

**Diseño e Implementación de un Sistema POS, con Módulo de Gestión de
Inventario de Productos para Clientes y Perfiles de Usuario, Aplicando Metodología
RUP**

Jaime Humberto Villamil Rodríguez

Universidad Nacional Abierta y a Distancia
Escuela de Ciencias Básicas, Tecnología e Ingeniería

Programa de Ingeniería de sistemas

Bogotá, Colombia

Junio 2021

**Diseño e Implementación de un Sistema POS, con Módulo de Gestión de
Inventario de Productos para Clientes y Perfiles de Usuario, Aplicando Metodología
RUP**

Jaime Humberto Villamil Rodríguez

Trabajo de grado presentado como requisito para optar al título de:

Ingeniero de sistemas

Director (a):

Ingeniero: Ismael Ángel Romero

Universidad Nacional Abierta y a Distancia

Escuela de Ciencias Básicas, Tecnología e Ingeniería

Programa de ingeniería de sistemas

Bogotá, Colombia

Junio 2021

Nota de aceptación:

Presidente del Jurado

Firma Jurado

Firma Jurado

No importa cuánto tiempo te tomes para alcanzar tus metas, el mundo no lo creó Dios en un solo día. Solo tú sabes el arduo trabajo al que te has sometido para llegar a la meta que te habías propuesto.

Llegado el momento, grita desde el fondo de tu corazón, LO LOGRE.

J.V.

Dedicatoria

A mi esposa Luz, que ha sido el complemento idóneo durante esta etapa y muchas otras, es ella quien ha estado al frente de nuestro hogar mientras yo dedico horas de estudio, sacrificando su tiempo y ofreciéndolo a mi causa, para ella mis respetos, amor y cariño.

A nuestros hijos Cynthia, Andrés y Sofia, porque han comprendido la importancia de soñar y nunca dejar de luchar para que esos sueños se realicen, ellos son la razón de ser, que ha impulsado todo este esfuerzo.

A mi madre la mujer valiente que me enseñó todo en la vida, se esforzó por brindarme en mis primeros años de vida la educación inicial, sin esa base no lo habría podido lograr.

A mis hermanos que anhelan orgullosos verme terminar este proceso, ellos han sabido esperar junto a sus hogares y tomar como ejemplo que nunca es tarde para alcanzar las metas.

A todas las personas que sin importar la edad han decidido cumplir sus sueños, que no se han detenido por las circunstancias y obstáculos que se han atravesado en la vida.

Agradecimiento

Mis palabras de agradecimiento primero a Dios todopoderoso que me ha permitido vivir este momento, es él quien me ha guiado en una comunicación personal, antes y después de todo el proceso de investigación y desarrollo de este proyecto.

Al ingeniero Ismael Ángel Romero por su valiosa colaboración en la materialización de las ideas del proyecto, sus aportes permitieron aterrizar de la manera más profesional cada línea de este documento.

Resumen

En el presente proyecto se evidencia el diseño y la implementación de un nuevo sistema POS (Punto de Venta) basado en ambiente Web para la empresa Broxxo Jeans que permite mejorar el sistema anterior de punto de venta, cambiar la imagen corporativa de la compañía, atraer nuevos clientes, aprovechando las ventajas de las tecnologías en la red de Internet, publicar una nueva línea de productos (ropa para dama y caballero) desde un catálogo que detalla cada uno de los productos, visualizar las características de cada uno de ellos, permitiendo al cliente tomar la decisión con mayor facilidad del producto que desea adquirir. Al mismo tiempo, como ventaja adicional, será posible registrar los clientes, alojar los datos en una base de datos y realizar seguimiento a cada uno de ellos.

El desarrollo se fundamenta en la aplicación de la metodología RUP (Rational Unified Process) la cual utiliza el lenguaje de Modelado Unificado (UML), este lenguaje tiene la ventaja de diseñar todos los diagramas de un sistema de software, fundamentado en tres principios básicos: Dirigido por casos de uso, Centrado en la arquitectura e Iterativo e incremental.

En la etapa de inicio del desarrollo web se contempla el establecimiento de requerimientos de negocio y requerimientos técnicos, definiendo las necesidades y deseos de la organización con relación a la consecución del proyecto, los técnicos se centrarán en las soluciones que harán posible la aplicación de las metas del proyecto.

Posterior a la etapa inicial se desarrollarán los prototipos de las vistas de las interfaces web, los contenidos, los menús, accesos y formularios, alcanzar el diseño web adaptable (diseño responsivo) como propósito de funcionalidad de las interfaces en cualquier dispositivo y resolución de pantalla.

El desarrollo se continua con la construcción del CRUD (Créate, Read, Update, Delete), aplicable a todos los gestores del sistema para punto de venta, gestor de inventarios, de usuarios, de clientes, de facturación.

Para el desarrollo del código se utiliza el patrón MVC (Modelo Vista Controlador) desarrollo en tres capas o niveles, implementándole una capa de seguridad al diseño. Sobre todos los gestores del sistema se implementa conexión a la base de datos MySQL utilizando la clase PHP PDO conexiones seguras.

Palabras claves: Sistema POS, metodología RUP, desarrollo WEB, Construcción CRUD.

Abstract

This project shows the design and implementation of a new POS (Point of Sale) system based on a Web environment for the company Broxxo Jeans that allows improving the previous point of sale system, changing the corporate image of the company, attracting new customers, taking advantage of the technologies in the Internet, publish a new line of products (clothing for women and men) from a catalog that details each of the products, view the characteristics of each one of them, allowing the customer make the decision more easily about the product to purchase. At the same time, as an additional advantage, it will be possible to register customers, host the data in a database and track each one of them.

The development is based on the application of the RUP (Rational Unified Process) methodology which uses the Unified Modeling language (UML), this language has the advantage of designing all the diagrams of a software system, based on three basic principles: Use Case Driven, Architecture Focused, and Iterative and Incremental.

In the initial stage of web development, the establishment of business requirements and technical requirements is contemplated, defining the needs and desires of the organization in relation to the achievement of the project, the technicians will focus on the solutions that will make possible the application of the project goals.

After the initial stage, the prototypes of the views of the web interfaces, the contents, the menus, accesses, and forms will be developed, to achieve the adaptive web design (responsive design) as the purpose of the functionality of the interfaces on any device and screen resolution.

The development continues with the construction of the CRUD (Create, Read, Update, Delete), applicable to all system managers for point of sale, inventory manager, users, customers, billing.

For the development of the code, the MVC pattern (Model View Controller) development in three layers or levels is used, implementing a security layer to the design. Connection to the MySQL database is implemented using the PHP PDO class secure connections on all system managers.

Keywords: POS Systems, RUP methodology, Web development, CRUD construction.

Contenido

Agradecimiento	6
Resumen	7
Abstract	9
Contenido	11
Lista de Figuras	15
Lista de Tablas	17
Introducción	18
Planteamiento del Problema	20
Justificación	21
Objetivos	23
3. Objetivo General.	23
3.1 Objetivos Específicos.	23
Marco Referencial	24
4. Antecedentes.	24
4.1.1 Software Punto de Venta POS (Point off Sale)	25
4.2 Marco Legal	26
4.3 Marco Conceptual	27
4.3.1 Metodología RUP	27
4.3.2 Fases e Iteraciones de la metodología RUP	28
4.3.3 Fase de Iniciación metodología RUP	29

	12
4.3.4 Fase de elaboración.....	29
4.3.5. Fase de Construcción.....	30
4.3.6 fase de Transición	31
4.4 UML Introducción.....	32
4.4.1 Notación básica de UML	33
4.4.2 Modelos	33
4.4.3 Diagrama de Estructura Estática	34
4.4.4 Diagrama de Casos de uso.....	34
4.4.5 Elementos de un Caso de Uso	34
4.4.6 Actores	34
4.4.7 Casos de Usos	34
4.4.8 Relaciones entre Casos de Uso.....	35
4.4.9 Conclusiones	37
4.5 Introducción a las Bases de Datos.....	38
4.5.1 ¿Qué es una Base de datos?	39
4.5.2. Gestor de Bases de Datos (SGBD)	40
4.5.3 Lenguajes de aplicación y lenguajes de datos	41
4.6 Introducción a PHP Lenguaje de Programación	43
4.6.1. Interfaces API para el desarrollo POS.....	44
4.6.2 Conectores en PHP para el desarrollo POS	44
4.6.3. Controladores en PHP para el desarrollo POS.....	45
4.6.4. Bootstrap (Framework).....	45
4.6.5. CSS (Hojas de estilo en cascada)	46

	13
4.6.6 Introducción a HTML	47
4.6.7. Protocolo HTTP	48
4.6.8 HTML 5	48
4.7 Marco Teórico	49
Diseño Metodológico	51
5. Fase 1, etapa de Iniciación	52
5.1 Implementación UML casos de Uso Sistema POS	56
5.2 Fase 2 Maquetación del diseño POS	62
5.2.1 Maquetación de los módulos de gestión del sistema POS	64
5.2.2 Modulo de usuarios.....	64
5.2.3 Maquetación del Módulo de venta del sistema POS.....	66
5.2.4 Módulo Administrar venta	67
5.2.5 Módulo Crear venta.....	68
5.2.6 Módulo de Reportes.....	69
5.2.7 Módulo de Inventarios.....	70
5.3.1 Modularizarían del patrón MVC en el desarrollo POS	73
5.4 Construcción del Modelo CRUD para las bases de datos.....	74
5.4.1 Diseño de las bases de datos para el sistema POS.....	75
5.4.2 Modelamiento del diseño lógico de la Base de Datos sistema POS	77
5.4.3 Diseño Físico de la Base de Datos sistema POS.....	79
5.4.4 Descripción de las tablas del sistema POS.....	80
5.4.4 Arquitectura del sistema POS	86
Resultados y Análisis.....	88

	14
6 Resultados	88
6.1 UML el componente estratégico del desarrollo POS	89
6.2 La Maquetación de las interfaces web, las ventajas del uso de plantillas	89
6.3 MySQL y PhpMyAdmin integran el SGBD de la base de datos del POS	89
6.4 Análisis	89
6.5 Pruebas de Producto final	90
Conclusiones & Recomendaciones	92
Conclusiones	92
Recomendaciones	93
Referencias	94
ANEXOS	96
Anexo 1	96

Lista de Figuras

Figura 1.....	27
Figura 2.....	28
Figura 3.....	35
Figura 4.....	36
Figura 5.....	36
Figura 6.....	37
Figura 7.....	51
Figura 8.....	52
Figura 9.....	56
Figura 10.....	57
Figura 11.....	58
Figura 12.....	59
Figura 13.....	60
Figura 14.....	61
Figura 15.....	62
Figura 16.....	63
Figura 17.....	65
Figura 18.....	66
Figura 19.....	67
Figura 20.....	68
Figura 21.....	69
Figura 22.....	71
Figura 23.....	72

Figura 24.	73
Figura 25.	76
Figura 26.	78
Figura 27.	87

Lista de Tablas

Tabla 1.	29
Tabla 2.	30
Tabla 3.	31
Tabla 4.	32
Tabla 5.	42
Tabla 6.	47
Tabla 7.	53
Tabla 8.	54
Tabla 9.	55
Tabla 10.	75
Tabla 11.	80
Tabla 12.	81
Tabla 13.	82
Tabla 14.	83
Tabla 15.	83
Tabla 16.	84
Tabla 17.	85
Tabla 18.	85
Tabla 19.	91

Introducción

Actualmente las pequeñas y medianas empresas (Pymes) no tienen un crecimiento tecnológico acorde con el desarrollo y demanda actual, se evidencia que en el mercado el crecimiento y desarrollo tecnológico avanzan rápidamente, por el contrario, las PYMES no realizan inversiones suficientes en tecnología, lo que genera un retraso en los proyectos de innovación a nivel corporativo, creando un ambiente negativo en la relación de comercio B2C, comercio entre empresas y consumidores. No se tiene en cuenta que, la inversión en innovación produce un efecto positivo en las organizaciones llegando incluso a generar cambios que mejoran la estructura organizacional. (Fernandez Portillo, Sanchez Escobedo, Jiménez Naranjo, & Hernandez Mogollon, 2015)

En los últimos años han surgido tecnologías de desarrollo de aplicaciones web que permiten a las empresas mejorar o migrar sus programas de gestión de clientes y puntos de venta a aplicaciones flexibles con capacidad de mantener un alto consumo de transacciones sin afectar su rendimiento, mejorando la interacción con el cliente de forma ágil y segura. Los lenguajes seguros y abiertos como PHP integrándose con bases de datos garantizan la protección de la información del cliente y optimizan los procesos en los sistemas operativos.

El presente proyecto se enfoca en la implementación de un sistema POS (Punto de Venta) para la organización Broxxo Jeans Company; actualmente la empresa cuenta con un sistema POS implementado hace más de 10 años con poca capacidad de crecer debido a su tecnología y dificultades de integración con otras tecnologías. La empresa Broxxo Jeans motivada por las nuevas generaciones de aplicaciones web considera apropiado migrar la actual aplicación POS a un sistema completo que integre clientes y productos por medio de un catálogo en un ambiente web de fácil acceso, sin necesidad de depender de un escritorio.

El desarrollo del proyecto cuenta con la implementación de metodologías ágiles como RUP (Rational Unified Process) utilizado en los procesos de construcción de software, RUP es un proceso basado en modelos y por componentes, orientado a los casos de uso describe en lenguaje UML lo que el usuario requiere incluyendo la secuencia completa de iteraciones entre el usuario y el sistema. (Pérez A., 2011)

Durante toda la implementación se realizará un proceso constante de comunicación con el cliente, sobre las diferentes actualizaciones y cambios que se vayan a ir realizando a medida que se ejecutan las tareas asignadas según el cronograma del diseño del proyecto.

Planteamiento del Problema

Broxxo Jeans es una empresa del sector de la moda en Colombia de ropa informal para dama y caballero, actualmente dedicada al diseño, la confección y venta de sus productos, con 10 puntos de venta (POS) a nivel nacional. Teniendo en cuenta las demandas actuales del mercado, decide liberar una nueva línea de productos de ropa para dama y caballero, la cual representara una ventaja competitiva frente a las empresas dedicadas a este tipo de servicios, sin embargo, el actual sistema de punto de venta ha presentado fallas graves en su operatividad y se ha quedado obsoleto para soportar la carga de procesamiento que requieren la gestión de los puntos de venta de las tiendas a nivel nacional. Para dar solución al problema generado por el actual sistema de punto de venta, se planea construir un nuevo sistema de punto de venta basado en un ambiente Web que incluya un módulo de gestión de inventario de productos para clientes y perfiles de usuario.

Con este nuevo sistema POS basado en Web, Broxxo Jeans pretende ponerse a la vanguardia en el mercado, ser más eficientes en la atención a los clientes, permitirá visualizar el portafolio completo de productos, ingresar los datos de cada cliente, evitar desplazamiento, personalizar la atención. Y lo más importante, mejorar su experiencia.

Una ventaja adicional el nuevo sistema se haría más portable, escalable hacia las nuevas tecnologías, permitiría convertirlo en Intranet de uso masivo en algunos módulos como el de usuarios, ventas, clientes, en toda la organización.

Justificación

Los avances en la ciencia de la informática han permitido la evolución de la industria de los contenidos web, con las primeras versiones del HTML se crearon las primeras páginas en el ecosistema World Wide Web, con el acceso a Internet en las últimas décadas crecieron también los desarrollos web, sin embargo con la globalización en el mundo de la economía a escala mundial se generan mercados entre países y entre continentes, el panorama cambió y el comercio se mueve por la nube de internet de una manera digital, pero para poder generar negocios se necesita una cara, una vista o presentación corporativa que identifique o diferencie una empresa de otra, para ello la tecnología de la industria del software desarrolla las aplicaciones basadas en ambiente web.

Según (Huertas , 2007) un desarrollo web corporativo para ser eficiente, debe estar integrado en los planes globales de la empresa, por ello la importancia de referenciar las aplicaciones web como la imagen corporativa y representativa de la organización.

