



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

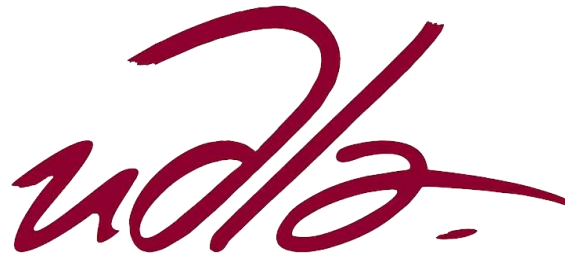
SISTEMA WEB DE CONTROL INTERNO BASADO EN WORDPRESS E
IMPLEMENTACIÓN DE UNA SOLUCIÓN DE BI PARA PYMES

AUTOR

Marco David Escobar Aleaga

AÑO

2020



FACULTAD DE INGENIERIAS Y CIENCIAS APLICADAS

SISTEMA WEB DE CONTROL INTERNO BASADO EN WORDPRESS E
IMPLEMENTACIÓN DE UNA SOLUCIÓN DE BI PARA PYMES

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniero en Sistemas de
Computación e Informática.

Profesor Guía

MSc. Pedro Manuel Nogales Cobas

Autor

Marco David Escobar Aleaga

Año

2020

DECLARACIÓN DEL PROFESOR GUÍA

"Declaro haber dirigido el trabajo, Sistema web de control interno basado en WordPress e implementación de una solución de BI para PYMEs, a través de reuniones periódicas con el estudiante Marco David Escobar Aleaga, en el semestre 202010, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".



Pedro Manuel Nogales Cobas

Máster en Gestión de Proyectos Informáticos

C.I. 1756760284

DECLARACIÓN DEL PROFESOR CORRECTOR

"Declaro haber revisado este trabajo, Sistema web de control interno basado en WordPress e implementación de una solución de BI para PYMEs, de Marco David Escobar Aleaga, en el semestre 202010, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".



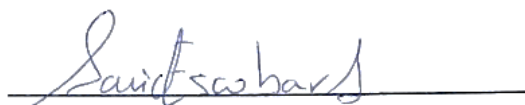
Eddy Mauricio Armas Pallasco

Magister en Gerencia de Sistemas y Tecnologías de Información

C.I. 1711715803

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

"Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes."



Marco David Escobar Aleaga

C.I: 1715925614

AGRADECIMIENTOS

A Dios, a mi familia, amigos y profesores que apoyaron y guiaron mi formación generando pasión por mis estudios. Un especial agradecimiento a José Manuel Rodríguez Padrino por ser un amigo y mentor que compartió gran parte de su conocimiento conmigo.

DEDICATORIA

A mis queridos papá y mamá por el apoyo incondicional en mi crecimiento personal y profesional; por los valores transmitidos y el afecto entregado.

Resumen

Considerando la importancia de la implementación de sistemas de control interno dentro de las PYMEs este proyecto de titulación busca el desarrollo completo y modular de diferentes requerimientos solicitados por A Plus Computer implementando *plugins* personalizados en el CMS *WordPress*, posteriormente se presenta información de valor para la empresa en un panel de *Business Intelligence* que ayudará a la toma de decisiones.

El desarrollo completo está basado en la metodología Scrum, que principalmente guía la gestión de proyectos ágiles a través de la entrega rápida, excelente calidad en los productos y optimizando recursos durante su desarrollo. Debido a esto, este proyecto cumple con los objetivos establecidos y entrega un producto útil para la empresa A Plus Computer.

Abstract

Considering the importance of the implementation of internal control systems within SMEs, this degree project seeks the complete and modular development of different requirements requested by A Plus Computer implementing customized Plugins in the WordPress CMS. Afterwards, valuable information for the company is presented in a Business Intelligence panel that will help decision making.

The complete development is based on the Scrum methodology, which mainly guides the management of agile projects through fast delivery, excellent product quality and optimizing resources during development. Due to this, this project fulfills the established objectives and delivers a useful product for the company A Plus Computer.

ÍNDICE

1. CAPÍTULO I. INTRODUCCIÓN	1
1.1 Antecedentes	1
1.2 Alcance.....	3
1.3 Justificación	6
1.4 Objetivo General	6
1.5 Objetivos Específicos	7
2. CAPÍTULO II. MARCO TEÓRICO.....	7
2.1 Control Interno.....	7
2.2 Sistemas de control interno	7
2.3 Herramientas de software similares	8
2.4 A Plus Computer	3
2.5 Marco de trabajo	5
2.5.1 Roles, eventos y artefactos.....	5
2.5.1.1 Roles	5
2.5.1.2 Eventos	6
2.5.1.3 Artefactos.....	6
2.5.2 Ciclo de vida Scrum	6
2.5.3 Medición y estimación ágil	7
2.6 Tecnologías de desarrollo	9
2.6.1 CMS.....	9
2.6.1.1 Template Hierarchy.....	10
2.6.1.2 Post Types	12

2.6.1.3	Modelo de base de datos	2
2.6.2	Lenguaje de programación PHP	7
2.6.3	Sistema gestor de bases de datos.....	7
2.6.4	Business Intelligence.....	7
2.6.5	Web Service	8
2.6.6	Hosting	8
2.6.7	Versionamiento de código.....	9
3.	CAPÍTULO III. PROPUESTA DE SOLUCIÓN	11
3.1	Product Backlog	11
3.2	Burn Down Charts	18
3.2.1	Sprint 1	18
3.2.2	Sprint 2	18
3.2.3	Sprint 3	19
3.3	Configuración de entorno del sistema	19
3.3.1	Dominio y Hosting	19
3.3.2	Instalación WordPress	21
3.3.3	Cuenta FTP	27
3.3.4	Repositorio Git.....	29
3.4	Consideraciones previas al desarrollo.....	29
3.5	Desarrollo / Sprints	35
3.5.1	Sprint 1	35
3.5.1.1	Sprint Planning.....	35
3.5.1.2	Sprint Backlog.....	35
3.5.1.3	Historias de Usuario.....	36
3.5.1.4	Desarrollo.....	39

3.5.1.5	Daily Scrum.....	52
3.5.1.6	Revisión	52
3.5.1.7	Retrospectiva	52
3.5.2	Sprint 2	52
3.5.2.1	Sprint Planning.....	52
3.5.2.2	Sprint Backlog.....	53
3.5.2.3	Historias de Usuario.....	53
3.5.2.4	Desarrollo.....	56
3.5.2.5	Daily Scrum.....	61
3.5.2.6	Revisión	62
3.5.2.7	Retrospectiva	62
3.5.3	Sprint 3	62
3.5.3.1	Sprint Planning.....	62
3.5.3.2	Sprint Backlog.....	63
3.5.3.3	Historias de Usuario.....	63
3.5.3.4	Desarrollo.....	66
3.5.3.5	Daily Scrum.....	72
3.5.3.6	Revisión	72
3.5.3.7	Retrospectiva	72
4.	CAPÍTULO IV. VALIDACIÓN Y PRUEBAS DE SOFTWARE	73
4.1	Pruebas de Caja Negra	73
4.2	Pruebas de Caja Blanca.....	85
4.3	Aprobación del Sistema por A Plus Computer	86
5.	CONCLUSIONES Y RECOMENDACIONES	87

5.1 Conclusiones.....	87
5.2 Recomendaciones.....	88
REFERENCIAS.....	89

ÍNDICE DE FIGURAS

<i>Figura 1: Proceso de Facturación</i>	4
<i>Figura 2: Ciclo de vida Scrum</i>	7
<i>Figura 3: Gráfico Burn Down</i>	9
<i>Figura 4: WordPress Template Hierarchy</i>	11
<i>Figura 5: Diagrama de Base de Datos de WordPress</i>	4
<i>Figura 6: Pasos para crear un nuevo registro de la entidad Cliente</i>	5
<i>Figura 7: Recuperar datos del cliente Juan Barahona</i>	6
<i>Figura 8: Información meta del cliente Juan Barahona</i>	6
<i>Figura 9: Flujo de trabajo entre ramas de Git</i>	10
<i>Figura 10: Flujo del guardado de datos en Git</i>	10
<i>Figura 11: Burn Down Chart Sprint 1</i>	18
<i>Figura 12: Burn Down Chart Sprint 2</i>	18
<i>Figura 13: Burn Down Chart Sprint 3</i>	19
<i>Figura 14: Link para administrar los subdominios</i>	20
<i>Figura 15: Crear el nuevo subdominio sci.aplusecuador.com</i>	20
<i>Figura 16: Link para administrar los archivos</i>	20
<i>Figura 17: Carpeta creada para el dominio sci.apluscomputer.com</i>	21
<i>Figura 18: Link para administrar sitios WordPress</i>	21
<i>Figura 19: Botón para realizar una nueva instalación de WordPress</i>	21
<i>Figura 20: Escoger el dominio sobre el cual se realizará la instalación</i>	22
<i>Figura 21: Ingresar datos de usuario administrador y sitio</i>	22
<i>Figura 22: Instalación finalizada. Link para configurar el sitio WordPress</i>	22
<i>Figura 23: Inicio de la configuración del sitio</i>	23
<i>Figura 24: Ingresar el tipo, industria, título y eslogan</i>	23
<i>Figura 25: Seleccionar si se desea agregar perfiles sociales</i>	24
<i>Figura 26: Elegir el tema</i>	24
<i>Figura 27: Dashboard del sitio “Sistema de Control Interno”</i>	25
<i>Figura 28: En plugins únicamente se mantendrá Classic Editor</i>	26
<i>Figura 29: En temas únicamente se mantendrá Twenty Seventeen</i>	26
<i>Figura 30: En páginas únicamente se mantendrá Home</i>	26
<i>Figura 31: En posts únicamente se dejará vacío</i>	26

<i>Figura 32:</i> En media únicamente se dejará vacío.	27
<i>Figura 33:</i> Se seleccionará el tipo de <i>Permantlinks</i>	27
<i>Figura 34:</i> Link para administrar cuentas <i>FTP</i>	27
<i>Figura 35:</i> Conexión de la cuenta <i>FTP</i> para <i>sci.apluscomputer.com</i>	28
<i>Figura 36:</i> Conexión exitosa con <i>FileZilla</i> utilizando la cuenta <i>FTP</i> creada.	28
<i>Figura 37:</i> Estructura de Artículo.	40
<i>Figura 38:</i> Estructura de Artículos en <i>Sublime Text 3</i> localmente.	41
<i>Figura 39:</i> Archivo <i>prerrequisitos.php</i>	42
<i>Figura 40:</i> <i>Plugin</i> activado desde el <i>dashboard</i> del sitio.	42
<i>Figura 41:</i> Archivo <i>cpt_articulo.php</i>	44
<i>Figura 42:</i> Archivo <i>cpt_articulo.php</i>	44
<i>Figura 43:</i> Anotación para indicar archivo de traducción.	45
<i>Figura 44:</i> Archivo <i>tax_articulo_marca.php</i>	46
<i>Figura 45:</i> Archivo <i>example-functions.php</i>	47
<i>Figura 46:</i> Archivo <i>prerrequisitos.php</i>	47
<i>Figura 47:</i> Archivo <i>mtx_articulo.php</i>	48
<i>Figura 48:</i> Archivo <i>mtx_articulo.php</i>	49
<i>Figura 49:</i> Representación de la entidad Artículo en el <i>dashboard</i>	49
<i>Figura 50:</i> Archivo <i>mtx_suscripción.php</i>	50
<i>Figura 51:</i> Archivo <i>mtx_suscripcion.php</i>	51
<i>Figura 52:</i> Representación de la entidad Suscripción en el <i>dashboard</i>	51
<i>Figura 53:</i> Archivo <i>mtx_ticket.php</i>	56
<i>Figura 54:</i> Representación de la entidad Ticket en el <i>dashboard</i>	57
<i>Figura 55:</i> Archivo <i>mtx_factura.php</i>	57
<i>Figura 56:</i> Archivo <i>mtx_factura.php</i>	58
<i>Figura 57:</i> Archivo <i>facturas.php</i>	58
<i>Figura 58:</i> Archivo <i>facturas.php</i> (Función que busca los Tickets).	59
<i>Figura 59:</i> Archivo <i>facturas.php</i>	59
<i>Figura 60:</i> Archivo <i>sciaplus-script.js</i> para listar los tickets.	60
<i>Figura 61:</i> Archivo <i>sciaplus-script.js</i> donde actualiza suscripción y artículos. ..	60
<i>Figura 62:</i> Representación de la entidad Factura en el <i>dashboard</i>	61
<i>Figura 63:</i> Factura en el <i>dashboard</i> con campos actualizados.	61

<i>Figura 64: Archivo <code>prerrequisitos.php</code>.</i>	66
<i>Figura 65: Petición <code>GET</code> a <code>endpoint</code> de clientes.</i>	67
<i>Figura 66: Elementos de <code>Spoon</code> para el proceso de los datos.</i>	68
<i>Figura 67: Configuración del elemento <code>Generate Rows</code>.</i>	68
<i>Figura 68: Configuración del elemento <code>Rest Client</code>.</i>	68
<i>Figura 69: Configuración del elemento <code>JSON Input</code>.</i>	69
<i>Figura 70: Configuración del elemento <code>Select Values</code>.</i>	69
<i>Figura 71: Creación de tabla de Cliente en la base de datos.</i>	70
<i>Figura 72: Configuración del elemento <code>Insert / Update</code>.</i>	70
<i>Figura 73: Log del proceso en <code>Spoon</code>.</i>	71
<i>Figura 74: Consulta de los datos procesados en la base de datos.</i>	71
<i>Figura 75: Consulta de datos de cliente en <code>Power BI</code>.</i>	72
<i>Figura 76: Resultado obtenido de la creación del Cliente.</i>	74
<i>Figura 77: Resultado obtenido de la creación del Técnico.</i>	75
<i>Figura 78: Resultado obtenido de la creación del Dispositivo.</i>	76
<i>Figura 79: Resultado obtenido de la creación del Artículo.</i>	77
<i>Figura 80: Resultado obtenido de la creación de la Suscripción.</i>	78
<i>Figura 81: Resultado superior obtenido de la creación del Ticket.</i>	80
<i>Figura 82: Resultado inferior obtenido de la creación del Ticket.</i>	80
<i>Figura 83: Resultado superior obtenido de la creación de la Factura.</i>	82
<i>Figura 84: Resultado inferior obtenido de la creación de la Factura.</i>	82
<i>Figura 85: Resultado obtenido del procesamiento de datos en <code>Spoon</code>.</i>	83
<i>Figura 86: Resultado obtenido de la presentación de datos en <code>Power BI</code>.</i>	84
<i>Figura 87: Resultado obtenido de la creación de usuarios con rol.</i>	85
<i>Figura 88: Carta de aprobación del sistema por parte de <code>A Plus Computer</code>.</i>	86

1. CAPÍTULO I. INTRODUCCIÓN

1.1 Antecedentes

La falta de herramientas tecnológicas para el registro, proceso y despliegue de datos han provocado que sus servicios sean inferiores a los de las empresas que sí lo han sabido implementar. Algunas empresas prestan sus servicios a través de sistemas de gestión poco eficientes como los diferentes formatos de registro de datos en papel, esto genera pérdida de clientes, gastos innecesarios en personal, riesgo en la fiabilidad de los datos y un nulo crecimiento en la inteligencia del negocio. Por otro lado, las empresas que han invertido sus recursos implementando herramientas de software que facilite su trabajo, proporcione un valor agregado y diversifique su mercado, han obtenido como resultado generar más ganancia económica, priorización y control de sus actividades, visualización de sus resultados entre otras ventajas. (Hernández , Cardona, & Del Rio, 2017)

La empresa *A Plus Computer* se dedica a la venta, mantenimiento y reparación de computadoras, así como la instalación de redes y cámaras de seguridad. Es una empresa pequeña ecuatoriana ubicada en Quito con más de 20 años de experiencia sobre su campo.

Actualmente *A Plus Computer* está enfrentando algunos problemas que afectan negativamente en el control interno de los procesos que se llevan a cabo. Entre los principales problemas identificados se puede mencionar:

- Las facturas emitidas a los clientes son generadas a través de una hoja de cálculo de Excel específica. Esto genera dificultades al momento de buscar una factura, conocer las que no se han pagado y las ganancias mensuales y anuales. Esto lo conoce el contador; sin embargo, su respuesta no es inmediata. No se cuenta con una base de datos de los clientes. Toda su información se encuentra en facturas como en órdenes de trabajo sobre equipos atendidos.

- Las órdenes de trabajo son pequeños papeles en los que se registra el cliente, sus datos de contacto y el problema de el o los equipos que desea reparar o dar mantenimiento. El problema radica en la pérdida de tiempo al registrar una y otra vez al mismo cliente si ya se lo conoce, esto genera que un técnico se demore tomando los datos del cliente o simplemente confíe de su memoria y recuerde lo que necesita. Debido a esto existe una confusión entre lo que se hizo, se debe hacer y lo que el cliente solicitó. Además de esto, existen clientes que regresan por la garantía de un trabajo luego del tiempo permitido, al no tener un control de las fechas se lo vuelve a realizar de forma gratuita con la finalidad de mantener el cliente.
- Además de lo anterior, la empresa vende suscripciones a servicios como licencias de antivirus y mantenimientos anuales. Durante los últimos años este sector del negocio se ha mantenido en constante aumento por lo cual *A Plus Computer* enfrenta la necesidad de tener un sistema que registre las suscripciones e informe cuándo estas están por vencerse. Actualmente este control se lo tiene en una hoja de cálculo de Excel en la cual se accede diariamente para registrar nuevas suscripciones como para conocer el vencimiento de otras.

Además de estas dificultades se debe tomar en cuenta los errores humanos como: mala memoria al recordar que era lo que el cliente dijo o necesitaba, como también la falta de conocimiento de los elementos que el cliente dejó (maletas, estuches, cargadores, USBs, CDs, entre otros); mala escritura en papel, olvido o pérdida de la orden de trabajo, ingreso o reemplazo de un puesto laboral en donde se desconoce el estado de sus actividades.

Tomando en cuenta lo anterior se pretende realizar un sistema que mejore los tiempos de respuesta ante los servicios que presta a sus clientes y de esta manera mejorar el control interno en los procesos que gestiona *A Plus Computer*. Además, se desea que estos datos puedan ser procesados, almacenados y representados en un panel para tener una mayor asimilación de los resultados

obtenidos sobre ciertos objetivos en diferentes periodos de tiempo; de esta forma facilitaría la toma de decisiones.

1.2 Alcance

Partiendo de la problemática planteada se pretende implementar un sistema web con los siguientes módulos:

- **Módulo de facturación:** realizará el registro e impresión de los artículos o servicios facturados por parte de *A Plus Computer* a sus clientes. En este se registrará la información del cliente, el detalle de la factura, el valor total y su estado. Se agregarán campos personalizados que no son necesarios en la factura como por ejemplo notas, ticket o servicio de referencia.
- **Módulo de gestión de tickets:** se encarga del seguimiento detallado del estado de un equipo que requiere mantenimiento o reparación. Estos equipos pueden ser computadoras de escritorio, laptops, impresoras, periféricos, UPSs, entre otros. Cuando el cliente entregue su equipo se registrarán los elementos adicionales como cargadores, USBs, CDs, estuches, etc; el o los problemas a resolver, el costo y tiempo de entrega estimado. Posteriormente el equipo pasa por diferentes procesos como limpieza, pruebas de software, pruebas de hardware, cambio de piezas, etc. los cuáles serán registrados hasta que finalmente se entregue al cliente.
- **Módulo de suscripción a servicios:** registrará los servicios adquiridos por cada cliente sobre licencias de antivirus, programas de mantenimiento, entre otros. Aquí se mostrará un panel con los servicios a vencerse, así como los clientes que no han adquirido dichos servicios además de notificaciones.

Estos tres módulos necesitarán la información guardada sobre clientes, técnicos, artículos y dispositivos para funcionar, por lo que los tickets, suscripciones y facturas se comporta como un *plugin* o una funcionalidad adicional. Se extenderá

la funcionalidad de *WordPress* en donde se lo utilizará como un sistema manejador de contenidos, pero además se implementará los métodos de *PHP* y *WordPress* para la persistencia y presentación de los datos. Posteriormente estos datos serán recolectados, almacenados, procesados y representados con herramientas de inteligencia de negocios.

Se conoce que *WordPress* es utilizado para proyectos pequeños cuyos objetivos son la elaboración de páginas web estáticas o dinámicas que muchas de las veces no logran la funcionalidad, flexibilidad y comportamiento exacto que se desea. *WordPress* gestiona entradas, páginas, adjuntos, revisiones y menú de navegaciones; sin embargo, los *Custom Post Types* nos permiten agregar entradas personalizadas y registrarlas en la base de datos *MySQL* como una entidad a manipular. De esta forma se crean módulos que pueden funcionar solos o conectados a muchos otros. Además, *WordPress* se encarga de administrar tanto la entidad, seguridad y representación gráfica para el manejo de usuarios. Por consiguiente, es posible enfocarse en la problemática a resolver con el nuevo software y apoyarse de esta tecnología para facilitar el desarrollo y mantenibilidad del sistema haciéndolo atómico, consistente, aislado y durable. (Leary, 2010)

Este proyecto se desarrollará sobre diferentes tecnologías específicas, las cuales en conjunto podrán proporcionar la arquitectura deseada para el despliegue del sistema. A continuación, se enlista cada una de ellas:

El lenguaje de programación que se usará será *PHP* en su versión 5.4. Este lenguaje se ha seleccionado debido a que *WordPress* se basa en *PHP* y a pesar de que admite la versión 7, la 5.4 es la más estable al momento con soporte de diferentes desarrollos existentes.

El sistema administrador de base de datos será *MySQL* en su versión *Community Server*. Debido a su gratuidad, rendimiento y fácil integración con *WordPress* se ha decidido implementarlo tomando en cuenta que A Plus Computer cuenta con no más de diez miembros del personal, lo cual no exige un potente rendimiento del *DBMS*.

El CMS a utilizar es *WordPress* en su versión 5.2 (última versión). Entre los principales sistemas manejadores de contenido podemos mencionar *WordPress*, *Joomla* y *Drupal*, entre muchos otros. Estos tres son considerados los mejores y cubren el 71% del mercado de CMS a nivel mundial; sin embargo, *WordPress* se destaca en pequeños y medianos negocios por el bajo costo de desarrollo frente a *Drupal* por ejemplo, el software existente que es adaptable con gran facilidad como *WooCommerce* lo cual pone en desventaja a *Joomla* y finalmente la facilidad y extensibilidad que tiene para desarrollar módulos de software a través de los *Custom Post Types* y taxonomías además de la disposición de un servicio *Rest*, lo coloca en la mejor opción para este proyecto que cumplirá las necesidades de A Plus Computer.

Estas tecnologías se mantendrán activas dentro de un *Hosting* pagado por A Plus Computer. *Godaddy* es la empresa que provee de este servicio y actualmente A Plus Computer cuenta con un dominio y una página web informativa alojada allí (aplusecuador.com) con un plan de "*Hosting Deluxe*" el cual provee sitios, almacenaje y subdominios ilimitados con 500Mb en Ram. Estas características son suficientes para este proyecto, sin embargo, si llegara a tener problemas en la velocidad de procesos y consultas A Plus Computer rentará más memoria Ram para solventar este tema.

Una vez que los módulos del sistema entren en funcionamiento, comenzarán a guardar grandes cantidades de datos diariamente. Para ayudar al usuario en su toma de decisiones se mostrarán los datos usando la herramienta *Spoon* de la suite de *Pentaho BI* para la extracción y procesamiento de la misma, posteriormente la herramienta *Power BI* se encargará de representar estos datos.

Pentaho por su parte es una herramienta libre para generar inteligencia empresarial la cual cuenta con herramientas integradas para la minería de datos y procesos *ETL*. Mientras que *Power BI* es una herramienta proporcionada por *Microsoft* para la visualización de datos e inteligencia de negocios. Ambas herramientas se destacan en su área y cuentan con la ventaja de poder implementarlas sin costo alguno; a pesar de esto mantienen una versión de paga

para mejores características. *A Plus Computer* puede comenzar y mantenerse por un largo periodo de tiempo con sus versiones gratuitas.

En base a lo explicado se propone realizar tres módulos de software: módulo de facturación, módulo de gestión de tickets y módulo de suscripciones a servicios; junto con un panel de inteligencia de negocios.

1.3 Justificación

Partiendo de la implementación de los módulos de facturación, gestión de tickets y suscripción a servicios se puede solventar problemas como errores humanos de escritura, lectura o falta de memoria en los trabajos realizados, creación de facturas y cobro; descuido de la renovación de servicios y manejo de garantías, además se obtendrá más tiempo en el cual todo el personal tendrá un mejor rendimiento para trabajar y despreocuparse del registro de datos en *A Plus Computer*. Los clientes podrán ser almacenados con toda su información, se podrá llevar el control de cada equipo tecnológico al que se preste un servicio, se conocerá el trabajo realizado por cada uno de sus colaboradores y se notificará los servicios a vencerse lo cual generará un mejor servicio al cliente.

Posteriormente, la recolección de todos estos datos brindará la posibilidad de representar gráficamente en una herramienta de *BI* los datos exactos sobre el rendimiento del negocio. De esta forma no solo resolverá los problemas actuales con los que *A Plus Computer* se enfrenta, sino que además tendrá el conocimiento para saber el tiempo estimado de demora en un trabajo, la necesidad de capacitaciones al personal, la oportunidad de ingresar en un nuevo mercado o simplemente expandirse geográficamente.

1.4 Objetivo General

Desarrollar un sistema web basado en *WordPress* y panel *BI* para el control interno de los servicios de tecnología que presta *A Plus Computer*.

1.5 Objetivos Específicos

- Realizar la fundamentación teórica para determinar el ambiente de desarrollo.
- Desarrollar los módulos de facturación, gestión de tickets y suscripción a servicios.
- Implementar en base a la información obtenida un panel de *Business Intelligence* desarrollado gracias a las herramientas *Spoon* y *Power BI*.
- Validar la propuesta de solución utilizando pruebas de calidad de software.

2. CAPÍTULO II. MARCO TEÓRICO

2.1 Control Interno

Es difícil indicar la definición exacta de lo que representa el control interno debido a su amplio campo de aplicación, sin embargo, la definición más cercana con respecto a la industria y empresas indica que es un proceso ejecutado por diferentes personas de una organización para brindar orden y seguridad que ayuden al logro de los objetivos y cumplimiento de normas. (Mantilla, 2018)

2.2 Sistemas de control interno

En el área de software se considera sistema de control interno al grupo de dispositivos que se encargan de administrar, ordenar, dirigir o regular la conducta de otro sistema para disminuir fallos y obtener resultados. (Ñeco, 2003)

Estos sistemas pueden aplicarse a diferentes áreas del negocio por lo que un sistema de control interno se encarga de los procesos tales como organización del personal, materiales de trabajo, trabajos realizados, área contable, seguridad y en general los procesos que son llevados a cabo para que la empresa pueda brindar un mejor servicio hacia el cliente desde una perspectiva propia y no exigiendo que el cliente realice un proceso complejo para obtener lo que desea.

En el caso de *A Plus Computer* se crearán los módulos de facturación, gestión de tickets y suscripción a servicios los cuales conforman un sistema de control interno específico para esta empresa. Es posible agregar nuevos módulos los cuales dependiendo de su función pueden ampliar el alcance de este sistema interno, por lo tanto, cuando se menciona el sistema de control interno se hace referencia a todos sus módulos que al integrarse cumplen con los resultados esperados por la empresa.

2.3 Herramientas de software similares

Actualmente existe software muy robusto que cumple con las expectativas de cada módulo, sin embargo, el costo de un sistema que integre estos módulos es muy alto debido a que este tipo de software está enfocado en procesos macros de negocio como lo son los *CRMs* en donde su implementación, mantenimiento y cambios son muy específicos. Es por esto que se ha decidido realizar un sistema de control interno personalizado para una PYME como lo es *A Plus Computer* en donde su costo será mucho menor, pero cumplirá con los requisitos de personalización que la empresa los desea. A continuación, se menciona tres tipos de software que cumplen con la función de los módulos propuestos.

El primero es Facturación 123 Pymes y Autónomos es un sistema de facturación gratuito para su versión básica y de paga para su versión completa la cual se muestra ideal para PYMEs, el sistema de paga cuenta con gestión de:

- Clientes
- Proveedores
- Productos
- Facturación
- Estadísticas
- Control de Gastos
- Listados
- Presupuestos y balances

(TPV Gratuito 123, 2019)

Este software es uno de los más completos en el área de facturación para pequeñas y medianas empresas. Sin embargo, su costo es de USD 150.00

anuales, no es posible agregar funcionalidades personalizadas, es un software local y no permite el consumo de sus datos a través de internet. De este software se puede tomar como ejemplo la administración de clientes y facturas.

