

**FACULTAD DE INGENIERÍA**

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

Diseño e implementación de sistema informático para mejorar el proceso de compra y venta de la tienda DeLua

**Autor(es):**

Jose Carlos Tirado Linares

Julio Daniel Junior Cuadros Medina

**Curso:**

Diseño y Arquitectura de Software

**Docente:**

Jorge Alfredo Guevara Jiménez

LIMA – PERÚ

2020-2

**RESUMEN**

Nuestro proyecto tiene como objetivo aumentar la productividad de la tienda "DeLua" está comprometida con la producción y venta de ropa, Aumente la productividad mediante la implementación de software dedicado para el control Compra, venta, inventario y clientes.

Como forma de resolver problemas de investigación, Aprenda de "Problemas en el uso de sistemas de inventario antiguos" Los problemas encontrados en la tienda DeLua y los problemas que encontraron Toda la gestión se realiza a la hora de gestionar las acciones de la empresa. Una Realizar una investigación más detallada para comprender y resolver las necesidades. Órdenes, eficiencia y producción del sistema comercial DeLua Varios casos de uso para comprender el correspondiente proceso de compra y venta de la tienda.

Los resultados obtenidos muestran que el software desarrollado con este software La optimización del proceso de compra y venta de la tienda se ha incrementado considerablemente. Y reduce el tiempo de uso de cada función. Con todo, el proyecto mostró eficiencia, seguridad, Mejore la productividad y el rendimiento de la empresa al poseer un nuevo sistema Computadora utilizada para la gestión de la información.

**Palabra clave:** *Software, gestión de inventarios, control de compras y ventas, indumentaria Perú, cultura peruana.*

**Abstract**

Our project aims to increase the productivity of the store "DeLua" is committed to the production and sale of clothing, increase productivity by implementing dedicated software to control Purchase, sale, inventory, and customers.

To solve research problems, learn from "Problems in the use of old inventory systems" The problems found in the DeLua store and the problems they encountered All management is done when managing the actions of the company. A Conduct more detailed research to understand and resolve needs. DeLua Commercial System Orders, Efficiency and Production Various use cases to understand the corresponding store buying and selling process.

The results obtained show that the software developed with this software the optimization of the store's buying and selling process has increased considerably. And it reduces the usage time of each function. All in all, the project showed efficiency, security, Improve the productivity and performance of the company by having a new computer system used for information management.

**Keyword:** *Software, inventory management, purchase and sales control, Peruvian clothing, Peruvian culture.*

**CAPÍTULO I: INTRODUCCIÓN**

1. **Identificación del problema**
2. **Planteamiento de la solución**

La solución propuesta por el grupo es implementar software especializado Gestión de inventarios a través de la compra y venta de tiendas. Se demostrará que los informes mensuales y de clientes tienen un buen control de inventario.

1. **Cómo implementar la solución**

El proyecto se llevará a cabo analizando los métodos de gestión de inventarios. La empresa utiliza técnicas de programación para crear una base de datos y adjuntar Software de gestión de inventarios.

1. **Justificación**

● Mejorar el proceso de registro de transacciones.

● Optimizar la gestión de inventario.

● Implementar software especial para mejorar el proceso.

1. **Limitaciones de la investigación**

● Los propietarios de la tienda tienen tiempo libre para coordinar Implemente el software en su base de datos existente.

● Falta de software profesional para la gestión de inventarios.

● Aplicar los procesos comerciales de la empresa al software que se está desarrollando.

1. **Ventajas de la solución**

● Servicio más rápido a la hora de realizar la venta.

● Mejor control del stock en la tienda.

● Tener más orden a la hora de realizar las compras de los proveedores.

● Mayor seguridad ya que el sistema cuenta con contraseña.

● Controlar mejor las ventas que se producirán.

● Lograr una mayor productividad a la hora de vender.

1. **Desventajas de la solución**

● Se tendrían pérdidas de inversión por los costos que conllevaría la

implementación del software.

● Desventaja temporal al tener que capacitar al empleado para que realice su

labor de atención al cliente.

● Posibles confusiones al principio a la hora de realizar los comandos de compra

y venta usando el software.

1. **Objetivos generales**

Desarrollar un software especializado para la mejora del proceso de compra, venta y manejo del stock de la tienda “DeLua”.

