



**UNIVERSIDAD TÉCNICA DE ORURO**

**FACULTAD NACIONAL DE INGENIERÍA**

**INGENIERÍA DE SISTEMAS E INGENIERÍA INFORMÁTICA**

**SISTEMA DE INFORMACION PARA EL TALLER DE SERVICIOS AUTOMOTRICES “LA RESISTENCIA”**

**MATERIA:** Ingeniería de Software I (INF 2720)

**SEMESTRE:** I/2020

**INTEGRANTES:** Benítez Díaz Néstor Alejandro

Lazcano Ballesteros Lexer

Ugarte Torrez Jenny Laura

**FECHA DE ENTREGA:** 09/07/2020

**ORURO – BOLIVIA**

**CONTENIDO**

[**TITULO DEL PROYECTO:** 5](#_Toc47560785)

[**1.** **INTRODUCCION** 5](#_Toc47560786)

[**2.** **ANTECEDENTES DEL PROYECTO** 5](#_Toc47560787)

[**2.1.**  **ORGANIGRAMA** 6](#_Toc47560788)

[**2.2.** **MISION** 6](#_Toc47560789)

[**2.3.** **VISION** 6](#_Toc47560790)

[**3.** **FORMULACION DEL PROBLEMA** 6](#_Toc47560791)

[**3.1.** **DESCRIPCION DE SITUACION PROBLEMICA** 6](#_Toc47560792)

[**3.2.** **PLANTEAMIENTO DEL PROBLEMA** 7](#_Toc47560793)

[**4.** **PLANTEAMIENTO DE OBJETIVOS** 7](#_Toc47560794)

[**4.1.** **OBJETIVOS GENERALES** 7](#_Toc47560795)

[**4.2.** **OBJETIVOS ESPECIFICOS** 7](#_Toc47560796)

[**5.** **DEFINICION DEL AMBITO DEL SOFTWARE** 8](#_Toc47560797)

[**5.1.** **PRINCIPALES FUNCIONES Y RENDIMIENTO** 8](#_Toc47560798)

[**5.1.1.** **REQUERIMIENTOS FUNCIONALES** 8](#_Toc47560799)

[**5.1.2.** **REQUERIMIENTOS NO FUNCIONALES** 8](#_Toc47560800)

[**5.2.** **UNIVERSO DE INFORMACION** 9](#_Toc47560801)

[**5.3.** **CONTEXTO** 9](#_Toc47560802)

[**6.** **ORGANIZACIÓN DE RECURSOS** 9](#_Toc47560803)

[**6.1.** **ORGANIZACIÓN DEL PERSONAL** 9](#_Toc47560804)

[**6.1.1.** **PARTICIPANTES** 10](#_Toc47560805)

[**6.1.2.** **JEFE DE EQUIPO** 10](#_Toc47560806)

[**6.1.3.** **EQUIPO DE SOFTWARE** 10](#_Toc47560807)

[**6.1.4.** **COORDINACION Y COMUNICACIÓN** 11](#_Toc47560808)

[**6.2.** **PLANIFICACION** 11](#_Toc47560809)

[**6.2.1.** **DIAGRAMA DE GANTT** 11](#_Toc47560810)

[**6.3.** **RECURSOS DE HARDWARE Y SOFTWARE** 15](#_Toc47560811)

[**7.** **PROCESO DE DESARROLLO** 15](#_Toc47560812)

[**7.1.** **ICONIX** 15](#_Toc47560813)

[**7.1.1.** **PROTOTIPO DE INTERFAZ DE USUARIO** 17](#_Toc47560814)

[**7.1.1.1.**  **LOGIN** 17](#_Toc47560815)

[**7.1.1.2.**  **PESTAÑA PRINCIPAL** 17](#_Toc47560816)

[**7.1.1.3.**  **PESTAÑA OPERADOR** 18](#_Toc47560817)

[**7.1.1.4.**  **PESTAÑA VENTAS** 18](#_Toc47560818)

[**7.1.1.5.**  **PESTAÑA RESERVAS** 18](#_Toc47560819)

[**7.1.1.6.**  **PESTAÑA USUARIO** 18](#_Toc47560820)

[**7.1.1.7.**  **PESTAÑA TECNICOS** 19](#_Toc47560821)

[**7.1.1.8.**  **PESTAÑA PRODUCTOS** 19](#_Toc47560822)

[**7.1.1.9.**  **PESTAÑA SERVICIOS** 19](#_Toc47560823)

[**7.1.2.** **MODELOS DE CASOS DE USO DEL SISTEMA** 20](#_Toc47560824)

[**7.1.3.** **ESPECIFICACION DE ACTORES DEL SISTEMA** 20](#_Toc47560825)

[**7.1.4.** **ESPECIFICACION DE CASOS DE USO DEL SISTEMA** 21](#_Toc47560826)

[**7.1.5.** **DIAGRAMA DE SECUENCIAS** 29](#_Toc47560827)

[**7.1.5.1.** **AUTENTICACION USUARIO** 29](#_Toc47560828)

[**7.1.5.2.** **EDITAR CONTRASEÑA** 30](#_Toc47560829)

[**7.1.5.3.**  **GESTIONAR RAMPAS** 30](#_Toc47560830)

[**8.** **ESTIMACION DEL PROYECTO DE SOFTWARE** 32](#_Toc47560831)

[**8.1.** **ESFUERZO** 36](#_Toc47560832)

[**8.2.** **TIEMPO** 37](#_Toc47560833)

[**8.3.** **COSTO** 37](#_Toc47560834)

[**9.** **GESTION DE RIESGOS** 38](#_Toc47560835)

[**9.1.** **LISTA DE CHQUEO DE RIESGOS** 38](#_Toc47560836)

[**9.1.1.** **RIESGOS DEL PROCESO** 38](#_Toc47560837)

[**9.1.2.** **TAMAÑO DEL PERSONAL Y RIESGOS DE EXPERIENCIA** 38](#_Toc47560838)

[**9.1.3.** **RIESGOS TECNOLOGICOS** 38](#_Toc47560839)

[**9.1.4.** **RIESGOS DEL ENTORNO DE DESARROLLO** 39](#_Toc47560840)

[**9.1.5.** **RIESGOS DEL IMPACTO DEL NEGOCIO** 39](#_Toc47560841)

[**9.1.6.** **RIESGOS RELACIONADOS CON EL CLIENTE** 39](#_Toc47560842)

[**9.2.** **TABLA DE EVALUACION DE RIESGOS** 40](#_Toc47560843)

[**9.2.1.** **TAMAÑO DEL PERSONAL Y RIESGOS DE EXPERIENCIA** 40](#_Toc47560844)

[**9.2.2.** **RIESGOS DEL PROCESO** 40](#_Toc47560845)

[**9.2.3.** **RIESGOS TECNOLOGICOS** 40](#_Toc47560846)

[**9.2.4.** **RIESGOS DEL ENTORNO DE DESARROLLO** 41](#_Toc47560847)

[**9.2.5.** **RIESGOS DEL IMPACTO DEL NEGOCIO** 41](#_Toc47560848)

[**9.2.6.** **RIESGOS RELACIONADOS CON EL CLIENTE** 41](#_Toc47560849)

[**9.3.** **PLAN RMMM** 42](#_Toc47560850)

[**9.3.1.** **RIESGO: POSIBLES CAMBIOS EN LOS REQUISITOS DEL PROYECTO** 42](#_Toc47560851)

[**9.3.2.** **RIESGO: UN INTEGRANTE NO COMPLETA SU PARTE DEL PROYECTO EN FECHA ESTABLECIDA** 42](#_Toc47560852)

[**9.3.3.** **RIESGO: QUE UN MIEMBRO ABANDONE** 43](#_Toc47560853)

[**9.3.4.** **RIESGO: LA POCA EXISTENCIA DE EXPERTOS A LOS CUALES SOLICITAR AYUDA ACERCA DE LAS HERRAMIENTAS** 43](#_Toc47560854)

[**9.3.5.** **RIESGO: NO EXISTA AYUDA EN INTERNET Y DOCUMENTACIÓN DISPONIBLE** 43](#_Toc47560855)

[**9.3.6.** **RIESGO: LOS INTEGRANTES DEL EQUIPO DE DESARROLLO TIENEN FORMACIÓN ADECUADA** 44](#_Toc47560856)

[**9.3.7.** **RIESGO: NO SE CUMPLE CON LAS CARACTERÍSTICAS ESPERAS POR EL CLIENTE** 44](#_Toc47560857)

[**9.3.8.** **RIESGO: SE PRESENTA ERRORES EN LA EJECUCIÓN DEL PROBLEMA** 44](#_Toc47560858)

[**9.3.9.** **RIESGO: EL CLIENTE ESTA INCONFORME CON EL SOFTWARE ENTREGADO** 45](#_Toc47560859)

[**10.** **ASEGURAMIENTO DE CALIDAD** 45](#_Toc47560860)

[**10.1.** **MARCO DE TRABAJO DE PLAN** 45](#_Toc47560861)

[**10.2.** **OBJETIVO GENERAL DE LA CALIDAD DEL PROYECTO** 46](#_Toc47560862)

[**11.** **PLANIFICACION DE LA CALIDAD** 46](#_Toc47560863)

[**11.1.** **IDENTIFICACION DE LOS OBJETIVOS ESPECIFICOS DE LA CALIDAD** 46](#_Toc47560864)

[**11.2.** **MANUAL DE ESTÁNDARES PARA EL PROYECTO** 47](#_Toc47560865)

[**11.2.1.** **ESTANDARES PARA LAS VENTANAS** 47](#_Toc47560866)

[**11.2.2.** **ESTANDARES PARA LOS REPORTES** 48](#_Toc47560867)

[**11.2.3.** **ESTANDARES PARA LA BASE DE DATOS** 48](#_Toc47560868)

[**11.2.4.** **ESTANDARES PARA LA CALIDAD DEL SOFTWARE** 49](#_Toc47560869)

[**11.3.** **LISTAS DE VERIFICACION DE SOFTWARE** 49](#_Toc47560870)

# **TITULO DEL PROYECTO:**

Sistema de Información para el Taller de Servicios Automotrices “La Resistencia”.

Caso de estudio: Taller de Servicios Automotrices “La Resistencia”.

1. **INTRODUCCION**

Hoy en día, todo el mundo habla de fenómenos globales, o de globalización, y de lo que se trata es de la condición de acceso irrestricto a bienes y servicios de naturaleza diversa para el consumo mundial. En consecuencia, las condiciones técnicas que exige la globalización implican una vinculación informática y tecnológica. Estar globalizado implica entonces, estar a la vanguardia o bien conocer la tecnología punta de una época determinada. Lo contrario sería rechazar la evolución significaría, en resumidas cuentas, lo que esboza la siguiente frase: dejar de tener relevancia, caer en el olvido, dejar de existir.

Es por esto que se ha producido un aumento de la competitividad entre los mercados y cada vez más empresas intentan buscar mejoras en su actividad. Por consiguiente, se intenta ofrecer mejores servicios con el fin de fidelizar clientes.

Es por esta razón que el presente trabajo implementa un sistema de información para realizar el registro detallado de los servicios que ejecuta el Taller Automotriz “La Resistencia” en su día a día, reduciendo el tiempo de atención y evitando todo tipo de retrasos en la entregas.

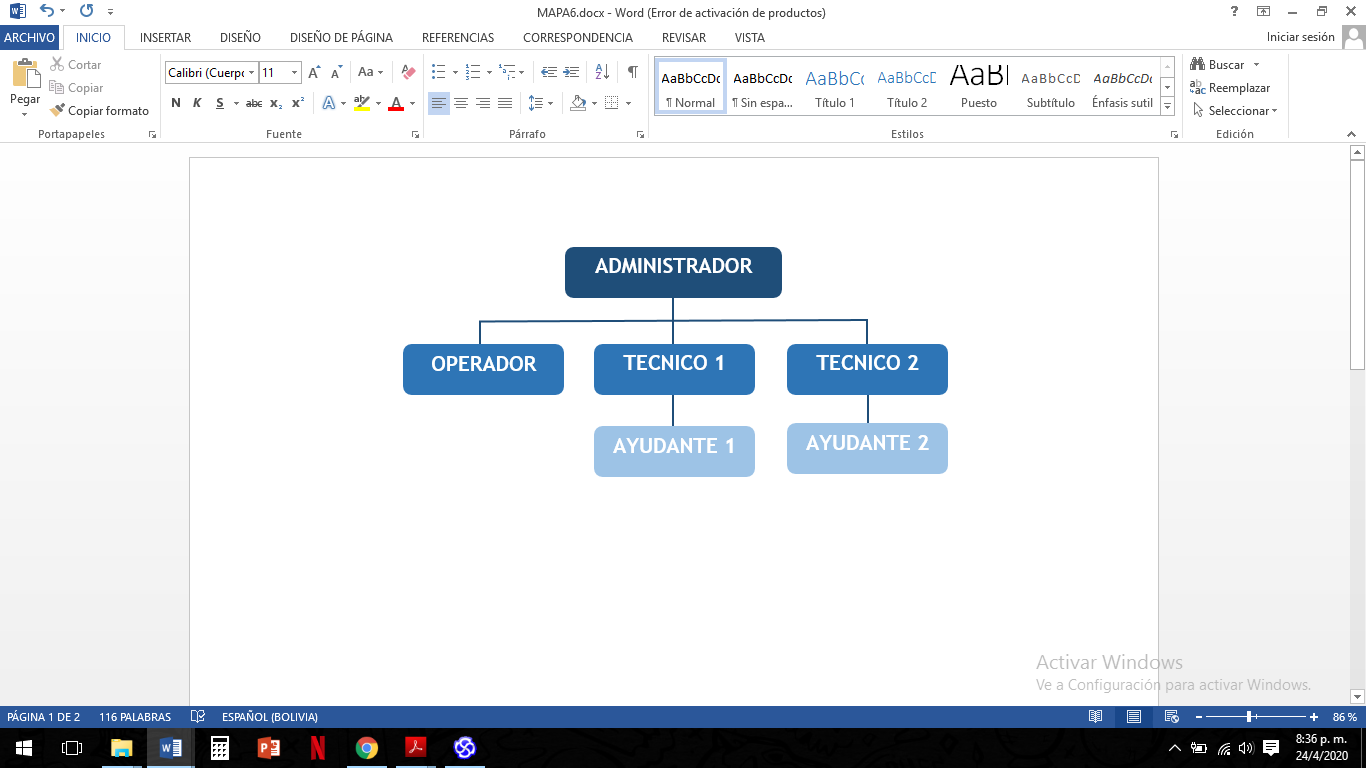
De esta manera se espera facilitar la tarea de los empleados y que los clientes del Taller Automotriz “La Resistencia” reciban una atención satisfactoria.