El objetivo que se pretende con este proyecto de diseño web es mejorar la imagen corporativa de la empresa Broxxo Jeans, aprovechando la red global de internet para ser identificados a nivel nacional, facilitar la comunicación con los clientes, publicar la nueva línea de productos, que pondrán a la venta, detallar cada producto mediante un catálogo de productos permitiendo visualizar sus características, vincular nuevos clientes y realizar un seguimiento mediante una base de datos.

El diseño web responderá a las nuevas exigencias del mercado en aplicaciones web orientadas a entregar soluciones en puntos de venta (POS), integrando reportes en línea, stock de inventarios, reporte de ventas, gestión de usuarios, gestión de clientes. El diseño responsivo le

permitirá al usuario acceder al portal desde dispositivos Android en general si no se tiene a la mano un monitor de alta resolución.

Objetivos

3. Objetivo General.

Diseñar un nuevo sistema POS basado en ambiente web que permita soportar la carga de procesamiento y transaccional de ventas, mejorar la imagen corporativa de la compañía Broxxo Jeans, dar un mejor desempeño en cada punto de venta y soportar la nueva línea de productos de ropa para dama y caballero.

3.1 Objetivos Específicos.

Establecer los Requerimientos funcionales para el desarrollo del nuevo sistema POS basado en ambiente web de la compañía Broxxo Jeans, aplicando la metodología RUP.

Determinar los prototipos del diseño del nuevo sistema POS basado en web, teniendo en cuenta el objetivo de los contenidos, vistas, logo, menús, formularios, accesos y perfiles de usuarios y clientes.

Construir el modelo CRUD de las bases de datos, y el modelado de clases utilizando herramientas y lenguajes Open Source.

Aplicar las políticas del diseño y administración del contenido web según la estructura organizacional de la empresa Broxxo Jeans.

Realizar las pruebas del producto final y despliegue del nuevo punto de venta.

Marco Referencial

El estado de arte para este proyecto de Investigación y aplicación se enfoca en la construcción de un software en ambiente web, con el objetivo de dar funcionalidad a un sistema POS o Punto de Venta (en inglés, Point off Sale), que cubra las necesidades de la organización Broxxo Moda Company en los sitios de atención, venta y comercialización de los productos de la empresa Broxxo Jeans.

4. Antecedentes.

La industria del software en Colombia es un modelo de desarrollo creciente, según datos de la Revista Portafolio (2019), en el país existen aproximadamente más de Seis mil (6.000) empresas en su mayoría Pymes que facturan 15 billones de pesos anuales (Ramirez, 2019), cifra muy importante para una industria en crecimiento. Sin embargo, de esa cantidad un 5 % corresponde a software especializado en Punto de Venta, lo anterior indica que es un pequeño porcentaje para la demanda que puede existir en el país sobre Puntos de venta.

Paralelo a la industria de desarrollo de software y hardware especializado en puntos de venta, este ha evolucionado, los diseños actuales tienden a ser más autónomos, casi robotizados, prueba de ello son los proyectos patentados a nivel mundial, William Bodin (United States Patente n° US 6,681,989 B2, 2004) patento un proyecto que permite realizar la compra de un producto sin la intervención del vendedor, el desarrollo implementa un sistema de seguridad que permite la identificación del cliente y el articulo a comprar, el sistema permite realizar la transacción de la cuenta del cliente y autorizar el retiro del articulo adquirido.

El siguiente es un proyecto realizado en México, presenta el diseño y el desarrollo de un sistema basado en computadoras para un punto de venta con aplicación móvil que ayuda a

mejorar las operaciones y procesos de un establecimiento que ofrece servicios de restaurante y bar. (Nuñez Islas, Ramirez Rosario, & Nava Rojo, 2016)

4.1.1 Software Punto de Venta POS (Point off Sale)

Los sistemas punto de venta POS, se utilizan en el momento del pago en un establecimiento comercial, entre sus funcionalidades están: contar los artículos, procesar el pago del artículo o los artículos, actualizar el inventario del establecimiento comercial, aplicar los descuentos o incrementos y entregar una factura al cliente. Los sistemas más robustos de punto de venta pueden realizar otras funciones adicionales, como integrarse con los programas contables de las organizaciones y afectar la contabilidad de la empresa.

Los desarrollos de Punto de Venta son considerados como un tipo de software de aplicación que cumple una tarea específica, su nivel de complejidad aumenta cuando se debe integrar con los sistemas contables de las organizaciones. Para nuestro proyecto la solución se presenta como un desarrollo apropiado para cubrir la venta de los productos de la organización Broxxo Jeans, dicho desarrollo se implementará inicialmente en un punto de venta central, su adecuación se debe hacer de una manera gradual a medida que los usuarios reciban la correspondiente capacitación de manejo y aplicación del software punto de venta.

El modelo de desarrollo propuesto para el sistema POS está basado en la metodología RUP, un estándar enfocado en la producción de software a gran escala y complejo, muy utilizado en la comunidad de producción de software.

4.2 Marco Legal

Normatividad: El desarrollo del trabajo de grado se enmarca en las normativas actuales aplicadas a los puntos de venta, el artículo 13 de la Resolución 00055 de 2016 establece que un sistema POS es el conjunto de un sistema de hardware y software que interactúa permitiendo la programación, control y ejecución de las funciones inherentes al punto de venta.

El decreto 2242 de 2015 y la Resolución 19 de 2016 indica que la facturación por el punto de venta está excluida de la facturación electrónica. Se puede aplicar facturación a través de un sistema POS en las tiendas y facturación electrónica en las demás operaciones de la compañía. Sin embargo, se requiere de una autorización de numeración para POS emitida por la DIAN.

Según el artículo 6 del decreto 1165 de 1996, todo documento emitido a través de una máquina registradora con sistema POS, debe cumplir las siguientes características:

- Razón social y Nit del que presta el servicio.
- Número consecutivo de la factura.
- Fecha de la factura.
- Descripción de los productos vendidos.
- Valor de la transacción.
- Debe tener impreso el número de resolución de la DIAN.

El artículo 14 de la Resolución 00055 de 2016 indica que las personas o entidades que utilicen para el registro de sus ventas máquinas registradoras POS o facturación por computador, deberán identificar los bienes o servicios, referenciando el departamento al cual corresponden y la tarifa del impuesto sobre las ventas e impuesto nacional al consumo asociada a cada bien o servicio.

4.3 Marco Conceptual

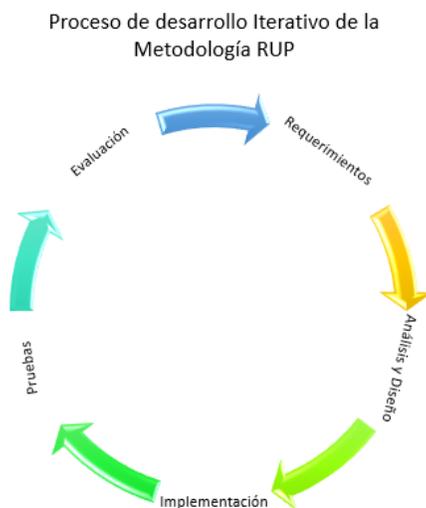
El sistema POS propuesto es un desarrollo web para la gestión y operación de los puntos de venta para la organización Broxxo Moda Company, en la actualidad Broxxo cuenta con dos puntos de venta en la ciudad de Bogotá, dos puntos de venta en la ciudad de Cali, dos puntos de venta en la ciudad de Pereira y uno en la ciudad de Medellín. Broxxo no cuenta con una comunicación centralizada, por tal razón el desarrollo se construye sobre una plataforma de gestión distribuida en cada ciudad.

4.3.1 Metodología RUP.

La metodología RUP es un proceso de Ingeniería de Software enfocado en el desarrollo y producción de software, muy utilizado en las organizaciones especializadas en el campo de la industria de aplicaciones web de alta calidad, sus procesos cubren ampliamente las necesidades de los clientes al aplicar tiempo y presupuestos estimados.

Figura 1.

Proceso de desarrollo Iterativo de la metodología RUP.



Fuente propia

La metodología RUP utiliza el Lenguaje de Modelado Unificado (UML) para preparar todos los diagramas de un sistema de software, ventaja importante respecto a otros procesos de desarrollo de software, este lenguaje permite describir en sencillos pasos la interacción del usuario y la aplicación en ejecución.

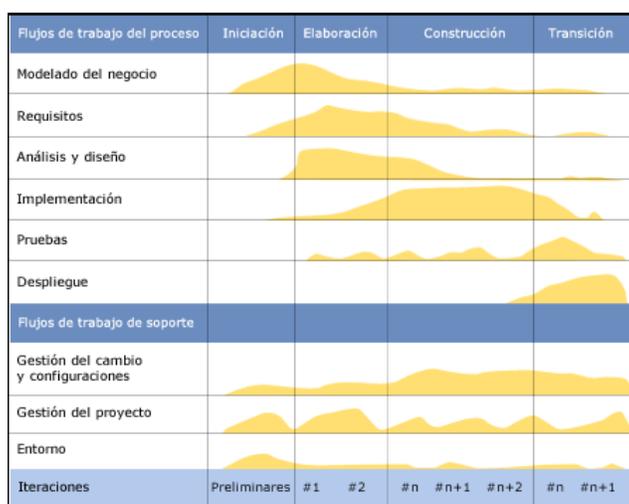
RUP se centra en los modelos utilizando el caso de uso, como una fracción del sistema que referencia los requisitos funcionales, describe que debe hacer el sistema por medio de una secuencia de acciones permitiendo una mejor visibilidad en lenguaje natural.

La importancia de los desarrollos interactivos como la metodología RUP es la relación a modo de dialogo entre los autores (usuario), y el sistema (aplicación POS), esta relación no se presenta al finalizar y entregar el producto, se inicia desde los requerimientos del diseño, donde el usuario establece criterios de uso y perspectivas del desarrollo, teniendo en cuenta esta recogida de información se adelanta el primer paso que son requerimientos funcionales.

4.3.2 Fases e Iteraciones de la metodología RUP

Figura 2.

Fases de la metodología RUP.



Fuente (Torossi , 2019)

4.3.3 Fase de Iniciación metodología RUP

La fase inicial se enfoca en la comprensión del problema y la tecnología sobre la cual se plantea una solución, las primeras iteraciones determinan el alcance del proyecto, la recogida de los requisitos es necesario identificar las asociaciones con las cuales interactuara (actores) la aplicación POS, usuarios en ambiente de punto de venta y las implicaciones que conllevan, como un entorno ágil, sin errores.

Tabla 1.

Lista de documentos al final de esta fase.

Modelo RUP
Documento con la visión del proyecto
Plan del proyecto que muestre fases
Requerimientos y características
Modelo de casos de uso
Prototipos del diseño
Evaluación de riesgo inicial
Reconocimiento de mercado

Fuente propia.

4.3.4 Fase de elaboración

El propósito de esta fase es saber si el proyecto es técnicamente viable, su importancia se basa en tener conocimiento suficiente de las soluciones para la elaboración de los diseños y los componentes de software que se aplicaran en la construcción del software.

Al finalizar esta fase se considera que la ingeniería difícil se ha completado, y que el proyecto ha superado una fase importante que determina estabilidad.

El desarrollo gana confianza en las directivas cuando se generan los primeros entregables del proyecto de desarrollo, los diseños de las vistas son relevantes, definen los prototipos estructurales y estéticos del desarrollo POS, su aprobación es el paso a la siguiente fase de construcción del código.

Tabla 2.

Fase de elaboración.

Modelo RUP
Realizar la arquitectura del software
Prototipos del diseño
Modelo de casos de uso
Revisar los requerimientos funcionales
Construir un prototipo de interfaz
Modelo de casos de uso
Prototipos del diseño
Lista revisada de riesgos
Reconocimiento de mercado

Fuente propia.

4.3.5. Fase de Construcción

La fase de construcción se caracteriza por la integración del producto de software, en esta fase se da finalización al proceso de desarrollo en todos sus módulos, se inician las primeras

pruebas de producto y se convoca al usuario final a interactuar en la ejecución del producto de software para realizar ajustes finales si es necesario.

Tabla 3.

Fase de construcción.

Modelo RUP
Actualizar plan de interacción
Revisar la lista de riesgos
Gerenciar los recursos (herramientas bases de datos)
Revisar los requerimientos funcionales
Completar el desarrollo de los componentes (prototipo funcional)
Probar los componentes contra los criterios de evaluación
Prototipos del diseño
Probar el diseño institucional, comunicacional y grafico contra los criterios de evaluación
Fuente propia

4.3.6 fase de Transición

La fase de transición del producto a los usuarios se caracteriza por una serie de pautas que se deben cumplir con una rigurosa atención, para que no se deforme la expectativa del nuevo producto de software, en este caso el nuevo desarrollo de software para puntos de venta. La transición requiere de manufactura, entrenamiento, soporte y mantenimiento del producto.

El cambio cultural del usuario a un nuevo producto de software debe concluir con el criterio de satisfacción completa del cliente.

Tabla 4.*Fase de transición.*

Modelo RUP
Realizar la evaluación de usuarios
Revisar y realizar los ajustes necesarios
Realizar un ajuste de gastos
Realizar la evaluación del producto por parte de los especialistas en ventas
Completar si es necesario el desarrollo de los componentes (prototipo funcional)
Probar los componentes contra los criterios de evaluación
Probar el diseño institucional, comunicacional y grafico contra los criterios de evaluación

Fuente propia.

4.4 UML Introducción

UML (*Unified Modeling Language*) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos. Para (Barahona Baquerizo & Calle Lema, 2018) “Los lenguajes de modelado son una abstracción de una entidad del mundo real que nos ayuda a entenderla y a comprenderla en todas sus características y funcionalidades”. El lenguaje de modelado UML está completamente asociado a los lenguajes orientados a objetos, y por su sencillez grafica puede detallarse para poder apreciar los elementos con diferentes enfoques, sin perder su naturalidad para comprender la funcionalidad.

James Rumbaugh (nacido en 1947) Doctor en ciencias de la computación, junto con Iván Jacobson y Grady Booch (“Los tres amigos”) desarrollaron el modelo UML (Lenguaje Unificado de Modelado), para (Ferre Grau & Sanchez Segura, 2008) “el objetivo de UML era posibilitar el

intercambio de modelos entre distintas herramientas”, definiendo una notación semántica que permitiera representar en un gráfico la sintaxis del lenguaje de modelado sin recurrir al código de los lenguajes de programación.

4.4.1 Notación básica de UML

La notación básica UML es la representación gráfica de los elementos que componen un modelo en especial orientado a objetos. En general describen las clases, dependencias y objetos y el detalle de asociaciones básicas.

4.4.2 Modelos

El modelo siempre va a representar un sistema de software con una perspectiva general y específica, la vista de los objetos da una idea de la funcionalidad y el flujo de los datos en diferentes fases. Cada modelo permitirá detallar un aspecto diferente del programa, es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, un algoritmo difícil de explicar en un lenguaje natural no técnico.

El estándar UML representa los siguientes tipos de diagramas:

- Diagrama de Estructura Estática.
- Diagrama de Casos de uso.
- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de estados.

4.4.3 Diagrama de Estructura Estática

Este tipo de diagramas incluidos en el estándar UML muestran en general el conjunto de clases y objetos que conforman la estructura del sistema de software en desarrollo. El concepto del diagrama debe enfocarse en las relaciones de objetos y clases.

4.4.4 Diagrama de Casos de uso

Es un tipo de diagrama que permite visualizar la relación entre los actores y los casos de uso del sistema, la diagramación del caso de uso busca representar la funcionalidad que desarrolla el programa en la interacción externa.

4.4.5 Elementos de un Caso de Uso

Los elementos que conforman un caso de uso son los siguientes: actores, casos de uso, relaciones entre casos de uso.

4.4.6 Actores

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el propio sistema, su representación en el caso de uso es una figura humana dibujada con palotes, los actores simbolizan una persona, o un elemento.

4.4.7 Casos de Usos

Los casos de uso son una descripción de una acción o actividad que se produce por la interacción de un actor y un sistema, son una representación en secuencia de una tarea específica, el caso de uso debe expresar una unidad coherente de funcionalidad, su representación se realiza en un plano gráfico y se le da el nombre de Caso de uso.