El segundo producto es Freshdesk, un software de soporte utilizado por pequeñas y grandes empresas reconocidas internacionalmente que ayuda principalmente a la:

- Administración de tickets
- Asignación automática de tickets
- Generación de informes empresariales
- Gestión de tickets por correo electrónico
- Detección de colisión de agentes
- Monitoreo y entrada de llamadas

(Freshworks, 2019)

Es un sistema que se acopla a todo tipo de negocio que brinde soporte y ha sido considerado entre los mejores productos de gestión de tickets. El costo de licenciamiento es de USD 50.00 anuales con posibilidad de implementar extensiones bajo un costo extra. Por otro lado, se limita al desarrollo y configuración bajo las consideraciones de *Freshworks* los cuales no se dedican a integrar diferentes sistemas. De este producto se puede tomar las entidades que requiere para la gestión de tickets, sus estados y sus flujos de procesos.

El tercer software es Billerk, un producto para la gestión de suscripciones y facturación que permite gestionar, controlar y facturar cualquier modelo de negocio o producto por suscripción automáticamente. (Billwerk, 2019)

Tiene un excelente soporte hacia el usuario y cuenta con un sistema de facturación, sin embargo, su costo es de USD 160.00 mensuales, no permite integración con otros sistemas ni consumo de sus datos. De este software se

puede destacar la sencillez en la estructura de sus procesos, la facilidad de manejo de las suscripciones y la relación directa con la facturación.

Estos sistemas son claros ejemplos de cada módulo que *A Plus Computer* necesita, sin embargo, la mayoría de ellos no se integran entre sí ni pueden generar *endpoints* de consulta para la sección de inteligencia de negocios. De todas formas, se ha tomado en cuenta el funcionamiento, proceso y almacenamiento de los datos como ejemplos muy cercanos a los requerimientos.

2.4 A Plus Computer

A Plus Computer es una empresa ecuatoriana creada en el año 2000 por su Gerente Propietario Marco Antonio Escobar que se dedica a la venta, reparación y mantenimiento de equipos tecnológicos como computadoras, impresoras, proyectores, equipos de monitoreo y periféricos, entre otros dispositivos electrónicos. Redes para hogares y oficinas. Además de asesoría para la implementación de nuevas tecnologías en el vivienda o negocio.

Misión: Nuestro compromiso es proveer de tecnología preeminente de venta, mantenimiento, reparación de equipos, instalación de redes, cámaras de seguridad y fines, para su empresa, negocio y hogar; escuchando sus necesidades y desarrollando soluciones que resuelvan sus problemas utilizando nuestro liderazgo y el compromiso mutuo de nuestros colaboradores logramos un trabajo eficiente.

Visión: Nos proyectamos a ser referencia en el mercado de TECNOLOGÍA EN COMPUTACION Hardware y Software, innovadores y competitivos a nivel nacional, conservando un trabajo ético, voluntad de servicio, confianza de clientes, bienestar y progreso conjunto de nuestra empresa.

Valores: Trabajo en equipo, Aprendizaje permanente, Optimismo e Innovación.

(A Plus Computer, 2019)

Está ubicada en el sector de la Jipijapa (Quito) y está conformada por su gerente que dedica gran parte de su tiempo a la administración del personal técnico, la

atención al cliente, ventas y facturación; existe un área contable, un departamento técnico interno conformado por 3 colaboradores y finalmente 3 técnicos externos en donde cada uno tiene un área de mejor desenvolvimiento.

En *A Plus Computer* existen varios procesos de negocio que a lo largo de los años se han mejorado gracias a las actitudes y aptitudes de las personas que lo conforman y las sugerencias de los clientes. La constante mejoría, la rapidez en respuesta a las solicitudes y una atención de calidad han logrado satisfacer a su cartera de clientes en donde su mayoría son empresas.

La metodología SIPOC se utiliza para establecer el alcance de un proceso mediante las etapas: proveedores, entradas, proceso, salidas y cliente. Dependiendo del proceso se usan todas o las más importantes. (Alonso, 2014)

El proceso de negocio llamado Facturación requiere de la administración de las entidades de cada módulo, este proceso se muestra a continuación:

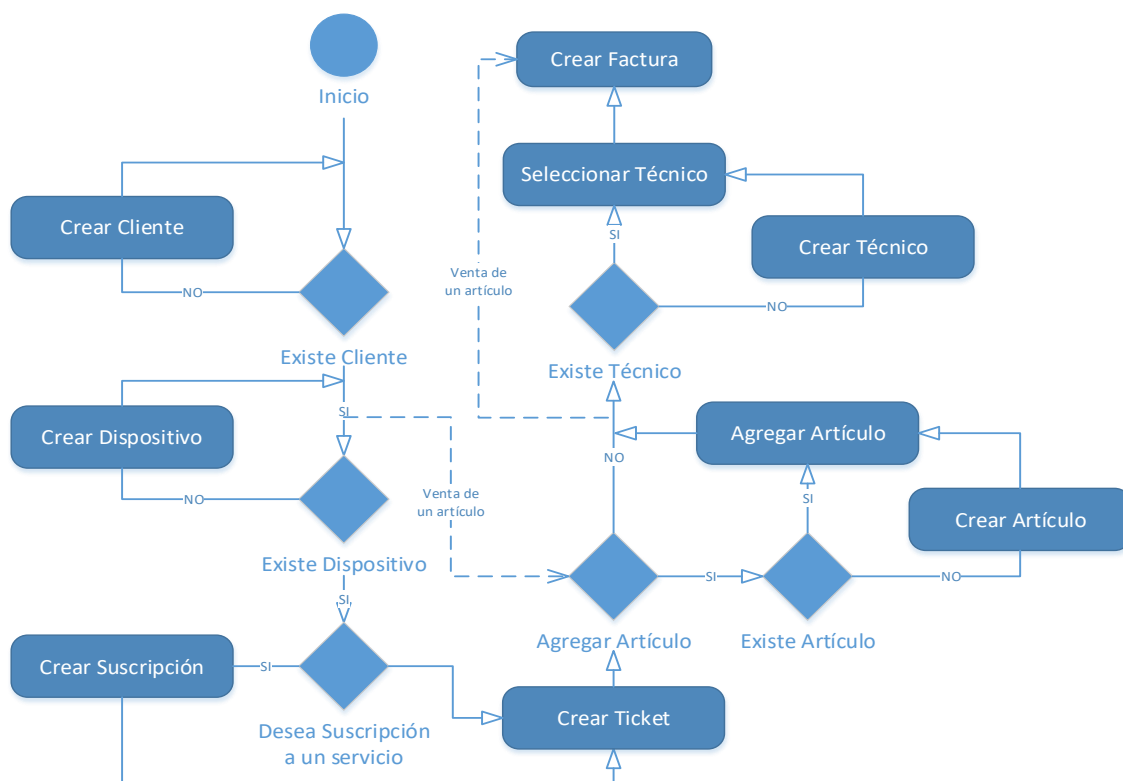


Figura 1: Proceso de Facturación

Tabla 1.

Diagrama SIPOC del Proceso de Facturación.

Entradas	Proceso	Salidas
<p>Cliente Dispositivo</p>	<p>Primero se identifica si existe o se debe crear el cliente y su dispositivo, dependiendo de si desea una suscripción a un servicio se genera un ticket el cual se factura una vez que ha finalizado.</p>	<p>Factura</p>

2.5 Marco de trabajo

El marco de trabajo es *Scrum*, el cual sirve para el desarrollo de proyectos muy difíciles. El principal objetivo de este marco de trabajo es la entrega de pequeños productos con gran valor siendo significativos para un producto más grande. Entender su metodología y ponerlo en práctica es muy fácil, lo complicado es llegar a dominarlo y generar con este los resultados esperados del proyecto.

2.5.1 Roles, eventos y artefactos

2.5.1.1 Roles

Existen tres tipos de roles dentro de la metodología *Scrum* que interactúan entre sí de forma particular.

El ***Scrum Master*** es un líder que tiene un vasto conocimiento sobre *Scrum* y es el responsable de la aplicación de la teoría, prácticas y reglas. Entre sus principales tareas están la moderación en reuniones, la organización entre el equipo y dueño de producto para la creación y estimación de las tareas de *Backlog*, y asesorar en general sobre la aplicación de la metodología *Scrum* al proyecto en particular.

El **dueño del producto** o también llamado *product owner* representa al cliente y es responsable de que el equipo de trabajo genere el producto que el cliente desea cumpliendo con todos sus requerimientos. Este rol lo toma una sola persona la cual conoce perfectamente el negocio del cliente, además sus decisiones son respetadas y se encarga principalmente de la gestión de *Backlog*.

El **equipo de desarrollo** está conformado de 3 a 9 profesionales multifuncionales auto organizados que desarrollan el incremento. Entre las habilidades a destacarse está el dominio de la metodología *Scrum*, capacidad para el trabajo en equipo y el dominio técnico sobre el producto en desarrollo. (Nogales & Rosales, 2020)

2.5.1.2 Eventos

Los siguientes eventos se basan en el ciclo de vida de la metodología *Scrum*:

Sprint: Es cada iteración del desarrollo y se considera el corazón de *Scrum*.

Sprint Planning: Es una reunión al inicio de cada *Sprint* que establece su objetivo.

Daily Scrum: Es una reunión diaria en donde se acuerda el trabajo a realizar y dura máximo 15 minutos.

Sprint Review: Es una reunión que evalúa el incremento y modifica el *Backlog*.

Sprint Retrospective: Es una reunión en donde el equipo tiene autocríticas y un plan de mejora.

2.5.1.3 Artefactos

Representan el trabajo o valor que se realiza en el proceso de desarrollo.

Product Backlog: Es una lista del conjunto de todas las tareas a realizar para lograr el objetivo del proyecto.

Sprint Backlog: Es una lista de las tareas a realizar durante el *sprint* activo.

Incremento: Es el resultado de cada *sprint* y se caracteriza por cumplir cierta funcionalidad que el cliente pueda probar y si desea integrar inmediatamente.

2.5.2 Ciclo de vida Scrum

Scrum estructura el trabajo a desarrollar en *sprints*, estos son ciclos cuyo intervalo va de una a cuatro semanas en donde su duración es fija, por lo que

nunca se alargan y se limitan en tiempo. En su inicio se seleccionan los elementos a trabajar durante el *sprint* y no se los puede cambiar. El equipo se reúne todos los días durante un tiempo corto para compartir los avances, identificar problemas y actualizar las gráficas sobre el estado actual del proyecto. Cuando el *sprint* termina, el equipo tiene un producto que funcione, se pueda poner a prueba y/o incorporar si así se lo desea. Finalmente se analiza lo construido con los interesados del proyecto y se toman recomendaciones para el siguiente *sprint*. Este proceso está representado en la Figura 2:

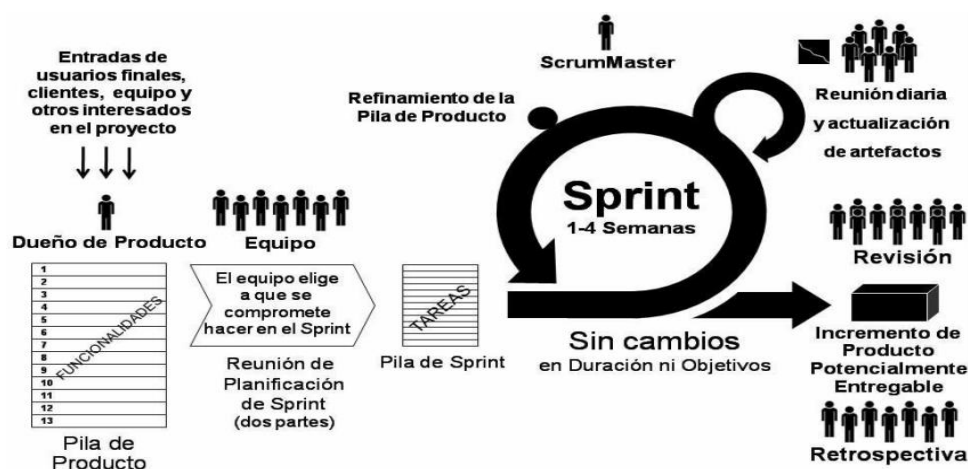


Figura 2: Ciclo de vida Scrum.

Tomado de (Deemer, Benefield, Larman, & Vodde, 2009)

2.5.3 Medición y estimación ágil

Determinar las métricas que se implementarán en el proyecto es un proceso complicado a pesar de que su comprensión sea muy sencilla, saber estimar las tareas tomando en cuenta el objetivo y la forma de trabajo del equipo es una habilidad que se adquiere con la experiencia en este campo.

Cuando se trabaja con Historias de Usuario o también llamadas HU se utilizan Puntos de Historia o PH y representa el tamaño de una actividad. Los PH son relativos ya que cada equipo puede tener el mismo número de PH, pero no significa que tengan el mismo peso de trabajo, todo esto depende de los requerimientos de las tareas y de las aptitudes del equipo. Para dar un valor al

PH es necesario tomar en cuenta la complejidad, el esfuerzo y el riesgo de la tarea.

La velocidad es el elemento que indica cuántos PH un equipo puede realizar en cada *sprint*, esta es utilizada para distribuir las tareas e indicar al cliente cuánto puede tardar la entrega del producto.

La estimación se la puede dar en base a la unidad de medida o limitaciones del punto de historia.

La unidad de medida indica que la tarea más sencilla tendrá 1PH, por lo que el resto de las tareas compararán su dificultad con dicha tarea y tendrán un valor; también se puede asumir que 1PH es una jornada laboral ideal de un integrante del equipo y a partir de allí cada tarea tendrá un peso según las horas que aproximadamente demore en completarse por una sola persona.

Las limitaciones del PH indican que se debe tomar en cuenta que el tiempo que se está estimando es un tiempo ideal donde no existen interrupciones o eventos en donde el equipo necesite invertir su tiempo por lo que se necesita conocer al equipo y determinar su velocidad para mitigar esta inexactitud. Además, mientras que la historia de usuario sea más grande, será más difícil determinar las jornadas laborales necesarias para completarla, por lo que se recomienda dividir en HU más pequeñas.

Existen técnicas como “Estimación de *Poker Variante Sucesión de Fibonacci*” la cual asigna el valor más cercano de la serie *Fibonacci* a cada tarea, también se utilizan cartas con símbolo de duda para informar que no se puede estimar con seguridad y otras cartas que indican descansos u otras actividades.

El gráfico *Burn Up* es una herramienta de planificación del dueño del producto para tener una guía o visión sobre la velocidad de desarrollo del equipo sobre todo el producto. Este es un diagrama cartesiano en el que el eje “Y” representa el esfuerzo estimado y en el eje “X” se representa el tiempo. Además, existe el gráfico *Burn Down* el cual mide la velocidad sobre el *sprint*.

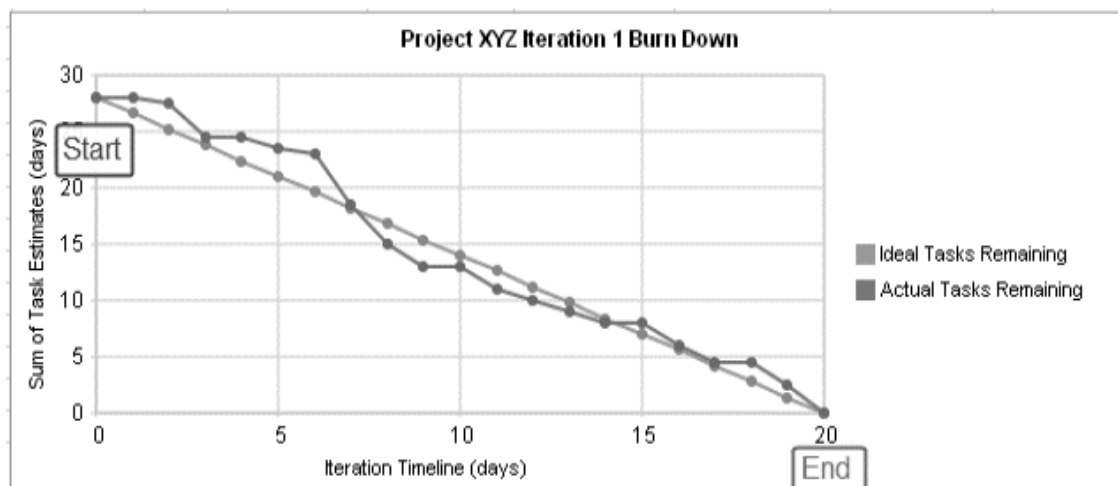


Figura 3: Gráfico *Burn Down*.

Tomado de: (Nogales & Rosales, 2020)

2.6 Tecnologías de desarrollo

A continuación, se da a conocer sobre las tecnologías utilizadas en el sistema de control interno de *A Plus Computer*.

2.6.1 CMS

Un *CMS* es un sistema de gestión de contenidos el cual hace referencia a los archivos que estructuran una página web y a un panel de control donde se configuran diferentes aspectos como diseño, plantillas, manejo de *URLs*, manejo de imágenes, *plugins* instalados, versiones del contenido, estadísticas entre muchas otras funcionalidades. (Patel, Rathod, & Prajapati, 2011)

Diferentes fuentes indican que *WordPress* es el *CMS* más utilizado a nivel mundial. La empresa *Buit With* se dedica a identificar las tecnologías utilizadas en páginas web, proveer de herramientas de *SEO* a las empresas e indicar las tendencias en todas las áreas de software. *Buit With Trends* posiciona a *WordPress* como el *CMS* más utilizado del 2019 con más de 30 millones de sitios activos seguido por *Wix* con 4 millones, *Squarespace* con 2 millones, *Joomla!* con 1.8 millones, *Shopify* con 1.1 millones y *Drupal* con medio millón. Esto se debe principalmente a la facilidad de uso tanto para programadores como para clientes de este *CMS*, además sus desarrolladores se han preocupado por

herramientas que potencien su funcionamiento especialmente en el mercadeo, es allí de donde se puede identificar ventajas como herramientas para *SEO*, visualización mejorada en dispositivos móviles, experiencia de usuario, creación de blogs y tiendas online con pasarelas de pago muy sofisticadas.

Siendo gratuito tanto *WordPress* como la mayoría de sus herramientas ha dado paso a que su estudio, desarrollo e implementación tenga un crecimiento exponencial durante la última década.

2.6.1.1 Template Hierarchy

WordPress utiliza una herencia de archivos o modelos para desplegar la información. Estos archivos son reusables y modulares, lo que posibilitan el uso de diferentes archivos a través de otros archivos, un ejemplo de esto es el uso del *header* o *footer* dentro de todas las páginas de una web.

WordPress identifica por la URL lo que se está solicitando y busca en su estructura de archivos los que den respuesta a la solicitud, si no existiesen pasan al siguiente escalón en su estructura. En la siguiente imagen se muestra desde el lado izquierdo en donde llega las diferentes solicitudes *URL* que pueden hacer referencia al sitio en sí, a una página específica, al conjunto o categorías de estas, resultados de una búsqueda o una página de error; pasando así por la herencia de estos archivos hacia otros que contienen por ejemplo *posts*, taxonomías, *slugs*; llegando finalmente al archivo principal *index.php* el cual en esta lógica, si no se encuentra ningún archivo referente a la petición solicitada por *URL*, este será presentado. (WordPress, 2019)

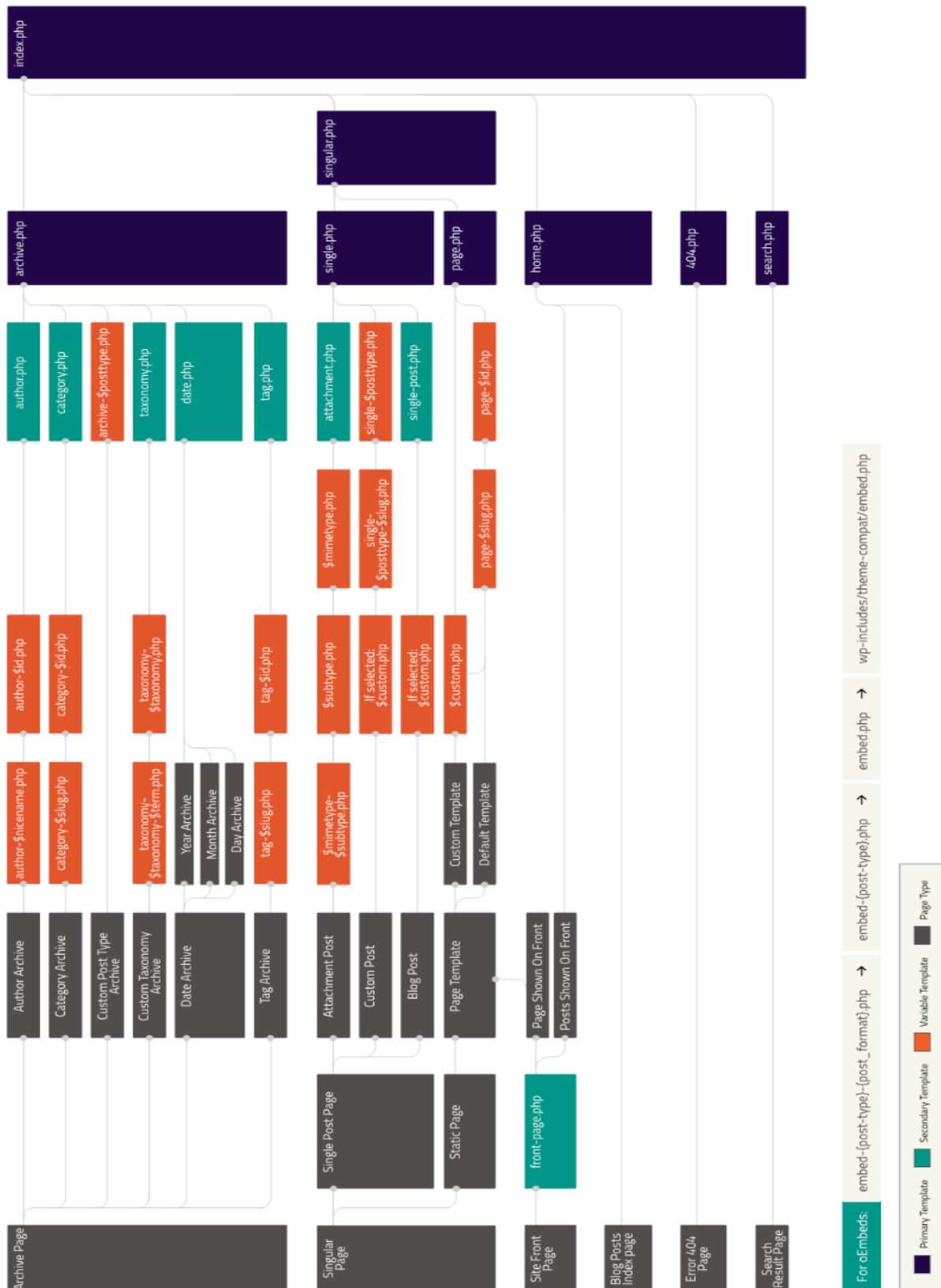


Figura 4: WordPress Template Hierarchy.

Tomado de (WordPress, 2019)

2.6.1.2 Post Types

WordPress maneja diferentes tipos de contenido, estos son llamados *Post Types* y suelen ser representados por los *Template Files*. Los *Post Types* pueden ser modificados o eliminados por el comportamiento de un *plugin* o tema, por defecto *WordPress* tiene los siguientes *Post Types*:

- *Post* (*Post Type*: '*post*'): *Post* es un tipo de *Post Types* el cual hace referencia a las entradas del sitio, por ejemplo, los artículos de lectura usados en blogs.
- *Page* (*Post Type*: '*page*'): Son las páginas creadas en las webs, por ejemplo, la página principal de una web, la página de contacto, la página donde se muestran los artículos en venta.
- *Attachment* (*Post Type*: '*attachment*'): Son los archivos multimedia.
- *Revision* (*Post Type*: '*revision*'): Son las revisiones o cambios realizados en los *Posts*.
- *Navigation menu* (*Post Type*: '*nav_menu_item*'): Son los menús de navegación, por lo general existe un principal y un secundario.

Adicionalmente, se pueden crear ***Custom Post Types*** los cuales extienden enormemente las capacidades de *WordPress* al posibilitar la creación de *Post Types* propios o personalizados, esto puede verse como la creación de entidades que se relacionan entre sí, persisten datos y forman un sistema completamente integrado. Los *Custom Post Types* pueden agregarse desde temas o *plugins*, sin embargo, se recomienda hacerlo desde este último para garantizar la portabilidad entre instalaciones *WordPress*. (WordPress, 2019)

El ejemplo más conocido de *Custom Post Type* es el *plugin WooCommerce* el cual ofrece la extensibilidad de una completa tienda en línea donde expone los productos, mantiene un carrito de compras y proporciona una pasarela de pagos además de muchas funcionalidades extras que se puede incorporar.

Taxonomies es el término utilizado para agrupar o clasificar características o atributos de un *Custom Post Type*, estas pueden ser jerárquicas o no y están compuestas por *Terms*. (WordPress, 2019)

A continuación, se presenta un ejemplo de la relación entre *Custom Post Types*, *Taxonomies* y *Terms*:

Se desea crear la entidad Persona, por lo cual el *Custom Post Type* se llamará 'persona', además se desea crear los atributos como nombre, apellido, fecha de nacimiento, género y ciudadanía; por lo que nombre, apellido y fecha de nacimiento serán atributos normales o también llamados *metafields* en donde el usuario ingresará su información personal, pero género y ciudadanía se puede tomar como *Taxonomies* debido a que es fácilmente agrupable. Finalmente, los *Terms* son las opciones de las *Taxonomies*. La estructura de la entidad Persona dentro de *WordPress* estaría representada de la siguiente forma:

- *Custom Post Type*:
 - Persona
- *Metafields*:
 - Nombre
 - Apellido
 - Fecha de nacimiento
- *Taxonomies*:
 - Género
 - Masculino (Term)
 - Femenino (Term)
 - Ciudadanía
 - Ecuatoriano (Term)
 - Ruso (Term)
 - Árabe (Term)

Al crear *Custom Post Types*, *WordPress* se encarga automáticamente de generar los métodos y visualizaciones de *Create*, *Read*, *Update* y *Delete* con la misma eficiencia que los *Post Types* por defecto.

De esta forma se puede crear diferentes *Custom Post Types* que representen una entidad específica, persistir en una base de datos y construir un sistema completo dentro de *WordPress*.

2.6.1.3 Modelo de base de datos

WordPress implementa un modelo relacional de base de datos que contiene las tablas principales para mantener los conceptos de estructura de una página web. A continuación, se describen sus tablas:

- *'wp_posts'*: Es considerada la tabla más importante y guarda los datos de cada post o entrada que se genere, además tiene una columna llamada *'post_type'* la cual indica el *Post Type* o *Custom Post Type* a la que pertenece el *Post*.
- *'wp_postmeta'*: Guarda metadatos de la tabla *'wp_posts'*.
- *'wp_comments'*: Almacena todos los comentarios de todos los *Posts* que admitan discusiones.
- *'wp_commentmeta'*: Guarda metadatos de la tabla *'wp_comments'*.
- *'wp_terms'*: Se registran los ítems de las *Taxonomies*.
- *'wp_term_taxonomy'*: Guarda el nombre de las agrupaciones o clasificaciones.
- *'wp_term_relationships'*: Registra la relación entre el *Post* y la *Taxonomy* correspondiente.
- *'wp_users'*: Guarda los usuarios del sistema.
- *'wp_usermeta'*: Guarda metadatos de la tabla *'wp_users'*.

- '*wp_options*': Almacena todas las opciones y configuraciones del sitio web como el tema, *plugins*, *widgets* y cache.
- '*wp_links*': Guarda los links hacia otros sitios.

(Poulson, 2015)

Esta relación entre las tablas de *WordPress* permite a los desarrolladores de temas y *plugins* crear varias entidades sin agregar más tablas e implementando los mismos métodos para la comunicación con la base de datos. Si se desea guardar específicamente una tabla con otros datos también es factible hacerlo, pero se debe tomar en cuenta que deben aplicarse métodos de comunicación personalizados con la base de datos.

Tomando el ejemplo Persona de la sección anterior, en la tabla '*wp_posts*' se guardará como tal una persona y estará "mezclada" con otras tuplas que guarden otro tipo de *Posts* como por ejemplo noticias, productos, compras, etc. *WordPress* sabrá diferenciarlos por su columna '*post_type*' en donde hará referencia al tipo de *Post Type* o *Custom Post Type* al que representa.

