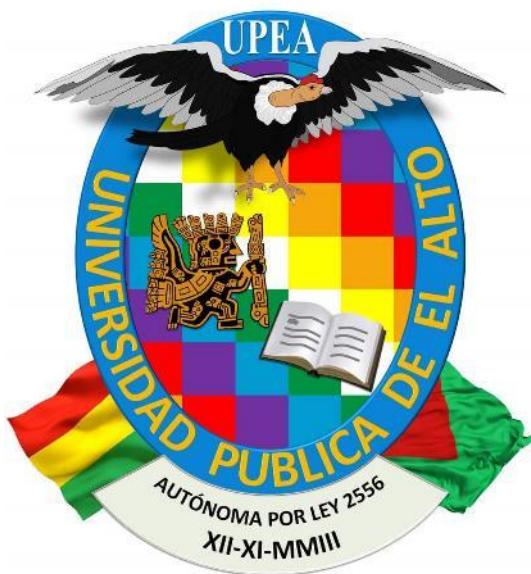


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA DE INFORMACIÓN PARA LA ADMINISTRACIÓN Y CONTROL DE VENTAS”

CASO: LINXS SRL

**Para Optar al Título de Licenciatura en Ingeniería de Sistemas
Mención: Informática y comunicaciones**

Postulante: Jimmy Yomar Mayta Jimenez

Tutor metodológico: Ing. Enrique Flores Baltazar

Tutor especialista: Lic. Mario Torrez Cupiticona

Tutor revisor: Lic. Jesus Juan Rocha Vera

El Alto - Bolivia

2020

DEDICATORIA

Dedico este trabajo a Dios quien siempre ha sido el autor de mi vida y el que guía mi destino, el mayor apoyo en tiempos difíciles.

A mi madre, quien fue mi mayor apoyo en tiempos de angustia, quiero honrar a mi padre, que hizo de todo para que la universidad fuera un sueño hecho realidad.

A mis tutores por guiarme e inculcarme sus conocimientos durante el desarrollo y culminación del presente proyecto.

AGRADECIMIENTOS

Quiero agradecer en primer lugar a Dios, por guiarme en el camino y fortalecerme espiritualmente para empezar un camino lleno de éxito.

Así, quiero mostrar mi gratitud a todas aquellas personas que estuvieron presentes en la realización de esta meta, de este sueño que es tan importante para mí, agradecer todas sus ayudas, sus palabras motivadoras, sus conocimientos, sus consejos y su dedicación.

Muestro mis más sinceros agradecimientos a mis tutores de proyecto, quien con sus conocimientos y sus guías fue una pieza clave para que pudiera desarrollar una sinfonía de hechos que fueron imprescindibles para cada etapa de desarrollo del trabajo.

A mis compañeros, quienes a través de tiempo fuimos fortaleciendo una amistad y creando una familia, muchas gracias por toda su colaboración, por convivir todo este tiempo conmigo, por compartir experiencias, alegrías, frustraciones, llantos, tristezas, peleas, celebraciones y múltiples factores que ayudaron a que hoy seamos como una familia, por aportarme confianza y por crecer juntos en este proyecto, muchas gracias.

Por último, quiero agradecer a la base de todo, a mi familia, en especial a mis padres, que quienes con sus consejos fueron el motor de arranque y mi constante motivación, muchas gracias por su paciencia y comprensión, y sobre todo por su amor.

¡Muchas gracias por todo!

RESUMEN

Las tecnologías de la información están transformando las actividades económicas y cotidianas. Indiscutiblemente los sistemas de información han invadido a todos y cada uno de los campos de la actividad humana: ciencia tecnología, arte, educación, recreación, administración y economía la tecnología relacionada con el manejo de la información hace que más personas y servicios estén conectados e interactúen entre si las tareas y funciones que tradicionalmente se realizaban manualmente ahora son hechas por medio de un sistema de información.

Linxs SRL es una empresa dedicada al desarrollo de aplicaciones de escritorio, páginas web, sistemas de reservas, material de publicidad, ventas de espacios publicitarios y motores, se encuentra en el mercado desde el año 2005 y ha ganado experiencia atendiendo a decenas de clientes en Bolivia, Estados Unidos y Puerto Rico. Ha diseñado páginas web, CD Multimedia y material de Merchandising para los más variados rubros y empresas, desde Hotelería y Turismo, hasta Industria, Banca y Gastronomía. Conocemos de calidad. Actualmente cuenta con alrededor de 7500 clientes en los nueve departamentos de Bolivia.

El presente proyecto tiene como objetivo desarrollar un sistema de información para la administración y control de ventas de manera que pueda ayudar en la administración y el control de ventas de la empresa Linxs utilizando metodología SCRUM que va enfocado al desarrollo exclusivo de sistemas de información.

Asimismo, se emplearon los factores de calidad (Métodos de evaluación de calidad). Para evaluar y medir la calidad del producto se aplicó la norma ISO 25000 y sus criterios de evaluación correspondientes.

Finalmente, los objetivos planteados en base a la problemática existente, han sido alcanzados de manera satisfactoria, de manera que se produjo un sistema de calidad que permite administrar y controlar las ventas a través del sistema.

Palabras clave: Sistema, información, administración, control y ventas.

SUMMARY

Information technologies are transforming economic and daily activities. Undeniably, information systems have invaded each and every one of the fields of human activity: science, technology, art, education, recreation, administration and economics. Technology related to information management means that more people and services are connected and interact with each other the tasks and functions that were traditionally performed manually are now done through an information system.

Linxs SRL is a company dedicated to the development of desktop applications, web pages, reservation systems, advertising material, sales of advertising space and engines, it has been in the market since 2005 and has gained experience serving dozens of clients in Bolivia, the United States and Puerto Rico. He has designed web pages, Multimedia CD and Merchandising material for the most varied areas and companies, from Hospitality and Tourism, to Industry, Banking and Gastronomy. We know quality. It currently has around 7,500 clients in the nine departments of Bolivia.

The objective of this project is to develop an information system for the administration and control of sales so that it can help in the administration and control of sales of the Linxs company using SCRUM methodology that is focused on the exclusive development of information systems.

Likewise, the quality factors (Quality evaluation methods) were used. To evaluate and measure the quality of the product, the ISO 25000 standard and its corresponding evaluation criteria were applied.

Finally, the objectives set based on the existing problems have been satisfactorily achieved, so that a quality system was produced that allows the administration and control of sales through the system.

Keywords: System, information, administration, control and sales.

ÍNDICE DE CONTENIDO

1. MARCO PRELIMINAR.....	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES	1
1.2.1. Antecedentes Institucionales	2
1.2.2. Antecedentes internacionales.....	3
1.2.3. Antecedentes nacionales.....	4
1.3. PLANTEAMIENTO DEL PROBLEMA	5
1.3.1. Problema principal	6
1.3.2. Problemas secundarios	6
1.4. OBJETIVOS	6
1.4.1. Objetivo general.....	6
1.4.2. Objetivos específicos.....	7
1.5. JUSTIFICACIÓN	7
1.5.1. Justificación técnica.....	7
1.5.2. Justificación económica.....	7
1.5.3. Justificación Social	8
1.6. METODOLOGÍA	8
1.6.1. Metodología de desarrollo	8
1.7. MÉTRICAS DE CALIDAD	8
1.8. HERRAMIENTAS.....	9
1.8.1. Lenguajes de Programación.....	9
1.8.2. Bases de datos	9
1.9. LÍMITES Y ALCANCES.....	10
1.9.1. Límite	10
1.9.2. Alcances	10
1.10. APORTES.....	10
1.10.1. Aporte practico.....	11
1.10.2. Aporte teórico	11
2. MARCO TEÓRICO	12
2.1. INTRODUCCIÓN	12

2.2.	SISTEMA	12
2.3.	INFORMACIÓN.....	12
2.4.	ADMINISTRACIÓN	12
2.5.	CONTROL DE VENTAS	13
2.6.	GESTIÓN DE VENTAS.....	13
2.7.	BUSINESS INTELLIGENCE	14
2.7.1.	Arquitectura del BI	15
2.7.2.	Desafíos en la implementación de BI	16
2.7.3.	Cuadrante de Gartner para herramientas de BI.....	17
2.8.	METODOLOGÍAS AGILES DE DESARROLLO DE SOFTWARE.....	18
2.8.1.	Metodología SCRUM.....	20
2.9.	LENGUAJES DE PROGRAMACIÓN WEB	25
2.9.1.	PHP	28
2.9.2.	JavaScript.....	30
2.10.	GESTIÓN DE CALIDAD DE SOFTWARE	32
2.10.1.	Normas ISO	33
2.10.2.	ISO/IEC 25000 (SQuaRE)	34
2.11.	ESTIMACIÓN DE COSTOS DE DESARROLLO DE SOFTWARE	35
2.11.1.	Método COSMIC	36
2.11.2.	Proceso COSMIC	36
2.12.	SEGURIDAD.....	40
2.12.1.	Seguridad de base de datos	40
2.12.2.	Seguridad lógica	41
2.12.3.	Seguridad física	41
2.13.	Pruebas de software	42
2.13.1.	Pruebas de caja blanca	42
2.13.2.	Pruebas de caja negra.....	42
2.13.3.	Pruebas de carga	42
2.13.4.	Pruebas de estrés.....	43
2.13.5.	Pruebas de resistencia	43
2.13.6.	Pruebas de escalabilidad.....	43

3. MARCO APlicativo	44
3.1. DIAGRAMAS CASOS DE USO	44
3.1.1. Casos de uso.....	44
3.1.2. Diagramas	44
3.2. ANÁLISIS DE REQUERIMIENTOS (PRODUCT BACKLOG)	48
3.2.1. Refinamiento del Product Backlog.....	50
3.2.2. Identificación de usuarios	50
3.2.3. Definición de roles por actor	50
3.2.4. Requerimientos de usuario.....	51
3.3. PLANIFICACIÓN DEL SPRINT.....	51
3.3.1. Sprint 1	52
3.3.2. Sprint 2	53
3.3.3. Sprint 3	54
3.4. DESARROLLO DEL SISTEMA.....	54
3.4.1. Semana 1	57
3.4.2. Semana 2	47
3.4.3. Semana 3	50
3.4.4. Semana 4	54
3.4.5. Semana 5	58
3.4.6. Semana 6	62
3.4.7. Semana 7	66
3.4.8. Semana 8	69
3.4.9. Semana 9	73
3.4.10. Semana 10	75
3.4.11. Semana 11	78
3.4.12. Semana 12	81
3.5. PRUEBAS DE EVALUACIÓN DE RESULTADOS.....	84
3.5.1. Métricas de calidad.....	84
3.5.2. Seguridad	92
3.5.3. Análisis de costos	93
3.5.4. Pruebas de software.....	97

4. CONCLUSIONES Y RECOMENDACIONES.....	105
4.1. CONCLUSIONES.....	105
4.2. RECOMENDACIONES	105
REFERENCIAS BIBLIOGRÁFICAS	107
ANEXOS A.....	110
ANEXOS B	111
ANEXOS C	114
ANEXOS D.....	118

ÍNDICE DE TABLAS

Tabla 1. Análisis de requerimientos	48
Tabla 2. Identificación de usuarios	50
Tabla 3. Definición de roles por actor.....	50
Tabla 4. Requerimientos de usuario.....	51
Tabla 5. Sprint 1	52
Tabla 6. Sprint 2	53
Tabla 7. Sprint 3.....	54
Tabla 8. Desarrollo del sistema	55
Tabla 9. Semana 1	57
Tabla 10. Semana 2	47
Tabla 11. Semana 3	50
Tabla 12. Semana 4	54
Tabla 13. Semana 5	58
Tabla 14. Semana 6	62
Tabla 15. Semana 7	66
Tabla 16. Semana 8	70
Tabla 17. Semana 9	73
Tabla 18. Semana 10	76
Tabla 19. Semana 11	78
Tabla 20. Semana 12	81
Tabla 21. Usabilidad.....	85
Tabla 22. Capacidad de mantenimiento	87
Tabla 23. Capacidad de mantenimiento	88
Tabla 24. Puntos de fusión.....	88
Tabla 25. Puntos de fusión.....	89
Tabla 26. Tabla de mapeo	94

ÍNDICE DE FIGURAS

Figura 1. Algunos clientes destacados de Linxs SRL.....	2
Figura 2. Organigrama de Linxs SRL	2
Figura 3. Esquema de aplicación de BI.....	15
Figura 4. Arquitectura de BI.....	16
Figura 5. Cuadrante de Gartner para BI.....	18
Figura 6. Metodología Scrum	24
Figura 7. Proceso de medición COSMIC	37
Figura 8. Tipos de sub-procesos definidos por el modelo COSMIC.....	39
Figura 9. Diagramas administradores	45
Figura 10. Diagramas compradores	45
Figura 11. Diagramas vendedores	46
Figura 12. Diagrama modulo Sistema	46
Figura 13. Diagrama modulo administrador	47
Figura 14. Diagrama modulo Usuario.....	47
Figura 15. Diagrama modulo ventas	48
Figura 16. Burn Down Chart semana 1	59
Figura 17. Base de datos	47
Figura 18. Burn Down Chart semana 2	49
Figura 19. Login	50
Figura 20. Burn Down Chart semana 3	53
Figura 21. Modulo Listado de Roles	53
Figura 22. Modulo Lista de Usuarios	54
Figura 23. Burn Down Chart semana 4	57
Figura 24. Modulo Clientes.....	57
Figura 25. Modulo Editar Usuario.....	58
Figura 26. Burn Down Chart semana 5	61
Figura 27. Modulo Clientes.....	61
Figura 28. Modulo Editar Cliente	62
Figura 29. Burn Down Chart semana 6	65
Figura 30. Modulo Categorías	65

Figura 31. Modulo Editar Categoría	66
Figura 32. Burn Down Chart semana 7	69
Figura 33. Modulo Detalle Ventas	69
Figura 34. Burn Down Chart semana 8	72
Figura 35. Modulo Productos	72
Figura 36. Modulo Editar Producto	73
Figura 37. Burn Down Chart semana 9	75
Figura 38. Modulo agregar venta	75
Figura 39. Burn Down Chart semana 10	77
Figura 40. Código fecha y hora	78
Figura 41. Burn Down Chart semana 11	80
Figura 42. Alerta Información	80
Figura 43. Burn Down Chart semana 12	83
Figura 44. Modulo Editar Usuario	83
Figura 45. Módulo Login (Dispositivo móvil)	84
Figura 46. ISO 25000	84
Figura 47. Calidad del Producto Software	85
Figura 48. Prueba de Caja Blanca	97
Figura 49. Login	98
Figura 50. Resultado de Datos	98
Figura 51. Base de Datos Query	99
Figura 52. Iniciar sesión con datos	99
Figura 53. Iniciar sesión con datos	100
Figura 54. Resultado de Datos	100
Figura 55. Base de Datos Query	100
Figura 56. Modulo Clientes	101
Figura 57. Login Test	101
Figura 58. Resultado Login Test	101
Figura 59. User Test	102
Figura 60. Resultado User Test	102
Figura 61. Usuario test	102

Figura 62. Resultado Usuario Test.....	103
Figura 63. Proveedores Test.....	103
Figura 64. Resultado Proveedores test.....	103
Figura 65. Productos Test	104
Figura 66. Resultado Productos Test	104

CAPÍTULO I

MARCO PRELIMINAR

1. MARCO PRELIMINAR

1.1. INTRODUCCIÓN

Las tecnologías de la información están transformando las actividades económicas y cotidianas. Indiscutiblemente los sistemas de información han invadido a todos y cada uno de los campos de la actividad humana: ciencia tecnología, arte, educación, recreación, administración y economía la tecnología relacionada con el manejo de la información hace que más personas y servicios estén conectados e interactúen entre si las tareas y funciones que tradicionalmente se realizaban manualmente ahora son hechas por medio de un sistema de información.

En este contexto el desarrollo de sistemas de información, administración y control juega un papel importante y cada vez más preponderante en la institución para poder competir y subsistir en el medio.

Por otra parte, las empresas se esfuerzan para tener una mayor participación en el mercado lo cual ha impulsado a que recurran a apoyarse en sistemas de información orientadas a la gestión y control de ventas. Estos sistemas ayudan a mejorar su administración mediante un seguimiento preciso de todas las transacciones que se realizan dentro de un negocio proporcionado reportes detallados de ventas que permiten a los directores tomar las decisiones más acatadas para el logro de sus objetivos.

En el presente proyecto se ha desarrollado un sistema de información para la administración y control de ventas para la empresa Linxs SRL, acudiendo a la metodología SCRUM ya que se adapta al desarrollo ágil y preciso para la creación de sistemas de información por medio del diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación. En cuanto a lenguaje de programación, se utilizará JavaScript y PHP.

1.2. ANTECEDENTES

Esta sección se destina a repasar brevemente las distintas investigaciones elaboradas en el ámbito académico y que se relacionan con el objeto de estudio del presente Proyecto de Grado.

1.2.1. Antecedentes Institucionales

Linxs SRL es una empresa dedicada al desarrollo de aplicaciones de escritorio, páginas web, sistemas de reservas, material de publicidad, ventas de espacios publicitarios y motores, se encuentra en el mercado desde el año 2005 y ha ganado experiencia atendiendo a decenas de clientes en Bolivia, Estados Unidos y Puerto Rico. Ha diseñado páginas web, CD Multimedia y material de Merchandising para los más variados rubros y empresas, desde Hotelería y Turismo, hasta Industria, Banca y Gastronomía. Conocemos de calidad. Actualmente cuenta con alrededor de 7500 clientes en los nueve departamentos de Bolivia.

De acuerdo son su sitio Web oficial, algunos de sus clientes más destacados son:

- ✓ Hotel Presidente
- ✓ Real Plaza Bolivia
- ✓ Don Pollo
- ✓ Senda Andina
- ✓ House Inn Apart Hotel
- ✓ Churrasqueria el Buen Gusto
- ✓ Visit Bolivia
- ✓ Gran Hotel Canada
- ✓ Alexander Palace Hotel
- ✓ Hotel Continental Park
- ✓ Hotel Copacabana
- ✓ Hotel Gloria
- ✓ Hoteles Ecologicos
- ✓ Hotel Yotau
- ✓ Sun Hotel
- ✓ Apart Hotel Premium Suites

Figura 1. Algunos clientes destacados de Linxs SRL

Fuente: Sitio Web oficial de Linxs SRL (www.linxs.com.bo)

Para un servicio de diseño web de calidad, LINXS DISEÑOS WEB cuenta con un equipo altamente talentoso en tecnología y diseño gráfico que proveen asesoramiento al cliente para desarrollar páginas Web dinámicas, amenas y con un atractivo diseño.



Figura 2. Organigrama de Linxs SRL

Fuente: Linxs SRL

Provee, además, ideas para desarrollar el concepto de la página Web, y al mismo tiempo, atiende la demanda de pequeñas, medianas y grandes empresas y utiliza las últimas tendencias de programación y diseño gráfico, también ofrece el servicio de base de datos y motores a los clientes que requieran manejar su información de manera dinámica.

Los clientes no tienen que preocuparse por buscar otras empresas de proveedores de dominio y hosting ya que la empresa cuenta con estos servicios a precios accesibles y cuenta con un hosting de confianza que provee la empresa Mega Link reconocida a nacional, cuenta con el personal capacitado para la atención a los clientes para el desarrollo de páginas web y además servicios que ofrece (Linx Srl, 2017).

1.2.1.1. Misión

La misión de la empresa es “Satisfacer las necesidades de publicidad, presencia en Internet e imagen corporativa de nuestros clientes utilizando medios tecnológicos, digitales, e innovadores procesos de impresión. Superar las exigencias de nuestros clientes y buscar la excelencia en todos nuestros servicios. Desarrollar a nuestros profesionales para que ellos trabajen de manera continua y progresiva en alcanzar siempre mayores estándares de calidad y eficiencia. Contribuir al desarrollo de

nuestro país, de las empresas que confían en nosotros y de las personas que se relacionan comercialmente con nuestra empresa”.

1.2.1.2. Visión

La visión de la empresa es “Ser la mejor alternativa para presentar productos y servicios empresariales en internet, en productos multimedia y materiales promocionales”.

1.2.2. Antecedentes internacionales

En primera instancia, se repasan estudios realizados más allá de las fronteras bolivianas, con el siguiente resultado.

- [Tufiño Guillén, Gabriela Nataly, 2012] SISTEMA DE CONTROL DE COMPRAS Y FACTURACIÓN DE VENTAS PARA LA COOPERATIVA DE PRODUCCIÓN DE PANELA (COPROPAP); Frente a la evidente necesidad de las pequeñas microempresas como la Cooperativa de Producción de Panela "El Paraíso" (COPROPAP) para agilizar los procesos que maneja con respecto al control de la producción y pagos de la misma, el proyecto de grado, plantea la creación de un sistema para administrar COPROPAP y así facilitar un control óptimo de la producción y llevado de registros claros que vayan en beneficio de todos los socios y familias que dependen de esta actividad económica [Santiago, Universidad Central de Chile]
- [Yacelga Maldonado, Edwin Santiago, 2016] IMPLEMENTACIÓN DE UN PORTAL WEB PARA LA EMPRESA LONG LIVE LEARNING, QUE PERMITA INTEGRARSE CON LA PLATAFORMA EDUCATIVA MOODLE Y AUTOMATIZAR EL PROCESO DEL PORTAFOLIO DE PRODUCTOS Y SERVICIOS; Long Live Learning es una empresa de capacitación que desea reestructurar la administración de negocio en relación al marketing que maneja. Con expectativas de mejorar los procesos internos y oportunidades de comercio de los productos y servicios que ofrecen, por la tanto se observa la necesidad de analizar, diseñar e implementar un sistema de administración académica, que integre un portal web y una herramienta de educación virtual [Santiago, Universidad Central de Chile].

- [Diaz Chavez, Carlos Eduardo; Medina Pico, David Eduardo, 2010] IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO ESPECIALIZADO PARA EL CONTROL Y RESERVACIÓN DE BOLETERÍA EN LA COOPERATIVA DE TRANSPORTES AEROTAXI; El proyecto de grado se enfoca en el desarrollo de una aplicación Web que una vez implementado, facilitará el control y reserva de boletería para la Cooperativa de Transportes Aerotaxi, el cual constituye una herramienta de apoyo y mejora para la atención al cliente, así como proveedor de información de calidad a los directivos [Madrid, Universidad Autónoma de Madrid]
- [Diaz Ortiz, Jimmy Jhonon; Romero Suarez, Mitcheli Anthony, 2017] DESARROLLO E IMPLEMENTACIÓN DE UN APlicativo WEB, UTILIZANDO LA METODOLOGÍA SCRUM, PARA MEJORAR EL PROCESO DE ATENCIÓN AL CLIENTE EN LA EMPRESA Z ADITIVOS S.A.; Z Aditivos S.A. es una empresa dedicada al desarrollo, fabricación y comercialización de aditivos para el concreto. Se ha observado que la información referente al seguimiento de los pedidos y los servicios de atención al cliente se encuentran archivados de manera rustica en folders y papel, lo cual hace difícil el monitoreo y control de los mismos, además de que dicha documentación tal como se almacena no aporta ningún valor al negocio [Madrid, Universidad Autónoma de Madrid].

1.2.3. Antecedentes nacionales

En cuanto a los estudios realizados en el territorio boliviano, se tienen los siguientes.

- [Limachi Machaca, Karla Belen, 2018] SISTEMA WEB PARA EL CONTROL Y SEGUIMIENTO DE VENTAS DE PRODUCTOS ARTESANALES. CASO: BOLIVIA TECH HUB; El proyecto de grado consiste en implementar un sistema web de control y seguimiento de ventas de productos artesanales para la empresa bolivia tech hub, y de acuerdo con la autora, surge del gran inconveniente que sufren los artesanos, que no disponen de oportunidades para publicar y promover su producción al mercado [La Paz, Universidad Privada Boliviana]

- [Ramos Huarachi, Liz Andrea, 2013] SISTEMA WEB DE ADMINISTRACIÓN DE VENTAS, BASADO EN LA GESTIÓN DE PROCESOS DE NEGOCIOS. CASO: EMPRESA DE MUEBLES ARTEMOBILE, DEPARTAMENTO DE VENTAS; El proyecto propuesto realiza la implementación de un sistema para la adecuada administración de todas las ventas que se realizan en la empresa ARTEMOBILE, dado que se llegó un punto en el cual la información se hizo imposible de manejar, saber cuál nos sirve o no y si es necesario hacer una modificación a dicha información. Todo este proceso tiene además un alto coste asociado puesto que no sólo se gastan recursos materiales sino humanos [La Paz, Universidad Privada Boliviana]
- [Gutierrez Vargas, Grover, 2015] SISTEMA DE CONTROL DE VENTAS E INVENTARIOS PARA ALMACENES DE ALUMINIOS UTILIZANDO DISPOSITIVOS MÓVILES. CASO: TÉCNICA DE ALUMINIO, VIDRIO Y SERVICIOS (TALVISER); El proyecto de grado pretende automatizar los procesos que la empresa realiza como las compras de materiales, venta de productos además de centralizar la información de materiales, usuarios, proveedores, para evitar duplicidad y poder acceder de manera inmediata a la información de los productos que la empresa comercializa, ya que estos procesos son de vital importancia para la empresa y es necesario controlar la información que se genera día a día [La Paz, Universidad Católica Boliviana “San Pablo”]

1.3. PLANTEAMIENTO DEL PROBLEMA

La empresa Linxs SRL tiene dificultades en la gestión de información debido a que la información se encuentra incompleta. Acceder a la información de ventas es muy difícil y lenta ya que el responsable del área contable debe llenar los datos manualmente en plantillas Excel y para poder visualizar se tiene que buscar los registros uno por uno. Los vendedores al momento de realizar una cobranza, renovación o venta de servicios a un cliente no cuentan con información fidedigna. No se pueden detectar a clientes con servicios vendidos. Por tal motivo existen pérdidas económicas, inconsistencia en la información, duplicidad de datos, insatisfacción de demanda, fallas en el control de ventas de cada cliente.

1.3.1. Problema principal

La empresa Linxs SRL experimenta pérdidas económicas por la falta de control y seguimiento en las ventas de los servicios y productos que realiza.

1.3.2. Problemas secundarios

- Demora en la ubicación de información debido a que la búsqueda se realizará en forma manual.
- Información desactualizada de ventas debido a que los datos se encuentran en hojas electrónicas dispersas.
- No cuenta con información completa en cuanto a las ventas porque se realiza de forma manual, lo que ocasiona no poder realizar resúmenes de los movimientos que hay y en determinados períodos de tiempo, porque el procesamiento es manual.
- Falta de disponibilidad de la información en soporte digital, debido a que no hay procesamiento de la información en un servidor central.
- Existe variaciones en los controles de inventarios físicos y lógicos en la mayoría de los casos existen software faltantes y sobrantes, se invierte mucho tiempo en verificar y corregir errores de los inventarios.
- Se usa excesivamente el papel al producir copias innecesarias de un mismo documento lo que ocasiona acumulación de documentos.
- No existe una base de datos centralizada debido a que los datos están dispersos.
- Los requerimientos del sistema actual se realizan manualmente.

¿Cómo apoyaría la administración y control de ventas con la implementación de un sistema de información en la empresa Linxs?

1.4. OBJETIVOS

1.4.1. Objetivo general

Desarrollar un sistema de información para administración y control de ventas, para apoyar el proceso de ventas en la empresa Linxs SRL de la Ciudad de La Paz.

1.4.2. Objetivos específicos

- Determinar los módulos para el desarrollo del sistema a partir del análisis de la información recopilada.
- Organización de la información y centralización de datos
- Desarrollar un sistema de información para apoyar a la institución.
- Desarrollo del sistema utilizando múltiples tecnologías.
- Realizar la ingeniería de requerimientos para establecer los requerimientos funcionales y no funcionales del sistema.
- Elaborar el diseño del sistema identificando las entidades y las relaciones que sustenten al sistema de información.
- Integrar toda la información del sistema de información en reportes para toma de decisiones utilizando Business Intelligence.
- Efectuar pruebas de control y calidad del sistema aplicando ISO 25000.
- Realizar el costeo del software a través de la metodología COSMIC.

1.5. JUSTIFICACIÓN

1.5.1. Justificación técnica

La empresa Linxs SRL cuenta con las herramientas adecuadas como recursos de software y hardware para desarrollar el sistema. La empresa cuenta con equipos de computación en cada uno de sus áreas, gerencia general, área contable, vendedores y áreas de sistemas, mismos que tienen acceso a una red internet e intranet. Así mismo la empresa tiene un hosting donde estará alojado la base de datos del sistema ya sea para uso interno como para el acceso externo, garantizando un acceso inmediato. Así mismo el personal está capacitado con el manejo de metodologías agiles, tienen conocimiento en herramientas de desarrollo.

1.5.2. Justificación económica

Con la implementación del sistema en la empresa disminuirán las pérdidas económicas a través del seguimiento y control de servicios a los clientes, evitando que existan servicios sin cobrar, clientes y servicios olvidados, permitiendo renovar con anticipación, también evitará las solicitudes de baja por parte de los clientes y disminuirán los gastos en llamadas telefónicas continuas por partes de los

vendedores y del mensajero. En cuanto a las herramientas para el desarrollo del sistema se empleará la licencia GPL, General Public License (Licencia Pública General) que es totalmente libre por lo que no se tendrá ningún costo.

También el sistema ayudará a lograr mejores tiempos de respuesta, siempre que se mejoren los procesos de venta, lo que redundará en beneficios de costo, tiempo y productividad para la empresa, reduciendo significativamente el costo total del proyecto, así como su duración.

1.5.3. Justificación Social

A medida que avanza el tiempo la cantidad de clientes va en aumento, la empresa Linxs SRL debe ser capaz de atender y mejorar sus servicios para que los mismos se sientan seguros y confíen en la compañía. Con el sistema los vendedores cuentan con información rápida oportuna y segura de los clientes suscritos y mejora la atención a los mismos, también es una ayuda para la gerencia general y el área contable porque les facilita en el control y seguimiento de las ventas.

