

	UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA			
	Documento	Código	Fecha	Revisión
	FORMATO HOJA DE RESUMEN PARA TRABAJO DE GRADO	F-AC-DBL-007	10-04-2012	A
Dependencia	Aprobado		Pág.	
DIVISIÓN DE BIBLIOTECA	SUBDIRECTOR ACADEMICO		i(95)	

RESUMEN – TRABAJO DE GRADO

AUTORES	JEIMMY CAROLINA URIBE LOZANO		
FACULTAD	INGENIERIAS		
PLAN DE ESTUDIOS	INGENIERIA DE SISTEMAS		
DIRECTOR	WILMAR ALIRIO GONZALEZ PEINADO		
TÍTULO DE LA TESIS	DESARROLLO DE PLUGIN SMS PARA EL SOFTWARE CRM DE LA EMPRESA TECHNET COMUNICACIONES S.A.S DE LA CIUDAD DE OCAÑA, NORTE DE SANTANDER.		
RESUMEN			
(70 palabras aproximadamente)			
<p>EL PRESENTE DOCUMENTO CONTIENE EL DESARROLLO DE UN PLUGIN DE MENSAJERIA PARA EL CRM (GESTION DE LAS RELACIONES CON CLIENTES) DEL SOFTWARE UNMS (SISTEMA DE GESTION DE RED DE UBIQUITI) DE LA EMPRESA TECHNET COMUNICACIONES S.A.S UBICADA EN OCAÑA, NORTE DE SANTANDER, CON EL FIN DE MANTENER INFORMADOS A LOS CLIENTES A TRAVES DEL ENVIO DE MENSAJES CORTOS (SMS) Y ASÍ MEJORAR LA COMUNICACION ENTRE LA ORGANIZACION Y SUS CLIENTES.</p>			
CARACTERÍSTICAS			
PÁGINAS: 95	PLANOS:	ILUSTRACIONES:108	CD-ROM:



**DESARROLLO DE PLUGIN SMS PARA EL SOFTWARE CRM DE LA EMPRESA
TECHNET COMUNICACIONES S.A.S DE LA CIUDAD DE OCAÑA, NORTE DE
SANTANDER**

AUTOR:

JEIMMY CAROLINA URIBE LOZANO

Trabajo de Grado presentado como requisito para optar al título de Ingeniero de Sistemas

Director

Ing. WILMAR ALIRIO GONZALEZ PEINADO

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS

Ocaña, Colombia

Noviembre, 2020

Índice

Resumen.....	11
Introducción	12
1. Desarrollo de Plugin SMS para el Software CRM de la empresa TechNet Comunicaciones S.A.S de la ciudad de Ocaña.....	1
1.1 TechNet Comunicaciones S.A.S.....	1
1.1.1 Misión.....	1
1.1.2 Visión.....	1
1.1.3 Objetivos de la empresa.....	1
1.1.4 Estructura organizacional.....	2
1.1.5 Dependencia y/o proyecto asignado.....	3
1.2 Diagnóstico inicial de la dependencia asignada.....	3
1.2.1 Planteamiento del problema.....	4
1.3 Objetivos de la pasantía.....	5
1.3.1 General.....	5
1.3.2 Específicos.....	5
1.4 Actividades a desarrollar.....	6
2. Enfoques referenciales.....	7
2.1 Enfoque conceptual.....	7
2.2 Enfoque legal.....	8
3. Informe de cumplimiento de trabajo	11
3.1 Presentación de resultados	11
3.1.1 Objetivo 1. Realizar una investigación sobre la creación de plugins y su implementación en el software de TechNet Comunicaciones S.A.S.....	11
3.1.2 Objetivo 2. Desarrollar el plugin de mensajería para el software CRM (Gestión de relaciones con los clientes) de la organización.....	56
3.1.3 Objetivo 3. Realizar las pruebas al plugin en la infraestructura tecnológica de la empresa.....	72
3.1.4 Objetivo 4. Implementar el plugin de mensajería para el software CRM (Gestión de relaciones con los clientes) de TechNet Comunicaciones S.A.S.....	76
4. Diagnostico final.....	77
5. Conclusiones.....	78

6. Recomendaciones	80
7. Referencias	81

Lista de figuras

<i>Figura 1.</i> Estructura organizacional (TechNet Comunicaciones S.A.S).....	2
<i>Figura 2.</i> Lenguaje del Sistema.....	13
<i>Figura 3.</i> Configuración e identificación de teclado	13
<i>Figura 4.</i> Autodetección del teclado.....	14
<i>Figura 5.</i> Autodetección del teclado.....	14
<i>Figura 6.</i> Teclado recomendado "Español (latinoamericano)"	15
<i>Figura 7.</i> Conexiones de red.....	15
<i>Figura 8.</i> Configuración de sistema de archivos	16
<i>Figura 9.</i> Configuración del perfil.....	16
<i>Figura 10.</i> Instalación del sistema.....	17
<i>Figura 11.</i> Logueo exitoso.....	17
<i>Figura 12.</i> Actualización de paquetes instalados	18
<i>Figura 13.</i> Configuración de IP estática.....	18
<i>Figura 14.</i> Configuración IP estática.....	19
<i>Figura 15.</i> Configuración IP estática.....	19
<i>Figura 16.</i> Dirección IP estática asignada	20
<i>Figura 17.</i> Comando "sudo apt-get update"	20
<i>Figura 18.</i> Comando "sudo apt-get install curl sudo bash"	21
<i>Figura 19.</i> Instalación de UNMS.....	21
<i>Figura 20.</i> Creación de la cuenta en UNMS.....	22
<i>Figura 21.</i> Vault key.....	22
<i>Figura 22.</i> Rango de IPs	23
<i>Figura 23.</i> Interfaz de inicio de UNMS.....	23
<i>Figura 24.</i> Tablero (Network)	24
<i>Figura 25.</i> Sitios (Módulo de red)	25
<i>Figura 26.</i> Sitios clientes (Módulo de red).....	25
<i>Figura 27.</i> Dispositivos (Módulo de red)	26
<i>Figura 28.</i> Reportes (Módulo de red).....	26
<i>Figura 29.</i> Administrador de firmware (Módulo de red)	27
<i>Figura 30.</i> Administrador de tareas (Módulo de red).....	27
<i>Figura 31.</i> Apagones (Módulo de red)	28
<i>Figura 32.</i> Ajustes (Módulo de red)	28
<i>Figura 33.</i> Tablero (Módulo CRM).....	29
<i>Figura 34.</i> Clientes (Módulo CRM).....	30
<i>Figura 35.</i> Facturas (Módulo CRM).....	30
<i>Figura 36.</i> Pagos (Módulo CRM).....	31
<i>Figura 37.</i> Planificación (Módulo CRM)	31
<i>Figura 38.</i> Ticketing (Módulo CRM).....	32

<i>Figura 39.</i> Reportes, uso de datos (Módulo CRM)	33
<i>Figura 40.</i> Reportes, facturación (Módulo CRM)	33
<i>Figura 41.</i> Ajustes (Módulo CRM)	34
<i>Figura 42.</i> Descarga de PHP	35
<i>Figura 43.</i> Carpeta PHP7	35
<i>Figura 44.</i> Ubicación de “php.ini”	36
<i>Figura 45.</i> Archivo "php.ini"	36
<i>Figura 46.</i> Variables de entorno	37
<i>Figura 47.</i> Variable "Path"	38
<i>Figura 48.</i> Creación de la variable "PHP7"	38
<i>Figura 49.</i> Instalación correcta de PHP	39
<i>Figura 50.</i> Descarga de git.....	40
<i>Figura 51.</i> Proceso de instalación.....	40
<i>Figura 52.</i> Componentes a instalar	41
<i>Figura 53.</i> Elección del editor	41
<i>Figura 54.</i> Ajuste de variables de entorno	42
<i>Figura 55.</i> Librería OpenSSL.....	42
<i>Figura 56.</i> Caracteres de fin de línea.....	43
<i>Figura 57.</i> Emulador MinTTY	43
<i>Figura 58.</i> Configuraciones extras de Git.....	44
<i>Figura 59.</i> Inicio de instalación de Git	44
<i>Figura 60.</i> Descarga de Composer	45
<i>Figura 61.</i> Instalación de Composer.....	45
<i>Figura 62.</i> Creación "pluginprueba"	49
<i>Figura 63.</i> Archivo ZIP esqueleto	50
<i>Figura 64.</i> Archivo “manifest.json”	52
<i>Figura 65.</i> Instalación de complementos.....	52
<i>Figura 66.</i> Archivo "public.php"	53
<i>Figura 67.</i> Archivo "main.php"	53
<i>Figura 68.</i> Comando “./vendor/bin/pack-plugin”	54
<i>Figura 69.</i> Archivo "plugin_prueba" comprimido	54
<i>Figura 70.</i> Proceso de "Subir complemento"	55
<i>Figura 71.</i> Cargar "plugin_prueba" al sistema	55
<i>Figura 72.</i> Plugin cargado a UNMS	56
<i>Figura 73.</i> Comunicación entre la API del servidor de SMS y UNMS	57
<i>Figura 74.</i> Función sendMessage()	58
<i>Figura 75.</i> Función getDataApi()	58
<i>Figura 76.</i> Función getSMS().....	59
<i>Figura 77.</i> main.php	60
<i>Figura 78.</i> public.php	60

<i>Figura 79.</i> manifest.json	61
<i>Figura 80.</i> composer.json	61
<i>Figura 81.</i> composer.lock	62
<i>Figura 82.</i> gitignore	62
<i>Figura 83.</i> ApiData.php	63
<i>Figura 84.</i> NotificationData.php	63
<i>Figura 85.</i> PluginData.php	64
<i>Figura 86.</i> UcrmData.php	64
<i>Figura 87.</i> CurlException.php	65
<i>Figura 88.</i> AbstractMessageNotifierFacade.php	65
<i>Figura 89.</i> TechnetNotifierFacade.php	66
<i>Figura 90.</i> ApiDataFactory.php	66
<i>Figura 91.</i> MessageTextFactory.php	67
<i>Figura 92.</i> NotificationDataFactory.php	67
<i>Figura 93.</i> ApiTechnetSMS.php	68
<i>Figura 94.</i> CurlExecutor.php	68
<i>Figura 95.</i> Logger.php	69
<i>Figura 96.</i> OptionsManager.php	69
<i>Figura 97.</i> PluginDataValidator.php	70
<i>Figura 98.</i> SmsNumberProvider.php	70
<i>Figura 99.</i> UcrmApi.php	71
<i>Figura 100.</i> Plugin.php	71
<i>Figura 101.</i> Plugin SMS en UNMS	72
<i>Figura 102.</i> Configuración del plugin SMS	73
<i>Figura 103.</i> Creación del cliente	73
<i>Figura 104.</i> Nombre de cliente modificado	74
<i>Figura 105.</i> Pago agregado	74
<i>Figura 106.</i> Mensajes cortos recibidos	75
<i>Figura 107.</i> Event/Request	75
<i>Figura 108.</i> Response Body	76

Lista de tablas

Tabla 1	3
Tabla 2	6

Resumen

Para este proyecto, fue llevado a cabo el desarrollo de un plugin de mensajería, para el CRM del software UNMS utilizado por la empresa TechNet Comunicaciones S.A.S ubicada en Ocaña, Norte de Santander, la cual se caracteriza por su servicio de calidad, buscando siempre satisfacer las necesidades de sus clientes con la creación de estrategias innovadoras y la incorporación de nuevas herramientas. La funcionalidad principal del complemento es mantener informados a los clientes a través del envío de SMS (mensajes cortos), así pues, mejorar la comunicación entre la organización y sus clientes.

Inicialmente se realizó un estudio sobre la creación de plugins y su implementación en el software de la empresa. A continuación de ese análisis, se efectuó el desarrollo del complemento para su posterior implementación en el CRM (Gestión de las relaciones con clientes) de UNMS (Sistema de gestión de red de Ubiquiti).

Introducción

En la actualidad, las organizaciones están expuestas a cambios y desafíos constantes, es por esto que buscan adaptarse a nuevos escenarios y entornos. Para cumplir con cada una de las exigencias las empresas han visto la necesidad de crear nuevas estrategias, como lo es el desarrollo de plugins o complementos en sus sistemas con el fin de mejorar la calidad de sus servicios y satisfacer completamente las necesidades de sus clientes.

La presencia de los plugins en los sistemas de las empresas es cada vez más notoria y el crecimiento de esta área es muy evidente, debido a que traen muchos beneficios las organizaciones desarrollan estas herramientas, pues permiten ampliar y mejorar ciertas funciones de un sistema sin la necesidad de adquirir otro sistema adicional.

Se ha desarrollado bajo la perspectiva mencionada anteriormente, un plugin de mensajería que facilite la comunicación entre el CRM (Gestión de las relaciones con clientes) y los clientes de TechNet Comunicaciones S.A.S y así brindar un servicio efectivo y de calidad.

1. Desarrollo de Plugin SMS para el Software CRM de la empresa TechNet Comunicaciones S.A.S de la ciudad de Ocaña

1.1 TechNet Comunicaciones S.A.S

TechNet Comunicaciones S.A.S es una empresa ocañera prestadora del servicio de internet que en los últimos tiempos se ha caracterizado por su excelente servicio pues cuenta con infraestructura tecnológica de calidad., buscando siempre satisfacer las necesidades de sus clientes a través de la calidad de su servicio y la innovación de los mismos.

1.1.1 Misión.

Proveer el servicio de conectividad efectiva, que satisfaga las necesidades de nuestros clientes, priorizando la privacidad, la seguridad, la disponibilidad de servicio y los tiempos de respuesta más bajos.

1.1.2 Visión.

Ser una de las empresas más influyentes en el servicio de conectividad y telecomunicaciones en el país.

1.1.3 Objetivos de la empresa.

General.

Garantizar a nuestros clientes un excelente servicio de conectividad monitoreada, con los mejores tiempos de respuesta ante cualquier interrupción del servicio.

Objetivos específicos.

- Brindar soluciones de conectividad efectiva, segura y con la disponibilidad necesaria.
- Diseñar e implementar redes a la medida de los clientes, con tecnología de punta, para brindar servicios con calidad para la transmisión de datos, Internet, IPTV y VoIP entre otros.

1.1.4 Estructura organizacional.

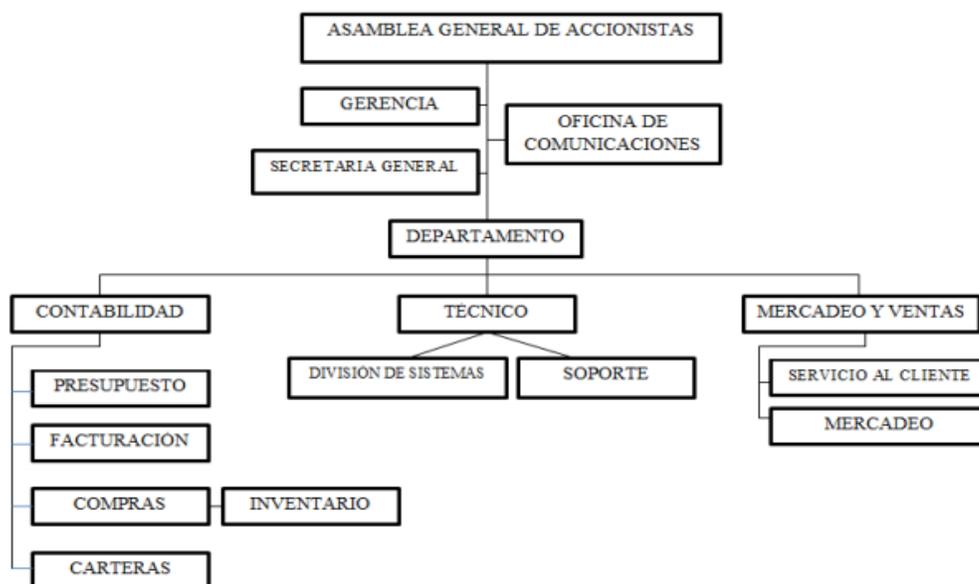


Figura 1. Estructura organizacional (TechNet Comunicaciones S.A.S)

1.1.5 Dependencia y/o proyecto asignado.

En el proyecto asignado se desarrollará un plugin de mensajería que facilite la comunicación entre el CRM (Gestión de las relaciones con clientes) y el software UMNS (Sistema de gestión de red de Ubiquiti) utilizado por TechNet Comunicaciones S.A.S, con el fin de que el cliente conozca a través del servicio de mensajes cortos (SMS) información acerca de su factura, así como las fallas del servicio ofrecido por la empresa.

Inicialmente se identificaron las fortalezas, debilidades, oportunidades y amenazas, las cuales se analizaron empleando la Matriz DOFA y las estrategias que se derivan de esta.

1.2 Diagnóstico inicial de la dependencia asignada.

Tabla 1
Matriz DOFA

	Fortalezas	Debilidades
Diagnóstico inicial	1. TechNet Comunicaciones S.A.S con el fin de ofrecer servicios de calidad y excelente atención al cliente, busca implementar herramientas que mejoren su estado actual. 2. TechNet Comunicaciones S.A.S cuenta con infraestructura tecnológica robusta para la realización del Plugin de mensajería.	1. Información limitada en el desarrollo y creación de plugins en la organización. 2. La caída y las fallas del servicio de internet ofrecido son provocadas por la ubicación geográfica y los cambios climáticos.
Oportunidades	Estrategia FO	Estrategia DO

Continuación

<p>1. Los clientes de otros proveedores de servicios de internet están poco satisfechos con el servicio debido a su mala calidad y desinformación del mismo.</p> <p>2. La organización conoce las necesidades del mercado y por ende de sus clientes, dando lugar al desarrollo de nuevas estrategias (Plugin de mensajería).</p>	<p>1. Estudiar las necesidades actuales de los clientes.</p> <p>2. Gestionar los mensajes enviados al cliente de forma ágil y óptima mediante la implementación del Plugin.</p>	<p>1. Investigar acerca de las herramientas a incorporar en la organización, con el fin de ampliar los conocimientos de los mismos.</p> <p>2. Evaluación de las necesidades del mercado y de sus clientes.</p>
Amenazas	Estrategia FA	Estrategia DA
<p>1. Competencia por parte de otros proveedores de servicio de internet abarcando un público extenso.</p> <p>2. La idea de la creación del Plugin de mensajería puede ser imitada por la competencia.</p>	<p>1. Programar capacitaciones para el personal sobre el Plugin de mensajería.</p> <p>2. Despliegue de campaña publicitaria para dar a conocer a la comunidad el Plugin de mensajería de la organización.</p>	<p>1. Descontinuar el uso del Plugin de mensajería en el software de TechNet Comunicaciones S.A.S.</p>

1.2.1 Planteamiento del problema.

En la actualidad, las organizaciones evolucionan en un ambiente de cambios y retos constantes, es por esto la importancia de analizar los diferentes factores que se manifiestan alrededor de ellas, por esta razón empresas como TechNet Comunicaciones S.A.S buscan crear nuevas estrategias enfocadas en mejorar la calidad de sus servicios.

TechNet Comunicaciones S.A.S cuenta con software avanzado como es el caso de UNMS (Sistema de gestión de red de Ubiquiti) el cual permite detectar las fallas y enviar alertas

tempranas a los funcionarios o clientes vía correo electrónico, sistema que se hace poco práctico debido a que no todos los clientes usan correo o carecen de plan de datos para revisarlo, es por esto que se plantea hacerlo vía SMS, actualmente no se posee un Plugin de mensajería, que mantenga informada al área técnica y de ingeniería sobre cualquier falla en el sistema y del mismo modo informar a sus clientes del estado de sus facturas e información de soporte técnico.

Considerando lo anteriormente expuesto y debido al creciente número de usuarios surge la necesidad de desarrollar un plugin de mensajería (SMS), para que los clientes además de recibir información acerca de su factura puedan estar al tanto del soporte técnico de una forma más automatizada.

1.3 Objetivos de la pasantía

1.3.1 General.

Desarrollar un Plugin SMS para el Software CRM de la empresa TechNet Comunicaciones S.A.S de la ciudad de Ocaña, Norte de Santander.

1.3.2 Específicos.

- Realizar una investigación sobre la creación de plugins y su incorporación en el software de TechNet Comunicaciones S.A.S.
- Desarrollar el Plugin de mensajería para el software CRM (Gestión de relaciones con los clientes) de la organización.
- Realizar las pruebas al Plugin en la infraestructura tecnológica de la empresa.

- Implementar el Plugin de mensajería para el software CRM (Gestión de relaciones con los clientes) de TechNet Comunicaciones S.A.S.

1.4 Actividades a desarrollar

Tabla 2
Actividades a desarrollar

Objetivo General	Objetivos Específicos	Actividades a desarrollar en la empresa para hacer posible el cumplimiento de los Obj. Específicos
Desarrollar un Plugin SMS para el Software CRM de la empresa TechNet Comunicaciones S.A.S de la ciudad de Ocaña, Norte de Santander.	Realizar una investigación sobre la creación de plugins y su implementación en el software de TechNet Comunicaciones S.A.S.	<ol style="list-style-type: none"> 1. Recopilar la información necesaria para la investigación sobre la creación de plugins y su implementación. 2. Documentar la información recolectada anteriormente.
	Desarrollar el Plugin de mensajería para el software CRM (Gestión de relaciones con los clientes) de la organización.	<ol style="list-style-type: none"> 1. Realizar el Plugin de mensajería encargado de notificar a los clientes el pago de sus facturas. 2. Realizar el plugin de mensajería que notifica información de soporte técnico a los clientes.
	Realizar las pruebas al Plugin en la infraestructura tecnológica de la empresa.	<ol style="list-style-type: none"> 1. Realizar las pruebas necesarias al Plugin de mensajería en un sistema aislado. 2. Corregir los errores identificados con la aplicación de la prueba.
	Implementar el Plugin de mensajería para el software CRM (Gestión de relaciones con los clientes) de TechNet Comunicaciones S.A.S.	<ol style="list-style-type: none"> 1. Poner en funcionamiento el Plugin de mensajería al software de la organización. 2. Llevar a cabo capacitaciones para el personal de la organización sobre el Plugin de mensajería.

2. Enfoques referenciales

2.1 Enfoque conceptual

Dentro del proyecto es importante entender conceptos como:

- **Plugin**

Se define plugin como:

Un plugin es una pieza de software que se relaciona con una aplicación (llamada aplicación anfitrión o host) para extender las funcionalidades de la última. El plugin de por sí carece de la capacidad de ejecutar la funcionalidad que implementa, es por esto que depende del anfitrión para ejecutarse. (Bustio, Coma, & Talavera, 2013)

- **Ubiquiti**

“Ubiquiti Networks es una compañía de tecnologías de comunicación de próxima generación fundada en 2005. Diseña y manufactura plataformas de tecnología disruptiva para mercados emergentes que operan con modelos de negocios productivos y conectividad omnipresente”. (WNI México, s.f.)

- **UNMS**

Se define UNMS como:

“UNMS (Ubiquiti Network Management System) de Ubiquiti es un software de monitorización basado en web disponible de forma gratuita para usuarios y proveedores de servicios”. (WifiSafe-Distribuidor WiFi, s.f.)

“Este sistema de gestión de red Ubiquiti (UNMS) le permite configurar, actualizar y monitorear varias plataformas de dispositivos Ubiquiti diferentes: airMAX, airCube, airFiber, EdgeMAX y UFiber”. (Ubiquiti, s.f.)

- **ISP**

Se define ISP como:

“Los ISP son empresas encargadas de prestar el servicio de acceso a la Internet”.

(Álvarez, Padilla, Garzón, & Muñoz, 2009)

- **Infraestructura tecnológica**

Se define infraestructura tecnológica como:

“Es el conjunto de hardware y software sobre el que se asientan los diferentes servicios que la institución necesita tener en funcionamiento para poder llevar a cabo toda su actividad”.

(EducareCorp, s.f.)

2.2 Enfoque legal

El marco legal para el desarrollo del plugin:

Derechos de autor:

Los derechos de autor en Colombia están protegidos por la ley 23 de 1982, la cual en el artículo 2 establece:

“Los derechos de autor recaen sobre las obras científicas, literarias y artísticas las cuales comprenden todas las creaciones del espíritu en el campo científico, literario y artístico, cualquiera que sea el modo o forma de expresión y cualquiera que sea su destinación...” (García, 2018).

Por medio de la ley 170 de 1994 se aprueba el Acuerdo por el que se establece la “Organización Mundial del Comercio (OMC)”, suscrito en Marrakech (Marruecos) el 15 de abril de 1994; a través del artículo 10:1 esta plantea:

“Los programas de ordenador, sean programas fuente o programas objeto serán protegidos como obras literarias en virtud del convenio de Berna (1971)”. (Función Pública Gobierno de Colombia)

Así mismo la Comunidad Andina de Naciones en la Decisión 351 respalda los derechos de autor de software, por medio del capítulo VIII “De los programas de ordenador y bases de datos”, artículo 23 establece:

Los programas de ordenador se protegen en los mismos términos que las obras literarias. Dicha protección se extiende tanto a los programas operativos como a los programas aplicativos, ya sea en forma de código fuente o código objeto... Sin perjuicio de ello, los autores o titulares de los programas de ordenador podrán autorizar las modificaciones necesarias para la correcta utilización de los programas. (Dirección Nacional de Derecho de Autor)

Por otro lado, la ley 565 de 2000 por medio de la cual se aprueba el “Tratado de la OMPI- Organización Mundial de la Propiedad Intelectual- sobre Derechos de Autor (WCT), adoptado en Ginebra el 20 de diciembre de 1996. A través del artículo 4 “Programas de ordenador” plantea lo siguiente:

“Los programas de ordenador están protegidos como obras literarias en el marco de los dispuesto en el artículo 2 del Convenio de Berna. Dicha protección se aplica a los programas de ordenador, cualquiera que sea su modo o forma de expresión”. (Suin - Juriscol)

3. Informe de cumplimiento de trabajo

3.1 Presentación de resultados

3.1.1 Objetivo 1. Realizar una investigación sobre la creación de plugins y su implementación en el software de TechNet Comunicaciones S.A.S.

- Actividad 1. Recopilar la información necesaria para la investigación sobre la creación de plugins y su implementación.
- Actividad 2. Documentar la información recolectada anteriormente.

Para hacer posible el cumplimiento del primer objetivo, fue necesario profundizar en la creación de plugins, conocer el funcionamiento del software de la organización y preparar el entorno de desarrollo.

TechNet Comunicaciones S.A.S cuenta con un sistema de gestión de red de Ubiquiti (UNMS), es un software de monitorización basado en web disponible para proveedores de servicio de internet (ISP) y usuarios, que permite monitorear, configurar, actualizar y administrar una red, permitiendo administrar todo en un solo lugar, UNMS es un software libre que está conformado por dos módulos:

- Red: Se encarga de la gestión de todos los dispositivos
- CRM (Gestión de relaciones con clientes): Permite la gestión de clientes y facturación.

Características:

- Instalación simple
- Descubrimiento rápido de dispositivos
- Monitoreo de red
- Administración de dispositivos
- Configuración del dispositivo
- Firmware y administrador de tareas (Ubiquiti, s.f.)

Para la instalación local de UNMS es necesario contar con Ubuntu Server en un servidor dedicado o en una máquina virtual, para el desarrollo del plan de trabajo TechNet Comunicaciones S.A.S ha facilitado un servidor con 2 GB de RAM, 297 GB de almacenamiento y un procesador Intel Celeron.

1. Instalación Ubuntu Server

Procedemos a realizar la instalación, en primera instancia se debe elegir el lenguaje del sistema.

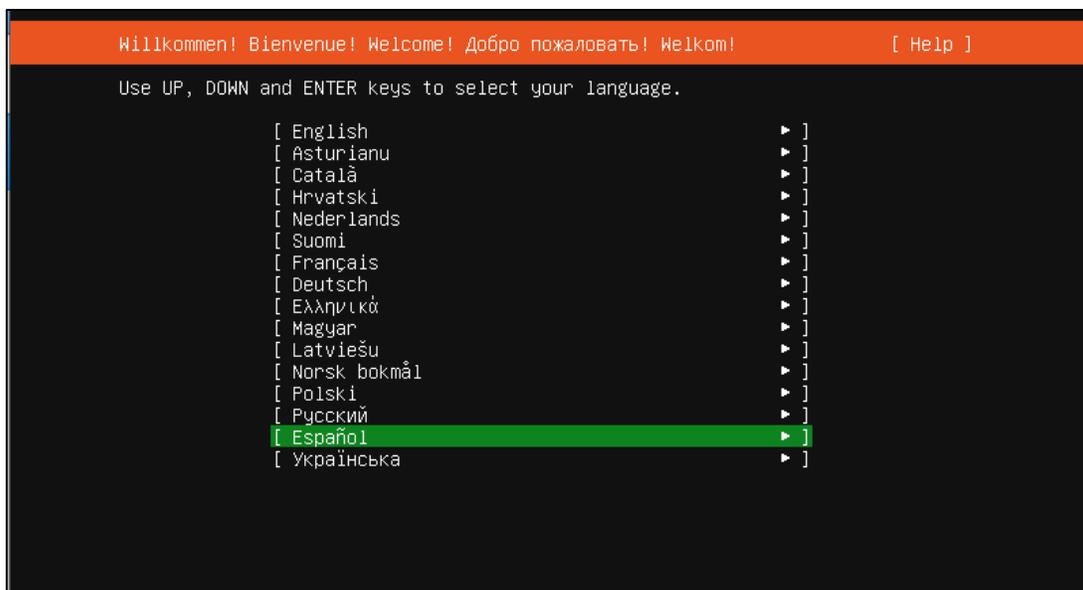


Figura 2. Lenguaje del Sistema.

Seguidamente se realiza la configuración e identificación del teclado.

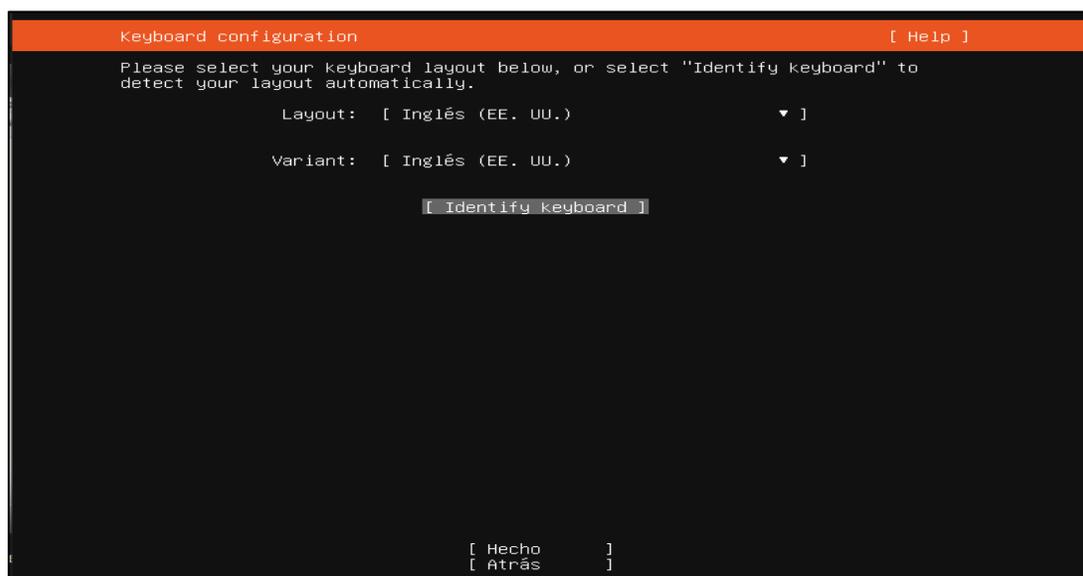


Figura 3. Configuración e identificación de teclado

En la identificación del teclado se efectúa una auto-detección, en esta se realizan una serie de preguntas acerca del teclado del dispositivo.