A continuación, se muestra las tablas que *WordPress* mantiene para el flujo de todo este proceso sin la necesidad de agregar nuevas:

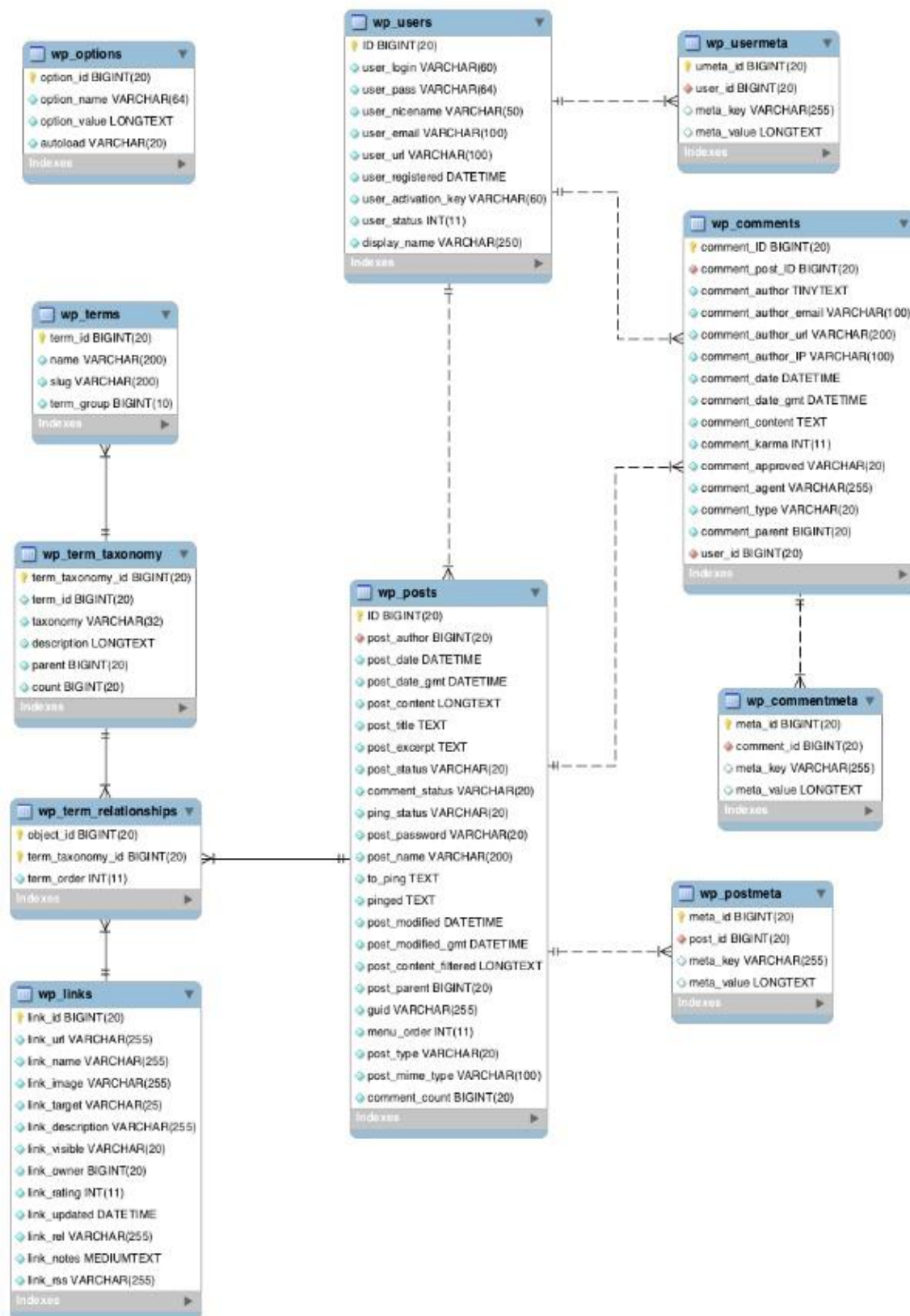


Figura 5: Diagrama de Base de Datos de WordPress.

Tomado de (Codex WordPress, 2019)

Por ejemplo, para crear un nuevo cliente llamado Juan Barahona es necesario seguir los siguientes pasos:

- 1) Dar clic en la opción Cientes del menú izquierdo.
- 2) Luego dar clic en Agregar Nuevo.
- 3) Llenar los campos correspondientes a Datos del Cliente.
- 4) Establecer la imagen destacada.
- 5) Dar clic en Publicar.

The screenshot shows the WordPress admin interface for adding a new client. The sidebar on the left has 'Clientes' selected (1). The main content area is titled 'Agregar Nuevo Cliente' and contains a form with the following fields: 'Datos del Cliente' (3), 'Cédula / RUC' (1715859654), 'Nombre' (Juan Barahona), 'Dirección' (El Condado), 'Teléfono Fijo' (2465896), and 'Celular' (985477886). On the right, there is a 'Publish' section with a 'Publish' button (5) and an 'Imagen Destacada' section with a placeholder image and an 'Eliminar Imagen Destacada' button (4).

Figura 6: Pasos para crear un nuevo registro de la entidad Cliente.

De esta forma la información principal como id, título y tipo de *post* se almacenará en la tabla *'wp_post'*, la columna *'post_type'* guardará el valor de *cliente* para identificar “el tipo de entidad del registro”.

Por otro lado, la información meta de los *metafields* que corresponde a los atributos del cliente se encuentran en la tabla *'wp_postmeta'*, la columna *'meta_key'* contiene el nombre del campo y la columna *'meta_value'* contiene los datos como se muestra a continuación:


```
SELECT id, post_title, post_type FROM `wp_posts` WHERE id = 489
```

Mostrar todo | Número de filas: 25 ▼ Filtrar filas

+ Opciones

id	post_title	post_type
489	1715859654 - Juan Barahona	cliente

Figura 7: Recuperar datos del cliente Juan Barahona.

```
SELECT * FROM `wp_postmeta` where post_id=489
```

Mostrar todo | Número de filas: 25 ▼ Filtrar filas: Buscar en esta tabla Ordenar según la clav

+ Opciones

	meta_id	post_id	meta_key	meta_value
<input type="checkbox"/> Editar Copiar Borrar	1305	489	_edit_lock	1581478179:1
<input type="checkbox"/> Editar Copiar Borrar	1308	489	_edit_last	1
<input type="checkbox"/> Editar Copiar Borrar	1309	489	_thumbnail_id	490
<input type="checkbox"/> Editar Copiar Borrar	1310	489	sciaplus_cliente_cedularuc	1715859654
<input type="checkbox"/> Editar Copiar Borrar	1311	489	sciaplus_cliente_nombres	Juan Barahona
<input type="checkbox"/> Editar Copiar Borrar	1312	489	sciaplus_cliente_direccion	El Condado
<input type="checkbox"/> Editar Copiar Borrar	1313	489	sciaplus_cliente_telefono	2465896
<input type="checkbox"/> Editar Copiar Borrar	1314	489	sciaplus_cliente_celular	985477886
<input type="checkbox"/> Editar Copiar Borrar	1315	489	sciaplus_cliente_fecha_nacimiento	16-07-1965
<input type="checkbox"/> Editar Copiar Borrar	1316	489	sciaplus_cliente_email	juan@gmail.com
<input type="checkbox"/> Editar Copiar Borrar	1317	489	yourprefix_demo_timezone	America/Guayaquil
<input type="checkbox"/> Editar Copiar Borrar	1318	489	yourprefix_demo_colorpicker	#ffffff
<input type="checkbox"/> Editar Copiar Borrar	1319	489	_wp_old_slug	auto-draft

Figura 8: Información meta del cliente Juan Barahona.

De esta forma se puede identificar cada registro de la tabla 'wp_post' como una entidad sin la creación de otras tablas y obtener todos sus campos adicionales con la tabla 'wp_postmeta'.

2.6.2 Lenguaje de programación PHP

WordPress está basado en *PHP*, actualmente su versión más estable es la 5 sin embargo existen muchas instalaciones con la versión 7 en donde no ha generado problemas.

PHP o también llamado *Hypertext Preprocessor* (preprocesador de hipertexto) es un lenguaje de programación de software libre orientado a servidores con fines muy generales. Fue creado en 1995 por Rasmus Lerdorf y su popularidad se debe a que fue un pionero en la implementación de su código dentro de los archivos *HTML*, esto facilitaba el desarrollo de páginas web a gran escala. (PHP, 2019)

2.6.3 Sistema gestor de bases de datos

Las únicas bases de datos soportadas por *WordPress* son *MySQL* desde la versión 5.0.15 y *MariaDB* en todas sus versiones, sin embargo, la más utilizada es *MySQL* manejada desde *phpMyAdmin*. (Codex WordPress, 2019)

MySQL es un gestor de base de datos relacional de código abierto basado en lenguaje de consulta estructurado. Se lo asocia con páginas o aplicaciones webs. Fue creado por *MySQL AB* y posteriormente en el 2008 adquirida por *Oracle*. (Rouse, 2015)

Esta herramienta se la implementará para el guardado y extracción de datos desde *Business Intelligence*.

2.6.4 Business Intelligence

Business Intelligence hace referencia a las estrategias implementadas para la recopilación, almacenamiento y análisis de datos existentes en una empresa para el proceso de toma de decisiones. Este término se originó en 1989 y a pesar de que no existían sistemas informáticos digitales, se aplicaba sobre los datos escritos y sobre el análisis de la competencia. (Negash & Gray, 2008)

Comúnmente la principal fuente de la inteligencia de negocios son los datos obtenidos por sistemas informáticos internos de la empresa, sin embargo,

muchas empresas desean tener una alta exactitud sobre sus decisiones e integran información de fuentes externas como el clima, redes sociales, encuestas en línea entre muchos otros sistemas que proporcionen información.

Pentaho Data Integration o también llamado *Kettle* es parte de la suite de *Pentaho* que sirve para la extracción, transformación y carga de datos también conocida como *ETL*. Fue creado en el 2004, es gratuito y es considerada una de las herramientas líderes en la extracción y procesamiento de datos. Es muy utilizado en *Big Data* y posteriormente en Internet de las Cosas (Martín, 2019)

Es una herramienta muy fácil de usar que ayudará a las empresas a procesar sus datos sin tener que crear un sistema que lo haga, lo cual es muy complicado. Además, existe bastante documentación y ejemplos de uso por lo que su aprendizaje es muy rápido.

Por otro lado, *Power BI* es una herramienta para análisis empresarial en la que se visualizan los datos procesados y comparte esta información con diferentes personas y aplicaciones externas. (Microsoft, 2019)

2.6.5 Web Service

Web Service es una tecnología diseñada para la interacción interoperable entre máquinas dentro de una red en donde se comunican por protocolo *HTTP* con una serialización *XML* junto a otros estándares. (Haas & Brown, 2004)

Gracias a los *Web Services* es posible que diferentes aplicaciones de software que están desarrolladas en diferentes lenguajes de programación, plataformas y sistemas operativos puedan intercambiar datos mientras tengan acceso a la red o internet. Al establecer estándares abiertos propuestos por *OASIS* y *W3C* ha sido posible mejorar la interacción entre aplicaciones.

2.6.6 Hosting

El *hosting* o alojamiento web es el servicio que proporciona el contenido web a los usuarios, comúnmente suele almacenarse en servidores y proporcionan más

servicios como correo, despliegue de aplicaciones, acceso *FTP* y creación de discos virtuales. (López, 2019)

GoDaddy es uno de los proveedores de dominios y *hosting* más reconocidos a nivel mundial debido a su eficiencia, precios, administración de productos y atención al cliente. Tiene servidores compartidos, dedicados y virtuales en *Linux* y *Windows*. (GoDaddy, 2019)

2.6.7 Versionamiento de código

Haciendo referencia al área de software, el control de versiones es la administración y revisión de los cambios generados en el código fuente y la estructura de un producto de software. Este se encarga de guardar e identificar los cambios realizados por el desarrollador o desarrolladores en proyectos colaborativos. (Práctica del Desarrollo de Software, 2019)

Git es una herramienta de código abierto diseñado por Linus Torvalds para el versionamiento de código en pequeños y grandes proyectos que no necesita un repositorio central. (Git, 2019)

Entre sus buenas prácticas de desarrollo indica que deben existir cuatro ramas de desarrollo:

- 1) *Master*: contiene el código que se encuentra en producción
- 2) *Development*: contiene los cambios pequeños y correcciones que se están realizando, cuando esta esté finalmente probada se unirá al *Master*.
- 3) *Features*: orientada a las nuevas funcionalidades.
- 4) *Hotfix*: las correcciones de errores se encuentran aquí.

(Bitbucket, 2019)

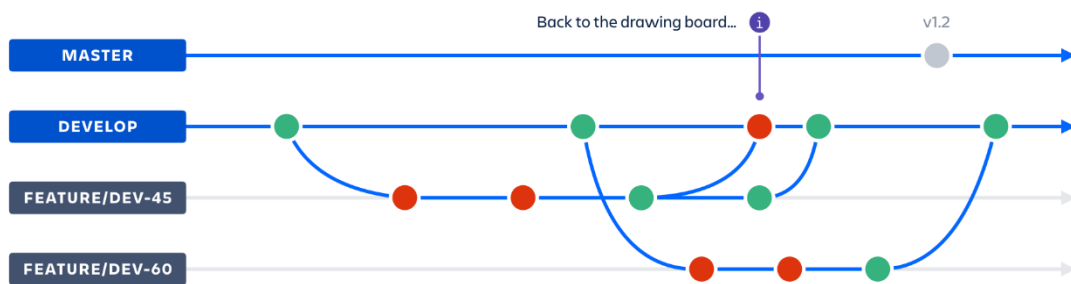


Figura 9: Flujo de trabajo entre ramas de *Git*.

Tomado de (Chubko, 2019)

El ciclo de vida de los archivos pasa por diferentes estados en los cuales se puede controlar los cambios que desean publicarse y el retroceso a alguna versión. En la siguiente imagen se puede observar su ciclo de vida en donde un archivo es editado para posteriormente pasar al área de *stage* donde comúnmente se encuentran todos los archivos editados referentes a la nueva funcionalidad y finalmente realizar un *commit* el cual tendrá un nombre y descripción de la nueva versión.

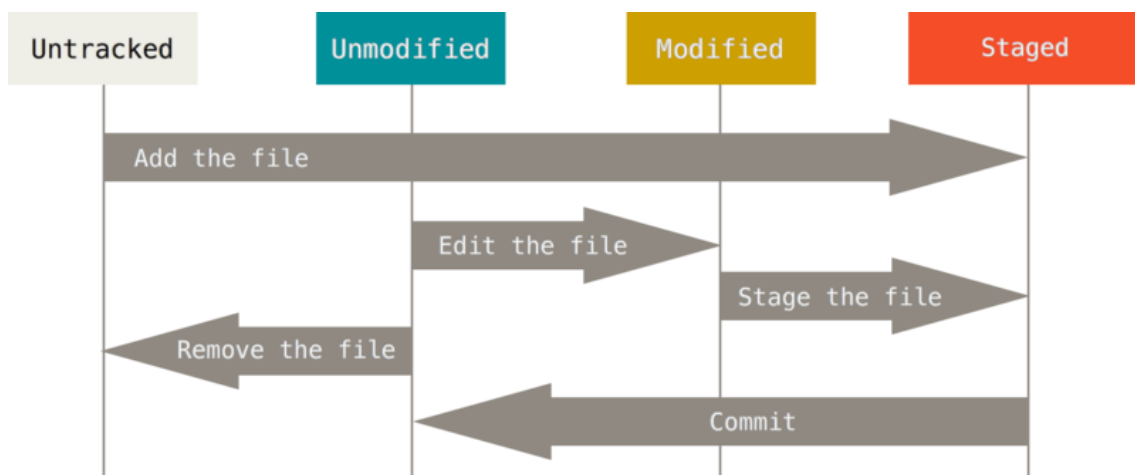


Figura 10: Flujo del guardado de datos en *Git*.

Tomado de (Git, 2019)

3. CAPÍTULO III. PROPUESTA DE SOLUCIÓN

Para el desarrollo del Sistema de Control Interno de *A Plus Computer* se ha considerado el esfuerzo y los puntos de historia de las entidades a desarrollar en cada módulo de *WordPress* y la presentación de datos en el panel de *BI* por lo tanto a continuación se presenta este análisis y planificación en base a la metodología *Scrum* dando como resultado tres *sprints* de dos semanas cada uno.

3.1 Product Backlog

Tabla 2.

Product Backlog.

ID	Nombre HU	Prioridad	Estimado
TIT01	Administrar Clientes	Media	2PH
TIT02	Administrar Técnicos	Media	2PH
TIT03	Administrar Dispositivos	Media	2PH
TIT04	Administrar Artículos	Media	2PH
TIT05	Administrar los Tickets	Alta	4PH
TIT06	Reportes personalizados de Tickets	Media	3PH
TIT07	Administrar Suscripciones	Alta	4PH
TIT08	Reportes personalizados de Suscripciones	Media	3PH
TIT09	Administrar Facturas	Alta	4PH
TIT10	Reportes personalizados de Facturas	Media	3PH
TIT11	Administrar Usuarios	Baja	1PH

Tabla 3.

Detailed Product Backlog.

ID	Descripción	Tareas	Criterios de Aceptación
TIT01	Como usuario quiero administrar los clientes de <i>A Plus Computer</i> para utilizar su información actual en procesos de negocio como facturación, tickets y suscripciones a servicios.	1) CRUD de Clientes	1) Debe poder leer los clientes. 2) Debe poder crear un cliente. 3) Debe poder modificar un cliente. 4) Debe poder eliminar un cliente. 5) Un cliente debe contener los campos: Cedula/RUC, Nombre, Dirección, Teléfono Fijo, Celular, Fecha Nacimiento, Correo y una imagen.
TIT02	Como usuario quiero administrar los técnicos de <i>A Plus Computer</i> para utilizar su información actual en procesos de negocio como tickets atendidos, suscripciones creadas y eventualmente creación de facturas.	1) CRUD de Técnicos	1) Debe poder leer los técnicos. 2) Debe poder crear un técnico. 3) Debe poder modificar un técnico. 4) Debe poder eliminar un técnico. 5) Un cliente debe contener los campos: Cédula, Nombre, Dirección, Teléfono, Celular, Fecha de Nacimiento, Correo Electrónico, Cargo y una imagen.

TIT03	Como usuario quiero administrar los dispositivos de los clientes para llevar un registro de sus datos como del historial.	1) CRUD de Dispositivos	<ol style="list-style-type: none"> 1) Debe poder leer los dispositivos. 2) Debe poder crear un dispositivo. 3) Debe poder modificar un dispositivo. 4) Debe poder eliminar un dispositivo. 5) Un cliente debe contener los campos: Serial, Tipo, Marca, Modelo, Sistema Operativo, <i>Mainboard</i>, Procesador, RAM, Almacenamiento, Fecha de Compra, Última Modificación, Persona de Contacto y una imagen. 6) Debe contener el cliente al que le corresponde este dispositivo.
TIT04	Como usuario quiero administrar los artículos para prevenir la falta de stock y conocer el dispositivo del cliente al que se instaló.	1) CRUD de Artículos	<ol style="list-style-type: none"> 1) Debe poder leer los artículos. 2) Debe poder crear un artículo. 3) Debe poder modificar un artículo. 4) Debe poder eliminar un artículo. 5) Un cliente debe contener los campos: Serial, Categoría, Marca, Modelo, Costo de Compra, Costo de Venta, Descripción y una imagen.

TIT05	Como usuario quiero administrar los tickets para llevar un control de los trabajos que los técnicos atienden sobre los dispositivos de los clientes.	<ol style="list-style-type: none"> 1) CRUD de Tickets 2) Vincular Técnicos 3) Vincular Clientes 4) Vincular Dispositivos 5) Vincular Artículos 	<ol style="list-style-type: none"> 1) Debe poder leer los tickets. 2) Debe poder crear un ticket. 3) Debe poder modificar un ticket. 4) Debe poder eliminar un ticket. 5) Un cliente debe contener los campos: Código, Descripción, Estado y una imagen. 6) Debe contener la referencia a los artículos utilizados, la suscripción implementada, el técnico que solventó el problema y el dispositivo atendido.
TIT06	Como usuario quiero obtener reportes personalizados de los tickets atendidos para identificar problemas y tomar decisiones de negocio.	<ol style="list-style-type: none"> 1) Crear <i>EndPoint</i> 2) Obtener información desde <i>Pentaho</i> 3) Presentar la información en <i>Power BI</i> 	<ol style="list-style-type: none"> 1) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los tickets. 2) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los clientes. 3) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los técnicos. 4) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los dispositivos. 5) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los artículos. 6) Debe guardar los datos en una base externa a la de <i>WordPress</i> con los

			<p>datos estandarizados por tablas.</p> <p>7) Debe generar reportes personalizables en <i>Power BI</i>.</p>
TIT07	<p>Como usuario quiero administrar las suscripciones para tener un control del vencimiento de las mismas y ofrecer al cliente la renovación con anterioridad.</p>	<ol style="list-style-type: none"> 1) CRUD de Suscripciones 2) Vincular Clientes 3) Vincular Dispositivos 4) Vincular Artículos 	<ol style="list-style-type: none"> 1) Debe poder leer las suscripciones. 2) Debe poder crear una suscripción. 3) Debe poder modificar una suscripción. 4) Debe poder eliminar una suscripción. 5) Un cliente debe contener los campos: Código, Fecha de Activación, Fecha de Caducidad, Fecha de Notificación, Descripción y una Imagen. 6) Debe contener la referencia del artículo a suscribirse y el dispositivo.
TIT08	<p>Como usuario quiero obtener reportes personalizados de las suscripciones para identificar problemas y tomar decisiones de negocio.</p>	<ol style="list-style-type: none"> 1) Crear <i>EndPoint</i> 2) Obtener información desde <i>Pentaho</i> 3) Presentar la información en <i>Power BI</i> 	<ol style="list-style-type: none"> 1) Debe existir un <i>endpoint</i> de <i>Web Service</i> para las suscripciones. 2) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los dispositivos. 3) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los artículos. 4) Debe guardar los datos en una base externa a la de <i>WordPress</i> con los

			<p>datos estandarizados por tablas.</p> <p>5) Debe generar reportes personalizables en <i>Power BI</i>.</p>
TIT09	<p>Como usuario quiero administrar las facturas para llevar una contabilidad detallada y registrar los ítems vendidos, servicios prestados y tickets atendidos en las facturas para controlar las garantías.</p>	<ol style="list-style-type: none"> 1) CRUD de Factura 2) Vincular Clientes 3) Vincular Tickets 4) Vincular Artículos 	<ol style="list-style-type: none"> 1) Debe poder leer las facturas. 2) Debe poder crear una factura. 3) Debe poder modificar una factura. 4) Debe poder eliminar una factura. 5) Un cliente debe contener los campos: Código, Subtotal, Descuento, IVA, Total y una Imagen. 6) Debe contener la referencia de los tickets facturados los cuales harán referencia a los artículos y suscripciones a cobrar.
TIT10	<p>Como usuario quiero obtener reportes personalizados de las facturas emitidas para identificar problemas y tomar decisiones de negocio.</p>	<ol style="list-style-type: none"> 1) Crear <i>EndPoint</i> 2) Obtener información desde <i>Pentaho</i> 3) Presentar la información en <i>Power BI</i> 	<ol style="list-style-type: none"> 1) Debe existir un <i>endpoint</i> de <i>Web Service</i> para las facturas. 2) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los tickets. 3) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los clientes. 4) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los técnicos.

			<p>5) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los dispositivos.</p> <p>6) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los artículos.</p> <p>7) Debe existir un <i>endpoint</i> de <i>Web Service</i> para las suscripciones.</p> <p>8) Debe guardar los datos en una base externa a la de <i>WordPress</i> con los datos estandarizados por tablas.</p> <p>9) Debe generar reportes personalizables en <i>Power BI</i>.</p>
TIT11	Como administrador quiero agregar usuarios para que puedan utilizar el sistema.	<p>1) Crear Roles</p> <p>2) Editar los permisos de acceso de los roles</p>	<p>1) Debe poder leer los usuarios.</p> <p>2) Debe poder crear un usuario.</p> <p>3) Debe poder modificar un usuario.</p> <p>4) Debe poder eliminar un usuario.</p> <p>5) Un cliente debe contener los campos: Nombre de Usuario, Email, Nombre, Apellido, Contraseña y Rol.</p> <p>6) Debe tener permisos para acceder como administrador a todo el sistema, como gerente a todos los módulos y como técnicos a todos los módulos tomando en cuenta que el módulo de facturas solo puede leer.</p>

3.2 Burn Down Charts

3.2.1 Sprint 1

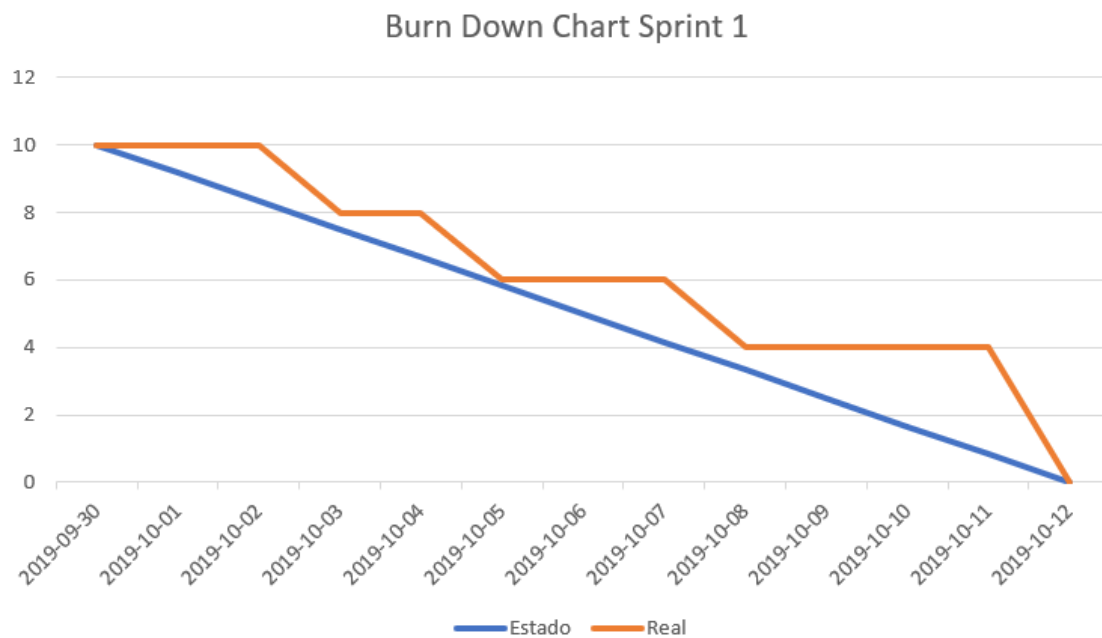


Figura 11: Burn Down Chart Sprint 1.

3.2.2 Sprint 2

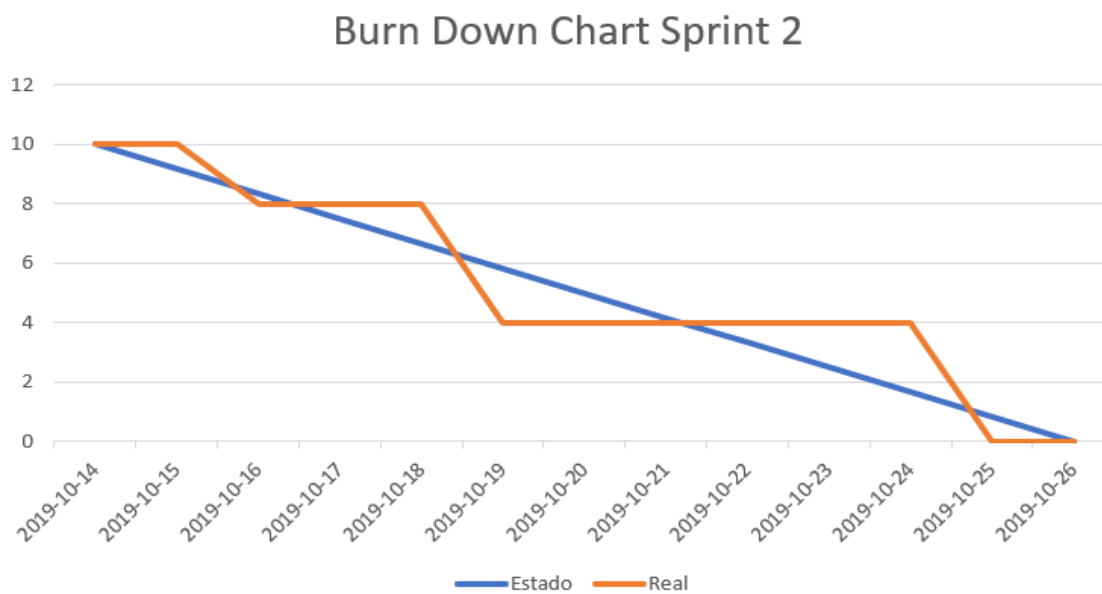


Figura 12: Burn Down Chart Sprint 2.