1. **Objetivos específicos**

• Desarrollar un sistema que permita iniciar sesión al usuario para registrar

productos y después mostrarlo en un catálogo.

• Realizar el mantenimiento de datos sobre los pedidos para realizar envíos y

poder visualizarlos.

• Desarrollar un sistema que permite realizar compras al proveedor y mantener

un control eficiente de los productos.

**CAPÍTULO II: MARCO TEÓRICO**

* 1. **Marco Teórico**
     1. **PROCESO DE NEGOCIOS**

Son procesos utilizados en los negocios para lograr algún objetivo del negocio. Un ejemplo de un proceso de negocio en una compañía de seguros sería emitir una política de seguros en una fábrica, un proceso de negocio sería aceptar un pedido para los productos y estipular el proceso de fabricación asociado. Los procesos de negocio pueden ser diseñados alrededor de un sistema heredado y restringidos por la funcionalidad que éste proporciona (Sommerville, 1989, p. 52).

* + 1. **Lenguaje de Modelado Unificado**

“Lenguaje gráfico utilizado en el desarrollo orientado a objetos que incluye varios tipos de modelos del sistema que proporcionan distintas vistas de un sistema. UML se ha convertido en un estándar de facto para el modelado orientado a objetos.” (Sommerville, 1989, p. 659)

* + 1. **Casos de Uso**

Rescatado de Sommerville (2005) “Ofrecen información adicional y detallada acerca de los estados de casos de uso analizados en el esquema y enunciado de problemas o casos con problemáticas, podemos decir que simplifica de cierta manera el trabajo de actores participantes”. (Capitulo 16). Los casos de relacionados reúnen toda la evidencia que demuestre que un sistema es seguro y confiable. El proceso debe incluir la identificación y la monitorización de peligros potenciales. Estos casos de uso relacionados deben ser representados de distintas formas, ya sea por medio de una inclusión, generalización, asociación, entre otros casos. Los participantes de los Casos de Uso no necesariamente comparten mismos roles en el sistema, dado que no todos tienen las mismas funcionalidades, pero se podrá simplificar tareas y casos de uso para una mejor distribución de actividades por parte de los actores. (Capitulo 16).

* + 1. **Modelo de Procesos del Software**

Sommerville (1989) afirma: Un modelo de procesos del software es una descripción simplificada de un proceso del software que presenta una visión de ese proceso. Estos modelos pueden incluir actividades que son parte de los procesos y productos de software y el papel de las personas involucradas en la ingeniería del software. (p. 25)

* + 1. **Proceso Unificado**

“Es un proceso del software diseñado como estructura para los métodos y herramientas del UML, y está impulsado por el caso de uso, centrado en la arquitectura, y es iterativo e incremental.” (Pressman, 2010, p. 80)

* + 1. **Requerimientos Funcionales**

Son aquellos que describen cualquier actividad que se deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones.

Por lo general, estos deben incluir funciones desempeñadas por pantallas específicas, descripciones de los flujos de trabajo a ser desempeñados por el sistema y otros requerimientos de negocio, cumplimiento, seguridad u otra índole. (Sommerville, 1989, p. 59).

* + 1. **Requerimientos No Funcionales**

Son los que especifican criterios para evaluar la operación de un servicio de tecnología de información, en contraste M. Fernando, H. Santiago, C. Marco pág. 12 con los requerimientos funcionales que especifican los comportamientos específicos. Por lo general, el plan para implementar los requerimientos no funcionales se detalla en la Arquitectura del Sistema, mientras que el de los requerimientos funcionales se especifica en el Diseño. (Pressman, 2010, p. 114)

* + 1. **Plan de pruebas**

Es un producto formal que define los objetivos de la prueba de un sistema, establece y coordina una estrategia de trabajo, y provee del marco adecuado para elaborar una planificación paso a paso de las actividades de prueba. El plan se inicia en el proceso Análisis del Sistema de Información (ASI), definiendo el marco general, y estableciendo los requisitos de prueba de aceptación, relacionados directamente con la especificación de requisitos. (Sommerville, 1989, p. 659)

* + 1. **Modelo Incremental:**

Tiene como objetivo un crecimiento progresivo de la funcionalidad. Es decir, el producto va evolucionando con cada una de las entregas previstas hasta que se amolda a lo requerido por el cliente o destinatario. (Sommerville, 1989, p. 425)