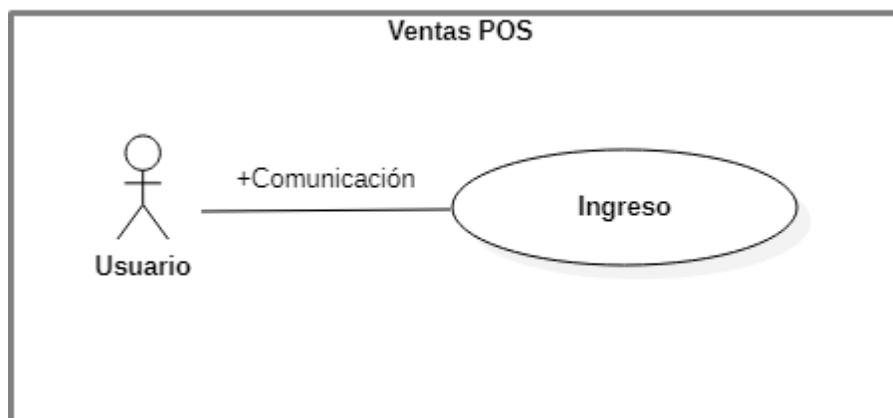
4.4.8 Relaciones entre Casos de Uso

Las relaciones son las interacciones que se generan entre dos o más casos de uso, se representan por medio de una línea que une los casos de uso, se definen cuatro tipos de relaciones:

➤ **Comunicación:** Relación asociación entre un actor y un caso de uso. Es un estereotipo básico.

Figura 3.

Relación Caso de Uso tipo Comunicación.

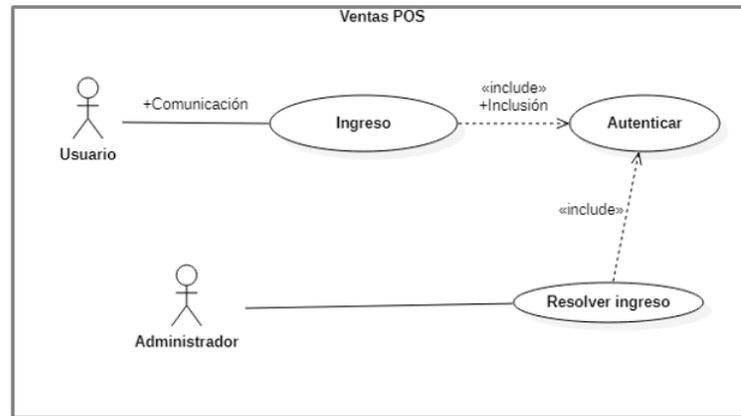


Fuente propia

➤ **Inclusión:** En este tipo de relación se incorpora explícitamente el comportamiento de otro caso de uso, el objetivo es enriquecer y compartir la funcionalidad común.

Figura 4.

Relación de inclusión caso de uso POS.

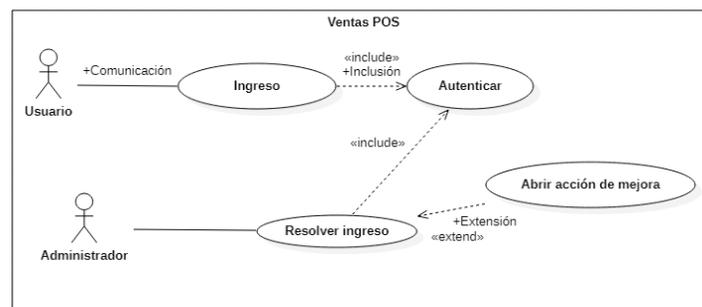


Fuente propia.

➤ Extensión: la relación de extensión sirve para modelar la parte opcional del sistema, es un subflujo que se ejecuta bajo ciertas condiciones. Incorpora implícitamente el comportamiento de otro caso de uso especificado indirectamente por este otro caso de uso. En el caso de uso base la extensión se crea por una serie de puntos concretos y previstos en el diseño (puntos de extensión).

Figura 5.

Relación de Extensión caso de uso POS.



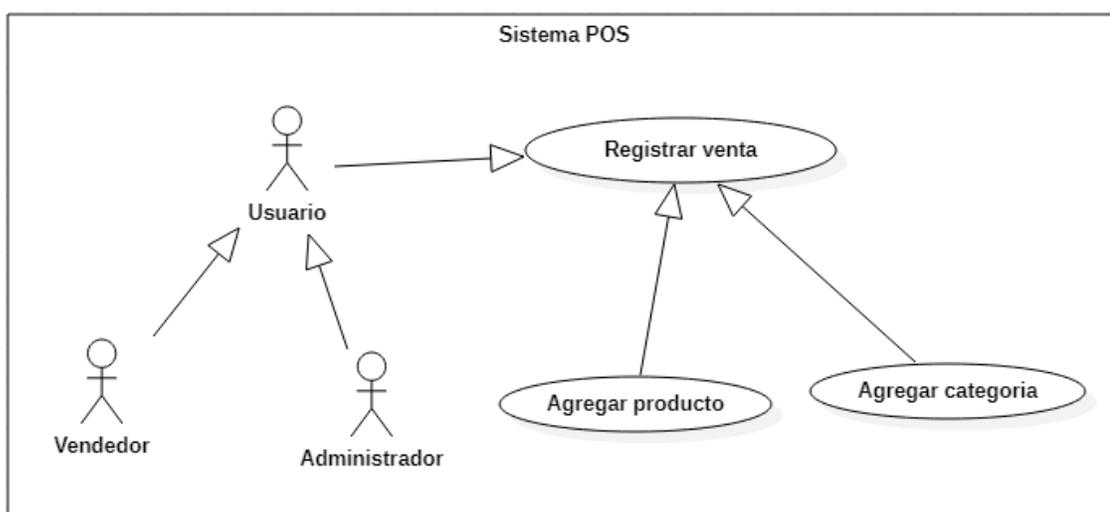
Fuente propia.

➤ Especialización y generalización de los casos

Es una relación que amplía la funcionalidad de un caso de uso, o mejora su funcionalidad original mediante el agregado de nuevas operaciones, atributos, y secuencias de acciones. En este tipo de caso de uso se heredan atributos que pueden ser utilizados en otros casos de uso.

Figura 6.

Relación de Generalización Caso de uso Sistema POS.



Fuente propia.

4.4.9 Conclusiones

La aplicación del Lenguaje Unificado de Modelado (UML) en el desarrollo de nuestro sistema POS, considerado como un estándar para describir sistemas de software ha permitido definir las diferentes relaciones entre los Casos de uso planteados, se ha obtenido una visión para capturar los requerimientos del sistema y guiar su proceso de desarrollo.

4.5 Introducción a las Bases de Datos

En la década de los años 70 del siglo XX, Edgar Frank Codd, definió el modelo Relacional en las bases de datos, el desarrollo consistía en el almacenamiento de datos en tablas compuestas por filas o “tuplas” y columnas o “campos” como una manera de organización y gestión de los datos, a la par que publicó una serie de reglas para los sistemas de datos relacionales, desde los inicios de su invento (BDR), la computación las ha utilizado convirtiéndolas en Sistemas de Gestión de Bases de Datos o SGBD.

“Hace 45 años, solo existían bases de datos en laboratorios de investigación especiales, donde los científicos computacionales buscaban modos de hacerlas eficientes y útiles”. (Oppel, 2010)

En la actualidad las bases de datos están disponibles para múltiples plataformas y ofrecen un eficiente servicio de almacenamiento de los datos organizados, evitando la duplicidad de la información, garantizando la integridad referencial de los registros, facilitando las búsquedas.

MySQL es un sistema gestor de bases de datos (SGBD) conocido ampliamente por la comunidad de informática, utilizado en soluciones donde se requiere un rendimiento alto de transaccionalidad, la simplicidad de implementación permite su aplicabilidad en la gran mayoría de plataformas (Windows, Linux, Unix).

MySQL fue desarrollado en sus inicios por Michael Widenius y David Axmark, en el año de 1995 fundan la empresa MySQL LAB y lanzan las primeras versiones de su producto, su principal característica desde sus fundadores es que sea rápida y de uso libre (free) licencias Community Edition, el producto continúa madurando y es aceptado por organizaciones que lo van implementando. En el año 2008 la empresa es adquirida por Sun Microsystem agregándole

funcionalidades de Gestor de Bases de Datos (SGDB) y en el año 2010 Oracle adquiere Sun Microsystems.

A partir de la versión 5.1 de MySQL se van añadiendo más características, según **(Suárez Cueto, 2010)**, en 1996 se lanza la versión 3.19 y en el año 2003 se producen avances importantes con la versión beta 4.0, en 2005 el gestor se acerca a un sistema completamente transaccional, incrementando su funcionalidad con procedimientos almacenados que aceleran las consultas a la base de datos.

4.5.1 ¿Qué es una Base de datos?

Una base de datos es un conjunto integrado de datos, almacenados de una manera organizada que puedan ser utilizados por los usuarios para facilitar sus búsquedas, visualizar la información, compartirla aplicando cláusulas para su posterior recuperación.

Según **(Acid Carrillo, Marin Ruiz, Medina Rodriguez, Pons Capote, & Vila Miranda, 2008)**, la definición de base de datos es: *“fondo común de información almacenada en una computadora para que cualquier persona o programa autorizada pueda acceder a ella, independientemente de su procedencia y del uso que haga de ella”*. El concepto anterior supone compartir el fondo común de información almacenada en un disco, servidor, o sitio, con un actor externo a la base de datos, con acceso restringido.

Sin embargo, buscando una definición más perfilada: “Conjunto de elementos de software con capacidad para definir, mantener y utilizar una base de datos”. **(Acid Carrillo, Marin Ruiz, Medina Rodriguez, Pons Capote, & Vila Miranda, 2008)** . La importancia del ultimo concepto es que introduce en un contexto de estructura de información junto con su contenido, dejando la parte de qué hacer con ese contenido al gestor de Bases de Datos (SGDB).

La Base de Datos debe contener un SGDB (Sistema de Gestión de Bases de datos), para el control de la información, aplicar cláusulas de seguridad para el control de acceso a la información y soportar las operaciones fundamentales al esquema lógico, como Crear, Leer, Modificar, Eliminar, (CRUD).

4.5.2. Gestor de Bases de Datos (SGBD)

Un SGBD está constituido por una serie de programas que permiten crear, alterar y eliminar bases de datos, el SGDB permite almacenar de una manera organizada la información, pero de igual manera recuperarla para darle un uso adecuado. La finalidad de implementar un SGDB es ofrecer un marco de trabajo con una serie de capas o niveles de abstracción, técnica que permite mediante lineamientos aislar elementos de la estructura de almacenamiento, la arquitectura ANSI/SPARC establece estos niveles, son los siguientes:

- Nivel conceptual: En general el nivel conceptual presenta la estructura de la base de datos de forma jerárquica, utiliza perfiles de usuario que de acuerdo con su rol puede visualizar la información, los usuarios restringidos tendrán una mínima vista en detalles del almacenamiento, el usuario programador tendrá una visión más detallada de la data.
- Nivel lógico: Es una representación de toda la información de la base de datos, es una vista lógica de los datos, como son realmente, a diferencia del nivel conceptual, el nivel lógico describe varios esquemas o vistas para el usuario.
- Nivel físico: Es una representación a bajo nivel de la base de datos completa, dispone como se almacenan los datos físicamente, este nivel incluirá los ficheros, las extensiones y las páginas.

4.5.3 Lenguajes de aplicación y lenguajes de datos

La arquitectura ANSI/SPARC recomienda que los SGBD deben incluir en sus herramientas de desarrollo un lenguaje específico enfocado en los datos, que brinde mecanismos necesarios para completar las tareas de control, definición, y manipulación de la información. MySQL implementa dentro de su arquitectura de desarrollo dichos lenguajes denominados: DSL (en inglés, Data Sub Language).

Lenguaje DDL (en inglés, Data Definition Language), es un lenguaje destinado a la definición de estructuras de datos y esquemas en la base de datos. El lenguaje DDL es usado para definir los esquemas físicos y lógicos de la base de datos, sin embargo, en arquitecturas DBMS se mantiene una separación entre los niveles físico y lógico y se utiliza un lenguaje alternativo denominado Lenguaje de Definición de Almacenamiento SDL (en inglés, Storage Definition Language) con el uso de este lenguaje se define el esquema conceptual.

Las sentencias DDL son aplicables para determinar las reglas de integridad (Constraints), muy importante para mantener integridad en la base de datos

El Lenguaje DML (en inglés, Data Manipulation Language), es un lenguaje que permite introducir datos en los esquemas de información de las bases de datos, de igual manera permite modificarlos, eliminarlos y consultarlos, permite también consultar la estructura de los esquemas definidos en la base de datos.

Finalizado el desarrollo de los esquemas de la base de datos, de igual manera se han cargado los datos iniciales, el lenguaje DML permite ejecutar las primeras instrucciones de manipulación de datos, los tipos de instrucciones utilizados en este esquema son: INSERT, UPDATE, DELETE, estos comandos permiten agregar, cambiar y eliminar filas de datos en las

tablas. En este punto del desarrollo de la base de datos esta lista para comenzar a recibir información.

Tabla 5.

Clasificación de los SGBD.

Rank Dic. 2020	SGBD más Populares	Base de Datos Modelo	Puntuación Diciembre 2020
1	Oracle	SGBD Relacional Multi-model	1325.60
2	MySQL	SGBD Relacional Multi-model	1255.45
3	Microsoft SQL Server	SGBD Relacional Multi-model	1038.09
4	PostgreSQL	SGBD Relacional Multi-model	547.57
5	MongoDB	SGBD Relacional Multi-model	457.53
6	IBM Db2	SGBD Relacional Multi-model	160.43
7	Redis	Key-valúe Multi-model	153.63
8	ElasticSearch	Search engine Multi-model	152.49
9	SQLite	Relacional	121.68
10	Cassandra	Wide Column	118.84

Fuente. (DB-Engines, 2020)

Según el sitio de información DB-Engines, encargado de recopilar y presentar información sobre SGDB, en su página oficial mensualmente publican las clasificaciones de los SGDB más populares y utilizados en internet, cumpliendo ciertos parámetros de los SGDB medidos de acuerdo a: consultas de motores de búsqueda, frecuencias de búsquedas de cada uno de los SGDB, discusiones técnicas y numero de ofertas de trabajo mencionando el conocimiento de los

sistemas SGDB, de acuerdo con los anteriores parámetros el sitio en la publicación de diciembre 2020, destaco a MySQL como uno de los SGDB relacionados en internet. Ubicándolo en la segunda posición por delante de SGBD como Microsoft SQL Server.

4.6 Introducción a PHP Lenguaje de Programación

PHP es un lenguaje de programación creado por Rasmus Lerdorf (1968, Groenlandia) en el año de 1995, desde los inicios de su creación PHP es considerado como software libre bajo licencia GNU. Su desarrollo se basa en aplicaciones web, para ejecutar desde un servidor apache, las implementaciones sobre PHP se caracterizan por ser ágiles y sencillas, puede utilizar formularios para registrar datos de usuarios, consultar productos en una base de datos, incluye todas las funcionalidades de un lenguaje de programación.

Las definiciones de las siglas de PHP son: en inglés Hypertext Pre-Processor, es un lenguaje de programación interpretado, se podría decir que su enfoque principal es desarrollar scripts que son interpretados por un servidor Web. **(Torres & Angel, 2014)**.

PHP tiene la facilidad de integrarse con otras plataformas de desarrollo como el Lenguaje de Marcado HTML (Hypertext Markup Language), lo cual le permite involucrar código en un bloque HTML, sin embargo, esta integración solo tiene una finalidad de diseño, PHP puede también integrar solo código nativo y lanzar desarrollos sin presentar problemas de ejecución de aplicación.

Para el desarrollo del sistema POS, PHP se integra con CSS (en inglés, Cascading Style Sheets) es un lenguaje creado para la composición y estructuración de páginas web, esta herramienta contiene elementos de codificación que permiten marcar la separación del contenido y la forma de presentar estilos visuales que mejoran la calidad de los desarrollos.

El lenguaje de programación PHP se integra muy bien con MySQL y sus bases de datos, para poder interactuar con dichas bases de datos utiliza una API (Interfaz de Programación de Aplicaciones), utiliza también un Conector, es una pieza de software que permite la conexión con el servidor MySQL, y por último utiliza un driver o Controlador, el cual llama una biblioteca cliente.