3.2.3 Sprint 3

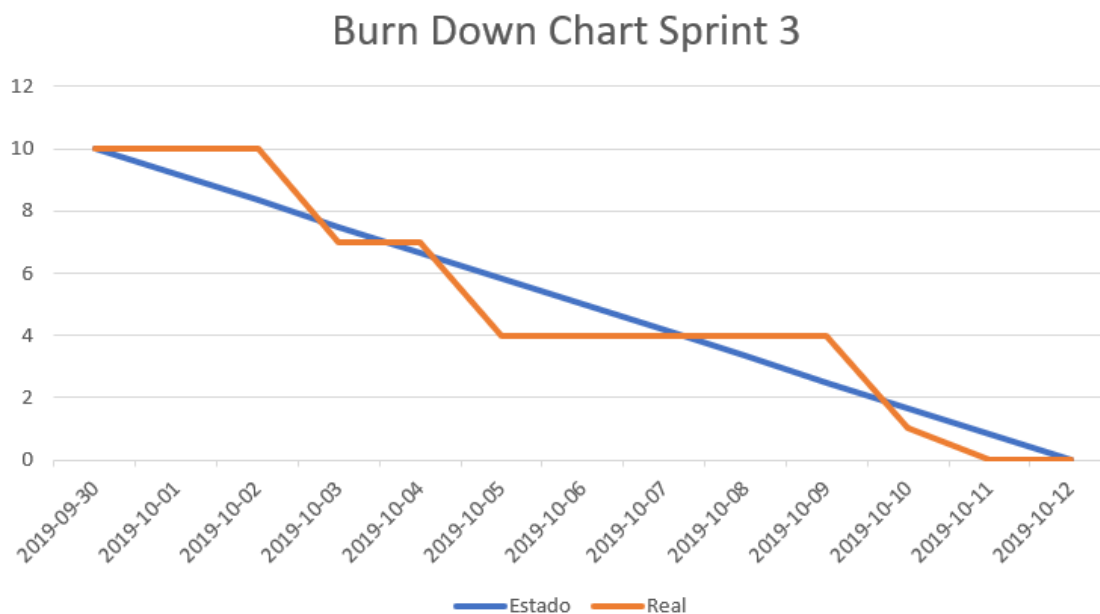


Figura 13: Burn Down Chart Sprint 3.

3.3 Configuración de entorno del sistema

A continuación, se muestran las consideraciones previas o configuración del entorno del sistema a desarrollar, a partir de aquí los recuadros rojos en las capturas de pantalla indican lo que se desea resaltar y los recuadros azules con fondo blanco ocultan información privada de *A Plus Computer*.

3.3.1 Dominio y Hosting

A Plus Computer tiene un contrato de dominio y hosting con el proveedor *GoDaddy*. El nombre de dominio es *aplusecuador.com* y el tipo de hosting es *Deluxe Linux Hosting with cPanel* el cual cuenta con sitios webs ilimitados, almacenamiento ilimitado y subdominios ilimitados.

A Plus Computer cuenta con una página web alojada en el dominio principal (*aplusecuador.com*) por lo que ahora se creará el sistema de control interno en un subdominio de la misma, para esto es necesario ingresar al *cPanel*, agregar subdominio, ingresar el nombre *sci.aplusecuador.com* (Sistema Control Interno)

e ingresar la carpeta donde contendrá la información como se muestra a continuación:

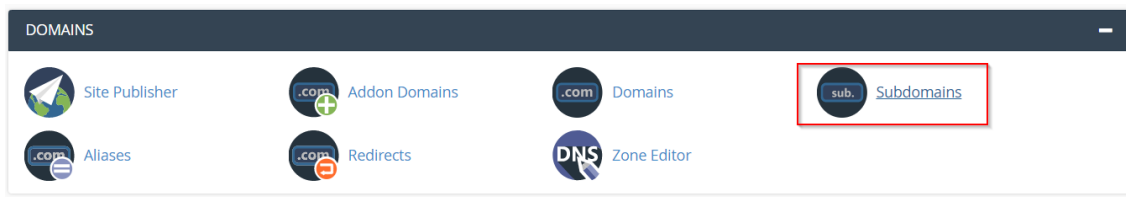


Figura 14: Link para administrar los subdominios.

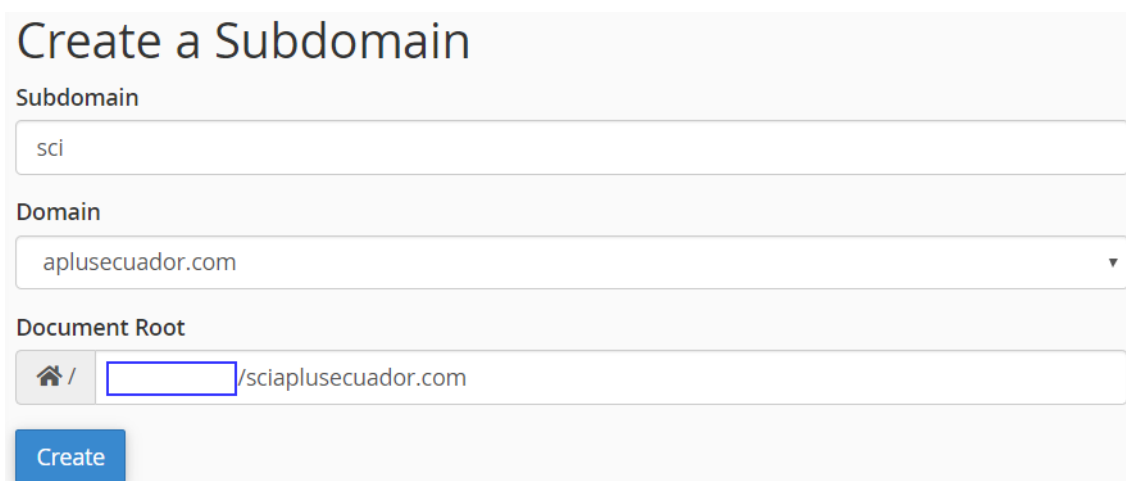
A screenshot of a 'Create a Subdomain' form. It has three input fields: 'Subdomain' with the value 'sci', 'Domain' with a dropdown menu showing 'aplusecuador.com', and 'Document Root' with a path '/sciaplusecuador.com'. A blue 'Create' button is at the bottom.

Figura 15: Crear el nuevo subdominio sci.aplusecuador.com.

Al dar clic en el botón *Create* tomará unos segundos en donde creará los registros para este nuevo subdominio y la carpeta que se especificó. Para confirmar la creación de esta carpeta es necesario ingresar al *File Manager* y navegar hasta la carpeta *scipluscomputer* especificada en la creación del subdominio, una vez allí ingresar en ella y observar que se ha creado la carpeta *cgi-bin* por defecto:

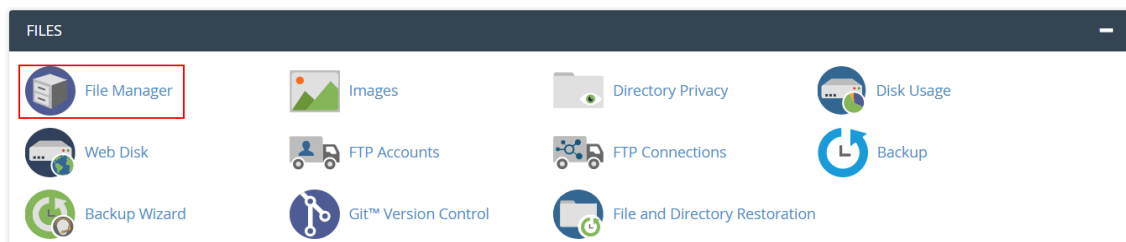


Figura 16: Link para administrar los archivos.

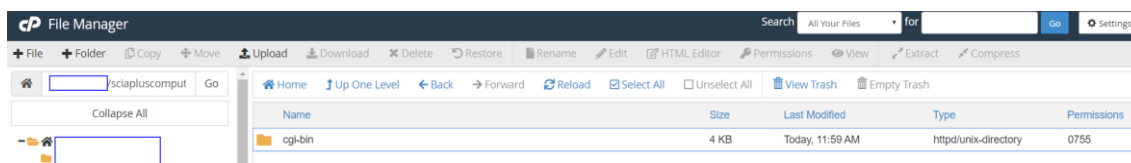


Figura 17: Carpeta creada para el dominio sci.apluscomputer.com.

En esta ruta (/scipluscomputer) se realizará la instalación de *WordPress* con la herramienta *Installatron*.

3.3.2 Instalación WordPress

Instalar *WordPress*, ingresar el dominio creado, confirmar la ruta de instalación, proporcionar un usuario, contraseña, email de administrador además de un título y subtítulo del sitio como se muestra a continuación:

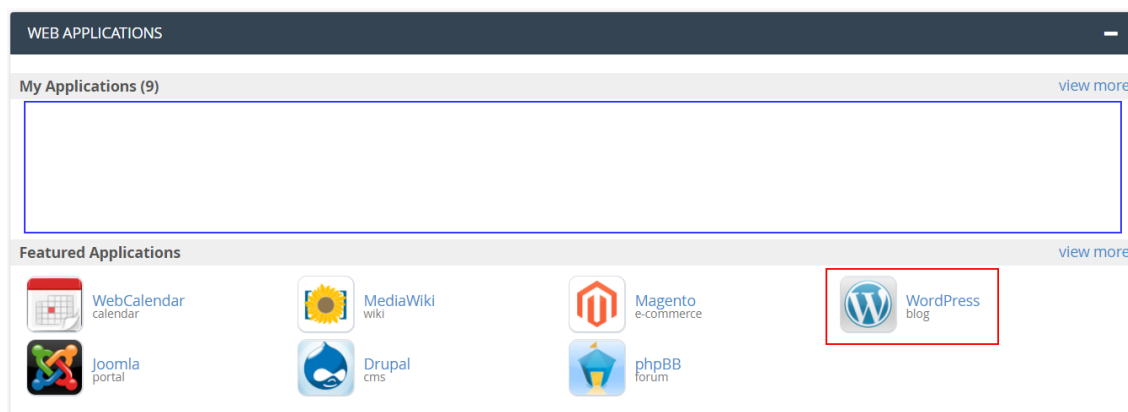


Figura 18: Link para administrar sitios *WordPress*.

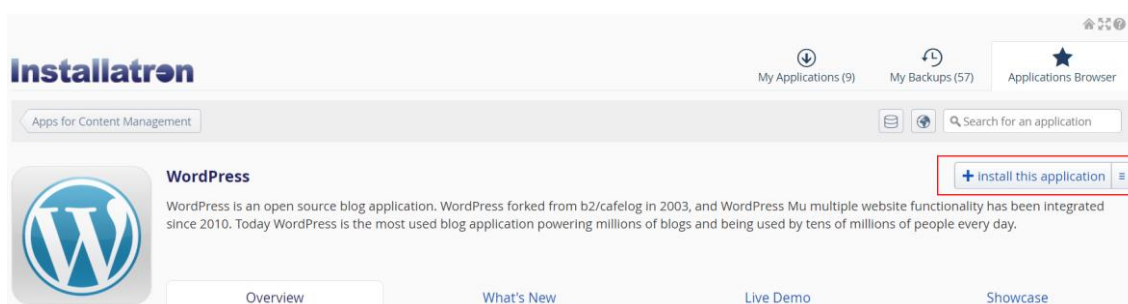


Figura 19: Botón para realizar una nueva instalación de *WordPress*.

Location

Please select a location to install WordPress.

The location is a combination of a domain and a directory that together determine where the files will be installed and where the application will be viewed in a web browser after installation.

Domain
http://sci.aplusecuador.com/

Directory (Optional)

The domain and directory selected above represent a URL (a web page link) and also a physical path on the server's hard drive, both of which are shown here:

http://sci.aplusecuador.com/
/csiapluscomputer/

Figura 20: Escoger el dominio sobre el cual se realizará la instalación.

Settings

These settings allow you to customize and personalize this installed application.

The values set can be modified later within installation or within the installed application.

Administrator Username

Administrator Password

Strength: Strong [Show Password](#) [Generate](#)

Administrator Email

Website Title

Sistema de Control Interno

Website Tagline

A Plus Computer

Figura 21: Ingresar datos de usuario administrador y sitio.

Posteriormente se debe ingresar al sitio para continuar con la instalación interna de *WordPress* la cual solicita el tipo de negocio, si se desea agregar perfiles de redes sociales y finalmente la elección del tema, este último no tiene importancia ya que el sistema de control interno se basa en el área administrativa de *WordPress*.

Sistema de Control Interno

sci.aplusecuador.com/wp-admin/

wordpress.org/support/

5.3 version

50.4MB files

1MB database

2019-11-24 installed

2019-11-24 updated

0 backups

Figura 22: Instalación finalizada. Link para configurar el sitio *WordPress*.

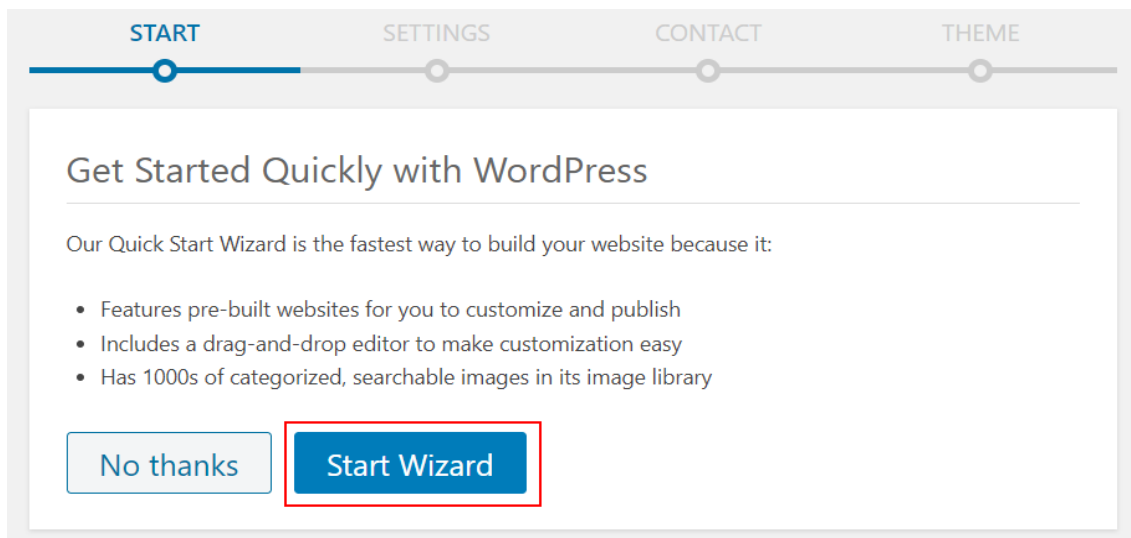


Figura 23: Inicio de la configuración del sitio.

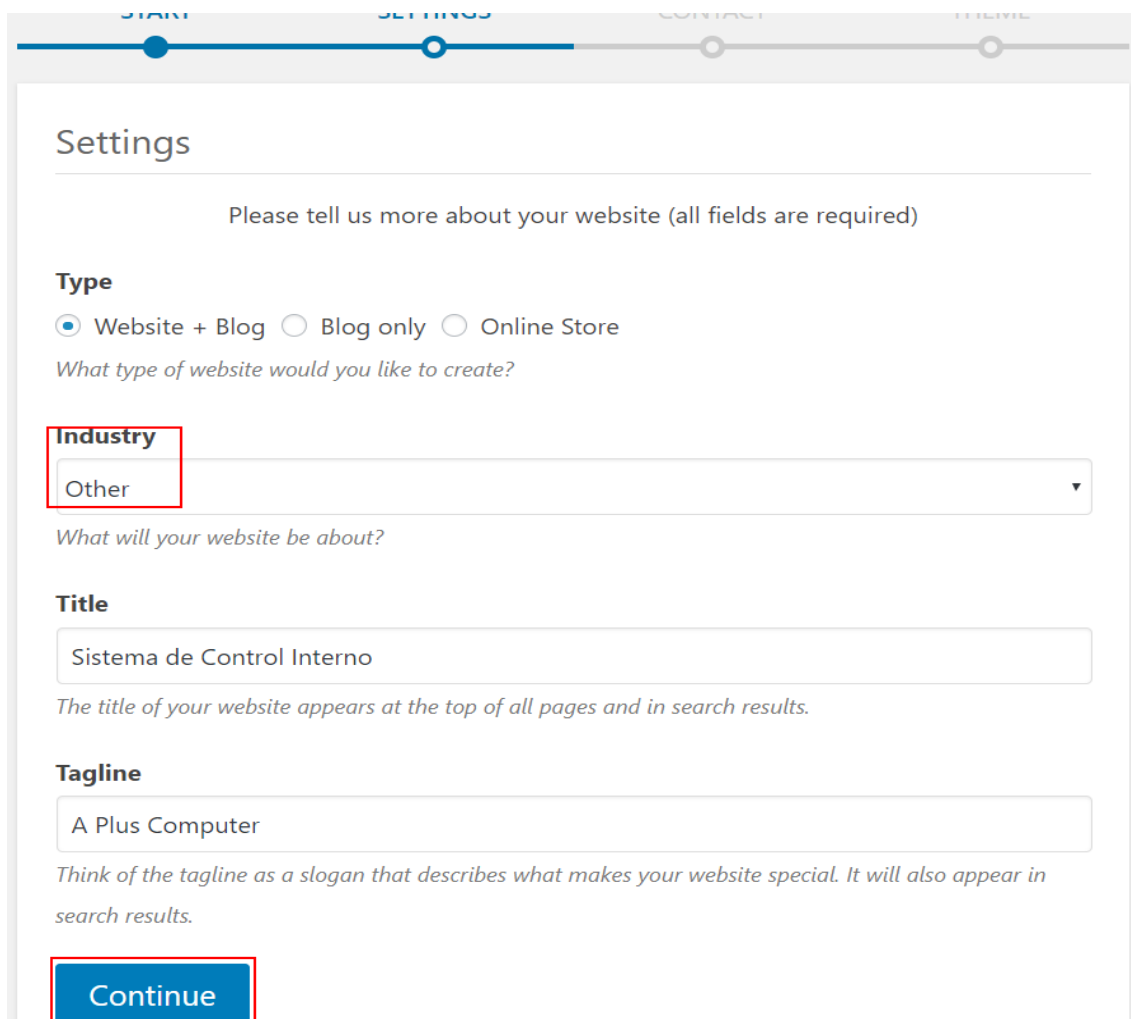


Figura 24: Ingresar el tipo, industria, título y eslogan.

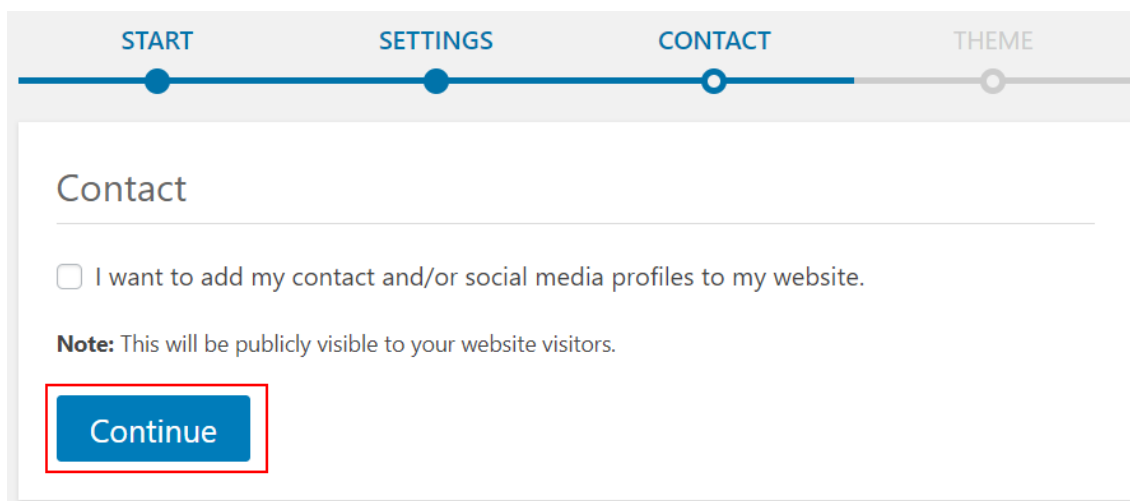


Figura 25: Seleccionar si se desea agregar perfiles sociales.

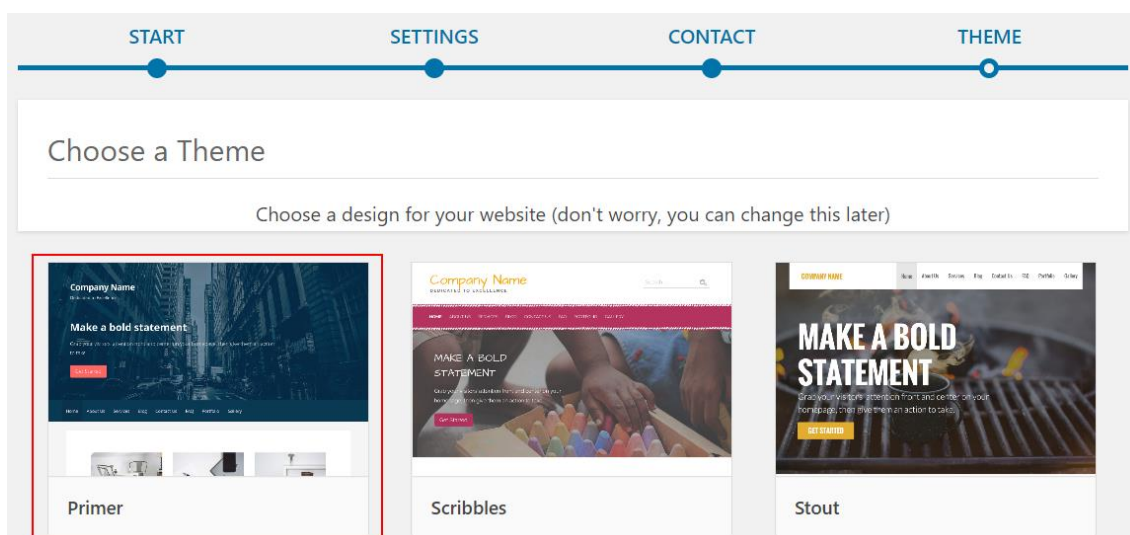


Figura 26: Elegir el tema.

Una vez terminada la instalación del tema es posible ingresar al panel administrativo del sitio a través de sci.aplusecuador.com/wp-admin/ como se muestra a continuación:

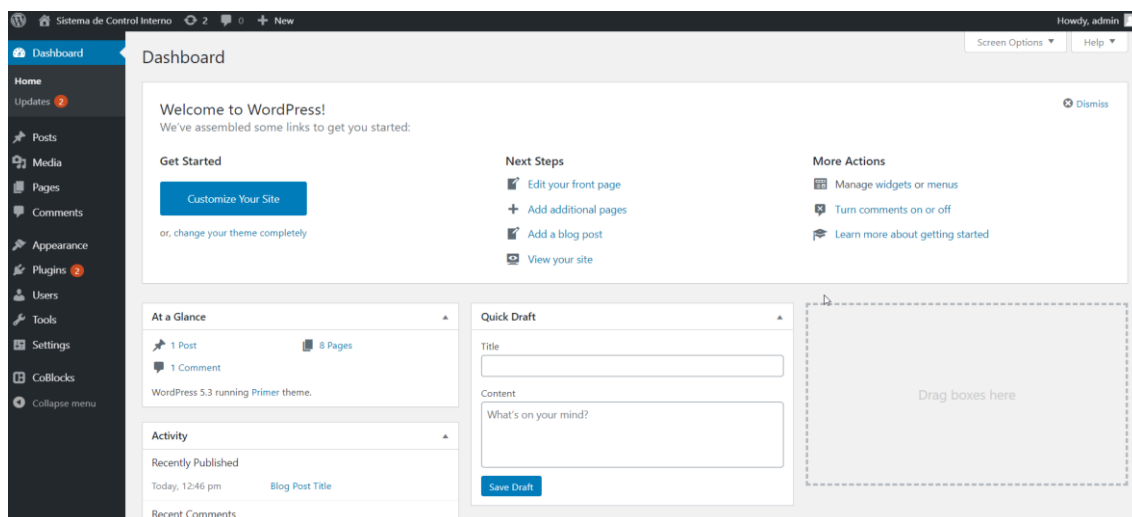


Figura 27: Dashboard del sitio “Sistema de Control Interno”.

Para dejar muy simple y limpio el sitio es necesario realizar lo siguiente:

1. *Plugins*

1.1. Eliminar todos.

1.2. Agregar únicamente *Classic Editor*.

2. *Temas*

2.1. Activar el tema *Twenty Seventeen*.

2.2. Eliminar los demás.

3. *Pages*

3.1. Únicamente dejar la página llamada *Home*.

4. *Posts*

4.1. Eliminar todo.

5. *Media Library*

5.1. Eliminar todo.

6. Actualizar los *Permanlinks*.

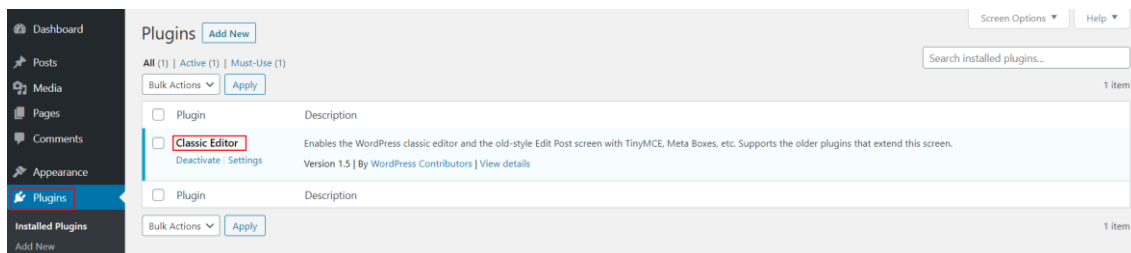


Figura 28: En plugins únicamente se mantendrá *Classic Editor*.

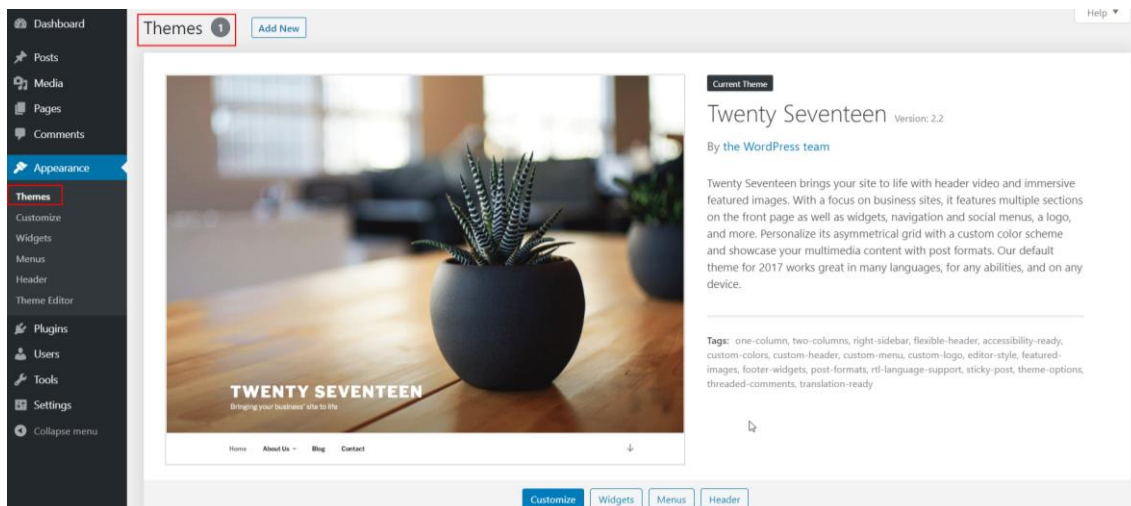


Figura 29: En temas únicamente se mantendrá *Twenty Seventeen*.

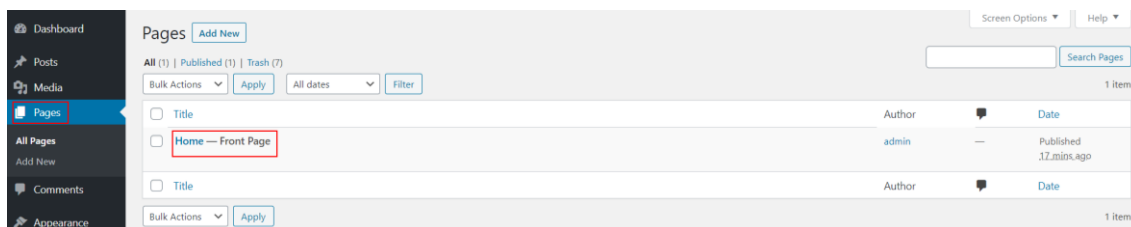


Figura 30: En páginas únicamente se mantendrá *Home*.