* + 1. **Diagrama de Secuencias y Clases**

El diagrama de secuencia es un tipo de diagrama del lenguaje unificado de modelado (UML) que, a su vez, se trata de un lenguaje orientado a objetos y está compuesto por elementos gráficos. UML modela sistemas y procesos de la programación orientada a objetos, así como procesos de negocio con el objetivo de presentar asuntos complejos de manera clara. (Sommerville, 1989, p. 659)

* + 1. **Modelo de Diseño de clases y datos**

El diagrama de clases recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos, pero no muestra información temporal.

* + 1. **Mapa de Procesos**

Es una representación interrelacionada entre sí de los procesos que ocurren dentro del negocio, así como también brinda la visión dentro del conjunto de actividades que realizan y también sobre el uso de recursos de soporte. (Sommerville, 1989)

* + 1. **Diagrama de Casos de Usos**

Rescatado de Sommerville (2005) “Un modelo independiente de plataforma que modele la operación del sistema sin referencia a su implementación. El PIM se describe usualmente mediante modelos UML, que muestran la estructura estática del sistema y como responde a eventos externos e internos”. (Capitulo 15). Estos casos de uso pueden representados de distintas formas, ya sea en una definición o de forma diagramática. Los participantes de los Casos de Uso no necesariamente comparten mismos roles en el sistema, dado que no todos tienen las mismas funcionalidades. (Capitulo 15).

* + 1. **Especificaciones de Casos de Uso**

Rescato por Sommerville (2005) “clientes e ingenieros definen el software que se producirá y las restricciones en su operación” (Capitulo 2). Las especificaciones formales no solo son importantes para una verificación del diseño e implementación del software, sino también son la forma más precisa de especificar sistemas y, por ende, de reducir el ámbito para las malas interpretaciones. Más aún, construir una especificación formal fuerza un análisis detallado de los requerimientos y esta es una forma efectiva de descubrir problemas de requerimientos. (Capitulo 12).

* + 1. **MySQL Workbench**

Rescatado de UbunLog.com Este programa es el sucesor de DBDesigner 4. MySQL Workbench es uno de los primeros productos de la familia MySQL que ofrece dos ediciones diferentes, una open source y una edición comercial. Este software da al usuario la posibilidad de modelado de datos, desarrollo de SQL y herramientas de administración completas para la configuración de servidores, administración de usuarios, copias de seguridad y mucho más. MySQL Workbench está disponible en Windows, GNU/Linux y Mac OS X. MySQL Workbench permite diseñar visualmente, modelar, generar y administrar bases de datos. Incluye todo lo que necesita un modelador de datos para crear modelos complejos de ER, ingeniería directa e inversa y también ofrece características clave para realizar tareas difíciles de gestión de cambios y documentación que normalmente requieren mucho tiempo y esfuerzo.

* + 1. **Spring**

Spring es un framework que es una alternativa al conjunto de tecnologías estándar en aplicaciones JavaEE que popularizo ideas como inyección de dependencias o el uso de objetos convencionales, como objetos de negocio, asimismo spring es un conjunto de portables entre servidores, estas ideas permiten el desarrollo más sencillo, rápido para aplicaciones. (Johnson, 2012)

* + 1. **Diagrama de Secuencias**

Este diagrama indica la estructura en la que los eventos provocan transiciones de un objeto a otro, estos son identificados por medio del análisis de casos de uso, es el modo en el cual los eventos causan el flujo de uno a otro como función del tiempo. (Pressman, 2010)

* + 1. **Diagrama de subsistencia**

Mediante este diagrama para poder facilitar la construcción de aplicaciones, permite agrupar en paquetes siguiendo un criterio lógico, a través de subsistemas, para añadir las nociones de bibliotecas de compilación, también de gestión de configuración a la semántica que ya se tiene en cada paquete. (Weitzenfeld, 2005)

* + 1. **Modelo tubería y filtro**

Es un estilo de la arquitectura de software donde los filtros se toman como un proceso y las tuberías conectan los siguientes filtros conectados entre sí, siendo un estilo unidireccional muy usado en el tipo de sistema operativo UNIX. (Sommerville, 1982)