1. **ANTECEDENTES DEL PROYECTO**

El Taller de Servicios Automotrices “La Resistencia” es una pequeña empresa especializada en la reparación de chapa, pintura y mecánica. La evolución positiva de esta empresa, a lo largo de los años, ha ido incrementando su número de clientes y proveedores. Por ello, desean invertir en un sistema de información, trayendo consigo las garantías de la satisfacción de sus clientes, a la vez facilitar la tarea de sus empleados y tener un mayor control en su inventario.

El registro de reservas, fechas de recepción, reparación y entrega de los automóviles se realiza en cuadernos y recibos, que impiden un control eficiente de ganancia y generan retrasos en las entregas, causando la molestia de los clientes.

Es por ello que se contara con una interfaz controlada por la administración en la cual se podrá realizar todas las tareas anteriormente descritas, facilitando el trabajo de los empleados y otorgando un mayor control de los servicios, inventario y ganancias.

* 1.  **ORGANIGRAMA**
  2. **MISION**

Proporcionar un buen servicio de mantenimiento automotriz en general, conservando los autos de nuestros clientes en un estado de operación eficiente y seguro, satisfaciendo sus necesidades y expectativas en el mejor tiempo posible. A sabiendas de que usted y su vehículo son lo más importante.

* 1. **VISION**

Llegar a ser el taller líder y confiable a nivel zona, en los próximos 3 años, una vez cumplido ese objetivo mantenernos como los mejores en el mercado, ofreciendo el servicio para automóviles modernos y unidades diésel, siendo reconocidos por la calidad en nuestros servicios, honestidad y precios justos.

1. **FORMULACION DEL PROBLEMA**
   1. **DESCRIPCION DE SITUACION PROBLEMICA**

Desde que el Taller “La Resistencia” empezó a ser reconocido y recibir más clientela, han presentado dificultades en cuanto al registro de reservaciones y fechas de recepción y entrega de los automóviles, de igual manera también ocasiona cierto desbalance de trabajo para los empleados y claramente afecta a las necesidades y satisfacción del cliente.

|  |  |
| --- | --- |
| **CAUSA** | **EFECTO** |
| Mal registro de fechas en el cuaderno del taller | Retrasos en la entrega de automóviles y componentes |
| Extravió del cuaderno de registros | Informes mal elaborados |
| Retrasos en las entregas | Clientes disconformes y/o perdidas de clientes |
| Reservas mal situadas | Exceso de trabajo para los empleados |

* 1. **PLANTEAMIENTO DEL PROBLEMA**

¿Cómo agilizar el proceso de registros del Taller de Servicios Automotrices “La Resistencia”, evitando pérdidas y retrasos, satisfaciendo las necesidades de los clientes?

1. **PLANTEAMIENTO DE OBJETIVOS**
   1. **OBJETIVOS GENERALES**

Desarrollar un sistema de información para el registro de servicios del Taller de Servicios Automotrices “La Resistencia”.

* 1. **OBJETIVOS ESPECIFICOS**
* Recolectar los diferentes requerimientos que necesita el Taller “La Resistencia” para estar al tanto de sus registros.
* Diseñar la estructura lógica del sistema de información para que el registro funcione correctamente.
* Crear una Base de Datos funcional para el Taller “La Resistencia” donde se podrá manipular y mantener los datos correctamente actualizados.
* Implementar la estructura ya diseñada para obtener los registros de servicios realizados en el taller.
* Realizar pruebas sobre el sistema de información para corregir y evitar inconvenientes.

1. **DEFINICION DEL AMBITO DEL SOFTWARE**
   1. **PRINCIPALES FUNCIONES Y RENDIMIENTO**
      1. **REQUERIMIENTOS FUNCIONALES**

El Sistema tendrá los siguientes requisitos funcionales:

* El sistema de información deberá ser accesible a través de un usuario y contraseña.
* El sistema de información deberá registrar cada servicio que realice el taller.
* El sistema de información permitirá la reservación, modificación y cancelación de fechas.
* El sistema debe generar reportes semanales.
* El sistema de información permitirá gestionar los productos del inventario del taller.
* El sistema de información permitirá gestionar los datos de los empleados del taller.
* El sistema de información permitirá gestionar los datos de todos los clientes del taller.
  + 1. **REQUERIMIENTOS NO FUNCIONALES**

El sistema tendrá los siguientes requisitos no funcionales**:**

* Confiable en cuanto al manejo de la información.
* La Interfaz del sistema debe ser simple, intuitiva, fácil de usar.
* El sistema de información debe ser escalable, para así poder adaptarse y mejorar a través del tiempo.
* El Sistema debe ser amigable y proporcionar mensajes de error que sean informativos y orientados al usuario final para facilitar su manejo.
* El software debe ser confiable al momento de ser usado por el usuario.
* El software debe ser capaz de controlar los tipos de entradas de datos.
* El software debe tener iconos de acceso en el escritorio.
* El software debe tener una paleta de colores agradable a la vista del usuario.
* El software debe tener tamaños de fuente acorde a la relevancia de la información mostrada.
  1. **UNIVERSO DE INFORMACION**

El sistema de información tendrá lo siguiente:

* **cliente** {id, nombre, ap\_paterno, ap\_materno, ci, telefono. direccion}
* **det\_producto**{id, id\_producto, id\_factura, cantidad, precio\_unitario, subtotal}
* **det\_servicio**{id, id\_factura, id\_servicio, id\_tecnico, id\_rampa, cantidad, precio\_unitario, subtotal}
* **factura**{id, nro, id\_cliente, id\_usuario, fecha, total}
* **producto**{id, nombre, precio\_compra, precio\_venta, tipo, stock, marca, modelo}
* **rampa**{id, nombre, tipo}
* **reservación**{id, id\_cliente, id\_rampa, horario. fecha, observacion}
* **servicio**{id, nombre, tipo, precio}
* **técnico**{id, nombre, ap\_paterno, ap\_materno, ci, telefono, direccion, especialidad, activo}
* **usuario**{id, nombre, ap\_paterno, ap\_materno, ci, telefono, direccion, tipo, cuenta, pass, activo}
  1. **CONTEXTO**

El sistema será implementado en el Taller de Servicios Automotrices “La Resistencia” ubicado en la Av. Circunvalación entre la intersección de vías férreas (sud este), gestionado por el administrador del taller, cumpliendo los requisitos de hardware y software para su correcto funcionamiento.

1. **ORGANIZACIÓN DE RECURSOS**
   1. **ORGANIZACIÓN DEL PERSONAL**

El factor humano será considerado lo más importante en el desarrollo del proyecto, cabe destacar que para alcanzar el éxito no se debe a las herramientas que utilizan, sino se debe a las personas y al trabajo en equipo, la importancia que tiene este factor es crucial para el desarrollo del proyecto, donde por medio de selección se incorporó al grupo teniendo en cuenta las capacidades, habilidades y destrezas de cada integrante.

* + 1. **PARTICIPANTES**

Estos se clasifican en:

* **Gestores técnicos del proyecto:** Encargados de planificar, organizar, y controlar el proceso de desarrollo del software.
* **Administrador:** Especificara los requerimientos.
* **Operador:** Se encarga de interactuar con el software.
  + 1. **JEFE DE EQUIPO**

Se contará con un solo jefe de equipo el cual será el gestor del proyecto, el cual deberá tener las siguientes facultades:

* Promover e incentivar las iniciativas y logros del equipo.
* Hacer conocer al equipo la importancia de la calidad.
* Tendrá la confianza para asumir el control del proyecto.
* Permitirá a todo el equipo aportar ideas para mejorar la calidad del proyecto.
  + 1. **EQUIPO DE SOFTWARE**

La organización del equipo será de nivel descentralizado democrático, ya que en este se especifica lo siguiente:

“Este equipo no tiene un jefe permanente y se nombran coordinadores a corto plazo, las decisiones se hacen por consenso del grupo. La comunicación entre los miembros del equipo es horizontal.”

En este nivel de organización es más eficiente a las demás por varios aspectos como:

* El número de integrantes es reducido (tres), la toma de decisiones será por consenso.
* La comunicación es horizontal, entonces se evoca más hacia los procesos y no a funciones.
* No es jerárquico (comunicación vertical).
* Todos los integrantes tendrán el conocimiento de todo el proceso del desarrollo del software.
  + 1. **COORDINACION Y COMUNICACIÓN**

Existen varios mecanismos de comunicación para coordinar el equipo de trabajo, se enfocará en tres acápites fundamentales para el éxito del proyecto:

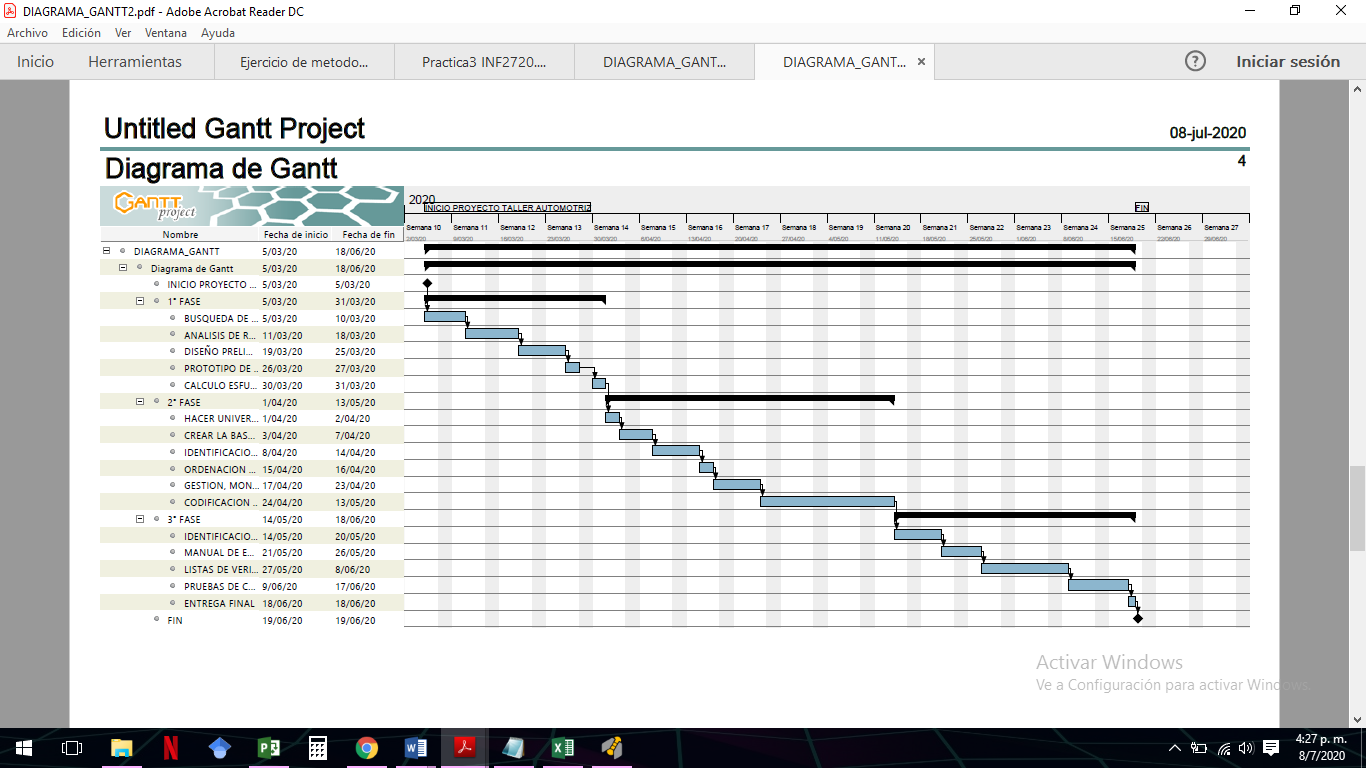
* **Comunicación formal:** Se lleva a cabo por escrito y reuniones. Incluye documentación e informes de seguimiento.
* **Comunicación informal:** Sera más personal, reuniones de grupo para la divulgación de información en cuanto al proyecto.
* **Comunicación electrónica:** Con el constante avance tecnológico como hacer de lado los nuevos medios para la comunicación entre el equipo.

