



**DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN**

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO DE SISTEMAS E INFORMÁTICA**

**TEMA: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA
CONTROL DE PROYECTOS DE LA EMPRESA METALEC**

AUTOR: COBA GÓMEZ, ALBERTO VLADIMIR

DIRECTOR: ING. CAMPAÑA, MAURICIO

SANGOLQUÍ, ABRIL 2016

AUTORÍA DE RESPONSABILIDAD



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Yo, *COBA GÓMEZ ALBERTO VLADIMIR* con cédula de identidad N° 1713173084, declaro que este trabajo de titulación “*ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DE PROYECTOS DE LA EMPRESA METALEC*” ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, abril del 2016



COBA GÓMEZ ALBERTO VLADIMIR

CC 1713173084

AUTORIZACIÓN




DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, *COBA GÓMEZ ALBERTO VLADIMIR* autorizo a la UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE publicar en la Biblioteca Virtual de la Institución el presente trabajo de titulación “*ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DE PROYECTOS DE LA EMPRESA METALEC*”, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, abril 2016



COBA GÓMEZ, ALBERTO VLADIMIR
CC 1713173084

AUTORIZACIÓN




DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, *COBA GÓMEZ ALBERTO VLADIMIR* autorizo a la UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE publicar en la Biblioteca Virtual de la Institución el presente trabajo de titulación “*ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DE PROYECTOS DE LA EMPRESA METALEC*”, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, abril 2016



COBA GÓMEZ, ALBERTO VLADIMIR
CC 1713173084

DEDICATORIA

Dedico el presente trabajo a mi familia, en especial a mis amados Padres, Hernán Coba y Margarita Gómez, verdaderos ejemplos de esfuerzo, dedicación y amor incondicional hacia sus hijos, motores que gracias al amor, valores y principios inculcados durante toda su vida a su familia, hacen que el mayor de los esfuerzo significa nada como retribución a lo recibido.

Para ustedes que jamás nos dejaron solos, que siempre estuvieron en los momentos más difíciles con palabras sabias, voz tranquilizadora y que han sido parte de mis mayores alegrías. A ustedes que me enseñaron el valor de la solidaridad, humildad, honradez e integridad. A ustedes mí razón de ser, ejemplo y bendición.

Alberto Vladimir Coba Gómez

AGRADECIMIENTO

Al esfuerzo y dedicación de mis Padres por siempre tener como principio de superación del ser humano la preparación intelectual, gracias por todo lo recibido, jamás podre retribuir todos sus esfuerzos.

A mis hermanas y hermano Diana, Ximena y Rommel que a imagen de mis Padres siempre me han brindado su apoyo, ejemplo y cariño siendo verdaderos ejemplos para mí.

A mi familia Miriam, Daniel y Martín fuentes de inspiración y el mayor de los motivos para que cualquier esfuerzo no sea suficiente para conseguir nuestros objetivos.

A mis amigos incondicionales que siempre han estado para brindar una mano solidaria, fraterna y desinteresada, muchas gracias por todo Michelle y Martín Carlier.

A todo el cuerpo docente de la Universidad de las Fuerzas Armadas (ESPE), por todos los conocimientos brindados, de manera especial a todos los docentes que lograron encender la curiosidad, la necesidad de investigar para construir nuestro conocimiento.

Mi gratitud infinita al apoyo incondicional tanto académico como humano brindado por el Ing. Mauricio Campaña durante mi vida universitaria, que me hace llevar el mejor de los ejemplos del significado de humildad, sencillez y calidad humana que un profesional debe llevar en su vida.

Muchas gracias por tanto

Alberto Vladimir Coba Gómez

TABLA DE CONTENIDOS

CAPÍTULO 1	1
1 GENERALIDADES	1
1.1 Introducción	1
1.2 Planteamiento del Problema	1
1.3 Alcance	4
1.4 Objetivos	7
1.4.1 Objetivo General	7
1.4.2 Objetivos Específicos.....	7
CAPÍTULO 2	9
2 MARCO TEORICO	9
2.1 Arquitectura	9
2.2 Definiciones.....	9
2.2.1 PHP	9
2.2.2 Bases de Datos MySQL.....	13
2.2.3 JQuery	15
2.2.4 HTML	30
2.2.5 CSS.....	41
2.2.6 API Google Maps.....	45
2.2.7 Código de Barras	48
2.2.8 Servidor Web Apache.....	49
2.3 Metodología SCRUM.....	51
2.3.1 Introducción	51
2.3.2 Proceso de SCRUM.....	52
2.3.3 Ejecución de la Iteración	54
2.3.4 Inspección y adaptación.....	54
2.3.5 Actividades	54
2.3.6 Ejecución de Iteración o Sprint	55
2.3.7 Reunión de Sincronización.....	55
2.3.8 Demostración de Requisitos Completados	55
2.3.9 Retrospectiva o Sprint Retrospective	56
2.3.10 Redefinición de lista de requerimiento y cambios.	56
2.3.11 Roles.....	56

2.3.12	Herramientas	57
CAPÍTULO 3	58
3	ANÁLISIS Y DISEÑO.....	58
3.1	Ejecución del Proyecto.....	58
3.1.1	Planificación del Proyecto.....	58
3.1.2	Alcance del Sistema	58
3.1.3	Equipo de Trabajo	61
3.1.4	Lista de requisitos priorizados	61
3.1.5	Diseño	64
3.1.6	Modelos de base de datos	64
3.1.7	Sprint Nro. 1 “Usuarios y Seguridades”	67
3.1.8	Sprint Nro. 2 Administración.....	75
3.1.9	Sprint Nro. 3 Gestión de Proyectos.....	83
3.1.10	Sprint Nro. 4 “Registro de Pagos y proformas”	105
3.1.11	Sprint Nro. 5 “Reportes”	112
CAPÍTULO 4	115
4	IMPLEMENTACIÓN.....	115
4.1	Implementación	115
4.1.1	Instalación de arquitectura definida.....	115
4.1.2	Instalación de Servidor.....	116
4.1.3	Instalación de sistema operativo	116
4.1.4	Implementación	130
4.1.5	Resultados.....	134
4.2	Organización	134
4.3	Seguimiento	136
4.4	Costos.....	137
CAPÍTULO 5	138
5	CONCLUSIONES Y RECOMENDACIONES.....	138
5.1	Conclusiones	138
5.2	Recomendaciones.....	138
6	REFERENCIAS BIBLIOGRÁFICAS	139
7	Bibliografía	139

INDICE DE FIGURAS

Figura 1. Principales clientes de MySQL:.....	15
Figura 2. Crecimiento de PHP:.....	15
Figura 3. Funcionamiento de AJAX:.....	16
Figura 4. Evento JQuery:.....	21
Figura 5. Funciones de JQuery:.....	25
Figura 6. Ejemplo de árbol de elementos HTML.....	26
Figura 7. Primer sitio web:	31
Figura 8. Nuevos elementos de la estructura semántica de HTML5.....	34
Figura 9. Código de barras IAN	49
Figura 10. Resultados de encuesta de servidores web 2014:.....	50
Figura 11. Planificación de la iteración o Sprint	53
Figura 12. Modelo conceptual de la aplicación.....	65
Figura 13. Modelo físico de la aplicación	66
Figura 14. Caso de uso autenticación	69
Figura 15. Caso de uso autenticación	70
Figura 16. Modelo de base de datos Sprint Nro. 1	74
Figura 17. Historia del Sprint Nro. 1	75
Figura 18. Avance del Sprint Nro. 1.....	75
Figura 19. Casos de uso menú rol	77
Figura 20. Casos de uso administración de proyecto	77
Figura 21. Casos de uso administración de catálogos	77

Figura 22. Historias del usuario y avance de Sprint Nro. 2.....	82
Figura 23. Seguimiento de Sprint Nro. 2.....	82
Figura 24. Casos de uso creación de proyecto	90
Figura 25. Casos de uso pre diseño	90
Figura 26. Casos de uso aprobación del proyecto	91
Figura 27. Casos de uso diseño definitivo de proyecto	91
Figura 28. Casos de uso producción asignación.....	91
Figura 29. Casos de uso producción corte.....	92
Figura 30. Casos de uso producción armando.....	92
Figura 31. Casos de uso instalación	92
Figura 32. Casos de uso cierre de proyecto.....	93
Figura 33. Avance del proyecto Historias del usuario	104
Figura 34. Seguimiento del proyecto.....	105
Figura 35. Casos de Uso registro de pago	107
Figura 36. Casos de uso proforma.....	107
Figura 37. Sprint Nro.4 Historias del usuario.....	111
Figura 38. Sprint Nro.4 Seguimiento de Sprint Nro.4.....	111
Figura 39. Casos de uso reportes.....	113
Figura 40. Instalación CentOS, selección de lenguaje	116
Figura 41. Instalación CentOS, configuración de instalación	117
Figura 42. Instalación CentOS, configuración contraseña root.....	117
Figura 43. Instalación CentOS, instalación finalizada	117
Figura 44. Arranque de sistema operativo CentOS	118
Figura 45. Configuración de servidor y conexiones de red nmtui	118

Figura 46. Cambio de nombre de host.....	119
Figura 47. Reinicio de hostnamed	119
Figura 48. Verificación del estado de hostname	120
Figura 49. Actualizando dependencias yum.....	120
Figura 50. Instalación de servicio httpd	121
Figura 51. Estado del servicio httpd	121
Figura 52. Acceso a servidor web mediante explorador	122
Figura 53. Instalación de paquete PHP.....	123
Figura 54. Archivo de prueba phpinfo	123
Figura 55. Página de phpinfo.....	124
Figura 56. Instalación de MariaDB y configuración de inicio	124
Figura 57. Estado del servicio de MariaDB	125
Figura 58. Instalación segura de servidor de base de datos.....	125
Figura 59. Agregar el repositorio EPEL.....	126
Figura 60. Actualización de bases de yum con el repositorio EPEL.....	126
Figura 61. Configuración de acceso para host al phpMyAdmin	127
Figura 62. Acceso al sistema phpMyAdmin	128
Figura 63. phpMyAdmin con acceso al servidor MariaDB	128
Figura 64. Configuración de zona horaria php.ini.....	129
Figura 65. Configuración de tamaño de archivos para carga	129
Figura 66. Creación de base de datos	130
Figura 67. Creación de usuario para base de datos	131
Figura 68. Carga de script de base de datos	131
Figura 69. Estructura de base de datos creada en el servidor	132

Figura 70. Configuración de parámetros de base de datos	132
Figura 71. Pantalla de inicio, comprobación de funcionamiento	133
Figura 72. Página principal del sistema.....	133

INDICE DE TABLAS

Tabla 1. Versiones de JQuery.....	17
Tabla 2. Diferencias entre métodos HTML, GET y POST	30
Tabla 3. Etiquetas reemplazadas en HTML5	33
Tabla 4. Principales elementos HTML 5.....	35
Tabla 5. Nuevos elementos de formularios	37
Tabla 6. Nuevas etiqueta input	37
Tabla 7. Atributos nuevos para etiquetas input	38
Tabla 8. Equipo de trabajo bajo metodología SCRUM.....	61
Tabla 9. Lista de requerimientos priorizados bajo metodología SCRUM	61
Tabla 10. Requerimientos funcionales Sprint Nro. 1	67
Tabla 11. Historias del usuario Sprint Nro. 1	68
Tabla 12. Roles para el Sprint Nro. 1	69
Tabla 13. Actores para el Sprint Nro. 1	69
Tabla 14. Especificación caso de uso autenticación.....	70
Tabla 15. Especificación caso de uso registro de acceso	71
Tabla 16. Especificación caso de uso recuperar contraseña.....	71
Tabla 17. Especificación caso de uso gestión de usuarios	72
Tabla 18. Especificación caso de uso asignar roles.....	73
Tabla 19. Requerimientos funcionales Sprint Nro. 2	75
Tabla 20. Equipo de trabajo Sprint Nro. 2.....	76
Tabla 21. Actores del Sprint Nro. 2.....	76
Tabla 22. Especificación caso de uso mostrar menú y obtener rol	78
Tabla 23. Especificación caso de uso administración de proyecto	79

Tabla 24. Especificación caso de uso buscar proyecto.....	80
Tabla 25. Especificación caso de lista de catalogo.....	80
Tabla 26. Especificación caso de edición de catalogo	81
Tabla 27. Requerimientos funcionales creación de proyecto	83
Tabla 28. Requerimientos funcionales pre diseño.....	84
Tabla 29. Requerimientos funcionales aprobación contrato	85
Tabla 30. Requerimientos funcionales diseño definitivo	86
Tabla 31. Requerimientos funcionales producción	86
Tabla 32. Requerimientos funcionales producción	87
Tabla 33. Requerimientos funcionales armado	87
Tabla 34. Requerimientos funcionales instalación.....	88
Tabla 35. Requerimientos funcionales cierre de proyecto	89
Tabla 36. Equipo de trabajo Sprint Nro. 3.....	89
Tabla 37. Actores del sistema Sprint Nro. 3.....	90
Tabla 38. Especificación de casos de uso crear proyecto.....	93
Tabla 39. Especificación de casos de uso generar mapa	94
Tabla 40. Especificación de casos de uso crear cliente	95
Tabla 41. Especificación de casos de uso generar solicitud	96
Tabla 42. Especificación de casos de uso bandeja de entrada.....	96
Tabla 43. Especificación de casos de uso cargar archivo.....	97
Tabla 44. Especificación de casos de uso ver proyecto.....	98
Tabla 45. Especificación de casos de uso ver proyecto.....	99
Tabla 46. Especificación de casos de uso generar contrato.....	100
Tabla 47. Especificación de casos de uso finalizar etapa producción	100

Tabla 48. Especificación de casos de uso finalizar instalación	101
Tabla 49. Especificación de casos de uso generar acta	102
Tabla 50. Especificación de casos de uso generar orden.....	102
Tabla 51. Especificación de casos de uso cerrar proyecto	103
Tabla 52. Requerimientos funcionales registro de pago	105
Tabla 53. Requerimientos funcionales proforma	106
Tabla 54. Equipo de trabajo Sprint Nro. 4.....	106
Tabla 55. Actores del sistema Sprint Nro. 4.....	107
Tabla 56. Especificación de casos de uso cerrar proyecto	107
Tabla 57. Especificación de casos de uso listado de pagos	108
Tabla 58. Especificación de casos de uso listado de pagos	109
Tabla 59. Especificación de casos de uso listado de pagos	109
Tabla 60. Especificación de casos de uso listado de pagos	110
Tabla 61. Requerimientos funcionales reportes	112
Tabla 62. Equipo de trabajo Sprint Nro. 5.....	112
Tabla 63. Actores del sistema Sprint Nro. 6.....	113
Tabla 64. Especificación de casos de uso Generar Reporte	113
Tabla 65. Especificación de casos de uso Generar Reporte	114

RESUMEN

El presente trabajo presenta el análisis, diseño e implementación del sistema de control de proyectos de la empresa METALEC. En el área de la industria el proceso de producción es sumamente importante para mantener y cumplir con estándares que permitan la competitividad en el mercado, por ello la implementación de un proceso de gestión de producción principalmente en la PYMES, logra la optimización de los recursos, mejoramiento continuo y una visión clara de responsabilidades permitiendo evaluar resultados e indicadores que aporte a mantener y mejorar la producción. Para ello en este trabajo se utilizó la metodología SCRUM que permite un ágil desarrollo en conjunto con la tecnología PHP, JQuery, Bootstrap que garantiza una implementación con presupuestos de hardware y software ajustados al sector y que permite llevar un control de cada uno de los proyectos permitiendo cumplir con términos contractuales y mantener la competitividad en el sector.

Palabras Clave:

CONTROL DE PROYECTOS,

METODOLOGIA ÁGIL SCRUM,

LENGUAJE PHP,

HERRAMIENTA JQUERY,

LIBRERIAS BOOSTRAP,

BASE DE DATOS MYSQL

ABSTRACT

This document presents the analysis, design and implementation of the METALEC company control system projects. Production process is extremely important in industrial area to maintain and comply with standards that allow high competition in the market, therefore a management production process implementation on SMEs mainly, obtains the resources optimization, continuous improvement and a clear vision of responsibilities allowing to evaluate results and indicators that contribute to maintain and improve production in general. To obtain those results, it is important to work with SCRUM methodology to agile development in conjunction with PHP, JQuery, Bootstrap technology that ensures implementation with hardware and software budgets that allows to control each of the projects enabling compliance with contractual terms and maintain competitiveness in the sector.

Keywords:

PROJECT CONTROL,

SCRUM METHODOLOGY,

PHP LANGUAGE,

JQUERY TOOLS,

BOOTSTRAP LIBRARY,

MYSQL DATABASE

CAPÍTULO 1

GENERALIDADES

1.1 Introducción

Los procesos productivos de las PYMES principalmente de los sectores de manufactura y construcción no cuentan con procesos estandarizados que permitan un control de sus procesos internos. El uso de herramientas manuales de control se ha limitado muchas veces a registros manuales o a través de hojas electrónicas que limitan en un alto grado la posibilidad de llevar un control detallado de actividades y compromisos establecidos.

Esta problemática inicialmente no se presenta directamente para los procesos productivos empresariales, sin embargo conforme la producción y demanda se incrementan los estos se hacen tangibles en las diferentes áreas que interactúan.

El principal problema que se evidencia es de carácter administrativo, principalmente en el seguimiento de cumplimiento de plazos y control de cambios afectando de forma directa al cumplimiento de los contratos que se traducen en pérdidas económicas por pago de multas o pérdida de materiales.

La tecnología de la información en este punto se transforma en un aliado fundamental para los procesos productivos, permitiendo a través de un adecuado diseño contar con las herramientas para el registro, administración, seguimiento y ejecución de los proyectos.

En este contexto, la empresa METALEC con el objetivo principal de mejorar su proceso de producción y control, busca una solución a través de un sistema informático capaz de hacer el acompañamiento en todas las áreas del proceso productivo manteniendo un registro actualizado de cada una de las actividades principales hasta la finalización, permitiendo brindar un seguimiento y control en tiempo real del avance de cada uno de los proyectos así como emitir alertas y comunicación efectiva entre los involucrados.

1.2 Planteamiento del Problema

Con el interés de comprender el problema es importante conocer el proceso manual que actualmente requiere ser automatizado. El proceso inicia cuando un cliente acude a las oficinas a fin de obtener una cotización de la solución modular que elija. Para ello

se realiza una orden de cotización la misma que es entregada a uno de los diseñadores a fin de que el mismo realice el proceso de medición y modelado que permite obtener un costo aproximado que incluye diseño, fabricación e instalación.

Una vez emitida la proforma el cliente está en la posibilidad de hacer modificaciones tanto en costos como en equipamiento hasta llegar a un acuerdo final en cuanto a productos y valores.

Ya aprobada la proforma se emite un contrato en el cual se establece formalmente el ámbito y alcance del proyecto. Una vez suscrito el contrato se inicia el proceso de diseño final en el cual se determina materiales, colores, accesorios y acabados específicos, mismos que son preparados por el encargado del proyecto que no necesariamente es la misma persona que preparo el diseño inicial de proforma.

Concluida la fase de diseño se remite a fábrica los modelos y materiales para su corte y producción. Esta actividad cuyo responsable es el supervisor se encarga de la coordinación del proceso de producción y de instalación.

El supervisor una vez finalizado el proceso de producción, emite una orden de instalación y coordina la visita al cliente donde con un equipo realiza la instalación de todos los muebles y accesorios conforme al contrato.

Una vez instalado, la orden de instalación y entrega son remitidos a la unidad administrativa a fin de que se procese el pago total del proyecto. El seguimiento del proceso de producción es realizado por la gerencia con el interés de garantizar cada una de las rutas críticas de los proyecto a fin de cumplir con lo establecido en los contratos suscritos, compromisos de entrega y diseños definitivos. Estos son puntos fundamentales para la imagen empresarial, planificación, atención al cliente que permita evitar posibles problemas legales posteriores.

Al realizar un primer análisis de los procesos que se manejan dentro de la empresa, se puede identificar que el proceso de gestión de proyectos/contratos se realiza de forma manual, y se usan herramientas de ofimática para llevar un registro y control del avance.

La información correspondiente a diseño, pagos y modificaciones se mantiene dispersa y frecuentemente es necesario acudir al archivo físico a fin de obtener información del proyecto generando tiempos de respuesta más largos y un complejo control de cumplimiento.

El uso constante del archivo físico hace susceptible que la documentación se vea afectada por su manipulación incluso la pérdida definitiva de la misma.

En cada uno de los proyectos se asigna un responsable quien es el encargado del contrato sin embargo esto hace complejo el cambio de responsables de proyecto ya que no hay suficiente información que permita que el proyecto sea independiente del diseñador.

El diseño cuenta con un software especializado mismo que es el encargado de la generación de los modelos 3D y planos requeridos para el área de corte y producción. Al ser generada esta información no se cuenta con un control de cambios de los mismos por lo cual es susceptible a errores y en muchos casos la información no está disponible ya que si el archivo no es remitido, no es posible continuar con el proceso de producción.

No se cuenta con un método que permita valorar correctamente la carga de trabajo asignado que permita optimizar y realizar una mejor distribución de la carga de trabajo al talento humano existente en el área de diseño.

La inexistencia de reportes, hace necesario que los mismos sean construidos de forma manual lo cual toma un tiempo considerable y limita la toma de decisiones de forma rápida disminuyendo los tiempos de respuesta.

En consecuencia el manejo de los proyectos puede ser susceptible a varios riesgos, mismos que tienen una probabilidad de ocurrencia a tomar en cuenta. Algunos de los riesgos identificados son:

- Pérdida de información importante para el proyecto como diseños definitivos, observaciones, cambios causaría el retraso, mal uso de materiales y malestar al cliente.

- Falta de información del estado de los proyectos causaría problemas de coordinación entre cada una de las áreas del proceso productivo.
- Ausencia de personal encargado del proyecto causaría que no se cuente con suficiente información a fin de que el proyecto pueda continuar.
- Falta de hitos de control, medición tiempos de respuesta y seguimiento causaría que no se esté usando los recursos de forma correcta.
- La no definición de un proceso estandarizado causaría desorganización al momento de ejecutar el proyecto.

Con estos antecedentes se puede considerar oportuno la implementación y definición de un proceso con el cual se pueda mitigar los riesgos identificados así como automatizar el proceso desde el inicio de la atención al cliente hasta la entrega conforme al mismo.

1.3 Alcance

En este contexto se ha establecido la necesidad de diseñar, desarrollar e implementar un sistema de información que permita llevar el control de los proyectos de la empresa METALEC permitiendo de esta manera contar con una herramienta que mantenga el proceso de producción estandarizado y que brinde información de forma oportuna del estado, avance y documentación de cada uno de los proyectos.

Para ello se ha establecido que debe contar con los siguientes módulos:

- **Módulo de seguridades**
 - **Administración de usuarios y roles**

Permitirá la creación y edición de usuarios que acceden al sistema así como los roles y permisos.

Adicionalmente mantendrá la administración de los registros de acciones y accesos del sistema.
- **Módulo de administración del sistema**
 - **Administración de catálogos**

Permitirá la creación, edición de todos los catálogos dinámicos y parametrización del sistema.

- **Módulo de utilitarios**
 - **Envío de notificaciones y correos electrónicos**

Herramienta para el envío de correos electrónicos de forma automática basado es plantillas personalizables para cada una de la acciones que sea requerido.
 - **Administración de localizaciones GPS**

Manejo de localizaciones geográficas que permita obtener resultados sin realizar implementaciones adicionales.
 - **Lector y Generador de código de barras**

Administración de parámetros para generación así como lectura de código de barras.
 - **Generador de Reportes**

Manejo de plantillas para reportes así como un generador de PDF que permita que los reportes sean portables.
- **Módulo ventas**
 - **Gestión de Clientes**

Permitirá el registro de clientes.
 - **Creación de proyectos**

Permitirá la creación de nuevos proyectos con su caracterización necesaria.
 - **Generación de contratos**

Editor de documentos contractuales basado en plantillas que permita su autogeneración
 - **Proformas y registro de Pagos**

Creación de proformas para la aprobación de clientes así como el registro de pagos de contratos vigentes.
 - **Asignación de proyectos**

Permite la asignación de un líder de proyecto que será el encargado del diseño y seguimiento del proyecto.
 - **Reportes**

Generador de reportes de los proyectos

- **Módulo de diseño**
 - **Gestión de proyectos.**

Acceso los proyectos de responsabilidad de cada uno de los usuarios del sistema para el registro de avance y carga de archivos relacionados al proyecto.
 - **Generación de cotizaciones.**

Permite la generación de documentos de cotización en base a plantillas personalizables.
 - **Carga de documentos digitales**

Permitirá la carga de distintos archivos digitales con versionamiento a nivel de la aplicación.
- **Módulo de producción**
 - **Gestión de proyectos.**

Acceso los proyectos de responsabilidad de cada uno de los usuarios del sistema para el registro de avance y carga de archivos relacionados al proyecto.
 - **Asignación y generación de órdenes de producción**

Permitirá el designar los responsables del proceso de producción en fábrica para el proyecto, usuario que será responsable de la actualización del estado de avance.
 - **Avance de producción**

Permitirá indicar el avance del proceso de producción por parte del usuario responsable.
 - **Generación orden de instalación y entrega**

Generación de documentos para la instalación y entrega basado en plantillas personalizables

Por la extensión de los módulos requeridos, se hace necesario contar con una metodología que permita un desarrollo ágil, por ello la principal opción como metodología es SCRUM, que permite tener entregables validados en periodos cortos de tiempo, así como flexibilizar el desarrollo permitiendo reducir el riesgo en el proyecto.

Sus principales características son:

- Rápido cambio de requisitos
- Proceso constructivo incremental y basado en la iteración (Sprints) que no deben superar los 30 días.
- Elevar la productividad
- Mantener reuniones diarias con el equipo de desarrollo.

Las actividades asociadas a la metodología establecen:

- Planificación de la iteración del proyecto – Sprint planning
- Ejecución de la Iteración - Sprint
- Reunión diaria de sincronización – Scrum daily meeting
- Demostración de requisitos completados dentro del sprint – Sprint Review
- Retrospectiva – Sprint Retrospective
- Re planificación del proyecto – Sprint Planning

Una vez concluida la fase de diseño el desarrollo del sistema se realizará usando una arquitectura distribuida bajo el uso de tecnología PHP que permite contar con sistemas de rápida respuesta y recursos de hardware y software que se ajustan a los niveles alcanzables de las PYMES.

1.4 Objetivos

1.4.1 Objetivo General

Analizar, Diseñar e implementar un sistema Web para la gestión de proyectos de producción de la empresa METALEC, que permita registrar, administrar y guiar el proceso productivo a través de los diferentes actores que intervienen, permitiendo mejorar los índices de efectividad dentro de la empresa así como contar con un registro digital de todos los proyectos garantizando la persistencia de la información de los mismos.

1.4.2 Objetivos Específicos

- Analizar los requerimientos funcionales necesarios para el desarrollo del sistema bajo una metodología de desarrollo ágil.

- Realizar un estudio de las diferentes tecnologías utilizadas dentro de un proyecto web a fin de establecer los beneficios de aplicarlas en el proyecto.
- Definir la tecnología más adecuada para el desarrollo del sistema así como su aporte al desarrollo del sistema que permita ajustarse a los requerimientos de la empresa.
- Definir la arquitectura de software y hardware necesario para el funcionamiento adecuado del sistema.
- Desarrollar los módulos requeridos en la tecnología y metodología definida que permita automatizar el proceso de producción.

CAPÍTULO 2

MARCO TEORICO

2.1 Arquitectura

Los diferentes conjuntos o elementos que componen un sistema están definidos por la arquitectura de software. Este conjunto de elementos indica el patrón o modelo que se usa en la solución de software y representa un guía para el proceso de desarrollo de software entre cada uno de los involucrados en el proyecto. La norma IEEE Std 1471-2000 define la arquitectura de software como la “organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán”. (Desarrollo Web, 2016)

Los sistemas web por su característica con aplicaciones basadas en varias capas mismas que interactúan a fin de cumplir con su función específica y que generalmente hacen referencia a un patrón. Por lo general la arquitectura estar definida por capas que implementan el manejo de datos, procesos y la vista de los sistemas permitiendo de cierta manera definir un orden claro dentro de un patrón como lo es modelo vista controlador, a lo cual se puede agregar capas correspondientes a otros patrones así como servicios.

2.2 Definiciones

2.2.1 PHP

Es un lenguaje de programación de código abierto orientado al desarrollo de aplicaciones web y cuyo nombre se origina del acrónimo PHP Hypertextual Preprocessor. SU principal característica es su simplicidad que permite un rápido aprendizaje sin dejar de lado operaciones de mayor complejidad que lo vuelve atractivo para los desarrolladores de aplicaciones.

Su misma facilidad se debe a que permite la ejecución de sus scripts del lado del servidor produciendo código HTML para el cliente, permitiendo de esta manera que la lógica de programación, como se mencionó antes, únicamente se ejecute en el servidor.

Para ello PHP hace uso de sus etiquetas especiales `<?php ?>` mismas que indican que las instrucciones contenidas en su interior corresponden ser ejecutadas por el servidor.

Otra de sus características principales es que puede ser empleado independientemente del sistema operativo del servidor entre los cuales se puede mencionar Linux, CentOS, Windows, HP-UX, Solaris, OpenBSD, Mac OS X, RISC OS. Por esta característica PHP actualmente es uno de los lenguajes más usados para la creación de CMS y sistemas de información donde se destacan desarrollos como Facebook, Wordpress, Joomla, Moodle entre otros.

Una característica que lo hace muy apreciable al momento de elegir PHP como lenguaje para desarrollar sistemas es su rapidez, la misma que se evidencia en el manejo de datos almacenados en bases de datos.

Su compatibilidad con el manejo de varios motores de base de datos lo han convertido en una alternativa seria para desarrollo de aplicaciones de gran acogida permitiendo conexiones a bases de datos como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL principalmente.

2.2.1.1 Historia

El inicio de PHP se da en los años 1994 cuando Rasmus Lerdorf crea un conjunto de código denominado PHP/FI 1994, el cual era un conjunto de ficheros binario Common Gateway Interface CGI escrito en lenguaje C y cuya principal función en ese momento era rastrear visitas en línea. Este conjunto de ficheros fue denominado Personal Home Page Tools conocido como PHP Tools siendo este el inicio del lenguaje como se conoce hoy.

En una posterior evolución el código fue escrito nuevamente y evolucionó incluyendo funcionalidades que le permitió ya el manejo de bases de datos y generando el salto necesario para que el lenguaje como tal pueda ser utilizado en el diseño de sitios dinámicos, sin embargo su manejo aún era limitado siendo incluso necesario la generación de código dentro de comentarios HTML para que se haga la inyección en las páginas web.

Uno de los principales hechos y motivo de un mayor crecimiento se dio en el año de 1995 cuando Lerdorf publica el código fuente, permitiendo de este modo que la comunidad se involucre y continúe el desarrollo de PHP lo que incluía la depuración y corrección de errores.

En el mes de octubre del año 1995 se libera el código bajo la denominación de Personal Home Page Construction Kit siendo considerada ya en ese momento la primera versión de una interfaz avanzada de scripts y cuya característica de éxito era su semejanza a la sintaxis utilizada en el lenguaje C. (PHP, 2016)

2.2.1.2 Evolución

Version 1.0

Denominada como PHP Tools fue un conjunto de librerías CGI orientado a búsquedas en línea y de uso personal, misma que fue evolucionando y teniendo varias denominaciones como PHP FI, FI y cuyo principal salto fue la liberación del código para el desarrollo del mismo de modo colaborativo permitiendo así la corrección y depuración del mismo, obteniendo soporte para bases de datos y manejando una sintaxis similar al muy difundido lenguaje C.

Versión 2.0

En el año de 1997 la versión 2.0 de PHP misma que tuvo un crecimiento considerable siendo que se estima llegaba a 60.000 dominio según una encuesta realizada por Netcraft en el años de 1998 avizorando que PHP sería uno de los principales lenguajes para el desarrollo de páginas web dinámicas, sin embargo esta versión no tuvo un desarrollo considerable ya que al ser liberada se estaba trabajando en la siguiente versión de PHP. Es importante indicar que en esta versión aún era necesario el uso de comentarios HTML para la codificación de los scripts PHP en las páginas.