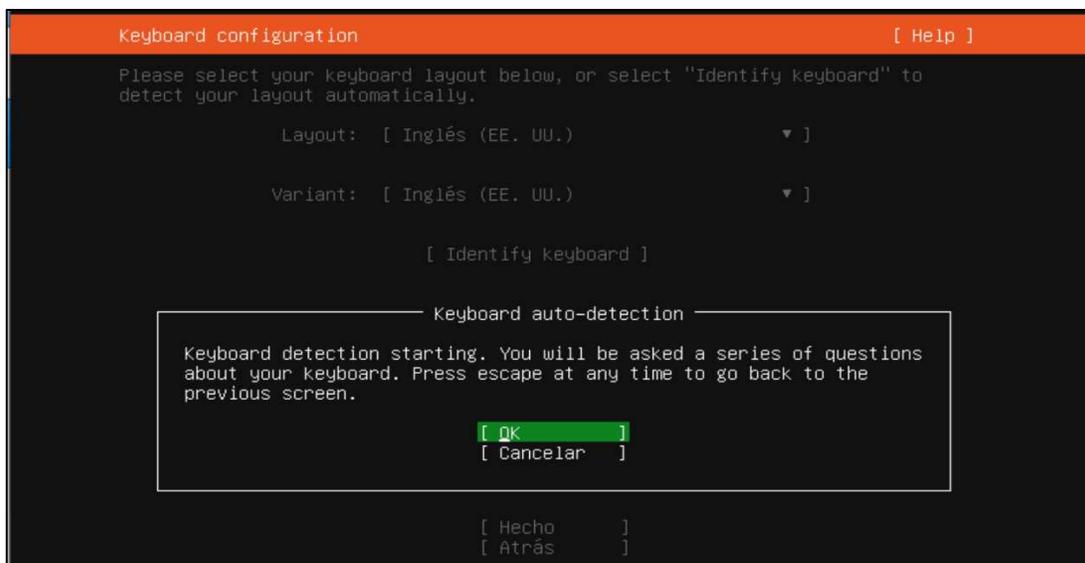


Figura 4. Autodetección del teclado

Para continuar con la auto-detección del teclado se debe seleccionar uno de los caracteres mostrados en pantalla y que estén presentes en el teclado del equipo.

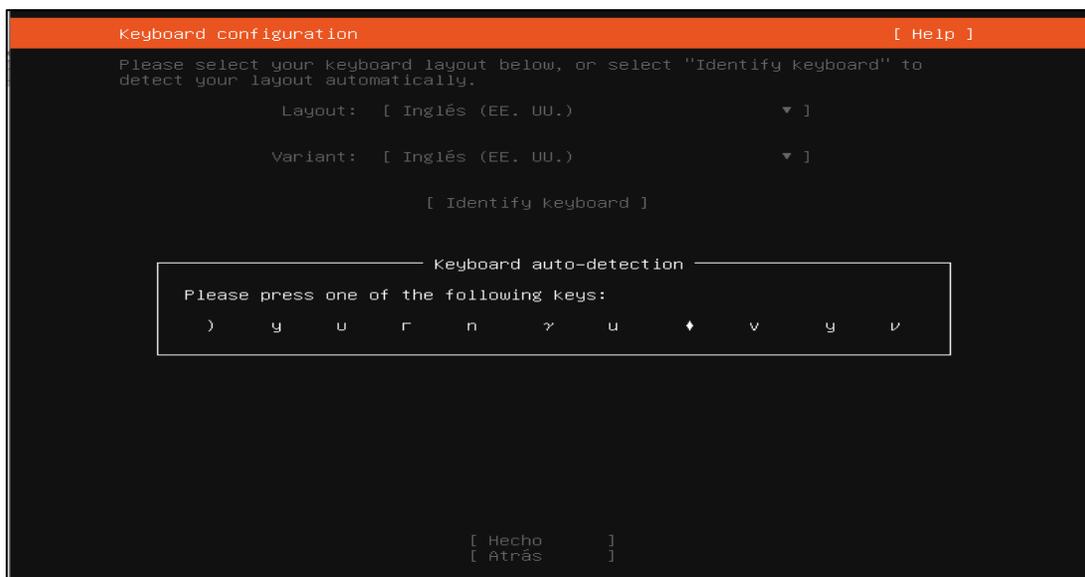


Figura 5. Autodetección del teclado

Al completar la auto-detección, el teclado recomendado es “Español (latinoamericano)”.

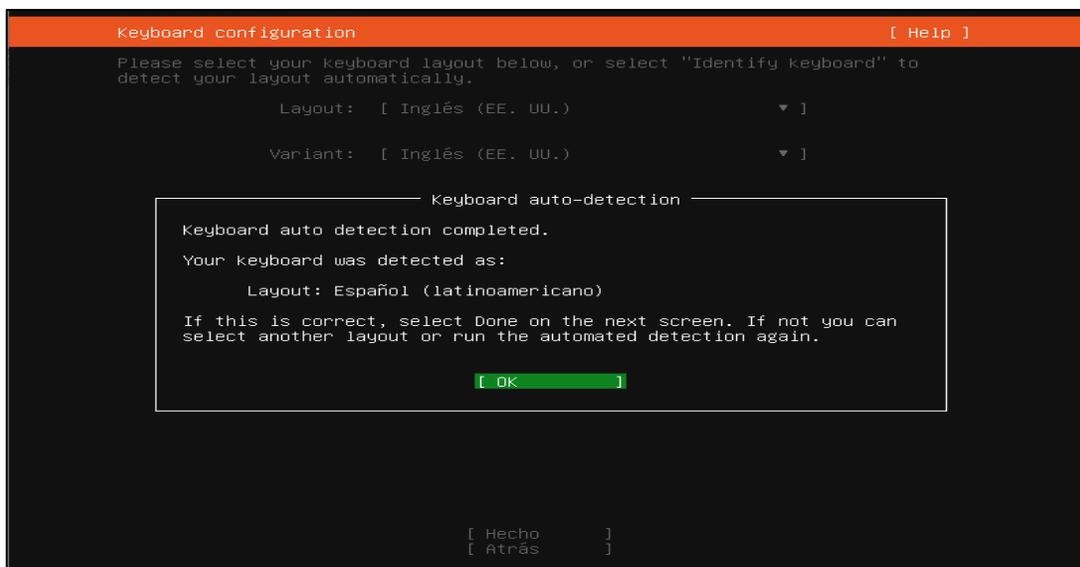


Figura 6. Teclado recomendado "Español (latinoamericano)"

En el área de conexiones de red debe configurarse al menos una interfaz para permitir la comunicación de este servidor con otros dispositivos, por otro parte, se muestra la dirección asignada por DHCPv4 (Protocolo de Configuración Dinámica de Hosts para IPv4).

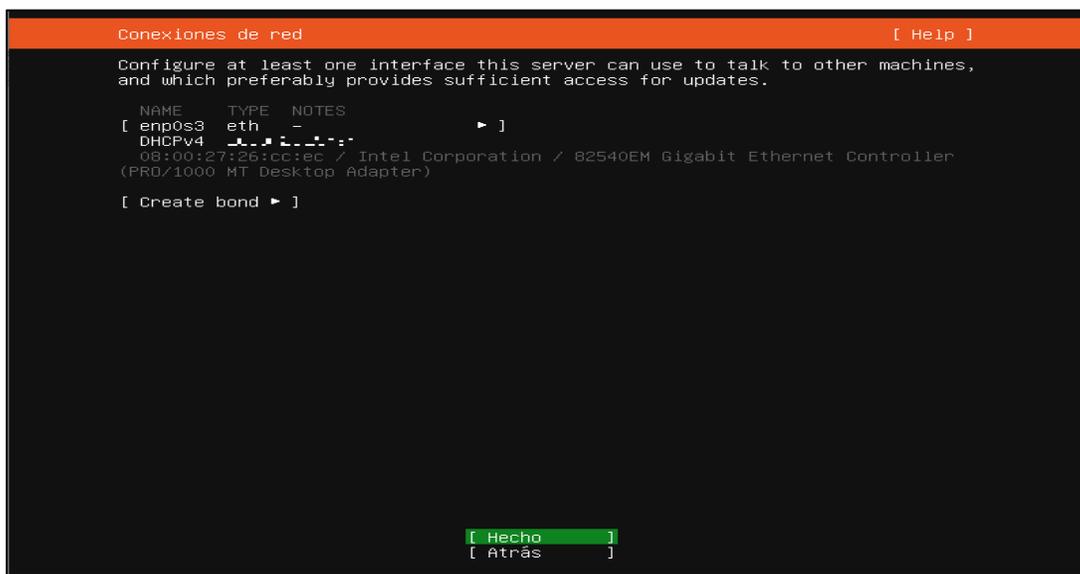


Figura 7. Conexiones de red

A continuación se configura el sistema de archivos, eligiendo la partición o el disco entero.

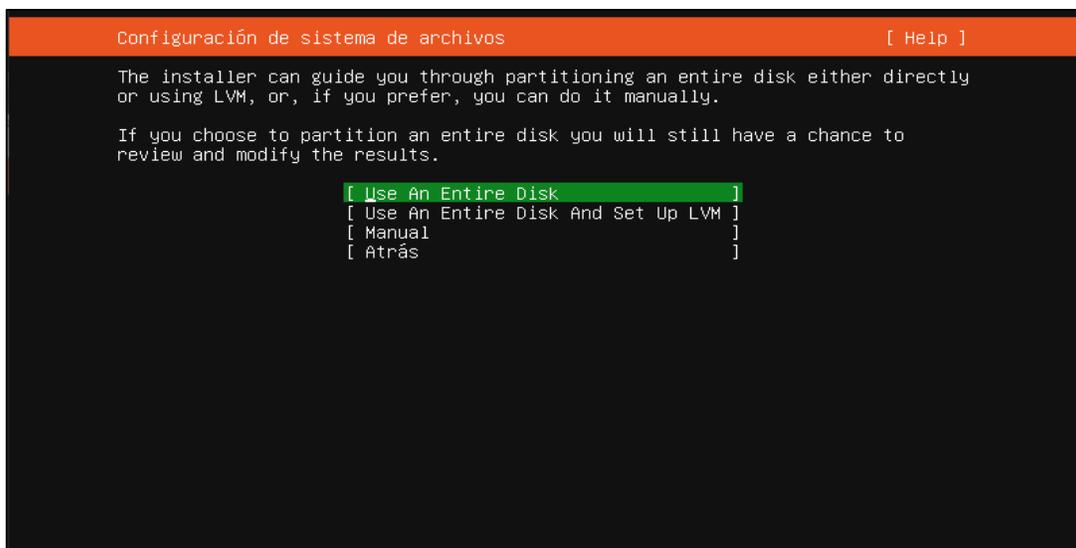


Figura 8. Configuración de sistema de archivos

Debe seleccionarse la partición de disco correcta, en este caso es un disco en blanco dedicado solamente para este sistema. Seguidamente, se pide la confirmación de la selección de partición correcta. Luego, se asigna un nombre, nombre del servidor, nombre de usuario (con este usuario se logeará en el sistema) y contraseña.

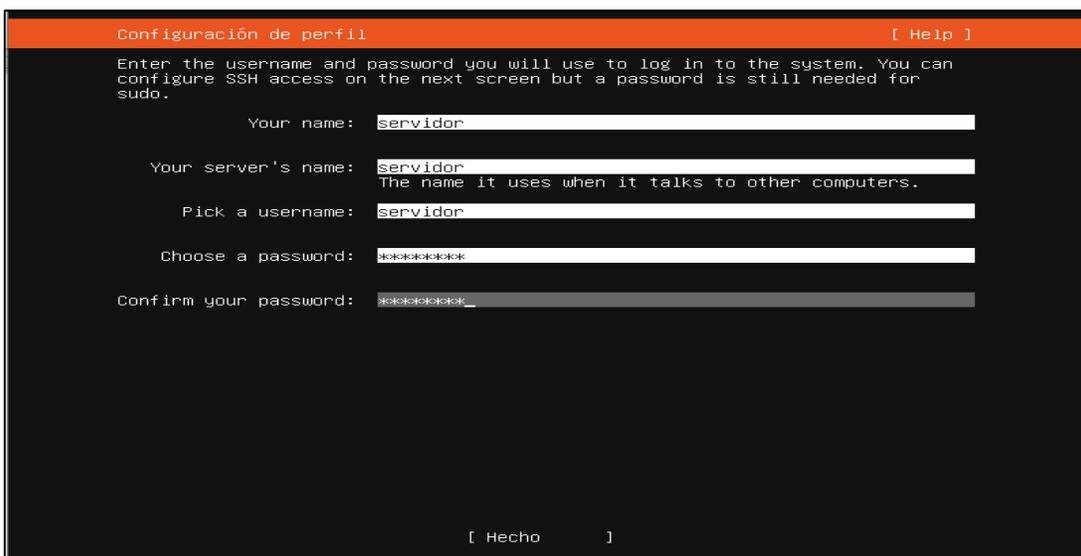


Figura 9. Configuración del perfil

Se da inicio a la instalación de Ubuntu Server.

```

Instalando el sistema [ Help ]

curtin command install
preparing for installation
configuring storage
  running 'curtin block-meta simple'
  curtin command block-meta
  removing previous storage devices
  configuring disk: disk-sda
  configuring partition: partition-0
  configuring partition: partition-1
  configuring format: format-0
  configuring mount: mount-0
configuring network
  running 'curtin net-meta auto'
  curtin command net-meta
writing install sources to disk
  running 'curtin extract'
  curtin command extract
  acquiring and extracting image from cp:///media/filesystem
configuring installed system
  running '/snap/bin/subiquity.subiquity-configure-run'
  running '/snap/bin/subiquity.subiquity-configure-apt'
/snap/subiquity/1459/usr/bin/python3 true'
curtin command apt-config
curtin command in-target \

[ View full log ]

```

Figura 10. Instalación del sistema

Para ingresar es necesario introducir las credenciales, anteriormente configuradas.

```

servidor login: servidor
Password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

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

System information disabled due to load higher than 1.0

14 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

servidor@servidor:~$

```

Figura 11. Logueo exitoso

2. Configuración de IP estática

Antes de empezar la configuración de la IP, se actualizan o se instalan las últimas versiones de todos los paquetes instalados, con el comando `apt-get update`.

```

servidor@servidor:~$ sudo su
root@servidor:/home/servidor# apt-get update
Obj:1 http://co.archive.ubuntu.com/ubuntu bionic InRelease
Obj:2 http://co.archive.ubuntu.com/ubuntu bionic-updates InRelease
Obj:3 http://co.archive.ubuntu.com/ubuntu bionic-backports InRelease
Obj:4 http://co.archive.ubuntu.com/ubuntu bionic-security InRelease
Des:5 http://co.archive.ubuntu.com/ubuntu bionic/main Translation-es [364 kB]
Des:6 http://co.archive.ubuntu.com/ubuntu bionic/restricted Translation-es [1.960 B]
Des:7 http://co.archive.ubuntu.com/ubuntu bionic/universe Translation-es [1.259 kB]
Des:8 http://co.archive.ubuntu.com/ubuntu bionic/multiverse Translation-es [74,9 kB]
Descargados 1.700 kB en 3s (524 kB/s)
Leyendo lista de paquetes... Hecho
root@servidor:/home/servidor#

```

Figura 12. Actualización de paquetes instalados

Para configurar la ip estática en Ubuntu Server, se accede a la ruta `/etc/netplan`, y así configurar con `netplan` el fichero `.yaml`.

```

Ubuntu 18.04.4 LTS servidor tty1
servidor login: servidor
Password:
Last login: Tue Mar 10 15:29:04 UTC 2020 on tty1
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

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

System information as of Tue Mar 10 15:30:16 UTC 2020

System load:  0.06          Processes:    88
Usage of /:   1.4% of 287.31GB  Users logged in:  0
Memory usage: 7%           IP address for enp0s3: 10.0.2.15
Swap usage:  0%

14 packages can be updated.
0 updates are security updates.

servidor@servidor:~$ sudo su
[sudo] password for servidor:
root@servidor:/home/servidor# cd /etc/netplan
root@servidor:/etc/netplan# ls -l
total 4
-rw-r--r-- 1 root root 391 mar 10 15:09 50-cloud-init.yaml
root@servidor:/etc/netplan# sudo nano 50-cloud-init.yaml_

```

Figura 13. Configuración de IP estática

A continuación se muestra el archivo `.yaml`

```

GNU nano 2.9.3                    50-cloud-init.yaml

# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: [config: disabled]
network:
  ethernets:
    enp0s3:
      dhcp4: true
      version: 2

[ Read 10 lines ]
G Get Help      O Write Out    W Where Is    K Cut Text    J Justify     C Cur Pos     M-U Undo
X Exit         R Read File    R Replace    U Uncut Text  T To Spell   G Go To Line  M-E Redo

```

Figura 14. Configuración IP estática

Dentro del fichero 50-cloud-init.yaml se asigna en la interfaz enp0s3 la dirección IP junto con la puerta de enlace, el DNS (Sistema de nombres de dominio) y se deniega el protocolo de Configuración Dinámica de Hosts para IPv4 e IPv6.

```

GNU nano 2.9.3                    50-cloud-init.yaml Modified

# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: [config: disabled]
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      dhcp6: no
      addresses: 192.168.1.10/24, 192.168.1.11/24
      gateway4: 192.168.1.1
      nameservers:
        addresses: 8.8.8.8, 8.8.4.4

```

Figura 15. Configuración IP estática

Se aplican los cambios con el comando “netplan apply” y para comprobar la IP estática se utiliza el comando “ifconfig -a” y efectivamente en la interfaz enp0s3 se encuentra asignada la dirección IP estática.

```

Password:
Last login: Tue Mar 10 15:41:27 UTC 2020 on tty1
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

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

System information disabled due to load higher than 1.0

14 packages can be updated.
0 updates are security updates.

servidor@servidor:~$ sudo su
[sudo] password for servidor:
root@servidor:/home/servidor# ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fedd:9e31 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:dd:9e:31 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 27 bytes 1806 (1.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 191 bytes 14575 (14.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 191 bytes 14575 (14.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@servidor:/home/servidor# _

```

Figura 16. Dirección IP estática asignada

3. Instalación de UNMS

Antes de iniciar la instalación de UNMS se debe verificar que las herramientas necesarias estén instaladas en el sistema con los siguientes comandos:

```
sudo apt-get update
```

```

servidor@servidor:~$ sudo apt-get update
[sudo] password for servidor:
Obj:1 http://co.archive.ubuntu.com/ubuntu bionic InRelease
Des:2 http://co.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]
Des:3 http://co.archive.ubuntu.com/ubuntu bionic-backports InRelease [74,6 kB]
Des:4 http://co.archive.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]
Descargados 252 kB en 4s (64,7 kB/s)
Leyendo lista de paquetes... Hecho
servidor@servidor:~$ _

```

Figura 17. Comando "sudo apt-get update"

sudo apt-get install curl sudo bash

```

servidor@servidor:~$ sudo apt-get install curl sudo bash
[sudo] password for servidor:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
bash ya está en su versión más reciente (4.4.18-2ubuntu1.2).
curl ya está en su versión más reciente (7.58.0-2ubuntu3.8).
fijado curl como instalado manualmente.
sudo ya está en su versión más reciente (1.8.21p2-3ubuntu1.2).
fijado sudo como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 14 no actualizados.
servidor@servidor:~$ _

```

Figura 18. Comando "sudo apt-get install curl sudo bash"

Para comenzar la instalación del sistema de gestión de red de Ubiquiti se ejecuta el siguiente comando en el host.

```
curl -fsSL https://unms.com/v1/install> /tmp/unms_inst.sh && sudo bash
```

/tmp/unms_inst.sh

```

conntrack v1.4.4 (conntrack-tools): 0 flow entries have been deleted.
Starting docker containers.
Starting UNMS...
conntrack v1.4.4 (conntrack-tools): 0 flow entries have been deleted.
Creating network "unms_internal" with the default driver
Creating network "unms_public" with the default driver
Creating unms-fluentd
Creating unms-postgres
Creating unms-nginx
Creating unms-redis
Creating unms-rabbitmq
Creating unms-netflow
Creating ucrm
Creating unms
UNMS started.
Removing old images
Waiting for UNMS to start

```

CONTAINER ID	PORTS	IMAGE	COMMAND	CREATED	STATUS
b748c2931ff2		ubnt/unms:1.1.5	"/usr/bin/dumb-init ..."	26 seconds ago	Up 12 sec
bnds			unms		
f47f05ac8a81		ubnt/unms-crm:3.1.5	"dumb-init -- make s..."	37 seconds ago	Up 25 sec
bnds	80-81/tcp, 443/tcp, 9000/tcp, 2055/udp		ucrm		
06e180e63f19		ubnt/unms-netflow:1.1.5	"/usr/bin/dumb-init ..."	37 seconds ago	Up 25 sec
bnds	0.0.0.0:2055->2055/udp		unms-netflow		
b9c9bdce90e4		redis:5.0.5-alpine	"docker-entrypoint.s..."	47 seconds ago	Up 37 sec
bnds			unms-redis		
b180ef9db7b1		rabbitmq:3.7.14-alpine	"docker-entrypoint.s..."	47 seconds ago	Up 41 sec
bnds			unms-rabbitmq		
c2319b670d79		ubnt/unms-nginx:1.1.5	"/entrypoint.sh ngin..."	47 seconds ago	Up 37 sec
bnds	0.0.0.0:80-81->80-81/tcp, 0.0.0.0:443->443/tcp		unms-nginx		
16cb5f183c05		postgres:9.6.12-alpine	"docker-entrypoint.s..."	47 seconds ago	Up 40 sec
bnds			unms-postgres		
b2ca70ec30fd		ubnt/unms-fluentd:1.1.5	"/entrypoint.sh /bin..."	50 seconds ago	Up 47 sec
bnds	5140/tcp, 1:1:1:1:24224->24224/tcp		unms-fluentd		

```

UNMS is running
servidor@servidor:~$ _

```

Figura 19. Instalación de UNMS

A continuación, se ingresa a UNMS con la dirección IP asignada, para iniciar en el sistema. Seguidamente, se realiza la creación de la cuenta con la que se va acceder a UNMS.

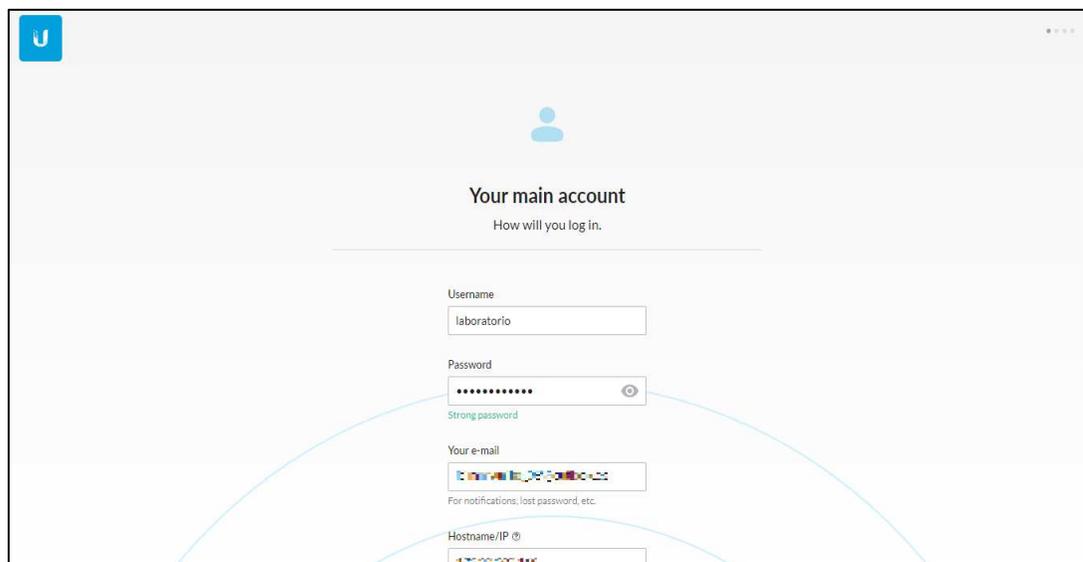
The screenshot shows a web interface for account creation. At the top left is a blue square with a white 'U'. Below it is a blue person icon. The heading is "Your main account" with the subtext "How will you log in." Below this are four input fields: "Username" with the value "laboratorio", "Password" with masked characters and a "Strong password" indicator, "Your e-mail" with a colorful email address, and "Hostname/IP" with a colorful IP address. A blue arc highlights the password and email fields.

Figura 20. Creación de la cuenta en UNMS

En este punto es de gran importancia guardar la “Vault key” generada por el sistema, esta llave es la que se entrega a los productos para que busquen directamente al servidor.

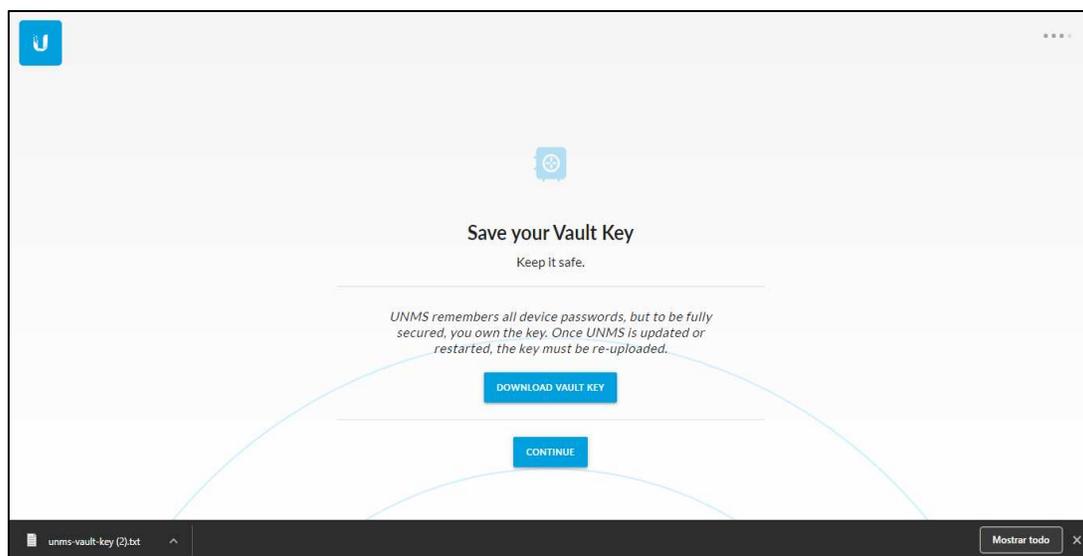
The screenshot shows a web interface for downloading a Vault Key. At the top left is a blue square with a white 'U'. Below it is a blue gear icon. The heading is "Save your Vault Key" with the subtext "Keep it safe." Below this is a paragraph of text: "UNMS remembers all device passwords, but to be fully secured, you own the key. Once UNMS is updated or restarted, the key must be re-uploaded." Below the text are two blue buttons: "DOWNLOAD VAULT KEY" and "CONTINUE". A blue arc highlights the "DOWNLOAD VAULT KEY" button. At the bottom left, there is a file name "unms-vault-key (2).txt" and at the bottom right, a button "Mostrar todo" with a close icon.

Figura 21. Vault key

Luego, se debe introducir el rango de IPs donde UNMS buscará los dispositivos.

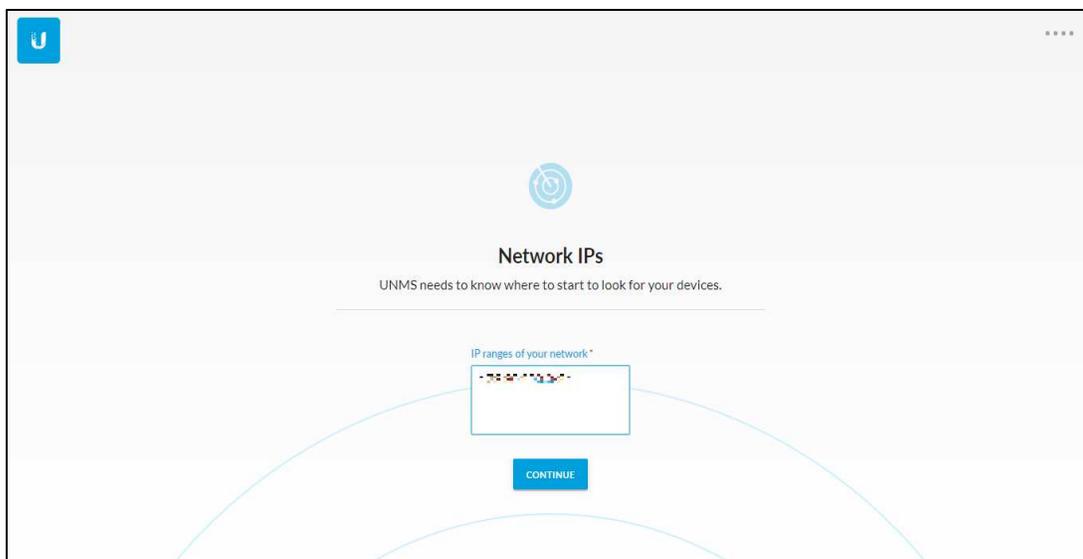


Figura 22. Rango de IPs

Al ingresar se muestran las novedades de UNMS, y la interfaz de inicio del sistema de gestión de red de Ubiquiti.

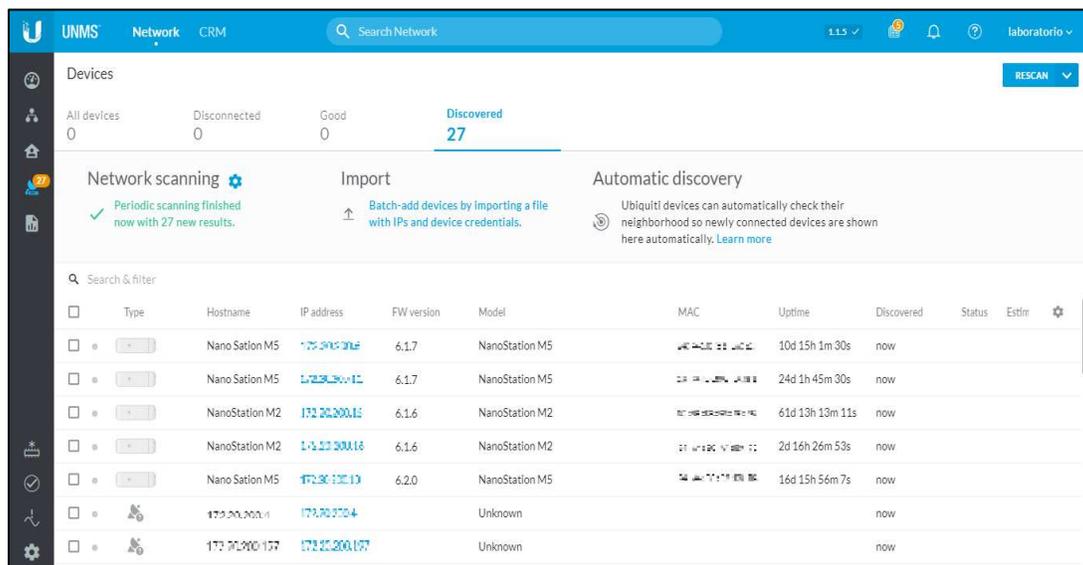


Figura 23. Interfaz de inicio de UNMS

- **Network (Módulo de red)**

El módulo de red de UNMS cuenta con un menú que se encuentra ubicado en la parte izquierda del sistema, en primera instancia se encuentra la opción “dashboard” o tablero, esta muestra el estado de la red (Sitios, sitios cliente, dispositivos descubiertos, desconectados, etc), así como las notificaciones de las actividades realizadas dentro del sistema.