* + 1. **Patrón CRC**

El patrón de diseño nombrado como tarjetas CRC nos sirve para poder tener una orientación entre los objetos de las clases, en relación con sus responsabilidades que son extraídas de los diagramas de colaboración como lo son los casos de uso o también puede ser representado en un diagrama de entidades. (Weitzenfeld, 2005)

* + 1. **Patrón SOA**

El patrón de diseño sobre la arquitectura orientada a servicios se caracteriza por ofrecer funcionalidades adicionales a un software, mediante la reutilización de servicios o aplicaciones que tienen la capacidad de brindar mayor soporte y funcionalidad a un sistema. (Monteagudo Mezo & Duran Heras, 2008)

* + 1. **Lucid chart**

Es un software que permite a los usuarios colaborar y trabajar juntos en tiempo real para crear diagramas de flujo, organigramas, estructuras alámbricas de sitio web, diseños UML, mapas mentales, prototipos de software, y muchos otros tipos de diagramas.

* + 1. **GitHub**

Es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. El programa aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto. Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls. Rescatado de kinsta.com (2020)

* + 1. **Balsamiq**

Es una aplicación/servicio pues no sólo cuenta con una aplicación nativa para OS X (también Windows y Linux) sino también con una versión web, de modo que podemos trabajar desde cualquier lugar. Y su finalidad no es otra que ayudar al desarrollo de aplicaciones con una herramienta que facilita la creación de esquemas y el desarrollo de diseños en o para aplicativos webs. Rescatado de isdi. education (2014)

**2.2 MARCO METEDOLÓGICO**

**2.1.1 PASO 1**

Buscamos una tienda que no disponga de un software especializado.

**2.1.2 PASO 2**

Nos ponemos en contacto con el dueño de la tienda.

**2.1.3 PASO 3**

Le realizamos la propuesta del software para la y esperamos su aprobación.

**2.1.4 PASO 4**

Comenzamos a realizar preguntas sobre la funcionalidad y el proceso de trabajo de la tienda.

**2.1.5 PASO 5**

En base a las respuestas obtenidas, realizamos el modelado y especificación de procesos del negocio.

**2.1.6 PASO 6**

Luego modelamos los casos de uso del software o sistema que se realizará para la tienda.

**2.1.7 PASO 7**

Se modeló el plan de iteraciones MVP para ayudarnos a entender un poco más las necesidades de la tienda.

**2.1.8 PASO 8**

Se modeló el SWAPP entendiendo así mejor los entornos en el que se ejecutara el programa.

**2.1.9 PASO 9**

Se usaron los diagramas MVC y ACME asignando roles y fechas otorgando un mejor orden a nuestro trabajo.

**2.1.10 PASO 10**

Se modeló el diagrama de tubería y filtro asegurando la conectividad de datos dentro de los componentes que conforman los casos de uso.

**2.1.11 PASO 11**

Se modeló el diagrama de repositorio dando al proyecto mayor seguridad de información.

**2.1.12 PASO 12**

Se modeló el diagrama de secuencia evidenciando de mejor manera el paso a paso de los casos de uso principales.

**2.1.13 PASO 13**

Se modeló el diagrama CRC del software.

**2.1.14 PASO 14**

Se modeló el diagrama SOA orientado al envío de información vía email.

**2.1.15 PASO 15**

Se modeló el diagrama de ingeniería inversa en nuestro sistema de ventas.

**2.1.16 PASO 16**

La tienda usará un método de entrega para clientes dentro de Lima Metropolitana a través de Glovo y Rappi y para provincias a través de Olva Courier

