistema recibe pedidos desde otro sistema.
- **Feedback:** consiste en la manifestación de una opinión, en la demostración de un punto de vista o incluso, en el análisis de algo.
- **Framework:** es un conjunto de herramientas y reglas predefinidas que facilitan el desarrollo de aplicaciones al proporcionar una estructura y funcionalidades comunes.
- **Frontend:** es la parte de una aplicación que los usuarios ven y con la que interactúan directamente, incluye la interfaz de usuario y la presentación de datos.
- **Insights:** se puede considerar como un entendimiento profundo de un problema o situación.
- **IT (*Information Technology*):** es la aplicación de ordenadores y equipos de telecomunicación para almacenar, recuperar, transmitir y manipular datos, con frecuencia utilizado en el de los negocios u otras empresas.

- **Jira:** plataforma de gestión de proyectos y seguimiento de problemas ampliamente utilizada en el desarrollo de software.
- **JSON (*JavaScript Object Notation*):** es un formato de intercambio de datos ligero y legible por humanos que se utiliza para representar datos estructurados. Los datos en JSON se organizan en pares clave-valor, lo que facilita su comprensión y manipulación.
- **LLM (*Large Language Model*):** sistema de inteligencia artificial diseñado para comprender y generar texto en lenguaje humano, entrenado en grandes cantidades de datos para realizar una variedad de tareas de procesamiento de lenguaje natural, gracias a su gran cantidad de parámetros de red neuronal.
- **MVP (*Minimum Viable Product*):** producto con suficientes características para satisfacer a los clientes iniciales, y proporcionar retroalimentación para el desarrollo futuro.
- **PaaS (*Platform as a Service*):** es un modelo de servicio en la nube que proporciona a los desarrolladores una plataforma completa que incluye herramientas y servicios para desarrollar, implementar y administrar aplicaciones sin preocuparse por la infraestructura subyacente. Esto simplifica el proceso de desarrollo y permite centrarse en la creación de aplicaciones en lugar de gestionar la infraestructura.
- **Prompt:** solicitud o entrada de texto proporcionada por el usuario que sirve como punto de partida para que el modelo de inteligencia artificial genere una respuesta o continúe una conversación.
- **Scheduler:** es una herramienta o componente de *software* que se utiliza para programar y automatizar la ejecución de tareas en un sistema informático en momentos específicos o según condiciones predefinidas, lo que ayuda a gestionar y optimizar procesos repetitivos o tareas programadas.
- **Secret Manager:** es un servicio de administración de secretos completamente gestionado que ayuda a proteger, rotar y recuperar de manera segura credenciales y otros datos confidenciales utilizados por aplicaciones en la nube de Amazon Web Services (AWS).

- **SQL (*Structured Query Language*):** es un lenguaje específico de dominio, diseñado para administrar y recuperar información de sistemas de gestión de bases de datos relacionales.
- **Story point:** los puntos de historia o story points permiten expresar en una medida el “esfuerzo” estimado de una tarea en base a su complejidad y tamaño.
- **Trello:** herramienta de gestión de proyectos basada en tableros, listas y tarjetas.
- **Tokens:** en el contexto de los modelos GPT de OpenAI, son grupos de caracteres que representan la unidad fundamental del texto.
- **UI:** User Interface o Interfaz de Usuario es la portada que tiene el usuario para interactuar con el sistema.
- **UX:** User Experience o experiencia de usuario es la forma en la que el usuario interactúa y experimenta el producto de Software.

Índice

1. Introducción	18
1.1. Descripción del equipo	18
1.2. Elección del proyecto.....	19
1.3. Descripción del cliente.....	19
1.4. Objetivos	20
1.4.1. Objetivos académicos	20
1.4.2. Objetivos personales	20
1.4.3. Objetivos de producto	21
1.5. Estructura del documento	21
2. Sobre el problema	23
2.1. Contexto.....	23
2.2. El problema propuesto	24
2.2.1. Situación actual del cliente	24
2.2.2. El problema.....	24
2.3. Objetivos del cliente	25
2.4. Interesados y necesidades	26
2.4.1. Análisis de interesados.....	26
2.4.2. Descripción de las necesidades de cada grupo	27
3. Descripción de la solución propuesta	28
3.1. Descripción funcional	28

3.1.1. Alcance inicial	28
3.1.2. Alcance final	29
3.2. Descripción general de la solución	30
3.3. Configuración de Odoo.....	30
3.4. Adapter.....	31
3.5. Extensión de Chrome.....	31
4. Primera etapa: Consultoría de CRMs	31
4.1. Proceso.....	32
4.1.1. Metodología de referencia	32
4.1.2. Operacionalización	34
4.2. Gestión del proyecto	39
4.3. Ingeniería de requerimientos.....	40
4.3.1. Relevamiento inicial	40
4.3.2. Especificación de requerimientos	41
4.3.3. Validación de criterios de selección	42
4.4. Informe de análisis de CRM	42
4.4.1. Principales resultados.....	43
4.4.2. Validación y aceptación de resultados	46
5. Segunda etapa: Implantación de Odoo CRM.....	48
5.1. Introducción	48
5.2. Ingeniería de requerimientos.....	48

5.2.1. Objetivos	48
5.2.2. Descripción general	48
5.2.3. Proceso de relevamiento	48
5.2.4. Resultados	50
5.2.5. Reflexiones del proceso	50
5.3. Requerimientos funcionales.....	51
5.4. Requerimientos no funcionales.....	53
5.5. Restricciones	54
5.6. Modelado de datos	54
5.7. Validación y aceptación de resultados.....	58
5.8. Reflexiones del proceso	60
6. Tercera etapa: Desarrollo de productos satélites	61
6.1. Introducción	61
6.2. Ingeniería de requerimientos.....	61
6.2.1. Objetivos	61
6.2.2. Descripción general del proceso	62
6.2.3. Metodologías utilizadas	62
6.2.3.1. Empatizar	62
6.2.3.2. Definir	63
6.2.3.3. Idear	64
6.2.3.3.1. Ideas obtenidas.....	66

6.2.3.4. Ausencia de la etapa de Prototipación	67
6.2.3.5. Validar.....	68
6.2.4. Resultados de la elicitación.....	68
6.3. Productos satélites.....	69
6.3.1. Adapter.....	69
6.3.1.1. Requerimientos funcionales.....	69
6.3.1.2. Requerimientos no funcionales.....	72
6.3.1.3. Restricciones	73
6.3.1.4. Arquitectura	73
6.3.1.4.1. Descripción de la arquitectura	73
6.3.1.4.2. Atributos de calidad, tácticas y patrones.....	77
6.3.2. Extensión de Google Chrome	81
6.3.2.1. Uso de la inteligencia artificial	82
6.3.2.2. Requerimientos funcionales.....	84
6.3.2.3. Requerimientos no funcionales.....	85
6.3.2.4. Restricciones	86
6.3.2.5. Arquitectura	86
6.3.2.5.1. Descripción de la arquitectura	88
6.3.2.5.2. Atributos de calidad, tácticas y patrones.....	90
6.3.3. Conclusiones	95
7. Gestión del proyecto	97

7.1. Proceso y ciclo de vida	97
7.1.1. Ciclo de vida	97
7.1.2. Metodología de referencia	97
7.1.3. Descripción general del proceso	97
7.2. Planificación temporal	102
7.2.1. Sobre la estimación	102
7.2.2. Proceso de planificación	103
7.3. Gestión del esfuerzo.....	103
7.3.1. Primera etapa	104
7.3.2. Segunda etapa	104
7.3.3. Tercera etapa.....	106
7.3.4. Documentación	107
7.3.5. Distribución de horas totales.....	107
7.4. Gestión de riesgos	108
7.4.1. Riesgos de cliente	109
7.4.2. Riesgos de producto.....	112
7.4.3. Riesgos personales	113
7.4.4. Riesgos académicos	113
7.5. Gestión de la comunicación	114
7.5.1. Comunicación interna	114
7.5.2. Comunicación con el cliente.....	115

7.5.3. Comunicación con el tutor	116
7.6. Métricas de gestión	117
7.7. Gestión de la configuración	119
7.7.1. Elementos de configuración.....	119
7.7.1.1. Elementos de software	120
7.7.1.1.1. Organización de repositorios	120
7.7.1.1.2. Uso de ramas.....	121
7.7.1.2. Elementos de documentación	122
7.8. Aseguramiento de la calidad.....	126
7.8.1. Estándares	126
7.8.1.1. Estándares de codificación.....	127
7.8.1.2. Estándares de documentación.....	128
7.8.2. Pruebas.....	129
8. Conclusiones y lecciones aprendidas.....	132
8.1. Análisis de objetivos planteados.....	132
8.1.1. Objetivos académicos	132
8.1.2. Objetivos personales	132
8.1.3. Objetivos de producto	133
8.1.4. Objetivos de cliente.....	133
8.2. Lecciones aprendidas	134
8.2.1. Aprendizajes de tecnología.....	134

8.2.2. Comunicación con el cliente.....	135
8.2.3. Uso de metodologías.....	135
8.2.4. Gestión de riesgos.....	136
8.2.5. Gestión de las emociones.....	136
8.2.6. Gestión del proyecto.....	137
9. Referencias bibliográficas.....	138
10. Anexos.....	141
10.1. Criterios de selección para el CRM.....	141
10.2. Informe de análisis de CRMs.....	144
10.3. Manual del desarrollador.....	184
10.4. Carta de conformidad del cliente.....	215

1. Introducción

Este documento tiene como propósito describir el proyecto de grado “Servicio de consultoría para la implantación de CRM y productos satélites”, llevado a cabo desde octubre de 2022 hasta septiembre de 2023, como parte de los requisitos para obtener el Título de Ingeniero de Sistemas de la Universidad ORT Uruguay.

Los objetivos del proyecto a gran escala fueron dos. El primero se trató sobre la realización de una investigación sobre los diferentes CRMs que hay actualmente en el mercado, teniendo en cuenta ciertos criterios de selección que se abordarán en el transcurso del documento. El segundo objetivo del proyecto, fue una vez realizada dicha investigación y habiendo obtenido un ranking de los mismos, implantar en la operativa de la empresa Evimed el CRM elegido por el equipo y validado por el cliente. Con esto, se buscará resolver una serie de problemas por los cuales la empresa Evimed presenta dicho proyecto.

A lo largo de todo el proyecto, se aplicaron diversas metodologías para los objetivos o etapas, así como técnicas de investigación y análisis para identificar necesidades y requerimientos de los empleados y directivos de la empresa Evimed.

Con la implantación de Odoo CRM (CRM elegido), la empresa Evimed logra tener de manera centralizada todos sus clientes, comunicaciones e inscripciones a cursos por parte de clientes. Además de la capacidad de poder segmentarlos para futuras campañas de marketing y visualizar todas estas entidades en tiempo real.

1.1. Descripción del equipo

El equipo está compuesto por tres estudiantes de la carrera Ingeniería en sistemas:

- **Marcos Bazzano:** estudiante de Ingeniería en Sistemas, software developer.
- **Juan Andrés Nervi:** estudiante de Ingeniería en Sistemas, software developer - BASF Services Américas.
- **Romina Rodríguez:** estudiante de Ingeniería en Sistemas, software engineer - MercadoLibre Uruguay.

Los integrantes se conocen desde los inicios de la carrera. Todos han estudiado juntos para diversas materias, aunque nunca habían tenido la oportunidad de trabajar los tres en un mismo proyecto lo que presentaba un desafío. Pese a ello, tenían la tranquilidad de conocer sus trabajos individuales.

1.2. Elección del proyecto

La elección del proyecto se toma de la “Feria de Proyectos” de la Universidad ORT Uruguay, de donde diferentes empresas de la industria, presentan sus proyectos año a año para que los estudiantes que estén comenzando su proyecto de grado puedan tomar de ella proyectos innovadores, desafiantes e interesantes que previamente se encuentran aprobados por la Universidad ORT Uruguay.

El proyecto de Evimed no fue la primera elección del equipo, dado a que no se trataba de un proyecto típico de desarrollo donde el equipo se sentía cómodo. Si bien tenía una parte de implantación y en ella integración de sistemas mediante desarrollo de código, también se debía investigar y convencer a la empresa Evimed que los CRMs elegidos por el equipo solucionarían los problemas planteados. Cabe destacar que, dependiendo del CRM elegido, se debían comprar licencias para su uso y negociar con el cliente, algo que a lo largo de la carrera no se ve con demasiada profundidad.

Por lo expuesto anteriormente, la selección del proyecto de Evimed, al cual el equipo decidió llamar “Servicio de consultoría para la implantación de CRM y productos satélites”, se considera que amplió sus conocimientos y lo hizo salir de su zona de confort.

1.3. Descripción del cliente

Evimed es una empresa líder en América Latina, dedicada a la educación médica. Trabajan en forma conjunta con importantes sociedades científicas de la región e internacionales. Es una empresa que mediante su plataforma de e-learning llamada “redEMC”¹, ofrece cursos para mantener a los profesionales de la salud actualizados.

Tiene aproximadamente 27 empleados y ha capacitado a más de 80.000 profesionales de la salud de toda la región.

¹ <https://redemc.net/>

1.4. Objetivos

Al inicio, el equipo se planteó los objetivos a cumplir para gestionar el proyecto de una manera exitosa, logrando estar alineados y enfocados en lo que se debía hacer para alcanzarlos.

Se optó por dividir estos objetivos en tres grupos principales: académicos, personales y de producto.

1.4.1. Objetivos académicos

- **Aprobar el proyecto final de carrera:** el principal objetivo planteado por el equipo es ser capaces de cumplir con los estándares exigidos por la Universidad ORT Uruguay y la industria, a modo de obtener el Título de grado de Ingeniero en Sistemas.
- **Relevar 12 CRMs:** ya que la implantación del CRM elegido corre por cuenta del equipo, debe cerciorarse de realizar una buena investigación para resolver los problemas del cliente.

1.4.2. Objetivos personales

- **Aprender a utilizar nuevas tecnologías:** para la implantación del producto seleccionado y la construcción de los productos satélites, el equipo debió de aprender tecnologías desconocidas de forma de cumplir con las restricciones del cliente para su posterior mantenimiento.
- **Aportar valor a la operativa de Evimed:** para lograr una solución que sea satisfactoria para el cliente, el equipo busca que se logre realizar tareas centralizadas con el nuevo software sin la necesidad de consultar a varios sistemas para obtener la información.
- **Obtener experiencia del rubro y dirección empresarial de la directiva de Evimed:** al mantener un estrecho contacto con uno de los socios fundadores de Evimed, quien tiene amplia experiencia dirigiendo la empresa, el equipo considera que esta es una gran oportunidad para aprender de él. De esta forma, no solo se obtiene experiencia en lo que conforma a la Ingeniería, sino que en las actividades de dirección empresarial, siendo esta una práctica útil para su vida profesional.

1.4.3. Objetivos de producto

- **Integrar un producto que satisfaga las necesidades del cliente:** el equipo deberá asegurarse que el producto final implantado y los productos satélites, cumplan con los requisitos y expectativas establecidos, disminuyendo las tareas manuales.
- **Cumplir con los plazos establecidos para las etapas y entregables acordados:** el equipo deberá de mostrar los avances y entregar los documentos acordados en tiempo y forma, asegurándose de cumplir con los requerimientos establecidos del proyecto.

1.5. Estructura del documento

Al tratarse de un proyecto atípico con grandes etapas, una de investigación de CRMs, otra de implantación del CRM elegido y por último una etapa adicional de realización de productos satélites (que será justificada más adelante), el equipo entiende que es pertinente explicar la estructura del documento.

Sobre el problema

Se pone en conocimiento del lector sobre quién es la empresa, su negocio y su contexto general. Luego se explica la situación actual del cliente, el problema en sí, los objetivos del cliente y los interesados y sus necesidades

Descripción de la solución propuesta

En este capítulo se describe el alcance inicial presentado por la empresa ante el equipo, el alcance final, la descripción general de la solución y los productos resultantes.

Primera etapa: Consultoría de CRMs

En este capítulo de investigación, se pone en conocimiento del lector acerca del proceso, la metodología aplicada para llevarlo a cabo, la gestión de esta parte del proyecto, la ingeniería de requerimientos aplicada y por último la elaboración de un informe de análisis sobre toda la investigación realizada entregada al cliente.

Segunda etapa: Implantación de Odoo CRM

En este capítulo se desarrolla el proceso de implantación de Odoo CRM, la especificación de requerimientos, su validación y la aceptación de los resultados obtenidos por el equipo.

Tercera etapa: Desarrollo de productos satélites

En este capítulo se especifica su ingeniería de requerimientos, los objetivos, la metodología utilizada, y por último se detallan los productos resultantes de la elicitación. En cada uno de los productos, se explica sus funcionalidades, restricciones encontradas, explicación de la arquitectura de los sistemas, la gestión y aseguramiento de la calidad llevadas a cabo.

Gestión del proyecto

Se detallan actividades y procesos relacionados con la gestión del proyecto.

Conclusiones y lecciones aprendidas

En este capítulo se realiza una retrospectiva general del proyecto. Se expone el análisis de los objetivos planteados, se identifican las oportunidades de mejora y se ponen en conocimiento las lecciones aprendidas por el equipo durante el transcurso del proyecto.

2. Sobre el problema

2.1. Contexto

Evimed es una empresa uruguaya, fundada en el 2004 por el ingeniero en computación Antonio López y el doctor Álvaro Margolis. Desde entonces, la empresa trabaja en forma conjunta con importantes sociedades científicas de la región y también internacionalmente, ofreciendo cursos médicos para profesionales mediante su plataforma de *e-learning* “redEMC”. En sus inicios Evimed fue incubada por Ingenio (Incubadora LATU/Universidad ORT).

La empresa ofrece cursos, tanto específicos para determinadas especialidades de la medicina, como generales. Existen cursos dictados en español, portugués e inglés. La estructura del curso está dada en módulos, donde cada curso tiene un máximo de 8 módulos. Los participantes pueden obtener tres posibles resultados:

- **Certificado:** el participante aprobó la cantidad de módulos necesarios para la obtención de un certificado de aprobación.
- **Constancia:** el participante no alcanzó la cantidad de módulos necesarios para la aprobación, pero aprobó suficientes para ser considerado partícipe del mismo y recibe una constancia de participación.
- **No aprobado:** el participante no alcanzó la cantidad de módulos aprobados necesarios para ser considerado partícipe del mismo.

Del punto de vista de negocio, es importante tener en cuenta que los dos primeros módulos de un curso pueden ser realizados sin necesidad de abonar el mismo, como un período de prueba que Evimed otorga a los participantes.

En sus inicios Evimed ofrecía cursos para empresas o instituciones establecidas, dejando de lado a los consumidores finales, es decir, su modelo de negocio al principio era B2B (*business to business*) y con el paso del tiempo adoptó también el modelo de negocio B2C (*business to customer*). Hoy en día, contemplando ambos modelos de negocios tienen alrededor de 140.500 clientes, donde 139.000 son clientes del tipo B2C y el restante 1500 del tipo B2B.

2.2. El problema propuesto

2.2.1. Situación actual del cliente

En este contexto planteado es que Evimed hoy en día realiza diversas actividades dependiendo de sus áreas. A efectos prácticos, se referirá sólo a las áreas de la empresa que dependen del problema: el área de IT y el área de marketing.

Evimed cuenta con un monitor provisto por la herramienta Looker Studio Overview², donde visualizan la cantidad de participantes activos en su plataforma de *e-learning*. Pueden realizar algunos filtros básicos sobre clientes y cursos, y se cuenta con gráficas donde se visualiza información general.

El equipo de marketing, cuando necesita alguna información adicional que no esté en este monitor, debe pedírsela al área de IT para que ellos, dado su conocimiento en la realización de consultas SQL, puedan brindarles la información solicitada.

2.2.2. El problema

La empresa Evimed enfrenta un problema crítico relacionado a la fragmentación de su base de datos. Esta situación se traduce en la dispersión de información esencial en múltiples sistemas y plataformas, lo que dificulta el acceso eficiente a los datos necesarios para la toma de decisiones estratégicas y operativas. La falta de integración de estos sistemas genera redundancias, incoherencias y errores que afectan la calidad de la información. La fragmentación de la base de datos no solo impacta la eficiencia operativa, sino que también representa un obstáculo significativo para la implementación de estrategias de análisis de datos.

Otro desafío importante que enfrenta Evimed se relaciona con la dificultad en la segmentación de clientes. La empresa opera con diversos sistemas y fuentes de datos para recopilar información sobre sus clientes. Esto implica que el proceso de segmentación requiera la extracción y consolidación de datos dispersos, lo que consume tiempo y recursos valiosos. La falta de una visión unificada de los clientes dificulta la identificación de patrones de comportamiento, preferencias, y necesidades, lo que limita la capacidad de la empresa para ofrecer ofertas personalizadas y relevantes. Además, esta falta de segmentación precisa

² <https://cloud.google.com/looker-studio?hl=es>

también afecta negativamente la eficacia de las estrategias de marketing y ventas, ya que no se pueden dirigir adecuadamente las campañas a segmentos específicos de clientes.

La empresa Evimed se enfrenta a un problema crucial en cuanto a la comunicación con sus clientes. La falta de un sistema centralizado de seguimiento y registro de las interacciones con los clientes ha llevado a situaciones en las que múltiples equipos de la empresa contactan a los mismos clientes repetidamente y, en ocasiones, con información contradictoria. Esto no solo crea una experiencia insatisfactoria para los clientes, sino que también puede dar lugar a la pérdida de clientes valiosos y dañar la reputación de la empresa. La falta de registro de estas comunicaciones también dificulta la capacidad de la empresa para aprender de las interacciones pasadas y mejorar la calidad de su servicio al cliente.

El cambio del modelo de negocio de Evimed, de uno exclusivamente B2B a uno que incorpora componentes B2B y B2C, ha expuesto una brecha en la arquitectura de la empresa. La infraestructura original fue diseñada para atender un alcance más limitado, y ahora enfrenta desafíos en términos de escalabilidad y adaptación. La transición a un enfoque B2B y B2C exige una mayor flexibilidad y capacidad de respuesta para satisfacer las demandas y expectativas de los consumidores individuales.

2.3. Objetivos del cliente

El equipo y Evimed elaboraron los objetivos para el proyecto.

- **Solucionar el problema de la fragmentación de la base de datos:** se deberá solucionar el problema de la fragmentación de la base de datos, consolidando en un único lugar la información que hoy en día la empresa Evimed cuenta con respecto a sus clientes, cursos y pagos.
- **Mejorar la segmentación de clientes:** se deberá mejorar la segmentación de clientes para permitir aplicar filtros que hoy en día son difíciles de aplicar, mejorando entonces la autonomía del área de marketing.
- **Registrar las comunicaciones entre clientes y Evimed:** se deberá llevar registro de las comunicaciones entre clientes y empresa, pudiendo ser posible visualizar las distintas comunicaciones realizadas a lo largo del tiempo y también el tipo de contacto (ya sea mediante *email*, *WhatsApp*, u otro).

- **Generar pruebas de performance:** dado al aumento exponencial de clientes, se busca generar pruebas de performance de la solución para ver cómo se comporta.

Estos objetivos fueron los que moldearon el alcance inicial y final de la solución y permitieron al equipo de desarrollo enfocar sus esfuerzos. Además, estos objetivos también sirvieron como guía para la evaluación del éxito del proyecto al comparar los resultados obtenidos con los objetivos establecidos.

2.4. Interesados y necesidades

2.4.1. Análisis de interesados

Para analizar las necesidades del cliente, es importante separarlo en los diferentes grupos de interesados que se puedan identificar. El equipo encontró tres grupos de interesados diferentes, estos son el equipo de marketing, el de IT y la directiva de Evimed, los cuales tienen necesidades distintas entre sí. Para definir las diferentes estrategias a aplicar con cada subgrupo, primero son categorizados mediante el uso de la matriz de poder/interés, la cual es una matriz que posiciona a los diferentes interesados según su nivel de interés en el proyecto y su poder de influencia en el mismo.

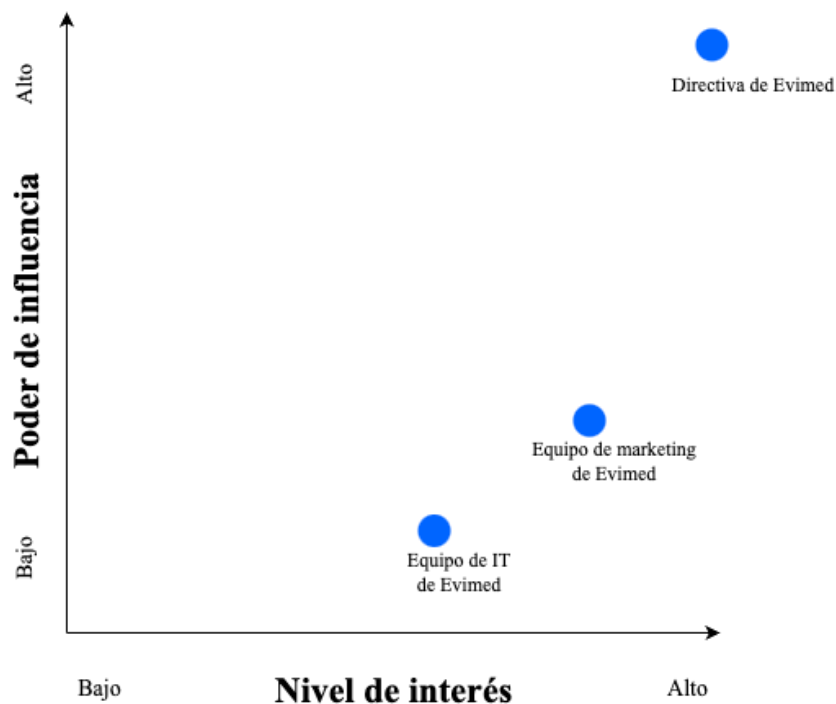


Gráfico 1: Poder e interés de grupos interesados

Observando el Gráfico 1, el interesado a priorizar es la directiva de Evimed, representada por Antonio López, a quien se debe mantener lo más cerca posible del proyecto, las actividades y decisiones del equipo. Esto por un lado es una ventaja, debido a que como fue mencionado anteriormente, Antonio no sólo es el interesado más importante, sino que el principal contacto con el cliente, lo cual hace más sencilla la tarea de mantener estrecho contacto con él. Por otro lado, genera un gran riesgo (riesgo R08, de la sección [7.4.1. Riesgos de cliente](#)) en caso de perder el contacto con él, no sólo se pierde la principal fuente de información del negocio del equipo, sino que también la cercanía con el principal interesado.

2.4.2. Descripción de las necesidades de cada grupo

Comenzando por el grupo de interesados más relevante, la directiva de Evimed, se destaca que sus necesidades son en mayor parte una aglomeración de las necesidades del resto de los subgrupos, ya que su principal necesidad es la completitud del proyecto en tiempo y forma. La otra necesidad de la directiva de Evimed que solo ellos tienen, es que el costo monetario del proyecto sea lo menor posible, lo cual fue un punto clave de consideración para el equipo al momento de tomar cualquier tipo de decisión respecto al proyecto.

Continuando con el segundo grupo de mayor nivel de interés, se tiene al equipo de marketing. Este equipo será el usuario final del producto resultante, por lo que es de vital importancia para ellos la usabilidad y la UI/UX del mismo, junto a que el producto entregado facilite y permita sus flujos regulares de trabajo de hoy en día. Es importante que el equipo los mantenga al tanto del progreso del producto y realice pruebas sobre la usabilidad del producto (si es posible) para obtener feedback y oportunidades de mejora.

Finalmente, el equipo de IT de Evimed tiene como necesidad primordial la mantenibilidad del sistema, dado que ellos serán los encargados de continuar con su ciclo de vida, realizando tareas de mantenimiento, actualización o desarrollo de nuevas funcionalidades que deseen a futuro. Es importante para ellos que estén familiarizados a trabajar con la tecnología utilizada en el proyecto, para minimizar su curva de aprendizaje al momento de transferir el proyecto finalizado.

3. Descripción de la solución propuesta

3.1. Descripción funcional

3.1.1. Alcance inicial

El alcance inicial fue dado entre Evimed y la Universidad ORT Uruguay, a través de lo presentado en el documento a la *Software Factory*.

El proyecto constaba de un MVP para validar la idea de la empresa de tener un omnicanal que no genere saturación por el trabajo manual. Se deseaba realizar una investigación de los CRMs que se encontraran en el mercado, comparándolos en base a los requerimientos del proyecto del cliente, y luego poder implantar e integrar su sistema en la herramienta CRM seleccionada. Esta solución fue solicitada por el cliente, no fue una sugerencia ni una idea planteada por el equipo.

Los sistemas que el cliente quería implantar e integrar eran los siguientes:

Sistema	Tipo	Mecanismo integración
Pagos	Interno	Base de datos
Consultas de contactos	Interno	Base de datos
Certificados	Interno	Base de datos
Estadísticas académicas	Interno	Base de datos
Actividad en Campus	Interno	Base de datos
Datos personales	Interno	Base de datos
Email marketing (Icegram ³)	Interno	Base de datos
Procesos automáticos	Interno	Base de datos
Marketing de influencers	Interno	Base de datos
WhatsApp (WATI ⁴)	Externo	API
Soporte (FreshDesk ⁵)	Externo	API
ZOOM ⁶	Externo	API
Gestión de contratos a proveedores	A contratar	A definir

Tabla 1: Sistemas de Evimed a implantar e integrar en CRM

³ <https://www.icegram.com/>

⁴ <https://www.wati.io/>

⁵ <https://support.freshdesk.com/en/support/home>

⁶ <https://zoom.us/es>

De aquí surgía que el proyecto tenía dos partes bien definidas. Una consultoría para la elección del CRM que mejor se adaptaba a las necesidades del cliente, que finalizaba con un informe del análisis de los CRMs evaluados. Y una parte de implantación de sus entidades en el CRM e integración con servicios externos, cuyo entregable sería la herramienta ya configurada y el código necesario para las integraciones extras.



Figura 1: Alcance inicial

3.1.2. Alcance final

Luego de realizada la primera parte del proyecto (consultoría), se decidió que la herramienta CRM sería Odoo. El cliente prefirió ir por la parte de las licencias pagas y que las bases de datos estuvieran hospedadas en los servidores de Odoo, pero se tuvo un conflicto con las licencias que permitían el desarrollo de código en los servidores de Odoo (Odoo.sh⁷), ya que elevaban considerablemente el precio. Se pidió comenzar con la implantación de los datos que vendrían por base de datos y luego se resolvería qué se haría con las integraciones externas.

Llegado el punto en el que se tenía que resolver esto, se decidió que las integraciones externas no se harían, y se tratarían las que fueran posibles como datos extra de sus bases de datos. Aquí se vio afectado el alcance del proyecto, por lo que se agregaron productos extras para complementar a la solución. Por un lado, un adaptador entre el sistema del cliente y la API de Odoo para actualizar los datos en tiempo real, y por otro lado una extensión de Google Chrome

⁷ <https://www.odoo.sh/>

para facilitar dos tareas que el equipo de marketing realiza regularmente, así facilitando y complementando al objetivo del cliente de disminuir el trabajo manual.

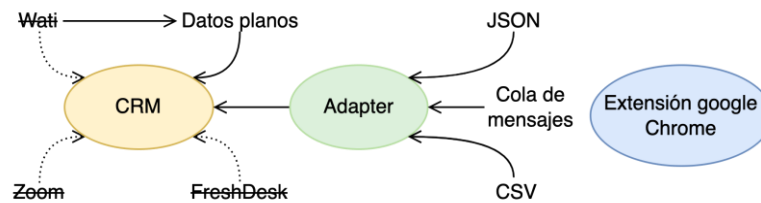


Figura 2: Alcance final

En la Figura 2 se muestra explícitamente el giro que se tomó respecto al alcance inicial (Figura 1).

3.2. Descripción general de la solución

Como se mencionó en la sección anterior, [3.1.2. Alcance Final](#), el proyecto tomó un nuevo rumbo a la mitad de la carrera. Se pasó de una gran implantación y programación sobre un CRM, a tener tres productos bien definidos que en conjunto aportan notoriamente a la solución del problema del cliente: disminuir el trabajo manual. En los siguientes puntos se hará un breve resumen de lo que aporta cada producto al problema del cliente, pero se profundizará más en las siguientes secciones.

3.3. Configuración de Odoo

Este producto es el principal y original, es lo que el equipo de marketing utilizará todos los días para llevar a cabo sus tareas. Aquí se concentrará toda la información importante de los clientes de Evimed, desde sus datos de contacto hasta el historial de los cursos que ha tomado y los contactos que ha tenido con la empresa. Teniendo este producto accesible a todos, hará que el equipo de marketing no tenga que depender del equipo de IT para realizar consultas y segmentar a los clientes para poder enviarles email marketing.

3.4. Adapter

Para acompañar a la integración entre Odoo y Evimed, se generó un producto al que se lo conocerá como “Adapter”. Este producto es como dice su nombre, un adaptador, un intermediario entre Odoo y Evimed para la importación de datos. Es una interfaz que recibe las actualizaciones de los clientes, ya sea por JSON, CSV o cola de mensajes, y las impacta directamente en Odoo. El desarrollo de las colas de mensajes es algo fundamental para el equipo de desarrollo, ya que permitirá una vez migradas sus bases de datos, mantenerlas actualizadas con la información entrante de sus plataformas sin la necesidad de que una persona las cargue manualmente.

3.5. Extensión de Chrome

El último producto es una extensión de Google Chrome. Al querer buscar agregar más valor para el cliente, se relevaron dos funcionalidades que ayudarán mucho en el día a día del equipo de marketing:

- **Generador de correos mediante api GPT:** esta herramienta ayudará día a día al equipo de marketing para elaborar correos electrónicos. Mediante pocos inputs, se generará un template de correo que ayudará al equipo de marketing a la redacción en vez de tener que pensar todo el mensaje.
- **Analizador de sentimientos:** esta herramienta ayudará a analizar los comentarios de las encuestas finales de curso, resultando en un comentario general y una puntuación de 1 a 5.

4. Primera etapa: Consultoría de CRMs

4.1. Proceso

En esta primera etapa se encuentra la Consultoría de CRMs. Como se mencionó en la sección anterior, el cliente estaba interesado en implantar un CRM del mercado. Para esto, fue necesario hacer un reconocimiento de algunos CRMs y evaluarlos para ver cuál se adaptaba mejor a las necesidades, concluyendo en un informe final y un top 3 de CRMs para que el cliente decida cuál era la mejor opción.

4.1.1. Metodología de referencia

La metodología que se utilizó para esta parte del proyecto fue la metodología PopcornFlow [2]. Esta metodología habla de siete pasos y cinco principios a seguir para poder experimentar rápidamente y tomar las mejores decisiones.

Los siete pasos son:

- *Problems and observations*: Problemas puntuales sobre los que se quiera trabajar.
- *Options*: Opciones/sugerencias para solucionar los problemas.
- *Possible experiments*: Experimentos posibles que surgen de las opciones prometedoras.
- *Committed*: Experimentos que se comprometen a llevar a cabo. Los experimentos tienen una acción, una razón, expectativas, duración y fecha de revisión.
- *Ongoing*: Cuando el experimento se puso en marcha.
- *Review*: Donde se hace la revisión de la iteración marcada por el equipo. Tiene específicamente las siguientes preguntas:
 - ¿Qué experimentos acordamos que haríamos?
 - ¿Cuáles realmente hicimos?
 - ¿Qué esperábamos que sucediera?
 - ¿Qué realmente sucedió?

- ¿Qué aprendimos?
- Basado en lo que aprendimos, ¿qué haremos a continuación?
- *Next*: Próximos pasos y experimentos a realizar.

Los cinco principios [3] son:

- *“If change is hard, make it continuous.”* – 1er principio de PopcornFlow, conocido como *“The Virus Principle”*.
- *“It’s not only what you do but also what you learn by doing it that matters.”* – 2do principio de PopcornFlow, conocido como *“The Ladder Principle”*.
- *“Everybody is entitled to their own opinion, but... a shared opinion is a fact.”* – 3er principio de PopcornFlow, conocido como el principio *“The Freedom Principle”*.
- *“It’s not “fail fast, fail often”... It’s “learn fast, learn often”.* – 4to principio de PopcornFlow, conocido como el principio *“The skateboarder”*.
- *“Small bets, big win.”* – 5to principio de PopcornFlow, aún en progreso.

4.1.2. Operacionalización

El equipo aplicó la metodología *PopcornFlow* mediante un tablero de Trello.



Figura 3: Tablero *PopcornFlow*

En este tablero se reflejan los primeros cinco pasos del *PopcornFlow*. En cada paso se encuentran las tarjetas que hacen referencia a cada uno de los problemas, opciones y experimentos que el equipo realizó, siguiendo la siguiente nomenclatura:

- **Problemas:** “ P_i - Descripción del problema”, donde la i hace referencia a una numeración, comenzando por el número uno.
- **Opciones:** “ P_iO_j - Descripción de la opción”, donde P_i hace referencia al problema para el cual es presentada la opción y la j hace referencia a una numeración, comenzando por el número uno.
- **Experimentos:** “ $P_iO_jE_k$ - Descripción del experimento”, donde P_i y O_j hacen referencia al problema y opción, para el cual es presentado el experimento, y la k hace referencia a una numeración, comenzando por el número uno.

Problems

El equipo tomó dos grandes problemas que se les presentó a la hora de dar comienzo con la consultoría:

- “P₁ - Hay mucha variedad de CRMs en el mercado para comenzar a analizar cada uno”
- “P₂ - No sabemos por dónde empezar con los requerimientos del cliente”

El primer problema surgió el día uno, cuando se comenzó a entender la metodología y enseguida se dio comienzo a la investigación. Más adelante, luego de realizados algunos experimentos y comenzando a documentar, surgió que no se tenía una especificación clara de los requerimientos del cliente ni un formato para identificar si lo investigado era acorde a lo que el cliente quería o no. Fue así que surgió el segundo problema, el equipo no sabía por dónde comenzar con los requerimientos del cliente.

Options

Para el primer problema, se plantearon las siguientes opciones:

- “P₁O₁ - Tomar CRMs, evaluarlos y generar documento de análisis”
- “P₁O₂ - Hacer *research* de opiniones generales del público sobre los CRMs”
- “P₁O₃ - Realizar demostraciones con expertos”
- “P₁O₄ - Usar técnica *Snowball Sampling* (no exacta) con los CRMs”

El primer objetivo era tomar CRMs y evaluarlos para corroborar si se adaptaban a las necesidades del cliente y así tomarlo como opción para hacer otros experimentos. Luego, una vez determinado que el CRM podría ser una posible solución, poder tener una demostración con el equipo de ventas para evaluar precios y funcionalidades que el sistema requiriera para poder implantar todo lo que necesitaba el cliente.

Además, se aplicó una copia de la técnica de *Snowball Sampling* [4], la cual constaba en pegar en los buscadores uno de los CRMs que eran potencial solución y poner “vs”, a modo de que el buscador de Google de CRMs muestre lo que el público en general toman como competencia directa.

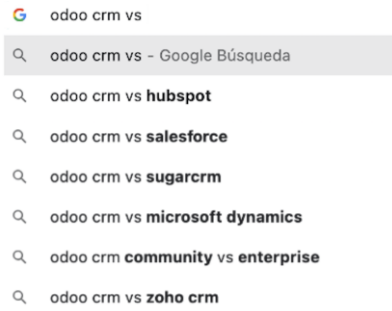


Figura 4: Ejemplo de *Snowball Sampling*

Para el segundo problema, la opción que se tenía era repasar más estrictamente con el cliente cuáles eran los requerimientos funcionales y no funcionales del CRM a elegir.

- “P₂O₁ - Repasar los requerimientos con el cliente”

Possible experiments

Si bien no todos los experimentos surgieron en la primer iteración, aquí se deja la lista completa de experimentos que se lograron finalizar:

- “P₁O₁E₁ - Evaluar Hubspot y agregar al documento de análisis”
- “P₁O₁E₂ - Evaluar Salesforce y agregar al documento de análisis”
- “P₁O₁E₃ - Evaluar Zoho y agregar al documento de análisis”
- “P₁O₁E₄ - Evaluar SugarCRM y agregar al documento de análisis”
- “P₁O₁E₅ - Evaluar Dynamics (Microsoft) y agregar al documento de análisis”
- “P₁O₁E₆ - Evaluar Odoo y agregar al documento de análisis”
- “P₁O₁E₇ - Evaluar Pipedrive y agregar al documento de análisis”
- “P₁O₁E₈ - Evaluar Groundhogg y agregar al documento de análisis”
- “P₁O₁E₉ - Evaluar FirmaO y agregar al documento de análisis”
- “P₁O₁E₁₀ - Evaluar Siebel CRM y agregar al documento de análisis”
- “P₁O₁E₁₁ - Evaluar SalesLogix y agregar al documento de análisis”

- “P₁O₂E₁ - Hacer un post en Reddit/CRM comentando nuestra problemática y requerimientos para obtener feedback”
- “P₁O₃E₁ - Demostración con experto de Odoo”
- “P₁O₃E₂ - Demostración con experto de Hubspot”
- “P₁O₃E₃ - Demostración con experto de Zoho”
- “P₁O₄E₁ - Utilizar snowball sampling con Odoo”
- “P₁O₄E₂ - Utilizar snowball sampling con Hubspot”
- “P₁O₄E₃ - Utilizar snowball sampling con Zoho”
- “P₂O₁E₁ - Repasar junto al cliente los servicios a integrar”
- “P₂O₁E₂ - Pedirle al cliente ejemplos de lo que hace en cada servicio a integrar”
- “P₂O₁E₃ - Validar con el cliente cuál sería la forma de integración con cada uno de los servicios (MySQL o API)”

Committed

Para el paso de *committed*, el equipo realizó reuniones en las que se planificaba qué experimento tomarían en esa iteración, teniendo en cuenta la carga horaria que cada uno podía dedicarle y planificando cada experimento con el formato que se requería.



Figura 5: Ejemplo de especificación de un experimento

Se asignaban los experimentos a los integrantes del equipo y luego se daba inicio a la iteración.

Ongoing

Para representar este paso, se utilizaba una columna en el Trello con el nombre “Doing”, como se puede apreciar en la Figura 2. Cada experimento tenía una duración específica, pero si algo no quedaba claro o no se encontraba la información, se avisaba al equipo y se agregaban horas a la duración, sabiendo que eso podría impactar en la planificación de la iteración.

Review

Se hicieron cuatro instancias de revisión a lo largo de la etapa de consultoría: el 9 de diciembre, el 10 de enero, el 10 de febrero y el 6 de marzo. En ellas, se planteaban las seis preguntas del paso como se muestra en la Figura 6, sacando accionables y puntos de mejora para la próxima iteración.



Figura 6: Review del 10 de enero

Next

Como próximos pasos el equipo tomaba lo aprendido en la iteración pasada, revisaba el *backlog* de experimentos y creaba nuevos que pudieran surgir con los experimentos concluidos. Luego se volvía al paso de *committed* para planificar una nueva iteración.

4.2. Gestión del proyecto

Para esta primera etapa, el equipo contaba con poca disponibilidad horaria ya que se encontraba finalizando el segundo semestre del año 2022, con preparaciones de exámenes y licencias que ya tenían planificadas.

	16/11/2022	30/11/2022	14/12/2022	28/12/2022	11/1/2023	25/1/2023	8/2/2023	22/2/2023	30 horas quincenales
Juan Andrés	Parciales	Obligatorios	Licencia						15 horas quincenales
Marcos	Parciales	Obligatorios	Fiestas	Licencia	Licencia		Exámenes		10 horas quincenales
Romina	Parciales	Obligatorios	Fiestas	Licencia	Exam. y obligatorio				5 horas quincenales

Figura 7: Disponibilidad del equipo para la primer etapa junto a la referencia

Se habló con el cliente del cronograma que se llevaría, dando a conocer que esta etapa comenzaría el 16 de noviembre de 2022 y finalizaría el 8 de marzo de 2023, con la entrega del informe. En ese tiempo, el equipo pudo evaluar sistemas que no conocían y entender las necesidades del cliente que eran complejas de obtener.

	16/11/2022	30/11/2022	14/12/2022	28/12/2022	11/1/2023	25/1/2023	8/2/2023	22/2/2023
Investigación	3 meses							
Matriz de Pugh		2 meses						
Informe de selección						1 mes y 1/2		

Figura 8: Roadmap primera etapa

En este período se contaron 276 horas. La investigación fue a lo que más tiempo se le dedicó, como se muestra en la Figura 8, ya que el equipo sabía que estaba limitado pero era la parte esencial. Esta herramienta era lo que más aportaría valor al cliente. Lo que representa cada proyecto que se muestra en la Figura 9, se explica en la sección 7.3. Gestión del esfuerzo.

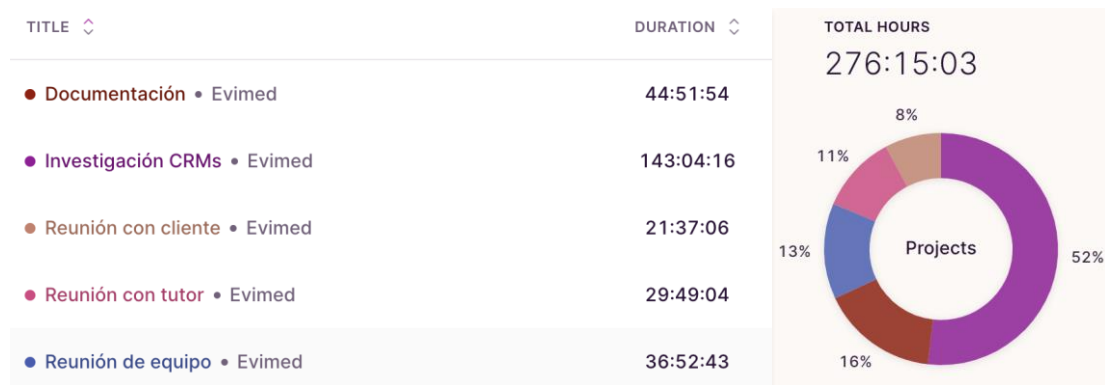


Figura 9: Distribución horaria respecto a los proyectos

Si no se hubiera tenido la metodología de Popcorn Flow, hubiera sido más compleja la organización del equipo para investigar y relevar información importante sobre los CRMs.

Cada paso y principio de la metodología ayudó a que el trabajo se diera de forma fluida y organizada. No es una metodología difícil de entender, pero a veces costaba dar seguimiento, sobre todo porque en esta parte faltó quien oficiara de Coach.

Pese a eso, el equipo logró llevar a cabo las cuatro iteraciones de forma correcta y es una nueva metodología aprendida que podrán aplicar en distintos proyectos a lo largo de su carrera.