Los medios de comunicación y plataformas a utilizar serán: GitHub, Correo Electrónico, Zoom y Whatsapp.

|  |  |  |  |
| --- | --- | --- | --- |
| **NOMBRE** | **CARGO** | **CORREO** | **USUARIO DE GIT** |
| Néstor Benítez | Tester | dban095@hotmail.com | dban73 |
| Lexer Lazcano | Desarrollador | lexerlb@gmail.com | lexerlb |
| Jenny Ugarte | Analista | jen.ut.x@gmail.com | Jenny939 |

* 1. **PLANIFICACION**
     1. **DIAGRAMA DE GANTT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modo de tarea | Nombre de tarea | Duración | Comienzo | Fin |
| **Programada automáticamente** | **Diagrama de Gantt** | **76 días** | **jue 5/3/20** | **jue 18/6/20** |
| Programada automáticamente | INICIO PROYECTO TALLER AUTOMOTRIZ | 0 días | jue 5/3/20 | jue 5/3/20 |
| **Programada automáticamente** | **1° FASE** | **19 días** | **jue 5/3/20** | **mar 31/3/20** |
| Programada automáticamente | BUSQUEDA DE ANTECEDENTES | 4 días | jue 5/3/20 | mar 10/3/20 |
| Programada automáticamente | ANALISIS DE REQUISITOS | 6 días | mié 11/3/20 | mié 18/3/20 |
| Programada automáticamente | DISEÑO PRELIMINAR | 5 días | jue 19/3/20 | mié 25/3/20 |
| Programada automáticamente | PROTOTIPO DE INTERFAZ DE USUARIO | 2 días | jue 26/3/20 | vie 27/3/20 |
| Programada automáticamente | CALCULO ESFUERZO TOTAL | 2 días | lun 30/3/20 | mar 31/3/20 |
| **Programada automáticamente** | **2° FASE** | **31 días** | **mié 1/4/20** | **mié 13/5/20** |
| Programada automáticamente | HACER UNIVERSO DE DISCURSO | 2 días | mié 1/4/20 | jue 2/4/20 |
| Programada automáticamente | CREAR LA BASE DE DATOS | 3 días | vie 3/4/20 | mar 7/4/20 |
| Programada automáticamente | IDENTIFICACION DE RIESGOS | 5 días | mié 8/4/20 | mar 14/4/20 |
| Programada automáticamente | ORDENACION Y FILTRADO DE RIESGOS | 2 días | mié 15/4/20 | jue 16/4/20 |
| Programada automáticamente | GESTION, MONITORIZACION Y MITIGACION DE RIESGOS | 5 días | vie 17/4/20 | jue 23/4/20 |
| Programada automáticamente | CODIFICACION DEL SOFTWARE | 14 días | vie 24/4/20 | mié 13/5/20 |
| **Programada automáticamente** | **3° FASE** | **26 días** | **jue 14/5/20** | **jue 18/6/20** |
| Programada automáticamente | IDENTIFICACION DE LOS OBJETIVOS ESPECIFICOS DE CALIDAD | 5 días | jue 14/5/20 | mié 20/5/20 |
| Programada automáticamente | MANUAL DE ESTANDARES PARA EL PROYECTO | 4 días | jue 21/5/20 | mar 26/5/20 |
| Programada automáticamente | LISTAS DE VERIFICACION DE SOFTWARE | 9 días | mié 27/5/20 | lun 8/6/20 |
| Programada automáticamente | PRUEBAS DE CAJA BLANCA Y NEGRA | 7 días | mar 9/6/20 | mié 17/6/20 |
| Programada automáticamente | ENTREGA FINAL | 1 día | jue 18/6/20 | jue 18/6/20 |
| Programada automáticamente | FIN | 0 días | jue18/6/20 | jue 18/6/20 |



Comienzo  
jue 5/3/20

Fin  
jue 18/6/20

9 mar '20

23 mar '20

6 abr '20

20 abr '20

4 may '20

18 may '20

1 jun '20

**1° FASE**  
**jue 5/3/20 - mar 31/3/20**

**2° FASE**  
**mié 1/4/20 - mié 13/5/20**

**3° FASE**  
**jue 14/5/20 - jue 18/6/20**

**BUSQUEDA DE ANTECEDENTES**  
**jue 5/3/20 - mar 10/3/20**

**ANALISIS DE REQUISITOS**  
**mié 11/3/20 - mié 18/3/20**

**DISEÑO PRELIMINAR**  
**jue 19/3/20 - mié 25/3/20**

**PROTOTIPO DE INTERFAZ DE USUARIO**  
**jue 26/3/20 - vie 27/3/20**

**CALCULO ESFUERZO TOTAL**  
**lun 30/3/20 - mar 31/3/20**

**CREAR LA BASE DE DATOS**  
**vie 3/4/20 - mar 7/4/20**

**HACER UNIVERSO DE DISCURSO**  
**mié 1/4/20 - jue 2/4/20**

**IDENTIFICACION DE RIESGOS**  
**mié 8/4/20 - mar 14/4/20**

**ORDENACION Y FILTRADO DE RIESGOS**  
**mié 15/4/20 - jue 16/4/20**

**GESTION, MONITORIZACION Y MITIGACION DE RIESGOS**  
**vie 17/4/20 - jue 23/4/20**

**CODIFICACION DEL SOFTWARE**  
**vie 24/4/20 - mié 13/5/20**

**IDENTIFICACION DE LOS OBJETIVOS ESPECIFICOS DE CALIDAD**  
**jue 14/5/20 - mié 20/5/20**

**MANUAL DE ESTANDARES PARA EL PROYECTO**  
**jue 21/5/20 - mar 26/5/20**

**LISTAS DE VERIFICACION DE SOFTWARE**  
**mié 27/5/20 - lun 8/6/20**

**PRUEBAS DE CAJA BLANCA Y NEGRA**  
**mar 9/6/20 - mié 17/6/20**

**ENTREGA FINAL**  
**jue 18/6/20**

**INICIO PROYECTO TALLER AUTOMOTRIZ  
jue 5/3/20**

**FIN  
jue 18/6/20**

* 1. **RECURSOS DE HARDWARE Y SOFTWARE**

Requisitos mínimos para ambos

**HARDWARE:**

* Procesador de 3ra generación como mínimo.
* Memoria de 2Gb de RAM.

**SOFTWARE:**

Para el correcto funcionamiento del sistema se requiere la instalación de los siguientes recursos en cuanto a software:

* Windows 7, 8, 10.
* SQL Server 2014.
* Framework 3.5 o superior.

1. **PROCESO DE DESARROLLO**
   1. **ICONIX**

ICONIX es una metodología pesada-ligera de Desarrollo del Software que se halla a medio camino entre RUP (Rational Unified Process) y XP (eXtreme Programming), es una metodología simplificada en comparación a otras más tradicionales, la cual unifica un conjunto de métodos de orientación a objetos con el objetivo de tener un control estricto sobre todo el ciclo de vida del producto a realizar, cuenta con una secuencia de pasos que se deben seguir y determina claramente las actividades a desarrollar en cada etapa del ciclo de vida del proyecto que la utilice.

**FASE 1: ANÁLISIS DE REQUISITOS**

Identificar en el mundo real, los objetos y todas las relaciones de agregación y generalización entre ellos. Se deben analizar todos los requisitos formaran parte del sistema y con estos construir el diagrama de clases, que representa las agrupaciones funcionales que estructuraran el sistema en desarrollo.

Para esta fase se utilizan 3 herramientas:

* Modelo de Dominio: esto se refiere a identificar objetos y cosas del mundo real que intervienen con nuestro sistema. (Estático)
* Modelo de Casos de Uso: describe las acciones o el comportamiento que un usuario realiza dentro del sistema. Comprende de actores, casos de uso y el sistema.
* Prototipo de Interfaz de Usuario: implica la creación de un modelo o modelos operativos del trabajo de un sistema, en el que analistas y clientes deben estar de acuerdo. (Dinámico/ los usuarios se hacen participantes activos en el desarrollo)

**FASE 2: ANÁLISIS Y DISEÑO PRELIMINAR**

En esta fase a partir de cada caso de uso se obtendrán una ficha de caso de uso, (la cual no pertenece a UML), está formada por un nombre, una descripción, una precondición que debe cumplir antes de iniciarse, una pos condición que debe cumplir al terminar si termina correctamente. Se deben describir los casos de uso, como un flujo principal de acciones, pudiendo contener los flujos alternativos y los flujos de excepción. la principal sugerencia de Iconix, en esta actividad es que no se debe perder mucho tiempo con la descripción textual. Debería usarse un estilo consistente que sea adecuado al contexto del proyecto. Realizar Diagrama de Robustez: es un híbrido entre un Diagrama de Clases y un Diagrama de Actividades. Es una herramienta que nos permite capturar el Que hacer y a partir de eso él Como hacerlo. Facilita el reconocimiento de objetos y hace más sencilla la lectura del sistema. Ayuda a identificar los objetos que participan en cada caso de uso.

El diagrama de Robustez se divide en:

* Objetos fronterizos: usado por los actores para comunicarse con el sistema.
* Objetos entidad: son objetos del modelo del dominio.
* Objetos de Control: es la unión entre la interfaz y los objetos de entidad.
* Diagrama de Clases: describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos

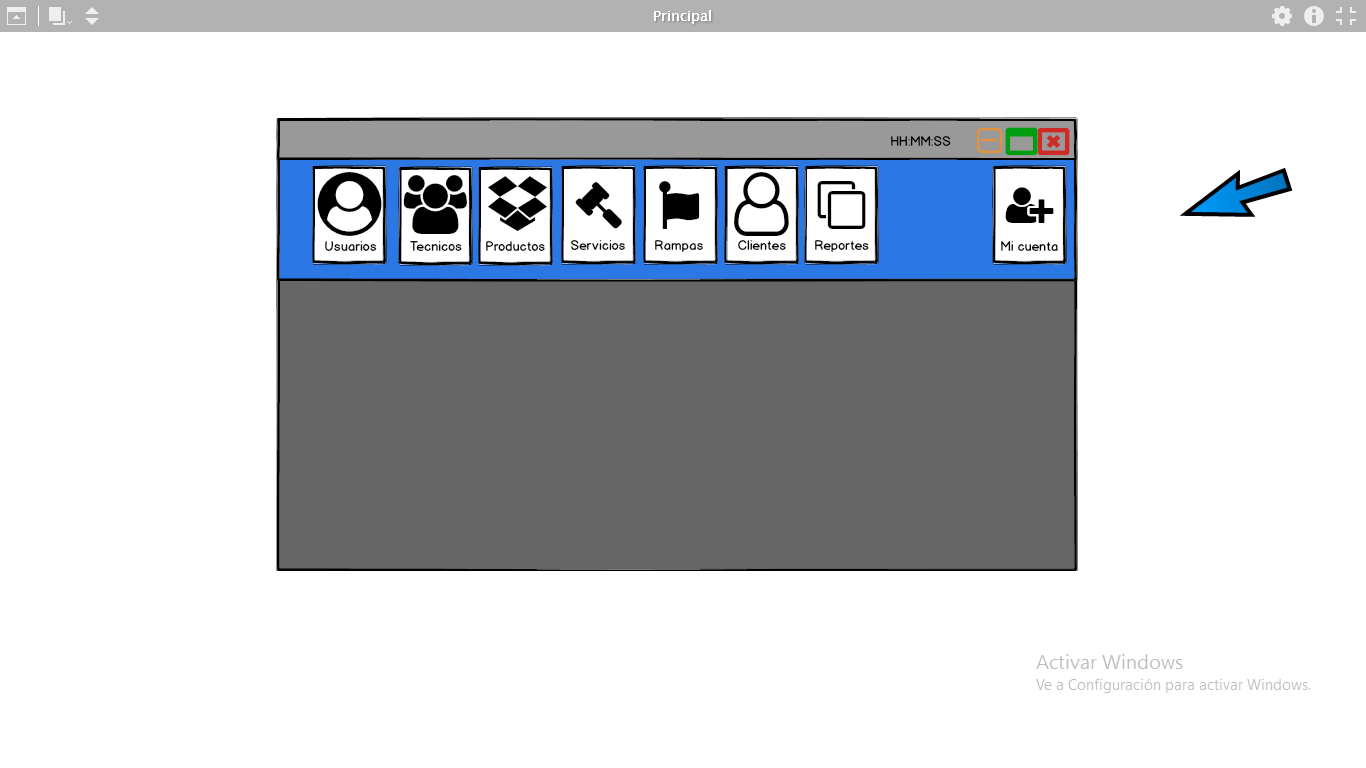
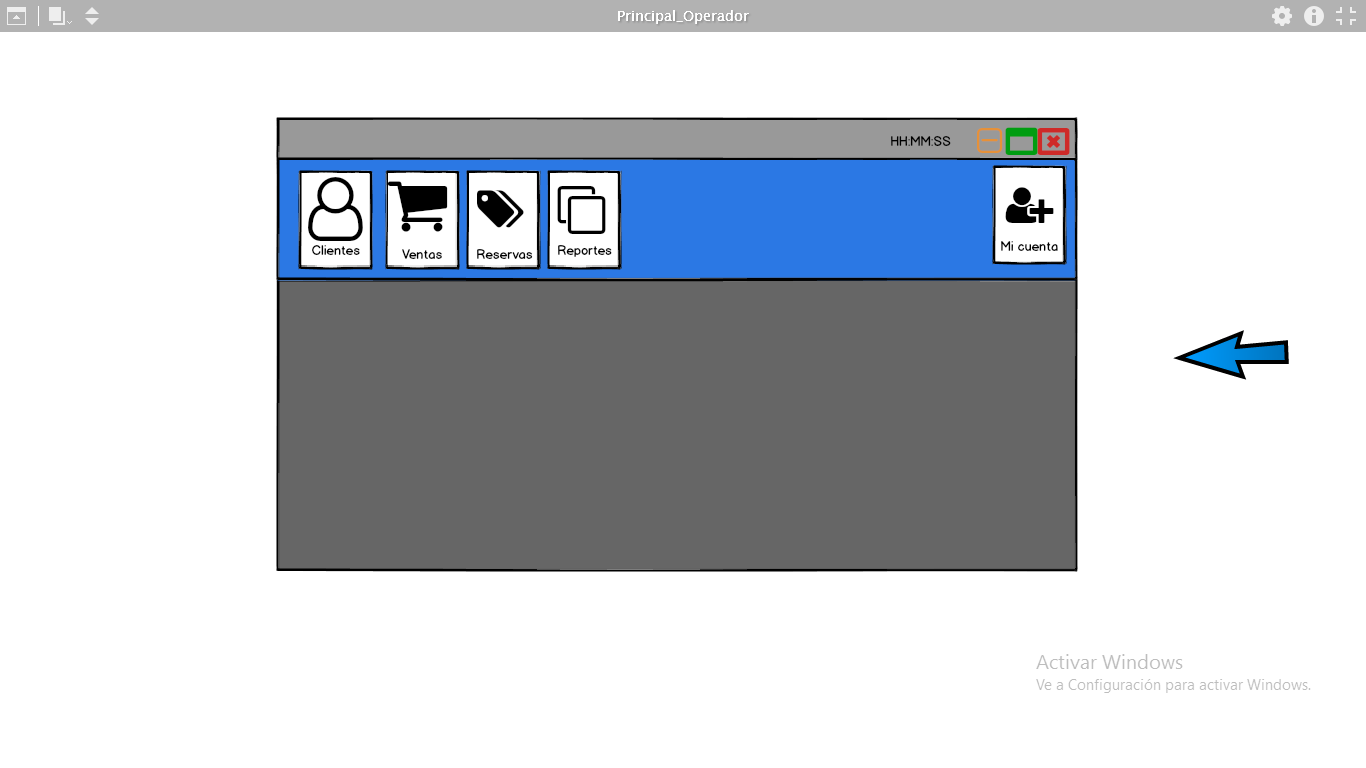
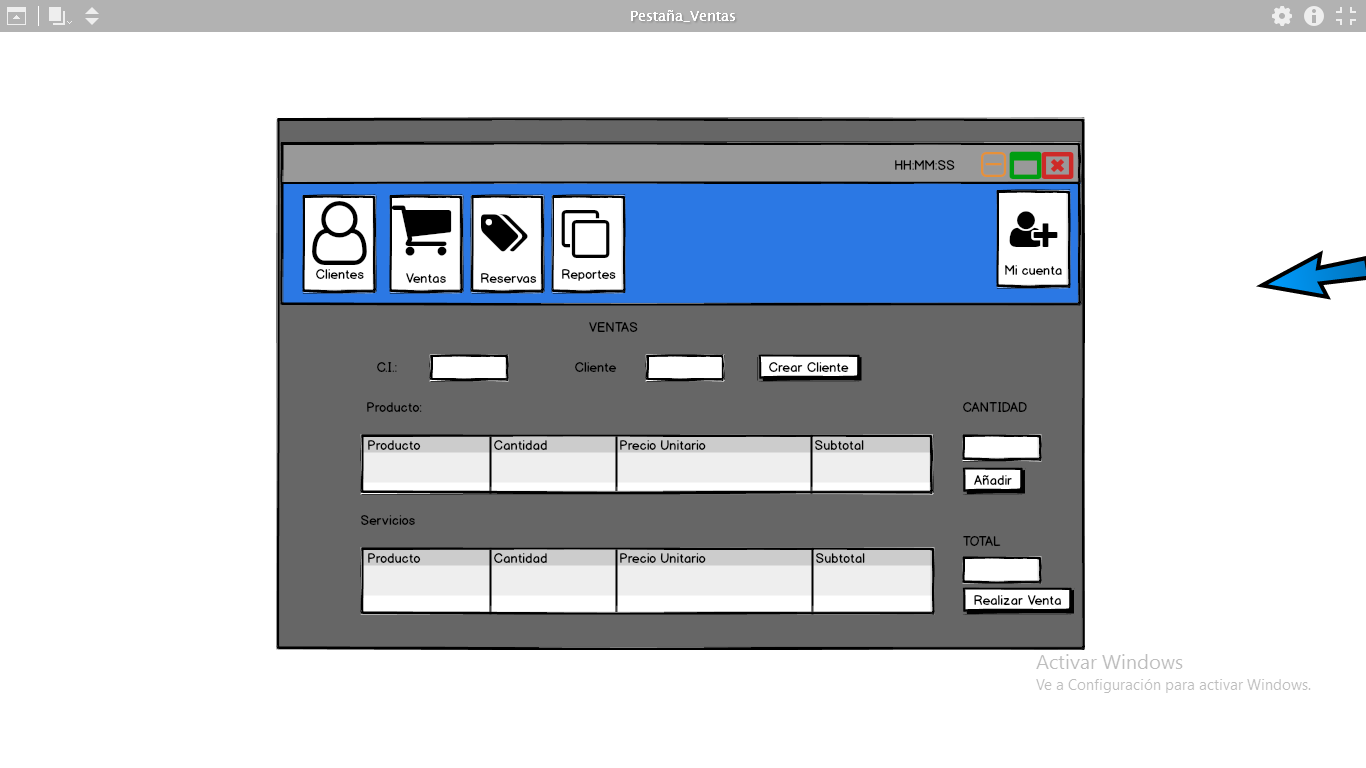
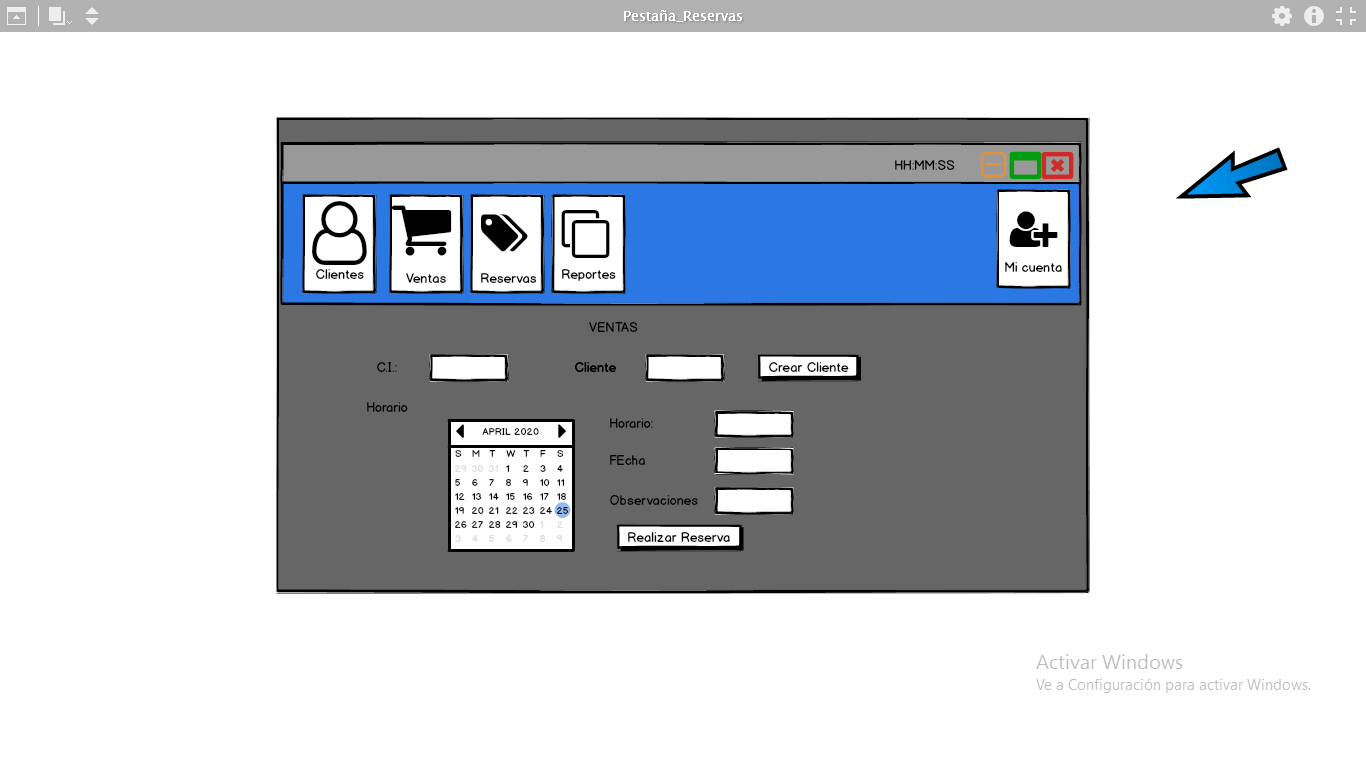