4.6.1. Interfaces API para el desarrollo POS

Un API en PHP, define las clases, métodos, funciones y variables que un desarrollo necesita llamar para ejecutar una tarea. En el caso del software POS, las API necesarias responden a un modelo MVC (Modelo Vista Controlador) lo anterior permite organizar el código de una forma ordenada, instanciando las clases y sentencias SQL en una capa de abstracción.

Las API en PHP pueden ser procedimentales u orientadas a objetos, con una API procedimental como la presentada en el modelo MVC, se instancian las clases y luego se llama a métodos de los objetos resultantes. Lo anterior produce un nivel de seguridad que no es visible desde la página de vista al usuario.

4.6.2 Conectores en PHP para el desarrollo POS

El termino Conector no es nativo de PHP, este término es generalizado en la documentación de bases de datos MySQL, define un mecanismo de comunicación con el servidor de base de datos MySQL, proporciona conectores para muchos lenguajes, incluido PHP. La aplicación de punto de venta POS necesita conectarse con la base de datos, se utilizará la API la cual contiene el modelo de conexión incluyendo las sentencias de consultas, el conector contendrá los parámetros de conexión que abren y cierran la conexión con la base de datos.

4.6.3. Controladores en PHP para el desarrollo POS

Es una pieza de software diseñada para la comunicación con un tipo específico de servidor de bases de datos. El controlador también llama a una Biblioteca, como la Biblioteca cliente de MySQL o el controlador nativo de MySQL. Estas bibliotecas implementan el protocolo de bajo nivel usado para comunicarse con el servidor de bases de datos de MySQL.

Uno de los controladores disponibles de MySQL es el controlador PDO, que permite hacer de interfaz con el servidor MySQL. Para el desarrollo del sistema POS, se empleará el controlador PDO que define una interfaz sencilla.

4.6.4. Bootstrap (Framework)

Bootstrap es una biblioteca multiplataforma, o kit de herramientas de código abierto utilizado en desarrollos web, Bootstrap permite adaptar la apariencia de las páginas web al dispositivo desde donde se está cargando la aplicación. El Framework hace uso de librerías CSS y JavaScript, incluye componentes como ventanas modales que facilitan las acciones de ingresar datos en formularios de usuarios, clientes, productos y ventas.

Bootstrap, en sus comienzos se conoció con el nombre de Blueprint utilizado en Twitter, desarrollado por Mark Otto y Jacob Thornton, desde su creación se pensó como un marco de trabajo (framework).

Bootstrap proporciona al desarrollo POS un conjunto de hojas de estilo para todos los componentes HTML, características como uniformidad al navegador, sistema de anchura, apariencia moderna para el formateo de elementos de texto, tablas y formularios.

4.6.5. CSS (Hojas de estilo en cascada)

Es un lenguaje utilizado en el diseño web para definir la presentación (aspecto visual) de un desarrollo web, CSS está enfocado en separar la información (contenido) de su representación visual, de modo que cambiar el diseño no implique modificar el contenido.

Según **(Manz, 2020)** “CSS es una tecnología utilizada para dotar de cualidades visuales y estéticas a una página web”. Los desarrollos web no se deben quedar en la sola presentación de contenido, si bien es cierto publicar información es una necesidad, publicarla de una manera agradable para el lector o el usuario, es una forma ética que los contenidos publicados tengan una apariencia ordenada y atractiva.

La estructura de CSS se basa en reglas con formato que determinan la apariencia del desarrollo web:

- **Selector:** Es el elemento HTML que se referencia del documento para aplicarle un estilo concreto (body, header, content).
- **Propiedad:** Son las características que ofrece el lenguaje CSS, color de texto, capacidad, propiedad de las líneas, etc.
- **Valor:** Las propiedades de los elementos en CSS tienen un valor que puede variar según el tipo de diseño planteado, las unidades de medida son:

Tabla 6.*Unidades de medida utilizadas en CSS.*

Longitudes utilizadas	Descripción
px	Píxeles (relativo al dispositivo)
em	Relativo al tamaño de la fuente del elemento (2em = 2 veces tamaño)
%	Porcentaje, relativo al elemento padre
in	Pulgadas (1 pulgada = 2.54 cm)
cm	Centímetros
Mm	Milímetros
pt	Puntos (1 punto = 1/72 pulgadas)

Fuente propia.

4.6.6 Introducción a HTML

El lenguaje HTML es un lenguaje estándar reconocido por todos los programadores y no programadores incluyendo la W3C (World Wide Web) consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento uso y desarrollo de internet. El objetivo del W3C es organizar las especificaciones técnicas y establecer directrices para el desarrollo de tecnologías web.

El significado de las siglas HTML como todos los productos y lenguajes de Informática establecen en si su funcionalidad, (en inglés, Hypertext Markup) Lenguaje de Marcas de Hipertexto. El origen de HTML se remonta a 1980, el físico Tim Berners-Lee propuso he invento un nuevo sistema de hipertexto para compartir documentos.

Importante referenciar que los sistemas de hipertexto ya habían sido creados años antes de 1980, sin embargo, la contribución de Tim Berners-Lee y Robert Cailliau motivo a unir sus esfuerzos y presentar un hipertexto hecho específico para el modelo de internet, llamada World Wide Web (W3).

4.6.7. Protocolo HTTP

Para que la comunicación sea posible, entre el navegador y el servidor donde se aloja la aplicación HTML (desarrollo POS) dicho de una manera coloquial “deben hablar el mismo idioma”, el protocolo HTTP se utiliza para que haya comunicación entre el ordenador (cliente) y el servidor. La comunicación se da en ambos sentidos, envió de comandos y respuestas en texto ASCII, es la base de cualquier intercambio de datos en la web.

Diseñado a comienzos de la década de 1990, HTTP es un protocolo ampliable que ha ido evolucionando con el tiempo, funciona sobre la capa de aplicación del modelo de red OSI, la transmisión se da sobre el estándar TCP, o también se puede implementar sobre el protocolo encriptado TLS muy utilizado en la actualidad, TLS ofrece seguridad en la transmisión de tramas o paquetes ya que utiliza un cifrado especial que impide sean vistos los datos en flujo.

4.6.8 HTML 5

Es la actual versión liberada para el estándar HTML, su lanzamiento se concretó en el año 2014, HTML 5 es conocida como la evolución de la web, por sus características que incorpora en su estructura, es importante referenciar que sitios tan importantes como wix.com, WordPress, LinkedIn ya han implementado desde hace varios años la versión HTML 5, y que la actual versión ha madurado en su funcionalidad.

Para el actual desarrollo del sistema POS, las características integradas de HTML 5 son necesarias, la combinación de CSS3 y PHP, junto con JQuery Data Tables permiten vincular tablas de uso para publicar los datos, lo cual mejora bastante el desempeño de las consultas. Para la declaración de los scripts se agrega una nueva etiqueta “script”, con lo cual se le indica al sistema que se va a cargar un script.

4.7 Marco Teórico

El marco teórico que orienta este desarrollo de sistemas POS, se basa en los conceptos fundamentales que definen un sistema: “la interacción, la globalidad, la organización y la complejidad” (Florez & Thomas, 1993). En concordancia con la Teoría General de Sistemas los conceptos expuestos con anterioridad actúan en conjunto para cumplir un propósito.

La interacción de un sistema de software se desarrolla en el principio de relación, retomando el lenguaje UML el Actor (Individuo, objeto) y el sistema (software) realizan un proceso con un propósito. El concepto de globalidad se cumple en dos sentidos (direcciones) el primero en la relación de cada uno de los conceptos (interacción, globalidad, organización, complejidad), el segundo sentido aplica en el servicio o función para el que fue dispuesto: “gestión de procesos de pago en un punto de venta”.

El principio de organización plantea (Florez & Thomas, 1993) “*hace referencia a la organización interna de los elementos, unión de individuos o elementos*”, la combinación de las funciones de todos los elementos se interpreta como un sistema, y un claro ejemplo de un sistema es una empresa en su totalidad, con todos sus departamentos, las normas que las rigen, la interacción entre todos los departamentos para cumplir las metas y los propósitos son el resultado de la organización interna.

La complejidad, este concepto no se puede confundir con el término “complicación”, para (Florez & Thomas, 1993) hace referencia al número de interacciones y las características de sus elementos. Enfocando el desarrollo POS en el principio de complejidad y las interacciones que un sistema de software requiere para su completa funcionalidad. La construcción de software no se limita al desarrollo del código, un sistema POS se construye bajo las normas que regulan la entrega de una factura al comprador, a llevar el control de un sistema de facturación, a generar reportes al terminar el tiempo de apertura del establecimiento comercial, aplicar los impuestos de acuerdo con las normas que rigen la venta y comercialización de artículos de ropa.

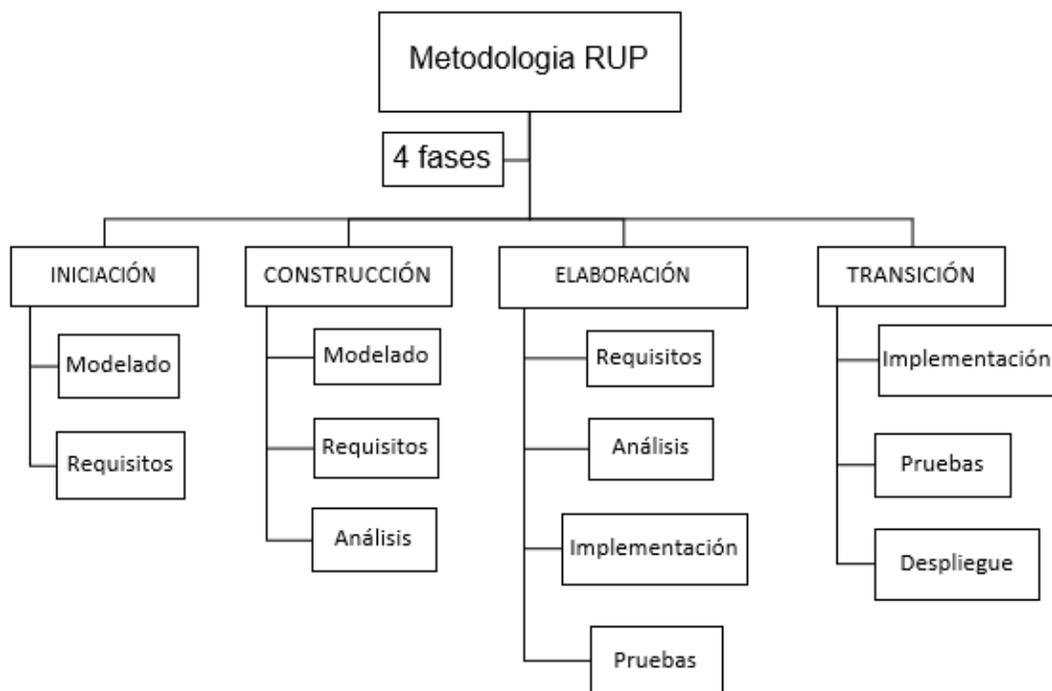
Diseño Metodológico

Para el desarrollo de este proyecto se aplica la metodología RUP, esta metodología representa una guía completa que permitirá definir un conjunto de actividades indispensables para convertir los requerimientos funcionales (que debe hacer el sistema) según el criterio de los usuarios, en un conjunto de elementos necesarios para la construcción de productos orientados a la Web.

El desarrollo del diseño POS se estructura en la recogida de los requerimientos funcionales, posterior a ello el modelo en cascada RUP permite generar los casos de uso utilizando la técnica UML, la diagramación de los casos describe un servicio que el usuario requiere del sistema e incluye la secuencia completa de interacciones entre el usuario y el sistema.

Figura 7.

Diagrama esquema metodología RUP.



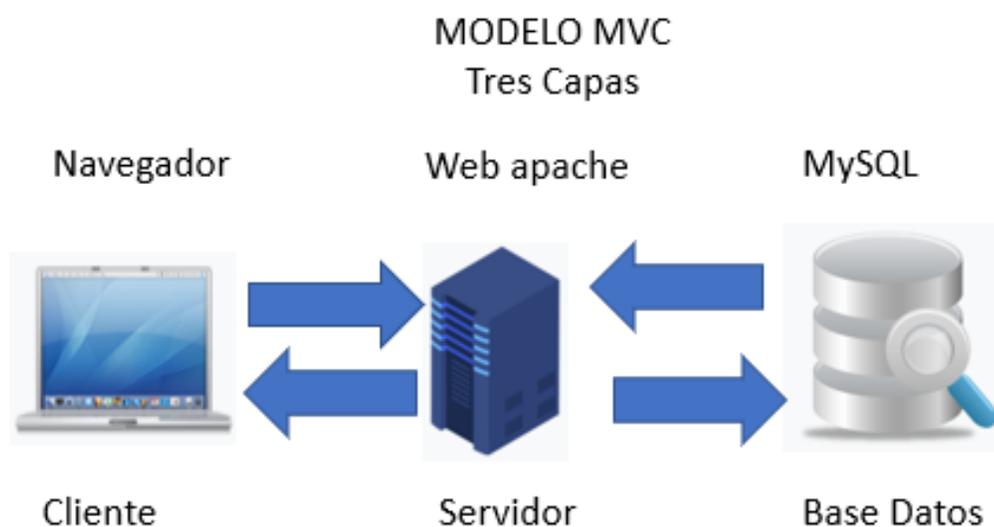
Fuente propia.

La metodología RUP sobre la cual está basado este proyecto, presenta cuatro (4) fases de desarrollo durante todo el ciclo de Vida, enfocado en la asignación de tareas y responsabilidades a cada uno de los actores del proyecto.

Según las necesidades del negocio y los requerimientos funcionales de la organización Broxxo Jeans, el almacenamiento y la manipulación de la información en el nuevo sistema POS se gestionará con motor MySQL complementado con PHP, en un ambiente cliente servidor de tres capas:

Figura 8.

Diagrama de tres capas.



Fuente propia.

5. Fase 1, etapa de Iniciación

El éxito de un desarrollo web se logra al aplicar correctamente la recogida de los requerimientos del usuario, darle importancia a la ingeniería de los requerimientos (IR) es una de las buenas prácticas que se ha sumado para el desarrollo de este Proyecto Web sistema POS para la organización Broxxo Jeans.

Al concluir la tarea de escuchar al usuario y poner atención a sus exigencias se concluyen los siguientes requerimientos, consolidados en las siguientes tablas al integrar las inquietudes del cliente con la experiencia del desarrollador, aplicando de igual manera la metodología RUP del desarrollo conforme a lo establecido en los objetivos del Proyecto.

Para (Oliveros, Danyans, & Mastropieto, 2014) las aplicaciones Web requieren nuevos enfoques de la IR que puedan dar una respuesta capaz de ajustarse a una comunidad cambiante, grande y abierta con amplios requerimientos de navegabilidad. (Oliveros, Danyans, & Mastropieto, 2014) Para la construcción del sistema, se determinan 17 requerimientos funcionales para las vistas e interfaces de la fase inicial del sistema POS.

Tabla 7.

Definición de requerimientos funcionales.

REQUERIMIENTOS	DESCRIPCION DEL REQUERIMIENTO
RF1	Construcción aplicación POS contara con un diseño web
RF2	Diseño de interfaz de inicio de sesión
RF3	Diseño de interfaz General
RF4	Diseño de la interfaz de gestión de usuarios
RF5	Diseño de la interfaz de gestión de categorías de productos
RF6	Diseño de la interfaz de gestión de productos
RF7	Diseño de la interfaz de gestión de clientes
RF8	Diseño de la interfaz de gestión de ventas
RF9	Diseño del Modal agregar usuario
RF10	Diseño del Modal agregar producto

RF11	Diseño del Modal agregar cliente
RF12	Diseño del Modal administrar venta
RF13	Diseño del Modal crear venta
RF14	Diseño del Modal reporte de ventas
RF15	Diseño del Menú desplegable
RF16	Diseño del Menú salida
RF17	Diseño del Menú estado usuario

Fuente propia.

Para el completo alojamiento de programas y aplicación se determinan los siguientes requerimientos a nivel de hardware, en el sistema POS.

Tabla 8.

Requerimientos Hardware.