Asimismo, este proyecto pondrá a disposición de los gerentes y personal de ventas la información necesaria con los reportes requeridos, con la facilidad de acceso directo cada vez que lo requieran. También se mejorará la atención al cliente al agilizar el proceso de venta de cada producto, evitando la deserción de clientes y al contrario aumentar la lista de nuestros clientes.

1.6. METODOLOGÍA

1.6.1. Metodología de desarrollo

La metodología ágil seleccionada para el desarrollo de la aplicación es SCRUM ya que ofrece Gestión regular de las expectativas del cliente, basada en resultados tangibles, resultados anticipados, flexibilidad y adaptación respecto a las necesidades del cliente, cambios en el mercado, mitigación sistemática de los riesgos del proyecto (Limachi, 2018).

1.7. MÉTRICAS DE CALIDAD

La ISO/IEC 25000, conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es una familia de normas que tiene por objetivo la

creación de un marco de trabajo común para evaluar la calidad del producto software.

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones.

- ISO/IEC 2500n - División de Gestión de Calidad
- ISO/IEC 2501n - División de Modelo de Calidad
- ISO/IEC 2502n - División de Medición de Calidad
- ISO/IEC 2503n - División de Requisitos de Calidad
- ISO/IEC 2504n - División de Evaluación de Calidad

1.8. HERRAMIENTAS

1.8.1. Lenguajes de Programación

- PHP: Para el desarrollo de frontend y backend en el sistema de información.
PHP es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web. [Knowlton Jim, 2009]
- JavaScript: Lenguaje de programación, scripting del lado del cliente
[Niederst, 2018]

1.8.2. Bases de datos

- MariaDB: Es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License).

1.8.3. Servidores

- Nginx: Se utilizará como servidor principal debido a que es mucho más rápido cuando se tiene mucho tráfico de datos. Servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3). Nginx se utiliza para aplicaciones de gran volumen con un gran número de conexiones únicas, a diferencia de las que tienen pocas conexiones que requieren muchos datos. [Hoffman, 2020]

1.9. LÍMITES Y ALCANCES

1.9.1. Límite

- Solo el personal autorizado podrá acceder al sistema
- El sistema no presentará proyección de ventas

1.9.2. Alcances

El sistema de información contemplara el desarrollo de los siguientes módulos:

- Módulos de autenticación de usuarios
- Módulo de clientes
- Módulo de categoría
- Módulo de productos
- Módulos de información de productos
- Módulo de proveedores
- Módulo ventas
- Módulo de usuarios
- Módulo de roles
- Módulos de reportes
- Módulos de control y seguimiento de ventas
- Medidas de seguridad de respaldo de información
- Permisos para múltiples usuarios
- Módulo de administración de recursos de datos
- Módulo de administración de la información
- Módulo de análisis de datos
- El sistema, cuyo funcionamiento es adaptable

1.10. APORTES

El sistema propuesto es de mucha utilidad para la empresa por que brinda información oportuna y segura a los usuarios, generando una mejor interacción entre el personal y buena atención a los clientes, además de realizar un mejor seguimiento a los servicios de clientes.

1.10.1. Aporte práctico

En este contexto, el aporte práctico del presente proyecto de grado permitirá optimizar el control y manejo de la información de clientes y ventas así como de los productos que ofrece la empresa. Esto también implica que la centralización y automatización de la información se traducirá en una mayor eficiencia en la administración del tiempo.

1.10.2. Aporte teórico

Al utilizar una metodología se brindará un servicio para el seguimiento y control de los movimientos de ventas e inventarios de manera precisa. La metodología SCRUM es una metodología ágil centrada en trabajar en equipo a partir de iteraciones o Sprints, como clave para el éxito en desarrollo de software.

El aporte del presente proyecto es utilizar una metodología para brindar servicio, que permita el seguimiento y control de los movimientos de ventas de manera precisa para la toma de decisiones.

Así mismo para aumentar las ventas el sistema utilizará la infraestructura y las herramientas de Business Intelligence (BI) para agilizar las ventas y tomar decisiones gerenciales, en las ventas, es la estrategia que puede representar la diferencia entre superar o no a los competidores. Con la ayuda de Business Intelligence, se facilita la organización de la información relacionada con el comportamiento del mercado y los clientes, lo que permite anticipar tendencias, redefinir los procesos internos y en última instancia, aumentar la ganancia de la empresa.

CAPÍTULO II

MARCO TEÓRICO

2. MARCO TEÓRICO

2.1. INTRODUCCIÓN

Este capítulo presenta los aspectos teóricos que fundamentan las referencias empleadas a lo largo del documento. Se dedica una sección para cada concepto aplicado. Así, se desarrolla la descripción de los sistemas de gestión de bases de datos MariaDB. También se desarrollan los aspectos conceptuales relativos a las metodologías ágiles y en específico se aborda la metodología SCRUM.

Otros aspectos complementarios que se trata en este capítulo, se refieren a los lenguajes de programación PHP y Java Script, la gestión de calidad de software y estimación de costos.

Asimismo, para su completo entendimiento, se describen los conceptos más importantes relacionados con el tema del proyecto de grado.

2.2. SISTEMA

Un sistema se define generalmente como un conjunto de componentes o elementos interrelacionados entre sí y que funcionan como un todo, para lograr un objetivo y a través de este obtener la solución de una situación problemática. (JJ. Daymod, 2019)

2.3. INFORMACIÓN

La información es un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje. (Vanessa TA, 2020)

2.4. ADMINISTRACIÓN

La Administración es el proceso de organizar y controlar el manejo de los recursos y las actividades de trabajo con el propósito de lograr los objetivos o metas propuestas de la organización de manera eficiente y eficaz. Permite implementar nuevas estrategias para el logro de las metas para tener éxito en la empresa u organización. (James Dalamont M., 2013)

2.5. CONTROL DE VENTAS

El control de ventas es el análisis y medición del desempeño de la fuerza de ventas, comparándolo con el desempeño estándar. Consiste en hacer notar y señalar las desviaciones, determinando sus causas, y tomar las medidas correctivas adecuadas para enfrentar diferentes situaciones. (Simon C., 2020)

2.6. GESTIÓN DE VENTAS

Cuando se habla de ventas, debe hacerse referencia al proceso de satisfacción de las necesidades del cliente, a cambio del beneficio tanto de la empresa como del vendedor. En consecuencia, la gestión comercial debe ser una parte importante del sistema de dirección en las organizaciones, lo que posibilita conocer el mercado, la competencia y adaptarse a las nuevas y cambiantes condiciones del entorno (Clarke Bloomfield, Cisneros Arias y Paneca González, 2018).

Para ello es importante considerar que la gestión de ventas involucra diferentes procesos: desarrollo de los productos, elaboración de los precios, servicio al cliente, marketing, planificación y evaluación de resultados, son algunas de las áreas que deben recibir una debida atención para que el negocio crezca y tenga mayores beneficios. Más que tener un equipo eficiente y capacitado, la cualificación del liderazgo es fundamental para la optimización de la gestión de ventas (Acosta Véliz, Salas Narváez, Jiménez Cercado y Guerra Tejada, 2018).

1. Control del proceso de ventas. En un artículo para Entrepreneur, Tony Parinello, autor del libro Getting to VITO, the Very Important Top Officer, 10 Steps to VITO's Office, sugiere que el gestor asuma la responsabilidad personal de asegurar que cada etapa del proceso de ventas se despliegue en un tiempo hábil. Parinello también afirma que, si cualquier etapa del proceso de ventas se olvida o no se completa de acuerdo con el tiempo estipulado previamente, los resultados cambian. Según él, lo ideal es establecer una planificación correcta y seguirla.
2. Planificación de ventas. La planificación de ventas es esencial para las empresas que quieren alcanzar mejores resultados en el mercado. Sirve como una guía para el equipo, pues establece metas y objetivos. Con este recurso

se hace una gestión de ventas más productiva. Para esto, es necesario definir algunos criterios, como el público objetivo de la empresa, el ciclo de compras y ventas, la estrategia de mercado y las formas de tratar con la competencia. Además, es necesario establecer metas de ventas. Estas deben ser realistas y comunicadas a todo el equipo. Mientras más comprometidos están los empleados, hay más oportunidades de llegar a los resultados esperados en determinado periodo.

3. Contratación de talentos. La gestión de ventas de una empresa se perjudica cuando el equipo carece de profesionales comprometidos y capacitados. Por eso, invertir en la contratación de talentos es una estrategia fundamental, que no debe ser vista como un gasto. Si el emprendedor quiere obtener mejores resultados en las ventas, entonces es necesario contar con los mejores empleados. Acuérdate: la falta de profesionales calificados genera aún más gastos para el emprendimiento, principalmente a largo plazo, debido a la rotación.
4. Capacitación del equipo. Además de contratar talentos para la empresa, es esencial apostar por la capacitación constante del equipo de ventas, con el fin de mejorar las habilidades de los profesionales, como la capacidad de relacionamiento y el trabajo con metas, por ejemplo. Los gestores tampoco deben olvidarse de capacitar a los empleados en cuanto al conocimiento sobre la realidad de la empresa. Es importante que ellos también estén informados sobre la competencia, productos, perfil del cliente y planificación del negocio.

2.7. BUSINESS INTELLIGENCE

De acuerdo con el sitio Web Software Testing Help, BI es una colección de software que se utiliza para apoyar el proceso de toma de decisiones de los analistas y administradores. La Inteligencia de Negocios puede definirse como el análisis y procesamiento de una gran cantidad de datos y su posterior conversión en información basada en el conocimiento para apoyar algunas decisiones de negocios rentables. El entorno BI comprende modelos de negocio, modelos de datos y herramientas ETL para organizar y transformar los datos en información útil.

BI utiliza algunos términos como:

- Big Data es una colección de grandes y complejos conjuntos de datos que contiene datos estructurados y no estructurados que pueden ser difíciles de procesar y analizar utilizando las herramientas tradicionales de gestión de bases de datos.
- Data Warehouse es un sistema integrado y orientado a la materia para reportar y analizar los datos para apoyar un proceso de toma de decisiones.
- La minería de datos (Data Mining) es un proceso de aplicación de algunas técnicas estadísticas a una gran cantidad de datos en bruto y los convierte en información útil con nuevas pautas y relaciones entre grandes bases de datos relacionales.

En la siguiente figura se presenta una representación esquemática del proceso de aplicación de la Inteligencia de Negocios, que ayudará a comprender más fácilmente la Inteligencia de Negocio.

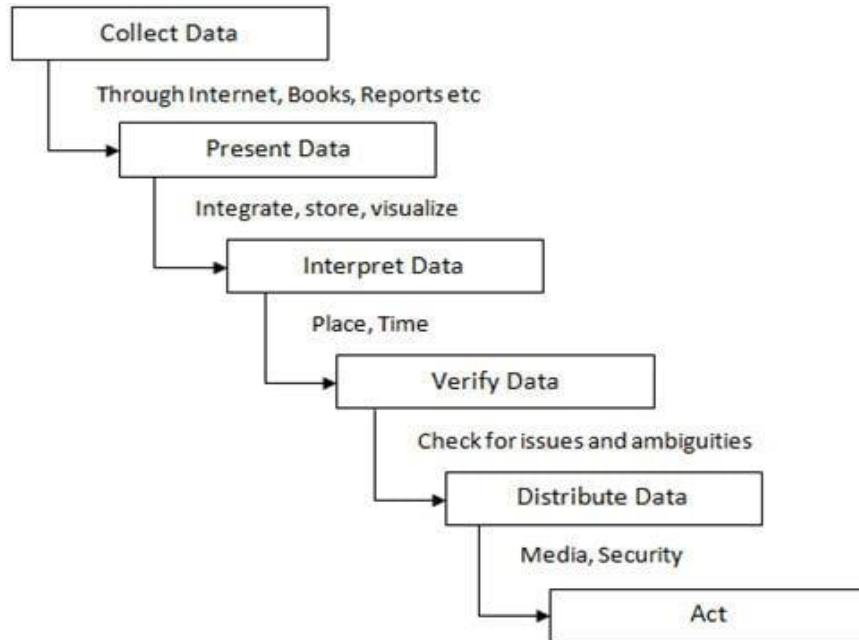


Figura 3. Esquema de aplicación de BI

Fuente: www.softwaretestinghelp.com

2.7.1. Arquitectura del BI

Un sistema BI, tiene cuatro grandes componentes: un DataWarehouse (DW), con su origen de datos; Business Analytics, una colección de herramientas para manejo,

minería y análisis de datos en el DW; un BPM para monitorear y analizar el rendimiento o desempeño; y una interfaz de usuario.

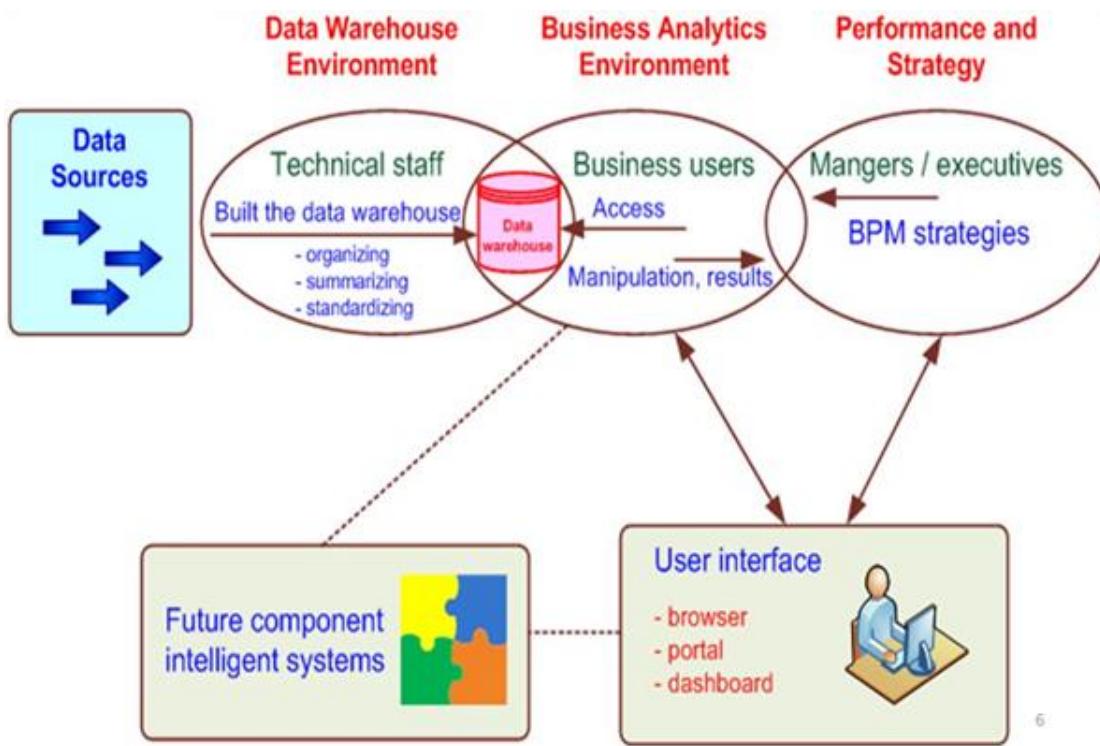


Figura 4. Arquitectura de BI
Fuente: Sharba, Denle, Turban (2018)

2.7.2. Desafíos en la implementación de BI

Aunque varias organizaciones están implementando activamente la BI y han demostrado su utilidad para sus negocios, existen algunos desafíos para implementarla. Algunos de ellos se enumeran a continuación:

- Hay una enorme cantidad de datos que se recogen todos los días pero que no es posible procesar todos en un momento dado.
- Falta de estrategia.
- La adopción que está impulsando a los usuarios o grupos de usuarios que utilizan diferentes formas no quiere cambiarla a menos que el método que están utilizando se vuelva lento e ineficiente.
- Justificación de las inversiones que están estimando el costo de descubrir nuevos métodos en el proceso comercial.
- Cambio en la gestión.

- Gestión de datos no transaccionales.
- Gobernanza de los datos de la empresa.
- Brecha de conexión entre las TI y los usuarios empresariales.
- Acceso a la información para una amplia gama de usuarios.
- Integración de la seguridad y la personalización.

2.7.3. Cuadrante de Gartner para herramientas de BI

Por decimotercer año consecutivo, Gartner ha nombrado a Microsoft líder del Cuadrante Mágico de 2020 en la categoría de plataformas de análisis e inteligencia empresarial.

El Cuadrante Mágico de Gartner es una publicación anual que mide la situación de mercado de un producto tecnológico. Contempla, por un lado, la innovación del proveedor y, por otro, la habilidad en el desarrollo de los productos. Su impulsora, Gartner, es la consultora de IT más importante del mundo, y se dedica a analizar las tendencias de mercado.

Este Cuadrante Mágico de Gartner se ha erigido como herramienta líder a nivel mundial en la industria IT, al permitir a las empresas que quieran contratar servicios y soluciones IT tener una visión global de un área de productos. Algo que, indudablemente, repercute después en su elección de proveedor.



Figura 5. Cuadrante de Gartner para BI

Fuente: Gartner (2020)

En el Cuadrante Mágico de Gartner, el eje X indica la “integridad de visión”, y clasifica la habilidad de los proveedores para aprovechar el momento que vive el mercado para generar valor. El eje Y indica la “capacidad de ejecutar” y clasifica la habilidad de los proveedores para ejecutar con éxito su visión de mercado.

2.8. METODOLOGÍAS AGILES DE DESARROLLO DE SOFTWARE

La metodología se refiere a un conjunto de procedimientos racionales que se utilizan para lograr una meta que requiere ciertas habilidades y conocimientos. La metodología es una de las fases específicas de un trabajo o proyecto que parte de una posición teórica y conduce a una selección de técnicas o métodos específicos sobre el procedimiento para la consecución de los objetivos. Son los métodos que

se utilizan en una determinada actividad para formalizarla y optimizarla y determina los pasos a seguir y cómo completar una tarea (Maida y Pacienzia, 2015). Al presente, pueden reconocerse dos metodologías: las metodologías tradicionales y las metodologías ágiles:

- Las metodologías tradicionales son denominadas a veces de forma despectiva, como metodologías pesadas. Centran su atención en llevar una documentación exhaustiva de todo el proyecto, la planificación y control del mismo en especificaciones precisas de requisitos y modelado y en cumplir con un plan de trabajo definido todo esto, en la fase inicial del desarrollo del proyecto. Estas metodologías tradicionales imponen una disciplina rigurosa de trabajo sobre el proceso de desarrollo del software con el fin de conseguir un software más eficiente. Para ello se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. Otra de las características importantes dentro de este enfoque, son los altos costes al implementar un cambio y la falta de flexibilidad en proyectos donde el entorno es volátil. (Maida y Pacienzia, 2015)
- Las metodologías ágiles, nacen como respuesta a los problemas que puedan ocasionar las metodologías tradicionales y se basa en dos aspectos fundamentales, retrasar las decisiones y la planificación adaptativa. Basan su fundamento en la adaptabilidad de los procesos de desarrollo. Un modelo de desarrollo ágil, generalmente es un proceso Incremental (entregas frecuentes con ciclos rápidos), también Cooperativo (clientes y desarrolladores trabajan constantemente con una comunicación muy fina y constante), Sencillo (el método es fácil de aprender y modificar para el equipo) y finalmente Adaptativo (capaz de permitir cambios de último momento). Las metodologías ágiles

proporcionan una serie de pautas y principios junto a técnicas pragmáticas que hacen que la entrega del proyecto sea menos complicada y más satisfactoria tanto para los clientes como para los equipos de trabajo, evitando de esta manera los caminos burocráticos de las metodologías tradicionales, generando poca documentación y no haciendo uso de métodos formales. Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. (Maida y Pacienzia, 2015)

2.8.1. Metodología SCRUM

La metodología de trabajo de Scrum, tiene sus principios fundamentales en la década de 1980, la cual fue desarrollada por su necesidad en procesos de reingeniería por Goldratt, Takeuchi y Nonaka.

El concepto de Scrum tiene su origen sobre los nuevos procesos de desarrollo utilizados en productos exitosos. Los equipos que desarrollaron estos productos partían de requisitos muy generales, así como novedosos y debían salir al mercado en mucho menos del tiempo del que se tardó en lanzar productos anteriores. Estos equipos seguían patrones de ejecución de proyecto muy similares. En este estudio se comparaba la forma de trabajo de estos equipos altamente productivos y multidisciplinares con la colaboración entre los jugadores de Rugby y su formación de Scrum, de la cual se tomó su nombre (Pacienzia, 2015).

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Podríamos decir que SCRUM se basa en cierto caos controlado, pero establece ciertos mecanismos para controlar esta indeterminación, manipular lo impredecible y controlar la flexibilidad.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está

especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

2.8.1.1. Fundamentos de SCRUM

Scrum se basa en los siguientes puntos (Pacienzia, 2015):

- El desarrollo incremental de los requisitos del proyecto en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita). Las iteraciones se pueden entender como mini proyectos, en todas las iteraciones se repite un proceso de trabajo similar (de ahí el nombre “iterativo”) para proporcionar un resultado completo sobre el producto final, de manera que el cliente pueda obtener los beneficios del proyecto de forma incremental. Para ello, cada requisito se debe completar en una única iteración. El equipo debe realizar todas las tareas necesarias para completarlo (incluyendo pruebas y documentación) y que esté preparado para ser entregado al cliente con el mínimo esfuerzo necesario. De esta manera no se deja para el final del proyecto ninguna actividad arriesgada relacionada con la entrega de requisitos.
- La priorización de los requisitos por valor para el cliente y coste de desarrollo en cada iteración. Para que un proyecto proporcione el mejor resultado posible, y como soporte fundamental al control empírico del proyecto, es necesario re priorizar los requisitos de manera regular, en cada iteración, según el valor que proporcionan al cliente en ese momento y su coste estimado de desarrollo. Como resultado de esta re priorización se actualiza la lista de requisitos priorizada (Product Backlog).
- El control empírico del proyecto. Por un lado, al final de cada iteración se demuestra al cliente el resultado real obtenido, de manera que pueda tomar las decisiones necesarias en función de lo que observa y del contexto del proyecto en ese momento. Por otro lado, el equipo se sincroniza diariamente y realiza las adaptaciones necesarias.

- La potenciación del equipo, que se compromete a entregar unos requisitos y para ello se le otorga la autoridad necesaria para organizar su trabajo.
- La sistematización de la colaboración y la comunicación tanto entre el equipo y como con el cliente.
- El timeboxing de las actividades del proyecto, para ayudar a la toma de decisiones y conseguir resultados. La técnica del timebox consiste en fijar el tiempo máximo para conseguir ciertos objetivos, tomar una decisión o realizar unas tareas, y hacer lo mejor que podamos en ese intervalo. De esta manera, en lugar de ponerse a trabajar en algo hasta que esté hecho, de antemano se acuerda que sólo se dedica un tiempo limitado. La conciencia de esta limitación temporal favorece la priorización de objetivos/tareas y fuerza la toma de decisiones.

2.8.1.2. Requisitos para poder utilizar SCRUM

Los siguientes puntos son de especial importancia para la implantación de una gestión ágil de proyectos como Scrum (Pacienza, 2015):

- Cultura de empresa basada en trabajo en equipo, delegación, creatividad y mejora continua.
- Compromiso del cliente en la dirección de los resultados del proyecto, gestión del ROI y disponibilidad para poder colaborar.
- Compromiso de la Dirección de la organización para resolver problemas frecuentes y realizar cambios organizativos, formando equipos autogestionados y multidisciplinares y fomentando una cultura de gestión basada en la colaboración y en la facilitación llevada a cabo por líderes al servicio del equipo.
- Compromiso conjunto y colaboración de los miembros del equipo.
- Relación entre proveedor y cliente basada en la colaboración y transparencia.
- Facilidad para realizar cambios en el proyecto.
- Tamaño de cada equipo entre 5 y 9 personas (con técnicas específicas de planificación y coordinación cuando varios equipos trabajan en el mismo proyecto).

- Equipo trabajando en un mismo espacio común para maximizar la comunicación.
- Dedicación del equipo a tiempo completo.
- Estabilidad de los miembros del equipo.

2.8.1.3. Roles en la metodología SCRUM

En la metodología ágil Scrum hay tres roles principales: el Scrum Master, Propietario del producto y Equipo de Scrum. Todas estas personas están comprometidas con el proyecto de Scrum.

- Scrum Master. El facilitador del proceso de desarrollo del Scrum. Además de mantener reuniones diarias con el equipo de Scrum, el maestro de Scrum se asegura de que las reglas de Scrum se cumplan y se apliquen según lo previsto. Las responsabilidades del maestro de Scrum también incluyen entrenar y motivar al equipo, eliminar los impedimentos para los Sprints y asegurarse de que el equipo tenga las mejores condiciones posibles para cumplir sus objetivos y producir productos entregables (Business News Daily, 2020).
- Propietario del producto. El propietario del producto representa a las partes interesadas, que suelen ser clientes. Para asegurar que el equipo de Scrum siempre esté entregando valor a las partes interesadas y al negocio, el propietario del producto determina las expectativas del producto, registra los cambios en el producto y administra una lista de tareas pendientes del Scrum, una lista detallada y constantemente actualizada para el proyecto de Scrum. El propietario del producto también es responsable de priorizar los objetivos de cada sprint, en función de su valor para las partes interesadas, de modo que las características más importantes y entregables se incorporen en cada iteración (Business News Daily, 2020).
- Equipo de Scrum. El equipo de Scrum es un grupo auto-organizado de tres a nueve personas que tienen las habilidades comerciales, de diseño, analíticas y de desarrollo para llevar a cabo el trabajo real, resolver problemas y producir productos entregables. Los miembros del equipo de Scrum autoadministran

las tareas y son responsables conjuntamente de cumplir los objetivos de cada sprint (Business News Daily, 2020).

2.8.1.4. Proceso de aplicación SCRUM

El ciclo de vida de Scrum consiste en una serie de Sprints, donde el resultado final es un incremento de producto potencialmente transportable. Dentro de estos sprints, todas las actividades necesarias para el desarrollo del producto se producen en un pequeño subconjunto del producto total.

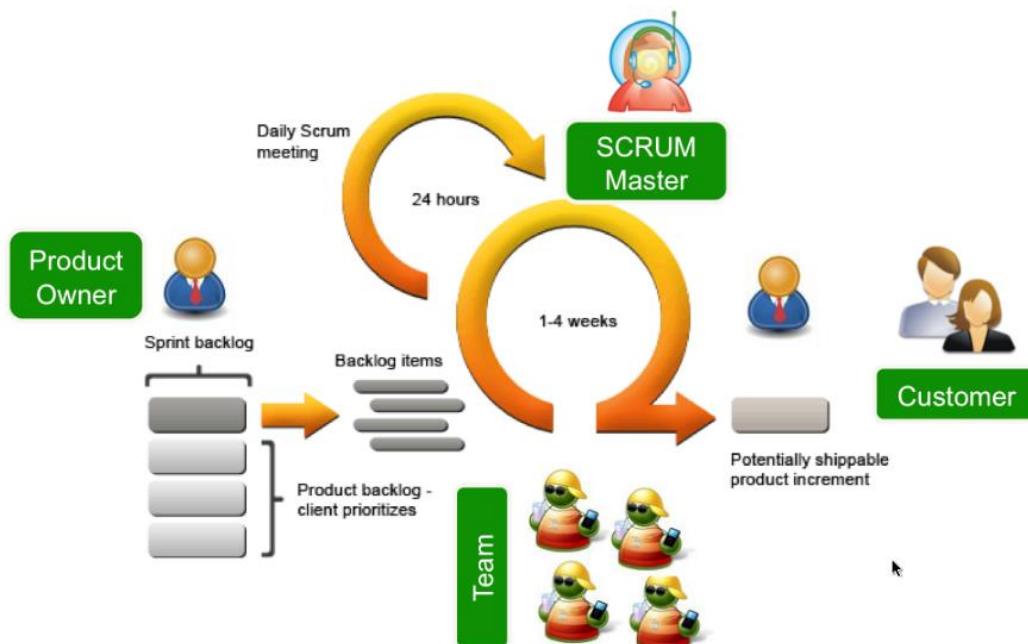


Figura 6. Metodología Scrum

Fuente: Pacienza (2015)

A continuación, se describen los pasos clave del ciclo de vida de Scrum (Agile Alliance, 2017):

1. Establecer el atraso del producto.
2. El propietario del producto y el equipo de desarrollo llevan a cabo la planificación del sprint. Determinar el alcance del Sprint en la primera parte del Sprint Planning y el plan para entregar ese alcance en la segunda mitad del Sprint Planning.
3. A medida que el Sprint progresá, el equipo de desarrollo realiza el trabajo necesario para entregar los elementos seleccionados del producto atrasado.
4. Diariamente, el equipo de desarrollo coordina su trabajo en un Scrum diario.

5. Al final del Sprint, el equipo de desarrollo entrega los artículos de productos atrasados seleccionados durante el Sprint Planning. El equipo de desarrollo lleva a cabo una Revisión Sprint para mostrar al cliente el incremento y obtener retroalimentación. El equipo de desarrollo y el propietario del producto también reflexionan sobre cómo ha procedido el Sprint hasta ahora y adaptan sus procesos en consecuencia durante una retrospectiva.
6. El equipo repite los pasos 2-5 hasta que el resultado deseado del producto se haya cumplido.