Versión 3.0

Andi Gutmans y Zeev Suraski mientras desarrollaban un proyecto de comercio electrónico proponen a Rasmus Lerdorf colaborar en el redesarrollo de PHP FI producto de lo cual se libera la versión 3.0 misma que lleva el nombre PHP acrónimo

de PHP: Hypertext Preprocessor como es conocido hasta ña actualidad y cuyas principales características fueron la madurez del lenguaje como tal permitiendo múltiples conexiones a bases de datos y protocolos que fueron uno de los principales atractivos sumados a su sencillez para que miles de desarrolladores lo adquieran para sus proyectos de desarrollo. Sus principales características fueron el soporte de programación orientada a objetos no completa como en lenguajes de mayor madurez como Java y una sintaxis de mayor potencia logrando así estar presente en un estimado del 10% de servidores web a nivel mundial.

Versión 4.0

A mediados del año 199 se introduce la versión 4.0 de PHP misma que basada en la ya exitosa versión 3.0 busca brindar una mayor robustez para aplicaciones más complejas siendo su motor denominado Zend haciendo referencia a Andi Gutmans y Zeev Suraski y liberado oficialmente el año 2000 siendo sus principales características la introducción de la “superglobales” `$_GET`, `$_POST` y `$_SESSION`.

Por motivos de seguridad se elimina el soporte para la opción `register_globals` que se consideró un problema de seguridad en las aplicaciones. Adicionalmente se introduce CLI para codificación en línea y CGI.

Versión 5.0

Lanzado en el año 2004 se caracteriza por estar compuesto por el motor Zend 2.0 mismo que permite una mejor visibilidad de métodos y clases en orientación a objetos incluyendo métodos finales, abstractos, interfaces y determinación de tipos permitiendo que los parámetros de las funciones sean objetos.

El manejo de los objetos se hace mediante referencia o manejadores que hace que las variables contengan una referencia a un objeto y no una copia del mismo permitiendo optimización de memoria.

Una de las principales y que permitirá un mayor desarrollo de otras tecnologías es el soporte nativo de cadenas JSON. Se elimina elementos tales como `register_globals`, `safe_mode`, `sesión_register()`, `sesión_unregister()` y se incluye un nuevo motor para el majeo de bucles.

Actualmente es la versión usada por la mayoría de aplicaciones web basadas en PHP y que se considera ya madura.

2.2.1.3 Nuevas Versiones

La versión 6.0 ha sido abandonada y muchas de las características consideradas para esta versión han sido añadidas a las diferentes versiones 5.x, esto ha desencadenado que la versión 6.0 no sea liberada y la versión 7.0 sea la próxima versión.

2.2.2 Bases de Datos MySQL

MySQL se presenta como una de las principales bases de datos utilizadas para el desarrollo de aplicaciones web, caracterizándose por su flexibilidad para la recuperación de datos, característica propia de ISAM que es un motor no transaccional que favorece la consulta de datos.

La gran difusión de MySQL esta principalmente dada por su licencia de GNU GPL que a pesar de no estar siendo desarrollado en comunidad le ha dado gran acogida. Una de las características que ha permitido su desarrollo, ha sido que es muy común relacionar y encontrar juntos a MySQL con PHP, lenguaje que ha tenido un desarrollo creciente y en comunidad orientado a sitios web principalmente, y que juntos se han tornado en una de las principales herramientas para el desarrollador web.

La existencia de distribuciones con prácticamente un mínimo requerimiento de configuración también ha sido un factor para que este conjunto sea elegido encontrando entre las principales distribuciones XAMP, LAMP, WAMP. Estas distribuciones multiplataforma están compuestas principalmente por un servidor Apache junto con PHP y MySQL que gracias a paneles de control y herramientas como PhpMyAdmin se puede usar de forma rápida facilitando el desarrollo sobre esta tecnología.

En la actualidad MySQL es soportado por los principales lenguajes de programación, entre los cuales se menciona Java, PHP, Perl, Python así como las conexiones mediante interfaces ODBC. Esto ha permitido que sitios de reconocimiento mundial la usen en algunas de sus divisiones de software como lo son Facebook, Twitter, Flickr, YouTube y Wikipedia entre otros. Su desarrollo continúa y su crecimiento se muestra respaldado por Sun Microsystem inicialmente, quienes a

partir de febrero del 2008 al comprar MySQL mantuvieron su desarrollo para posteriormente ser comprados por Oracle quienes hasta el momento mantienen el desarrollo.

2.2.2.1 Historia

Se estima que MySQL se remonta a los años noventa y como muchos de los desarrollos actuales estaba basado en la necesidad por mejorar lo existente, en este caso en particular Michael Windenis crea sus propias funciones para conectar tablas basado en ISAM, sin embargo se encontraron algunas limitaciones propias de MySQL, por lo cual se desarrollan nuevas funciones principalmente en lenguaje ANSI C logrando una interfaz SQL que era compatible con MySQL que no contaba con integridad referencial ni transacciones a pesar de lo cual ya llamo la atención de los desarrolladores de sitios web que reconocieron su velocidad. Por ello se funda MySQL AB empresa encargada de su desarrollo durante sus versiones 3.23 Alfa a mediados del año 1999.

Una vez ya establecida la empresa MySQL AB, generó expectativas y en el año 2005 su motor InnoDB desarrollado por la empresa Innobase OY que comprada por Oracle acerco cada vez más a la empresa a una negociación misma que se dio en el año 2008 cuando Sun Microsystems adquiere MySQL y que posteriormente fue adquirida por Oracle Corporation haciéndose de los derechos de la misma y quienes actualmente son los encargados de su desarrollo tanto la versión Privada como Libre. El respaldo alcanzado por una empresa de gran envergadura como Oracle no hizo más que afianzar su nivel de preferencia al incrementar características que aumentaron su robustez sin perder su rapidez, característica fundamental de MySQL. Esto ha derivado en una gran preferencia: (MySQL, 2016).

ESPACIO EN BLANCO INTENCIONAL

Partial List of MySQL Customers		
Web/End-User Customers	Embedded/ISV Customers	Telecom Customers
• Amazon.com	• Adobe	• Alcatel-Lucent
• Bank of America	• Airbus	• AT&T Wireless
• BBC News	• CA NetQoS	• BT Plusnet
• Disney	• Dell Kace	• Cable & Wireless
• Facebook	• EMC	• Cell C
• Google	• F-Secure	• Cisco
• Lufthansa	• F5 Networks	• Comcast
• NASA	• Hewlett Packard	• Ericsson
• Priceline.com	• Intel	• France Telecom
• Puma	• McAfee	• Italtel
• Shopatron	• Motorola	• Oi Telecom
• Twitter	• Sage Group	• Motorola
• Walmart	• SAP BusinessObjects	• Siemens
• YouTube	• SonicWALL	• Telenor
• Zappos	• Symantec	• Teligent

Figura 1. Principales clientes de MySQL, (MYSQL, 2016)

No es ninguna novedad que grandes empresas a nivel mundial confían los bienes más preciados como lo es la información a esta base de datos.

2.2.2.2 Evolución



Figura 2. Crecimiento de PHP, (Databasefriends, 2016)

2.2.3 JQuery

Es una librería que permite simplificar el manejo de páginas HTML en el cliente a través de Javascript. Esta funcionalidad asíncrona permite que las llamadas al servidor se las realice en segundo plano y de forma específica, reduciendo la cantidad de datos

obtenidos del servidor y brindando una mejor experiencia al usuario al eliminar la recarga total de la página. (W3C JQuery, 2016)

Esto se logra a través de la manipulación de HTML y del árbol DOM (Document Object Model) con la ayuda de AJAX, acrónimo de Asynchronous JavaScript And XML, que es una combinación de cuatro tecnologías:

- XHTML
- DOM
- XMLHttpRequest
- XML

El uso adecuado de estas tecnologías permite que muchas de las operaciones se realicen en segundo plano y se carguen sin interferir con otros elementos del DOM.

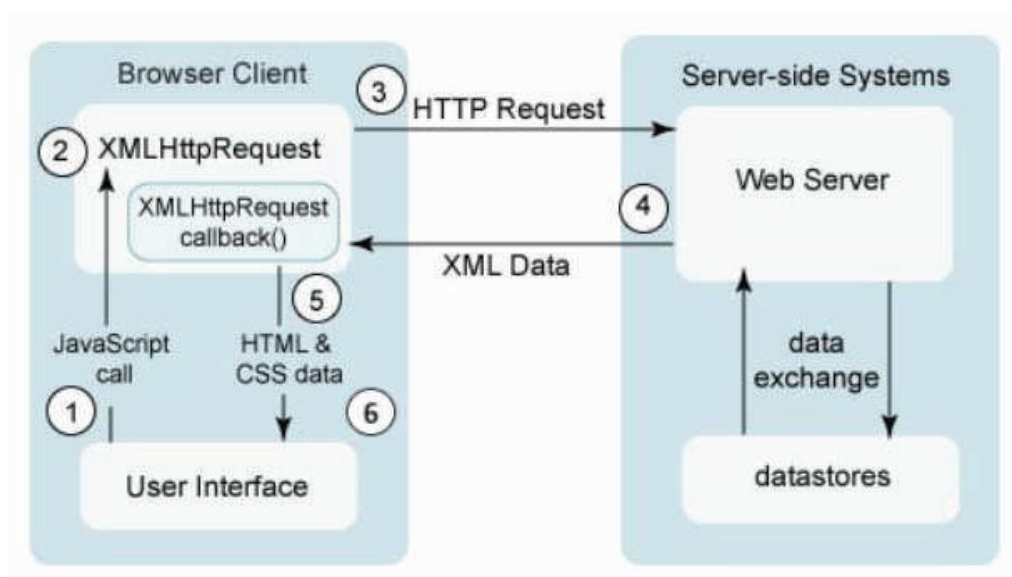


Figura 3. Funcionamiento de AJAX, *(Universidad de Washington, 2016)*

2.2.3.1 Historia

Hace su aparición en enero del 2006 en el BarCamp NYC. Su primera versión estable es la 1.0 liberada en agosto del 2006, desde entonces su desarrollo se mantiene activo y se ha mantenido consistentemente como se muestra a continuación:

Tabla 1.
Versiones de JQuery

<i>Versión</i>	<i>Fecha</i>	<i>Características</i>
Versión 1.0	Agosto 2006	Corrección de errores y primera versión estable
Versión 2.0	Septiembre 2007	Incremento de selectores Partial load Ajax Efectos .map(), .prevAll(), .nextAll().
Versión 1.3	Enero 2009	Sizzle, motor CSS HTML injection Rewrite Live Events
Versión 1.4	Enero 2010	Creación de portal propio de JQuery
Versión 1.5	Enero 2011	Re escritura de AJAX corrigiendo errores
Versión 1.6	Mayo 2011	Adiciona especificaciones HTML5 Corrección errores AJAX
Versión 1.7	Noviembre 2011	Nuevas API .on() .off() Mejora el soporte para HTML5
Versión 1.8	Agosto 2012	Re arquitectura de Sizzle, Prefijado automático de CSS. Corrección de errores
Versión 1.9	Enero 2013	Mayor de soporte de exploradores Nuevo método .finish()
Versión 1.10	Mayo 2014	Corrección de errores
Versión 1.11	Enero 2014	

2.2.3.2 Evolución

Es imposible hablar de la evolución de JQuery sin mencionar a JavaScript, que es uno de los principales lenguajes de programación que tiene su principal característica que no se ejecuta en el servidor, siendo necesario un programa adicional para su ejecución, en este caso el navegador web por lo cual su ejecución la realiza en el cliente.

JQuery al estar compuesto por varias librerías JScript aprovecha su ejecución en el cliente permitiendo la manipulación del árbol DOM con la ventaja que sus librerías están orientadas a facilitar considerablemente la programación.

Esta característica es la que ha propiciado su gran aceptación a avance, que ha permitido realizar tareas sencillas de manejo de las páginas hasta permitir la interacción con bases de datos principalmente a través del uso AJAX y PHP.

Todas estas características lograron que en 2008 Nokia y Microsoft anunciaran que brindarían soporte para JQuery, y se orientaría para .Net en el caso de Microsoft y para el desarrollo de diferentes componentes en el caso de Nokia.

La creciente comunidad ha logrado que JQuery tenga un versionamiento muy acelerado lo cual si bien significa un rápido desarrollo de esta herramienta, también puede causar inconvenientes por su velocidad de cambio, sin embargo de ello es una de las herramientas más atractivas para el desarrollo por su relativa facilidad, velocidad de desarrollo y compatibilidad con CSS3 sumado al soporte de las principales plataformas web y móviles.

2.2.3.3 Características JQuery

Es una librería orientada a facilitar el desarrollo de aplicaciones web permitiendo manipulación de HTML/DOM, CSS, Eventos HTML, Efectos y AJAX.

Para usar las librerías JQuery se puede realizar de dos formas principales:

La primera es necesario descargar del sitio jquery.com las librerías mismas que deberán ser referenciadas en el <head> de nuestras páginas para que sean accesibles, para ello incluir:

```
<head>
<script src="jquery-2.1.4.min.js"></script>
</head>
```

La segunda opción para usar JQuery consiste en agregar la referencia usando CDN (Content Delivery Network), lo cual evita que la librería sea descargada y únicamente se referencia a la url de este contenido siendo los principales Google y Microsoft.

```
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/
jquery.min.js"></script>
</head>
```

Una vez que se ha realizado la referencia a JQuery, sus librerías ya están disponibles para su uso dentro de la aplicación.

Su sintaxis está definida principalmente por los caracteres `$()` mismos que permiten seleccionar a los elementos del DOM a utilizar para realizar diferentes tareas.

```
Sintaxis
#(selector).acción
```

Al ser JQuery prácticamente código JavaScript y por lo tanto se ejecuta del lado del cliente, es importante garantizar que cualquier acción que se realice en la página se realice una vez completada la transmisión de datos entre el servidor y el cliente, para ello es necesario agregar como una buena práctica que el código JQuery se ejecute una vez que el documento este cargado por completo.

Esta buena práctica se la realiza gracias al evento Ready del documento

```
$(document).ready(function() {
    // Código a ejecutarse una vez que se cargue el
    documento
});
```

Su funcionamiento se explica prácticamente solo, indicando que se ejecute cualquier código dentro de `function()` una vez cargado el documento.

Para el uso de cada uno de los elementos existentes en el DOM de la página web JQuery utilizar los selectores, mismo que permite la manipulación de las

características, datos de dichos elementos permitiendo seleccionarlos de diferentes formas: tipos, atributos, identificadores, clases que facilitan y dan un gran poder de selección tanto individual como grupal.

Las principales formas de usar selectores son:

Por Id:

```
HTML
<input id="miIdentificador" type="button" value="test"/>

jQuery
$("#miIdentificador")
```

Por Clase

```
HTML
<input id="miIdentificador" class="miClase" type="button"
value="test"/>

jQuery
$(".miClase")
```

El conjunto de funciones que se generan a través de los selectores tiene una gran probabilidad de que puedan ser usados en varias páginas o ser genéricos para acciones comunes dentro de un sistema, para ello se puede agrupar las funciones en archivos independientes de las páginas HTML, para ello se los puede unificar en un archivo JavaScript mismo que deberá ser adjuntado a las páginas HTML que requieran estas funciones para lo cual se deberá incluir en la cabecera <head> su llamado.

```
HTML
<head>
<script src="miArchivojQuery.js"></script>
</head>
```

2.2.3.4 Eventos en JQuery

Cada uno de los elementos dentro de DOM puede ejecutar reaccionar fruto de una acción específica sobre él, mismo que se denomina Evento.

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

Figura 4. Evento JQuery, (W3C JQuery, 2016)

La mayoría de eventos del DOM tiene uno relacionado en JQuery, para lo cual se debe usar conjuntamente con el selector para crear una función que se ejecute al momento de ocurrir el evento indicado.

```

$("selector").click(function() {
  //Código a ejecutarse al hacer clic en el selector
});

```

Con el interés de poder asociar varios eventos a él o los elementos del selector establecido, se utiliza la función on.

```

$("p").on({
  mouseenter: function() {
    $(this).css("background-color", "lightgray");
  },
  click: function() {
    $(this).css("background-color", "yellow");
  }
});

```

Para eliminar la asociación realizada con la función on se puede realizar usando la función off.

2.2.3.5 Efectos JQuery

Permite manipular los efectos de los elementos seleccionados entre los cuales se menciona:

Hide/Show/Toggle: Oculta o muestra elementos, en el caso de toggle lo hace de forma automática entre las dos funciones.

Fade: Mediante la variación de visibilidad hace aparecer u oculta elementos con el uso de las funciones `fadeIn()` y `fadeOut()`.

Sliding: Permite desplegar de forma vertical un panel usando las funciones `slideDown()`, `slideUp()` y `slideToggle()`.

Animate: Permite animar un elemento mediante varios parámetros de velocidad como `slow` o `fast` en un tiempo y distancia determinados. Las características de la animación requieren la definición de estas en el CSS del elemento.

Stop: permite detener un efecto o animación al hacer la llamada a esta función.

Callback: Al usar efectos en los elementos y si es necesario que se ejecute otra acción o función junto a ellos podría generarse errores, para ello se cuenta con la opción de `callback` la cual permite realizar la ejecución línea a línea, esto en otras palabras significa que primero se ejecutará por ejemplo la animación y una vez concluida esta se ejecutará una acción específica, para ello se requiere tener en cuenta su sintaxis:

```
$(selector).hide(speed, callback);
```

Chaining: Esta técnica permite que se ejecuten varios métodos o efectos de forma consecutiva en un elemento para lo cual cada uno de los métodos o efectos estarán separados por un punto bajo la siguiente sintaxis:

```
$(selector).método
    .efecto
    .efecto;
```

2.2.3.6 Manejo de HTML con JQuery

JQuery como se conoce permite acceder al DOM de manera mucho más sencilla con el uso de los selectores y las funciones propias que son:

text(): permite asignar o recuperar texto de un elemento del DOM

html(): Obtiene el código HTML de un elemento del DOM

val(): Obtiene el valor de un elemento del DOM.

attr(): Permite obtener los atributos de un elemento del DOM.

```
$(selector).text()  
$(selector).html()  
$(selector).val()  
$(selector).attr("atributo")
```

Para agregar HTML a los elementos también se puede usar funciones como:

append(): Inserta contenido al final del elemento del selector.

prepend(): Inserta contenido al inicio del elemento del selector.

after(): Inserta elementos después del selector.

before(): Insertar elementos antes del selector.

Con el uso de estas funciones se puede hacer varias inserciones de contenidos HTML en los elementos del DOM.

Con el objetivo de eliminar elementos se puede usar dos funciones como los son **remove()** y **empty()**, su diferencia de funcionalidad se da en los hijos de los elementos del selectos. Mientras al usar la función **remove()** se elimina el elemento del selector así como sus hijos, la función **empty()** lo que realiza es la eliminación de todos los elementos hijos mas no del selector.

Para la función **remove()** también se puede especificar qué tipo de elementos se requiere eliminar a través del uso de clases de dichos elementos, para ello se tiene la siguiente sintaxis:

```
$(selector).remove(".clase1, .clase2")
```


2.2.3.7 Manipulación de CSS con JQuery

JQuery adicionalmente al manejo de elementos de DOM a nivel HTML, también permite el manejo de CSS de dichos elementos para ello se usa las funciones:

addClass(): Permite agregar en este caso clases establecidas CSS a elementos del selectos.

removeClass(): Remueve una o varias clases CSS que estén siendo aplicadas al elemento del selectos

toggleClass(): Permite unificar en unas sola función el addClass y removeClass(), ejecutándose de forma alternada entre las dos permitiendo agregar y quitar clases CSS del elemento del selector.

```
$("#selector").evento(function(){
    $("#selector, selector, ...").toggleClass("ClaseCSS");
});
```

Adicionalmente JQuery cuenta con la función css(), misma que permite el obtener o determinar atributos CSS al elemento del selector:

```
Obtener:
    $("#selector").css("atributo");
Establecer:
    $("#selector").css("atributo", "valor");
```

Los elementos HTML del DOM cuentan con dimensiones establecidas a nivel de CSS, las mismas que pueden ser establecidas o recuperadas con funciones específicas de JQuery para esta necesidad de acuerdo a la figura se puede determinar su función.

ESPACIO EN BLANCO INTENCIONAL

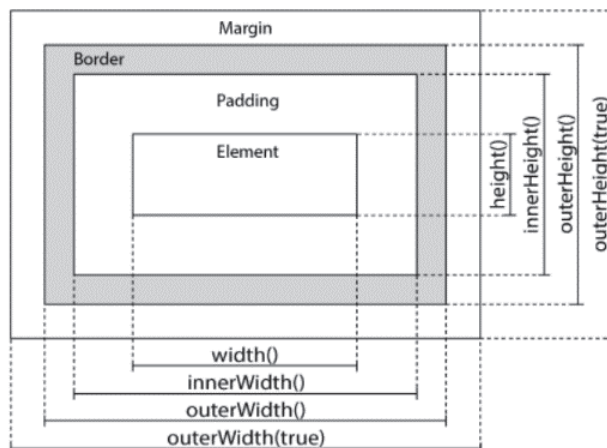


Figura 5. Funciones de JQuery para manejo de tamaños de elementos,

(W3C JQUERY, 2016)

Su sintaxis se ajusta al estándar propio de JQuery de las demás funciones como se muestra a continuación y que se aplica a todas las funciones de dimensión:

```

Obtener:
    $("selector").height();
Establecer:
    $("selector").height(200);
  
```

2.2.3.8 *Desplazamiento entre elementos Transversing*

El desplazamiento entre los elementos del árbol del DOM permite seleccionar elementos gracias al movimiento entre el árbol. Para ello es necesario comprender la estructura y posibles desplazamientos.

La estructura del árbol del DOM está compuesta por ancestros, parientes, hijos y hermanos.

ESPACIO EN BLANCO INTENCIONAL

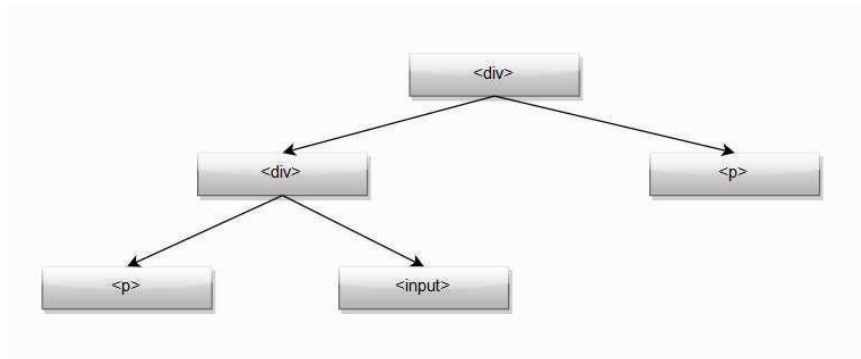


Figura 6. Ejemplo de árbol de elementos HTML

Si se toma la figura como punto de partida, se puede determinar la relación entre los elementos.

El elemento `<input>` como padre al elemento `<div>` y `<p>` que se encuentra a su mismo nivel correspondería a su hermano. Manteniendo la misma lógica se tendría como ancestros a los dos elementos `<div>`.

Con esto y gracias a las funciones de desplazamiento de JQuery se puede desplazar por los elementos del árbol de forma vertical como horizontal a lo cual se adiciona la posibilidad de filtrar las selecciones a través de clases de los elementos y búsquedas con lo cual la variedad de recorridos posibles permite que todos los elementos del árbol puedan ser seleccionados en referencia a sus parientes a nivel vertical u horizontal

Para el desplazamiento vertical entre elementos parientes existen las siguientes funciones.

2.2.3.9 Selección de ancestros

parent(): Permite obtener el elemento padre del selector

parents(): Obtiene todos los ancestros del elemento de selector

parentUntil: Obtiene todos los ancestros entre el elemento del selector y uno de sus ancestros.

```

$("selector").parent()
$("selector").parents()
$("selector").parentUntil("elemento")
  
```

2.2.3.10 Selección de descendientes

children(): permite seleccionar los hijos del selector

find(): Permite buscar un elemento hijo a partir del selector, para lo cual es necesario especificar el elemento a buscar como hijo, para el caso de hacer una búsqueda de todos los elementos hijos se utilizará el asterisco como parámetro de la función.

```
$("#selector").children()  
$("#selector").find("elemento")  
$("#selector").find("*")
```

2.2.3.11 Desplazamiento lateral entre hermanos

sibling(): permite seleccionar a todos los hermanos del selector.

next(): permite seleccionar al siguiente elemento hermano del selector.

nextAll(): obtiene todos los hermanos a partir del selector.

nextUntil(): permite obtener los hermanos entre el selector y un elemento hermanos establecido como parámetro de la función.

prev(): selecciona al hermano anterior al selector.

prevAll(): Obtiene todos los hermanos previos al selector.

prevUntil(): Obtiene los hermanos entre el selector y un elemento hermano establecido como parámetro de la función.

```
$("#selector").sibling()  
$("#selector").next()  
$("#selector").nextAll()  
$("#selector").nextUntil("elemento")
```

ESPACIO EN BLANCO INTENCIONAL

2.2.3.12 Selección basada en posición

JQuery cuenta con funciones de selección de elementos basada en la posición del grupo de elementos del árbol, las cuales son:

first(): retorna el primer elemento de la selección.

last(): retorna el último elemento de la selección.

eq(): obtiene el elemento de acuerdo al índice de posición indicado como parámetro.

filter(): obtienen los elementos filtrados a través de la clase de los mismos que es indicada como parámetro de la función.

not(): obtiene los elementos filtrados que no pertenecen a la clase con la cual se ha filtrado, siendo una función opuesta a find().

```
$("#selector").first()  
$("#selector").last()  
$("#selector").eq(posición)  
$("#selector").filter("clase")  
$("#selector").not("clase")
```

2.2.3.13 Ajax y JQuery

Ajax es un conjunto de diferentes tecnologías que permite la transmisión asíncrona de información entre el servidor y la aplicación web que permite actualizar solo elementos específicos del DOM sin necesidad de recargar toda la página.

Esta característica de Ajax ha permitido que las aplicaciones Web cada vez cuenten con características más cercanas a las aplicaciones de escritorio.

Para el manejo de datos en general JQuery ha desarrollado sus propias funciones que permiten el manejar llamadas Ajax.

load(): Permite cargar los datos obtenidos a través de un url llamado como parámetro en un elemento del selector.

```
$("#selector").load(URL, callback)
```

La función para el método load permite contar con 3 parámetros: Respuesta, estado y error.

```
$("#selector").load(URL, function(respuesta, estado, err){
})
```

get(): Permite realizar una petición tipo HTTP GET al servidor

```
$("#selector").get(URL, function(respuesta, estado){
})
```

post(): Permite realizar un petición tipo HTTP Post al servidor.

```
$("#selector").post(URL,
{
  Var1: "valor",
  Var2: "valor"
},function(respuesta, estado){
  }
)
```

Como se puede observar en la sintaxis de los dos métodos de acceso get y post, su funcionalidad es prácticamente la misma, teniendo la principal diferencia de uso en el paso de variables, en el método post se indica los nombres y valores a enviar a través de una cadena. Para el caso del método get, se deberá incluir los parámetros y valores dentro de la cadena del URL.

Para la función del callback es necesario contar con dos parámetros, el primero que devuelve la respuesta de la URL a la cual se accede mediante el método y el estado de la llamada. Las principales diferencias entre el método HTML GET y POST son:

ESPACIO EN BLANCO INTENCIONAL

Tabla 2. Diferencias entre métodos HTML, GET y POST

<i>Característica</i>	<i>GET</i>	<i>POST</i>
Botón Atrás/Recargar de explorador	No sucede nada	Los datos pueden ser reenviados en la llamada
Historial	Se mantiene los valores en el historial de navegación	No se registra
Cache	Puede mantenerse en cache	No usa cache
Tamaño de datos	Limitado a 2048 caracteres en la URL	No hay restricción
Seguridad	NO es segura ya que muestra los valores en URL y requiere codificación	Más seguro que get al no ser almacenados en cache ni historial, sin embargo puede ser accesible mediante los paquetes en la red
Visibilidad	Datos visibles a través de la URL	Datos no son visibles

2.2.4 HTML

Es un estándar para la creación de páginas web gracias al uso de etiquetas que referencia el contenido para ser interpretados por los explotadores. Sus siglas provienen de HyperText Markup Lenguaje, en español lenguaje de marcas de hipertexto. El lenguaje HTML ha ido evolucionando a través de la historia incluyendo y dejando atrás características que han marcado la evolución tanto de las páginas web como de los exploradores de internet.

2.2.4.1 Evolución

Los orígenes de HTML se remontan a la década de los 40 cuando la Organización Europea de Investigación Nuclear CERN desarrollo un lenguaje que permita referenciar información con el uso de hipervínculos. Es así como en 1965 Ted Nelson

oficializo el nombre hipervínculo siendo la base estructural para que en el año de 1989 surja la World Wide Web.

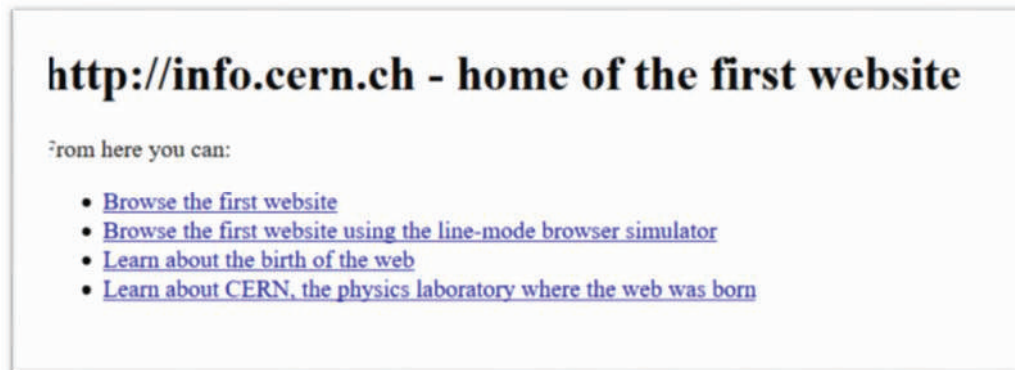


Figura 7. Primer sitio web, (Web Facts, 2016)

Tim Berners-Lee en el año de 1991 fue el primero en generar una descripción de HTML mismo que se basaba en el SGML (ISO 8879:1986), mismo que contaba con una descripción de 22 elementos. El desarrollo de exploradores de principalmente por parte de Microsoft y Netscape hizo necesario que el lenguaje HTML se desarrolle, es dentro de esta lógica que su desarrollo pasa a manos de W3C.

En 1995 IETF publica la norma RFC 1866 que contiene la definición de HTML 2.0 misma que fue escrita por Berners-Lee y Connolly sin embargo IETF dejó su desarrollo a año siguiente a pesar de que en el año 2000 se publicó la RFC 2854.

La versión de HTML 3.2 publicada en enero del 1997 por W3C incorpora muchas de las estructuras y etiquetas que actualmente se usan y fue escrita por Dave Raggett.

FORMATO DE DOCUMENTO HTML 3.2

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
  <HEAD>
    <TITLE>A study of population dynamics</TITLE>
    ... other head elements
  </HEAD>
  <BODY>
    ... document body
  </BODY>
</HTML>
```


En abril de 1998 la versión de HTML 4.0 es publicada estando a cargo de Dave Raggett incluyéndose en esta entrega algo muy importante como lo es el manejo de hojas de estilo CSS. UN año después se publica HTML 4.01 definiendo tres tipos principales de documentos: Strict, Frameset y Transitional. Sus características difieren principalmente por temas de visualización por el uso de frames y la eliminación de etiquetas como tipo de fuente en el Strict.

La aparición de CSS muestra claramente la postura de separar la presentación gráfica de la estructura misma del HTML. Posteriormente W3C avanza en el desarrollo de los estándares XML y XHTML mismos que considero la evolución de la versión de HTML 4.01 sin embargo el cambio hacia XHTML realmente nunca se dio como fue planeado.

En el año 2004 se creó WHATWG (Web Hypertext Application Technology Working Group) compuesto principalmente por Mozilla, Opera, Apple y Google entre las más relevantes y su objetivo fue el desarrollo de abandonado HTML dando como resultado de ello la especificación de HTML 5. A este esfuerzo en el año 2007 W3C hace nuevamente su aparición y no es hasta el año 2011 donde anuncio su compromiso definitivo para el desarrollo de HTML 5.

HTML5

```
<!DOCTYPE html>
```

HTML4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

XHTML 1.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
```

2.2.4.2 Características de HTML 5

HTML 5 es el nuevo formato para el desarrollo de páginas web y sus principales características han sido la adopción de nuevas etiquetas que permitan el manejo de errores disminuyendo el uso de plugins externos además de implementar nuevas etiquetas semánticas para la maquetación de la página a desarrollar.