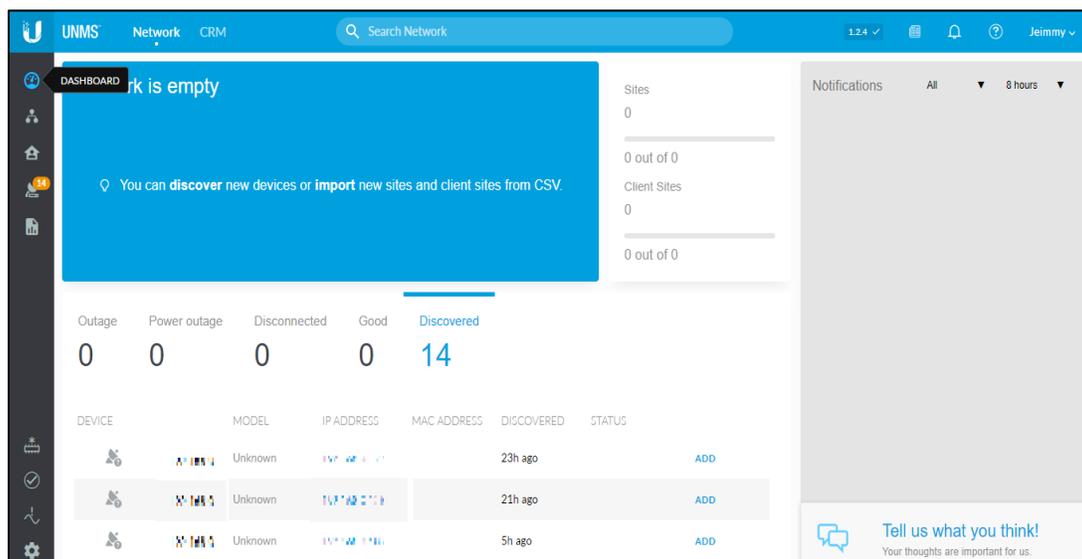


Figura 24. Tablero (Network)

A continuación, se encuentra el apartado de “sites” o sitios, en este se encuentran los sitios creados, su localización en el mapa, así como la topología de red de cada sitio en concreto y se permite la creación de nuevos sitios.

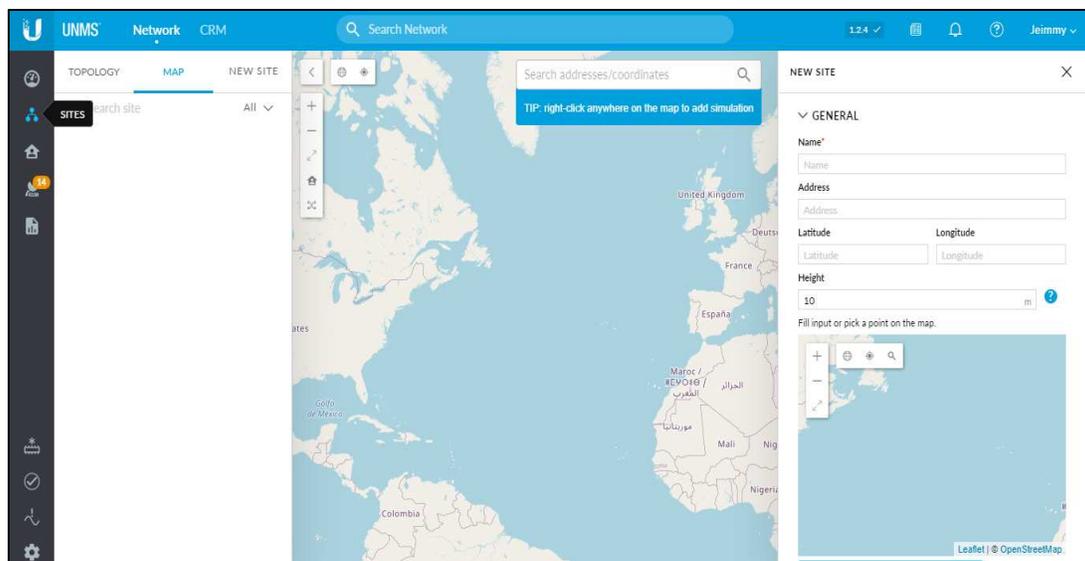


Figura 25. Sitios (Módulo de red)

El sistema de gestión de red de Ubiquiti da la opción de “Client sites” para la creación de sitios clientes, cabe resaltar que cada uno de sus dispositivos debe asignarse a un sitio o un sitio de Cliente.

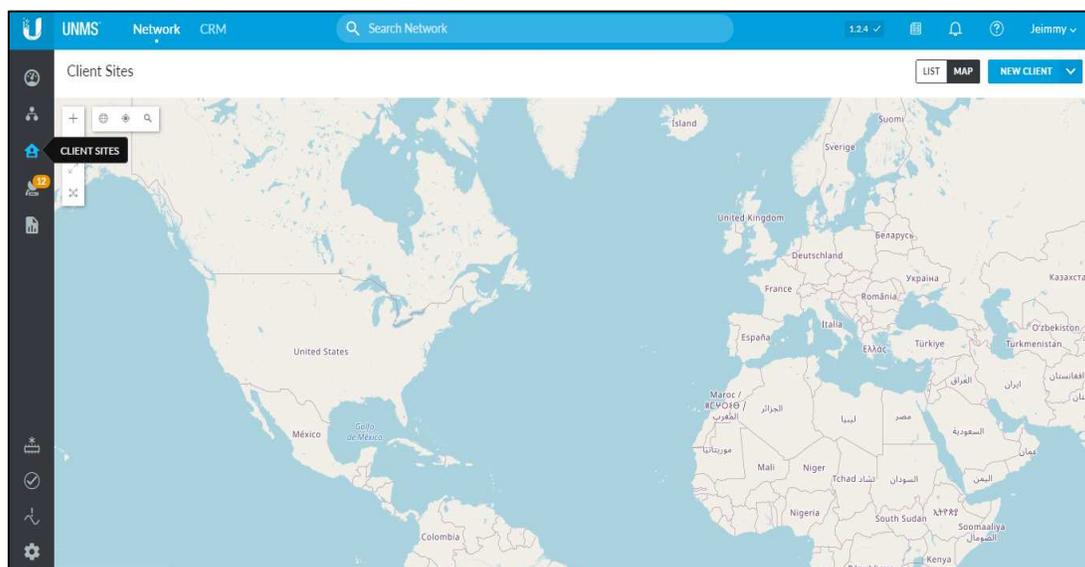


Figura 26. Sitios clientes (Módulo de red)

Otra opción en este módulo de UNMS es “Devices” o dispositivos, en esta sección se muestra el estado de los dispositivos (buenos, conectados y descubiertos) dentro de la misma red a través de un escaneo realizado por el sistema, así como los que han sido o serán importados de archivos CSV (Valores separados por comas).

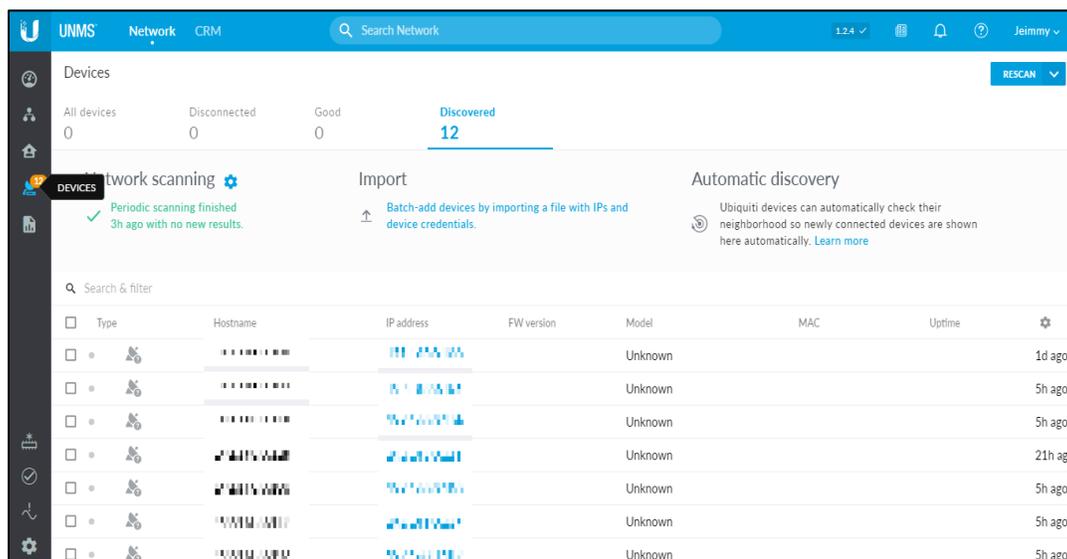


Figura 27. Dispositivos (Módulo de red)

Otra opción disponible en UNMS es la de “Reports” o informes de IPs desconocidas, enlaces de datos y potencial de enlaces.

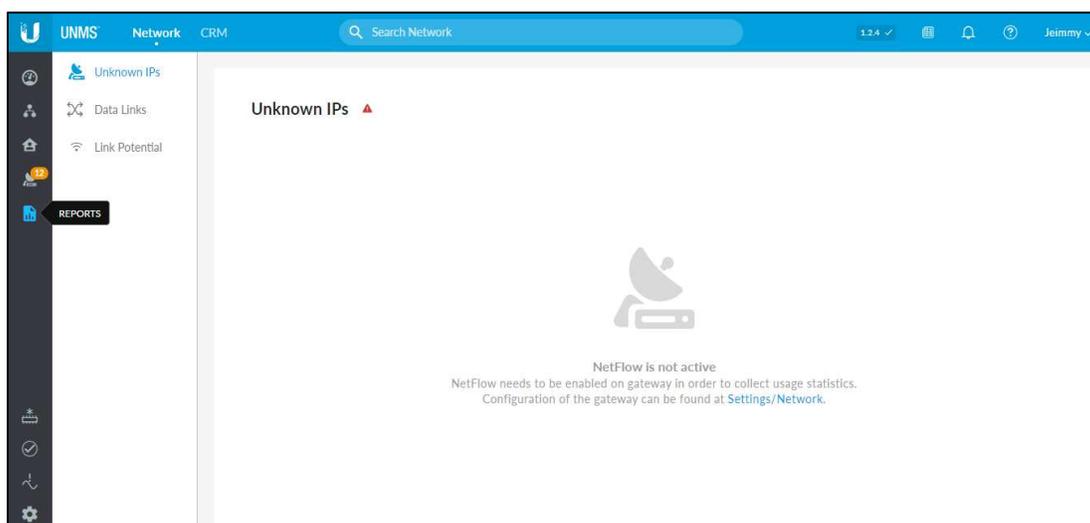
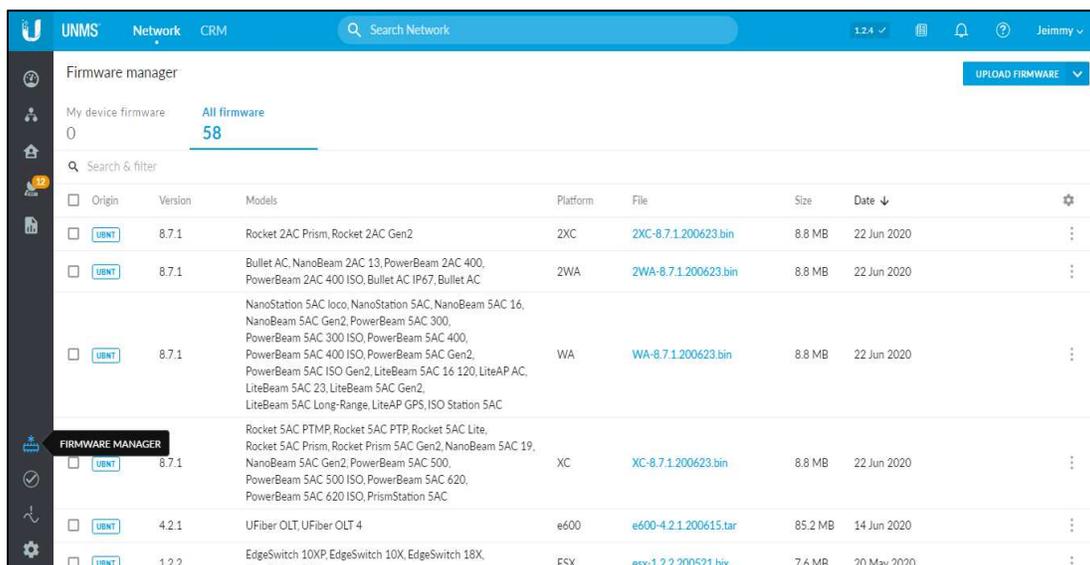


Figura 28. Reportes (Módulo de red)

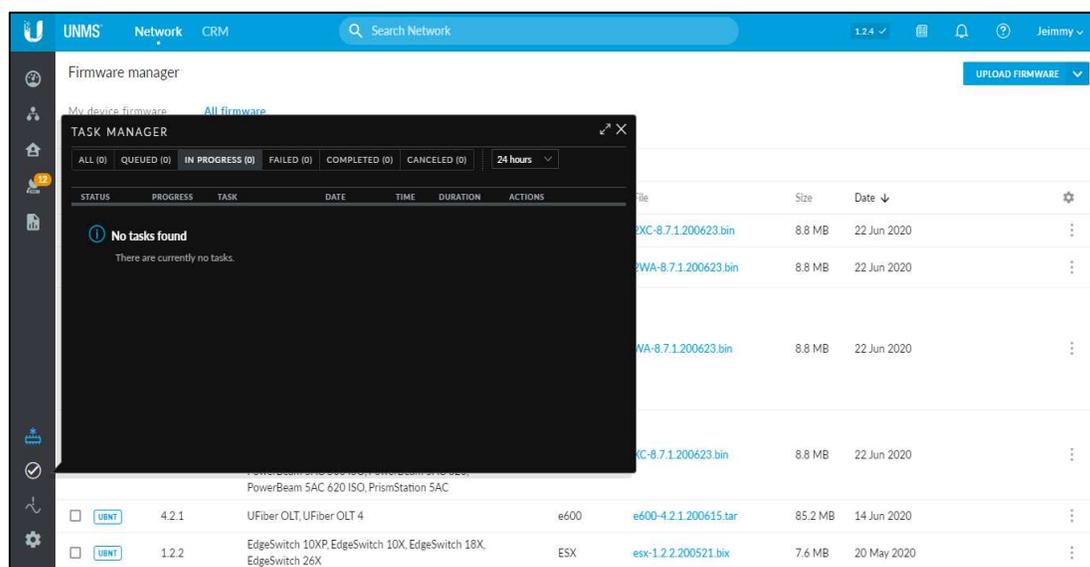
En este módulo también se tiene la opción de “firmware manager” o administrador de firmware, que permitirá actualizar los equipos de forma automática y centralizada. Así mismo, mostrará la lista de la última versión disponible para todos los dispositivos compatibles.



Origin	Version	Models	Platform	File	Size	Date
UBNT	8.7.1	Rocket 2AC Prism, Rocket 2AC Gen2	2XC	2XC-8.7.1.200623.bin	8.8 MB	22 Jun 2020
UBNT	8.7.1	Bullet AC, NanoBeam 2AC 13, PowerBeam 2AC 400, PowerBeam 2AC 400 ISO, Bullet AC IP67, Bullet AC	2WA	2WA-8.7.1.200623.bin	8.8 MB	22 Jun 2020
UBNT	8.7.1	NanoStation 5AC loco, NanoStation 5AC, NanoBeam 5AC 16, NanoBeam 5AC Gen2, PowerBeam 5AC 300, PowerBeam 5AC 300 ISO, PowerBeam 5AC 400, PowerBeam 5AC 400 ISO, PowerBeam 5AC Gen2, PowerBeam 5AC ISO Gen2, LiteBeam 5AC 16 120, LiteAP AC, LiteBeam 5AC 23, LiteBeam 5AC Gen2, LiteBeam 5AC Long-Range, LiteAP GPS, ISO Station 5AC	WA	WA-8.7.1.200623.bin	8.8 MB	22 Jun 2020
UBNT	8.7.1	Rocket 5AC PTMP, Rocket 5AC PTP, Rocket 5AC Lite, Rocket 5AC Prism, Rocket Prism 5AC Gen2, NanoBeam 5AC 19, NanoBeam 5AC Gen2, PowerBeam 5AC 500, PowerBeam 5AC 500 ISO, PowerBeam 5AC 620, PowerBeam 5AC 620 ISO, PrismStation 5AC	XC	XC-8.7.1.200623.bin	8.8 MB	22 Jun 2020
UBNT	4.2.1	UFiber OLT, UFiber OLT 4	e600	e600-4.2.1.200615.tar	85.2 MB	14 Jun 2020
UBNT	1.2.2	EdgeSwitch 10XP, EdgeSwitch 10X, EdgeSwitch 18X, EdgeSwitch 26X	ESX	esx-1.2.2.200521.bix	7.6 MB	20 May 2020

Figura 29. Administrador de firmware (Módulo de red)

El “task manager” o administrador de tareas, proporciona información importante sobre los procesos que se están ejecutando dentro del sistema.



STATUS	PROGRESS	TASK	DATE	TIME	DURATION	ACTIONS
No tasks found						
There are currently no tasks.						

Figura 30. Administrador de tareas (Módulo de red)

UNMS muestra los “outages” o cortes eléctricos en caso de que haya habido alguno.

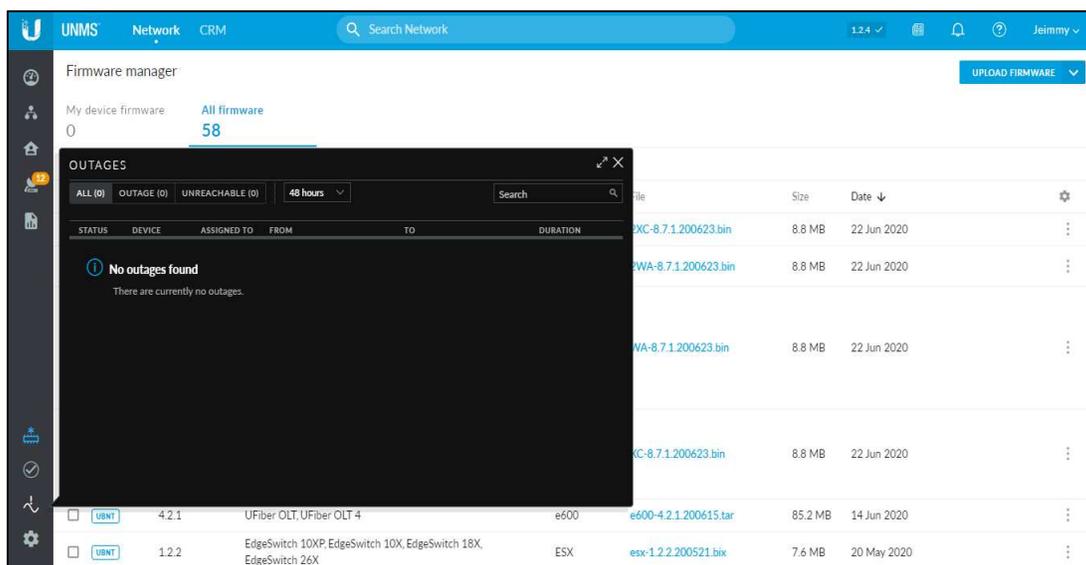


Figura 31. Apagones (Módulo de red)

Por último, se encuentran la opción de “settings” o ajustes, en este punto se muestran los parámetros generales de UNMS y se permiten modificar las configuraciones de la red (servidor UNMS, mantenimiento, copias de seguridad, usuarios, dispositivos, bóveda de credenciales, red e importaciones).

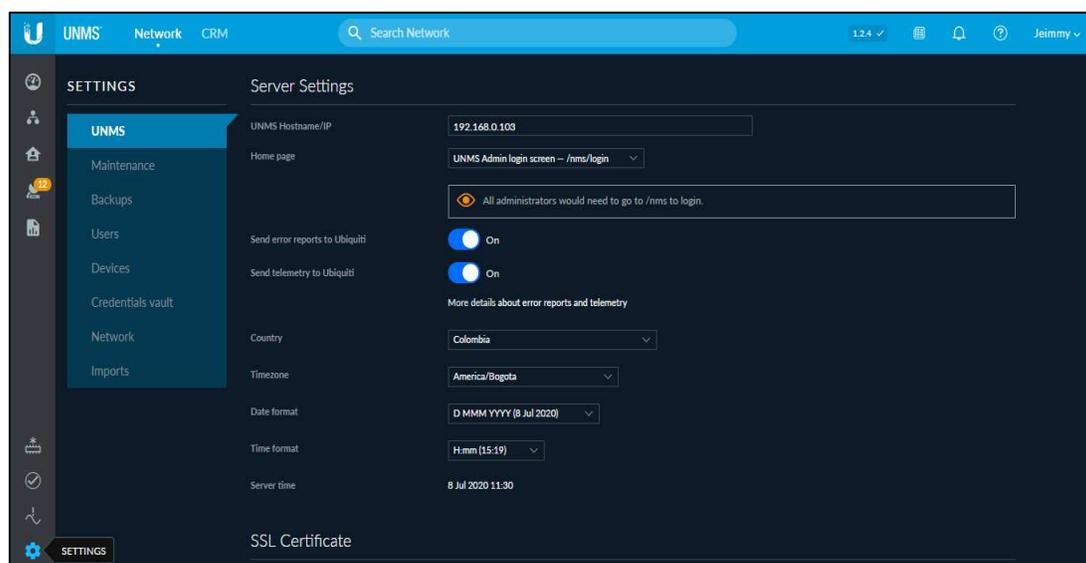


Figura 32. Ajustes (Módulo de red)

- **CRM (Módulo CRM)**

UNMS cuenta con el módulo de CRM, el cual permite la gestión optimizada de clientes y negocios. Al igual que el módulo de red, este cuenta con la opción de “dashboard” o tablero, donde se muestra una visión general de clientes y el estado de la red.

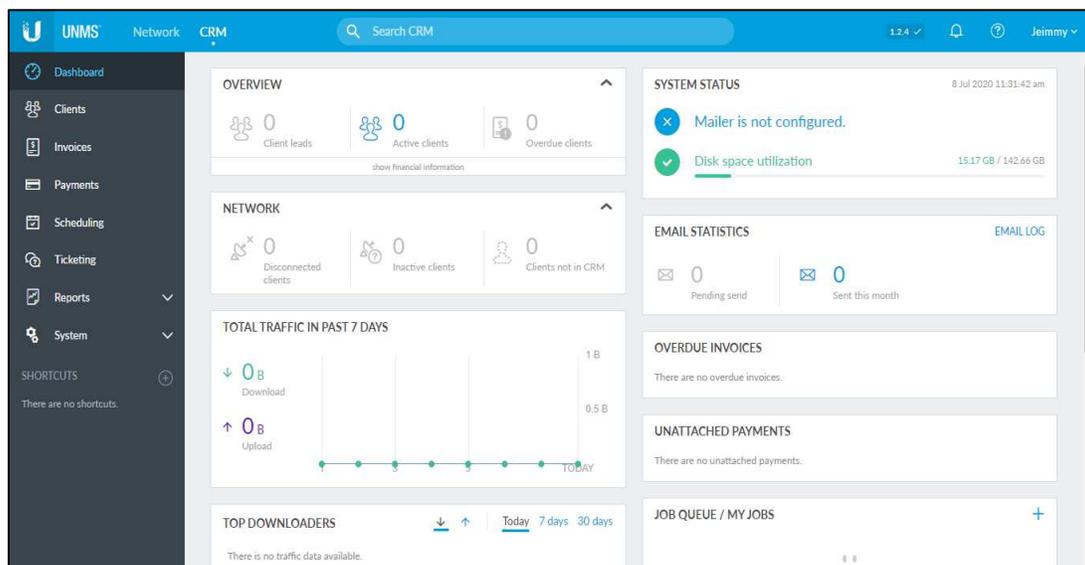


Figura 33. Tablero (Módulo CRM)

En segundo lugar, está la opción “Clients” o clientes, donde se puede agregar un nuevo cliente completando los campos necesarios, entre los cuales se encuentran sus datos personales y ubicación o se pueden importar clientes de archivos CSV (Valores separados por comas).

Figura 34. Clientes (Módulo CRM)

En el área de “Invoices” o facturas, se realiza la creación y personalización de facturación y facturas recurrentes.

INVOICE NUMBER	CLIENT	TOTAL	AMOUNT DUE	CREATED DATE	DUE DATE
There are no rows in the table...					

Figura 35. Facturas (Módulo CRM)

El apartado de “Payments” se permite añadir pagos, multipagos, pagos de importación y reembolsos.

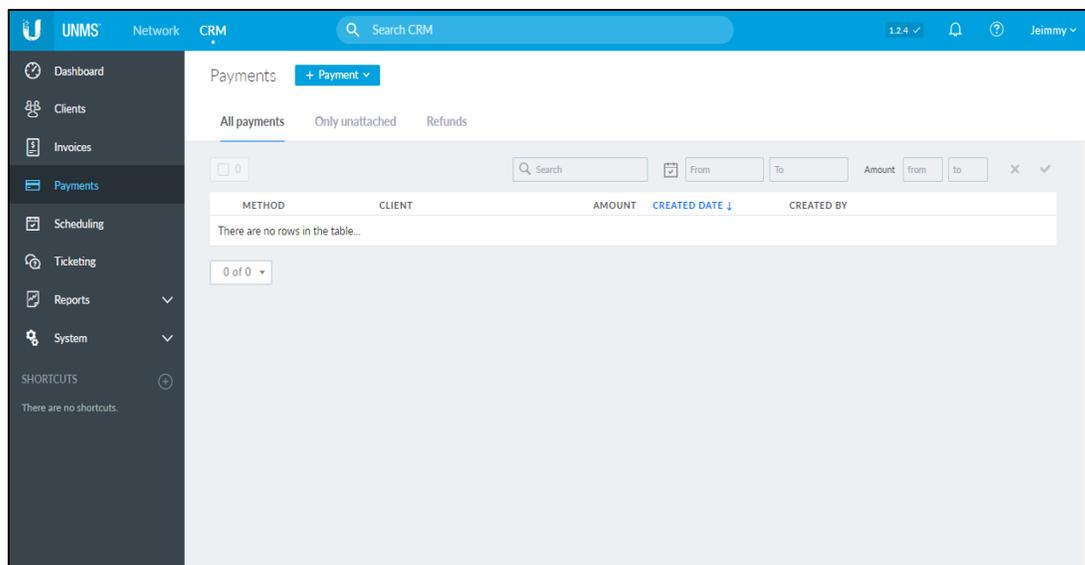


Figura 36. Pagos (Módulo CRM)

Por otra parte, está disponible la opción “Scheduling” o planificación, donde se programan trabajos o tareas para el equipo dentro de la organización.

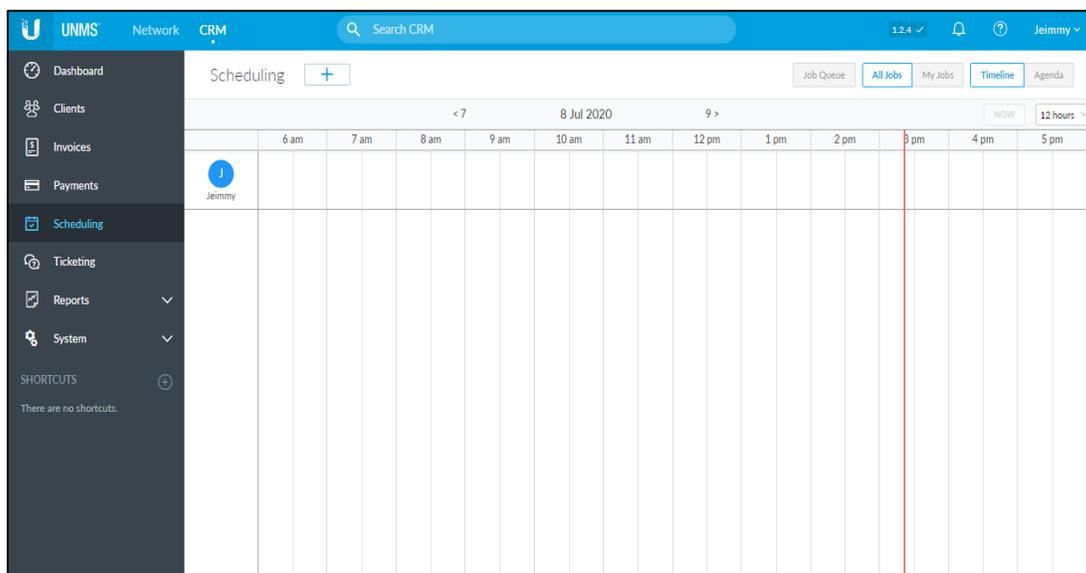


Figura 37. Planificación (Módulo CRM)

La sección de “ticketing” mejora la atención al cliente, permite la interacción con los clientes a través del sistema de venta de entrada integrado en la zona de clientes o a través de correos electrónicos extraídos automáticamente por UNMS.

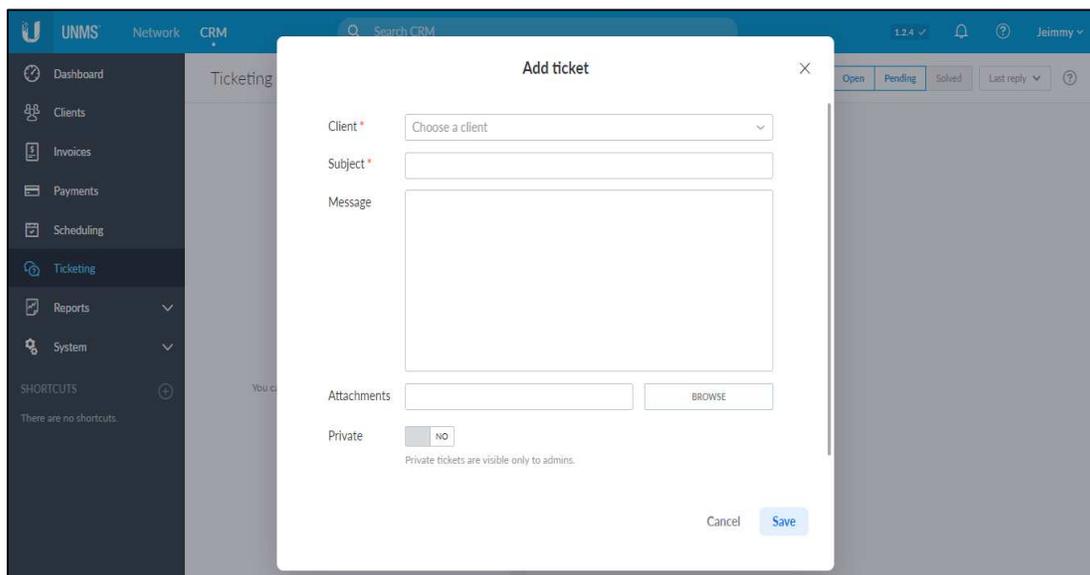


Figura 38. Ticketing (Módulo CRM)

La opción de “Reports” en este módulo se divide en dos secciones, en primera instancia se encuentra el uso de datos, el cual muestra información de los clientes, con respecto al consumo de internet.