REQUERIMIENTOS	DESCRIPCION DEL REQUERIMIENTO
RFH1	Servidor Dell 2900 alojar aplicación
RFH2	Portátil Lenovo V310 8 GB RAM, Disco 500 GB, Desa 1
RFH3	Portátil Lenovo V310 8 GB RAM, Disco 500 GB, Desa 2
RFH4	PC Escritorio Lenovo punto de venta 1 pruebas
RFH5	PC Escritorio Lenovo punto de venta 2 pruebas
RFH6	Switch de comunicaciones 100 MB 1 GB
RFH7	Impresora POS

Fuente propia.

Complementario a los Requerimientos funcionales para el desarrollo POS, las herramientas que permiten el diseño la implementación el desarrollo y el despliegue del producto de software son los programas y aplicaciones que permiten la funcionalidad del requerimiento y constituyen la base para la construcción de cada componente de software.

Sin estas herramientas no se puede generar el servicio o el consumo del servicio, a continuación, se describen dichas herramientas.

Tabla 9.

Requerimientos funcionales & herramientas.

REQUERIMIENTOS	DESCRIPCION DEL REQUERIMIENTO
RFD1	XAMPP para Windows 7.3.9 64 bit
RFD2	Sublime Text 3 para Windows
RFD3	Apache versión 7 para Windows
RFD4	Plugin de Java
RFD5	Plugin Bootstrap
RFD6	Plantilla Admin. LTE Software libre
RFD7	Data Table (CSS Java Script)
RFD8	StarUML aplicación de Desarrollo casos de Uso

Fuente propia.

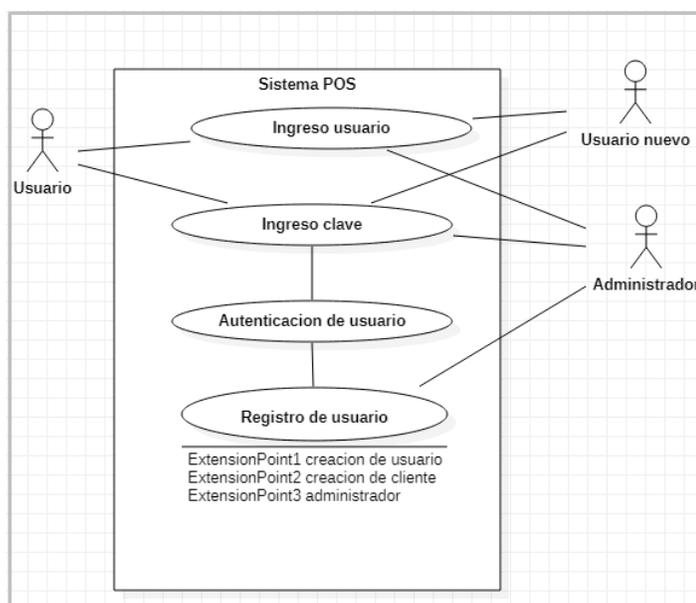
En la anterior tabla se describen los requerimientos funcionales enfocados al uso de las herramientas que integran el paquete de desarrollo, las aplicaciones corren bajo un ambiente Windows 10, el uso de estas herramientas se establece bajo las políticas de software libre, licencias GNU GPL.

5.1 Implementación UML casos de Uso Sistema POS

El RUP es un proceso basado en los modelos en cascada y por componentes, facilita el uso de casos de uso en el desarrollo de aplicaciones, esta herramienta permite realizar el paso a paso del comportamiento de la comunicación mediante la interacción del usuario y la aplicación POS.

Figura 9.

Caso de uso 1 Ingreso al sistema POS.



Fuente propia.

Flujo principal caso de uso 1.

Paso 1 – Usuario: Digita el nombre de usuario.

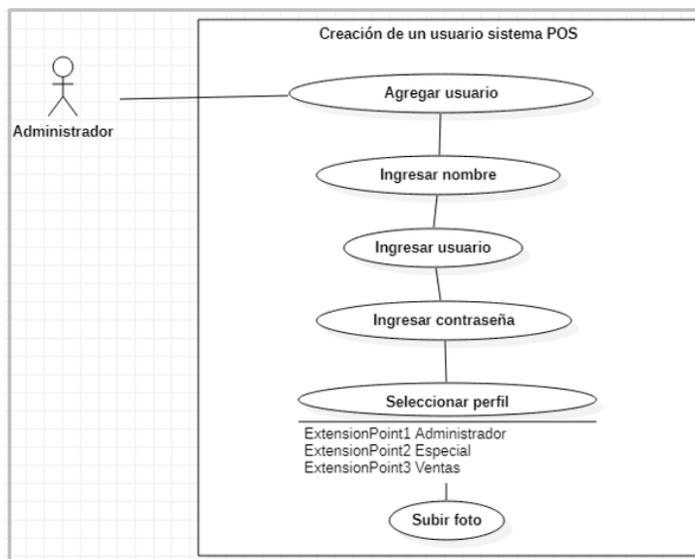
Paso 2 – Sistema: Ingrese la clave de usuario.

Paso 3 – Usuario: Ingresa la clave de usuario.

Paso 4 – Sistema: Realiza autenticación, permite el ingreso.

Figura 10.

Caso de uso 2, creación de una cuenta de usuario.



Fuente propia.

Flujo principal caso de uso 2.

Paso 1 – Usuario: clic derecho botón Agregar usuario.

Paso 2 – Sistema: abre el modal agregar usuario.

Paso 3 – Usuario: ingresa nombre completo.

Paso 4 – Usuario: ingresa nombre de usuario.

Paso 5 – Usuario: ingresa contraseña.

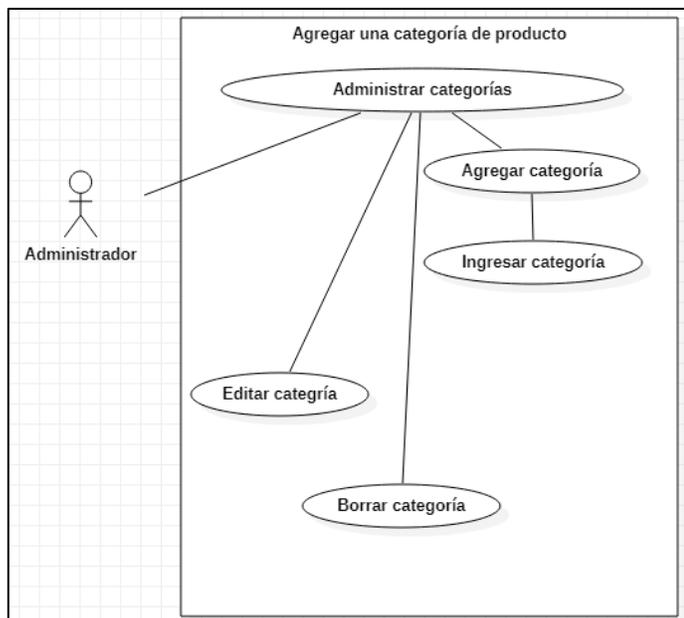
Paso 6 – Usuario: selecciona el perfil del usuario.

Paso 7 – Usuario: carga la foto del usuario.

Paso 8 – Usuario: clic sobre el botón guardar usuario.

Figura 11.

Caso de uso 3, agregar una categoría al sistema POS.



Fuente propia.

Flujo principal caso de uso 3, agregar una categoría de producto desde el Administrador de categorías.

Paso 1 – Usuario: Desde el gestor de categorías clic botón agregar categoría.

Paso 2 – Sistema: despliega un modal, ingresar nombre de la categoría.

Paso 3 – Usuario: Ingresa el nombre de la categoría, botón guardar categoría.

Paso 4 – Usuario: Editar una categoría, clic en el icono lápiz (acciones).

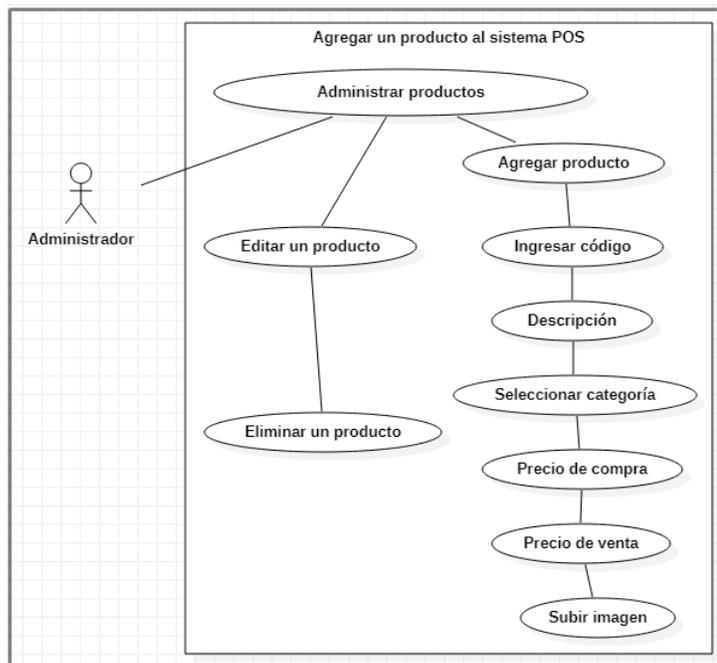
Paso 5 – Sistema: abre un modal para realizar cambio en la categoría.

Paso 6 – Usuario: realiza los cambios en la categoría (nombre, tipo).

Paso 7 – Usuario: clic en el icono x para borrar una categoría.

Figura 12.

Caso de uso 4, agregar un producto al sistema POS.



Fuente propia.

Flujo principal del caso de uso 4, agregar un producto al sistema POS.

Paso 1 – Usuario: Interfaz de gestión de productos, clic agregar producto.

Paso 2 – Sistema: abre el modal, agregar producto.

Paso 3 – Usuario: ingresar el código de producto.

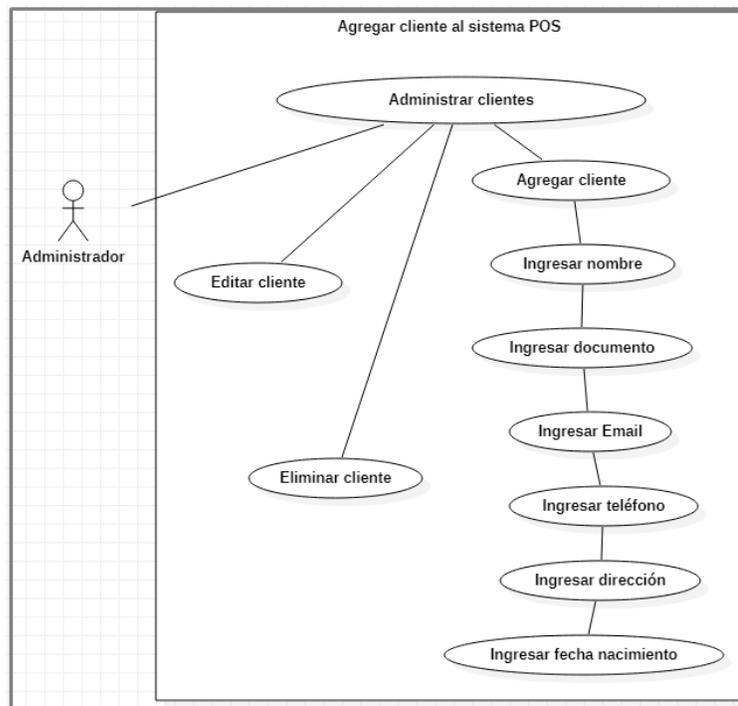
Paso 4 – Usuario: ingresar descripción, seleccionar categoría.

Paso 5 – Usuario: ingresar precio de venta.

Paso 6 – Usuario: guarda los cambios.

Figura 13.

Caso de uso 5, agregar un cliente al sistema POS.



Fuente propia.

Flujo principal del caso de uso 5, agregar un cliente al sistema POS.

Paso 1 – Usuario: Desde interfaz de clientes clic agregar cliente.

Paso 2 – Sistema: carga interfaz modal agregar cliente.

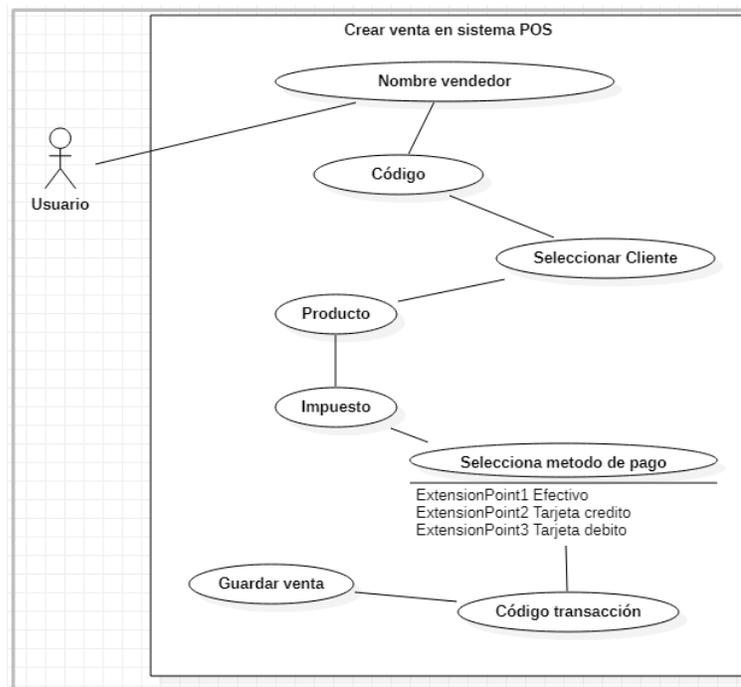
Paso 3 – Usuario: Ingresar datos de cliente: nombre, documento, email.

Paso 4 – Usuario: Ingresar teléfono, dirección, fecha de nacimiento.

Paso 5 – Usuario: Guardar datos de cliente.

Figura 14.

Caso de uso 6, agregar venta.



Fuente propia.

Flujo principal del caso de uso 6, crear venta.

Paso 1 – Usuario: ingresa nombre de usuario.

Paso 2 – Usuario: ingresa código de vendedor.

Paso 3 – Usuario: ingresa descripción de producto, precio, impuesto.

Paso 4 – Usuario: selecciona método de pago.

Paso 5 – Usuario: ingresa código de transacción.

Paso 6 – Usuario: Guardar venta.

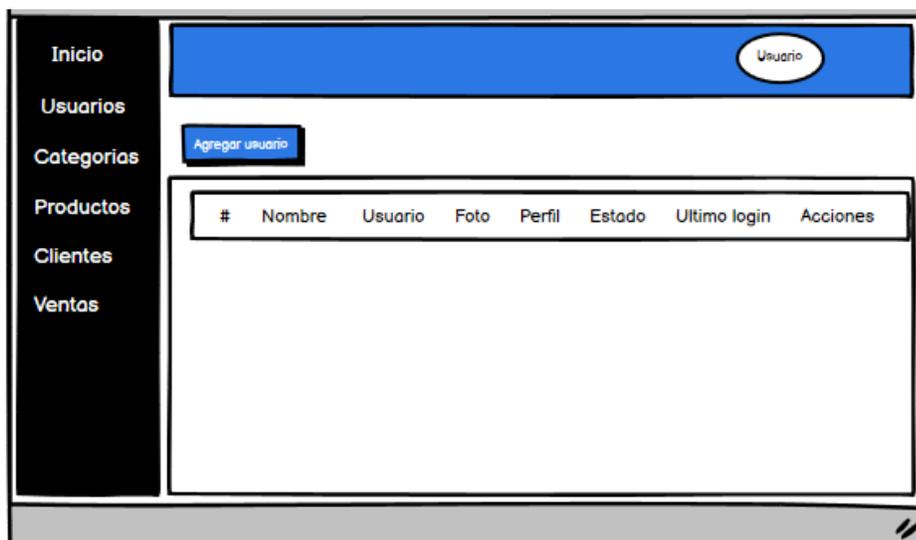
5.2 Fase 2 Maquetación del diseño POS

El diseño de las interfaces del proyecto está basado en una plantilla que permite la gestión de contenidos para un sistema informático, es una plantilla gratuita, fácil de adaptar en la construcción de interfaces web, sus características de integración con HTML5, CSS 3, BOOTSTRAP 3 y múltiples plugin, hacen que el desarrollo sea más ágil y productivo.