4.3. Ingeniería de requerimientos

4.3.1. Relevamiento inicial

En el comienzo, el equipo comenzó directamente con la investigación, teniendo como requerimientos la carta de presentación de la empresa. A medida que se avanzaba, se notó que lo que se estaba documentando estaba siendo desprolijo, dado que no se tenían requerimientos específicos que el sistema debiera cumplir para poder evaluar si los CRMs los cumplían o no. Es por esto que el equipo, además de seguir investigando, decidió realizar un documento llamado “Criterios de selección para el CRM” (Anexo 10.1).

Este documento surgió como la especificación de requerimientos de la implantación del sistema.

4.3.2. Especificación de requerimientos

Como se mencionaba en la parte anterior, la especificación de requerimientos se dio mediante un documento que realizó el equipo de desarrollo y se corrigió junto al cliente.

Este documento habla de doce requerimientos que el sistema debe intentar cumplir. Cada uno de ellos está ponderado con una nota del 1 al 5, siendo 1 el valor más bajo y 5 el más alto. Esta ponderación se aplica para determinar qué requerimiento es más importante que otro, dado que no se podía asegurar que algún CRM pueda cumplir con todos.

Requerimientos funcionales con su debida ponderación:

- Customización de datos: Personalizar entidades (5)
- Distinción B2B y B2C (3)
- Segmentación de clientes: Filtros de segmentación (5)
- Soporte a usuarios: Soporte para usuarios de EviMed (4)
- Importación de datos (3)

Requerimientos no funcionales con su debida ponderación:

- Infraestructura: Lenguaje, sistema operativo y base de datos (5)
- Modo de licenciamiento: Gratuito o con licencias (5)
- Integraciones: Disposición de APIs, integración SMTP (5)
- Interfaz modificable (1)
- Soporte para mantenimiento del CRM: Soporte para usuarios de Odoo (2)
- Existencia de aplicación móvil (1)
- Hosting: On premise - On Cloud (3)

Con estos requerimientos se creó una Matriz de Pugh, en la que se pusieron como criterios los requerimientos, y como conceptos cada CRM evaluado como se muestra en la Figura 10.

Criterios	Conceptos											
	Peso	Odoo	Hubspot	Zoho	Pipedrive	Firmao	SugarCRM	Groundhogg	Salesforce	Dynamics	Siebel CRM	SalesLogix
Infraestructura	5											
Modo de licenciamiento	5											
Integraciones	5											
Customización de datos	5											
Distinción B2B & B2C	3											
Segmentación de clientes	5											
Interfaz modificable	1											
Soporte para mantenimiento del CRM	2											
Soporte a usuarios	4											
Existencia de aplicación móvil	1											
Importación de datos	3											
Hosting	3											
Estabilidad de licenciamiento	1											
Suma positivos (+)	-	0	0	0	0	0	0	0	0	0	0	0
Suma negativos (-)	-	0	0	0	0	0	0	0	0	0	0	0
Total	-	0	0	0	0	0	0	0	0	0	0	0

Figura 10: Matriz de Pugh

Como se puede apreciar, el tema del licenciamiento, infraestructura, customización (incluyendo integraciones) y segmentación eran los más relevantes para el cliente. Entonces se debía buscar un CRM que mínimamente cumpliera con ellos.

4.3.3. Validación de criterios de selección

Para validar estos requerimientos, que se les adjudicó el nombre de “Criterios de selección”, se hizo el documento y se validó con el cliente uno por uno. Primero, asíncronamente por el documento compartido de Google, y luego en una reunión de Google Meet para revisarlos en conjunto y evacuar dudas existentes. Allí el cliente dio el visto bueno, y a partir de ellos se comenzó a evaluar y describir cada uno de los CRMs investigados.

4.4. Informe de análisis de CRM

Como cierre de la etapa de consultoría, se acordó un entregable que mostrara el análisis realizado por el equipo, junto a una conclusión de cuáles CRMs eran los que el equipo consideraba mejor para realizar el trabajo.

4.4.1. Principales resultados

El informe constaba inicialmente de dos partes:

1) Análisis de CRMs

En esta sección, se evaluaron los once CRMs que se muestran en la Figura 10. Para cada uno de ellos, se le dio investigación bajo los Criterios de selección.

El objetivo era poder validar si cumplía o no con el criterio y agregar información extra si era necesario. En casos donde se observaba que a grandes rasgos no se cumplía con alguno de los requerimientos mayor ponderados, se hacía la investigación pero no se le daba la profundidad necesaria ya que se sabía que se descartaría.

Los once CRMs evaluados fueron los siguientes:

Criterios	Peso	Conceptos										
		Odoo	Hubspot	Zoho	Pipedrive	Firmao	SugarCRM	Groundhogg	Salesforce	Dynamics	Siebel CRM	SalesLogix
Infraestructura	5	1	1	1	1	0	1	0	1	-1	-1	-1
Modo de licenciamiento	5	1	0	1	1	1	-1	1	-1	-1	-1	-1
Integraciones	5	1	1	1	1	1	1	1	1	1	1	1
Customización de datos	5	1	1	1	1	1	1	1	1	1	1	1
Distinción B2B & B2C	3	1	1	1	1	0	1	1	1	1	1	1
Segmentación de clientes	5	1	1	1	1	1	1	1	1	1	1	1
Interfaz modificable	1	1	1	1	1	1	1	-1	1	1	1	1
Soporte para mantenimiento del CRM	2	1	1	1	1	-1	1	1	0	1	1	1
Soporte a usuarios	4	1	1	1	0	1	1	0	1	1	1	1
Existencia de aplicación móvil	1	1	1	1	1	1	1	-1	1	1	0	1
Importación de datos	3	1	1	1	1	1	1	1	1	1	1	1
Hosting	3	1	1	1	0	1	1	1	1	-1	1	1
Estabilidad de licenciamiento	1	1	1	-	-	-	-	-	-	-	-	-
Suma positivos (+)	-	43	38	42	35	32	37	31	35	29	31	32
Suma negativos (-)	-	0	0	0	0	2	5	2	5	13	10	10
Total	-	43	38	42	35	30	32	29	30	16	21	22

Figura 11: Matriz de Pugh completa

En esta matriz se muestran los CRMs evaluados bajo los criterios y la puntuación que obtuvieron, teniendo como referencia “1 - Bueno”, “0 - Neutro”, “-1 - Malo”.

2) Top 3 CRMs

El objetivo era que el equipo pudiera darle al cliente un top 3 de los CRMs que evaluaron, para darle varias opciones y el cliente poder elegir la que más le conviniera.

En este top 3 se encontraban:

- Odoo

- Hubspot
- Zoho

Luego de este análisis y la evaluación del top 3, se planificó una reunión con el cliente siguiendo las pautas para una reunión efectiva:

Fecha: 23 de febrero, 16:30hrs.		
Lugar: Google meet (Reunión quincenal Evimed - Equipo desarrollo).		
Duración: 1 hora.		
Se tratarán los siguientes temas:		
<ul style="list-style-type: none"> • Nombrar el top 3 de CRMs de acuerdo a los criterios establecidos. • Conocer el CRM analizado por el cliente y, de acuerdo a ello, ejecutar 1 de los 3 posibles planes que tiene el equipo de desarrollo. • Plantear tema del alcance. Surgió el tema con el revisor y también con el tutor, se quiere realizar una priorización de los servicios para comenzar por los más importante. 		
Aviso previo: Se comunicó la agenda el día 22 de febrero, dado que la reunión ya era recurrente.		
Tema	Preparación	Proceso Propuesto
Introducción Tiempo: 10 min Propósito: hablar con el cliente sobre lo que hemos trabajado y lo que vamos a mostrar en la presentación Responsable: Juan Andrés	Tener pronto el documento de análisis para mostrar al cliente	Luego de los saludos, compartir pantalla mostrando el documento y hablar sobre el resumen y el proceso que tomamos para la investigación
Mostrar CRMs evaluados Tiempo: 5 min Propósito: repasar con el equipo todos los CRMs evaluados Responsable: Juan Andrés	Avanzar a la Matriz de Pugh	Repasar cada CRM brevemente para mostrar las debilidades o fortalezas a grandes rasgos
Mostrar Top 3 Tiempo: 15 min Propósito: repasar específicamente el top 3 Responsable: Juan Andrés	-	Repasar cada uno de los CRMs del top 3 y expresar nuestras conclusiones

Conocer el CRM evaluado por el cliente Tiempo: 10 min Propósito: consultar al cliente cuál era el CRM que ellos evaluaron Responsable: Juan Andrés	-	Consultar al cliente
Alcance Tiempo: 20 min Propósito: plantear la priorización de servicios a implantar Responsable: Juan Andrés	Tabla con los servicios a implantar	Realizar en conjunto un análisis de priorización de servicios a implantar

Tabla 2: Agenda efectiva

El equipo se planteó tres escenarios posibles de conocer el CRM evaluado por el cliente:

- **El CRM del cliente está incluido en el top 3:** este sería el escenario ideal, ya que validaría lo realizado por el equipo y agilizaría la toma de decisiones y licencias (en caso de ser necesario).
- **El CRM del cliente está incluido en la investigación pero no en el top 3:** en este caso, el equipo volvería a evaluar el CRM para entender por qué fue descartado y plantear al cliente la sugerencia, de si sigue no siendo parte del top, o si puede ser una opción más de solución.
- **El CRM del cliente no fue investigado:** en este caso, el equipo investigaría el CRM planteado y lo sumaría al análisis y Matriz de Pugh, evaluando si formaría parte del top 3 o si siguen firmes con su sugerencia.

En esa reunión, surgieron dos puntos a resaltar:

- Se descartó Zoho CRM, dado a que la interfaz era anticuada y poco intuitiva.
- El CRM que el cliente investigó era uno del top 3 identificado por el equipo, por lo que se ubicaba en el primer escenario.

El primer punto marcó una diferencia en el análisis, teniendo que hacer una segunda versión del documento, generando una nueva sección: Odoo vs HubSpot.

El segundo punto, se tomó como una validación de lo realizado por el equipo, dado que del mundo enorme de CRMs que existen, el cliente investigara y eligiera el mismo CRM que el equipo era uno de los escenarios menos probables.

Luego de un par de iteraciones que se detallarán en la siguiente subsección, se llegó al documento final el 20 de marzo, resultando como CRM elegido, Odoo.

4.4.2. Validación y aceptación de resultados

Como se mencionaba anteriormente, el informe de análisis tuvo cuatro iteraciones.

Una primera entrega fue el 23 de febrero, donde se mostró al cliente la Matriz de Pugh con los CRMs evaluados y se pidió el CRM analizado por ellos.

La segunda fue el 8 de marzo, que como surgió en la reunión del 23 de febrero, pidieron generar una tabla de comparación entre Odoo y HubSpot. En la reunión asociada a esa entrega se repasó el entregable y se vio que había un problema con las licencias de Odoo y de HubSpot. Se estaban comparando cantidades y funcionalidades diferentes, cosa que el cliente no estaba de acuerdo, entonces pidió que se generara un nuevo presupuesto evaluando las mismas características de ambos lados. Esto hizo que el equipo se reuniera nuevamente con el equipo de ventas de Odoo y HubSpot y repasaran los datos y presupuestos existentes.

La tercera iteración fue el 14 de marzo, donde se entregó un documento actualizado con los datos requeridos. En la siguiente reunión, el cliente ya se había decidido que el CRM elegido sería Odoo, pero le preocupaba el presupuesto total anual. El presupuesto presentado en este punto era de US\$1406,40, el cual incluía:

- Cuatro licencias: tres para el equipo de desarrollo y una para el equipo de Evimed
- 8GB de almacenamiento (Odoo.sh): el equipo de ventas recomendaba 2GB por persona para el correcto funcionamiento del almacenamiento
- 1 worker (Odoo.sh): manejador de request del *backend*. El equipo de ventas recomendaba 1 cada 25 personas que utilizaran Odoo.
- 1 ambiente de prueba (Odoo.sh): ayudaría al equipo a tener el entorno de producción con una versión estable, pudiendo realizar pruebas en un ambiente aparte.

El cliente dijo que repensaría la idea de la integración con WhatsApp, que era lo que implicaba pagar la licencia de Odoo.sh, y así bajar fuertemente el presupuesto. Le pidió al equipo volver

a tener la reunión con el equipo de ventas de Odoo y así abaratar los costos de las licencias, resultando en eliminar Odoo.sh y la licencia para Evimed.

La última iteración respecto al informe y presupuesto fue el 20 de marzo, donde lo que se incluían eran las tres licencias para el equipo de desarrollo.

El cliente comentó que Evimed era una empresa amiga de Quanam, empresa importante en Uruguay que utiliza Odoo, y querían realizar una reunión con ellos para que cuenten su experiencia y de alguna forma validar lo investigado. El equipo accedió a la reunión y el cliente quedó en contactarse para agendarla.

La reunión con Quanam fue recién el 3 de abril. Pasaron dos semanas desde que se había tomado la decisión de utilizar Odoo. Para ir avanzando, pese a que el equipo sabía que el cliente quería esperar a la reunión para obtener las licencias, comenzaron a pedir datos y requerimientos más específicos para ir pensando el esquema de implantación en el CRM. Se le asignó la tarea a la representante de marketing de Evimed, quien sería la encargada de otorgar los cinco flujos más importantes que ella realiza en su día a día, para mapear entidades importantes del sistema y así, ir pidiendo al equipo de sistemas los datos de esas entidades. Estos flujos nunca llegaron.

El equipo insistió repetidamente desde el 20 de marzo en obtener información, flujos, requerimientos, algo por lo cual poder empezar, pero nada de eso llegó hasta que el mismo equipo decidió crear un diagrama de cómo se imaginaban las entidades con la poca información que tenían, para poder ayudar al cliente a transmitir la información. Este diagrama se entregó el 28 de abril. Fue una decisión tardía, que dejó su aprendizaje que será detallado más adelante en la sección correspondiente, pero el equipo tomó un enfoque proactivo y eso activó al cliente a entregar la información pedida.

Finalmente, se termina la primera etapa de consultoría, con una validación de Quanam, principal cliente de Odoo en Uruguay, las tres licencias para acceder a Odoo y un diagrama de entidad de relaciones del sistema de Evimed.

5. Segunda etapa: Implantación de Odoo CRM

5.1. Introducción

En esta segunda etapa se encuentra la Implantación de Odoo CRM. Como se mencionó en la sección anterior, el cliente se decidió por la herramienta Odoo para adaptar su sistema y dejar una única base de datos para segmentar clientes. Para este proceso, fue necesario relevar las entidades del sistema de Evimed, así como qué clase de segmentaciones se querían realizar para poder generar una configuración personalizada para el cliente.

Cabe destacar que de los tres productos entregados, este es el requerido desde un inicio por el cliente, por lo que el entregable de este producto era el más importante para el equipo.

5.2. Ingeniería de requerimientos

5.2.1. Objetivos

Como objetivos de la ingeniería de requerimientos se plantearon:

- Conocer el contexto del negocio para entender cómo representar mejor los datos.
- Conocer el modelo actual de los datos.

5.2.2. Descripción general

Para cumplir los objetivos, se necesitaba poder relevar específicamente los modelos de datos del sistema para evaluar cómo podrían quedar mejor representados.

Además de los modelos, se necesitaba un enfoque que agilice la segmentación al equipo de marketing, pudiendo obtener todos los datos en un único lugar sin necesidad de realizar segmentaciones partidas en distintas tablas para obtener sus objetivos.

5.2.3. Proceso de relevamiento

Para relevar los requerimientos del sistema se tuvieron distintas instancias hasta obtener un resultado que permitiera avanzar en el desarrollo del modelo.

El equipo de desarrollo comunicó al cliente la necesidad de tener al menos cinco flujos del sistema, con el fin de poder entender cómo era el trabajo del equipo de marketing, más allá de

los requerimientos previos que debía cumplir la herramienta. Estos flujos fueron pedidos en la reunión en la que se entregó el informe de análisis de CRMs, en la que participaba Sofía, representante del equipo de marketing.

Esta tarea quedó asignada a Sofía, quien explicaría qué segmentaciones realizaban con más frecuencia y así poder identificar un modelo de datos que pueda ayudarla.

Se tuvo un imprevisto del lado del cliente, y estos flujos no pudieron presentarse, por lo que el equipo decidió ir directamente con el equipo de IT para obtener el modelado de datos actual de sus sistemas.

Tampoco se pudieron obtener resultados del equipo de IT, por lo que se decidió realizar un diagrama y un documento del lado del equipo de estudiantes, con el fin de que el cliente sea capaz de validar si ese esqueleto inicial era el correcto y así poder avanzar con el modelado de datos. Se envió el diagrama y el documento, y se validaron en una reunión en la que no estaba el representante principal de la empresa, Antonio, quien había quedado a cargo de otorgar la información necesaria.

En esa reunión, se pidió que se desestimara el documento que el equipo había generado, dado a que la realidad de la empresa era diferente a la planteada. Luego, Antonio agendó una nueva reunión dos días posterior a esa, en la que entregó el modelo de entidad de relación (MER) que se imaginaba para representar en Odo.

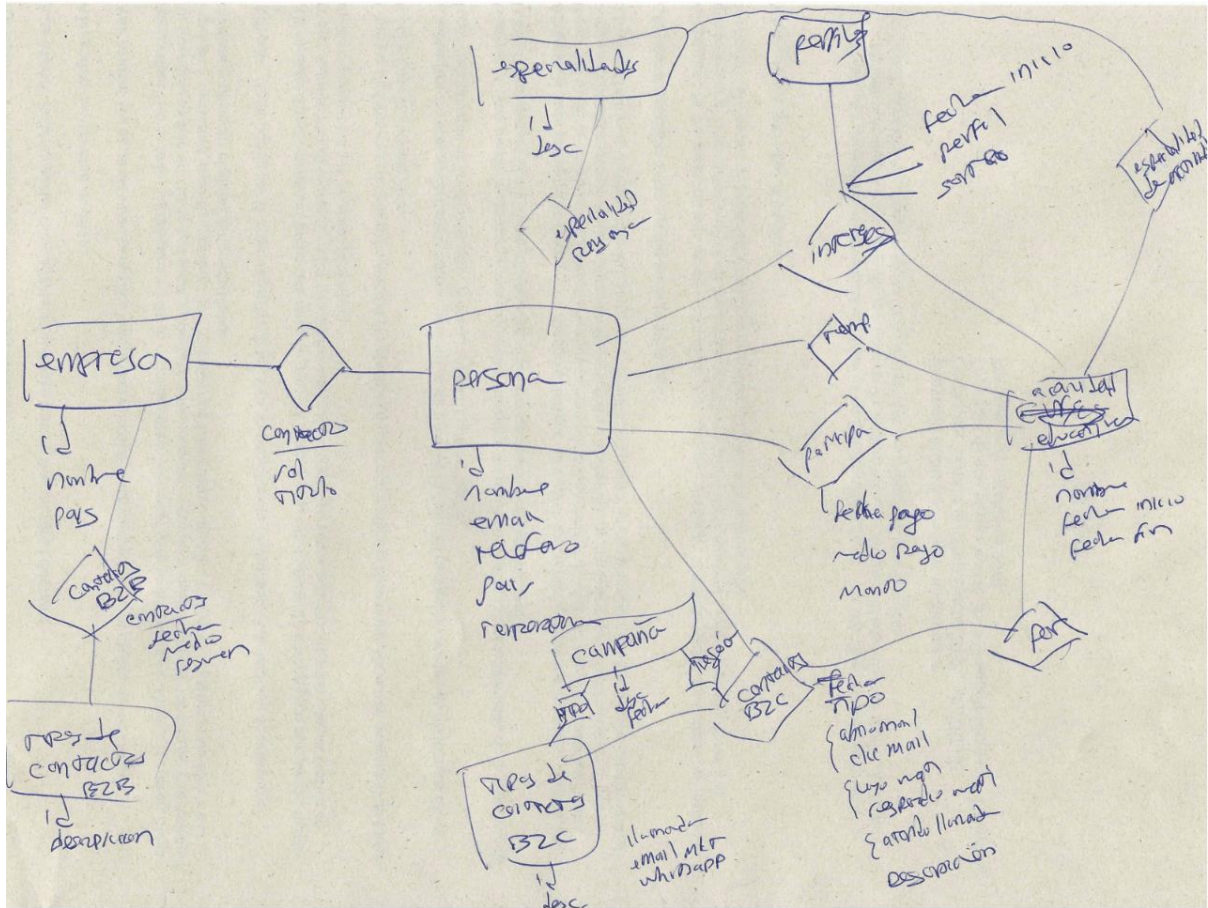


Figura 12: Modelo de entidad de relación de Evimed

5.2.4. Resultados

Gracias al MER que se muestra en la Figura 11, el equipo pudo comenzar con el modelado de datos en Odoo. Allí se representaron todas las entidades que eran necesarias tener en el sistema.

Se realizaron algunas iteraciones para refinar el modelado, sobre todo a medida que se fue modelando directamente en Odoo, pero el modelo es básicamente lo que se representó.

5.2.5. Reflexiones del proceso

Si bien se llegó al objetivo de poder representar de la mejor forma el modelo de datos, el equipo entiende que podría haber sido más proactivo a la hora de relevar los requerimientos. Se llegó a una instancia crítica, con demoras importantes, por la inexperiencia de manejar un cliente y fechas límites.

La proactividad del equipo en realizar el documento y diagrama que el cliente marcó como equivocados, fue el hincapié a que ellos realmente pensarán y otorgaran la información correcta.

También, el equipo se confió de cómo venía siendo la relación con el cliente en cuanto a la respuesta de sus consultas en la etapa de investigación, que al ser tan buenas, no creyó necesario una práctica tan estructurada para esta relevación. Allí también se encontró un punto de mejoría. Se podrían haber implementado técnicas más específicas, como entrevistas estructuradas, un prototipo mejor y más tempranamente definido, en vez de sólo empatizar con el cliente.

5.3. Requerimientos funcionales

Épica EV-1: Configuración Odoo

[EV-1.RF-1]: Configuración de cliente.

Como representante de marketing,

Quiero crear, modificar o eliminar un cliente, con sus datos correspondientes,

Para mantener nuestro sistema actualizado.

[EV-1.RF-2]: Configuración de empresa.

Como representante de marketing,

Quiero crear, modificar o eliminar una empresa, con sus datos correspondientes,

Para mantener nuestro sistema actualizado.

[EV-1.RF-3]: Configuración de actividad educativa.

Como representante de marketing,

Quiero crear, modificar o eliminar una actividad educativa, con sus datos correspondientes,

Para mantener nuestro sistema actualizado.

[EV-1.RF-4]: Configuración de interés.

Como representante de marketing,

Quiero crear, modificar o eliminar un interés, con sus datos correspondientes,

Para mantener nuestro sistema actualizado.

[EV-1.RF-5]: Configuración de contacto con clientes.

Como representante de marketing,

Quiero crear, modificar o eliminar un contacto con un cliente, con sus datos correspondientes,

Para mantener nuestro sistema actualizado.

[EV-1.RF-6]: Configuración de contacto con empresa.

Como representante de marketing,

Quiero crear, modificar o eliminar un contacto con una empresa, con sus datos correspondientes,

Para mantener nuestro sistema actualizado.

[EV-1.RF-7]: Configuración de campaña.

Como representante de marketing,

Quiero crear, modificar o eliminar una campaña, con sus datos correspondientes,

Para mantener nuestro sistema actualizado.

[EV-1.RF-8]: Configuración de filtros.

Como representante de marketing,

Quiero filtrar los clientes,

Para realizar una segmentación y realizar email marketing.

[EV-1.RF-9]: Exportación de datos.

Como representante de marketing,

Quiero disponer de templates de exportación de datos,

Para exportar fácilmente nuestros datos.

5.4. Requerimientos no funcionales

EV-1.RNF-1 - [Usabilidad] Tener accesibles las entidades del sistema para facilitar la vista.

EV-1.RNF-2 - [Usabilidad] Dentro de un cliente, poder visualizar sus inscripciones, intereses y empresas relacionadas.

5.5. Restricciones

EV-1.RES-1 - El sistema debe mantener la unicidad de los clientes, tomando como clave única el correo electrónico.

5.6. Modelado de datos

Para el modelado de datos, se usó como referencia el modelo de la Figura 12.

El equipo evaluó cómo se representaban actualmente cada entidad ya definida de Odoo, e hizo los diagramas que acompañarían a la verdadera implementación en Odoo.

Se detectaron cuatro flujos base. El primero, sería las entidades primarias del sistema de Evimed. Aquí se tienen a los clientes, tanto B2C (*client*) como B2B (*company*), y a los cursos que dictan (*education_activity*).

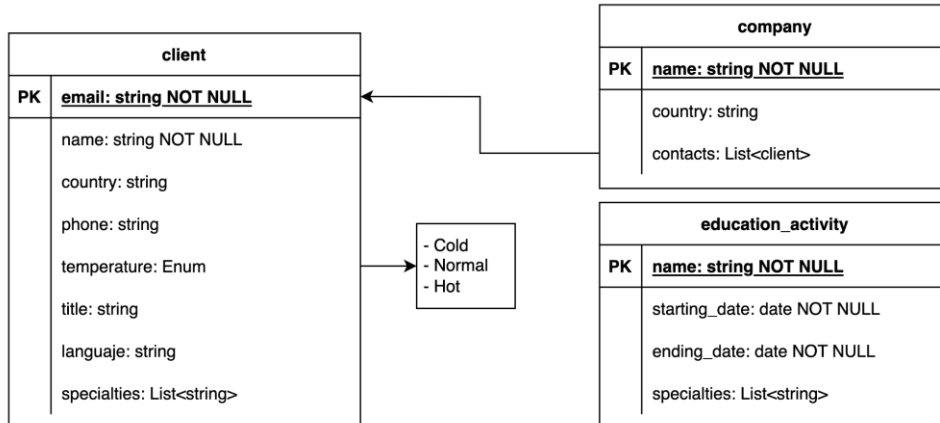


Figura 13: Entidades primarias del sistema

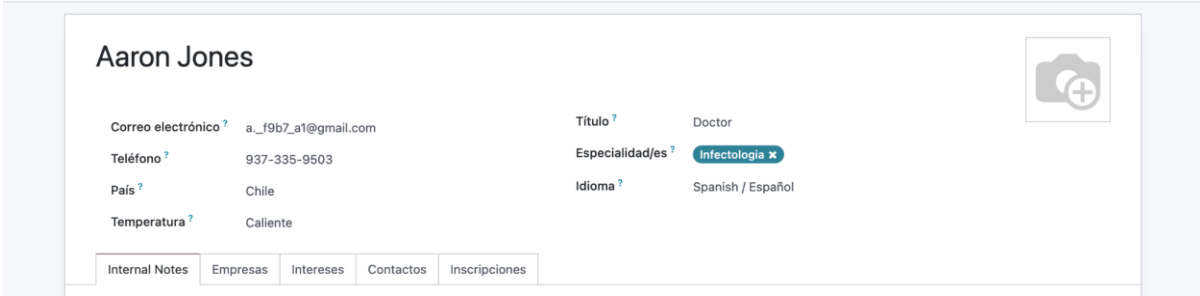


Figura 14: Ejemplo de la representación de un cliente en Odoo

El segundo flujo, serían las interacciones que tienen esos clientes con los cursos, estos serían los intereses (*interest*) o las inscripciones (*inscription*). Lo que diferencia un interés de una inscripción es que en el interés se marca cuando un cliente llena un formulario de inscripción y por algún motivo no lo completa, lo que hace marcar como un posible interesado para luego dar un input al equipo de marketing para enviar un mail con una oferta o información.

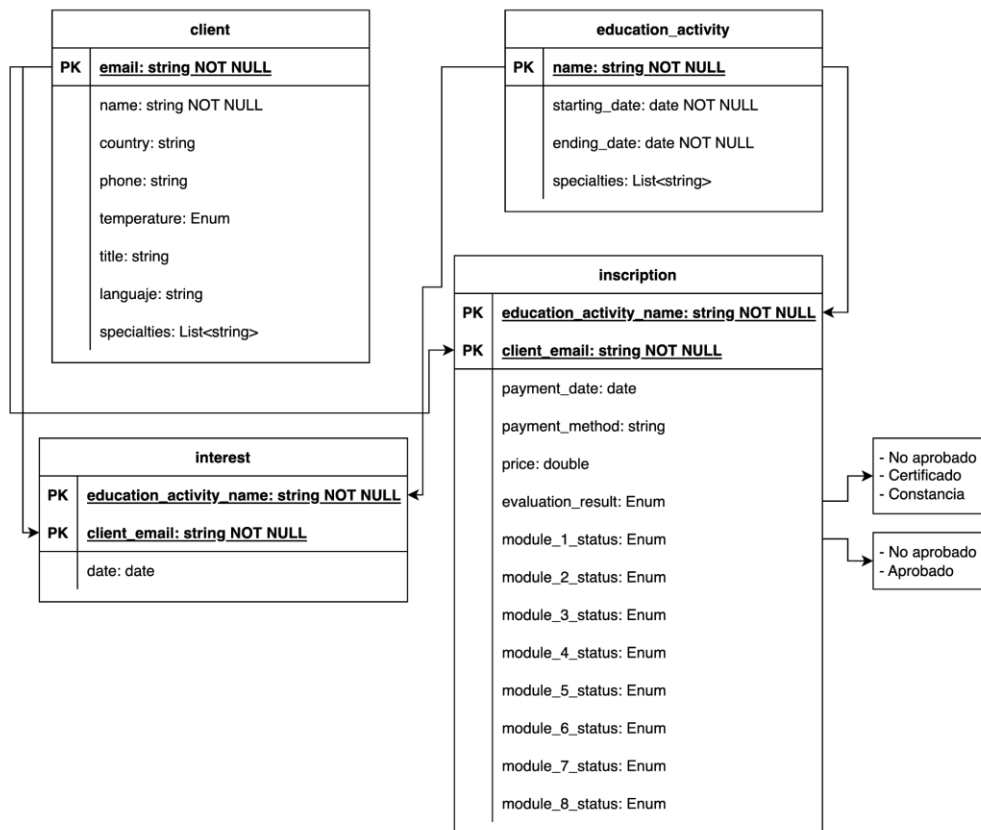


Figura 15: Flujo de interacciones del cliente

Id ?	Ai_1a63_a@hotmail.com_interpretacao-do-antibiograma ES	Resultado de evaluación ?	Constancia
Cliente ?	Timothy Bishop	Certificado ?	
Actividad Educativa ?	interpretacao-do-antibiograma	Precio ?	115,00
Fecha de pago ?	12/05/2023		
Medio de pago ?	<input type="radio"/> pago_local ✕		
Módulo 1 ?	Aprobado	Módulo 5 ?	No aprobado
Módulo 2 ?	Aprobado	Módulo 6 ?	No aprobado
Módulo 3 ?	No aprobado	Módulo 7 ?	No aprobado
Módulo 4 ?	No aprobado	Módulo 8 ?	No aprobado

Figura 16: Ejemplo de la representación de una inscripción en Odoo

El tercer flujo, serían los contactos individuales que el equipo de Evimed realiza con cada uno de sus clientes. Estos contactos pueden ser tanto con clientes persona como con empresas, mediante un contacto específico.

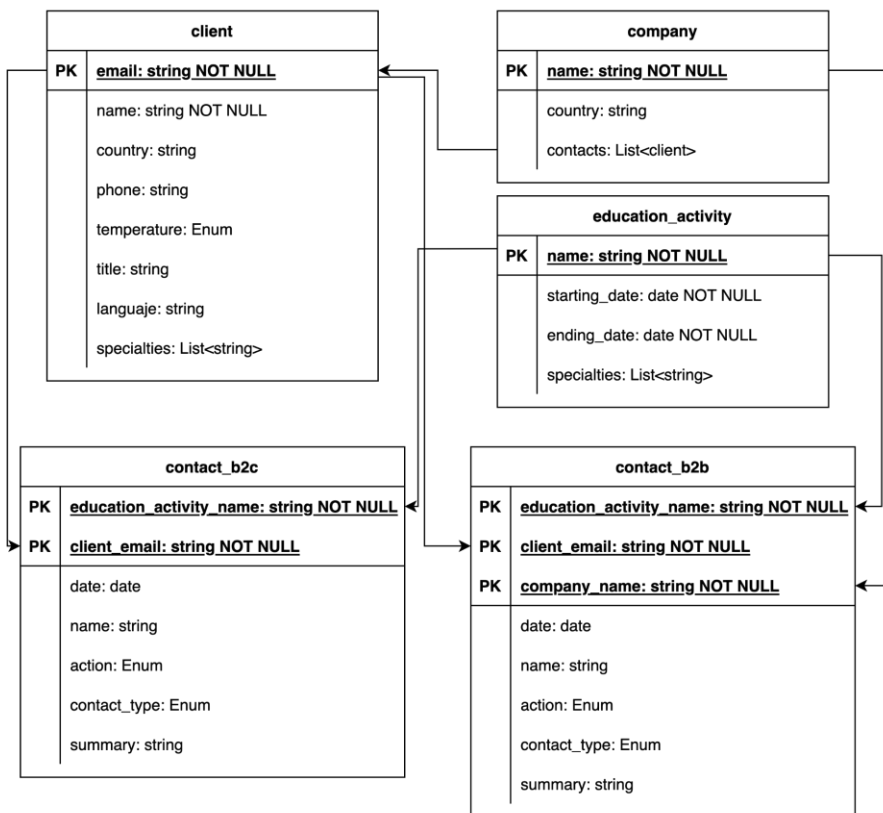


Figura 17: Flujo de interacciones de equipo Evimed con los clientes

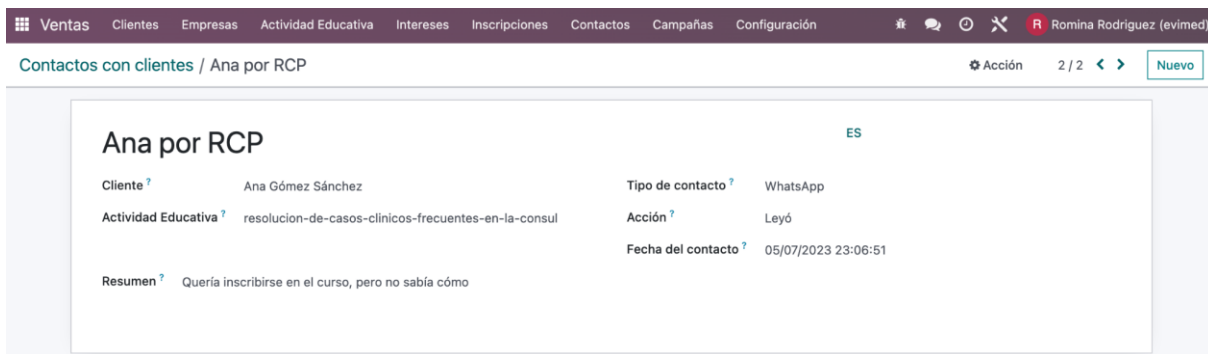


Figura 18: Ejemplo de la representación de un contacto con un cliente en Odoo

Y finalmente, se identificó un último flujo, que depende de estos contactos con los clientes. Estas son las campañas de marketing, en el que se agrupan unos contactos específicos que son referentes a las campañas de marketing de Evimed. Es una manera de identificar rápidamente qué contactos pertenecieron a qué movimiento específico de Evimed.

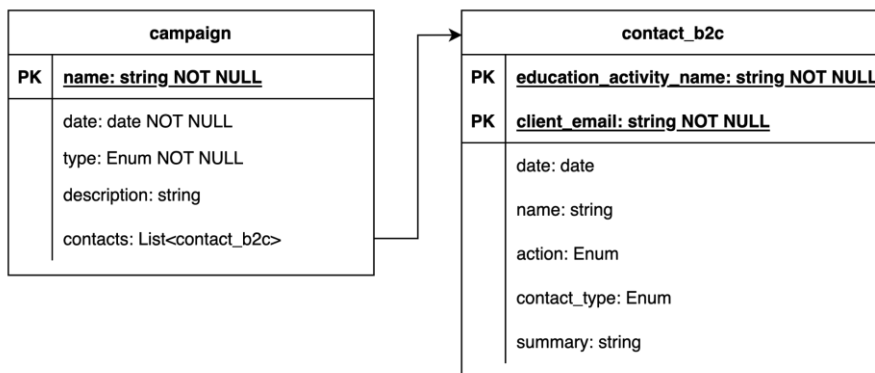


Figura 19: Flujo de las campañas

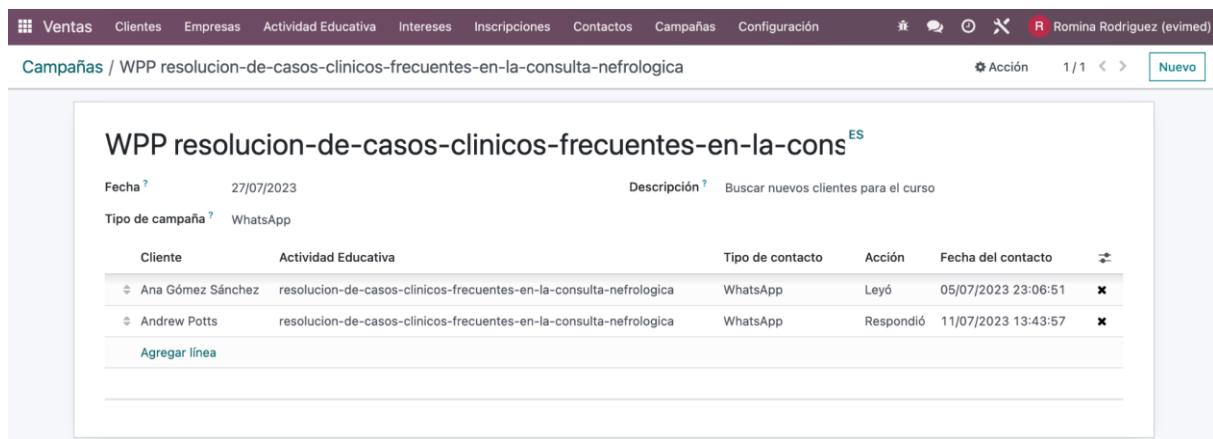


Figura 20: Ejemplo de la representación de una campaña en Odoo

Este modelado fue clave para poder vincular todo con el cliente, de modo que el equipo de marketing pueda segmentar específicamente desde la vista del cliente. Para esto el equipo realizó una serie de filtros personalizados que ayudan al equipo de marketing a realizar sus tareas de segmentación más fácilmente.

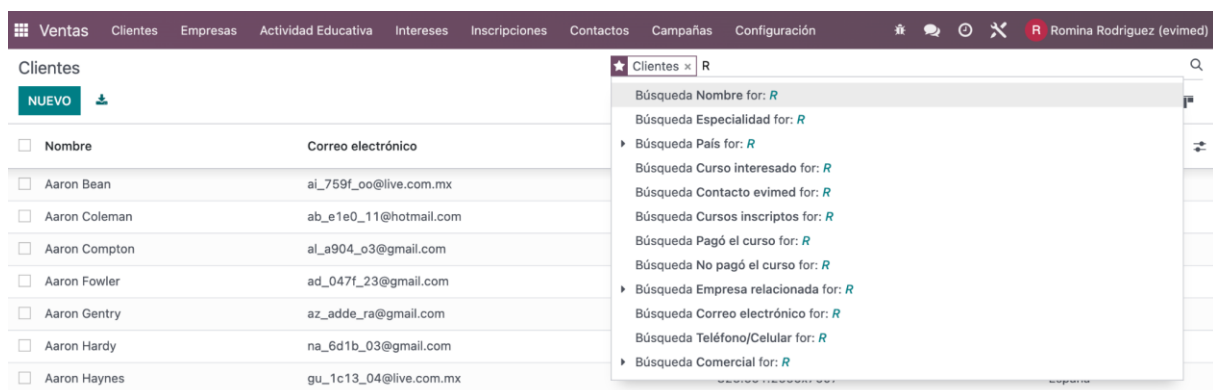


Figura 21: Ejemplo de la representación de una campaña en Odoo

Estos filtros fueron los especificados por el cliente para facilitar sus búsquedas, pero dado el modelado, es muy fácil agregar un filtro nuevo si así se desea.

5.7. Validación y aceptación de resultados

El equipo de desarrollo y el cliente plantearon desde el comienzo una reunión quincenal, en la que se utilizaba el tiempo para lo que fuera necesario. Cuando se comenzó la etapa de implantación y representación de los modelos, el equipo aprovechó esas instancias para mostrar los avances que habían tenido y dudas que le hayan surgido.

Los datos con los que se validó el sistema tienen parte real del cliente, y parte generada por el equipo de desarrollo, dado que el cliente no tuvo el tiempo de pasar datos de prueba. De cualquier forma, los datos generados fueron avalados por el cliente en las demostraciones, cambiando algunos valores según los feedbacks que se iban recibiendo.

En la última iteración en la que se trabajó con Odoo, se hizo una última demo con los flujos que el cliente había requerido como necesarios para evaluar si el sistema soportaba lo que marketing necesitaba.

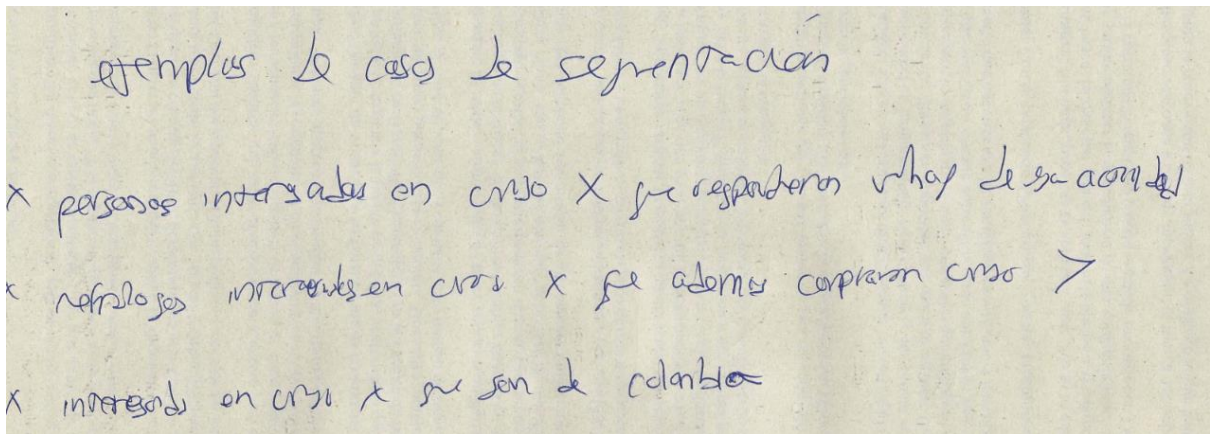


Figura 22: Ejemplos de casos de segmentación

También se validó que se pudiera enviar un email mediante Odoo por un servidor propio.

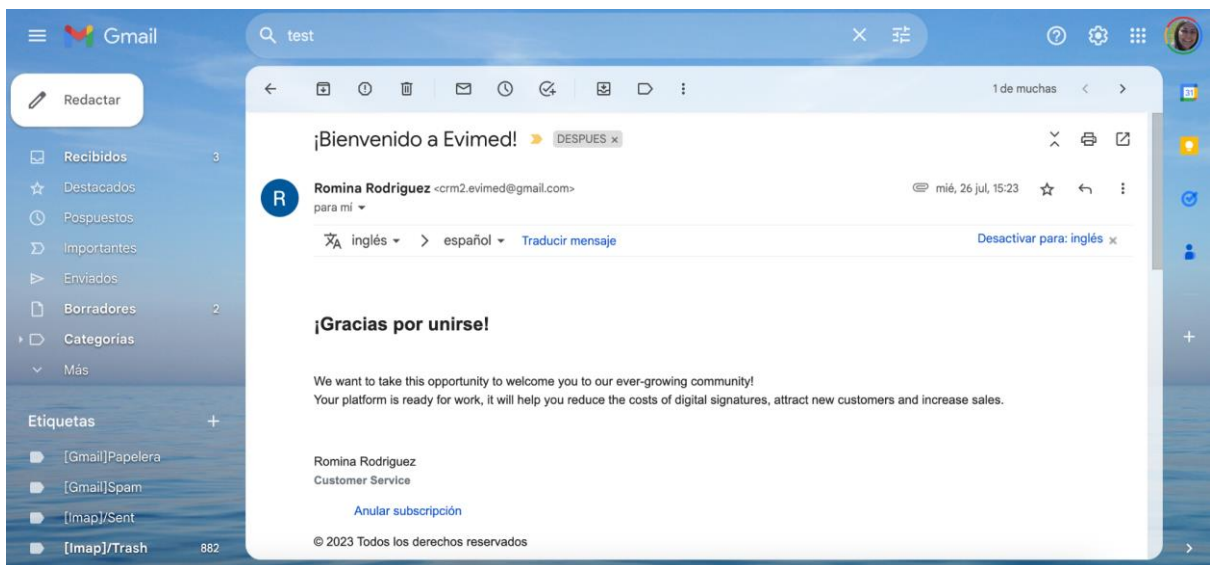


Figura 23: Ejemplo de envío correo electrónico

En esta última demo, el cliente exclamó que el trabajo que realizó el equipo fue muy bueno. Ellos esperaban un MVP que les permitiera migrar su trabajo y luego ir mejorándolo, pero que el trabajo que el equipo realizó, **superaba sus expectativas**.

5.8. Reflexiones del proceso

Como se mencionó en la sección [5.2.5. Reflexiones del proceso de ingeniería de requerimientos](#), el equipo reconoce que para esta etapa de implantación le faltó proactividad al presentarse problemas.

El equipo estaba a la espera de respuesta del cliente, pero dado a la temporada en la que se encontraban, era difícil obtener la información requerida. Esto sirvió como un enorme aprendizaje para el equipo. En el día a día de trabajo, uno no puede estar dependiendo 100% de otra parte y atarse a ello. Se debe buscar las maneras de poder desbloquearse y seguir adelante, o buscar una alternativa para lograr los objetivos.

Se cree que en cierto punto, lo sucedido ayudó a crecer al equipo, porque los motivó a pensar en alternativas, en poder pivotar de ser necesario para no perder de vista el objetivo académico que es convertirse finalmente en un profesional.

En cuanto a la herramienta, sin dudas fue un gran aprendizaje. El equipo comenzó el proyecto sin saber claramente lo que era un CRM, y terminó amoldando una herramienta a un negocio que poco conocía, agregando valor a la empresa y superando las expectativas del cliente según muestra su carta de conformidad (Anexo [10.4 Carta de conformidad del cliente](#)).