**CAPÍTULO III: DESARROLLO DE LA SOLUCIÓN**

**3.1 REQUESITOS FUNCIONALES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | REQUISITO | CASO DE USO | Prototipo | Versión |
| 1 | El sistema permite validar al usuario y su contraseña | Valida al usuario | si | V1.0 |
| 2 | El sistema permite agregar usuarios | Permite agregar usuario | si | V1.0 |
| 3 | El sistema permite eliminar usuarios | Permite eliminar usuario | si | V1.0 |
| 4 | El sistema muestra un menú de opciones | Muestra menú | si | V1.0 |
| 5 | El sistema muestra el catálogo | Muestra catálogo | si | V1.0 |
| 6 | Permite modificar productos | Agregar productos al stock | si | V1.0 |
| 7 | Permite modificar productos | Elimina productos en el stock | si | V1.0 |
| 8 | Permite modificar productos | Editar producto | si | V1.0 |
| 9 | El sistema muestra el listado de clientes | Muestra clientes | si | V2.0 |
| 10 | Permite modificar el listado de clientes | Permite agregar clientes | si | V2.0 |
| 11 | Permite modificar el listado de clientes | Permite eliminar clientes | si | V2.0 |
| 12 | Permite modificar el listado de clientes | Permite editar clientes | si | V2.0 |
| 13 | El sistema muestra el listado de pedidos | Muestra pedidos | si | V2.0 |
| 14 | Permite modificar el listado de pedidos | Permite agregar pedidos | si | V2.0 |
| 15 | El sistema muestra el listado de materiales | Muestra materiales | si | V2.0 |
| 16 | Permite modificar materiales | Permite agregar materiales | si | V2.0 |
| 17 | Permite modificar materiales | Permite eliminar materiales | si | V2.0 |
| 18 | Permite modificar materiales | Permite editar materiales | si | V2.0 |
| 19 | El sistema muestra el listado de proveedores | Muestra proveedores | si | V2.0 |
| 20 | Permite modificar los proveedores | Permite agregar proveedor | si | V2.0 |
| 21 | Permite modificar los proveedores | Permite eliminar proveedores | si | V2.0 |
| 22 | Permite modificar los proveedores | Permite editar proveedores | si | V2.0 |
| 23 | El sistema muestra el listado de ventas | Muestra ventas | si | V3.0 |
| 24 | Permite modificar el listado de ventas | Permite agregar ventas | si | V3.0 |
| 25 | El sistema muestra el listado de empleados | Muestra empleados | si | V3.0 |
| 26 | Permite modificar empleados | Permite agregar empleados | si | V3.0 |
| 27 | Permite modificar empleados | Permite eliminar empleados | si | V3.0 |
| 28 | Permite modificar empleados | Permite modificar empleados | si | V3.0 |

**Se identifico los siguientes requisitos funcionales de la tienda estudiada para después crear los casos de uso que serán incluidos en un diagrama.**

**3.1.2 MAPA DE PROCESOS**

Diagrama

Descripción generada automáticamente

**3.1.3 DIAGRAMA DE CASOS DE USO RELACIONADO**

Diagrama

Descripción generada automáticamente

**3.1.4 ESPECIFICACION DE CASOS DE USO**

|  |  |
| --- | --- |
| ÍTEM | DESCRIPCIÓN |
| Nombre Corto | Registrar productos |
| Actores | Vendedor |
| Objetivo | Registrar productos para la venta de la empresa DeLua |
| Disparador | Disponibilidad de productos |
| Pre-condiciones | Usuario registrado como vendedor |
| Post-condiciones | Producto registrado |
| Escenario Básico | * Si no abre la sesión indicará un error. * Si no abre la sesión indicará una solución. * Si no inserta o confirma los datos indicará solución. |
| Escenario Alternativo | * Si no abre la aplicación indicara un error. * Si no abre la sesión indicara una solución. * Si no inserta o confirma los datos indicara la solución. |
| Prioridad | Alta |

**Se tomo un caso de uso principal del diagrama para poder describirlo, conocer el objetivo de este, y saber cómo se desempeñará el sistema en cada escenario.**

|  |  |  |
| --- | --- | --- |
| ID | ÍTEM | DESCRIPCIÓN |
| 1 | Nombre Corto | Registrar clientes |
| 2 | Actores | Vendedor |
| 3 | Objetivo | Registrar clientes que realizan compras a la empresa DeLua |
| 4 | Disparador | Venta de prendas a clientes |
| 5 | Pre-condiciones | Cliente realiza compra por primera vez |
| 6 | Post-condiciones | Cliente registrado |
| 7 | Escenario Básico | * Abrir sistema. * Iniciar Sesión como vendedor. * Ingresar datos de clientes. * Confirmar datos ingresados. * Grabar información. |
| 8 | Escenario Alternativo | * Si no abre la aplicación indicara un error. * Si no abre la sesión indicara una solución. * Si no inserta o confirma los datos indicara la solución. |
| 9 | Prioridad | Alta |