La plantilla base es Admin LTE 2.4.2, apropiada para adaptar en desarrollos web, incluye menús laterales desplegable, ocultos para aprovechar al máximo el espacio de los datos de vista al usuario y los productos.

Figura 15.

Boceto interfaz de usuarios.



Fuente propia.

La anterior imagen muestra el diseño de la página de inicio en el sistema POS, el diseño consta de un menú y un panel principal lateral, aplicando el diseño responsivo para acceder desde

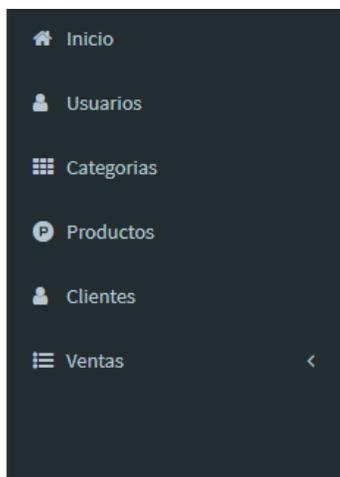
dispositivos de resolución pequeña como un móvil o tabletas, el diseño responde a 320, 768, 1024 pixeles.

El marco se basa en CSS Bootstrap 3, personalizado para el uso en computadores de escritorio, puntos de venta, el código de la maquetación aplicando HTML 5, Plugin CSS y JavaScript.

Los menús laterales del sistema POS, se integran a partir de una técnica de ensamblado de código que permite ocultar y mostrar contenido, con esta característica el menú lateral se plegara para abrir todo el espacio de datos de la interfaz, o por el contrario se desplegara para navegar en los módulos de la aplicación. Las vistas de los módulos de la aplicación de inicio, usuario, categorías, productos, clientes, ventas, reportes, crear-venta, llamaran la página correspondiente al nombre, desde el navegador se identificará el módulo en el cual se está navegando.

Figura 16.

Diseño del panel lateral.



Fuente propia.

5.2.1 Maquetación de los módulos de gestión del sistema POS

Cada módulo de gestión del sistema POS, corresponde a un sistema interno de gestión de la aplicación POS, la vista de cada módulo parte de una plantilla estándar creada en HTML5 combinando con etiquetas PHP, la mayor parte del código esta generado en PHP con la herramienta sublime Text 3.

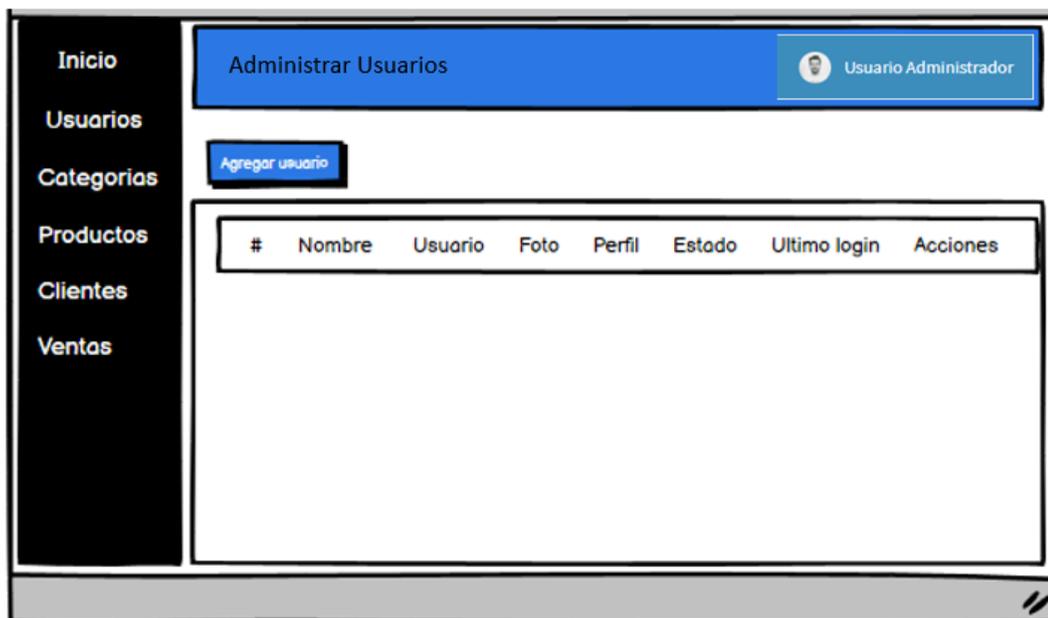
El contenido de cada módulo se publica por la acción de la herramienta Data Table, un Plugin que permite desplegar tablas de una manera dinámica sin afectar los datos de la base de datos, tampoco compromete la seguridad la información. Las funcionalidades que ofrece el plugin Data Table son: Define criterios de búsquedas, paginación de la información, datos organizados, se complementa con Ajax para función asíncrona.

5.2.2 Modulo de usuarios

El diseño de las interfaces, del módulo de usuarios, responden a la necesidad básica de gestión de usuarios, parten desde la autenticación del usuario (Registro y login), hasta la creación de un nuevo usuario en la cual se asigna un perfil de gestión del usuario. La aplicación web permite generar tres tipos de perfiles de usuarios (Administrador, Especial o Vendedor).

Figura 17.

Modulo Gestión de usuarios.

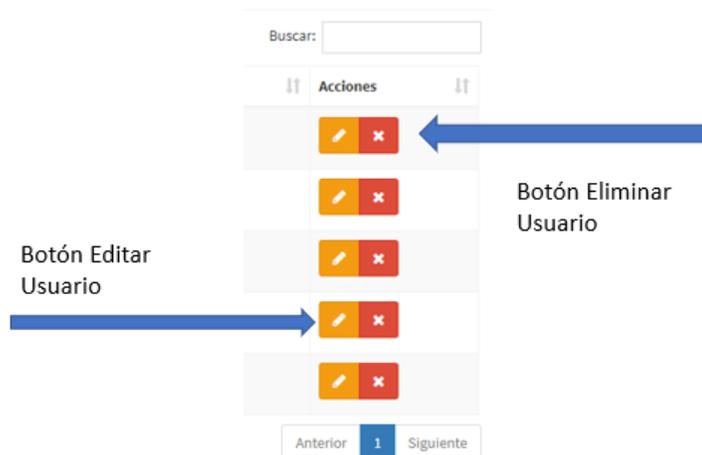


Fuente propia.

El gestor de usuarios está construido en una estructura de código PHP, el diseño incluye un botón “Agregar usuario” con la función de cargar una ventana modal en forma de formulario con los respectivos campos para agregar un nuevo perfil de usuario. El formulario toma la información de usuario de una base de datos en MySQL. La interfaz permite visualizar en forma de tabla los datos de cada usuario, los datos corresponden a: Nombre, Usuario, Foto, Perfil, Estado, Ultimo login, Acciones. Las Acciones permiten la edición de cada usuario y realizar cambios según los requerimientos, también incluye la función de eliminar al usuario.

Figura 18.

Boceto del botón Acciones.



Fuente propia.

La ventana modal se carga con la herramienta Bootstrap 3, desde la plantilla de la aplicación se carga el plugin de Bootstrap, el contenido del código es adaptable a toda clase de soluciones web.

5.2.3 Maquetación del Módulo de venta del sistema POS

El módulo de ventas presenta un diseño dividido en tres secciones, cada sección cumple una función diferente, sin embargo, se integran para poder completar el ciclo de sistema de venta, inventario y reportes. El módulo de Administrar venta gestiona cada una de las ventas realizadas en el punto de pago del establecimiento. El módulo Crear venta es la encargada de ejecutar la venta directa, agrega los productos, los cuenta, y realiza los cálculos y envía la información al módulo de Reporte de ventas y al módulo Administrar venta para su correspondiente impresión y entrega de factura al cliente.

5.2.4 Módulo Administrar venta

El módulo Administrar venta es una interface que cumple el propósito de gestionar las ventas realizadas desde el punto de pago, este módulo recibe los datos organizados de la venta realizada, los captura en forma de lista y los prepara para su correspondiente gestión, el usuario puede visualizar los datos de la venta realizada de una manera generalizada, los datos capturados evidencian código de factura, cliente, vendedor, total de la venta, fecha en la cual se generó la venta y la forma de pago.

Figura 19.

Boceto del módulo Administrar venta del Sistema POS.

#	Código factura	Cliente	Vendedor	Forma de pago	Neto	Total	Fecha	Acciones
1	10001	Ismael Romero	Usuario Administrador	Efectivo	\$ 6,300.00	\$ 7,497.00	2020-12-30 16:18:05	  

Fuente propia.

La columna Acciones del módulo de Administrar venta cumple tres funciones importantes, el icono en forma de impresora ejecuta la tarea de imprimir la factura; el icono en forma de lápiz

ejecuta la tarea de editar la factura en el caso de requerir algún ajuste, y la opción de eliminar la factura en caso de solicitar su eliminación.

5.2.5 Módulo Crear venta.

El módulo crear venta es el encargado de ejecutar la venta, realiza los cálculos asociados a la venta, cuenta los artículos, genera la sumatoria y totaliza el valor de la venta, guarda la venta y la envía al módulo Administrar venta para su correspondiente impresión. La interfaz se divide en dos secciones, una sección corresponde a los parámetros que integran una venta, la otra sección (derecha) visualiza el listado de productos (Inventario) debidamente registrado con la opción del botón “Agregar” que incluirá los productos que serán referidos en la venta.

Figura 20.

Boceto del módulo Crear venta.

#	Imagen	Código	Descripcion	Stock	Acciones
1		101	Jean tiro alto	20	Agregar
2		102	Jeggins ajustado	16	Agregar
3		103	Jogger semi ajustado	15	Agregar
4		104	Jeans ajustados	19	Agregar
5		105	Denim stretch	18	Agregar
6		106	Overol semi ajustado	20	Agregar
7		107	Leggins negro	20	Agregar
8		108	Leggins azul	20	Agregar
9		109	Leggins rojo	20	Agregar
10		110	Leggins ajustado poliester	20	Agregar

Fuente propia.

5.2.6 Módulo de Reportes

El módulo de reportes de venta es el encargado de construir los informes de los productos vendidos en el sistema POS, cumple la función también de imprimir las facturas generadas en cada una de las ventas realizadas en el punto de venta, su diseño presenta una interfaz gráfica con estadísticas generales de las ventas. El usuario puede consultar fácilmente los productos más vendidos visualizando un reporte gráfico general de ventas.

El diseño de la interfaz presenta cuatro secciones de información: Grafico de ventas, Productos más vendidos, Información de vendedores, Información de compradores.

Figura 21.

Boceto del módulo de Reportes.



Fuente propia.

La impresión de la factura se realiza desde la interfaz de Administrar venta, en la columna de acciones, la factura se genera en formato PDF, al pulsar clic sobre el icono de impresora carga

una página con la información de la venta. El módulo TCPDF se integra completamente con PHP el cual permite crear archivos PDF de una forma dinámica, TCPDF tiene la capacidad de interpretar código XHTML sobre desarrollos web.

TCPDF se basa en la clase FPDF, es un proyecto muy utilizado en los sistemas que requieren agilidad para generar los archivos PDF, las últimas versiones de este producto se han construido para soportar cargas altas de generación de archivos. TCPDF soporta escritura en HTML sobre múltiples columnas. Importante resaltar del producto que permite crear los diseños de la factura en secciones o módulos.

5.2.7 Módulo de Inventarios

El módulo de inventarios cumple la función de garantizar y mantener la información actualizada de cada uno de los productos de acuerdo con sus categorías, debe informar también sobre el stock de productos en el punto de venta. Su funcionalidad le indica al usuario si hay existencia de productos para la venta, de no existir en el listado publicado sobre la columna Stock de la tabla impresa en un recuadro de color rojo con el indicador 0, indicara cero productos para la venta.

El módulo contiene un botón con la función “Agregar producto”, el botón carga una ventana modal en forma de formulario que permite ingresar un producto nuevo, la información se almacena en la tabla productos, el diseño de la interfaz carga una matriz que permite desplegar la información detallada de cada uno de los productos. Desde la columna “Acciones” se pueden editar los productos, también permite remover el producto del listado sin afectar las ventas anteriores del producto, solo afectaría el stock del inventario.

Figura 22.

Boceto módulo de inventarios sistema POS.

#	Imagen	Código	Descripción	Categoría	Stock	Precio de compra	Precio de venta	Agregado	Acciones
60		101	Jean tiro alto	PANTALÓN PARA DAMA	OK	1000	1200	2021-01-08 23:29:32	
59		102	Jejginis ajustado	PANTALÓN PARA DAMA	Baja	4500	6300	2021-01-08 23:13:21	
58		103	Jejginis semi ajustado	PANTALÓN PARA DAMA	OK	3000	4200	2021-01-08 23:25:48	
57		104	Jeans ajustados	PANTALÓN PARA DAMA	OK	4000	5600	2021-01-08 23:25:48	
56		105	Denim stretch	PANTALÓN PARA DAMA	OK	1540	2156	2021-01-09 01:07:54	
55		106	Overol semi ajustado	PANTALÓN PARA DAMA	Baja	1100	1540	2021-01-08 20:53:38	
54		107	Jeans Trece	PANTALÓN PARA DAMA	OK	1540	2156	2021-01-09 01:07:54	
53		108	Jeans Julia	PANTALÓN PARA DAMA	OK	1540	2156	2021-01-08 23:29:32	
52		109	Jeans Tacio	PANTALÓN PARA DAMA	OK	2600	3640	2021-01-09 01:07:54	
51		110	Jeans Roser	PANTALÓN PARA DAMA	OK	2210	3094	2021-01-08 23:29:32	

Fuente propia.

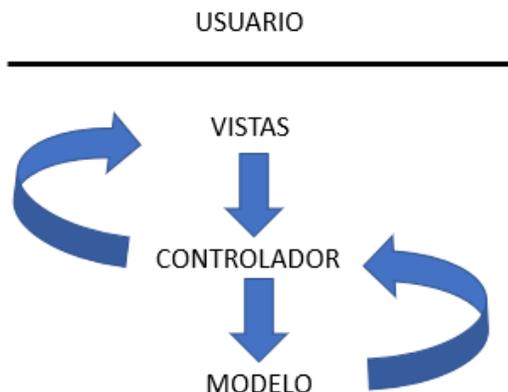
5.3 Construcción del Modelo Vista Controlador

El patrón MVC (Modelo, Vista, Controlador) es un tipo de diseño que separa en capas bien definidas el desarrollo de una aplicación, esas partes son tres:

- El Modelo: Encargado de la lógica de la aplicación y la conexión con la base de datos.
- El Controlador: Es el encargado de gestionar las peticiones del usuario, procesando, invocando al modelo y mostrando al usuario a través de las vistas.
- Vistas: Son las responsables de mostrar al usuario el resultado que obtiene del modelo a través del controlador.

Figura 23.

Modelo Vista Controlador.



Fuente propia.

En el modelo MVC, el usuario siempre se va a encontrar con las vistas (interfaces), en el momento que hay una solicitud para traer información de la base de datos, esta solicitud pasa al Controlador, el controlador depura la información y es enviada al Modelo, el modelo por lo general está protegido y puede con libertad traer la información de la base datos, entregársela al controlador, este la recibe y la retorna a la Vista, la vista imprime la información requerida por el usuario desde la interfaz de usuario.

El patrón MVC busca con este proceso de tres capas proteger la información de la base de datos, y dificultar su acceso por parte de terceros que deseen acceder a los datos confidenciales de los desarrollos web. Con la implementación de este modelo MVC se evitan ataques a la información sensible de la organización, se agrega seguridad al ocultar la data en la capa más profunda del desarrollo web.