6. Tercera etapa: Desarrollo de productos satélites

6.1. Introducción

Debido a la reducción de requerimientos mencionada en la [sección 3.1.2. Alcance final](#), surge la necesidad de generar nuevas funcionalidades para complementar el CRM implantado. Esta inquietud surgió inicialmente en el equipo al ir avanzando en la implantación y configuración de Odoo. Fue en la segunda revisión de la tesis donde la revisora levantó formalmente el riesgo de que este alcance reducido podría no ser suficiente académicamente y se debía buscar cómo agregar más valor al cliente.

Esta tercera etapa es diferente a las dos anteriores, dado que en ellas se partió de objetivos definidos y problemas a solucionar. En esta se necesitó dar un paso hacia atrás, volver a analizar el negocio y los problemas del cliente desde su perspectiva para encontrar nuevas formas de otorgarle valor. Esto implicó un cambio de metodología, procesos y más importante, se tuvo que adoptar un enfoque más creativo.

El mayor desafío a afrontar en esta tercera etapa fue el tiempo, ya que habían transcurrido dos tercios del tiempo estipulado del proyecto. Esto implicó la necesidad de realizar todo el descubrimiento de las nuevas funcionalidades, diseñar la solución a medida y desarrollar un producto funcional y entregable, en el menor tiempo posible.

6.2. Ingeniería de requerimientos

Los dos productos satélites resultantes, surgieron del mismo proceso de ingeniería de requerimientos sobre el que se va a profundizar a continuación.

6.2.1. Objetivos

Las funcionalidades que había que crear tenían que cumplir los siguientes dos objetivos:

1. Otorgarle más valor a Evimed. Para ello, era necesario que estas funcionalidades no quedaran alineadas a la implantación ya realizada, de forma de respetar la centralización que buscaba Evimed con Odoo.

2. Estas nuevas funcionalidades deberían implicar un desafío técnico, para complementar la parte del proyecto ya realizada y dar al equipo la oportunidad de conocer y aprender a trabajar con nuevas tecnologías.

6.2.2. Descripción general del proceso

Para cumplir con estos dos objetivos simultáneamente, surge la pregunta: ¿Cómo se puede aportar más valor a Evimed? Se identificaron tres diferentes pilares con los que trabajar:

1. Aumentar la información que se obtiene de los datos ya existentes.
2. Automatizar procesos regulares de la empresa.
3. Aumentar la cantidad de posibles clientes alcanzados.

Una vez identificados estos pilares, se utilizaron diferentes metodologías para fomentar el pensamiento creativo, tomando cada uno de estos individualmente y pensando qué se podría desarrollar para otorgar valor de dicha forma. Al obtener una lista de ideas tentativas, se iteró sobre cada una de ellas, evaluando su viabilidad basado en el tiempo y complejidad de desarrollo y aporte de valor al cliente tanto bajo nuestro criterio, y luego validándolas con Evimed.

6.2.3. Metodologías utilizadas

Para el relevamiento de las nuevas funcionalidades se aplicaron diferentes etapas del proceso de *Design Thinking* [5], el cual es un proceso de innovación, centrado en encontrar ideas y soluciones creativas para satisfacer necesidades existentes. Ha de tenerse en cuenta, que la cuarta etapa en el proceso ideal de *Design Thinking*, la de prototipación, no fue realizada por motivos que se profundizarán en la sección [6.2.3.4. Ausencia de la etapa de Prototipación](#). En las siguientes subsecciones, se explicarán las diferentes etapas de *Design Thinking* que utilizó el equipo y cómo fueron aplicadas.

6.2.3.1. Empatizar

Para otorgarle más valor a Evimed, es importante analizar los diferentes grupos de interesados especificados en la sección [2.4. Interesados y necesidades](#) y separarlos en el análisis, así logrando empatizar específicamente con cada uno de ellos y no simplemente con la empresa entera como un solo punto de referencia. El proyecto inicial, a pesar de otorgar valor al equipo

de IT mediante la facilitación del mantenimiento del sistema dada la centralización de todos sus datos en Odoo, estaba principalmente enfocado en el equipo de marketing, dado que lo que se buscaba era facilitar la segmentación y el envío de campañas de marketing.

Las necesidades del equipo de IT, están relacionadas al mantenimiento del sistema a largo plazo y por otro lado, a la transición a Odoo, buscando que sea lo más fluida posible. El equipo de marketing, con sus necesidades cubiertas por la implantación de Odoo, puede seguir beneficiándose de mejoras, simplificaciones o cualquier facilitación de sus flujos de trabajo diarios. Finalmente, la directiva de Evimed se ve beneficiada con cualquier producto agregado que aumente la productividad o ganancia de la empresa.

6.2.3.2. Definir

Para definir las áreas de mejora en las cuales trabajar, se comenzó volviendo al inicio del proyecto para analizar la problemática inicial explicada en la sección [2. Sobre el problema](#) que causó que Evimed presente el proyecto. La implantación de un CRM y migración de los sistemas actuales al mismo, surgió de que Evimed pasó de un modelo únicamente B2B a uno B2B y B2C. Esto incrementó la cantidad de clientes con los que trabajaba a una velocidad que la estructura de su sistema actual no podía soportar. A pesar de que el problema de la interacción de estos datos se soluciona mediante la centralización en Odoo, sigue existiendo la problemática de que el flujo de registro de nuevos clientes depende de sus sistemas actuales para ser ingresados en Odoo, que como fue recién mencionado, no funcionan bien con el volumen de datos que maneja Evimed actualmente. Luego de analizarlo, se obtuvo que una de las áreas de mejora principales en las que profundizar es el conjunto de flujos ubicados en el sistema actual de Evimed, encargados del alta, baja y modificación de los datos en Odoo.

Las otras dos principales motivaciones de Evimed para contratar un CRM eran la segmentación y el envío de campañas. ¿Se pueden mejorar estos dos flujos? La segmentación está totalmente cubierta con las customizaciones realizadas sobre Odoo, pero el envío de correos incluido en Odoo solo tiene capacidad de enviar correos y crear plantillas visuales, no trabaja con el contenido del correo en sí. De esto surge una oportunidad de mejora respecto a la generación del contenido de los correos.

Una vez analizados los problemas y motivaciones iniciales de Evimed, hay que encontrar áreas desconocidas donde mejorar, lo cual el equipo realizó consultando con ellos las diferentes áreas de la empresa, y adentrándose en cada una de ellas, buscando oportunidades de mejora. La que

más llamó la atención fue la del *feedback* de los cursos por parte de los alumnos mediante encuestas, ya que poseían mucha información recopilada de cada curso que no estaba siendo aprovechada ya que no se estaba analizando regularmente.

6.2.3.3. Idear

Con el objetivo de obtener ideas basadas en estas áreas de mejora obtenidas en la parte anterior, se utilizó *Brainstorming* y *Brainwriting*, actividades que fomentan el pensamiento creativo en equipo con el objetivo de materializar ideas nuevas.

ROMI	Recomendación de cursos a través de entrenamiento de deep learning	Asistencia de usuarios en tiempo real con sugerencias	Integrar chatGPT a la plataforma de e-learning para que responda como un chat bot a los usuarios	Que las recomendaciones salgan de un plugin, con el filtro armado para poder hacer el envío de mails	Estimar probabilidad de éxito de un contacto mediante comunicaciones e interacciones pasadas	Poder sacar métricas para evaluar los contenidos de los primeros dos módulos gratuitos de cuáles llamaron la atención y trajeron más inscriptos y cuáles no
MARCOS	Automatización de flujo de campañas de marketing	Pedirle a chatGPT que realice un template de email para enviar a clientes	Poder integrar odoo para la gestión de los mails de chatGPT para mantener el registro en la plataforma	Analizar posibles candidatos en redes sociales para marketing de influencers mediante ai	Entrenar un modelo de IA para segmentar clientes, medir el porcentaje en el que ese cliente compra el curso	Analizar qué curso es más llamativo para hacerle publicidad en redes sociales
NERVI	Integrar chatGPT a Odoo	Usarlo para redactar emails y whatsapps con el formato que precisan	Utilizar datos de cursantes para personalizar la comunicación	Con los datos de cursantes, realizar un análisis de sentimiento para recomendar futuros cursos	Que chatGPT te de los filtros para utilizar en Odoo para hacer el análisis de seguimiento (cuántas veces entró o no y por qué, qué módulo va completando, etc)	Poder consultar la probabilidad de desercion de un estudiante y en caso de ser critica para cualquiera, generar una alerta automáticamente.

Figura 24: *Brainwriting*

Para complementar estas actividades, el equipo decidió realizar una investigación sobre aquellas funcionalidades que Odoo CRM no tiene disponible y que sí lo presentan CRMs evaluados anteriormente por el equipo, como fue el caso de Hubspot CRM y Salesforce CRM. En ellos se notó que existe la posibilidad de, por medio de la inteligencia artificial, realizar diferentes acciones que buscan asistir al usuario en diferentes aspectos. Estas acciones van desde la escritura de un simple mail hasta analizar grandes volúmenes de datos provenientes de otros sistemas como ERPs, *data warehouses* y archivos de logs para generar poderosos *insights*. El análisis de grandes volúmenes de datos es gracias a la funcionalidad de Salesforce llamada Einstein⁸.

⁸ <https://www.salesforce.com/products/einstein/overview/>

El equipo identificó gracias a las actividades realizadas, que con la inteligencia artificial aparecía una oportunidad para explotar innovadora, lo cual se apega fuertemente al segundo objetivo definido en la sección [6.2.1. Objetivos](#). Para ello se aplicó ingeniería inversa [6], que es un proceso que implica ir hacia atrás en el proceso de construcción, se parte del resultado final y se observa cómo está construido un objeto, programa o sistema, ver cómo funciona, con la intención de hacer mejoras o duplicarlo, en este caso. Este proceso fue aplicado a la funcionalidad “Redactor de correos electrónicos con IA”⁹ de Hubspot CRM que utiliza inteligencia artificial para poder generar mails creativos para ser usados en campañas de marketing o simplemente generar seguimiento a los clientes. Este proceso será explicado más detalladamente en la sección [6.3.2. Extensión de Google Chrome](#).

Generar contenido ✕

Tipo de correo electrónico *

Correo electrónico de introducción ▼

¿Qué vendes? *

Por ejemplo: software de ventas que incrementa la productividad de los representantes y ger

¿A quién le vendes? *

Por ejemplo: Mary Doe, directora de ventas de Acme, Inc.

Describe en pocas palabras qué deseas comunicar en este correo electrónico

Describe por qué tu producto es una excelente opción para el prospecto e incluye los beneficios. Menciona las conexiones personales o argumentos que desees.

Select the tone of your email

Profesional Útil Ingenioso Formal Optimista

Al utilizar esta función, reconoces y aceptas las [Políticas de uso](#) y la [Política de uso compartido y publicación](#) de OpenAI.HubSpot registrará y almacenará tus indicaciones, idioma generado y métricas de uso cuando uses el asistente de contenido con IA para mejorar el producto. Tus indicaciones serán atribuibles a ti y se compartirán con OpenAI con fines de moderación de contenido. No compartas información confidencial en tus indicaciones.

Generate

Figura 25: Redactor de correos electrónicos con IA - Hubspot CRM

⁹ <https://www.hubspot.es/products/marketing/ai-email-writer>

6.2.3.3.1. Ideas obtenidas

De estas actividades surge una primera lista tentativa de ideas:

1. Extensión de Chrome como motor para funcionalidades que interactúen con Odoo.
2. Escribir correos para marketing mediante ChatGPT.
3. Motor de recomendación de cursos para personas que ya cursaron algún curso de Evimed.
4. *Sentiment analysis* según encuestas de fin de curso.
5. Obtener métricas de los primeros dos módulos gratuitos del curso, para mejorar el ratio de permanencia del cursante.
6. Análisis de probabilidad de deserción dado un curso y un estudiante, generar alertas para comunicarse si un valor es crítico.
7. Personalización de la comunicación con los cursantes, utilizando datos respecto a su relación con el curso.
8. Utilizar ChatGPT para facilitar la redacción de mensajes de WhatsApp que correspondan a las restricciones de Wati.
9. Entrenar un modelo para categorizar a los usuarios relacionados con Evimed según su probabilidad de comprar un nuevo curso (cursos realizados anteriormente, comunicaciones pasadas, cursos en los que se interesó).
10. Automatización del flujo de generación de campañas de marketing.
11. Dado un mail de posible cliente, mediante ChatGPT categorizar, identificar curso, cantidad de integrantes y región, para generar un ticket de posible compra y que marketing se comunique.
12. Dado un conjunto de parámetros de filtros/segmentación, mediante ChatGPT realizar un template de email para enviar.
13. Integración con medios de pago.

14. Configuración de alerta de cursante inactivo: mediante el parámetro de entradas al campus y de módulos completados, obtener aquellos que se encuentran por debajo de cierto nivel de actividad fijado por el cliente, con la opción automática de envío de mail.
15. Configuración de alerta de cursante próximo a finalizar trial: al finalizar los dos primeros módulos satisfactoriamente y de no haber comprado el curso, enviar mail de captación y/o retención del cliente.
16. Motor de recomendación de cursos para clientes. Dado un curso nuevo poder correr esta funcionalidad para obtener los potenciales clientes, para los cuales podría enviarse un email y convertirse en una inscripción. El mismo se haría teniendo en cuenta para quienes está dirigido el nuevo curso, si el cliente ya ha cursado alguno similar, etc.

6.2.3.4. Ausencia de la etapa de Prototipación

Antes de profundizar en el proceso de validación realizado, es importante tener en cuenta que según la metodología de *Design Thinking*, se ha de prototipar luego de la etapa de ideación y usar esos prototipos como base de la validación. En este proyecto se dejó de lado esta etapa, alejándose del proceso ideal de *Design Thinking*, lo cual hizo aparecer el riesgo de que surjan cambios imprevistos en el momento de desarrollo por falta de validación por parte del cliente de las visuales del sistema. Este riesgo no fue materializado, pero fue tenido en cuenta por el equipo al tomar esta decisión.

El proceso fue modificado de esta forma por las siguientes razones:

- La lista de ideas relevadas era muy amplia y no se disponía con la cantidad de tiempo necesaria para realizar prototipos y que cada uno de estos sea validado por el cliente, dados los problemas de comunicación existentes, esto implicaba un riesgo muy importante que no era asumible.
- Una gran parte de las ideas no tienen una parte de UI, lo cual implica realizar un POC para prototipar el funcionamiento. Estas ideas también implican uso de tecnologías con las que ningún miembro del equipo estaba familiarizado, lo cual implicaba una inversión adicional de tiempo para poder realizar los POC necesarios para validar las ideas.

6.2.3.5. Validar

El proceso de validación consistió de tres etapas distintas.

- Primera etapa: Validación interna de listado de ideas tentativas según la viabilidad por tiempo del equipo.
- Segunda etapa: Validación interna de las ideas resultantes, adentrándose en cada una de ellas como equipo e identificando si cumplen los objetivos definidos en la sección [6.2.1. Objetivos](#).
- Tercera etapa: Reunión de validación con Evimed. Se realizó una primera explicación de las ideas obtenidas, luego se profundizó en las que no fueron descartadas por ellos, discutiendo la viabilidad, interés y valor aportado por las mismas.

6.2.4. Resultados de la elicitación

Se culminó el proceso de elicitación con los siguientes productos:

1. “Adapter”, que actúa como puente entre los sistemas actuales del cliente y Odo. Aporta valor al cliente mediante facilidad de transferencia de datos para la migración a corto plazo y aumento de posibles clientes alcanzados mediante la transferencia de datos de nuevos clientes al CRM en tiempo real. Desde el punto de vista de la arquitectura, aporta mejoras en la escalabilidad del sistema, evitando que se repita el principal problema inicial de Evimed. También aporta valor académicamente al equipo, dado que se utilizan diversas tecnologías de comunicación entre servicios.
2. “Odo CRM Marketing Assistant”, que aporta valor a Evimed mediante la introducción de la inteligencia artificial tanto para el análisis de datos ya existentes en su sistema como para la automatización en la generación de templates para el envío de email marketing. A su vez, introduce tanto al equipo de Evimed como al equipo a trabajar con inteligencia artificial, lo cual es altamente beneficioso para ambas partes, debido a la gran disrupción que está generando hoy en día y lo mucho que les permite innovar tanto como ingenieros o empresarios.

6.3. Productos satélites

En esta sección se profundizará sobre los requerimientos y estructura de los productos satélites resultantes del proceso de ingeniería de requerimientos de la sección [6.2. Ingeniería de requerimientos](#).

6.3.1. Adapter

El Adapter surge con el objetivo de cubrir las necesidades del equipo de IT de Evimed mencionadas en la [sección 6.2.3.1](#): la simplificación del mantenimiento del sistema a largo plazo y la fluidez de la transición de los datos de sus bases de datos actuales a Odoo.

Está compuesto por cuatro módulos diferentes sobre los cuales se profundizará en la [sección 6.3.1.4.1](#):

- Productores, que se encargan de enviar mensajes a la cola de mensajes con los datos de registro o actualización de una entidad.
- Consumidores, que reciben estos mensajes, los formatean y los derivan a la capa de comunicación con Odoo.
- Capa de comunicación, que se encarga de realizar las comunicaciones con los diferentes *endpoints* que Odoo expone para el ingreso de datos e interacción con sus datos ya almacenados.
- API, exponiendo un conjunto de *endpoints* para que el equipo de IT pueda realizar operaciones CRUD sobre los datos de Odoo sin necesidad de ingresar a la interfaz provista.

6.3.1.1. Requerimientos funcionales

Épica EV-2: Adapter

[EV-2.RF-1]: Envío de usuarios registrados a la cola de mensajes.

Como sistema de registro de usuario de Evimed,

Quiero enviar sus datos a una cola de mensajes,

Para que otros componentes del sistema puedan procesar los datos de manera asíncrona.

[EV-2.RF-2]: Consumición de mensajes de la cola de mensajes correspondiente.

Como consumidor de mensajes,

Quiero recibir datos de la cola de mensajes,

Para analizar y parsear la información contenida en los mensajes.

[EV-2.RF-3]: Creación de entidad desde el mensaje recibido de la cola de mensaje correspondiente.

Como ejecutor del Adapter,

Quiero que se cree la entidad del dominio de negocio correspondiente al mensaje consumido,

Para enviarlo a Odoo mediante su API.

[EV-2.RF-4]: Envío de entidades a Odoo mediante su API.

Como como ejecutor del Adapter,

Quiero que se envíen las entidades generadas a Odoo mediante su API,

Para que se ingresen los datos a Odoo y poder trabajar con ellos.

[EV-2.RF-5]: Modificación de entidades existentes en Odoo mediante su API.

Como como ejecutor del Adapter,

Quiero que se modifiquen las entidades existentes en Odoo mediante su API,

Para mantener la integridad de los datos en Odoo.

[EV-2.RF-6]: Eliminado de entidades existentes en Odoo mediante su API.

Como como ejecutor del Adapter,

Quiero que se eliminen entidades existentes en Odoo mediante su API,

Para mantener la integridad de los datos en Odoo.

[EV-2.RF-7]: Ingreso de entidades mediante CSV.

Como usuario de marketing de Evimed,

Quiero poder subir archivos con datos de una entidad existente en formato CSV a un *endpoint*,

Para facilitar el ingreso de cantidades masivas de clientes al sistema.

[EV-2.RF-8]: Ingreso de entidades mediante API.

Como usuario de Evimed,

Quiero ingresar nuevas entidades mediante un *endpoint*,

Para facilitar el ingreso de nuevas entidades a Odoo.

[EV-2.RF-9]: Modificación de entidad existente mediante API.

Como usuario del equipo de IT de Evimed,

Quiero modificar una entidad existente mediante un *endpoint*,

Para facilitar la modificación de los datos existentes en Odoo.

[EV-2.RF-10]: Eliminación de entidad mediante API.

Como usuario del equipo de IT de Evimed,

Quiero eliminar una entidad existentes mediante un *endpoint*,

Para facilitar la eliminación de los datos existentes en Odoo.

[EV-2.RF-11]: Obtención de todos los registros de un tipo de entidad mediante API.

Como usuario de Evimed,

Quiero obtener todos los registros de un tipo de entidad existentes mediante un *endpoint* en formato JSON,

Para facilitar el análisis sobre entidades existentes en Odoo.

[EV-2.RF-12]: Obtención una entidad mediante API.

Como usuario de Evimed,

Quiero obtener una entidad existente por su clave única mediante un *endpoint*,

Para facilitar el análisis sobre entidades existentes en Odoo.

6.3.1.2. Requerimientos no funcionales

EV-2.RNF-1 - [Disponibilidad] El cliente debe recibir una documentación de como instalar el Adapter, para así utilizarlo y mantenerlo a futuro.

EV-2.RNF-2 - [Deployabilidad] El cliente se encargará de desplegar el sistema en su licencia de AWS, por lo tanto, se debe entregar una documentación con toda la información necesaria para el correcto despliegue del sistema.

EV-2.RNF-3 - [Interoperabilidad] Los productores encargados de enviar los mensajes a las colas van a estar incorporados a los flujos existentes del sistema de Evimed, por lo que tienen que ser fácilmente integrables por el equipo de IT de Evimed.

6.3.1.3. Restricciones

EV-2.RES-1 - El sistema debe estar codificado en su totalidad en Python o PHP para posibilitar su correcto mantenimiento, esto es necesario debido a la inexperiencia del equipo de IT de Evimed en otros lenguajes de programación.

6.3.1.4. Arquitectura

Esta sección se centrará en el análisis de la arquitectura del Adapter, adentrándose en los diferentes módulos en los que está dividido el sistema y sus interacciones, tanto entre sí como con sistemas externos.

6.3.1.4.1. Descripción de la arquitectura

Representación primaria del Adapter

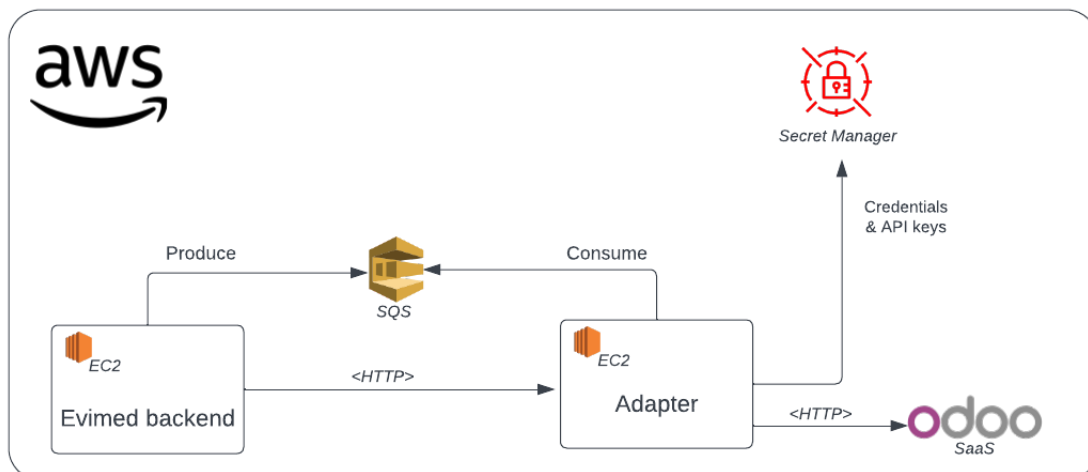


Figura 26: Diagrama de la arquitectura del Adapter

El sistema está compuesto por cuatro diferentes módulos: los consumidores, la API, la capa de comunicación con Odoo y los productores. Los tres primeros están encapsulados dentro del Adapter. Se habla de productores y consumidores en plural, ya que se dispone de uno de cada uno de ellos por cada entidad del dominio de negocio de Evimed y de una cola de mensajes de Amazon SQS para cada pareja de productor y consumidor.

Los diferentes productores no se encuentran con el resto del Adapter debido a que deben integrarse a los flujos de registro existentes de Evimed, para así introducir el procesamiento asíncrono mediante la delegación del envío de los datos hacia Odoo desde el Adapter. Esto implica la necesidad de que los productores tengan que existir junto a los sistemas actuales de Evimed. Como su equipo de IT no tenía conocimiento del uso de colas de mensajes y el equipo tampoco tenía acceso a su código, se decidió desarrollar los productores de forma genérica y reutilizable, de forma que puedan ser fácilmente integrables a sus sistemas actuales.

Los consumidores están constantemente escuchando la cola de mensajes correspondiente a su entidad, esperando la llegada de nuevos mensajes para construir la entidad correspondiente con el formato especificado por la API de Odoo¹⁰.

¹⁰ https://www.odoo.com/documentation/16.0/developer/reference/external_api.html

La capa de comunicación con Odoo tiene la responsabilidad de tomar estos conjuntos de datos transformados en entidades por el consumidor correspondiente y realizar la llamada HTTP al *endpoint* de la API de Odoo que corresponda.

Finalmente, el Adapter expone su propia API para el uso del personal de Evimed, con la posibilidad de realizar las mismas operaciones CRUD que se realizan desde la interfaz de Odoo por dicha API en formato JSON. A su vez, esta API también posee un endpoint destinado al ingreso de entidades en formato CSV, lo que facilita la carga masiva de los datos ya existentes en el sistema actual a Odoo, facilitando la transición.

Las diferentes credenciales que son necesarias para el funcionamiento de los sistemas externos, como API keys y credenciales, están almacenadas en *AWS Secret Manager*.

A continuación, se muestra la interacción entre los diferentes elementos del sistema tanto en el caso de la entrada de datos por la API como por la cola de mensajes. En ambos diagramas se usará como flujo de ejemplo el ingreso de un nuevo cliente a Odoo.

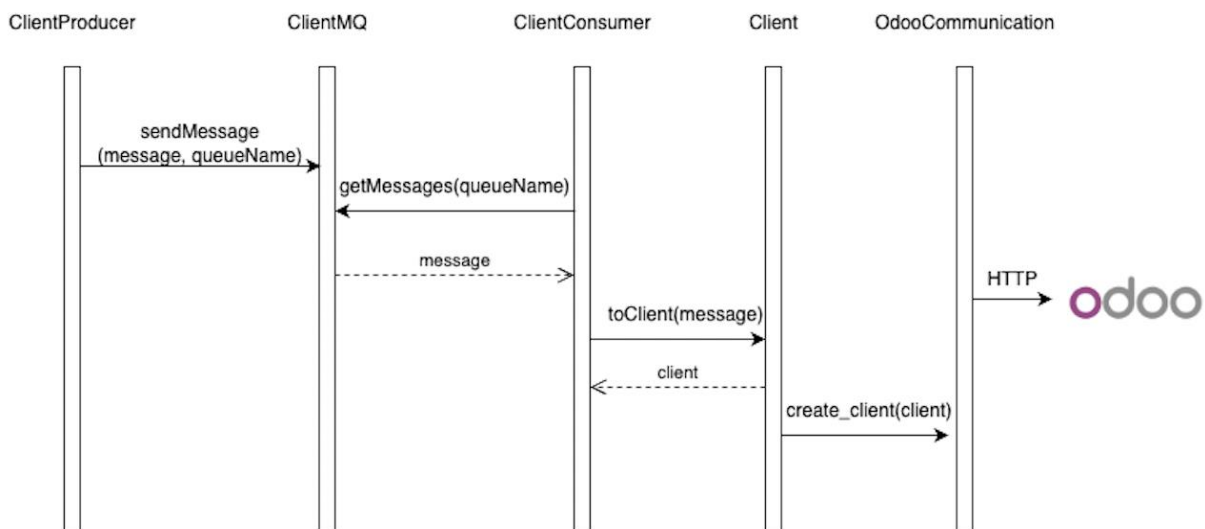


Figura 27: Diagrama de interacción del de registro de nuevo cliente mediante productores y consumidores

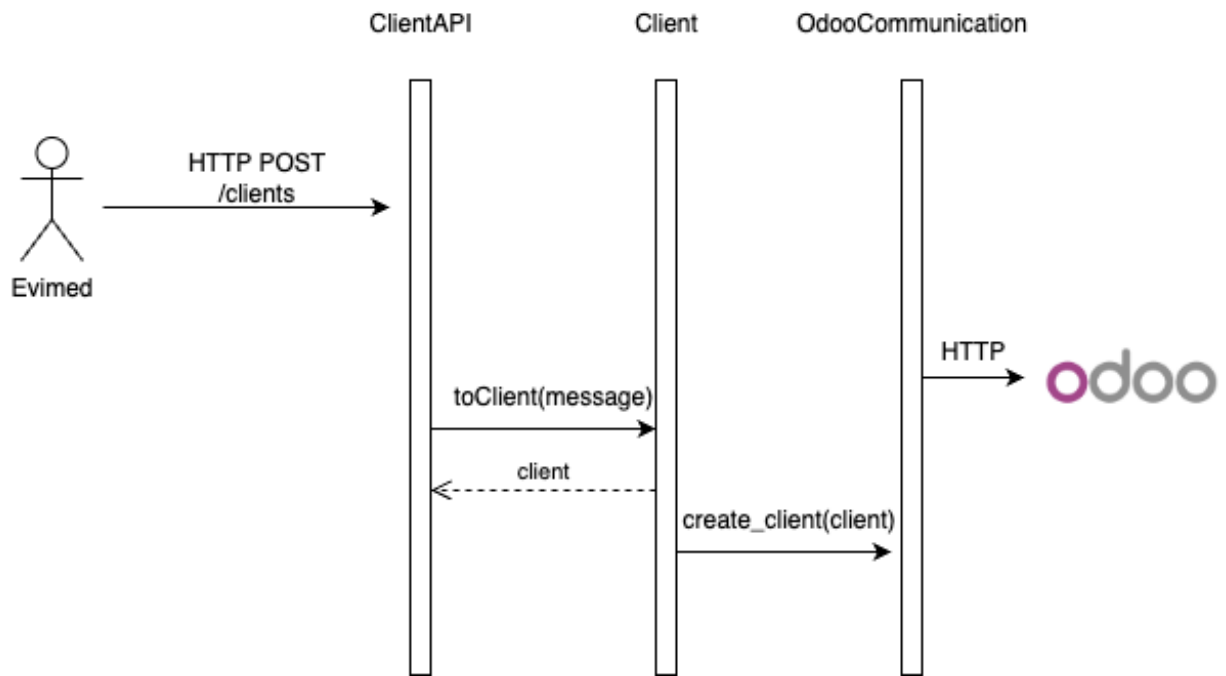


Figura 28: Diagrama de interacción de registro de nuevo cliente mediante API

Catálogo de elementos

Al igual que en los diagramas anteriores, se tomará como ejemplo la entidad cliente para describir al sistema, en lugar de hacer hincapié en cada una de las diferentes entidades, con el objetivo de facilitar el entendimiento de los diferentes elementos del sistema.

Elemento	Tipo	Responsabilidad
clientMqConsumer.py	Archivo	Encargado de leer mensajes de la clientMq y derivar la creación de la entidad a client.py.
clientMqProducer.py	Archivo	Encargado de enviar mensajes con los datos de registro de un nuevo cliente a la clientMq.
odooCommuncation.py	Archivo	Encargado de realizar las comunicaciones necesarias con la API externa de Odoo para enviarle los datos a ser reflejados.
secretManager.py	Archivo	Responsable de comunicarse con AWS <i>Secret Manager</i> para obtener las credenciales necesarias para el funcionamiento del sistema.
client.py	Archivo	Responsable de la creación de las entidades a partir de los diferentes formatos de entrada.
Odoo <i>external</i> API	API	API externa de Odoo, la cual efectúa acciones sobre los datos en el mismo.
client_csv.py	Archivo	Responsable de la entrada de múltiples clientes en formato CSV.
api.py	Archivo	Encargado de exponer los diferentes endpoints a Evimed y recibir sus requests de modificaciones de datos mediante HTTP.
clientMq	Cola de mensajes Amazon SQS	Encargada de proporcionar un mecanismo de mensajería asíncrona que permite la comunicación entre el flujo de registro de clientes y el Adapter.

Tabla 3: Catálogo de elementos Adapter

6.3.1.4.2. Atributos de calidad, tácticas y patrones

En esta sección se profundizará sobre los atributos de calidad priorizados al momento de diseñar la solución junto a principales tácticas utilizadas por el equipo para satisfacerlos. Tanto los atributos de calidad, como las tácticas, se encuentran extraídas del libro de arquitectura de software, *Software Architecture in Practice* [7]

Seguridad

La seguridad es un atributo de calidad fundamental en el desarrollo de software que se enfoca en proteger los sistemas y datos de posibles amenazas y vulnerabilidades. Este atributo busca garantizar la confidencialidad, integridad y disponibilidad de la información, así como prevenir el acceso no autorizado, el robo de datos y otros riesgos cibernéticos.

A pesar de que el Adapter no va a alojar datos de ningún tipo, al ser un canal de comunicación entre Evimed y Odoo, es importante protegerlo de acceso no deseado, ya que es justamente, un acceso a Odoo. Para hacer esto, se desplegó el sistema utilizando Elastic Beanstalk, el cual ofrece servicios que benefician a la seguridad del sistema, principalmente:

- Seguridad de la infraestructura: AWS se encarga de la seguridad física de los centros de datos, la red y otros componentes de la infraestructura subyacente. Esto incluye medidas como el control de acceso físico, la segregación de redes y la protección contra ataques DDoS (*Distributed Denial of Service*).
- Seguridad de red: Elastic Beanstalk se ejecuta dentro de una red virtual (VPC) de AWS, lo que permite aislar las aplicaciones y controlar el tráfico de red. Los grupos de seguridad y las listas de control de acceso pueden configurarse para controlar quién puede acceder a las instancias de la aplicación. Esto permite la implementación de la táctica *limit access*.

Se utilizaron las siguientes tácticas de resistencia a ataques:

Encrypt Data

AWS Secret Manager utiliza un proceso de cifrado de doble capa con claves de AWS KMS (*Key Management Service*) para proteger cada valor de secreto. Cuando el valor de un secreto cambia, *Secret Manager* solicita una nueva clave de datos a AWS KMS para protegerlo. La clave de datos se cifra bajo una clave de KMS y se almacena en los metadatos del secreto. Para

descifrar el secreto, *Secret Manager* primero descifra la clave de datos cifrada utilizando la clave de KMS en AWS KMS.

Importante destacar que *Secret Manager* no utiliza directamente la clave de KMS para cifrar el valor del secreto. En cambio, emplea la clave de KMS para generar y cifrar una clave de datos simétrica de 256 bits utilizando el Estándar de Cifrado Avanzado (AES, por sus siglas en inglés). Luego, utiliza esa clave de datos para cifrar el valor del secreto. *Secret Manager* utiliza la versión en texto sin formato de la clave de datos para cifrar el valor del secreto fuera de AWS KMS, y luego la elimina de la memoria. La copia cifrada de la clave de datos se almacena en los metadatos del secreto. Este enfoque proporciona una capa adicional de seguridad para proteger los secretos.

Authenticate actors

La táctica de arquitectura *Authenticate Actors* se centra en verificar la identidad de los actores, como usuarios o sistemas, que intentan acceder a un sistema de software. En el Adapter esto se realiza mediante el uso de una *API key* necesaria para el acceso a la API de Odoo, la cual está almacenada en AWS *Secret Manager* encriptada, protegiendo los datos de acceso no deseado.

Separate entities

Esta táctica implica diseñar un sistema de software dividiéndolo en componentes independientes que se comunican a través de interfaces definidas. La separación fue lograda mediante el despliegue de cada módulo en diferentes aplicaciones de *Elastic Beanstalk*, cada una con su diferente instancia de EC2. Esto no solo beneficia al sistema respecto a su seguridad, sino que a su disponibilidad y escalabilidad.

Modificabilidad

La modificabilidad es un atributo de calidad que refiere a la capacidad del sistema para ser modificado o adaptado de manera eficiente y efectiva en respuesta a cambios en los requisitos, las condiciones o los entornos. Este atributo implica que el software pueda ser fácilmente mantenido y mejorado sin causar efectos secundarios no deseados en otras partes del sistema.

Hubo mucho enfoque del equipo en mantener desacoplados los diferentes módulos del sistema, lo cual implica reducir la interdependencia entre los módulos, componentes o partes de un sistema.

Abstract common services

El equipo consideró que una alternativa a realizar en el Adapter es la aplicación de esta táctica de modificabilidad, que consiste en unificar múltiples servicios similares en una sola interfaz abstracta. Esto reduciría considerablemente el costo de cambio de ese servicio, debido que solo deberían realizarse los cambios en la nueva interfaz abstracta. Esto es muy aplicable a los diferentes consumidores y fue una decisión muy discutida, pero el equipo finalmente optó por separarlos. Con esta decisión se priorizó que Evimed pudiera administrar el estado de cada consumidor independientemente, así minimizando el costo de servicios iniciados cuando no son necesarios.

Use of intermediaries

La táctica de uso de intermediarios implica introducir componentes intermedios o intermediarios entre módulos o partes del sistema, lo que facilita la adaptación y el cambio de estos módulos sin afectar directamente a otros. Estos intermediarios actúan como capas de abstracción que separan las partes del sistema y, por lo tanto, reducen el acoplamiento.

Esta es una de las principales tácticas en las que el equipo se basó al desarrollar el Adapter, lo cual se ve claramente en el nombre que se seleccionó para el sistema. Esto es debido a que el sistema en cuestión actúa como una interfaz adaptadora de la API de Odoo, funcionando como intermediario entre los sistemas actuales de Evimed y Odoo.

Performance

La *performance* de un sistema se refiere a la velocidad, eficiencia y capacidad con la que realiza tareas o procesa datos, utilizando sus recursos de manera óptima. Implica la minimización de la latencia, la maximización de la velocidad de respuesta, la eficiencia de los recursos y la capacidad de escalar para manejar una mayor carga de trabajo. La optimización del rendimiento es esencial para garantizar una experiencia eficaz y satisfactoria tanto para usuarios como para aplicaciones.

Introduce concurrency

La táctica de rendimiento "uso de concurrencia" se refiere a la estrategia de permitir que múltiples tareas o procesos se ejecuten simultáneamente en un sistema.

Una de las principales características del Adapter es el uso de una Cola de mensajes para implementar la comunicación asíncrona entre los sistemas actuales de Evimed y Odo. Esto implica la implementación de concurrencia dentro de los flujos de Evimed, haciendo que su flujo de registro delegue la tarea de la validación lógica y registro de los datos del usuario al Adapter. Esto, en conjunto con la modularización de cada consumidor, favorece enormemente la escalabilidad del sistema, que como fue mencionado, fue uno de los principales problemas de Evimed, como fue explicado en la sección [2.2.2. El problema](#).

Disponibilidad

La disponibilidad de un sistema refiere a la capacidad de un sistema de software para estar en funcionamiento y accesible cuando se necesita, manteniendo un alto grado de operatividad y reduciendo al mínimo las interrupciones no planificadas. Este atributo se relaciona con la capacidad del sistema para ofrecer sus servicios de manera constante y confiable, garantizando que los usuarios puedan acceder y utilizar la aplicación o servicio sin experimentar tiempos significativos de inactividad o interrupciones.

Al estar desplegado en AWS, se estima un tiempo esperado de disponibilidad del servicio de al menos 99.99% del tiempo. En caso de no cumplirse esto, Amazon presenta una devolución de un porcentaje variable de la cuota del mes dependiendo de que tanto la disponibilidad haya disminuido. A continuación se muestran los valores exactos.

Monthly Uptime Percentage	Service Credit Percentage
Less than 99.99% but equal to or greater than 99.0%	10%
Less than 99.0% but equal to or greater than 95.0%	30%
Less than 95.0%	100%

Figura 29: Devolución por *downtime* de AWS EC2

6.3.2. Extensión de Google Chrome

El objetivo de construir una extensión de Google Chrome por parte del equipo fue aumentar el alcance, punto que había sido levantado por la segunda revisora como se explicó en la sección [6.1. Introducción](#), y como consecuencia de las actividades de ideación comentadas en la sección [6.2.3.3 Idear](#).

La extensión, “Odoo CRM Marketing Assistant” se encuentra deployada y disponible en la Chrome Web Store de Google de donde se puede añadir al navegador como cualquier otra.¹¹ Fue desarrollada bajo las guidelines de Chrome Developers [7], con el manifiesto v3, Javascript y Vue.js framework.

Como profundización a la ingeniería de requerimientos especificada en la sección [6.2 ingeniería de requerimientos](#), el equipo cree necesario ahondar sobre ingeniería inversa realizada sobre la funcionalidad de Hubspot CRM “Redactor de correos electrónicos con IA”, de dónde se vio en la Figura 25 de la sección [6.2.3.3 Idear](#) cómo luce la misma. El equipo comenzó probando dicha funcionalidad, se pidió una demostración gratuita de Hubspot CRM para interactuar con el formulario y ver cómo se comportaba. Se probaron los distintos tipos de tonalidades de la funcionalidad para ver si verdaderamente el texto brindado cambiaba su estilo. Luego, se pensó cómo la funcionalidad de Hubspot CRM interactuaba con la inteligencia artificial, más precisamente, cómo del formulario del *frontend* era posible enviar el *prompt* y obtener una respuesta. La idea de esta actividad fue pensar cómo está construido por detrás el “Redactor de correos electrónicos con IA” de Hubspot CRM, entendiendo cómo interactúa el usuario con el formulario para el posterior envío de esa información al *backend* del sistema y de esa manera obtener los requerimientos funcionales y no funcionales para replicar la funcionalidad.

Con el desarrollo de esta extensión, se buscó mejorar la forma en que el equipo de marketing de Evimed se dirige a clientes o futuros clientes con el fin de promocionar, dar seguimiento o realizar mails introductorios sobre los cursos ofrecidos. Para lograr dicho objetivo fue que el equipo desarrolló la funcionalidad de generación de templates de mails mediante la inteligencia artificial. Y también, siguiendo la misión de la empresa Evimed referida a la mejora continua, es que se realizó la segunda funcionalidad pensada para los directivos de la empresa, análisis

¹¹ Link: <https://tinyurl.com/Odoo-CRM-Marketing-Assistant>

de sentimiento de las encuestas a clientes para obtener *feedback* del curso y un puntaje general sobre el total de encuestados.

6.3.2.1. Uso de la inteligencia artificial

Para poder replicar la funcionalidad “Redactor de correos electrónicos con IA” de Hubspot CRM, el equipo tenía que hacer uso de la inteligencia artificial. Aquí se plantearon diferentes alternativas e interrogantes. Para comenzar, ninguno de los tres integrantes del equipo contaba con experiencia en dicha área, por lo que todo lo que el equipo hiciera, iba a requerir de una curva de aprendizaje. Se preguntaron si para esta funcionalidad el equipo debía entrenar un modelo de inteligencia artificial, para que vaya aprendiendo de sí mismo y que la generación de contenido sea lo más acertada posible a cursos anteriores que Evimed haya comercializado. Pero en esta etapa del proyecto, el tiempo comenzó a ser un recurso escaso, por lo que entrenar un modelo desde cero, no parecía muy atractivo, dado a que el entrenamiento toma tiempo y no se aprecian los resultados en un corto plazo.

De esta manera surgió la idea de utilizar la API *chat completions* de GPT, propiedad de OpenAI¹², que si bien el equipo no estaba familiarizado con ella, la curva de aprendizaje comparado con la creación de un modelo de inteligencia artificial desde cero, y el tiempo para implementarla, era mucho menor. OpenAI es la empresa creadora de ChatGPT - aplicación chatbot de inteligencia artificial compuesta por varios LLMs (*large language models*). En este caso se usó el LLM “gpt-3.5-turbo” [8].

La API¹³ tiene como único requisito contar con una API key, que se puede generar desde el sitio web de OpenAI¹⁴. Es importante destacar que dicha API no es gratuita y el cobro de la misma viene dada por la cantidad de *tokens* que se envían (*prompt*) y se reciben en la respuesta (el mensaje). Como parámetros de la solicitud, se debe incluir la utilización del modelo (gpt-3.5-turbo) y mensajes. En este último se incluye el rol del usuario que hace la petición (un usuario, el sistema o un asistente) y por último el contenido (*prompt*). Un *prompt* es una instrucción, pregunta o un texto que se utiliza para interactuar con sistemas de inteligencia artificial, en este caso, la API *chat completions*. En rol representa el rol del autor del mensaje. De acuerdo a este, *chat completions* entiende cómo debe comportarse y quien es que está haciendo la solicitud. El rol de "usuario" representa al usuario final real que envía el mensaje

¹² <https://platform.openai.com/docs/guides/gpt/chat-completions-api>

¹³ <https://platform.openai.com/docs/api-reference/chat/create>

¹⁴ <https://platform.openai.com/account/api-keys>

a la API *chat completions*. El rol de "asistente" representa quién responde al mensaje del usuario final. Este rol indica que el mensaje fue una respuesta del asistente (modelo de API *chat completions*). El rol de "asistente" es útil para establecer la respuesta anterior del modelo en la solicitud actual para mantener la continuidad de la conversación. Por último, el rol "sistema" es útil para crear un contexto o alcance de la conversación asignando un comportamiento específico al asistente de chat, por ejemplo: "Eres un experto en la utilización de CRMs".

Luego existe un campo auxiliar llamado temperatura, esto sirve para controlar la creatividad o aleatoriedad del texto generado por el modelo. Este es un valor entre 0 y 1 donde los valores más altos como 0,8 generarán salidas más randómicas, y los valores más bajos como 0,2 generarán salidas más acertadas y deterministas. A esto se lo conoce como alucinaciones [9] (*hallucinations*, en inglés). Una alucinación es cuando un modelo de inteligencia artificial genera información incorrecta pero la presenta como si fuera un hecho real. Este atributo al principio no fue utilizado por el equipo, pero luego se utilizó para mejorar la creatividad de la funcionalidad seteándolo en 0,7.

```
const url = 'https://api.openai.com/v1/chat/completions';
const response = await fetch(url, init: {
  'method': 'POST',
  'headers': {
    'Content-Type': 'application/json',
    'Authorization': `Bearer ${result.openAiApiKey}`,
  },
  'body': JSON.stringify(value: {
    'model': 'gpt-3.5-turbo',
    'messages': [{ 'role': 'user', 'content': prompt }],
    "temperature": 0.7
  }),
});
```

Figura 30: Llamada a la API *chat completions* de OpenAI

Como se puede apreciar en la Figura 30, se hace uso de la API nativa *fetch* de Javascript para realizar la solicitud, y de esa manera, obtener la información con los parámetros antes descritos. Luego en los headers de la solicitud se puede ver la API key que el usuario debe brindar a la

extensión para autenticarse con OpenAI. Por último, en el body de la solicitud, como se explicó anteriormente, se muestran los atributos del modelo utilizado, los mensajes y la temperatura.