2.9. LENGUAJES DE PROGRAMACIÓN WEB

Un lenguaje de programación es un vocabulario y un conjunto de reglas gramaticales para instruir a una computadora o dispositivo informático para realizar tareas específicas. Dicho de otra manera, un lenguaje de programación es una notación diseñada para conectar instrucciones a una máquina o a un ordenador. Los lenguajes de programación se utilizan principalmente para controlar el rendimiento de una máquina o para expresar algoritmos. En la actualidad, se han implementado miles de lenguajes de programación. En el campo de la informática, muchos lenguajes necesitan ser enunciados en forma imperativa, mientras que otros lenguajes de programación utilizan la forma declarativa. El programa puede dividirse en dos formas, como la sintaxis y la semántica. El término lenguaje de programación suele referirse a los lenguajes de alto nivel, como BASIC, C, C++, COBOL, Java, FORTRAN, Ada y Pascal. Cada lenguaje de programación tiene un conjunto único de palabras clave (palabras que entiende) y una sintaxis especial para organizar las instrucciones de los programas (Beal, 2020).

Los lenguajes de programación pueden ser de diferentes tipos:

- Lenguaje de programación de procedimientos. El lenguaje de programación de procedimientos se utiliza para ejecutar una secuencia de declaraciones que conducen a un resultado. Normalmente, este tipo de lenguaje de programación utiliza múltiples variables, bucles pesados y otros elementos, lo que los separa de los lenguajes de programación funcionales. Las funciones del lenguaje

procedimental pueden controlar variables, aparte de los retornos de valor de la función. Por ejemplo, la impresión de información.

- Lenguaje de programación funcional. El lenguaje de programación funcional utiliza típicamente datos almacenados, evitando frecuentemente los bucles en favor de las funciones recursivas. La programación funcional se centra principalmente en los valores de retorno de las funciones, y los efectos secundarios y diferentes sugieren que el estado de almacenamiento se desaconseja poderosamente. Por ejemplo, en un lenguaje útil sumamente puro, si se denomina una función, se espera que ésta no modifique o realice ningún o/p. Sin embargo, puede construir llamadas algorítmicas y alterar los parámetros de esas llamadas. Los lenguajes funcionales suelen ser más fáciles y construirse más fácilmente para figurar en cuestiones abstractas, sin embargo, estarán incluso "más lejos de la máquina" en ella su modelo de programación hace difícil conocerlo con precisión, pero el código se decodifica en lenguaje de máquina (que a menudo son problemáticos para la programación de sistemas).
- Lenguaje de programación orientado a objetos. Este lenguaje de programación ve el mundo como un grupo de objetos que tienen datos internos y partes de acceso externo a esos datos. El objetivo de este lenguaje de programación es pensar en la falla separándola en una colección de objetos que ofrecen servicios que pueden ser usados para resolver un problema específico. Uno de los principios fundamentales del lenguaje de programación orientado a objetos es la encapsulación de que todo lo que un objeto necesitará debe estar dentro del objeto. Este lenguaje también hace hincapié en la reutilización a través de la herencia y la capacidad de difundir las implementaciones actuales sin tener que cambiar mucho código mediante el uso del polimorfismo.
- Lenguaje de programación de scripts. Estos lenguajes de programación suelen ser de procedimiento y pueden comprender elementos de lenguaje orientados a objetos, pero entran en su propia categoría, ya que normalmente no son lenguajes de programación propiamente dichos con apoyo para el desarrollo de grandes sistemas. Por ejemplo, es posible que no dispongan de verificación

del tipo de tiempo de compilación. Por lo general, estos lenguajes requieren una sintaxis diminuta para su puesta en marcha.

- Lenguaje de programación lógica. Este tipo de lenguajes permiten a los programadores hacer declaraciones y luego permiten a la máquina razonar sobre las consecuencias de esas declaraciones. En cierto sentido, este lenguaje no le dice a la computadora cómo hacer algo, sino que emplea restricciones sobre lo que debe considerar hacer.

Los lenguajes de programación se usan para controlar el rendimiento de la computadora o máquina. En la actualidad, el programador de computadoras tiene muchas opciones para elegir el lenguaje, pero hay muchas diferencias entre los lenguajes de programación y de acuerdo con Eastwood (2020) y Veeraraghavan (2020), los más populares entre los desarrolladores son Python, C, C#, C++, Java y PHP.

Por su parte, la programación web se refiere a la escritura, el marcado y la codificación que intervienen en el desarrollo de la web, que incluye el contenido web, la escritura de clientes y servidores web y la seguridad de la red. Los lenguajes más comunes utilizados para la programación de la Web son XML, HTML, JavaScript, Perl 5 y PHP. La programación de la Web es diferente de la mera programación, que requiere conocimientos interdisciplinarios sobre el área de aplicación, la escritura de clientes y servidores y la tecnología de las bases de datos.

- Python, es uno de los lenguajes de programación más utilizados hoy en día y es un lenguaje fácil de aprender para los principiantes debido a su legibilidad. Es un lenguaje de programación libre y de código abierto con amplios módulos de apoyo y desarrollo comunitario, fácil integración con los servicios de la web, estructuras de datos fáciles de usar y aplicaciones de escritorio basadas en la interfaz gráfica de usuario. Es un lenguaje de programación popular para el aprendizaje automático y las aplicaciones de aprendizaje profundo. Python se utiliza para desarrollar imágenes 2D y paquetes de animación 3D como Blender, Inkscape y Autodesk. También se ha usado para crear videojuegos populares, incluyendo Civilization IV, Vegas Trike, y Toontown. Python se

utiliza para aplicaciones científicas y computacionales como FreeCAD y Abacus, y también en sitios web populares como YouTube, Quora, Pinterest, e Instagram (Veeraraghavan, 2020).

- Java, es un lenguaje multiplataforma que es particularmente útil en la creación de redes. Por supuesto, la mayor parte de este lenguaje se utiliza en la web con los applets de Java. Sin embargo, este lenguaje se usa para diseñar programas multiplataforma, ya que es similar al C++ en estructura y sintaxis. Para los programadores de C++, el lenguaje Java es muy fácil de aprender y ofrece algunas ventajas proporcionadas por la programación orientada a objetos. Como la reutilización y puede ser difícil escribir código eficiente en Java. Pero, hoy en día la velocidad del lenguaje Java ha aumentado y la versión 1.5 ofrece algunas buenas características para hacer programas fáciles (Eastwood, 2020).

2.9.1. PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML (Achour et. al., 2020).

En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo". De aquí que el código de PHP esté encerrado entre las etiquetas especiales de comienzo y final <?php y ?> que permiten entrar y salir del "modo PHP".

Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.

2.9.1.1. Aplicaciones de PHP

PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Aunque PHP puede hacer mucho más.

Existen principalmente tres campos principales donde se usan scripts de PHP (Achour et. al., 2020).

- Scripts del lado del servidor. Este es el campo más tradicional y el foco principal. Son necesarias tres cosas para que esto funcione: el analizador de PHP (módulo CGI o servidor), un servidor web y un navegador web. Es necesario ejecutar el servidor con una instalación de PHP conectada. Se puede acceder al resultado del programa de PHP con un navegador, viendo la página de PHP a través del servidor. Todo esto se puede ejecutar en su máquina si está experimentando con la programación de PHP. Véase la sección sobre las instrucciones de instalación para más información.
- Scripts desde la línea de comandos. Se puede crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Solamente es necesario el analizador de PHP para utilizarlo de esta manera. Este tipo de uso es ideal para scripts que se ejecuten con regularidad empleando cron (en *nix o Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden usarse para tareas simples de procesamiento de texto. Véase la sección Uso de PHP en la línea de comandos para más información.
- Escribir aplicaciones de escritorio. Probablemente PHP no sea el lenguaje más apropiado para crear aplicaciones de escritorio con una interfaz gráfica de usuario, pero si se conoce bien PHP, y se quisiera utilizar algunas características avanzadas de PHP en aplicaciones del lado del cliente, se puede utilizar PHP-GTK para escribir dichos programas. También es posible de esta manera escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal. Si está interesado en PHP-GTK, puede visitar su propio sitio web.

Scripts del lado del servidor. Este es el campo más tradicional y el foco principal. Son necesarias tres cosas para que esto funcione: el analizador de PHP (módulo CGI o servidor), un servidor web y un navegador web. Es necesario ejecutar el servidor con una instalación de PHP conectada. Se puede acceder al resultado del programa de PHP con un navegador, viendo la página de PHP a través del servidor. Todo esto se puede ejecutar en su máquina si está experimentando con la programación de PHP. Véase la sección sobre las instrucciones de instalación para más información.

Scripts desde la línea de comandos. Se puede crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Solamente es necesario el analizador de PHP para utilizarlo de esta manera. Este tipo de uso es ideal para scripts que se ejecuten con regularidad empleando cron (en *nix o Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden usarse para tareas simples de procesamiento de texto. Véase la sección Uso de PHP en la línea de comandos para más información.

Escribir aplicaciones de escritorio. Probablemente PHP no sea el lenguaje más apropiado para crear aplicaciones de escritorio con una interfaz gráfica de usuario, pero si se conoce bien PHP, y se quisiera utilizar algunas características avanzadas de PHP en aplicaciones del lado del cliente, se puede utilizar PHP-GTK para escribir dichos programas. También es posible de esta manera escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal. Si está interesado en PHP-GTK, puede visitar su propio sitio web.

2.9.2. JavaScript

JavaScript es un lenguaje de programación que añade interactividad y comportamientos personalizados a sitios Web. Es un lenguaje de scripting del lado del cliente, lo que significa que se ejecuta en la máquina del usuario y no en el servidor, como lo hacen otros lenguajes de programación web como PHP y Ruby. Eso significa que JavaScript (y la forma en que se usa) depende de las capacidades y ajustes del navegador. Puede que ni siquiera esté disponible, ya sea porque el

usuario ha elegido apagarlo o porque el dispositivo no lo soporta, lo cual los buenos desarrolladores tienen en cuenta y planifican (Niederst, 2018).

JavaScript es también lo que se conoce como un lenguaje de programación dinámico y de escritura libre. Como lenguaje de programación dinámico, JavaScript no necesita ser ejecutado a través de ningún tipo de compilador que interprete el código legible para los humanos en algo que el navegador pueda entender. El navegador lee efectivamente el código de la misma manera que un usuario y lo interpreta sobre la marcha (Niederst, 2018).

Javascript también está escrito de forma suelta. Todo esto significa que no necesariamente hay que decirle a JavaScript qué es una variable. Si establecemos una variable con un valor de 5, no hay que especificar programáticamente esa variable como un número; es decir, 5 es un número, y JavaScript lo reconoce como tal.

En resumen, JavaScript permite crear interfaces de gran capacidad de respuesta que mejoran la experiencia del usuario y proporcionan una funcionalidad dinámica, sin esperar a que el servidor cargue una nueva página. Por ejemplo, podemos usar JavaScript para hacer cualquiera de las siguientes acciones:

- Sugerir el término completo que un usuario podría estar introduciendo en un cuadro de búsqueda mientras escribe.
- Solicitar contenido e información del servidor e injectarlo en el documento actual según sea necesario, sin recargar la página entera. Esto se conoce comúnmente como "Ajax".
- Mostrar y ocultar el contenido basado en que el usuario haga clic en un enlace o encabezamiento, para crear un área de contenido "plegable"
- Probar las características y capacidades individuales de los navegadores. Por ejemplo, se puede comprobar la presencia de "eventos táctiles", que indican que el usuario está interactuando con la página a través del navegador de un dispositivo móvil, y añadir estilos y métodos de interacción más fáciles de tocar.

- Llenar los vacíos en los que la funcionalidad incorporada de un navegador se queda corta, o añadir algunas de las características que se encuentran en los navegadores más nuevos a los navegadores más antiguos. Este tipo de scripts se suelen denominar "shims" o "polyfills".
- Cargar una imagen o contenido en una "caja de luz" personalizada -aislada en la página con CSS- después de que un usuario hace clic en una versión en miniatura de la imagen.

2.10. GESTIÓN DE CALIDAD DE SOFTWARE

Actualmente, la gente está muy interesada en la calidad de los productos o servicios. En el mercado altamente competitivo de hoy, no es suficiente producir y distribuir productos o servicios. Las ventas son importantes, pues allí se traduce la aprobación del cliente. Se dice que la calidad no tiene concepto, solo aprobación. Sin embargo, la calidad del software es un concepto complejo y no se puede comparar directamente con la calidad del producto. El software se ha convertido en uno de los principales objetivos estratégicos de la organización, porque el proceso más importante de la organización y su supervivencia dependen de la función del software todos los días.

De aquí que calidad del software se puede ver en la función o atributo. Como atributos, la calidad se refiere a características medibles, es decir, cosas comparables a los estándares, como longitud, color, características eléctricas y ductilidad. Sin embargo, el software como entidad de conocimiento tiene la complejidad de caracterizar objetos físicos. Sin embargo, algunas medidas nos permiten evaluar las características del programa. Estas características incluyen complejidad mental y física, puntos de función, líneas de código, etc. (Enrique, 2017). Cuando se examina un elemento sus características mensurables se pueden encontrar dos tipos de calidad:

- Calidad de diseño; la calidad de diseño se refiere a las características que los diseñadores especifican para un elemento.
- Calidad de concordancia; la calidad de concordancia es el grado en el que las especificaciones de diseño se aplican durante la fabricación.

Actualmente existen en el mercado “normativo” diversas opciones, los estándares y modelos de evaluación y mejora de los procesos de software que están relacionados con la calidad (Ramos, 2013). Es este contexto, las normas ISO, adquieren gran relevancia, toda vez que ISO es un marco de trabajo que permite gestionar, controlar y mejorar a una organización, con el objetivo de mejorar la calidad de los protocolos y servicios que produce.

2.10.1. Normas ISO

La Organización Internacional para la Estandarización, mejor conocida como ISO, es la agencia especializada en estandarización, conformada por representantes de los cuerpos normalizadores, fue establecida oficialmente el 23 de febrero de 1947 con el objeto de promover la estandarización internacional, de tal manera que se facilitara el intercambio internacional de bienes y servicios casi como el desarrollo científico y tecnológico. Actualmente abarca los estándares nacionales de 91 países. En los Estados Unidos, la representación se llama The American National Standards Institute o ANSI (Enrique, 2017).

- ISO 25000:2005 (SQuaRE -Software Quality Requirements and Evaluation) es una nueva serie de normas que se basa en ISO 9126 y en ISO 14598 (Evaluación del software). Uno de los principales objetivos de la serie SQuaRE es la coordinación y harmonización del contenido de ISO 9126 y de ISO 15939:2002 (Measurement Information Model). ISO 15939 tiene un modelo de información que ayuda a determinar que se debe especificar durante la planificación, performance y evaluación de la medición. Para su aplicación, cuenta con los siguientes pasos: Recopilar los datos, Preparación de los datos y Análisis de los datos. (ISO, 2017)

El objetivo de la calidad del software es proporcionar información adecuada sobre los datos necesarios referentes a la calidad del producto. En esta fase es deseable que las métricas técnicas proporcionen una visión interna a la calidad del modelo de análisis. Estas métricas examinan el modelo de análisis con la intención de predecir el tamaño del sistema resultante.

La medición de la calidad sobre aspectos del software se los realiza a través de métricas de control de calidad, para medir aspectos del software como funcionalidad, mantenibilidad y portabilidad.

En este documento se acude al desarrollo de la medición de la calidad del software mediante la métrica ISO-25000 lo cual es un estándar internacional para la evaluación de software, que establece que cualquier componente de la calidad pueda ser descrito por las características de Funcionalidad, Confiabilidad, Mantenibilidad y Portabilidad.

2.10.2. ISO/IEC 25000 (SQuaRE)

La familia de normas ISO/IEC 25000 surge para sustituir a las antiguas ISO/IEC 9126 e ISO/IEC 14598, unificando el contenido de estas y definiendo a lo largo de sus distintas partes:

- **UN MODELO DE CALIDAD:** con las características y sub características de calidad que se pueden evaluar de un producto software.
- **MÉTRICAS Y REQUISITOS DE CALIDAD:** que se pueden aplicar al producto software.
- **UN PROCESO DE EVALUACIÓN:** a seguir para determinar la calidad de los productos software.

Cumpliendo con los requisitos de esta familia de normas, el presente proyecto de grado realiza un proceso de evaluación compuestos por cinco actividades:

- **ACTIVIDAD 1** Establecer los requisitos de Evaluación. Cuyo objetivo es definir el propósito de la evaluación, los requisitos de calidad que se deben considerar, las partes involucradas y el rigor de la evaluación.
- **ACTIVIDAD 2** Especificar la Evaluación, cuyo objetivo es determinar las métricas, técnicas y herramientas que se utilizaran para llevar a cabo la evaluación.
- **ACTIVIDAD 3** Diseñar la Evaluación, cuyo fin es definir el plan con las actividades de evaluación que se deben llevar a cabo.
- **ACTIVIDAD 4** Ejecutar la Evaluación, cuya meta es obtener las mediciones y aplicar los criterios de evaluación determinados en las actividades anteriores

- ACTIVIDAD 5 Concluir la Evaluación, que finalmente permite analizar los resultados y elaborar un informe descriptivo para que la organización evaluada conozca la calidad de su producto software.

2.11. ESTIMACIÓN DE COSTOS DE DESARROLLO DE SOFTWARE

La evaluación del costo de desarrollo del programa es un factor muy importante en el análisis de proyectos de TI, ya que es estratégico contar con indicadores para medir su costo y garantizar eficiencia, excelencia, calidad y competitividad. El análisis de costos es el proceso de determinar los recursos necesarios para llevar a cabo un trabajo o proyecto de manera efectiva. El costeo determina la calidad y cantidad de recursos necesarios en función del dinero, la energía, las habilidades, el conocimiento y el tiempo, lo que influye en la gestión empresarial. Actualmente, existen muchos indicadores no utilizados que se pueden aplicar a cualquier tipo de proyecto de software para calcular su costo (Del Valle Roque, 2014).

Adicionalmente, el costo del desarrollo de programas informáticos depende de una variedad de factores diferentes. La complejidad y el tamaño del proyecto, la tecnología utilizada e incluso la ubicación geográfica de los desarrolladores. Todos estos aspectos se reflejan en el precio final (Kurilo, 2020). Algunos factores que pueden incidir en el costo del proyecto son la complejidad del proyecto, el tamaño, el diseño, plataformas soportadas, tecnología, equipo de desarrollo, localización, mantenimiento, etc. Por lo tanto es conveniente emplear algún tipo de criterio para estimar tales costos.

Como resultado, al predeterminar el alcance del proyecto, utilizar los indicadores del software como base para la evaluación y desglosar el proyecto en partes más pequeñas que pueden evaluarse individualmente, se desarrollaron varias técnicas de evaluación para el desarrollo de software, tales como COCOMO II, CoCots, CoStar, CostModeler, CostXpert, SoftCost. Y mientras estos instrumentos de estimación fueron desarrollados por empresas diferentes, existen otras formas de estimación tales como la opinión de expertos, técnica Delphi, Analogía.

2.11.1. Método COSMIC

El Common Software Measurement International Consortium o COSMIC es un método de análisis de puntos de función de segunda generación, en el cual se determina el tamaño funcional del software a partir del número de interacciones entre los procesos funcionales. El método IFPUG (previo a COSMIC) depende de los “tipos de funciones” definidos por Albrecht, los cuales eran adecuados para la década de los 70, pero se ha hecho difícil asignárselos a formas modernas de modelar los requerimientos de software, como por ejemplo cuando el software se construye como servicios (SOA) y en áreas como el software de tiempo real o de infraestructura. Para los años 90 la industria estaba demandando un método de análisis de puntos de función estándar, sin embargo, no existía acuerdo para seleccionar alguno de los métodos de medición que existían. El principal cambio de COSMIC frente a IFPUG, es que en lugar de clasificar las funcionalidades y asignar una cantidad predeterminada de puntos de función a cada tipo, se enfoca en contar las interacciones entre funcionalidades y de estas con archivos de datos (Symons y Lesterhuis, 2017).

Como consecuencia, el método COSMIC define los principios, reglas y un proceso para medir un tamaño funcional estándar de una pieza de software. El "tamaño funcional" es una medida de la cantidad de funcionalidad expresada en términos que un usuario puede entender. Es completamente independiente de la tecnología utilizada para crearlo. El tamaño del COSMIC suele determinarse a partir de los requisitos, pero también puede calcularse de forma inversa a partir de otros artefactos de software, por ejemplo, diseños, sistemas instalados, etc. (COSMIC, 2019).

2.11.2. Proceso COSMIC

El método de medición consiste en un conjunto de modelos, principios, reglas y procedimientos que se aplican para determinar el valor de una magnitud para la funcionalidad entregada por el software. Dicha magnitud de la funcionalidad del software es expresada en puntos de función COSMIC (Vasquez, 2018). A muy grandes rasgos, el proceso consiste en:

- Establecer la frontera entre el sistema y los actores con los que interactúa (que pueden ser personas u otros sistemas).
- Identificar los procesos funcionales que los actores reciben del sistema.
- Para cada proceso funcional, identificar los movimientos de datos que genera. Cada que el usuario provee al sistema un conjunto de datos para realizar un proceso, hay un movimiento de datos. También es un movimiento de datos cuando el sistema provee grupos de datos al usuario, y cuando se acceden o almacenan grupos de datos en almacenamiento. Cada movimiento de datos contribuye con el equivalente a un punto de función COSMIC.

COSMIC tiene la ventaja de no establecer límites arbitrarios al tamaño funcional, así puede medir componentes de software muy grandes o pequeños. Adicionalmente, está basado en el desglose funcional de los componentes de software, alineado con las prácticas de Ingeniería de software. Paralelamente, COSMIC fue diseñado para trabajar con requisitos funcionales en cualquier capa de la arquitectura de software y en cualquier grado de desglose de componentes, por lo que el proceso de medición COSMIC, consta de tres fases (Symons y Lesterhuis, 2016).

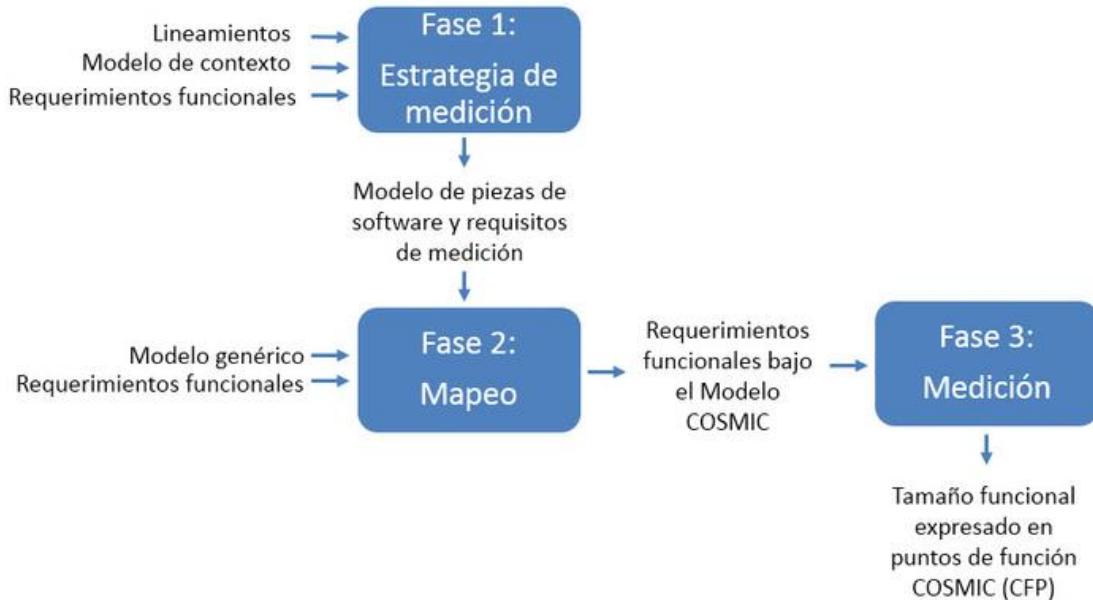


Figura 7. Proceso de medición COSMIC

Fuente: Symons y Lesterhuis (2016)

2.11.2.1. Fase 1: Estrategia de medición

Lo primero que se realiza en una medición y estimación de software con COSMIC, es determinar qué es lo que se va a medir.

Una medición de software depende del punto de vista de lo que definimos como usuarios funcionales, por ejemplo, personas, dispositivos de hardware u otros sistemas que interactúan con el software.

En esta primera fase se define el propósito y alcance de la medición de software, que incluye cuales son los requerimientos funcionales de usuario que se van a medir, quienes son los usuarios funcionales y otros parámetros. Previo a esto, es necesario haber aplicado técnicas para el levantamiento de requerimientos de software.

Es importante dejar documentados los parámetros de la medición de software, para asegurar que esta pueda ser interpretada adecuadamente por quienes harán uso de ella para realizar las estimaciones y presupuestos.

2.11.2.2. Fase 2: Mapeo

En una medición COSMIC, el mapeo se realiza para crear un modelo COSMIC de los requerimientos funcionales de usuario. El punto de partida para el mapeo son los artefactos disponibles, como por ejemplo un esquema o especificación de requerimientos detallada, modelos de diseño como por ejemplo los casos de uso, software que está instalado físicamente, entre otros.

Para elaborar este modelo, se utilizan los principios del Modelo genérico de software COSMIC, aplicados a los requerimientos de software que se van a medir (Symons y Lesterhuis, 2016). El modelo de requerimientos de software COSMIC tiene 4 principios:

1. La funcionalidad de software está comprendida de procesos funcionales. La tarea de cada proceso funcional es responder a un evento ocurrido fuera de la frontera del sistema (el mundo de los usuarios funcionales).
2. Los procesos funcionales están compuestos de sub-procesos:
 - a) Cada sub-proceso puede mover datos o manipular datos.

- b) Los sub-procesos de movimiento de datos que mueven datos de un usuario funcional a un proceso funcional se les llama "Entradas".
- c) Los sub-procesos que mueven datos desde un proceso funcional hacia el exterior se les llama salidas.
- d) Los sub-procesos que mueven datos hacia un almacén de datos se les llama "Escrituras" mientras que a los que mueven datos desde dichos almacenes se les conoce como "lecturas".
3. Cada movimiento de datos (Entrada, salida, lectura o escritura) moviliza solamente un grupo de datos, cuyos atributos describen un solo objeto de interés.
4. Se asume que la manipulación de datos forma parte de las entradas, salidas, lecturas o escrituras, por lo tanto, estas no se miden por separado (En la medición solo se cuentan los movimientos de datos).

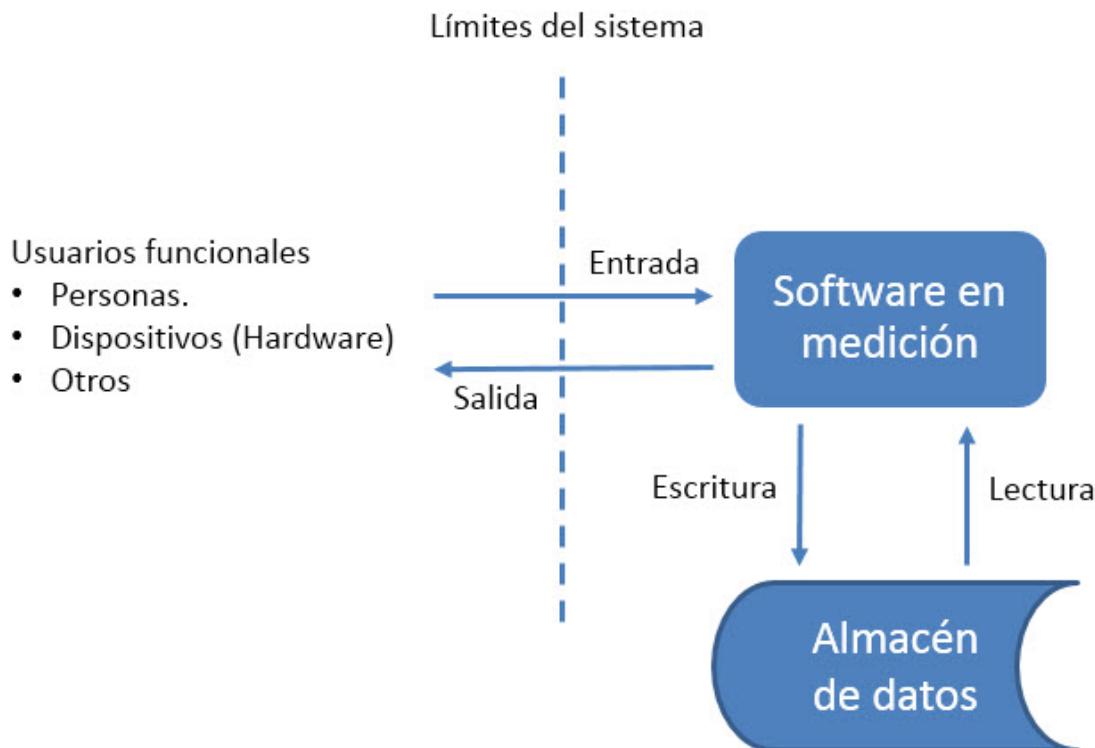


Figura 8. Tipos de sub-procesos definidos por el modelo COSMIC
Fuente: Symons y Lesterhuis (2016)

Se entiende que un proceso funcional termina su ejecución cuando ha realizado todos los sub-procesos necesarios para responder a los datos que recibió del evento.

2.11.2.3. Fase 3: Medición

La unidad de medida del método COSMIC es el “punto de función COSMIC” (CFP). Cada movimiento de datos es medido como un (Decreto Supremo N° 156) CFP.

La medición de la nueva pieza de software se realiza identificando todos los movimientos de datos, es decir todas las entradas, salidas, lecturas y escrituras de cada proceso funcional. Luego sumándolas todas.

Todo proceso funcional debe tener al menos dos movimientos de datos (al menos una entrada y una salida o una escritura). Solo de esta forma se garantiza que el proceso funcional modelado proporciona un servicio completo. Por lo tanto, el tamaño funcional mínimo de un proceso es de 2 CFP. No existe un límite superior al tamaño de un proceso funcional.