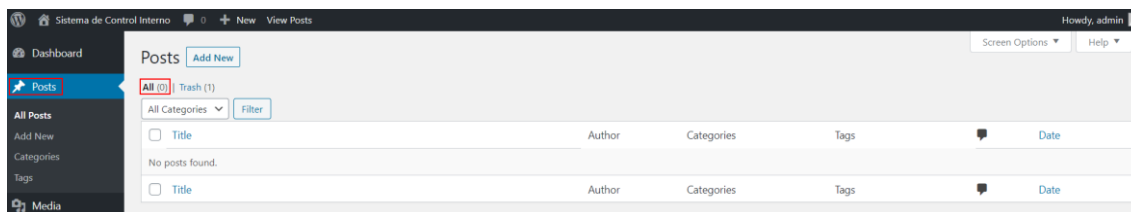


Figura 31: En posts únicamente se dejará vacío.

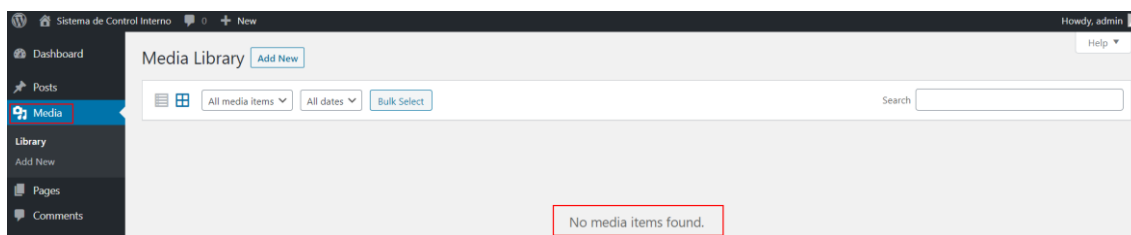


Figura 32: En media únicamente se dejará vacío.

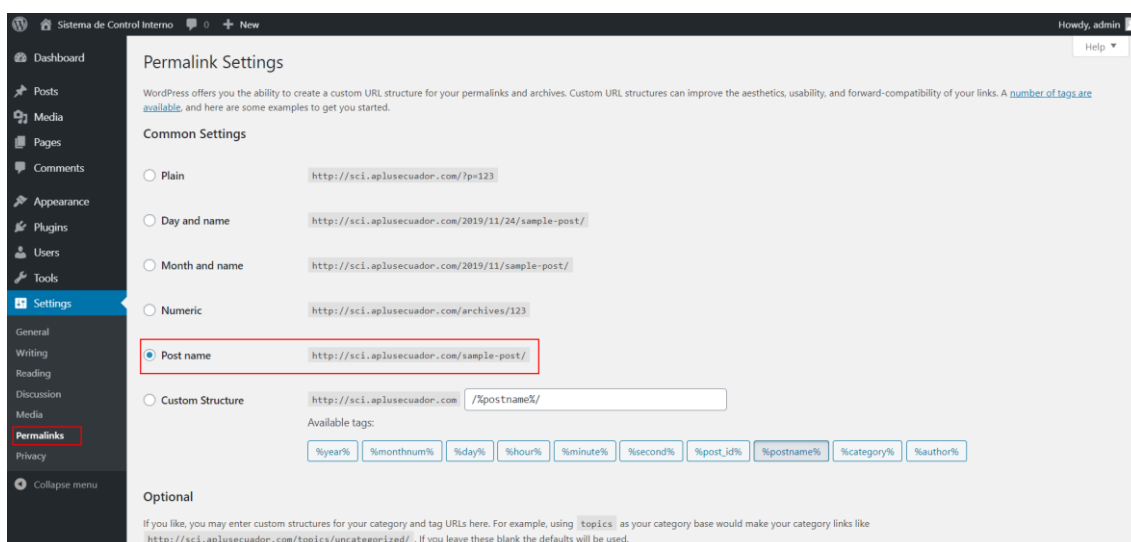


Figura 33: Se seleccionará el tipo de *Permantlinks*.

3.3.3 Cuenta FTP

Para la transferencia de archivos es necesario crear una cuenta *FTP* y acceder desde *FileZilla*. Para esto dirigirse a *FTP Accounts* en *cPanel* e ingresar credenciales de conexión para la carpeta */scipluscomputer*.

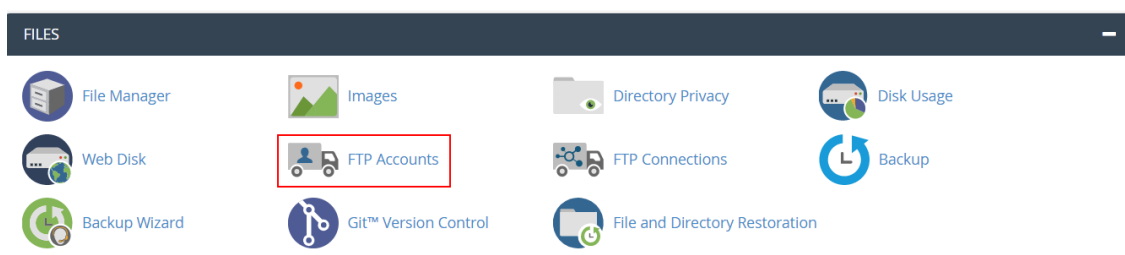


Figura 34: Link para administrar cuentas *FTP*.

Add FTP Account

Log In

Domain

Password

Password (Again)

Strength ?

Very Strong (100/100)

Directory

/home//scipluscomputer

Quota

2000 MB

Unlimited

Figura 35: Conexión de la cuenta *FTP* para sci.apluscomputer.com.

Finalmente acceder a los archivos desde la herramienta *FileZilla* (filezilla-project.org/) y editarlos con la herramienta *SublimeText 3* (sublimetext.com/3).

Remote site: /wp-content

/

- cgi-bin
- wp-admin
- wp-content
- wp-includes

Filename	Filesize	Filetype	Last modifi...	Permissi...	Owner/G...
..					
languages		File folder	11/24/2019...	0755	4248647...
mu-plugins		File folder	11/24/2019...	0755	4248647...
plugins		File folder	11/24/2019...	0755	4248647...
themes		File folder	11/24/2019...	0755	4248647...
upgrade		File folder	11/24/2019...	0755	4248647...
uploads		File folder	11/24/2019...	0755	4248647...
index.php	28	PHP File	1/8/2012 1...	0644	4248647...

Figura 36: Conexión exitosa con *FileZilla* utilizando la cuenta *FTP* creada.

3.3.4 Repositorio Git

El repositorio de este código se encuentra en *Git* en el siguiente link:

github.com/dabarlea/sciaplusecuador

3.4 Consideraciones previas al desarrollo

Antes de comenzar, es necesario determinar las entidades que dependen de otras, así como los atributos, su tipo, taxonomías y/o *Custom Post Type* al que hace referencia. Para visualizar estos datos se crearán *Metaboxes* los cuales tendrán *Metafields* que serán creados gracias a las librerías de CMB2, el repositorio de este es el siguiente github.com/CMB2/CMB2.

Tabla 4.

Estructura de la entidad Cliente.

Entidad	Atributo	Tipo de Entrada	Tipo de Dato
Cliente	Cédula/RUC	Texto	Simple
	Nombre Empresa	Texto	Simple
	Nombre	Texto	Simple
	Apellido	Texto	Simple
	Dirección	Texto	Simple
	Teléfono	Texto	Simple
	Celular	Texto	Simple
	Fecha Nacimiento	Date	Simple
	<i>Email</i>	<i>Email</i>	Simple

Tabla 5.

Estructura de la entidad Técnico.

Entidad	Atributo	Tipo de Entrada	Taxonomía
Técnico	Cédula	Texto	Simple
	Nombre	Texto	Simple
	Apellido	Texto	Simple
	Dirección	Texto	Simple
	Teléfono	Texto	Simple
	Celular	Texto	Simple
	Fecha Nacimiento	<i>Date</i>	Simple
	<i>Email</i>	<i>Email</i>	Simple
	Cargo	<i>Combo Box</i>	Taxonomía Jerárquica

Tabla 6.

Estructura de la entidad Dispositivo.

Entidad	Atributo	Tipo de Entrada	Taxonomía
Dispositivo	Serial	Texto	Simple
	Tipo	<i>Combo Box</i>	Taxonomía Jerárquica
	Marca	<i>Combo Box</i>	Taxonomía No Jerárquica
	Modelo	Texto	Simple
	Sistema Operativo	<i>Combo Box</i>	Taxonomía Jerárquica
	<i>MainBoard</i>	Texto	Simple

	Procesador	Texto	Simple
	RAM	Texto	Simple
	Almacenamiento	Texto	Simple
	Fecha de Compra	<i>Date</i>	Simple
	Última Modificación	<i>Date</i>	Simple
	Persona de Contacto	Texto	Simple
	ID Cliente	<i>Combo Box</i>	<i>Custom Post Type (Cliente)</i>

Tabla 7.

Estructura de la entidad Artículo.

Entidad	Atributo	Tipo de Entrada	Taxonomía
Artículo	Serial	Texto	Simple
	Categoría	<i>Combo Box</i>	Taxonomía Jerárquica
	Marca	<i>Combo Box</i>	Taxonomía No Jerárquica
	Modelo	Texto	Simple
	Costo Compra	<i>Money</i>	Simple
	Costo Venta	<i>Money</i>	Simple
	Descripción	Texto	Simple

Tabla 8.

Estructura de la entidad Ticket.

Entidad	Atributo	Tipo de Entrada	Taxonomía
---------	----------	-----------------	-----------

Tickets	Código	Texto	Simple
	ID Cliente	<i>Combo Box</i>	<i>Custom Post Type (Cliente)</i>
	ID Técnico	<i>Combo Box</i>	<i>Custom Post Type (Técnico)</i>
	ID Dispositivo	<i>Combo Box</i>	<i>Custom Post Type (Dispositivo)</i>
	Artículos Usados	<i>Combo Box</i>	<i>Custom Post Type (Artículo)</i>
	Descripción	Texto	Simple
	Estado	Texto	Simple

Tabla 9.

Estructura de la entidad Suscripción.

Entidad	Atributo	Tipo de Entrada	Taxonomía
Suscripción	Código	Texto	Simple
	ID Cliente	<i>Combo Box</i>	<i>Custom Post Type (Cliente)</i>
	ID Artículo	<i>Combo Box</i>	<i>Custom Post Type (Artículo)</i>
	ID Dispositivo	<i>Combo Box</i>	<i>Custom Post Type (Dispositivo)</i>
	Fecha de Activación	<i>Date</i>	Simple
	Fecha de Caducidad	<i>Date</i>	Simple
	Fecha de Notificación	<i>Date</i>	Simple
	Descripción	Texto	Simple

Tabla 10.

Estructura de la entidad Factura.

Entidad	Atributo	Tipo de Entrada	Taxonomía
Facturación	Código	Texto	Simple
	Empresa	Texto	Simple
	ID Cliente	<i>Combo Box</i>	<i>Custom Post Type (Cliente)</i>
	ID Tickets	<i>Combo Box</i>	<i>Custom Post Type (Tickets)</i>
	ID Artículos	<i>Combo Box</i>	<i>Custom Post Type (Artículo)</i>
	Subtotal	<i>Money</i>	Simple
	Descuento	<i>Money</i>	Simple
	IVA	<i>Money</i>	Simple
	Total	<i>Money</i>	Simple

En la Tabla 10, se presentan 3 *sprints* con 10PH cada uno divididos de la siguiente forma tomando en cuenta que las entidades Tickets, Suscripciones y Facturación necesitan de Clientes, Técnicos, Artículos y Dispositivos respectivamente:

Tabla 11.

Planificación de sprints.

<i>Sprint</i>	Historia de Usuario	Puntos de Historia
<i>Sprint 1</i>	Administrar Clientes	2
	Administrar Dispositivos	2

	Administrar Artículos	2
	Administrar Suscripciones	4
<i>Sprint 2</i>	Administrar Técnicos	2
	Administrar Tickets	4
	Administrar Facturas	4
<i>Sprint 3</i>	Reporte Tickets	3
	Reporte Suscripciones	3
	Reporte Facturas	3
	Administrar Usuarios	1

Cada módulo estará representado por un *plugin* el cual contendrá una o varias entidades como se muestra en la Tabla 12:

Tabla 12.

Organización de los plugins según sus entidades.

<i>Plugin</i>	Entidades	Explicación
Prerrequisitos	Clientes	Los módulos necesitarán que estas entidades se encuentren creadas para utilizarlas.
	Técnicos	
	Dispositivos	
	Artículos	
Módulo Suscripciones	Suscripciones	Únicamente Suscripciones
Módulo Tickets	Tickets	Únicamente Tickets
Módulo Facturas	Facturas	Únicamente Facturas

Nota: Se mostrará el desarrollo de las entidades Artículos y Suscripciones debido a la extensa explicación que esto implica; sin embargo, las siguientes entidades son desarrolladas implementando los mismos conceptos. De la misma forma

para la extracción, manipulación y despliegue de datos en *Power BI* se presentará con las mismas entidades.

3.5 Desarrollo / Sprints

3.5.1 Sprint 1

3.5.1.1 Sprint Planning

Los objetivos para este *sprint* consisten en la creación de entidades básicas necesarias para la creación de una más compleja como lo es Suscripciones, esta podrá ser probada e implementada si así se lo desea.

3.5.1.2 Sprint Backlog

Tabla 13.

Sprint Backlog del Spring 1.

ID	Nombre HU	Descripción	Responsable	Esfuerzo
TIT01	Administrar Clientes	Como usuario quiero administrar los clientes para utilizar su información actual en procesos de negocio como facturación, tickets y suscripciones a servicios.	David Escobar	2PH
TIT03	Administrar Dispositivos	Como usuario quiero administrar los dispositivos de los clientes para llevar un registro de sus datos como del historial.	David Escobar	2PH
TIT04	Administrar Artículos	Como usuario quiero administrar los artículos para prevenir la falta de stock y conocer el dispositivo del cliente al que se instaló.	David Escobar	2PH
TIT07	Administrar Suscripciones	Como usuario quiero administrar las suscripciones para tener un control del vencimiento de las mismas y ofrecer al cliente la renovación con anterioridad.	David Escobar	4PH

3.5.1.3 Historias de Usuario

Tabla 14.

Historia de usuario TIT01.

Historia de usuario	
ID: TIT01	Usuario: Usuario
Nombre de historia: Administrar Clientes	
Prioridad en negocio: Media	Responsable: David Escobar
Descripción: Como usuario quiero administrar los clientes de <i>A Plus Computer</i> para utilizar su información actual en procesos de negocio como facturación, tickets y suscripciones a servicios.	
Criterios de aceptación:	
<ol style="list-style-type: none"> 1) Debe poder leer los clientes. 2) Debe poder crear un cliente. 3) Debe poder modificar un cliente. 4) Debe poder eliminar un cliente. 5) Un cliente debe contener los campos: Cedula/RUC, Nombre, Dirección, Teléfono Fijo, Celular, Fecha Nacimiento, Correo y una imagen. 	

Tabla 15.

Historia de usuario TIT03.

Historia de usuario	
ID: TIT03	Usuario: Usuario
Nombre de historia: Administrar Dispositivos	
Prioridad en negocio: Media	Responsable: David Escobar

Descripción: Como usuario quiero administrar los dispositivos de los clientes para llevar un registro de sus datos como del historial.

Criterios de aceptación:

- 1) Debe poder leer los dispositivos.
- 2) Debe poder crear un dispositivo.
- 3) Debe poder modificar un dispositivo.
- 4) Debe poder eliminar un dispositivo.
- 5) Un cliente debe contener los campos: Serial, Tipo, Marca, Modelo, Sistema Operativo, *Mainboard*, Procesador, RAM, Almacenamiento, Fecha de Compra, Última Modificación, Persona de Contacto y una imagen.
- 6) Debe contener el cliente al que le corresponde este dispositivo.

Tabla 16.

Historia de usuario TIT04.

Historia de usuario	
ID: TIT04	Usuario: Usuario
Nombre de historia: Administrar Artículos	
Prioridad en negocio: Media	Responsable: David Escobar
Descripción: Como usuario quiero administrar los artículos para prevenir la falta de stock y conocer el dispositivo del cliente al que se instaló.	
Criterios de aceptación:	
<ol style="list-style-type: none"> 1) Debe poder leer los artículos. 2) Debe poder crear un artículo. 3) Debe poder modificar un artículo. 	

- 4) Debe poder eliminar un artículo.
- 5) Un cliente debe contener los campos: Serial, Categoría, Marca, Modelo, Costo de Compra, Costo de Venta, Descripción y una imagen.

Tabla 17.

Historia de usuario TIT07.

Historia de usuario	
ID: TIT07	Usuario: Usuario
Nombre de historia: Administrar Suscripciones	
Prioridad en negocio: Alta	Responsable: David Escobar
Descripción: Como usuario quiero administrar las suscripciones para tener un control del vencimiento de las mismas y ofrecer al cliente la renovación con anterioridad.	
Criterios de aceptación:	
<ol style="list-style-type: none"> 1) Debe poder leer las suscripciones. 2) Debe poder crear una suscripción. 3) Debe poder modificar una suscripción. 4) Debe poder eliminar una suscripción. 5) Un cliente debe contener los campos: Código, Fecha de Activación, Fecha de Caducidad, Fecha de Notificación, Descripción y una Imagen. 6) Debe contener la referencia del artículo a suscribirse y el dispositivo. 	

3.5.1.4 Desarrollo

Para la creación de la entidad Artículos es necesario conectarse utilizando *FileZilla* por la cuenta *FTP* previamente creada en *cPanel*, una vez allí se navegará hasta la carpeta */wp-content/plugins* y se creará la siguiente estructura de archivos y carpetas:

1. Prerrequisitos (carpeta)

1.1. *backend* (carpeta)

1.1.1. *articulo* (carpeta)

1.1.1.1. *cpt_articulo.php* (archivo *php*)

1.1.1.2. *mtx_articulo.php* (archivo *php*)

1.1.1.3. *tax_articulo_categoria.php* (archivo *php*)

1.1.2. *tax_articulo_marca.php* (archivo *php*)

1.2. *prerrequisitos.php* (archivo *php*)

Remote site: /wp-content/plugins/Prerrequisitos/backend/articulo

Filename	Filesize	Filetype	Last modifi...	Permissi...	Owner/G...
..					
cpt_articulo.php	2,959	PHP File	12/6/2019 ...	0644	4248647...
mtx_articulo.php	2,461	PHP File	12/6/2019 ...	0644	4248647...
tax_articulo_categoria.php	2,200	PHP File	12/6/2019 ...	0644	4248647...
tax_articulo_marca.php	2,091	PHP File	12/6/2019 ...	0644	4248647...

Figura 37: Estructura de Artículo.

Para un desarrollo más rápido se sugiere descargar el directorio Prerrequisitos en la computadora local, editar los archivos con *Sublime Text 3* y luego transferirlos (sobrescribir) al servidor. En la Figura 38 se muestra la estructura en local:

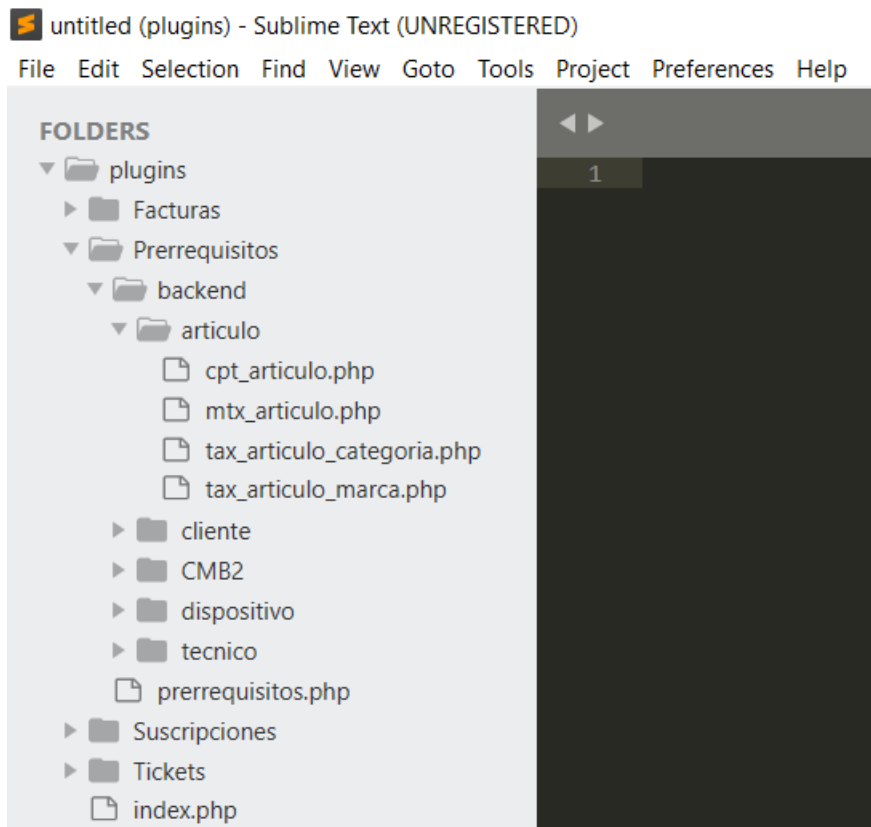


Figura 38: Estructura de Artículos en *Sublime Text 3* localmente.

Primero es necesario editar el archivo *prerrequisitos.php* el cual indicará el nombre, descripción, autor, versión entre otras características que se pueden agregar sobre el *plugin*. Luego se llama a los archivos requeridos y finalmente se agrega una función para eliminar el *metabox* creado automáticamente en los *Custom Post Types* cuando se crean taxonomías. (Esto se realiza ya que por defecto son de tipo *check list* y se necesita *drop down list*)

```

prerrequisitos.php x
1 <?php
2 /*
3 Plugin Name: Prerrequisitos SCI A Plus Computer
4 Plugin URI: http://wordpress.org/
5 Description: Este plugin contiene las entidades necesarias para la implementación de los
6 módulos del Sistema de Control Interno de A Plus Computer.
7 Author: David Escobar
8 Version: 1.1
9 */
10 /* Clientes */
11 require_once "backend/cliente/cpt_cliente.php" ;
12 require_once "backend/cliente/mtx_cliente.php" ;
13
14 /* Técnicos */
15 require_once "backend/tecnico/cpt_tecnico.php" ;
16 require_once "backend/tecnico/mtx_tecnico.php" ;
17
18 require_once "backend/tecnico/tax_tecnico_cargo.php" ;
19
20
21 function remove_post_custofields_tecnico() {
22     remove_meta_box( 'cargodiv' , 'tecnico' , 'side' );
23 }
24 add_action( 'admin_menu' , 'remove_post_custofields_tecnico' );
25
26
27 /* Artículos */
28 require_once "backend/articulo/cpt_articulo.php" ;
29 require_once "backend/articulo/mtx_articulo.php" ;
30
31 require_once "backend/articulo/tax_articulo_categoria.php" ;
32 require_once "backend/articulo/tax_articulo_marca.php" ;
33
34 function remove_post_custofields_articulo() {
35     remove_meta_box( 'categoriadiv' , 'articulo' , 'side' );
36     remove_meta_box( 'tagsdiv-marca' , 'articulo' , 'side' );
37 }
38 add_action( 'admin_menu' , 'remove_post_custofields_articulo' );
39
40 /* Dispositivos */

```

Figura 39: Archivo *prerrequisitos.php*.

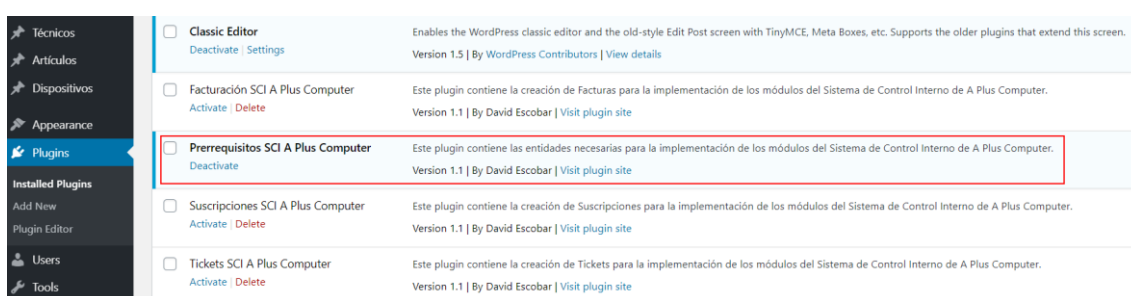


Figura 40: Plugin activado desde el *dashboard* del sitio.

Luego se debe editar el *Custom Post Type cpt_articulo.php* el contendrá una función que indicará el comportamiento de la entidad y su representación visual. Estas características se detallan a continuación:

Tabla 18.

Atributos de Custom Post Type.

<i>'label'</i>	Nombre en singular y plural.
<i>'description'</i>	Descripción corta.
<i>'labels'</i>	Hace referencia a todos los nombres que se desplegarán durante el flujo del <i>CPT</i> .
<i>'supports'</i>	Campos por defecto que <i>WordPress</i> tiene y se los puede agregar.
<i>'hierarchical'</i>	Si el <i>CPT</i> tiene jerarquía. Ejemplo: Noticias, Noticias de Deportes.
<i>'public'</i>	Si todos los usuarios pueden accederla.
<i>'show_ui'</i>	Si se muestra en la interfaz gráfica (Pueden existir entidades que no es necesario que el usuario pueda manipularlas).
<i>'show_in_menu'</i>	Si se muestra en el menú principal.
<i>'menu_position'</i>	Posición donde se desplegará en el menú.
<i>'show_in_admin_bar'</i>	Mostrarlo en la barra de administración.
<i>'show_in_nav_menus'</i>	Mostrar en la barra de navegación.
<i>'can_export'</i>	Permite exportar la información.
<i>'has_archive'</i>	Si tiene una página que agrupa todos los <i>CPT</i> de este tipo.
<i>'exclude_from_search'</i>	Excluir de los resultados de búsqueda del sistema
<i>'publicly_queryable'</i>	Si es posible realizar consultas sobre este <i>CPT</i> .
<i>'capability_type'</i>	El tipo de <i>Post</i> .
<i>'show_in_rest'</i>	Habilitar el <i>endpoint</i> de <i>API Rest</i> . (Importante)
<i>'rest_base'</i>	El nombre del <i>endpoint</i> de la <i>API Rest</i> .

```

1 <?php
2 if ( ! function_exists('cpt_articulos') ) {
3
4 // Register Custom Post Type
5 function cpt_articulos() {
6
7     $labels = array(
8         'name' => _x( 'Artículos', 'Post Type General Name', 'articulos' ),
9         'singular_name' => _x( 'Artículo', 'Post Type Singular Name', 'articulos' ),
10        'menu_name' => __( 'Artículos', 'articulos' ),
11        'name_admin_bar' => __( 'Artículo', 'articulos' ),
12        'archives' => __( 'Archivos de Artículo', 'articulos' ),
13        'attributes' => __( 'Atributos del Artículo', 'articulos' ),
14        'parent_item_colon' => __( 'Artículo Padre:', 'articulos' ),
15        'all_items' => __( 'Todos los Artículos', 'articulos' ),
16        'add_new_item' => __( 'Agregar Nuevo Artículo', 'articulos' ),
17        'add_new' => __( 'Agregar Nuevo', 'articulos' ),
18        'new_item' => __( 'Nuevo Artículo', 'articulos' ),
19        'edit_item' => __( 'Editar Artículo', 'articulos' ),
20        'update_item' => __( 'Actualizar Artículo', 'articulos' ),
21        'view_item' => __( 'Consultar Artículo', 'articulos' ),
22        'view_items' => __( 'Consultar Artículos', 'articulos' ),
23        'search_items' => __( 'Buscar Artículo', 'articulos' ),
24        'not_found' => __( 'No encontrado', 'articulos' ),
25        'not_found_in_trash' => __( 'No encontrado en la basura', 'articulos' ),
26        'featured_image' => __( 'Imágen Destacada', 'articulos' ),
27        'set_featured_image' => __( 'Establecer Imágen Destacada', 'articulos' ),
28        'remove_featured_image' => __( 'Eliminar Imágen Destacada', 'articulos' ),
29        'use_featured_image' => __( 'Usar como Imágen Destacada', 'articulos' ),
30        'insert_into_item' => __( 'Insertar en Artículo', 'articulos' ),
31        'uploaded_to_this_item' => __( 'Cargado a este Artículo', 'articulos' ),
32        'items_list' => __( 'Lista de Artículos', 'articulos' ),
33        'items_list_navigation' => __( 'Navegación en lista de Artículos', 'articulos' ),
34        'filter_items_list' => __( 'Lista de posiciones de filtro', 'articulos' ),
35    );
36    $args = array(
37        'label' => __( 'Artículo', 'articulos' ),

```

Figura 41: Archivo `cpt_articulo.php`.

```

36    $args = array(
37        'label' => __( 'Artículo', 'articulos' ),
38        'description' => __( 'Artículos de la empresa', 'articulos' ),
39        'labels' => $labels,
40        'supports' => array( 'title', 'thumbnail' ),
41        'hierarchical' => false,
42        'public' => true,
43        'show_ui' => true,
44        'show_in_menu' => true,
45        'menu_position' => 26,
46        'show_in_admin_bar' => true,
47        'show_in_nav_menus' => true,
48        'can_export' => true,
49        'has_archive' => true,
50        'exclude_from_search' => false,
51        'publicly_queryable' => true,
52        'capability_type' => 'post',
53        'show_in_rest' => true,
54        'rest_base' => 'articulosrest',
55    );
56    register_post_type( 'articulo', $args );
57
58 }
59 add_action( 'init', 'cpt_articulos', 0 );
60
61 }
62

```

Figura 42: Archivo `cpt_articulo.php`.