Las nuevas características que brinda HTML 5 son apreciables únicamente con el soporte de los exploradores formándose una relación conjunta para su aprovechamiento real. El desarrollo de aplicaciones y páginas web para ser visualizadas en móviles y la variedad de dispositivos que pueden acceder a la web impulsó el desarrollo de HTML 5.

Para que el explorador pueda reconocer que se está usando HTML 5 es necesario que la página cuente con el siguiente formato de documento:

```

FORMATO DE DOCUMENTO HTML 5

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
</head>

<body>
Content of the document.....
</body>

</html>

```

Los principales cambios que se encuentran en HTML 5 principalmente están dados por la eliminación de etiquetas, creación de nuevos controles para los formularios, cambios de la estructura semántica, la creación de etiquetas que permiten el manejo de multimedia directamente desde HTML evitando así el uso de plugins externos y la posibilidad de contener y generar gráficos 2D Y 3D.

Las etiquetas eliminadas son las siguientes:

Tabla 3.
Etiquetas reemplazadas en HTML5

<i>Eliminadas</i>	<i>Actuales</i>
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS (continúa)

<center>	CSS
<dir>	
	CSS
<frame>	-
<frameset>	-
<noframes>	-
<strike>	CSS
<tt>	CSS

La estructura semántica de HTML ha variado ya que en la maquetación de la páginas HTML 4.01 era común el uso de etiquetas <div> con el fin de crear áreas para determinados contenidos, en HTML 5 este ha sido uno de los cambios ya que implementa etiquetas específicas para el maquetado de páginas.

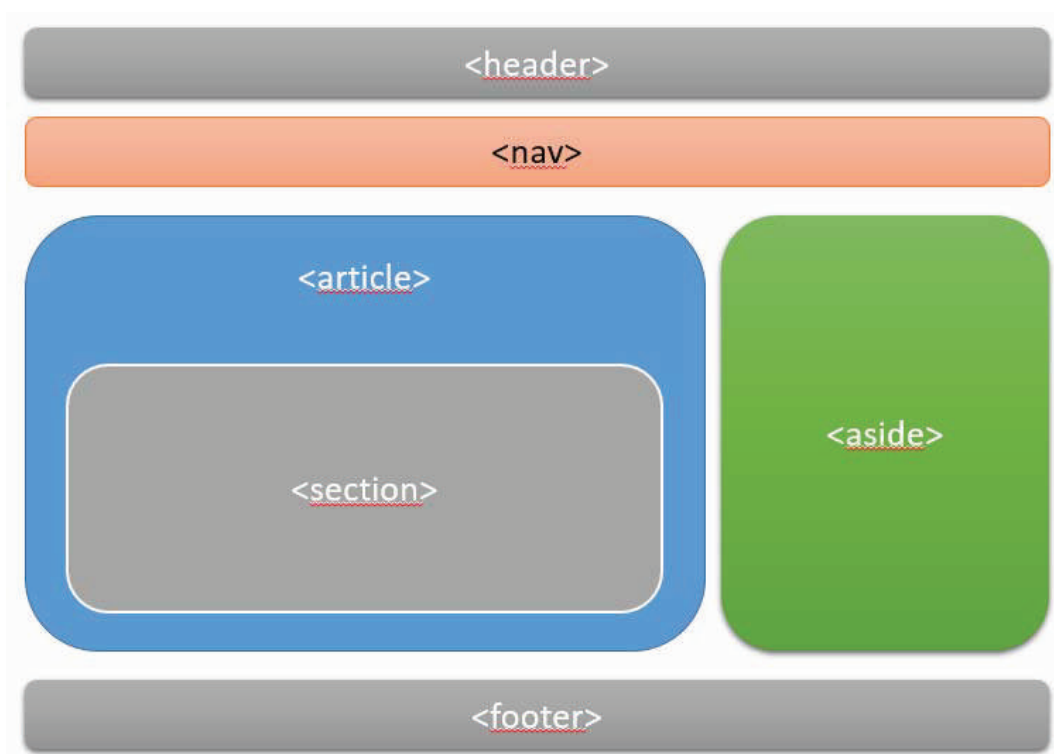


Figura 8. Nuevos elementos de la estructura semántica de HTML5

Los principales elementos semánticos que añade HTML 5 son los siguientes:

Tabla 4.
Principales elementos HTML 5

<i>Etiqueta</i>	<i>Descripción</i>
<article>	Define un artículo en el documento
<aside>	Define el contenido lateral en el documento
<bdi>	Define una parte del texto que podrían ser formateado en una dirección diferente de otro texto
<details>	Define los detalles adicionales que el usuario pueda ver u ocultar
<dialog>	Define un caja de dialogo
<figcaption>	Define un título para un elemento <figure>
<figure>	Define el contenido en sí misma, como ilustraciones , diagramas , fotos, listados de código , etc.
<footer>	Define un pie de página para el documento o una sección
<header>	Define un encabezado para el documento o una sección
<main>	Define el contenido principal de un documento
<mark>	Define marcada o texto resaltado
<menuitem>	Define un elemento de comando / menú que el usuario puede invocar desde un menú emergente
<nav>	Define enlaces de navegación en el documento
<progress>	Define el progreso de una tarea
<rp>	Define qué mostrar en los navegadores que no soportan anotaciones rubí
<rt>	Define una explicación / pronunciación de caracteres (para la tipografía de Asia Oriental)
<ruby>	Define una anotación ruby (para la tipografía de Asia Oriental)
<section>	Define una sección en el documento
<summary>	Define un encabezado visible para un < detalles > elemento
<time>	Define fecha/hora
<wbr>	Define una posible salto de línea

Adicionalmente a estos elementos semánticos, HTML 5 permite la creación de nuevos elementos, acción que es posible en los exploradores modernos tales como Mozilla Firefox o Google Chrome. Para ello se requiere crear el elemento en el documento:

```
<head>
  <title>Nuevo Elemento</title>
  <script>document.createElement("miElemento")</script>
  <style>
    miElemento {
      display: block;
      background-color: #ded;
      font-size: 20px;
      padding: 20px; }
  </style>
</head>
<body>
  <miElemento>Este es mi elemento</miElemento>
</body>
```

Para mantener la compatibilidad con navegadores antiguos así como el caso de Internet Explorer es necesario el uso de un Script que permite su compatibilidad para ello se debe hacer la siguiente llamada en el Head.

```
<!--[if lt IE 9]>
  <script
  src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></scri
  pt>
<![endif]-->
```

Para explorados anteriores es necesario indicar que los nuevos elementos semánticos deben mostrarse en bloque para ello a nivel de CSS es recomendable indicar lo siguiente:

```
header, section, footer, aside, nav, main, article, figure {
  display: block;
}
```

2.2.4.3 Nuevos Elementos de Formulario

Lo siguientes elementos pertenecerán a un formulario del documento y son los siguientes:

Tabla 5.
Nuevos elementos de formularios

<i>Etiqueta</i>	<i>Descripción</i>
<datalist>	Define una lista de opciones
<keygen>	Define un campo generador de par de claves
<output>	Define el resultado de un cálculo

Adicionalmente se han adicionado varias tipos para la etiqueta <input>, así como atributos que se listan a continuación:

Tabla 6.
Nuevas etiqueta input

<i>Inputs</i>	<i>Descripción</i>
color	Selector de Color
date	Ingreso de formato fecha
datetime	Ingreso de formato fecha y hora (No soportado aun)
datetime-local	Ingreso de formato fecha y hora local
email	Ingreso de email
month	Seleccionar un mes
number	Permite seleccionar un número a partir de picks de incremento y disminución
range	Permite seleccionar un rango de forma visual
search	Campo de búsqueda
tel	Permite ingresar números telefónicos
time	Permite seleccionar un hora
url	Permite el ingreso de URLs
week	Selecciona una semana

Tabla 7.
Atributos nuevos para etiquetas input

<i>Atributos</i>	<i>Descripción</i>
autocomplete	Indica si es susceptible a autocompletar
autofocus	Permite que una vez que se cargue la página este input tome el foco automáticamente
form	Permite indicar a que formulario pertenece
formaction	Permite indicar la URL que se va a procesar anulando la del atributo action del formulario.
formenctype	Permite indicar la codificación del formulario anulando la del form
formmethod	Permite indicar el método HTTP anulando el establecido en el formulario
formnovalidate	Indica que el input no debe ser validado
formtarget	Indica el destino para mostrar la respuesta del submit
height y width	Permite indicar el tamaño en ancho y altura del input
list	Relaciona a un datalist con el input
min y max	Permite determinar el mínimo y máximo para un número, fecha, mes, semana
multiple	Indica que se puede ingresar más de un elemento
pattern (regexp)	Especifica un expresión regular para el input
placeholder	Indica una descripción breve o indicación que se muestra en el input
required	Indica que el campo es requerido
step	Indica el intervalo de valores requeridos para el input

HTML 5 permite el manejo de gráficos a través de <canvas> y <svg>, para el caso de canvas se puede trazar dibujos usando java script.

ESPACIO EN BLANCO INTENCIONAL

CANVAS

```

<!DOCTYPE html>
<html>
<body>
<canvas id="miCanvas" width="200" height="200"
style="border:1px solid #eee;">
</canvas>
<script>
    var c = document.getElementById("miCanvas");
    var ctx = c.getContext("2d");
    ctx.fillStyle = "#ddd";
    ctx.fillRect(0,0,100,105);
</script>
</body>
</html>

```

SVG

```

<!DOCTYPE html>
<html>
<body>
<svg width="300" height="300">
  <circle cx="100" cy="100" r="70"
    stroke="red" stroke-width="2" fill="blue" />
</svg>
</body>
</html>

```

Mediante el conjunto de instrucciones se puede definir de acuerdo a la necesidad los gráficos que sean requeridos, teniendo en cuenta la compatibilidad con los exploradores así como la funcionalidad de los gráficos que se desee generar.

2.2.4.4 Multimedia

Para el manejo multimedia era necesario el uso de plugins externos como el muy conocido Flash Player, sin embargo en HTML5 se cuenta ya con soporte multimedia nativo para contenidos multimedia. Para el uso de multimedia en HTML5 se reconocen los siguientes formatos de audio y video:

Audio:

- MP3
- WAV
- OGG

Video:

- MP4
- WebM
- OGG

```
<!DOCTYPE html>
<html>
<body>

<video width="320" height="240" controls>
  <source src="mipelicula.mp4" type="video/mp4">
  <source src=" mipelicula.ogg" type="video/ogg">
</video>
</body>
</html>
```

Para el uso de otro tipo de multimedia se la considera como un objeto embebido para ello se usa la etiqueta `<object>`

```
<object width="400" height="150" data="miflash.swf"></object>
```

También es posible hacerlo con la etiqueta `<embed>` misma que incrusta un elemento en el HTML.

```
<embed width="400" height="150" src="miflash.swf">
```

2.2.5 CSS

Las hojas de estilo en cascada (Cascading Style Sheets) es un lenguaje que permite manejar la apariencia ubicación y comportamientos de páginas web. Su uso ha permitido mejorar considerablemente la legibilidad de la página web separando la funcionalidad de las mismas de la apariencia, simplificando el código y aumentando la posibilidad de manejo gráfico.

2.2.5.1 Historia CSS

CSS hace su aparición conjuntamente a HTML 4 siendo una necesidad para solucionar un gran problema que tenía HTML hasta ese momento. El principal problema era que la parte funcional así como la presentación estaban en un solo archivo HTML y que al estar juntos producía código extenso y que era más difícil de comprender y mantener. Al implementar CSS se separa las características gráficas y de comportamiento de los elementos HTML permitiendo tener código limpio y mucho más fácil de manejar y entender, Adicionalmente permitió facilitar el desarrollo de páginas al abrir la posibilidad de crear clases y aplicar estilos personalizados así como generalizados a etiquetas.

CSS hace su aparición oficialmente a través de W3C en el año de 1996 siendo sus principales funcionalidades:

- Manejo de fuentes
- Colores de elementos
- Atributos de texto
- Alineación de elementos
- Atributos de margen, espaciado y relleno
- Propiedades de identificación

Luego cada una de las evoluciones de CSS han sido un conjunto de mejoras, en el año de 1998 es publicada la versión 2 de CSS siendo una aplicación de CSS 1 y como características principales presenta:

- Posicionamiento relativo, absoluto y fijo.
- Soporte de CSS auditivos
- Características de texto decorativas

El mismo año se hace una nueva revisión de CSS 2 misma que está orientada a la corrección de errores y eliminación de funcionalidades que no son necesarias.

Prácticamente a la par que se hace la liberación de CSS 2 se inicia el trabajo de los que se convertiría en CSS 3, mismo que se caracteriza principalmente por su popularización que hizo que cada uno de ellos se fuera desarrollando y profundizando permitiendo contra con un soporte robusto de cada una de las características de CSS.

2.2.5.2 Características CSS

La determinación de las características individuales o grupales a los elemento HTML CSS permite a través de diferentes selectores acceder a uno o varios grupos de elementos, es por ello que se puede determinar 3 principales selectores:

- Elemento
- Clase
- Identificador

2.2.5.3 Selector por elemento

El selector por elemento aplicara a un elemento HTML las características establecidas dentro de la definición CSS. Esto significa que todos los elementos de este tipo tendrán características comunes definidas en el CSS.

```
p {  
  text-align: center;  
  color: blue;  
}
```

Es posible también agrupar los elementos a los cuales se les va a aplicar las propiedades CSS, para ello cada elemento HTML debe ser descrito en el selector separado por comas.

```
p, h1, h3 {  
    text-align: center;  
    color: blue;  
}
```

2.2.5.4 *Selector por Identificador*

Los elementos HTML tienen a través de la propiedad ID un identificador único que los define, es gracias a este identificador que es posible aplicar propiedades CSS únicas para un elemento.

```
#identificador1 {  
    text-align: center;  
    color: blue;  
}
```

2.2.5.5 *Selector por Clases*

Permite especificar las características para una clase establecida para los elementos HTML, lo que permite que varios elementos cuenten con las características establecidas en el CSS.

```
.miclase {  
    text-align: center;  
    color: blue;  
}
```

Adicionalmente se puede especificar un elemento específico que pertenezca a determinada clase.

```
p.miclase {  
    text-align: center;  
    color: blue;  
}
```

Las hojas de estilos CSS tiene tres vías principales para ser adjuntadas a páginas HTML, así como un orden de precedencia entre cada uno de los métodos de inserción mismos que son:

- Archivo CSS independiente
- Dentro de <head> del HTML
- En línea, dentro del elemento HTML

La precedencia se da de la siguiente manera:

1. Definido por el Explorador.
2. CSS de archivo o en el encabezado HTML
3. En línea

2.2.5.6 Declaración de CSS

En un archivo independiente CSS (Externo)

```
<head>
  <link rel="stylesheet" type="text/css" href="miestilo.css">
</head>
```

En el <head> de HTML (Interno)

```
<head>
<style>
p {
  color: red;
  margin-left: 20px;
}
</style>
</head>
```

En línea, directamente en el elemento HTML (Inline)

```
<p style="color:blue; ">Mi párrafo es azul!!!</p>
```

Ya en lo concerniente específicamente de CSS3 sus características principales o particularmente sus módulos son los siguientes:

- Selectores
- Modelos de Cajas
- Bordos y fondos
- Valores de imágenes
- Efectos de texto
- Transformaciones 2D y 3D
- Animaciones
- Layouts

2.2.6 API Google Maps

Durante la evolución de la tecnología de la información se puede identificar el reemplazo continuo de herramientas tradicionales o manuales en diferentes aspectos de la vida cotidiana de la sociedad. No estuvo alejado de esta realidad el manejo a nivel de software de mapas geográficos con diferente información de acuerdo a las necesidades específicas.

Esta evolución trajo consigo la necesidad de globalizar los servicios de mapas dejando a un lado sistemas pagados y específicos como los navegadores GPS, sistemas cartográficos que si bien cumplían con su objetivo no eran de acceso abierto a la sociedad.

Esta necesidad es la fuente para el desarrollo de los que hoy se conoce como Google Maps, desarrollo independiente que posteriormente es adquirido por Google y que permitió, gracias a las propias características de la empresa, un desarrollo continuo y que al momento brinda un sin número de servicios basados en localización GPS, imágenes satelitales, navegación, imágenes y un sin número de funcionalidades que lo han convertido en una herramienta de libre acceso de mayor difusión y que brinda una amplia gama de beneficios.

Uno de los pilares para su amplio uso ha sido la liberación de su SDK y herramientas para desarrolladores que han permitido incluir sus funcionalidades en aplicaciones y

en la actualidad ser parte fundamental de la tecnología móvil que gracias a su interoperabilidad.

Por ello Google Maps se presenta como la principal librería o fuente de mapas digitalizados con soporte en imágenes satelitales.

2.2.6.1 Historia

Sus orígenes se remontan al año 2014 donde la empresa Where 2 Technologies inicia su desarrollo ya bajo una plataforma web misma que es adquirida por Google lo que da como producto el nacimiento de Google Maps.

Inicialmente su versión beta incluyó ubicaciones digitales de las principales ciudades de América del Norte para posteriormente ir incluyendo información de otros países como los europeos quienes tuvieron acceso al servicio en el año 2006.

Conforme fue expandiendo el proceso de digitalización de los mapas se incluyeron ubicaciones, información de tráfico, siendo estos servicios así como la ubicación de coordenadas GPS los principales atractivos de la herramienta.

Ya en el año 2007 se incorpora la posibilidad de incluir imágenes de sitios específicos y que fue denominado como StreetView que inicialmente estuvo disponible únicamente en 5 ciudades sin embargo dichas imágenes han ido en un crecimiento vertiginoso hasta que al momento prácticamente a cubierto todos los rincones del mundo con las opciones antes indicadas.

Siendo Maps un servicio orientado a la web si bien brindaba un modo de acceso mundial ciertas funcionalidades requerían de una plataforma más específica que la web. Es así que se desarrolla Google Earth que es un software específico que cuenta con información de Google Maps pero que incluye varias herramientas y una navegación sobre todo el globo y con servicios como vista interior de edificios, estructuras en tres dimensiones, creación de mapas, rutas, áreas y mediciones.

2.2.6.2 Evolución

Las primeras versiones de Google Maps contaban con imágenes digitales de las principales ciudades, mismas que fueron incrementándose conforme Google Maps hacia incursión en nuevos países y regiones.

La tecnología de imágenes por satélite permite relacionar a nivel de capas la imagen vectorial de los mapas con imágenes satelitales reales.

Conforme el crecimiento de información fue dándose se incluyeron datos específicos de lugares emblemáticos, imágenes y StreetView.

Se desarrolló el SDK y herramientas de desarrollo que permite la integración de los servicios en aplicaciones desarrolladas.

El incremento de funcionalidades permitió la expansión a nuevas tecnologías como lo es la tecnología móvil que gracias a su SDK permitió su integración a servicios como redes sociales, sistemas geográficos permitiendo la interacción de dichas aplicaciones con servicios específicos como compartir ubicación, cálculo de rutas, tráfico entre otras.

La evolución ha permitido el ampliar sus servicios como:

- Google Local: Búsquedas locales
- Ride Finder: Ubicación de servicios de transporte
- Business View: Fotografías propietarias publicadas
- Google Moon: Geografía Lunar
- Google Mars: Geografía de Marte con diferentes tipos de vistas
- Send-to-car: Integración de GPS de vehículos
- Integración con Redes Sociales

2.2.6.3 Características de API Google Maps

El API de Google permite incluir los servicios de Google Maps aplicaciones desarrolladas sea bajo tecnología web o específicas para móviles bajo Android o iOS principalmente.

Para acceder a los servicios inicialmente de debe adquirir un llave de uso (Key). Esta llave permite la autenticación del usuario que va a utilizar los servicios, para ello es necesario seguir el siguiente procedimiento:

1. Crear Proyecto o seleccionar uno existente
2. Activar el API

3. Obtener las Credenciales y la llave

Una vez con la llave se acceder a los servicios de Google Maps de varias maneras siendo la más común a través de JavaScript

```
<script src = https://maps.googleapis.com/maps/api/js?key=API\_KEY type="text/javascript"></script>
```

Para el uso del API se debe incluir en el HTML la llamada al API conjuntamente con la llave. Para inicializar el mapa es necesario indicar el contener dentro del DOM del documento para lo cual se hace uso de la etiqueta HTML <div> mismas que como id se indica el nombre del objeto creado en Javascript y que permitirá la inyección del mapa a mostrar.

```
<div id="map"></div>

<script type="text/javascript">
var map;
function initMap() {
  map = new google.maps.Map(document.getElementById('map'),
  {
    center: {lat: -34.397, lng: 150.644},
    zoom: 8
  });
}
</script>

<script async defer
  src="https://maps.googleapis.com/maps/api/js?key=API_K
  EY&callback=initMap">
</script>
```

De esta manera se ha integrado el servicio de Google Maps a nuestra aplicación, siendo su funcionalidad provista a través del API.

2.2.7 Código de Barras

Una entrada de datos muy utilizada por sus grandes ventajas, se ha convertido en un estándar a nivel mundial que permite la rápida identificación de objetos. Los códigos

de barras tienen como principal característica la representación de un identificador único de forma gráfica y estandarizada que con el uso de sistemas de información permite asociar la imagen con documentos, objetos, productos entre otros.



Figura 9. Código de barras IAN

La patente de creación de código de barras es registrada en el año de 1952 estando principalmente orientado a la identificación de vagones de tren y su invención está determinada para Joseph Woodland, Jordin Johanson y Bernard Silver.

En la de cada de los 80 es cuando el sistema de identificación cobra mayor valor al difundirse y utilizarse en muchos productos.

Existen varios tipos de códigos de barras, mismos que han sido desarrollados de acuerdo a las necesidades sin embargo los más usados son los compuestos por barras verticales conocidos como EAN.

Las principales ventajas de su uso son:

- Rapidez en el manejo y obtención de información relacionada a los productos etiquetados.
- Porcentaje de error reducido al momento de identificación de datos.
- Integración con sistemas de información.
- Su uso es prácticamente un estándar en la industria

2.2.8 Servidor Web Apache

Los orígenes del servidor web apache están dados por el servidor NCSA httpd mismo que es desarrollado por National Center for Supercomputing Applications por Robert McCool, y que inicialmente sirvió de fuente para que Apache sea únicamente un conjunto de parches a este servidor web.

Al transcurrir el tiempo, apache empieza la reescritura del código y se establece Apache Software Foundation quien se encarga de mantener el desarrollo de apache siendo su cumbre el año 2005 donde se estima que alcanza aproximadamente el 70% de los sitios web a nivel mundial sentando el precedente de ser el servidor web más usado a nivel mundial.

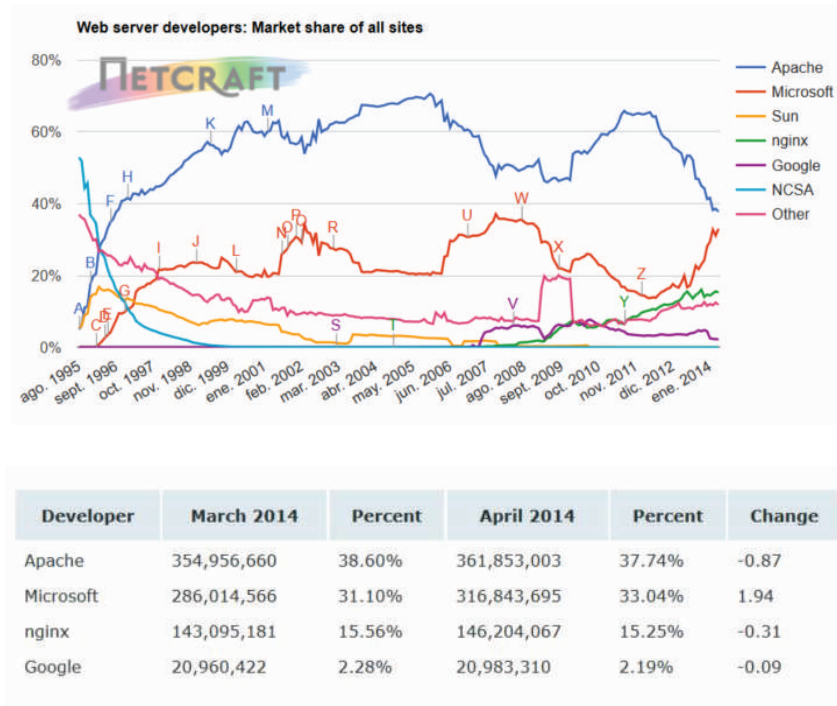


Figura 10. Resultados de encuesta de servidores web 2014, (Netcraft, 2016)

Las principales características de servidor web apache están definidas por su flexibilidad sin embargo también se puede mencionar:

- Modularidad
- Multiplataforma
- Extensible
- Altamente popular
- Requisitos de software bajos
- Rápido

Su modularidad permite mejorar y extender su funcionalidad siendo algunos de sus principales módulos los siguientes por ejemplo:

- `mod_ssl`: Comunicaciones basadas en TLS para seguridad.
- `mod_rewrite`: Rescritura de direcciones.
- `mod_cband`: Limitador de ancho de banda y tráfico.
- `mod_php`: Integración con páginas dinámicas basadas en lenguaje PHP.
- `mod_perl`: Páginas dinámicas con perl.

La principal implementación del servidor apache se da a través de sistemas Linux, siendo su configuración principal realizada a través de un único archivo de configuración en el cual se define sus principales propiedades y el mismo está ubicado en el directorio `/etc/httpd/conf/` el srchivo a modificar será `httpd.conf` siendo las principales configuraciones:

- `ListenPort`
- `ServerName`
- `Extensiones`
- `Virtual Hosts`

Para el caso de los Virtual Host generalmente apache hace referencia a la ubicación `/etc/httpd/conf.d/` donde se deberán ubicar los archivos relacionados a servidores virtuales.

Una vez configurado podrá ser iniciado como un servicio del sistema operativo a través de la línea de comando:

```
#service httpd start
```

2.3 Metodología SCRUM

2.3.1 *Introducción*

Una metodología lo que busca es determinar un conjunto de pasos que garanticen el cumplimiento de los objetivos planteados siempre buscando el mejor desarrollo del proyecto. Dentro de las metodologías existentes en el desarrollo de software se han

desarrollado varias que han ido adaptándose a las características de los proyectos así como los lenguajes de programación y realidad de los proyectos.

Generalmente los proyectos que requieren de un largo aliento para el desarrollo se han orientado por metodologías tradicionales donde la arquitectura de software es fundamental y sus requerimientos están plenamente definidos.

Bajo esta consideración y conforme el desarrollo es llevado adelante en las metodologías tradicionales un cambio en los requisitos se vuelve en uno de los principales factores de fracaso de los proyectos, ante esto se definen nuevas opciones de metodologías ágiles de desarrollo como Extreme Programming y SCRUM.

Las metodologías ágiles toman como premisa que existirán cambios en los proyectos, es por ello que definen periodos de desarrollo rápidos conocidos como Sprints donde prácticamente se da todo el proceso de desarrollo de una porción del proyecto garantizando la colaboración con el cliente, cambios y calidad. Es así que nace o inicia la metodología SCRUM que la permite definir como un conjunto de buenas prácticas que permite un desarrollo colaborativo dentro del proyecto y cuyo objetivo es obtener el mejor resultado de un proyecto basado en las entregas parciales siendo la productividad flexibilidad e innovación fundamentales para su éxito. El proceso colaborativo está basado en el desarrollo de una lista de requerimientos priorizados que son la base para el sprint en el cual se cumplen todas la etapas propias del desarrollo de aplicaciones de inicio a fin esto es análisis hasta pruebas permitiendo así que al final de cada sprint el cliente valide una fase del proyecto ya funcional.

2.3.2 *Proceso de SCRUM*

El proceso de SCRUM está definido por varios Sprints, mismos que se ejecutan en un par de semanas a un mes, siendo esta una de sus principales característica, dentro del cual se debe esperar un resultado completo de acuerdo a lo definido. Para ello es necesario contar con una lista de requerimiento priorizada, que es el plan del proyecto, la misma debe ser balanceada entre aporte al proyecto así como al costo.

ESPACIO EN BLANCO INTENCIONAL

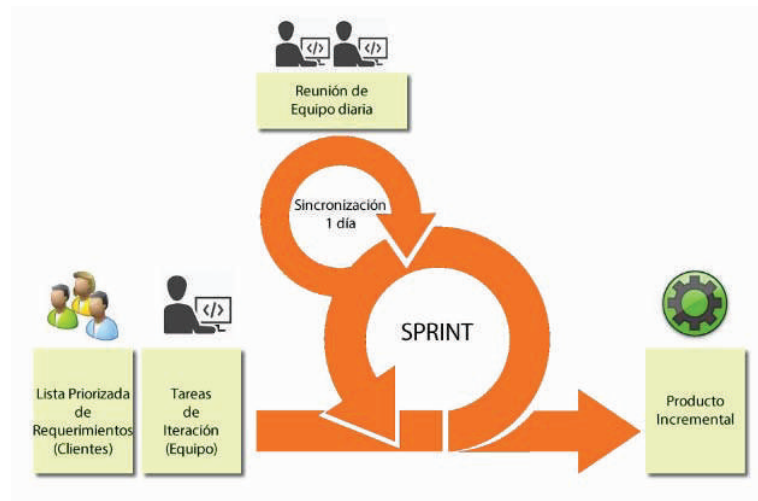


Figura 11. Planificación de la iteración o Sprint

La fase de planificación determina la iteración dentro de la metodología SCRUM, inicialmente es requerido que los clientes o usuarios determinen una lista de Requisitos Priorizados, los mismos que deberán contener el listado de selección de requisitos que son considerados necesarios para la iteración misma que será presentada al equipo del proyecto, se considera que esta reunión deberá estar planificada para no extenderse más allá de cuatro horas. Durante la reunión el equipo de desarrollo del proyecto deberá aclarar cualquier duda en relación a la lista presentada por el cliente y a su vez iniciar el proceso de selección de los requerimientos presentados por su prioridad teniendo en cuenta a su vez aspectos propios del proceso de desarrollo como dependencias, requerimientos y demás que asocien a determinado requerimiento con otros.

Una vez obtenido ya el listado de requerimientos priorizados los cuales deberán estar balanceados respecto al costo y funcionalidad requerida se deberá realizar una segunda reunión esta vez conformada por el equipo de desarrollo en la cual se estimará el esfuerzo requerido así como la distribución entre el equipo.

Los productos esperados de estas dos actividades serán:

- Lista de requisitos priorizados
- Lista de tareas de la iteración

2.3.3 Ejecución de la Iteración

Para el proceso de ejecución se considerara la lista de tareas de la iteración a fin de realizar el seguimiento de cumplimiento de las mismas, para ello todos los días se deberá realizar una reunión corta, de no más de quince minutos, en la cual se establecerá el avance, dependencias y problemas encontrados dentro del desarrollo de la iteración, para ello el Scrum Master del proyecto deberá ejercer su responsabilidad de facilitador y lograr solucionar cualquier dificultad y mantener la productividad del equipo.

Durante la reunión cada uno de los miembros del equipo deberá indicar los avances desde la última reunión de sincronización, sus próximas actividades y cualquier posible impedimento o problema a enfrentar.

Adicionalmente se podrá cambiar o redefinir los requerimientos seleccionados con el apoyo del cliente a fin de maximizar la productividad de la iteración y el proyecto en general.

2.3.4 Inspección y adaptación

Una vez llegado al fin de la iteración o sprint, se deberá realizar la reunión de demostración en la cual en un tiempo no superior a cuatro horas se realizará la presentación de los requisitos completados así como de los cambios que han sido necesarios realizar dentro de la iteración, de esta presentación el cliente deberá re planificar lo necesario a fin de garantizar el avance adecuado del proyecto con miras al siguiente sprint.