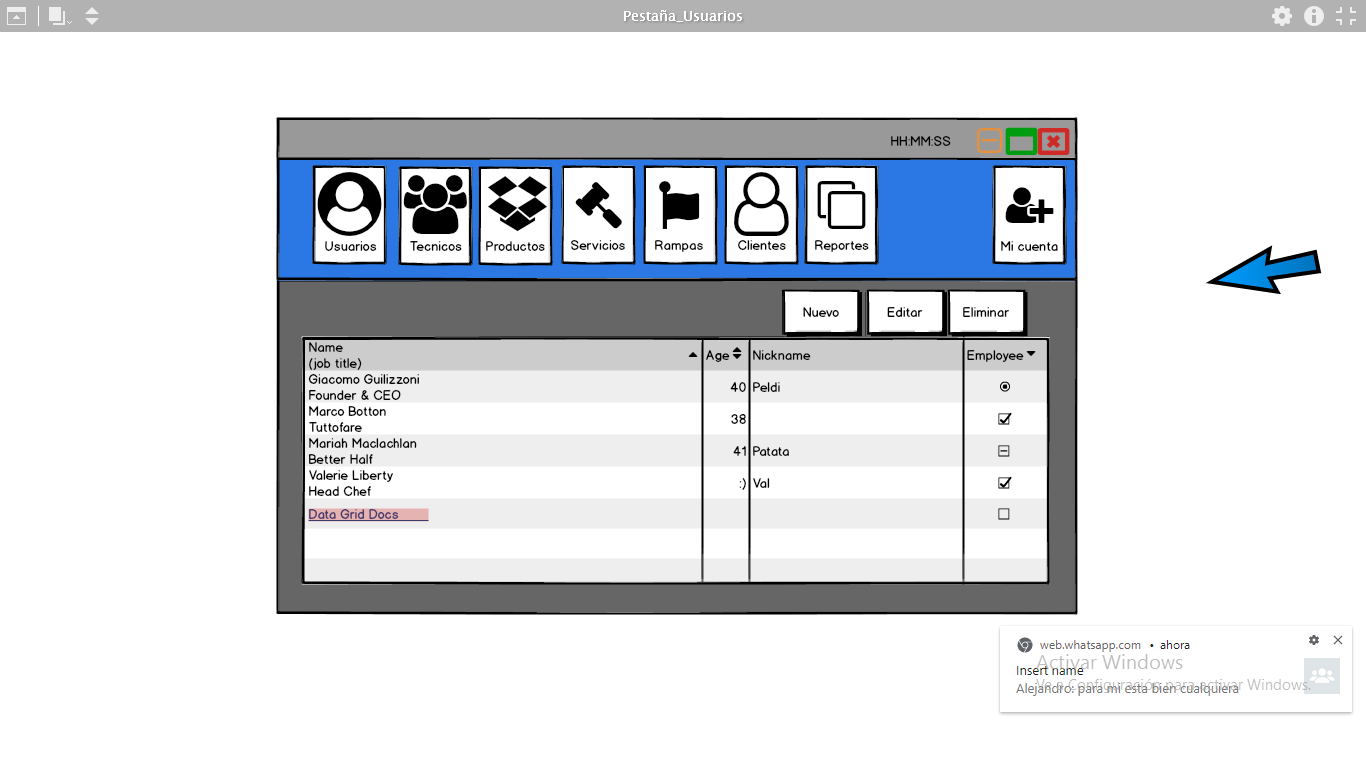
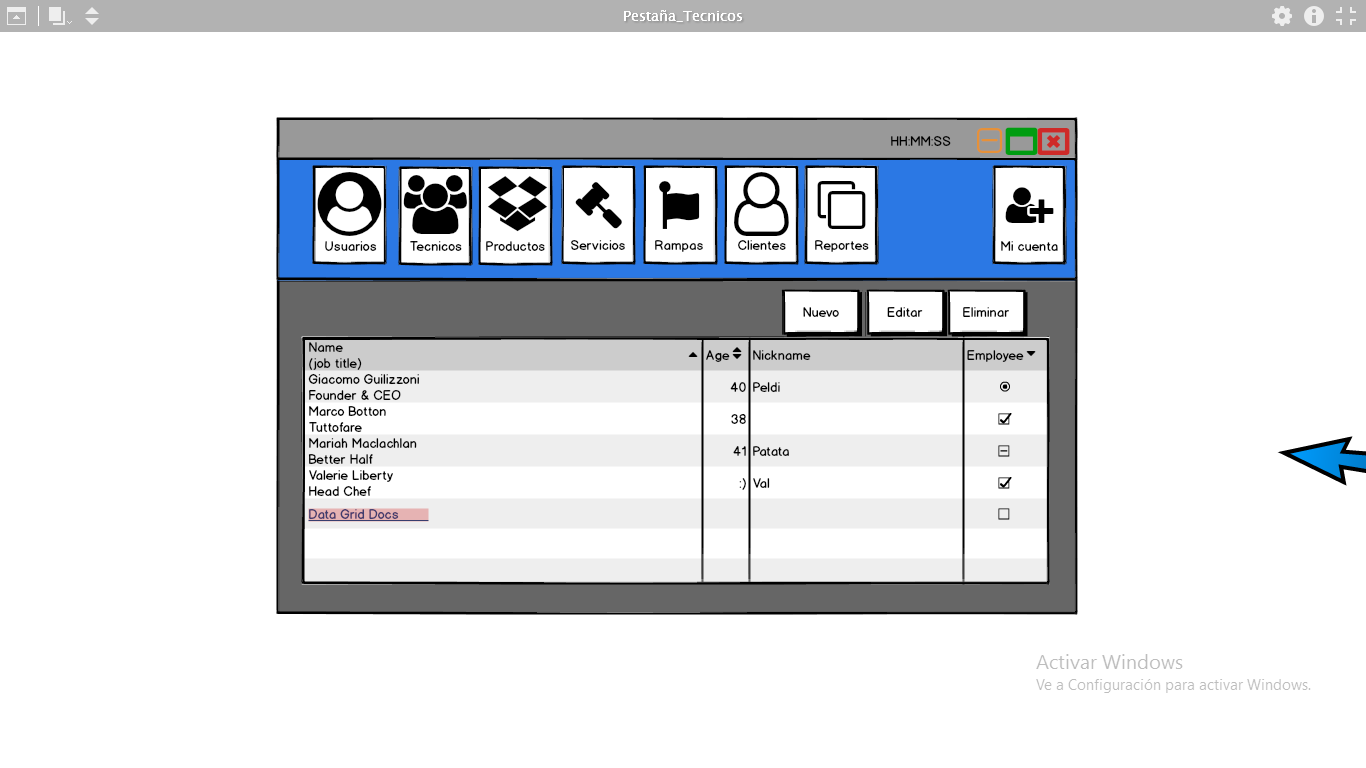
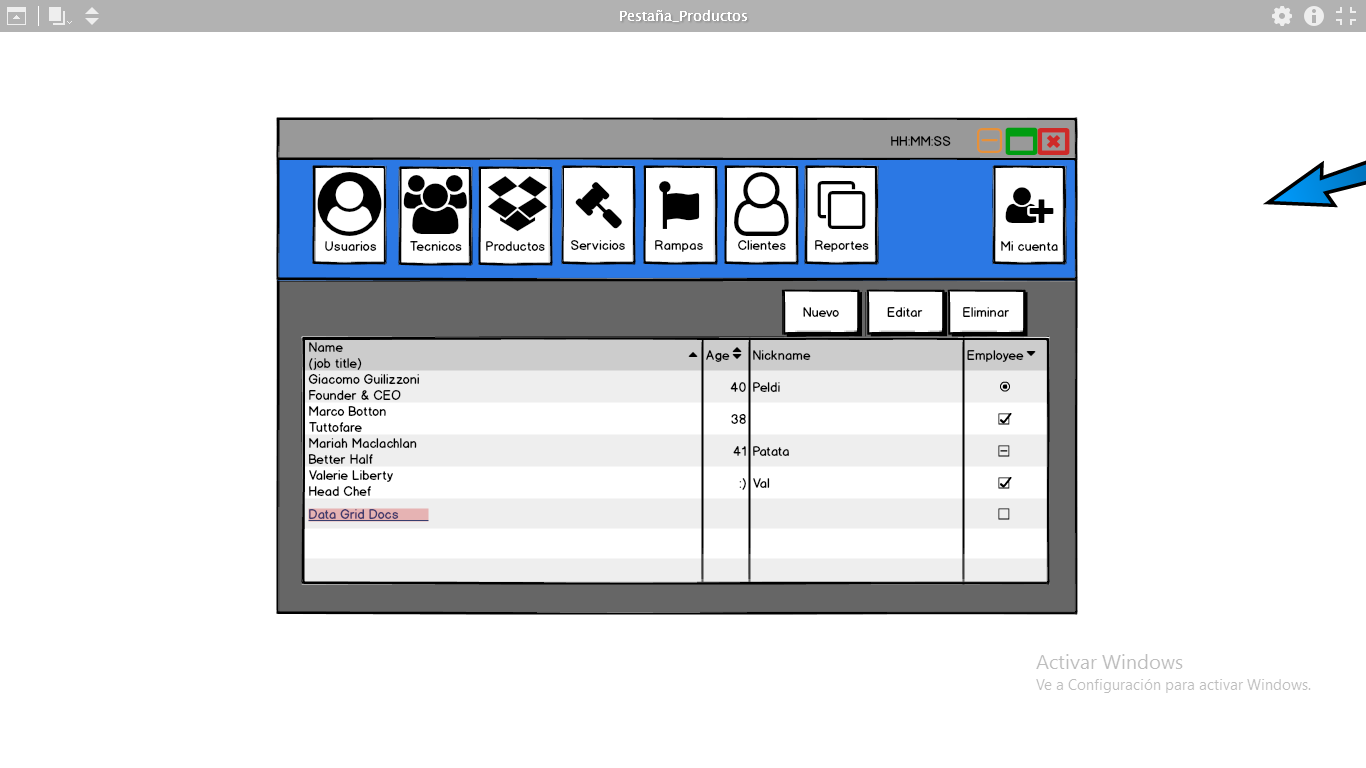
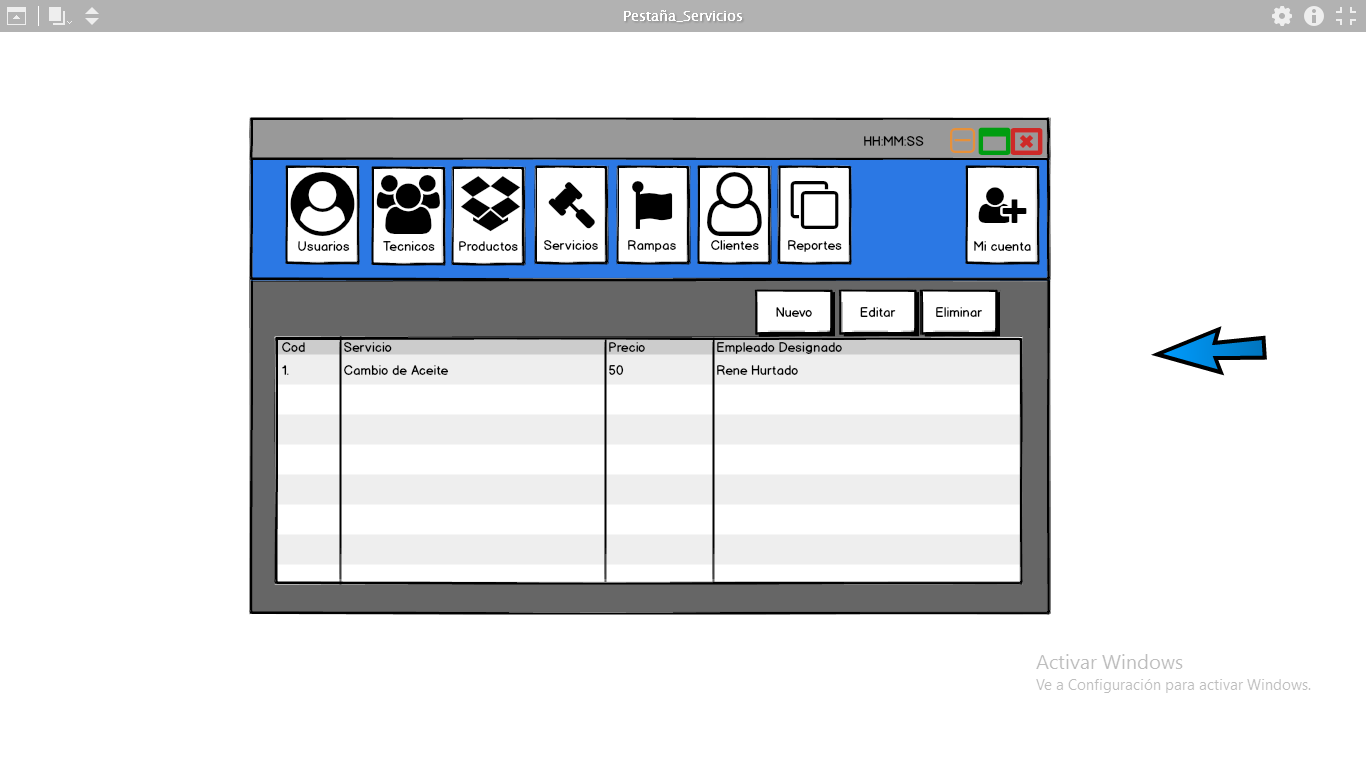
**FASE 3: DISEÑO**