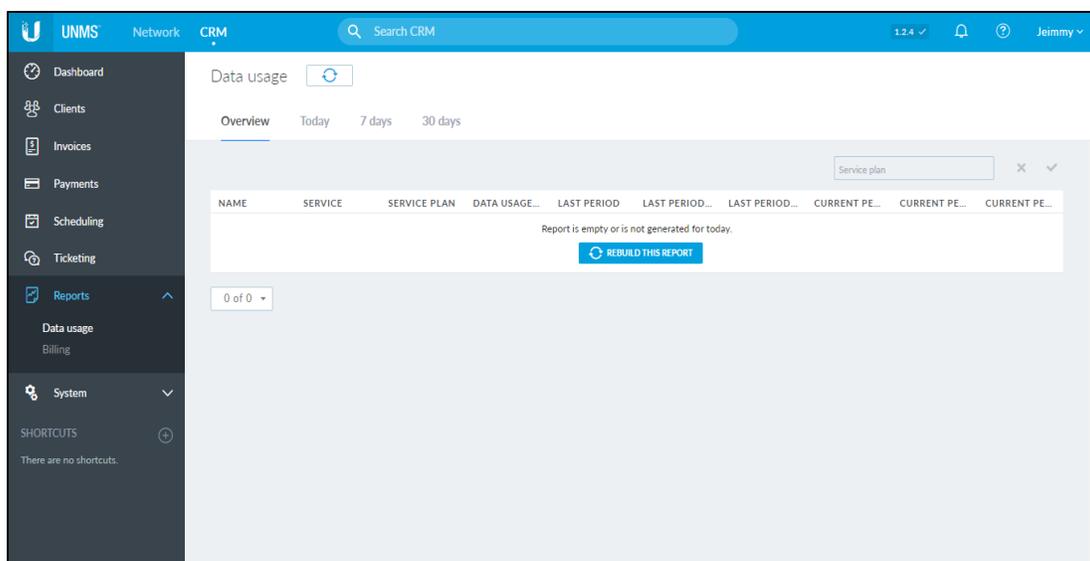


Figura 39. Reportes, uso de datos (Módulo CRM)

En segunda instancia, la sección de facturación muestra todas las facturas y los ingresos facturados (total en pesos de facturas no pagadas, atrasadas y pagadas)

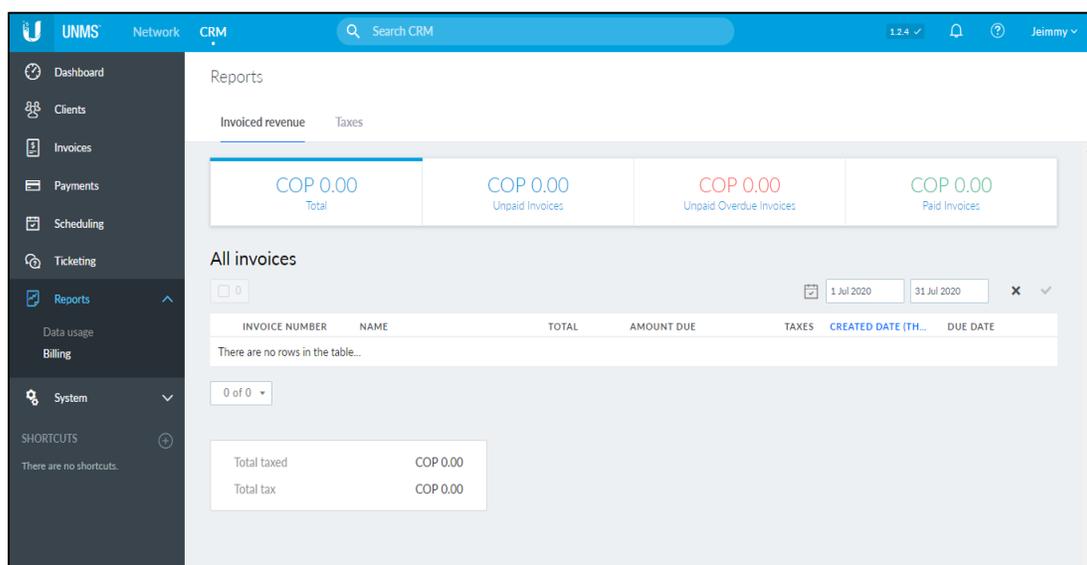


Figura 40. Reportes, facturación (Módulo CRM)

El módulo CRM cuenta con la sección de “System” o sistema, en la cual se permite realizar ajustes generales de clientes, organizaciones, planes de servicios y productos,

facturación, personalización, herramientas, seguridad, complementos, webhooks, registros y otros.

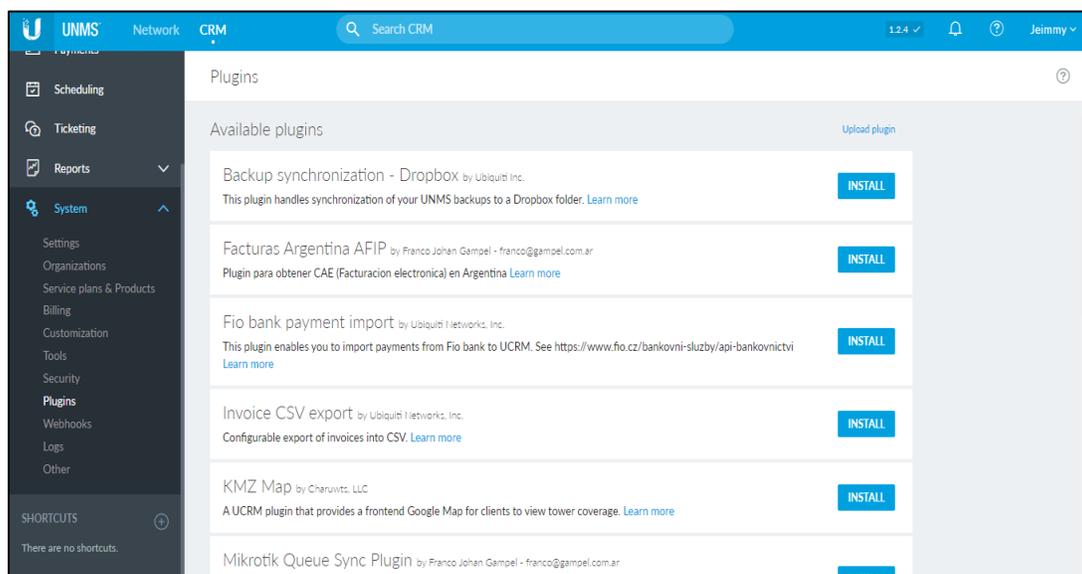


Figura 41. Ajustes (Módulo CRM)

4. Entorno de desarrollo

Para la creación del plugin fue necesario preparar el entorno de desarrollo. Se realizó la instalación de PHP, Git y Composer.

- **PHP**

PHP (Hypertext Preprocessor) es un lenguaje de programación de código abierto utilizado frecuentemente en el desarrollo web, con una gran librería de funciones (php.net, s.f.).

Para descargar e instalar PHP, se ingresa a <https://windows.php.net/download/>

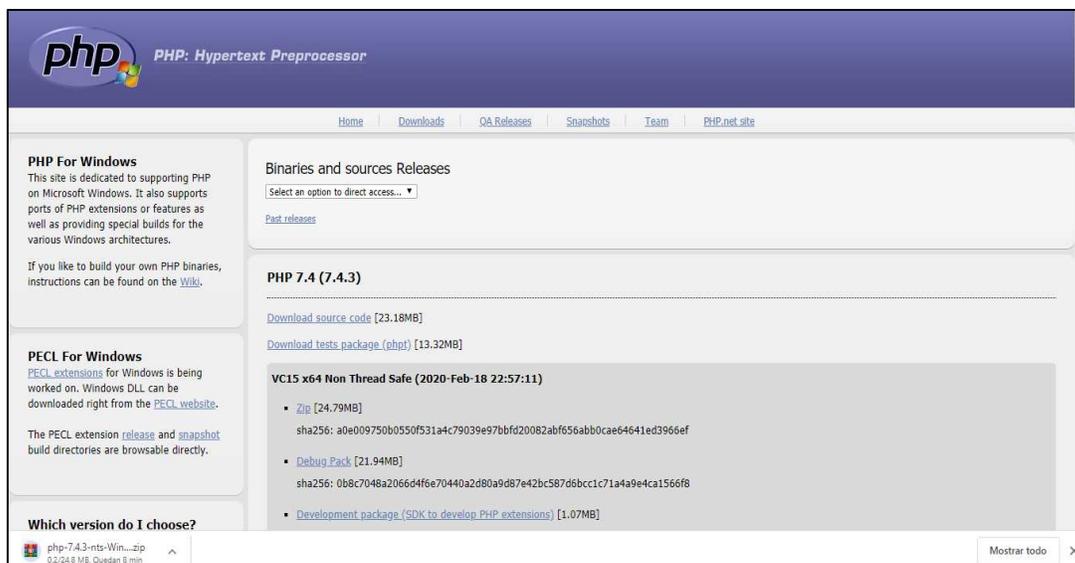


Figura 42. Descarga de PHP

Se crea una carpeta llamada PHP7, en Disco local(C:). Una vez descargado el archivo comprimido, se extrae en la carpeta creada anteriormente.

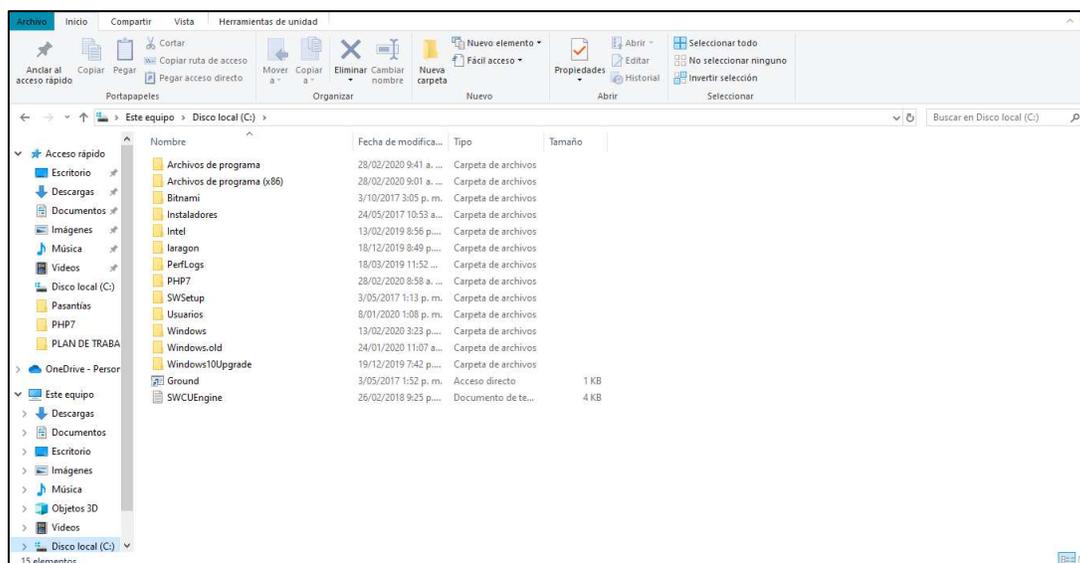


Figura 43. Carpeta PHP7

Dentro de la carpeta, se localiza el archivo “php.ini-development” para duplicarlo y modificar el nombre a “php.ini”.

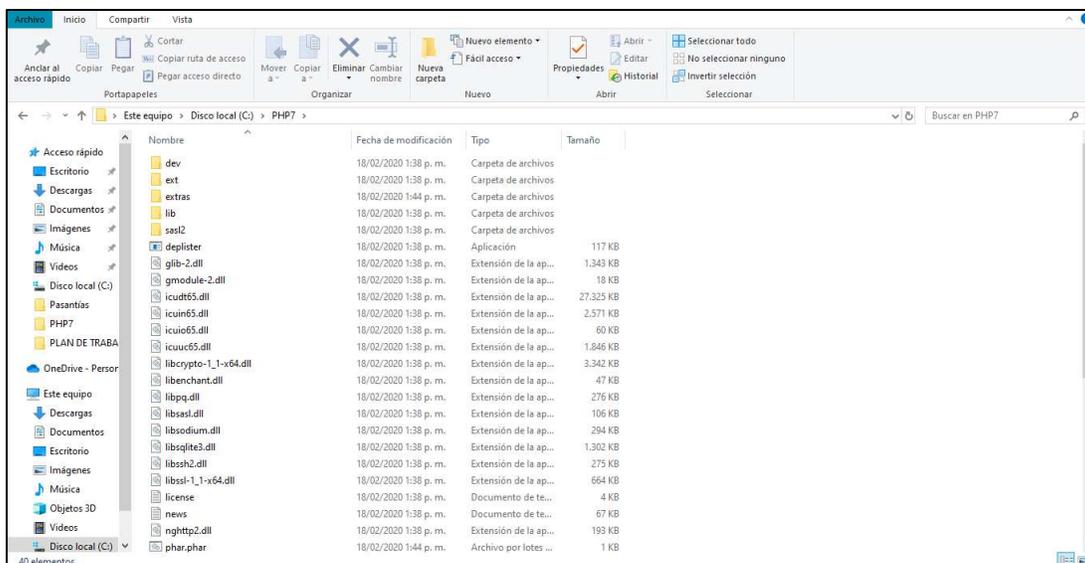


Figura 44. Ubicación de “php.ini”

Se abre el archivo php.ini que se creó a partir de la copia.

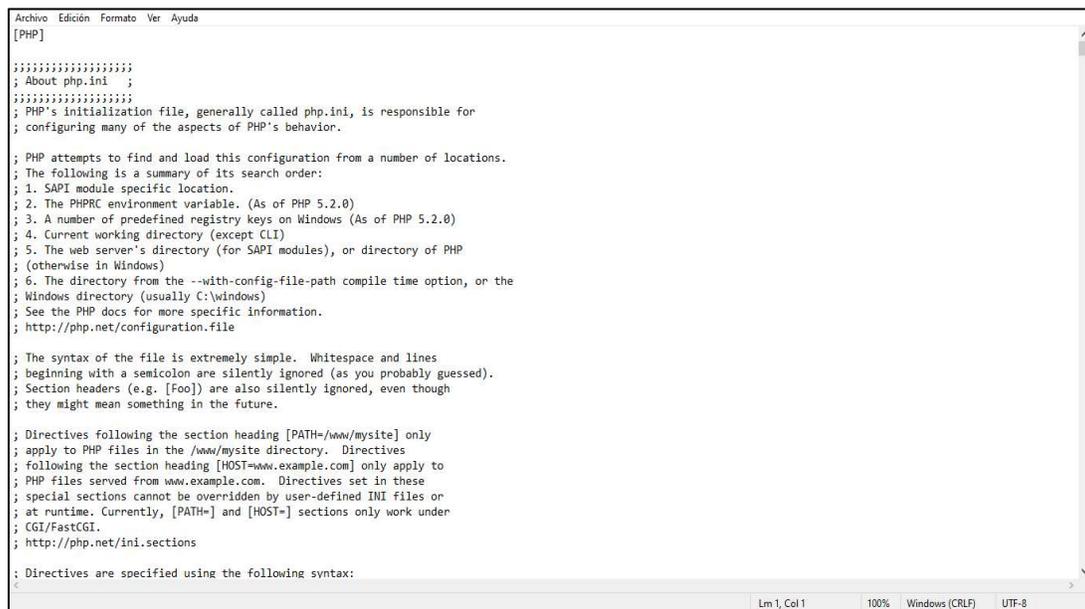


Figura 45. Archivo "php.ini"

Se ubican las siguientes líneas y se les quita el “;” para evitar que sean comentarios.

```
;extension_dir = "ext"
```

```
;extension=curl
```

```
;extension=fileinfo
```

```
;extension=gd2
```

```
;extension=gettext
```

```
;extension=gmp
```

```
;extension=mysqli
```

```
;extension=pdo_sqlite
```

```
;extension=pgsql
```

```
;extension=shmop
```

Luego, se entra al sistema, para modificar las variables de entorno.

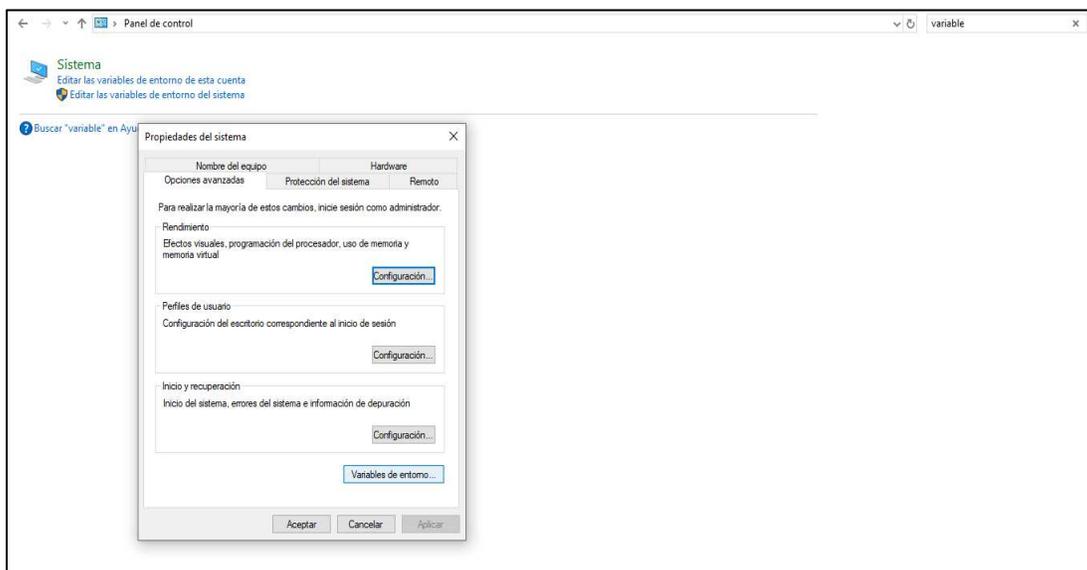


Figura 46. Variables de entorno

Se edita la variable del sistema “Path”.

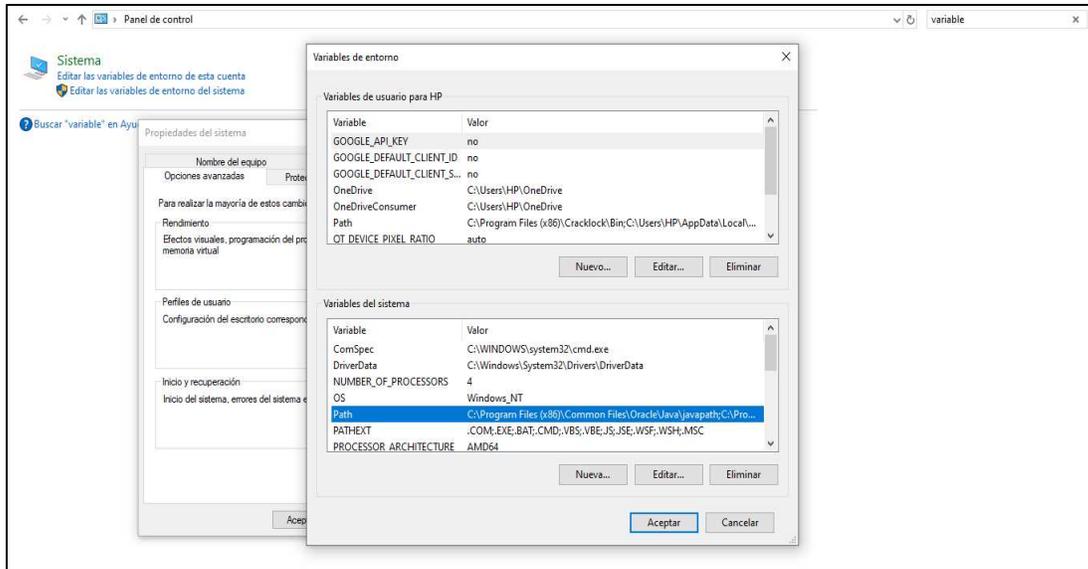


Figura 47. Variable "Path"

Seguidamente, se crea la variable “PHP7”.

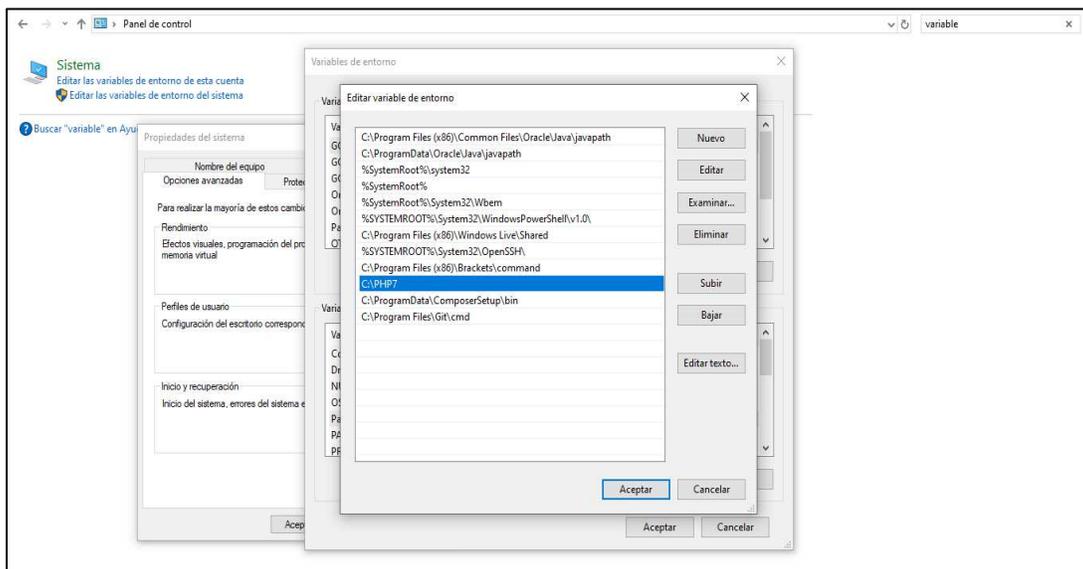
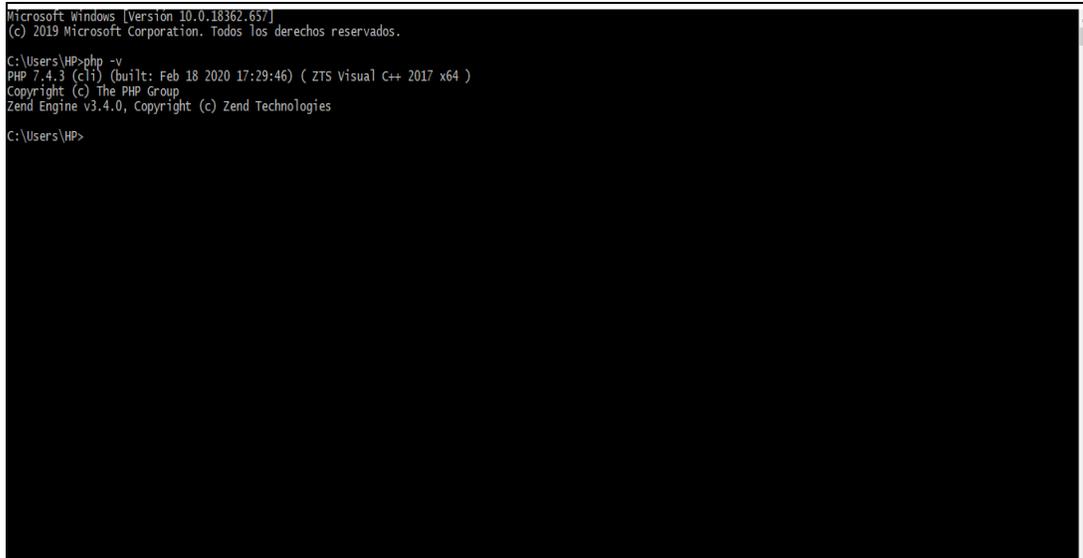


Figura 48. Creación de la variable "PHP7"

Para comprobar que PHP se instaló correctamente se ingresa al simbolo del sistema y se ejecuta el comando “php -v”.



```
Microsoft Windows [Versión 10.0.18362.657]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\HP>php -v
PHP 7.4.3 (cli) (built: Feb 18 2020 17:29:46) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies

C:\Users\HP>
```

Figura 49. Instalación correcta de PHP

- **Git**

Git es un software de control de versiones distribuido que modela sus datos, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones con gran número de archivos de código fuente (Gómez, Aula software libre, Universidad de Córdoba, pág. 10)

Para su instalación se ingresa a <https://git-scm.com/download/win> el sitio web detecta el sistema operativo del equipo y sugiere descargar una versión de Git.



Figura 50. Descarga de git

Una vez finalizada la descarga, se inicia el proceso de instalación.

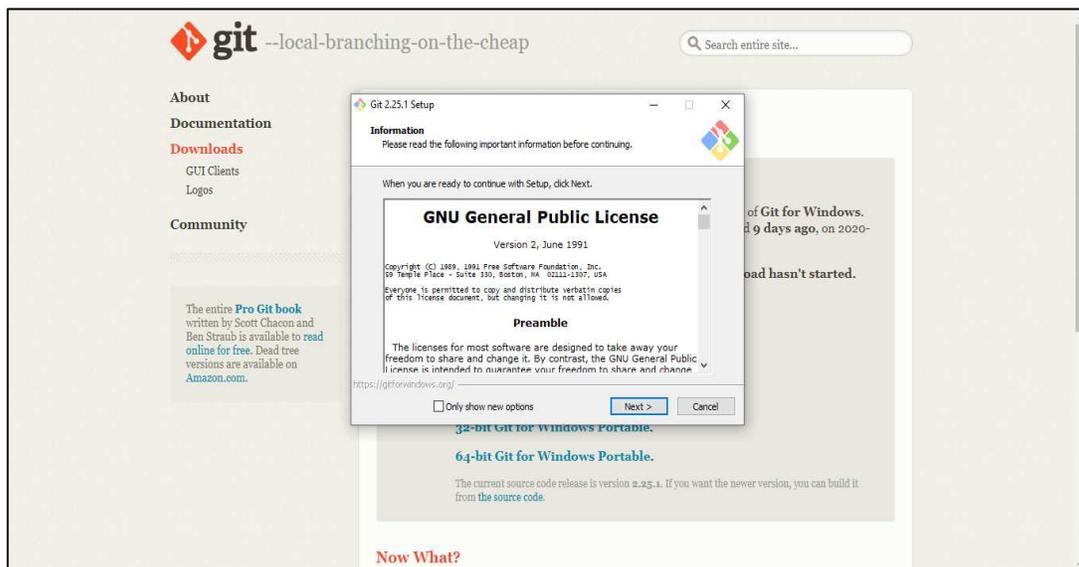


Figura 51. Proceso de instalación

Luego, se seleccionan los componentes que deben ser instalados.

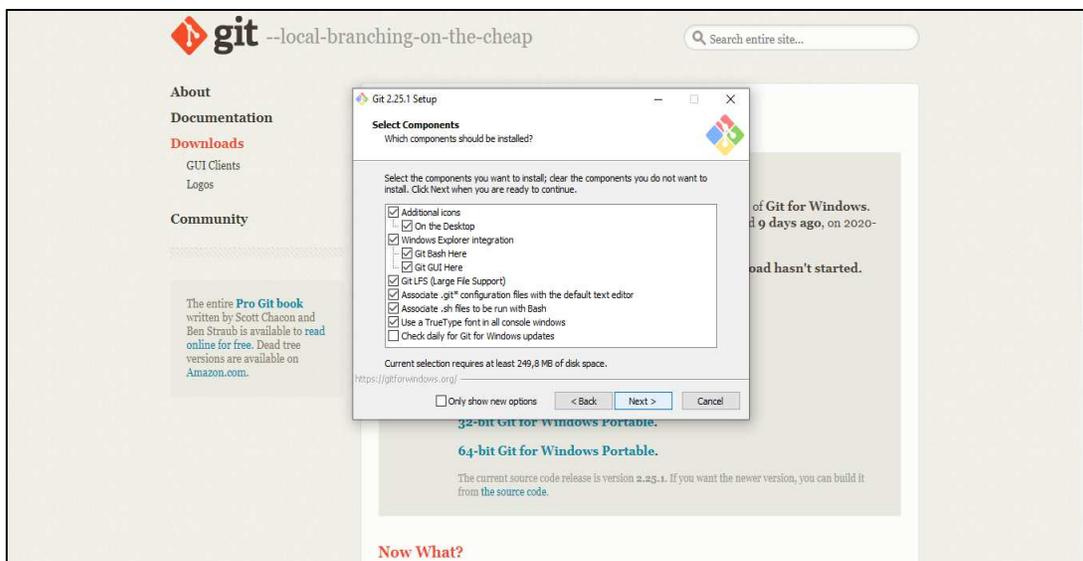


Figura 52. Componentes a instalar

Se elige el editor que se asocia con Git.

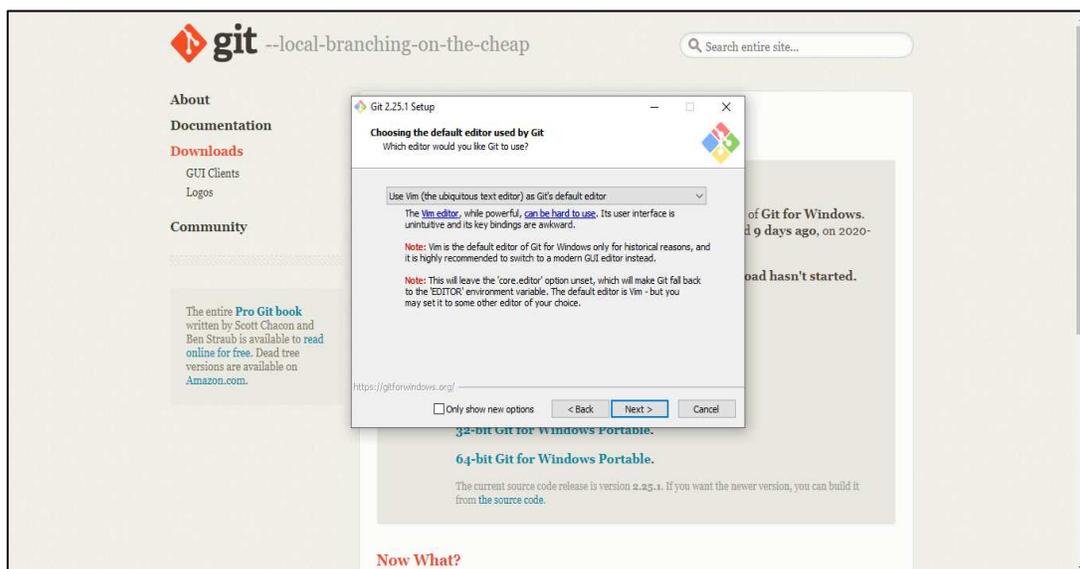


Figura 53. Elección del editor

Se selecciona la forma en que se ajustarán las variables del entorno (PATH).