Posteriormente se realizará una reunión del equipo de desarrollo del proyecto denominado de Retrospectiva donde se analizará la forma de trabajo y los posibles problemas o errores encontrados dentro de la iteración buscando el mejoramiento continuo del proceso.

2.3.5 Actividades

SCRUM define un conjunto de actividades que se realizan en cada una de las iteraciones y que son:

Planificación de iteración o Sprint Planning

Básicamente compuesta por dos reuniones de trabajo, la primera con el cliente (Product Owner) donde el establecerá la lista de requerimientos priorizados al equipo siendo importante su interacción a fin de tener clara la lista priorizada.

Se realizará la reunión del equipo del proyecto en la cual se hará una selección de la lista de requerimientos priorizados así como la estimación de esfuerzo y responsabilidades que cada uno asumirá para el cumplimiento de lo planificado, de esta manera se definirá la lista de Tareas de la iteración basada en la Lista de requerimientos priorizados. De esta manera se han definido dos de las herramientas con las que trabaja la metodología SCRUM.

2.3.6 Ejecución de Iteración o Sprint

En la ejecución se deberá dar prioridad a los requisitos más funcionales para el cliente como medida de completar con éxito la iteración. Durante la ejecución de la iteración se deberá realizar reuniones de sincronización las mismas que permitirán a los miembros del equipo conocer los avances, problemas y los pasos a seguir entre todos los miembros. El facilitador o SCRUM Master será el encargado durante esta fase de mantener la productividad del equipo así como responsable de eliminar cualquier obstáculo que el equipo encuentre en la ejecución del proyecto.

2.3.7 Reunión de Sincronización

Las reuniones de sincronización deberán ser cortas y diarias permitiendo actualizar continuamente las tareas de la iteración así como el grafico de trabajo pendiente (Burndown chart).

2.3.8 Demostración de Requisitos Completados

Permite la presentación del avance obtenido durante la iteración para lo cual el equipo hará la demostración de los requisitos completados de forma incremental al cliente. De esta manera se permite conocer claramente el avance y como se han plasmado ya los requisitos del cliente. De esta manera el cliente podrá re planificar el proyecto.

2.3.9 Retrospectiva o Sprint Retrospective

La Reunión de retrospectiva busca identificar los factores que han aportado al éxito o fracaso de la iteración reconociendo no solo los puntos de vista del equipo sino también del cliente, esta evaluación permitirá identificar que actividades están funcionando adecuadamente e identificar cuáles son necesarias cambiarlas o mejorarlas a fin de que se cumpla con el proyecto.

2.3.10 Redefinición de lista de requerimiento y cambios.

La lista de requisitos priorizados durante cada una de las iteraciones podrá ser modificada ajustándose a las necesidades del proyecto sea por ajustes internos o exógenos por cambios en el entorno del proyecto. Los ajustes se deberán realizar durante las reuniones de Demostración. Los ajustes jamás se harán con los requerimientos que se están desarrollando en la iteración.

2.3.11 Roles

Dentro de la metodología SCRUM se establece tres roles principales los mismos que cumplen con funciones específicas dentro del proyecto y son:

2.3.11.1 Cliente (Product Owner)

Es el representante de todas las partes interesadas del proyecto y es el encargado de definir los objetivos del proyecto por ende el único interlocutor con el equipo del trabajo. Su colaboración en el proyecto es fundamental para planificar, revisar y definir detalles del proyecto.

2.3.11.2 Facilitador (SCRUM Master)

Es el encargado de guiar el proyecto para que mantenga su productividad siendo el intermediario entre el equipo de desarrollo y los clientes. Sus funciones son las facilitar el trabajo del equipo permitiendo su aislamiento de problemas que no estén relacionados al desarrollo así como mantener un ambiente adecuado para el trabajo colaborativo. Dentro del mismo contexto así como facilitador deberá ser el responsable de garantizar el cumplimiento de los requisitos priorizados.

2.3.11.3 Equipo (Team)

Son los responsables del cumplimiento de lo planificado tanto como su calidad, su trabajo es esencial y está compuesto por un equipo multidisciplinario que tiene la

obligación de priorizar, seleccionar y hacerse responsable de los requerimientos que se están desarrollando durante la iteración. Adicionalmente serán los encargados de realizar la demostración de los avances a los clientes una vez finalizada la iteración.

2.3.12 Herramientas

Las herramientas denominadas por SCRUM son:

2.3.12.1 Lista de requisitos priorizados o Product Backlog

Es la representación de la visión del producto por parte del cliente quien es responsable de crear y gestionar la lista. La lista esta expresada como historias del usuario y en base a ella se establecerá las iteraciones. Su definición es recomendable se la realice con la ayuda del SCRUM master a fin de valorar adecuadamente su costo, esfuerzo y entrega.

2.3.12.2 Lista de Tarea de la iteración o Sprint Backlog

Es la lista de los requerimientos seleccionados por parte del equipo de desarrollo y que se compromete a completar dentro del periodo de iteración para su demostración al cliente. La lista permite evaluar el desarrollo del proyecto así como determinar áreas de dificultad por ello las tareas deberán ser detalladas para ser cumplidas entre 4 a 16 horas a fin de que sea rápidamente identificado un punto de dificultad.

2.3.12.3 Gráfico de trabajo pendiente (Burndown Chart)

Es un gráfico que representa el avance de cumplimiento de las tareas a manera de pendiente basado en horas/días pendientes de trabajo.

CAPÍTULO 3

ANÁLISIS Y DISEÑO

3.1 Ejecución del Proyecto

3.1.1 *Planificación del Proyecto*

El desarrollo del producto está definido para realizarlo bajo la metodología SCRUM, para lo cual de determinar la lista de requerimientos priorizados conjuntamente con el cliente, adicionalmente a ello se deberá establecer el proceso adecuado para que la automatización del proceso alcancen los objetivos planteados.

Para ello se establecerá inicialmente el acercamiento a fin de establecer el equipo de trabajo dentro del proyecto preparando los requerimientos necesarios para ajustarse a la metodología SCRUM con el interés de iniciar el primer sprint.

3.1.2 *Alcance del Sistema*

El Sistema de gestión de proceso de producción está planificado para que sea una herramienta que permita mantener el proceso de producción estandarizado dentro de la empresa y que brinde información de forma oportuna del estado, avance y documentación de los proyectos.

Para ello se ha establecido que debe contar con los siguientes módulos:

Módulo de seguridades

Administración de usuarios y roles

Permitirá la creación y edición de usuarios que acceden al sistema así como los roles y permisos. Adicionalmente mantendrá la administración de los registros de acciones y accesos del sistema.

Módulo de administración del sistema

Administración de catálogos

Permitirá la creación, edición de todos los catálogos dinámicos y parametrización del sistema.

Módulo de utilitarios

Envío de notificaciones y correos electrónicos

Herramienta para el envío de correos electrónicos de forma automática basado es plantillas personalizables para cada una de la acciones que sea requerido.

Administración de localizaciones GPS

Manejo de localizaciones geográficas que permita obtener resultados sin realizar implementaciones adicionales.

Generador de Reportes

Manejo de plantillas para reportes así como un generador de PDF que permita que los reportes sean portables.

Módulo Atención al Cliente

Gestión de Clientes

Permitirá el registro de clientes.

Creación de proyectos

Permitirá la creación de nuevos proyectos con su caracterización necesaria.

Generación de contratos

Editor de documentos contractuales basado en plantillas que permita su autogeneración

Proformas y registro de Pagos

Creación de proformas para la aprobación de clientes así como el registro de pagos de contratos vigentes.

Asignación de proyectos

Permite la asignación de un líder de proyecto que será el encargado del diseño y seguimiento del proyecto.

Reportes

Generador de reportes de los proyectos.

Módulo de diseño

Gestión de proyectos.

Acceso los proyectos de responsabilidad de cada uno de los usuarios del sistema para el registro de avance y carga de archivos relacionados al proyecto.

Generación de cotizaciones.

Permite la generación de documentos de cotización en base a plantillas personalizables.

Carga de documentos digitales

Permitirá la carga de distintos archivos digitales con versionamiento a nivel de la aplicación.

Módulo de producción

Gestión de proyectos.

Acceso los proyectos de responsabilidad de cada uno de los usuarios del sistema para el registro de avance y carga de archivos relacionados al proyecto.

Asignación y generación de órdenes de producción

Permitirá el designar los responsables del proceso de producción en fábrica para el proyecto, usuario que será responsable de la actualización del estado de avance.

Avance de producción

Permitirá indicar el avance del proceso de producción por parte del usuario responsable.

Generación orden de instalación y entrega

Generación de documentos para la instalación y entrega basado en plantillas personalizables.

3.1.3 Equipo de Trabajo

El equipo de trabajo para el desarrollo del sistema se ha definido bajo los lineamientos de la metodología y está conformado de la siguiente manera:

Tabla 8.
Equipo de trabajo bajo metodología SCRUM

<i>Rol</i>	<i>Persona</i>	<i>Área</i>
Producto Owner	Xavier Sandoval	Gerencia
SCRUM Master	Alberto Coba	Desarrollo
Team	Alberto Coba	Desarrollo
	Xavier Sandoval	Gerencia

3.1.4 Lista de requisitos priorizados

La lista de requerimientos priorizada contiene la definición de los requerimientos del sistema, mismos que de acuerdo a la metodología se desarrolló en conjunto entre los miembros del equipo de trabajo y que está acordado de acuerdo al documento DOC-001-METALEC en cuyo contenido se define la siguiente lista de requerimiento:

Tabla 9.
Lista de requerimientos priorizados bajo metodología SCRUM

<i>Lista de Requerimientos Priorizados</i>			
<i>Nro.</i>	<i>Prioridad</i>	<i>Requerimiento</i>	<i>Modulo</i>
1	ALTA	Gestión de usuarios	Administrador Aplicación(continúa)

2	ALTA	Creación de clientes bajo persona natural o jurídica	Administrativo
3	ALTA	Uso de una presentación visual minimalista sin uso de colores brillantes o que no tengan el contraste adecuado	General
4	ALTA	El sistema deberá contar con la seguridad para que cada usuario cuente con un acceso personalizado de acuerdo a sus funciones	Seguridades
5	ALTA	La autenticación se deberá hacer mediante un nombre de usuario y contraseña misma que deberán estar encriptada	Seguridades
6	ALTA	El acceso a cada uno de los roles de usuario deberá ser transparente para el usuario y su modificación deberá realizarlo el administrador de la aplicación.	Seguridades
7	ALTA	La aplicación deberá estar compuesta por tres secciones: Un encabezado con el logo de la empresa, datos de identificación del usuario	Seguridades
8	ALTA	Todo ingreso al sistema deberá contar con un registro o campos de auditoria	Seguridades
9	BAJA	Si es aceptada la proforma final el/la encargada/o administrativa generará el contrato que estará previamente cargado gracias a formatos generados automáticamente	Administrativo
10	BAJA	Los documentos deberán incluir logo institucional y una presentación estandarizada.	General (continúa)

11	BAJA	Registrar la ubicación del proyecto mediante el uso de mapas y coordenadas.	General
12	BAJA	Envío de correos de notificación en cada etapa del proceso	General
13	BAJA	Deberá funcionar en el ambiente WEB	General
14	BAJA	Cada responsable del área de producción será encargado de actualizar el estado del proyecto relacionado al avance en el proceso de producción así como problemas encontrados.	Producción
15	BAJA	Concluido el proceso de producción e instalación de los módulos se deberá emitir el acta de entrega recepción del contrato	Producción
16	MEDIA	Módulo de administración de catálogos que requieran ser gestionados	Administrador Aplicación
17	MEDIA	Crear solicitud de cotización misma que deberá dirigirse a un diseñador que será seleccionado por la persona encargada del rol de Administrativo	Administrativo
18	MEDIA	Una vez firmado el contrato deberá ser digitalizado y cargado a la aplicación y su enlace deberá ser accesible desde todos los roles, siendo asignado ya al diseñador que ejecutara el contrato	Administrativo
19	MEDIA	El área administrativa podrá registrar y emitir recibos de pagos por concepto de abonos	Administrativo
20	MEDIA	El diseñador deberá cargar al sistema su diseño del requerimiento, así como la imagen 3D del diseño y en base a ello generar los datos de la	Diseño (continúa)

		Cotización/proforma que se envía al cliente inicialmente por correo electrónico o físico.	
21	MEDIA	El diseñador tendrá la posibilidad de actualizar las versiones de los archivos de diseño y emitir nuevas cotizaciones o proformas	Diseño
22	MEDIA	Los diseñadores podrán agregar nuevas versiones de los archivos de diseño así como de imagen 3d del proyecto manteniendo el versionamiento de los mismos asociados a los cambios que sean requeridos	Diseño
23	MEDIA	Concluido el diseño se deberá enviar a producción donde el responsable de coordinar el proceso de producción emitirá la orden de producción y designara responsables para las etapas específicas de producción.	Producción

La escala de importancia se basa en tres estados específicos definidos: Alta, Media y Baja mismas que han sido definidos por el Producto Owner.

3.1.5 Diseño

3.1.6 Modelos de base de datos

De acuerdo a la lista de requerimientos priorizados se ha establecido una solución del modelo de datos que permita el manejo de etapas para el avance de los proyectos, para ellos el modelo conceptual de base de datos se establece de la siguiente forma:

ESPACIO EN BLANCO INTENCIONAL

Modelo Conceptual de Base de Datos

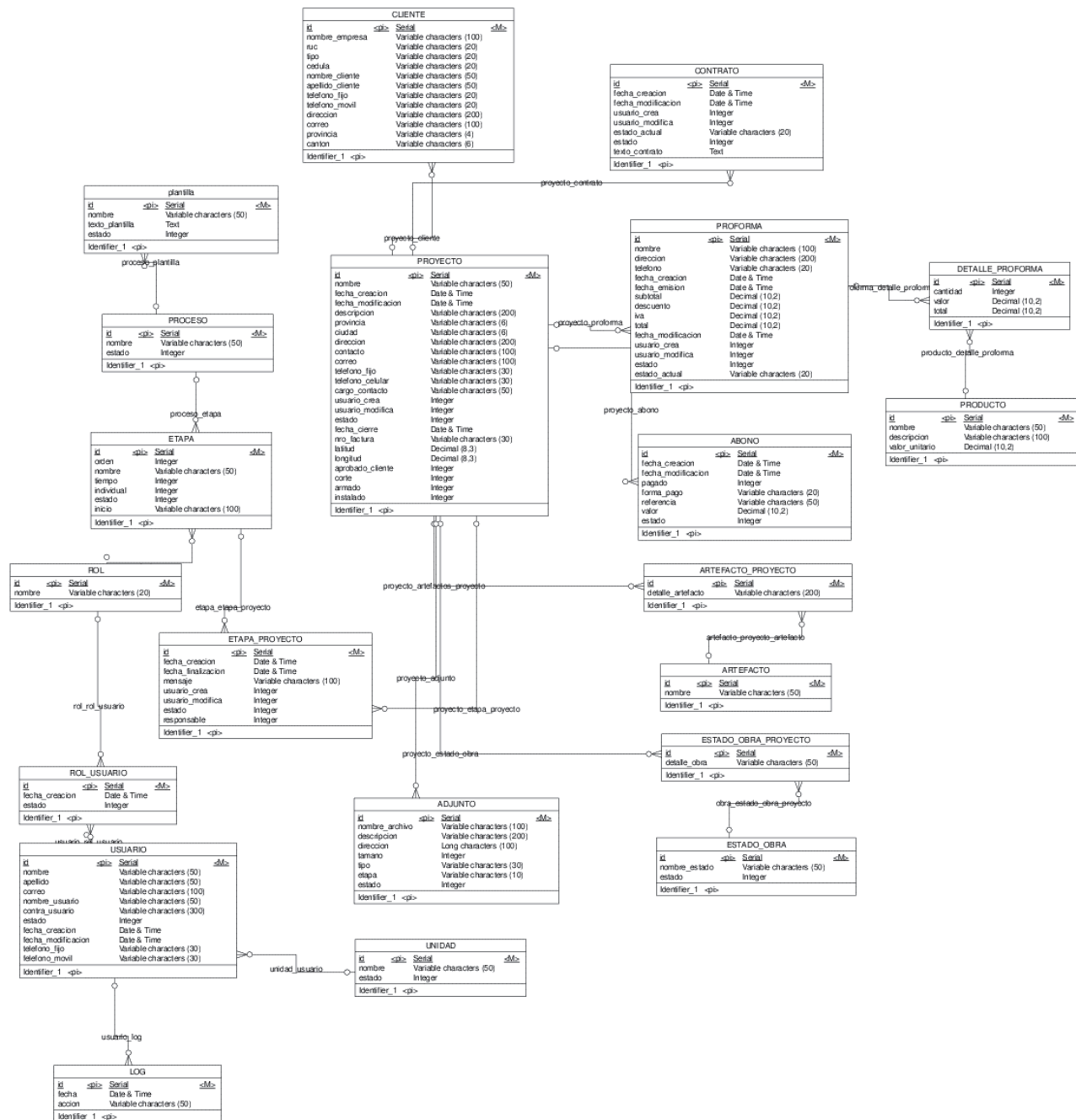


Figura 12. Modelo conceptual de la aplicación

Modelo Físico de Base de Datos

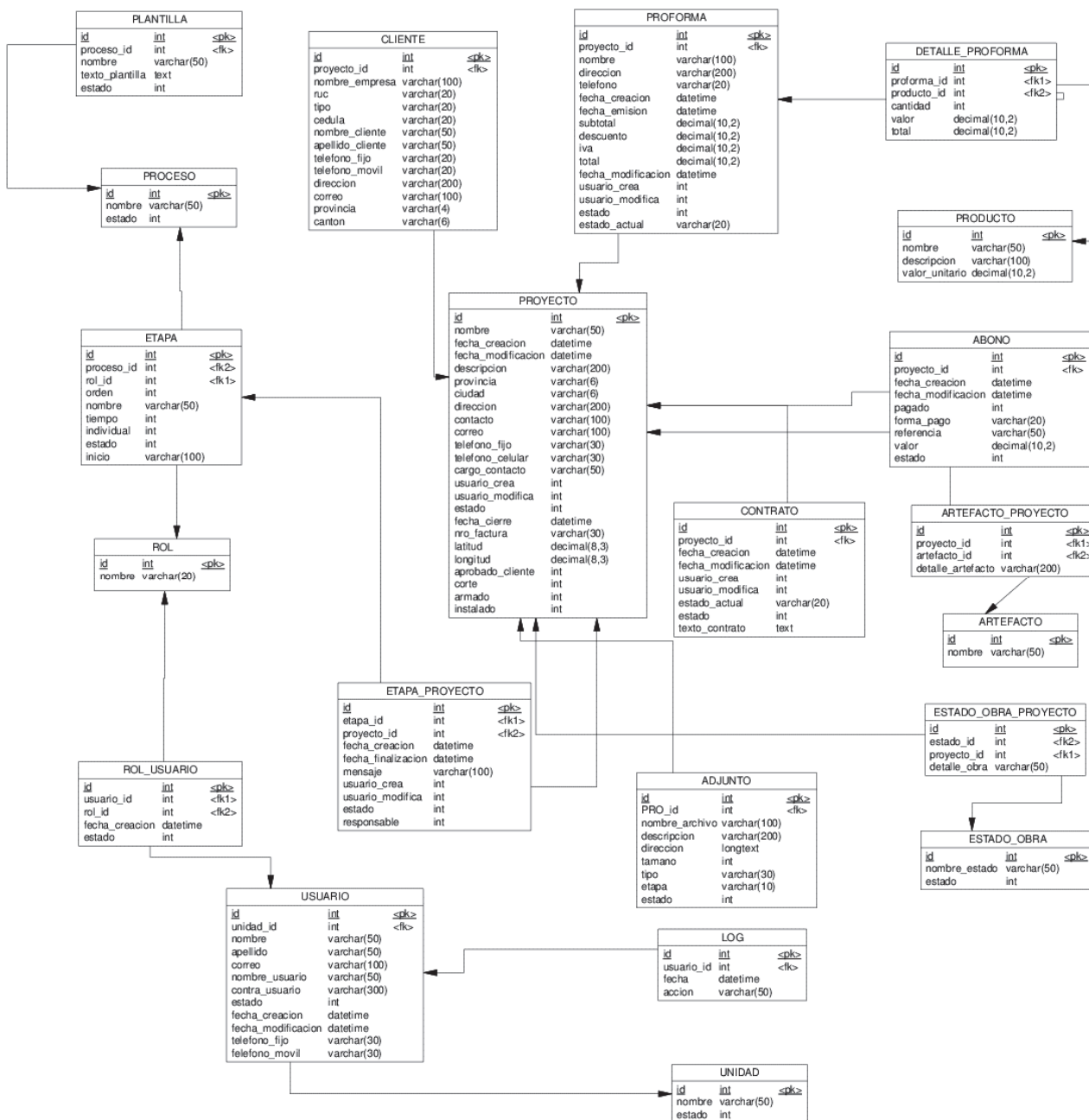


Figura 13. Modelo físico de la aplicación

3.1.7 *Sprint Nro. 1 “Usuarios y Seguridades”*

El primer sprint está definido para implementar las funcionalidades de creación, modificación y administración de usuarios, así como de los roles que cada uno de ellos requiere para llevar a cabo las diferentes responsabilidades dentro del proceso del sistema.

3.1.7.1 *Planificación*

Se convoca a la reunión de planificación del sprint con la participación del equipo de trabajo establecido y que mediante acta NRO-002-METALEC, se definen los requisitos tanto funcionales como no funcionales y que con la suscripción de la misma se aprueba como están definidos a continuación:

3.1.7.2 *Requerimientos funcionales y no funcionales*

Tabla 10.
Requerimientos funcionales Sprint Nro. 1

<i>Requerimientos Funcionales</i>	
1	El sistema deberá permitir la creación de usuarios, mismos que contarán con un usuario único y una contraseña como método de autenticación
2	La contraseña deberá estar encriptada y no deberá poder conocerse
3	Cada uno de los usuarios deberá poder contar con uno o varios roles
4	Los roles establecidos para cada uno de los usuarios estarán relacionados a las actividades que deben cumplir dentro del proceso de producción
5	Los usuarios una vez creados no podrán ser eliminados, únicamente deberán ser deshabilitados
6	Al autenticarse el usuario lo realizará únicamente con su usuario y contraseña
7	Si la autenticación es correcta el sistema deberá asignar el primer rol establecido para el usuario como predeterminado
8	Deberá existir una funcionalidad que permita seleccionar el rol requerido de acuerdo a los roles asignados
9	La aplicación deberá mostrar siempre el nombre del usuario así como el rol en un lugar visible de la aplicación.
10	El único que podrá activar o desactivar usuarios será el rol de administrador
11	El usuario deberá poder cambiar su contraseña para lo cual deberá ingresar la contraseña actual y la nueva.
12	Los roles podrán ser habilitados o deshabilitados para cada usuario (continúa)

Requerimiento no funcionales	
1	El sistema deberá funcionar independiente del sistema operativo
2	Manejar una presentación minimalista

3.1.7.3 Historias del usuario

La definición de historial del usuario para el sprint se usa una reproducción parcial de la lista de requerimiento priorizados conforme se estableció dentro del ACTA NRO-001-METALEC.

Tabla 11.
Historias del usuario Sprint Nro. 1

<i>Lista de Requerimientos Priorizados</i>			
<i>Nro</i>	<i>Prioridad</i>	<i>Requerimiento</i>	<i>Modulo</i>
1	ALTA	Gestión de usuarios	Administrador Aplicación
4	ALTA	El sistema deberá contar con la seguridad para que cada usuario cuente con un acceso personalizado de acuerdo a sus funciones	Seguridades
5	ALTA	La autenticación se deberá hacer mediante un nombre de usuario y contraseña misma que deberán estar encriptada	Seguridades
6	ALTA	El acceso a cada uno de los roles de usuario deberá ser transparente para el usuario y su modificación deberá realizarlo el administrador de la aplicación.	Seguridades
7	ALTA	La aplicación deberá estar compuesta por tres secciones: Un encabezado con el logo de la empresa, datos de identificación del usuario	Seguridades
8	ALTA	Todo ingreso al sistema deberá contar con un registro o campos de auditoria	Seguridades (continúa)

3.1.7.4 Equipo de trabajo

El equipo de trabajo para el primer sprint se mantiene lo establecido en el proyecto manejando dos funciones principales como lo es desarrollo y pruebas.

Tabla 12.
Roles para el Sprint Nro. 1

<i>Rol</i>	<i>Persona</i>	<i>Área</i>
Producto Owner	Xavier Sandoval	Gerencia
SCRUM Master	Alberto Coba	Desarrollo
Team	Alberto Coba	Desarrollo
	Xavier Sandoval	Pruebas

3.1.7.5 Actores del Sistema

Los actores identificados para el primer sprint corresponden a los siguientes:

Tabla 13.
Actores para el Sprint Nro. 1

<i>Actor</i>	<i>Descripción</i>
Usuario Administrador	Es el usuario encargado de gestionar el sistema por lo cual cuenta con privilegios de cambiar parámetros y datos del sistema.

3.1.7.6 Diagramas de Casos de Uso

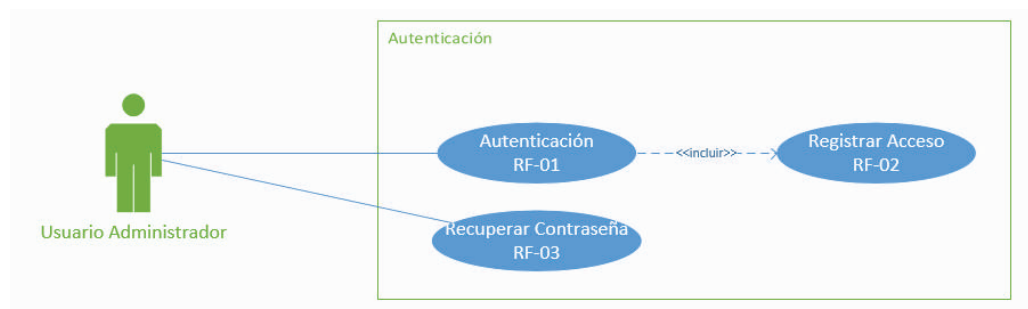


Figura 14. Caso de uso autenticación

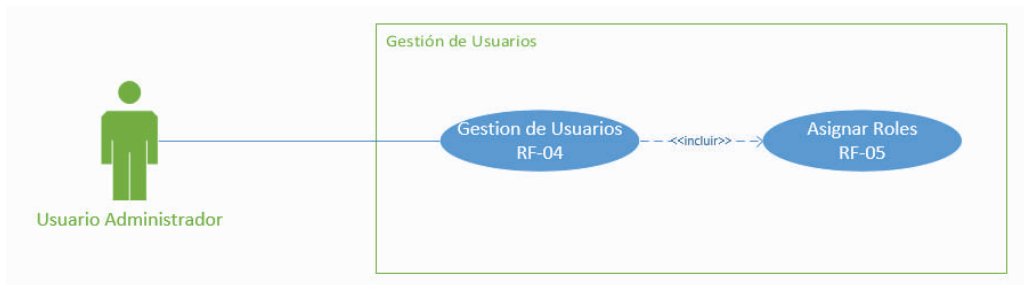


Figura 15. Caso de uso autenticación

3.1.7.7 Especificaciones de Casos de Uso

Tabla 14.
Especificación caso de uso autenticación

ID:	RF-01
Descripción:	Autenticación de Usuarios registrados en el sistema
Precondición:	El usuario debe estar registrado en el sistema
Flujo normal:	<ol style="list-style-type: none"> 1 El sistema solicita nombre de usuario y contraseña al usuario 2 El sistema valida que el usuario este registrado, activo y sus datos validos 3 El sistema registra el acceso en log de auditoria 4 Cargar su perfil de acuerdo al rol y dirigir a la página principal
Flujos Alternos:	<ol style="list-style-type: none"> 1 Se podrá recuperar la contraseña mediante el ingreso del usuario y la dirección de correo registrada
Excepciones:	<ol style="list-style-type: none"> 1 Si el usuario ha sido desactivado o eliminado deberá mostrar la página de error 2 El sistema en caso de no existir el usuario deberá envía a la página de error
Notas:	<ol style="list-style-type: none"> 1 El nombre de usuario deberá ser único para cada usuario

2	Las contraseñas deberán estar encriptadas
3	Para recordar la contraseña se enviará un correo electrónico con la opción de actualización de contraseña en línea registrando datos de auditoria y solicitando cambio de contraseña.