Para realizar mediciones sobre mejoras a piezas de software existente, se identifican todos los movimientos de datos que se van a agregar, modificar o eliminar, sumándolos todos en cada uno de sus procesos funcionales. El tamaño mínimo de una modificación es de un CFP.

2.12. SEGURIDAD

2.12.1. Seguridad de base de datos

La seguridad de datos, también conocida como seguridad de la información o seguridad informática, es un aspecto esencial en las organizaciones de cualquier tamaño y tipo. Se trata de un aspecto que tiene que ver con la protección de datos contra accesos no autorizados y para protegerlos de una posible corrupción durante todo su ciclo de vida. Seguridad de datos incluye conceptos como encriptación de datos, tokenización y prácticas de gestión de claves que ayudan a proteger los datos en todas las aplicaciones y plataformas de una organización.

Hoy en día, organizaciones de todo el mundo invierten fuertemente en la tecnología de información relacionada con la ciberdefensa con el fin de proteger sus activos críticos: su marca, capital intelectual y la información de sus clientes.

En todos los temas de seguridad de datos existen elementos comunes que todas las organizaciones deben tener en cuenta a la hora de aplicar sus medidas: las personas, los procesos y la tecnología. (Elle, 2019)

2.12.2. Seguridad lógica

La seguridad lógica de un sistema informático consiste en la aplicación de barreras y procedimientos que protejan el acceso a los datos y a la información contenida en él. El activo más importante de un sistema informático es la información y por tanto, la seguridad lógica se plantea como uno de los objetivos más importantes.

La seguridad lógica trata de conseguir los siguientes objetivos:

- Restringir el acceso a los programas y archivos.
- Asegurar que los usuarios puedan trabajar sin supervisión y no puedan modificar los programas ni los archivos que no correspondan.
- Asegurar que se estén utilizados los datos, archivos y programas correctos en y por el procedimiento correcto.
- Verificar que la información transmitida sea recibida sólo por el destinatario al cual ha sido enviada y que la información recibida sea la misma que la transmitida.
- Disponer de pasos alternativos de emergencia para la transmisión de información.

2.12.3. Seguridad física

La seguridad física de un sistema informático consiste en la aplicación de barreras físicas y procedimientos de control frente a amenazas físicas al hardware.

Este tipo de seguridad está enfocado a cubrir las amenazas ocasionadas tanto por el hombre como por la naturaleza del medio físico en que se encuentra ubicado el sistema. Las principales amenazas que se prevén son:

- Desastres naturales, incendios accidentales y cualquier variación producida por las condiciones ambientales.
- Amenazas ocasionadas por el hombre como robos o sabotajes.
- Disturbios internos y externos deliberados.

Evaluar y controlar permanentemente la seguridad física del sistema es la base para comenzar a integrar la seguridad como función primordial del mismo. Tener controlado el ambiente y acceso físico permite disminuir siniestros y tener los medios para luchar contra accidentes.

2.13. Pruebas de software

2.13.1. Pruebas de caja blanca

El método de pruebas White box (también conocido como clear box testing, glass box testing, transparent box testing, and structural testing) es el cual mira el código y la estructura del producto que se va a probar y usa ese conocimiento para la realización de las pruebas.

Este método se usa en la fase de Unit testing, aunque también puede ocurrir en otras fases como en las pruebas de sistema o de integración. Para la ejecución de este método es necesario que el tester o la persona que valla a usar el método tengan amplios conocimientos de la tecnología y arquitectura usada para desarrollar el programa. (Diamond, 2017)

2.13.2. Pruebas de caja negra

Caja Negra (black box) es el método en el cual el elemento es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. Estas pruebas son realizadas desde la interfaz gráfica. (Diamond, 2017)

2.13.3. Pruebas de carga

La prueba de carga (load testing) nos permite identificar la cantidad de peticiones que un sistema puede soportar. Por ejemplo, un sistema de transporte tiene una cantidad establecida de usuarios que puede movilizar. De manera similar se puede

probar un programa para identificar si se cumple con la cantidad de peticiones concurrentes que se establece en un inicio. (platzi.com, 2018)

2.13.4. Pruebas de estrés

Una prueba de estrés (stress) consiste en probar los límites que un sistema puede soportar. En este tipo de pruebas se suele enviar más peticiones de las que el software podría atender normalmente para saber el comportamiento de la aplicación. (platzi.com, 2018)

2.13.5. Pruebas de resistencia

El test de resistencia (endurance) se hace enviando peticiones a un sistema en ciertos intervalos de tiempo. (platzi.com, 2018)

2.13.6. Pruebas de escalabilidad

La escalabilidad (scalability) es un factor que al evaluar nos permite identificar las mejoras a la infraestructura donde reside nuestra aplicación. Y así podemos repetir las tres pruebas anteriores para conocer cómo cambia el performance. (platzi.com, 2018)

CAPÍTULO III

MARCO APLICATIVO

3. MARCO APLICATIVO

3.1. DIAGRAMAS CASOS DE USO

3.1.1. Casos de uso

Los casos de uso son descripción de una acción o actividad. Un diagrama de caso de uso es una descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso. Los personajes o entidades que participarán en un diagrama de caso de uso se denominan actores. En el contexto de ingeniería del software, un diagrama de caso de uso representa a un sistema o subsistema como un conjunto de interacciones que se desarrollarán entre casos de uso y entre estos y sus actores en respuesta a un evento que inicia un actor principal. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo, la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requisitos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

3.1.2. Diagramas

3.1.2.1. Diagramas administradores

Los administradores tienen privilegios elevados, pueden registrar a usuarios, asignar privilegios a usuarios, registrar a clientes, realizar ventas, registrar categorías, registrar productos, dar de alta y baja a ciertos productos, categorías, usuarios, clientes.

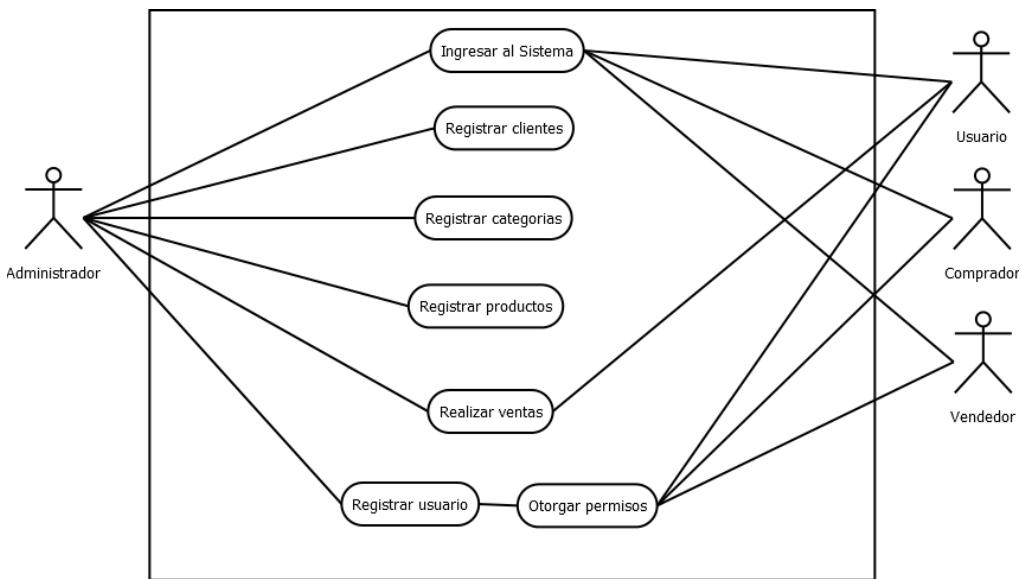


Figura 9. Diagramas administradores

Fuente: Elaboración propia

3.1.2.2. Diagramas compradores

Un comprador tiene privilegios limitados que solo puede hacer ciertas tareas, como ser ingresar al sistema registrar productos, registrar categorías lo cual es evaluado con el administrador.

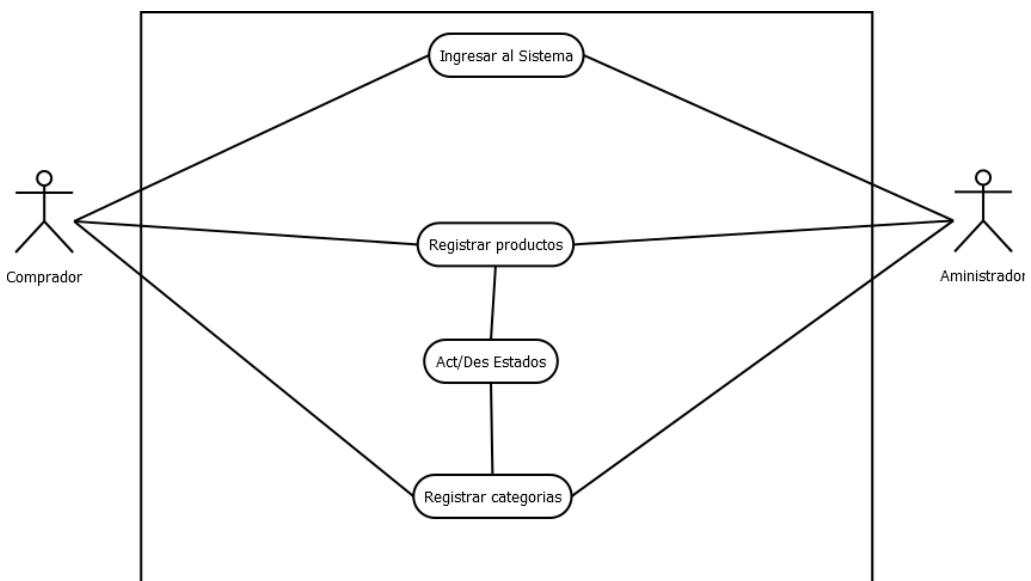


Figura 10. Diagramas compradores

Fuente: Elaboración propia

3.1.2.3. Diagramas vendedores

Los vendedores tienen ciertas tareas lo cual son limitados por los privilegios que el administrador asigna al momento de crear vendedores, cada vendedor puede ingresar al sistema, registrar clientes, dar de alta y baja a clientes, registrar categorías, registrar productos que son supervisados por los administradores, puede realizar ventas a clientes.

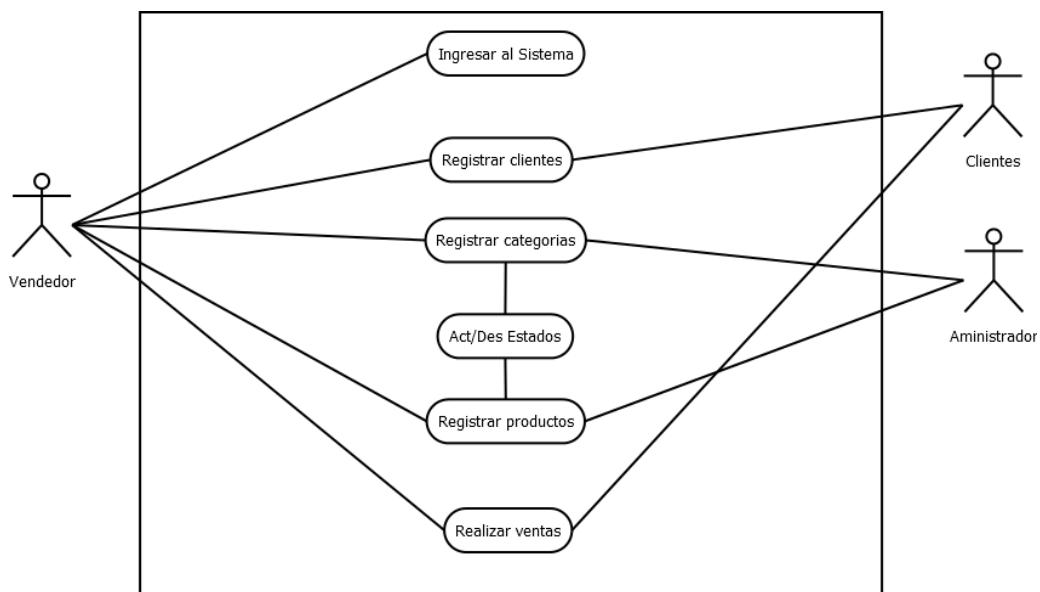


Figura 11. Diagramas vendedores

Fuente: Elaboración propia

3.1.2.4. Diagrama modulo Sistema

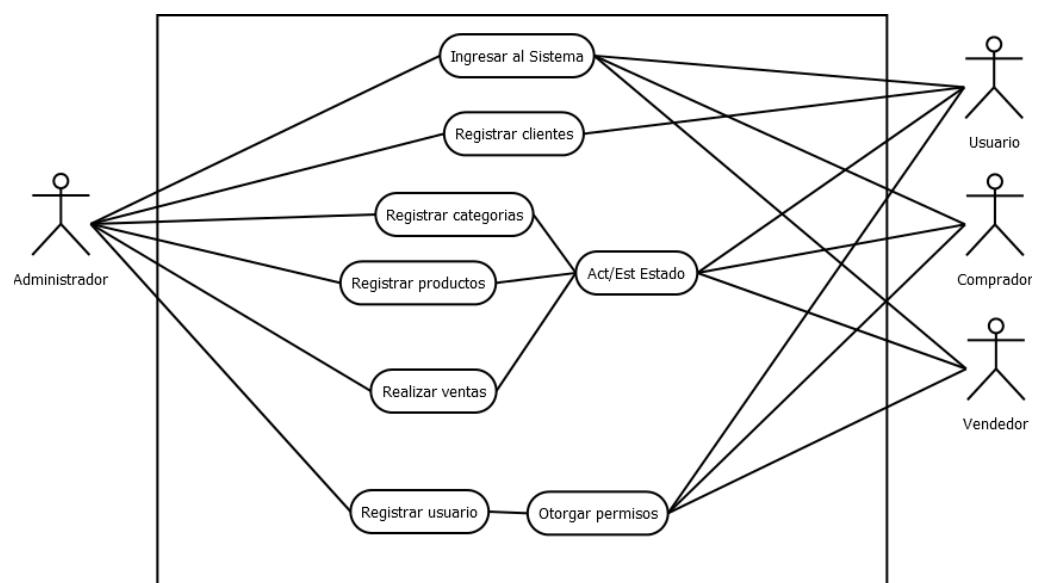


Figura 12. Diagrama modulo Sistema

Fuente: Elaboración propia

3.1.2.5. Diagrama modulo administrador

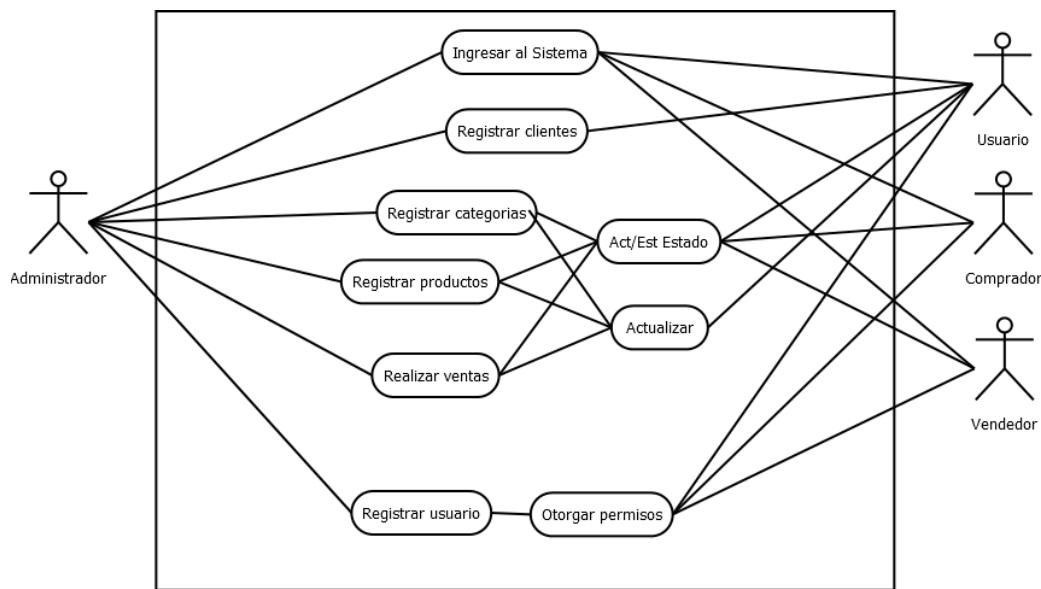


Figura 13. Diagrama modulo administrador

Fuente: Elaboración propia

3.1.2.6. Diagrama modulo Usuario

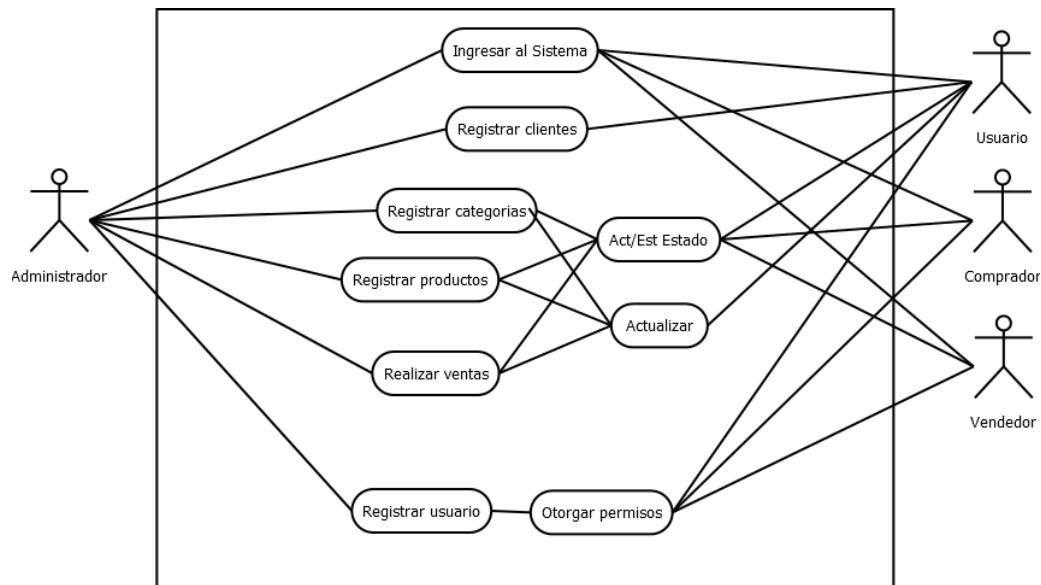


Figura 14. Diagrama modulo Usuario

Fuente: Elaboración propia

3.1.2.7. Diagrama modulo ventas

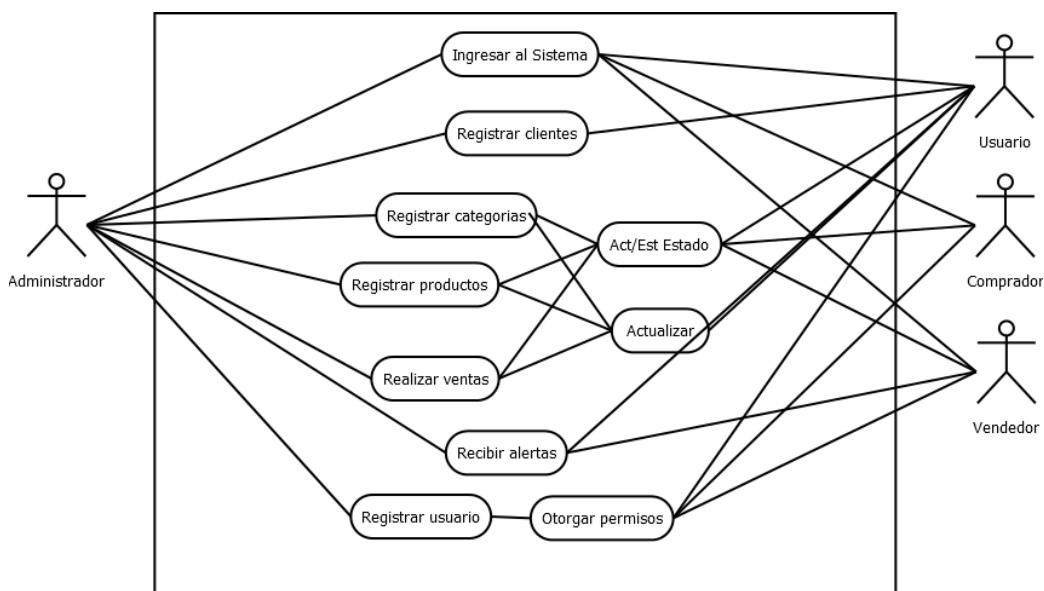


Figura 15. Diagrama modulo ventas

Fuente: Elaboración propia

3.2. ANÁLISIS DE REQUERIMIENTOS (PRODUCT BACKLOG)

Todo lo que el equipo SCRUM ha desarrollado fue en base al Product Backlog el cual fue proporcionado por el Product Owner, siendo este la única fuente de requerimientos para el sistema de información para la administración y control de ventas.

El Product Backlog no es nada más que Historia de Usuarios priorizados que son descritos de forma natural, con el correr del tiempo se va convirtiendo más extenso y concreto.

Tabla 1. Análisis de requerimientos

HISTORIA DE USUARIO

ID	Como	Quiero	Para
H001	Sistema	Registrar datos del cliente	Tener un control de clientes
H002	Sistema	Actualizar datos del cliente	Poder modificar o añadir algún dato que sea necesario

H003	Sistema	Cambiar estado de un servicio	Visualizar el estado
H004	Administrador	Registrar datos de usuario	Tener control de clientes
H005	Administrador	Actualizar datos de usuario	Poder modificar o añadir algún dato que sea necesario
H006	Administrador	Otorgar permisos del sistema a usuario	Administrar acceso del sistema por tipo de usuario
H007	Administrador	Tener control de todos los módulos del sistema	Administrar todos los módulos
H008	Venta	Registrar datos del cliente	Tener control de clientes
H009	Venta	Actualizar datos del cliente	Poder modificar o añadir algún dato que sea necesario
H010	Venta	Registrar datos del producto	Tener un control de productos
H011	Venta	Actualizar datos de producto	Poder modificar algún dato que sea necesario
H012	Venta	Recibir alertas	Para verificar y confirmar
H013	Venta	Cotizar solicitud de cliente	Ampliar información de costo de los productos
H014	Venta	Emitir una orden de venta	Dejar constancia de la orden
H015	Usuario	Iniciar sesión	Poder identificarse como usuario
H016	Usuario	Actualizar usuario	Actualizar usuario

H017	Usuario	Ingresar al sistema por una Tablet	Aumenta la productividad
H018	Usuario	Ingresar al sistema desde cualquier localización	Ingresar los datos a una base de datos

Fuente: Elaboración propia

3.2.1. Refinamiento del Product Backlog

En esta actividad el beneficio clave es la preparación de los Sprints subsiguientes, por lo que el equipo SCRUM en conjunto incluye, pero no se limita a:

- Mantener el product backlog ordenado
- Eliminar o degradar ítems que ya no sean importantes
- Agregar o promover ítems que surgen o se vuelven importantes
- Dividir ítems en ítems más pequeños
- Unir ítems en ítems más grandes
- Estimar ítems

3.2.2. Identificación de usuarios

Tabla 2. Identificación de usuarios

Product Owner	Simon Callizaya
Scrum master	Jimmy Mayta
Equipo de desarrollo	Jimmy Mayta

Fuente: Elaboración propia

3.2.3. Definición de roles por actor

Tabla 3. Definición de roles por actor

ROL	PERSONA A CARGO	DESCRIPCIÓN
-----	-----------------	-------------

Product Owner	Simon Callizaya	Responsable de maximizar el valor del producto desarrollado por el Equipo de Desarrollo
---------------	-----------------	---

Scrum master	Jimmy Mayta	Guía de equipo en la metodología scrum
--------------	-------------	--

Equipo de desarrollo	Jimmy Mayta	Encargados de diseñar y desarrollar la aplicación
----------------------	-------------	---

Fuente: Elaboración propia

Los roles se definieron en base a las capacidades de cada persona: Jimmy Mayta y Simon Callizaya se ha desenvuelto como Product Owner debido a su conocimiento en el campo de accesibilidad y comprensión de los requerimientos de usuarios, asimismo el rol de Scrum Master ha estado a cargo de Jimmy Mayta, debido a su especialización en la metodología Scrum, por lo cual dirige y apoya al equipo en el uso de la metodología. Como equipo de desarrollo únicamente participo Jimmy Mayta, encargado para el diseño y desarrollo de la aplicación, así como la ejecución de las pruebas, teniendo en consideración de que la cantidad de participantes pueda incrementarse en el futuro.

3.2.4. Requerimientos de usuario

Tabla 4. Requerimientos de usuario

Product Owner	Simon Callizaya	Conocimientos en área de ventas, compras, administración, desarrollo
Scrum master	Jimmy Mayta	Conocimientos en desarrollo
Equipo de desarrollo	Jimmy Mayta	Conocimientos en desarrollo

Fuente: Elaboración propia

3.3. PLANIFICACIÓN DEL SPRINT

Antes de comenzar con el desarrollo en cada Sprint, el equipo Scrum se reunió para seleccionar y comprender el trabajo que será realizado.

El equipo de desarrollo y el product owner negociaron para llegar a un acuerdo compartido respecto a qué trabajos serán realizados y en qué tiempo se finalizará, tomando en cuenta el performance del equipo de desarrollo y la cantidad de tareas que tiene cada ítem del product backlog. Tomando en cuenta la auto organización del equipo de desarrollo, se propusieron y establecieron los siguientes Sprints.

3.3.1. Sprint 1

En este primer Sprint se estimó cinco semanas de desarrollo para los siguientes ítems del Product Backlog:

Tabla 5. Sprint 1

ID	COMO...	QUIERO...	PARA...
H001	Sistema	Crear BD	BD sistema
H002	Usuario	Iniciar sesión	Poder identificarse como usuario
H003	Usuario	Ingresar al sistema por una Tablet	Aumenta la productividad
H004	Administrador	Registrar datos de usuario	Tener un control de clientes
H005	Administrador	Otorgar permisos del sistema a usuario	Administrador de accesos del sistema por tipo de usuario
H006	Administrador	Tener un control de todos los nodulos	Administrador todos los nodulos del sistema
H007	Administrador	Actualizar datos de usuario	Poder identificarse o añadir algún

				dato que sea necesario
H008	Sistema	Registrar datos del cliente	Tener control de clientes	
H009	Venta	Registrar datos del cliente	Tener un control de clientes	
H010	Sistema	Actualizar datos del cliente	Poder modificar o añadir algún dato que sea necesario	
H011	Venta	Actualizar datos del cliente	Poder modificar o añadir algún dato que sea necesario	

Fuente: Elaboración propia

3.3.2. Sprint 2

En este segundo Sprint se estimó cuatro semanas de desarrollo para los siguientes ítems:

Tabla 6. Sprint 2

ID	COMO...	QUIERO...	PARA...
H001	Sistema	Registrar datos	Tener un control de diagnóstico por cliente
H002	Sistema	Actualizar datos	Poder identificar algún dato que sea necesario
H003	Sistema	Verificar cliente	Verificar cliente estado de cliente

H004	Venta	Registrar datos del producto	Tener un control de productos
H005	Venta	Actualizar datos del producto	Poder modificar algún dato que sea necesario
H006	Venta	Cotizar solicitud de cliente	Ampliar información de costo de los productos en local

Fuente: Elaboración propia

3.3.3. Sprint 3

Para el tercer Sprint se estimó tres semanas de desarrollo para los siguientes ítems:

Tabla 7. Sprint 3

ID	COMO...	QUIERO...	PARA...
H001	Sistema	Programar f/h de entrega	Especificar y tener un estado
H002	Venta	Venta alertas	Recibir Poder verificar y confirmar entregas
H003	Usuario	Actualizar usuario	Actualizar usuario
H004	Usuario	Ingresar sistema desde cualquier localización	Ingresar los datos a una base de datos

Fuente: Elaboración propia

3.4. DESARROLLO DEL SISTEMA

Se presenta el Taskboard de desarrollo inicial del proyecto con todas las historias y la condición inicial de cada uno de los Sprint.

Tabla 8. Desarrollo del sistema

INICIO	3/8/2020	NOMBRE	DESARROLLO DEL SISTEMA
Fin	30/10/2020	Semana	1
		Historia de Pendiente usuario	En curso Hecho
Sprint 1		Crear Base de datos	✓
		Iniciar sesión	✓
		Ingresar al sistema por Tablet	✓
		Registrar datos de usuario	✓
		Otorgar permisos del sistema a usuario	✓
		Tener un control de los módulos del sistema	✓
		Actualizar datos de usuario	✓

	Registrar	✓
	datos del cliente	
	Actualizar	✓
	datos del cliente	
Sprint 2	Registrar	✓
	datos	
	Actualizar	✓
	datos	
	Verificar	✓
	cliente	
	Registrar	✓
	datos del producto	
	Actualizar	✓
	datos del producto	
	Cotizar	✓
	solicitud de cliente	
Sprint 3	Programar	✓
	fecha y hora de entrega	
	Venta recibir	✓
	alerta	

Actualizar ✓
usuario

Ingresar al ✓
sistema desde
cualquier
localización

Fuente: Elaboración propia

3.4.1. Semana 1

Se muestra el Taskboard de la Semana 1 en donde, en el Sprint 1 y la historia de usuario “Creación de Base de Datos” se encuentra en curso.