5.3.1 Modularización del patrón MVC en el desarrollo POS

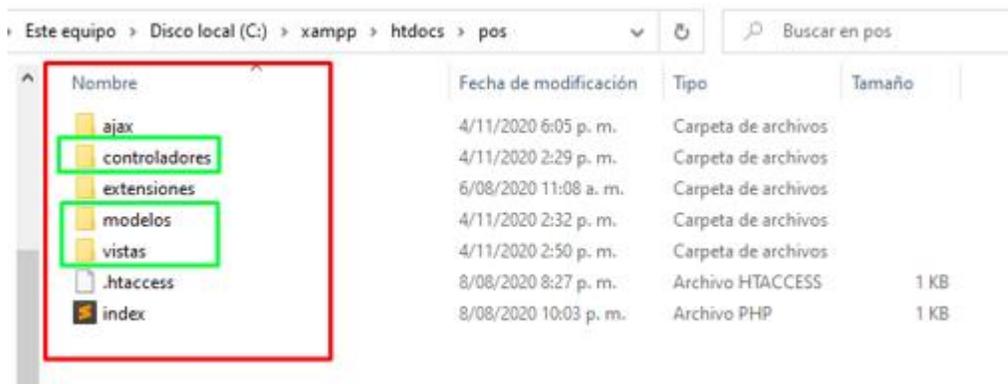
La modularización del patrón MVC es una técnica aplicada al desarrollo que permite su construcción por medio de módulos, cada módulo identifica un proceso, como la gestión de usuarios o la gestión de productos, los módulos se pueden ir agregando a la medida de las necesidades del proyecto, esto garantiza un orden en el desarrollo del código y la protección de la información sensible.

La estructura de directorio que se requiere para que se cumpla la jerarquía de acceso a los datos y se aplique la capa de seguridad se integra desde la estructura de directorio de la aplicación, en el directorio raíz de la aplicación se crearan tres directorios claves que identificarán la jerarquía del patrón MVC, ya que su objetivo principal es alejar la vista del modelo.

- Controladores
- Modelos
- Vistas

Figura 24.

Directorios patrón MVC.



Fuente propia.

El directorio vistas contiene toda información que permite la visualización de los datos de lado del usuario, los contenidos de las páginas de gestión de usuarios, gestión de productos, gestión de clientes, gestión de ventas.

El directorio “controladores” contiene la estructura de clases y los métodos de programación, recibe los eventos de entrada, como la respuesta al dar clic sobre un botón, contiene reglas de gestión de eventos. Estas acciones pueden suponer peticiones al modelo o a las vistas. Para nuestro proyecto desde el controlador se generan los llamados a los métodos en PHP, peticiones hechas al modelo para obtener datos de una tabla de la base de datos del proyecto.

El directorio modelos tiene la capacidad de acceder a la capa de almacenamiento de datos. Lo ideal para el desarrollo del POS es que este sea independiente del sistema de almacenamiento, el modelo define las reglas de las peticiones realizadas desde el controlador, es decir declara las clases de los métodos que provienen del controlador, el directorio modelos contiene los archivos que componen cada módulo de gestión de la aplicación POS, el código de cada archivo ejecuta las cláusulas de conexión con la base de datos y las tablas de datos:

5.4 Construcción del Modelo CRUD para las bases de datos

CRUD es el acrónimo que se aplica en la gestión de bases de datos (SGDB), se interpreta en el lenguaje básico de bases de datos, como una acción de Crear, Leer, Actualizar, Borrar, en otras palabras, CRUD resume las funciones requeridas por un usuario para crear, actualizar y gestionar datos en un ambiente de bases de datos relacionales.

Tabla 10.*Descripción del CRUD.*

CRUD	DESCRIPCION DE LA CLAUSULA
Créate	Crear registro (CREATE)
Read	Leer registro (RETRIEVE, SELECT)
Update	Actualizar registro (UPDATE)
Delete	Borrar registro (DELETE)

Fuente propia.

MySQL permite el acceso a través de operaciones CRUD, para manipular la base de datos se utiliza lenguaje script del lado del servidor de PHP y la extensión PHP Data Objects (PDO), PDO provee una capa de abstracción de acceso a datos para PHP, es una interfaz ligera pero consistente, de esta manera se consigue acceder desde el mismo programa PHP conectar con la base de datos POS (Nombre de la base de datos del sistema POS), para nuestro proyecto PDO es la interfaz ligera que transmite las consultas CRUD a la base de datos.

El requerimiento para utilizar PDO en el CRUD es crear las tablas desde el gestor de bases de datos, PhpMyAdmin es la interfaz de gestión de las bases de datos incluida en el programa XAMPP, esta aplicación integra Apache, MySQL y PHP versión 7 para Windows 64 bits. La conexión a la base de datos se alcanza con el método estático (static).

5.4.1 Diseño de las bases de datos para el sistema POS

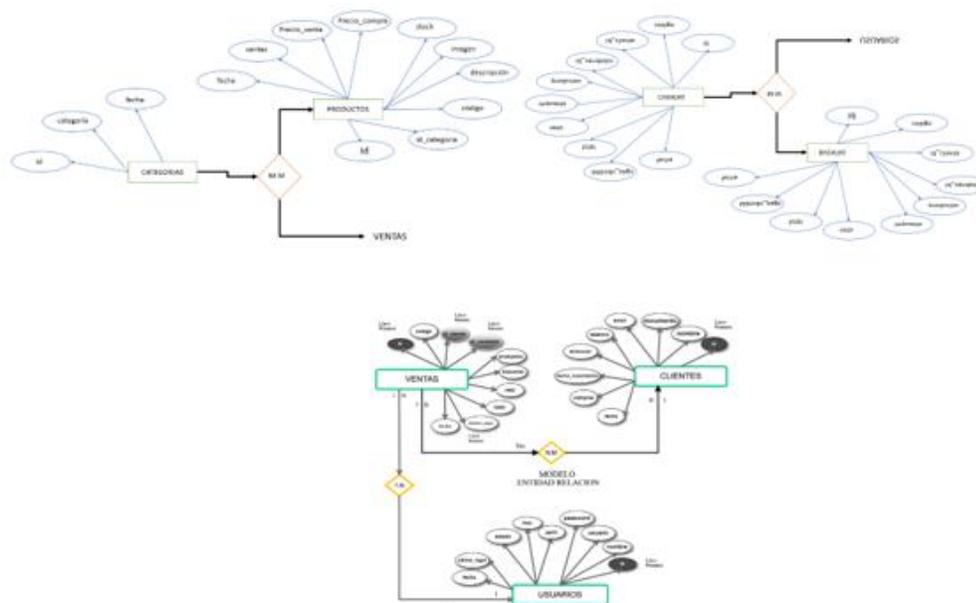
Antes de realizar el diseño físico y lógico de la base de datos, se maqueta el diseño conceptual, en este modelo inicial se captura la información relevante de los datos del problema a resolver, para el proyecto de implementación de un software que permita la automatización de los

procesos de punto de venta se presta atención a los actores, los elementos, los procesos, los artículos que interactúan en el registro y control de una venta.

En conclusión, el diseño conceptual es una interpretación directa de los actores (cliente) futuros usuarios del sistema y los requerimientos indicados en lenguaje no técnico pero detallado de los eventos relacionados con los puntos de venta.

Figura 25.

Diagrama del Diseño conceptual, base de datos POS.



Fuente propia.

El diagrama anterior recoge las interacciones que corresponden al modelo conceptual, en el diseño se presentan los elementos importantes encerrados en rectángulos que representan las entidades y luego se convertirán en tablas en el diseño lógico y físico de la base de datos, los elementos encerrados en semi círculos son los atributos de las tablas.

5.4.2 Modelamiento del diseño lógico de la Base de Datos sistema POS

En esta fase del diseño lógico se inicia la normalización, el proceso consiste en la conversión de la información del modelo conceptual en un esquema lógico, apoyado por las normas para depurar la información el diseño se lleva a columnas y filas para derivar en un conjunto de relaciones (tablas) para cada esquema lógico.

Para la derivación del esquema conceptual al diseño lógico, se transforman las entidades en una estructura de tablas organizadas que permiten alojar los datos en forma de filas o tuplas (registros), sin embargo, en la construcción de la base de datos se aplica un nivel más avanzado de organización (Normalización) que depurara los elementos que se puedan duplicar en la carga de la información.

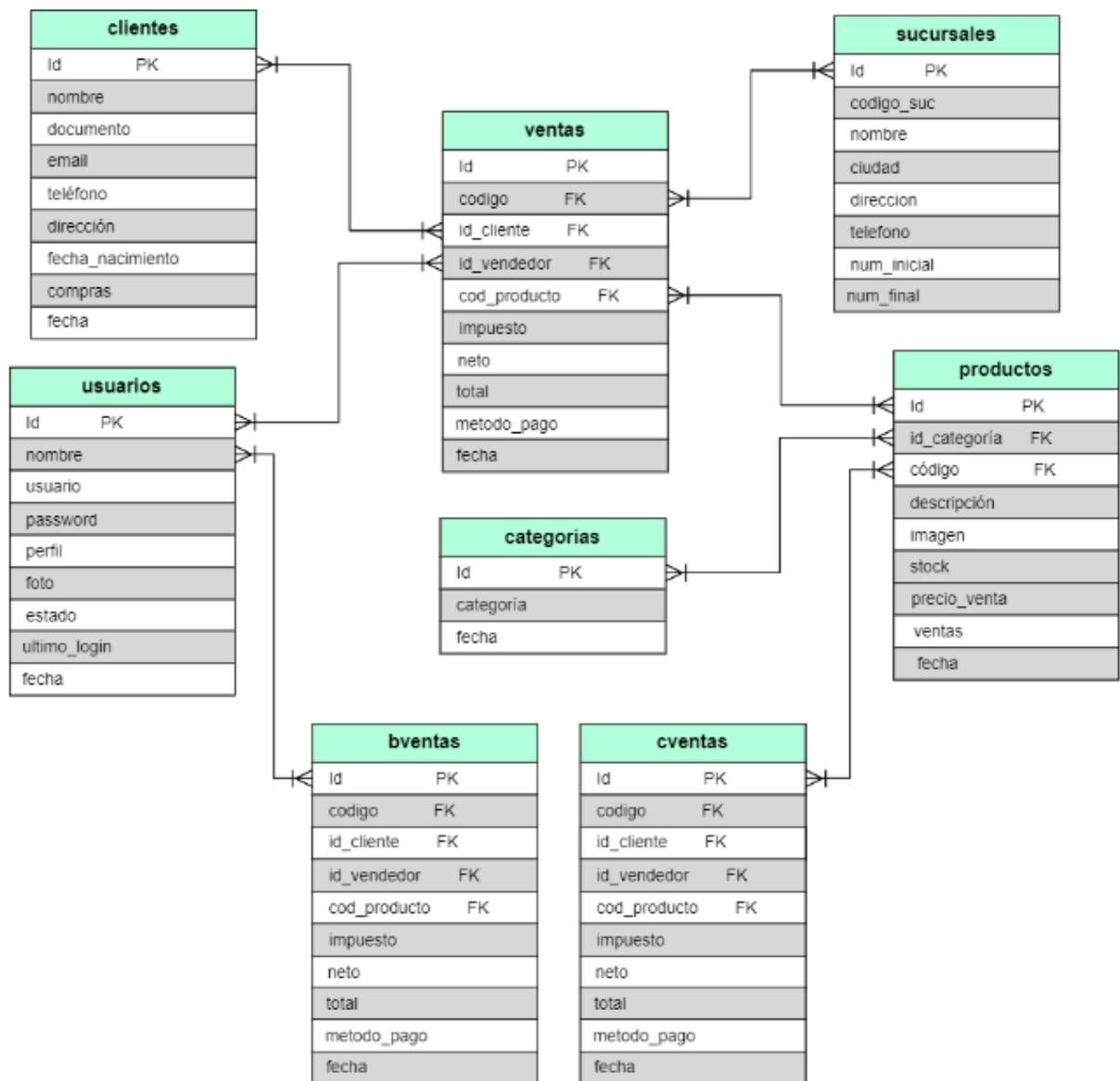
Aplicando la primera forma normal (1FN) se determina si existen en los atributos datos duplicados o redundancia de la información, también se puede identificar consultas innecesarias o complejas que pongan en un futuro errores de ejecución en la aplicación web.

Para que una tabla pueda cumplir con la segunda forma normal (2FN) debe cumplir con la 1FN, además que ningún campo este “fuera de lugar”, que haga parte de otra tabla, debe depender completamente de la llave primaria (PK).

El desarrollo POS aplica para la tercera forma normal 3FN depurando las tablas y revisando que sus atributos dependan directamente de la clave principal (PK), aquellos atributos que no dependan de la clave principal se agregaron a una segunda tabla relacionando ambas tablas.

Figura 26.

Diagrama Lógico de la base de datos, sistema POS.



Fuente propia.

Perfilada cada una de las tablas con sus atributos y columnas, se articula el modelo en un diseño relacional, la selección de las claves primarias permite al diseño la unión de las tablas y el concepto total de una base de datos relacional. MySQL en su estructura interna permite por medio de una clave primaria relacional una llave foránea en una tabla adicional.

5.4.3 Diseño Físico de la Base de Datos sistema POS

El modelo Físico de la base de datos representa la implementación del modelo lógico directamente sobre un gestor de bases de datos, en el caso de MySQL integra varios gestores de bases de datos (SGBD), la herramienta XAMPP incluye el gestor PHPMYADMIN, este manager facilita la construcción y gestión de las bases de datos MySQL, permitiendo el diseño de las relaciones preparadas en el modelo lógico.

El modelo Entidad-Relación es una correspondencia entre dos o más entidades, cada entidad tiene atributos, ejemplo: tabla categorías posee tres columnas las cuales se pueden nombrar también como atributos:

Los identificadores (PK) (FK), son los encargados de diferenciar una instancia de la entidad de otra distinta, los identificadores primarios (PK) o llaves primarias permiten identificar una entidad de una manera única, para el desarrollo POS se seleccionaron cada uno de los atributos id de cada tabla como identificadores primarios, tienen la tarea de dar identidad única a cada tabla y permitir la relación con otras tablas y atributos foráneos (FK).

5.4.4 Descripción de las tablas del sistema POS

La forma técnica para la descripción de las tablas en una base de datos es utilizando la herramienta del Diccionario de datos, el diccionario permite acceder a los detalles de la información de forma simple y ágil.

Tabla 11.

Tabla clientes base de datos POS.

clientes				
Contiene los datos de los clientes				
Nombre columna	Tipo de dato	K	N	Observaciones
id	Int.			Identificador único para la tabla clientes
nombre	varchar			Nombre del cliente
documento	varchar			Identifica el número documento del cliente
email	varchar			Identifica el correo del cliente
teléfono	varchar			Identifica el número de teléfono del cliente
dirección	varchar			Identifica la ubicación del cliente
fecha_nacimiento	date			Fecha de nacimiento del cliente
compras	int			Número de compras del cliente
fecha	timestamp			Fecha

Fuente propia.

Tabla 12.*Tabla usuarios base de datos POS.*

usuarios				
Contiene los datos de los usuarios				
Nombre columna	Tipo de dato	K	N	Observaciones
id	int			Identificador único para la tabla usuarios
nombre	varchar			Aloja nombre completo del usuario
usuario	varchar			Aloja el nombre del usuario
password	varchar			Aloja las claves cifradas de usuario
perfil	varchar			Aloja el tipo de perfil seleccionado
foto	varchar			Almacena la ruta de directorio de la foto
estado	Int			Aloja el estado, 0 o 1 del usuario
ultimo_login	datetime			Identifica la fecha del ultimo ingreso
fecha	timestamp			Toma la fecha de la creación del usuario

Fuente propia.

Tabla 13.*Tabla ventas base de datos POS.*

ventas				
Contiene los datos de las ventas				
Nombre columna	Tipo de dato	K	N	Observaciones
id	int			Identificador único para la tabla ventas
código	int			Identifica código de la factura de sucursal
id_cliente	int			Identificador del cliente en la factura
id_vendedor	int			Identifica el vendedor en la factura
productos	varchar			Almacena información de la factura
impuesto	float			Aloja los datos impuestos aplicado venta
neto	float			Total, de la venta antes de impuesto
total	float			Total, de la venta incluyendo impuesto
método_pago	varchar			Datos del método de pago
fecha	timestamp			Fecha de la venta

Fuente propia.