Tabla 15.
Especificación caso de uso registro de acceso

ID:	RF-02
Descripción:	Registro de acceso
Precondición:	El usuario debe estar registrado en el sistema
Flujo normal:	<ol style="list-style-type: none"> 1 SI la autenticación es exitosas se registrará el usuario, fecha y hora de acceso 2 El registro incluirá la dirección IP desde donde se realiza la autenticación
Flujos	Ninguno
Alternos:	
Excepciones:	<ol style="list-style-type: none"> 1 Si la autenticación falla por contraseña se registrará el nombre de usuario, fecha y dirección IP 2 Si la autenticación falla por el usuario se registrará el nombre de usuario, fecha y dirección IP
Notas:	<ol style="list-style-type: none"> 1 Ninguna

Tabla 16. Especificación caso de uso recuperar contraseña

ID:	RF-03
Descripción:	Recuperar contraseña
Precondición:	El usuario debe estar registrado en el sistema
	(continúa)

Flujo normal:	<ol style="list-style-type: none"> 1 Ingresar el nombre de usuario y el correo electrónico registrado 2 Se genera una nueva clave segura y se remitirá mediante correo electrónico junto al link para ingreso 3 El usuario será obligado a cambiar su contraseña
Flujos	Ninguno
Alternos:	
Excepciones:	<ol style="list-style-type: none"> 1 Si el usuario no recibe el correo, el usuario deberá ser gestionado por el administrador del sistema 2 Si los datos ingresados para recuperación no coinciden, el sistema no realizará ninguna acción
Notas:	<ol style="list-style-type: none"> 1 El correo deberá ser generado mediante un formato preestablecido

Tabla 17.
Especificación caso de uso gestión de usuarios

ID:	RF-04
Descripción:	Gestión de Usuarios
Precondición:	Ser administrador del sistema
Flujo normal:	<ol style="list-style-type: none"> 1 Búsqueda de usuarios de la lista de registrados 2 Selección de usuario mediante un botón que indique edición permitiendo la actualización de los datos 3 Incluir un botón que permita abrir un formulario de creación de un nuevo usuario. 4 Deberá incluir un botones que indique asignar permisos
Flujos	
Alternos:	
Excepciones:	(continúa)

	1	No se podrá eliminar ningún usuario, únicamente desactivarlo
	2	Todos los usuarios se crean con la opción de cambiar su contraseña al inicio
Notas:		
	1	El nombre de usuario deberá ser único para cada usuario

Tabla 18.
Especificación caso de uso asignar roles

ID:	RF-05
Descripción:	Asignar roles
Precondición:	El usuario debe estar registrado en el sistema
Flujo normal:	
	1 Se asignaran los roles a los cuales el usuarios seleccionado tendrá acceso
	2 De ser necesario se podrá quitar los roles ya asignados
	3 Se registra el cambio o creación en un log transaccional
Flujos	
Alternos:	1
Excepciones:	
	1 Todos los usuarios deberán contar con al menos un rol asignado
Notas:	
	1 Los roles podrán ser activados o desactivados de acuerdo a las funciones que se tenga

ESPACIO EN BLANCO INTENCIONAL

3.1.7.8 Modelo de Datos

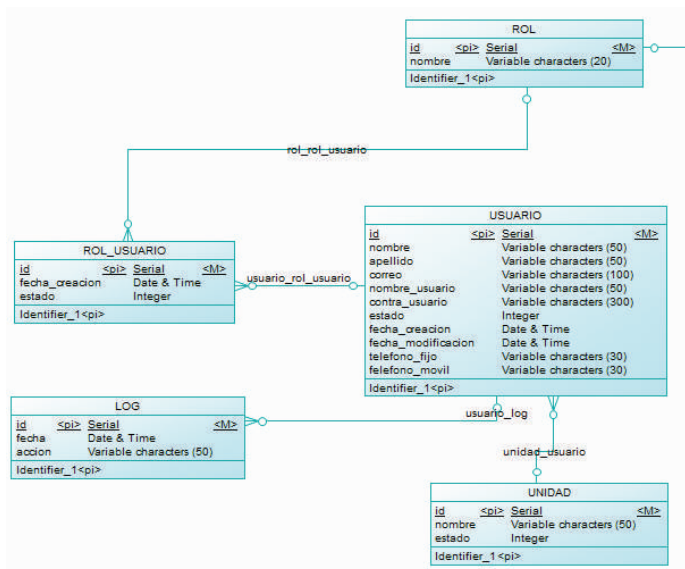


Figura 16. Modelo de base de datos Sprint Nro. 1

3.1.7.9 Construcción y Pruebas

Para la etapa de construcción se utilizó como medio de control la herramienta Sprint to Meter, misma que permite dar un seguimiento diario al avance del equipo tanto a nivel de codificación como pruebas:

Nº	Historia	Nº tareas	Nombre	Historia	Nombre	Tareas	Asignado 1	Asignado 2	Probado hoy/Pasa hacer	Hecho %	Oct 24	Oct 25	Oct 26	Oct 27
1	Desarrollo de imágenes y componentes	3	Alberto Ceba	3					27% (30)	27% (32)	24% (14)	23% (11/6)	30% (21)	
2	Creación de páginas básicas para sistema	2	Alberto Ceba	2					27% (11)	27% (66)	24% (12)	23% (11/6)	30% (19)	
3	Autenticación de Usuarios	5	Alberto Ceba	5					17% (12)	33% (8)	30% (2)	30% (12)	33% (11)	
4	Creación de Roles	5	Alberto Ceba	5					30% (6)	30% (12)	30% (4)	30% (4)	33% (11)	
5	Pruebas de funcionamiento	5	Xavier Sandoval	5					17% (8)	30% (12)	30% (2)	30% (4)	33% (11)	
6	Pruebas de funcionamiento	5	Xavier Sandoval	5					17% (8)	30% (12)	30% (2)	30% (4)	33% (11)	

Figura 17. Historia del Sprint Nro. 1

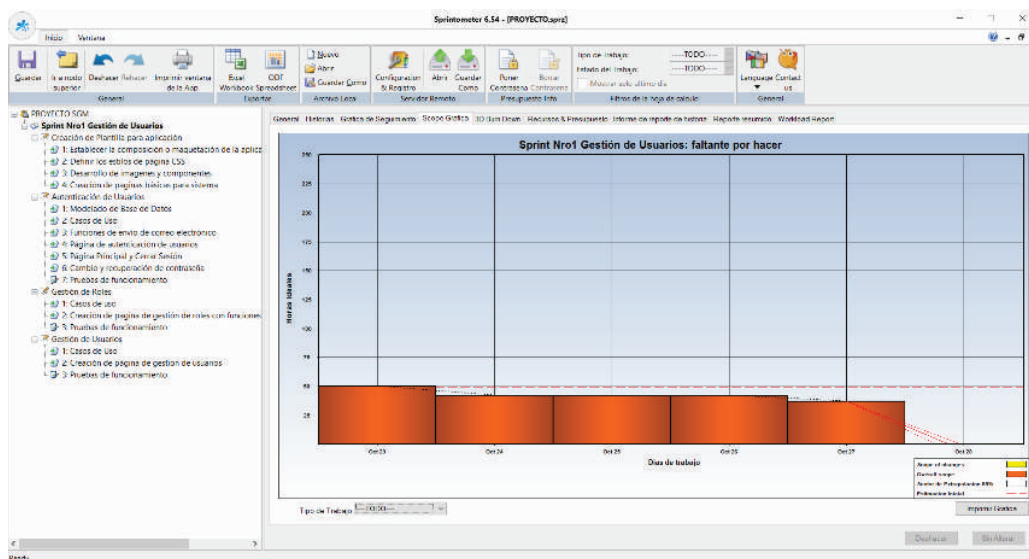


Figura 18. Avance del Sprint Nro. 1

3.1.8 Sprint Nro. 2 Administración

3.1.8.1 Planificación

Para el desarrollo del segundo sprint el equipo del proyecto fue convocado para la definición de los requerimientos funcionales y no funcionales para este sprint, mismo que está basado en la lista de requerimientos priorizados inicial. Dentro de la reunión se establecieron los requerimientos, mismos que fueron aprobados y suscritos mediante ACTA-003.METALES.

3.1.8.2 Requerimientos funcionales y no funcionales

Tabla 19.
Requerimientos funcionales Sprint Nro. 2

<i>Requerimientos Funcionales</i>	
1	Deberá permitir la administración de los datos
2	Eliminación lógica de datos mediante el uso de un campo de estado
3	Se deberá mostrar una lista de los datos existentes con su estado y la opción para realizar modificaciones.
4	Cada que se vaya a realizar un cambio se deberá confirmar si se lo desea hacer
5	En caso de eliminación, siempre se podría reversar el proceso. (continúa)

6	En el caso de tablas que sean de uso exclusivo del proceso no se debe poder modificar
7	Para acceder a los cambios se deberá contar con una lista de los datos registrados para la selección del registro exclusivo a modificar
8	Las listas en caso de considerarse puedan crecer deberán contar con buscador y paginación

Requerimiento no funcionales

1	En las listas de datos se deberá poder diferenciar las filas para una mejor visualización.
2	Incluir botones de acción con colores claros que los identifiquen fácilmente

3.1.8.3 Equipo de trabajo

Tabla 20.
Equipo de trabajo Sprint Nro. 2

<i>ROL</i>	<i>Persona</i>	<i>Área</i>
Producto Owner	Xavier Sandoval	Gerencia
SCRUM Master	Alberto Coba	Desarrollo
Team	Alberto Coba	Desarrollo
	Xavier Sandoval	Pruebas

3.1.8.4 Actores del Sistema

Los actores identificados para el primer sprint corresponden a los siguientes:

Tabla 21.
Actores del Sprint Nro. 2

<i>Actor</i>	<i>Descripción</i>
Usuario	Es el usuario encargado de gestionar el sistema por lo cual cuenta con privilegios de cambiar parámetros y datos del sistema.
Administrador	

3.1.8.5 Casos de Uso

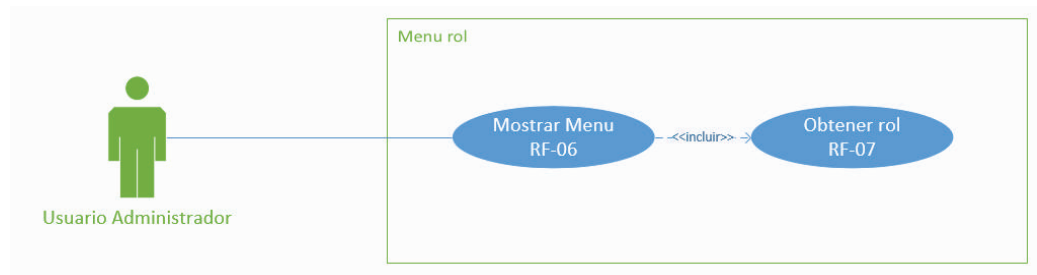


Figura 19. Casos de uso menú rol

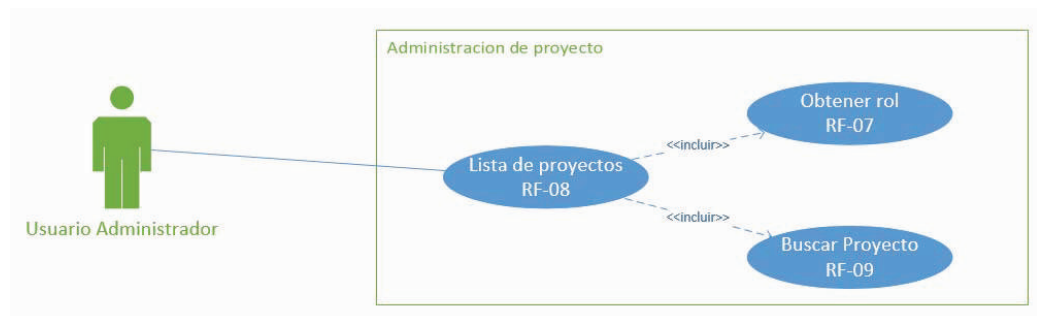


Figura 20. Casos de uso administración de proyecto

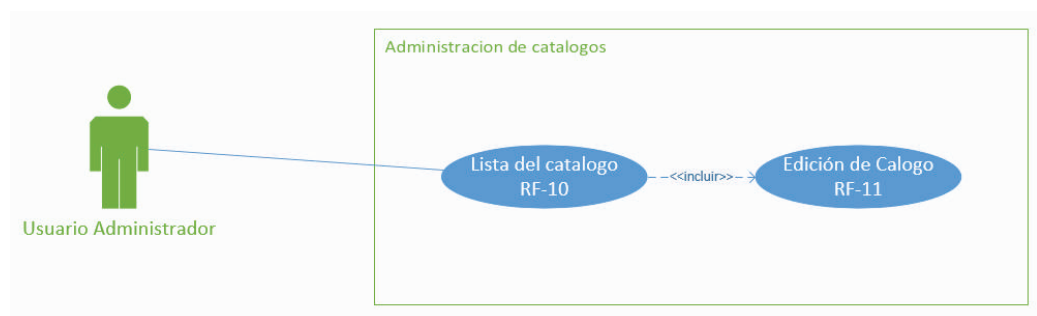


Figura 21. Casos de uso administración de catálogos

3.1.8.6 Especificaciones de Casos de Uso

Tabla 22.

Especificación caso de uso mostrar menú y obtener rol

ID:	RF-06
Descripción:	Mostrar Menú
Precondición:	Ser administrador del sistema
Flujo normal:	<ol style="list-style-type: none"> 1 El menú de administración de mostrará únicamente al administrador 2 Las opciones llevaran a una página específica 3 El control de acceso a los elementos será controlado por una variable de sesión 4 El menú estará disponible desde todas las páginas
Flujos Alternos:	<ol style="list-style-type: none"> 1 La opciones no serán visibles para ningún usuario que no sea administrador
Notas:	<ol style="list-style-type: none"> 1 El menú será desplegable ubicado bajo en encabezado de cada página
ID:	RF-07
Descripción:	Obtener rol
Precondición:	Estar autenticado en el sistema
Flujo normal:	<ol style="list-style-type: none"> 1 El usuario al autenticarse tendrá un rol activo 2 Se obtendrá el rol activo de la sesión y se obtendrá el registro de rol activo 3 El rol activo permitirá realizar varias funciones dentro del sistema por lo cual estará disponible en toda la aplicación

(continúa)

Flujos	
Alternos:	
Excepciones:	
1	Si el rol no está definido la aplicación direccionara a la página de error
Notas:	
1	En toda la sesión estará disponible

Tabla 23.
Especificación caso de uso administración de proyecto

ID:	RF-08
Descripción:	Administración de proyecto
Precondición:	Estar autenticado en el sistema
Flujo normal:	<ol style="list-style-type: none"> 1 Obtiene lista de todos los proyectos registrados 2 Estos se representaran en orden de fecha de creación y paginados 3 De acuerdo a la opción se presentara un submenú de selección de acciones posibles 4 Si se establece edición, permitirá abrir una ventana de cambio de datos relacionados al proyecto solo disponible para administrador
Flujos Alternos:	<ol style="list-style-type: none"> 1 NO se cambiaran campos de auditoria como fechas ni usuarios de creación
Notas:	<ol style="list-style-type: none"> 1 Agregar una alerta antes de grabar

Tabla 24.
Especificación caso de uso buscar proyecto

ID:	RF-09
Descripción:	Buscar Proyecto
Precondición:	Estar autenticado en el sistema
Flujo normal:	<ol style="list-style-type: none"> 1 En primera instancia obtener los proyectos ordenados por fecha de creación 2 Incluir un buscador 3 Todos los resultados deberán ser paginados 4 La búsqueda podrá realizarse por varios campos, esto es nombre proyecto, descripción o cliente
Flujos	
Alternos:	
Excepciones:	<ol style="list-style-type: none"> 1 Cada usuario en base a su rol tendrá una bandeja de entrada con las etapas activas de acuerdo a su rol
Notas:	<ol style="list-style-type: none"> 1 Todos los usuarios podrán tener acceso al buscador y las funciones dependerán del rol

Tabla 25.
Especificación caso de lista de catalogo

ID:	RF-10
Descripción:	Lista de catalogo
Precondición:	Estar autenticado en el sistema y ser administrador
Flujo normal:	<ol style="list-style-type: none"> 1 Se cargaran los datos del catálogo específico ordenado por llave primaria 2 De ser necesario por el tipo de catálogo se deberá incluir paginación 3 Cada registro contara con un enlace para abrir el registro en una página de edición específica (continúa)

	4 El acceso a la lista de cualquier catalogo será únicamente disponible para el administrador dela aplicación.
Flujos	Ninguno
Alternos:	
Excepciones:	Ninguna
Notas:	
	1 Ninguna

Tabla 26.
Especificación caso de edición de catalogo

ID:	RF-11
Descripción:	Edición de catalogo
Precondición:	Estar autenticado en el sistema y ser administrador
Flujo normal:	<ol style="list-style-type: none"> 1 El catalogo abrirá un registro identificado por la llave primaria 2 Los campos podrán ser modificados con restricciones según el caso 3 La eliminación será lógica por lo cual se establece un estado 1 como activo 0 como pasivo
Flujos	Ninguno
Alternos:	
Excepciones:	
	1 Ninguna
Notas:	
	1 Los campos editables dependen del catálogo

3.1.8.7 Modelo de Datos

Para el segundo sprint el modelo de datos abarca casi todas las tablas de catálogos para el sistema y adicionalmente las tabla de proyectos y clientes, por lo cual hay que hacer referencia al modelo Conceptual y Físico del proyecto de proyecto en conjunto.

3.1.8.8 Construcción y Pruebas

Para el segundo sprint la herramienta SprintToMeter se mantiene como la herramienta de seguimiento del proceso de desarrollo de la aplicación.

Gráfico de avance del segundo sprint.

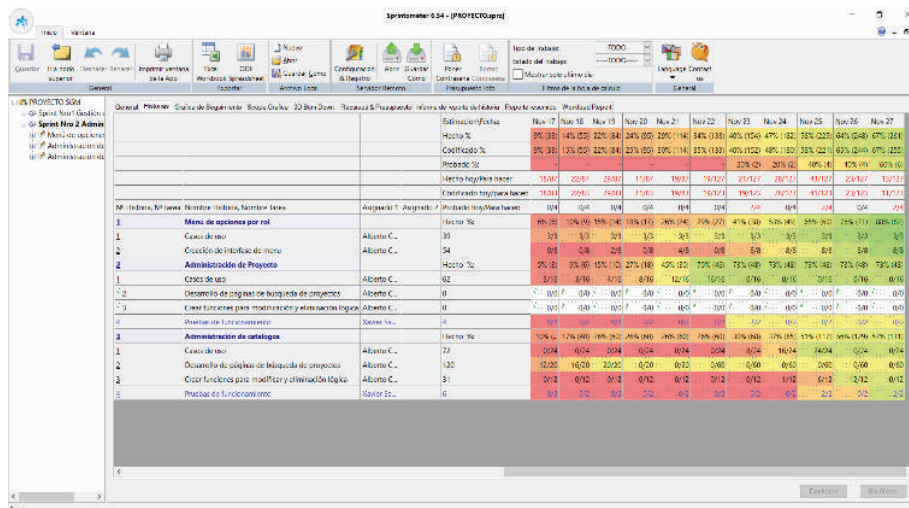


Figura 22. Historias del usuario y avance de Sprint Nro. 2

Gráfico de seguimiento de sprint 2

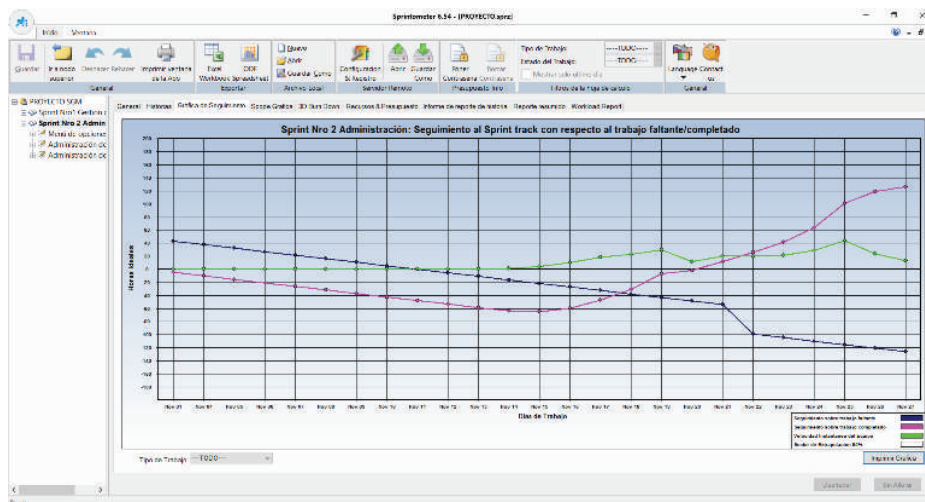


Figura 23. Seguimiento de Sprint Nro. 2

3.1.9 Sprint Nro. 3 Gestión de Proyectos

3.1.9.1 Planificación

El tercer Sprint está definido para ser el que automatice cada una de las etapas dentro del proceso de producción de los proyectos, por ello se hace un reconocimiento de validación de cada una de estas etapas por lo que se establece un flujo definido. En el mismo sentido se realizó el levantamiento de los requerimientos funcionales y no funcionales de cada una de las etapas, mismos que fueron plasmados en el ACTA-004-METALEC.

3.1.9.2 Requerimientos Funcionales y No funcionales

Proceso de Producción

El proceso de producción estará compuesto por las siguientes etapas:

1. Creación del Proyecto
2. Diseño de propuesta y proforma
3. Aprobación y contrato
4. Diseño Definitivo
5. Producción
6. Corte y preparación
7. Ensamblaje de modulares, armado
8. Instalación de modulares y muebles
9. Cierre del proyecto y registro de factura.

Para estas etapas se establecen los siguientes requerimientos:

Creación de proyecto

Tabla 27.
Requerimientos funcionales creación de proyecto

<i>Requerimientos Funcionales</i>	
1	Se requiere el ingreso de datos del Proyecto
2	Al ingresar la dirección el sistema deberá ubicar en un mapa la ubicación geográfica y almacenarla (continúa)

3	Se deberá registrar al cliente el mismo que podría ser persona natural o jurídica
4	Una vez creado el proyecto el mismo permitirá generar una orden de diseño la cual deberá ser generada para la firma del cliente
5	El proyecto una vez creado deberá permitir seleccionar el diseñador que se encargue de la propuesta, para ello se deberá mostrar una lista de usuarios que tengan el rol de diseño.
6	Al asignar el proyecto a un usuario del rol de diseño, el proyecto deberá ser accesible para la función específica que cumplirá el siguiente rol.
7	El proyecto mientras no esté en el rol solo será accesible para verificar avance del mismo.
<i>Requerimiento no funcionales</i>	
1	La página de ingreso deberá ser clara y con todos los capos necesarios accesibles
2	La ubicación geográfica podrá ser afinada de acuerdo a la necesidad

Diseño y propuesta (Pre diseño)

Tabla 28.
Requerimientos funcionales pre diseño

<i>Requerimientos Funcionales</i>	
1	Una vez que se haya creado el proyecto este deberá llegar a una bandeja de entrada correspondiente al pre diseño del mismo
2	El pre diseño deberá permitir al usuario específico de diseño cargar al sistema los archivos de diseño, sea AutoCAD/kitchen draw
3	Se deberá generar la proforma referente al proyecto
4	El sistema valida si se ha cumplido con estos requisitos para poder finalizar la etapa
5	Al finalizar la etapa se deberá determinar el siguiente responsable, de acuerdo al proceso determinado la siguiente etapa sería el registrar la

(continúa)

	aprobación, y no está bajo ninguna responsabilidad, por ello es enviada de forma general al rol de Atención al cliente
6	Los archivos cargados al sistema así como la proforma estarán disponibles mientras la etapa este en PREDISEÑO
7	Se registrará la fecha y hora de creación de la etapa así como de finalización
Requerimiento no funcionales	
1	Los archivos creados se mostrarán en listas fácilmente identificables y con botón para descarga

Aprobación Contrato

Tabla 29.
Requerimientos funcionales aprobación contrato

<i>Requerimientos Funcionales</i>	
1	Una vez que se cargaron el pre diseños, el rol de ATENCION AL CLIENTE recibe en su bandeja de entrada la etapa correspondiente a aprobación.
2	La aprobación deberá ser indicada y afirmada por parte del usuario, para lo cual deberá cargar desde una plantilla precargada el formato de contrato para su edición.
3	La edición del contrato sería mediante un editor que cuente las herramientas necesarias de un editor de texto.
4	Los cambios del contrato podrán ser guardados en cualquier momento garantizando que no se pierdan.
5	Una vez concluida la edición del contrato, este se emitirá en formato PDF para ser impreso y suscrito por las partes.
6	Si el cliente está de acuerdo, se definirá el proyecto como aprobado y se podrá seleccionar el responsable del diseño definitivo, mismo que podrá ser cualquiera que tenga el rol de DISEÑO
7	Al finalizar se registrará la fecha y hora de finalización
Requerimiento no funcionales	
1	La edición de contrato debe ser flexible en relación a visualización

Diseño Definitivo

Tabla 30.
Requerimientos funcionales diseño definitivo

<i>Requerimientos Funcionales</i>	
1	Aprobado el proyecto se genera una nueva etapa para el rol DISEÑO a través de la bandeja de entrada, misma que deberá permitir cargar los diseños definitivos.
2	Los diseños definitivos consisten el planos de AutoCAD/kitchen Draw referentes a los modulares o muebles requeridos.
3	La página tendrá disponible los archivos de pre diseño realizados así como los datos de proforma.
4	El usuario deberá cargar los archivos de esta etapa los que serán identificados como correspondientes a DISEÑO.
5	Una vez cargados los archivos se podrá finalizar la etapa en la cual se enviará al rol de PRODUCCIÓN de forma general sin un responsable específico.
6	Se registrará la fecha y hora de finalización de la etapa
Requerimiento no funcionales	
1	Todos los archivos deben ser fácilmente accesibles y para descarga

Producción

Tabla 31.
Requerimientos funcionales producción

<i>Requerimientos Funcionales</i>	
1	Concluida la etapa de DISEÑO, se crea la etapa de PRODUCCIÓN
2	La etapa al no tener un producto específico, y facilidad de registro se establece que solo se afirmará que se ha realizado la acción específica dependiendo de la tapa que corresponda.
3	Una vez registrado el cumplimiento de la etapa se registrará, quien la realizo así como fecha y hora.

(continúa)

4	Al finalizar la etapa se deberá elegir el responsable de la siguiente etapa
Requerimiento no funcionales	
1	Todos los archivos deben ser fácilmente accesibles y para descarga

Producción Corte

Tabla 32.
Requerimientos funcionales producción

<i>Requerimientos Funcionales</i>	
1	Para esta etapa se deberá poder acceder a los archivos de diseño a fin de que sean fuente para realizar los cortes de los materiales necesarios de acuerdo a lo contratado
2	La etapa al no tener un producto específico, y facilidad de registro se establece que solo se afirmará que se ha realizado la acción específica dependiendo de la tapa que corresponda.
3	Una vez registrado el cumplimiento de la etapa se registrará, quien la realizo así como fecha y hora.
4	Al finalizar la etapa se deberá elegir el responsable de la siguiente etapa
<i>Requerimiento no funcionales</i>	
1	Todos los archivos deben ser fácilmente accesibles y para descarga

Producción Armado

Tabla 33.
Requerimientos funcionales armado

<i>Requerimientos Funcionales</i>	
1	Para esta etapa se deberá poder acceder a los archivos de diseño a fin de que sean fuente para realizar los cortes de los materiales necesarios de acuerdo a lo contratado

(continúa)

2	La etapa al no tener un producto específico, y facilidad de registro se establece que solo se afirmará que se ha realizado la acción específica dependiendo de la tapa que corresponda.
3	Una vez registrado el cumplimiento de la etapa se registrará, quien la realizo así como fecha y hora.
4	Al finalizar la etapa se deberá elegir el responsable de la siguiente etapa
<i>Requerimiento no funcionales</i>	
1	Todos los archivos deben ser fácilmente accesibles y para descarga

Instalación

**Tabla 34.
Requerimientos funcionales instalación**

<i>Requerimientos Funcionales</i>	
1	Para la etapa de instalación se tendrá acceso a información del proyecto así como del cliente.
2	Se deberá incluir un mapa que muestre la ubicación del proyecto
3	Incluye los archivos de diseño del proyecto
4	Se deberá generar una orden de instalación, misma que será suscrita por los responsables de la misma así como del supervisor de obra.
5	Se emitirá un acta de entrega recepción para firma del cliente
6	Cumplida la instalación el responsable deberá indicarlo y asignar la siguiente ETAPA
	Registrar la fecha, usuario de finalización de etapa
<i>Requerimiento no funcionales</i>	
1	Todos los archivos deben ser fácilmente accesibles y para descarga

ESPACIO EN BLANCO INTENCIONAL

Cierre

Tabla 35.
Requerimientos funcionales cierre de proyecto

<i>Requerimientos Funcionales</i>	
1	Para la etapa de instalación se tendrá acceso a información del proyecto así como del cliente.
2	Se deberá incluir un mapa que muestre la ubicación del proyecto
3	Incluye los archivos de diseño del proyecto
4	Se deberá generar una orden de instalación, misma que será suscrita por los responsables de la misma así como del supervisor de obra.
5	Se emitirá un acta de entrega recepción para firma del cliente
6	Cumplida la instalación el responsable deberá indicarlo y asignar la siguiente ETAPA
	Registrar la fecha, usuario de finalización de etapa
<i>Requerimiento no funcionales</i>	
1	Todos los archivos deben ser fácilmente accesibles y para descarga

3.1.9.3 Equipo de trabajo

Tabla 36.
Equipo de trabajo Sprint Nro. 3

<i>Rol</i>	<i>Persona</i>	<i>Área</i>
Producto Owner	Xavier Sandoval	Gerencia
SCRUM Master	Alberto Coba	Desarrollo
Team	Alberto Coba	Desarrollo
	Xavier Sandoval	Pruebas

ESPACIO EN BLANCO INTENCIONAL

3.1.9.4 Actores del Sistema

Los actores identificados para el primer sprint corresponden a los siguientes:

Tabla 37.
Actores del sistema Sprint Nro. 3

<i>Actor</i>	<i>Descripción</i>
Usuario	Es el usuario encargado de gestionar el de acuerdo a su rol asignado