La tarea de definición del prompt no fue una tarea sencilla, en especial por la salida requerida por el equipo (objeto donde se especifica el asunto y el contenido del mail). Para ello el equipo tuvo que investigar acerca de las mejores prácticas para que el modelo sea certero. En la documentación oficial de la API de OpenAI, existe una guía de ellas para una implementación exitosa.¹⁵

6.3.2.2. Requerimientos funcionales

Épica EV-3: Odoo CRM Marketing Assistant

[EV-3.RF-1]: Guardar la API key de OpenAI.				
Como	usuario	de		Evimed
Quiero que al presionar el botón “Guardar”, se almacene la API key de OpenAI				
Para que la API pueda autenticarse con OpenAI				

[EV-3.RF-2]: Obtener la API key de OpenAI.				
Como	usuario	de		Evimed
Quiero que al entrar en el apartado de configuración, se despliegue la API key almacenada				
Para poder verificarla o modificarla.				

[EV-3.RF-3]: Generación de template de mail.					
Como	usuario	de	marketing	de	Evimed
Quiero que al completar los campos requeridos y presionar “Generar Contenido”, se genere					
el	cuerpo	de	un	mail	

¹⁵ <https://platform.openai.com/docs/guides/gpt-best-practices>

Para la generación de mails que sirvan como template para una futura comunicación con clientes.

[EV-3.RF-4]: Importar la encuesta .xlsx.

Como usuario de Evimed
Quiero poder cargar el documento de Google Sheets de extensión .xlsx correspondiente a la encuesta de un curso
Para que pueda ser analizado posteriormente

[EV-3.RF-5]: Generar el feedback y asignar puntaje a curso.

Como usuario de Evimed
Quiero poder presionar el botón de “Generar análisis”
Para que se genere un análisis de sentimiento tomando como entrada las respuestas en texto plano de los clientes devolviendo un comentario general y un puntaje al curso.

[EV-3.RF-6]: Copiar información generada.

Como usuario de Evimed
Quiero poder presionar el icono de copiar
Para poder utilizar el template del mail generado en Odoo CRM

6.3.2.3. Requerimientos no funcionales

EV-3.RNF-1 - [Disponibilidad] El cliente debe recibir una documentación de como instalar la extensión y el repositorio para realizar futuros cambios y versiones.

EV-3.RNF-3 - [Deployabilidad] El sistema debe ser deployado en la Chrome Web Store para su correcta distribución.

EV-3.RNF-5 - [Usabilidad] Añadir Vue.js framework¹⁶ para construir interfaces de usuario dinámicas, efectivas y estructurar el código en componentes.

EV-3.RNF-6 - [Usabilidad] Añadir la librería Ant Design of Vue¹⁷ para utilizar componentes y mejorar la UX/UI de la extensión.

6.3.2.4. Restricciones

EV-3.RES-1 - Debido a que las extensiones de Google Chrome utilizan Javascript para sus scripts es necesario desarrollarla en dicho lenguaje.

EV-3.RES-2 - El sistema debe desarrollarse utilizando su manifiesto en la versión 3, debido a que Chrome Web Store no acepta nuevos deployments con versiones anteriores.

6.3.2.5. Arquitectura

En esta sección se explicará la arquitectura de “Odo CRM Marketing Assistant”, las decisiones tomadas, el diseño del código y los desafíos que el equipo tuvo que sortear.

Antes de explicitar la arquitectura, el equipo entiende que puede ser útil comentar acerca del funcionamiento general de una extensión de Google Chrome. Lo único que es mandatorio en cada una de las extensiones es el archivo manifest.json, el propósito del mismo es brindar al navegador toda la información necesaria acerca de la extensión. Partiendo desde el nombre, la descripción hasta configuraciones más complejas.

¹⁶ <https://vuejs.org/>

¹⁷ <https://antdv.com/docs/vue/introduce>

```

{
  "manifest_version": 3,
  "name": "Odoo CRM Marketing Assistant",
  "description": "Potencia tu Odoo CRM con esta extensión.",
  "version": "1.0.1",
  "background": {
    "service_worker": "/background.js"
  },
  "permissions": [
    "storage",
    "scripting",
    "activeTab"
  ],
  "host_permissions": [
    "*/**/*.odoo.com/*"
  ],
  "action": {
    "default_popup": "popup.html",
    "default_icon": "logo.png"
  },
  "content_scripts": [
    {
      "matches": [
        "*/**/*.odoo.com/*"
      ],
      "js": [
        "/content.js"
      ]
    }
  ]
}

```

Figura 31: Odoo CRM Marketing Assistant manifest.json

6.3.2.5.1. Descripción de la arquitectura

La arquitectura de una extensión de Google Chrome [10] viene dada en gran medida por su manifiesto. Como podemos observar el equipo utilizó la última versión del manifiesto disponible, esto es la versión v3, publicada inicialmente a fines del año 2021.

Representación primaria de la extensión

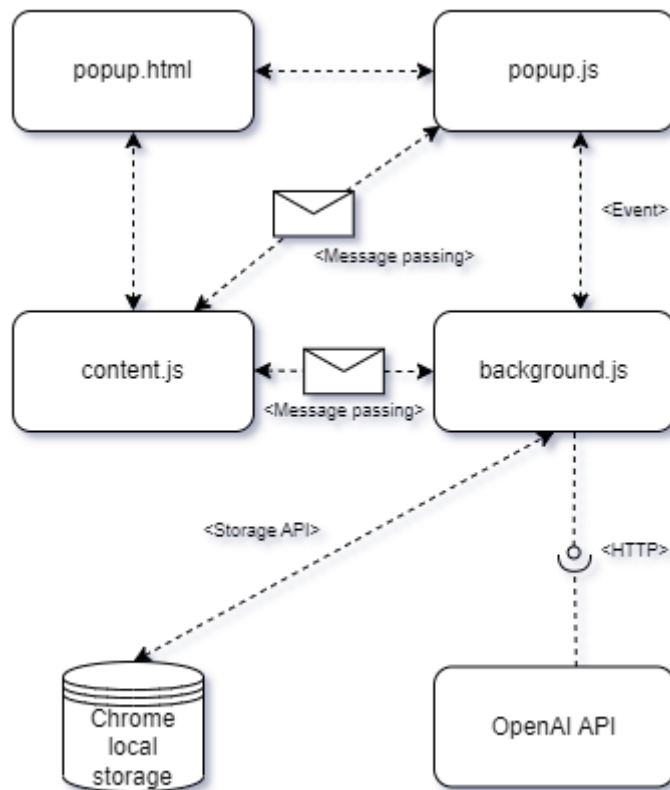


Figura 32: Arquitectura de Odoo CRM Marketing Assistant

En cuanto a la base de datos, el equipo definió utilizar *chrome.storage* API¹⁸ para almacenar información del usuario y estados. La *Storage API* está dividida en cuatro *buckets* o áreas, ellas son la *storage.local*, *storage.sync*, *storage.session* y *storage.managed*.

El *bucket storage.local* es un tipo de almacenamiento muy utilizado donde el acceso a la información es rápido, sus principales características son que almacena pares clave-valor, no tiene fecha de expiración, puede almacenar hasta 10MB de datos, solo scripts del mismo origen

¹⁸ <https://developer.chrome.com/docs/extensions/reference/storage/>

pueden acceder a los datos, se puede compartir entre diferentes pestañas y no se envían datos al servidor, es de uso exclusivo del cliente y por último los datos almacenados son de tipo texto.

El *bucket storage.sync* no tiene grandes diferencias con el *storage.local*, solo varía en si el usuario está logueado al navegador Chrome y si la funcionalidad de sincronización está activada, entonces la data también va a estar sincronizada.

El *bucket storage.session* mantiene la información en memoria hasta la duración de la sesión del navegador y por último el *bucket storage.managed* es un almacenamiento de tipo solo escritura (*read-only*) donde los administradores pueden usar un esquema y políticas empresariales para manejar las configuraciones de una extensión.

Por lo anterior expuesto, el equipo consideró que el *bucket* o área que más le servía para almacenar las diferentes API keys de OpenAI era el *storage.local*, dado que almacena pares clave-valor, y por lo tanto, podría estar implementada con tablas de hash [11] accediendo a los valores en orden 1 en el peor caso, algo muy eficiente.

Catálogo de elementos

Elemento	Tipo	Responsabilidad
popup.html	Archivo	Encargado de la vista de la extensión de Google.
popup.js	Archivo	Encargado de la lógica de la vista.
content.js	Archivo	Script que se ejecuta en las urls definidas en el manifest.json y que pueden escuchar del DOM (<i>Document Object Model</i>). Puede comunicarse con background.js mediante el evento message.
background.js	Archivo	Permite a la aplicación reaccionar ante eventos del buscador, como el guardado de una API key. Importante destacar que no puede escuchar del DOM, pero si se puede comunicar con los content scripts utilizando el

		evento message.
Chrome local storage	Storage API	Responsable de la obtención y el guardado de la API key de OpenAI proporcionada por el usuario.
OpenAI API	API	Llamada externa a la API de OpenAI mediante la librería fetch.

Tabla 4: Catálogo de elementos Odo CRM Marketing Assistant

6.3.2.5.2. Atributos de calidad, tácticas y patrones

En esta sección se comentarán los principales atributos de calidad tomados en cuenta para desarrollar la solución, también se hablará de las principales tácticas utilizadas por el equipo para satisfacerlos. Tanto los atributos de calidad, como las tácticas, se encuentran extraídos del libro de arquitectura de software, *Software Architecture in Practice* [12].

Disponibilidad

Como se explicó en la sección [6.3.1.4.2. Atributos de calidad, tácticas y patrones](#) del Adapter, la disponibilidad refiere a una propiedad del software, el cual debe estar listo para realizar una tarea cuando el usuario estime que es necesario y este debe responder correctamente al estímulo del usuario.

Como Odo CRM Marketing Assistant se encuentra deployada en la Chrome Web Store (dando cumplimiento al requerimiento no funcional EV-3.RNF-3 de la [sección de requerimientos no funcionales](#)) y por tratarse de sistemas de Google que se sabe que son altamente disponibles y brindan buen servicio a sus clientes, es que se puede decir que el sistema va a estar disponible la enorme mayoría del tiempo. Dado a que la extensión desarrollada por el equipo no es algo pedido por el cliente, ni reviste una gran responsabilidad su disponibilidad, el equipo cree que no va a ser un riesgo. De todas maneras, se buscaron los porcentajes de disponibilidad, se realizó una investigación sobre los SLAs (*Service Level Agreement*) y desafortunadamente, a la fecha de entrega de este documento, no existen tales acuerdos respecto de la Chrome Web Store y las APIs de OpenAI (sí existen acuerdos para los servicios Cloud de Google). En el caso de OpenAI, están en camino de tener un SLA, y mientras tanto existe un monitor sobre sus sistemas. Podría decirse que aplican tácticas de detección de fallas, como *ping/eco*, *monitor*, *heartbeat*, etc, para detectar las mismas valga la

redundancia. Dicho monitor¹⁹ refleja que las APIs de dicho proveedor tienen una disponibilidad de 99.83 % (solo es posible tomar un periodo de tiempo de noventa días).

El equipo pretendió realizar un promedio entre la disponibilidad de OpenAI y Chrome Web Store para obtener un porcentaje real de la disponibilidad de su extensión pero, dado a la falta de acuerdos, esto no es posible.

Modificabilidad

Como se explicó en la sección [6.3.1.4.2. Atributos de calidad, tácticas y patrones](#) del Adapter, la modificabilidad se trata del cambio, y el interés en él se centra en el costo y en el riesgo de hacer cambios. Entonces, la modificabilidad es la capacidad de un programa de implementar cambios necesarios, minimizando el impacto que generan y los costos de realizarlos.

Increase semantic coherence

Dicha táctica se encuentra dentro de las tácticas para aumentar la cohesión de un sistema y habla de que si responsabilidades A y B están en un módulo y no cumplen el mismo propósito, entonces deberían estar en módulos distintos. Entonces, si se toman a los módulos como los componentes realizados por el equipo en la extensión y tomando al *Single Responsibility Principle* de los principios SOLID [13], se puede decir que se separaron las responsabilidades en los componentes presentes. Vue.js en ese sentido da cierta ventaja dado a que ayuda a la componentización del sistema dividiendo responsabilidades.

Encapsulate

Esta táctica implica que un cambio a un módulo no debe propagarse a otro, mediante el rechazo consciente a la exposición de detalles de implementación internos y la dependencia de otros módulos. Si bien esta táctica habla de interfaces, en el caso de que un elemento deba utilizar dicho módulo, este solamente conozca su interfaz y no la implementación específica. En el contexto de la extensión, un cambio en un componente no debe propagarse a otro dado a que cada uno de ellos cumple una responsabilidad específica.

¹⁹ <https://status.openai.com/>

Seguridad

La seguridad es la habilidad de un sistema de proteger la información de los accesos no autorizados mientras continúa brindando acceso a las personas y sistemas que sí lo están, como también se explicó en la sección [6.3.1.4.2. Atributos de calidad, tácticas y patrones](#) del Adapter.

En el contexto de la extensión, puede ser un aspecto que podría mejorar. La extensión no maneja información sensible de los usuarios, solamente almacena la *API key* de OpenAI necesaria para interactuar con dicha API. Aquí el equipo discutió sobre las posibles acciones a implementar para dar una capa de seguridad al almacenar esta *API key*, de las que surgieron distintas ideas. La primera idea que se analizó fue la de hashear su valor con algún algoritmo de encriptación como AES [14] y luego guardarla en el Chrome Storage, pero aquí, el equipo debía guardar el *secret key* requerido para luego poder descryptar dicha *API key*. Por lo que no tenía sentido, dado a que no se cuenta con un archivo `.env` (archivo donde se guardan atributos clave-valor) para almacenar el *secret key* en la extensión y hardcodear dicho valor no es una opción. Por otro lado se pensó de implementar un *login* utilizando Firebase Authentication o algún otro mecanismo de autenticación, aquí se tenía el problema de que si existiera un *man in the middle* [15] podría interceptar la solicitud de obtención de la *API key* dejando sin efecto la implementación de algún factor de autenticación por no cumplir con el cometido inicial. La única manera que tendrían de estar seguros, sería implementando un servidor, donde al igual que el Adapter, se utilice *Secret Manager* de AWS.

Resultando: primero, dado que la *API key* de OpenAI se puede monitorear desde el sitio web de la compañía, dándola de baja si se entiende que hay un uso inadecuado de ella. Segundo, dado que el equipo entiende que implementar un servidor para esta extensión no sería redituable y tercero y último, dado que existen variadas extensiones en la Chrome Web Store donde se utilizan *API keys* de esta manera; el equipo resuelve levantar el punto pero seguir utilizando la *API key* de la manera desarrollada, monitoreando en la página web de la compañía OpenAI sobre la actividad de la misma.

Authorize actors

La técnica de autorizar actores de la categoría resistir ataques, se basa en asegurarse de que un usuario tenga los permisos necesarios para acceder y/o modificar recursos que está solicitando. Las peticiones que el usuario realiza, como la generación de contenido y la generación del análisis dado una encuesta de curso, deben realizarse proporcionando una *API key* previamente

ingresada en la sección de administración de la extensión (identificada con el icono de la rueda). En caso de que la API *key* no sea válida, ya sea porque los fondos se hayan consumido o la API *key* haya caducado, la API de OpenAI negará la petición. De esta manera, el sistema se desprende de la responsabilidad de autenticar a los usuarios y se delega al proveedor de la API dicha tarea.

Limit Access

La técnica de limitar el acceso, al igual que la anterior, se encuentra dentro de la categoría de resistir ante ataques. Se basa en limitar el acceso a recursos como conexiones a la red que pueden obtenerse por medio de bloqueos a hosts, cerrando puertos o rechazando un protocolo. En este caso mediante el manifiesto de la extensión, se limitó la inyección de scripts solo a la página de Odoo CRM, esto se puede verificar en la [Figura 31 donde se muestra el manifest.json](#). Si se intentara inyectar scripts que no sean para el dominio de Odoo CRM, esto se vería bloqueado por la política de CORS (*Cross-Origin Resource Sharing*) y CSP (*Content Security Policy*) [16].

Usabilidad

La usabilidad de un sistema es todo lo relacionado con la facilidad que un usuario tiene para completar una tarea deseada y el tipo de ayuda que el sistema provee como forma de dar soporte al usuario.

Para lograr una UX/UI acorde, el equipo decidió utilizar *Vue.js framework v2* dado a que cuentan con experiencia en dicho *framework*. Con esto se buscó poder desarrollar la parte visual de la extensión en un tiempo menor, y también con la ayuda de la librería *Ant Design* para el *framework* de *Vue.js* poder acceder a mejores estilos, transiciones y elementos, ya sean *inputs*, *dropdowns*, etc para que la visual sea estéticamente atractiva e intuitiva.

Figura 33: Asistente de contenido de Odoo CRM Marketing Assistant

Las pantallas de la extensión fueron creadas manteniendo coherencia de estilos, tamaño de los botones, *inputs*, *placeholders* para dar *feedback* a los usuarios de qué es lo que se espera que ingresen en dichos campos, manteniendo la consistencia en toda la extensión. Mediante un menú superior se pretende mostrar al usuario mediante una asistencia visual de color azul donde esta ubicado dentro de la extensión. Como se puede apreciar en la Figura 33, el botón “Generar contenido” se mantiene desactivado, es decir, sin poder ser seleccionado, hasta que todos los campos sean completados dando *feedback* al usuario sobre la falta de información.

Figura 34: Análisis de sentimiento de Odoo CRM Marketing Assistant

Otro punto a través del cual el equipo intentó mejorar la usabilidad del sistema, fue en cuanto a brindar *feedback* sobre entradas incorrectas por parte del usuario. A modo de ejemplo, si se observa la Figura 34, a la hora de ingresar encuestas para el análisis de sentimiento, se espera se ingrese un archivo .xlsx (encuestas de satisfacción de Evimed descargadas de Google Forms), por lo que el usuario al subir un archivo incorrecto será alertado oportunamente.

Este atributo de calidad no es fácil de medir y tampoco de definir dado que depende del tipo de usuarios que sean objetivo para la aplicación, su familiarización con otros sistemas de software y demás factores. El equipo pensó esta extensión y las funcionalidades para ser extensibles al público general, pero para esta instancia se tomó como usuarios, los empleados de Evimed, ya sea equipo de tecnología, como equipo de marketing. Aquí se tiene una posible mejora, dado a que el equipo no realizó prototipos en cuanto a la UI como se explicó en la sección [6.2.3.4. Ausencia de la etapa de prototipación](#). También las UI de la extensión son simples, es decir, el tiempo que se debía invertir por parte del equipo para la realización de estos prototipos en alguna herramienta de prototipación como Figma²⁰ eran muy parecidos o iguales al tiempo de desarrollo, por lo que se optó por mostrar las funcionalidades y UI una vez concluidas las mismas. De todas formas, el *feedback* proporcionado por los usuarios (empleados de Evimed) una vez hecha la demostración fue positivo.

6.3.3. Conclusiones

Con respecto a la extensión de Google Chrome “Odoo CRM Marketing Assistant”, el equipo logró sortear las dificultades de desarrollar una extensión propia, teniendo en cuenta que no se tenía experiencia en esta área. También se tomó como algo positivo el proceso de deployar su extensión en la Chrome Web Store para los usuarios de Evimed y que el equipo de Google la haya aprobado para pertenecer al numeroso catálogo de extensiones existentes. El proceso, al tener que justificar todos los permisos que la extensión utiliza, les permitió reflexionar sobre lo que el equipo le estaba pidiendo al usuario, logrando en ciertos casos eliminar ciertos permisos que no eran necesarios o en el caso del *content script*, enfocar solo la inyección de scripts en el entorno de Odoo CRM y no en todas las páginas webs.

²⁰ <https://www.figma.com/>

A su vez, el equipo cree que la experiencia interactuando con la API de OpenAI, introduciéndose en la inteligencia artificial, el cuidado por el *prompt engineering* y el manejo de la temperatura, fue muy positiva y enriquecedora.

En relación al Adapter, no solo se lograron en su totalidad los requerimientos funcionales y no funcionales relevados, sino que el equipo tuvo la experiencia de sumergirse en el negocio, analizando las diferentes necesidades del cliente con el objetivo de diseñar un sistema hecho a medida para ellos. Esto implicó el uso de diversos conceptos aprendidos en la carrera para discutir las diferentes decisiones del diseño de la arquitectura del sistema, lo cual fue una experiencia muy enriquecedora para el equipo, al permitir aplicar los conceptos teóricos en un ambiente real.

Finalmente, en ambos productos, por las restricciones temporales mencionadas en la sección [6.1. Introducción](#), se tuvo que lidiar con el desafío de idear, diseñar y desarrollar rápidamente un producto funcional. Esto implicó un gran esfuerzo sostenido y constante adaptabilidad del equipo, lo cual aportó mucho valor a su desarrollo profesional.

7. Gestión del proyecto

En esta sección se desarrollará la gestión de las etapas dos y tres, implantación del CRM elegido y productos satélites, respectivamente. Con la excepción que en la sección [7.4. Gestión de riesgos](#) que se evaluará la totalidad del proyecto (las tres etapas). La gestión de la primera etapa del proyecto fue detallada en la sección [4.2. Gestión del proyecto](#).

7.1. Proceso y ciclo de vida

7.1.1. Ciclo de vida

El ciclo de vida de un proyecto de software determina cómo se planifica, se desarrolla y entrega el proyecto. El equipo consideró que el ciclo de vida incremental iterativo era el indicado, dado a que permite ir entregando valor al cliente en cada iteración y también ir evolucionando el producto (incremental) [17].

7.1.2. Metodología de referencia

A la hora de elegir una metodología de referencia, el equipo se enfocó en adoptar una metodología ágil, es decir, se adoptaron principios mencionados en el manifiesto ágil [18] para dicho fin. Decidieron no enfocarse en un marco de trabajo en particular, como Scrum, pero sí se tomaron ciertas ceremonias y artefactos.

No se eligió un marco de trabajo en particular, dado que el equipo de trabajo debido a sus obligaciones externas, como trabajo y realidades académicas, no se quería sobrecargar con todas las ceremonias que Scrum plantea, justamente para poder desarrollar ágilmente y optimizar el tiempo.

Como se mencionó anteriormente, del marco de trabajo Scrum se tomó el manejo de los requerimientos con un formato de “historias de usuario”. Estas son descripciones breves y sencillas de los requisitos funcionales del cliente.

7.1.3. Descripción general del proceso

Para una mejor gestión, el equipo dividió el proyecto en tres grandes épicas. Ellas fueron la configuración de Odo CRM, el Adapter y la extensión de Google Chrome. Para llevar a cabo

dichas épicas se definieron varias historias de usuario por cada una de ellas, que se atacaron en iteraciones de dos semanas cada una.

Las historias de usuario tienen como objetivo describir una funcionalidad o característica del sistema desde la perspectiva del usuario. Tienen un **título** donde hacen referencia a la funcionalidad o característica a desarrollar y una **descripción o narrativa** donde se especifica el siguiente formato: “como [actor], quiero [funcionalidad], para [valor o beneficio]”. Luego le siguen los **casos de pruebas o escenarios**, donde se definen criterios de aceptación, confirmación de lo que el sistema debe hacer y podría también contener una sección de **anotaciones** que ayudan a ver en qué se quedó con el cliente o preguntas a realizar, cuando la historia de usuario llega al desarrollador, dicha sección debe estar vacía.

Las historias de usuario fueron divididas en tres tipos de tickets, las épicas, las historias de usuario y los *spikes*. Las épicas son grandes historias de usuario que se encuentran formadas por muchas historias de usuario más pequeñas. Estas últimas son las que se trabajan en las diferentes iteraciones. Por último se tienen los *spikes*, que son aquellos que representan las investigaciones que se hicieron para luego poder realizar las historias de usuario, estos tickets tienen la particularidad que no tienen puntos de historia pero si tienen una duración estimada en horas.

Para los criterios de aceptación, se realiza una historia de usuario llamada “*Definition of Done*” (DOD, por sus siglas en inglés) para asegurar la calidad y para evaluar cuando el equipo termina una historia de usuario o no.

Como es habitual en la industria, el equipo tomó del marco de trabajo Kanban, su tablero. El tablero de Kanban es una herramienta ágil de gestión de proyectos diseñada para ayudar a visualizar el trabajo, limitar el trabajo en curso y maximizar la eficiencia.

En las siguientes secciones se detallan los roles adoptados por el equipo, artefactos y ceremonias resultantes aplicadas.

Roles

Como en el marco de trabajo Scrum, la definición de roles es algo fundamental para que cada integrante del proyecto sepa sus responsabilidades y que se espera de él.

En primer lugar, tenemos al equipo de **Developers** quienes son los responsables de integrar y construir el producto. Este rol abarca tareas que cubren desde el diseño, análisis, desarrollo, pruebas, generación de documentación, etc. Todos los integrantes del equipo cumplieron este rol y no se diferenciaron entre *frontend*, *backend* u otras tareas como arquitectura.

En segundo lugar, el equipo tuvo la necesidad de contar con un **Cloud Architecture**, encargado de diseñar la arquitectura del sistema y todo lo relacionado con las colas de mensaje de Amazon SQS, la creación del contenedor en Docker, el *deployment* del Adapter en Amazon, entre otras tareas. También cumplió el rol de consultor para los demás integrantes del equipo en relación a la construcción de la extensión de Google Chrome. Dicho rol fue desempeñado por Marcos Bazzano.

En tercer lugar, el equipo definió contar con un **Agile Coach** encargado de llevar adelante la metodología ágil de manera eficiente y responsable, llevando a cabo las ceremonias planteadas, controlando el desarrollo de la iteración y velando por el cuidado del equipo. Dicho rol fue desempeñado por Romina Rodríguez.

Por último, el equipo tuvo la necesidad de incluir un **Client Manager** encargado de organizar las reuniones con el cliente y de llevarlas a cabo, pedir por datos e información actuando de cierta manera como un **Product Owner**, poniéndose en lugar del cliente para lograr un mejor producto. Dicho rol fue desempeñado por Juan Andrés Nervi.

Artefactos

Product Backlog

El *Product Backlog* contiene historias de usuario con un número de identificación. En cuanto a la priorización de las historias de usuario, se dio total libertad al equipo a la hora de ordenarlas, teniendo en cuenta que para Evimed, lo más importante era la segmentación de clientes para un posterior envío de mails para ofertas y demás acciones.

Se utilizó la herramienta JIRA de Atlassian²¹ para gestionarlo y actualizarlo conforme surgen nuevas tareas. Las tareas eran identificadas en base a alguna de las tres épicas definidas.

²¹ <https://www.atlassian.com/software/jira>.

Backlog de la iteración

El *Backlog* de la iteración contiene las historias de usuario a ser desarrolladas para cada iteración. Están estimadas en *story points* o puntos de historia. El equipo decidió trabajar con tres columnas en el tablero Kanban, la columna de “*Backlog*”, la columna de “*Doing*” y la columna de “*Done*”. En la primera, el equipo tenía priorizada las historias de usuario a trabajar en esa iteración y luego el responsable de dicha historia de usuario era el encargado de actualizar su estado en dicho tablero, pasandola a la columna de “*Doing*” cuando se estaba desarrollando la funcionalidad y una vez terminada y cumplida la historia de usuario DOD, mover la historia de usuario a la columna de “*Done*”.

Ceremonias

Planning

Dicha ceremonia tomaba lugar al inicio de cada iteración. En la misma se encontraban todos los miembros del equipo. El objetivo de dicha ceremonia es obtener el alcance de una iteración mediante la selección de historias de usuario para el *Backlog* de la iteración. En ella, también se fija el objetivo de la iteración.

El equipo de desarrollo junto con el *Agile Coach* puntuaba las historias de usuario con la serie de Fibonacci (1, 2, 3, 5, 8, 13) donde los puntos representan los puntos de historia. Con la ayuda de la aplicación Easy Retro, se construyó un tablero con columnas (una por cada integrante del equipo) donde se siguió la técnica de *Planning Poker* [19] para el proceso de puntuación de cada historia de usuario, con la diferencia que en vez de iterar hasta lograr el consenso como la técnica así lo explica, los desarrolladores que puntuaron dichas historias con los puntos más alejados, es decir, los más bajos y los más altos debían explicitar sus argumentos y de esa manera lograr el consenso en una sola vuelta, simplemente para optimizar el tiempo.

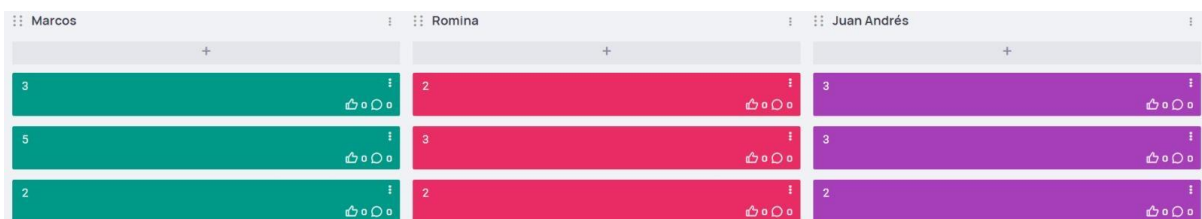


Figura 35: Votación de ejemplo en la aplicación Easy Retro

Bi-Weekly meetings

El equipo definió realizar reuniones de seguimiento dos veces por semana. Las mismas tomaron lugar los días martes y viernes, estratégicamente definidas. Los martes, para que los días miércoles donde estaban seteadas las reuniones quincenales con el tutor del equipo y con el cliente, ya tuvieran definidos los temas a tratar. Y luego los días viernes, al terminar la semana, para un status general. De todas maneras, el objetivo de las *bi-weekly* no es preparar reuniones ni hacer reflexiones sino que el objetivo es comunicar lo que el desarrollador hizo, en que está y si se encuentra bloqueado. Las reuniones quincenales entre el cliente y tutor fueron estratégicamente seteadas para que todos los miércoles el equipo tenga reunión, ya sea con el tutor o cliente.

Review

La *Review* fue una ceremonia llevada a cabo por todos los integrantes del equipo conjuntamente con el cliente, representado generalmente por los directivos de la empresa y en algunas ocasiones previamente comunicadas, solicitamos la presencia de empleados del área de marketing para que valide lo construido. En dichas ceremonias, el objetivo era mostrar el avance que el equipo de desarrollo realizó en cada iteración.

Previo a dicha ceremonia el equipo se preparaba con anterioridad para revisar lo implementado en la última iteración para organizar lo que se iba a mostrar y como.

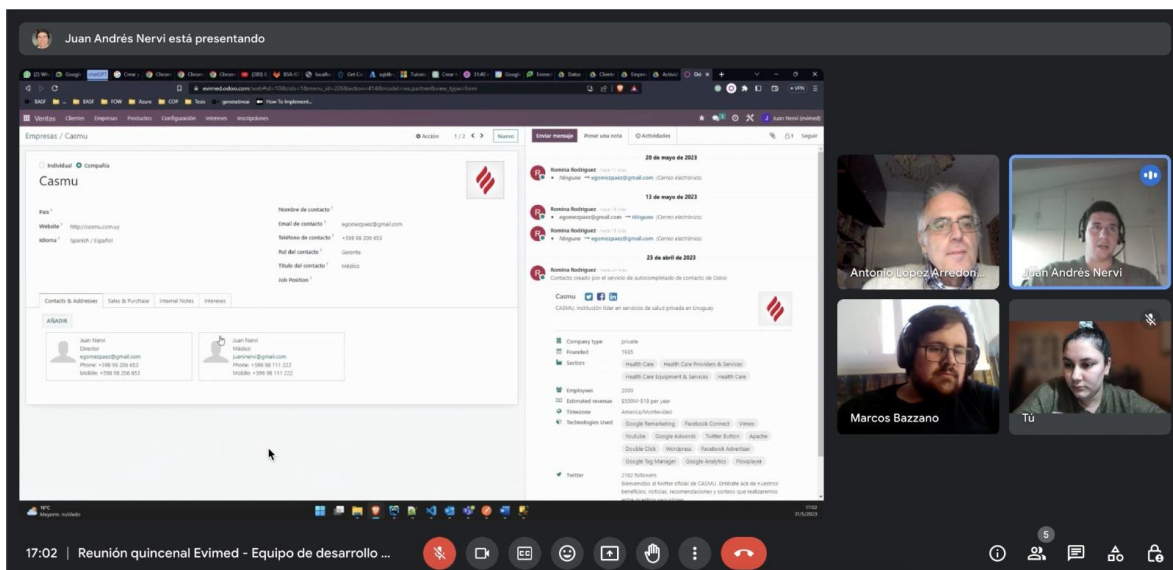


Figura 36: *Review* con el cliente

Retrospective

Las *retrospectives* son ceremonias que toman lugar luego de realizar las *reviews* con el cliente. El objetivo de esta ceremonia es reflexionar sobre posibles cambios, mejoras y actitudes que se dieron durante la iteración, y que entonces estaría bueno volverlas a repetir (en caso de que sean positivas), que el equipo debería abandonar con rapidez (en caso de que no aporten) o adoptar las nuevas. De estas reflexiones se obtienen *action points* de la iteración, donde se pasa en limpio lo que se quiere mejorar o dejar de hacer para futuras iteraciones. Dichos *action points* no son obligatorios.

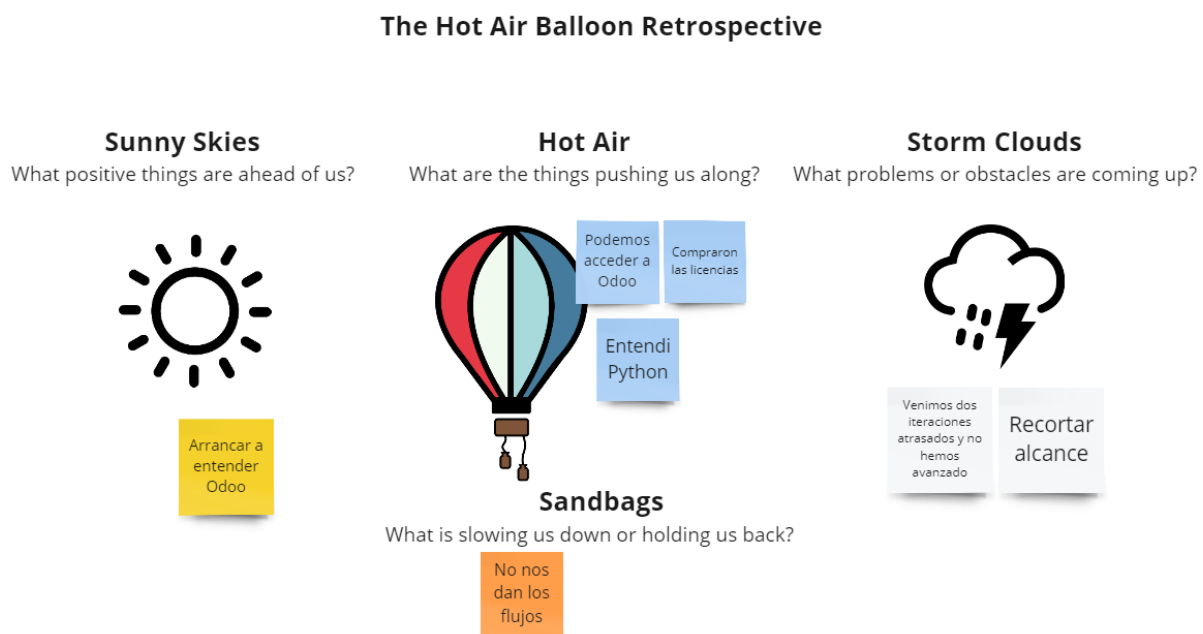


Figura 37: Ejemplo de *Retrospective* en la herramienta Miro

7.2. Planificación temporal

7.2.1. Sobre la estimación

La estimación del esfuerzo en el desarrollo de *software* es una faceta fundamental en la gestión de proyectos. Este proceso se traduce en predecir la cantidad de esfuerzo requerido, expresado en términos de puntos de historia, para dar forma al producto. Dichas evaluaciones proveen al equipo de una previsión del tiempo necesario para llevar a cabo el proyecto.

La obtención de una estimación precisa en las etapas iniciales del proyecto reduce la probabilidad de incumplir con el compromiso establecido. A medida que se acumula un mayor conocimiento, las estimaciones se vuelven más precisas y de mayor utilidad.

El equipo empleó una técnica conocida como "estimación relativa," que se basa en la secuencia de Fibonacci. En esta metodología, a cada tarea que requería ser desarrollada se le asignó un número, también denominado puntos, según esta secuencia. Se la denomina relativa debido a que se fundamenta en la premisa de que una tarea valorada en 2 puntos demandará menos de la mitad del esfuerzo que se necesita para completar una tarea valorada en 5 puntos.

La cantidad de puntos que un equipo logra completar durante una iteración se considera su velocidad. Es crucial destacar que estos puntos no representan horas, sino una estimación del esfuerzo basada en la complejidad inherente a cada tarea.

7.2.2. Proceso de planificación

Una vez que el equipo contaba con el product backlog refinado, el proceso para realizar las estimaciones de las historias de usuario tomó lugar cada dos semanas, al finalizar cada iteración.

El equipo llevó a cabo sesiones de planificación para ver cuantas y cuales tareas debían abordarse para la siguiente iteración. Para definir la cantidad de tareas por iteración se tomó en cuenta la velocidad de las iteraciones anteriores. Mientras la suma de puntos de todas las tareas agregadas a la iteración sea menor que la velocidad del equipo, entonces se pueden seguir agregando tareas, pero una vez que se alcanza este valor el equipo no puede seguir agregando tareas, ya que lo más probable que sucediera sería que no se llegara a completarlas.

7.3. Gestión del esfuerzo

Para gestionar el tiempo empleado en cada área del proyecto, el equipo utilizó la herramienta Toggl²². Cuando cada estudiante comenzaba a trabajar en el proyecto, encendía la herramienta para mantener un registro horario y la guardaba bajo alguno de los siguientes proyectos:

- **Investigación CRMs:** en este proyecto se engloba todo lo relacionado a los experimentos de PopCorn Flow. Toda la investigación, pruebas, reuniones con equipos de ventas, etc.
- **Implantación (Desarrollo):** en este proyecto inicialmente se registraba todo lo que abarcaba la implantación de Odo. Luego, cuando se agregaron los productos satélites,

²² <https://toggl.com/track/>

se agregó a la misma categoría. Esta decisión fue del equipo, ya que el fin era el mismo, desarrollar productos.

- **Documentación:** en este proyecto se encuentra todo lo referenciado a documentos entregables, tanto al cliente como a la Universidad, preparación de revisiones, etc.
- **Reunión con cliente:** reuniones quincenales con el cliente, y algunas excepcionales que se hicieron en medio.
- **Reunión de equipo:** reuniones bi-weekly, retrospectivas, plannings, brainstorming, etc.
- **Reunión con tutor:** reuniones quincenales con el tutor.

Estos proyectos (identificados por nombre y colores diferentes) fueron creados con el fin de diferenciar y agrupar las áreas de trabajo más fácilmente, pudiendo sacar métricas por área, por etapa, por integrante, y así poder visualizar el trabajo que se estaba dedicando, detectar problemas y proponer mejoras.

Para estas etapas, la disponibilidad horaria que el equipo presentaba era la siguiente:

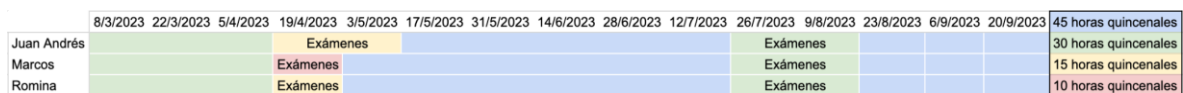


Figura 38: Disponibilidad horaria del equipo

A continuación, se irá haciendo un breve análisis respecto a cada etapa del proyecto.

7.3.1. Primera etapa

De esta etapa se habló en la sección [4.2. Gestión del proyecto](#).

7.3.2. Segunda etapa

Para esta segunda etapa, se tenía el plan que se muestra en la Figura 39.

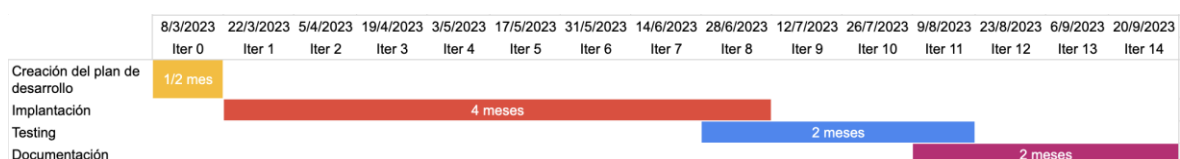


Figura 39: Roadmap inicial segunda etapa

Como se mencionó en la sección [5.2.3. Proceso de relevamiento](#), se tuvo un percance en cuanto a la obtención de requerimientos y por ende, a la creación del plan de desarrollo. Todo ese tiempo, hasta obtener las licencias y el MER, el equipo lo tomó como una iteración 0.



Figura 40: Roadmap actualizado segunda etapa

Como se puede apreciar, en la Figura 41, el equipo destinó muchas horas a reuniones de equipo y documentación. Estas horas se tratan de los momentos de ideación que tuvo que recurrir, cuando a mitad de la etapa surgió la advertencia de que el alcance podría ser corto. Allí también se encuentran las dos revisiones que el equipo debió preparar intensamente para poder reflejar la situación en la que se encontraban.

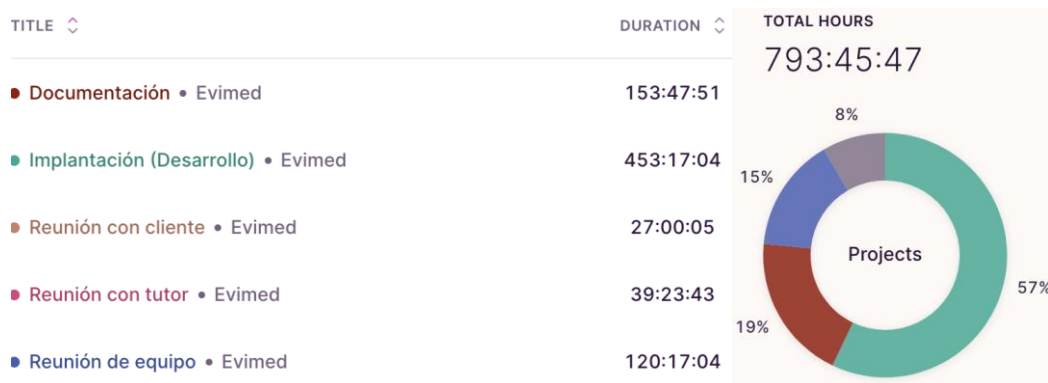


Figura 41: Distribución horaria segunda y tercera etapa

7.3.3. Tercera etapa

Esta tercera etapa surgió de la segunda revisión, que tuvo lugar el 16 de mayo. En esta revisión, como se comentó previamente, recalcó que el alcance que tenía el proyecto hasta el momento podría no ser suficiente para una tesis de grado de tres ingenieros, dado que todas las integraciones que parecían existir en un comienzo, se fueron desvaneciendo. He ahí que el

equipo tuvo que volver a plantearse el roadmap, agregando esta nueva etapa que llamaron desarrollo de productos satélites

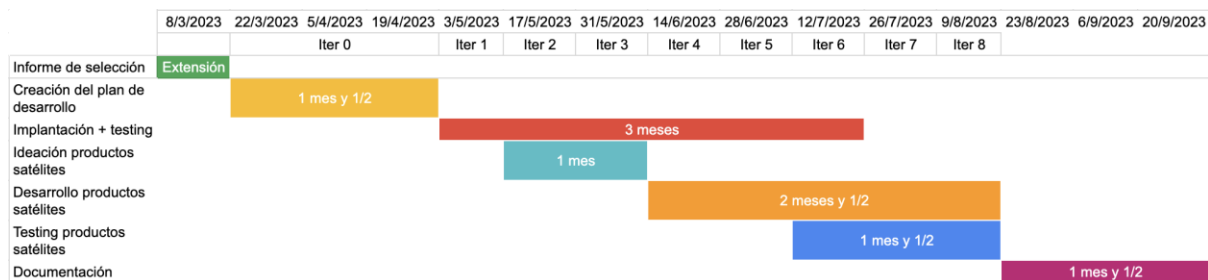


Figura 42: Roadmap tercera etapa

En este nuevo roadmap, se muestra cómo se agregó el desarrollo de productos satélites que tuvo que convivir con la implantación en Odo. El equipo se pudo adaptar, pudo integrar la etapa de implantación y testing en los tres meses que marcaron y hacer un MVP de productos satélite en menos de tres meses. Para esto, tuvieron que dividir y conquistar, cada uno siendo referente en cada proyecto y llevándolo a buen término.

Desde el momento en que el equipo comenzó a desarrollar los productos satélites, se puede observar que las horas de reuniones disminuyeron notoriamente, enfocándose más que nada en la implantación, desarrollo y revisiones (documentación).

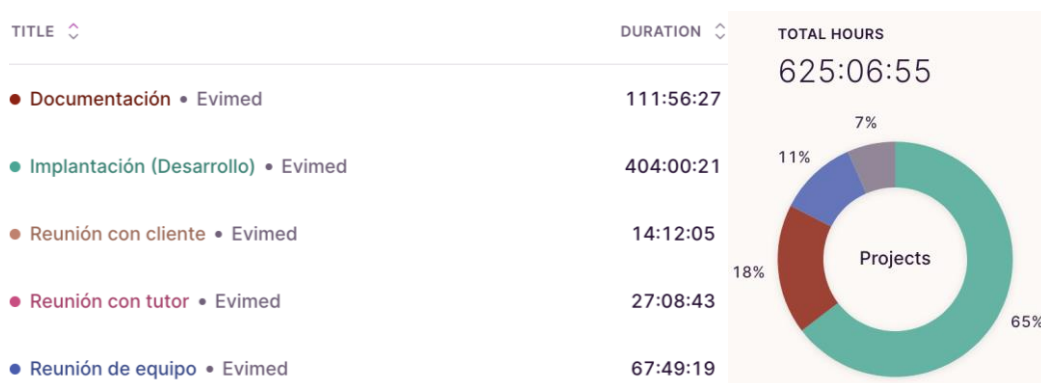


Figura 43: Distribución horaria a partir de comenzado el desarrollo tercera etapa

7.3.4. Documentación

Y para finalizar, en el período de documentación el equipo tomó una licencia en sus respectivos trabajos para poder garantizar el horario que se había estipulado. Hubo algunos contratiempos como enfermedades, exámenes que no estaban previstos o arreglos de pequeños bugs, pero fueron bien mitigados logrando finalizar con la documentación en tiempo y forma.