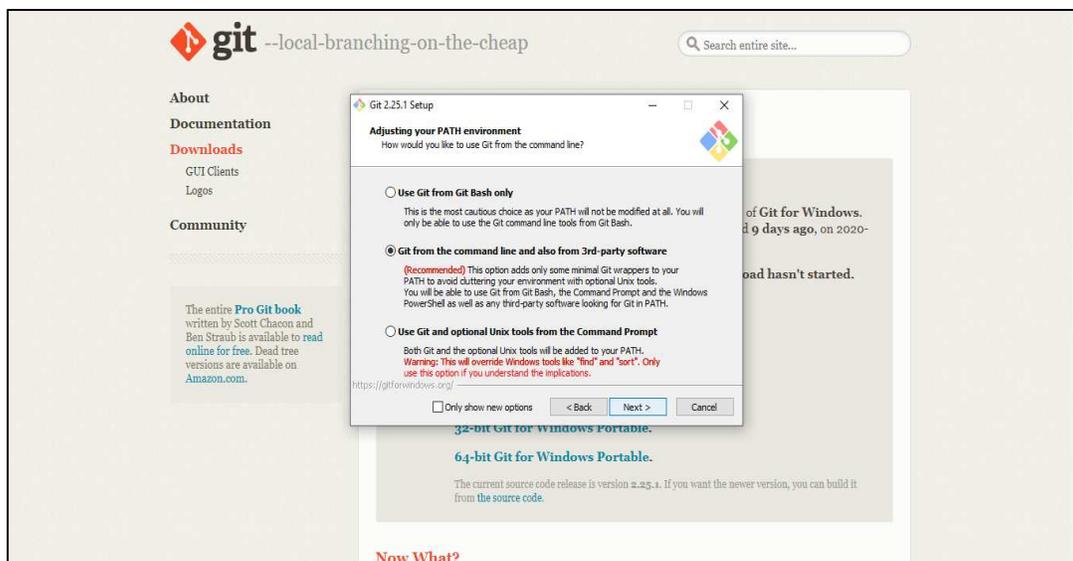


Figura 54. Ajuste de variables de entorno

Se deja por defecto la librería OpenSSL (El paquete OpenSSL contiene herramientas de administración y librerías).

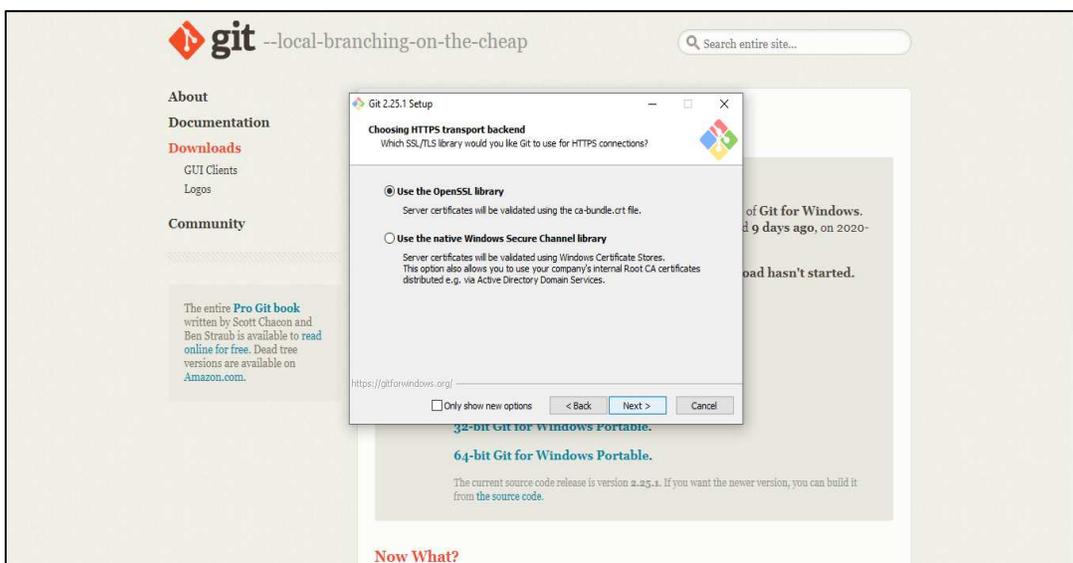


Figura 55. Librería OpenSSL

A continuación, se deja la opción por defecto para los caracteres de fin de línea.

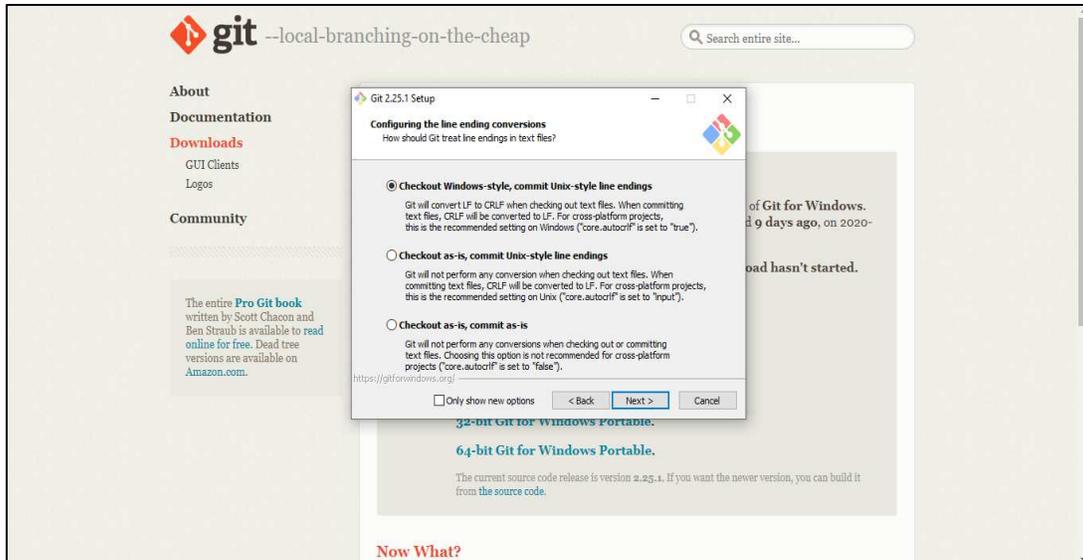


Figura 56. Caracteres de fin de línea

Se usa MinTTY, es cual es un emulador de terminal gratuito y de código abierto.

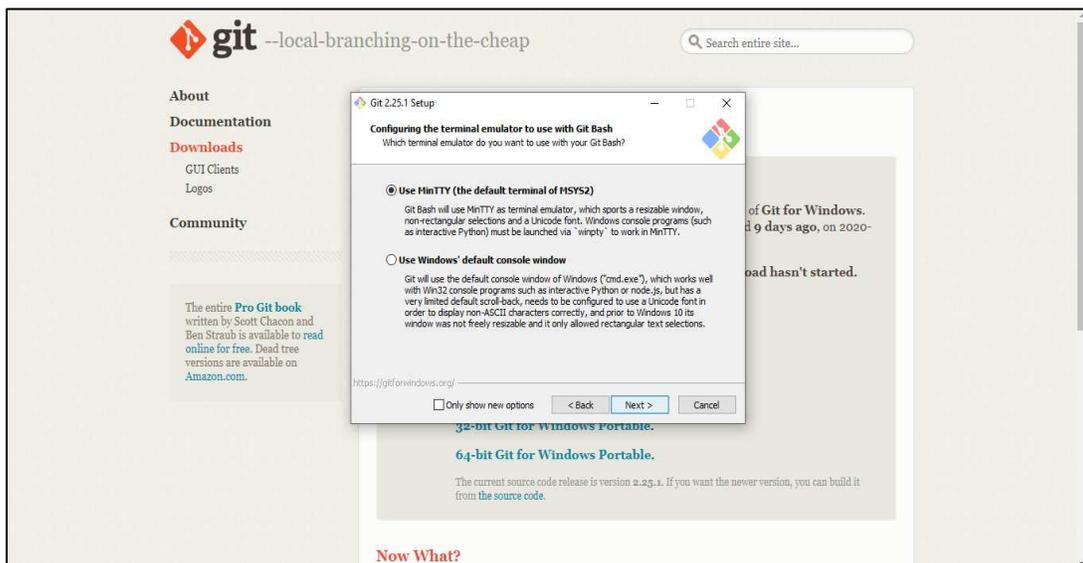


Figura 57. Emulador MinTTY

Seguidamente, se muestran todas las configuraciones extras de Git.

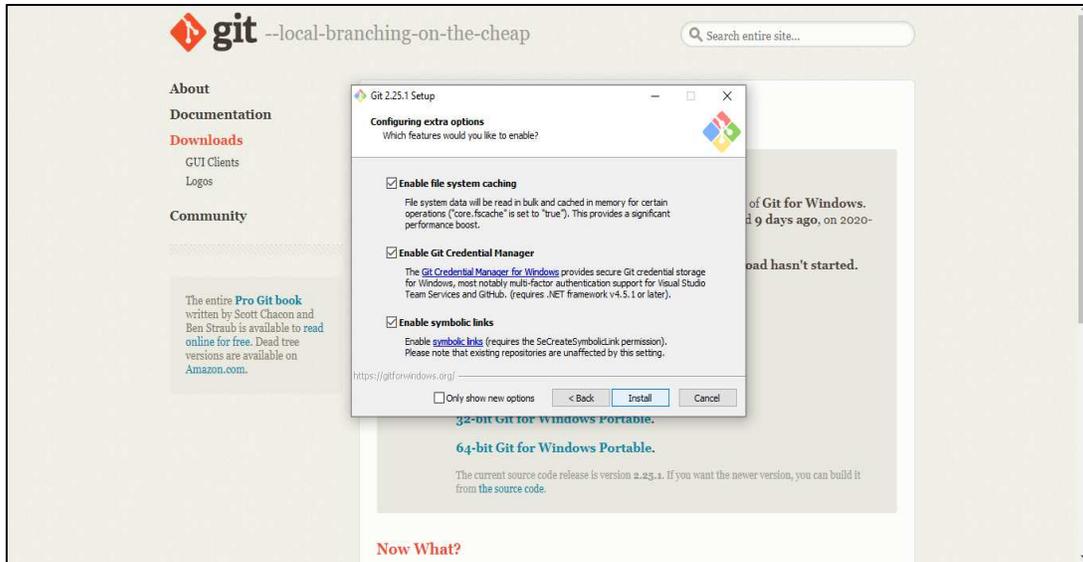


Figura 58. Configuraciones extras de Git

Por ultimo, se inicia la instalación.

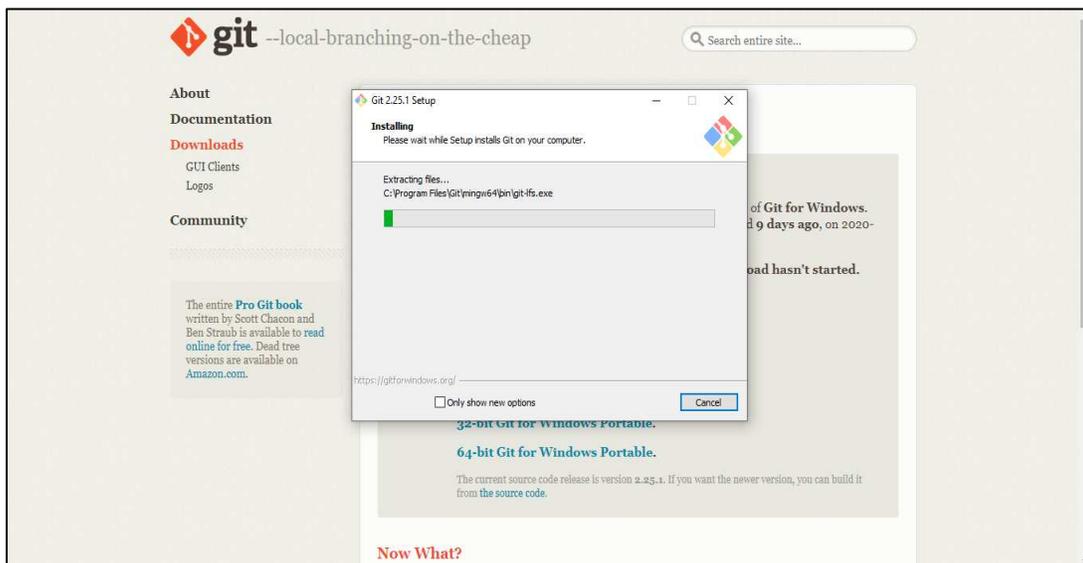


Figura 59. Inicio de instalación de Git

- **Composer**

Composer es una herramienta utilizada para la gestión de dependencias en PHP. Permite declarar las bibliotecas de las que depende un proyecto y las administra (instala / actualiza).

(Composer)

Para instalar esta herramienta se ingresa a <https://getcomposer.org/> y se inicia la descarga “Composer-Setup.exe”.

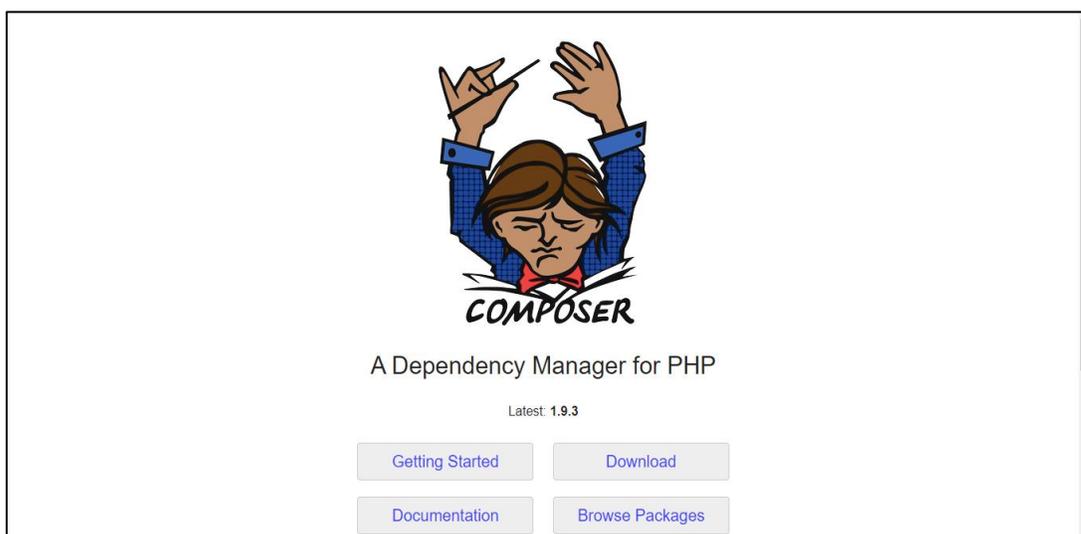


Figura 60. Descarga de Composer

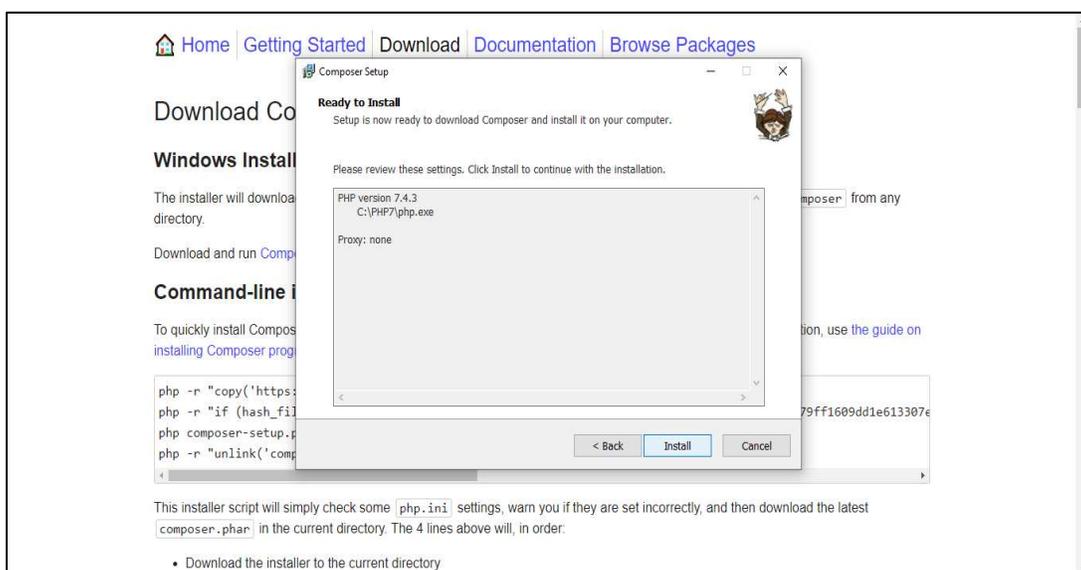


Figura 61. Instalación de Composer

5. Plugins

El sistema de gestión de red de Ubiquiti (UNMS) permite desarrollar complementos propios o mejorar uno existente, añadiendo una nueva funcionalidad o característica al software. (Ubiquiti)

Estos Plugins están bajo licencia MIT (Massachusetts Institute of Technology), lo que permite a cualquier persona contribuir con cualquier actualización a los complementos existentes o crear otros nuevos. (php.net, s.f.)

Los complementos CRM tienen una estructura de archivos específica. (Exner & Procházka, GitHub - Ubiquiti-App, 2019) (Exner, Tousek, & Vavra, GitHub - Ubiquiti-App, 2019)

- Archivos requeridos u obligatorios

manifest.json: Este archivo contiene información necesaria sobre el Plugin (nombre, autor, configuración requerida, etc.)

main.php: Archivo principal, se ejecutará cuando el complemento sea ejecutado por CRM.

- Archivos opcionales

public.php: Generará una URL pública para el Plugin que apuntará a este archivo.

“public” directory: Este directorio solo funcionará si se cuenta con un archivo “public.php” y debe estar ubicado en la raíz del Plugin. Todos los archivos dentro de él serán accesibles públicamente sin ninguna restricción.

public/admin-zone.js: Si este archivo existe, se cargará automáticamente en cada página de zona de administración en CRM.

public/client-zone.js: A diferencia del anterior si este archivo existe, se cargará automáticamente en cada página de zona de cliente en CRM.

hook_*.php files: Al instalar, actualizar, configurar, habilitar, deshabilitar o eliminar el Plugin, CRM llamará automáticamente al respectivo archivo de enlace que puede usar para ejecutar cualquier comando personalizado. Estos archivos deben colocarse en la raíz del complemento, al lado del archivo “public.php”.

"data" directory: Este directorio está protegido entre actualizaciones del complemento. No se tocará nada en este directorio.

data/config.json: La configuración del complemento (es decir, los parámetros del complemento y sus valores) se guardarán en este archivo.

data/plugin.log: Todo lo que contenga este archivo se mostrará como texto en la página de detalles del complemento en CRM.

data/files directory: Contiene los archivos cargados con la configuración de tipo “file”.

- Archivos reservados

ucrm.json: Archivo generado automáticamente, creado después de instalar el Plugin en CRM. Contiene configuración preparada, que el complemento puede usar de inmediato. La configuración se actualiza automáticamente cuando se realizan cambios en la configuración de CRM.

.ucrm-plugin-running: Este archivo se utiliza para evitar múltiples ejecuciones de complementos si la instancia anterior aún se está ejecutando. Esta medida se usa solo para la ejecución manual y para la ejecución automática.

.ucrm-plugin-execution-requested: Este archivo se utiliza para planificar la ejecución del complemento desde la interfaz CRM, independientemente del período de ejecución elegido.

- **Plugin de prueba**

A continuación, se muestra un Plugin desarrollado como una prueba, con el fin de conocer su estructura, el proceso de instalación en UNMS y familiarizarse con el entorno.

Se crea una carpeta llamada “pluginprueba”

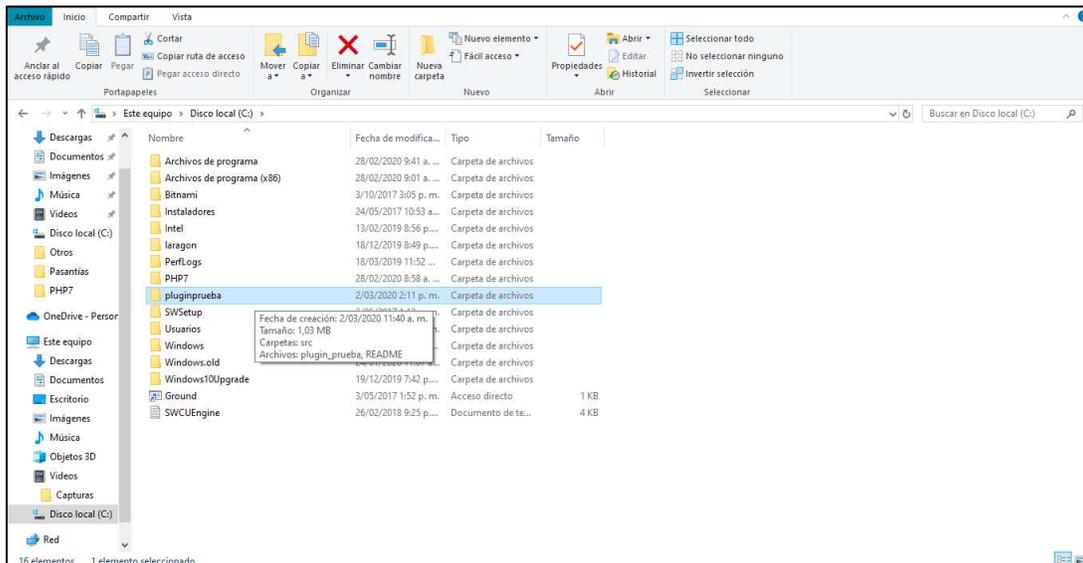


Figura 62. Creación "pluginprueba"

Dentro de la carpeta creada anteriormente se ubica el “archivo ZIP esqueleto” previamente descomprimido, descargado de <https://github.com/Ubiquiti-App/UCRM-plugins/blob/master/docs/tutorials/first-plugin.md>.

Contiene la siguiente estructura:

- README.md
- src/.gitignore
- src/composer.json
- src/main.php
- src/manifest.json

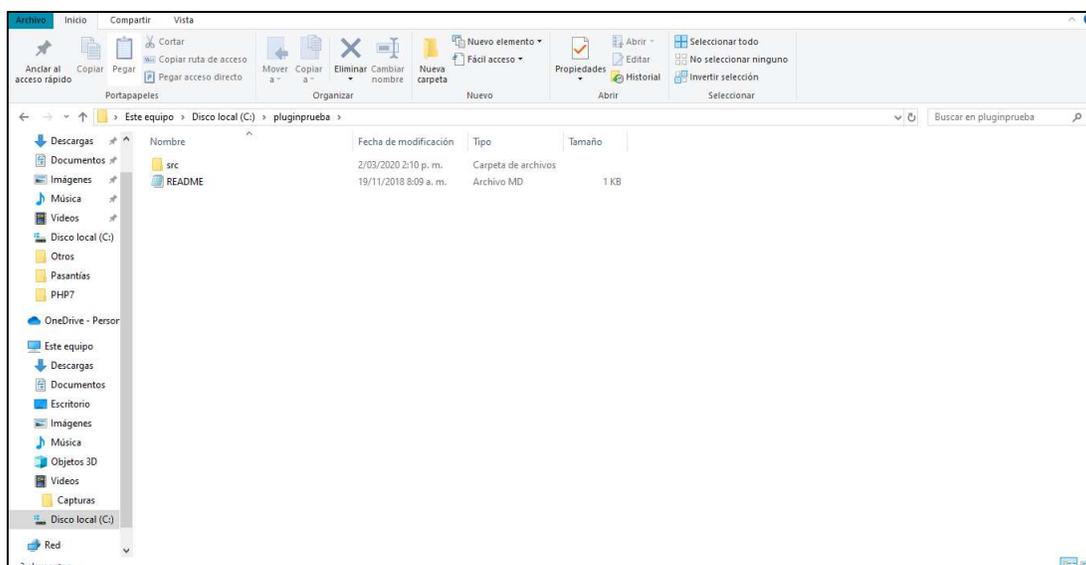


Figura 63. Archivo ZIP esqueleto

Se continua con la preparación del archivo “manifest.json”, el cual describe el Plugin para CRM y contiene lo siguiente:

- version: Determina la versión del archivo de configuración.
- information: Describe el Plugin.
 - name - Nombre del Plugin
 - displayName - Nombre del Plugin como se mostrará en la interfaz de CRM.
 - Description - Descripción del Plugin.
 - url - Enlace a la página del Plugin.
 - unmsVersionCompliance/ucrmVersionCompliance: Versión del UNMS y CRM en las que funcionara el Plugin.
 - Autor - Creador del Plugin.
- configuration: Determina las claves de configuración del Plugin.
 - key - clave de propiedad.

label - Etiqueta de la propiedad.

description - Descripción de la propiedad.

required - La propiedad puede ser obligatoria u opcional.

type - Tipo de elemento de configuración.

choices - Define las posibles opciones para el choice.

- menu: Agrega enlaces al complemento en el menú CRM.

type - Puede ser “admin” (se mostrará en la zona de administración) o “client” (se mostrará en la zona del cliente)

target - Puede ser “blank” (El enlace lo llevará a la página de destino) o “iframe” (El enlace lo llevará a una página especial dentro de CRM).

key - Categoría de menú para insertar el enlace.

label – Etiqueta del enlace.

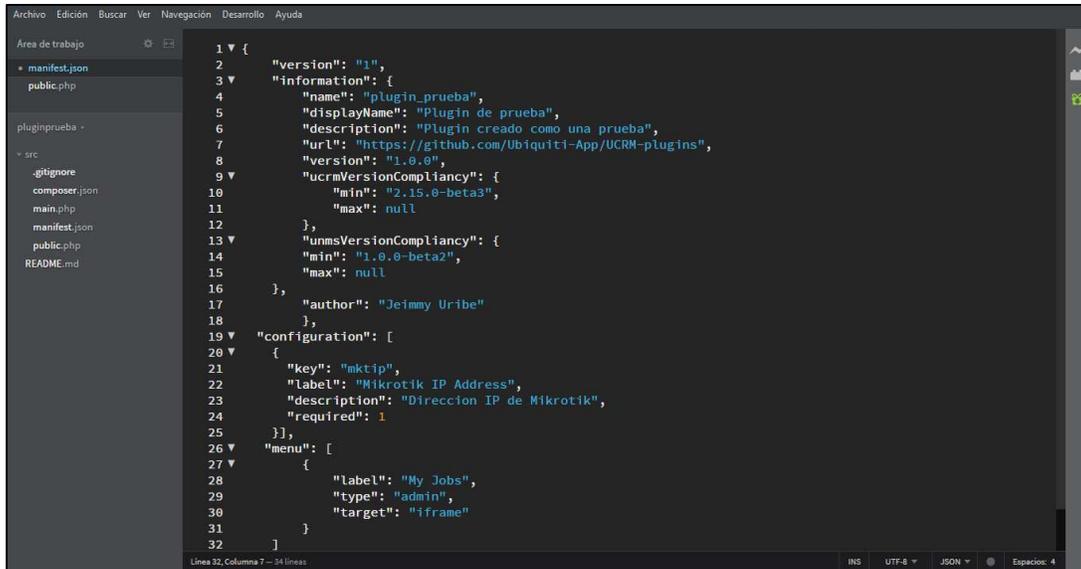
parameters – Matriz de parámetros para el enlace.

- widgets: Muestran la página pública del complemento en posiciones especificadas.

position - Determina donde mostrar el widget.

iframeHeight - Altura del iframe del widget.

IframeUrlParameters - Matriz de parámetros para el enlace.



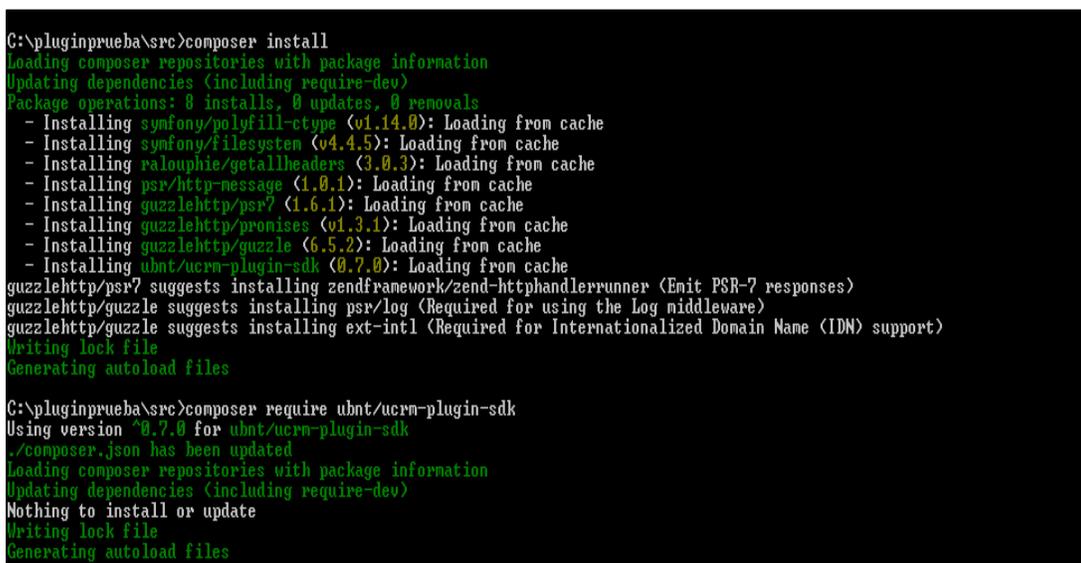
```

1 {
2   "version": "1",
3   "information": {
4     "name": "plugin_prueba",
5     "displayName": "Plugin de prueba",
6     "description": "Plugin creado como una prueba",
7     "url": "https://github.com/Ubiquiti-App/UCRM-plugins",
8     "version": "1.0.0",
9     "ucrnVersionCompliance": {
10      "min": "2.15.0-beta3",
11      "max": null
12    },
13    "ummsVersionCompliance": {
14      "min": "1.0.0-beta2",
15      "max": null
16    },
17    "author": "Jeimmy Uribe"
18  },
19  "configuration": [
20    {
21      "key": "mktip",
22      "label": "Mikrotik IP Address",
23      "description": "Direccion IP de Mikrotik",
24      "required": 1
25    }
26  ],
27  "menu": [
28    {
29      "label": "My Jobs",
30      "type": "admin",
31      "target": "iframe"
32    }
33  ]
34 }

```

Figura 64. Archivo “manifest.json”

Es necesario Instalar dependencias de complementos, el esqueleto contiene UCRM Plugin SDK como una dependencia de Composer. Para instalarlo se debe ir al directorio “src” y ejecutar “composer install”.



```

C:\pluginprueba\src>composer install
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 8 installs, 0 updates, 0 removals
 - Installing symfony/polyfill-ctype (v1.14.0): Loading from cache
 - Installing symfony/filesystem (v4.4.5): Loading from cache
 - Installing ralouphie/getallheaders (3.0.3): Loading from cache
 - Installing psr/http-message (1.0.1): Loading from cache
 - Installing guzzlehttp/psr7 (1.6.1): Loading from cache
 - Installing guzzlehttp/promises (v1.3.1): Loading from cache
 - Installing guzzlehttp/guzzle (6.5.2): Loading from cache
 - Installing ubnt/ucrn-plugin-sdk (0.7.0): Loading from cache
guzzlehttp/psr7 suggests installing zendframework/zend-httpdierrunner (Emit PSR-7 responses)
guzzlehttp/guzzle suggests installing psr/log (Required for using the Log middleware)
guzzlehttp/guzzle suggests installing ext-intl (Required for Internationalized Domain Name (IDN) support)
Writing lock file
Generating autoload files

C:\pluginprueba\src>composer require ubnt/ucrn-plugin-sdk
Using version ^0.7.0 for ubnt/ucrn-plugin-sdk
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Nothing to install or update
Writing lock file
Generating autoload files

```

Figura 65. Instalación de complementos

A continuación, se crea un archivo llamado “public.php”.

```

1 <?php
2
3 require_once __DIR__ . '/vendor/autoload.php';
4
5 $api = \Ubn\UcrmPluginSdk\Service\UcrmApi::create();
6 $security = \Ubn\UcrmPluginSdk\Service\UcrmSecurity::create();
7
8 $user = $security->getUser();
9 if (! $user || ! $user->hasViewPermission(\Ubn\UcrmPluginSdk\Security\PermissionNames::SCHEDULING_MY_JOBS)) {
10     die('You do not have permission to view this page.');
```

Figura 66. Archivo "public.php"

Se muestra el archivo “main.php”.

```

1 <?php
2
3 require_once __DIR__ . '/vendor/autoload.php';
4
```

Figura 67. Archivo "main.php"

Para probar el plugin y su funcionamiento en el sistema (UNMS) se debe comprimir el archivo, ejecutando el comando “./vendor/bin/pack-plugin” dentro del src.

```

C:\pluginprueba\src>
C:\pluginprueba\src>cd vendor

C:\pluginprueba\src\vendor>cd bin

C:\pluginprueba\src\vendor\bin>pack-plugin
Plugin root dir not given, trying to find automatically: "C:\pluginprueba\src"
Created plugin ZIP archive: "C:\pluginprueba\plugin_prueba.zip"

C:\pluginprueba\src\vendor\bin>

```

Figura 68. Comando “./vendor/bin/pack-plugin”

Se creará un archivo ZIP junto al README.md, ese archivo es el que se cargará en la sección Sistema -> Complementos de UNMS.