3.1.9.5 Casos de uso

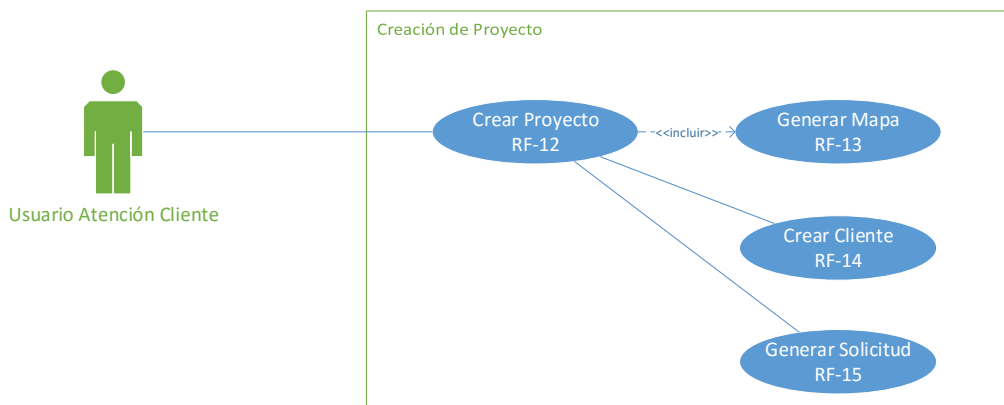


Figura 24. Casos de uso creación de proyecto

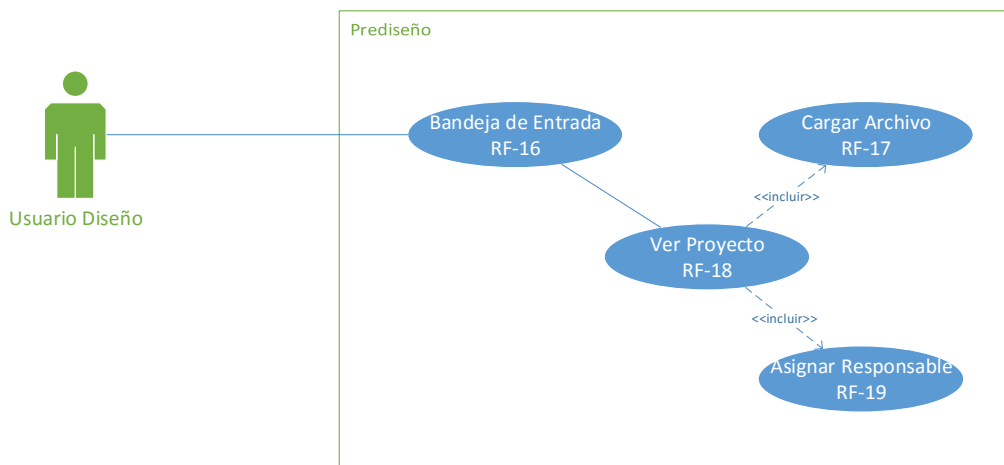


Figura 25. Casos de uso pre diseño

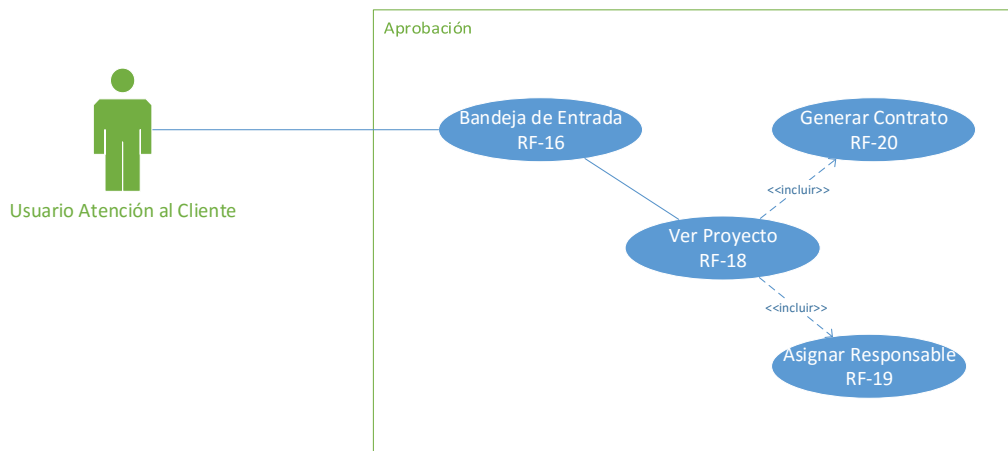


Figura 26. Casos de uso aprobación del proyecto

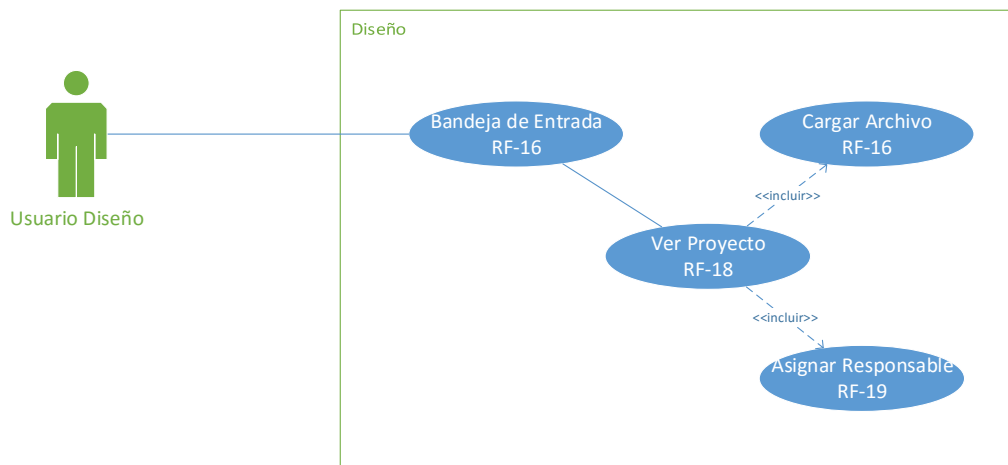


Figura 27. Casos de uso diseño definitivo de proyecto

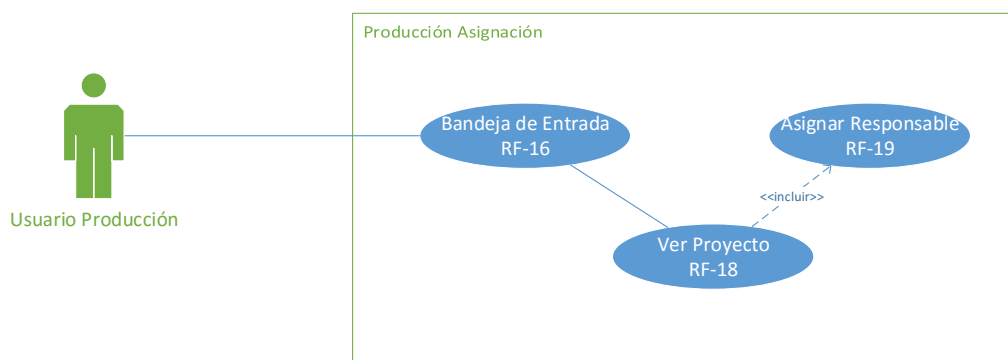


Figura 28. Casos de uso producción asignación

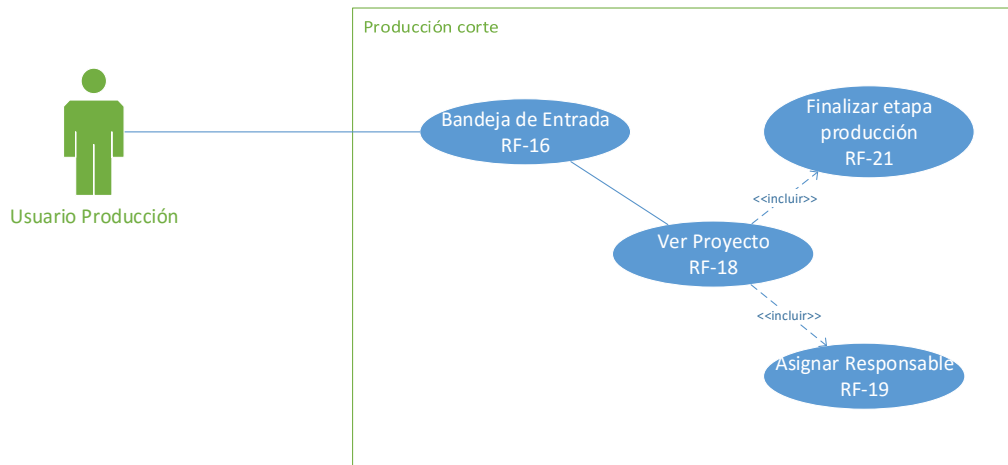


Figura 29. Casos de uso producción corte

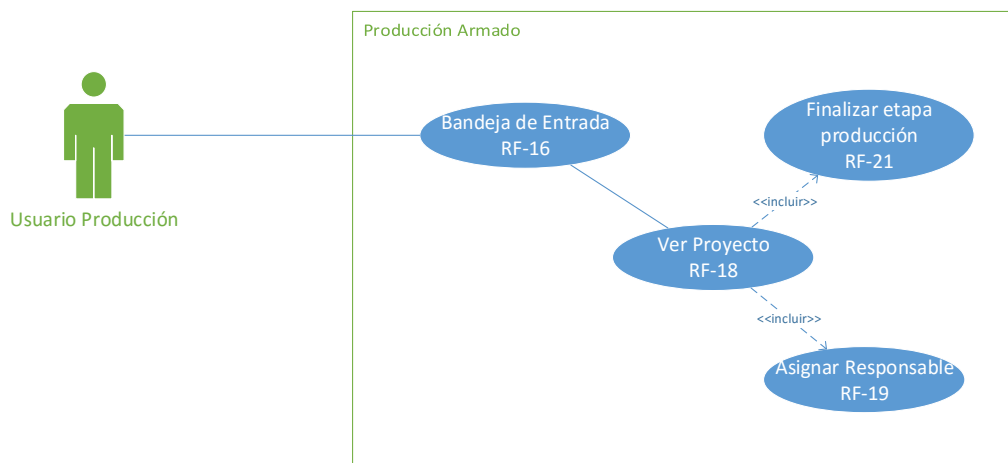


Figura 30. Casos de uso producción armado

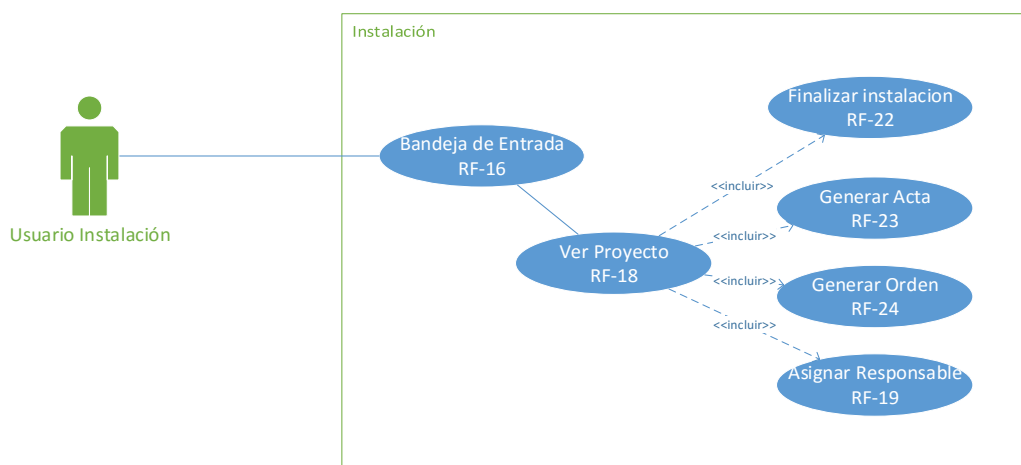


Figura 31. Casos de uso instalación

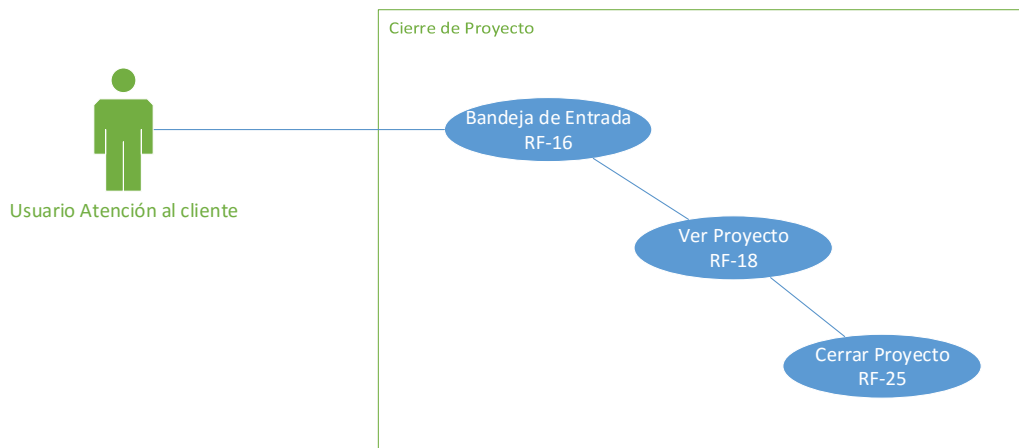


Figura 32. Casos de uso cierre de proyecto

3.1.9.6 Especificaciones de Casos de Uso

Tabla 38.
Especificación de casos de uso crear proyecto

ID:	RF-12
Descripción:	Crear proyecto
Precondición:	Estar con el rol Atención al Cliente
Flujo normal:	<ol style="list-style-type: none"> 1 Cargar datos predefinidos de los catálogos 2 Ingresar datos y validar correspondientes al proyecto y al cliente 3 Garantizar el almacenamiento de datos en mayúsculas 4 Al momento de ingresar la dirección del proyecto se deberá cargar la ubicación geográfica en el mapa 5 El cliente estará relacionado al proyecto 6 Se deberá seleccionar de una lista los artefactos que podrían incluirse en el proyecto. 7 Se establecerá características de la obra civil de inmueble 8 Al crear el proyecto se registrara la primera Etapa dentro del proyecto
	(continúa)

	9	Se creara una carpeta donde se almacenaran los archivos adjuntos del proyecto.
	10	Guardado el proyecto se genera la orden de diseño, misma que estará en formato PDF con la fecha de emisión del documento para su suscripción
Flujos	Ninguno	
Alternos:		
Excepciones:		
	1	Ninguna
Notas:		
	1	Siempre se deberá poder crear el proyecto

Tabla 39.
Especificación de casos de uso generar mapa

ID:	RF-13	
Descripción:	Generar mapa	
Precondición:	Estar con el rol Atención al Cliente	
Flujo normal:	1	Incluir el API de Google Maps al inicio de la página cargando una ubicación específica.
	2	Los cambios en las marcas dentro del mapa se deberán ejecutar y actualizar el campo de latitud y longitud.
	3	Mediante el evento de cambio del campo de dirección del proyecto, deberá obtener el texto de la dirección y cargarse en el mapa automáticamente.
	4	Si la dirección es cargada en el mapa se podrá arrastras la marca para ubicarla de forma más específica dentro de la aplicación.
	5	Los cambios del marcador se reflejara en los campos de latitud y longitud
		(continúa)

Flujos	Ninguno
Alternos:	
Excepciones:	1 Si la ubicación no es encontrada emitir mensaje de error
Notas:	1 Las direcciones se deberán centrar en la ciudad de Quito

Tabla 40.
Especificación de casos de uso crear cliente

ID:	RF-14
Descripción:	Crear cliente
Precondición:	Estar con el rol Atención al Cliente
Flujo normal:	1 Se tendrá un formulario relacionado al proyecto para obtener los datos del cliente
	2 Cada cliente estará relacionado a un proyecto
	3 El cliente podrá ser una persona natural o una persona jurídica por lo cual se deberá registrar tanto cédula como RUC
	4 Los datos deberán ser estandarizados en mayúsculas
	5 Los campos de ciudad y provincia deberán estar pre cargados
	6 El número de cédula podrá ser validado
Flujos	Ninguno
Alternos:	
Excepciones:	1 Ninguna
Notas:	1 Siempre se deberá poder crear el proyecto

Tabla 41.
Especificación de casos de uso generar solicitud

ID:	RF-15
Descripción:	Generar Solicitud
Precondición:	Estar con el rol Atención al Cliente
Flujo normal:	<ol style="list-style-type: none"> 1 Se tendrá un formulario relacionado al proyecto para ver información relevante del mismo 2 El código del proyecto permitirá contener datos relacionados y cargarlos en una plantilla 3 Crear la estructura de la solicitud con la imagen empresarial 4 El archivo HTML generado, deberá ser generado con el formato especificado mediante estilos específicos para impresión 5 La solicitud tendrá formato A4. 6 Generar a partir de la información recuperada el archivo PDF para impresión.
Flujos	Ninguno
Alternos:	
Excepciones:	<ol style="list-style-type: none"> 1 Ninguna
Notas:	<ol style="list-style-type: none"> 1 Siempre se deberá poder crear el proyecto

Tabla 42.
Especificación de casos de uso bandeja de entrada

ID:	RF-16
Descripción:	Bandeja de entrada
Precondición:	Estar autenticado en el sistema
Flujo normal:	<ol style="list-style-type: none"> 1 Se deberá obtener el rol con el que esta autenticado el usuario

(continúa)

	2 Realizar una búsqueda de las etapas relacionadas al rol que se encuentren activas para el rol
	3 Realizar la búsqueda de las etapas activas relacionadas al rol y al usuario específico.
	4 Las etapas recuperadas deberán estar ordenadas en relación a su fecha de creación ascendente
	5 La lista de etapas recuperadas deberá estar paginada
	6 Cada etapa tendrá un botón que abra la página específica para la tarea que corresponda
Flujos	El administrador tendrá acceso a todos las etapas de requerirse
Alternos:	
Excepciones:	
	1 Ninguna
Notas:	
	1 Ninguna

Tabla 43.
Especificación de casos de uso cargar archivo

ID:	RF-17
Descripción:	Cargar archivo
Precondición:	Estar autenticado en el sistema
Flujo normal:	1 Formulario para cargar archivos adjuntos a la aplicación
	2 Sera una función disponible para todo el sistema
	3 Sebera tener control de sesión para su acceso
	4 Deber registrar los datos del archivo así como la etapa en que se carga
	5 Almacenara los datos del archivo como ubicación descripción tamaño y tipo
	(continúa)

	6 Los archivos se cargaran en un directorio propio para cada proyecto (continúa)
Flujos	El administrador tendrá acceso a todos las etapas de requerirse
Alternos:	
Excepciones:	
	1 Si se excede el parámetro establecidos el archivo se generará un error.
Notas:	
	1 Ninguna

Tabla 44.
Especificación de casos de uso ver proyecto

ID:	RF-18
Descripción:	Ver proyecto
Precondición:	Estar autenticado en el sistema
Flujo normal:	1 Se deberá abrir el proyecto mediante un identificador
	2 Se deberá recuperar los principales elementos aportados en cada etapa
	3 Debe incluir un mapa con la ubicación geográfica de ejecución del proyecto
	4 Mediante diagramas de Gantt se mostrará de forma gráfica
	5 Deberá indicar las etapas y sus responsables así como los tiempos tomados en realizarlas, calculando el número de días en cada una
	6 Todos los datos serán únicamente de lectura
Flujos	Ninguno
Alternos:	
Excepciones:	

	1 Si se excede el parámetro establecidos el archivo se generará un error.
--	---------------------------------------------------------------------------

Notas:

	1 Ninguna
--	-----------

Tabla 45.
Especificación de casos de uso ver proyecto

ID:	RF-19
Descripción:	Asignar responsable
Precondición:	Estar autenticado en el sistema

Flujo normal:	1 Se deberá obtener la etapa actual del sistema
	2 Se presentara en base a la etapa actual los usuarios que tienen los permisos para la siguiente etapa
	3 Una vez seleccionado al siguiente usuario de registrará la finalización de la etapa actual
	4 Al finalizar la etapa actual se registrará la fecha y hora de finalización en la etapa
	5 Se creará una nueva etapa conforme al proceso establecido con los datos de responsable, estado fecha y hora.
	6 Se deberá re direccionar a la bandeja de entrada una vez concluido

Flujos En caso de que sea una etapa sin un usuario responsable directamente sino general se definirá la tarea sin responsable, sin embargo al cumplirse deberá registrarse que usuario realizo la acción.

Alternos:

Excepciones:	1 Ninguna
---------------------	-----------

Notas:

	1 Ninguna
--	-----------

Tabla 46.
Especificación de casos de uso generar contrato

ID:	RF-20
Descripción:	Generar contrato
Precondición:	Estar autenticado en el sistema
Flujo normal:	<ol style="list-style-type: none"> 1 Obtener la plantilla del contrato 2 Cargar en un editor WYSIWYG, que preste las facilidades para editar un documento. 3 La plantilla cargada será editable y sus cambios se guardaran y podrán recuperar posteriormente. 4 El edito podrá ampliarse a página completa para facilitar la edición 5 Concluida la edición el contenido del contrato se podrá generar en formato PDF para su impresión 6 Una vez generado el contrato no se podrá modificar.
Flujos	De ser necesario el administrador podrá modificar el contrato
Alternos:	generando una copia que permita mantener el original
Excepciones:	<ol style="list-style-type: none"> 1 Ninguna
Notas:	<ol style="list-style-type: none"> 1 Ninguna

Tabla 47.
Especificación de casos de uso finalizar etapa producción

ID:	RF-21
Descripción:	Finalizar etapa producción
Precondición:	Estar autenticado en el sistema (continúa)

Flujo normal:	1 Si se está en una etapa de producción presentar la opción de respuesta de finalización de tareas.
	2 La opción de ser afirmativa se registrará en el proyecto indicando que se completó determinada etapa de producción
	3 Al registrar la etapa como finalizada se permitirá ver la opción para finalización de etapa así como asignar el responsable
Flujos	Ninguno
Alternos:	
Excepciones:	
	1 Ninguna
Notas:	
	1 Ninguna

Tabla 48.
Especificación de casos de uso finalizar instalación

ID:	RF-22
Descripción:	Finalizar instalación
Precondición:	Estar autenticado en el sistema
Flujo normal:	1 Si se está en una etapa de instalación presentar la opción de respuesta de finalización de tareas.
	2 La opción de ser afirmativa se registrará en el proyecto indicando que se completó determinada etapa de instalación
	3 Al registrar la etapa como finalizada se permitirá ver la opción para finalización de etapa así como asignar el responsable para el cierre
Flujos	Ninguno
Alternos:	
Excepciones:	(continúa)

1	Ninguna
Notas:	
1	Ninguna

Tabla 49.
Especificación de casos de uso generar acta

ID:	RF-23
Descripción:	Generar Acta
Precondición:	Estar autenticado en el sistema
1	El código del proyecto permitirá contener datos relacionados y cargarlos en una plantilla
2	Crear la estructura de la acta con la imagen empresarial
3	El archivo HTML generado, deberá tener formato especificado mediante estilos para impresión
4	El acta tendrá formato A4.
5	Generar a partir de la información recuperada el archivo PDF para impresión.
Flujos	Ninguno
Alternos:	
Excepciones:	
1	Ninguna
Notas:	
1	Ninguna

Tabla 50.
Especificación de casos de uso generar orden

ID:	RF-24
Descripción:	Generar Orden (continúa)

Precondición:	Estar autenticado en el sistema
	1 El código del proyecto permitirá contener datos relacionados y cargarlos en una plantilla
	2 Crear la estructura de la acta con la imagen empresarial
	3 El archivo HTML generado, deberá tener formato especificado mediante estilos para impresión
	4 La orden de instalación tendrá formato A4.
	5 Generar a partir de la información recuperada el archivo PDF para impresión.
Flujos	Ninguno
Alternos:	
Excepciones:	
	1 Ninguna
Notas:	
	1 Ninguna

Tabla 51.
Especificación de casos de uso cerrar proyecto

ID:	RF-25
Descripción:	Cerrar proyecto
Precondición:	Estar autenticado en el sistema
	1 Se cargan en el formulario los abonos realizados durante la ejecución del proyecto
	2 Se carga el valor de la proforma y los productos determinados.
	3 Los valores deben mantener un balance de 0 a fin de que se pueda finalizar el proyecto (continúa)

	4 Una vez liquidado el proyecto, se deberá ingresar el valor y datos de la factura emitida para finalizar el proyecto
	5 Una vez finalizado se registrara la fecha de cierre del proyecto así como de las etapas.
Flujos Alternos:	Si no se ha saldado el total se deberá registrar los abonos hasta completar los saldos.
Excepciones:	1 Ninguna
Notas:	1 Ninguna

Historias del Usuario

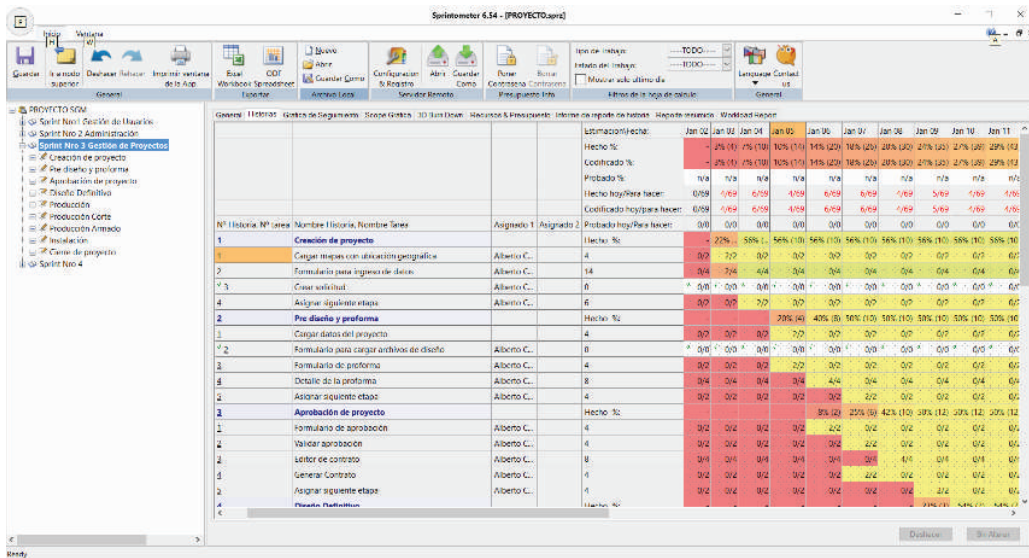


Figura 33. Avance del proyecto Historias del usuario

ESPACIO EN BLANCO INTENCIONAL

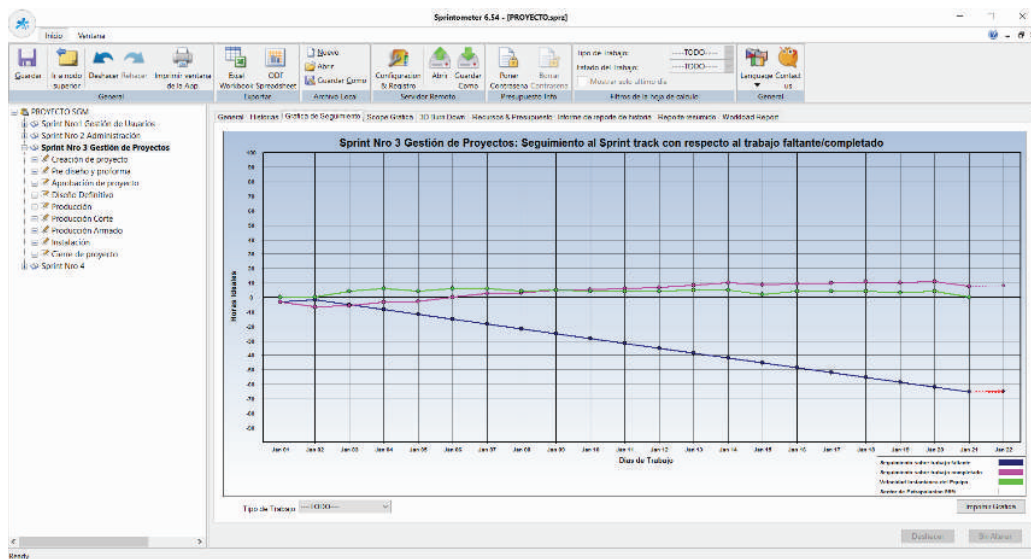


Figura 34. Seguimiento del proyecto

3.1.10 Sprint Nro. 4 “Registro de Pagos y proformas”

3.1.10.1 Planificación

El desarrollo del Sprint nro. 4 se realiza con el fin de determinar los requerimientos funcionales para el registro de pagos, para ello el equipo de trabajo mediante ACTA-005.METALEC, establece los requerimientos funcionales para la fase de desarrollo.

3.1.10.2 Requerimientos funcionales y no funcionales

Registro de Pagos

Tabla 52.
Requerimientos funcionales registro de pago

Requerimientos Funcionales	
1	El registro de pagos se podrá realizar en cualquier momento que sea necesario.
2	El acceso a registro de pago será establecido únicamente al personal que se encarga de atención al cliente.
3	El usuario deberá registrar el pago, así como la forma que se realiza el mismo.

(continúa)

- | | |
|---|-------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 | Al ser estos abonos Y no pagos definitivos, no se registrará cargar impositivas como el IVA, mismo que deberá determinarse en la factura final. |
| 5 | Se deberá emitir un recibo correspondiente al abono. |

Proforma

Tabla 53.
Requerimientos funcionales proforma

Requerimientos Funcionales	
1	La proforma será emitida por el encargado del diseño.
2	Se presentará una lista de los productos(modulares) que están disponibles para la cotización
3	Se indicará el número de productos calculándose los valores automáticamente así como los totales
4	La proforma deberá incluir valores como el IVA y DESCUENTOS.
5	Se emitirá la proforma en formato PDF para su impresión y entrega.
Requerimiento no funcionales	
1	La proforma deberá mantener la imagen corporativa

3.1.10.3 Equipo de trabajo

Tabla 54.
Equipo de trabajo Sprint Nro. 4

<i>Rol</i>	<i>Persona</i>	<i>Área</i>
Producto Owner	Xavier Sandoval	Gerencia
SCRUM Master	Alberto Coba	Desarrollo
Team	Alberto Coba	Desarrollo
	Xavier Sandoval	Pruebas

3.1.10.4 Actores del Sistema

Los actores identificados para el primer sprint corresponden a los siguientes:

Tabla 55.
Actores del sistema Sprint Nro. 4

<i>Actor</i>	<i>Descripción</i>
Usuario	Es el usuario encargado de gestionar el de acuerdo a su rol asignado

3.1.10.5 Casos de uso

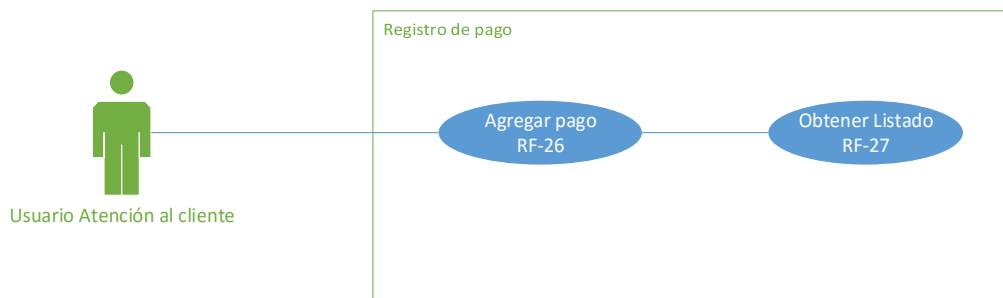


Figura 35. Casos de Uso registro de pago

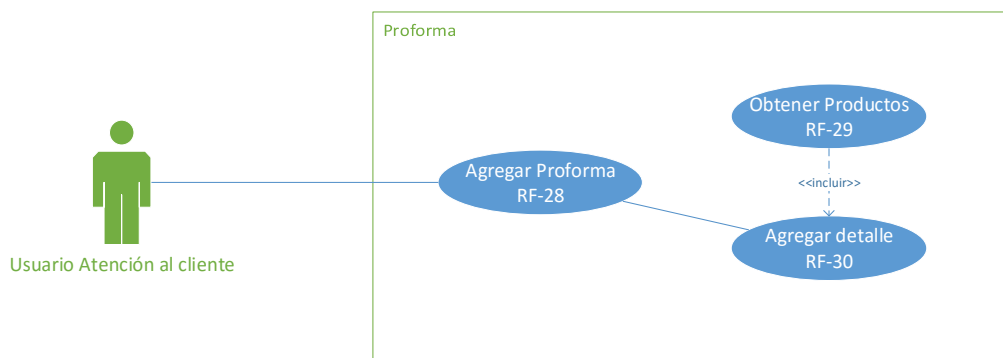


Figura 36. Casos de uso proforma

3.1.10.6 Especificaciones Casos de uso

Tabla 56.
Especificación de casos de uso cerrar proyecto

ID:	RF-26
Descripción:	Agregar pago
Precondición:	Estar autenticado en el sistema como Atención al cliente

(continúa)

	1	Cargar los datos de abonos o pagos ya realizados
	2	Al agregar un nuevo pago se deberá ingresar la forma de pago, el valor y una referencia del pago como voucher, nro cheque, banco, o efectivo.
	3	Al agregar el pago este deberá actualizar el valor total de pago registrado
	4	Se podrá emitir un recibo por el abono realizado en formato PDF.
	5	Los registro no se podrán eliminar de forma lógica
Flujos		Ninguno
Alternos:		
Excepciones:		
	1	Ninguna
Notas:		
	1	Ninguna

Tabla 57.
Especificación de casos de uso listado de pagos

ID:	RF-27
Descripción:	Listado pagos/abono
Precondición:	Estar autenticado en el sistema
	1 Se obtendrá de la base de datos todos los registros de pago referentes a un proyecto
	2 Cargar en la página mediante una tabla ordenada por orden de creación
	3 Al final de la lista indicar el valor total de abonos realizados
Flujos	Ninguno
Alternos:	
Excepciones:	(continúa)

1	Ninguna
<i>Notas:</i>	
1	Ninguna

Tabla 58.
Especificación de casos de uso listado de pagos

ID:	RF-28
Descripción:	Agregar proforma
Precondición:	Estar autenticado en el sistema
1	Desde la página de pre diseño se deberá mostrar un botón con la acción de crear proforma
2	Al crear la proforma se registraran los datos del usuario que crea la proforma así como datos de fecha y hora
3	Se registraran los datos de totales, descuentos e impuestos en el encabezado de la proforma
4	Es posible generar un archivo PDF con la proforma y este obtendrá los datos registrados.
Flujos	Ninguno
Alternos:	
Excepciones:	
1	Ninguna
<i>Notas:</i>	
1	Ninguna

Tabla 59.
Especificación de casos de uso listado de pagos

ID:	RF-29
Descripción:	Obtener productos
Precondición:	Estar autenticado en el sistema

(continúa)

	1	Generar un alista con los productos registrados en el sistema
	2	La lista estará compuesta por los productos activos con su identificador
	3	Se genera un campos de selección para su selección a anexo al detalle de la proforma
Flujos		Ninguno
Alternos:		
Excepciones:		
	1	Ninguna
Notas:		
	1	Ninguna

Tabla 60.
Especificación de casos de uso listado de pagos

ID:	RF-30
Descripción:	Agregar detalle proforma
Precondición:	Estar autenticado en el sistema
	1 Agregar un formulario compuesto por los productos disponibles de la lista que incluya valor unitario y cantidad
	2 Al agregar a la lista se deberá actualizar el total de la proforma así como los campos correspondientes a descuento e impuesto
	3 Los productos agregados podrán ser eliminados del detalle.
Flujos	Ninguno
Alternos:	
Excepciones:	
	1 Ninguna
Notas:	
	1 Ninguna

Historias del Usuario

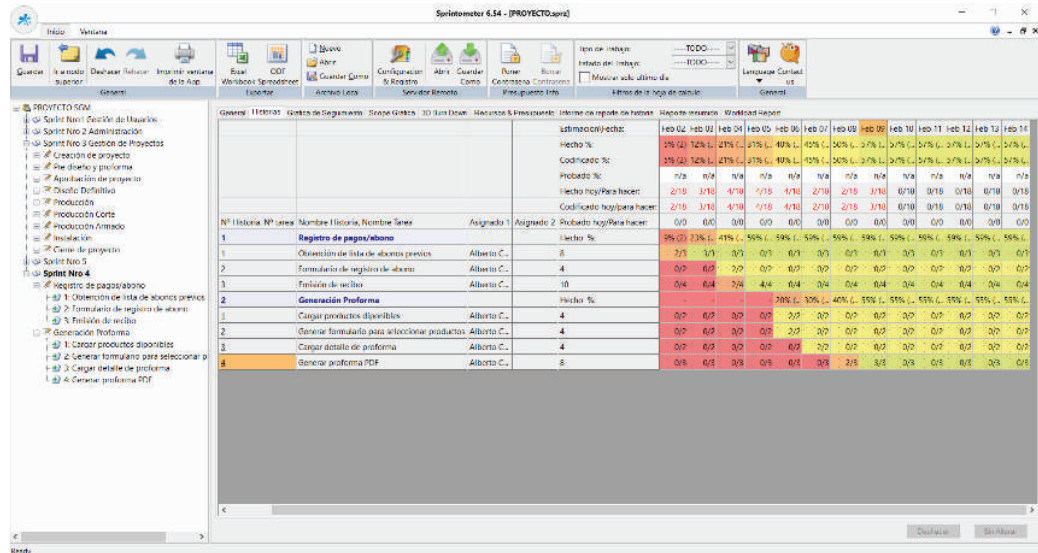


Figura 37. Sprint Nro.4 Historias del usuario

Gráfico de Seguimiento

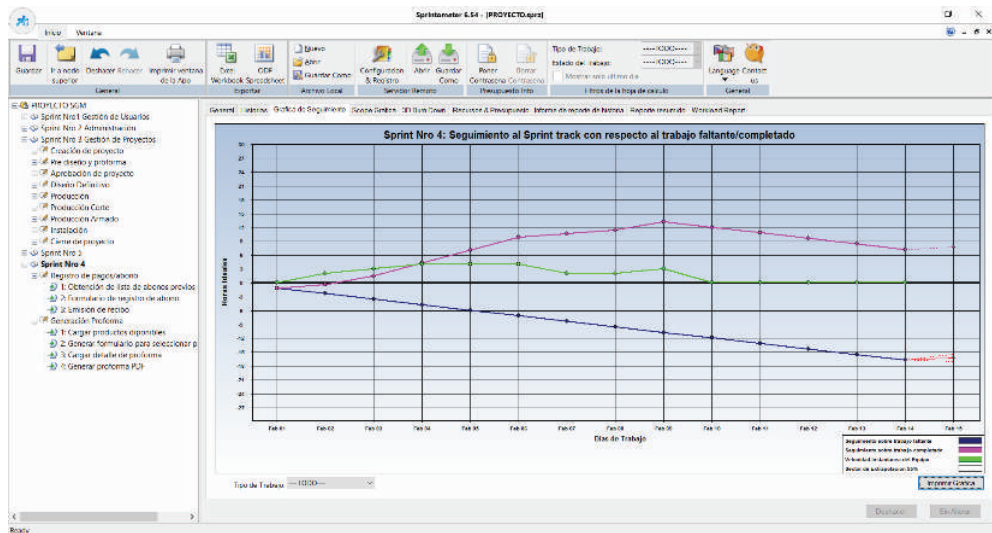


Figura 38. Sprint Nro.4 Seguimiento de Sprint Nro.4

3.1.11 Sprint Nro. 5 “Reportes”

3.1.11.1 Planificación

Se establece la necesidad de contar con los reportes necesarios que permitan evaluar el proceso de producción siendo importante contar con datos basados en tiempos de cada una de las etapas. Siendo ese el principal requerimiento se realiza la reunión del equipo de trabajo de donde las resoluciones se determinan en el ACTA-006-METALEC.