Figura 44: Distribución horaria de la etapa de documentación

7.3.5. Distribución de horas totales

Para dar una visión general de lo que fue todo el proyecto, desde el 16 de noviembre de 2022 hasta el 28 de septiembre de 2023, el equipo empleó 1426 horas.

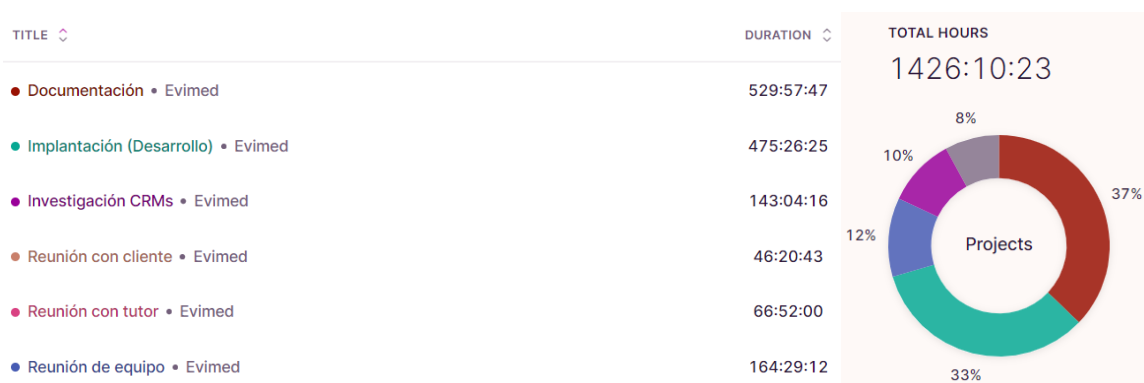


Figura 45: Distribución horaria total del proyecto

7.4. Gestión de riesgos

En esta sección se profundizará sobre los diferentes riesgos que encontró el equipo a lo largo de la realización del proyecto. Los mismos se identificaban y se actualizaban en la ceremonia *Planning*.

Se categorizan los riesgos respecto a su temática principal, para facilitar el entendimiento del contexto en el que se encontraba el equipo y el proyecto en su momento. Los riesgos R02 y R05, surgen del *feedback* del comité de selección de proyectos de la Universidad ORT Uruguay

al momento de presentar la carta de proyecto, este período no está incluido en el Gráfico 2, debido a que aún no se tenían iteraciones.

Se especificarán los riesgos con el siguiente formato:

RXX: Nombre del riesgo (Fecha de descubrimiento)

- Probabilidad: **Probabilidad del riesgo y comentarios adicionales**
- Impacto: **Impacto del riesgo y comentarios adicionales**

En caso de que algunos de los atributos del riesgo sean actualizados en el tiempo, se agrega un ítem con la siguiente notación:

- Actualización **fecha de la actualización: Nueva probabilidad o riesgo, comentarios sobre la razón de actualización**

A continuación se especifican los diferentes valores posibles para ambos atributos de un riesgo.

Impacto	Grado
Ninguno	0
Leve	1
Medio	2
Importante	3
Crítico	4
Catastrófico	5

Tabla 5: Grado de impacto de los riesgos

Probabilidad	Grado
Imposible	0
Poco probable	0,2
Probable	0,4
Muy probable	0,6
Altamente probable	0,8
Problema	1

Tabla 6: Grado de probabilidad de los riesgos

7.4.1. Riesgos de cliente

R02: Demoras de parte del cliente en la adquisición del CRM elegido (15/11/22)

- Riesgo de cliente.
- Probabilidad: **Probable**, ya que pueden surgir eventualidades con el sector de pagos y financiamiento de EviMed.
- Impacto: **Crítico**, porque generaría un tiempo donde no tendríamos la Probabilidad de avanzar en la implementación.
- Actualización 15/03: Probabilidad: **Problema**. Ya que no sólo no tuvimos noticias de la adquisición de las licencias, sino que también no se presentaron a la reunión estipulada.
- Actualización 17/04 Probabilidad: **Imposible**. Compraron las licencias

R05: Que el cliente no quiera comprar nuestro CRM elegido (15/11/22).

- Probabilidad: **Poco probable**
- Impacto: **Crítico**, ya que tendríamos que debatir con ellos y probablemente buscar otra opción, congelando el comienzo del desarrollo hasta una nueva decisión

- Actualización 15/03: Probabilidad: **Probable**, ya que el cliente no envía actualizaciones sobre la situación respecto a la compra de Odoo ni responde los mensajes hace una semana.
- Actualización 17/04: Probabilidad: **Imposible**, se nos entregaron las licencias de Odoo.

R08: Problemas de comunicación con el cliente (16/04/23)

- Probabilidad: **Altamente probable**, dado que ya ha sucedido en el pasado
- Impacto: **Crítico**, debido a que esto dificulta la obtención de datos y feedback

R11: Disgusto de Evimed al ser informados del nuevo alcance (10/06/23)

- Probabilidad: **Poco probable**, dado que el cliente no está mostrando especial interés en el proyecto.
- Impacto: **Crítico**, esto podría comprometer nuestra relación con el cliente e implicar una posible desvinculación del proyecto.
- Actualización 18/06 Probabilidad: **Imposible**, hablamos con EviMed y tuvieron buena reacción.

R12: Ausencia de los datos de prueba para Odoo (10/06/23)

- Probabilidad: **Altamente probable**, dadas las experiencias pasadas con dependencias de Evimed.
- Impacto: **Medio**, implicaría invertir tiempo del equipo en conseguir un dataset de pruebas público.
- Fecha: 24/06 probabilidad: **Problema**, dada la aún ausencia de los datos de prueba, se asumió su no existencia para dejar de depender de Evimed
- Fecha: 28/06 probabilidad: **Poco Probable**, nos dieron datos.
- Fecha: 01/07 impacto: **Importante**, al analizarlos, se notaron fallas tanto de integridad como de estructura de los mismos. Surgieron nuevos campos y los campos existentes se eliminaron. Se pidió reestructuración de los datos
- Fecha: 22/07 probabilidad: **Problema**, aún no tenemos los datos de forma correcta, se generan datos ficticios.

R15: Cesión de pago de las licencias por parte de Evimed (05/05/23)

- Probabilidad: **Poco probable**, dado que Evimed está más que conforme con los avances en lo pedido por ellos, expresando regularmente su visto bueno sobre el estado del proyecto.
- Impacto: **Catastrófico**, esto implicaría que no podríamos continuar la configuración de Odoo.
- Fecha: 16/05: Probabilidad: **Imposible**, dado que el desarrollo fue finalizado, no hay razón ni atraso posible para generar una disconformidad en el cliente. Impacto: **Medio**

7.4.2. Riesgos de producto

R01: No tener experiencia en el lenguaje de implementación del CRM y las integraciones que debamos hacer. (15/01/22)

- Probabilidad: **Altamente probable**, ya que por pedido explícito del cliente, debe ser PHP o Python y ninguno de los desarrolladores posee experiencia en ellos.
- Impacto: **Importante**

R10: Falta de conocimiento en tecnologías nuevas a utilizar (25/05/23)

- Probabilidad: **Altamente probable**, estamos buscando alternativas innovadoras para aumentar el alcance del proyecto, lo cual va acompañado de tecnologías desconocidas para el equipo
- Impacto: **Importante**, ya que esto haría que el equipo pierda tiempo, que es nuestro recurso más preciado ahora mismo
- Fecha: 24/06 Probabilidad: **poco probable**, estudiamos las tecnologías a utilizar y tenemos mas expertise, sigue siendo posible porque puede surgir algo nuevo
- Fecha: 10/07 Probabilidad: **Imposible**. El equipo superó las barreras técnicas de las nuevas tecnologías sin grandes inconvenientes o impacto en el producto final.

R13: Interrupción momentánea de algún servicio de terceros necesario al desarrollar (Servidores de Odoo, OpenAI API) (24/06/23)

- Probabilidad: **Poco probable**, dado que los servicios que consumimos son servicios de gran calibre y tienen alta disponibilidad.
- Impacto: **Importante**, dado que interrumpe el desarrollo o prueba de la funcionalidad que dependa de ese servicio

R14: Interrupción momentánea de algún servicio de terceros necesario al momento de una demo o presentación (24/06/23)

- Probabilidad: **Poco probable**

- Impacto: **Crítico**, dado que nos podría imposibilitar mostrar la correcta funcionalidad del sistema

7.4.3. Riesgos personales

R06: Problemas personales inesperados de alguno de los integrantes que reduzcan su capacidad horaria temporalmente (15/01/23)

- Probabilidad: **Probable**, dado que todos tenemos otras responsabilidades y pueden llegar a interferir levemente en algún caso.
- Impacto: **Importante**, debido a que se puede planificar alrededor de esas responsabilidades y mitigar el impacto lo más posible.
- Actualización 29/03: Probabilidad: **Problema**, un integrante del equipo debe tener una cirugía con aproximadamente dos semanas de recuperación el 14/04.

R07: Problemas personales inesperados de alguno de los integrantes que imposibiliten su trabajo en el proyecto por un tiempo limitado o indefinidamente (15/01/23)

- Probabilidad: **Poco probable**, ya que todos estamos bien de salud actualmente.
- Impacto: **Catastrófico**, esto reduciría en hasta un tercio la productividad del equipo probablemente necesitando una reestructuración del proceso de ingeniería y el calendario, junto al alcance del proyecto.

7.4.4. Riesgos académicos

R09: Alcance del proyecto insuficiente académicamente (26/05/23)

- Probabilidad: **Probable**, las features pedidas por el cliente actualmente no son suficientes para el alcance de un proyecto de tesis de ingeniería
- Impacto: **Catastrófico**
- Fecha: 26/05

- Fecha: 18/06, Probabilidad: **Poco probable**, están definidas y en desarrollo las features a realizar. No tenemos feedback alguno de la correctitud del nuevo alcance, por lo que la probabilidad disminuye un poco, pero no totalmente.

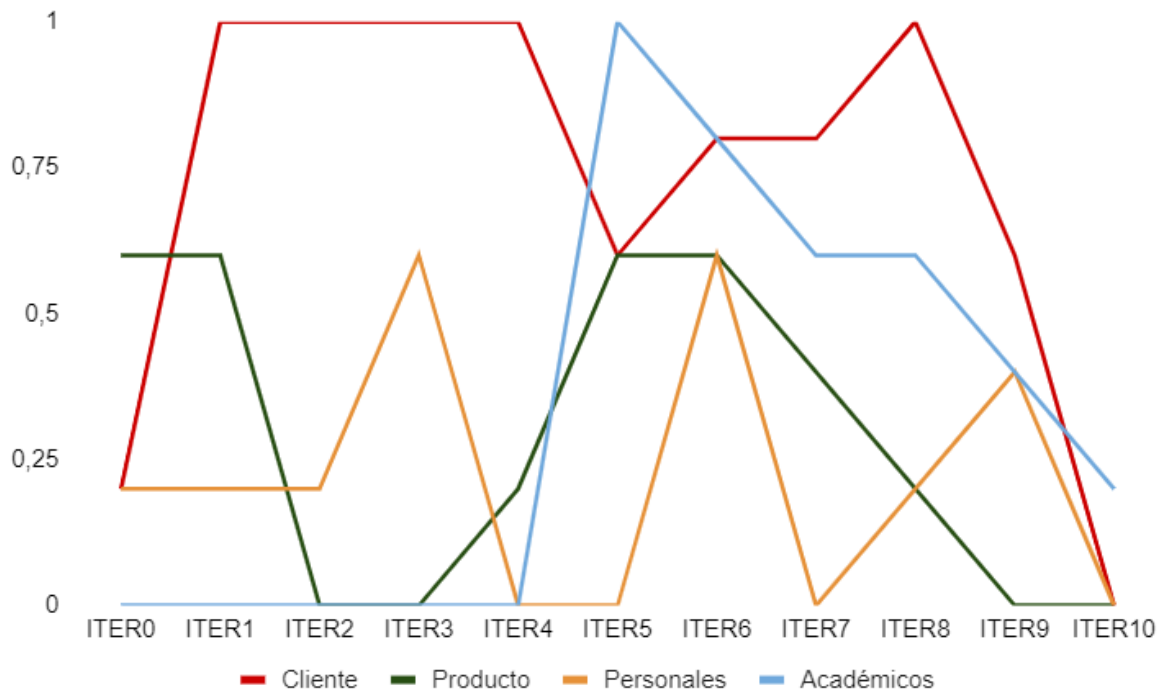


Gráfico 2: Riesgos

7.5. Gestión de la comunicación

La comunicación jugó un rol muy importante en lo que es el proyecto. Desde un comienzo se definieron los canales, tanto para la comunicación interna del equipo como para la comunicación del cliente.

Se optaron por realizar reuniones online, dado que el horario que les convenía a todos era en horario laboral, y tener que desplazarse hasta un lugar para tener reuniones presenciales implicaba un enorme esfuerzo extra que era difícil de llevar a cabo todas las semanas.

7.5.1. Comunicación interna

A lo largo del proyecto, el equipo mantuvo una comunicación fluida por varios canales. Primero, como se mencionó anteriormente, se manejaban reuniones *bi-weekly* mediante

Google Meet²³ con el fin de comentar lo que se había realizado, lo planificado y si habían bloqueos. Además, el equipo se mantenía sincronizado diariamente mediante WhatsApp²⁴. Se tenían dos grupos, uno para las conversaciones informales, planificación y conversaciones espontáneas, y otro para enviar información específica y formal que se quería mantener aparte de la conversación diaria.

Además de reunirse mediante reuniones virtuales, el equipo consideró importante reunirse presencialmente para situaciones en específico, como largas jornadas de desarrollo los fines de semana, prácticas para las revisiones, sesiones de ideaciones, etc. Esto fomentaba no sólo en la comunicación, sino también en el bienestar del equipo.

Para mantener la información organizada, se tenía una carpeta de Google Drive²⁵, donde el equipo iba comunicando qué agregaba y en qué carpeta, y así unificando los lugares donde guardar la información.

7.5.2. Comunicación con el cliente

Las reuniones con el cliente fueron llevadas a cabo mediante la aplicación Google Meet. Se agendaron reuniones quincenales (cada dos miércoles) para reservar el espacio. En varias de ellas se fue con una agenda para tratar puntos en específico, en otras simplemente eran una muestra o sincronización del estado del proyecto.

A su vez, se contaba con un canal de Google Chat²⁶ para comunicaciones en general, como planificación de reuniones y pedido de datos.

Cuando se comenzó la segunda parte (implantación de Odoo), se tuvieron algunos problemas de comunicación. Evimed tenía el curso más grande del año, por lo que todos los equipos estaban enfocados en atender la demanda de sus clientes. Esto fue un desafío muy grande para el equipo, dado que estaba atrasando el cronograma estipulado.

Se planteó al cliente la oportunidad de cambiar el medio de comunicación, dado que por Google Chat no estaba siendo eficiente, obteniendo respuestas esporádicas e incompletas, dificultando el avance del proyecto. Ellos decidieron no cambiar los medios de comunicación, ya que era

²³ <https://meet.google.com/>

²⁴ <https://www.whatsapp.com/>

²⁵ <https://www.google.com/intl/es/drive/>

²⁶ <https://mail.google.com/chat/u/0/#onboarding>

fácil para ellos acceder a estos medios y que Google Chat les enviaba un mail cada vez que les escribíamos.

También se manejaba la comunicación vía Google Mail (Gmail)²⁷. Este espacio estaba enfocado a comunicaciones formales y entrega de documentos. Aquí se planteó la preocupación de las demoras que se estaban teniendo y fue que se obtuvo la explicación del por qué de la situación.

Pese a estas complicaciones, el equipo logró avanzar por su cuenta y mejorar su proactividad. No se esperó a tener toda la información que se le iba pidiendo al cliente, sino que se tomó un enfoque de generarla así se mitigaba ese riesgo.

7.5.3. Comunicación con el tutor

Las reuniones con el tutor también fueron llevadas a cabo mediante Google Meet. Se marcaron quincenalmente, al igual que con el cliente. En un comienzo era los miércoles, a la misma hora que con el cliente pero en el miércoles intercalado, luego se movieron a los jueves por la disponibilidad de cada uno. En los últimos dos meses, se cambiaron las reuniones a semanales todos los viernes, generando un gran apoyo del lado del tutor hacia el equipo.

Estas reuniones fueron de gran ayuda para el equipo. Existieron varios momentos de grandes incertidumbres, desde cómo enfrentar el desconocimiento de cómo realizar una consultoría, hasta los momentos en que se perdía la comunicación con el cliente y no se sabía cómo avanzar, el tutor estuvo en cada momento acompañando al equipo y dando herramientas para poder salir adelante, intentando no involucrarse inmediatamente para que ellos pudieran aprender del proceso.

También se tenía un grupo de WhatsApp, en el que se iban haciendo sincronizaciones, planteando dudas y dando feedback de lo que se había avanzado. La comunicación siempre fue fluida y el tutor se encontraba siempre disponible para el equipo.

7.6. Métricas de gestión

Como se comentó en la sección [7.3. Gestión del esfuerzo](#), el tiempo del proyecto que quedó disponible para implantación y desarrollo fue de ocho iteraciones. En estas iteraciones, se tuvo

²⁷ <https://www.google.com/intl/es/gmail/about/>

que administrar para que las tres épicas (implantación, Adapter y extensión de Google Chrome) se llevaran a cabo de forma ordenada y eficiente.

En principio, el equipo trabajó en conjunto en la implantación del CRM. Para esto, generó un *backlog* del producto el cual fue convirtiéndose en *backlog* de la iteración a medida que avanzaba el proyecto.

Esto se pudo realizar hasta el momento que la etapa de ideación fuera concluida, ya que a partir de allí se generaron las otras dos épicas y el equipo tendría que dividirse.

Como se muestra en el Gráfico 3, en las primeras tres iteraciones el equipo estaba enfocado exclusivamente en la implantación del CRM. Luego, en las siguientes iteraciones el rendimiento bajó dado que sólo una persona continuó con la configuración, mientras que el resto se migró al desarrollo de los productos satélites.

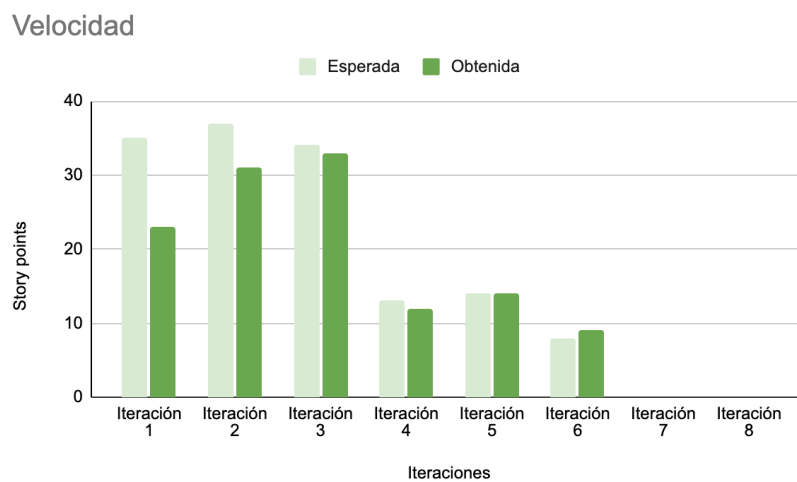


Gráfico 3: Velocidades de la implantación del CRM

Así como el rendimiento bajó respecto a las velocidades de implantación, se compensa en las épicas del Adapter (Gráfico 4) y extensión de Google Chrome (Gráfico 5).

Velocidad

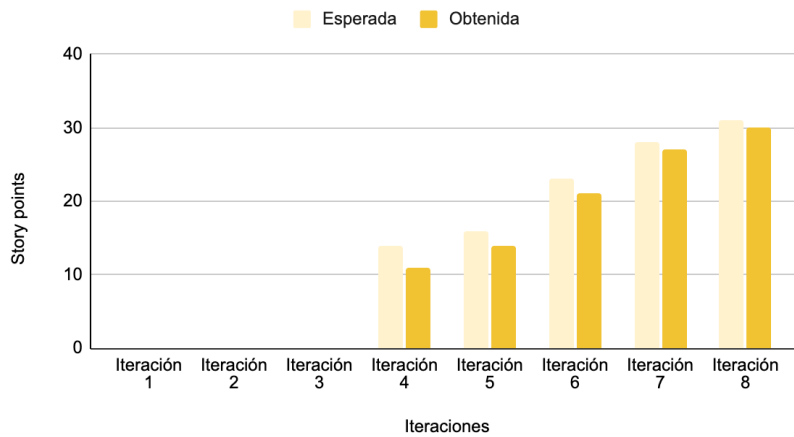


Gráfico 4: Velocidades del Adapter

¿Por qué en las velocidades del Adapter se fueron aumentando los *story points*? Porque a medida que la implantación del CRM se iba finalizando, la persona del equipo que estaba encargada de llevar a cabo la épica fue tomando tareas del Adapter, dado que también se necesitaba mucho conocimiento de la implantación para poder interactuar con la API de Odo.

Luego, para la épica de la extensión de Google Chrome, se puede ver que se comenzó bastante optimista con los story points, pero luego se fue equilibrando y encontrando un balance para esa persona.

Velocidad

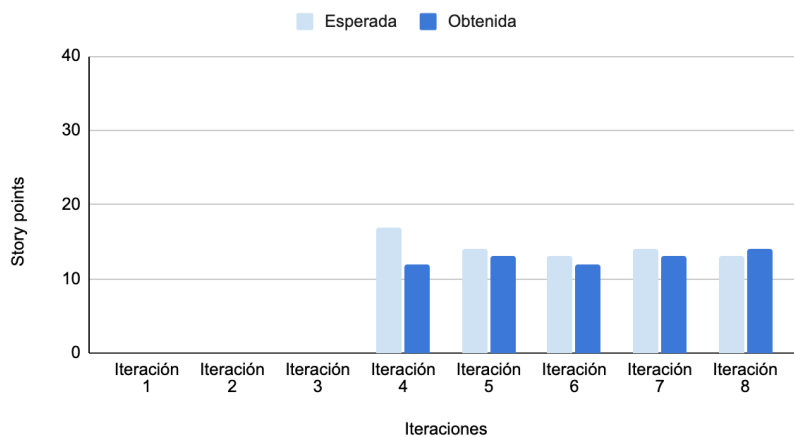


Gráfico 5: Velocidades de la extensión de Google Chrome

7.7. Gestión de la configuración

Esta sección se enfocará en la gestión de la configuración, analizando diferencialmente los elementos de *software* de los de documentación. También se profundizará en las tácticas implementadas para controlar los cambios. Cabe destacar que la integración de cambios resultó fundamental para garantizar la calidad del producto final entregado.

7.7.1. Elementos de configuración

Los elementos de configuración fueron categorizados según si son elementos de *software* o documentación.

Software:

- Código del Adapter
- Código de la extensión de Chrome

Documentación:

- Informe de análisis de CRMs
- Manual del desarrollador para Evimed
- Criterios de selección de CRMs
- Documento de gestión de riesgos
- Informes de revisiones académicas
- Estructura de modelos de datos
- Conjuntos de datos de prueba
- El presente documento

A continuación se profundizará sobre las diferentes herramientas y procesos aplicados para gestionar los diferentes elementos.

7.7.1.1. Elementos de software

7.7.1.1.1. Organización de repositorios

Para la gestión del código fuente se utilizó Git, con GitHub²⁸ como plataforma elegida por el equipo.

Se generaron dos repositorios, uno para el Adapter y uno para la extensión de Chrome. De esta forma se podía mantener separados ambos productos, que aunque pertenezcan al mismo proyecto, tienen diferentes objetivos y están desarrollados y desplegados con tecnologías diferentes.

²⁸ <https://github.com/>

7.7.1.1.2. Uso de ramas

Para una correcta separación de responsabilidades y una colaboración eficiente, se hizo uso del concepto de *branching*, que implica la definición de diferentes ramas paralelas al momento de desarrollar para evitar conflictos entre el equipo al introducir nuevo código al mismo archivo concurrentemente. Para el proceso de definición de las ramas se utilizó *Git Flow* [20], esto consiste en que de la rama principal *main* se realiza una rama *develop*, de la cual derivan las ramas en las que el equipo va trabajando cada funcionalidad. *Git flow* también especifica la nomenclatura de las ramas la cual es la siguiente:

categoría/nombre-funcionalidad

Siendo las categorías *feature*, *hotfix*, *bugfix* y *test*.

Cabe destacar que el nombre de la funcionalidad tiene que ser descriptivo de la misma, para el entendimiento de los desarrolladores y que no debe necesariamente estar escrito en *kebab-case* como el ejemplo sino que es a preferencia del equipo, aunque es importante que sea consistente dentro del repositorio para evitar confusiones.

Una vez finalizado el desarrollo correspondiente a la rama, se crea un *pull request*, comparando la rama padre con la actualización que se quiere hacer. Se definió como requisito para realizar el *merge* y cerrar el *pull request* la aprobación de al menos otro miembro del equipo de desarrollo, fomentando la revisión interna y regular del código entre los diferentes miembros, buscando minimizar los errores.

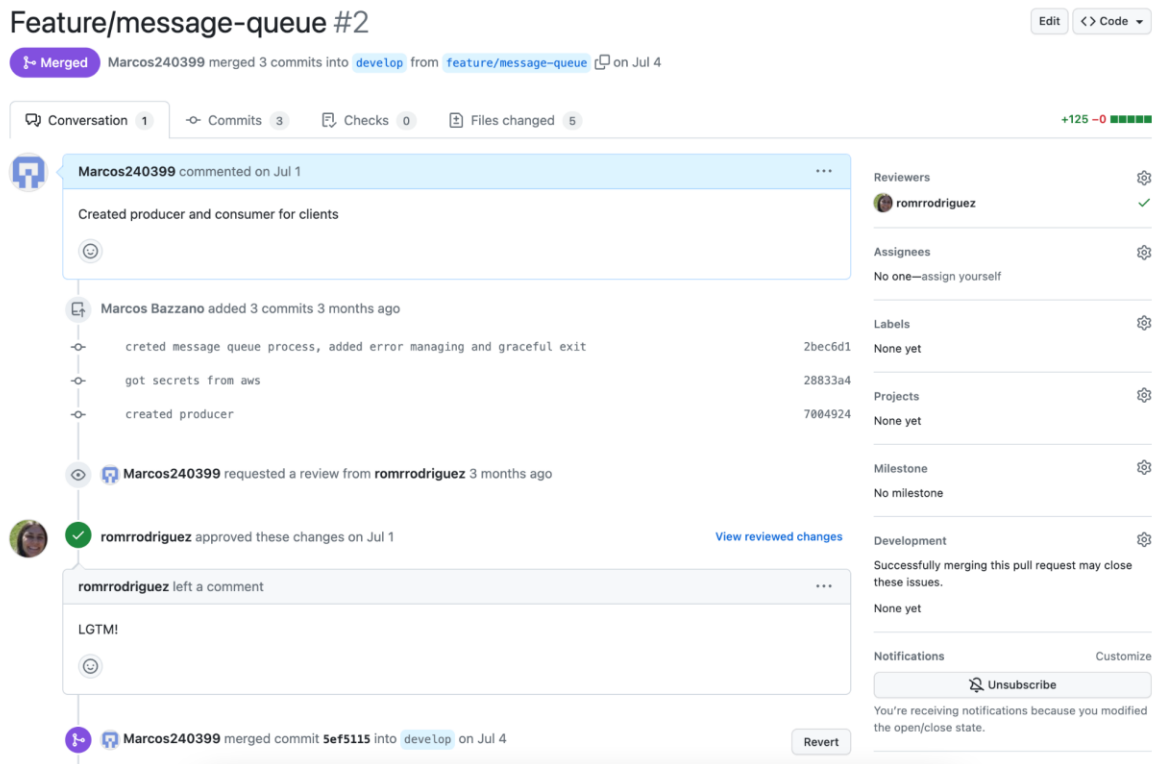


Figura 46: Ejemplo de *Pull request*

7.7.1.2. Elementos de documentación

Toda la documentación se encuentra en una carpeta de Google Drive compartida entre los integrantes del equipo y el tutor. Dentro de la misma, existe otra carpeta compartida también con Evimed, destinada para el compartimiento de documentos del equipo a Evimed y viceversa. Todos los documentos entregables fueron realizados en Google Docs.

Tipo ▾ Personas ▾ Modificado ▾

























Nombre ▾	Propietario	Última modifi... ▾	Tamaño de a	:
 Informe de revisiones	 Juan Andrés ...	25 may 2023	—	:
 Cliente	 Juan Andrés ...	28 abr 2023	—	:
 Chrome Extension	 Juan Andrés ...	8 ago 2023	—	:
 3era Revisión	 Juan Andrés ...	2 ago 2023	—	:
 Template Documentación (luego de partir) 👥	 Juan Andrés ...	20 ago 2023	77 kB	:
 Template Documentación (antes de partir) 👥	 Juan Andrés ...	15 ago 2023	28 kB	:
 Retrospectivas 👥	 Romina Rodri...	25 may 2023	3 kB	:
 Redefinición del Alcance 👥	 Juan Andrés ...	24 ago 2023	348 kB	:
 paisesPersonasods.ods 👥	 Juan Andrés ...	11 abr 2023	3,4 MB	:
 Metricas 👥	 Juan Andrés ...	5 ago 2023	1 kB	:
 Mapa Clientes interactivo 👥	 Juan Andrés ...	6 may 2023	1 kB	:
 Entendiendo un proyecto de Chrome extension 👥	 Juan Andrés ...	25 jun 2023	1 kB	:

Figura 47: Carpeta de Google Drive del equipo

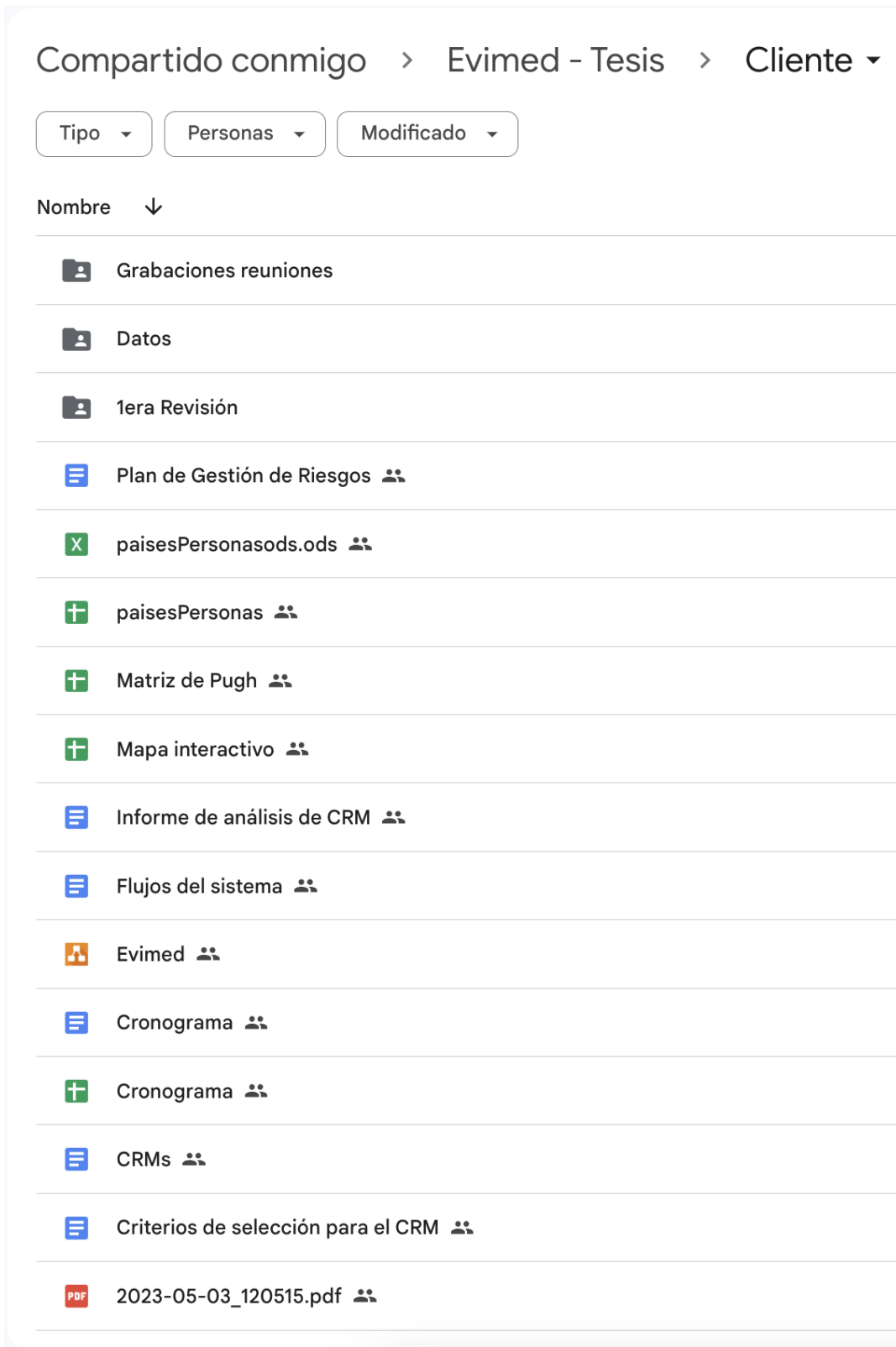


Figura 48: Carpeta de Google Drive compartida con Evimed

Para el desarrollo de los tres diferentes entregables a lo largo del proyecto, se utilizaron diferentes metodologías de trabajo para gestionar los documentos.

Para la realización del Informe de análisis, se efectuó la metodología especificada en la sección [4.1.2. Operacionalización](#).

Tanto el manual del desarrollador como este mismo documento, se escribió siguiendo el siguiente proceso.

Se comenzó mediante la estructuración del documento, generando un esqueleto de secciones inicial según el contenido que se quería reflejar. En esta etapa se hizo especial énfasis en los formatos especificados en el documento 302, explicado en la sección [7.8.1.2. Estándares de documentación](#). A su vez, se generó un archivo de seguimiento del documento, cuya función fue el mantenimiento del estado de cada sección del documento, junto a su responsable y su fecha límite establecida o *deadline*.

Para el proceso de escritura, cada miembro del equipo fue tomando secciones para escribir. Cuando alguien termina una sección, se cambia el estado de la misma a “Revisión” y es revisada por otro miembro del equipo o por el tutor, quien realiza comentarios en la sección del documento y cambia el estado de la misma a “Corregir”. Una vez corregidos, se resuelven los comentarios y en caso de haber agregado nuevo contenido como parte de una corrección, se vuelve a pasar al estado “Revisión”. En caso contrario, la sección pasa a estado “*Done*”.

Los posibles estados de cada sección son los siguientes:

- To do: Aún no se comenzó la escritura de la sección.
- WIP: La sección está actualmente siendo escrita.
- Revisión: La sección está esperando ser revisada por otro miembro del equipo o el tutor.
- Corregir: La sección fue revisada y tiene comentarios a corregir.
- *Done*: La sección fue revisada y corregida al menos una vez y no tiene nuevas correcciones.

1	Sección	Estado	Deadline	Responsable
42	4.3.1. Relevamiento inicial	DONE	7/9/2023	Romi
43	4.3.2. Especificación de requerimientos	DONE	7/9/2023	Romi
44	4.3.3. Validación de criterios de selección	DONE	7/9/2023	Romi
45	4.4. Informe de análisis de CRM	WIP	22/9/2023	Romi
46	4.4.1. Principales resultados	WIP	22/9/2023	Romi
47	4.4.2. Validación y aceptación de resultados	WIP	22/9/2023	Romi
48	5. Segunda etapa: Implantación de Odoo CRM	REVISIÓN	22/9/2023	Romi
49	5.1. Proceso	REVISIÓN	22/9/2023	Romi
50	5.2. Ingeniería de requerimientos	REVISIÓN	22/9/2023	Romi
51	5.2.1. Objetivos	REVISIÓN	22/9/2023	Romi
52	5.2.2. Descripción general	REVISIÓN	22/9/2023	Romi
53	5.2.3. Metodologías utilizadas	REVISIÓN	22/9/2023	Romi
54	5.2.4. Resultados	REVISIÓN	22/9/2023	Romi
55	5.2.5. Reflexiones del proceso	REVISIÓN	22/9/2023	Romi
56	5.3. Requerimientos funcionales	CORREGIR	22/9/2023	Romi
57	5.4. Requerimientos no funcionales	CORREGIR	22/9/2023	Romi
58	5.5. Restricciones	CORREGIR	22/9/2023	Romi
59	5.6. Arquitectura / Modelado de datos	REVISIÓN	22/9/2023	Romi
60	5.7. Validación y aceptación de resultados	REVISIÓN	22/9/2023	Romi
61	5.8. Reflexiones del proceso	REVISIÓN	22/9/2023	Romi
62	6. Tercera etapa: Desarrollo de productos satélites	DONE	8/9/2023	Marcos
63	6.1. Introducción	DONE	8/9/2023	Marcos
64	6.2. Ingeniería de requerimientos	DONE	8/9/2023	Marcos
65	6.2.1. Objetivos	DONE	8/9/2023	Marcos
66	6.2.2. Descripción general del proceso	DONE	8/9/2023	Marcos
67	6.2.3. Metodologías utilizadas	DONE	8/9/2023	Marcos
68	6.2.3.1. Empatizar	DONE	8/9/2023	Marcos

Figura 49: Documento de gestión de las secciones

7.8. Aseguramiento de la calidad

En esta sección se detallarán tareas y actividades realizadas por el equipo para lograr asegurar la calidad del producto de software.

7.8.1. Estándares

En cuanto a los estándares de calidad, el equipo estableció una diferencia en cuanto a los estándares de codificación y estándares de documentación. Los estándares de codificación son pautas establecidas por el equipo adoptados por la industria en cuanto a la construcción y desarrollo de software para lograr productos mantenibles y de buena calidad. Los estándares de documentación son fijados por la Universidad ORT Uruguay para la entrega formal del proyecto.

7.8.1.1. Estándares de codificación

En este apartado se comentarán los estándares de codificación llevados a cabo por el equipo, de esta manera el equipo se aseguraba de ir en busca de un código de calidad, escalable a futuro y mantenible.

- **Clean Code:** es un estándar ampliamente utilizado en la industria basado en el libro Clean Code de Robert Martin [21]. El contenido de este libro fue introducido en materias intermedias en la carrera por el cual el equipo considera lo tiene asimilado.
- **Refactoring:** durante el proceso de desarrollo, surgen oportunidades de mejora y optimización del código existente mediante un proceso llamado *refactoring*. El proceso implica revisar y ajustar elementos del código existente en busca de mejorar la calidad y evitar la repetición de código. Esto se explica en el libro de Martin Fowler y Kent Beck llamado “Refactoring: Improving the Design of Existing Code” [22].
- **REST:** es el acrónimo en inglés para referirse a *Representational State Transfer* [23]. Dicho estándar refiere a un estilo de arquitectura para las APIs desarrolladas. Al seguir este estándar adoptado por la industria, se buscó que las APIs fueran fáciles de entender, utilizar y mantener.
- **SOLID:** los principios SOLID [24] son una buena práctica a la hora de codificar ya que nos habla de una serie de principios para escribir un código limpio, flexible, escalable, reutilizable y mantenible. Los mismos fueron introducidos al equipo en etapas tempranas de la carrera por lo que el equipo los tiene incorporados y presentes
- **Linters:** un *linter* es un *software* que ayuda al equipo a encontrar y resolver problemas con el código. Resuelve problemas de formateo, de sintaxis, muestra errores cuando no se siguen buenas prácticas, provee sugerencias para mejorar el código y puede mantener un estilo consistente o reforzar reglas internas del propio código. En el caso de Odo CRM Marketing Assistant se utilizó ESLint²⁹ y en el caso del Adapter se utilizó una extensión para el editor de texto Visual Studio Code llamada Python de Microsoft en la cual tenía un linter embebido.

²⁹ <https://eslint.org/>

Adicionalmente a dichos estándares de codificación, también se siguieron buenas prácticas en cuanto a la creación de *pull requests* para que antes de mergear la funcionalidad a la rama *develop* se realice una *code review* por otro desarrollador del equipo para asegurarse la calidad del código generado, poder sugerir cambios si algo no está claro y que el integrante extra analizando el código ajeno pueda descubrir *bugs* que no fueron vistos por el integrante que desarrolló la funcionalidad. Luego de que el revisor verificara que todo esté bien, el *pull request* era aprobado y el desarrollador encargado de crear la funcionalidad entonces podía *mergear* la misma a la rama *develop*. En la Figura 50 podemos ver el proceso descrito anteriormente.

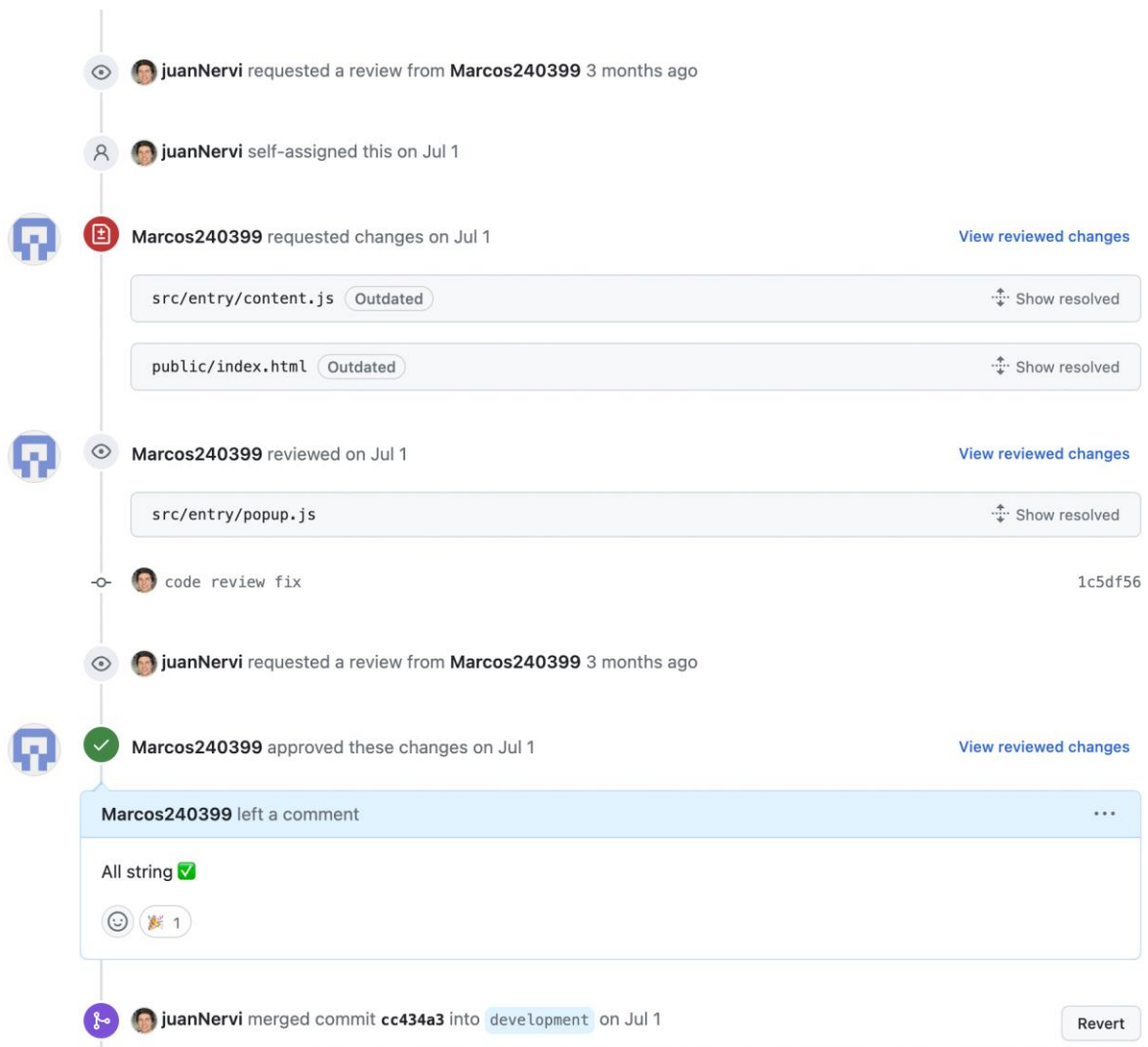


Figura 50: Proceso de revisión de *pull request*

7.8.1.2. Estándares de documentación

Para asegurar la calidad de la documentación, como se mencionó en la sección [7.8.1. Estándares](#), el equipo utilizó los documentos brindados por la Universidad ORT Uruguay para los proyectos finales de carrera.

- **Documento 302:** este documento corresponde a las normas específicas para la presentación de trabajos finales de carrera de la Facultad de Ingeniería excepto Biotecnología [25].
- **Documento 303:** este documento corresponde a la hoja de verificación de pautas de presentación de trabajos finales de carreras de la Facultad de Ingeniería [26].
- **Documento 304:** en este documento se encuentran las normas para el desarrollo de trabajos finales de carrera [27].
- **Documento 306:** este documento corresponde a la orientación para títulos, resúmenes o abstracts e informes de corrección de trabajos finales de carrera [28].

7.8.2. Pruebas

El equipo de desarrollo decidió enfocarse en realizar pruebas funcionales o exploratorias, pruebas de performance y pruebas de aceptación del usuario, de las cuales las pruebas de performance y las pruebas de aceptación con usuarios no se llevaron a cabo. Las primeras debido a que hubo una propuesta por parte del equipo de desarrollo que se tomó como válida por el equipo de Evimed y las segundas por falta de tiempo y recursos humanos por parte de Evimed. A continuación, se describen los distintos tipos de pruebas pensadas por el equipo.

Pruebas funcionales o exploratorias

Las pruebas funcionales se llevaron a cabo en los productos satélite, Adapter y Odo CRM Marketing Assistant (Extensión de Google Chrome). No se realizaron estas pruebas en Odo CRM dado que se entiende que al ser un software externo, las funcionalidades allí establecidas ya cuentan con este tipo de pruebas.

El equipo para realizar las pruebas funcionales del Adapter realizó una colección de pruebas en Postman³⁰ (Figura 51) a modo de probar el correcto funcionamiento de la API. La colección se encuentra dividida en diferentes carpetas. Existe una carpeta por entidad y en ellas se encuentran todos los verbos HTTP de las operaciones básicas CRUD (por sus siglas en inglés, referido a Crear, Leer, Actualizar y Borrar). El proceso que siguió el equipo fue la verificación de la correctitud en el alta, baja, modificación o lectura de cualquiera entidad contra la información disponible en Odoo CRM.