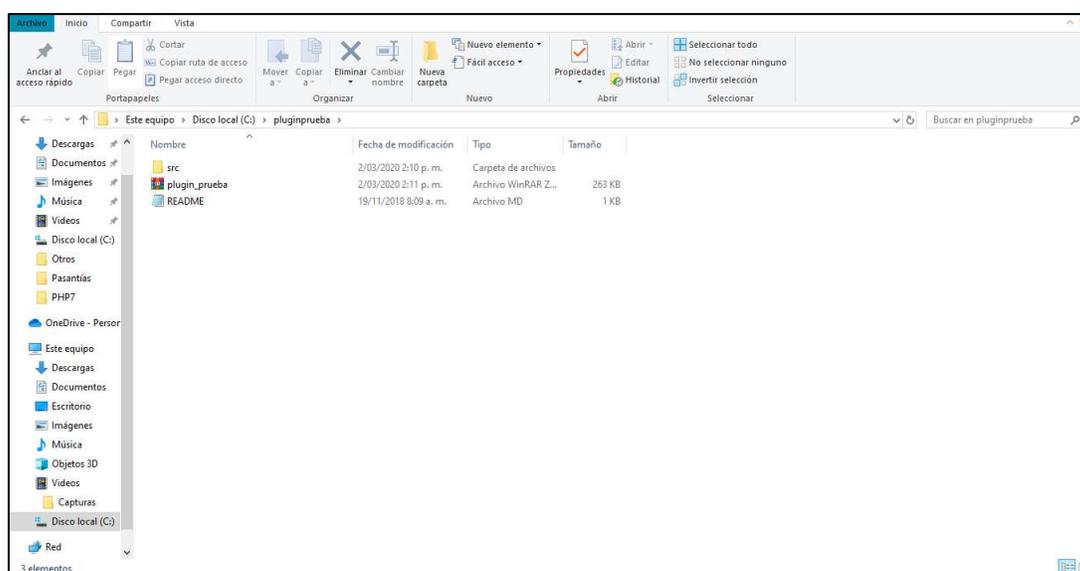


Figura 69. Archivo "plugin_prueba" comprimido

Se inicia el proceso para habilitarlo en el sistema, ingresando a la opción “Sistema -> Complementos -> Subir complemento” del módulo CRM de UNMS.

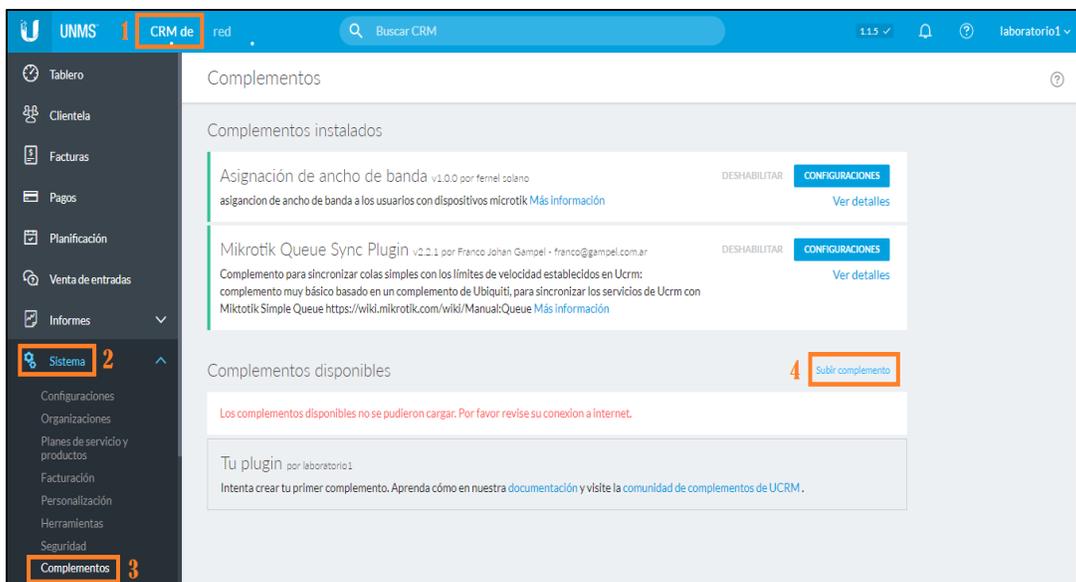


Figura 70. Proceso de "Subir complemento"

Luego, se carga el archivo ZIP "plugin_prueba" y se sube al sistema.

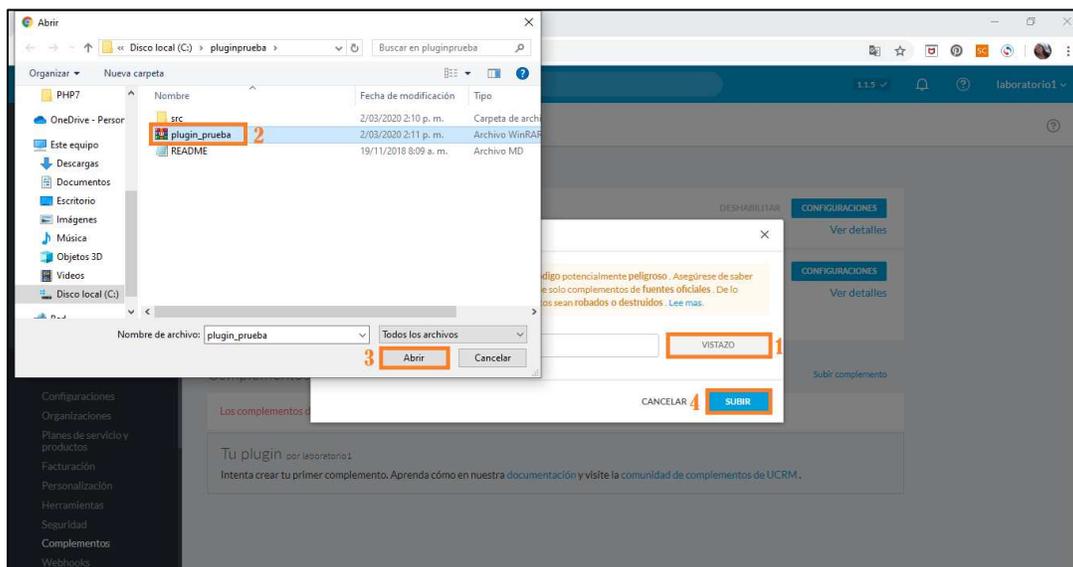


Figura 71. Cargar "plugin_prueba" al sistema

Solo resta habilitarlo y ya estará disponible para usarse.

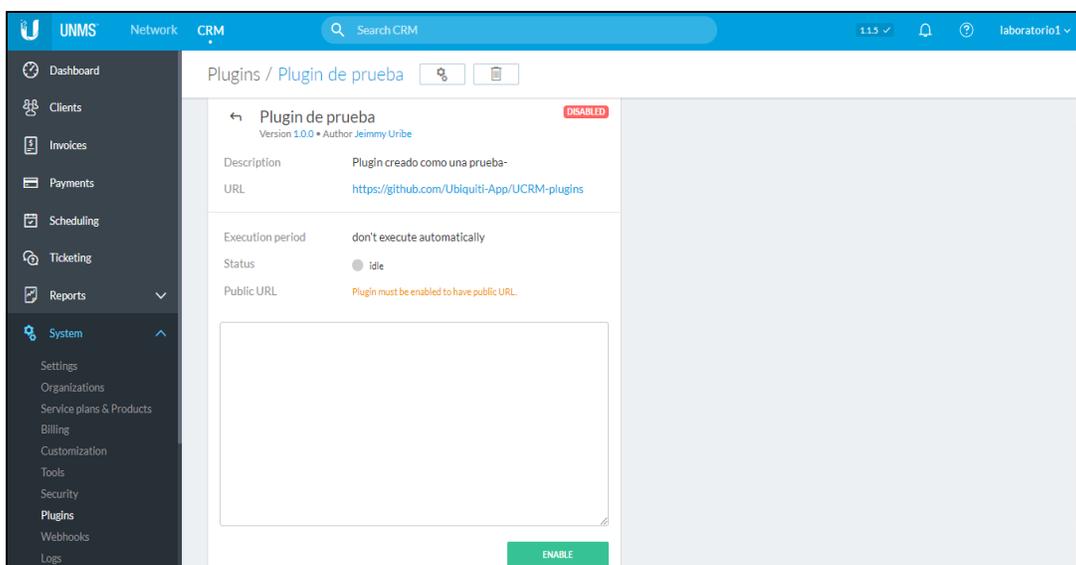


Figura 72. Plugin cargado a UNMS

3.1.2 Objetivo 2. Desarrollar el plugin de mensajería para el software CRM (Gestión de relaciones con los clientes) de la organización.

- Actividad 1. Realizar el plugin de mensajería encargado de notificar a los clientes el pago de sus facturas.
- Actividad 2. Realizar el plugin de mensajería que notifica información de soporte técnico a los clientes.

Inicialmente se planteó realizar dos plugins de mensajería, en el transcurso del proyecto se hizo notorio que ambos se complementaban, debido a esto se decidió hacer uno que cumpliera con todas las especificaciones. El plugin desarrollado en este proyecto se basa en “SMS notifier-Twilio”, un complemento de por Ubiquiti Networks, Inc, el cual utiliza los servicios de Twilio (Plataforma de comunicaciones).

El plugin de mensajería pretende mejorar la comunicación entre TechNet Comunicaciones S.A.S y sus clientes, enviándoles notificaciones SMS. El mensaje es activado por un evento que ocurrió en el CRM (Gestión de relaciones con los clientes) de UNMS (Sistema de gestión de red Ubiquiti), es importante resaltar que se enviarán dichos mensajes solo a los clientes que en sus datos tienen registrado el número de teléfono celular.

Este complemento debe realizar una llamada a la API (interfaz de programación de aplicaciones) de SMSrutas, el servidor contratado por la empresa para envío de mensajes cortos (SMS), para esto fue necesario profundizar en la documentación brindada por SMSrutas, donde especifica los parámetros y métodos utilizados por la API y hacer la respectiva comunicación.

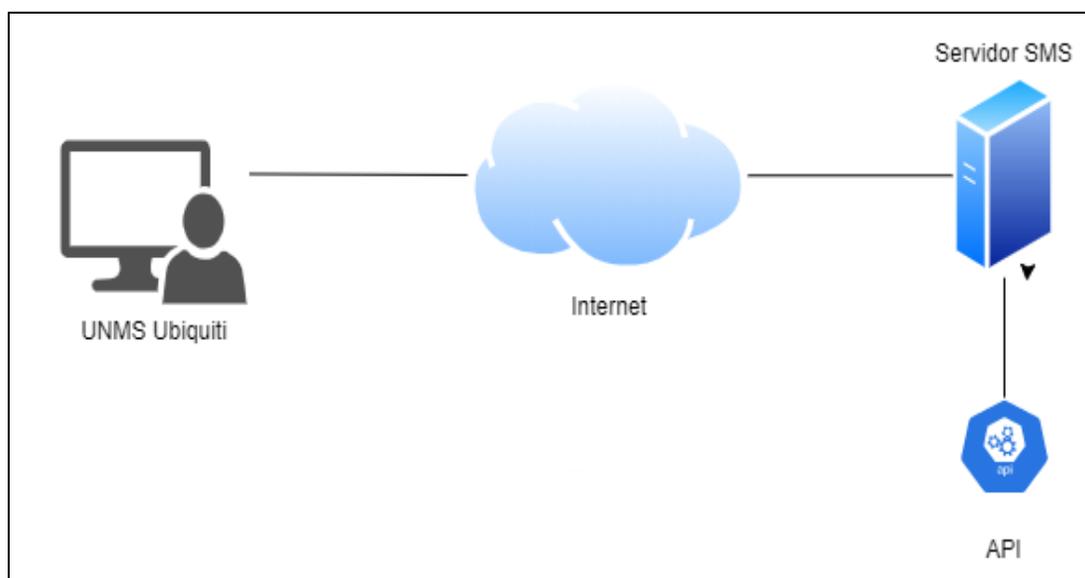


Figura 73. Comunicación entre la API del servidor de SMS y UNMS

La comunicación se realiza en la función `sendMessage()` de la clase `TechNetNotifierFacade.php` que extiende de `AbstractMessageNotifierFacade.php`, esta función recibe tres parámetros:

- \$notificationData
- \$clientSmsNumber
- \$messageBody

```
protected function sendMessage( NotificationData $notificationData, string $clientSmsNumber, string $messageBody): void
{
    $dataApi= $this->datosApiData->getDataApi($clientSmsNumber,$messageBody);
    $this->apiTechnetSMS->getSMS($dataApi);
}
```

Figura 74. Función sendMessage()

Dos de los cuales son enviados a la función getDataApi de la clase ApiDataFactory, en esta se crea un arreglo, donde llegan y se asignan las variables:

- \$SenderId
- \$ApiKey
- \$ClientId
- \$message
- \$numbers

```
public function getDataApi($numbers,$message): array {
    $dato['SenderId']= "SMS";
    $dato['ApiKey']= $_SESSION['ApiKeySms'];
    $dato['ClientId']= $_SESSION['clientIdSms'];
    $dato['Message']= $message;
    $dato['MobileNumbers']= $numbers;
    return $dato;
}
```

Figura 75. Función getDataApi()

Este arreglo es asignado a la variable `$dataApi` y enviado a la función `getSMS()` de la clase `ApiTechnetSMS`, la cual contiene un Try-catch utilizado para el control de errores, en el try se abre la petición a la API, donde se crean las cabeceras o headers como lo indicaba la documentación de la API ('Content-Type' => 'application/json'), por otro lado, se especifica el método y la ruta por los cuales se realizará la conexión y en el body es enviado el arreglo de la función `getDataApi()` en formato json.

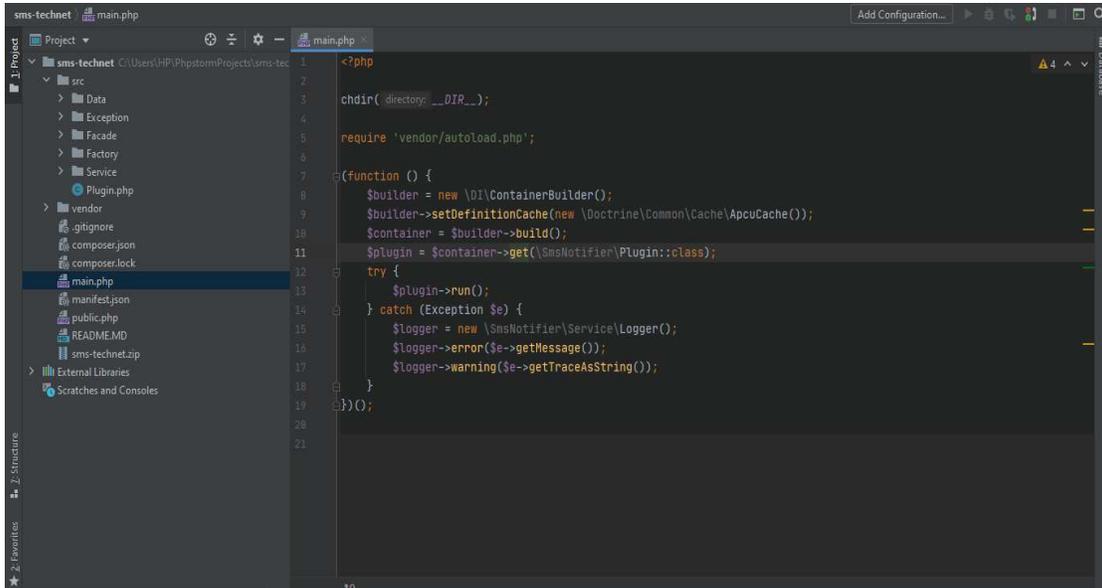
```
public function getSMS($data){  
  
    try {  
        $this->httpClient = HttpClient::create(['headers' => [  
            'User-Agent' => 'PHP console app',  
            'Content-Type' => 'application/json',  
        ]]);  
        $this->response = $this->httpClient->request(  
            method: 'POST',  
            url: 'http://198.136.58.34:6005/api/v2/SendSMS', ['body'=> json_encode($data)];  
        );  
    } catch (TransportExceptionInterface $e) {  
        var_export($e);  
    }  
}
```

Figura 76. Función `getSMS()`

Estructura de archivos

A continuación, se muestra la estructura de archivos del plugin de mensajería desarrollado para TechNet Comunicaciones S.A.S.

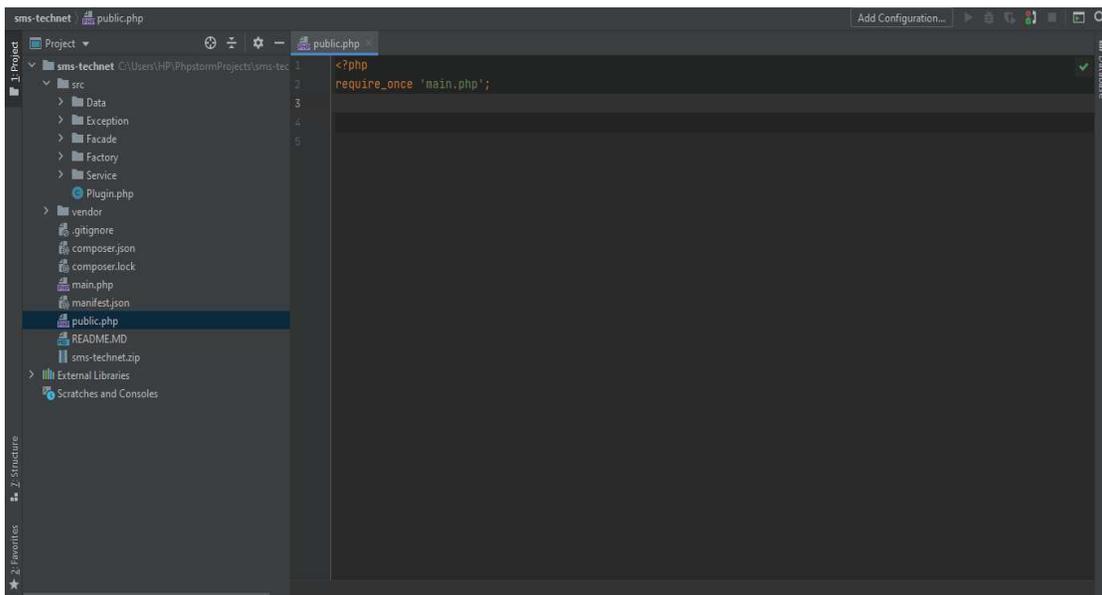
- main.php: Archivo principal del plugin, es ejecutado cuando el complemento se ejecuta en el CRM.



```
<?php
1
2
3 chdir( __DIR__ );
4
5 require 'vendor/autoload.php';
6
7
8 function () {
9     $builder = new \DI\ContainerBuilder();
10    $builder->setDefinitionCache(new \Doctrine\Common\Cache\ApcuCache());
11    $container = $builder->build();
12    $plugin = $container->get(\SmsNotifier\Plugin::class);
13    try {
14        $plugin->run();
15    } catch (Exception $e) {
16        $logger = new \SmsNotifier\Service\Logger();
17        $logger->error($e->getMessage());
18        $logger->warning($e->getTraceAsString());
19    }
20 }();
21
```

Figura 77. main.php

- public.php: Este archivo es el que genera la URL pública para el plugin.



```
<?php
1
2 require_once 'main.php';
3
4
5
```

Figura 78. public.php

- manifest.json: Contiene información sobre el complemento (versión, nombre, descripción, autor, configuración requerida, etc).

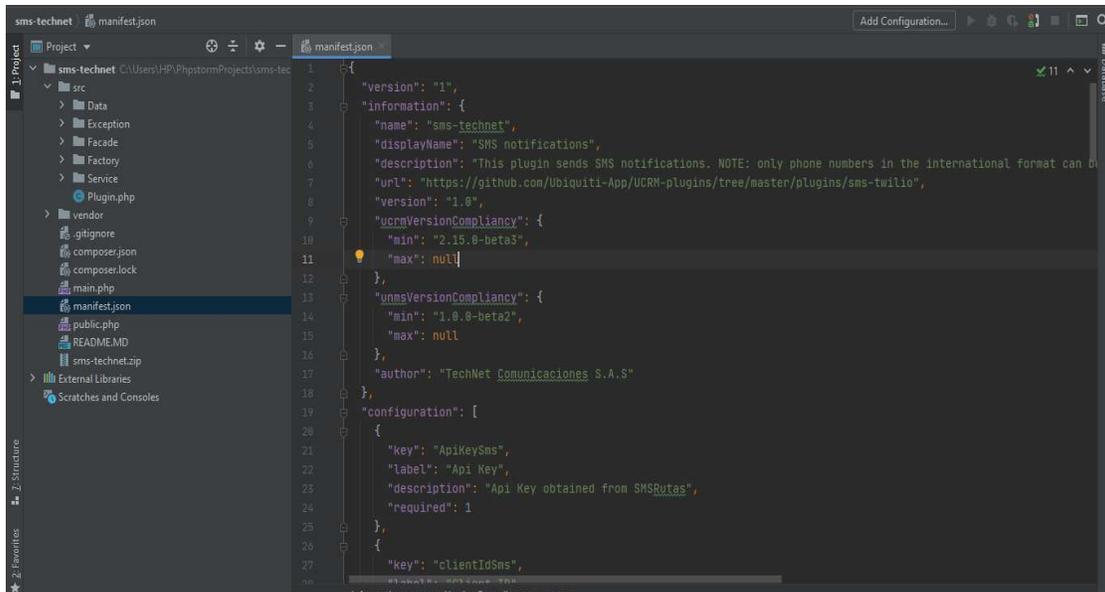


Figura 79. manifest.json

- composer.json: Describe todas las dependencias del proyecto que serán instaladas posteriormente a través de Composer.

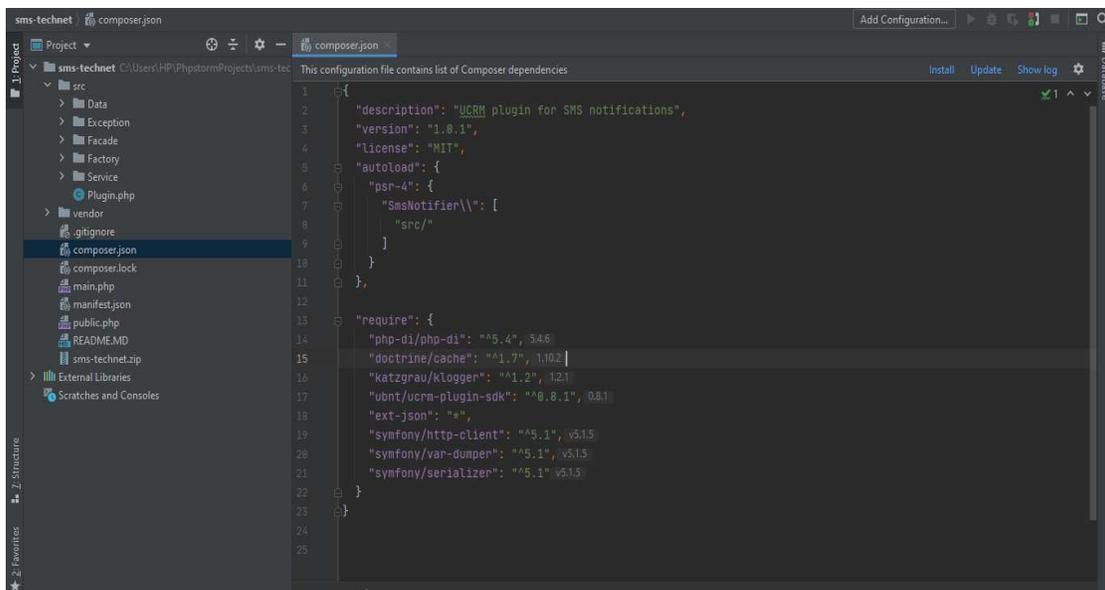


Figura 80. composer.json

- ApiData.php: Este archivo contiene los datos rellenos automáticamente por la API de SMSrutas.

```

1 declare(strict_types=1);
2
3 namespace SmsNotifier\Data;
4
5 /**
6  * automatically filled data for API DATA
7  */
8
9 class ApiData
10 {
11     /**
12      * @var string
13      */
14     public $SenderId;
15
16     /**
17      * @var boolean|null
18      */
19     public $Is_Unicode;
20
21     /**
22      * @var boolean|null
23      */
24     public $Is_Flash;
25 }

```

Figura 83. ApiData.php

- NotificationData.php src\Data: En este archivo se encuentran los datos recibidos en Webhook.

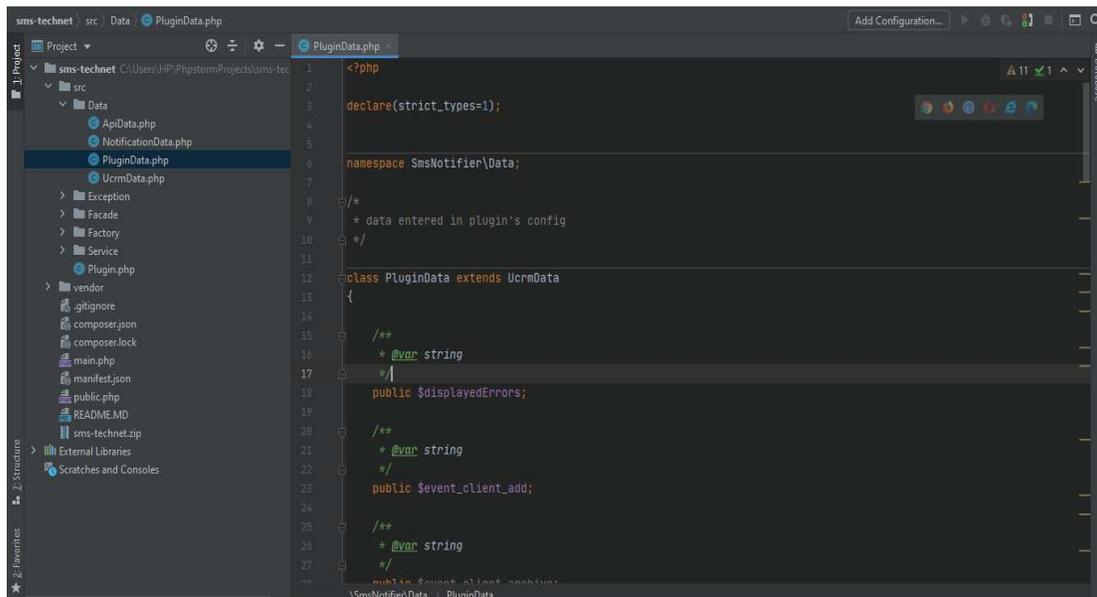
```

1 declare(strict_types=1);
2
3 namespace SmsNotifier\Data;
4
5 /**
6  * data received in webhook, plus details loaded from UCRM API
7  */
8
9 class NotificationData
10 {
11     /** @var string */
12     public $Uuid;
13
14     /** @var string */
15     public $changeType;
16
17     /** @var string */
18     public $entity;
19
20     /** @var int|null */
21     public $entityId;
22
23     /** @var int|null */
24     public $clientId;
25
26     /** @var string */
27     public $entity;
28 }

```

Figura 84. NotificationData.php

- PluginData.php src\Data: Contiene los datos ingresados en la configuración del plugin.



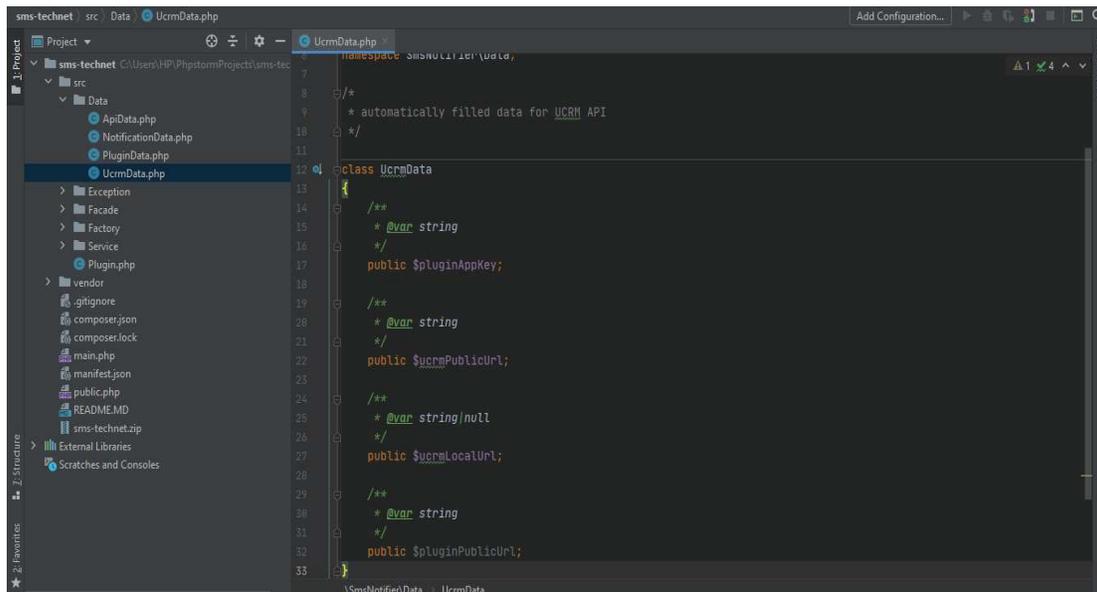
```

1 <?php
2
3 declare(strict_types=1);
4
5
6 namespace SmsNotifier\Data;
7
8 /**
9  * data entered in plugin's config
10  */
11
12 class PluginData extends UcrmData
13 {
14
15     /**
16      * @var string
17      */
18     public $displayedErrors;
19
20     /**
21      * @var string
22      */
23     public $event_client_add;
24
25     /**
26      * @var string
27      */
28     public $event_client_archive;
29 }

```

Figura 85. PluginData.php

- UcrmData.php src\Data: En este se encuentran los datos rellenos automáticamente por la API de UCRM.