**Se tomo un caso de uso principal del diagrama donde se puede realizar una explicación del flujo que tendrá dentro del sistema.**

|  |  |  |
| --- | --- | --- |
| ID | ÍTEM | DESCRIPCIÓN |
| 1 | Nombre Corto | Registrar pedido |
| 2 | Actores | Cliente |
| 3 | Objetivo | Registrar pedido de producto en la empresa DeLua |
| 4 | Disparador | Necesidad de vestir |
| 5 | Pre-condiciones | Usuario con acceso a la red social de DeLua |
| 6 | Post-condiciones | Pedido realizado |
| 7 | Escenario Básico | * Abrir sistema. * Iniciar Sesión como usuario. * Revisión del catálogo y elección pedido. * Confirmación de pedido. * Grabar información. |
| 8 | Escenario Alternativo | * Si no abre la aplicación indicara un error. * Si no abre la sesión indicara una solución. * Si no inserta o confirma los datos indicara la solución. |
| 9 | Prioridad | Alta |

**Se tomo un caso de uso principal del diagrama donde se puede realizar una explicación del flujo que tiene la realización de un pedido para su seguimiento.**

|  |  |  |
| --- | --- | --- |
| ID | ÍTEM | DESCRIPCIÓN |
| 1 | Nombre Corto | Registrar compra |
| 2 | Actores | Empleado |
| 3 | Objetivo | Registrar compra de materiales en la empresa DeLua |
| 4 | Disparador | Necesidad de abastecerse de materiales |
| 5 | Pre-condiciones | Usuario con acceso a la red social de DeLua |
| 6 | Post-condiciones | Compra realizada |
| 7 | Escenario Básico | * Abrir sistema. * Iniciar Sesión como usuario. * Revisión de materiales y elección de compra. * Confirmación de compra. * Grabar información. |
| 8 | Escenario Alternativo | * Si no abre la aplicación indicara un error. * Si no abre la sesión indicara una solución. * Si no inserta o confirma los datos indicara la solución. |
| 9 | Prioridad | Alta |

**Se tomo un caso de uso principal del diagrama donde se puede realizar una explicación del flujo que tiene la realización de un pedido para su seguimiento.**

**4. REPOSITORIO GitHub**

**5. PLAN DE ITERACIONES**

**Escala de tiempo

Descripción generada automáticamente**

**6. DIAGRAMA MVC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Casos de Uso | Clases | Fecha Inicio | Fecha Final | Responsable |
| Inicio de sesión con credenciales | M\_INICIO\_SESION | 5-abril | 15-abril | José Carlos |
| V\_INICIO\_SESION |
| C\_INICIO\_SESION |
| M\_VAL\_INICIO\_SESION |
| Registrar Usuario  Modificar Usuario  Eliminar Usuario  Registrar Producto  Modificar Producto  Eliminar Producto | A\_REG\_ USER | 15-abril | 30-abril | Julio Cuadros |
| M\_REG\_USER |
| C\_REG\_USER |
| D\_REG\_USER |
| V\_CAT\_PROD |
| C\_PRO |
| Mostrar Catalogo | M\_CAT\_PROD | 30-abril | 9-mayo | Kathy Pacara |
| V\_CAT\_PROD |

**Se implemento el plan de unidades de programación para poder establecer los plazos que tendrán cada uno de los incrementos planteados.**

**7. DIAGRAMA ACME**

|  |  |  |  |
| --- | --- | --- | --- |
| Tabla | SP | Fecha Inicio | Fecha Final |
| USUARIO | A\_TABLA\_VENDEDOR | 7-Abril | 9-Abril |
| C\_TABLA\_VENDEDOR | 9-Abril | 11-Abril |
| M\_TABLA\_VENDEDOR | 11-Abril | 13-Abril |
| E\_TABLA\_VENDEDOR | 13-Abril | 15-Abril |
| PRODUCTOS | A\_TABLA\_PRODUCTO | 15-Abril | 17-Abril |
| C\_TABLA\_PRODUCTO | 17-Abril | 19-Abril |
| M\_TABLA\_PRODUCTO | 19-Abril | 21-Abril |
| E\_TABLA\_PRODUCTO | 21-Abril | 23-Abril |