3.1.11.2 Requerimientos funcionales y no funcionales

Tabla 61.
Requerimientos funcionales reportes

Requerimientos Funcionales	
1	Los reportes deberán estar determinados por fecha de inicio y fin
2	Se reasentaran los datos en formato de unidad y porcentaje
3	Reporte de etapa de proyectos
4	Reporte de proyectos abiertos
5	Reporte de tiempos de producción proyectos
6	Reporte de tiempo de producción
7	Reporte individual de proyecto

3.1.11.3 Equipo de trabajo

Tabla 62.
Equipo de trabajo Sprint Nro. 5

<i>Rol</i>	<i>Persona</i>	<i>Área</i>
Producto Owner	Xavier Sandoval	Gerencia
SCRUM Master	Alberto Coba	Desarrollo
Team	Alberto Coba	Desarrollo
	Xavier Sandoval	Pruebas

3.1.11.4 Actores del Sistema

Los actores identificados para el primer sprint corresponden a los siguientes:

Tabla 63.
Actores del sistema Sprint Nro. 6

<i>Actor</i>	<i>Descripción</i>
Usuario	Es el usuario encargado de gestionar el de acuerdo a su rol asignado

Casos de uso

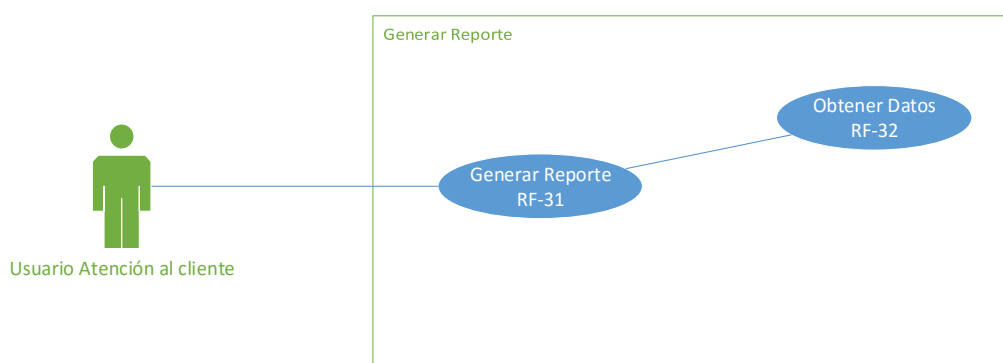


Figura 39. Casos de uso reportes

3.1.11.5 Especificaciones Casos de uso

Tabla 64.
Especificación de casos de uso Generar Reporte

ID:	RF-31
Descripción:	Generar Reporte
Precondición:	Estar autenticado en el sistema
	1 Determinar fechas para la generación del reporte
	2 En base al tipo de reporte se cargaran los datos requeridos
	3 Incluir la opción de impresión del reporte obtenido
Flujos	Ninguno
Alternos:	
Excepciones:	
	1 Ninguna

Tabla 65.
Especificación de casos de uso Generar Reporte

ID:	RF-32
Descripción:	Obtener datos reporte
Precondición:	Estar autenticado en el sistema
	1 En base a los parámetros generar las consultas a base de datos
	2 Devolver el arreglo con los resultados obtenidos
Flujos	Ninguno
Alternos:	
Excepciones:	
	1 Ninguna
Notas:	
	1 Ninguna

CAPÍTULO 4

IMPLEMENTACIÓN

4.1 Implementación

4.1.1 *Instalación de arquitectura definida*

El proyecto está basado en tecnología PHP y por sus características puede ser implementado bajo dos ambientes principales.

El primero mediante el uso de alojamiento pagado Hosting, donde se ofrecen varias opciones para alojar el sistema que cumple con la arquitectura requerida. La segunda opción es la preparación de un servidor especializado para la aplicación, siendo posible presentarlo como un servicio virtualizado a fin de que permita mantener un respaldo completo y su recuperación sea más eficiente en el caso de un fallo.

La virtualización ofrece un sin número de ventajas entre las cuales se establece la flexibilidad, seguridad, portabilidad como las principales.

Flexibilidad.

Una implementación usando virtualización brinda flexibilidad ya que los recursos asignados pueden variar de acuerdo a las necesidades del sistema mientras existan los recursos físicos. Esto permite optimizar el uso de hardware.

Seguridad.

Al ser un elemento virtualizado cuenta con su propia seguridad a nivel de sistema operativo, adicionalmente cada una de la máquinas virtuales tienen su propio sistema de seguridad que de ser vulnerado afectaría solo al servicio vulnerado.

Portabilidad.

La virtualización brinda una gran portabilidad, permitiendo que una imagen de la máquina virtual pueda ser restaurada sin mayores inconvenientes en otro hardware de ser necesario. Esto es fundamental para la respuesta ante fallo que puedan producirse.

4.1.2 Instalación de Servidor

Para la implementación se establece la necesidad de contar con un sistema operativo seguro por lo cual se consideró el uso de sistema operativo basado en Linux, para el caso puntual de esta implementación CentOS 7.

4.1.3 Instalación de sistema operativo

Seleccionar el idioma para la instalación, en este caso español.

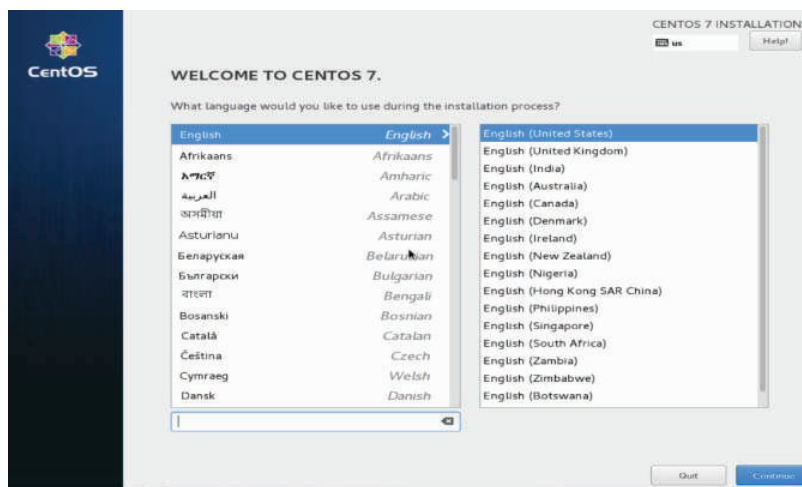


Figura 40. Instalación CentOS, selección de lenguaje

Es necesario especificar el particionamiento de disco duro, así como configuraciones generales necesarias:



Figura 41. Instalación CentOS, configuración de instalación

Se establecerá una contraseña de administrador la misma que deberá contar con la suficiente robustez que garanticen la seguridad.

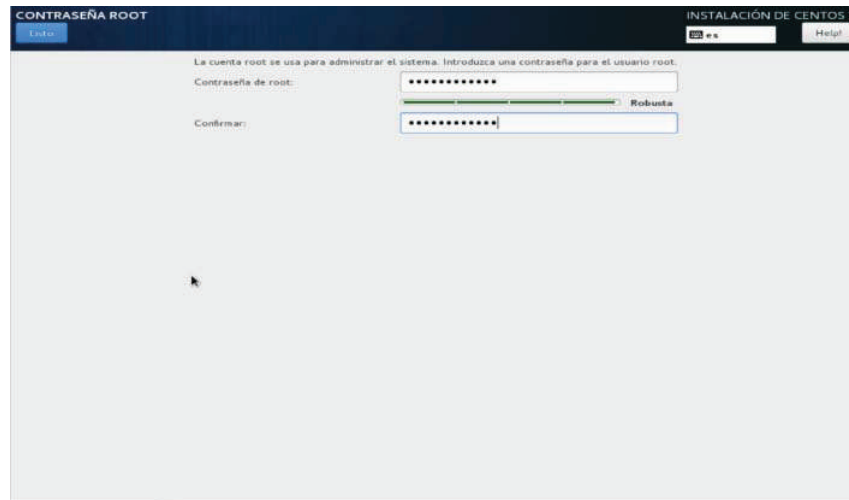


Figura 42. Instalación CentOS, configuración contraseña root

Establecida la contraseña se espera se concluya la instalación y se reinicia el equipo



Figura 43. Instalación CentOS, instalación finalizada

ESPACIO EN BLANCO INTENCIONAL

Ya iniciado el equipo se procede a arrancar con la instalación realizada para su configuración:

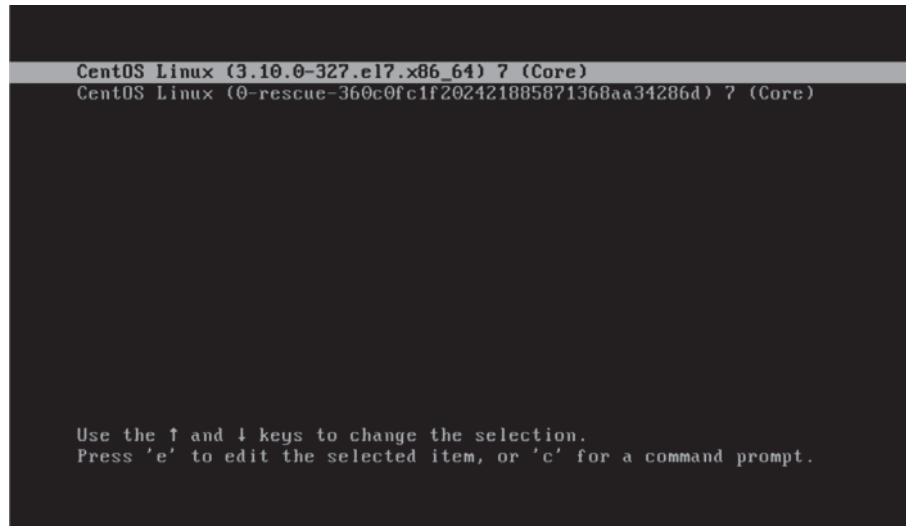


Figura 44. Arranque de sistema operativo CentOS

Se debe en primera instancia verificar el hostname, que es el nombre del servidor. Para modificarlo se puede usar e línea de comando la **nmtui**. Con ello se presenta una ventana para configuración rápida de hostname, y red de forma fácil.



Figura 45. Configuración de nombre de servidor y conexiones de red nmtui

ESPACIO EN BLANCO INTENCIONAL

Establecido el nombre del host, se acepta la configuración y se guardan los cambios.

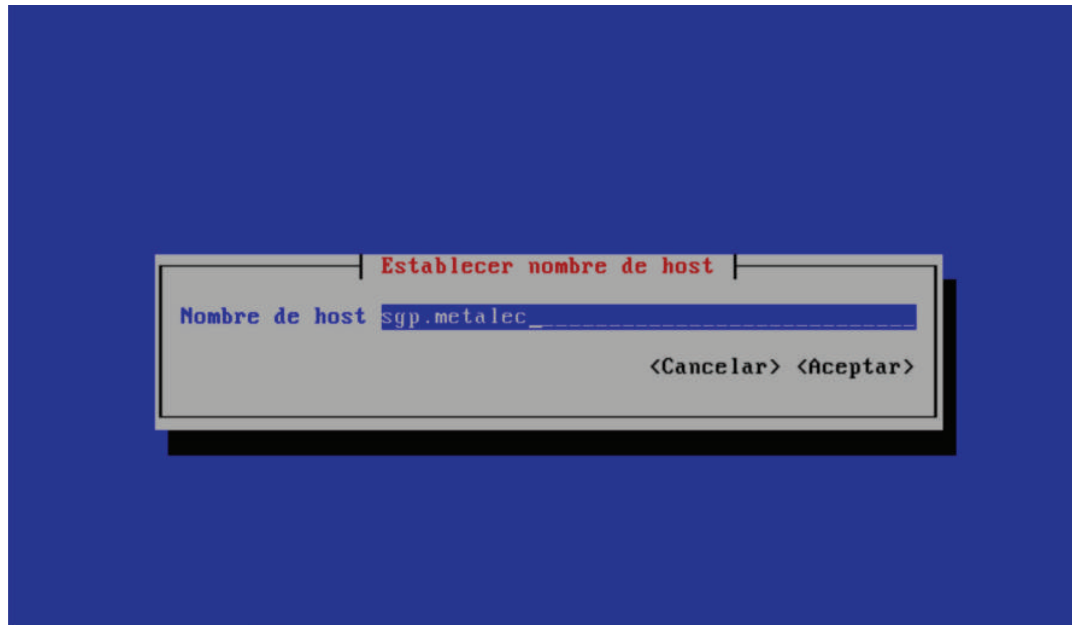


Figura 46. Cambio de nombre de host

Reiniciar el systemd-hostnamed

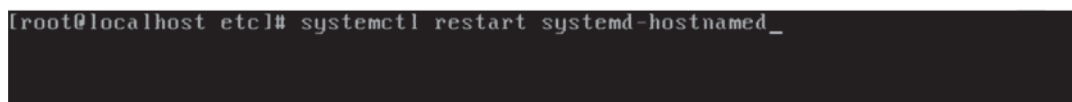


Figura 47. Reinicio de hostnamed

Reiniciado el servicio es recomendable reiniciar el sistema operativo para luego realizar una revisión de los datos de hostname verificando si fueron cambiados correctamente.

Para ello lo usar el comando hostnamectl status

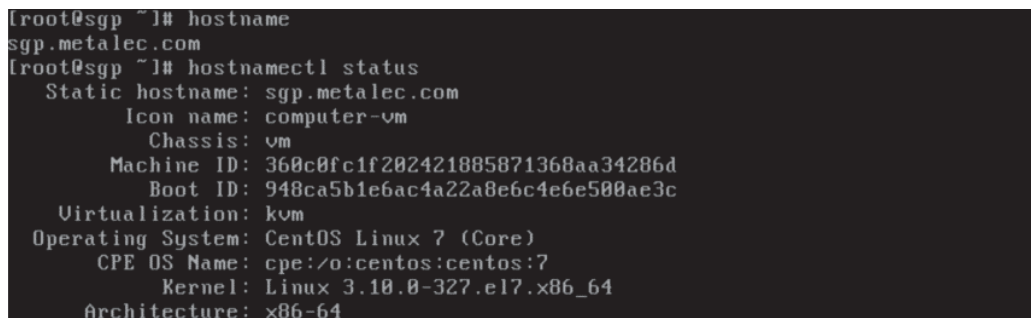


Figura 48. Verificación del estado de hostname

Para la instalación de paquetes y servicios necesarios para que el sistema operativo funcione de acuerdo a las necesidades es necesario contar con los repositorios adecuados. Toda instalación al ser una distribución CentOS de Linux principalmente será mediante el comando **yum**. Para contar con las versiones estables se realizará una actualización de los repositorios mediante el comando `yum update`.

```
[root@spp ~]# yum -y update
Complementos cargados:fastestmirror
base                               | 3.6 kB      00:00
extras                             | 3.4 kB      00:00
updates                           | 3.4 kB      00:00
(3/4): extras/7/x86_64/pri 2% [ 1 13 kB/s | 192 kB 10:10 ETA
```

Figura 49. Actualizando dependencias yum

Realizado esto se considera instalado el sistema operativo y listo para instalar los servicios requeridos.

4.1.3.1 Instalación de servidor web Apache

Una vez que se cuenta con el sistema operativo es el momento de instalar el servidor web que se encargará de contener nuestro sistema, para ello dentro del servidor se realiza lo siguiente:

En línea de comandos usando el comando `yum` se instalará el servicio `httpd` que corresponde al servidor Apache.

Para esta instalación se usa la versión estable del repositorio oficial que es la **httpd 2.4.6**.

```
#yum install httpd
```

ESPACIO EN BLANCO INTENCIONAL

```

Dependencias resueltas
=====
Package           Arquitectura  Versión           Repositorio  Tamaño
=====
Instalando:
httpd             x86_64       2.4.6-40.el7.centos  base         2.7 M
Instalando para las dependencias:
apr              x86_64       1.4.8-3.el7        base         103 k
apr-util         x86_64       1.5.2-6.el7        base         92 k
httpd-tools      x86_64       2.4.6-40.el7.centos  base         82 k
mailcap          noarch       2.1.41-2.el7        base         31 k

Resumen de la transacción
=====
Instalar 1 Paquete (+4 Paquetes dependientes)

Tamaño total de la descarga: 3.0 M
Tamaño instalado: 10 M
Is this ok [y/d/N]: _

```

Figura 50. Instalación de servicio httpd

Concluida la instalación se debe agregar, de ser necesario, al firewall los puertos 80 y 443. Se iniciara el servicio del Apache por lo cual es necesario usar el comando `systemctl`

```

[root@spp ~]# systemctl start httpd
[root@spp ~]# systemctl status httpd
■ httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese
t: disabled)
   Active: active (running) since lun 2016-02-15 19:01:28 CET; 7s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 11093 (httpd)
   Status: "Processing requests..."
    CGroup: /system.slice/httpd.service
           └─11093 /usr/sbin/httpd -DFOREGROUND
             └─11094 /usr/sbin/httpd -DFOREGROUND
               └─11095 /usr/sbin/httpd -DFOREGROUND
                 └─11096 /usr/sbin/httpd -DFOREGROUND
                   └─11097 /usr/sbin/httpd -DFOREGROUND
                     └─11098 /usr/sbin/httpd -DFOREGROUND

feb 15 19:01:27 spp.metalec.com systemd[1]: Starting The Apache HTTP Server...
feb 15 19:01:28 spp.metalec.com httpd[11093]: AH00557: httpd: apr_sockaddr_i...m
feb 15 19:01:28 spp.metalec.com httpd[11093]: AH00558: httpd: Could not reli...e
feb 15 19:01:28 spp.metalec.com systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[root@spp ~]# _

```

Figura 51. Estado del servicio httpd

Se establece que el servicio inicie cuando se encienda el servidor. Para ello se utiliza el comando:

```
#systemctl enable httpd
```

Habilitado el servicio, iniciará automáticamente cada que inicie el sistema operativo. Con el servicio arriba el servidor web comienza a escuchar las peticiones por el puerto por defecto que es el 80. En el caso de usar certificado de seguridad el puerto sería el 443.

Se puede en este momento verificar que ya esté disponible el servicio httpd accediendo mediante un explorador a la dirección del servidor web.

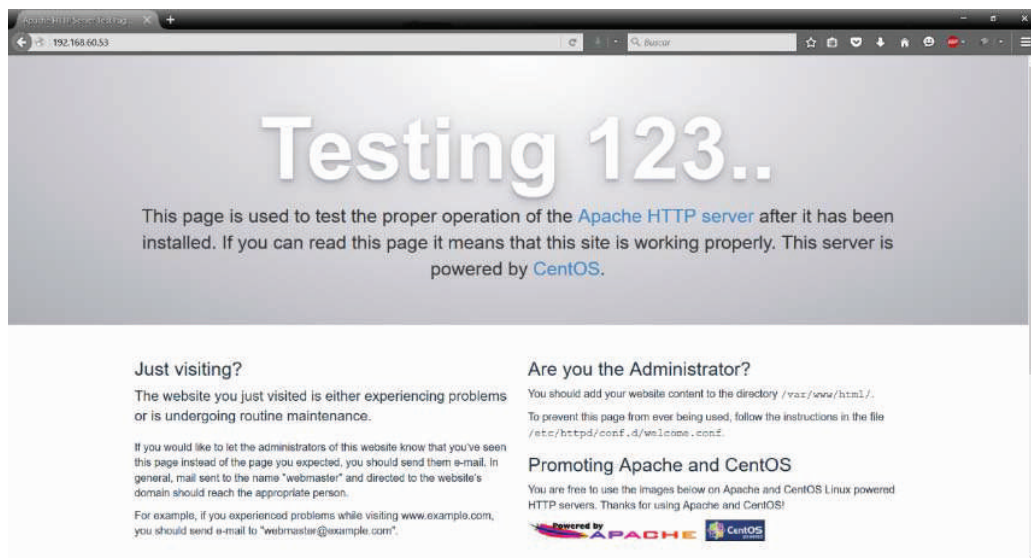


Figura 52. Acceso a servidor web mediante explorador

4.1.3.2 Instalación de PHP

Al ser un sistema desarrollado con PHP es necesario la instalación de PHP a fin de que su interprete permita el funcionamiento del sistema, para lo cual se usa el comando yum.

Primero se listan los paquetes disponibles usando:

```
#yum available | grep php
```

Seleccionar los módulos requeridos, en este caso:

```
mysql, pdo, gd, common, mbstring, xml
```

Posteriormente instalar PHP usando:

```
#yum install php php-mysql php-pdo php-gd php-common php.mbstring php-xml
```

```

Comprobando : 2:libpng-1.5.13-7.el7_2.x86_64 11/18
Comprobando : php-pdo-5.4.16-36.el7_1.x86_64 12/18
Comprobando : php-gd-5.4.16-36.el7_1.x86_64 13/18
Comprobando : php-xml-5.4.16-36.el7_1.x86_64 14/18
Comprobando : php-mysql-5.4.16-36.el7_1.x86_64 15/18
Comprobando : libX11-1.6.3-2.el7.x86_64 16/18
Comprobando : libXau-1.0.8-2.1.el7.x86_64 17/18
Comprobando : php-common-5.4.16-36.el7_1.x86_64 18/18

Instalado:
php.x86_64 0:5.4.16-36.el7_1 php-common.x86_64 0:5.4.16-36.el7_1
php-gd.x86_64 0:5.4.16-36.el7_1 php-mbstring.x86_64 0:5.4.16-36.el7_1
php-mysql.x86_64 0:5.4.16-36.el7_1 php-pdo.x86_64 0:5.4.16-36.el7_1
php-xml.x86_64 0:5.4.16-36.el7_1

Dependencia(s) instalada(s):
libX11.x86_64 0:1.6.3-2.el7 libX11-common.noarch 0:1.6.3-2.el7
libXau.x86_64 0:1.0.8-2.1.el7 libXpm.x86_64 0:3.5.11-3.el7
libjpeg-turbo.x86_64 0:1.2.90-5.el7 libpng.x86_64 2:1.5.13-7.el7_2
libxcb.x86_64 0:1.11-4.el7 libxslt.x86_64 0:1.1.28-5.el7
libzip.x86_64 0:0.18.1-8.el7 php-cli.x86_64 0:5.4.16-36.el7_1
t1lib.x86_64 0:5.1.2-14.el7

¡Listo!
[root@sqq ~]#

```

Figura 53. Instalación de paquete PHP

Finalizada la instalación es necesario comprobar la instalación de todos los módulos mediante una página php con el script `phpinfo()`.

```

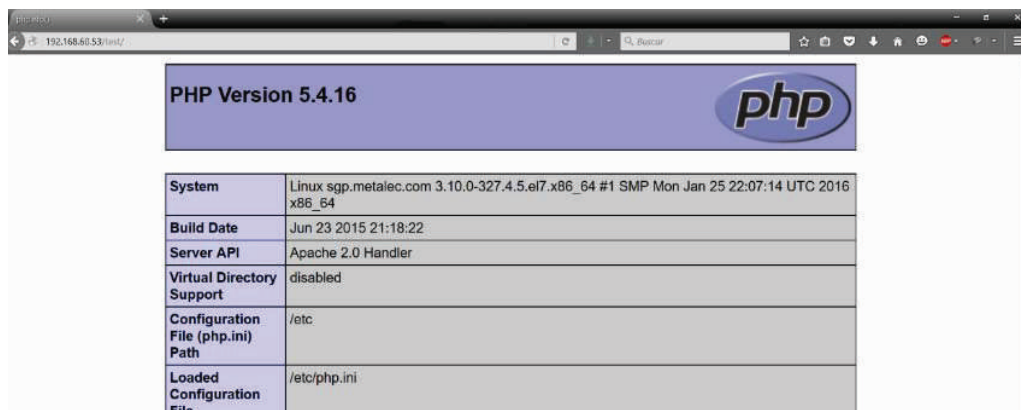
GNU nano 2.3.1 Fichero: index.php
<?php
phpinfo();
?>

```

Figura 54. Archivo de prueba phpinfo

Al acceder al archivo de prueba creado se presenta las características de PHP que se están ejecutando en este comentario como versiones, módulos disponibles.

ESPACIO EN BLANCO INTENCIONAL



PHP Version 5.4.16	
System	Linux sgp.metalec.com 3.10.0-327.4.5.el7.x86_64 #1 SMP Mon Jan 25 22:07:14 UTC 2016 x86_64
Build Date	Jun 23 2015 21:18:22
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini

Figura 55. Página de phpinfo

4.1.3.3 Instalación servidor de base de datos MariaDB/MySQL

El servidor MariaDB es el reemplazo compatible que se presenta en la versión de CentOS 7 a la versión de MySQL, siendo su compatibilidad una gran ayuda para prácticamente sea transparente su uso.

Para su instalación se hace uso del comando yum:

```
#yum install mariadb mariadb-server
```

```
perl-Scalar-List-Utils.x86_64 0:1.27-248.e17
perl-Socket.x86_64 0:2.010-3.e17
perl-Storable.x86_64 0:2.45-3.e17
perl-Text-ParseWords.noarch 0:3.29-4.e17
perl-Time-HiRes.x86_64 4:1.9725-3.e17
perl-Time-Local.noarch 0:1.2300-2.e17
perl-constant.noarch 0:1.27-2.e17
perl-libs.x86_64 4:5.16.3-286.e17
perl-macros.x86_64 4:5.16.3-286.e17
perl-parent.noarch 1:0.225-244.e17
perl-podlators.noarch 0:2.5.1-3.e17
perl-threads.x86_64 0:1.87-4.e17
perl-threads-shared.x86_64 0:1.43-6.e17

¡Listo!
[root@sgp ~]# systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service
to /usr/lib/systemd/system/mariadb.service.
[root@sgp ~]#
```

Figura 56. Instalación de MariaDB y configuración de inicio

Una vez instalado se establece que se ejecute el servicio al iniciar el sistema operativo. Ahora es necesario la nueva instalación del servidor de base de datos para ello se debe iniciar el servicio y verificar que se encuentra activo:

```
#systemctl start mariadb
```

```
#systemctl status mariadb
```

```

■ mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since lun 2016-02-15 19:27:03 CET; 7s ago
     Process: 18153 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
     Process: 18070 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
    Main PID: 18152 (mysqld_safe)
      CGroup: /system.slice/mariadb.service
             └─18152 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
                └─18309 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql...

feb 15 19:27:01 sgp.metalec.com mariadb-prepare-db-dir[18070]: The latest inf...
feb 15 19:27:01 sgp.metalec.com mariadb-prepare-db-dir[18070]: You can find a...
feb 15 19:27:01 sgp.metalec.com mariadb-prepare-db-dir[18070]: http://dev.mys...
feb 15 19:27:01 sgp.metalec.com mariadb-prepare-db-dir[18070]: Support MariaD...
feb 15 19:27:01 sgp.metalec.com mariadb-prepare-db-dir[18070]: Corporation Ab...
feb 15 19:27:01 sgp.metalec.com mariadb-prepare-db-dir[18070]: Alternatively ...
feb 15 19:27:01 sgp.metalec.com mariadb-prepare-db-dir[18070]: http://mariadb...
feb 15 19:27:01 sgp.metalec.com mysqld_safe[18152]: 160215 19:27:01 mysqld_sa...
feb 15 19:27:01 sgp.metalec.com mysqld_safe[18152]: 160215 19:27:01 mysqld_sa...
feb 15 19:27:03 sgp.metalec.com systemd[1]: Started MariaDB database server.
Hint: Some lines were ellipsized, use -l to show in full.
[root@sgp ~]# _

```

Figura 57. Estado del servicio de MariaDB

Con el servicio disponible se realiza la instalación segura, misma que permite la configuración de clave de root para el servidor de base de datos, así como un conjunto de configuraciones que buscan garantizar una instalación segura del sistema.

```

[root@sgp ~]# mysql_secure_installation
/bin/mysql_secure_installation: línea 379: find_mysql_client: no se encontró la orden

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none): _

```

Figura 58. Instalación segura de servidor de base de datos

Se establece como password provicional: espe2016

4.1.3.4 Instalación de PhpMyAdmin

Este paquete permite una fácil administración de la base de datos de forma remota y con una configuración segura, para lo cual se realiza las siguientes acciones:

Agregar el repositorio EPEL que permite acceder a los paquetes correspondientes a phpMyAdmin.

```
[root@sgp ~]# rpm -iUvh http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
Recuperando http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
advertencia:/var/tmp/rpm-tmp.NQtSpJ: EncabezadoU3 RSA/SHA256 Signature, ID de clave 352c64e5: NOKEY
Preparando... ##### [100%]
Actualizando / instalando...
 1:epel-release-7-5 ##### [100%]
[root@sgp ~]# _
```

Figura 59. Agregar el repositorio EPEL

Ya instalado el repositorio, es necesario actualizar la base de yum a fin de que muestre los paquetes disponibles en este nuevo repositorio, de esta manera se podrá instalar el PhpMyAdmin. Para actualizar nuevamente se utilizará el comando yum update.

```
[root@sgp ~]# rpm -iUvh http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
Recuperando http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
advertencia:/var/tmp/rpm-tmp.NQtSpJ: EncabezadoU3 RSA/SHA256 Signature, ID de clave 352c64e5: NOKEY
Preparando... ##### [100%]
Actualizando / instalando...
 1:epel-release-7-5 ##### [100%]
[root@sgp ~]# yum -y update
Complementos cargados:fastestmirror
epel/x86_64/metalink | 2.3 kB 00:00
epel | 4.3 kB 00:00
(1/3): epel/x86_64/group_gz | 169 kB 00:00
(2/3): epel/x86_64/updateinfo | 485 kB 00:01
(3/3): epel/x86_64/primary_db | 3.8 MB 00:02
Loading mirror speeds from cached hostfile
 * base: mirror.cedia.org.ec
 * epel: epel.gtdinternet.com
 * extras: mirror.nbtelecom.com.br
 * updates: centos.brnet.net.br
No packages marked for update
[root@sgp ~]# _
```

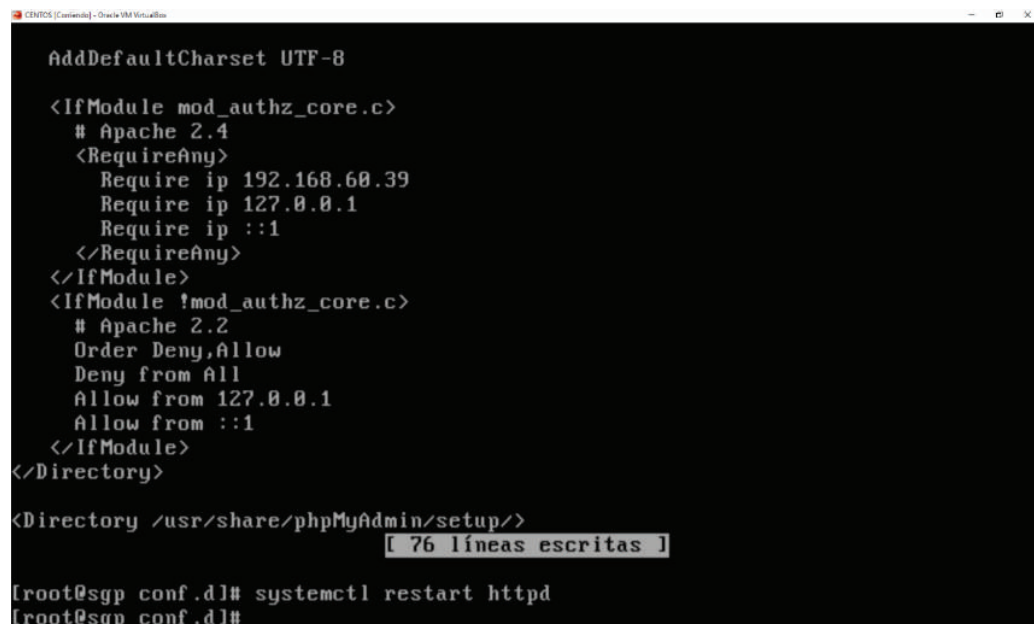
Figura 60. Actualización de bases de yum con el repositorio EPEL

Actualizado se procede a instalar phpMyAdmin en este caso la versión 4.4.15 usando el comando `yum install phpmyadmin`. Una vez instalado se debe configurar el acceso para hay que dirigirse a la siguiente ubicación:

```
#cd /etc/httpd/conf.d
```

Este sitio cuenta con la configuración de virtual host de apache así como de acceso para phpMyAdmin.