En esta fase se reconocen todos los elementos que forman parte de nuestro sistema. Diagramas de Secuencia: muestra los métodos que llevaran las clases de nuestro sistema. Muestra todos los cursos alternos que pueden tomar todos nuestros casos de uso. Se debe terminar el modelo estático, añadiendo los detalles del diseño en el diagrama de clases y verificar si el diseño satisface todos los requisitos identificados.

**FASE 4: IMPLEMENTACIÓN Y PRUEBAS**

En esta fase a partir del buen diseño logrado se creara el software; que posteriormente se entregara. Se debe utilizar el diagrama de componentes si fuera necesario para apoyar el desarrollo, es decir mostrar una distribución física de los elementos que componen la estructura interna del sistema. Así como escribir y generar el código.

Pero además se debe tener en cuenta factores como:

* Reusabilidad: Es la posibilidad de hacer uso de los componentes en diferentes aplicaciones.
* Extensibilidad: Consiste en modificar con facilidad el software.
* Confiabilidad: Realización de sistemas descartando las posibilidades de error.
* Realizar pruebas: Test de unidades, de casos, datos y resultados. Test de integración con los usuarios para verificar la aceptación de los resultados.
  + 1. **PROTOTIPO DE INTERFAZ DE USUARIO**
       1.  **LOGIN**
       2.  **PESTAÑA PRINCIPAL**
       3.  **PESTAÑA OPERADOR**
       4.  **PESTAÑA VENTAS**
       5.  **PESTAÑA RESERVAS**
       6.  **PESTAÑA USUARIO**
       7.  **PESTAÑA TECNICOS**
       8.  **PESTAÑA PRODUCTOS**
       9.  **PESTAÑA SERVICIOS**
    2. **MODELOS DE CASOS DE USO DEL SISTEMA**

****

* + 1. **ESPECIFICACION DE ACTORES DEL SISTEMA**

|  |  |
| --- | --- |
| **ACTORES** | **ESPECIFICACION** |
|  | El Usuario representa al Administrador y el Operador del sistema, ambos podrán ingresar al sistema y ejercer distintas acciones. |
|  | El Administrador se encarga de gestionar todo lo referente a la base de datos además de los respectivos reportes sobre los cambios en los datos del sistema. |
|  | El operador es el encargado de registrar las distintas ventas que se harán a los clientes y además gestionar las reservaciones. |

* + 1. **ESPECIFICACION DE CASOS DE USO DEL SISTEMA**

|  |  |
| --- | --- |
| **CASO DE USO** | **ESPECIFICACIÓN** |
|  | El Sistema autentifica al usuario (Administrador u Operario) para poder hacer un uso correcto del mismo. |
|  | El Usuario puede editar su contraseña en cualquier momento por razones de seguridad. |
|  | El administrador gestiona Usuarios donde: agrega, actualiza o elimina al Usuario. Los Usuarios del Sistema además de sus datos personales cuentan con una cuenta y contraseña para ingresar al sistema. |
|  | El administrador gestiona Clientes donde: agrega, actualiza o elimina a algún Cliente. |
|  | El administrador gestiona Rampas donde: agrega, actualiza o elimina alguna rampa. Existen Rampas de Limpieza y de Mantenimiento. |
|  | El administrador gestiona Técnicos donde: agrega, actualiza o elimina Técnicos, además se especifica si el Técnico está activo o no. |
|  | El administrador gestiona Servicios donde: agrega, actualiza o elimina servicios ya sean de mantenimiento o limpieza. |
|  | El administrador gestiona Productos donde: agrega, actualiza o elimina productos, además de verificar el stock. |
|  | El sistema realizara un reporte de ventas por cantidad o fechas al administrador. |
|  | El Operario puede solicitar un reporte con las ventas que realizo su persona ordenadas por fecha. |
|  | El administrador gestiona las reservaciones donde: agrega, actualiza o elimina reservaciones según la actividad en el taller y las solicitudes de los clientes. |
|  | El Operador registra las ventas del Taller en el sistema. |

|  |  |  |
| --- | --- | --- |
| **Nombre:** | Autenticar Usuario | |
| **Autor:** | Lazcano Ballesteros Lexer | |
| **Fecha:** | 21/04/2020 | **REF:** |
| **Actor**  Usuario | | |
| **Descripción**  Este caso de uso permite que el Administrador u Operador pueda validarse como usuario del sistema. | | |
| **Condiciones previas**  El Usuario debe entrar a la aplicación. | | |
| **Flujo principal**  1. El Usuario ingresa su cuenta, su contraseña y hace click en el botón Ingresar.  2. El sistema verifica los datos  3. El sistema despliega la pantalla principal. | | |
| **Flujo alterno**  3.1 El Usuario ingresa mal su cuenta y/o contraseña, el sistema muestra un mensaje de error. | | |
| **Condiciones posteriores**  El Usuario entra al sistema y despliega pantalla principal. | | |

|  |  |  |
| --- | --- | --- |
| **Nombre:** | Editar Contraseña | |
| **Autor:** | Benítez Díaz Néstor Alejandro | |
| **Fecha:** | 19/04/2020 | **REF:** |
| **Actor**  Usuario | | |
| **Descripción**  Este caso de uso permite que el Administrador u Operador pueda cambiar su contraseña. | | |
| **Condiciones previas**  El Usuario debe entrar al sistema. | | |
| **Flujo principal**  1. El Usuario hace click en el botón Editar.  2. El sistema despliega una ventana donde puede ingresarse la nueva contraseña.  3. El usuario ingresa la nueva contraseña y presiona Aceptar.  4. El sistema cambia la contraseña.  5. El sistema despliega la pantalla principal. | | |
| **Flujo alterno**  2.1 El Usuario presiona Cancelar para salir de la ventana de Editar. | | |
| **Condiciones posteriores**  El Usuario se encuentra dentro del sistema | | |

|  |  |  |
| --- | --- | --- |
| **Nombre:** | Solicitar Reporte de Ventas | |
| **Autor:** | Ugarte Torrez Jenny Laura | |
| **Fecha:** | 24/04/2020 | **REF:** |
| **Actor**  Operador | | |
| **Descripción**  Este caso de uso permite al operador poder solicitar un reporte detallado de las ventas que realizo en una fecha indicada. | | |
| **Condiciones previas**  El operador debe estar dentro el sistema | | |
| **Flujo principal**  1. El operador hace click en el botón “Reporte Ventas”.  2. El sistema despliega una ventana donde el operador ingresa una fecha.  3. El sistema despliega el reporte de la fecha indicada.  4. El operador puede imprimir el reporte. | | |
| **Flujo alterno**  3.1 Si no existe un reporte de esa fecha el sistema despliega una ventana emergente que diga “No existe el reporte en la fecha indicada” | | |
| **Condiciones posteriores**  El operador tiene conocimiento del reporte deseado | | |

|  |  |  |
| --- | --- | --- |
| **Nombre:** | Gestionar Reservaciones | |
| **Autor:** | Benítez Díaz Néstor Alejandro | |
| **Fecha:** |  | **REF:** |
| **Actor**  Operador | | |
| **Descripción**  Este caso de uso permite que el Operador pueda insertar, modificar y eliminar las reservaciones realizadas por los clientes. | | |
| **Condiciones previas**  El Operador debe entrar al sistema. | | |
| **Flujo principal**  1. El Operador hace click en el botón buscar para conocer los datos del cliente mediante su CI.  2. El Operador visualiza las fechas y horarios disponibles.  3. El Operador hace click en Reservar Servicio en una fecha y hora disponible y realiza su reserva.  4. El sistema verifica que la fecha y hora esté disponible y almacena | | |
| **Flujo alterno**   * 1. De no existir el cliente en la base de datos el operador puede crear uno nuevo   2. El operador ingresa los datos del nuevo cliente   3. El operador presiona guardar | | |
| **Condiciones posteriores**  La reserva se guarda en la base de datos y regresa pantalla principal. | | |

|  |  |  |
| --- | --- | --- |
| **Nombre:** | **Registrar Ventas** | |
| **Autor:** | Lazcano Ballesteros Lexer | |
| **Fecha:** |  | **REF:** |
| **Actor**  Operador | | |
| **Descripción**  El Operador registra las ventas del Taller en el sistema. | | |
| **Condiciones previas**  El usuario debe entrar a la página web del Taller. | | |
| **Flujo principal**  1. El operador ingresa el ci del cliente.  2. El operador selecciona los servicios y productos que requiere el cliente además de su cantidad  3. El operador hace click en añadir.  4. El operador hace click en Realizar venta.  5. El sistema emite un mensaje de confirmación.  6. Los detalles se envían a Factura.  7. El sistema imprime la factura | | |
| **Flujo alterno**   * 1. De no existir el cliente en la base de datos el operador puede crear uno nuevo   2. El operador ingresa los datos del nuevo cliente   3. El operador presiona guardar   4.1 El operador puede cancelar la venta | | |
| **Condiciones posteriores**  Los registros de detalles y factura se guardan en la BdD | | |

|  |  |  |
| --- | --- | --- |
| **Nombre:** | Gestionar Cliente | |
| **Autor:** | Ugarte Torrez Jenny Laura | |
| **Fecha:** |  | **REF:** |
| **Actor**  Usuario | | |
| **Descripción**  Este caso de uso permite que el Operador y el Administrador puedan insertar, modificar y eliminar los datos de los clientes. | | |
| **Condiciones previas**  El operador u administrador deben estar dentro el sistema. | | |
| **Flujo principal**   1. El operador/administrador ingresa en la ventana clientes. 2. Se despliega la información de los clientes. 3. El operador/administrador selecciona uno de los clientes de ser necesario para “Editar” o “Eliminarlo”. 4. El operador/administrador selecciona una de las opciones “Nuevo”, “Editar” ó “Eliminar” según sea su requerimiento. 5. En caso de presionar Nuevo o Editar, ingresa los datos del cliente y hace click en guardar. 6. Se despliega la información de los clientes. | | |
| **Flujo alterno**  4.1 Si se presiona “Eliminar”, se elimina al cliente seleccionado y se refresca la información de los clientes. | | |
| **Condiciones posteriores**  La información de los clientes es actualizada según las acciones realizadas. | | |

|  |  |  |
| --- | --- | --- |
| **Nombre:** | Generar Reportes | |
| **Autor:** | Ugarte Torrez Jenny Laura | |
| **Fecha:** |  | **REF:** |
| **Actor**  Administrador | | |
| **Descripción**  Este caso de uso permite que el Administrador pueda generar o imprimir los reportes de actividad realizada en el taller. | | |
| **Condiciones previas**  El administrador debe ingresar al sistema | | |
| **Flujo principal**   1. El administrador ingresa a la ventana “Reportes” mediante el botón reportes. 2. El sistema despliega una ventana para seleccionar los distintos reportes. 3. El administrador selecciona el reporte requerido. 4. El administrador puede solicitar un reporte diario o mensual. 5. El administrador ingresa la fecha requerida. 6. El administrador presiona el botón “imprimir” de ser necesario. | | |
| **Flujo alterno** | | |
| **Condiciones posteriores**  El administrador tiene conocimiento de los reportes necesarios. | | |

|  |  |  |
| --- | --- | --- |
| **Nombre:** | Gestionar Productos | |
| **Autor:** | Benítez Díaz Néstor Alejandro | |
| **Fecha:** |  | **REF:** |
| **Actor**  Administrador | | |
| **Descripción**  Este caso de uso permite que el Administrador podrá insertar, modificar y eliminar productos de su inventario. | | |
| **Condiciones previas**  El Administrador debe estar dentro el sistema. | | |
| **Flujo principal**   1. El Administrador ingresa en la ventana productos. 2. Se despliega la información de los productos. 3. El Administrador selecciona uno de los productos de ser necesario para “Editar” o “Eliminarlo”. 4. El administrador selecciona una de las opciones “Nuevo”, “Editar” ó “Eliminar” según sea su requerimiento. 5. En caso de presionar Nuevo o Editar, ingresa los datos del producto y hace click en guardar. 6. Se despliega la información de los clientes. | | |
| **Flujo alterno**  4.1 Si se presiona “Eliminar”, se elimina al cliente seleccionado y se refresca la información de los productos. | | |
| **Condiciones posteriores**  La información de los productos es actualizada según las acciones realizadas. | | |

|  |  |  |
| --- | --- | --- |
| **Nombre:** | Gestionar Servicios | |
| **Autor:** | Lazcano Ballesteros Lexer | |
| **Fecha:** |  | **REF:** |
| **Actor**  Administrador | | |
| **Descripción**  Este caso de uso permite que el Administrador podrá insertar, modificar y eliminar los servicios que se realizan en el taller. | | |
| **Condiciones previas**  El Administrador debe estar dentro el sistema. | | |
| **Flujo principal**   1. El Administrador ingresa en la ventana servicios. 2. Se despliega la información de los servicios. 3. El Administrador selecciona uno de los servicios de ser necesario para “Editar” o “Eliminarlo”. 4. El administrador selecciona una de las opciones “Nuevo”, “Editar” ó “Eliminar” según sea su requerimiento. 5. En caso de presionar Nuevo o Editar, ingresa los datos del servicio y hace click en guardar. 6. Se despliega la información de los servicios. | | |
| **Flujo alterno**  4.1 Si se presiona “Eliminar”, se elimina al servicio seleccionado y se refresca la información de los servicios. | | |
| **Condiciones posteriores**  La información de los servicios es actualizada según las acciones realizadas. | | |

|  |  |  |
| --- | --- | --- |
| **Nombre:** | Gestionar Técnicos | |
| **Autor:** | Lazcano Ballesteros Lexer | |
| **Fecha:** |  | **REF:** |
| **Actor**  Administrador | | |
| **Descripción**  Este caso de uso permite que el Administrador podrá insertar, modificar y eliminar los datos de los técnicos que forman parte del taller. | | |
| **Condiciones previas**  El administrador debe estar dentro el sistema | | |
| **Flujo principal**   1. El Administrador ingresa en la ventana “técnicos”. 2. Se despliega la información de los técnicos. 3. El Administrador selecciona uno de los técnicos de ser necesario para “Editar” o “Eliminarlo”. 4. El administrador selecciona una de las opciones “Nuevo”, “Editar” ó “Eliminar” según sea su requerimiento. 5. En caso de presionar Nuevo o Editar, ingresa los datos del técnico y hace click en guardar. 6. Se despliega la información de los técnicos. | | |
| **Flujo alterno**  4.1 Si se presiona “Eliminar”, se elimina al técnico seleccionado y se refresca la información de los técnicos. | | |
| **Condiciones posteriores**  La información de los técnicos es actualizada según las acciones realizadas. | | |

|  |  |  |
| --- | --- | --- |
| **Nombre:** | Gestionar Rampas | |
| **Autor:** | Benítez Díaz Néstor Alejandro | |
| **Fecha:** |  | **REF:** |
| **Actor**  Administrador | | |
| **Descripción**  Este caso de uso permite que el Administrador podrá insertar, modificar y eliminar los datos de las rampas que forman parte del taller. | | |
| **Condiciones previas**  El administrador debe estar dentro el sistema | | |
| **Flujo principal**   1. El Administrador ingresa en la ventana “rampas”. 2. Se despliega la información de las rampas. 3. El Administrador selecciona uno de las rampas de ser necesario para “Editar” o “Eliminarlo”. 4. El administrador selecciona una de las opciones “Nuevo”, “Editar” ó “Eliminar” según sea su requerimiento. 5. En caso de presionar Nuevo o Editar, ingresa los datos de la rampa y hace click en guardar. 6. Se despliega la información de las rampas. | | |
| **Flujo alterno**  4.1 Si se presiona “Eliminar”, se elimina la rampa seleccionada y se refresca la información de las rampas. | | |
| **Condiciones posteriores**  La información de las rampas es actualizada según las acciones realizadas. | | |

|  |  |  |
| --- | --- | --- |
| **Nombre:** | Gestionar Usuarios | |
| **Autor:** | Ugarte Torrez Jenny Laura | |
| **Fecha:** |  | **REF:** |
| **Actor**  Administrador | | |
| **Descripción**  Este caso de uso permite que el Administrador podrá insertar, modificar y eliminar los usuarios que podrán ingresar al sistema. | | |
| **Condiciones previas**  El administrador debe estar dentro el sistema. | | |
| **Flujo principal**   1. administrador ingresa en la ventana usuarios. 2. Se despliega la información de los usuarios. 3. El administrador selecciona uno de los usuarios de ser necesario para “Editar” o “Eliminarlo”. 4. El administrador selecciona una de las opciones “Nuevo”, “Editar” ó “Eliminar” según sea su requerimiento. 5. En caso de presionar Nuevo o Editar, ingresa los datos del usuario y hace click en guardar. 6. Se despliega la información de los usuarios. | | |
| **Flujo alterno**  4.1 Si se presiona “Eliminar”, se elimina al usuario seleccionado y se refresca la información de los clientes. | | |
| **Condiciones posteriores**  La información de los usuarios es actualizada según las acciones realizadas. | | |