Posteriormente se identifica las Taxonomías de este *Custom Post Type*, para el caso de los artículos son categorías y marcas. Para el caso de la taxonomía *tax_articulo_marca.php* se especificará los siguientes campos:

Tabla 19.

Atributos de las Taxonomías.

<i>'labels'</i>	Nombre en singular y plural.
<i>'hierarchical'</i>	Si la taxonomía tiene jerarquía. Ejemplo: automóvil, camioneta.
<i>'public'</i>	Si todos los usuarios pueden accederla.
<i>'show_ui'</i>	Si se muestra en la interfaz gráfica (Pueden existir entidades que no es necesario que el usuario pueda manipularlas).
<i>'show_admin_column'</i>	Mostrarlo en la columna de administración.
<i>'show_in_nav_menus'</i>	Mostrar en la barra de navegación.
<i>'show_tagcloud'</i>	Mostrar <i>tagcloud</i> .
<i>'show_in_rest'</i>	Habilitar el <i>endpoint</i> de <i>API Rest</i> . (Importante)
<i>'rest_base'</i>	El nombre del <i>endpoint</i> de la <i>API Rest</i> .

La parte izquierda contiene el valor del campo al que hace referencia, la parte de la derecha hace referencia a un archivo de traducción que pueda existir en el sistema:

```
'menu_name' => __( 'Marca', 'sciaplus' ),
'all_items' => __( 'Todos los Marcas', 'sciaplus' ),
```

Figura 43: Anotación para indicar archivo de traducción.


```

1 <?php
2 if ( ! function_exists( 'sciaplus_tax_marca' ) ) {
3
4 // Register Custom Taxonomy
5 function sciaplus_tax_marca() {
6
7     $labels = array(
8         'name' => _x( 'Marcas', 'Taxonomy General Name', 'sciaplus' ),
9         'singular_name' => _x( 'Marca', 'Taxonomy Singular Name', 'sciaplus' ),
10        'menu_name' => __( 'Marca', 'sciaplus' ),
11        'all_items' => __( 'Todos los Marcas', 'sciaplus' ),
12        'parent_item' => __( 'Marca Padre', 'sciaplus' ),
13        'parent_item_colon' => __( 'Marca Padre:', 'sciaplus' ),
14        'new_item_name' => __( 'Nuevo Marca', 'sciaplus' ),
15        'add_new_item' => __( 'Agregar Marca', 'sciaplus' ),
16        'edit_item' => __( 'Editar Marca', 'sciaplus' ),
17        'update_item' => __( 'Actualizar Marca', 'sciaplus' ),
18        'view_item' => __( 'Ver Marca', 'sciaplus' ),
19        'separate_items_with_commas' => __( 'Separar Marcas con comas', 'sciaplus' ),
20        'add_or_remove_items' => __( 'Agregar o Remover Marcas', 'sciaplus' ),
21        'choose_from_most_used' => __( 'Seleccionar del mas usado', 'sciaplus' ),
22        'popular_items' => __( 'Marcas Populares', 'sciaplus' ),
23        'search_items' => __( 'Buscar Marca', 'sciaplus' ),
24        'not_found' => __( 'No Existe', 'sciaplus' ),
25        'no_terms' => __( 'No hay Marcas', 'sciaplus' ),
26        'items_list' => __( 'Lista de Marcas', 'sciaplus' ),
27        'items_list_navigation' => __( 'Navegación de Marcas', 'sciaplus' ),
28    );
29    $args = array(
30        'labels' => $labels,
31        'hierarchical' => false,
32        'public' => true,
33        'show_ui' => true,
34        'show_admin_column' => true,
35        'show_in_nav_menus' => true,
36        'show_tagcloud' => false,
37        'show_in_rest' => true,
38        'rest_base' => 'marcasrest',
39    );
40    register_taxonomy( 'marca', array( 'articulo' ), $args );
41
42 }
43 add_action( 'init', 'sciaplus_tax_marca', 0 );
44
45 }

```

Figura 44: Archivo `tax_articulo_marca.php`.

De la misma forma se lo realiza con `tax_articulo_categoria.php`. Finalmente se debe crear los *metaboxes* y *metafields* especificando los inputs que el usuario llenará, esto se puede realizar “desde cero”; sin embargo, existe una librería muy útil llamada *CMB2* para crear fácilmente estos campos. Se puede copiar el contenido del ejemplo que contiene esta carpeta llamado `example-functions.php` y pegarlo en `mtx_articulo`, luego como lo indica reemplazar el prefijo en todos los lugares donde se encuentre la cadena de texto `yourprefix_`, esto se muestra en la Figura 45:

```

1 <?php
2 /**
3  * Include and setup custom metaboxes and fields. (make sure you copy this file to outside
4  * the CMB2 directory)
5  *
6  * Be sure to replace all instances of 'yourprefix_' with your project's prefix.
7  * http://nacin.com/2010/05/11/in-wordpress-prefix-everything/
8  *
9  * @category YourThemeOrPlugin
10 * @package Demo_CMB2
11 * @license http://www.opensource.org/licenses/gpl-license.php GPL v2.0 (or later)
12 * @link https://github.com/CMB2/CMB2
13 */
14 /**

```

Figura 45: Archivo *example-functions.php*.

Posteriormente se borra la información detallada sobre la herramienta de *CMB2* tomando en cuenta que se debe agregar al archivo *prerrequisitos* las funciones que llaman a sus librerías y se utilizarán para todos los *metaboxes* y *metafields* como se muestra a continuación:

```

54
55 /* CMB2 */
56
57 if ( file_exists( dirname( __FILE__ ) . '/backend/cmb2/init.php' ) ) {
58     require_once dirname( __FILE__ ) . '/backend/cmb2/init.php';
59 } elseif ( file_exists( dirname( __FILE__ ) . '/backend/CMB2/init.php' ) ) {
60     require_once dirname( __FILE__ ) . '/backend/CMB2/init.php';
61 }
62

```

Figura 46: Archivo *prerrequisitos.php*.

Una vez cambiado los prefijos en todo el archivo *mtx_articulo.php* el cual le dan una identidad única a cada elemento en todo el contexto del sistema, se procede a agregar un *metabox* y luego los *metafields* según corresponda. Del archivo de ejemplo se puede tomar los *metafields* que se necesiten y borrar el resto, de esta forma el *CPT* artículo contendrá los *metafields*:

```

1 <?php
2 /**
3  * Be sure to replace all instances of 'sciaplus.' with your project's prefix.
4  */
5
6 add_action( 'cmb2_admin_init', 'articulo_register_metabox' );
7
8 function articulo_register_metabox() {
9     $prefix = 'sciaplus_articulo_';
10
11     $cmd_datos_articulo = new_cmb2_box( array(
12         'id'           => $prefix . 'datosarticulo',
13         'title'        => esc_html__( 'Datos del Artículo', 'sciaplus' ),
14         'object_types' => array( 'articulo' ), // Post type
15     ) );
16
17
18     $cmd_datos_articulo->add_field( array(
19         'name'         => esc_html__( 'Serial', 'sciaplus' ),
20         'desc'         => esc_html__( 'Ej: 123456 ', 'sciaplus' ),
21         'id'           => $prefix . 'serial',
22         'type'         => 'text_medium',
23         'attributes'  => array(
24             'pattern' => '[0-9]*',
25             'min'     => '0100000000',
26             'max'     => '2499999999',
27             'required' => 'required',
28         ),
29     ) );
30
31     $cmd_datos_articulo->add_field( array(
32         'name'         => esc_html__( 'Categoria', 'sciaplus' ),
33         'desc'         => esc_html__( 'Seleccinar', 'sciaplus' ),
34         'id'           => $prefix . 'categoria',
35         'type'         => 'taxonomy_select',
36         'taxonomy'     => 'categoria', // Taxonomy Slug
37     ) );
38
39     $cmd_datos_articulo->add_field( array(
40         'name'         => esc_html__( 'Marca', 'sciaplus' ),
41         'desc'         => esc_html__( 'Seleccinar', 'sciaplus' ),
42         'id'           => $prefix . 'marca',
43         'type'         => 'taxonomy_select',
44         'taxonomy'     => 'marca', // Taxonomy Slug
45     ) );
46

```

Prefijo

Metabox

Metafields. Sus implementaciones dependen del tipo de campo

Figura 47: Archivo `mtx_articulo.php`.

```

46
47 $cmd_datos_articulo->add_field( array(
48     'name'    => esc_html__( 'Modelo', 'sciaplus' ),
49     'desc'    => esc_html__( 'Ej: A1 ', 'sciaplus' ),
50     'id'      => $prefix . 'modelo',
51     'type'    => 'text_medium',
52 ) );
53
54 $cmd_datos_articulo->add_field( array(
55     'name' => esc_html__( 'Costo Compra', 'sciaplus' ),
56     'desc' => esc_html__( 'Ej: $10', 'sciaplus' ),
57     'id'  => $prefix . 'costo_compra',
58     'type' => 'text_money',
59     // 'before_field' => 'E', // override '$' symbol if needed
60     // 'repeatable' => true,
61 ) );
62
63 $cmd_datos_articulo->add_field( array(
64     'name' => esc_html__( 'Costo Venta', 'sciaplus' ),
65     'desc' => esc_html__( 'Ej: $20', 'sciaplus' ),
66     'id'  => $prefix . 'costo_venta',
67     'type' => 'text_money',
68     // 'before_field' => 'E', // override '$' symbol if needed
69     // 'repeatable' => true,
70 ) );
71
72 $cmd_datos_articulo->add_field( array(
73     'name'    => esc_html__( 'Descripción', 'sciaplus' ),
74     'desc'    => esc_html__( 'Ej: 1 Par de parlantes ', 'sciaplus' ),
75     'id'      => $prefix . 'descripcion',
76     'type'    => 'text_medium',
77 ) );
78
79 }

```

Figura 48: Archivo `mtx_articulo.php`.

De esta forma se realiza un CRUD de la entidad artículo con sus atributos y taxonomías lista para guardar datos/artículos en la tabla `wp_posts` cuyo campo `post_type` hace referencia a artículo.

The screenshot shows the WordPress dashboard interface for adding a new article. The main content area is titled 'Agregar Nuevo Artículo' and contains a form with several fields. A 'Metabox' titled 'Datos del Artículo' is visible, containing fields for 'Serial', 'Categoría', 'Marca', 'Modelo', and 'Costo Compra'. Each field has a corresponding input box and a placeholder example (e.g., 'Ej: 123456', 'No terms', 'Ej: A1', '\$ Ej: \$10'). A 'Metafields' label with arrows points to these input fields. On the right side, there is a 'Publish' section with buttons for 'Save Draft', 'Preview', and 'Publish', along with status and visibility options. Below the 'Publish' section is a 'Imagen Destacada' section with a button to 'Establecer Imagen Destacada'.

Figura 49: Representación de la entidad Artículo en el *dashboard*.

De esta forma se realiza un CRUD de la entidad artículo con sus atributos y taxonomías listas para guardar datos/artículos en la tabla 'wp_posts' cuyo campo 'post_type' hace referencia a artículo.

Un ejemplo de una entidad que necesita de otra entidad es la Suscripción ya que necesita tanto del dispositivo como el artículo para crear una relación y registrarlos. La creación del *Custom Post Type* y Taxonomías es igual al de artículos, las diferencias se encuentran en la creación de otro *metabox* y *metafield* respectivamente los cuales listan los dispositivos y artículos existentes como se muestra a continuación:

```

1  <?php
2  /**
3   * Be sure to replace all instances of 'sciaplus_' with your project's prefix.
4   */
5
6  add_action( 'cmb2_admin_init', 'suscripcion_register_metabox' );
7
8
9  function suscripcion_register_metabox() {
10     $prefix = 'sciaplus_suscripcion_';
11
12     $arg = array(
13         "post_type" => "dispositivo",
14         "posts_per_page" => -1,
15     );
16     $dispositivos = new WP_Query($arg);
17     $listadispositivos = array();
18     if($dispositivos->have_posts())
19     {
20         while($dispositivos->have_posts())
21         {
22             $dispositivos->the_post();
23             $id = get_the_ID();
24             $listadispositivos[$id] = get_the_title();
25         }
26     }
27     $arg = array(
28         "post_type" => "articulo",
29         "posts_per_page" => -1,
30     );
31     $articulos = new WP_Query($arg);
32     $listaarticulos = array();
33     if($articulos->have_posts())
34     {
35         while($articulos->have_posts())
36         {
37             $articulos->the_post();
38             $id = get_the_ID();
39             $listaarticulos[$id] = get_the_title();
40         }
41     }

```

Figura 50: Archivo *mtx_suscripción.php*.

```

118 $cmd_datos_dispositivo = new_cmb2_box( array(
119     'id'           => $prefix . 'datosdispositivo',
120     'title'       => esc_html__( 'Datos del Dispositivo', 'sciaplus' ),
121     'object_types' => array( 'suscripcion' ), // Post type
122     'context'     => 'side',
123     'priority'    => 'high',
124 ) );
125
126 $cmd_datos_dispositivo->add_field( array(
127     'name'       => esc_html__( 'Dispositivo', 'sciaplus' ),
128     'desc'       => esc_html__( 'Seleccione', 'sciaplus' ),
129     'id'         => $prefix . 'id_dispositivo',
130     'show_option_none' => true,
131     'type'       => 'select',
132     'options'    => $listadispositivos,
133     'attributes' => array(
134         'required' => 'required',
135     ),
136 ) );
137
138 $cmd_datos_articulo = new_cmb2_box( array(
139     'id'           => $prefix . 'datosarticulo',
140     'title'       => esc_html__( 'Datos del Artículo', 'sciaplus' ),
141     'object_types' => array( 'suscripcion' ), // Post type
142     'context'     => 'side',
143     'priority'    => 'high',
144 ) );
145
146 $cmd_datos_articulo->add_field( array(
147     'name'       => esc_html__( 'Artículo', 'sciaplus' ),
148     'desc'       => esc_html__( 'Seleccione', 'sciaplus' ),
149     'id'         => $prefix . 'id_articulo',
150     'show_option_none' => true,
151     'type'       => 'select',
152     'options'    => $listaarticulos,
153     'attributes' => array(
154         'required' => 'required',
155     ),
156 ) );

```

Figura 51: Archivo *mtx_suscripcion.php*.

De esta forma se crean dos *metaboxes* al costado superior derecho de las entidades con las que se relacionará la suscripción:

The screenshot shows the 'Agregar Nueva Suscripción' form in a WordPress dashboard. The form has a sidebar menu on the left with 'Suscripciones' selected. The main form area contains an 'Add title' field, a 'Datos de la Suscripción' section with sub-fields for 'Código', 'Fecha de Activación', 'Fecha de Caducidad', and 'Fecha de Notificación', and two metaboxes on the right: 'Datos del Dispositivo' and 'Datos del Artículo', each containing a 'None' dropdown menu.

Figura 52: Representación de la entidad Suscripción en el *dashboard*.

3.5.1.5 Daily Scrum

Las reuniones se realizarán en los primeros quince minutos del día en donde se revisará lo realizado, los problemas que se tiene y lo que se realizará durante aquel día con respecto a las entidades Clientes, Dispositivos, Artículos y Suscripciones.

3.5.1.6 Revisión

Se considera que el producto generado está listo para ser entregado y cumple con los criterios de aceptación, los cambios que puedan generarse no deberán representar un esfuerzo considerable.

3.5.1.7 Retrospectiva

El trabajo realizado sobre este *sprint* ha cumplido con las expectativas, sin embargo, es necesario considerar los siguientes puntos para el siguiente *sprint*:

Ventajas

- 1) Conocimiento completo sobre las tecnologías a implementar.
- 2) Comprensión total de los requerimientos del cliente.

Desventajas

- 1) Desarrollo tardío como lo representa el *burn down chart* debido al detalle en la presentación e implementación de buenas prácticas de desarrollo sobre el código.

3.5.2 Sprint 2

3.5.2.1 Sprint Planning

Los objetivos para este *sprint* consisten en la creación de entidades complejas faltantes como lo Tickets y Facturas las cuales darán fin al desarrollo de los módulos y podrán ser probadas e implementadas si así se lo desea.

3.5.2.2 Sprint Backlog

Tabla 20.

Sprint Backlog del Sprint 2.

ID	Nombre HU	Descripción	Responsable	Esfuerzo
TIT02	Administrar Técnicos	Como usuario quiero administrar los técnicos de <i>A Plus Computer</i> para utilizar su información actual en procesos de negocio como tickets atendidos, suscripciones creadas y eventualmente creación de facturas.	David Escobar	2PH
TIT05	Administrar los Tickets	Como usuario quiero administrar los tickets para llevar un control de los trabajos que los técnicos atienden sobre los dispositivos de los clientes.	David Escobar	4PH
TIT09	Administrar Facturas	Como usuario quiero administrar las facturas para llevar una contabilidad detallada y registrar los ítems vendidos, servicios prestados y tickets atendidos en las facturas para controlar las garantías.	David Escobar	4PH

3.5.2.3 Historias de Usuario

Tabla 21.

Historia de usuario TIT02.

Historia de usuario	
ID: TIT02	Usuario: Usuario
Nombre de historia: Administrar Técnicos	

Prioridad en negocio: Media	Responsable: David Escobar
Descripción: Como usuario quiero administrar los técnicos de <i>A Plus Computer</i> para utilizar su información actual en procesos de negocio como tickets atendidos, suscripciones creadas y eventualmente creación de facturas.	
Criterios de aceptación:	
<ol style="list-style-type: none"> 1) Debe poder leer los técnicos. 2) Debe poder crear un técnico. 3) Debe poder modificar un técnico. 4) Debe poder eliminar un técnico. 5) Un cliente debe contener los campos: Cédula, Nombre, Dirección, Teléfono, Celular, Fecha de Nacimiento, Correo Electrónico, Cargo y una imagen. 	

Tabla 22.

Historia de usuario TIT05.

Historia de usuario	
ID: TIT05	Usuario: Usuario
Nombre de historia: Administrar los Tickets	
Prioridad en negocio: Alta	Responsable: David Escobar
Descripción: Como usuario quiero administrar los tickets para llevar un control de los trabajos que los técnicos atienden sobre los dispositivos de los clientes.	
Criterios de aceptación:	
<ol style="list-style-type: none"> 1) Debe poder leer los tickets. 	

- 2) Debe poder crear un ticket.
- 3) Debe poder modificar un ticket.
- 4) Debe poder eliminar un ticket.
- 5) Un cliente debe contener los campos: Código, Descripción, Estado y una imagen.
- 6) Debe contener la referencia a los artículos utilizados, la suscripción implementada, el técnico que solventó el problema y el dispositivo atendido.

Tabla 23.

Historia de usuario TIT09.

Historia de usuario	
ID: TIT09	Usuario: Usuario
Nombre de historia: Administrar Facturas	
Prioridad en negocio: Alta	Responsable: David Escobar
Descripción: Como usuario quiero administrar las facturas para llevar una contabilidad detallada y registrar los ítems vendidos, servicios prestados y tickets atendidos en las facturas para controlar las garantías.	
Criterios de aceptación:	
<ol style="list-style-type: none"> 1) Debe poder leer las facturas. 2) Debe poder crear una factura. 3) Debe poder modificar una factura. 4) Debe poder eliminar una factura. 	

- 5) Un cliente debe contener los campos: Código, Subtotal, Descuento, IVA, Total y una Imagen.
- 6) Debe contener la referencia de los tickets facturados los cuales harán referencia a los artículos y suscripciones a cobrar.

3.5.2.4 Desarrollo

El desarrollo de las entidades Técnicos, Tickets y Facturas son similares a las de Artículos y Suscripciones, lo que las diferencia es la creación de varios campos para Artículos dentro Tickets lo cual se soluciona agregando un *metabox* con grupos repetibles compuestos por *metafields* de la siguiente forma:

```

128 $cmb_repeat_test = new_cmb2_box( array(
129     'id'           => $prefix . 'metaboxjff',
130     'title'        => __( 'Artículos', 'your-text-domain' ),
131     'object_types' => array( 'ticket' ), // post type
132     // 'show_on'    => array( 'key' => 'page-template', 'value' => 'page-test.php' ),
133     'context'      => 'normal',
134     'priority'     => 'high',
135     'show_names'  => true,
136 ) );
137
138 $group_repeat_test = $cmb_repeat_test->add_field( array(
139     'id'           => $prefix . 'metaboxjff_sections',
140     'type'         => 'group',
141     'options'      => array(
142         'group_title' => __( 'Artículo', 'your-text-domain' ) . ' {#}', // {#} gets
replaced by row number
143         'add_button'  => __( 'Agregar otro Artículo', 'your-text-domain' ),
144         'remove_button' => __( 'Remover Artículo', 'your-text-domain' ),
145         'sortable'   => true, // beta
146     ),
147 ) );
148 $cmb_repeat_test->add_group_field( $group_repeat_test, array(
149     'name'         => esc_html__( 'Artículo', 'sciaplus' ),
150     'desc'         => esc_html__( 'Seleccione', 'sciaplus' ),
151     'id'           => $prefix . 'id_articulo',
152     'show_option_none' => true,
153     'type'         => 'select',
154     'options'      => $listaarticulos,
155     'attributes'  => array(
156         'required'  => 'required',
157     ),
158 ) );
159 $cmb_repeat_test->add_group_field( $group_repeat_test, array(
160     'name'         => __( 'Cantidad', 'your-text-domain' ),
161     'id'           => $prefix . 'id_cantidad_articulo',
162     'type'         => 'text',
163 ) );
164

```

The image shows a code editor window titled 'm_mtx_ticket.php'. The code defines a metabox and its fields. Annotations with arrows point to specific parts of the code:

- Metabox:** Points to the `new_cmb2_box` function call (lines 128-135).
- Grupo:** Points to the `add_field` function call (lines 138-147).
- Metafield:** Points to the `add_group_field` function calls (lines 148-158 and 159-163).

Figura 53: Archivo *mtx_ticket.php*.

Lo cual genera una entrada de datos como se muestra en la Figura 54:

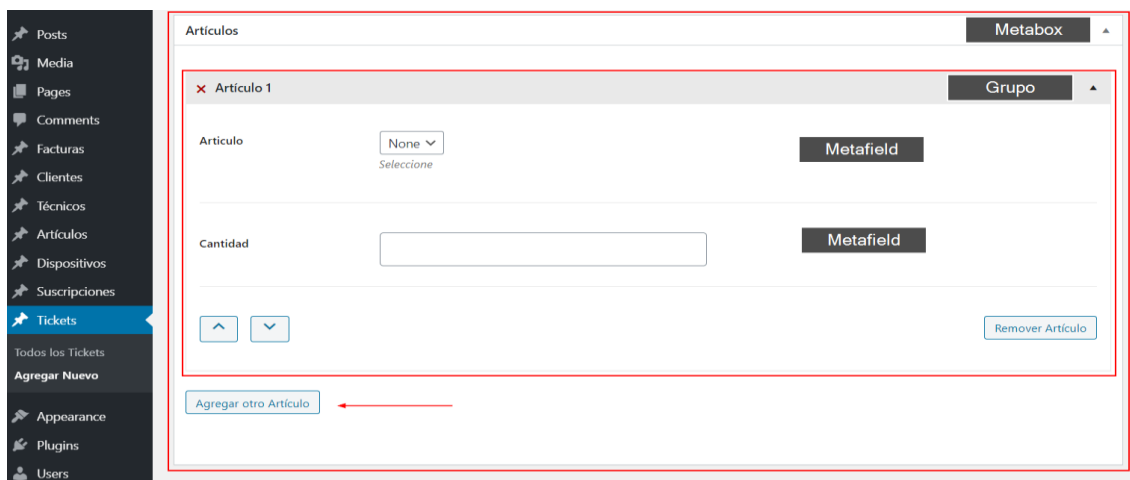


Figura 54: Representación de la entidad Ticket en el *dashboard*.

Por otro lado, la entidad Factura en un inicio muestra los clientes para según esto cargar los Tickets en estado “Finalizado” los cuales estarían pendientes de cobro. Una vez cargados en la vista, al seleccionar cada uno debe desplegar los artículos o suscripciones que se usaron o activaron respectivamente. Esto se lo realiza de la siguiente forma:

```

1  <?php
2  /**
3   * Be sure to replace all instances of 'sciaplus_' with your project's prefix.
4   */
5
6  add_action( 'cmb2_admin_init', 'factura_register_metabox' );
7
8  function factura_register_metabox() {
9      $prefix = 'sciaplus_factura_';
10
11
12     $arg = array(
13         "post_type" => "cliente",
14         "posts_per_page" => -1,
15     );
16     $clientes = new WP_Query($arg);
17     $listaclientes = array();
18     if($clientes->have_posts())
19     {
20         while($clientes->have_posts())
21         {
22             $clientes->the_post();
23             $id = get_the_ID();
24             $listaclientes[$id] = get_the_title();
25         }
26     }

```

Figura 55: Archivo *mtx_factura.php*.

```

28 function yourprefix_render_row_cb( $field_args, $field ) {
29     $classes = $field->row_classes();
30     $id = $field->args( 'id' );
31     $label = $field->args( 'name' );
32     $name = $field->args( '_name' );
33     $value = $field->escaped_value();
34     $description = $field->args( 'description' );
35     ?>
36     <div class="custom-field-row <?php echo esc_attr( $classes ); ?>">
37         <div class="search_bar">
38             <label>Seleccione un Ticket</label>
39             <select id="ticket_list_input"></select>
40         </div>
41         <div id="ticket_list" class="ticket_list"></div>
42     </div>
43     <?php
44 }
45

```

Creación de Metabox con Metafield customizado para el despliegue de los tickets, artículos y suscripciones a cobrar

Aquí se implementará Ajax para la carga de los datos según lo que el usuario seleccione

Figura 56: Archivo *mtx_factura.php*.