Tabla 9. Semana 1

INICIO	3/8/2020	NOMBRE	DESARROLLO DEL SISTEMA
Fin	30/10/2020	Semana	1
		Historia de Pendiente usuario	En curso Hecho
Sprint 1	Crear Base de datos		✓
		Iniciar sesión ✓	
		Ingresar al ✓ sistema por Tablet	
		Registrar ✓ datos de usuario	
		Otorgar ✓ permisos del	

	sistema	a
	usuario	
	Tener	un ✓
	control de los	
	módulos del	
	sistema	
	Actualizar	✓
	datos	de
	usuario	
	Registrar	✓
	datos	del
	cliente	
	Actualizar	✓
	datos	del
	cliente	
Sprint 2	Registrar	✓
	datos	
	Actualizar	✓
	datos	
	Verificar	✓
	cliente	
	Registrar	✓
	datos	del
	producto	
	Actualizar	✓
	datos	del
	producto	

	Cotizar	✓
	solicitud de	
	cliente	
Sprint 3	Programar	✓
	fecha y hora	
	de entrega	
	Venta recibir	✓
	alerta	
	Actualizar	✓
	usuario	
	Ingresar al	✓
	sistema desde	
	cualquier	
	localización	

Fuente: Elaboración propia

3.4.1.1. Burn Down Chart semana 1

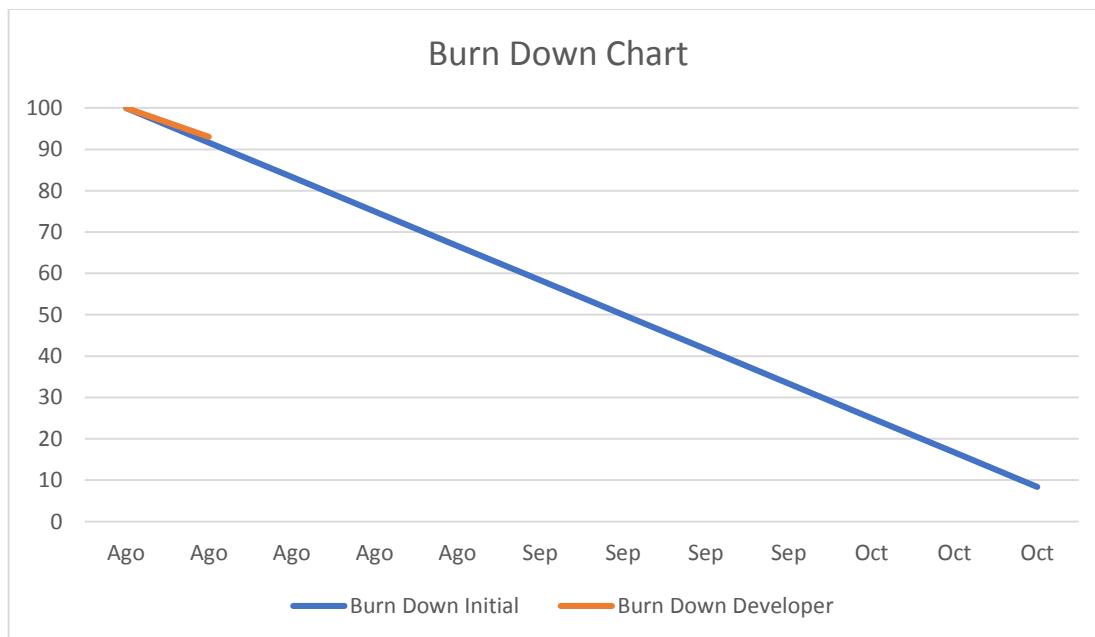


Figura 16. Burn Down Chart semana 1

Fuente: Elaboración propia

Base de datos

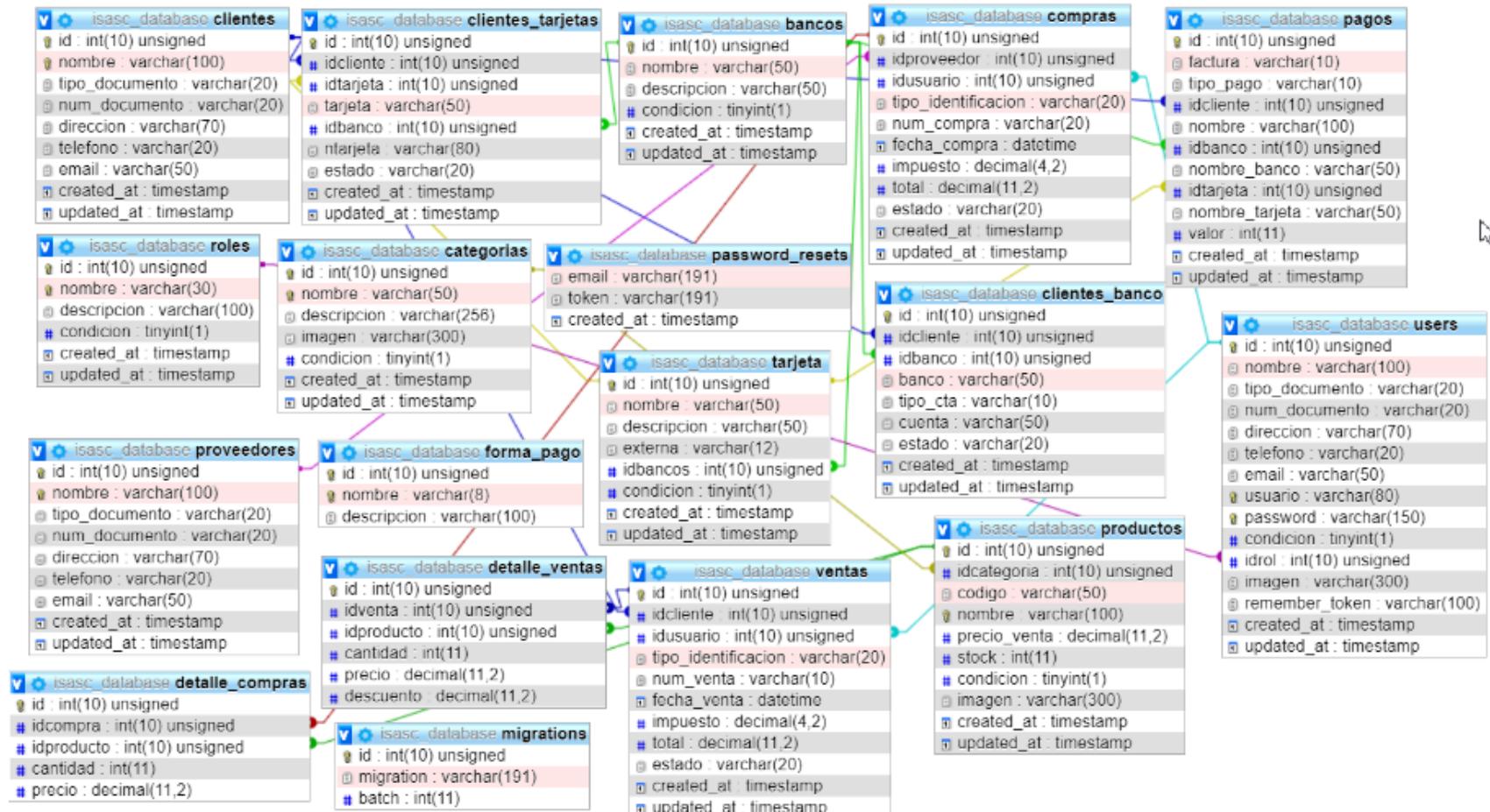


Figura 17. Base de datos
Fuente: Elaboración propia

3.4.2. Semana 2

Se muestra el Taskboard de la Semana 2 en donde, en el Sprint 1 ya se terminó e inicia “Iniciar sesión” y “Ingresar al sistema por Tablet” se encuentra en curso.

Tabla 10. Semana 2

INICIO	3/8/2020	NOMBRE	DESARROLLO DEL SISTEMA
Fin	30/10/2020	Semana	2
		Historia de Pendiente usuario	En curso Hecho
Sprint 1		Crear Base de datos	✓
		Iniciar sesión	✓
		Ingresar al sistema por Tablet	✓
		Registrar datos de usuario	✓
		Otorgar permisos del sistema a usuario	✓
		Tener un control de los módulos del sistema	✓

	Actualizar	✓
	datos de	
	usuario	
	Registrar	✓
	datos del	
	cliente	
	Actualizar	✓
	datos del	
	cliente	
Sprint 2	Registrar	✓
	datos	
	Actualizar	✓
	datos	
	Verificar	✓
	cliente	
	Registrar	✓
	datos del	
	producto	
	Actualizar	✓
	datos del	
	producto	
	Cotizar	✓
	solicitud de	
	cliente	
Sprint 3	Programar	✓
	fecha y hora	
	de entrega	

Venta recibir ✓
alerta

Actualizar ✓
usuario

Ingresar al ✓
sistema desde
cualquier
localización

3.4.2.1. Burn Down Chart Semana 2

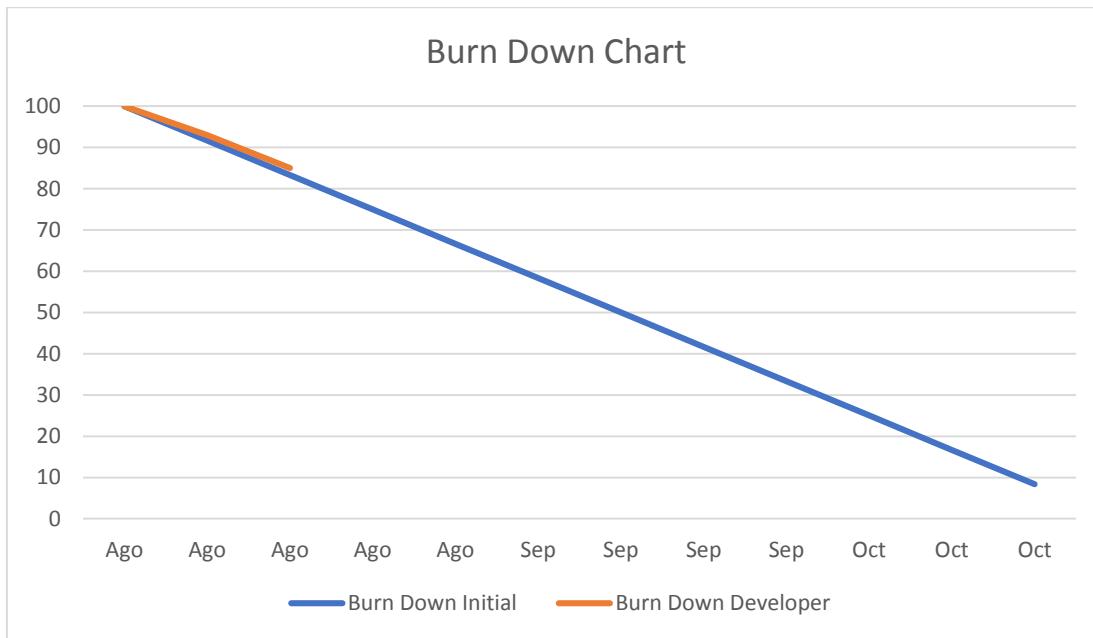


Figura 18. Burn Down Chart semana 2

Fuente: Elaboración propia

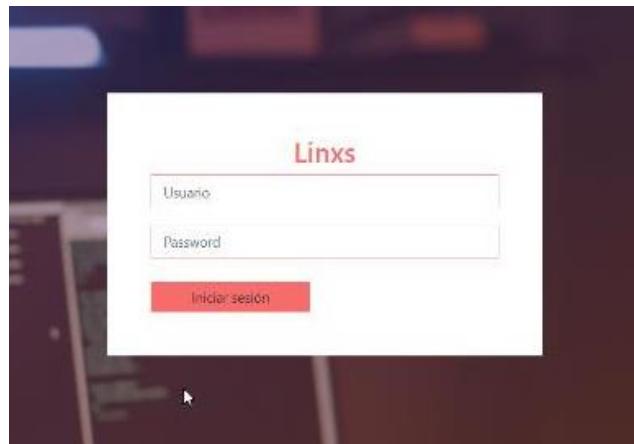


Figura 19. Login

Fuente: Elaboración propia

3.4.3. Semana 3

Se muestra el Taskboard de la Semana 3 en donde, en el Sprint 2 ya se terminó e inicia “Registrar datos de usuario” y “Otorgar permisos del sistema a usuario” se encuentra en curso.

Tabla 11. Semana 3

INICIO	3/8/2020	NOMBRE	DESARROLLO DEL SISTEMA
Fin	30/10/2020	Semana	2
		Historia de usuario	Pendiente
			En curso
			Hecho
Sprint	Crear Base		✓
1	de datos		
		Iniciar sesión	✓
		Ingresar al sistema por	
		Tablet	✓

	Registrar	✓
	datos de	
	usuario	
	Otorgar	✓
	permisos del	
	sistema a	
	usuario	
	Tener un	✓
	control de los	
	módulos del	
	sistema	
	Actualizar	✓
	datos de	
	usuario	
	Registrar	✓
	datos del	
	cliente	
	Actualizar	✓
	datos del	
	cliente	
Sprint 2	Registrar	✓
	datos	
	Actualizar	✓
	datos	
	Verificar	✓
	cliente	

Registrar ✓
datos del
producto

Actualizar ✓
datos del
producto

Cotizar ✓
solicitud de
cliente

Sprint 3 Programar ✓
fecha y hora
de entrega

Venta recibir ✓
alerta

Actualizar ✓
usuario

Ingresar al ✓
sistema
desde
cualquier
localización

Fuente: Elaboración propia

3.4.3.1. Burn Down Chart Semana 3

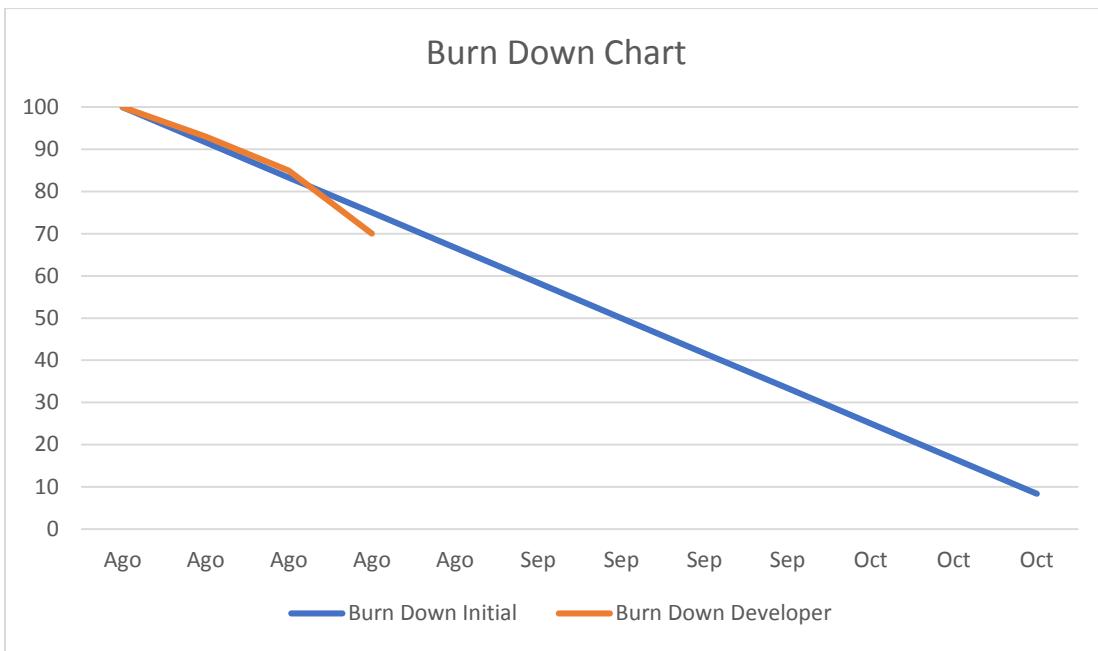


Figura 20. Burn Down Chart semana 3

Fuente: Elaboración propia

Listado de Roles		
Rol	Descripción	Estado
Comprador	Comprador	Activo
Vendedor	Vendedor	Activo
Administrador	Administrador	Activo

Figura 21. Modulo Listado de Roles

Fuente: Elaboración propia

Listado de Usuarios											
Nombre	Tipo Documento	Número	Dirección	Teléfono	Email	Usuario	Rol	Imagen	Editar	Estado	
yomar	CEDULA	473784973948	ave montesco	7839239082	yomar@gmail.com	yomar123	Comprador		 		
Jimmy Mayta	CEDULA	623568989	Ninguna	Ninguna	jimmyymayta@gmail.com	jimmy123	Administrador		 		
Jorge Zamora	CEDULA	0923456835	Cdla. Sauces 7	2237896	jorgezamorajb@hotmail.com	jorge2019	Vendedor		 		

+

Anterior 1 2 3 Siguiente

Figura 22. Modulo Lista de Usuarios

Fuente: Elaboración propia

3.4.4. Semana 4

Se muestra el Taskboard de la Semana 4 en donde, en el Sprint 3 ya se terminó e inicia “Registrar datos de usuario” y “Otorgar permisos del sistema a usuario” se encuentra en curso.

Tabla 12. Semana 4

INICIO	3/8/2020	NOMBRE	DESARROLLO DEL SISTEMA
Fin	30/10/2020	Semana	2
Sprint 1	Crear Base de datos	Historia de usuario	Pendiente
	Iniciar sesión	En curso	Hecho
	Ingresar al sistema por Tablet		✓

	Registrar datos de usuario	✓
	Otorgar permisos del sistema a usuario	✓
	Tener un control de los módulos del sistema	✓
	Actualizar datos de usuario	✓
	Registrar datos del cliente	✓
	Actualizar datos del cliente	✓
Sprint 2	Registrar datos	✓
	Actualizar datos	✓
	Verificar cliente	✓

Registrar ✓
datos del
producto

Actualizar ✓
datos del
producto

Cotizar ✓
solicitud de
cliente

Sprint 3 Programar ✓
fecha y hora
de entrega

Venta recibir ✓
alerta

Actualizar ✓
usuario

Ingresar al ✓
sistema desde
cualquier
localización

Fuente: Elaboración propia

3.4.4.1. Burn Down Chart Semana 4

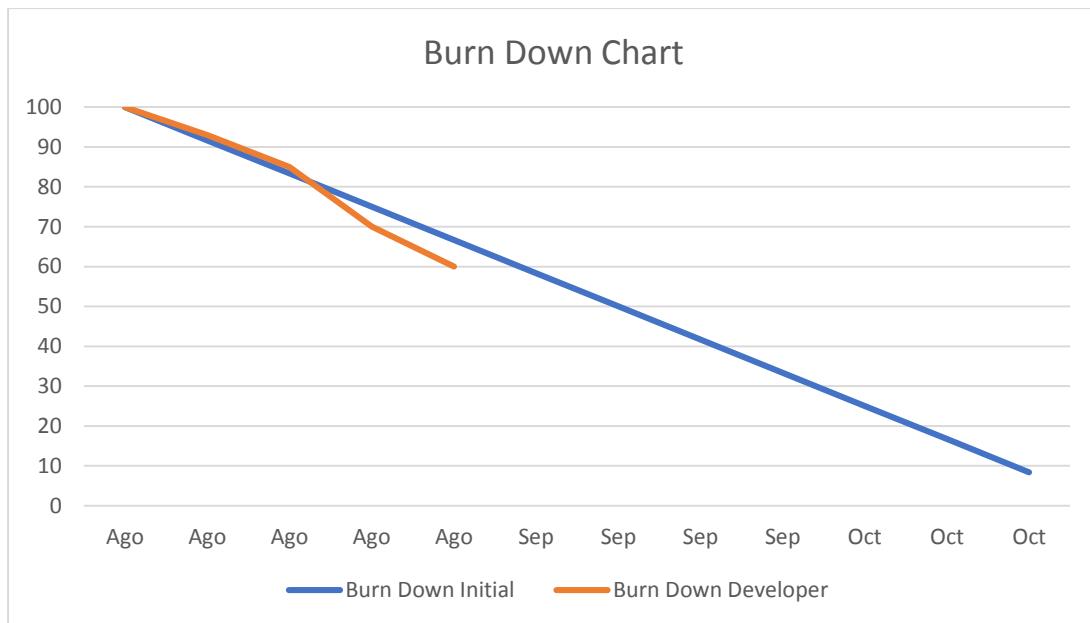


Figura 23. Burn Down Chart semana 4

Fuente: Elaboración propia

Clientes						
Nombre	Buscar texto					
Cliente	Tipo de Documento	Número Documento	Teléfono	Email	Dirección	Editar
Teresa	CEDULA	90138473874	2834784	teresa@gmail.com	ave. libertad	
Adam Montesco	CEDULA	3874814	24040404	adam@hotmail.com	Ave. juan de la cruz	
Helena	CEDULA	67364373	2439049304	helena@gmail.com	Plaza murrillo	

+

Anterior 1 2 Siguiente

Figura 24. Modulo Clientes

Fuente: Elaboración propia

Editar Usuario

Usuario (*)	Jimmy Mayta
Tipo Documento	CEDULA
Número	623568989
Dirección	Ninguna
Teléfono	Ninguna
Email	jimmyymaytaj@gmail.com
Rol (*)	Administrador
Usuario (*)	jimmy123
Password (*)	Contraseña de acceso
Imagen	<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado

Figura 25. Modulo Editar Usuario

Fuente: Elaboración propia

3.4.5. Semana 5

Se muestra el Taskboard de la Semana 4 en donde, en el Sprint 4 ya se terminó e inicia “Registrar datos de usuario” y “Otorgar permisos del sistema a usuario” se encuentra en curso.

Tabla 13. Semana 5

INICIO	3/8/2020	NOMBRE	DESARROLLO DEL SISTEMA
Fin	30/10/2020	Semana	2
		Historia de usuario	Pendiente
			En curso
Sprint 1	Crear Base de datos		Hecho
	Iniciar sesión		✓

	Ingresar al sistema por Tablet	✓
	Registrar datos de usuario	✓
	Otorgar permisos del sistema a usuario	✓
	Tener un control de los módulos del sistema	✓
	Actualizar datos de usuario	✓
	Registrar datos del cliente	✓
	Actualizar datos del cliente	✓
Sprint 2	Registrar datos	✓
	Actualizar datos	✓

Verificar ✓
cliente

Registrar ✓
datos del
producto

Actualizar ✓
datos del
producto

Cotizar ✓
solicitud de
cliente

Sprint 3 Programar ✓
fecha y hora
de entrega

Venta recibir ✓
alerta

Actualizar ✓
usuario

Ingresar al ✓
sistema desde
cualquier
localización

Fuente: Elaboración propia

3.4.5.1. Burn Down Chart Semana 5

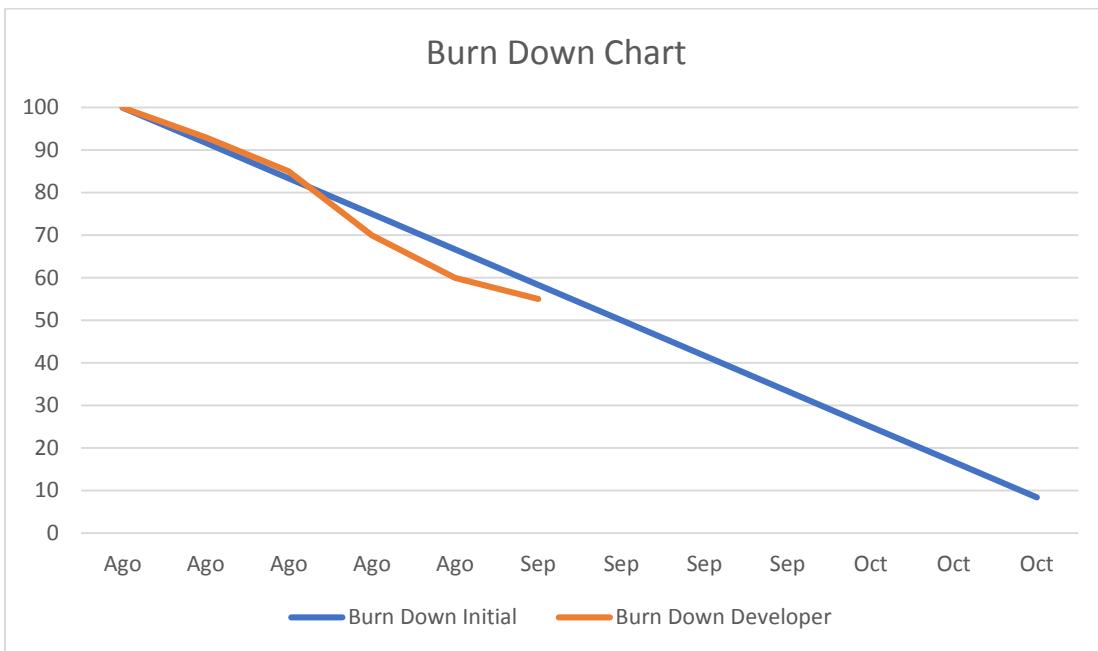


Figura 26. Burn Down Chart semana 5

Fuente: Elaboración propia

Clientes						
Nombre	Buscar texto					
Cliente	Tipo de Documento	Número Documento	Teléfono	Email	Dirección	Editar
Teresa	CEDULA	90438473874	2834784	teresa@gmail.com	ave. libertad	
Adam Montesca	CEDULA	3874814	24040404	adam@hotmail.com	Ave. juan de la cruz	
Helena	CEDULA	67364373	2435049304	helena@gmail.com	Plaza murillo	

+

Anterior 1 2 Siguiente

Figura 27. Modulo Clientes

Fuente: Elaboración propia

Editar Cliente

Cliente (*)	teresa
Tipo Documento	CEDULA
Número	90438473874
Dirección	ave. libertad
Teléfono	2834784
Email	teresa@gmail.com

x 

Figura 28. Modulo Editar Cliente

Fuente: Elaboración propia

3.4.6. Semana 6

Se muestra el Taskboard de la Semana 6 en donde, en el Sprint 5 ya se terminó e inicia “Registrar datos” y “Actualizar datos” se encuentra en curso.

Tabla 14. Semana 6

INICIO	3/8/2020	NOMBRE	DESARROLLO DEL SISTEMA
Fin	30/10/2020	Semana	2
		Historia de Pendiente usuario	En curso Hecho
Sprint 1	Crear Base de datos		✓
	Iniciar sesión		✓
	Ingresar al sistema por Tablet		✓

	Registrar datos de usuario	✓
	Otorgar permisos del sistema a usuario	✓
	Tener un control de los módulos del sistema	✓
	Actualizar datos de usuario	✓
	Registrar datos del cliente	✓
	Actualizar datos del cliente	✓
Sprint 2	Registrar datos	✓
	Actualizar datos	✓
	Verificar cliente	✓

Registrar ✓
datos del
producto

Actualizar ✓
datos del
producto

Cotizar ✓
solicitud de
cliente

Sprint 3 Programar ✓
fecha y hora
de entrega

Venta recibir ✓
alerta

Actualizar ✓
usuario

Ingresar al ✓
sistema desde
cualquier
localización

Fuente: Elaboración propia

3.4.6.1. Burn Down Chart Semana 6

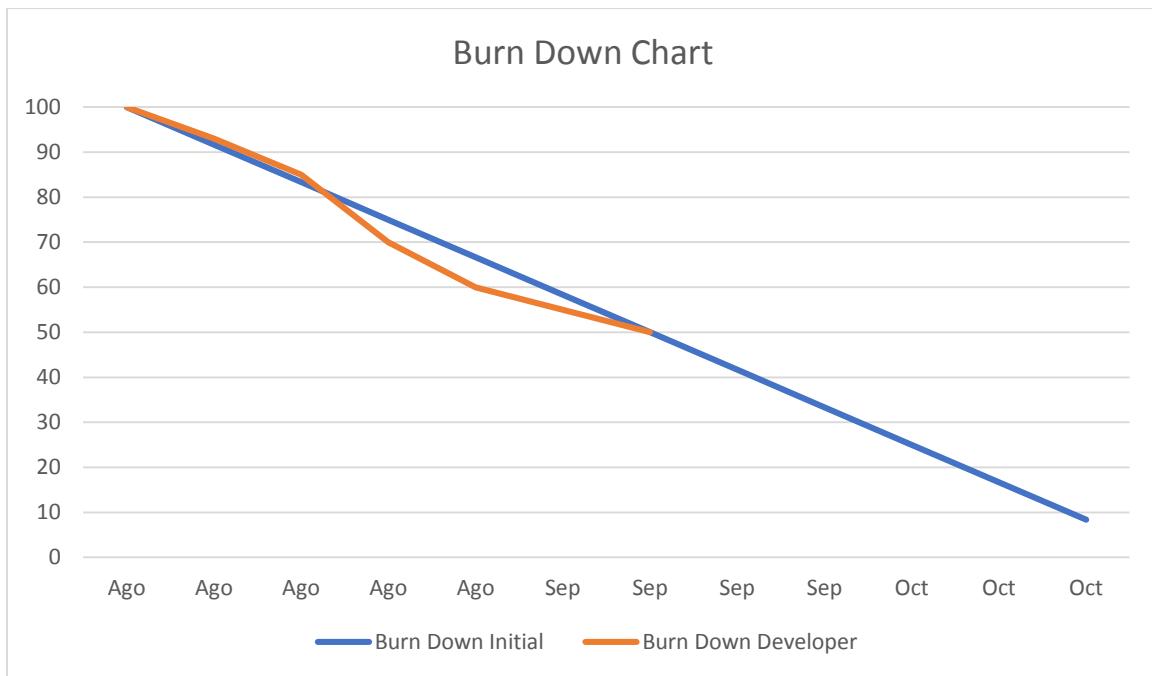


Figura 29. Burn Down Chart semana 6

Fuente: Elaboración propia

Categorías				
Categoría	Descripción	Imagen	Estado	
Página Web	Página Web		Activo	
BETM	Bolivia en tus Manos		Activo	
Hosting	Hosting		Activo	

+

Anterior 1 2 3 Siguiente

Figura 30. Modulo Categorías

Fuente: Elaboración propia

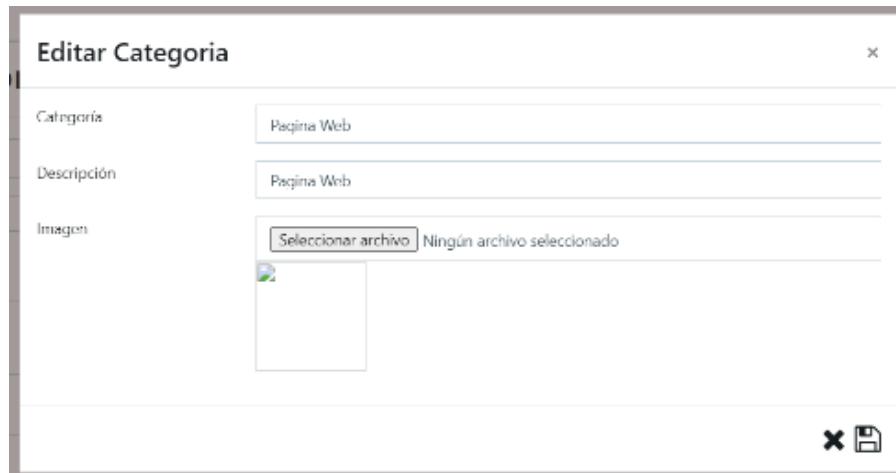


Figura 31. Modulo Editar Categoría

Fuente: Elaboración propia

3.4.7. Semana 7

Se muestra el Dashboard de la Semana 7 en donde, en el Sprint 6 ya se terminó e inicia “Verificar cliente” se encuentra en curso.