**Se implemento la siguiente programación de las tablas que será parte de la base de datos que contendrá la información del software.**

**8. DIAGRAMA ENTIDAD RELACIÓN**

**Diagrama

Descripción generada automáticamente**

**9. DIAGRAMA DE ACTIVIDADES**

**Diagrama

Descripción generada automáticamente**

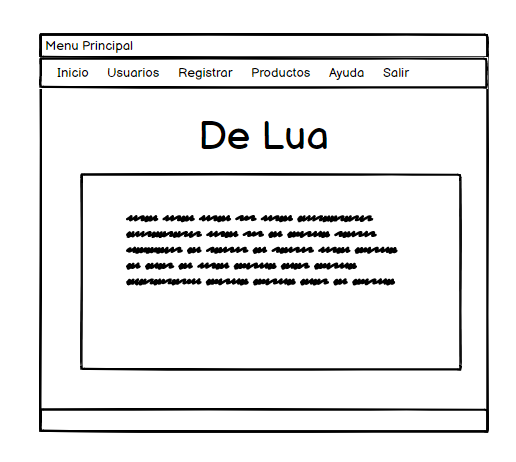
**10. PROTOTIPOS DE SOFTWARE**

**Texto

Descripción generada automáticamente con confianza media**

**Diagrama

Descripción generada automáticamente con confianza media**

**Texto

Descripción generada automáticamente con confianza media**

**Texto

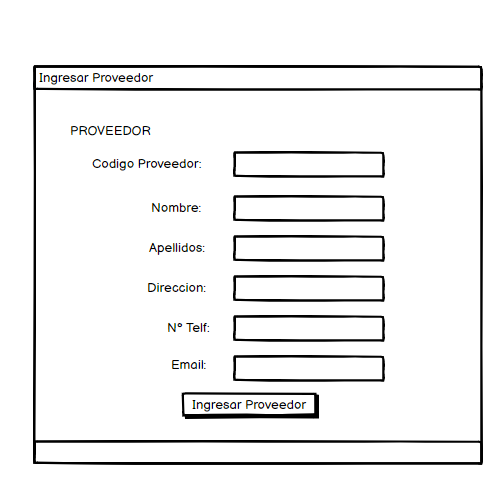
Descripción generada automáticamente con confianza mediaTexto

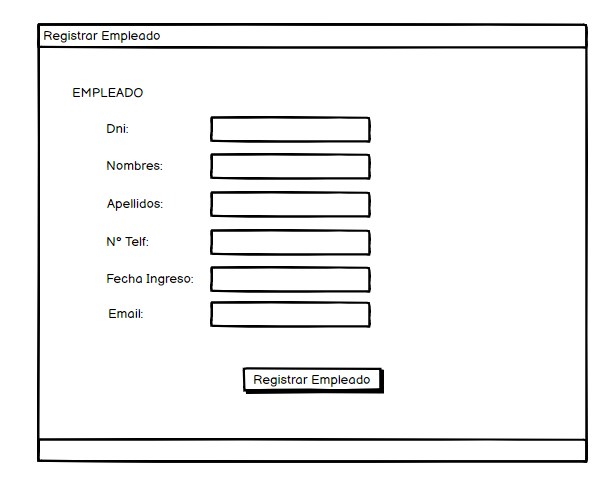
Descripción generada automáticamente con confianza media**

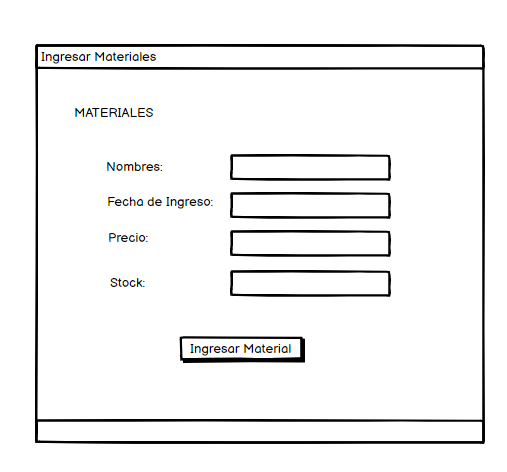
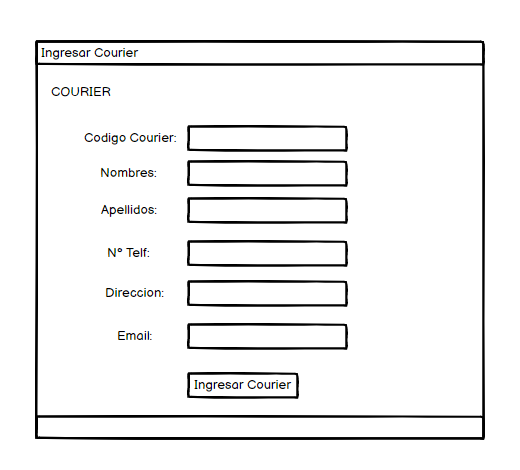
**Tabla