* + 1. **DIAGRAMA DE SECUENCIAS**
       1. **AUTENTICACION USUARIO**

****

* + - 1. **EDITAR CONTRASEÑA**

****

* + - 1. ** GESTIONAR RAMPAS**
      2. **REALIZAR VENTA DE SERVICIO**

****

* + - 1. **GESTIONAR USUARIOS**

1. **ESTIMACION DEL PROYECTO DE SOFTWARE**

**ESTIMACIÓN POR CASOS DE USO:**

****

UUCP = UAW + UUCW  
UUCP = Puntos de Casos de uso sin ajustar  
UAW = Factor de peso de los actores sin ajustar  
UUCW = Factor de peso de los casos de uso sin ajustar

|  |  |  |
| --- | --- | --- |
| **Tipo de Actor** | **Descripción** | **De peso Factor** |
| **Simple** | Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, aplication Programming Interface) | 1 |
| **Medio** | Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto | 2 |
| **Complejo** | Una persona que interactúa con el sistema mediante una interfaz grafica | 3 |

UAW = 2 \* 3  
UAW = 6

|  |  |  |
| --- | --- | --- |
| **Tipo de caso de uso** | **Descripción** | **Factor De peso** |
| Simple | El caso de uso contiene de 1 a 3 transacciones | 5 |
| Medio | El caso de uso contiene de 4 a 7 transacciones | 10 |
| Complejo | El caso de uso contiene más de 8 transacciones | 15 |

|  |  |  |
| --- | --- | --- |
| **CASO DE USO** | **TIPO** | **VALOR DADO** |
| **Autenticar Usuario** | Simple | 5 |
| **Editar Contraseña** | Simple | 5 |
| **Gestionar Usuarios** | Medio | 10 |
| **Gestionar Clientes** | Medio | 10 |
| **Gestionar Rampas** | Medio | 10 |
| **Gestionar Técnicos** | Medio | 10 |
| **Gestionar Servicios** | Medio | 10 |
| **Gestionar Productos** | Medio | 10 |
| **Generar Reportes** | Simple | 5 |
| **Solicitar Reporte Ventas** | Simple | 5 |
| **Gestionar Reservaciones** | Medio | 10 |
| **Registrar Venta** | Complejo | 15 |

UUCW = 4\*5 + 7\*10+1\*15 = 20 + 70 + 15  
UUCW = 105

**PUNTOS DE CASOS DE USO SIN AJUSTAR:**

UUCP = UAW +UUCW

UUCP = 6+105 = 111

**Cálculo puntos de Casos de uso ajustados UCP:**

UCP = UUCP \* TCF \* EF

Dónde:

UCP = Puntos de Casos de uso ajustados.

UUCP = Puntos de casos de uso sin ajustar.

TCF = Factor de complejidad técnica.

EF = Factor del ambiente.

**CÁLCULO TCF (FACTOR DE COMPLEJIDAD TÉCNICA):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Factor** | **Descripción** | **Peso** | **Valor Est.** | **Producto** | **Descripción** |
| **T1** | Sistema distribuido. | 2 | 0 | 0 | El Sistema es centralizado |
| **T2** | Objetivos de performance o tiempo de respuesta. | 1 | 2 | 2 | Limitada por la cantidad de productos vendidos |
| **T3** | Eficiencia del usuario final. | 1 | 4 | 4 | Los roles necesitan estar relacionados con el Sistema para su funcionamiento |
| **T4** | Procesamiento interno complejo. | 1 | 2 | 2 | No hay cálculos complejos |
| **T5** | El código debe ser reutilizable. | 1 | 4 | 4 | Podría ser usado por sistemas similares |
| **T6** | Facilidad de instalación. | 0.5 | 5 | 2.5 | Facilidad en la instalación |
| **T7** | Facilidad de uso. | 0.5 | 4 | 2 | El sistema debe ser fácil de usar |
| **T8** | Portabilidad. | 2 | 0 | 0 | No es necesario |
| **T9** | Facilidad de cambio. | 1 | 0 | 0 | Los cambios no afectan las funcionalidades del sistema |
| **T10** | Concurrencia. | 1 | 0 | 0 | No hay concurrencia |
| **T11** | Incluye objetivos especiales de seguridad. | 1 | 3 | 3 | El sistema sólo permite que usuario realice las funcionalidades correspondientes a su rol dentro del sistema. |
| **T12** | Provee acceso directo a terceras partes. | 1 | 0 | 0 | Solo acceso a usuarios |
| **T13** | Se requiere facilidades especiales de entrenamiento a usuario. | 1 | 1 | 1 | Sistema fácil de usar |
| **Total** |  |  |  | 20.5 |  |

TFC = 0.6 + 0.01 \* (total)

TFC = 0.6 + 0.01 \* (20.5)

TFC = 0.805

**CÁLCULO EF (FACTOR DEL AMBIENTE):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Factor** | **Descripción** | **Peso** | **Valor** | **Producto** | **Descripción** |
| **E1** | Familiaridad con el modelo de utilizado proyecto | 1.5 | 3 | 4.5 | Se está familiarizado con el modelo, pero la experiencia en el modelado es media. |
| **E2** | Experiencia en la aplicación | 0.5 | 2 | 1 | Se tiene experiencia |
| **E3** | Experiencia orientación a en objetos | 1 | 4 | 4 | Buena experiencia en objetos. |
| **E4** | Capacidad del analista líder | 0.5 | 4 | 2 | No existe analista líder, los analistas que integran el equipo de trabajo poseen capacidad buena. |
| **E5** | Motivación | 1 | 5 | 5 | El grupo está muy motivado |
| **E6** | Estabilidad de los requerimientos | 2 | 3 | 6 | Algunos cambios |
| **E7** | Personal part-time | -1 | 1 | -1 | Casi tiempo completo |
| **E8** | Dificultad del lenguaje de programación | -1 | 3 | -3 | Se usara lenguaje C# |
| **Total** |  |  |  | **18.5** |  |

**EF = 1.4 + (-0.03 \* Total)**

**EF = 1.4 + (-0.03 \* 18.5)**

**EF = 0.85**

**Entonces:**

**UCP = UUCP \* TCF \* EF**

**UCP = 111 \* 0.805 \* 0.85**

**UCP = 75.95**

* 1. **ESFUERZO**

**ESTIMACIÓN DEL ESFUERZO (CASOS DE USO)**

|  |  |
| --- | --- |
| **Factor** | **Filtro** |
| **De E1 a E6** | Factor < 3 |
| **De E7 a E8** | Factor > 3 |

**Tenemos:**

Factor < 3 = 2

Factor > 3 = 0

El total es 2

**Entonces para evaluar el resultado o la cantidad total según la siguiente tabla:**

|  |  |
| --- | --- |
| **Horas-Persona (CF)** | **Descripción** |
| **20** | Si el valor es<=2 |
| **28** | Si el valor es<=4 |
| **36** | Si el valor es>= |

**Si es 2 entonces:**

**CF = 20[h − hom]**

**Entonces:**

**E = CF \* UCP**

**E = 20 \* 75.95**

**E = 1519 [h − hom]**

* 1. **TIEMPO**

T = 1519[h - hom]/3[hom]  
T = 506.33 [h]

* 1. **COSTO**

Costo Total = 1519 [h - hom] \* 20[Bs/h - hom]  
**Costo Total = 30380 [Bs]**

**ESFUERZO TOTAL DEL PROYECTO**

|  |  |  |
| --- | --- | --- |
| **ACTIVIDAD** | **PORCENTAJE** | **H-HOMBRE** |
| Análisis | 10.00% | 379.75 |
| Diseño | 20.00% | 759.5 |
| Programación | 40.00% | 1519 |
| Pruebas | 15.00% | 569.25 |
| Sobrecarga | 15.00% | 569.25 |
| **Total** | **100%** | **3769.75** |

1. **GESTION DE RIESGOS**
   1. **LISTA DE CHQUEO DE RIESGOS**
      1. **RIESGOS DEL PROCESO**

|  |  |  |
| --- | --- | --- |
| **PREGUNTA** | **SI** | **NO** |
| ¿Su organización ha desarrollado una descripción escrita de proceso de software que se utilizara en este proyecto? | **x** |  |
| ¿Se llevan a cabo regularmente revisiones técnicas formales de las especificaciones de requisitos, diseño y código? | **x** |  |
| ¿Se utiliza un mecanismo para controlar los cambios en los requisitos del cliente que afectan el software? |  | **x** |

* + 1. **TAMAÑO DEL PERSONAL Y RIESGOS DE EXPERIENCIA**

|  |  |  |
| --- | --- | --- |
| **PREGUNTA** | **SI** | **NO** |
| ¿Hay suficientes personas disponibles? | **x** |  |
| ¿El personal está comprometido durante toda la duración del proyecto? | **x** |  |
| ¿Algún personal del proyecto trabajara solo a tiempo parcial en este proyecto? |  | **x** |

* + 1. **RIESGOS TECNOLOGICOS**

|  |  |  |
| --- | --- | --- |
| **PREGUNTA** | **SI** | **NO** |
| ¿La tecnología que se va a construir es nueva para su organización? | **x** |  |
| ¿El software interactúa con hardware nuevo o no probado? |  | **x** |
| ¿Los requisitos para el producto exigen la creación de componentes de programa que no se hayan desarrollado previamente por su organización? | **x** |  |

* + 1. **RIESGOS DEL ENTORNO DE DESARROLLO**

|  |  |  |
| --- | --- | --- |
| **PREGUNTA** | **SI** | **NO** |
| ¿Hay disponible una herramienta de gestión de proyectos de software? | **x** |  |
| ¿Los compiladores o generadores de código están disponibles y son apropiados para el producto que se va a construir? | **x** |  |
| ¿Están disponibles las herramientas de análisis y diseño? | **x** |  |

* + 1. **RIESGOS DEL IMPACTO DEL NEGOCIO**

|  |  |  |
| --- | --- | --- |
| **PREGUNTA** | **SI** | **NO** |
| ¿La calidad del software cumple con la funcionalidad esperada? | **x** |  |
| ¿El software presenta genera algún problema en la actividad diaria del negocio? |  | **x** |
| ¿El software fue entregado en el tiempo pactado? | **x** |  |

* + 1. **RIESGOS RELACIONADOS CON EL CLIENTE**

|  |  |  |
| --- | --- | --- |
| **PREGUNTA** | **SI** | **NO** |
| ¿El personal está capacitado para el manejo del software? | **x** |  |
| ¿El personal presenta algún rechazo a la implementación de software? |  | **x** |
| ¿El cliente se encuentra satisfecho con el software implementado? | **x** |  |

* 1. **TABLA DE EVALUACION DE RIESGOS**
     1. **TAMAÑO DEL PERSONAL Y RIESGOS DE EXPERIENCIA**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RIESGO** | **CATEGORIA** | **AREA** | **PROB.** | **IMPACTO** | **COSTO[BS]** | **ER** | **IMPACTO** |
| Un integrante no completa su parte del proyecto en fecha establecida | Proyecto | Participantes | 0,4 | 0,7 | 2000 | 560 | Marginal |
| Falta de un integrante a las reuniones del equipo | Proyecto | Participantes | 0,5 | 0,4 | 100 | 20 | Marginal |
| Abandono de un integrante | Proyecto | Participantes | 0,4 | 0,8 | 5000 | 1600 | Marginal |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RIESGO** | **CATEGORIA** | **AREA** | **PROB.** | **IMPACTO** | **COSTO[BS]** | **ER** | **IMPACTO** |
| Posibles cambios en los requisitos del proyecto | Proyecto | Requisitos | 0,3 | 0,6 | 1000 | 720 | Marginal |