Tabla 15. Semana 7

INICIO	3/8/2020	NOMBRE	DESARROLLO DEL SISTEMA
Fin	30/10/2020	Semana	2
		Historia de Pendiente usuario	En curso
Sprint 1	Crear Base de datos		✓
	Iniciar sesión		✓
	Ingresar al sistema por Tablet		✓

	Registrar datos de usuario	✓
	Otorgar permisos del sistema a usuario	✓
	Tener un control de los módulos del sistema	✓
	Actualizar datos de usuario	✓
	Registrar datos del cliente	✓
	Actualizar datos del cliente	✓
Sprint 2	Registrar datos	✓
	Actualizar datos	✓
	Verificar cliente	✓

Registrar ✓
datos del
producto

Actualizar ✓
datos del
producto

Cotizar ✓
solicitud de
cliente

Sprint 3 Programar ✓
fecha y hora
de entrega

Venta recibir ✓
alerta

Actualizar ✓
usuario

Ingresar al ✓
sistema desde
cualquier
localización

Fuente: Elaboración propia

3.4.7.1. Burn Down Chart Semana 7

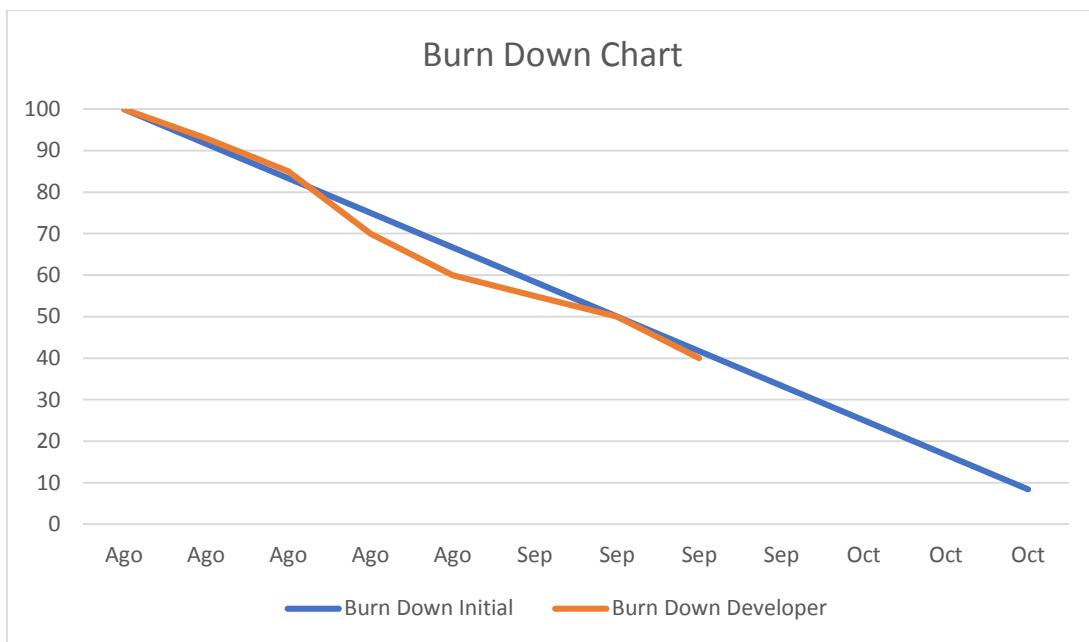


Figura 32. Burn Down Chart semana 7

Fuente: Elaboración propia

Detalle Ventas				
CLIENTE	TIPO DE IDENTIFICACIÓN	NÚMERO VENTA	IMUESTRO	
Marisol	TICKET	12341234	0.12	
Producto	Precio (Bs)	Cantidad	Descuento	Total (Bs)
Motor de Compras	13000.00	1	0.00	13000
Motor de Reserva Mayta	19000.00	1	0.00	19000
Sub-Total: 28160.00 (Bs)				
Impuesto: 3840.00 (Bs)				
Total: 32000.00 (Bs)				

x

Figura 33. Modulo Detalle Ventas

Fuente: Elaboración propia

3.4.8. Semana 8

Se muestra el Taskboard de la Semana 8 en donde, en el Sprint 6 ya se terminó e inicia “Registrar datos del producto” y “Actualizar datos del producto” se encuentra en curso.

Tabla 16. Semana 8

INICIO	3/8/2020	NOMBRE	DESARROLLO DEL SISTEMA
Fin	30/10/2020	Semana	2
		Historia de Pendiente usuario	En curso
Sprint 1		Crear Base de datos	✓
		Iniciar sesión	✓
		Ingresar al sistema por Tablet	✓
		Registrar datos de usuario	✓
		Otorgar permisos del sistema a usuario	✓
		Tener un control de los módulos del sistema	✓
		Actualizar datos de usuario	✓

	Registrar datos del cliente	✓
	Actualizar datos del cliente	✓
Sprint 2	Registrar datos	✓
	Actualizar datos	✓
	Verificar cliente	✓
	Registrar datos del producto	✓
	Actualizar datos del producto	✓
	Cotizar solicitud de cliente	✓
Sprint 3	Programar ✓ fecha y hora de entrega	
	Venta recibir ✓ alerta	

- Actualizar ✓
- usuario
- Ingresar al ✓
- sistema desde
- cualquier
- localización

Fuente: Elaboración propia

3.4.8.1. Burn Down Chart Semana 8

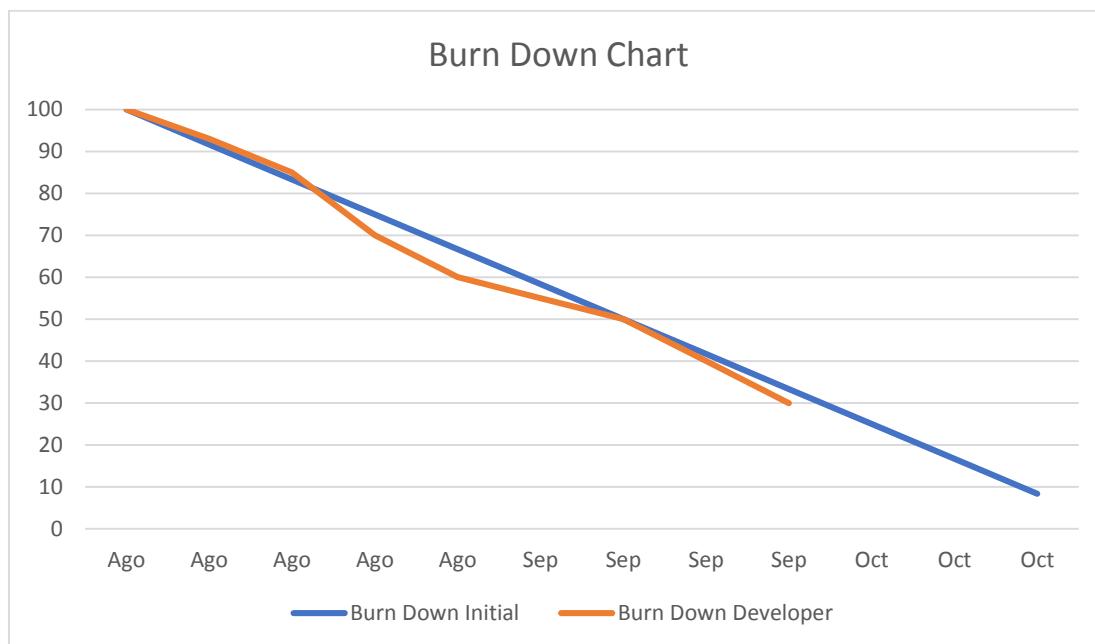


Figura 34. Burn Down Chart semana 8

Fuente: Elaboración propia

Productos								
Producto	Buscar texto							
Categoría	Producto	Código	Precio Venta (Bs)	Imagen	Estado	Editar	Cambiar Estado	
Desarrollo de Sistemas	Sistema de contabilidad	100	1200.00		Activo			
Sistemas de Reservas	Motor Hoteles	456745	12000.00		Desactivado			
Sistemas de Reservas	Motor de Compras	4578985	13000.00		Activo			

+

Anterior 1 2 Siguiente

Figura 35. Modulo Productos

Fuente: Elaboración propia

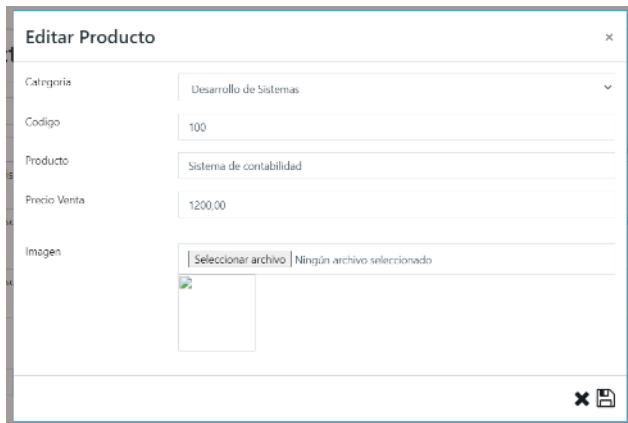


Figura 36. Modulo Editar Producto

Fuente: Elaboración propia

3.4.9. Semana 9

Se muestra el Taskboard de la Semana 9 en donde, en el Sprint 8 ya se terminó e inicia “Cotizar solicitud de cliente” se encuentra en curso.

Tabla 17. Semana 9

INICIO	3/8/2020	NOMBRE	DESARROLLO	DEL SISTEMA
Fin	30/10/2020	Semana	2	
		Historia de usuario	Pendiente	En curso
Sprint 1	Crear Base de datos			✓
	Iniciar sesión			✓
	Ingresar al sistema por Tablet			✓
	Registrar datos de usuario			✓
	Otorgar permisos del sistema a usuario			✓

	Tener un control de los módulos del sistema	✓
	Actualizar datos de usuario	✓
	Registrar datos del cliente	✓
	Actualizar datos del cliente	✓
Sprint 2	Registrar datos	✓
	Actualizar datos	✓
	Verificar cliente	✓
	Registrar datos del producto	✓
	Actualizar datos del producto	✓
	Cotizar solicitud de cliente	✓
Sprint 3	Programar fecha y hora de entrega	✓
	Venta recibir alerta	✓
	Actualizar usuario	✓
	Ingresar al sistema desde cualquier localización	✓

Fuente: Elaboración propia

3.4.9.1. Burn Down Chart Semana 9

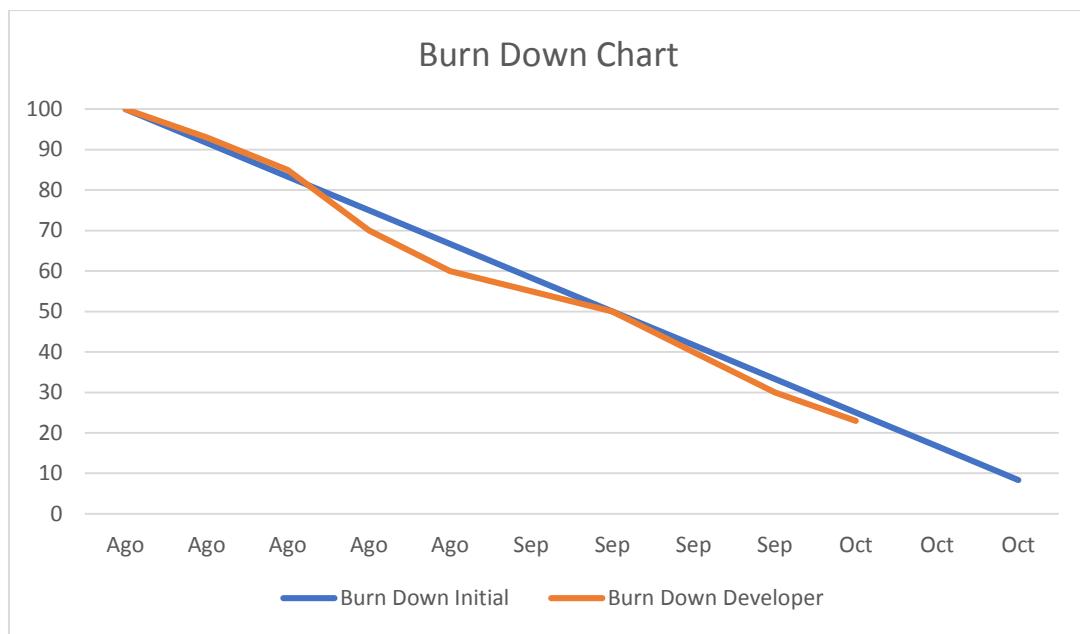


Figura 37. Burn Down Chart semana 9

Fuente: Elaboración propia

Producto	Precio (Bs)	Descuento	Total (Bs)	Eliminar
No se han agregado productos				

Figura 38. Modulo agregar venta

Fuente: Elaboración propia

3.4.10. Semana 10

Se muestra el Taskboard de la Semana 10 en donde, en el Sprint 9 ya se terminó e inicia “Programar fecha y hora de entrega” se encuentra en curso.

Tabla 18. Semana 10

INICIO 3/8/2020

NOMBRE DESARROLLO
DEL SISTEMA

Fin	30/10/2020	Semana	2	
				Historia de usuario
Sprint 1	Crear Base de datos	Pendiente	En curso	Hecho
	Iniciar sesión			✓
	Ingresar al sistema por Tablet			✓
	Registrar datos de usuario			✓
	Otorgar permisos del sistema a usuario			✓
	Tener un control de los módulos del sistema			✓
	Actualizar datos de usuario			✓
	Registrar datos del cliente			✓
	Actualizar datos del cliente			✓
Sprint 2	Registrar datos			✓
	Actualizar datos			✓
	Verificar cliente			✓

	Registrar datos del producto	✓
	Actualizar datos del producto	✓
	Cotizar solicitud de cliente	✓
Sprint 3	Programar fecha y hora de entrega	✓
	Venta recibir alerta	✓
	Actualizar usuario	✓
	Ingresar al sistema desde cualquier localización	✓

Fuente: Elaboración propia

3.4.10.1. Burn Down Chart Semana 10

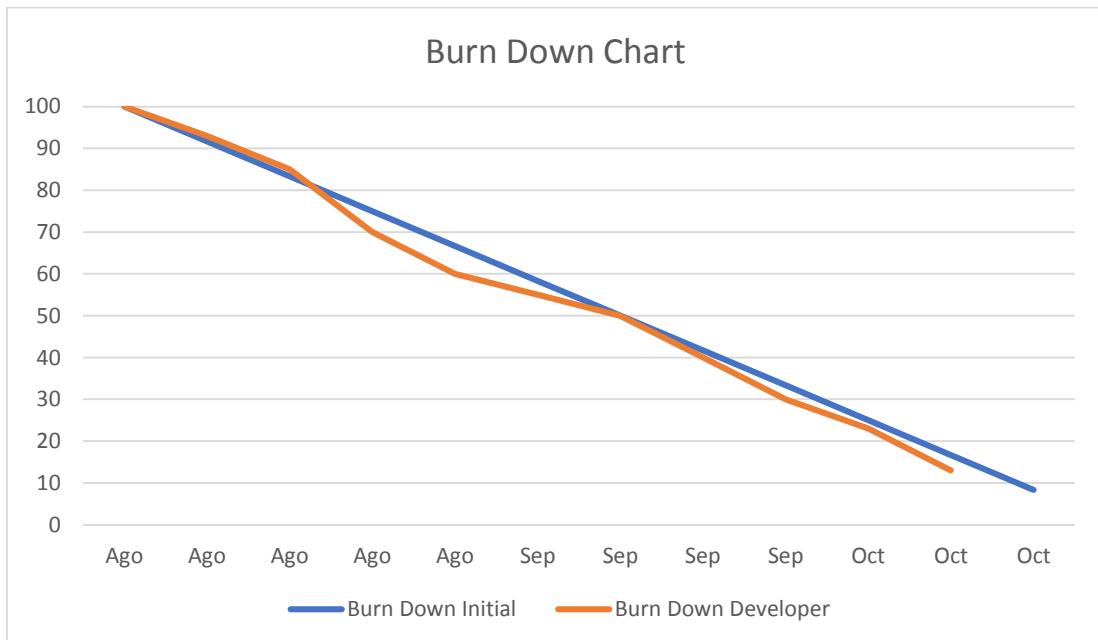


Figura 39. Burn Down Chart semana 10

Fuente: Elaboración propia

```

    public static function DateTime() {
        $dateTime = new DateTime('now', new DateTimeZone('America/La_Paz'));
        return $dateTime->format('Y-m-d H:i:s');
    }

    public static function DateLinez($date) {
        $dateTime = new DateTime($date, new DateTimeZone('America/La_Paz'));
        return $dateTime->format('Y-m-d H:i:s');
    }

    public static function DateDifference($date) {
        $date1 = new DateTime($date, new DateTimezone('America/La_Paz'));
        $date2 = new DateTime('now', new Datetimezone('America/La_Paz'));
        $diff = $date1->diff($date2);
        if ($date1->format('Y-m-d') > $date2->format('Y-m-d')) {
            return $diff->days * -1;
        } else {
            return $diff->days;
        }
    }

    public static function DateDifferencez($date1, $date2) {
        $date1 = new DateTime($date1, new DateTimeZone('America/La_Paz'));
        $date2 = new DateTime($date2, new DateTimeZone('America/La_Paz'));
        $diff = $date1->diff($date2);
        if ($date1->format('Y-m-d') > $date2->format('Y-m-d')) {
            return $diff->days * -1;
        } else {
            return $diff->days;
        }
    }

    public static function DateDifferences($date1, $date2) {
        $date1 = new DateTime($date1, new DateTimeZone('America/La_Paz'));
        $date2 = new DateTime($date2, new DateTimeZone('America/La_Paz'));
        $diff = $date1->diff($date2);
        return $diff->days;
    }
}

```

Figura 40. Código fecha y hora

Fuente: Elaboración propia

3.4.11. Semana 11

Se muestra el Dashboard de la Semana 11 en donde, en el Sprint 10 ya se terminó e inicia “Programar fecha y hora de entrega” se encuentra en curso.

Tabla 19. Semana 11

INICIO	3/8/2020	NOMBRE	DESARROLLO
			DEL SISTEMA

Fin	30/10/2020	Semana	2	
		Pendiente	En curso	Hecho
Sprint 1	Crear Base de datos			✓
	Iniciar sesión			✓
	Ingresar al sistema por Tablet			✓

	Registrar datos de usuario	✓
	Otorgar permisos del sistema a usuario	✓
	Tener un control de los módulos del sistema	✓
	Actualizar datos de usuario	✓
	Registrar datos del cliente	✓
	Actualizar datos del cliente	✓
Sprint 2	Registrar datos	✓
	Actualizar datos	✓
	Verificar cliente	✓
	Registrar datos del producto	✓
	Actualizar datos del producto	✓
	Cotizar solicitud de cliente	✓
Sprint 3	Programar fecha y hora de entrega	✓
	Venta recibir alerta	✓
	Actualizar usuario	✓

Ingresar al sistema ✓
desde cualquier
localización

Fuente: Elaboración propia

3.4.11.1. Burn Down Chart Semana 11

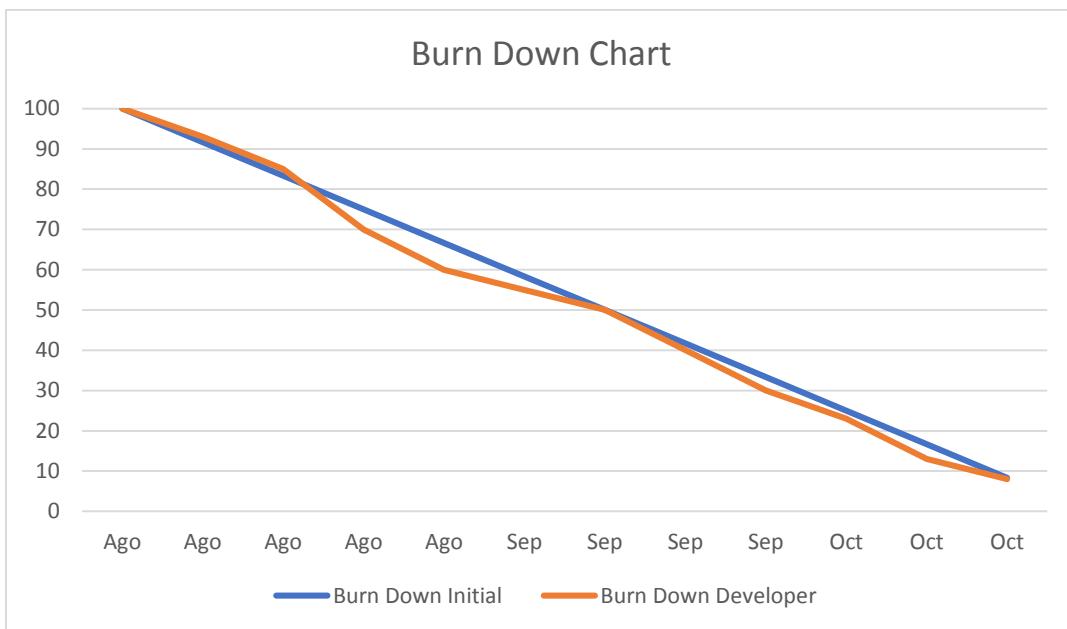


Figura 41. Burn Down Chart semana 11

Fuente: Elaboración propia

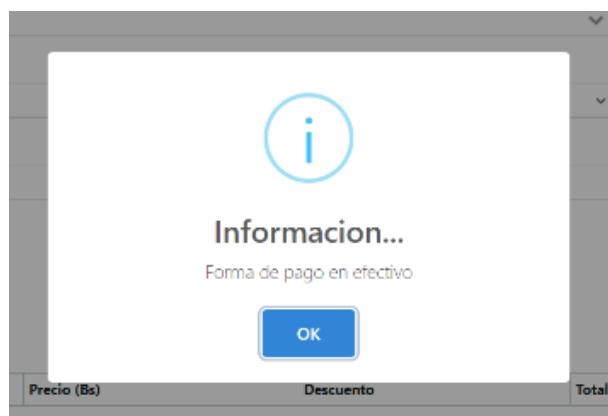


Figura 42. Alerta Información

Fuente: Elaboración propia

3.4.12. Semana 12

Se muestra el Taskboard de la Semana 12 en donde, en el Sprint 11 ya se terminó e inicia “Programar fecha y hora de entrega” se encuentra en curso.

Tabla 20. Semana 12

INICIO	3/8/2020	NOMBRE	DESARROLLO DEL SISTEMA
Fin	30/10/2020	Semana 2	
Sprint 1		Historia de usuario	Pendiente
		Crear Base de datos	En curso
		Iniciar sesión	Hecho
		Ingresar al sistema por Tablet	
		Registrar datos de usuario	✓
		Otorgar permisos del sistema a usuario	✓
		Tener un control de los módulos del sistema	✓
		Actualizar datos de usuario	✓
		Registrar datos del cliente	✓

	Actualizar datos del cliente	✓
Sprint 2	Registrar datos	✓
	Actualizar datos	✓
	Verificar cliente	✓
	Registrar datos del producto	✓
	Actualizar datos del producto	✓
	Cotizar solicitud de cliente	✓
Sprint 3	Programar fecha y hora de entrega	✓
	Venta recibir alerta	✓
	Actualizar usuario	✓
	Ingresar al sistema desde cualquier localización	✓

Fuente: Elaboración propia

3.4.12.1. Burn Down Chart Semana 12

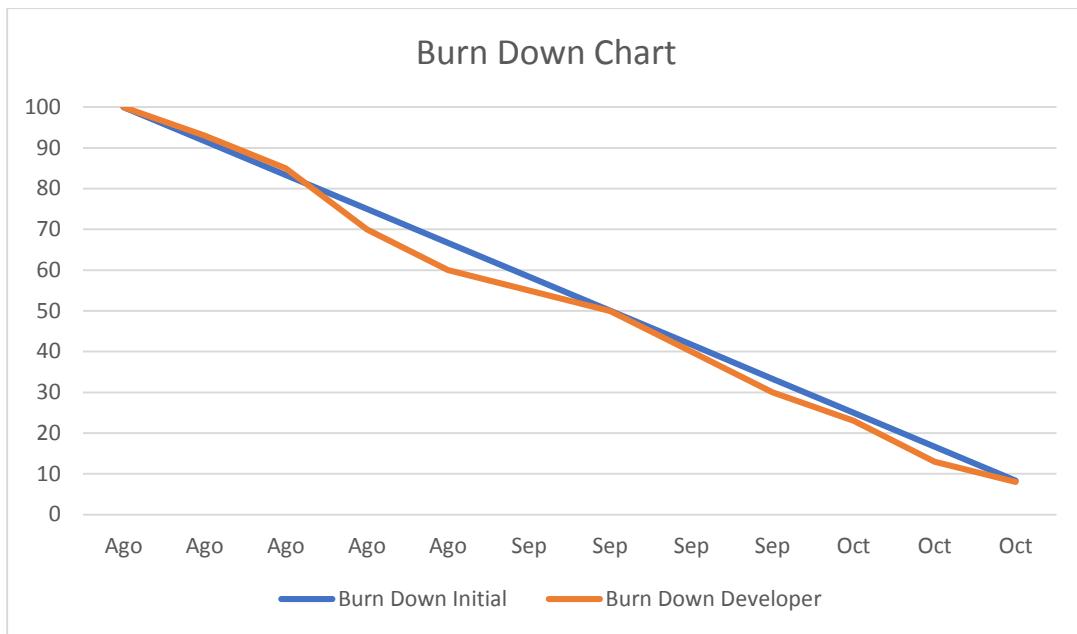


Figura 43. Burn Down Chart semana 12

Fuente: Elaboración propia

Nombre (*)	eva12345
Tipo Documento	CEDULA
Número	46546465
Dirección	vafadad
Teléfono	7465465
Email	eva@gmail.com
Role (*)	Vendedor
Usuario (*)	eva12345
Password (*)	password de acceso
Imagen	Seleccionar archivo Ningún archivo seleccionado

Figura 44. Modulo Editar Usuario

Fuente: Elaboración propia

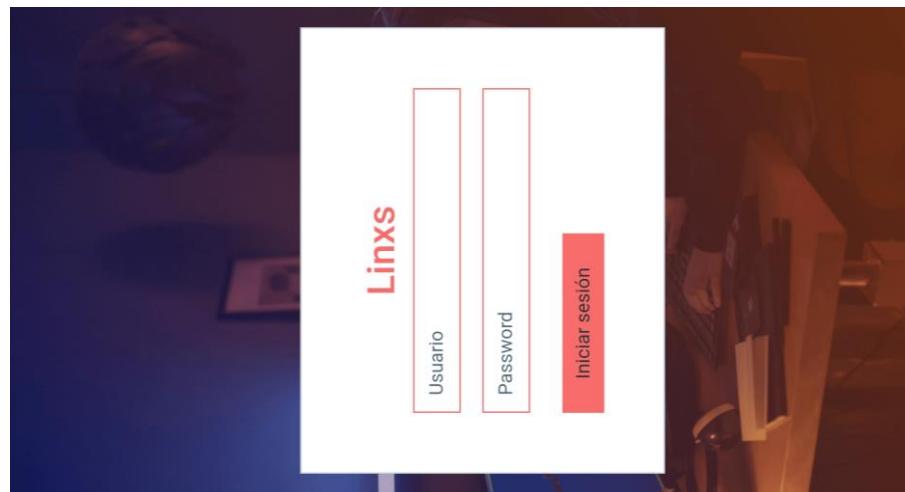


Figura 45. Módulo Login (Dispositivo móvil)

Fuente: Elaboración propia

3.5. PRUEBAS DE EVALUACIÓN DE RESULTADOS

3.5.1. Métricas de calidad

Las Métricas de Calidad proporcionan una indicación de cómo se ajusta el software a los requerimientos implícitos y explícitos del cliente.

Para lo cual se aplicará la norma ISO 25000 conocida como SQuaRE (Software Product Quality Requirements and Evaluation) tiene por objetivo la creación de un marco de trabajo para definir los requisitos y evaluar la calidad del producto software.