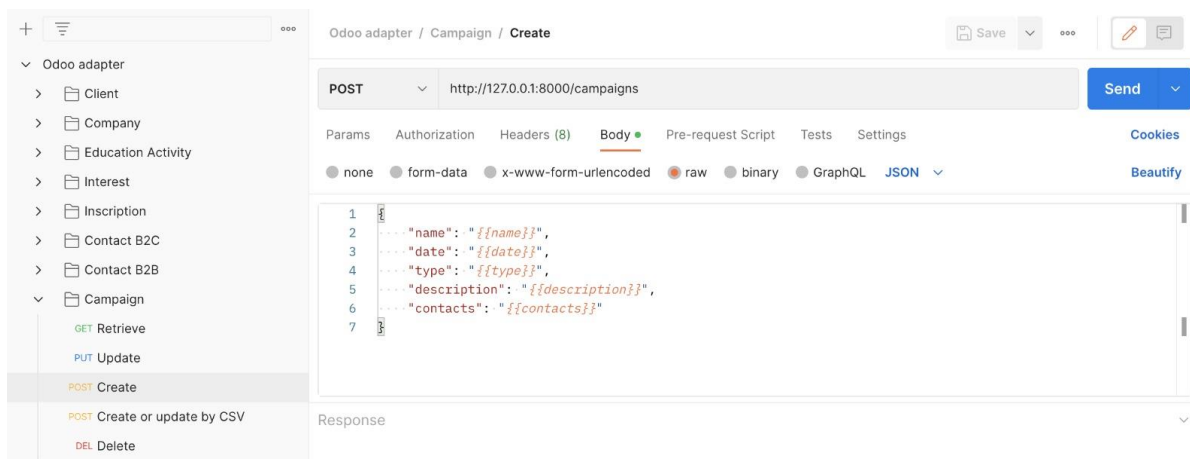


Figura 51: Ejemplo de colección de pruebas de Postman

Para las pruebas funcionales en Odoo CRM Marketing Assistant, se realizaron pruebas exploratorias donde la misión fue encontrar errores. Se probó la correctitud de la extensión una vez deployada en Chrome Web Store, en dos sistemas operativos diferentes, Windows 11 y macOS Monterey. De encontrar defectos, el equipo los registraba en Jira para la correcta gestión.

Pruebas de performance

Al comienzo del proyecto el cliente Evimed dejó en claro que uno de los aspectos importantes a cubrir por el equipo de desarrollo era poder medir cómo se comporta la solución en cuanto a las cargas de clientes, inscripciones y demás entidades. Entonces el equipo en una primera instancia decidió que para realizar dichas pruebas se utilizaría Apache JMeter³¹ para realizar *tests* de estrés dado que se entendía que el CRM elegido debía de contener la información actualizada de los sistemas de Evimed una vez al día (mediante un *scheduler*). Luego, el equipo

³⁰ <https://www.postman.com/>

³¹ <https://jmeter.apache.org/>

introdujo tecnología (*Amazon SQS*) para no tener que realizar dichas tareas que demandan tiempo, contienen grandes cantidades de datos y que por lo tanto pueden traducirse en pérdida de información. Al introducir esta funcionalidad, la carga masiva de datos solo se realizaría una vez, es decir, inicialmente. Luego, el sistema se mantendrá con dicha tecnología haciendo que el traspaso de información a Odoo CRM sea instantánea lo que el equipo de desarrollo y Evimed creen es satisfactorio.

Pruebas de Aceptación de Usuario (UAT)

Como se introdujo anteriormente, estas pruebas no fueron llevadas a cabo por cuestiones de tiempo y de recursos humanos por parte de Evimed (la falta de un actor clave del área de marketing). El equipo de desarrollo priorizó la realización de las pruebas funcionales comentadas anteriormente antes que esta última, dado a que se cree es más importante entregar un *software* con la mínima cantidad de errores posibles y performante y de esa manera cumplir con los objetivos del cliente comentados en la sección [2.3. Objetivos del cliente](#).

Si bien Evimed fue acompañando el proceso de integración de Odoo CRM mediante la ceremonia de *Review* llevada a cabo por el equipo, se entiende que haber realizado este tipo de pruebas hubiera aportado a entregar un *software* de mejor calidad. Dado a que no es lo mismo la utilización que se le dio por parte del equipo de desarrollo al que se le dará por el equipo de Evimed probando reflejar la información de sus sistemas actuales en el CRM. De allí, se podría haber trabajado en los hallazgos reportados y realizar las optimizaciones y correcciones necesarias, para que el software entregado, se adapte aún más a la realidad de la empresa.

8. Conclusiones y lecciones aprendidas

8.1. Análisis de objetivos planteados

En esta sección se analizarán los objetivos planteados por el equipo en la sección [1.4. Objetivos](#), evaluando su cumplimiento.

8.1.1. Objetivos académicos

- **Aprobar el proyecto final de carrera:** No es posible demostrar si este objetivo fue cumplido ya que al momento de la escritura de este documento se desconoce cuál será el resultado que tendrá la evaluación final. De todas formas, el equipo concluye que hizo todo lo que podía hacer para lograr este objetivo, encontrándose satisfechos con el resultado obtenido.
- **Relevar 12 CRMs:** El equipo realizó la relevación y análisis de 11 CRMs distintos. A pesar de esto, el número elegido como objetivo fue hecho arbitrariamente y se logró aprender sobre los diferentes CRMs del mercado, sus puntos de venta y sus funcionalidades diferenciales. Este conocimiento fue muy útil al momento de realizar la [6.2. Ingeniería de requerimientos](#) de los productos satélites. Debido a esto y a que el número de CRMs relevados fue 11, como se ve en la sección [4.3.2. Especificación de requerimientos](#), se considera el objetivo como cumplido.

8.1.2. Objetivos personales

- **Aprender a utilizar nuevas tecnologías:** Dado que la totalidad del Adapter está codificado en Python, que el equipo desarrolló una extensión de Google Chrome agregando valor con la API *chat completions* de GPT, se considera que el equipo afrontó el desafío de forma correcta y logró cumplir el objetivo planteado.
- **Aportar valor a la operativa de Evimed:** Aún no es posible demostrar si este objetivo fue cumplido debido a que Evimed aún no está utilizando Odoon ni los productos satélites. De todas maneras, dado la carta presentada en el anexo [10.4. Carta de conformidad del cliente](#), el equipo cree que el trabajo realizado les generará valor cumpliendo con este objetivo.

- **Obtener experiencia del rubro y dirección empresarial de la directiva de Evimed:** A pesar de que no se aprendió específicamente de la dirección empresarial, el equipo está satisfecho con la cantidad de conocimiento del rubro de la enseñanza, como de los flujos diarios de una empresa, a partir del estudio de los diferentes casos de uso de los CRMs.

8.1.3. Objetivos de producto

- **Integrar un producto que satisfaga las necesidades del cliente:** Los productos entregados a Evimed cumplen con la totalidad de los requerimientos funcionales y no funcionales relevados. Además, tanto la selección de Odoos como CRM como las funcionalidades realizadas para complementarlo fueron enseñadas al cliente, el cual mostró su contento con el resultado.
- **Cumplir con los plazos establecidos para las etapas y entregables acordados:** A lo largo del proyecto, el equipo cumplió con todos los plazos establecidos con el cliente para las entregas y completitud de funcionalidades.

8.1.4. Objetivos de cliente

Estos objetivos fueron planteados en la sección [2.3 Objetivos del cliente](#).

- **Solucionar el problema de la fragmentación de la base de datos:** Este objetivo fue cumplido, dado que la base de datos de Evimed se encuentra totalmente centralizada en Odoos CRM.
- **Mejorar la segmentación de clientes:** Debido a la existencia de datos dentro de Odoos CRM, se puede utilizar su funcionalidad de segmentación integrada, para resolver segmentaciones que antes requerían múltiples *queries* de SQL en un solo filtrado.
- **Registrar las comunicaciones entre clientes y Evimed:** Este objetivo se logró al crear y almacenar las entidades “Contact_B2B” y “Contact_B2C” en el nuevo modelo de datos representado en Odoos.
- **Generar pruebas de performance:** Este objetivo se vio modificado, debido a la propuesta del equipo de desarrollo de incluir tecnología de colas de mensajes (*Amazon SQS*) para no tener que depender de un *scheduler* donde se pudiera ver comprometida

la performance del Adapter. Si bien el objetivo se vio modificado porque planteaba la generación de pruebas de performance, el equipo y Evimed entienden que fue cubierto.

8.2. Lecciones aprendidas

En esta sección se reflexionará sobre los diferentes aprendizajes y lecciones que otorgó el proyecto.

8.2.1. Aprendizajes de tecnología

La naturaleza inusual y evolutiva del proyecto permitió al equipo llevarse consigo mucha experiencia de diferentes herramientas y tecnologías, para el equipo esto es muy valioso, dado que pocas veces se tiene la posibilidad de trabajar en un proyecto con un cliente real, de comenzar a trabajar con tecnologías que uno no conoce y aprenderlas a medida avanza el mismo.

Las principales tecnologías en las que se ganó experiencia fueron:

- **Inteligencia Artificial, uso de LLMs (*Large Language Models*)**

Al realizar las integraciones de la extensión de Google Chrome con la API de GPT (sección [6.3.2.1 Uso de la inteligencia artificial](#)), el equipo tuvo la oportunidad de aprender como funciona detrás de escena el disruptivo ChatGPT. Esta fue una gran oportunidad no solo por lo útil que fue en el proyecto, sino que es una herramienta sumamente interesante y explotable para el uso diario, tanto en lo personal como profesional.

- **Python**

Debido a una [restricción](#) que surgió del cliente, el equipo tuvo que aprender Python para desarrollar el Adapter. Esto en un comienzo fue un gran desafío, dado que es un lenguaje de programación sumamente distinto a los que el equipo acostumbra a utilizar en cuanto a estructura, sintaxis y funcionamiento general. Debido a que es un lenguaje pedido en el mercado laboral y que está estrechamente asociado a la Inteligencia Artificial, tecnología en la que el equipo encontró mucho interés, el equipo considera que valió la pena desde el punto de vista personal además del académico.

- **CRMs**

Fuera de las tecnologías que se utilizaron desde el desarrollo, se destaca el aprendizaje que se obtuvo sobre los CRMs, dado que ninguno de los integrantes del equipo sabía o había siquiera escuchado el término CRM previo a la introducción del proyecto. Mediante la investigación realizada, se obtuvo una gran cantidad de conocimiento sobre el funcionamiento y razón de ser de los CRMs, concluyendo unánimemente que son una herramienta sumamente eficaz a tener en cuenta para cualquier emprendimiento o empresa de mediana a grande escala.

8.2.2. Comunicación con el cliente

Como se comentó en la sección [7.5.2. Gestión de comunicación con el cliente](#), hubieron muchos aprendizajes respecto a la gestión con el cliente.

El equipo se llevó como una gran lección aprendida el poner fechas límites. Dar a entender los objetivos de cada iteración para que el cliente comprenda la necesidad de cada pedido, es fundamental para que se colabore.

Otra lección aprendida corresponde a la disponibilidad. El equipo mostró desde el momento inicial un calendario reflejando cuándo tenían menos disponibilidad (licencias, exámenes, parciales, etc), pero no consultó la del cliente. Esto se aprendió y se ejecutó correctamente luego de sucedido el incidente del relevamiento de requerimientos del CRM, explicado en la sección [4.4.2. Validación y aceptación de resultados](#).

Si bien el equipo se comprometió con el proyecto, cree que habiendo mejorado la escucha, proactividad y comprensión, se podrían haber mitigado algunos riesgos.

8.2.3. Uso de metodologías

Comenzando por la parte de la consultoría, el equipo siente que fue una de las mayores lecciones aprendidas. Nunca había experimentado lo que sería el hecho de realizar un informe en base a sus investigaciones, recomendar un producto y que un cliente invirtiera dinero en ello. El ida y vuelta con el precio de las licencias fue algo novedoso para el equipo. Tener que comunicar las necesidades del cliente, actuando como representante frente al equipo de Odoo se sintió como una gran lección.

Fue un proceso complejo, de mucha adaptación y la metodología Popcorn Flow que introdujo el tutor fue de gran ayuda.

Luego, con la metodología ágil, el equipo ya está más familiarizado con ella, dado que se ha tocado en distintos puntos de la carrera y además lo llevan a la práctica en su rutina laboral. Lo desafiante de esta metodología fue justamente lo mencionado anteriormente. El equipo trabaja con prácticas ágiles en su día a día, pero son todas diferentes. Se tienen algunas ceremonias en común, pero la forma de manejarlas era distinta y eso fue un desafío para el equipo.

8.2.4. Gestión de riesgos

Se tuvo un proyecto con altibajos. Desde riesgos en la comunicación hasta ausencias por enfermedad. No adaptarse no era una opción, era algo necesario para que el proyecto siguiera a flote.

Se cree que el equipo logró un gran balance. Entre los trabajos demandantes de cada uno, las ausencias por enfermedad que afectaron en gran medida y la inestabilidad del proyecto por momentos, se tuvo respuesta ágil del equipo y eso se pudo notar.

8.2.5. Gestión de las emociones

A pesar de que los miembros del equipo ya se conocían, no habían tenido la oportunidad de trabajar todos juntos. En el correr del proyecto, debido al estrés acumulado y el cansancio, surgieron algunos conflictos intragrupal que tuvieron que ser resueltos mediante el diálogo, discutiendo los intereses de cada uno. Al conversar se encontró que la principal razón de esto fue que diferentes miembros del equipo tienen características de liderazgo muy marcadas en su personalidad, lo cual causaba choques ya que ambos inconscientemente intentaban cumplir el rol.

Otro conflicto que ocurrió sobre el comienzo del proyecto, fue respecto a la diferencia de horarios habituales de productividad de los miembros del equipo, habiendo integrantes que preferían trabajar en la mañana y otros en la noche, lo cual dificultó la definición de un itinerario para las reuniones regulares.

Concluyendo, el compartir un proyecto de tanta importancia para todos por un año entero fue una experiencia muy enriquecedora, debido a que fue necesario ser flexibles para poder respetar las posiciones e ideas de todos los miembros.

Superar juntos todos los obstáculos que este proyecto implicó, resultó en una relación de colegas y amigos estrecha, lo cual genera mucha satisfacción a todos los integrantes.

8.2.6. Gestión del proyecto

Como mejora a la gestión del proyecto hubiera sido bueno desafiar la solución inicial que el cliente planteaba una vez entendidas las necesidades y problemas identificados. Ese desafío existió, pero se podría haber planteado de manera diferente. Plantear desarrollar una solución alternativa sin tener que adquirir un CRM en su totalidad, teniendo en cuenta la cantidad de funcionalidades que tienen. Para ello se podría haber realizado reuniones presenciales para ver cómo trabaja el equipo de Evimed individualmente en su día a día, entendiendo en dónde invierten mayoritariamente su tiempo tratando de optimizar dichas tareas.

Reunirse por separado con cada interesado, para que no todo pase por la gerencia en sí y obtener la mayor cantidad de información posible.

9. Referencias bibliográficas

- [1] “Laboratorio ORT Software Factory.”
<http://fi.ort.edu.uy/innovaportal/v/3393/5/fi.ort.front/inicio.html> (accessed Sep. 03, 2023).
- [2] “PopcornFlow: Continuous Evolution Through Ultra-Rapid Experimentation.”
<https://www.slideshare.net/cperrone/popcornflow-continuous-evolution-through-ultrarapid-experimentation> (accessed Sep. 04, 2023).
- [3] “LinkedIn.” <https://www.linkedin.com/pulse/discover-popcorn-flow-interview-claudio-perrone-paolo-sammicheli/> (accessed Sep. 03, 2023).
- [4] K. Nikolopoulou, “What Is Snowball Sampling?,” *Scribbr*, Aug. 17, 2022.
<https://www.scribbr.com/methodology/snowball-sampling/> (accessed Sep. 04, 2023).
- [5] H. Plattner, C. Meinel, and L. Leifer, *Design Thinking Research: Making Design Thinking Foundational*. Springer, 2015.
- [6] I. Sommerville, *Software Engineering, 9/e*. Pearson Education India, 2011.
- [7] “Chrome Extensions getting started guides,” *Chrome Developers*.
<https://developer.chrome.com/docs/extensions/mv3/getstarted/> (accessed Sep. 03, 2023).
- [8] L. Mearian, “What are LLMs, and how are they used in generative AI?,”
Computerworld, May 30, 2023. <https://www.computerworld.com/article/3697649/what-are-large-language-models-and-how-are-they-used-in-generative-ai.html> (accessed Sep. 28, 2023).
- [9] E. Alston, “What are AI hallucinations—and how do you prevent them?,” Apr. 05, 2023.
<https://zapier.com/blog/ai-hallucinations/> (accessed Sep. 21, 2023).
- [10] “Chrome Extensions architecture overview,” *Chrome Developers*.
<https://developer.chrome.com/docs/extensions/mv3/architecture-overview/> (accessed Sep. 10, 2023).
- [11] J. A. Storer, *An Introduction to Data Structures and Algorithms*. Springer Science & Business Media, 2012.

- [12] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*. Addison-Wesley, 2012.
- [13] “ArticleS.UncleBob.PrinciplesOfOod.”
<http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod> (accessed Sep. 12, 2023).
- [14] C. Bernstein and M. Cobb, “Advanced Encryption Standard (AES),” *Security*, Sep. 24, 2021. <https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard> (accessed Sep. 10, 2023).
- [15] “man-in-the-middle attack (MitM) - Glossary.”
https://csrc.nist.gov/glossary/term/man_in_the_middle_attack (accessed Sep. 10, 2023).
- [16] “Chrome Extensions: cross-origin network requests,” *Chrome Developers*.
<https://developer.chrome.com/docs/extensions/mv3/network-requests/> (accessed Sep. 11, 2023).
- [17] I. Sommerville, *Ingeniería del software*. Pearson Educación, 2005.
- [18] “Principios del Manifiesto Ágil.” <https://agilemanifesto.org/iso/es/principles.html> (accessed Sep. 23, 2023).
- [19] M. N. Adams, “Scrum Poker for Agile Projects,” *Work Life by Atlassian*, Sep. 10, 2021. <https://www.atlassian.com/blog/platform/a-brief-overview-of-planning-poker> (accessed Sep. 03, 2023).
- [20] “A successful Git branching model,” *nvie.com*. <http://nvie.com/posts/a-successful-git-branching-model/> (accessed Sep. 25, 2023).
- [21] R. C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*. Pearson Education, 2009.
- [22] P. Becker, M. Fowler, K. Beck, J. Brant, W. Opdyke, and D. Roberts, *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional, 1999.
- [23] “RESTful Web Services.” <https://www.w3schools.in/restful-web-services/intro> (accessed Sep. 25, 2023).
- [24] “Getting a SOLID start.” <https://sites.google.com/site/unclebobconsultingllc/getting-a->

solid-start (accessed Sep. 25, 2023).

- [25] Universidad ORT Uruguay, “NORMAS ESPECÍFICAS PARA LA PRESENTACIÓN DE TRABAJOS FINALES DE CARRERA (TFDC) FACULTAD DE INGENIERÍA,” Sep. 26, 2019. https://www.ort.edu.uy/innovaportal/file/95484/1/normas-especificas-para-la-presentacion-de-trabajos-finales-de-carrera-facultad-de-ingenieria-excepto-biotecnologia__documento-302.pdf (accessed Sep. 25, 2023).
- [26] Universidad ORT Uruguay, “HOJA DE VERIFICACIÓN DE FORMATO DE TRABAJOS FINALES DE CARRERAS DE LA FACULTAD DE INGENIERÍA,” Sep. 26, 2012. https://www.ort.edu.uy/innovaportal/file/95484/1/hoja-de-verificacion-de-pautas-de-presentacion-de-trabajos-finales-de-carreras-excepto-biotecnologia__documento-303.pdf (accessed Sep. 25, 2023).
- [27] Universidad ORT Uruguay, “NORMAS PARA EL DESARROLLO DE TRABAJOS FINALES DE CARRERA,” Aug. 20, 2012. https://www.ort.edu.uy/innovaportal/file/95484/1/normas-para-el-desarrollo-de-trabajos-finales-de-carrera__documento-304.pdf (accessed Sep. 25, 2023).
- [28] Universidad ORT Uruguay, ““Orientación para títulos, resúmenes o abstracts e informes de corrección de trabajos finales de carrera,”” Mar. 01, 2012. https://www.ort.edu.uy/innovaportal/file/95484/1/orientacion-para-titulos-resumenes-o-abstracts-e-informes-de-correccion-de-trabajos-finales-de-carrera__documento-306.pdf (accessed Sep. 25, 2023).

10. Anexos

10.1. Criterios de selección para el CRM

Infraestructura (5)

El lenguaje debe ser Python o Php.

El Sistema operativo debe ser Linux.

La base de datos debe ser MySQL.

Modo de licenciamiento (5)

Preferentemente que sea gratuito y open source.

Si es pago, deberá ser en función de los usuarios administradores que utilicen el CRM (por mes o por año). Que no esté basada en interacciones con los clientes ni en cantidad de clientes.

Integraciones (5)

El CRM deberá disponer de APIs que permitan la alimentación con datos de fuentes externas en tiempo real (por ejemplo: se realiza un pago, se emite un certificado, etc). La lista de sistemas externos es la siguiente:

Sistema	Tipo	Mecanismo integración
Pagos	Interno	Base de datos
Consultas de contactos	Interno	Base de datos
Certificados	Interno	Base de datos
Estadísticas académicas	Interno	Base de datos
Actividad en Campus	Interno	Base de datos
Datos personales	Interno	Base de datos
Email marketing (Icegram)	Interno	Base de datos
Procesos automáticos	Interno	Base de datos
Marketing de influencers	Interno	Base de datos

WhatsApp (WATI)	Externo	API
Soporte (FreshDesk)	Externo	API
ZOOM	Externo	API
Gestión de contratos a proveedores	A contratar	A definir

Tabla 7: Sistemas a integrar

Debe permitir la integración de correos con SMTP.

La integración con WhatsApp debe ser específicamente a través de la herramienta WATI.

La integración con ZOOM no es necesaria para este punto ya que no se sabe qué requerimientos necesitan de su integración.

Customización de datos (5)

Se debe poder modificar los campos de la entidad persona, agregando por ejemplo, especialidad, años de experiencia, entre otros. Y también la creación de nuevas entidades (por ejemplo: cursos). Deberá soportar que haya relaciones n a n.

Ej: Personas que toman cursos. Dentro de esa relación debería agregarse la beca, nota, certificado, etc.

Distinción B2B y B2C (3)

Deberá soportar distinguir entre los usuarios de tipo persona de los tipo empresa. (Esto podría ser con un simple atributo.)

Segmentación de clientes (5)

A partir de los datos se deberán poder ejecutar queries o filtros que permitan segmentar a los clientes para poder realizar marketing. Es importante también que tengan Scoring.

Ej: Mujeres mayores de 30 que estudien cardiología y hayan cursado “Curso 1”.

Interfaz modificable (1)

Se valora que el CRM otorgue la posibilidad de modificar la estructura del mismo, tanto la ubicación de las diferentes funcionalidades como la organización de cada uno de sus elementos en la pantalla (ej., ubicación de formularios y orden de campos o datos).

Soporte para mantenimiento del CRM (2)

Es beneficioso que el CRM elegido tenga una sección de soporte para desarrolladores y/o documentación, facilitando las tareas de mantenimiento correctivo-adaptativo de EviMed una vez ya haya sido entregada la integración.

Se valora la existencia de discusiones en foros existentes y activos sobre problemáticas recurrentes al momento de mantenimiento o uso de la misma.

Soporte a usuarios (4)

Aporta que tenga incorporado la gestión de consultas y soporte de los usuarios. Pudiendo ser que lleguen en un mail y que el usuario asignado a dicha tarea pueda responderlos por la aplicación.

Existencia de aplicación móvil (1)

Es positivo la posibilidad de tener el CRM en una aplicación móvil, para comodidad del equipo de EviMed.

Importación de datos (3)

El CRM tiene que soportar la importación de datos de los sistemas que puedan eventualmente surgir en un futuro (las fuentes existentes actualmente están contempladas en el capítulo “Integraciones”) a partir de un scheduler.

Hosting (3)

Idealmente el CRM se podrá instalar en los propios servidores de EviMed para facilitar la implementación de las integraciones.

De lo contrario, sería bueno que el despliegue esté incluido en la licencia del CRM y no haya que contratarlo aparte. En este caso se deberá asegurar que las integraciones necesarias sean posibles.

10.2. Informe de análisis de CRMs

Odoo CRM

Infraestructura (5)

La integración se puede hacer tanto en Python como en PHP. Se puede hacer On Premise, pero esto limitaría la integración con terceros. Para eso es necesario utilizar Odoo.sh, que es un PaaS (Platform as a Service) que ofrece Odoo. Es para hosteo en la nube que facilita a los desarrolladores en la integración e implementación del CRM.

Modo de licenciamiento (5)

El licenciamiento es por usuario que utilice el CRM, no por cliente de Evimed. Son las únicas licencias que maneja Odoo, no hay de desarrolladores ni de otro tipo.

Licenciamiento por mes:

- Estándar 9,1 USD/Mes
- Personalizado 13,6 USD/Mes *

Licenciamiento anual:

- Estándar 7,25 USD/Mes
- Personalizado 10,9 USD/Mes *

* (Incluye el licenciamiento de Odoo.sh), se requiere de esta licencia y no la estándar dado a que se necesita consultar APIs externas.

Odoo.sh

Odoo.sh es la plataforma de integración continua donde puedes alojar tu desarrollo y ramas de prueba y producción. Es necesario adquirir dicha herramienta dado que la misma oficia de cuenta para el desarrollo, para poder realizar las integraciones y poder desarrollar. Para tener Odoo.sh, se tiene que pagar aparte el almacenamiento y los workers (hilos de ejecución, se recomienda un worker cada 25 comunicaciones al backend y 5000 visitantes de frontend por

día). También se cuenta con la posibilidad de pagar por un ambiente de pruebas (staging environment) con datos de producción (altamente recomendable).

Licenciamiento por mes:

- Workers (trabajador Odoo.sh) 72 USD/Worker/Mes
- Storage (almacenamiento) 0,25 USD/GB/Mes
- Staging environment (ambiente de test) 18 USD/Ambiente/Mes

Licenciamiento anual:

- Workers (trabajador Odoo.sh) 57,6 USD/Worker/Mes
- Storage (almacenamiento) 0,20 USD/GB/Mes
- Staging environment (ambiente de test) 14,4 USD/Ambiente/Mes

En caso de necesitar más GB, staging environment o workers, y se había pago anualmente, se hace un prorrateo de lo que quede de año la diferencia del precio.

También existe el módulo adicional de “Mantenimiento”. Pagando este módulo Odoo se encarga de realizar el mantenimiento de las líneas de desarrollo que fueron hechas en sus servidores. Se hace automáticamente cobrando 72 USD/100 líneas de código. Este módulo se puede rechazar al momento de comenzar a desarrollar, simplemente enviando un mail diciendo que del mantenimiento de nuestro desarrollo se hará cargo la empresa misma (equipo de desarrollo de Evimed). A futuro, si Odoo cambia de versión, la actualización la tendrá que hacer el equipo de desarrollo de Evimed y si no lo pueden realizar, deberán comprar un pack de horas de desarrollo para que lo haga Odoo (a no ser que tengan el módulo de mantenimiento activo).

La referente de ventas que está haciendo el seguimiento es Laura Pérez. Ella envió un presupuesto para diez personas aproximadamente que fue reenviado al cliente. Posteriormente y atendiendo lo charlado en la reunión con Evimed, se bajó la cantidad de licencias a cuatro. Tres de ellas para los desarrolladores y la última para el cliente Evimed. De estas 3 cuentas que serían del equipo de desarrolladores, es recomendable que sean del dominio de la empresa, para que podamos definir los diferentes roles para probar diferentes cosas y para que de ser

necesario puedan usar más de una en diferentes ocasiones de la implementación. Lo recomendable es seguir el contacto con ella que ya entiende el proyecto.

lpz@odoo.com

Integraciones (5)

Se puede enviar correos mediante la aplicación con un mail personalizado y en multitud (tiene una limitante inicial de 200 mails por día, si estos fueron exitosos, se va aumentando el límite. Esto es para no cargar los servidores de Odoo innecesariamente).

Se puede integrar con SMTP de Gmail si es necesario, configurando para permitir los mails de entrada y salida. (Esta opción es con mails ilimitados, ya que al tener un servidor propio no hay restricciones de uso)

https://www.odoo.com/documentation/16.0/es/applications/general/email_communication/email_servers.html

Ejemplo de envío:

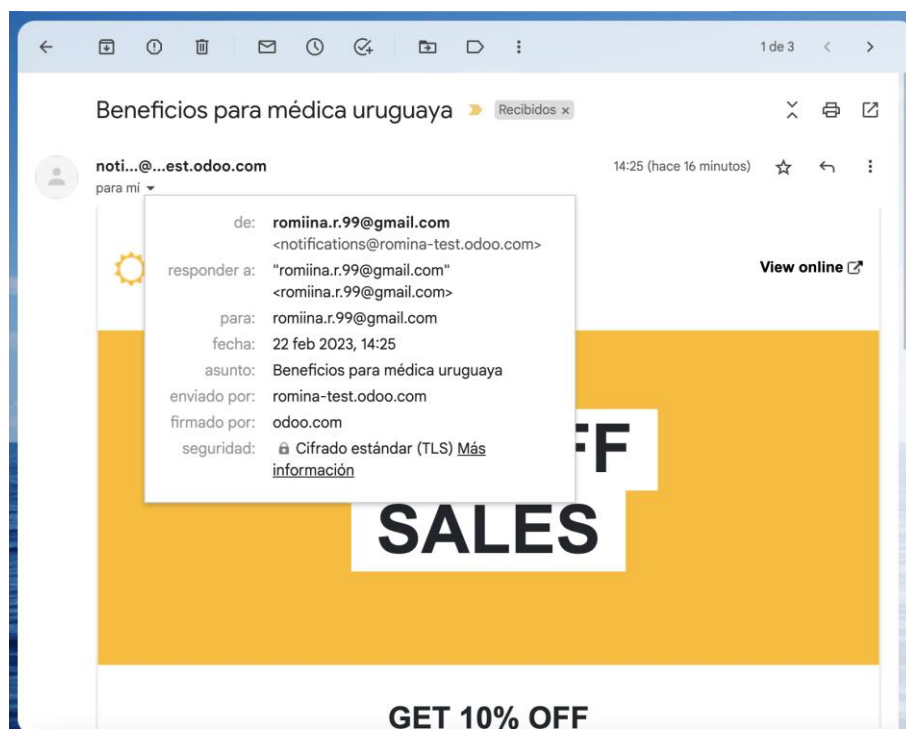


Figura 52: Ejemplo de envío de mail con Odoo CRM

La integración con WATI se puede realizar mediante Zapier.

<https://zapier.com/apps/wati/integrations/odoo>

Licenciamiento por mes:

- Starter 29,99 USD/Mes
- Professional 73,5 USD/Mes

Licenciamiento anual:

- Starter 19,99 USD/Mes
- Professional 49 USD/Mes

<https://zapier.com/app/pricing>

También hay plugins de Odoo para integrarlos:

<https://apps.odoo.com/apps/modules/browse?search=whatsapp>

Y por último existe la posibilidad de desarrollar la integración con WhatsApp por cuenta propia mediante Odoo.sh.

Customización de datos (5)

Odoo studio es la herramienta de personalización. Con ella se podrá personalizar los datos de los clientes.

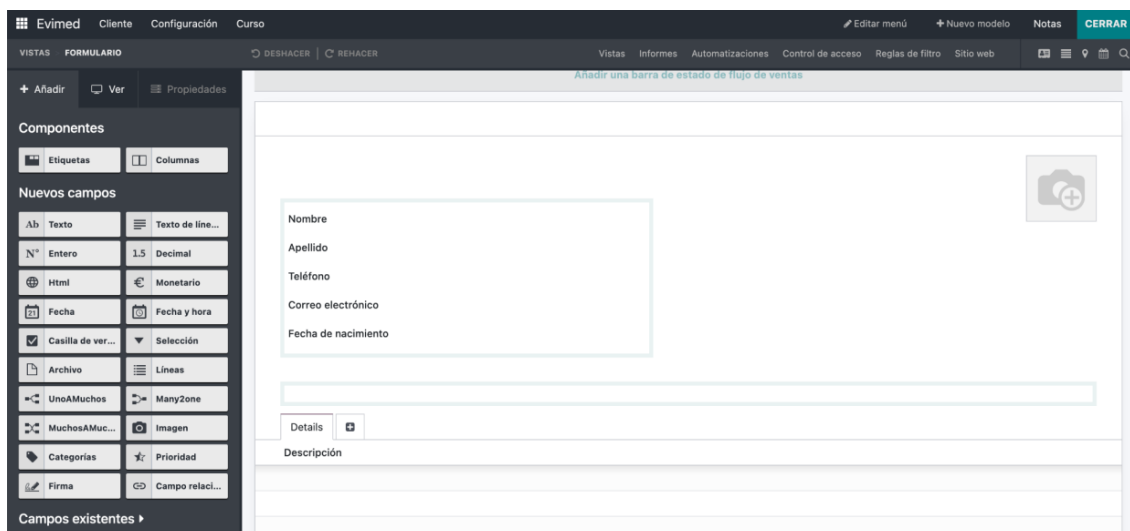


Figura 53: Odoo Studio para modificar interfaces

La entidad curso debería ser representada mediante los productos, dado que en realidad sí es el producto que vende Evimed. Para las segmentaciones se puede realizar fácilmente del lado de clientes, y para los cursos se puede filtrar por productos vendidos, con rangos de fechas y otras especificaciones.

Distinción B2B y B2C (3)

Con Odoo studio se puede agregar un campo para distinguir el tipo de cliente.

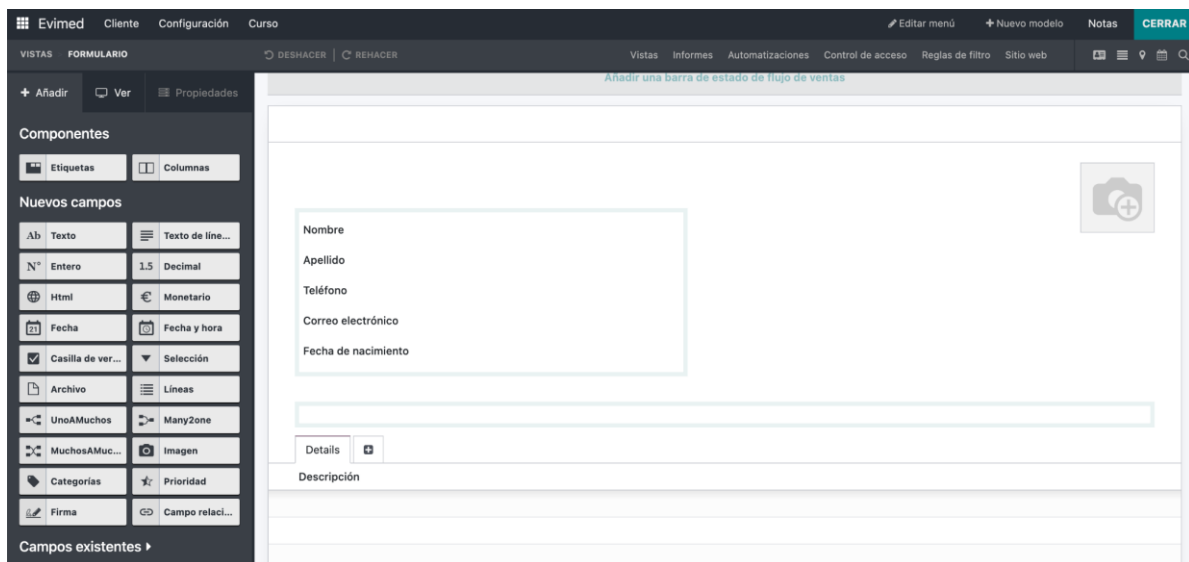


Figura 54: Interfaz de entidad cliente empresa siendo creada

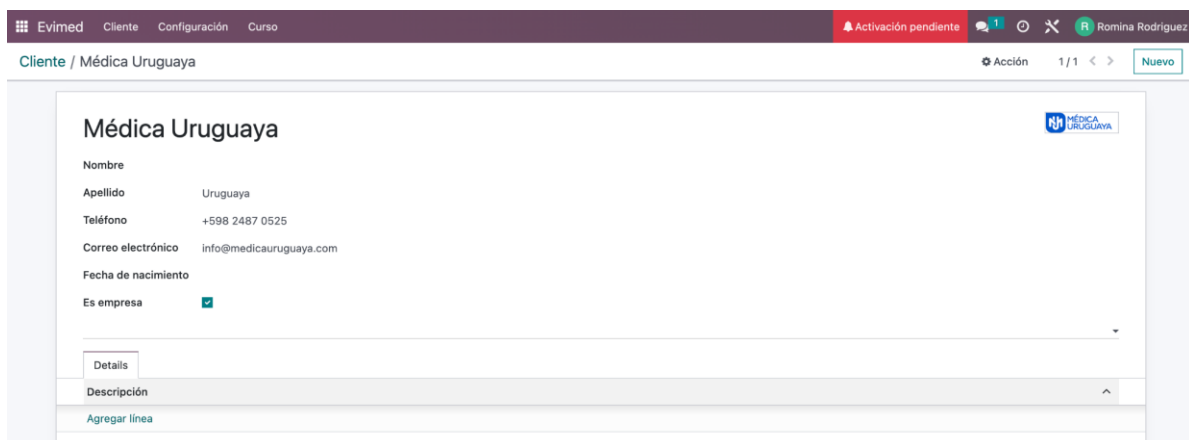


Figura 55: Interfaz de entidad cliente empresa terminada

Segmentación de clientes (5)

Se puede segmentar por clientes y obtener los cursos. Al tener los cursos como productos, se puede visualizar las “compras” de productos específicos, en rangos de fecha y qué personas. Por lo que se podría segmentar correctamente todo.

También cuenta con el módulo de Lead Scoring.

Interfaz modificable (1)

Con Odoo Studio se puede modificar cómo se muestran los datos de los clientes, pudiendo modificar el orden, ocultar y mostrar otros datos, etc.

Soporte para mantenimiento del CRM (2)

Tiene documentación (<https://www.odoo.com/documentation/16.0/es/>), videos en YouTube y personas disponibles a las que se le puede preguntar. Existe el equipo de partners que también brindan apoyo para la integración con el CRM (tienen un costo aparte). Si se precisa una funcionalidad específica para un reporte nuevo, se le puede pedir la estimación de horas para el desarrollo y así ver qué plan conviene más.

Soporte a usuarios (4)

Si alguien entra a un link en Facebook o Instagram de “contact us” se genera un lead en la planilla para que alguien lo contacte. Tiene helpdesk.

Se puede integrar con FreshDesk mediante Zapier u otra app misma de Odoo.

Existencia de aplicación móvil (1)

Tiene aplicación móvil en AppStore (<https://apps.apple.com/es/app/odoo/id1272543640>) y Google Play (https://play.google.com/store/apps/details?id=com.odoo.mobile&hl=es_UY&gl=US).

Importación de datos (3)

Se pueden importar datos a partir de la API de integración en cualquier momento.

Hosting (3)

Se puede hostear en los servidores de Evimed. Odoo sólo recomienda esta opción para clientes con conocimientos técnicos avanzados, ya que Odoo SA no puede ofrecer ninguna garantía ni acuerdo de nivel de servicio con respecto a la solidez, el rendimiento y la escalabilidad de su infraestructura.

Odoo tiene su propio hosting llamado Odoo.sh. Es un PaaS (Platform as a Service). Las licencias para Odoo.sh están incluidas en el plan Personalizado y es necesario si se necesitan integraciones con aplicaciones de terceros.

Estabilidad de licenciamiento (requerimiento especial)

Según lo conversado con la representante de Odoo, el modo de licenciamiento varió en el año 2022 pero a favor del cliente. Antes, las aplicaciones se pagaban por separado haciendo que la cuota mensual sea más alta. Ahora es un sólo pago, y citamos:

“Nuestra misión es proporcionar a las empresas acceso al software que necesitan para dirigir y expandir su empresa. Nuestro sueño es hacer que la digitalización sea accesible para todos, incluso pequeñas empresas con un presupuesto limitado.

Con cada versión de Odoo, continuamos haciendo que nuestro conjunto de aplicaciones empresariales sea más fuerte, inteligente y rápido, con muchas nuevas aplicaciones, funciones, servicios mejorados y material de aprendizaje. Aún había una función grande que debíamos actualizar para hacer nuestro sueño realidad: el precio.”

https://www.odoo.com/es_ES/blog/odoo-news-5/el-nuevo-precio-de-odoo-970

Pipedrive

Infraestructura (5)

La integración se puede hacer tanto en Python como en PHP.

Modo de licenciamiento (5)

El licenciamiento se da por usuario.

Licenciamiento por mes:

- Básico 15 USD/Mes
- Avanzado 29 USD/Mes
- Profesional 59 USD/Mes
- Corporativo 99 USD/Mes

Licenciamiento anual:

- Básico 12,5 USD/Mes
- Avanzado 24,9 USD/Mes
- Profesional 49,9 USD/Mes
- Corporativo 74,9 USD/Mes

Integraciones (5)

Se pueden enviar mails desde la página web. Se sincroniza la cuenta con la del mail y recibes y envías mail por la página. Ejemplo de envío:

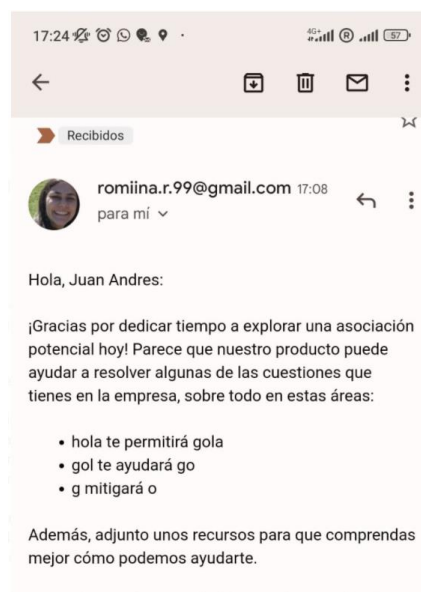


Figura 56: Ejemplo de envío de mail en Pipedrive

A diferencia de otros, el mail es únicamente el personal. No está de intermediario el CRM.

La integración con WATI se puede realizar mediante Zapier: <https://zapier.com/apps/pipedrive/integrations/wati>

Licenciamiento por mes:

- Starter 29,99 USD/Mes
- Professional 73,5 USD/Mes

Licenciamiento anual:

- Starter 19,99 USD/Mes
- Professional 49 USD/Mes

<https://zapier.com/app/pricing>

Customización de datos (5)

Mediante la configuración en campos de datos, se pueden agregar campos personalizados a las cuatro entidades definidas. No se encontró forma de agregar nuevas entidades, por lo que se quiera representar deberá hacerse mediante estas, personalizando los campos de datos.

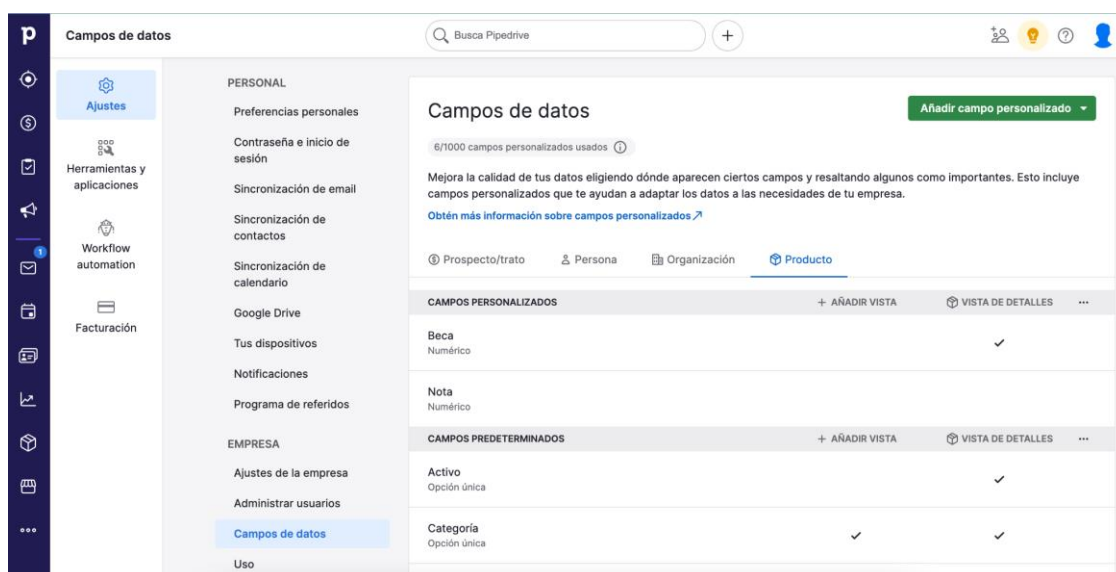


Figura 57: Ejemplo de interfaz en Pipedrive

La relación cliente-curso se puede hacer mediante los tratos, siendo el cliente la “Persona”, el curso un “Producto” y el “Trato” la relación entre ambos.

Distinción B2B y B2C (3)

Se le puede agregar un campo a la persona, representando si es un cliente o una empresa.

Segmentación de clientes (5)

Permite generar filtros en los clientes muy fácilmente.



<input type="checkbox"/>	Nombre completo	Especialidad	Título	Valor
<input type="checkbox"/>	Juan Andres Nervi	Anestesista	Trato Curso 2	UYU10,000

Figura 58: Tratos en Pipedrive

Interfaz modificable (1)

Se pueden modificar el lugar de las presentaciones y los campos.



<input type="checkbox"/>	Nombre completo	Especialidad	Título	Valor
<input type="checkbox"/>	Juan Andres Nervi	Anestesista	Trato Curso 2	UYU10,000

Elige las columnas

Buscar

VISIBLE

- Nombre completo Persona
- Especialidad Persona
- Título Trato
- Valor Trato

NO VISIBLE

Trato Persona Organización

- Actividades completadas
- Actividades por hacer
- Actividades totales
- Cantidad de producto

Por defecto Cancelar Guardar

Figura 59: Modificación de interfaz en Pipedrive

Soporte para mantenimiento del CRM (2)

Hay una gran documentación de ayuda al desarrollador con una colección de Postman lista. También es fácil de resolver dudas en foros, YouTube y Google.