Se deberá configurar el acceso en el archivo `phpMyAdmin.conf`. Configurado se reinicia el servicio de Apache y PhpMyAdmin ya estará disponible desde el equipo que se haya establecido.



```

AddDefaultCharset UTF-8

<IfModule mod_authz_core.c>
    # Apache 2.4
    <RequireAny>
        Require ip 192.168.68.39
        Require ip 127.0.0.1
        Require ip ::1
    </RequireAny>
</IfModule>
<IfModule !mod_authz_core.c>
    # Apache 2.2
    Order Deny,Allow
    Deny from All
    Allow from 127.0.0.1
    Allow from ::1
</IfModule>
</Directory>

<Directory /usr/share/phpMyAdmin/setup/>
    [ 76 líneas escritas ]

[root@sqp conf.d]# systemctl restart httpd
[root@sqp conf.d]#

```

Figura 61. Configuración de acceso para host al phpMyAdmin

Configurado el acceso a PhpMyAdmin es necesario reiniciar el servicio `httpd` lo cual permitirá de acuerdo a los permisos establecidos acceder al servidor de base de datos mediante phpMyAdmin.

ESPACIO EN BLANCO INTENCIONAL

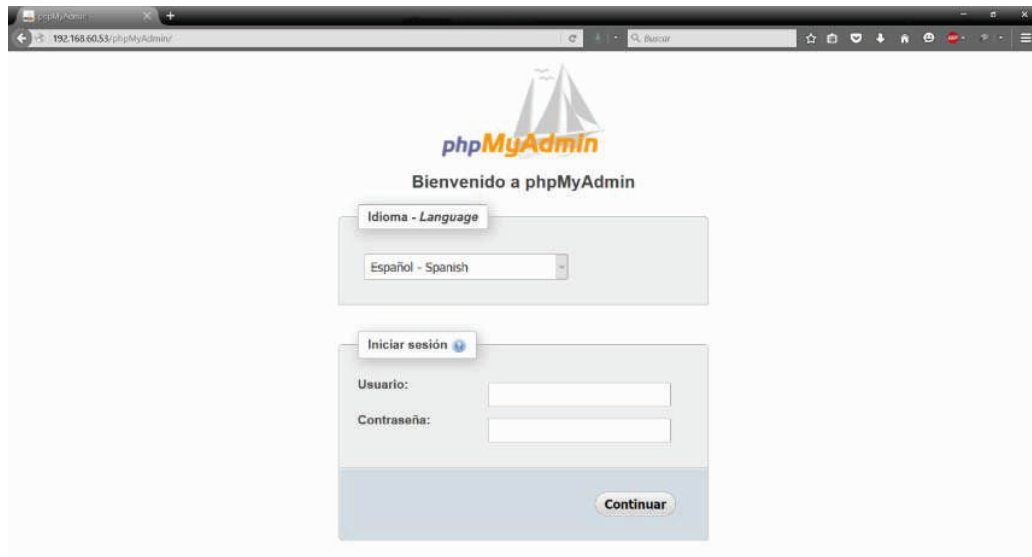


Figura 62. Acceso al sistema phpMyAdmin

Una vez que presenta la pantalla de autenticación se debe ingresar con las contraseñas establecidas en la instalación, y de esta manera acceder a las funcionalidades de esta herramienta:

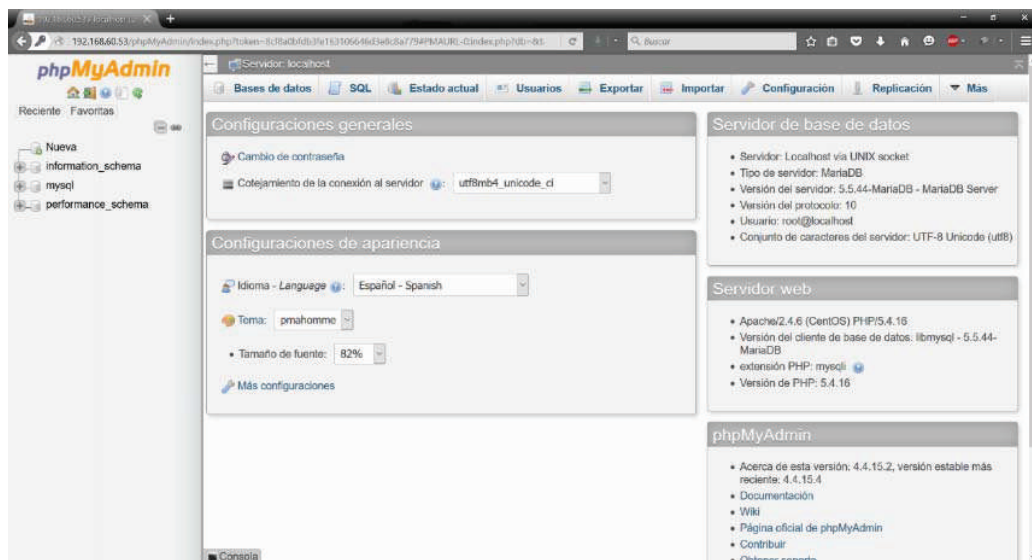


Figura 63. phpMyAdmin con acceso al servidor MariaDB

ESPACIO EN BLANCO INTENCIONAL

4.1.3.5 Configuraciones Adicionales

Cambio de Zona Horaria:

Para configurar correctamente la zona horaria se debe establecer la misma en el archivo de configuración de PHP denominado php.ini, el mismo que se encuentra en el directorio /etc.

```

;;;;;;;;;;;;;;;;;;;;
; Module Settings ;
;;;;;;;;;;;;;;;;;;;;

[CLI Server]
; Whether the CLI web server uses ANSI color coding in its terminal output.
cli_server.color = On

[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = America/Guayaquil

```

Figura 64. Configuración de zona horaria php.ini

Tamaño de carga de archivos al servidor

Es importante definir el límite que se tendrá para la carga de archivos adjuntos al servidor, por ello se configura el límite, que deberá ser coherente con el configurado en la aplicación.

```

; http://php.net/file-uploads
file_uploads = On

; Temporary directory for HTTP uploaded files (will use system default if not
; specified).
; http://php.net/upload-tmp-dir
upload_tmp_dir =

; Maximum allowed size for uploaded files.
; http://php.net/upload-max-filesize
upload_max_filesize = 10M

; Maximum number of files that can be uploaded via a single request
max_file_uploads = 20

;;;;;;;;;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;;;;;;;;;

; Whether to allow the treatment of URLs (like http:// or ftp://) as files.

```

[^]G Ver ayuda [^]O Guardar [^]R Leer Fich [^]Y Pág Ant [^]K CortarTxt [^]C Pos actual
[^]X Salir [^]J Justificar [^]W Buscar [^]U Pág Sig [^]U PegarTxt [^]T Ortografía

Figura 65. Configuración de tamaño de archivos para carga

4.1.4 Implementación

4.1.4.1 Creación de base de datos

La creación de la base de datos se realiza mediante el script generado durante los Sprints de desarrollo, mismo que cuenta con la información básica para que el sistema funcione, para ello se crea la base de datos y se usa el cotejamiento utf8_general_ci.

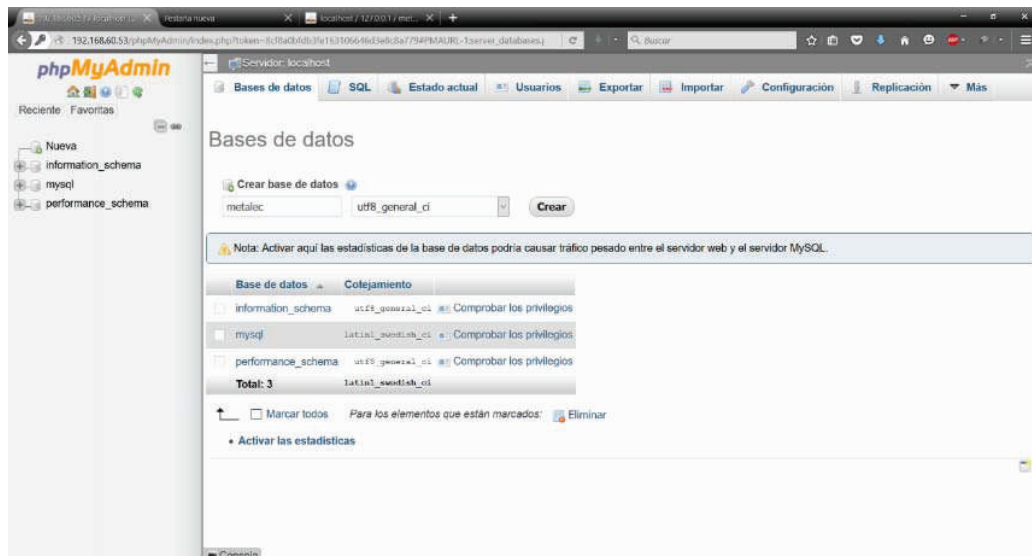


Figura 66. Creación de base de datos

Una vez creada la base de datos se debe crear un usuario específico para uso del sistema para ello se crea el usuario mismo que solo tendrá acceso a la base de datos del sistema.

ESPACIO EN BLANCO INTENCIONAL

4.1.4.2 Creación de usuario para base de datos

Para este caso se generará una clave segura a fin de que el sistema se autentique con la base de datos y únicamente lo pueda realizar el mismo servidor y no otros equipos.

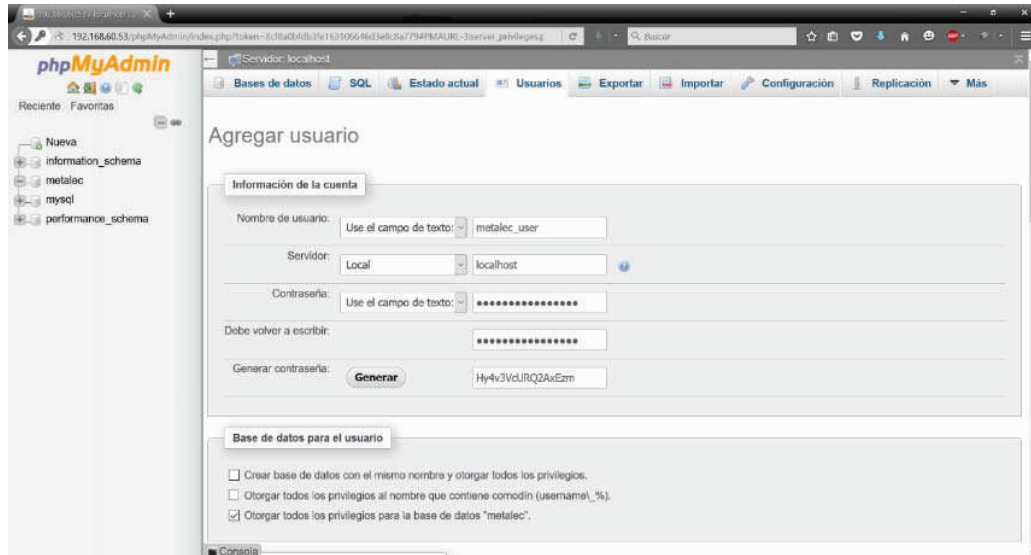


Figura 67. Creación de usuario para base de datos

Ya disponible la base de datos se deberá realizar la carga de script de la base de datos

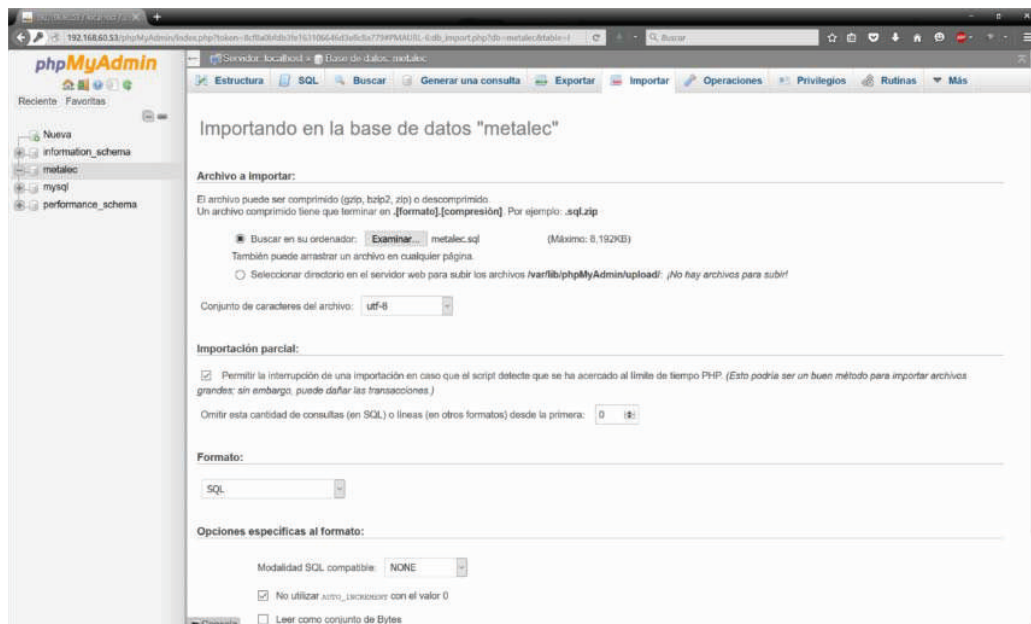


Figura 68. Carga de script de base de datos

Al finalizar la importación de la base de datos se podrá comprobar que la estructura se ha creado y al momento está lista para su funcionamiento.

The screenshot shows the phpMyAdmin interface with the 'metalec' database selected. The 'Estructura' (Structure) tab is active, displaying a list of tables. The table list includes columns for 'Tabla', 'Acción', 'Filas', 'Tipo', 'Cotejamiento', 'Tamaño', and 'Residuo a depurar'. The tables listed are: abono, adjunto, artefacto, artefacto_proyecto, cliente, contrato, detalle_factura, detalle_proforma, estado_obra, estado_obra_proyecto, etapa, etapa_proyecto, factura, log, plantilla, proceso, producto, proforma, proyecto, rol, rol_usuario, and usuario.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
abono	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	22 KB	-
adjunto	Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8_general_ci	32 KB	-
artefacto	Examinar Estructura Buscar Insertar Vaciar Eliminar	10	InnoDB	utf8_general_ci	14 KB	-
artefacto_proyecto	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8_general_ci	11 KB	-
cliente	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	1 KB	-
contrato	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	81 KB	-
detalle_factura	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	45 KB	-
detalle_proforma	Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8_general_ci	41 KB	-
estado_obra	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	14 KB	-
estado_obra_proyecto	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8_general_ci	41 KB	-
etapa	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_general_ci	45 KB	-
etapa_proyecto	Examinar Estructura Buscar Insertar Vaciar Eliminar	14	InnoDB	utf8_general_ci	41 KB	-
factura	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	32 KB	-
log	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	32 KB	-
plantilla	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	45 KB	-
proceso	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	14 KB	-
producto	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_general_ci	14 KB	-
proforma	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	32 KB	-
proyecto	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_general_ci	14 KB	-
rol	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	14 KB	-
rol_usuario	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	45 KB	-
usuario	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	45 KB	-

Figura 69. Estructura de base de datos creada en el servidor

La aplicación deberá ser copiada al directorio por defecto html ubicado en el directorio /var/www/html en el cual se debe acceder al archivo de configuración de los parámetros de conexión a base de datos, en el cual se parametriza los datos conforme se han creado en la implementación:

```

/var/www/sgp/Connections/conn.php - root@192.168.60.53 - Editor - Win...
Encoding • Color •
<?php
# FileName="Connection_php_mysql.htm"
# Type="MYSQL"
# HTTP="true"
$hostname_conn = "localhost";
$databse_conn = "metalec";
$username_conn = "metalec_user";
$password_conn = "Hy4v3VcURQ2AxEzm";
$conn = mysql_pconnect($hostname_conn, $username_conn, $password_conn)
mysql_query("SET NAMES 'utf8'");
mysql_select_db($databse_conn, $conn);
?>
Line: 8/12 Character: 34 (0x22) Encoding: 1252 (ANSI - Laf)

```

Figura 70. Configuración de parámetros de base de datos

Guardados todos los cambios, el sistema está listo para su utilización en el servidor configurado, a fin de probar que todo funcione correctamente se accede mediante un explorador al sistema:

La primera pantalla indica que el servidor httpd está respondiendo a las peticiones web, sin embargo no hace hasta el momento ninguna conexión a base de datos.



Figura 71. Pantalla de inicio, comprobación de funcionamiento

Para la comprobación que la conexión a la base de datos se realiza el proceso de autenticación mismo que de ser correcto, llevará a la página principal del sistema, caso contrario indicaría un error.

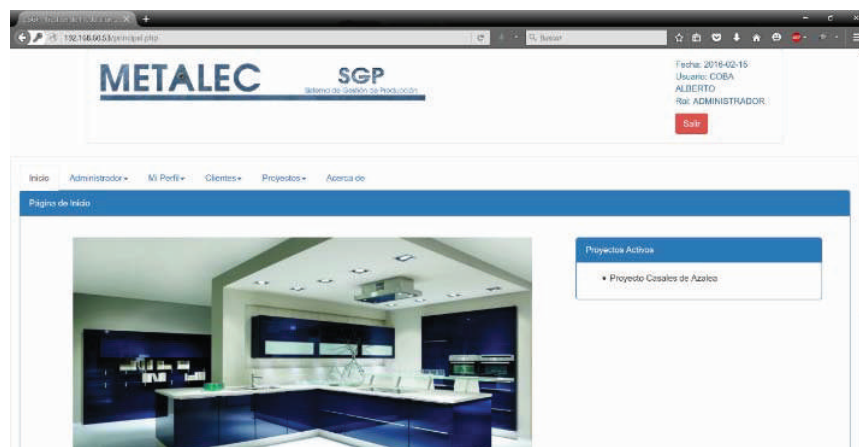


Figura 72. Página principal del sistema

Una vez que se ha realizado la comprobación se determina que todos los servicios están respondiendo correctamente y la aplicación al momento se encuentra implementada.

4.1.5 Resultados

La implementación del proyecto evidencio varios resultados mismos que han permitido potenciar el trabajo del talento humano de la empresa, optimizar el uso de recursos y disminuir el riesgo, centrando sus resultados en áreas específicas como lo son organización, seguimiento y costos visto desde el punto de vista del proceso.

4.2 Organización

Cada uno de los usuarios del sistema tiene acceso a una Bandeja de entrada (Figura 73) la misma que permite de primera mano conocer cuáles son sus responsabilidades pendientes dentro del proceso, lo cual se replica para cada uno de los roles que intervienen y que ha derivado en un menor tiempo de respuesta entre los diferentes actores.

PROYECTO	CLIENTE	ETAPA	MENSAJE	ASIGNADO	ACCION
LAS ENRIE-DADI-BAS NRG2	Hacer Ceros	DIS-NO (H-IND)	Producción	2016-02-04 22:17:41	ABRIR

Figura 73. Bandeja de entrada

De esta manera se registran todas las acciones realizadas dentro del proyecto por el o los usuarios responsables de cada etapa determinada, adicionalmente todos los proyectos puedan accederse mediante la opción de búsqueda (Figura 74) con lo cual se tiene acceso rápido a la documentación de cada proyecto. Estas opciones han permitido reducir en siete veces el tiempo de acceso a información de los proyectos.



Figura 74. Opciones sobre proyectos

Para analizar el grado de impacto se realizó una muestra de los tiempos promedio (Tabla 66) de cambio entre las etapas, donde se pudo evidenciar que con el proceso automatizado el tiempo disminuye sustancialmente y que muestra que se cumple con uno de los objetivos de optimizar el proceso de producción y mejorar los niveles de cumplimiento.

Tabla 66. Tiempos de cambio de etapa

Etapa del Proyecto	SIN SGP (min)	SGP (min)	Diferencia	Mejora
Asistente - Pre diseño	15	4	11	73,33 %
Contrato - Diseño	15	5	10	66,67 %
Diseño - Producción	15	3	12	80 %
Producción - Instalación	45	5	40	88,89 %
Instalación - Cierre	60	5	55	91,67 %
TOTALES	150	22	25,6	80,11 %

Los resultados obtenidos muestran que el trabajo entre cada uno de los roles se agiliza claramente (Figura 75), lo cual no solo se expresa en tiempos sino que permite una mejora subjetiva en la interacción entre el talento humano.

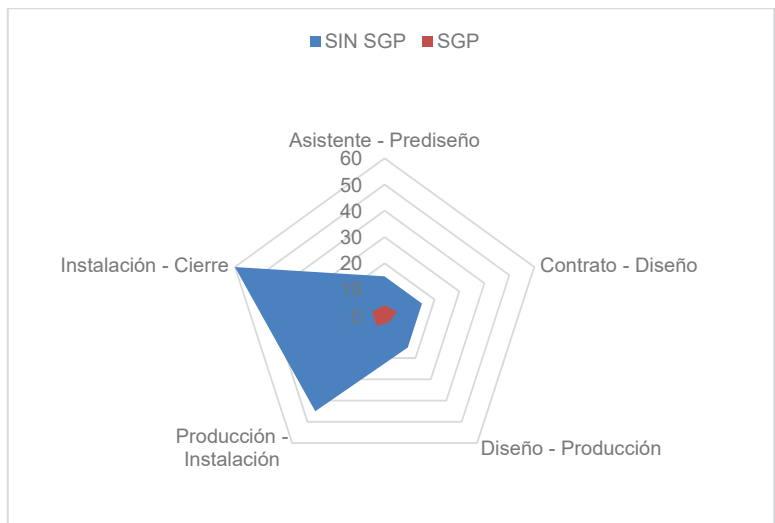


Figura 75. Tiempo de cambio de etapa

4.3 Seguimiento

La asignación de responsabilidades por etapa ha mostrado un mayor nivel de compromiso y cumplimiento de los trabajadores, lo que aporta a la eficiencia de todo el proceso empresarial, requerimiento fundamental para mantener la competitividad en el sector, lo cual es producto del seguimiento de cada una de las etapas (Figura 7).

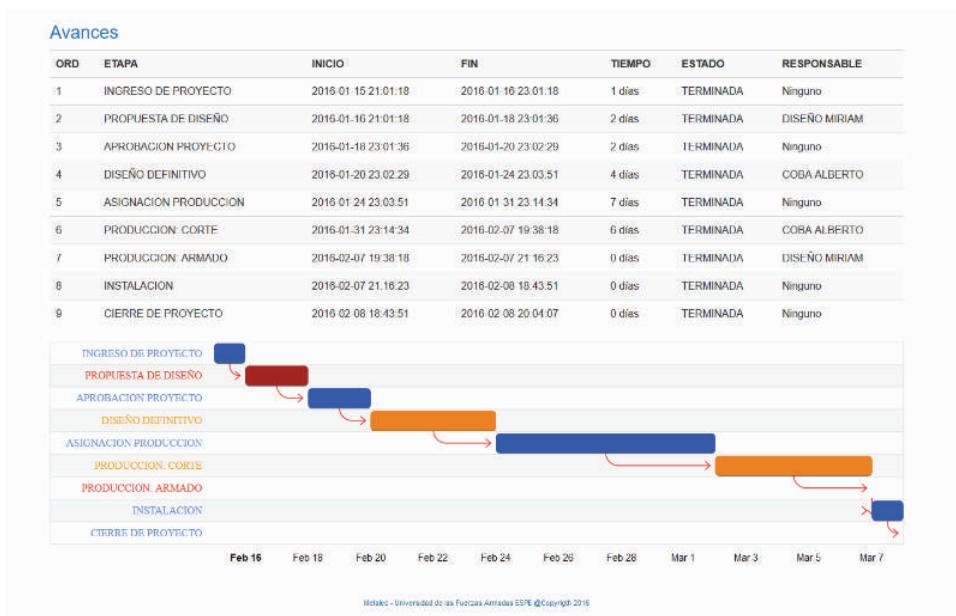


Figura 76. Seguimiento de etapas del proyecto

La posibilidad de control de tiempos permite a nivel gerencial determinar cuellos de botella así como puntos pendientes de gestión que le permitan optimizar el proceso.

4.4 Costos

Los costos directos de producción no han variado ya que reflejan principalmente la materia prima, remuneraciones entre otros rubros, sin embargo los tiempos de cambio de etapas (Figura 6) y el mayor compromiso de los trabajadores al visibilizar su trabajo generan ahorro que se traduce en un incremento de la capacidad de producción y disminuye el costo individual del producto manufacturado.

Del análisis de costos por error en los últimos 10 proyectos muestran que el sistema no ha influido en la prevención de la mayoría de los errores asociados a procedimientos manuales. A pesar de ello analizando los errores en características (materiales y acabado) su principal causa es la ausencia de información clara disponible. Con la implementación del sistema se evidencia (Figura 8) la disminución importante de este tipo de errores lo cual se asocia a un mejor acceso a información clara que está incidiendo en la calidad de los productos.

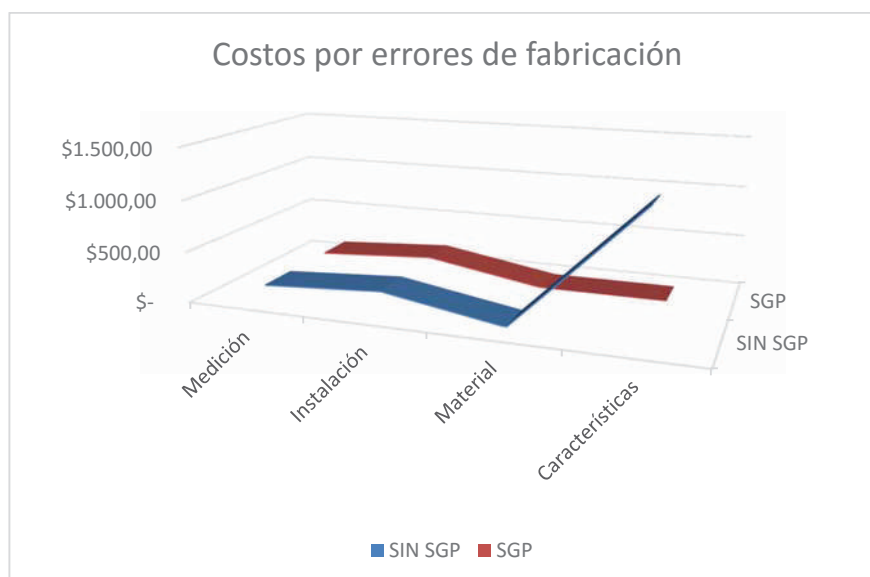


Figura 8. Costo por errores de producción

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La implementación del sistema de control ha permitido que los procesos de producción se mantengan con estados y etapas claros, garantizando la persistencia de la información y documentación de los proyectos de forma organizada y accesible de manera rápida.
- El uso de PHP ha permitido que el sistema sea flexible en relación a los recursos que las pequeñas y medianas empresas cuentan, permitiendo la optimización de los costos inherentes a hardware y software.
- La asignación de actividades organizadas por roles y responsable ha permitido llevar un control de las actividades pendientes mejorando la organización y optimizando los tiempos de atención en cada una de las etapas.
- El uso de tecnologías basadas en Javascript y CSS como JQuery y Bootstrap permiten una presentación y usabilidad del sistema sin incrementar tiempos de desarrollo o complejidad.
- La metodología SCRUM sin lugar a duda es la ideal para proyectos de desarrollo de software en este tipo de proyectos por sus características de la colaboración entre el equipo y el cliente, adaptabilidad a cambios y entregas funcionales que permite mejorar ampliamente el éxito del desarrollo.

5.2 Recomendaciones

- Continuar con el desarrollo que permita incluir nuevos procesos como compras, inventarios que permitan generar mayor información y control dentro de la empresa.
- Desarrollar servicios web así como aplicaciones móviles que permitan explotar con mayor amplitud el sistema desarrollado y así brindar mejores servicios que aporten a la competitividad dentro del sector.

- Mantener procesos de capacitación constante a los usuarios del sistema que permita garantizar su uso adecuado.

REFERENCIAS BIBLIOGRÁFICAS

Bibliografía

- Databasefriends. (22 de 02 de 2016). *Historia de MySQL*. Obtenido de <http://www.databasefriends.co/2014/02/history-of-mysql.html>
- Desarrollo Web. (22 de 02 de 2016). *Usabilidad y arquitectura de software*. Obtenido de <http://www.desarrolloweb.com/articulos/1622.php>
- Ministerio de Telecomunicaciones y de la Sociedad de la Información. (Enero de 2014). *Tecnologías de la información y comunicaciones para el desarrollo*. Obtenido de Ministerio de Industrias: <http://www.industrias.ec/archivos/CIG/file/CARTELERA/MINTEL-TIC%20para%20el%20Desarrollo.pdf>
- MYSQL. (22 de 02 de 2016). *MySQL Database*. Obtenido de <https://www.mysql.com/products/enterprise/mysql-datasheet.en.pdf>
- MySQL. (01 de 02 de 2016). *MySQL Documentation*. Obtenido de <http://dev.mysql.com/doc/>
- Netcraft. (22 de 02 de 2016). *Netcraft*. Obtenido de <http://goo.gl/zJU6aJ>
- PHP. (22 de 02 de 2016). *Sitio oficial PHP*. Obtenido de <http://php.net/manual/es/history.php.php>
- Universidad de Washington. (22 de 02 de 2016). *Curso de Ajax*. Obtenido de <https://courses.cs.washington.edu/courses/cse190m/11sp/lectures/slides/lecture18-ajax.shtml#slide5>
- W3C JQUERY. (22 de 02 de 2016). *Diseño JQuery*. Obtenido de http://www.w3schools.com/jquery/jquery_dimensions.asp
- W3C JQUERY. (22 de 02 de 2016). *Eventos JQuery*. Obtenido de http://www.w3schools.com/jquery/jquery_events.asp
- W3C JQuery. (22 de 02 de 2016). *W3C JQuery School*. Obtenido de <http://www.w3schools.com/jquery/default.asp>
- Web Facts. (22 de 02 de 2016). *Web Facts*. Obtenido de http://www.w3schools.com/jquery/jquery_dimensions.asp