Figura 46. ISO 25000

Fuente: iso25000.com

3.5.1.1. Modelos de calidad de software y sistema

Describe el modelo de calidad para el producto software y calidad en uso



Figura 47. Calidad del Producto Software

Fuente: iso25000.com

3.5.1.2. Usabilidad

La usabilidad consiste en la evaluación del esfuerzo necesario que el usuario invertirá para usar el sistema, en base a su comprensión y estructura lógica que el sistema tiene. Esta comprensión por parte de los usuarios con relación al sistema evalúa los siguientes casos:

- Comprensibilidad
- Facilidad de Aprender
- Operabilidad

Se realizan encuestas a los usuarios finales sobre el manejo, la compresión, y la facilidad de aprender el sistema para medir la usabilidad según la siguiente tabla:

Tabla 21. Usabilidad

PREGUNTAS	CALIFICACIÓN	OBSERVACIÓN
¿Tiempo que los usuarios se toman para completar tareas específicas?	93.1	Calificación obtenida
¿El número de tareas de diversos tipos que puede ser completada dentro de un tiempo límite dado?	91.3	Calificación obtenida
¿La relación entre interacciones exitosas y los errores?	93.3	Calificación obtenida
¿El tiempo empleado en la recuperación de los errores?	89.3	Calificación obtenida

¿El número de acciones erróneas inmediatamente posteriores?	90	Calificación obtenida
¿La frecuencia del uso de manuales y/o ayuda del sistema resolviendo el problema del usuario?	90	Calificación obtenida
¿El número de ocasiones que el usuario se debía de la tarea real?	90.1	Calificación obtenida
Promedio	91.01	

Fuente: Elaboración propia

Entonces el sistema tiene una usabilidad del 91.01 %.

Adicionalmente, el usuario podrá evaluar desde su propio punto de vista la usabilidad en lo que respecta a la facilidad de aprendizaje.

3.5.1.3. Mantenimiento

Para hallar la mantenibilidad del sistema se utiliza el índice de madurez de software (IMS), que proporciona una indicación de la estabilidad de un producto de software (basado en los cambios que ocurren con cada versión del producto).

Se determina la siguiente fórmula para hallar el (IMS):

$$IMS = \frac{M_t - (F_c + F_a + FE)}{M_t}$$

Donde:

- M_t : Numero de módulos en la versión actual
- F_c : Numero de modulos en la versión actual que se han cambiado
- F_a : Numero de módulos en la versión actual que se han añadido
- FE : Numero de módulos en la versión anterior que se han borrado en la versión actual

Recopilando la información requerida por la formula se obtuvo la información que se muestra en la siguiente tabla:

Tabla 22. Capacidad de mantenimiento

INFORMACIÓN

VALOR

M_t	16
F_c	1
F_a	0
FE	0

Fuente: Elaboración propia

Calculando el índice de madurez de software (IMS), usando los valores obtenidos, se obtiene:

$$\text{IMS} = \frac{16 - (1 + 0 + 0)}{16} = \frac{15}{16} = 0.93$$

$$\text{IMS} = 1 * 100\% = 93\%$$

Con ese resultado se concluye que el sistema “Sistema de Información para la administración y control de ventas”, tiene un índice de madurez de software del 93%.

3.5.1.4. Funcionalidad

La funcionalidad examina si el sistema satisface los requisitos funcionales esperados. El objetivo es revelar problemas y errores en lo que concierne a la funcionalidad del sistema y su conformidad al comportamiento, expresado o deseado por el usuario. Haremos uso de cinco características de dominios de información y se proporcionan las cuentas en la posición apropiada. Los valores de los dominios de información, y se definen de la siguiente manera:

- Número de entradas de usuario. Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.
- Número de entradas de usuario. Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se

deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.

- Número de salidas de usuario. En este contexto la salida de usuario se refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.
- Número de peticiones de usuario. Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
- Número de archivos. Se cuenta cada archivo maestro lógico (se refiere a un grupo lógico de datos que puede ser parte de una gran base de datos o un archivo independiente).
- Número de interfaces externas. Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de disco) que se utilizan para transmitir información a oro sistema [Pressman, 2005].

Tabla 23. Capacidad de mantenimiento

PARÁMETROS DE ENTRADA	CUENTA
Número de entradas de usuario	2
Número de salidas de usuario	2
Número de peticiones de usuario	3
Numero de archivos	1
Numero de interfaces externas	0
Total	8

Fuente: Elaboración propia

Calculando los puntos de fusión considerando los factores de ponderación medio.

Tabla 24. Puntos de fusión

PARÁMETROS DE ENTRADA	CUENTA	FACTOR	TOTALES
		PENDERACIÓN	
		MEDIO	
Número de entradas de usuario	2	8	16

Número de salidas de usuario	2	8	16
Numero peticiones de usuario	3	4	12
Numero de archivos	1	10	10
Numero de interfaces externas	0	7	0
Total			54

Fuente: Elaboración propia

Calcularemos la relación para calcular el punto de fusión:

$$PF = \text{Cuenta Total} * (\text{Grado de confiabilidad} + \text{Tasa de error}) * \sum F$$

Donde:

PF = Media de confiabilidad

Cuenta Total = Es la suma del valor de las entradas, salidas, peticiones, interfaces externas y archivos.

Grado de confiabilidad = Confiabilidad estimada del sistema

Taza de error = Probabilidad subjetiva estimada del dominio de la información este error es el 1%.

F = Son los valores de ajuste de complejidad.

**Tabla 25. Puntos de fusión
IMPORTANCIA**

	SIN	IMPORTANCIA	INCREMENTAL	MODERADO	MEDIO	SIGNIFICATIVO	ESENCIAL
Requiere el sistema de información copias de seguridad y es de fácil recuperación						X	
Se requiere comunicación de datos							X

Existe funciones de procesos distribuidos	X
Es critico el rendimiento	X
El sistema web será ejecutado en un sistema operativo	X
Se requiere una entrada interactiva para el sistema	X
Se requiere que el sistema tenga entradas a datos con múltiples ventanas	X
Se actualiza los archivos de forma interactiva	X
Son complejas las entradas, salidas, los archivos o procesos	X
Es complejo el procesamiento interno del sistema	X
Se ha diseñado el código para ser reutilizado	X
Se ha diseñado el sistema para facilitar al usuario el trabajo	x
Cuenta Total	45

Hallamos el punto de valor máximo para comparar los valores del sistema.

$$PF = \text{Cuenta Total} * (\text{Grado de confiabilidad} + \text{Tasa de error}) * \sum F$$

$$PF = 54 * (0.91 + 0.01) * 45 = 73.44$$

El punto función ideal al 100 % de los factores en la ecuación tiene el valor de 48, a continuación, se halla el PF para obtener la estimación con el resultado antes obtenido.

$$PF = 54 * (0.91 + 0.01) * 48 = 75.06$$

Con ambos resultados se calcula el porcentaje de funcionalidad del sistema:

$$PF \text{ Real} = \frac{PF \text{ Obtenida}}{PF \text{ Ideal}} * 100$$

$$PF \text{ Real} = \frac{48.96}{50.84} * 100 = 97.84 \%$$

El resultado obtenido muestra que el sistema tiene una funcionalidad o utilidad del 97,84% para la organización, lo que indica que él es sistema cumple con los requisitos funcionales de forma satisfactoria.

3.5.1.5. Confiabilidad

Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido. [Valle, 2009]

La función siguiente muestra el nivel de confiabilidad del sistema:

$$F(t) = (\text{Funcionalidad}) * e^{-\lambda t}$$

Se observa el trabajo hasta que se observa un fallo en un instante, la función es la siguiente.

Probabilidad de hallar una falla: $P(T \leq t) = F(t)$

Probabilidad de hallar una falla: $P(T > t) = 1 - F(t)$

Valor de Funcionalidad previo = 97.84 %

$\lambda = 0.01$ (es decir 1 error en cada 6 ejecuciones)

$t = 3$ meses

Hallamos la confiabilidad del sistema.

$$F(3) = 97.84 * e^{-\frac{1}{6}*12} = 13.24 \%$$

La probabilidad de hallar una falla es de un 13.24% durante los próximos 12 meses.

Por lo tanto, la probabilidad de no hallar un error es del 86.76 %

3.5.1.6. Portabilidad

El sistema está desarrollado para ejecutarse en plataforma Windows (XP, 7, 8, 8.1, 10), Linux (Debian, Ubuntu, otros)

Por lo tanto, puede ser ejecutado en cualquier computador que posee alguno de los sistemas operativos mencionados anteriormente, obteniendo un porcentaje de portabilidad de un 91%.

3.5.1.7. Eficiencia

Eficiencia del sistema, se evalúa tomando en cuenta a algunas medidas realizadas anteriormente para la usabilidad.

El tiempo que los usuarios se toman para contemplar tareas específicas y el número de tareas de diversos tipos que puedan ser completadas dentro de un tiempo límite dado, nos da una estimación del comportamiento temporal del sistema.

En este sentido, la calificación que obtiene el sistema es de 91,3%.

Evaluación de calidad

Característica	Resultado (%)
Usabilidad	91 %
Mantenimiento	93 %
Funcionalidad	97 %
Confiabilidad	86 %
Portabilidad	91 %
Eficiencia	91 %
Evaluación total de calidad	91 %

Fuente: Elaboración propia

3.5.2. Seguridad

3.5.2.1. Seguridad de la base de datos

3.5.2.1.1. Seguridad lógica

Se evalúan los controles de acceso de los usuarios a las plataformas de procesamiento informático y a los datos que estas gestionan con el fin de evitar las

irregularidades que obstaculicen la confidencialidad, exactitud y disponibilidad de la información.

Gestión de comunicación y operaciones.

- Los respaldos (Back-up) de la base de datos (Poveda).
- Los personales involucrados en el proceso del sistema deberán cambiar el Password del sistema periódicamente una vez al mes.
- Recomendamos al personal involucrado en el sistema cada vez que ingrese al sistema en recomendable que cierre sesión.

3.5.2.1.2. Seguridad física

Para mantener la seguridad física de los archivos donde se guarda la base de datos se crearon procesos que realizan back-ups.

Seguridad física del entorno de alojamiento de sistema.

- Se recomienda los back-up 2 a 3 copias en distintos discos de almacenamiento.
- Las copias de la base de datos deberán ser protegidos en un área donde solo el personal autorizado tenga el acceso.

3.5.3. Análisis de costos

3.5.3.1. Estrategias de medición

En el caso de este sistema es una propuesta que aportara de mucho a la institución, el cual cuenta con una persona asignada al equipo de desarrollo de software y se sabe que su costo mensual es de Bs. 3000.

Al mismo tiempo se pudo determinar que, en los últimos 3 meses, el equipo de trabajo ha producido un promedio de 67 puntos de función.

Para determinar cuánto cuesta desarrollar cada punto de función se utiliza la siguiente formula:

$$\text{Costo por punto de función} = \frac{\text{Costo mes del equipo de trabajo}}{\text{Puntos de función del mes}}$$

$$\text{Costo por punto de función} = \frac{3000}{67}$$

Costo por punto de función = 10.45 [Bs]

Aplicando la fórmula se determinó el costo por punto de función en nuestro sistema que es 10.45 Bs.

3.5.3.2. Mapeo

El método de medida Cosmic CFP (Cosmic Function Points) para el presente proyecto de grado muestra los siguientes puntos de función:

Tabla 26. Tabla de mapeo

N.	PROCESOS FUNCIONALES	CFP
----	----------------------	-----

1	LOGIN (Inicio de sesión)	
	Entrada	30
2	ADMINISTRACIÓN DE USUARIOS	
	Entrar	
	Agregar nuevo usuario	10
	Listar usuarios	30
	Editar usuarios	30
	Modificar estado usuario	33
	Eliminar usuario	39
	Alertas	19
		13
3	ADMINISTRACIÓN DE CLIENTES	
	Entrar	
	Agregar nuevo cliente	10
	Listar clientes	30

Editar clientes	39
Modificar cliente	33
Eliminar cliente	39
Alertas	19
	13
4 ADMINISTRACIÓN DE VENTAS	
Entrar	10
Agregar nueva venta	33
Listar ventas	39
Editar estado	31
Eliminar venta	19
Alertas	13
5 ADMINISTRACIÓN DE PRODUCTOS	
Entrar	
Agregar nuevo producto	10
Listar productos	39
Editar producto	39
Estado venta	33
Alertas	30
	13
6 ADMINISTRACIÓN CATEGORÍAS	
Entrar	

Agregar nueva categoría	10
Listar categoría	39
Editar categoría	39
Estado categoría	33
Alertas	30
	13
Total	860

Fuente: Elaboración propia

De esta forma se ha determinado que el proyecto tiene una medición de 860 CFP puntos de función.

3.5.3.3. Medición

Una vez efectuada la medición del tamaño del software y el costo por unidad de medida, se procede a determinar el costo del proyecto de software usando la siguiente formula:

Costo del Proyecto de Software=860*10.45

Costo del Proyecto de Software=8,989.54 [Bs]

Por lo tanto, la estimación del costo del sistema es de Bs 8,989.54.

3.5.4. Pruebas de software

3.5.4.1. Pruebas de caja negra y blanca

3.5.4.1.1. Caja blanca

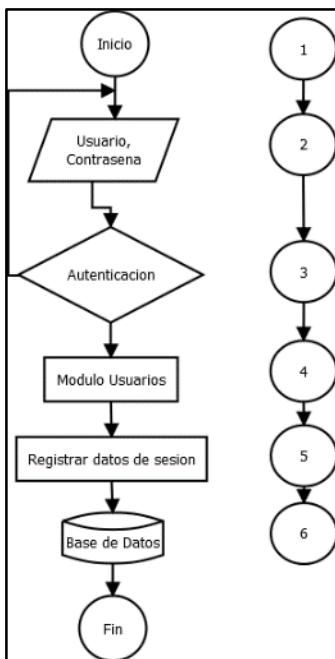


Figura 48. Prueba de Caja Blanca

Fuente: Elaboración propia

La complejidad se obtiene en base al grafo de flujo se define como:

$$V(G)=E-N+2$$

Dónde: E= número de aristas y N= número de nodos.

$$V(G)=6-6+2=2$$

Por lo tanto, las existencias son de 2 caminos independientes, para los que se realizan las pruebas.

Caso de prueba camino 1

Valor (usuario='Eva12345', contraseña='Jhdu7dsjh') = entrada validada ya que los datos ingresados existen en la base de datos, rol como admin.

Ingresar al sistema, visualiza los módulos. Resultados esperados correctos.

Caso de prueba camino 2

Valor (usuario='Eva12345', contraseña='ninguno') = entrada no validada ya que los datos ingresados no existen en la base de datos, vuelva a introducir datos.

Actualiza datos de entrada.

Resultados esperados correctos.

3.5.4.1.2. Caja blanca

La prueba de caja negra de caja negra se realizó en la parte de construcción de Proyecto, codificando resultados el cual nos permite realizar las pruebas para verificar la correcta funcionalidad del sistema.

Prueba de datos fallidos

Ingreso de datos

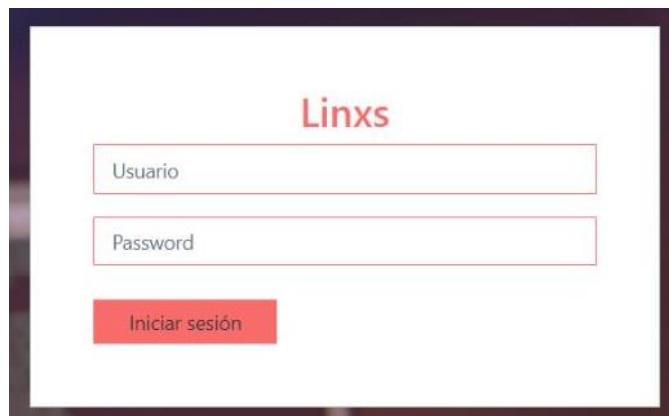


Figura 49. Login

Fuente: Elaboración propia

Resultado de datos

```
User:eva12345  
Password:$2y$10$08OVZOb4PDnF6Sm60aqwf.Ua4cAjpI09wE4zUR8Yx5rUy5yrbIJ0G
```

Figura 50. Resultado de Datos

Fuente: Elaboración propia

Búsqueda de datos

```
MariaDB [isasc_database]> SELECT usuario, password FROM Users;
+-----+-----+
| usuario | password |
+-----+-----+
| Admin1970 | $2y$10$lo/iYvMVo9pSQhjyuBxrh.9JMA1UnnrVY1aXEEedNAZfWIyY5snK2. |
| patricio2 | $2y$10$h.fkAyZeTrPoPwEH2w0LfLfuLEgg09Z.i.NGwpasDvqf77mD2Nkx3xW |
| jessileo1974 | $2y$10$tZT58igSfw1VIWD6/apGSuTXkgEE.wdMLcv2fK1SZ/TWAIG0Uczwa |
| jorge2019 | $2y$10$OA/1EY7HhD0p.BFQXzNAGu0f8rg80KJ.ex3lEoa20Pp2tq0b8y8nC |
| jimmy123 | $2y$10$iJcfDD8KqHks6uOp15tmR.1tRifonryepYBE8bLJBQY32H2N0.zzW |
| yomar123 | $2y$10$y6nh5i3ViMhbS.2bXPPuaOseoNlsgas2ombc403cPbSFPa64Y1H0K |
| eva12345 | $2y$10$5RItwHzLQGFB.CAzcAidx6jmeZ/m0Eq190ZYahG0PcwSN0Mc8mCS |
+-----+-----+
7 rows in set (0.000 sec)

MariaDB [isasc_database]> |
```

Figura 51. Base de Datos Query

Fuente: Elaboración propia

El resultado te regresa al módulo de autenticación



Figura 52. Iniciar sesión con datos

Fuente: Elaboración propia

Prueba de datos correctos

Ingreso de datos

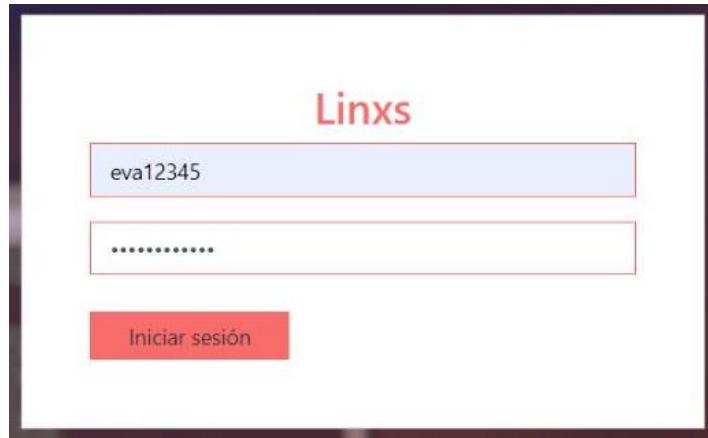


Figura 53. Iniciar sesión con datos

Fuente: Elaboración propia

Resultado de datos

```
User:eva12345  
Password:$2y$10$08OVZOb4PDnF6Sm60aqwf.Ua4cAjpI09wE4zUR8Yx5rUy5yrbIJ0G
```

Figura 54. Resultado de Datos

Fuente: Elaboración propia

Búsqueda de datos

```
MariaDB [isasc_database]> SELECT usuario, password FROM Users;  
+-----+-----+  
| usuario | password |  
+-----+-----+  
| Admin1970 | $2y$10$lo/iYvMVo9pSQuhjyuBxrh.9JMA1UnnrVY1aXEEedNAZfWIyY5snK2. |  
| patricio2 | $2y$10$h.fkAyZeTrPoPwEH2w0LfuLEgg09Z.i.NGwpasDvqf77mD2Nkx3xW |  
| jessileo1974 | $2y$10$tZT58igSfw1VINd6/ap6SuTXkgEE.wdMLcv2fK1SZ/TWAIG0Uczwa |  
| jorge2019 | $2y$10$0A/1EY7HhD0p.BFQXzNAGu0f8rg80KJ.ex3lEoa20Pp2tq0b8y8nC |  
| jimmy123 | $2y$10$iJcfDD8KqHks6uOP15tmR.1tRifonryepYBE8bLJBQY32H2N0.zzW |  
| yomar123 | $2y$10$y6nh5i3ViMhbS.2bXPPua0seoNlsgas2ombc403cPbSFPa64Y1H0K |  
| eva12345 | $2y$10$5RItwHzLQ6FB.CAzcAidxe6jmeZ/m0Eq190ZYaHGOpCwSN0Mc8mCS |  
+-----+  
7 rows in set (0.000 sec)  
MariaDB [isasc_database]> |
```

Figura 55. Base de Datos Query

Fuente: Elaboración propia

El resultado te ingresa al módulo de clientes:

Clientes						
 						
Nombre	Buscar texto					
Cliente	Tipo de Documento	Número Documento	Teléfono	Email	Dirección	Editar
vanessa Quispe	CEDULA	90438473874	2834784	vanessa@gmail.com	ave. libertad	
Adam Montesco	CEDULA	3874834	24040404	adam@hotmail.com	Ave. juan de la cruz	
Helena	CEDULA	67364373	2439049304	helena@gmail.com	Plaza murillo	

Figura 56. Modulo Clientes

Fuente: Elaboración propia

3.5.4.2. Pruebas automatizadas

Login Test

```
4
5  use Tests\TestCase;
6  use Illuminate\Foundation\Testing\WithFaker;
7  use Illuminate\Foundation\Testing\RefreshDatabase;
8
9  class LoginTest extends TestCase
10 {
11     /**
12      * A basic feature test example.
13      *
14      * @return void
15      */
16     public function testExample()
17     {
18         $response = $this->get('/');
19
20         $response->assertStatus(200);
21     }
22 }
23
```

Figura 57. Login Test

Fuente: Elaboración propia

```
E:\Projects\ISASC (JimmyMayta -> origin)
λ vendor\bin\phpunit
PHPUnit 7.5.20 by Sebastian Bergmann and contributors.

.....
10 / 10 (100%)

Time: 422 ms, Memory: 18.00 MB

OK (10 tests, 10 assertions)

E:\Projects\ISASC (JimmyMayta -> origin)
λ
```

Figura 58. Resultado Login Test

Fuente: Elaboración propia

User test

```
class UserTest extends TestCase
{
    /**
     * A basic feature test example.
     *
     * @return void
     */
    public function testExample()
    {
        $response = $this->get('/main');

        $response->assertStatus(200);
    }
}
```

Figura 59. User Test

Fuente: Elaboración propia

```
E:\Projects\ISASC (JimmyMayta -> origin)
λ vendor\bin\phpunit
PHPUnit 7.5.20 by Sebastian Bergmann and contributors.

.....
10 / 10 (100%)

Time: 422 ms, Memory: 18.00 MB

OK (10 tests, 10 assertions)
```

```
E:\Projects\ISASC (JimmyMayta -> origin)
λ
```

Search

Figura 60. Resultado User Test

Fuente: Elaboración propia

Usuario test

```
9  class UsuariosTest extends TestCase
10 {main/user
11 /**
12  * A basic feature test example.
13 *
14  * @return void
15 */
16 public function testExample()
17 {
18     $response = $this->get('/main/user');
19
20     $response->assertStatus(200);
21 }
22 }
```

Figura 61. Usuario test

Fuente: Elaboración propia

```
E:\Projects\ISASC (JimmyMayta -> origin)
λ vendor\bin\phpunit
PHPUnit 7.5.20 by Sebastian Bergmann and contributors.

.....
10 / 10 (100%)

Time: 422 ms, Memory: 18.00 MB
OK (10 tests, 10 assertions)

E:\Projects\ISASC (JimmyMayta -> origin)
λ
```

Figura 62. Resultado Usuario Test

Fuente: Elaboración propia

Proveedores

```
2
3 namespace Tests\Feature;
4
5 use Tests\TestCase;
6 use Illuminate\Foundation\Testing\WithFaker;
7 use Illuminate\Foundation\Testing\RefreshDatabase;
8
9 class ProveedoresTest extends TestCase
10 {
11     /**
12      * A basic feature test example.
13      *
14      * @return void
15     */
16    public function testExample()
17    {
18        $response = $this->get('/Proveedores');
19
20        $response->assertStatus(200);
21    }
22 }
23
```

Figura 63. Proveedores Test

Fuente: Elaboración propia

```
E:\Projects\ISASC (JimmyMayta -> origin)
λ vendor\bin\phpunit
PHPUnit 7.5.20 by Sebastian Bergmann and contributors.

.....
10 / 10 (100%)

Time: 422 ms, Memory: 18.00 MB
OK (10 tests, 10 assertions)

E:\Projects\ISASC (JimmyMayta -> origin)
λ
```

Figura 64. Resultado Proveedores test

Fuente: Elaboración propia

Productos test

```
3  
9     class ProductosTest extends TestCase  
0     {  
1         /**  
2          * A basic feature test example.  
3          *  
4          * @return void  
5          */  
6         public function testExample()  
7         {  
8             $response = $this->get('/productos');  
9             $response->assertStatus(200);  
0             $response->assertBodyContains('Hello World');  
1         }  
2     }  
3
```

Figura 65. Productos Test

Fuente: Elaboración propia

```
E:\Projects\ISASC (JimmyMayta -> origin)  
λ vendor\bin\phpunit  
PHPUnit 7.5.20 by Sebastian Bergmann and contributors.  
.....  
10 / 10 (100%)  
Time: 422 ms, Memory: 18.00 MB  
OK (10 tests, 10 assertions)
```

Figura 66. Resultado Productos Test

Fuente: Elaboración propia

CAPÍTULO IV

CONCLUSIONES Y

RECOMENDACIONES

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

A lo largo del presente proyecto de grado, se ha logrado alcanzar satisfactoriamente los objetivos planteados.

- Se logró automatizar los procesos de movimiento de las ventas, brindando un control detallado, reduciendo perdida de productos, además brindando información actual, verídica, pudiendo tomar decisiones en tiempos oportunos, es decir que esté disponible la información cuando se la requiera.
- Se realizó la ingeniería de requerimientos para la recolección de datos del sistema de información.
- Se diseñó el diagrama de entidades y relaciones del sistema de información.
- Se determinó los módulos a partir de la información recopilada para el desarrollo del sistema
- Se efectuó pruebas de control y calidad aplicando la ISO 25000
- Se realizó las estimaciones del costo del software utilizando la metodología COSMIC con CFP (Cosmic Function Points) se utilizaron las fórmulas de Costo por Puntos de Función.

4.2. RECOMENDACIONES

Tras la culminación del presente proyecto de grado, se sugieren las siguientes recomendaciones:

- Cuando se requiera la ampliación y creación de nuevos módulos, se recomienda primero revisar la documentación para tomar una buena decisión, ya que el sistema presenta elementos reutilizables para el desarrollo de módulos nuevos.
- Se recomienda la utilización de herramientas de programación brindadas por PHP y JavaScript debido a la interfaz amigable para el desarrollador.
- Se debe tener cuidad respecto a las claves de acceso, siempre mantenerlas seguras y protegidas, también cambiarlas con regularidad.
- Se recomienda a la empresa, implementar, utilizar y administrar el sistema de acuerdo las instrucciones brindadas.

- Realizar el backup de toda información almacenada por lo menos al final de cada mes, para evitar problemas jurídicos puesto que se manejan facturas que son prueba de una transacción monetaria.
- La revisión periódica por cierto periodo de tiempo es recomendable para mantener la eficiencia y un funcionamiento adecuado del sistema.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

- Acosta Véliz, M., Salas Narváez, L., Jiménez Cercado, M., y Guerra Tejada, A. M. (2018). La administración de las ventas. Conceptos Clave en el Siglo XXI. España: Editorial Área de Innovación y Desarrollo, S.L.
- Achour, M., Betz, F., Dovgal, A., Lopes, N., Magnusson, H., Richter, G., . . . Vrana, J. (2020). Manual de PHP. Recuperado el 23/10/2020
- Agile Alliance. (2017). Extreme Programming. Recuperado el 23/10/2020, 2020, de <https://www.agilealliance.org/glossary/xp/>
- Beal, V. (2020). Programming language. Recuperado el 23/10/2020, 2020, de https://www.webopedia.com/TERM/P/programming_language.html
- Business News Daily. (2020). What Is Agile Scrum Methodology? Recuperado el 23/10/2020, 2020, de <https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html>
- Clarke Bloomfield, M., Cisneros Arias, Y., y Paneca González, Y. (2018). Gestión Comercial: diagnóstico del atractivo y rentabilidad del punto de ventas. Ciencias Holguín, 24 (4), 1-11.
- COSMIC. (2019). The standard methodology for sizing software: Cosmic.
- Del Valle Roque, D. (2014). Estimación de costos de desarrollo de software. Recuperado el 23/10/2020, 2020, de <https://www.gestiopolis.com/estimacion-de-costos-de-desarrollo-de-software/>
- Díaz, C. E., y Medina, D. (2010). Implementación de un sistema informático especializado para el control y reservación de boletería en la Cooperativa de Transportes Aerotaxi. Escuela Politécnica del Ejército, Ecuador.
- Eastwood, B. (2020). The 10 most popular programming languages to learn in 2020. Recuperado, de <https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/>
- Enrique, C. (2017). Estándares de Calidad en el Diseño de Algoritmos y Construcción de Programas. Recuperado el 23/10/2020, 2020,

- Fontes, S., García-Gallego, C., Quintanilla, L., Rodríguez, R., Rubio, P., y Sarriá, E. (2015). Fundamentos de investigación en Psicología. Madrid: Universidad Nacional de Educación a Distancia.
- Gupta, S., y Mittal, A. (2008). Introduction to database management system. New Dheli, India: Laxmi Publications (P) Ltd.
- Hernández, R., Fernández, C., y Baptista, P. (2006). Metodología de la investigación (Cuarta ed.). México D.F.: McGraw-Hill Interamericana.
- Hoffman, A. (2020). Web Application Security. Exploitation and Countermeasures for Modern Web Applications. United States of America: O'Reilly Media, Inc.
- Ibáñez, B. (2002). Manual para la elaboración de tesis (Segunda ed.). México: Trillas.
- ISO. (2017). Quality management standards. Recuperado el 23/10/2020, 2020, de <https://www.iso.org/about-us.html>
- Johnston, J. (2019). How to build a web app: A beginner's guide. Recuperado el 23/10/2020, 2020, de <https://www.budibase.com/blog/how-to-make-a-web-app/>
- Juba, S., y Volkov, A. (2019). Learning PostgreSQL 11. A beginner's guide to building high-performance PostgreSQL database solutions. Birmingham, United Kingdom: Packt Publishing Ltd.
- Lara Castro, L. O., y Vaca Sierra, T. N. (2008). Análisis Comparativo de Metodologías Web para el Desarrollo de un Webquest En Entornos Virtuales de Aprendizaje. Paper presented at the Programa de Maestría en Ingeniería del Software, Ibarra, Ecuador.
- Limachi, C. (2018). Sistema Web para el control y seguimiento de ventas de productos artesanales. Caso: Bolivia Tech Hub. Universidad Mayor de San Andrés, La Paz, Bolivia.