Soporte a usuarios (4)

Se puede integrar el FreshDesk mediante Zapier.

<https://zapier.com/apps/freshdesk/integrations/pipedrive>

Existencia de aplicación móvil (1)

Tiene aplicación móvil en AppStore (<https://apps.apple.com/es/app/pipedrive-crm-de-ventas/id921456160>) y Google Play (https://play.google.com/store/apps/details?id=com.pipedrive&hl=es_UY&gl=US).

Importación de datos (3)

Se pueden importar en tiempo real mediante la API de Pipedrive.

Hosting (3)

No tiene la opción de On Premise, es un cloud-based CRM. El precio del hosting está integrado en las licencias.

Zoho

Infraestructura (5)

Posee un SDK para desarrollar e integrar funcionalidades tanto en Python como en Php. Tiene una extensión para enlazarse a bases de datos mySQL. Se puede instalar y utilizar en Linux.

Modo de licenciamiento (5)

Tiene diferentes planes, tanto mensuales como anuales (descuentos en planes anuales).

La forma de licenciamiento es precio/usuario/tiempo.

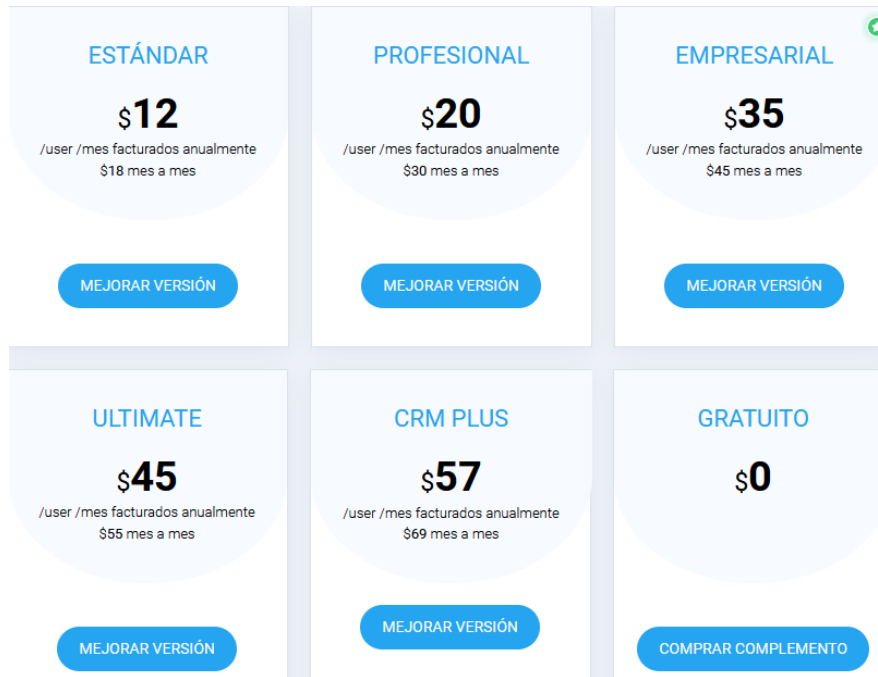


Figura 60: Modo de licenciamiento Zoho CRM

<https://www.zoho.com/es-xl/crm/comparison.html?src=crmpricing-middle>

Existe una versión gratuita, pero no permite la edición de campos del formulario de clientes, lo cual es de vital importancia y descalifica esta versión. También, solamente permite 3 usuarios.

Integraciones (5)

Posee APIs a las cuales podemos integrarnos. Posee integración a WhatsApp como servicio de correo, junto a 500+ integraciones a diferentes aplicaciones.

<https://www.zoho.com/es-xl/crm/integrations.html>

Customización de datos (5)

Se puede customizar los campos de los clientes (<https://help.zoho.com/portal/en/kb/crm/customize-crm-account/customizing-fields/articles/use-custom-fields>)

Se puede crear productos y asociarlos al resto de los datos del CRM, tanto como clientes o ventas (<https://help.zoho.com/portal/en/kb/crm/manage-inventory/products/articles/working-with-products>)

Distinción B2B y B2C (3)

Permite la creación de negocios para mantener relaciones B2B además de las B2C.

<https://help.zoho.com/portal/en/community/topic/how-to-set-up-for-both-b2c-and-b2b>

Segmentación de clientes (5)

Se puede segmentar los clientes por cualquiera (o conjunto) de sus atributos mediante filtros personalizables.

<https://help.zoho.com/portal/en/kb/crm/automate-business-processes/segmentation/articles/segmentation#Segmentation in Zoho CRM>

Interfaz modificable (5)

Se puede modificar la interfaz y las diferentes pestañas a demanda del usuario.

Soporte para mantenimiento del CRM (2)

El CRM posee una extensa documentación sobre sus APIs y de sus diferentes funcionalidades tanto para desarrolladores a nivel técnico como para usuarios “comunes”, junto a muchísima actividad del equipo de soporte en los foros de Zoho. También posee soporte en vivo y sistema de tickets.

Soporte a usuarios (4)

Una de las aplicaciones de Zoho con las que se puede integrar su CRM, es Zoho Desk, la cual es un sistema de manejo de consultas y soporte de clientes mediante tickets.

<https://www.zoho.com/desk/helpdesk-ticketing-system.html>

Existencia de aplicación móvil (1)

Zoho posee aplicaciones tanto para Android

(https://play.google.com/store/apps/details?id=com.zoho.crm&hl=es_419&gl=US&pli=1) como para iOS (<https://apps.apple.com/us/app/zoho-crm-sales-marketing/id444908810>).

Importación de datos (3)

Se puede importar los datos ya existentes mediante las diferentes APIs de Zoho.

Hosting (3)

Zoho no tiene versión On Premise, pero no es necesario contratar un servicio externo para el hosteo del mismo, ya está incluido.

Hubspot CRM

Infraestructura (5)

Hubspot tiene librerías para importar en el código para mejorar la fricción entre las APIs de Hubspot y nosotros. Existen librerías en Node.js, Ruby, PHP, Python. De las cuales nos interesan las últimas dos por requerimiento del cliente, las mismas están actualizadas y con READMEs para su utilización.

Modo de licenciamiento (5)

CRM gratuito. Luego existe un bundle que incluye Marketing Hub, Sales Hub, Service Hub, CMS Hub, Operations Hub todos ellos con 2 usuarios por U\$\$ 50 y sino se paga todo el año por adelantado se accede a un descuento, quedando en U\$\$ 45 al mes. Esto es necesario para crear productos (cursos) y asignarlos a los clientes.

Con respecto a campañas de marketing solo se puede realizar hasta 1000 clientes con la versión starter, existe la posibilidad de aumentar este número pero no es recomendable dado que excede el presupuesto.

Comienza a enviar correos electrónicos de forma gratuita

Envía 2000 correos electrónicos al mes, crea 25 listas estáticas y 5 listas inteligentes, revisa informes básicos y más.

Comenzar

Figura 61: Correos electrónicos en Hubspot CRM

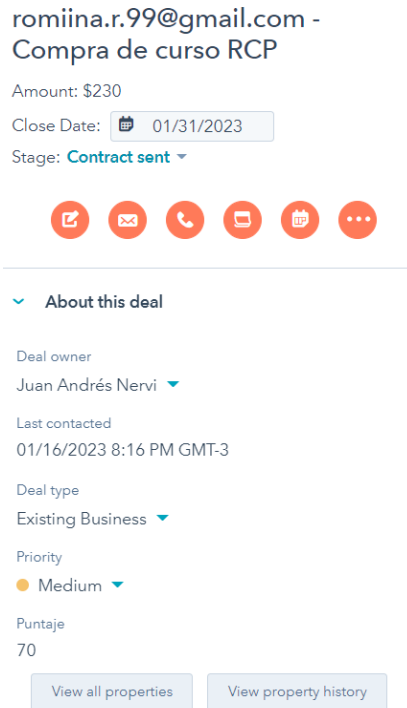


Figura 62: Hubspot CRM

Distinción B2B y B2C (3)

Existen tabs para clientes y empresas. Al crear un cliente, el CRM toma del email del mismo la empresa, si no existe en el sistema, se crea automáticamente.

Segmentación de clientes (5)

A partir de la creación de listas o lists, es posible segmentar a nuestros leads o clientes. Hay dos formas de hacerlo, creación de lists estáticas y creación de lists activas. Las primeras se crean con la información que el CRM tiene de nuestros leads hasta el momento y la segunda al ser activa, corresponde a la información en tiempo real del CRM. En la versión starter se puede tener hasta 25 listas activas o 1000 listas estáticas.

Interfaz modificable (1)

Se pueden modificar los datos de los clientes a su antojo, agregar eliminar o reordenar campos es posible con dicho CRM.

Soporte para mantenimiento del CRM (2)

Developers Documentation: [HubSpot Developer Site](#)

Hubspot Community: [HubSpot Community](#)

Hubspot Slack Community: [Join the HubSpot Developer Slack Community](#)

Hubspot Courses: [Courses & Lessons | HubSpot Academy](#)

Soporte a usuarios (4)

Tiene un apartado de meetings, donde los clientes pueden agendar una meet en el calendario de la persona o departamento buscando un slot de tiempo disponible para ello. Ideal para el equipo de ventas.

Existencia de aplicación móvil (1)

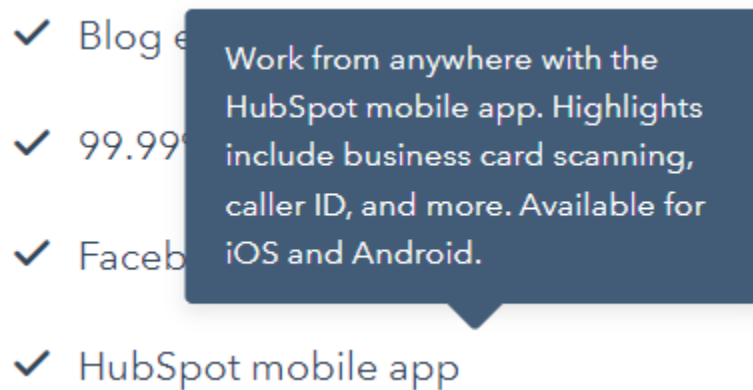


Figura 63: Evidencia existencia de *app mobile*

Efectivamente, HubSpot cuenta con versiones móviles para Android (<https://play.google.com/store/apps/details?id=com.hubspot.android&hl=es&gl=US>) e iOS (<https://apps.apple.com/es/app/crm-de-hubspot-crecer-mejor/id1107711722>).

Importación de datos (3)

El CRM tiene que soportar la importación de datos de los sistemas existentes regularmente a partir de un scheduler: Es posible. La importación de datos se puede realizar a través de un archivo CSV.

Para importar datos en HubSpot, el usuario debe seguir los siguientes pasos:

1. Acceder a la cuenta de HubSpot y hacer clic en "Contacts", "Companies", "Deals" o "Tickets", según el tipo de registro que se desee importar.
2. Hacer clic en el botón "Import" en la esquina superior derecha.
3. Seleccionar la opción "File import" para importar un archivo CSV o "Service import" para importar desde un servicio de terceros.
4. Configurar las opciones de importación, como el tipo de registro, etc.
5. Seleccionar el archivo CSV o el servicio de terceros que se desea importar y seguir las instrucciones en pantalla para completar la importación.

Hosting (3)

Es Cloud. No existe versión On Premise.

Estabilidad de licenciamiento (requerimiento especial)

Al consultar mediante chat a un representante de ventas de dicha compañía se nos especificó que una vez se realice el contrato anual, el precio finalizado dicho año va a seguir siendo el mismo si se renueva la suscripción. Para productos nuevos puede variar el precio pero siempre se pueden renegociar los términos.

Salesforce CRM

Infraestructura (5)

Salesforce ofrece varias opciones para integrar sus servicios con aplicaciones de terceros, incluyendo bibliotecas y SDKs para Python y PHP.

Para integrar Salesforce con Python, los desarrolladores pueden utilizar la biblioteca de Python para Salesforce (simple-salesforce), la cual provee una interfaz para interactuar con la API de Salesforce desde Python.

Modo de licenciamiento (5)

Por la cantidad de información y contactos, necesitamos Salesforce Enterprise a 150 dólares / usuario / mes.

Integraciones (5)

[APIs and Integration | Salesforce Developer Center](#)

Mediante APIs, en particular se puede integrar con una REST API (vista en la carrera).

Customización de datos (5)

Salesforce CRM permite crear campos personalizados para los objetos de contacto (y otros objetos) que ya existen, así como también definir nuevos objetos personalizados que contengan los campos específicos que se requieren. Estos campos personalizados se pueden crear a través de la interfaz de usuario de Salesforce o mediante programación utilizando la API de Salesforce.

Es posible crear relaciones n a n entre clientes y una entidad de cursos en Salesforce CRM, y se pueden agregar atributos a esa relación mediante campos personalizados en el objeto de unión u objeto de relación.

Para crear esta relación, se necesitan tres objetos personalizados: un objeto para los clientes, un objeto para los cursos y un objeto de unión que los conecte. El objeto de unión tendrá atributos que describen la relación, como la fecha de inicio del curso, el estado del curso, etc.

Para agregar atributos a la relación, se pueden crear campos personalizados en el objeto de unión. Estos campos se pueden utilizar para almacenar información adicional sobre la relación, como el progreso del cliente en el curso o la calificación obtenida.

Una vez que se han creado los objetos y campos personalizados necesarios, se pueden configurar reglas y flujos de trabajo en Salesforce para automatizar la gestión de las relaciones entre clientes y cursos. Por ejemplo, se pueden crear flujos de trabajo que envíen correos

electrónicos automáticos a los clientes cuando se agregue un nuevo curso a su cuenta, o se puede crear un panel de control para monitorear el progreso del cliente en varios cursos.

Distinción B2B y B2C (3)

Salesforce CRM permite distinguir entre las relaciones comerciales B2B y B2C y ofrece diferentes funcionalidades para cada uno de ellos.

Segmentación de clientes (5)

Es posible, se pueden aplicar filtros a los leads o clientes. Mediante la utilización de una variedad de herramientas, como listas de contacto, segmentación de campañas, informes personalizados y analíticas de ventas. Esto ayuda a los usuarios a identificar y clasificar a sus clientes según diferentes criterios, y crear estrategias de marketing y ventas más efectivas.

Interfaz modificable (1)

Salesforce CRM es altamente personalizable, y ofrece una gran cantidad de herramientas para adaptarla a las necesidades y preferencias de los usuarios. Desde la creación de páginas personalizadas hasta la utilización de la API de Salesforce, hay muchas formas de personalizar la plataforma para mejorar la productividad y la eficiencia de la empresa.

Soporte para mantenimiento del CRM (1)

Developers Documentation: [Developer Documentation | Salesforce Developers](#)

Esta buena la documentación, tiene una curva de aprendizaje un poco elevada para no tener experiencia previamente.

Soporte a usuarios (4)

Cuando el usuario tenga dudas puede contactarse, la información está en el apartado de Cases y podemos elegir cómo contactar al mismo.

Existencia de aplicación móvil (1)

Salesforce ofrece aplicaciones móviles para iOS (<https://apps.apple.com/es/app/salesforce/id404249815>) y Android (<https://play.google.com/store/apps/details?id=com.salesforce>).

[chatter&hl=es_UY&gl=US](#)) que permiten a los usuarios acceder a la plataforma CRM en la nube desde sus dispositivos móviles.

La aplicación móvil de Salesforce proporciona una experiencia de usuario optimizada para dispositivos móviles y permite a los usuarios ver y actualizar registros, colaborar con colegas, realizar tareas y tomar decisiones importantes en cualquier momento y lugar.

Importación de datos (3)

Es posible, la misma se puede hacer como importación de archivos CSV, importaciones a través de integraciones e importaciones a través de API. Es importante tener en cuenta que antes de importar datos a Salesforce, se debe realizar una planificación cuidadosa para asegurarse de que los datos se importen de manera correcta y que se mantenga la integridad de los mismos.

Hosting (3)

Está disponible solamente en la nube, el hosteo está incluido en el precio inicial en los planes essentials, professional y enterprise.

Firmao CRM

Infraestructura (5)

Su versión On Premise está hecha en Java.

Modo de licenciamiento (5)

Tiene tres planes diferentes de licenciamiento, no se considera el primero dado que este no tiene la posibilidad de customización de datos de los clientes.



Figura 64: Planes Firmao CRM

También tiene una modalidad on premise, con pago único (ver en apartado Hosteo en la nube).

Características y funcionalidades de las versiones

Integraciones (5)

Tiene integración con Gmail para automatización de envíos de correos, no tiene con WhatsApp, pero tiene con muchas otras aplicaciones que podrían ser de interés.

<https://firmao.net/integrations>

Customización de datos (5)

No tiene customización de los datos para los usuarios en la versión más barata, pero si en la Professional y la Enterprise.

Es posible en estas versiones, crear productos o servicios y asociar sus ventas a diferentes contactos.

Distinción B2B y B2C (3)

Se pueden crear contactos y compañías de forma separada, como también hacer pertenecer a ciertos contactos a diferentes compañías.

Segmentación de clientes (5)

Posee la capacidad de crear diferentes filtros para segmentar los contactos según sus diferentes campos y actuar sobre ellos.

Interfaz modificable (1)

No se puede modificar la interfaz de usuario.

Soporte para mantenimiento del CRM (2)

Tiene soporte técnico vía correo o llamada, pero no posee documentación para desarrolladores más allá de su API https://firmao.net/API-Documentation_EN.pdf.

Soporte a usuarios (4)

Tiene posibilidad de configurar meet con los diferentes contactos y a su vez mandarles mails.

Existencia de aplicación móvil (1)

Existe una aplicación móvil tanto para Android (<https://play.google.com/store/apps/details?id=pl.firmaomobile&hl=es&gl=US>) como iOS (<https://apps.apple.com/ro/app/firmao-crm/id1541176154>).

Importación de datos (3)

El CRM tiene que soportar la importación de datos de los sistemas existentes regularmente a partir de un scheduler.

Hosting (3)

Al ser una Web app, está en la nube por defecto, no hay que hacer nada al respecto. Se puede entrar desde cualquier web browser. Tiene opción on premise, con pago único.

Service	Cost
License for 1-10 users	\$7 250
License for 10-25 users	\$12 250
License for 25-50 users	\$19 750
License for 50-100 users	\$24 750
License without limits	\$29 750
Instalation	\$1 950
Technical support 1 year	\$1 450

Figura 65: Pagos Firmao CRM

Groundhogg CRM

Infraestructura (5)

Groundhogg es un plugin disponible para WordPress (el cual es open source). Está desarrollado en PHP y utiliza MySQL como motor de base de datos.

Modo de licenciamiento (5)

WordPress es gratuito y Groundhogg tiene un modo de licenciamiento fijo por mes, independiente de la cantidad de conexiones que se manejen.

Agency	Pro	Plus	Basic
\$ 100 /mo <small>(billed annually)</small>	\$ 50 /mo <small>(billed annually)</small>	\$ 40 /mo <small>(billed annually)</small>	\$ 20 /mo <small>(billed annually)</small>
"Deliver client marketing & sales projects faster and cheaper."	"Everything you need to maximize results, and more."	"Level up your sales and email marketing."	"The essentials every business needs."
Start Now	Start Now	Start Now	Start Now
Everything in Pro, and: <ul style="list-style-type: none">+ White Label CRM+ Access to our Certified Partner Program+ VIP Support+ Use for your clients+ 25 Sites <p>Have questions about using Groundhogg for your agency? Book an interview with us.</p>	Everything in Plus, and: <ul style="list-style-type: none">+ Complimentary Setup+ AffiliateWP Integration+ AWS Integration+ BuddyBoss Integration+ Conditional Logic+ EDD Integration+ GiveWP Integration+ HelpScout Integration+ LearnDash Integration+ LifterLMS Integration+ MemberPress Integration+ Postmark Integration+ Presto Player Integration+ SendGrid Integration+ SMS77 Integration+ Texting (SMS)+ ThriveCart Integration+ TutorLMS Integration+ Twilio Integration+ WhatsApp Integration+ Woocommerce Integration+ WP Simple Pay Integration+ Zapier Integration+ ZeroBounce Integration+ Use on up to 5 Sites	Everything in Basic, and: <ul style="list-style-type: none">+ Complimentary Setup+ All form integrations+ Backup & Restore+ Birthday automation+ Booking calendar+ Company Management+ Content restriction+ Divi Integration+ Elementor Integration+ Lead scoring+ Really simple payments+ Sales pipeline automation+ Signed Contracts+ Social Proof popup+ Use on up to 3 Sites	The core essentials: <ul style="list-style-type: none">+ Complimentary Setup+ Drag & drop email editor+ Use your SMTP service+ Advanced email preferences+ Custom replacement codes+ Email countdown timers+ Advanced form styling+ Superlinks (tracking links)+ 8 advanced funnel steps+ Premium support+ 1 Site

Figura 66: Precios Groundhogg CRM

Integraciones (5)

Permite la integración con servicios SMTP en todas sus versiones.

Posee integración con WhatsApp incluida en la versión PRO (US\$ 50 mensuales, más caro que otros evaluados).

Customización de datos (5)

Se puede agregar campos personalizados ilimitados a los contactos y agruparlos en categorías también personalizadas (puede servir para los cursos).

Distinción B2B y B2C (3)

Permite distinguir entre contactos y empresas.

<https://www.groundhogg.io/downloads/companies/>

Segmentación de clientes (5)

Se puede segmentar a los clientes mediante cualquiera de las propiedades, conjuntos de propiedades o su relación con productos previamente adquiridos.

Interfaz modificable (1)

No es posible.

Soporte para mantenimiento del CRM (2)

Groundhogg posee soporte técnico personalizado, junto a una extensa documentación de funcionalidades y otra específicamente para desarrolladores sobre la implementación de integraciones.

Soporte a usuarios (4)

Provee la posibilidad de agendar reuniones de zoom automáticamente con los diferentes contactos con sincronización en los diferentes calendarios web.

Existencia de aplicación móvil (1)

No tiene aplicación móvil.

Importación de datos (3)

Se pueden importar datos mediante CSV.

Hosting (1)

Al ser un plugin de WordPress, se puede realizar la página en WordPress en tus propios servidores y hostearla On Premise. Se puede hostear On Cloud, pero habría que contratar un servicio aparte.

SugarCRM

Infraestructura (5)

CRM de código abierto, lo que significa que su código fuente está disponible públicamente y puede ser modificado y distribuido por cualquier persona.

El núcleo de SugarCRM está desarrollado en el lenguaje de programación PHP, un lenguaje popular para el desarrollo de aplicaciones web dinámicas. Además, SugarCRM también utiliza otros lenguajes y tecnologías como JavaScript, HTML y CSS para la implementación de la interfaz de usuario y la funcionalidad del cliente en el navegador web.

Modo de licenciamiento (5)

No cumple. Se paga por cantidad de clientes (en la herramienta para marketing por ejemplo) y mínimo 50 euros por cada usuario (con un mínimo de tres usuarios) para las herramientas de sell y 80 euros para la herramienta de asistencia (serve) por cada usuario y con un mínimo de tres usuarios.

Integraciones (5)

Existe una amplia variedad de integraciones que se realizan mediante API. También, SugarCRM tiene la capacidad de integrarse con servidores SMTP para enviar correos electrónicos desde la plataforma.

Customización de datos (5)

Permite personalizar los datos que se muestran en la plataforma y ajustarlos a las necesidades específicas de la empresa. Se podrá crear campos personalizados, personalizar la apariencia y el diseño de módulos, permite definir flujos de trabajo personalizados, dashboards y vistas personalizadas.

Distinción B2B y B2C (3)

Se distingue entre leads (contacts) y accounts que son las empresas.

Segmentación de clientes (5)

La segmentación de clientes es posible con la herramienta Market que es una solución de automatización de marketing integrada con el CRM que permite crear y ejecutar campañas de marketing eficaces para llegar a sus clientes.

Interfaz modificable (1)

Es posible. Cuenta con una amplia variedad de personalizaciones como cambiar la apariencia, personalizar el menú de navegación, crear paneles personalizados, personalizar las visitas y también personalizar informes.

Soporte para mantenimiento del CRM (2)

SugarCRM proporciona soporte completo para mantener su plataforma actualizada, segura y eficiente, lo que permite a los usuarios concentrarse en su negocio en lugar de preocuparse por problemas técnicos.

- Documentación oficial de SugarCRM: https://support.sugarcrm.com/Documentation/Sugar_Versions/
- Foro de la comunidad de SugarCRM: <https://community.sugarcrm.com/community/developer>
- Centro de recursos de SugarCRM: <https://resources.sugarcrm.com/>
- Página de soporte de SugarCRM: <https://support.sugarcrm.com/>

Soporte a usuarios (4)

Con la herramienta Serve se le brinda atención al cliente a los usuarios de la empresa. Es un conjunto de características que se centran en mejorar la atención al cliente, la gestión de casos y la resolución de problemas. Es una herramienta de automatización de servicios que ayuda a los equipos de servicio al cliente a gestionar mejor los casos y a ofrecer una experiencia de atención al cliente superior.

Existencia de aplicación móvil (1)

Tiene aplicación móvil en AppStore (<https://apps.apple.com/us/app/sugarcrm/id599208633>) y Google Play (<https://play.google.com/store/apps/details?id=com.sugarcrm.nomad&hl=es&gl=US>).

Importación de datos (3)

Se puede importar datos desde hojas de Excel o en formato CSV.

Hosting (3)

SugarCRM ofrece tanto opciones de implementación en la nube como on-premise.

En la opción de implementación on-premise, se instala SugarCRM directamente en los servidores de la empresa y se administra internamente. Esto ofrece a las empresas un mayor control sobre su plataforma de CRM y una mayor capacidad para personalizar y adaptar la plataforma a sus necesidades específicas.

Sin embargo, la opción on-premise de SugarCRM también requiere más recursos internos, como servidores y personal de TI, para implementar y mantener la plataforma. Por lo tanto, esta opción suele ser más adecuada para empresas con un equipo de TI sólido y la capacidad de invertir en la infraestructura necesaria.

Dynamics

Infraestructura (5)

Al ser una aplicación de Microsoft, Dynamics está claramente enfocada en ser utilizada en Windows, ya que requiere de diferentes aspectos no disponibles en Linux (Microsoft SQL Server, Microsoft .NET Framework y Microsoft Internet Information Services).

Modo de licenciamiento (5)

Los modos de licenciamiento son por usuario por mes, pero tienen unos precios extremadamente elevados en comparación al resto del mercado (<https://dynamics.microsoft.com/en-us/pricing/>) la funcionalidad de marketing sale US\$1500 al mes.

Integraciones (5)

Se pueden realizar integraciones mediante la API de Dynamics, la cual es compatible con Python, habiendo bibliotecas en el mismo para facilitar la comunicación.

Posee integración con WhatsApp business.

Customización de datos (5)

Dynamics CRM permite personalizar los campos de datos de las entidades predefinidas, como los contactos y las cuentas, así como crear entidades personalizadas.

Para modificar los campos de datos de una entidad existente o crear nuevas entidades personalizadas, se pueden utilizar las herramientas de personalización integradas de Dynamics CRM, como el Editor de Personalización de Entidades. A través de esta herramienta, se puede agregar, eliminar o renombrar campos de datos, y definir reglas de validación y formatos para los datos de entrada.

Distinción B2B y B2C (3)

Sí, es posible distinguir entre clientes empresa (B2B) y usuarios finales (B2C) en Microsoft Dynamics CRM. Dynamics CRM proporciona herramientas y funcionalidades para segmentar y administrar a los clientes empresariales y a los usuarios finales de manera efectiva.

Segmentación de clientes (5)

Dynamics CRM proporciona una funcionalidad avanzada de segmentación de clientes, que te permite filtrar y segmentar los clientes en función de múltiples atributos, incluyendo los datos de contacto, los datos de transacciones, la información de productos, los datos demográficos, los datos de comportamiento y más.

Para segmentar a los clientes en Dynamics CRM, se puede utilizar la herramienta de "Segmentación de Listas". Con esta herramienta, es posible crear filtros complejos que permiten combinar múltiples criterios de búsqueda y crear segmentos de clientes personalizados.

Interfaz modificable (1)

Existen varias formas de modificar la interfaz de usuario en Dynamics CRM:

1. Personalización de formularios: Es posible personalizar los formularios en Dynamics CRM para agregar, eliminar o mover campos, y personalizar las etiquetas de campo y también crear formularios personalizados para diferentes tipos de registros, como contactos, cuentas, oportunidades y casos.
2. Personalización de vistas: Se puede personalizar las vistas en Dynamics CRM para mostrar solo los datos que necesitas, y organizar los datos en el orden que prefieras o crear vistas personalizadas para diferentes tipos de registros.
3. Personalización de paneles de navegación: Se pueden personalizar los paneles de navegación en Dynamics CRM para agregar, eliminar o reorganizar los elementos del menú.

Soporte para mantenimiento del CRM (2)

Tiene:

1. Centro de Ayuda y Soporte: Microsoft tiene un Centro de Ayuda y Soporte en línea, donde hay documentación, artículos y tutoriales para aprender a utilizar Dynamics CRM.

2. Comunidad de usuarios: Microsoft tiene una comunidad de usuarios en línea, donde se pueden hacer preguntas y obtener respuestas de otros usuarios de Dynamics CRM.
3. Soporte técnico en línea: Microsoft proporciona soporte técnico en línea a través de su portal de soporte técnico en línea, donde puedes abrir un caso de soporte técnico y recibir ayuda de un experto en Dynamics CRM.
4. Soporte telefónico: Microsoft también proporciona soporte telefónico a sus clientes, donde puedes hablar con un representante de soporte técnico para obtener ayuda con Dynamics CRM.

Soporte a usuarios (4)

Microsoft Dynamics CRM cuenta con una herramienta de levantado de tickets y soporte para clientes llamada Microsoft Dynamics 365 Customer Service. Esta herramienta permite a los usuarios registrar y realizar un seguimiento de los casos de soporte técnico y las solicitudes de asistencia de los clientes.

Existencia de aplicación móvil (1)

Sí, Microsoft Dynamics CRM tiene una aplicación móvil llamada "Microsoft Dynamics 365 for phones and tablets" que está disponible para dispositivos móviles iOS (<https://apps.apple.com/us/app/microsoft-dynamics-365/id678800460>), Android (<https://play.google.com/store/apps/details?id=com.microsoft.crm.crmphone&hl=es&gl=US>) y Windows (<https://learn.microsoft.com/en-us/dynamics365/mobile-app/install-dynamics-365-for-phones-and-tablets>). Esta aplicación móvil permite a los usuarios acceder a sus datos de Dynamics CRM desde sus dispositivos móviles y realizar tareas de forma remota, como ver, actualizar y crear registros, agregar notas y actividades, y trabajar con dashboards y reportes.

Importación de datos (3)

Se puede importar datos desde la API mediante un script o mediante un CSV o bases de datos.

Hosting (3)

El alojamiento en la nube para Microsoft Dynamics CRM es proporcionado por Microsoft a través de su plataforma en la nube, Azure. El precio del alojamiento en la nube de Azure es

independiente del precio de la licencia de Dynamics CRM. En otras palabras, el alojamiento en la nube es un costo adicional al precio de la licencia.

Siebel CRM

Infraestructura (5)

Lenguaje: Siebel CRM no utiliza Python ni PHP. Está basado en tecnologías propias de Oracle, como Java y Oracle Database.

Sistema operativo: Puede ejecutarse en sistemas operativos Windows y Linux.

Base de datos: Siebel CRM es compatible con Oracle Database, no con MySQL.

Modo de licenciamiento (5)

Siebel CRM es una solución comercial de Oracle, por lo que no es gratuita ni de código abierto. El costo de licenciamiento se basa en una variedad de factores, como el número de usuarios y módulos utilizados. No se basa en interacciones con los clientes ni en la cantidad de clientes.

Integraciones (5)

Siebel CRM ofrece APIs para integración con fuentes de datos externas en tiempo real.

Soporta integración con correo electrónico a través de SMTP.

No tiene una integración específica con WhatsApp o Zoom, por lo que requeriría desarrollo personalizado.

Customización de datos (5)

Siebel CRM permite la personalización de campos y la creación de nuevas entidades, lo que incluye relaciones n a n.

Distinción B2B y B2C (3)

Siebel CRM permite distinguir entre usuarios de tipo persona y empresas mediante la personalización de entidades y atributos.

Segmentación de clientes (5)

Siebel CRM permite la ejecución de consultas y filtros para segmentar clientes y realizar actividades de marketing. También ofrece capacidades de Scoring.

Interfaz modificable (1)

Siebel CRM ofrece cierto grado de personalización de la interfaz, pero no es muy flexible en comparación con algunas soluciones más modernas.

Soporte para mantenimiento del CRM (2)

Oracle proporciona soporte y documentación para Siebel CRM, pero la comunidad de usuarios y foros activos puede ser limitada en comparación con soluciones de código abierto.

Soporte a usuarios (4)

Siebel CRM incluye funcionalidades para la gestión de consultas y soporte de usuarios a través de la aplicación.

Existencia de aplicación móvil (1)

Siebel CRM ofrece una aplicación móvil que permite a los usuarios acceder a ciertas funcionalidades desde dispositivos móviles.

Importación de datos (3)

Siebel CRM es capaz de importar datos de sistemas externos, pero la implementación de programación por lotes y programación automática puede requerir desarrollo personalizado.

Hosting (3)

Siebel CRM se puede instalar en los servidores de EviMed, pero ten en cuenta que se trata de una solución empresarial que puede requerir infraestructura significativa. El despliegue puede estar incluido en el costo de licencia, pero es necesario asegurarse de que las integraciones necesarias sean posibles.

Saleslogix CRM

Infraestructura (5)

Lenguaje: Saleslogix CRM se basa en .NET y utiliza C# en lugar de Python o PHP.

Sistema operativo: Puede ejecutarse en sistemas operativos Windows y Linux (mediante emulación o virtualización).

Base de datos: Saleslogix CRM es compatible con bases de datos como Microsoft SQL Server, no con MySQL.

Modo de licenciamiento (5)

Saleslogix CRM es una solución comercial y no es gratuita. El costo de licenciamiento se basa en el número de usuarios y módulos utilizados.

Integraciones (5)

Saleslogix CRM ofrece APIs para integración con fuentes de datos externas en tiempo real.

Soporta integración con correo electrónico a través de SMTP.

Las integraciones con WhatsApp y Zoom requerirían desarrollo personalizado.

Customización de datos (5)

Saleslogix CRM permite la personalización de campos y la creación de nuevas entidades, incluyendo relaciones n a n.

Distinción B2B y B2C (3)

Saleslogix CRM permite distinguir entre usuarios de tipo persona y empresas mediante la personalización de entidades y atributos.

Segmentación de clientes (5)

Saleslogix CRM permite la ejecución de consultas y filtros para segmentar clientes y realizar actividades de marketing. También ofrece capacidades de Scoring.

Interfaz modificable (1)

Saleslogix CRM ofrece opciones de personalización de la interfaz, pero su flexibilidad puede ser limitada en comparación con algunas soluciones más modernas.

Soporte para mantenimiento del CRM (2)

Infor, la empresa detrás de Saleslogix CRM, proporciona soporte y documentación. La comunidad de usuarios y foros activos puede variar.

Soporte a usuarios (4)

Saleslogix CRM incluye funcionalidades para la gestión de consultas y soporte de usuarios a través de la aplicación.

Existencia de aplicación móvil (1)

Saleslogix CRM ofrece una aplicación móvil que permite a los usuarios acceder a ciertas funcionalidades desde dispositivos móviles.

Importación de datos (3)

Saleslogix CRM es capaz de importar datos de sistemas externos, pero la implementación de programación por lotes y programación automática puede requerir desarrollo personalizado.

Hosting (3)

Saleslogix CRM se puede instalar en los servidores de EviMed, pero ten en cuenta que se trata de una solución empresarial que puede requerir infraestructura significativa. El despliegue puede estar incluido en el costo de licencia, pero es necesario asegurarse de que las integraciones necesarias sean posibles.

Top 3 CRMs

Para evaluar los CRMs de una forma más objetiva y cuantificable, de forma de poder entregar una selección acotada con las mejores opciones para luego discutir y analizar esos candidatos en mayor profundidad, decidimos construir una matriz de Pugh, la cual es la matriz que se presenta debajo.

Funciona de la siguiente manera: mediante los criterios definidos en conjunto con sus pesos asociados, podemos calificar cada CRM en cada uno de esos aspectos en -1,0,1. Al multiplicar ese valor por el peso, repetir eso para cada valor y sumar la totalidad de puntos para cada CRM, obtenemos un valor numérico que nos permite comparar objetivamente según estos criterios cuáles son mejores opciones.

En el siguiente documento, detallamos qué se buscaba en cada CRM para evaluar cada punto. Criterios de selección para el CRM.

Criterios	Peso	Conceptos										
		Odoo	Hubspot	Zoho	Pipedrive	Firmao	SugarCRM	Groundhogg	Salesforce	Dynamics	Siebel CRM	SalesLogix
Infraestructura	5	1	1	1	1	0	1	0	1	-1	-1	-1
Modo de licenciamiento	5	1	0	1	1	1	-1	1	-1	-1	-1	-1
Integraciones	5	1	1	1	1	1	1	1	1	1	1	1
Customización de datos	5	1	1	1	1	1	1	1	1	1	1	1
Distinción B2B & B2C	3	1	1	1	1	0	1	1	1	1	1	1
Segmentación de clientes	5	1	1	1	1	1	1	1	1	1	1	1
Interfaz modificable	1	1	1	1	1	1	1	-1	1	1	1	1
Soporte para mantenimiento del CRM	2	1	1	1	1	-1	1	1	0	1	1	1
Soporte a usuarios	4	1	1	1	0	1	1	0	1	1	1	1
Existencia de aplicación móvil	1	1	1	1	1	1	1	-1	1	1	0	1
Importación de datos	3	1	1	1	1	1	1	1	1	1	1	1
Hosting	3	1	1	1	0	1	1	1	1	-1	1	1
Estabilidad de licenciamiento	1	1	1	-	-	-	-	-	-	-	-	-
Suma positivos (+)	-	43	38	42	35	32	37	31	35	29	31	32
Suma negativos (-)	-	0	0	0	0	2	5	2	5	13	10	10
Total	-	43	38	42	35	30	32	29	30	16	21	22

Tabla 8: Matriz de Pugh

Realizados los análisis, concluimos que nuestro top 3 de CRMs es: Zoho, Hubspot y Odoo. Por lo hablado en la reunión donde se presentó este top 3, se destacó la interfaz de usuario compleja, poco intuitiva y “anticuada” que posee Zoho. Por más que esto no fue definido como un criterio, creemos que este feedback es de suma importancia y debido a él decidimos que Zoho sea el primer eliminado de la selección final.

Odoo CRM vs HubSpot CRM

Conceptos			
Criterios	Odoo	Hubspot	¿Cuál es mejor?
Infraestructura	Se puede instalar on premise, pero lo recomendable es on cloud ya que para conectarse con terceros es necesario esta funcionalidad. La integración es en Python.	Herramienta cloud, tiene múltiples librerías para realizar integraciones. En especial nos interesa las librerías de Python y PHP.	Iguales
Modo de licenciamiento	Es necesario licenciamiento con Odoo.sh para poder conectarse con terceros (ejemplo WhatsApp). Los precios rondan entre 10,9 US\$/Mes y 13,6 US\$/Mes cada licencia dependiendo si se paga anualmente o por mes respectivamente. También, Odoo.sh nos da la posibilidad de mantener nuestro propio código por nuestra cuenta o de lo contrario un equipo se encargará de ello por US\$72 cada 100 líneas de código al año. Según el presupuesto que nos enviaron, el año saldría US\$1.435,20, se logró realizar un enfoque bottom up, es decir, de lo mínimo indispensable para poder desarrollar y testear, y en un futuro adquirir más licencias para el abarcamiento de distintas áreas de la empresa Evimed.	Si bien presenta versiones gratuitas, para los criterios establecidos es necesario adquirir la versión Suite CRM Starter de HubSpot dado a que debemos crear productos (que son propiedad del módulo de ventas). Para realizar campañas de marketing encontramos que en dicha versión se puede hasta 1000 clientes. Dicha versión viene con 2 licencias de Service Hub y 2 licencias de Sales Hub, comprando 8 más de cada una (en total 20 licencias de ambos servicios) rondaría un precio de US\$4000 aproximadamente, lo que se puede rever. Capaz que no todos los usuarios necesitan todas las funcionalidades.	Odoo
Integraciones	Se puede integrar con SMTP lo cual permite envíos ilimitados a nivel de CRM por mes, con WhatsApp mediante aplicaciones de terceros o desarrollo propio y con todas las demás integraciones mediante API.	Se puede realizar integraciones de servidores de e-mail externos por SMTP, integraciones de datos desde otros crms, entre una infinidad de otras integraciones.	Iguales
Customización de datos	Existe una herramienta de personalización llamada Odoo Studio que permite personalizar los datos del cliente, del producto (curso) y de la forma de	Existe una amplia variedad de customizaciones, agregación de campos como puntaje de los cursos. Se pueden realizar relaciones entre productos (cursos) y clientes. Para ello es necesario adquirir la	Iguales

	presentarlo.	licencia de Sales Hub dado a que la definición de productos no es gratuita.	
Distinción B2B & B2C	Existen varias formas fáciles de poder representar esta distinción.	Existe la posibilidad de crear empresas y usuarios clientes. Las empresas pueden crearse manualmente o pueden ser tomadas del mail del cliente, fijándose si estas ya integran nuestra lista o creándolas en otro caso.	Iguales
Segmentación de clientes	Se puede hacer una fácil segmentación de clientes mediante filtros. También se pueden filtrar cursos a partir de las compras, de los clientes y demás datos. A su vez, cuenta con un módulo de Lead Scoring.	Con la versión CRM Suite Starter es posible segmentar clientes de dos maneras distintas. Creando hasta 25 listas activas y hasta 1000 listas estáticas. Dicha segmentación puede exportarse a un CSV o un Excel si así se desea. Con respecto a la creación de campañas de marketing, sólo se puede realizar con 1000 clientes de marketing que pueden ir variando mes a mes de acuerdo a lo que se requiera, pudiendo enviar hasta 5000 mails al mes (desde el CRM).	Odoo
Interfaz modificable	Con Odoo Studio se pueden modificar las vistas de las entidades y del proyecto.	Se pueden modificar los datos de los clientes a su antojo, agregar eliminar o reordenar campos es posible con dicho CRM.	Iguales
Soporte para mantenimiento del CRM	Tiene una extensa documentación, videos de YouTube y foros de comunidades. Además, cuenta con un servicio de desarrolladores que si se necesita desarrollar algo por parte de Odoo, ellos se encargan. (Tiene un costo extra).	Amplia comunidad de slack, documentación y servicio al cliente por parte de HubSpot.	Iguales
Soporte a usuarios	Se puede integrar FreshDesk y tiene varias aplicaciones de terceros para realizar el soporte.	Tiene un apartado de meetings, donde los clientes pueden agendar una meet en el calendario de la persona o departamento buscando un slot de tiempo disponible para ello. Ideal para el equipo de ventas.	Iguales
Existencia de aplicación móvil	Cuenta con aplicación móvil.	Cuenta con aplicación móvil.	Iguales
Importación de datos	Se puede importar mediante API en cualquier momento.	Importación mediante CSV o mediante las integraciones existentes.	Iguales
Hosting	Se debe pagar por Odoo.sh aparte.	Al ser una herramienta Cloud, no hay que preocuparse por el hosting.	Iguales
Estabilidad de licenciamiento	La misión de Odoo es hacer el mejor trabajo en conjunto para que las aplicaciones sean fuertes, inteligentes, rápidas, entre otro. Para lograr ese objetivo entiende que deben mantener el precio en el	Al consultar mediante chat a un representante de ventas de dicha compañía se especificó que una vez se realice el contrato anual, el precio finalizado dicho año va a seguir siendo el mismo si se renueva la suscripción. Para productos	Iguales

	formato actual (que mejoró respecto al año 2022) para crecer junto a sus clientes. Por lo que da a entender que NO cambiará su forma de licenciamiento.	nuevos puede variar el precio pero siempre se pueden renegociar los términos.	
--	---	---	--

Tabla 9: Odoo CRM vs HubSpot CRM

Comentarios y conclusiones

En cuanto a lo académico, creemos que la etapa de investigación de los diferentes CRMs que existen en el mercado fue enriquecedora para nosotros, teniendo en cuenta que ninguno de los tres tenía experiencia previa con este tipo de softwares. Podemos decir que hemos aprendido sobre las diferentes funcionalidades que los principales CRMs manejan, sus diferencias y hacia donde apuntan.

Dicho eso, como mencionamos en el resumen del documento, para aquellos CRMs que tenían mayor puntaje realizamos contactos con expertos y a su vez probamos los mismos solicitando versiones de prueba. Fue allí que a parte de cotejar los datos, con la puesta en práctica aprendimos aún más. Muchas veces los licenciamientos, si bien tienen partes gratuitas, no se adaptan a lo que nosotros buscamos y era necesario encontrar una versión de pago que se adecue al presupuesto.

Uno de los motivos que nos llevó a pensar más en adquirir Odoo fue la creación de campañas de marketing, dado que HubSpot cobra por tiers de clientes de marketing siendo complejo en la situación actual de Evimed (más aún si su número de clientes seguirá aumentando). Otro motivo fue el costo anual de mantenimiento, Odoo no nos cobrará por las líneas de código a mantener si así se lo expresamos vía mail, la herramienta detecta si existen líneas de código a mantener, ya sea por una nueva versión del software de Odoo u otro motivo. Odoo ofrece según nuestros criterios de selección las funcionalidades por un precio menor, lo que creemos que es conveniente.

Por todo esto y lo explicitado en el documento, **creemos que Odoo es el CRM que se debería adquirir.**

El presupuesto antes descrito se puede encontrar [aquí](#) para mayor información. Se incluye el contacto de la persona que nos estuvo dando seguimiento en todo este tiempo, la que sugerimos se siga manteniendo.

10.3. Manual del desarrollador

Objetivo

El objetivo del presente documento es familiarizar al personal técnico de Evimed sobre la instalación y modo de funcionamiento de los productos satélites (llamados Adapter y Extensión de Google Chrome en el resto del documento) y de Odo CRM. Se asume conocimiento previo básico del funcionamiento de Amazon Web Services (AWS).