Debido a que el Ticket contiene el Dispositivo atendido y este hace referencia a al propietario o Cliente, primero se debe realizar presentar los Clientes para posteriormente mostrar únicamente los Tickets por cobrar, seleccionado el Ticket carga los Artículos y Suscripción implementados como se muestra a continuación:

```

19 add_action( 'admin_enqueue_scripts', 'my_enqueue' );
20 function my_enqueue($hook) {
21     wp_enqueue_script( 'sciaplus-script', plugins_url( 'Facturas/js/sciaplus-script.js',
22     wp_localize_script( 'sciaplus-script', 'sciaplus_object', array( 'ajax_url' => admin_url
23     ( 'admin-ajax.php' ), 'we_value' => 1234 ));
24 }
25
26 add_action( 'wp_ajax_nopriv_buscar_tickets', 'buscar_tickets' );
27 add_action( 'wp_ajax_buscar_tickets', 'buscar_tickets' );
28 function buscar_tickets() {
29     global $wpdb;
30
31     $arg = array(
32         "post_type" => "dispositivo",
33         "posts_per_page" => -1,
34         'meta_key' => 'sciaplus_dispositivo_id_cliente',
35         'meta_value' => $_GET['client_id']
36     );
37     $dispositivos = new WP_Query($arg);
38     $listadispositivos = array();
39     if($dispositivos->have_posts())
40     {
41         while($dispositivos->have_posts())
42         {
43             $dispositivos->the_post();
44             $listadispositivos[] = get_the_ID();
45         }
46     }

```

Agregar el archivo sciaplus-script.js el cual llamará y mostrará los resultados que solicite el usuario implementando las siguientes funciones

Función para buscar los Tickets

Figura 57: Archivo *facturas.php*.

```

facturas.php
47     $listatickets = array();
48     foreach ($listadispositivos as $dispositivo) {
49         $arg = array(
50             "post_type" => "ticket",
51             "posts_per_page" => -1,
52             'meta_query' => array(
53                 'relation' => 'AND',
54                 array(
55                     'key' => 'sciaplus_ticket_id_dispositivo',
56                     'value' => $dispositivo,
57                     'compare' => '=',
58                 ),
59                 array(
60                     'key' => 'sciaplus_ticket_estado',
61                     'value' => 'finalizado',
62                     'compare' => '=',
63                 ),
64             ),
65         );
66         $tickets = new WP_Query($arg);
67         if($tickets->have_posts())
68         {
69             while($tickets->have_posts())
70             {
71                 $tickets->the_post();
72                 $id = get_the_ID();
73                 $listatickets[] = array(
74                     'id' => $id,
75                     'value' => get_post_meta($id, 'sciaplus_ticket_descripcion', true)
76                 );
77             }
78         }
79     }
80     //wp_die();
81 }

```

Figura 58: Archivo *facturas.php* (Función que busca los Tickets).

```

facturas.php
84 add_action( 'wp_ajax_nopriv_buscar_datos_de_ticket', 'buscar_datos_de_ticket' );
85 add_action( 'wp_ajax_buscar_datos_de_ticket', 'buscar_datos_de_ticket' );
86
87 function buscar_datos_de_ticket(){
88     $ticket = NULL;
89     $elementos_ticket = NULL;
90     $ticket = get_post($_GET['ticket_id']);
91     $elemento_ticket = NULL;
92     $ticket_items = get_post_meta($_GET['ticket_id'], 'sciaplus_ticket_metaboxjff_sections',
93         true);
94     foreach($ticket_items as $item){
95         $articulo = get_post($item['sciaplus_ticket_id_articulo']);
96         $articulo_nombre = $articulo->post_title;
97         $articulo_precio = get_post_meta($articulo->ID, 'sciaplus_articulo_costo_venta',
98             true);
99         $elementos_ticket[] = array(
100             'id' => $articulo->ID,
101             'cantidad' => $item['sciaplus_ticket_id_cantidad_articulo'],
102             'nombre' => $articulo_nombre,
103             'precio' => $articulo_precio
104         );
105     }
106     $elemento_ticket = array(
107         'id' => $_GET['ticket_id'],
108         'nombre' => $ticket->post_title,
109         'elementos' => $elementos_ticket
110     );
111     echo json_encode($elemento_ticket);
112     wp_die();
113 }

```

Función para extraer los datos a facturar del Ticket resuelto

Figura 59: Archivo *facturas.php*.

```

sciaplus-script.js
3  jQuery(document).ready(function(){
4      jQuery('#sciaplus_factura_id_cliente').on('change',function(){
5          jQuery.ajax({
6              url: sciaplus_object.ajax_url,
7              data: {
8                  action: 'buscar_tickets',
9                  client_id: jQuery('#sciaplus_factura_id_cliente').val()
10             },
11             success: function(response){
12                 var option = '';
13                 option += option += '<option>Selecione</option>';
14                 jQuery.each( JSON.parse(response), function(index,value){
15                     option += '<option value="' + value.id + '">' + value.id + ' - ' +
16                     value.value + '</option>';
17                 });
18                 jQuery('#ticket_list_input').append(option);
19             },
20             error: function(response){
21                 console.log(response);
22             }
23         });
24     });
25

```

Figura 60: Archivo *sciaplus-script.js* para listar los tickets.

```

sciaplus-script.js
27  jQuery('#ticket_list_input').on('change',function(){
28      console.log('cargando los datos del ticket');
29      var itemdetail = '';
30      jQuery('#ticket_list').html('');
31      jQuery.ajax({
32          url: sciaplus_object.ajax_url,
33          data: {
34              action: 'buscar_datos_de_ticket',
35              ticket_id: jQuery(this).val()
36          },
37          success: function(response){
38              console.log(response);
39              var listaItems = JSON.parse(response);
40              itemdetail = '<div id="ticket-' + listaItems.id + '"><h3>'+
41              listaItems.nombre +
42              '</h3><button id="agregarTicket" type="button">Agregar</button><ul>';
43
44              jQuery.each( listaItems.elementos, function(index,value){
45                  console.log(value);
46                  itemdetail += '<li class="single-item" data-id="'+value.id+'><strong>'
47 + value.nombre + '</strong> | <span><input name="precio" type="number" step="0,01" value="'+
48 value.precio + '"></span> | <span><input name="cantidad" type="number" value="'+
49 value.cantidad + '"></span></li>';
50              });
51              itemdetail += '</ul></div>';
52
53              jQuery('#ticket_list').append(itemdetail);
54          },
55          error: function(response){
56              console.log(response);
57          }
58      });
59  });
60

```

Figura 61: Archivo *sciaplus-script.js* donde actualiza suscripción y artículos.

Lo cual crea el dinamismo que tiene la Factura en su creación:

The screenshot shows the 'Editar Factura' (Edit Invoice) interface. At the top, there is a text input field containing '001001000006' and a 'Permalink' field with the URL 'http://sciaplusecuador.com/factura/001001000006/'. Below this is a 'Tickets' section with a dropdown menu showing '415 - El cliente necesita que se le de mantenimiento de software, hardware, revisar el estado del disco duro y desea que sus docum'. Underneath the dropdown is an 'Agregar' (Add) button and a table with two rows of input fields for items:

10000058 - Kingston	20.00	2
1000000002 - M2020	210.00	1

Below the items table is the 'Datos de la Factura' (Invoice Data) section with a 'Código' (Code) field containing '001001000006' and an example 'Ej: 123456'. On the right side, there is a 'Datos del Cliente' (Client Data) section with a dropdown menu showing '1818181820 - David' and a 'Publish' section with buttons for 'Save Draft', 'Preview', 'Status: Draft Edit', 'Visibility: Public Edit', 'Publish immediately Edit', 'Move to Trash', and 'Publish'.

Figura 62: Representación de la entidad Factura en el *dashboard*.

The screenshot shows the 'Editar Factura' interface with updated values. The 'Datos de la Factura' section now has the following values:

Código	001001000006	Ej: 123456
Subtotal	\$ 250	Ej: \$100
Descuento	\$ 240	Ej: \$0
IVA	\$ 28.8	Ej: 12%
Total	\$ 268.8	Ej: \$120

The right side of the interface remains the same, showing the 'Publish' section with buttons for 'Move to Trash' and 'Publish'.

Figura 63: Factura en el *dashboard* con campos actualizados.

De esta forma se termina la creación de las entidades, módulos o *plugins* del Sistema de Control Interno en *WordPress* siendo fácil de mantener y adaptable a futuras necesidades de *A Plus Computer*.

3.5.2.5 Daily Scrum

Las reuniones se realizarán en los primeros quince minutos del día en donde se revisará lo realizado, los problemas que se tiene y lo que se realizará durante aquel día con respecto a las entidades Técnicos, Tickets y Facturas.

3.5.2.6 Revisión

Se considera que el producto generado está listo para ser entregado y cumple con los criterios de aceptación, los cambios que puedan generarse no deberán representar un esfuerzo considerable.

3.5.2.7 Retrospectiva

El trabajo realizado sobre este *sprint* ha cumplido con las expectativas, sin embargo, es necesario considerar los siguientes puntos para el siguiente *sprint*:

Ventajas

- 1) Conocimiento completo sobre las tecnologías a implementar.
- 2) Comprensión de los requerimientos del cliente.
- 3) Reutilización de código.

Desventajas

- 1) Inconvenientes al agregar más de una entidad simple a entidades complejas, por ejemplo, la inclusión de varios tickets dentro de las facturas.
- 2) Validaciones de campos no tomadas en cuenta.

3.5.3 Sprint 3

3.5.3.1 Sprint Planning

Los objetivos para este *sprint* consisten en la creación de usuarios y roles y la extracción de los datos ingresados sobre los módulos creados previamente y presentándolos sobre un panel *BI*.

3.5.3.2 Sprint Backlog

Tabla 24.

Sprint Backlog del Sprint 3.

ID	Nombre HU	Descripción	Responsable	Esfuerzo
TIT06	Reportes personalizados de Tickets	Como usuario quiero obtener reportes personalizados de los tickets atendidos para identificar problemas y tomar decisiones de negocio.	David Escobar	3PH
TIT08	Reportes personalizados de Suscripciones	Como usuario quiero obtener reportes personalizados de las suscripciones para identificar problemas y tomar decisiones de negocio.	David Escobar	3PH
TIT10	Reportes personalizados de Facturas	Como usuario quiero obtener reportes personalizados de las facturas emitidas para identificar problemas y tomar decisiones de negocio.	David Escobar	3PH
TIT11	Administrar Usuarios	Como administrador quiero agregar usuarios para que puedan utilizar el sistema.	David Escobar	1PH

3.5.3.3 Historias de Usuario

Tabla 25.

Historia de usuario TIT06.

Historia de usuario	
ID: TIT06	Usuario: Usuario
Nombre de historia: Reportes personalizados de Tickets	
Prioridad en negocio: Media	Responsable: David Escobar
Descripción: Como usuario quiero obtener reportes personalizados de los tickets atendidos para identificar problemas y tomar decisiones de negocio.	

Criterios de aceptación:

- 1) Debe existir un *endpoint* de *Web Service* para los tickets.
- 2) Debe existir un *endpoint* de *Web Service* para los clientes.
- 3) Debe existir un *endpoint* de *Web Service* para los técnicos.
- 4) Debe existir un *endpoint* de *Web Service* para los dispositivos.
- 5) Debe existir un *endpoint* de *Web Service* para los artículos.
- 6) Debe guardar los datos en una base externa a la de *WordPress* con los datos estandarizados por tablas.
- 7) Debe generar reportes personalizables en *Power BI*.

Tabla 26.

Historia de usuario TIT08.

Historia de usuario	
ID: TIT08	Usuario: Usuario
Nombre de historia: Reportes personalizados de Suscripciones	
Prioridad en negocio: Media	Responsable: David Escobar
Descripción: Como usuario quiero obtener reportes personalizados de las suscripciones para identificar problemas y tomar decisiones de negocio.	
Criterios de aceptación:	
<ol style="list-style-type: none"> 1) Debe existir un <i>endpoint</i> de <i>Web Service</i> para las suscripciones. 2) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los dispositivos. 3) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los artículos. 4) Debe guardar los datos en una base externa a la de <i>WordPress</i> con los datos estandarizados por tablas. 5) Debe generar reportes personalizables en <i>Power BI</i>. 	

Tabla 27.

Historia de usuario TIT10.

Historia de usuario	
ID: TIT10	Usuario: Usuario
Nombre de historia: Reportes personalizados de Facturas	
Prioridad en negocio: Media	Responsable: David Escobar
Descripción: Como usuario quiero obtener reportes personalizados de las facturas emitidas para identificar problemas y tomar decisiones de negocio.	
Criterios de aceptación:	
<ol style="list-style-type: none"> 1) Debe existir un <i>endpoint</i> de <i>Web Service</i> para las facturas. 2) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los tickets. 3) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los clientes. 4) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los técnicos. 5) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los dispositivos. 6) Debe existir un <i>endpoint</i> de <i>Web Service</i> para los artículos. 7) Debe existir un <i>endpoint</i> de <i>Web Service</i> para las suscripciones. 8) Debe guardar los datos en una base externa a la de <i>WordPress</i> con los datos estandarizados por tablas. 9) Debe generar reportes personalizables en <i>Power BI</i>. 	

Tabla 28.

Historia de usuario TIT11.

Historia de usuario	
ID: TIT11	Usuario: Administrador
Nombre de historia: Administrar Usuarios	
Prioridad en negocio: Baja	Responsable: David Escobar
Descripción: Como administrador quiero agregar usuarios para que puedan utilizar el sistema.	
Criterios de aceptación:	
<ol style="list-style-type: none"> 1) Debe poder leer los usuarios. 2) Debe poder crear un usuario. 	

- 3) Debe poder modificar un usuario.
- 4) Debe poder eliminar un usuario.
- 5) Un usuario debe contener la información completa.
- 6) Debe tener permisos para acceder como administrador a todo el sistema, como gerente a todos los módulos y como técnicos a todos los módulos tomando en cuenta que el módulo de facturas solo puede leer.

3.5.3.4 Desarrollo

Cuando se crearon los *Custom Post Types* de cada entidad se habilitó el *endpoint* de *API Rest* el cual permite la extracción de datos, sin embargo, por defecto solo indica el tipo de post mas no los *postmeta* relacionados a dicho *post*, es por ello que es necesario agregar una función que, al momento de solicitar la información de los *posts*, responda con la información completa incluyendo el *postmeta* en el archivo *Json* como se muestra a continuación:

```
prerrequisitos.php x
81
82 add_action( 'rest_api_init', 'articulo_api_rest_post_meta' );
83
84 function articulo_api_rest_post_meta() {
85     register_rest_field( 'articulo',
86         'post_meta_fields',
87         array(
88             'get_callback' => 'callback_articulo_postmeta'
89         )
90     );
91 }
92
93 function callback_articulo_postmeta( $object ) {
94     $post_id = $object['id'];
95     return get_post_meta( $post_id );
96 }
```

Figura 64: Archivo *prerrequisitos.php*.

De esta forma se obtiene la información completa de cada entidad, esto se puede verificar ingresando datos de artículos y realizando la petición con la herramienta *Postman* de la siguiente forma:

```

3      "id": 343,
4      "date": "2019-12-08T21:29:52",
5      "date_gmt": "2019-12-09T02:29:52",
6      "guid": {
7        "rendered": "http://sci.aplusecuador.com/?post_type=articulo&#038;p=343"
8      },
9      "modified": "2019-12-08T21:29:52",
10     "modified_gmt": "2019-12-09T02:29:52",
11     "slug": "10000000002",
12     "status": "publish",
13     "type": "articulo",
14     "link": "http://sci.aplusecuador.com/articulo/10000000002/",
15     "title": {
16       "rendered": "10000000002"
17     },
18     "featured_media": 0,
19     "template": "",
20     "categoriasrest": [
21       19
22     ],
23     "marcasrest": [
24       30
25     ],
26     "post_meta_fields": {
27       "_edit_lock": [
28         "1575858783:1"
29       ],
30       "_edit_last": [
31         "1"
32       ],
33       "sciaplus_articulo_serial": [
34         "10000000002"

```

Figura 65: Petición *GET* a *endpoint* de clientes.

Posteriormente es necesario instalar *Pentaho* siguiendo los siguientes pasos:

- 1) Instalar *Java JDK 8*.
- 2) Configurar `JAVA_HOME` al directorio bin de la instalación de Java.
- 3) Descargar *Pentaho* desde el siguiente enlace <https://sourceforge.net/projects/pentaho/>
- 4) Copiar a una carpeta (*Pentaho* no requiere instalación, es portable)
- 5) Abrir *Spoon* ubicado en `\pdi-ce-8.2.0.0-342\data-integration`

Dentro de la herramienta *Spoon* es necesario agregar y conectar los siguientes elementos que conforman el proceso de extracción, procesamiento y guardado de los datos presentados desde el *endpoint* de *WordPress*:



Figura 66: Elementos de *Spoon* para el proceso de los datos.

Generate Rows setea el *endpoint* del que se consumirá:

Step name: Generate Rows

Limit: 1

Never stop generating rows:

Interval in ms (delay): 5000

Current row time field name: now

Previous row time field name: FiveSecondsAgo

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set emp
1	endpoint	String							http://sci.aplusecuador.com/wp-json/wp/v2/clientesrest/	N

Figura 67: Configuración del elemento *Generate Rows*.

REST Client hará la petición al servidor:

Step name: REST client

General | Authentication | SSL | Headers | Parameters | Matrix Parameters

Settings

URL: []

Accept URL from field?

URL field name: endpoint

HTTP method: GET

Get Method from field:

Method field name: []

Body field: []

Application type: TEXT PLAIN

Output fields

Result field name: result

HTTP status code field name: []

Response time (milliseconds) field name: []

Response header field name: []

Figura 68: Configuración del elemento *Rest Client*.

Y con *JSON input* se obtendrá los campos necesarios para organizarlos y guardarlos en la base de datos que posteriormente serán presentados con *Power BI*.

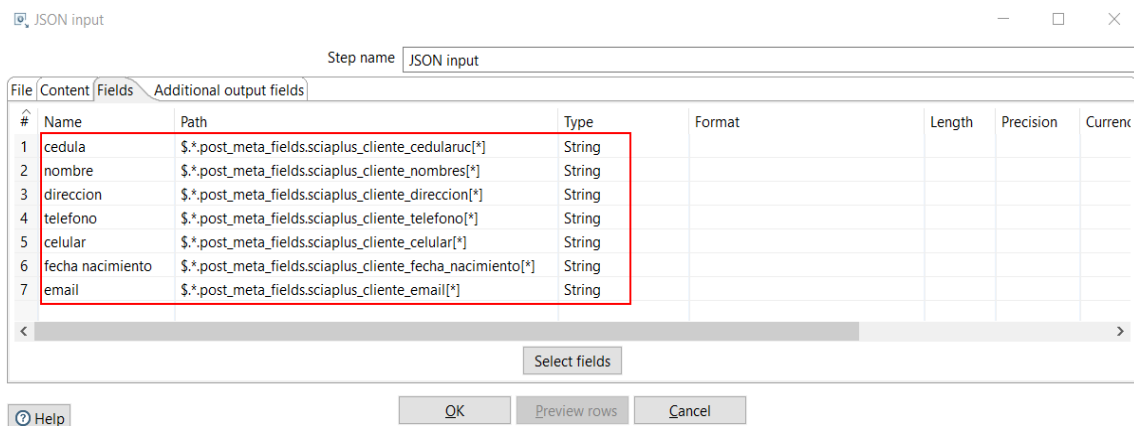


Figura 69: Configuración del elemento *JSON Input*.

Select values tomará estos campos:

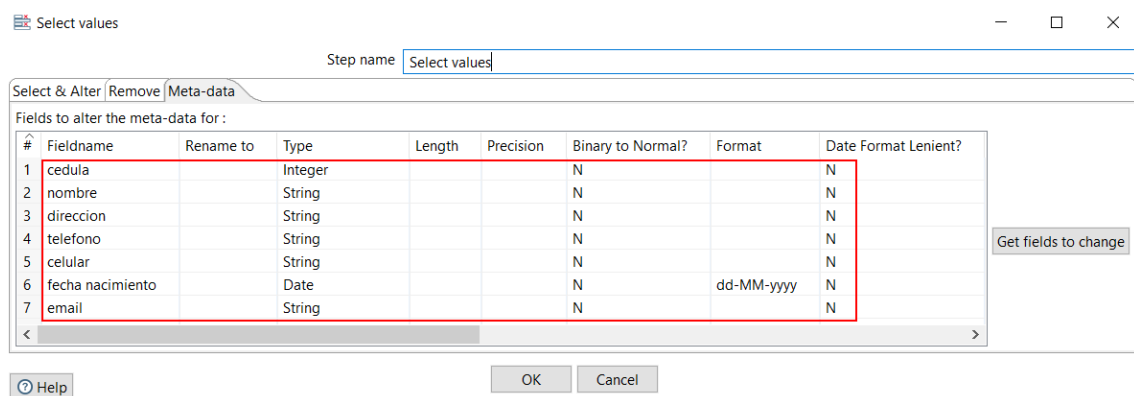


Figura 70: Configuración del elemento *Select Values*.

Insert / Update creará la conexión con la base de datos e inserta o actualiza los datos que son constantemente ingresados:

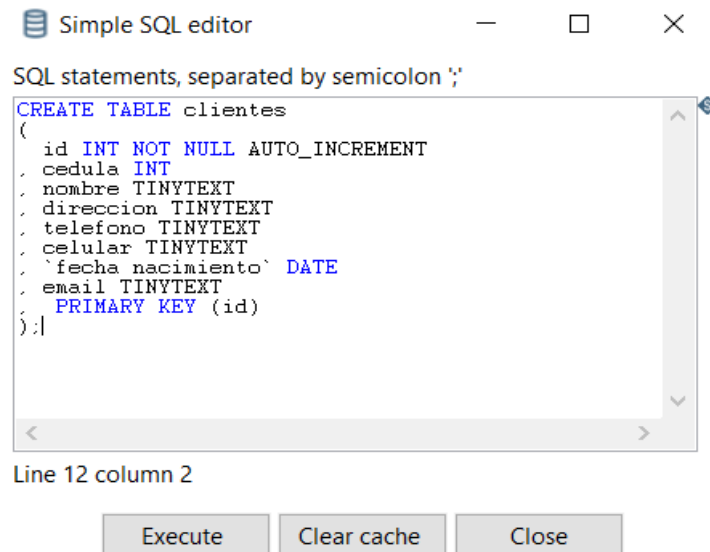


Figura 71: Creación de tabla de Cliente en la base de datos.

Primero se debe especificar el campo de referencia para la actualización y posteriormente los campos a actualizar. En este caso se puede usar el campo cedula el cual es único.

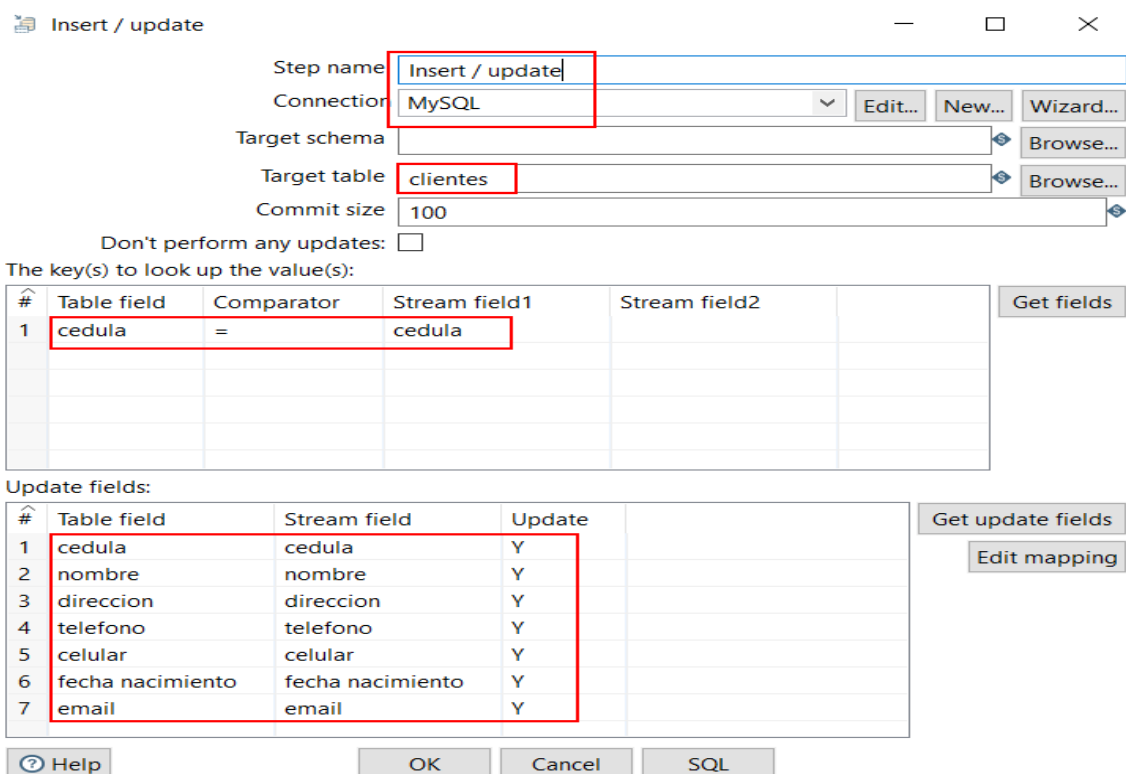


Figura 72: Configuración del elemento *Insert / Update*.

Como resultado al ejecutar este proceso en *Spoon* se observa en el log que se ha ejecutado correctamente y se puede observar el proceso de los datos:

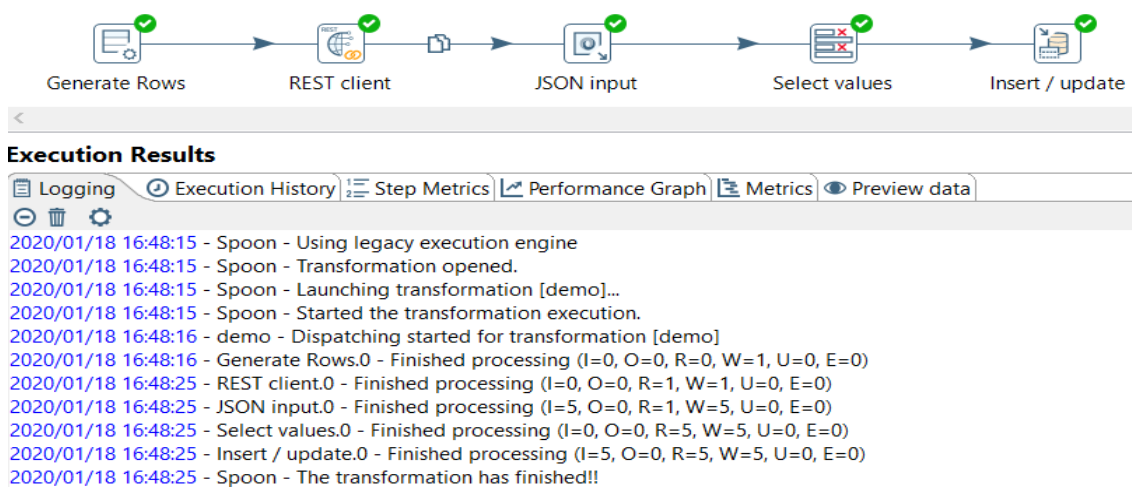


Figura 73: Log del proceso en *Spoon*.

Con esto se puede evidenciar en la base de datos la creación o actualización de los datos extraídos:

SCHMAS

Filter objects

phpmyadmin
powerbi
Tables
clientes
Views
Stored Procedures
Functions
test
wordpress

1 • SELECT * FROM powerbi.clientes;

Limit to 1000 rows

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

id	cedula	nombre	direccion	telefono	celular	fecha nacimiento	email
1	1715859654	Juan Barahona	El Condado	2465896	985477886	1965-07-16	juan@gmail.com
2	1818181820	David	Sna Cariks	256254123	984211554	2019-12-17	dasda@asdasd.com
3	1919191919	Cristina Paz	Cumbaya	2626262626	0907070707	1979-08-06	cristinapaz@gmail.com
4	1818181818	Juan Castro	Condado	2525252525	0908080808	1985-12-10	juancastro@gmail.com
5	1717171717	Pedro Almeida	El Inca	2424242424	0909090909	1990-12-01	pedroalmeida@gmail.com

Figura 74: Consulta de los datos procesados en la base de datos.

De esta forma se han procesado los datos y se puede integrar nuevas fuentes de datos para su almacenamiento. Luego se necesita instalar *Power BI* de la siguiente forma:

- 1) Descargar desde su página oficial <https://powerbi.microsoft.com/es-es/desktop/>
- 2) Seguir los pasos de instalación.

3) Abrir *Power BI*

Agregar la fuente de datos ingresando las credenciales de la base de datos en *Get Data*, de esta forma la alta gerencia puede realizar consultas con respecto a su negocio como por ejemplo, listar los nombres de los clientes según la dirección o en este caso barrio donde están ubicados como se muestra a continuación:

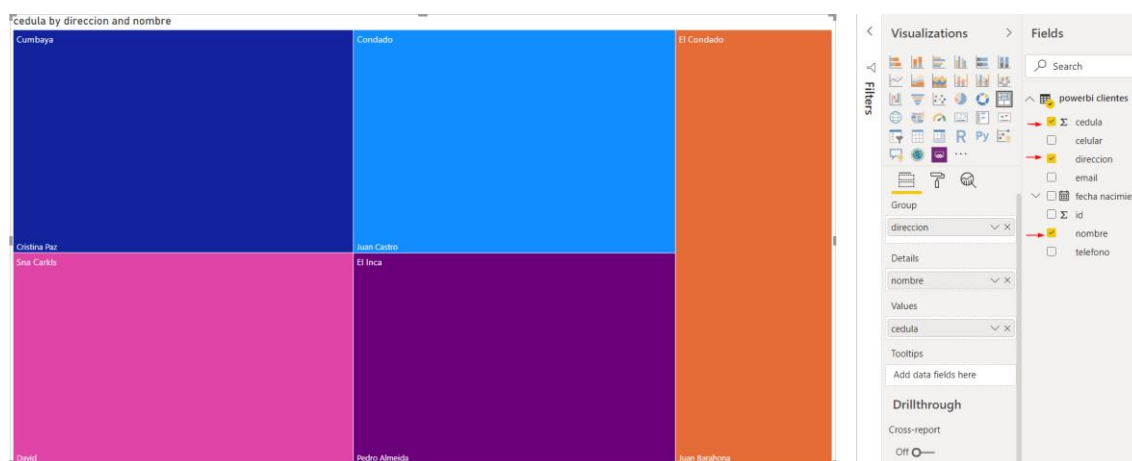


Figura 75: Consulta de datos de cliente en *Power BI*.