Linx Srl. (2017). Diseño Web en Bolivia. Recuperado el 01/10/2020, 2020, de <http://www.linxs.com.bo/diseno-web-bolivia.html>

Lungariello, R. (2017). The 10 Best Database Software Systems For Business Professionals. Recuperado el 23/10/2020, 2020, de <https://mytechdecisions.com/it-infrastructure/10-best-database-software-systems-business-professionals/>

MariaDB Foundation. (2020). About MariaDB Server. Recuperado el 15/10/2020, de <https://mariadb.org/about/>

Marín, R. (2019). Los gestores de bases de datos más usados en la actualidad. Recuperado el 23/10/2020, 2020, de <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>

Molina Ríos, J. R., y Zea Ordoñez, M. (2017). Metodologías de desarrollo en aplicaciones Web. Revista Arjé, 11 (21), 245-270.

Molina Ríos, J. R., Zea Ordóñez, M. P., Contento Segarra, M. J., y García Zerda, F. G. (2017). Estado del arte: Metodologías de desarrollo en aplicaciones Web. 3C Tecnología, 6 (3), 54-71.

Niederst, J. (2018). Learning Web Design. A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. Canadá: O'Reilly Media, Inc.

Obe, R., y Hsu, L. (2018). PostgreSQL: Up and Running. United States of America: O'Reilly Media, Inc.

Silberschatz, A., Korth, H., y Sudarshan, S. (2020). Databases systems concepts. New York: McGraw-Hill Education.

Vasquez, C. E. (2018). Estimación de Software con COSMIC. Recuperado el 23/10/2020, 2020, de <https://sg.com.mx/revista/49/estimacion-software-cosmic>.

ANEXOS

ANEXOS A

Árbol de objetivos y problemas

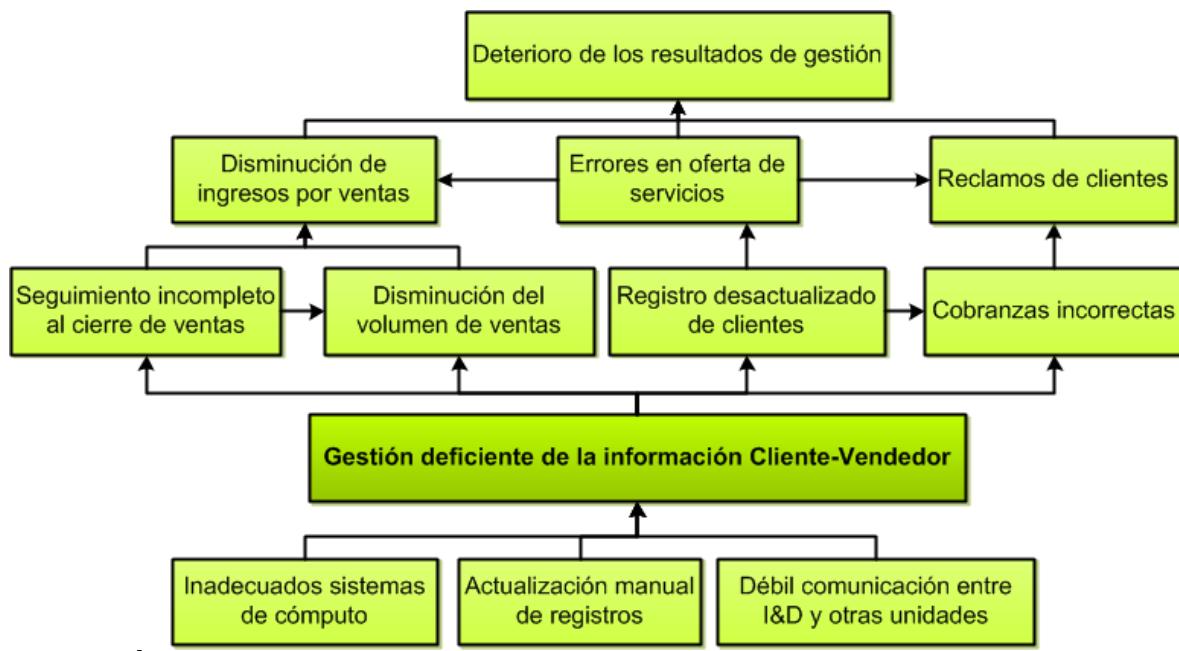


Figura. Árbol de problemas

Fuente: Elaboración propia con base en la información preliminar recolectada.

ANEXOS B

Guía de observación

CONTEXTO DE OBSERVACIÓN	ÁREA DE INTERÉS	¿QUÉ?	¿CÓMO?	¿POR QUÉ?
Contexto Físico	Información sobre la distribución de las oficinas, la ubicación de las zonas comunes, la en división del espacio dentro de las oficinas	¿Qué es lo que está haciendo? ¿Requiere esfuerzo? ¿Parece que está agitado? ¿En dolor? ¿Feliz? ¿La actividad impactando al usuario positivamente o negativamente?	¿Cómo lo está haciendo? ¿Requiere esfuerzo? ¿Parece que está agitado? ¿En dolor? ¿Feliz? ¿La actividad impactando al usuario positivamente o negativamente?	¿Por qué está haciendo lo que está haciendo? ¿La actividad requiere que se hagan conjeturas o deducciones sobre las emociones. También en esta etapa es donde se le preguntará al usuario
Contexto Social	Organigrama del personal y las oficinas, las características de las personas que trabajan allí, sus funciones y tareas.	e. Reportar todos los hechos.	?	motivaciones o emociones.

Interacciones Formales	Comunicación entre los distintos niveles jerárquico de la empresa, tipología de las decisiones.	supuestos y con frecuencia se descubrirán hallazgos inesperados .
	Procesos (¿Quién es consultado? ¿De qué forma? ¿Se hacen reuniones? ¿Cómo? ¿Cuándo se convocan? ¿Cuáles son las consecuencias si una decisión resulta errónea?).	
Interacciones Informales	Comportamiento cotidiano	
Fuente:	Geilfus (1997)	

Guía de entrevista

1 Introducción

Presentación del entrevistador

Porqué estoy aquí

Presentación de la metodología

2 Información general

El cargo que desempeña

Tiempo de estancia en la empresa

Cambios en los cargos que ejerció

3 Caracterización de la empresa

Aspectos destacables

Principales problemas

Relaciones entre áreas

4 Caracterización de la unidad donde trabaja

Aspectos destacables

Principales problemas

Relaciones entre compañeros

5 Comentarios adicionales

7 Conclusión

Qué se hará después con lo conversado

Agradecimientos

Fuente: Elaboración propia

ANEXOS C

Aval tutor metodológico

El Alto, diciembre de 2020

Señor
Ing. David Carlos Mamani Quispe
DIRECTOR
CARRERA INGENIERÍA DE SISTEMAS - UPEA
Presente.-

REF.: AVAL DE CONFORMIDAD

Distinguido ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado "**SISTEMA DE INFORMACIÓN PARA LA ADMINISTRACIÓN Y CONTROL DE VENTAS. CASO: LINXS SRL**", elaborado por el universitario: **JIMMY YOMAR MAYTA JIMENEZ**, con cedula de identidad: **6913193 L. P.** y registro universitario: **15000111**, para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.


Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO

Aval tutor especialista

El Alto, noviembre de 2020

Señor
Ing. David Carlos Mamani Quispe
DIRECTOR
CARRERA INGENIERÍA DE SISTEMAS - UPEA
Presente.-

REF.: AVAL DE CONFORMIDAD

Distinguido ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado "**SISTEMA DE INFORMACIÓN PARA LA ADMINISTRACIÓN Y CONTROL DE VENTAS. CASO: LINXS SRL**", elaborado por el universitario: **JIMMY YOMAR MAYTA JIMENEZ**, con cedula de identidad: **6913193 L. P.** y registro universitario: **15000111**, para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.



Lic. Mario Torrez Cupiticona
TUTOR ESPECIALISTA

Aval tutor revisor

El Alto, diciembre de 2020

Señor
Ing. David Carlos Mamani Quispe
DIRECTOR
CARRERA INGENIERÍA DE SISTEMAS - UPEA
Presente.-

REF.: AVAL DE CONFORMIDAD

Distinguido ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado "**SISTEMA DE INFORMACIÓN PARA LA ADMINISTRACIÓN Y CONTROL DE VENTAS. CASO: LINXS SRL**", elaborado por el universitario: **JIMMY YOMAR MAYTA JIMENEZ**, con cedula de identidad: **6913193 L. P.** y registro universitario: **15000111**, para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.

Lic. Jesus Juan Rocha Vera
TUTOR REVISOR

Aval institucional

LINXS S.R.L.

El Alto, 19 de noviembre de 2020

El (la) suscrito (a) JEFE DE UNIDAD DE RRHH, en uso de sus atribuciones.

CERTIFICA:

Que, siendo auspiciantes del proyecto de grado del Sr. **JIMMY YOMAR MAYTA JIMENEZ**, con Cedula de Identidad: **6913193 LP** y Registro Universitario: **15000111**, quien desarrolló su trabajo con el tema "**SISTEMA DE INFORMACIÓN PARA LA ADMINISTRACIÓN Y CONTROL DE VENTAS**" es grato informar que el proceso de implementación del sistema de información ha superado satisfactoriamente, asimismo, las pruebas técnicas y los requerimientos de la institución, por lo que extiendo el presente documento como constancia de aceptación y finalización del proyecto.

Autorizo al Sr. **JIMMY YOMAR MAYTA JIMENEZ**, puede hacer uso de este documento para los fines pertinentes en la Universidad Pública de El Alto.

Atentamente.


Simon Callisaya Q.

RECURSOS HUMANOS



ANEXOS D

Manual de usuario

Manual de Usuario

SISTEMA DE INFORMACIÓN PARA LA ADMINISTRACIÓN Y CONTROL DE VENTAS



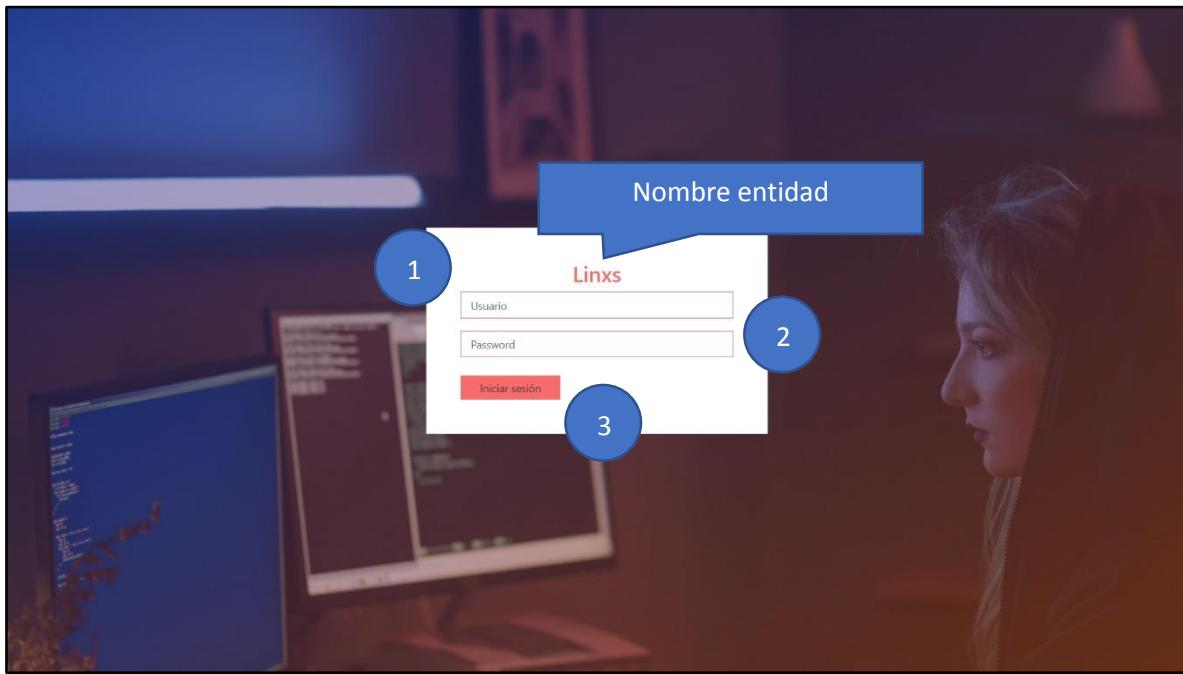
Desarrollador: Jimmy Yomar Mayta Jimenez

Supervisor: Simon Callisaya Q.

Sistema: ISASC

1. Login (Inicio de sesión)

El Login una interfaz de acceso al sistema con un usuario y contraseña.



- 1) Usuario: Se debe ingresar un usuario ya designado por el administrador
- 2) Password: Se debe ingresar una contraseña
- 3) Se debe presionar “Iniciar sesión” para ingresar al sistema

A screenshot of a web-based application interface titled 'Linxs'. On the left is a sidebar with navigation links: 'Cuentas', 'Categorías', 'Productos', 'Compras', 'Proveedores' (which is highlighted in purple), 'Ventas', 'Usuarios', 'Roles', and 'PDF Reporte'. The main content area shows a table titled 'Clientes' with columns: 'Cliente', 'Tipo de Documento', 'Número Documento', 'Teléfono', 'Email', 'Dirección', and 'Editar'. The table contains three rows of data. At the bottom of the table is a navigation bar with buttons for 'Anterior', page numbers '2 3 4 5', and 'Siguiente'. A blue callout bubble points to the top right of the table with the text 'Buscador'. In the bottom left corner of the main area, there is a small 'BI' icon.

Ingresamos al sistema y dependiendo de los permisos nos aparecerá diferentes módulos.

Linxs

Clients

Categorías

Productos

Compras

Proveedores

Ventas

Usuarios

Roles

Bancos

Tarjetas

BI

Clientes

Nombre Buscar texto

Cliente Tipo de Documento Número Documento Teléfono Email Dirección Editar

Vela	CEDULA	dasd	asdas	dasd	asd	
SARA	CEDULA	fdsaf	dafsd	fadsf	fasf	
vaensa	CEDULA	sdfd	sdf	asdfasd	fsdf	

Anterior 1 2 3 4 5 Siguiente

1

jimmy Cuenta Cerrar sesión

Tabla de clientes

- 1) Para salir o cerrar sesión click en tu nombre o usuario, seleccionar la opción "Cerrar sesión"

2. Módulos

2.1. Modulo para administradores

El módulo de administradores contiene más módulos a diferencia de otros permisos, es capaz de crear usuarios, ingresar al BI (Business Intelligence) y realizar tareas de vendedores y compradores.

Linxs

Clients

Categorías

Productos

Compras

Proveedores

Ventas

Usuarios

Roles

Bancos

Tarjetas

BI

Categorías

Categoría Descripción Imagen Estado

Sistema de Reservas	Desarrollo de Sistemas de Reservas		Activo
Sistemas de Informacion	Desarrollo de Sistemas de Informacion		Activo
Sistemas de Seguridad	Desarrollo de Sistemas de Seguridad		Activo

+

Anterior 1 2 3 Siguiente

jimmy Cuenta

Editar Categoría

Cambiar estado

2.2. Módulo para vendedores

El módulo de vendedores es capaz de realizar ventas registrar clientes, crear productos y categorías.

2.3. Modulo Comprador

El módulo comprador simplemente pude registrar productos y categorías

3. Clientes

Módulo de clientes, se puede registrar, editar clientes

- 1) Para crear o registrar un nuevo cliente click “+” y se desplegará un formulario para llenar datos del cliente
- 2) Formulario de agregar cliente

4. Productos

El módulo de productos una interfaz amigable en donde se puede registrar productos, se puede editar y cambiar estados de los productos.

- 1) Crear o registrar un nuevo producto click “+” se desplegará un formulario que deberá llenar datos del producto
- 2) Formulario para agregar producto

Ingresamos

5. Compras

Módulo de compras que contiene una vista para ver detalle de cada compra

3

LLenar el formulario

Número Compra
45687546

Proveedor(*)
Buscar Proveedores (Linxs)

Factura - Ticket(*)
Factura

Impuesto(*)
0.12

Producto (Ingrese código del producto)
Agregar Producto

Agregar producto

Lista de Compras

Producto	Precio (Bs)	Total (Bs)	Eliminar
No se han agregado productos			

x

4

Detalle de Compra

PROVEEDOR	TIPO IDENTIFICACION	NUMERO COMPRA	IMUESTO
Linxs SRL	FACTURA	42342	0.12

Producto	Precio (Bs)	Cantidad	Total (Bs)
Software de Musica	8000.00	1	8000
Sub-Total: 7040.00 (Bs)			
Impuesto: 960.00 Bs			
Total: 8000.00 Bs			

x **Cerrar modulo**

- 1) Crear o registrar una compra click “+” desplegará un formulario para registrar datos
- 2) Para ver detalle de cada compra click “Ojo” abrirá un modulo de “Detalle de compra”

6. Lista proveedores

Lista de proveedores de diferentes entidades

Linxs

Cuentas Categorías Productos Compras Proveedores Ventas Usuarios Roles Bancos Tarjetas BI

Listado de Proveedores

Nombre Buscar texto

Proveedor	Tipo de Documento	Número Documento	Teléfono	Email	Dirección	Editar
Linxs SRL	CEDULA	0963452346	2 2141908	info@linxs.com.bo	C. Jorge Saenz Nro. 1263 PB (Miraflores), La Paz	
MegaLink	CEDULA	1034678396	2 2129000	info.MegaLink24@yahoo.com	Edificio Anibal, Planta Baja, Sanchez Lima 2520, La Paz	
SumaDevs	CEDULA	09103456788	28237865	luis36@hotmail.com	Cdla Samanes 4 Mz 238 Villa 10	

Anterior 1

Editar

7. Ventas

Módulo de ventas, cuenta con un buscador, con un cambio de estado y se puede ver a detalle cada venta

Linxs

Cuentas Categorías Productos Compras Proveedores Ventas Usuarios Roles Bancos Tarjetas BI

Ventas

Número Venta Buscar texto

Ver Detalle	Fecha Venta	Número Venta	Cliente	Tipo de identificación	Vendedor	Total (Bs)	Impuesto	Estado	Cambiar estado (Eliminar)
	2020-03-23 00:00:00	35	Felix Orosco Pedraza	FACTURA	jimmy	30.00	0.12	Registrado	
	2020-03-23 00:00:00	34	Carla Cuzman Alarcon	FACTURA	jimmy	3.50	0.12	Registrado	
	2020-03-22 00:00:00	33	Carla Cuzman Alarcon	FACTURA	jimmy	3.50	0.12	Registrado	

Anterior 1 2 3 4 5 6 7 Siguiente

Cambiar estado

Detalle Ventas

CLIENTE TIPO DE IDENTIFICACIÓN NÚMERO VENTA IMPUESTO

Andres	FACTURA	4132342342	0.13
Producto	Precio (Bs)	Descuento	Total (Bs)
Publicidad avanzada	1500.00	0.00	1500
Sub-Total: 1305.00 (Bs)			
Impuesto: 195.00 (Bs)			
Total: 1500.00 (Bs)			

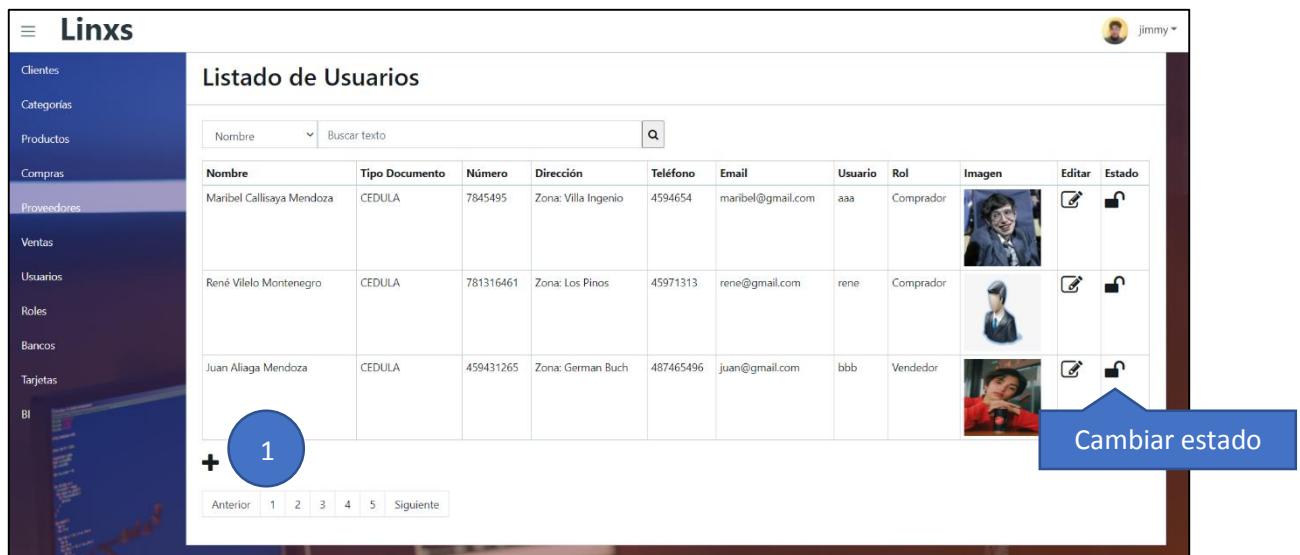
x

- 1) Para ver cada detalle de una venta click en “Ojo” para ver y desplegará un módulo de “Detalle ventas”

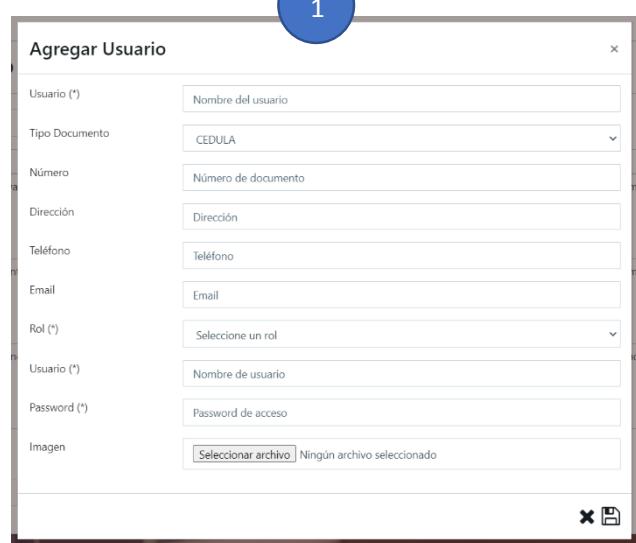
2) Modulo detalle ventas

8. Usuarios

Módulo de usuarios, donde se puede registrar parte del personas de la entidad, se puede editar, cambiar estado, agregar un fotografía del personal e incluso editar



Nombre	Tipo Documento	Número	Dirección	Teléfono	Email	Usuario	Rol	Imagen	Editar	Estado
Maribel Callisaya Mendoza	CEDULA	7845495	Zona: Villa Ingenio	4594654	maribel@gmail.com	aaa	Comprador			
René Vilelo Montenegro	CEDULA	781316461	Zona: Los Pinos	45971313	rene@gmail.com	rene	Comprador			
Juan Aliaga Mendoza	CEDULA	459431265	Zona: German Buch	487465496	juan@gmail.com	bbb	Vendedor			



Usuario (*)	Nombre del usuario
Tipo Documento	CEDULA
Número	Número de documento
Dirección	Dirección
Teléfono	Teléfono
Email	Email
Rol (*)	Seleccione un rol
Usuario (*)	Nombre de usuario
Password (*)	Password de acceso
Imagen	Seleccionar archivo Ningún archivo seleccionado

- 1) Crear o registrar un nuevo usuario
- 2) Llenar datos de usuario

9. BI (Business Intelligence)

Módulo de Business Intelligence que ayuda a la entidad a la toma de decisiones

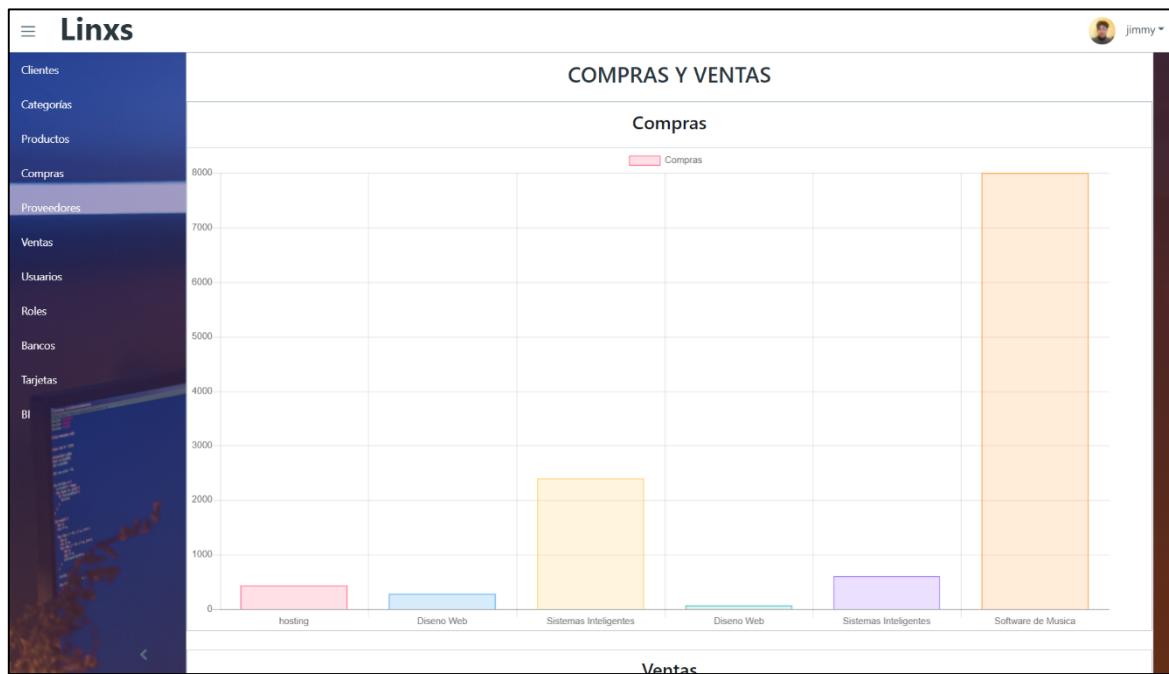


Imagen 1

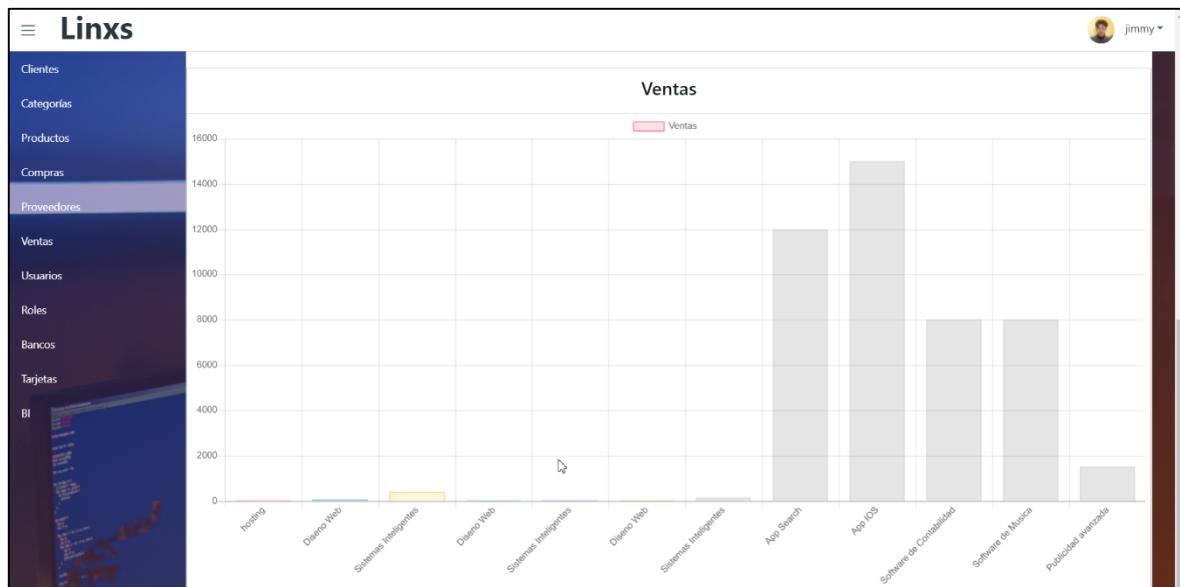


Imagen 2

Business Intelligence una herramienta que está en constante procesamiento para el análisis de datos sobre ventas de productos. Analiza una gran cantidad de datos para el apoyo de las decisiones de la entidad.