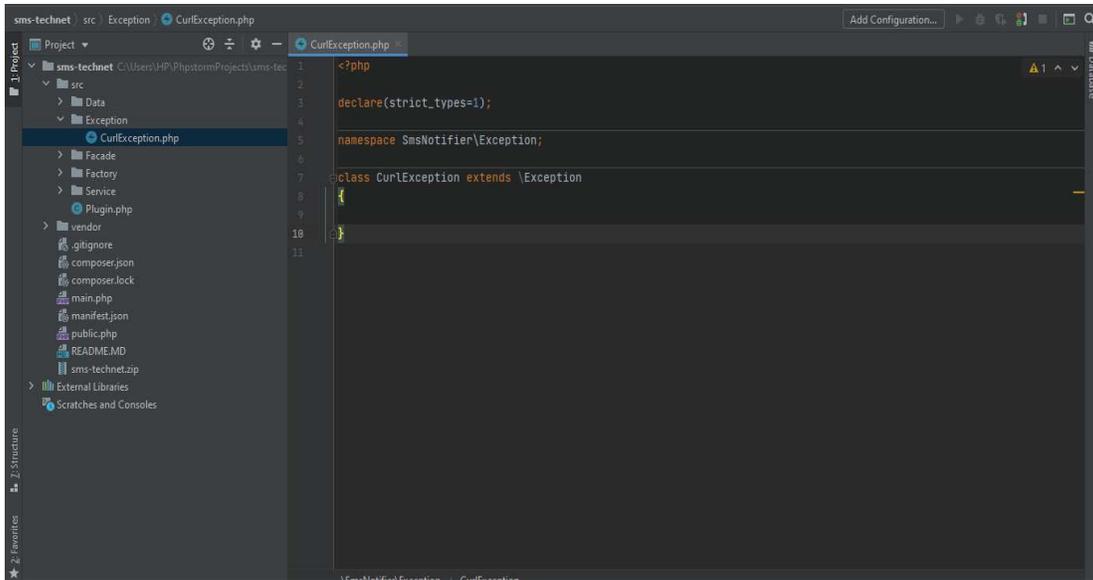
```

1 namespace SmsNotifier\Data;
2
3 /**
4  * automatically filled data for UCRM API
5  */
6
7 class UcrmData
8 {
9     /**
10      * @var string
11      */
12     public $pluginAppKey;
13
14     /**
15      * @var string
16      */
17     public $ucrmPublicUrl;
18
19     /**
20      * @var string|null
21      */
22     public $ucrmLocalUrl;
23
24     /**
25      * @var string
26      */
27     public $pluginPublicUrl;
28 }

```

Figura 86. UcrmData.php

- CurlException.php src\Exception: Este archivo extiende de la clase “Exception”, la cual es la clase base para todas las excepciones.



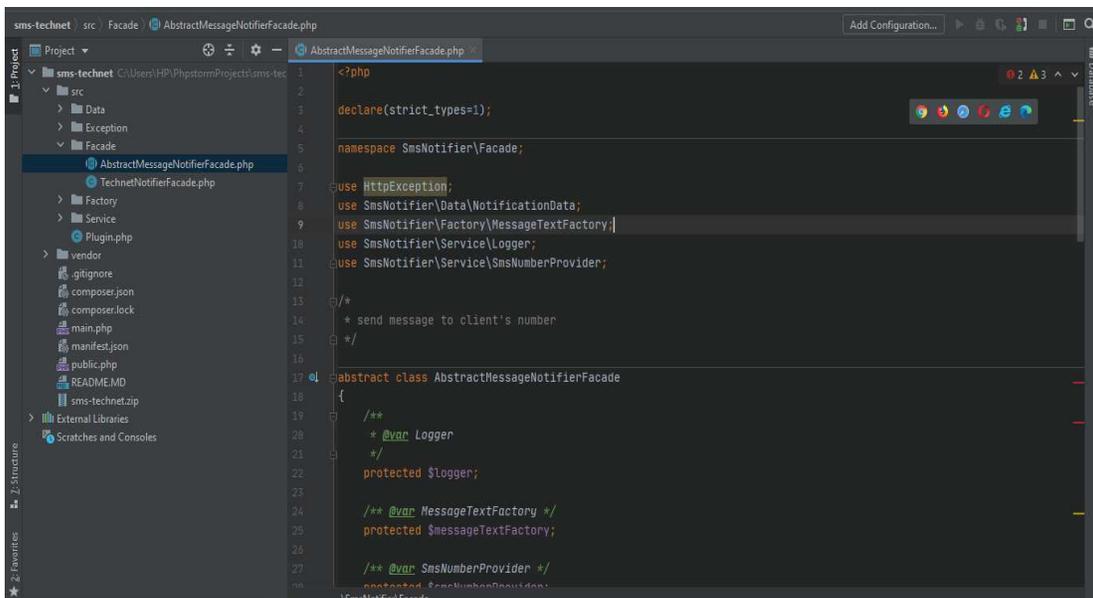
```

1 <?php
2
3 declare(strict_types=1);
4
5 namespace SmsNotifier\Exception;
6
7 class CurlException extends \Exception
8 {
9 }
10
11

```

Figura 87. CurlException.php

- AbstractMessageNotifierFacade.php src\Facade: Son configurados los números de teléfono y el cuerpo del mensaje a enviar.



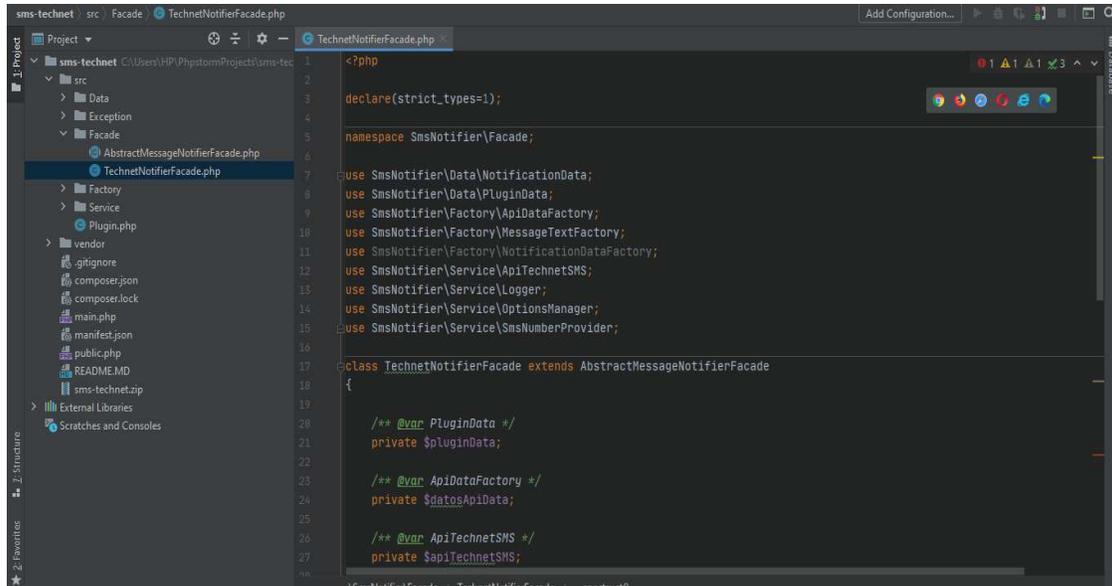
```

1 <?php
2
3 declare(strict_types=1);
4
5 namespace SmsNotifier\Facade;
6
7 use HttpException;
8 use SmsNotifier\Data\NotificationData;
9 use SmsNotifier\Factory\MessageTextFactory;
10 use SmsNotifier\Service\Logger;
11 use SmsNotifier\Service\SmsNumberProvider;
12
13 /**
14  * send message to client's number
15  */
16
17 abstract class AbstractMessageNotifierFacade
18 {
19     /**
20      * @var Logger
21      */
22     protected $logger;
23
24     /** @var MessageTextFactory */
25     protected $messageTextFactory;
26
27     /** @var SmsNumberProvider */
28     protected $smsNumberProvider;
29
30 }

```

Figura 88. AbstractMessageNotifierFacade.php

- TechnetNotifierFacade.php src\Facade: Extiende de AbstractMessageNotifierFacade, en este archivo se especifica el proveedor de mensajería específico (SMSrutas).



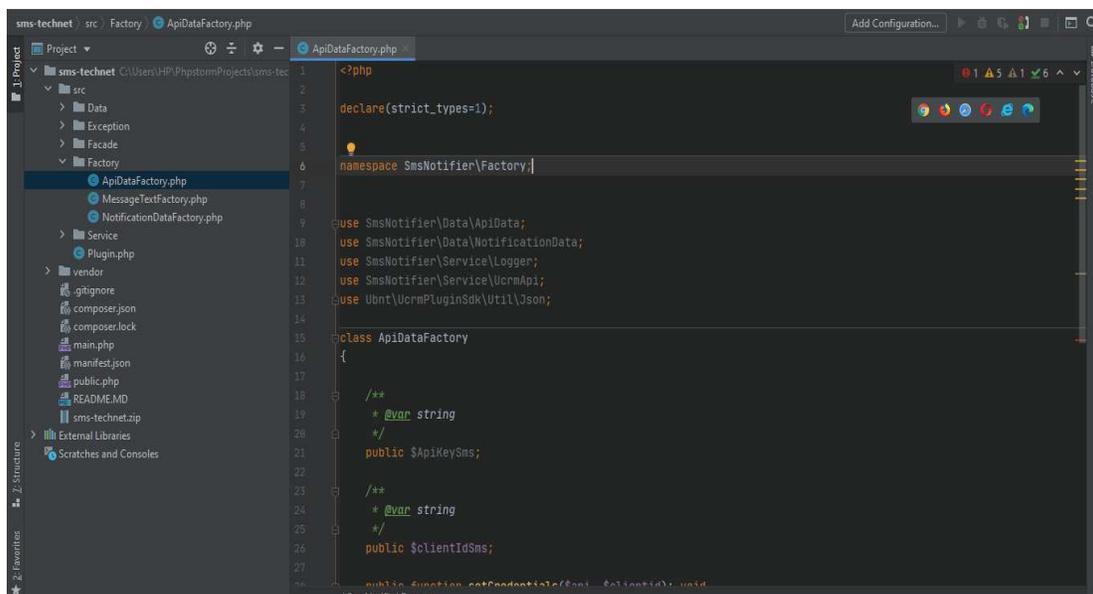
```

1 <?php
2
3 declare(strict_types=1);
4
5 namespace SmsNotifier\Facade;
6
7 use SmsNotifier\Data\NotificationData;
8 use SmsNotifier\Data\PluginData;
9 use SmsNotifier\Factory\ApiDataFactory;
10 use SmsNotifier\Factory\MessageTextFactory;
11 use SmsNotifier\Factory\NotificationDataFactory;
12 use SmsNotifier\Service\ApiTechnetSMS;
13 use SmsNotifier\Service\Logger;
14 use SmsNotifier\Service\OptionsManager;
15 use SmsNotifier\Service\SmsNumberProvider;
16
17 class TechnetNotifierFacade extends AbstractMessageNotifierFacade
18 {
19     /** @var PluginData */
20     private $pluginData;
21
22     /** @var ApiDataFactory */
23     private $datosApiData;
24
25     /** @var ApiTechnetSMS */
26     private $apiTechnetSMS;
27
28     public function __construct()
29     {
30     }
31 }

```

Figura 89. TechnetNotifierFacade.php

- ApiDataFactory.php src\Factory: En este archivo se encuentra ubicada la función getDataApi(), la cual recibe y asigna las variables de la API de SMSrutas.



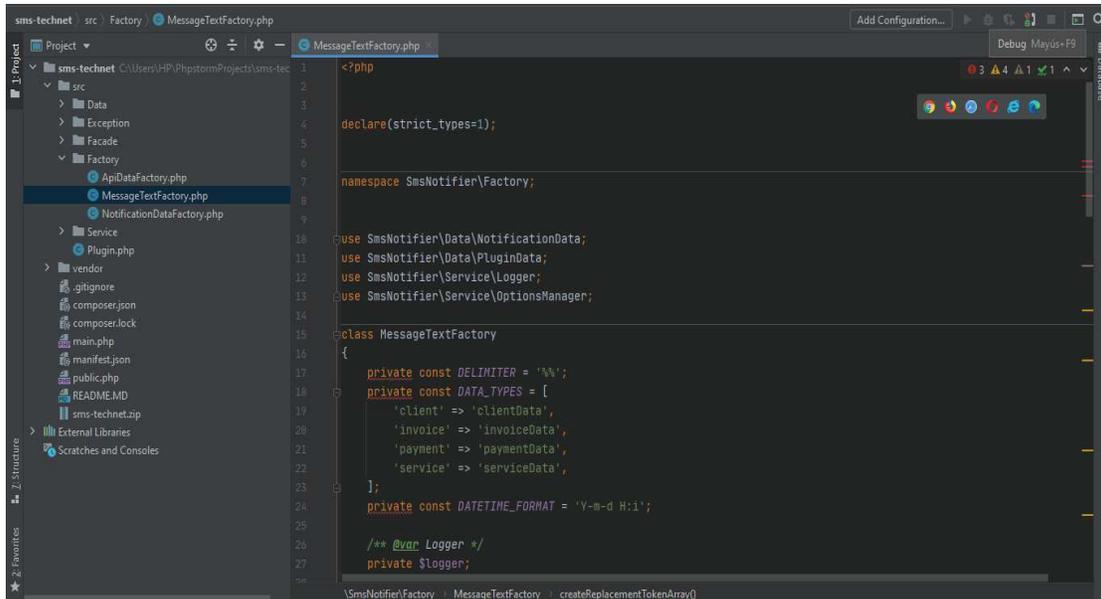
```

1 <?php
2
3 declare(strict_types=1);
4
5 namespace SmsNotifier\Factory;
6
7 use SmsNotifier\Data\ApiData;
8 use SmsNotifier\Data\NotificationData;
9 use SmsNotifier\Service\Logger;
10 use SmsNotifier\Service\UcrnApi;
11 use Ubn\UcrnPluginSdk\Util\Json;
12
13 class ApiDataFactory
14 {
15     /**
16      * @var string
17      */
18     public $ApiKeySms;
19
20     /**
21      * @var string
22      */
23     public $ClientIdSms;
24
25     public function getDataApi($api_key, $client_id): void
26     {
27     }
28 }

```

Figura 90. ApiDataFactory.php

- MessageTextFactory.php src\Factory: Verifica si existe y está configurado un mensaje de texto para un evento específico.

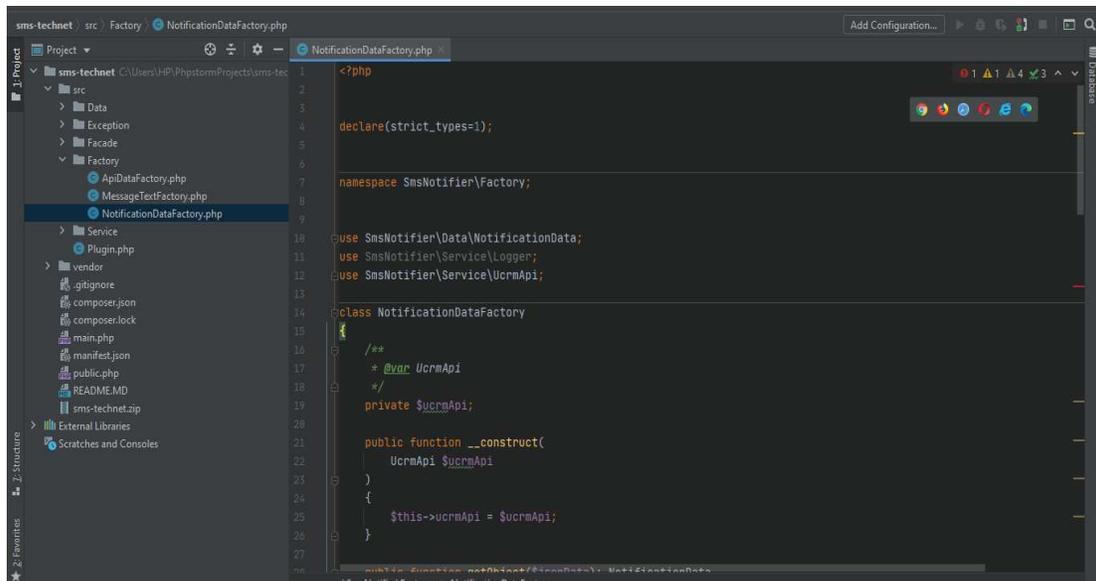


```

1 <?php
2
3
4 declare(strict_types=1);
5
6
7 namespace SmsNotifier\Factory;
8
9
10 use SmsNotifier\Data\NotificationData;
11 use SmsNotifier\Data\PluginData;
12 use SmsNotifier\Service\Logger;
13 use SmsNotifier\Service\OptionsManager;
14
15 class MessageTextFactory
16 {
17     private const DELIMITER = '\n';
18     private const DATA_TYPES = [
19         'client' => 'clientData',
20         'invoice' => 'invoiceData',
21         'payment' => 'paymentData',
22         'service' => 'serviceData',
23     ];
24     private const DATETIME_FORMAT = 'Y-m-d H:i';
25
26     /** @var Logger */
27     private $logger;
  
```

Figura 91. MessageTextFactory.php

- NotificationDataFactory.php src\Factory: En este archivo se reciben los datos del Webhook.



```

1 <?php
2
3
4 declare(strict_types=1);
5
6
7 namespace SmsNotifier\Factory;
8
9
10 use SmsNotifier\Data\NotificationData;
11 use SmsNotifier\Service\Logger;
12 use SmsNotifier\Service\UcrmApi;
13
14 class NotificationDataFactory
15 {
16     /**
17      * @var UcrmApi
18      */
19     private $ucrmApi;
20
21     public function __construct(
22         UcrmApi $ucrmApi
23     )
24     {
25         $this->ucrmApi = $ucrmApi;
26     }
27
  
```

Figura 92. NotificationDataFactory.php

- ApiTechnetSMS.php\Service: Este archivo contiene la función getSMS(), la cual realiza la petición a la API de SMSrutas.

```

1 <?php
2
3
4 namespace SmsNotifier\Service;
5
6
7 use SmsNotifier\Data\ApiData;
8 use SmsNotifier\Data\PluginData;
9 use Symfony\Component\HttpClient\HttpClient;
10 use SmsNotifier\Exception\CurlException;
11 use Symfony\Contracts\HttpClient\Exception\TransportExceptionInterface;
12 use UbntUcrmPluginSdk\Util\Json;
13
14 class ApiTechnetSMS
15 {
16     /**
17      * @var HttpClient
18      */
19     private $httpClient;
20
21     /**
22      * @var Json
23      */
24     private $response;
25
26
27     public function __construct()
28     {
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figura 93. ApiTechnetSMS.php

- CurlExecutor.php src\Service: Permite realizar peticiones HTTP con el objetivo de transferir información con sintaxis de URL al servidor remoto.

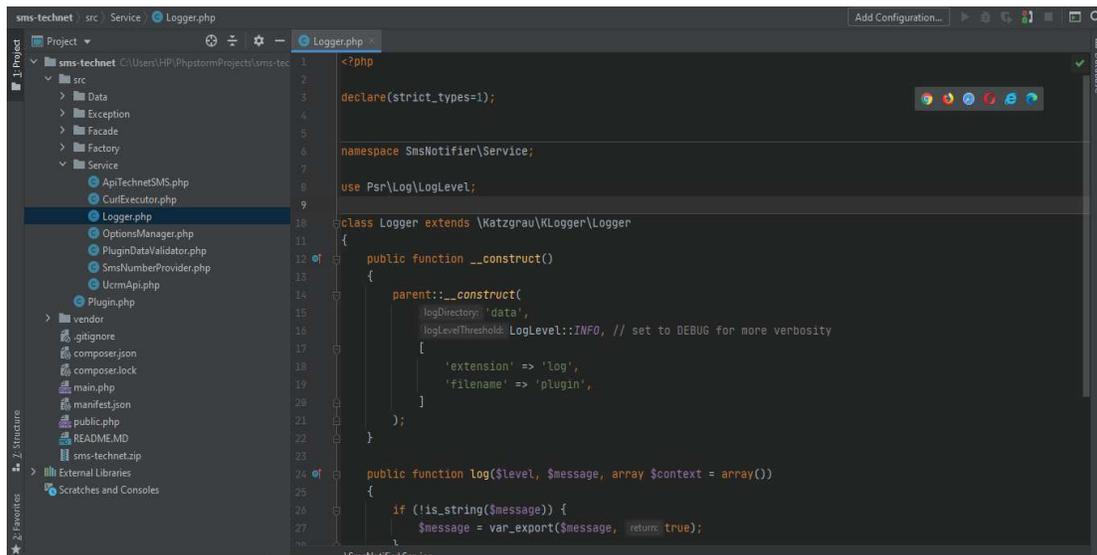
```

1 <?php
2
3 declare(strict_types=1);
4
5
6 namespace SmsNotifier\Service;
7
8
9 use SmsNotifier\Exception\CurlException;
10
11 class CurlExecutor
12 {
13     /**
14      * @throws CurlException
15      */
16     public function curlCommand($url, $method, array $headers = [], $data = null, bool $verifySsl = true): void
17     {
18         $c = curl_init();
19         curl_setopt($c, option: CURLOPT_URL, $url);
20         curl_setopt($c, option: CURLOPT_HTTPHEADER, $headers);
21         curl_setopt($c, option: CURLOPT_CUSTOMREQUEST, $method);
22
23         if ($data) {
24             curl_setopt($c, option: CURLOPT_POST, value: true);
25             curl_setopt($c, option: CURLOPT_POSTFIELDS, $data);
26         }
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figura 94. CurlExecutor.php

- `Logger.php` `src\Service`: Se configuran los niveles de registro, el cual se establece para advertir, lo que significa todo lo que se registra. Una vez que el registrador acepta un mensaje, se envía a la consola.



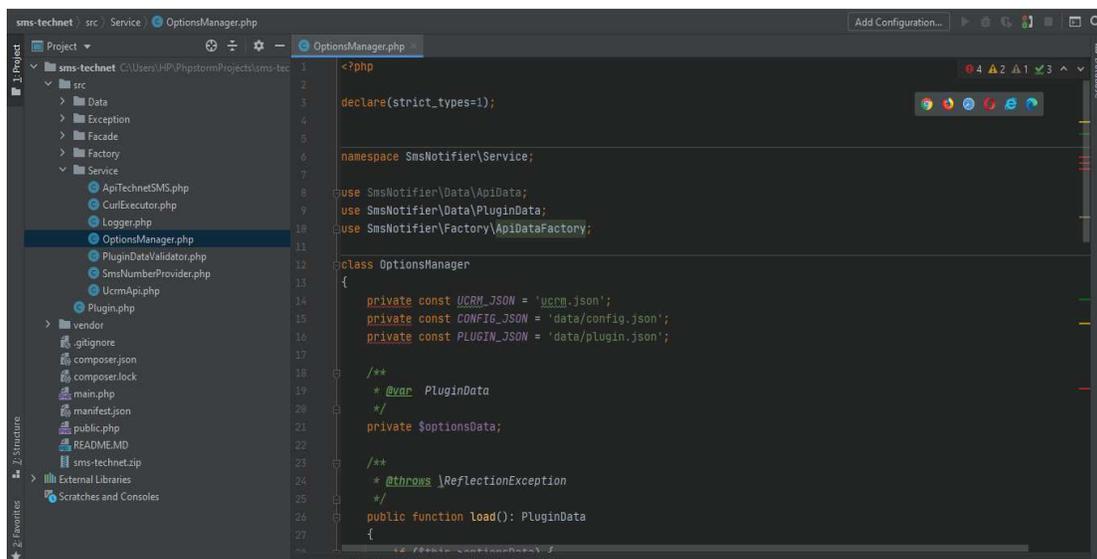
```

1 <?php
2
3 declare(strict_types=1);
4
5
6 namespace SmsNotifier\Service;
7
8 use Psr\Log\LogLevel;
9
10
11 class Logger extends \Katzgrau\KLogger\Logger
12 {
13     public function __construct()
14     {
15         parent::__construct(
16             logDirectory: 'data',
17             logLevelThreshold: LogLevel::INFO, // set to DEBUG for more verbosity
18             [
19                 'extension' => 'log',
20                 'filename' => 'plugin',
21             ]
22         );
23     }
24
25     public function log($level, $message, array $context = array())
26     {
27         if (!is_string($message)) {
28             $message = var_export($message, true);
29         }
30     }
31 }

```

Figura 95. `Logger.php`

- `OptionsManager.php` `src\Service`: En constantes privadas se almacenan `ucrm.json`, `config.json` y `plugin.json`, para posteriormente ser combinados en un `array_merge`.



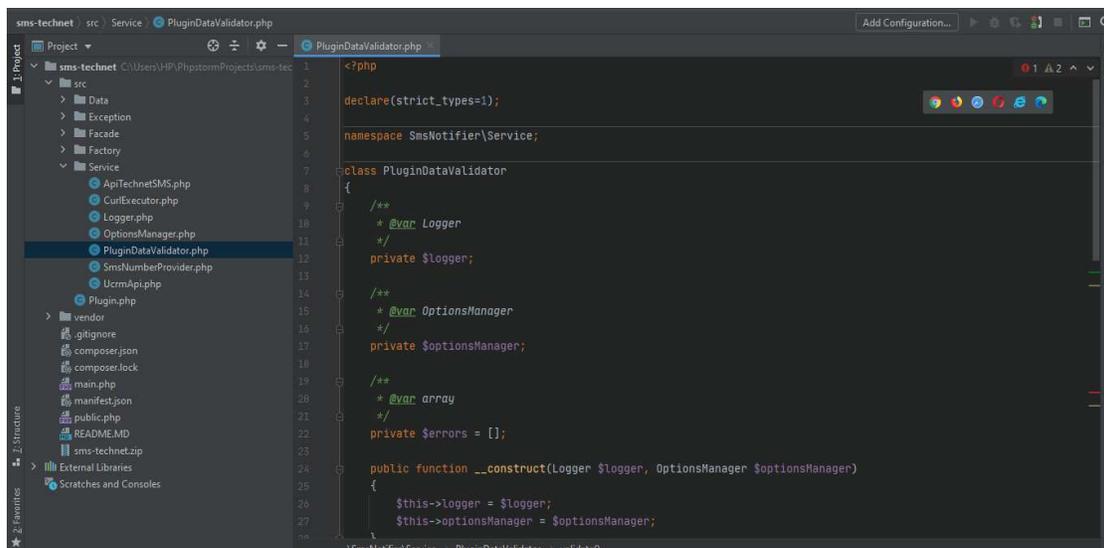
```

1 <?php
2
3 declare(strict_types=1);
4
5
6 namespace SmsNotifier\Service;
7
8 use SmsNotifier\Data\ApiData;
9 use SmsNotifier\Data\PluginData;
10 use SmsNotifier\Factory\ApiDataFactory;
11
12 class OptionsManager
13 {
14     private const UCRM_JSON = 'ucrm.json';
15     private const CONFIG_JSON = 'data/config.json';
16     private const PLUGIN_JSON = 'data/plugin.json';
17
18     /**
19      * @var PluginData
20      */
21     private $optionsData;
22
23     /**
24      * @throws \ReflectionException
25      */
26     public function load(): PluginData
27     {
28         // ...
29     }
30 }

```

Figura 96. `OptionsManager.php`

- PluginDataValidator.php src\Service: Este archivo cuenta con la función validate(), la cual como su nombre lo indica valida la Api Key y client ID de la API de SMSrutas.



```

<?php
declare(strict_types=1);

namespace SmsNotifier\Service;

class PluginDataValidator
{
    /**
     * @var Logger
     */
    private $logger;

    /**
     * @var OptionsManager
     */
    private $optionsManager;

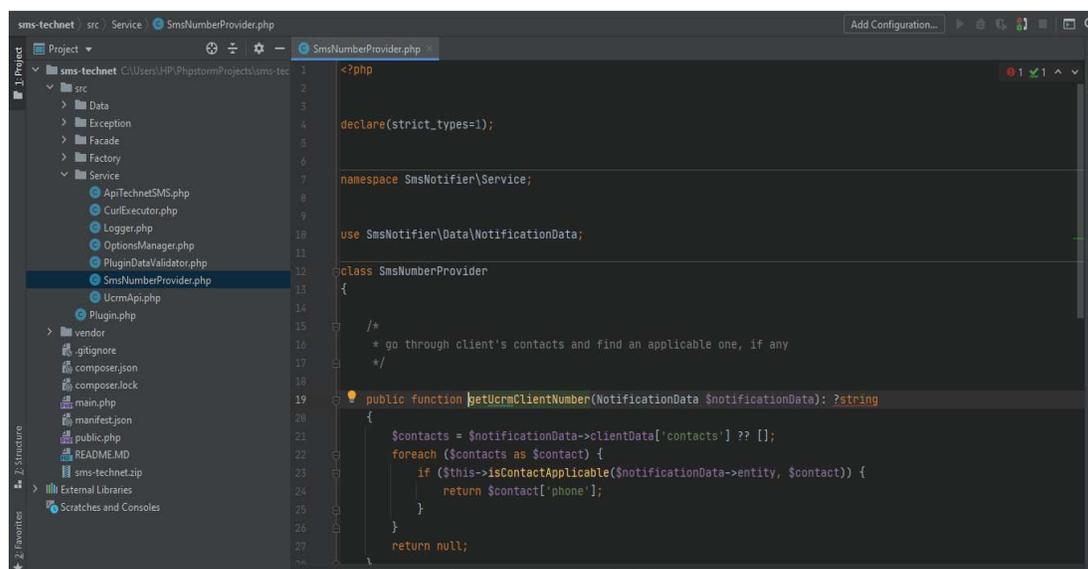
    /**
     * @var array
     */
    private $errors = [];

    public function __construct(Logger $logger, OptionsManager $optionsManager)
    {
        $this->logger = $logger;
        $this->optionsManager = $optionsManager;
    }
}

```

Figura 97. PluginDataValidator.php

- SmsNumberProvider.php src\Service: Se revisan los contactos del cliente y se busca uno que corresponda. No todos los contactos tienen el número de teléfono en sus datos; por eso en este archivo se comprueba si el tipo de notificación es aplicable al contacto.



```

<?php
declare(strict_types=1);

namespace SmsNotifier\Service;

use SmsNotifier\Data\NotificationData;

class SmsNumberProvider
{
    /**
     * go through client's contacts and find an applicable one, if any
     */

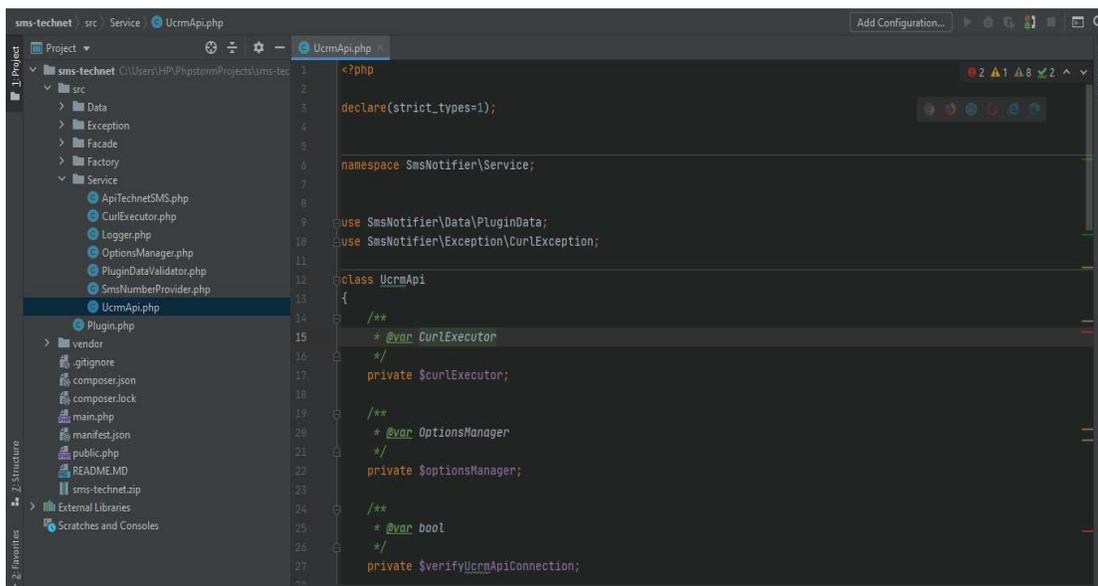
    public function getUcrmClientNumber(NotificationData $notificationData): ?string
    {
        $contacts = $notificationData->clientData['contacts'] ?? [];
        foreach ($contacts as $contact) {
            if ($this->isContactApplicable($notificationData->entity, $contact)) {
                return $contact['phone'];
            }
        }

        return null;
    }
}

```

Figura 98. SmsNumberProvider.php

- UcrmApi.php src\Service: En este archivo, se verifica la conexión de la API de UCRM.