3.5.3.5 Daily Scrum

Las reuniones se realizarán en los primeros quince minutos del día en donde se revisará lo realizado, los problemas que se tiene y lo que se realizará durante aquel día con respecto a la extracción de datos de *Pentaho* y el despliegue en *Power BI*.

3.5.3.6 Revisión

Se considera que el producto generado está listo para ser entregado y cumple con los criterios de aceptación, los cambios que puedan generarse no deberán representar un esfuerzo considerable.

3.5.3.7 Retrospectiva

El trabajo realizado sobre este *sprint* ha cumplido con las expectativas, sin embargo, es necesario considerar los siguientes puntos para el siguiente *sprint*:

Ventajas

- 1) Experiencia sobre las tecnologías a implementar.
- 2) Comprensión de los requerimientos del cliente.
- 3) Facilidad de instalaciones.

Desventajas

- 1) Inconvenientes en la creación de reportes debido a las diferentes solicitudes del cliente.
- 2) Falta de conocimiento en la creación de roles para *Custom Post Types*.

4. CAPÍTULO IV. VALIDACIÓN Y PRUEBAS DE SOFTWARE

Las pruebas de caja negra se enfocan a la funcionalidad de los requerimientos sin importar el código fuente, las pruebas de caja blanca valida tanto la funcionalidad como los requerimientos, estas pruebas se presentan a continuación:

4.1 Pruebas de Caja Negra

Tabla 29.

Caso de Prueba TES01.

Caso de Prueba de Software	
ID: TES01	Fecha: 14 – Dic – 2019
Nombre: Creación de Clientes	
Descripción: Proveyendo los siguientes datos, el sistema debe crear un cliente correctamente: <ul style="list-style-type: none"> • Cedula/RUC: 1715859654 • Nombre: Juan Barahona • Dirección: El Condado • Teléfono Fijo: 2465896 	

- Celular: 0985477888
- Fecha Nacimiento: 16 de Julio de 1965
- Correo: juan@gmail.com
- Imagen

Resultado Obtenido: El sistema crea un cliente correctamente.

Editar Cliente [Agregar Nuevo](#)

Post published. [View post](#)

Datos del Cliente

Cédula / RUC Ej: 1715925614001

Nombre Ej: Marco

Dirección Ej: Av. Shyris N45-95 y Av. 6 de Diciembre

Teléfono Fijo Ej: 2462468

Publish [Preview Changes](#)


Status: **Published** [Edit](#)

Visibility: **Public** [Edit](#)

Published on: Jan 1, 2020 at 16:59 [Edit](#)

[Move to Trash](#) [Update](#)

Imagen Destacada



Click the image to edit or update

Figura 76: Resultado obtenido de la creación del Cliente.

Tabla 30.

Caso de Prueba TES02.

Caso de Prueba de Software	
ID: TES02	Fecha: 14 – Dic – 2019
Nombre: Creación de Técnicos	
Descripción: Proveyendo los siguientes datos, el sistema debe crear un técnico correctamente:	
<ul style="list-style-type: none"> • Cédula: 1845965874 • Nombre: Jeovani Diaz • Dirección: Quicentro • Teléfono: 2857458 	

- Celular: 0987581226
- Fecha de Nacimiento: 5 de diciembre de 1975
- Correo Electrónico: jeovani@gmail.com
- Cargo: Software
- Imagen

Resultado Obtenido: El sistema crea un técnico correctamente.

Figura 77: Resultado obtenido de la creación del Técnico.

Tabla 31.

Caso de Prueba TES03.

Caso de Prueba de Software	
ID: TES03	Fecha: 14 – Dic – 2019
Nombre: Creación de Dispositivos	
Descripción: Proveyendo los siguientes datos, el sistema debe crear un dispositivo correctamente:	
<ul style="list-style-type: none"> • Serial: 10000022 • Tipo: Laptop • Marca: Sony • Modelo: Vaio 	

- Sistema Operativo: *Windows*
- *Mainboard*: Intel
- Procesador: Core i3
- RAM: 8 Ram
- Almacenamiento: 512 GB
- Fecha de Compra: 8 de enero del 2012
- Última Modificación: 26 de agosto de 2015
- Persona de Contacto: Pedro
- Imagen
- Pertenece al cliente: Juan Barahona

Resultado Obtenido: El sistema crea un dispositivo correctamente.

The screenshot shows a web interface for editing a device. At the top, there's a header 'Editar Dispositivo' with a 'Agregar Nuevo' button. Below it, a notification says 'Post published. View post'. The main form is titled '10000022 - Sony' and includes a permalink: 'http://sci.aplusecuador.com/dispositivo/10000022-sony/'. The 'Datos del Dispositivo' section contains:

- Serial: 10000022 (with an example 'Ej: 123456')
- Tipo: laptop (with a 'Seleccionar' dropdown)
- Marca: Sony (with a 'Seleccionar' dropdown)

 On the right, the 'Datos del Cliente' section shows 'Cliente' as '1715859654 - Juan Barahona'. The 'Publish' section shows 'Status: Published', 'Visibility: Public', and 'Published on: Jan 1, 2020 at 17:42'. There are 'Move to Trash' and 'Update' buttons at the bottom right.

Figura 78: Resultado obtenido de la creación del Dispositivo.

Tabla 32.

Caso de Prueba TES04.

Caso de Prueba de Software	
ID: TES04	Fecha: 14 – Dic – 2019
Nombre: Creación de Artículos	

Descripción: Proveyendo los siguientes datos, el sistema debe crear un artículo correctamente:

- Serial: 20000033
- Categoría: Almacenamiento
- Marca: *Western Digital*
- Modelo: WDS250G2B0A
- Costo de Compra: 50
- Costo de Venta: 70
- Descripción: Sata 2,5 Pulgadas
- Imagen

Resultado Obtenido: El sistema crea un artículo correctamente.

The screenshot shows a CMS interface for editing an article. At the top, there are buttons for 'Editar Artículo' and 'Agregar Nuevo'. A notification bar indicates 'Post published. View post'. The article ID '20000033' is displayed in a text box, with a 'Permalink' field containing 'http://sci.aplusecuador.com/articulo/20000033/'. Below this is a 'Datos del Artículo' section with three rows: 'Serial' (20000033), 'Categoría' (Almacenamiento), and 'Marca' (Western Digital). To the right, a 'Publish' sidebar shows the article status as 'Published', visibility as 'Public', and a publish date of 'Jan 4, 2020 at 15:14'. An 'Imágen Destacada' section displays an image of a WD Blue 2.5" SSD. Buttons for 'Move to Trash', 'Update', and 'Preview Changes' are also visible.

Figura 79: Resultado obtenido de la creación del Artículo.

Tabla 33.

Caso de Prueba TES05.

Caso de Prueba de Software	
ID: TES05	Fecha: 14 – Dic – 2019
Nombre: Creación de Suscripciones	

Descripción: Proveyendo los siguientes datos, el sistema debe crear una suscripción correctamente:

- Código: 300000035
- Fecha de Activación: 16 de diciembre de 2019
- Fecha de Caducidad: 16 de diciembre de 2020
- Fecha de Notificación: 9 de diciembre de 2020
- Descripción: Suscripción anual de Antivirus
- Imagen
- Pertenece al dispositivo: 10000022
- Utiliza el artículo: 20000001 (Licencia ESET)

Resultado Obtenido: El sistema crea una suscripción correctamente.

Editar Suscripción [Agregar Nueva](#)

Post published. [View post](#)

300000035

Permalink: <http://sci.aplusecuador.com/suscripcion/300000035/> [Edit](#)

Datos de la Suscripción

Código: Ej: 123456

Fecha de Activación: Ej: 20-12-2019

Fecha de Caducidad: Ej: 20-12-2020

Fecha de Notificación: Ej: 15-12-2020

Datos del Dispositivo

Dispositivo: Seleccione

Datos del Artículo

Artículo: Seleccione

Figura 80: Resultado obtenido de la creación de la Suscripción.

Tabla 34.

Caso de Prueba TES06.

Caso de Prueba de Software	
ID: TES06	Fecha: 14 – Dic – 2019
Nombre: Creación de Tickets	

Descripción: Proveyendo los siguientes datos, el sistema debe crear un ticket correctamente:

- Código: 80000005
- Descripción: La computadora necesita un mantenimiento de hardware y software. El cliente indica que prefiere que se la formatee y se cargue software desde cero. Además, agregar una licencia de antivirus. Revisar si el disco duro tiene sectores dañados, si es así cambiarlo.
- Estado: En Proceso
- Imagen
- El técnico que lo resolverá será: 1845965874 (Jeovani Diaz)
- El dispositivo atendido es: 10000022
- Implementará la suscripción: 30000035

Resultado Obtenido: El sistema crea un ticket correctamente.

Editar Ticket [Agregar Nuevo](#)

Post published. [View post](#)

80000005

Permalink: <http://sciplusecuador.com/ticket/80000005/> [Edit](#)

Datos del Ticket

Código: Ej: 123456

Descripción:
 La computadora necesita un mantenimiento de hardware y software. El cliente indica que prefiere que se la formatee y se cargue software desde cero. Además, agregar una licencia de antivirus. Revisar si el disco duro tiene sectores dañados, si es así cambiarlo.

F: Se desea formatear, cargar programas, cambiar el disco duro y la fuente de poder

Estado
 Estado:
 Seleccione

Datos del Técnico
 Técnico:
 Seleccione

Figura 81: Resultado superior obtenido de la creación del Ticket.

Artículos

Artículo 1

Artículo:
 Seleccione

Cantidad:

Datos del Dispositivo
 Dispositivo:
 Seleccione

Datos de la Suscripciones
 Suscripción:
 Seleccione

Publish

Figura 82: Resultado inferior obtenido de la creación del Ticket.

Tabla 35.

Caso de Prueba TES07.

Caso de Prueba de Software	
ID: TES07	Fecha: 14 – Dic – 2019
Nombre: Creación de Facturas	

Descripción: Proveyendo los siguientes datos, el sistema debe crear una factura correctamente:

- Código: 001001000018
- Cliente a facturar: 1715859654 (Juan Barahona)
- Ticket finalizado a facturar: 80000005
- Además, desea comprar el artículo: 10000058 (Kingston Technology)
- Subtotal: 115.00
- Descuento: 0.00
- IVA: 13.80
- Total: 128.80
- Imagen

Resultado Obtenido: El sistema crea una factura correctamente.

Editar Factura [Agregar Nuevo](#)

Post draft updated. [Preview post](#)

001001000018

Permalink: <http://sci.aplusecuador.com/factura/001001000018/> [Edit](#)

Tickets

Seleccione un Ticket

503 - La computadora necesita un mantenimiento de hardware y software. El cliente indica que prefiere que se la formatee y se cargu. ▾

80000005

[Agregar](#)

20000033 - Western Digital | 70.00 | 1

20000001 - ESET | 30.00 | 1

10000058 - Kingston Technology | 15.00 | 1

Datos del Cliente

Cliente

1715859654 - Juan Barahona ▾

Seleccione un cliente

Publish

[Save Draft](#) [Preview](#)

Status: Draft [Edit](#)

Visibility: Public [Edit](#)

Publish immediately [Edit](#)

[Move to Trash](#) [Publish](#)

Datos de la Factura

Figura 83: Resultado superior obtenido de la creación de la Factura.

Código 001001000018 *Ej: 123456*


Subtotal \$ 115.00 *Ej: \$100*

Descuento \$ 0.00 *Ej: \$0*

IVA \$ 13.80 *Ej: 12%*

Total \$ 128.80 *Ej: \$120*

Imagen Destacada



Click the image to edit or update

[Eliminar Imagen Destacada](#)

Figura 84: Resultado inferior obtenido de la creación de la Factura.

Tabla 36.

Caso de Prueba TES08.

Caso de Prueba de Software	
ID: TES08	Fecha: 14 – Dic – 2019
Nombre: Procesamiento de datos por parte de Spoon (Pentaho).	

Descripción: El sistema debe extraer los datos desde el *endpoint* `http://sci.aplusecuador.com/wp-json/wp/v2/clientesrest/` cuyo resultado en formato *JSON* será procesado y almacenado en la base de datos *MySQL* según los campos que tenga el Cliente concordando con el tipo de dato.

Resultado Obtenido: El sistema extrae, procesa y guarda datos procesados a partir de la información generada en el Sistema de Control Interno de *A Plus Computer*.

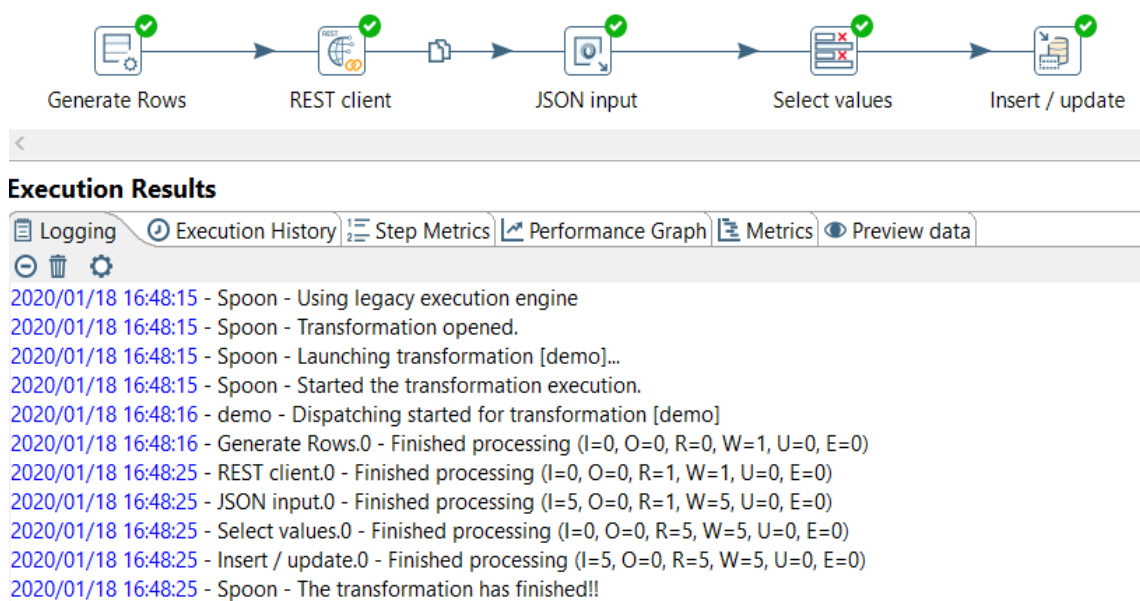


Figura 85: Resultado obtenido del procesamiento de datos en *Spoon*.

Tabla 37.

Caso de Prueba TES09.

Caso de Prueba de Software	
ID: TES09	Fecha: 14 – Dic – 2019
Nombre: Presentación de datos en <i>Power BI</i> .	
<p>Descripción: La herramienta <i>Power BI</i> debe poder mostrar diferentes tipos de gráficos con respecto a los datos obtenidos de la base de datos <i>MySQL</i> que <i>Spoon</i> previamente los ha procesado. Debe existir la posibilidad de filtrar datos, agregar texto descriptivo, mostrar diferentes gráficos en una misma área, entre otras opciones.</p>	

Resultado Obtenido: *Power BI* presenta correctamente gráficos a partir de los datos proporcionados.

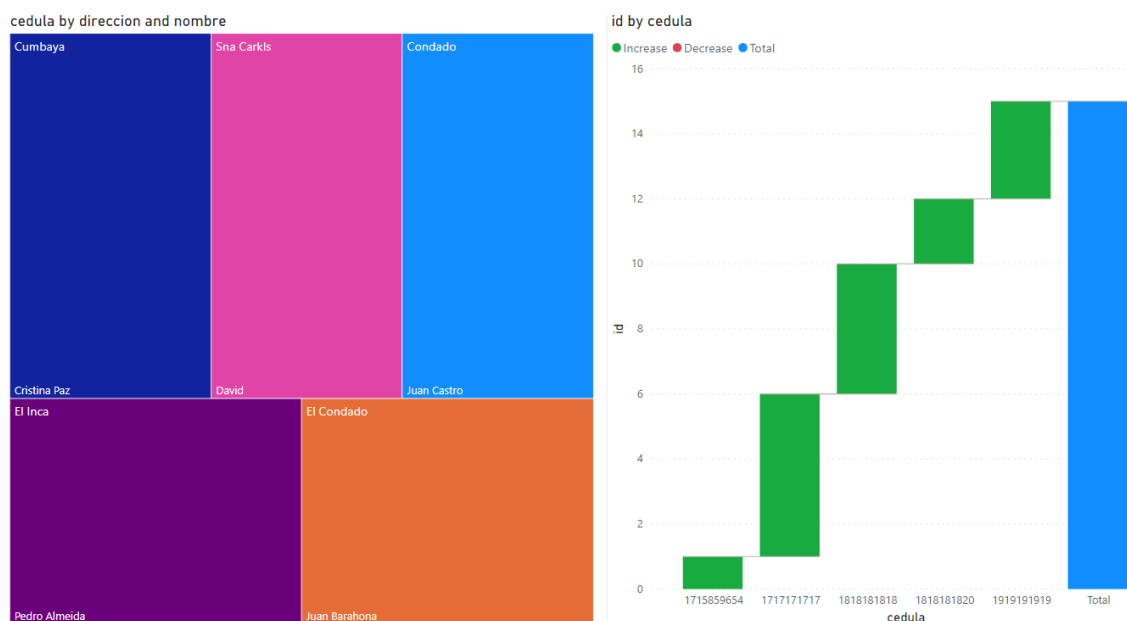


Figura 86: Resultado obtenido de la presentación de datos en *Power BI*.

Tabla 38.

Caso de Prueba TES10.

Caso de Prueba de Software	
ID: TES10	Fecha: 14 – Dic – 2019
Nombre: Creación de usuarios con rol personalizado	
Descripción: El sistema debe crear un rol personalizado para los técnicos el cual pueda acceder únicamente a los módulos del sistema, no a las opciones de configuración ni código del sistema.	
Resultado Obtenido: El sistema crea un usuario con rol de técnico que accede únicamente a los módulos del sistema.	

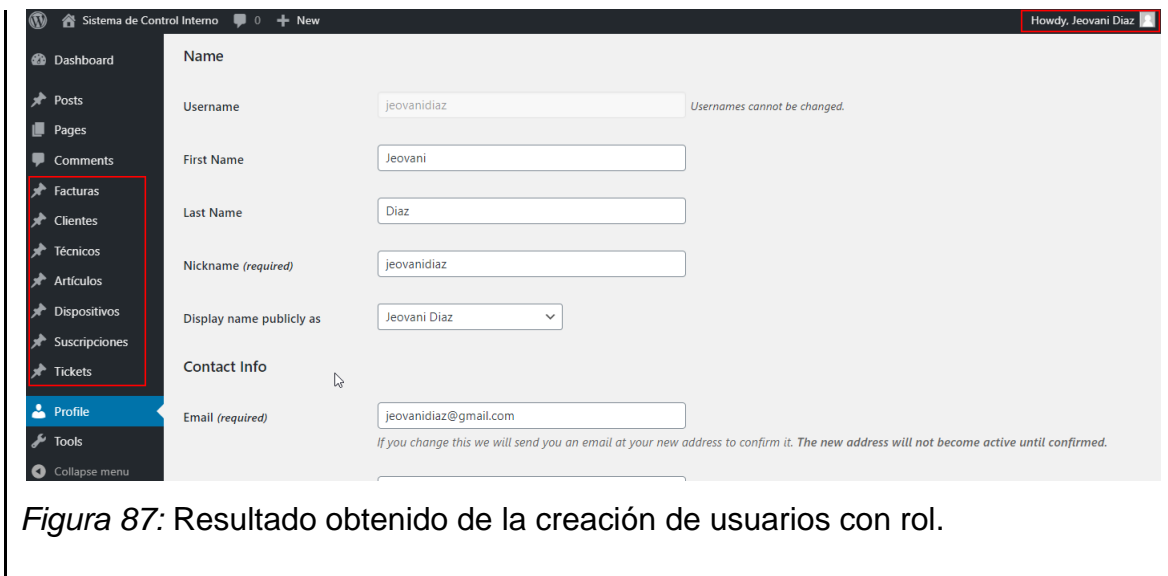


Figura 87: Resultado obtenido de la creación de usuarios con rol.

4.2 Pruebas de Caja Blanca

Estas pruebas se basan en la estructura de archivos y organización de código cuyo objetivo es la mantenibilidad del sistema y evitar una deuda técnica, la explicación de esta estructura de archivos y del código se la ha realizado durante el Capítulo 3 *Propuesta de Solución* y más específicamente en la sección 3.5 *Desarrollo / Sprints*, en donde se han implementado buenas prácticas de desarrollo que indica la documentación de *WordPress*.

4.3 Aprobación del Sistema por A Plus Computer



REPARACION Y MANTENIMIENTO

Quito, 16 de diciembre del 2019

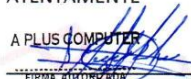
A PLUS COMPUTER
MARCO ANTONIO ESCOBAR TIRADO
GERENTE

Por medio de la presente hago constar que el Sistema Web de Control Interno con implementación de BI desarrollado por MARCO DAVID ESCOBAR ALEAGA con C.I. 171592561-4 dentro de su proceso de titulación como Ingeniero en Sistemas de Computación e Informática, ha sido aceptado por nuestra empresa bajo los requerimientos acordados.

Sin más por el momento, quedo ante cualquier aclaración o duda.

ATENTAMENTE

A PLUS COMPUTER



FRANCISCA ESCOBAR
Marco Escobar
C.I. 1001610300

Av de los Shyris N4595 y Av. 6 de Diciembre Of. 204 (Frente Ex. Disc. Cerebro)
Telfs: 2462 468 • Cel: 09 9476 855 • E-mail: aplus@aplusecuador.com • www.aplusecuador.com

Figura 88: Carta de aprobación del sistema por parte de A Plus Computer.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Desde hace mucho tiempo atrás existen sistemas de control interno que a pesar de no contar con tecnología controlaban los procesos internos de las empresas o industrias; sin embargo, implementar software en el desarrollo de estos sistemas potencia sus resultados.

Existe software que cumple con las necesidades individuales de cada módulo de *A Plus Computer*, muchos de ellos tienen hasta funcionalidades extras que en un momento dado podrían implementarlas; sin embargo, *A Plus Computer* necesita que estas herramientas estén integradas, presenten un servicio de *Api Rest* y su desarrollo sea económico; por lo que un software personalizado con estas características no se ha encontrado.

A Plus Computer es un claro ejemplo de una PYME ecuatoriana que a pesar de la calidad de su atención y experiencia en su campo no ha logrado tener un crecimiento considerable por la falta de implementación de tecnología que les permita expandirse o crecer sin perder el control de sus procesos.

Gracias a la metodología *Scrum* se puede estimar con bastante precisión la duración y esfuerzo que tomará un proyecto de desarrollo de software, de esto depende el éxito o fracaso del proyecto.

WordPress como un *CMS* para el desarrollo de páginas web se considera muy intuitivo y fácil de aprenderlo, además de esto la factibilidad del desarrollo de *plugins*, la analítica sobre los datos, la actual tendencia de consumo de micro servicios e integración con otros sistemas sumado a la amplia comunidad y soporte que mantiene a un costo muy bajo, lo posiciona como una solución apta para el desarrollo de sistemas actuales para pequeñas y medianas empresas que buscan mejorar sus procesos a través de eficientes sistemas informáticos.

La implementación de inteligencia de negocios sobre empresas que mantienen sistemas informáticos es muy amigable debido a que está pensada en manejar diferentes tipos de fuentes y estructuras de datos, además esta se extiende y se

enfoca en los resultados según la granularidad de las preguntas que la alta gerencia busque resolver.

5.2 Recomendaciones

Recomiendo leer la documentación de *WordPress* para tener un conocimiento sólido sobre los objetivos a los que se quiere llegar y las herramientas con las que se cuenta para lograrlos.

Sugiero que el código sea descargado localmente, modificado y posteriormente subirlo al servidor para una mayor rapidez de desarrollo y evitar un alto uso de la memoria del hosting.

Recomiendo leer la documentación de *CMB2* ya que con esta librería se puede extender su funcionalidad en base a requerimientos más complejos.

Sugiero aprender *Ajax* sobre *WordPress* para la extracción de datos desde la base de datos.

Recomiendo enfáticamente entender teóricamente la diferencia entre *Custom Post Types*, *Taxonomías* y *Términos* tomando en cuenta la administración que tienen en la base de datos de *WordPress* ya que la estructura de guardado es diferente a la que comúnmente se conoce.

REFERENCIAS

- A Plus Computer. (2019). *Empresa A Plus Computer*. Recuperado el 10 de Septiembre de 2019 de <http://www.aplusecuador.com>
- Alonso, C. (2014). Orientaciones para implementar una gestión basada en procesos. Chile.
- Billwerk. (2019). Recuperado el 26 de Septiembre de 2019 de <https://billwerk.es/producto/>
- Bitbucket. (2019). *Atlassian*. Recuperado el 26 de Noviembre de 2019 de <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>
- Chubko, A. (2019). *ProAndroidDev*. Recuperado el 1 de Diciembre de 2019 de <https://proandroiddev.com/how-to-set-up-an-efficient-development-workflow-with-git-and-ci-cd-5e8916f6bece?gi=d870a4cd1dd8>
- Codex WordPress. (2019). *WordPress.Org*. Recuperado el 27 de Noviembre de 2019 de https://codex.wordpress.org/Database_Description#Database_Diagram
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2009). *Información básica de Scrum*. España: Creative Commons.
- Freshworks. (2019). *Freshdesk*. Recuperado el 3 de Diciembre de 2019 de <https://freshdesk.com/latam/>
- Git. (2019). *Git*. Recuperado el 7 de Noviembre de 2019 de <https://git-scm.com/book/es/v2/Fundamentos-de-Git-Guardando-cambios-en-el-Repositorio>
- Git. (2019). *Git SCM*. Recuperado el 18 de Noviembre de 2019 de <https://git-scm.com>

- GoDaddy. (2019). *GoDaddy*. Recuperado el 3 de Enero de 2020 de <https://www.godaddy.com/>
- Haas, H., & Brown, A. (2004). *W3C*. Recuperado el 20 de Diciembre de 2019 de <https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>
- Hernández, H., Cardona, D., & Del Rio, J. (2017). *Scielo*. Recuperado el 20 de Diciembre de 2019 de https://scielo.conicyt.cl/scielo.php?pid=S0718-07642017000500003&script=sci_arttext
- Leary, S. (2010). *Beginning WordPress 3*. Estados Unidos: Apress.
- López, B. (2019). *Ciudadano 2 Cero*. Recuperado el 7 de Noviembre de 2019 de <https://www.ciudadano2cero.com/que-es-un-hosting-web-tipos-alojamiento-cual-elegir/>
- Mantilla, S. A. (2018). *Auditoría del Control Interno*. Bogotá: ECOE.
- Martín, P. (2019). *itop Academy*. Recuperado el 27 de Noviembre de 2019 de <https://www.itop.academy/blog/item/que-es-pentaho-data-integration-pdi-y-para-que-sirve.html>
- Microsoft. (2019). *Power BI*. Recuperado el 2 de Diciembre de 2019 de <https://powerbi.microsoft.com/es-es/what-is-power-bi/>
- Ñeco, R. (2003). *Apuntes de sistemas de control*. Editorial Club Universitario.
- Negash, S., & Gray, P. (2008). Business Intelligence. En *Handbook on Decision Support Systems 2* (págs. 175-193). Berlin: Handbook.
- Nogales, P., & Rosales, A. (2020). *AGILIDAD & SCRUM*. Quito.
- Patel, S., Rathod, V., & Prajapati, J. (2011). Performance Analysis of Content Management. *International Journal of Computer Applications*, 5.
- PHP. (2019). *PHP Manual*. Recuperado el 2 de Diciembre de 2019 de <https://www.php.net/manual/en/faq.general.php>

Poulson, I. (2015). *Delicious Brains*. Recuperado el 27 de noviembre de 2019 de <https://deliciousbrains.com/tour-wordpress-database/>

Práctica del Desarrollo de Software. (2019). *Práctica del Desarrollo de Software*. Recuperado el 3 de Enero de 2019 de <https://sites.google.com/site/practicadesarrollosoft/temario/sistemas-de-versionado-de-codigo>

Rouse, M. (2015). *Search Data Center Techtarget*. Recuperado el 10 de Septiembre de 2019 de <https://searchdatacenter.techtarget.com/es/definicion/MySQL>

TPV Gratuito 123. (2019). Recuperado el 11 de Septiembre de 2019 de <https://tpvgratis.com/facturacion-pymes-y-autonomos>

WordPress. (2019). *WordPress.org*. Recuperado el 15 de Septiembre de 2019 de <https://developer.wordpress.org/themes/basics/template-hierarchy/>

WordPress. (2019). *WordPress.Org*. Recuperado el 11 de Septiembre de 2019 de <https://developer.wordpress.org/themes/basics/post-types/>