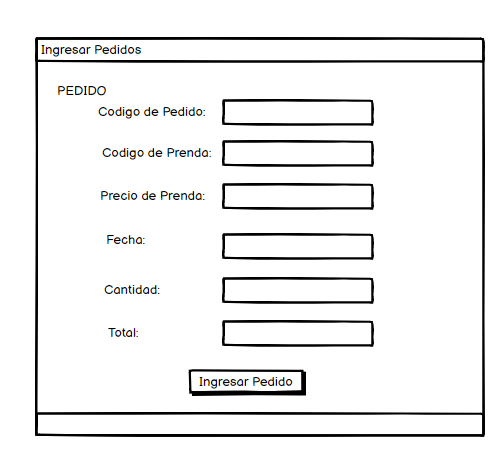
Descripción generada automáticamenteTabla

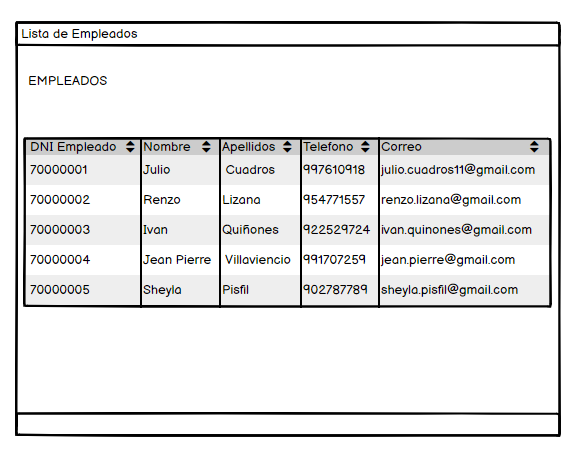
Descripción generada automáticamente**











**CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES**

**11.1 CONCLUSIONES**

**11.2 RECOMENDACIONES**

**11.3 REFERENCIA BIBLIOGRÁFICAS**

Sánchez, M. S. L., & Lidia, M. V. L. (2011, enero). Sistema de Información para el Control de Inventarios del Almacén del ITS. Recuperado de <https://dialnet.unirioja.es/descarga/articulo/3707498.pdf>

Martínez, L. T. M. (2017, 31 enero). Evaluación del Manejo de Inventarios en industrias PYMES. Recuperado de [http://reaxion.utleon.edu.mx/Art\_Evaluacion\_del\_Manejo\_de\_Inventarios\_en\_in dustrias\_PYMES.html](http://reaxion.utleon.edu.mx/Art_Evaluacion_del_Manejo_de_Inventarios_en_in%20dustrias_PYMES.html)

Roger, Pressman (2010) INGENIERÍA DE SOFTWARE. UN ENFOQUE PRÁCTICO. Recuperado de <http://artemisa.unicauca.edu.co/~cardila/Libro_Pressman_7.pdf>

Ávila, D. Y. A. A. (2013, 15 febrero). IMPLEMENTACIÓN DE UN SISTEMA DE INVENTARIOS EN FD FILTROS Y REPUESTOS LTDA. Recuperado de<https://repository.unilibre.edu.co/bitstream/handle/10901/9385/Proyecto%20de%20grado%20(2).pdf?sequence=1&isAllowed=y>

Ian Sommerville (1989). INGENIERÍA DE SOFTWARE. SÉPTIMA EDICIÓN. Recuperado de<https://ulagos.files.wordpress.com/2010/07/ian-sommerville-ingenieria-de-software-7-ed.pdf>

**11.4 ANEXOS**