Tabla 14.*Tabla productos base de datos POS.*

productos				
Almacena la información de productos, mantiene el stock y la imagen				
Nombre columna	Tipo de dato	K	N	Observaciones
id	int			Identificador único para la tabla productos
Id_categoria	int			Identifica la categoría del producto
código	int			Identificador del producto
descripción	varchar			Descripción del producto
imagen	varchar			Almacena la ruta de la imagen producto
stock	int			Identifica cantidad de productos disponibles
precio_venta	float			Precio de venta del producto
ventas	int			Cantidades vendidas del producto
fecha	timestamp			Fecha ingreso producto

Fuente propia.

Tabla 15.*Tabla categorías base de datos POS.*

categorías				
Aloja la información referente a los tipos de categorías				
Nombre columna	Tipo de dato	K	N	Observaciones

id	int	Identificador único para la tabla productos
categoría	int	Identifica la categoría del producto
fecha	timestamp	Fecha ingreso categoría

Fuente propia.

Tabla 16.

Tabla bventas base de datos POS.

bventas				
Tabla bventas almacena información sucursal Bogotá				
Nombre columna	Tipo de dato	K	N	Observaciones
id	int			Identificador único para la tabla ventas
código	int			Identifica código de la factura de sucursal
id_cliente	int			Identificador del cliente en la factura
id_vendedor	int			Identifica el vendedor en la factura
productos	varchar			Almacena información de la factura
impuesto	float			Aloja los datos impuestos aplicado venta
neto	float			Total, de la venta antes de impuesto
total	float			Total, de la venta incluyendo impuesto
método_pago	varchar			Datos del método de pago
fecha	timestamp			Fecha de la venta

Fuente propia.

Tabla 17.*Tabla sucursales base de datos POS.*

sucursales				
Aloja la información referente a los tipos de categorías				
Nombre columna	Tipo de dato	K	N	Observaciones
id	int			Identificador único para la tabla productos
nombre	varchar			Nombre de la sucursal
código	int			Identifica la numeración de facturación
ciudad	varchar			Ciudad de la sucursal
fecha	timestamp			Fecha ingreso categoría

Fuente propia.

Tabla 18.*Tabla cventas sucursal Cali base de datos POS.*

cventas				
Tabla cventas almacena información sucursal ciudad de Cali				
Nombre columna	Tipo de dato	K	N	Observaciones
id	int			Identificador único para la tabla ventas
código	int			Identifica código de la factura de sucursal
id_cliente	int			Identificador del cliente en la factura
id_vendedor	int			Identifica el vendedor en la factura
productos	varchar			Almacena información de la factura

impuesto	float	Aloja los datos impuestos aplicado venta
neto	float	Total, de la venta antes de impuesto
total	float	Total, de la venta incluyendo impuesto
método_pago	varchar	Datos del método de pago
fecha	timestamp	Fecha de la venta

Fuente propia.

5.4.4 Arquitectura del sistema POS

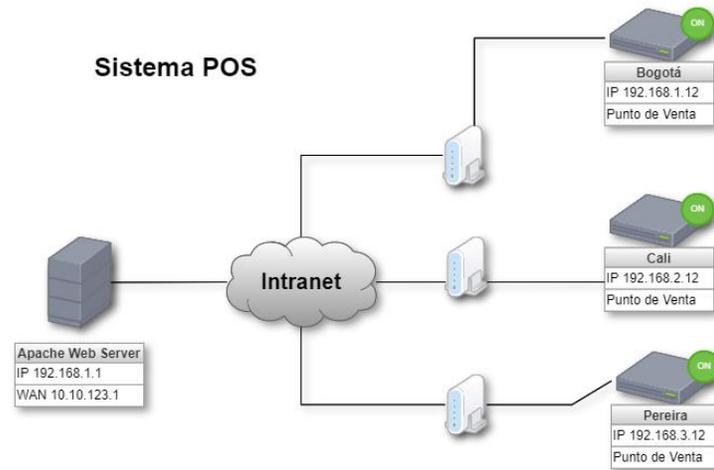
El desarrollo presenta una arquitectura web cliente servidor, diseñado para ser utilizado desde un navegador común, con la ventaja de implementarse en un ambiente de hosting o intranet corporativo, el modelo responde a interfaces de resolución grafica desde 1366 x 768, y resoluciones pequeñas en caso de llegar a necesitarse.

El diseño y arquitectura web permiten que los usuarios puedan acceder al sistema POS mediante un usuario autenticado y una contraseña cifrada en su almacenamiento, evitando que sea descifrada en el momento del ingreso. Los contenidos están estructurados y etiquetados de manera que al usuario le sea muy sencillo su navegabilidad una vez ha ingresado al sitio.

La distribución de los puntos de venta para las tiendas Broxxo se implementa de acuerdo con su infraestructura de red y comunicaciones, en la actualidad las tiendas están ubicadas en tres ciudades: Bogotá, Cali y Pereira, las sucursales se intercomunican utilizando una red tipo WAN, por tal razón el sistema POS se desarrolla en un ambiente web que facilita el acceso desde un cliente web.

Figura 27.

Distribución de los puntos de venta.



Fuente propia.

Resultados y Análisis

6 Resultados

En esta sección se presentan los resultados producto de las investigaciones realizadas, en el desarrollo de las diferentes fases del proyecto aplicando la metodología RUP, el modelamiento de los casos de uso aplicando el lenguaje UML, la maquetación de las interfaces del desarrollo web, y la construcción del código, creación del modelo CRUD para la base de datos, para dar cumplimiento al objetivo general de fabricar el sistema de software para puntos de venta POS.

La primera parte de este análisis la enfocaremos en los alcances del Proyecto, teniendo en cuenta que los desarrollos de software, orientados a cubrir soluciones de punto de venta tienden a ser complejos por la relación estrecha que existe entre las ventas y el proceso administrativo de la empresa, la venta de un producto afecta no solamente el inventario, también afecta la contabilidad central de la organización.

Según (Vidal Holguín, Londoño Ortega, & Contreras Rengifo, 2011; Vidal Holguín, Londoño Ortega, & Contreras Rengifo, 2011), “Uno de los problemas más complejos que afectan las empresas industriales y comerciales locales son los inventarios”. El actual desarrollo presenta una solución básica de un punto de venta para la organización Broxxo moda company, se agrega un módulo de inventario porque sin un inventario el punto de venta no aplica, se agrega un sistema de gestión de usuario porque necesita validar el usuario que realiza la venta, lo demás es el proceso interno de la aplicación que realiza los cálculos, conteo y actualiza stock de productos. Lo anterior es la funcionalidad básica de un punto venta, ese es el alcance del proyecto, está enfocado en cubrir la operatividad del punto de venta.

La implementación de la metodología RUP para el desarrollo del sistema POS, es garantía para producir software de calidad, agilizando los procesos de construcción por fases y generando

entregables en cada fase, que permitieron su evaluación, y solución a fallas generadas durante el desarrollo, obteniendo mejores resultados en el software.

6.1 UML el componente estratégico del desarrollo POS

La definición de los casos de uso aplicados a los módulos de sistema POS, proyecta una visión real del comportamiento del software y las diferentes interacciones de cambio y decisión que se puedan presentar durante la ejecución del sistema POS.

6.2 La Maquetación de las interfaces web, las ventajas del uso de plantillas

Parte del desarrollo de las interfaces del proyecto aplicado se basó en la plantilla AdminLTE, esta herramienta de código abierto y su diseño modular permite agilizar la construcción de las interfaces y ofrece la integración completa de las herramientas de desarrollo como JavaScript, CSS3, PHP, Bootstrap.

6.3 MySQL y PhpMyAdmin integran el SGBD de la base de datos del POS

MySQL potencia el desarrollo del sistema POS, su mantenimiento es sencillo, y permite la gestión de la información del punto de venta en línea, optimiza el uso de recursos, las transacciones del punto de venta no alteran las funcionalidades de toda la aplicación. El motor MySQL resulto ser rápido, las consultas de los productos a juicio y vista del usuario, no generan tardanza, en conclusión, no se perciben tiempos de espera.

6.4 Análisis

La integración de metodologías ágiles como RUP y lenguajes descriptivos enfocados en casos de uso como UML, permiten proyectar una técnica eficiente en la construcción de software, la invención de metodologías, para el desarrollo de software en el campo de la industria están dejando de lado la complejidad, y es un hecho que las técnicas de modelado de software permiten su desarrollo en productos de calidad.

El producto de software generado cumple con las especificaciones objetivas de la propuesta, el desarrollo en ambiente web es una ventaja frente a desarrollos para aplicaciones de escritorio o desarrollos no gráficos. El performance de la aplicación es multiusuario, puede prestar el servicio a varios usuarios registrados en la aplicación de punto de venta POS.

La aplicación permite al usuario mantener un inventario actualizado y stock de productos disponibles para la venta, el software desarrollado garantiza la salida del producto en tiempo y forma, manteniendo un umbral óptimo para la organización.

6.5 Pruebas de Producto final

El modelo de desarrollo RUP contiene en su fase de transición la aplicabilidad de producto desarrollado; dentro de este marco de trabajo se incluyen las pruebas de producto como parte del avance del proyecto, para nuestro sistema POS una serie de pruebas de concepto antes de llegar al producto final previo a la puesta en producción y despliegue del desarrollo, estas pruebas piloto detallan el funcionamiento correcto del proyecto y garantizan al usuario final como producto esperado que el desarrollo cumple con las funcionalidades descritas al iniciar el Proyecto.

Así como se describe en los objetivos propuestos del Proyecto de desarrollo POS, las pruebas de producto final integran todas las funcionalidades para las cuales fue construido, estas pruebas constituyen una serie de verificaciones prácticas de carga de productos cumpliendo todo el ciclo de consumo de servicios POS.

Estas pruebas se desarrollaron a lo largo del ciclo de desarrollo de producto en compañía del director encargado Ismael Ángel Romero, cada una de estas sesiones determinaron la continuidad del Proyecto y el avance en el desarrollo del producto de software. (ver ANEXO 1).

Tabla 19.*Fechas pruebas de concepto.*

FECHA PRUEBA	DESCRIPCION DE LA PRUEBA
07-02-2021	Modelo multifuncional de ambientes para varias sucursales
28-02-2021	Presentación de los módulos y consumo de servicios POS.
21-03-2021	Presentación de documento final y consumo de servicios POS

Fuente propia.

La integración del desarrollo POS con una metodología RUP entrega en cada fase una serie de pruebas de producto donde se detalla el avance del proyecto, se socializan en reuniones cortas cada una de las finalizaciones de la fase y su respectiva prueba de aplicación. (ver ANEXO 1)

Conclusiones & Recomendaciones

Conclusiones

El desarrollo del presente Proyecto de Aplicación ha contribuido a dinamizar una de las metodologías estándar, utilizadas en la construcción de software y su ciclo de vida, orientadas a los casos de uso, el Rational Unified Process (RUP), su eficacia se determina en el presente desarrollo por el manejo y la claridad de los diagramas que condujeron a tener una visión completa de los procesos requeridos para el sistema de Punto de Venta.

Otro factor relevante para destacar está enmarcado en la modularidad del software extensible con la cual se planteó y se aplicó el diseño del sistema punto de venta. Si bien es cierto que el concepto de modularidad abarca otros tópicos como la autonomía, composición, entendibilidad y protección, la extensibilidad se adhiere al proyecto de manera que se proveen las herramientas para que, en el desarrollo de futuros proyectos de sucursales nuevas, se puedan agregar módulos adicionales al sistema sin la necesidad de recurrir a cambios drásticos en el proyecto en producción.

En la introducción de este proyecto se menciona el surgimiento de tecnologías nuevas y la evolución de otras, que integradas componen una suite de aplicaciones flexibles que facilitan los desarrollos de software para puntos de venta, sin embargo, detrás de esta llamativa industria de software esta la base que abrió el camino “El Software libre”. El acceso, uso y aplicación gratuita a lenguajes de programación como PHP, gestores de bases de datos como MySQL, PhpMyAdmin, editores de código como Sublime Text 3, XAMPP distribución de Apache que incluye varios softwares libres como MariaDB, así como Perl y PHP, constituyen para el Proyecto y desarrolladores una gran oportunidad corporativa para la continuidad de nuevos modelos y el crecimiento del actual.

La funcionalidad proyectada en los objetivos del desarrollo POS son esenciales para los resultados alcanzados, aplicados en la solución del software punto de venta se convirtieron en el derrotero, priorizaron las actividades iniciales como los resultados de la recogida de los requisitos del proyecto, determinar los diseños de las interfaces de las páginas del sistema POS, la construcción de la base de datos y el modelado de clases y métodos del código del proyecto.

Analizadas las fases de desarrollo del software es importante orientar estas conclusiones a la problemática presentada en la propuesta inicial. El nuevo desarrollo de software de punto de venta parte de la premisa de implementar un sistema que mejore las circunstancias expuestas, como un sistema POS que permite el acceso desde un navegador web dejando a un lado el diseño del pos anterior, el rendimiento en los ambientes cliente servidor es compartido y la estructura interna del sistema está preparada para soportar la carga en cada punto físico de cada sucursal, el desarrollo no es un ambiente limitado a un sistema operativo, este puede ser instalado en un sistema operativo Windows o Linux realizando ajustes de orden general no en el código.

Recomendaciones

El sistema está diseñado para un ambiente web sin embargo es importante aclarar que no hay software invulnerable, para su alojamiento en un ambiente de hosting o publicación hacia internet es importante mantener un diseño de red seguro con un firewall y cifrado de encriptación como SSL.

Referencias

- Acid Carrillo, S., Marin Ruiz, N., Medina Rodriguez, M., Pons Capote, O., & Vila Miranda, A. (2008). *Introducción a las Bases de Datos*. Madrid, España: Ediciones Paraninfo S.A.
- Barahona Baquerizo, I., & Calle Lema, J. (Mayo de 2018). Modelado UML en el Diseño de Software. Milagro, Milaro, Ecuador.
- Bodin, W. (2004). *United States Patente nº US 6,681,989 B2*.
- DB-Engines. (2020). *DB-Engines Ranking*.
- Fernandez Portillo, A., Sanchez Escobedo, M., Jimenéz Naranjo, H., & Hernandez Mogollon, R. (2015). La Importancia de la Innovación en el Comercio electrónico. *Universia Business Review*.
- Ferre Grau, X., & Sanchez Segura, I. (2008). Desarrollo Orientado a Objetos con UML.
- Florez, A., & Thomas, J. (1993). La Teoria General de Sistemas. *Dialnet*, 10-16.
- Huertas , O. (2007). Importancia del portal o sitio web en la comunicación empresarial o corporativa. *Revista RE*.
- Manz. (12 de 09 de 2020). *Introducción a CSS*.
- Nuñez Islas, J. M., Ramirez Rosario, C. G., & Nava Rojo, M. (Mayo de 2016). Diseño y Desarrollo de un Punto de Venta con Aplicación Movil. Ciudad de Mexico, Distrito, Mexico.
- Oliveros, A., Danyans, F., & Mastropieto, M. (2014). Practica de Ingeniería de Requerimientos en el desarrollo de aplicaiones Web. *Faculta de Ingeniería y Ciencias Exactas*.
- Oppel, A. (2010). Fundamentos de Bases de Datos. En A. Oppel, *Fundamentos de Bases de Datos* (págs. 1-10). Bogotá.
- Pérez A., O. (2011). Cuatro enfoques metodológicos para el desarrollo de software. *Revista Uniminuto*.
- Ramirez, J. (2019). Industria Nacional de Software. *Portafolio*, 1.
- Suárez Cueto, A. (23 de 10 de 2010). Brevísimas historia de MySQL. Alicante, Raspeig, España.
- Torossi , G. (2019). El Proceso Unificado de Desarrollo de Software. *Diseño de sistemas*.

Torres, R., & Angel, M. (2014). *Desarrollo de Aplicaciones con PHP*. Madrid, España: Macro.

Vidal Holguín, C., Londoño Ortega, J., & Contreras Rengifo, F. (06 de 06 de 2011). Aplicación de los Modelos de Inventarios en una Cadena de Abastecimiento de Productos de Consumo Masivo con una Bodega y N Puntos de Venta. *Revista Universidad del Valle*, 1-10.

ANEXOS**Anexo 1*****Enlace de acceso al video Presentación Proyecto, (Julio 2021)***

Tema	Descripción
Presentación Proyecto (Video)	https://youtu.be/C93L5jzPRDY