1. Adapter

1.1. Link al repositorio

Para ingresar al repositorio es necesario que el usuario sea invitado a participar del mismo.

Link: <https://github.com/romrrodriguez/data-enhancer-for-odoo>

1.2. Despliegue en la nube (AWS)

Todos los diferentes elementos del sistema tendrán el mismo proceso de despliegue, partimos de nuestro código y un Dockerfile, y se sube a Elastic Beanstalk (EB). En las siguientes subsecciones se especificarán los archivos necesarios para cada despliegue.

Es muy importante que se haga un comprimido con los archivos a desplegar en el mismo directorio padre, es decir, que no haya una carpeta dentro del comprimido y dentro de esa carpeta nuestros archivos).

Todos estos archivos que se mencionan a continuación se encuentran en el directorio data-enhancer-for-odoo/project/core

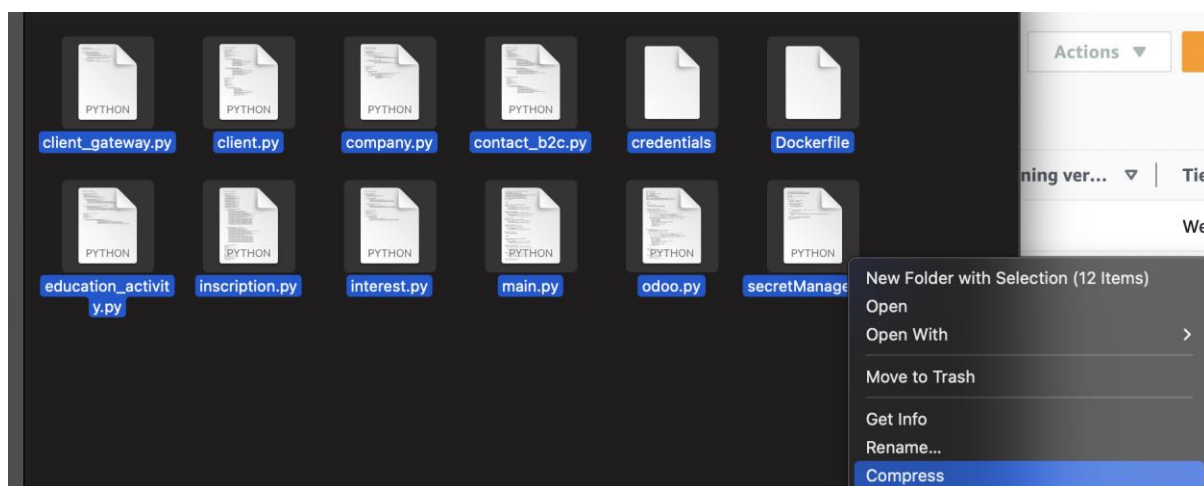


Figura 67: Comprimido de archivos para EB

1.2.1. API

1.2.1.1. Archivos a desplegar

- client_gateway.py

- client.py
- company.py
- contact_b2c.py
- credentials
- Dockerfile
- education_activity.py
- inscription.py
- interest.py
- main.py
- odoo.py
- secretManager.py

1.2.1.2. Dockerfile

FROM python:3.10

EXPOSE 80

ADD main.py odoo.py secretManager.py client.py company.py client_gateway.py
contact_b2c.py education_activity.py inscription.py interest.py ./

ADD credentials /root/.aws/

RUN pip install python-multipart pandas anyio==3.7.0 boto3==1.26.160 botocore==1.29.160
certifi==2023.5.7 charset-normalizer==3.1.0 click==8.1.3 datadog==0.45.0 decouple==0.0.7
fastapi==0.98.0 h11==0.14.0 httptools==0.5.0 idna==3.4 jmespath==1.0.1 pydantic==1.10.9
python-dateutil==2.8.2 python-dotenv==1.0.0 PyYAML==6.0 requests==2.31.0
s3transfer==0.6.1 six==1.16.0 sniffio==1.3.0 starlette==0.27.0 typing_extensions==4.6.3
urllib3==1.26.16 uvicorn==0.22.0 uvloop==0.17.0 watchfiles==0.19.0 websockets==11.0.3

CMD ["uvicorn", "main:app", "--proxy-headers", "--host", "0.0.0.0", "--port", "80"]

1.2.2. Consumidores

Los consumidores son desplegados individualmente de la misma forma que la API, esto se hizo para que se puedan prender o apagar los consumidores a necesidad, minimizando costos.

A continuación, los Dockerfile de cada uno junto a los archivos necesarios en el comprimido.

1.2.2.1. Clientes

1.2.2.1.1. Archivos a desplegar

- odoo.py
- secretManager.py
- credentials
- Dockerfile
- client.py
- clientMqConsumer.py

1.2.2.1.2. Dockerfile

```
FROM python:3.10
```

```
ADD secretManager.py client.py odoo.py clientMqConsumer.py ./
```

```
ADD credentials /root/.aws/
```

```
RUN pip install python-multipart pandas anyio==3.7.0 boto3==1.26.160 botocore==1.29.160  
certifi==2023.5.7 charset-normalizer==3.1.0 click==8.1.3 datadog==0.45.0 decouple==0.0.7  
fastapi==0.98.0 h11==0.14.0 httptools==0.5.0 idna==3.4 jmespath==1.0.1 pydantic==1.10.9  
python-dateutil==2.8.2 python-dotenv==1.0.0 PyYAML==6.0 requests==2.31.0  
s3transfer==0.6.1 six==1.16.0 sniffio==1.3.0 starlette==0.27.0 typing_extensions==4.6.3  
urllib3==1.26.16 uvicorn==0.22.0 uvloop==0.17.0 watchfiles==0.19.0 websockets==11.0.3
```

```
CMD ["python", "clientMqConsumer.py"]
```

1.2.2.2. Compañías

1.2.2.2.1. Archivos a desplegar

- odoo.py
- secretManager.py
- credentials
- Dockerfile
- company.py
- companyMqConsumer.py

1.2.2.2.2. Dockerfile

FROM python:3.10

ADD secretManager.py company.py odoo.py companyMqConsumer.py ./

ADD credentials /root/.aws/

RUN pip install python-multipart pandas anyio==3.7.0 boto3==1.26.160 botocore==1.29.160 certifi==2023.5.7 charset-normalizer==3.1.0 click==8.1.3 datadog==0.45.0 decouple==0.0.7 fastapi==0.98.0 h11==0.14.0 httpx==0.25.0 idna==3.4 jmespath==1.0.1 pydantic==1.10.9 python-dateutil==2.8.2 python-dotenv==1.0.0 PyYAML==6.0 requests==2.31.0 s3transfer==0.6.1 six==1.16.0 sniffio==1.3.0 starlette==0.27.0 typing_extensions==4.6.3 urllib3==1.26.16 uvicorn==0.22.0 uvloop==0.17.0 watchfiles==0.19.0 websockets==11.0.3

CMD ["python", "companyMqConsumer.py"]

1.2.2.3. Actividades educativas

1.2.2.3.1. Archivos a desplegar

- odoo.py
- secretManager.py

- credentials
- Dockerfile
- education_activity.py
- educationActivityMqConsumer.py

1.2.2.3.2. Dockerfile

FROM python:3.10

ADD secretManager.py education_activity.py odoo.py educationActivityMqConsumer.py ./

ADD credentials /root/.aws/

RUN pip install python-multipart pandas anyio==3.7.0 boto3==1.26.160 botocore==1.29.160 certifi==2023.5.7 charset-normalizer==3.1.0 click==8.1.3 datadog==0.45.0 decouple==0.0.7 fastapi==0.98.0 h11==0.14.0 httptools==0.5.0 idna==3.4 jmespath==1.0.1 pydantic==1.10.9 python-dateutil==2.8.2 python-dotenv==1.0.0 PyYAML==6.0 requests==2.31.0 s3transfer==0.6.1 six==1.16.0 sniffio==1.3.0 starlette==0.27.0 typing_extensions==4.6.3 urllib3==1.26.16 uvicorn==0.22.0 uvloop==0.17.0 watchfiles==0.19.0 websockets==11.0.3

CMD ["python", "educationActivityMqConsumer.py"]

1.2.2.4. Inscripciones

1.2.2.4.1. Archivos a desplegar

- odoo.py
- secretManager.py
- credentials
- Dockerfile
- inscription.py
- inscriptionMqConsumer.py

1.2.2.4.2. Dockerfile

```
FROM python:3.10
```

```
ADD secretManager.py inscription.py odoo.py inscriptionMqConsumer.py ./
```

```
ADD credentials /root/.aws/
```

```
RUN pip install python-multipart pandas anyio==3.7.0 boto3==1.26.160 botocore==1.29.160  
certifi==2023.5.7 charset-normalizer==3.1.0 click==8.1.3 datadog==0.45.0 decouple==0.0.7  
fastapi==0.98.0 h11==0.14.0 httptools==0.5.0 idna==3.4 jmespath==1.0.1 pydantic==1.10.9  
python-dateutil==2.8.2 python-dotenv==1.0.0 PyYAML==6.0 requests==2.31.0  
s3transfer==0.6.1 six==1.16.0 sniffio==1.3.0 starlette==0.27.0 typing_extensions==4.6.3  
urllib3==1.26.16 uvicorn==0.22.0 uvloop==0.17.0 watchfiles==0.19.0 websockets==11.0.3
```

```
CMD ["python", "inscriptionMqConsumer.py"]
```

1.2.2.5. Intereses

1.2.2.5.1. Archivos a desplegar

- odoo.py
- secretManager.py
- credentials
- Dockerfile
- interest.py
- interestsMqConsumer.py

1.2.2.5.2. Dockerfile

```
FROM python:3.10
```

```
ADD secretManager.py interest.py odoo.py interestsMqConsumer.py ./
```

```
ADD credentials /root/.aws/
```

```
RUN pip install python-multipart pandas anyio==3.7.0 boto3==1.26.160 botocore==1.29.160
certifi==2023.5.7 charset-normalizer==3.1.0 click==8.1.3 datadog==0.45.0 decouple==0.0.7
fastapi==0.98.0 h11==0.14.0 httptools==0.5.0 idna==3.4 jmespath==1.0.1 pydantic==1.10.9
python-dateutil==2.8.2 python-dotenv==1.0.0 PyYAML==6.0 requests==2.31.0
s3transfer==0.6.1 six==1.16.0 sniffio==1.3.0 starlette==0.27.0 typing_extensions==4.6.3
urllib3==1.26.16 uvicorn==0.22.0 uvloop==0.17.0 watchfiles==0.19.0 websockets==11.0.3
```

```
CMD ["python", "interestsMqConsumer.py"]
```

1.2.2.6. Campañas

1.2.2.6.1. Archivos a desplegar

- odoo.py
- secretManager.py
- credentials
- Dockerfile
- campaign.py
- campaignMqConsumer.py

1.2.2.6.2. Dockerfile

```
FROM python:3.10
```

```
ADD secretManager.py campaign.py odoo.py campaignMqConsumer.py ./
```

```
ADD credentials /root/.aws/
```

```
RUN pip install python-multipart pandas anyio==3.7.0 boto3==1.26.160 botocore==1.29.160
certifi==2023.5.7 charset-normalizer==3.1.0 click==8.1.3 datadog==0.45.0 decouple==0.0.7
fastapi==0.98.0 h11==0.14.0 httptools==0.5.0 idna==3.4 jmespath==1.0.1 pydantic==1.10.9
python-dateutil==2.8.2 python-dotenv==1.0.0 PyYAML==6.0 requests==2.31.0
s3transfer==0.6.1 six==1.16.0 sniffio==1.3.0 starlette==0.27.0 typing_extensions==4.6.3
urllib3==1.26.16 uvicorn==0.22.0 uvloop==0.17.0 watchfiles==0.19.0 websockets==11.0.3
```

```
CMD ["python", "campaignMqConsumer.py"]
```

1.2.2.7. Contactos (B2B)

1.2.2.7.1. Archivos a desplegar

- odoo.py
- secretManager.py
- credentials
- Dockerfile
- contact_b2b.py
- contact_b2bMqConsumer.py

1.2.2.7.2. Dockerfile

```
FROM python:3.10
```

```
ADD secretManager.py contact_b2b.py odoo.py contact_b2b.py ./
```

```
ADD credentials /root/.aws/
```

```
RUN pip install python-multipart pandas anyio==3.7.0 boto3==1.26.160 botocore==1.29.160  
certifi==2023.5.7 charset-normalizer==3.1.0 click==8.1.3 datadog==0.45.0 decouple==0.0.7  
fastapi==0.98.0 h11==0.14.0 httptools==0.5.0 idna==3.4 jmespath==1.0.1 pydantic==1.10.9  
python-dateutil==2.8.2 python-dotenv==1.0.0 PyYAML==6.0 requests==2.31.0  
s3transfer==0.6.1 six==1.16.0 sniffio==1.3.0 starlette==0.27.0 typing_extensions==4.6.3  
urllib3==1.26.16 uvicorn==0.22.0 uvloop==0.17.0 watchfiles==0.19.0 websockets==11.0.3
```

```
CMD ["python", "contact_b2bMqConsumer.py"]
```

1.2.2.8. Contactos (B2C)

1.2.2.8.1. Archivos a desplegar

- odoo.py

- secretManager.py
- credentials
- Dockerfile
- contact_b2c.py
- contact_b2cMqConsumer.py

1.2.2.8.2. Dockerfile

FROM python:3.10

ADD secretManager.py contact_b2c.py odoo.py contact_b2c.py ./

ADD credentials /root/.aws/

RUN pip install python-multipart pandas anyio==3.7.0 boto3==1.26.160 botocore==1.29.160 certifi==2023.5.7 charset-normalizer==3.1.0 click==8.1.3 datadog==0.45.0 decouple==0.0.7 fastapi==0.98.0 h11==0.14.0 httpptools==0.5.0 idna==3.4 jmespath==1.0.1 pydantic==1.10.9 python-dateutil==2.8.2 python-dotenv==1.0.0 PyYAML==6.0 requests==2.31.0 s3transfer==0.6.1 six==1.16.0 sniffio==1.3.0 starlette==0.27.0 typing_extensions==4.6.3 urllib3==1.26.16 uvicorn==0.22.0 uvloop==0.17.0 watchfiles==0.19.0 websockets==11.0.3

CMD ["python", "contact_b2cMqConsumer.py"]

1.3. Creación de colas de mensajes

Cada consumidor consume los mensajes recibidos de su respectiva cola de mensajes, las cuales han de crearse en Amazon Simple Queue Service (SQS). De las muchas configuraciones posibles a realizar, la única de vital importancia es que la cola sea de tipo FIFO (First In First Out) lo cual minimiza los problemas del orden de ingreso de nuevas entidades.

Details

Type
Choose the queue type for your application or cloud infrastructure.

Standard Info
At-least-once delivery, message ordering isn't preserved

- At-least once delivery
- Best-effort ordering

FIFO Info
First-in-first-out delivery, message ordering is preserved

- First-in-first-out delivery
- Exactly-once processing


 You can't change the queue type after you create a queue.

Figura 68: Selección FIFO en SQS

1.3.1. Nombres de las colas

Es importante que el nombre de las colas sean los correctos, dado que los diferentes consumidores utilizan el identificador para saber de qué cola consumir.

- EvimedClientQueue.fifo
- EvimedCompanyQueue.fifo
- EvimedCoursesQueue.fifo
- EvimedInscriptionQueue.fifo
- EvimedInterestsQueue.fifo
- EvimedCampaignQueue.fifo
- EvimedContactB2BQueue.fifo
- EvimedContactB2CQueue.fifo

1.4. Credenciales AWS

1.4.1. Archivo Credentials

El archivo “credentials” al cual se le hace referencia en todos los listados de archivos a desplegar, es necesario para que el Adapter pueda comunicarse con los diferentes servicios de AWS.

El archivo debe contener las credenciales de la misma cuenta de AWS que realiza el despliegue correspondiente. Estas credenciales se encuentran en la sección “My Security Credentials” del perfil de AWS.

1.5. Región de AWS

En diferentes lugares del código fuente, está especificada la región en la que se encuentran los servicios de AWS a utilizar, en nuestro caso es “us-east-1” debido a que es la región disponible en la licencia académica que nos provee la facultad. En caso de utilizar una región diferente, importante que sean cambiados estos valores en:

- main.py
- secretManager.py

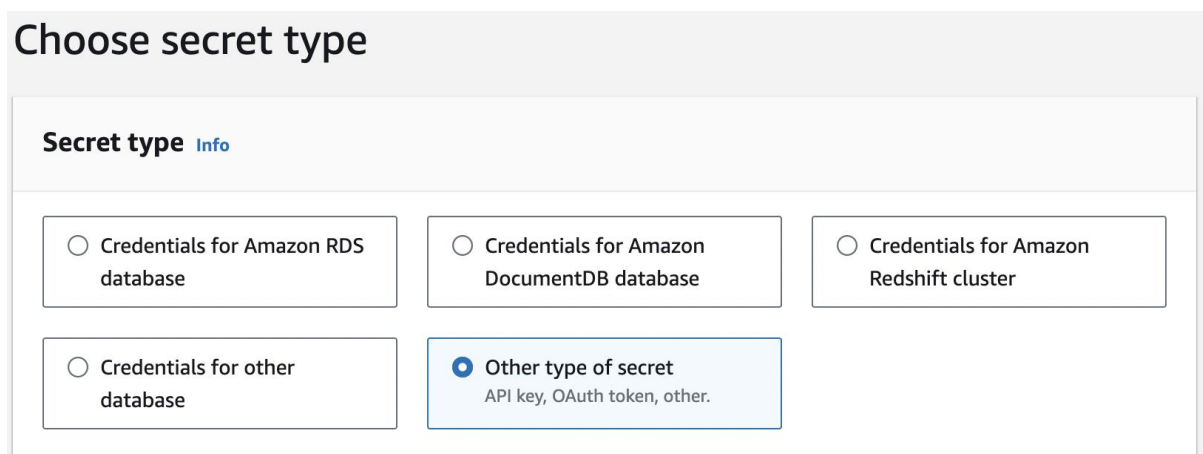
1.6. Secret Manager

Para almacenar las credenciales de Odoo y api keys necesarias de forma segura y acceder a ellas fácilmente, utilizamos AWS Secret Manager. Estas han de ser creadas desde dicho servicio.

1.6.1. Configuraciones importantes

1.6.1.1. Tipo de secreto

El tipo de secreto a almacenar debe ser “otro tipo de secreto” ya que lo vamos a utilizar desde nuestra aplicación y no directamente desde otro servicio de aws.



The screenshot shows the 'Choose secret type' dialog in the AWS console. It features a title bar 'Choose secret type' and a sub-header 'Secret type Info'. Below this, there are five radio button options arranged in two rows. The first row contains three options: 'Credentials for Amazon RDS database', 'Credentials for Amazon DocumentDB database', and 'Credentials for Amazon Redshift cluster'. The second row contains two options: 'Credentials for other database' and 'Other type of secret'. The 'Other type of secret' option is selected, indicated by a blue dot and a blue border around its box. Below the selected option, the text 'API key, OAuth token, other.' is visible.

Figura 69: Selección de tipo de secreto

1.6.1.2. Claves de las credenciales

Estas son las claves de las credenciales, las cuales, de la misma forma que se explicó en la sección de los nombres de las colas de mensajes, deben ser idénticas a las que definimos a continuación.

- API_KEY_ODOO (Api key de la cuenta asociada a Odoo)
- URL_ODOO (https://evimed.odoo.com/)

- DB_ODOO (EviMed)
- USERNAME (Correo electrónico de la cuenta asociada a Odo)

1.6.1.3. Nombre del secreto

- prod/evimed/odoo

1.7. Descripción de los endpoints expuestos por la API

1.7.1. Client

1.7.1.1. Retrieve by email (GET)

- Parámetros:
 - email (Query)
- URI: /clients

1.7.1.2. Create (POST)

- Parámetros: Ninguno
- URI: /clients
- Cuerpo: Datos del cliente (name, email, country, phone, temperature, title, specialties, language)

1.7.1.3. Update by email (PATCH)

- Parámetros:
 - email (URI)
- URI: /clients/{email}
- Cuerpo: Datos actualizados del cliente (name, email, country, phone, temperature, title, specialties, language)

1.7.1.4. Delete by email (DELETE)

- Parámetros:
 - email (Query)

- URI: `/clients`

1.7.2. Company

1.7.2.1. Retrieve by name (GET)

- Parámetros:
 - name (Query)
- URI: `/companies`

1.7.2.2. Create (POST)

- Parámetros: Ninguno
- URI: `/companies`
- Cuerpo: Datos de la compañía (name, contacts, country)

1.7.2.3. Update by name (PATCH)

- Parámetros:
 - name (URI)
- URI: `/companies/{name}`
- Cuerpo: Datos actualizados de la compañía (name, contacts)

1.7.2.4. Delete by name (DELETE)

- Parámetros: Ninguno
- URI: `/companies/{name}`

1.7.3. Education Activity

1.7.3.1. Retrieve by name (GET)

- Parámetros:
 - name (Query)
- URI: `/education_activity`

1.7.3.2. Update by name (PATCH)

- **Parámetros:**
 - name (URI)
- **URI:** /education_activity/{name}
- **Cuerpo:** Datos actualizados de la actividad educativa (name, dates, specialties)

1.7.3.3. Create (POST)

- **Parámetros:** Ninguno
- **URI:** /education_activity
- **Cuerpo:** Datos de la actividad educativa (name, dates, and specialties)

1.7.3.4. Delete by name (DELETE)

- **Parámetros:**
 - name (URI)
- **URI:** /education_activity/{name}

1.7.4. Interest

1.7.4.1. Retrieve by education activity name and client email (GET)

- **Parámetros:**
 - education_activity_name (Query)
 - client_email (Query)
- **URI:** /interests

1.7.4.2. Create (POST)

- **Parámetros:** Ninguno
- **URI:** /interests
- **Cuerpo:** Datos de interés (creation_date, client_email, and education_activity_name)

1.7.4.3. Update by education activity name and client email (PATCH)

- **Parámetros:**
 - `education_activity_name` (Query)
 - `client_email` (Query)
- **URI:** `/interests/{id}`
- **Cuerpo:** Datos actualizados de interés (`creation_date`, `education_activity_name`)

1.7.4.4. Delete (DELETE)

- **Parámetros:**
 - `id` (URI)
- **URI:** `/interests/{id}`

1.7.5. Inscription

1.7.5.1. Get by educational_activity_name and client_email (GET)

- **Parámetros:**
 - `education_activity_name` (Query)
 - `client_email` (Query)
- **URI:** `/inscriptions`

1.7.5.2. Update (PUT)

- **Parámetros:** Ninguno
- **URI:** `/inscriptions`
- **Cuerpo:** Datos actualizados de la inscripción (`creation_date`, `client_email`, `education_activity_name`)

1.7.5.3. Create (POST)

- **Parámetros:** Ninguno
- **URI:** `/inscriptions`
- **Cuerpo:** Datos de la inscripción (`client_email`, `education_activity_name`, `payment_date`, `payment_method`, `price`, `evaluation_results`)

1.7.5.4. Delete by ID (DELETE)

- **Parámetros:**
 - id (URI)
- **URI:** /inscriptions/{id}

1.7.6. Contact B2C

1.7.6.1. Retrieve by name (GET):

- **URI:** /contacts/b2c?name={{name}}
- **Parámetros:**
 - name (URI)

1.7.6.2. Update (PATCH):

- **URI:** /contacts/b2c/{{name}}
- **Cuerpo:**
 - name
 - date
 - action
 - contact_type
 - client_email
 - education_activity_name
 - summary

1.7.6.3. Create (POST):

- **URI:** /contacts/b2c
- **Cuerpo:**
 - name
 - date
 - action
 - contact_type
 - client_email
 - education_activity_name

- summary

1.7.6.4. Delete by id (DELETE):

- URI: /contacts/b2c/{{id}}
- Parámetros: id (URI)

1.7.7. Contact B2B

1.7.7.1. Retrieve by name (GET):

- URI: /contacts/b2b?name={{name}}
- Parámetros:
 - name (URI)

1.7.7.2. Update (PATCH):

- URI: /contacts/b2b/{{name}}
- Parámetros: name (URI)
- Cuerpo:
 - name
 - date
 - action
 - contact_type
 - client_email
 - company_name
 - education_activity_name
 - summary

1.7.7.3. Create (POST):

- URI: /contacts/b2b
- Cuerpo:
 - name
 - date
 - action

- contact_type
- client_email
- company_name
- education_activity_name
- summary

1.7.7.4. Delete by id (DELETE):

- URI: /contacts/b2b/{{id}}
- Parámetros: id (URI)

1.7.8. Campaign

1.7.8.1. Retrieve by education activity and client (GET):

- URI:
/campaigns?education_activity_name={{name}}&client_email={{email}}
}
- Parámetros: education_activity_name, client_email (URI)

1.7.8.2. Update (PATCH):

- URI: /campaigns/{{id}}
- Parámetros: id (URI)
- Cuerpo:
 - name
 - date
 - type
 - description
 - contacts

1.7.8.3. Create (POST):

- URI: /campaigns
- Cuerpo:
 - name

- date
- type
- description
- contacts

1.7.8.4. Delete by id (DELETE):

- URI: `/campaigns/{{id}}`
- Parámetros: `id` (URI)

2. Odoo

La URL para acceder a la web de Odoo es: evimed.odoo.com. Las credenciales de las licencias que fueron utilizadas para desarrollo son las siguientes.

- Correo: crm1.evimed@gmail.com, contraseña: EvimedOrt2023.
- Correo: crm2.evimed@gmail.com, contraseña: EvimedOrt2023.
- Correo: crm3.evimed@gmail.com, contraseña: EvimedOrt2023.

A continuación se explicará el acceso a dos funcionalidades importantes de Odoo al momento de desarrollar.

2.1. Modo de desarrollador

El modo de desarrollador permite al usuario visualizar información de la plataforma a la que normalmente no tendría acceso. Resultó de mucha utilidad al momento de realizar *debugging* de diferentes partes de la aplicación.

Para acceder al modo de desarrollador, en el menú principal seleccionar Ajustes.



Figura 70: Opción ajustes del menú principal

Una vez allí, *scrollear* hasta el final de la página y seleccionar “activar modo desarrollador”

Figura 71: Menú de ajustes

Se puede ver si el modo de desarrollador está activado por menú de bicho que aparece en la parte superior derecha de la pantalla.



Figura 72: Indicador de activación del modo de desarrollador

2.2. Odo studio

Odo studio es una herramienta muy necesaria, es lo que permitió al equipo la creación de las diferentes vistas y formularios que se ven a lo largo de toda la plataforma. Para acceder a ella, seleccionar el botón de “lápiz y llave” que se ve en la figura anterior.

Dentro de esta herramienta, se encuentran dos cosas muy relevantes que han de ser destacadas.

- Catálogo de campos a agregar: muestra los diferentes campos que son posibles de agregar a la vista (se agregan mediante *drag and drop*)



Figura 73: Catálogo de campos

- Tipo de vista: aquí es donde se selecciona con qué vista dentro de una sección de la plataforma se está trabajando. Como ejemplo, la vista que posee un listado de todos los clientes y permite filtrarlos es la vista de **lista** de la sección de Clientes. Por otro lado, la vista que permite el ingreso o modificación de los clientes es la vista de **formulario** de la sección de Clientes. Se puede acceder a estas vistas diferentes para modificarlas dentro de Odoo studio con los botones de arriba a la derecha del mismo.

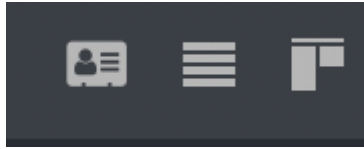


Figura 74: Botones para cambios de vista

Name	Email	Phone	País
Aaron Bean	ai_759f_oo@live.com.mx	079-327-5332x76826	México
Aaron Coleman	ab_e1e0_11@hotmail.com	1-613-408-7736x75877	Venezuela
Aaron Compton	ai_a904_c3@gmail.com	741.920.6381x187	Argentina
Aaron Fowler	ad_047f_23@gmail.com	217.769.7030x2338	Ecuador
Aaron Gentry	az_adde_ra@gmail.com	1-630-519-8474x15902	México
Aaron Hardy	na_6d1b_03@gmail.com	(159)811-1722x94159	Brasil
Aaron Haynes	gu_1c13_04@live.com.mx	823.591.2556x7607	España
Aaron Johnson	jc_7409_12@hotmail.com	(395)341-3786x864	Colombia
Aaron Johnson	ta_d39c_un@gmail.com	(857)643-3138x75584	Ecuador
Aaron Jones	a_f9b7_a1@gmail.com	937-335-9503	Chile
Aaron Lara	ad_8202_02@gmail.com	005-507-0728x973	Perú
Aaron Martinez	fg_7cba_08@hotmail.com	888-343-1517x456	Bolivia
Aaron Reilly	AP_c0e1_to@cas.austral.edu.ar	(843)051-1323x984	Brasil
Aaron Reyes	al_3e55_12@gmail.com	+47(3)1685309254	Ecuador

Figura 75: Vista de lista de Clientes

Añadir una barra de estado de flujo de ventas

Aaron Bean Image

<p>Email ? ai_759f_oo@live.com.mx</p> <p>Phone ? 079-327-5332x76826 SMS</p> <p>País ? México</p> <p>Temperatura ? Normal</p>	<p>Title ? Doctor</p> <p>Especialidad/es ? Enfermería</p> <p>Language ? Spanish / Español</p>
---	--

Internal Notes
Empresas
Intereses
Contactos
Inscripciones
+

Figura 76: Vista de formulario de Clientes

3. Extensión de Chrome

Para seguir esta guía será necesario contar con “Git Bash”, algún editor de texto como “Visual Studio Code”, o también algún IDE como “WebStorm” de JetBrains.

3.1. Chrome Web Store

La extensión se encuentra deployada en la Chrome Web Store: <https://tinyurl.com/Odoo-CRM-Marketing-Assistant>

3.2. Repositorio

Para ingresar al repositorio es necesario que el usuario sea invitado a participar del mismo.

Link: <https://github.com/juanNervi/odoo-chrome-extension>

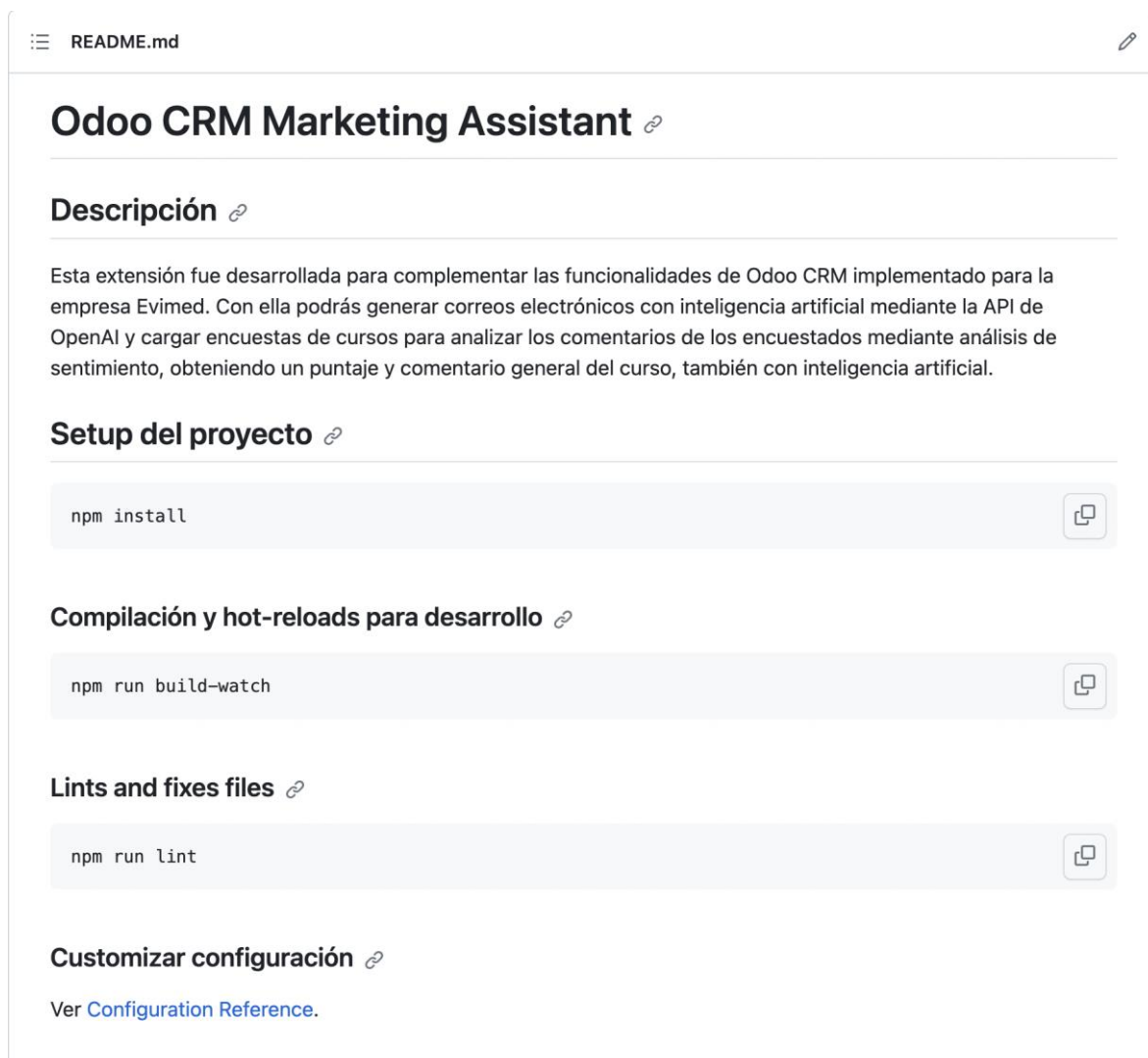


Figura 77: Readme del repositorio Odoo CRM Marketing Assistant

En él se especifican los comandos utilizados por el equipo para desarrollar la extensión. El primer paso a realizar, es clonar el repositorio de Odoo CRM Marketing Assistant desde el GitHub. Luego hay que instalar las dependencias del proyecto con el comando especificado en el *readme* ***npm install*** (Figura 78). El siguiente paso para poder desarrollar o mantener el software es correr el comando ***npm run build-watch*** (Figura 79) de modo que el código compile y al realizar un cambio en el mismo, se visualice en la interfaz (*hot-reloads*). Con el comando anterior se generará una carpeta de código compilado llamada *dist*. Dicha carpeta es donde vive el código de nuestra extensión y la que hay que subir a Google Chrome para visualizar los cambios.

```
Terminal: Local x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\NerviJu\OneDrive - BASF\Desktop\Repos\odoo-chrome-extension> npm install
```

Figura 78: Terminal de WebStorm IDE utilizando el comando “*npm install*”

```
PS C:\Users\NerviJu\OneDrive - BASF\Desktop\Repos\odoo-chrome-extension> npm run build-watch

> my-project@0.1.0 build-watch
> vue-cli-service --env.NODE_ENV=development build-watch --mode development

All browser targets in the browserslist configuration have supported ES module.
Therefore we don't build two separate bundles for differential loading.

| Building for development...

DONE Compiled successfully in 21918ms

File                Size
dist\popup.js       14994.44 KiB
dist\background.js  0.67 KiB
dist\background.js  0.67 KiB
dist\content.js     0.66 KiB

Images and other types of assets omitted.
Build at: 2023-09-25T15:23:44.701Z - Hash: 9be013cbcf8b828c - Time: 975ms

DONE Build complete. The dist directory is ready to be deployed.
INFO Check out deployment instructions at https://cli.vuejs.org/guide/deployment.html
```

Figura 79: Terminal de WebStorm IDE utilizando el comando “*npm run build-watch*”

El proceso para subir una extensión que no esté deployada en la Chrome Web Store es el siguiente:

1. En un navegador de Google Chrome o en el navegador de su preferencia busque las configuraciones del mismo.
2. Luego dirigirse a la sección “Extensiones” y hacer click en “Gestionar extensiones”.

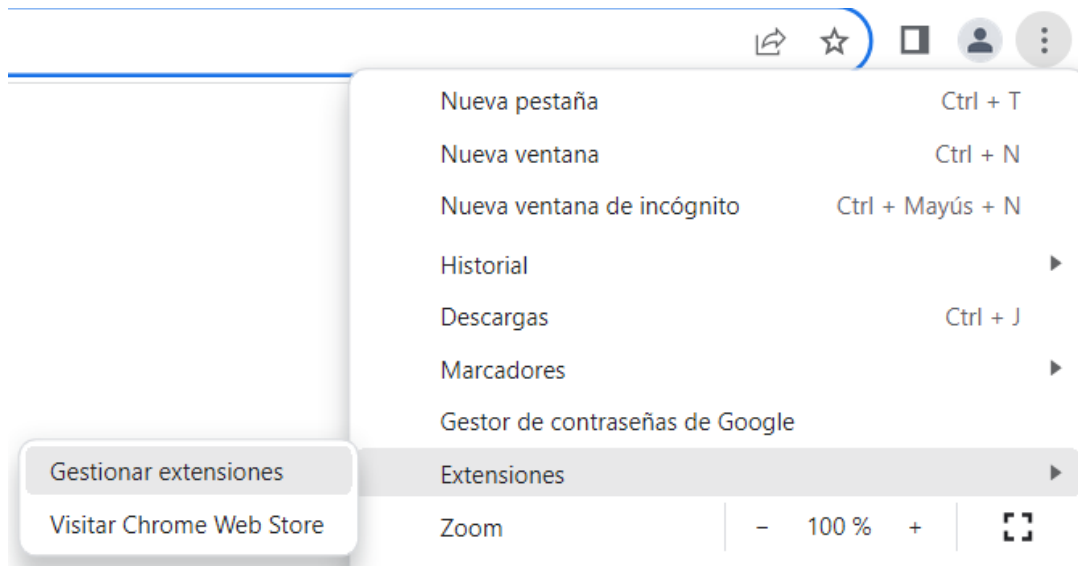


Figura 80: Pasos 1 y 2 a seguir en Google Chrome

3. Una vez dentro del apartado de “Gestionar extensiones” es necesario habilitar el modo desarrollador del navegador situado en el extremo superior derecho de la interfaz.
4. Al realizar el paso anterior, se desplegarán varias opciones, para cargar nuestra extensión daremos click en “Carga descomprimida”.

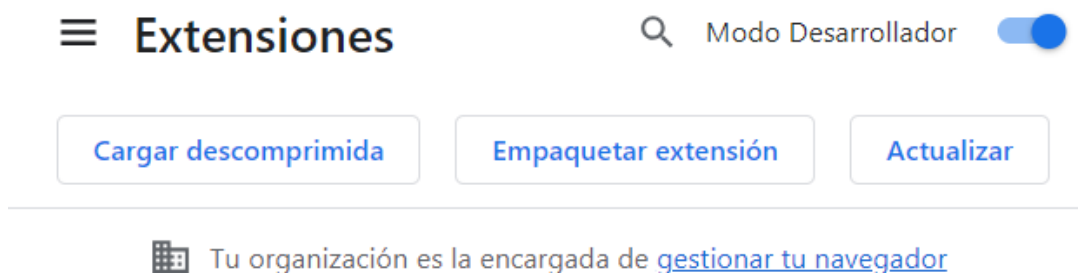


Figura 81: Pasos 3 y 4 a seguir en Google Chrome

5. Al realizar el paso anterior se abrirá un *file chooser* en el cual debemos navegar hasta nuestra extensión y elegir la carpeta “*dist*” previamente generada al correr el comando ***npm run build-watch*** explicado anteriormente. Luego hacer click en seleccionar carpeta.

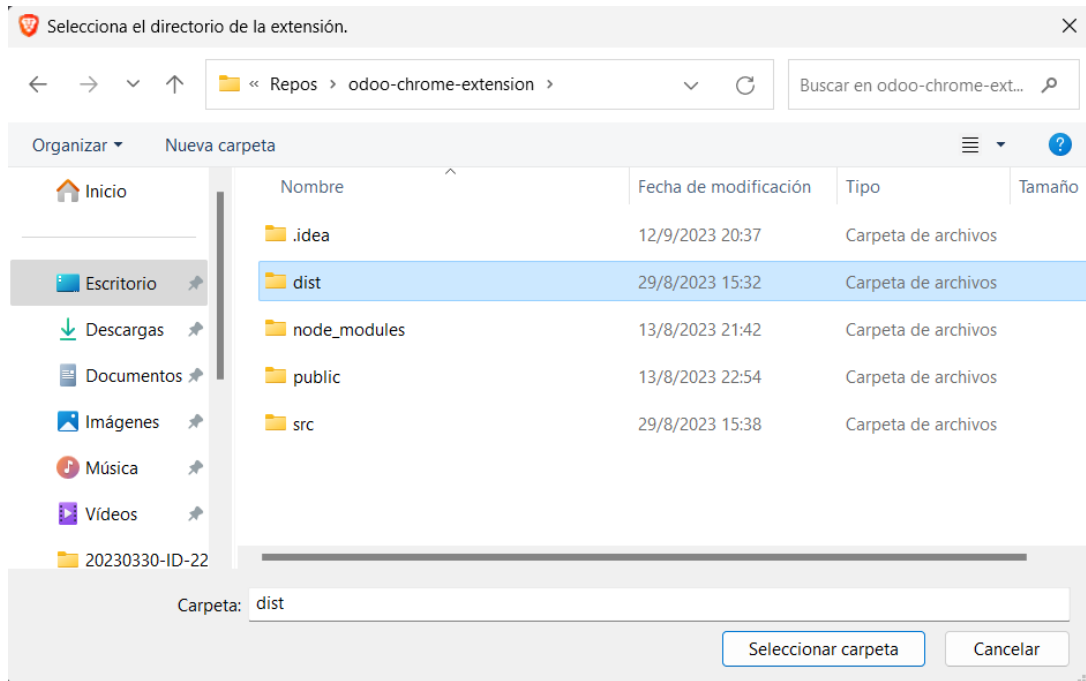


Figura 82: File chooser de Google Chrome

- Una vez realizado el paso anterior, nuestra extensión quedará cargada como una extensión más de nuestro navegador.

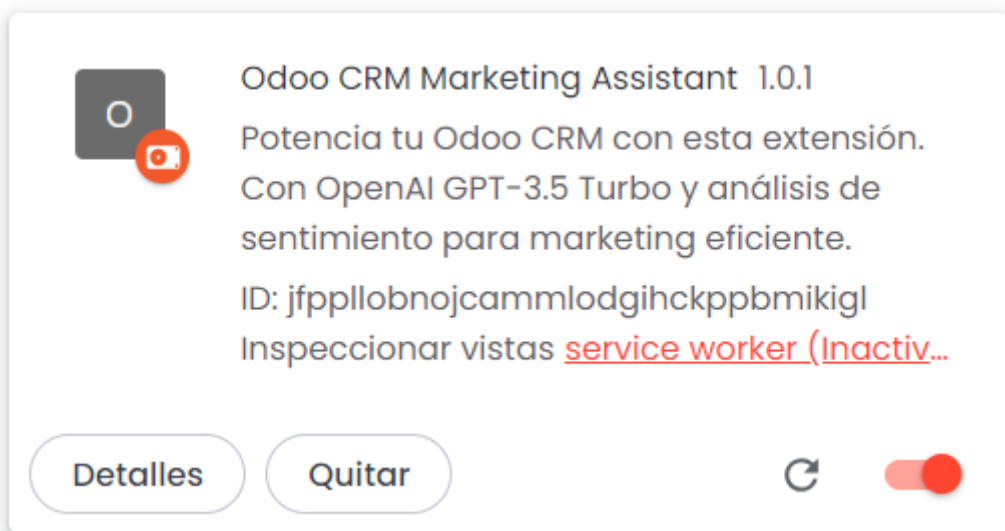


Figura 83: Extensión “Odoo CRM Marketing Assistant” cargada en Google Chrome

Por último, si se quiere que Odoo CRM Marketing Assistant aparezca “fijada” en la barra del navegador será necesario presionar el “pin” azul mostrado en la Figura 84.

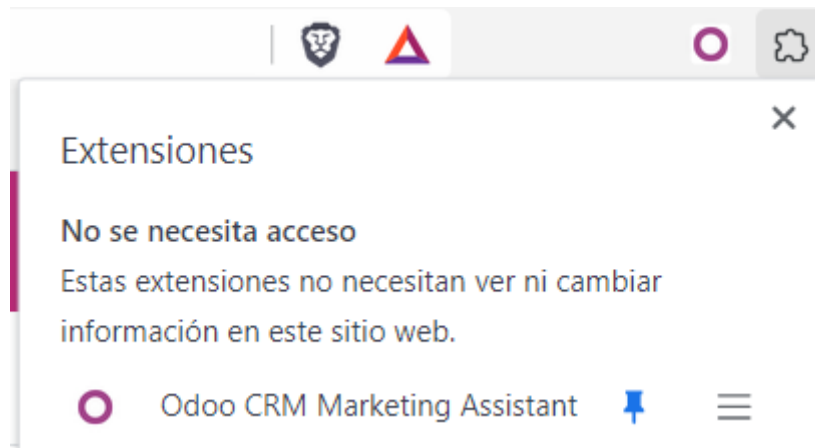



Figura 84: Barra de navegación de Google Chrome

10.4. Carta de conformidad del cliente



Montevideo, 22 de setiembre de 2023

Sres. Universidad ORT

PRESENTE

De nuestra consideración:

Por medio de la presente y a solicitud de la Universidad les informamos que el equipo de estudiantes conformado por Marcos Bazzano, Juan Andrés Nervi y Romina Rodríguez

En nuestra opinión, en una escala donde 1 representa totalmente insatisfactorio y 5 representa totalmente satisfactorio, el desempeño del equipo se puede evaluar como:

	1	2	3	4	5
<i>Compromiso con el éxito del proyecto</i>					X
<i>Trato profesional, cordial y respetuoso con nuestro personal</i>					X
<i>Calidad del software recibido</i>					X

Basamos nuestra opinión en que el grupo desde las reuniones previas de presentación del proyecto ya mostraron interés genuino en el mismo y luego durante la ejecución se involucraron para llevarlo adelante con el alcance y calidad requerido respetando el cronograma planificado. Las reuniones quincenales de coordinación se realizaron sin excepción, y siempre con un trato muy ameno y profesional. El resultado obtenido supera con creces las expectativas de nuestra empresa.

Quedamos a disposición por si se necesita información adicional.

Saludos cordiales,

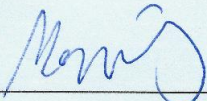

Antonio Lopez
Director TI Por EviMed

Figura 85: Carta de conformidad del cliente