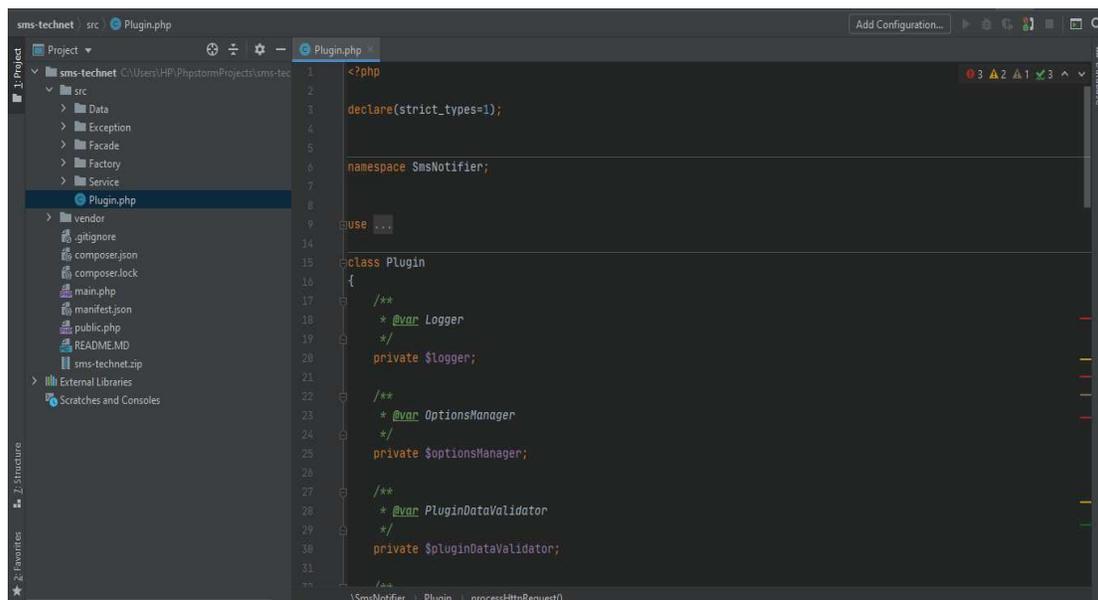
```

1 <?php
2
3 declare(strict_types=1);
4
5
6 namespace SmsNotifier\Service;
7
8
9 use SmsNotifier\Data\PluginData;
10 use SmsNotifier\Exception\CurlException;
11
12 class UcrmApi
13 {
14     /**
15      * @var CurlExecutor
16      */
17     private $curlExecutor;
18
19     /**
20      * @var OptionsManager
21      */
22     private $optionsManager;
23
24     /**
25      * @var bool
26      */
27     private $verifyUcrmApiConnection;
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figura 99. UcrmApi.php

- Plugin.php src: En la clase Plugin se encuentra la función principal, llamada run(), si todo se encuentra bien se inicia el complemento en UNMS.



```

1 <?php
2
3 declare(strict_types=1);
4
5
6 namespace SmsNotifier;
7
8
9 use SmsNotifier\Data\PluginData;
10 use SmsNotifier\Exception\CurlException;
11 use SmsNotifier\Service\UcrmApi;
12
13 class Plugin
14 {
15     /**
16      * @var Logger
17      */
18     private $logger;
19
20     /**
21      * @var OptionsManager
22      */
23     private $optionsManager;
24
25     /**
26      * @var PluginDataValidator
27      */
28     private $pluginDataValidator;
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figura 100. Plugin.php

3.1.3 Objetivo 3. Realizar las pruebas al plugin en la infraestructura tecnológica de la empresa.

- Actividad 1. Realizar las pruebas necesarias al plugin de mensajería en un sistema aislado.
- Actividad 2. Corregir los errores identificados con la aplicación de la prueba.

Es importante mencionar que las pruebas realizadas al plugin de mensajería se llevaron a cabo en un sistema aislado para prevenir posibles alteraciones en la infraestructura real de la empresa.

Inicialmente se cargó el plugin al CRM (Gestión de relaciones con los clientes) de UNMS (Sistema de gestión de red Ubiquiti).

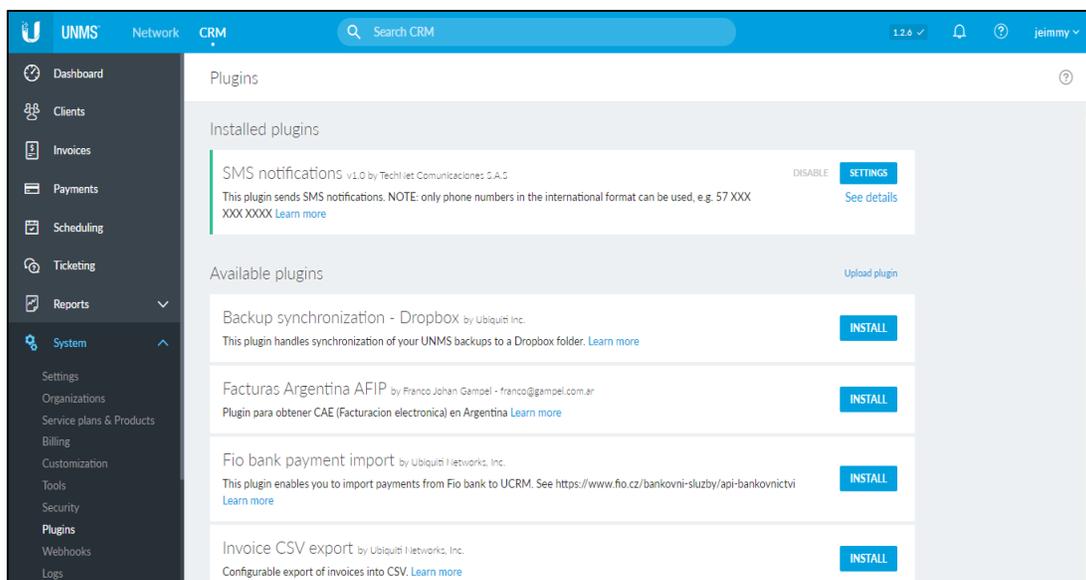


Figura 101. Plugin SMS en UNMS

Seguidamente, fueron rellenados los campos obligatorios (Api key, client ID, sender ID) con los datos que se obtienen en SMSRutas (Servidor de mensajería contratado) y los mensajes

cortos a enviar a los clientes de la empresa, para esta prueba se utilizaron dos tipos de situaciones, la primera en la que el cliente es modificado (Estimado(a) %%client.firstName ha sido editado%%) y la segunda en la que el cliente realiza el pago de su factura (Estimado(a) %%client.firstName%% el pago de su factura fue exitoso.). Se guardó la configuración y se habilitó el complemento.

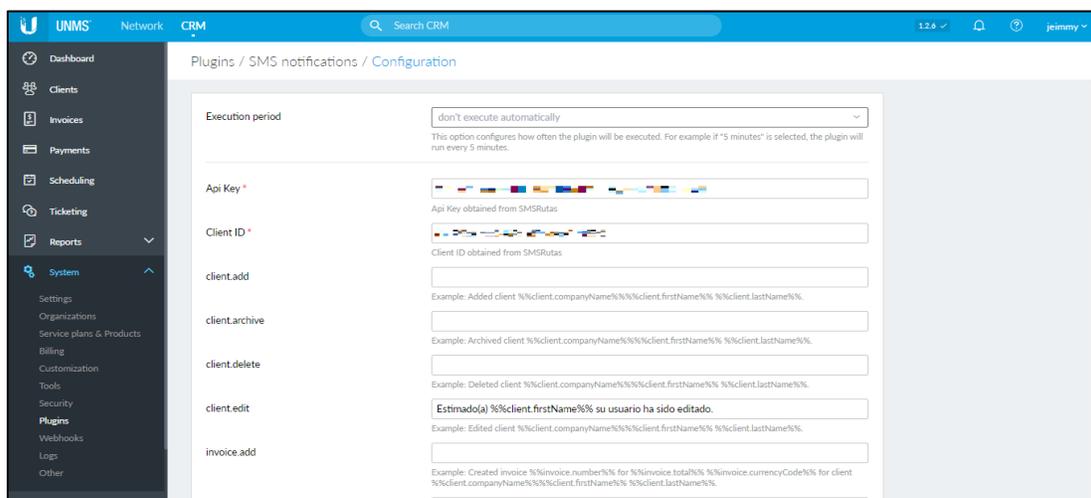


Figura 102. Configuración del plugin SMS

Para llevar a cabo la prueba fue necesario la creación de un usuario con el rol de cliente.

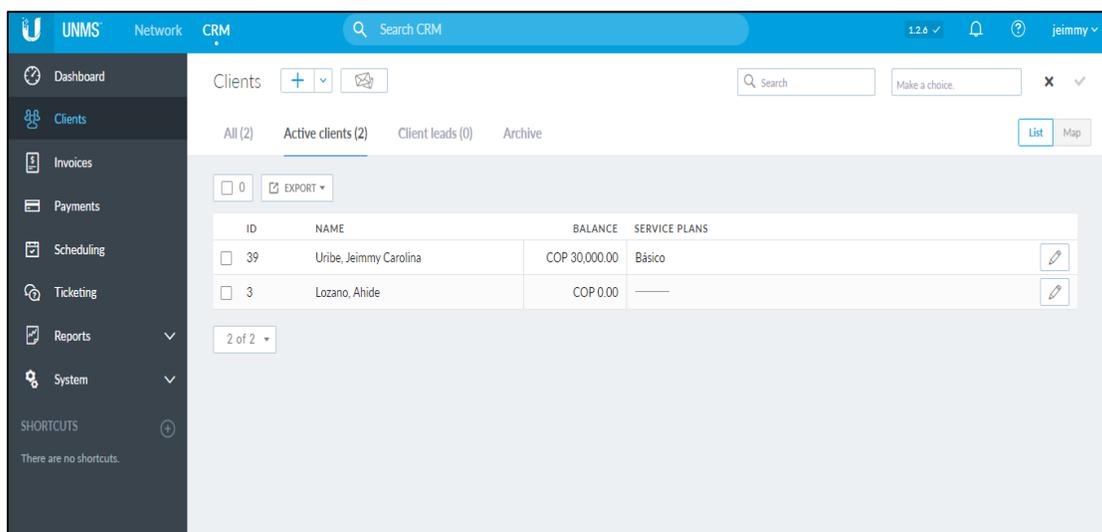
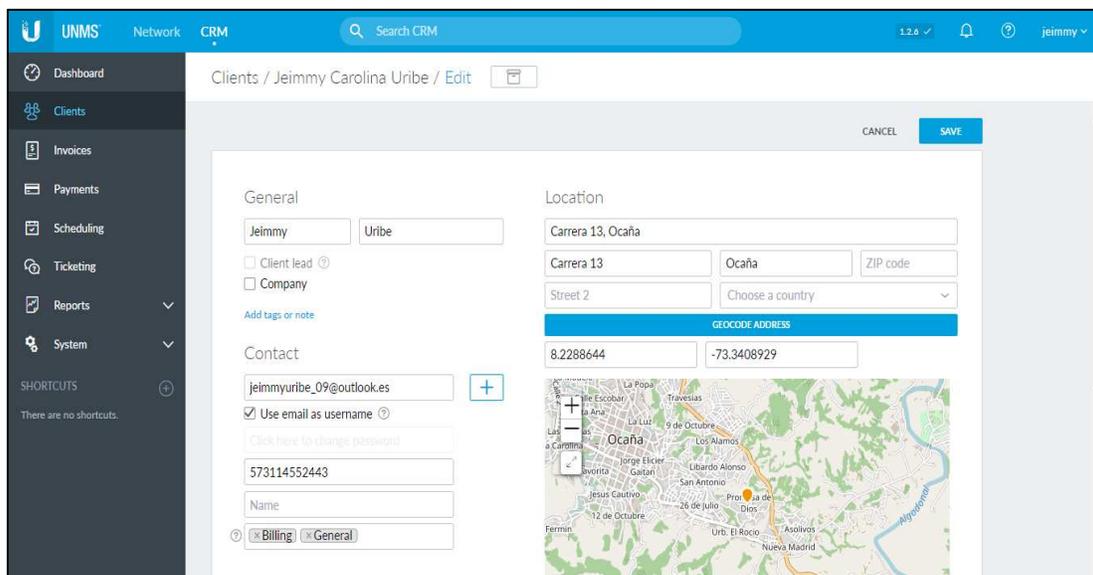


Figura 103. Creación del cliente

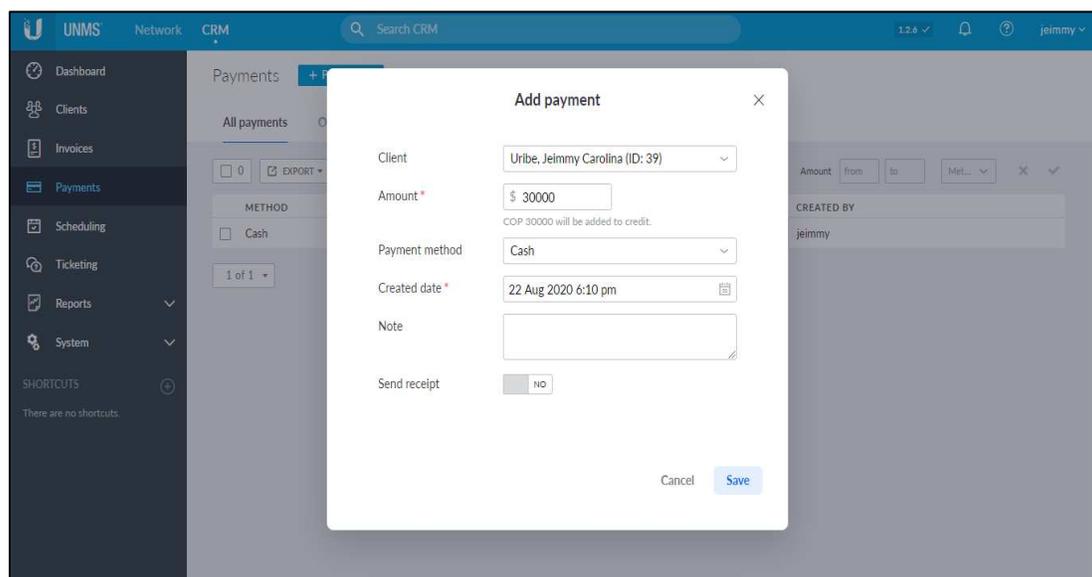
Luego de su creación, se hizo la respectiva modificación, en este caso en el nombre (“Jeimmy Carolina” a “Jeimmy”).



The screenshot displays the 'Edit' form for a client in the UNMS CRM system. The form is titled 'Clients / Jeimmy Carolina Uribe / Edit'. It features a sidebar with navigation options like Dashboard, Clients, Invoices, Payments, etc. The main form area is divided into three sections: 'General', 'Location', and 'Contact'. In the 'General' section, the name is 'Jeimmy' and the last name is 'Uribe'. There are checkboxes for 'Client lead' and 'Company'. The 'Location' section shows the address 'Carrera 13, Ocaña' and includes a map of the area. The 'Contact' section shows the email 'jeimmyuribe_09@outlook.es' and a phone number '573114552443'. There are 'CANCEL' and 'SAVE' buttons at the top right of the form.

Figura 104. Nombre de cliente modificado

Por otra parte, se agregó el pago a dicho cliente.



The screenshot shows the 'Add payment' modal form in the UNMS CRM system. The form is titled 'Add payment' and includes the following fields: 'Client' (Uribe, Jeimmy Carolina (ID: 39)), 'Amount' (\$ 30000), 'Payment method' (Cash), 'Created date' (22 Aug 2020 6:10 pm), and 'Send receipt' (NO). There are 'Cancel' and 'Save' buttons at the bottom of the modal.

Figura 105. Pago agregado

En el número del teléfono del cliente especificado se recibieron los siguientes mensajes.

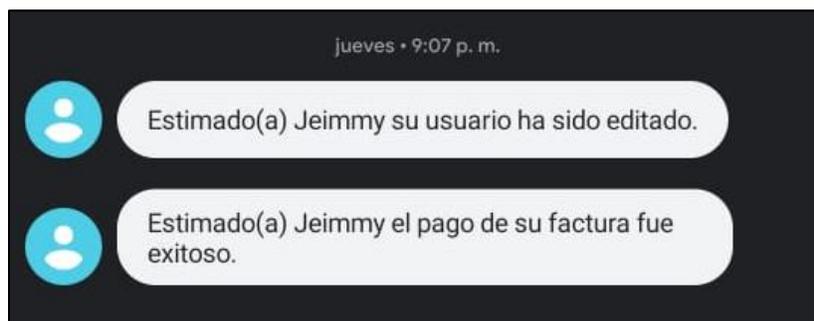
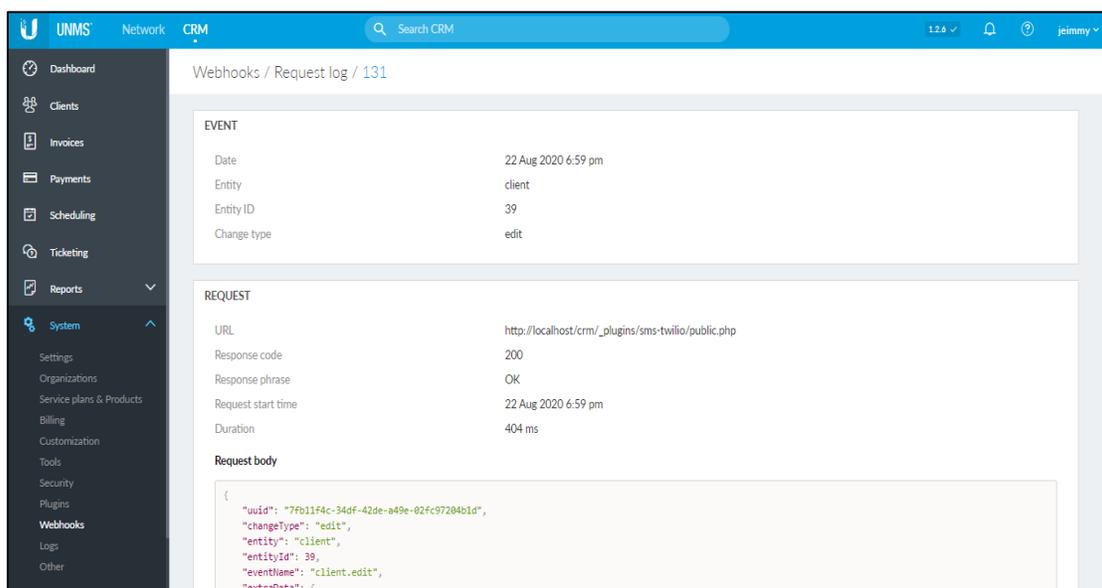


Figura 106. Mensajes cortos recibidos

En el registro de solicitudes (Request log) se muestra lo siguiente:



UNMS Network CRM Search CRM 1.2.0 jeimmy

Dashboard Clients Invoices Payments Scheduling Ticketing Reports System Settings Organizations Service plans & Products Billing Customization Tools Security Plugins Webhooks Logs Other

Webhooks / Request log / 131

EVENT

Date	22 Aug 2020 6:59 pm
Entity	client
Entity ID	39
Change type	edit

REQUEST

URL	http://localhost/crm/_plugins/sms-twilio/public.php
Response code	200
Response phrase	OK
Request start time	22 Aug 2020 6:59 pm
Duration	404 ms

Request body

```
{
  "uuid": "7fb11f4c-34df-42de-a49e-92fc97204b1d",
  "changeType": "edit",
  "entity": "client",
  "entityId": 39,
  "eventName": "client.edit",
  "extraData": {
```

Figura 107. Event/Request

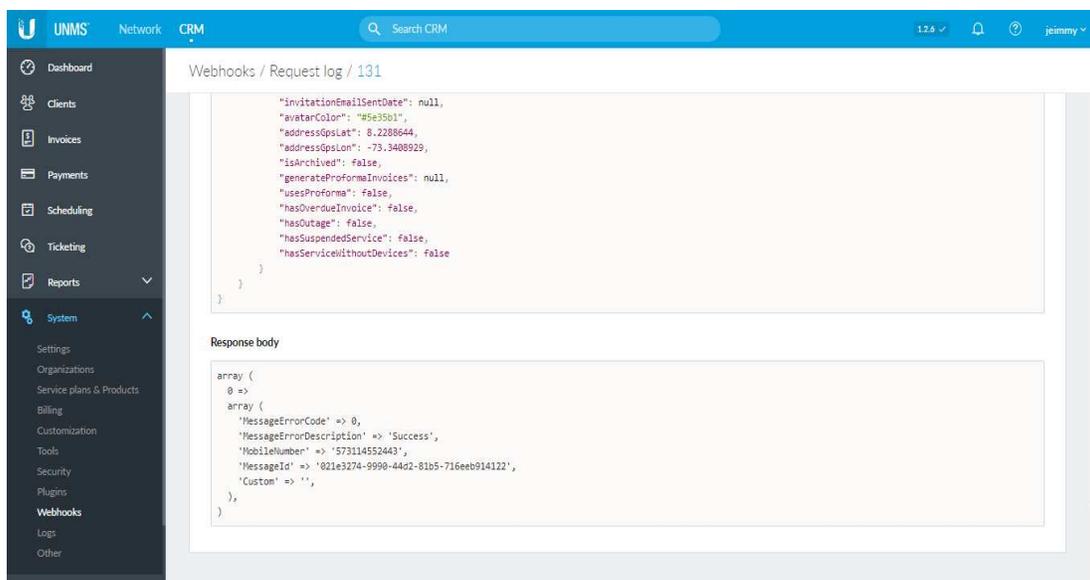


Figura 108. Response Body

3.1.4 Objetivo 4. Implementar el plugin de mensajería para el software CRM (Gestión de relaciones con los clientes) de TechNet Comunicaciones S.A.S.

- Actividad 1. Poner el funcionamiento el plugin de mensajería al software de la organización.
- Actividad 2. Llevar a cabo capacitaciones para el personal de la organización sobre el plugin de mensajería.

Para implementar el plugin de mensajería en el software de la empresa fue necesario la presencia del ingeniero de TechNet Comunicaciones S.A.S a cargo de UNMS. El plugin de mensajería se instaló y posteriormente se activó en el sistema, para esto fue necesario cargar el archivo .zip del complemento (Sistema > plugins). Seguidamente se personalizaron los mensajes de texto que se enviarían a los clientes de la organización cuando ocurre un evento en el sistema.

4. Diagnostico final

La evolución de TechNet Comunicaciones S.A.S ha permitido la creación de nuevas estrategias con el fin de mejorar la calidad de sus servicios, la implementación del plugin de mensajería en UNMS (Sistema de gestión de red de Ubiquiti) ha sido una de ellas.

Este complemento permite ofrecer excelente atención a sus clientes, ya que los mantiene informados del estado actual de sus facturas e información de soporte técnico vía SMS (Mensajes cortos), un medio accesible ya que la mayoría de los clientes cuenta con un teléfono celular y no se requiere de conexión a internet para recibir dicho mensaje.

5. Conclusiones

Los plugins han dejado gran impacto en los últimos tiempos, gracias a los avances tecnológicos se ha podido profundizar en esta área, han sido una herramienta esencial para ampliar las funciones de sistemas o plataformas, como es el caso de UNMS (Sistema de gestión de red de Ubiquiti), sistema utilizado por TechNet Comunicaciones S.A.S.

Dentro del desarrollo de este proyecto fue de vital importancia estudiar a fondo la definición, estructura y creación de los plugins, así como su incorporación en el software de la organización.

El desarrollo del plugin de mensajería como estrategia, fue imprescindible para mejorar la comunicación vía SMS con los clientes de TechNet Comunicaciones S.A.S, para llevarse a cabo previamente se preparó el entorno de desarrollo y se profundizó sobre el manejo del sistema de gestión de red de Ubiquiti.

En base a las pruebas realizadas para verificar el correcto funcionamiento del plugin fue pertinente realizarlas en un sistema aislado de la estructura tecnológica real de la empresa y así evitar posibles alteraciones en el sistema de TechNet Comunicaciones S.A.S.

Por último, una vez validado el plugin de mensajería a través de las pruebas, se verificó el correcto funcionamiento del mismo y seguidamente fue instalado en la infraestructura tecnológica real de la organización.

6. Recomendaciones

A TechNet Comunicaciones S.A.S:

Se recomienda que continúe desarrollando nuevas estrategias e implementando herramientas que mejoren su estado actual para satisfacer las necesidades del mercado y de sus clientes y así seguir ofreciendo un servicio de calidad.

De la experiencia recogida en este proyecto, es recomendable igualmente aprovechar al máximo el software utilizado por la organización con la idea de emprender mejoras al plugin de mensajería implementado o incluir nuevos complementos en un futuro próximo para integrarlos a UNMS (Sistema de gestión de red Ubiquiti) y así este sistema cada vez adquiera más funciones, agilizando el trabajo y calidad del servicio prestado por TechNet Comunicaciones S.A.S a los ocañeros.

7. Referencias

- Álvarez, D., Padilla, J., Garzón, A., & Muñoz, L. (2009). Proveedores de servicios de Internet y de contenidos, responsabilidad civil y derechos de autor. *Studiositas*, 54.
- Bustio, L., Coma, Y., & Talavera, I. (2013). Arquitectura basada en plugins para el desarrollo de software científico. *II Conferencia Internacional de Ciencias Computacionales e Informáticas, CICCI 2013* (pág. 3). La Habana: 15 Ediciones pensando en el futuro.
- Climent, A., & Guillem, M. (03 de Diciembre de 2017). *Algoritmo de reconocimiento de patrones basado en codificación fisiológica en cerebro de primates*. Obtenido de Universidad Politecnica de Valencia:
<https://riunet.upv.es/bitstream/handle/10251/92671/CASTEL%20-%20Algoritmo%20de%20reconocimiento%20de%20patrones%20basado%20en%20codificaci%C3%B3n%20fisiol%C3%B3gica%20en%20cerebro%20de....pdf?sequence=1>
- Composer. (s.f.). *Composer*. Obtenido de A Dependency Manager for PHP:
<https://getcomposer.org/>
- Composer. (s.f.). *Get Composer*. Obtenido de Introducción: <https://getcomposer.org/doc/00-intro.md>
- Dirección Nacional de Derecho de Autor. (s.f.). *Mininterior - Gobierno de Colombia*. Obtenido de Decisión Andina 351 de 1993: <http://derechodeautor.gov.co/decision-andina>
- EducareCorp. (s.f.). *EducareCorp Sector Empresarial*. Obtenido de Infraestructura tecnológica:
<https://www.grupoeducare.com/dcespresarial/index.php/2-principal/38-infraestructura-tecnologica#:~:text=%C2%BFQu%C3%A9%20es%20la%20infraestructura%20tecnol%C3%B3gica,investigaci%C3%B3n%20administraci%C3%B3n%20o%20gesti%C3%B3n%20interna>.
- Exner, O., & Procházka, J. (02 de Octubre de 2019). *GitHub - Ubiquiti-App*. Obtenido de Complementos UCRM- Cómo crear tu propio complemento UCRM:
<https://github.com/Ubiquiti-App/UCRM-plugins/blob/master/docs/tutorials/first-plugin.md#how-to-create-your-first-ucrm-plugin>
- Exner, O., Tousek, J., & Vavra, P. (30 de Octubre de 2019). *GitHub - Ubiquiti-App*. Obtenido de Complementos UCRM - Estructura de archivo: <https://github.com/Ubiquiti-App/UCRM-plugins/blob/master/docs/file-structure.md>
- Función Pública Gobierno de Colombia. (s.f.). *Gestor Normativo*. Obtenido de Ley 170 de 1994:
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=37805>

- García, G. A. (20 de Octubre de 2018). *Aspectos jurídicos del software libre en Colombia*. Obtenido de <http://revistavirtual.ucn.edu.co/index.php/RevistaUCN/article/viewFile/158/303>
- git-scm. (s.f.). *git --local-braching-on-the-cheap*. Obtenido de Downloading Git: <https://git-scm.com/download/win>
- Gómez, S. (s.f.). *Aula Software Libre*. Obtenido de Introducción a Git y Github - Día 1: <https://www.uco.es/aulasoftwarelibre/wp-content/uploads/2015/11/git-cosfera-dia-1.pdf>
- Gómez, S. (s.f.). *Aula software libre, Universidad de Córdoba*. Obtenido de Introducción a Git y GitHub -Día 1: <https://www.uco.es/aulasoftwarelibre/wp-content/uploads/2015/11/git-cosfera-dia-1.pdf>
- NeoAttack. (s.f.). *NeoAttack*. Obtenido de Plugin: <https://neoattack.com/neowiki/plugin/>
- php.net. (s.f.). *PHP: Hypertext Preprocessor*. Obtenido de PHP para Windows: <https://windows.php.net/download/>
- php.net. (s.f.). *Php-Manual*. Obtenido de ¿Qué es php?: <https://www.php.net/manual/es/intro-what-is.php>
- Sistema de Información sobre Comercio Exterior. (s.f.). *SICE*. Obtenido de Decisión 351, Régimen común sobre derecho de autor y derechos conexos: <http://www.sice.oas.org/trade/junac/decisiones/Dec351s.asp>
- Suin - Juriscol. (s.f.). *Sistema único de Información Normativa - Minjusticia*. Obtenido de Ley 565 de 2000: <http://www.suin-juriscol.gov.co/viewDocument.asp?ruta=Leyes/1662698>
- TechNet Comunicaciones S.A.S. (s.f.). TechNet Comunicaciones S.A.S. Ocaña, Norte de Santander, Colombia.
- Ubiquiti. (s.f.). *Ayuda de Ubiquiti*. Obtenido de UNMS - Sistema de gestión de red Ubiquiti: <https://help.ui.com/hc/en-us/articles/115012195187-UNMS-Getting-Started>
- Ubiquiti. (s.f.). *Ayuda de Ubiquiti*. Obtenido de UNMS-CREM: Complementos: <https://help.ui.com/hc/en-us/articles/360002433113-UCRM-Plugins>
- Ubiquiti. (s.f.). *Ayuda de Ubiquiti*. Obtenido de UNMS - Introducción: <https://help.ui.com/hc/en-us/articles/360020400574-UNMS-v1-Introduction>
- WifiSafe-Distribuidor WiFi. (s.f.). *Distribuidor WiFi*. Obtenido de Configuración UNMS de UBIQUITI: <https://www.wifisafe.com/blog/configuracion-unms-de-ubiquiti/>

WNI México. (s.f.). *WNI México*. Obtenido de Ubiquiti:

https://wni.mx/index.php?option=com_virtuemart&page=shop.browse&manufacturer_id=2