* + 1. **RIESGOS DEL PROCESO**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RIESGO** | **CATEGORIA** | **AREA** | **PROB.** | **IMPACTO** | **COSTO[BS]** | **ER** | **IMPACTO** |
| Se trata de una tecnología nueva para la organización. | Proyecto | Mantenimiento | 0,4 | 0,6 | 3000 | 720 | Marginal |
| Existen dudas de que el proyecto sea realizable. | Proyecto | Coste y mantenimiento | 0,2 | 0,2 | 5000 | 200 | Despreciable |

* + 1. **RIESGOS TECNOLOGICOS**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RIESGO** | **CATEGORIA** | **AREA** | **PROB.** | **IMPACTO** | **COSTO[BS]** | **ER** | **IMPACTO** |
| No exista ayuda en internet y documentación disponible. | Proyecto | Planificación | 0,4 | 0,6 | 6000 | 1440 | Marginal |
| La poca existencia de expertos a los cuales solicitar ayuda acerca de las herramientas | Proyecto | Planificación | 0,6 | 0,8 | 6000 | 2880 | Critico |
| Los integrantes del equipo de desarrollo tienen formación adecuada | Proyecto | Participantes | 0,2 | 0,4 | 5000 | 1600 | Despreciable |
| Están todas las Herramientas de desarrollo integradas | Proyecto | Planificación y rendimiento | 0,2 | 0,2 | 2000 | 80 | Despreciable |
| La existencia de generadores de código apropiados para la aplicación | Proyecto | Planificación | 0,2 | 0,2 | 4000 | 160 | Despreciable |

* + 1. **RIESGOS DEL ENTORNO DE DESARROLLO**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RIESGO** | **CATEGORIA** | **AREA** | **PROB.** | **IMPACTO** | **COSTO[BS]** | **ER** | **IMPACTO** |
| No se cumple con las características esperas por el cliente | Proyecto | Participantes y requisitos | 0.2 | 0,8 | 6000 | 960 | Marginal |
| Se presenta errores en la ejecución del problema | Técnico | Implementación | 0,2 | 0,8 | 4000 | 640 | Marginal |
| No se entrega el software en la fecha indicada | Proyecto | Calendario | 0,3 | 0,7 | 800 | 168 | Marginal |

* + 1. **RIESGOS DEL IMPACTO DEL NEGOCIO**
    2. **RIESGOS RELACIONADOS CON EL CLIENTE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RIESGO** | **CATEGORIA** | **AREA** | **PROB.** | **IMPACTO** | **COSTO[BS]** | **ER** | **IMPACTO** |
| El usuario no tiene conocimiento para manejar la aplicación | Proyecto | Personal | 0.5 | 0,3 | 200 | 30 | Marginal |
| El usuario no está dispuesto a adecuarse a la nueva tecnología | Proyecto | Personal | 0,4 | 0,3 | 200 | 24 | Despreciable |
| El Cliente está inconforme con el software entregado | Proyecto | Participantes y requisitos | 0,2 | 0,9 | 7000 | 1260 | Marginal |

* 1. **PLAN RMMM**

Se incluirán los riesgos que se consideren críticos y además tengan un costo mayor o igual a Bs. 1000.

* + 1. **RIESGO: POSIBLES CAMBIOS EN LOS REQUISITOS DEL PROYECTO**
* **Mitigación:**

Tener los requisitos muy claros desde el inicio. Tener buena comunicación con el cliente. Tener comentado todo el código del proyecto.

* **Monitoreo:**

Presentar informes constantes de avance al cliente. Verificar el cumplimento de los estándares de programación.

* **Gestión:**

Designar al programador que realizo la programación de esa área o una cercana, al desarrollo e implantación del nuevo requisito.

* + 1. **RIESGO: UN INTEGRANTE NO COMPLETA SU PARTE DEL PROYECTO EN FECHA ESTABLECIDA**
* **Mitigación:**

Asignar las tareas según las habilidades de todos los integrantes. Brindar un tiempo razonable para las entregas.

* **Monitoreo:**

Presentar avances y sus respectivos reportes en las reuniones de equipo.

* **Gestión:**

Asignar un tiempo de corrección para que el líder realice lo que no se entregó.

* + 1. **RIESGO: QUE UN MIEMBRO ABANDONE**
* **Mitigación:**

Tener buena comunicación con todos los miembros del proyecto. Que cada integrante respalde su información a su debido tiempo.

* **Monitoreo:**

Tener reuniones constantes sobre el avance del proyecto. Plan en caso de salida de integrantes.

* **Gestión:**

Destinar al líder o miembros del equipo a las áreas de las que era responsable el miembro que abandono.

* + 1. **RIESGO: LA POCA EXISTENCIA DE EXPERTOS A LOS CUALES SOLICITAR AYUDA ACERCA DE LAS HERRAMIENTAS**
* **Mitigación:**

Practicar con las herramientas antes de desarrollar el software, ver apuntes de materias anteriores, buscar tutoriales, leer manuales.

* **Monitoreo:**

Ver las dificultades en desarrollar el software si son con respecto al entorno de desarrollo.

* **Gestión:**

Buscar un experto en el entorno de desarrollo. Se tiene que buscar de inmediato si se nota desde un principio dificultades con el entorno de desarrollo.

* + 1. **RIESGO: NO EXISTA AYUDA EN INTERNET Y DOCUMENTACIÓN DISPONIBLE**
* **Mitigación:**

Recolectar la mayor cantidad de información, también acudir a libros y videos interactivos

* **Monitoreo:**

Revisión periódica de la documentación y ver si se va de acuerdo con lo planificado.

* **Gestión:**

Buscar ayuda de personas que aprobaron la materia con proyectos similares.

* + 1. **RIESGO: LOS INTEGRANTES DEL EQUIPO DE DESARROLLO TIENEN FORMACIÓN ADECUADA**
* **Mitigación:**

Asistir a clases de docencia como de auxiliatura. Comprometerse con la materia y tomar apuntes de lo avanzado.

* **Monitoreo:**

Verificar con el tiempo si el proyecto va de acuerdo con lo planeado.

* **Gestión:**

Obtener conocimiento mediante búsquedas en internet.

* + 1. **RIESGO: NO SE CUMPLE CON LAS CARACTERÍSTICAS ESPERAS POR EL CLIENTE**
* **Mitigación:**

Mantenerse siempre en contacto con el cliente para tener en cuenta lo que espera del software, el personal debe estar al tanto de los cambios que el cliente solicita.

* **Monitoreo:**

Informes constantes, designar un encargado de mantener la comunicación entre el cliente y el proceso de desarrollo

* **Gestión:**

Mantener informes sobre las solicitudes del cliente y determinar si se cumplió con lo pactado, el equipo de trabajo debe estar preparado si ser necesario realizar algún cambio.

* + 1. **RIESGO: SE PRESENTA ERRORES EN LA EJECUCIÓN DEL PROBLEMA**
* **Mitigación:**

Se debe realizar el depurado de código constantemente, se debe realizar las pruebas necesarias para evitar fallas, el código debe estar documentado.

* **Monitoreo:**

Se debe realizar la documentación necesaria del proyecto.

* **Gestión:**

Revisar la documentación del software, revisar y depurar el código.

* + 1. **RIESGO: EL CLIENTE ESTA INCONFORME CON EL SOFTWARE ENTREGADO**
* **Mitigación:**

Se debe coordinar reuniones con el cliente en fases criticas del proyecto, mantener informado al cliente de los alcances del proyecto.

* **Monitoreo:**

Se debe presentar avances del proyecto cada cierto tiempo, asignar a un encargado que se encargue de mostrar los avances y recibir algunas expectativas del cliente.

* **Gestión:**

Minimizar los daños realizando cambios mínimos en el software

1. **ASEGURAMIENTO DE CALIDAD**
   1. **MARCO DE TRABAJO DE PLAN**

En este proyecto el equipo se basará en 3 estándares para el aseguramiento de la calidad del software:

**• ISO-14598:**

Proveen métodos para las mediciones y evaluaciones de la calidad de un producto de software, indicando los requisitos que serán medidos, y analizados en este proceso. Los desarrolladores pueden usar los resultados de la evaluación con el fin de hacer mejoras en el producto o tomar decisiones sobre la estrategia de evolución que seguirá el producto software.

**• ISO 9126:**

Creemos en un software que cumpla los estándares de calidad exigidos en todas las etapas de su ciclo de vida, seleccionando los servicios, técnicas de test y herramientas más adecuadas, según las necesidades y expectativas marcadas, en materia de: Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenimiento y Portabilidad.

**• ISO 9001:**

Sistema de calidad-modelo de aseguramiento de la calidad en diseño/ desarrollo producción, instalación y servicio. La calidad sus productos y servicios.

* 1. **OBJETIVO GENERAL DE LA CALIDAD DEL PROYECTO**

Entregar al cliente un software con todo el contenido, las funciones y las características especificadas como parte del modelo de requerimientos, de manera que el mismo este avalado por los estándares de calidad utilizados.

1. **PLANIFICACION DE LA CALIDAD**
   1. **IDENTIFICACION DE LOS OBJETIVOS ESPECIFICOS DE LA CALIDAD**

De cada estándar se tomará en cuenta solo los siguientes atributos de calidad:

**• ISO-9126:**

**Confiabilidad:** Calidad de tiempo que el software se encuentra disponible para su uso según lo indican los siguientes atributos: madurez, tolerancia a fallas y recuperación.

**Usabilidad:** Grado en el que el software es fácil de usar, según lo indican los siguientes sub-atributos: entendible, aprendible y operables.

**• ISO-14598:**

**ISO/IEC 14598-5:**

**Proceso evaluador:** este proceso es utilizado por organizaciones encargadas de evaluar, provee los requisitos y guías para la evaluación del producto software. Promueve las siguientescaracterísticas de proceso (repetible, Imparcial, Objetivo).

**Actividades:** (Trazabilidad, Resultados, Problemas, Mejoras, Conclusiones)

**• ISO-9001:**

**Enfoque al cliente:** El enfoque consiste en satisfacer las necesidades de los clientes, incluyendo también las expectativas.

**Descripción:** La gestión de la calidad tiene entre sus objetivos satisfacer las necesidades de los clientes y esforzarse en superar las expectativas de los mismos.

**Justificación:** El éxito de una organización se alcanza cuando atrae y retiene la confianza de los clientes. Para esto hay que tener en cuenta las necesidades presentes y futuras que puedan existir.

* 1. **MANUAL DE ESTÁNDARES PARA EL PROYECTO**
     1. **ESTANDARES PARA LAS VENTANAS**
* La ventana principal deberá contener imágenes adecuadas para un buen diseño y vista del usuario.
* Las ventanas deben tener un título que represente la acción que esta realiza.
* El tamaño de las letras del resto de la ventana será de 12px.
* Todos los campos de entrada deberán tener sus respectivas etiquetas.
* Para las ventanas de ingreso de datos se debe tener un mínimo de opciones:
  + Crear una entrada de datos.
  + Guardar la entrada de datos.
  + Eliminar la entrada de datos.
  + Buscar una entrada de datos.
  + Salir de la ventana.
* Las ventanas emergentes de información saldrán para informar un error, información adicional, solicitar una confirmación o la ejecución satisfactoria de un procedimiento.
  + 1. **ESTANDARES PARA LOS REPORTES**
* Cada reporte debe tener un título que este adecuadamente identificando al tipo de reporte.
* Cada reporte deberá tener la fecha de generación de la misma.
* Los datos de cada reporte deberán estar ordenados en forma ascendente por productos.
* El título de cada reporte deberá estar con un tamaño de 15 px, el resto de los datos estarán en 12 px.
* Cada fila de un reporte deberá estar enumeradas.
  + 1. **ESTANDARES PARA LA BASE DE DATOS**
* Los registros que involucren la utilización de cadenas de texto deberán estar definidas en el tipo de dato VARCHAR.
* Los registros que involucren la utilización de números deberán estar definidas en el tipo de dato INT.
* Se especificará las longitudes de los registros con los nombres: NOMBRE longitud 20 caracteres, APELLIDO longitud 50 caracteres, CI longitud 20 caracteres, DIRECCIÓN longitud 200 caracteres y TELÉFONO longitud 50 caracteres.
* Los registros que involucren CUENTA, CONTRASEÑA, ROL y TIPOUSUARIO tendrán una longitud de 50 caracteres.
* Los registros que involucren PRECIO, EFECTIVO, CAMBIO, CANTIDAD, TOTAL, SUBTOTAL estarán definidas con DECIMAL (8,2) donde el primer argumento indica el total de dígitos y el segundo, la cantidad de decimales.
* Se debe usar el formato de fecha DD/MM/AAAA en las tablas.
* Los nombres de las tablas deben hacer alusión a la entidad que lo representa.
* Los campos de las tablas no serán redundantes.
* Los registros en las tablas deberán tener un campo que los identifique del resto.
  + 1. **ESTANDARES PARA LA CALIDAD DEL SOFTWARE**
* El software deberá satisfacer las características que específico el cliente.
* El software deberá ser intuitivo para el manejo de un usuario no experimentado en el área de la informática.
  1. **LISTAS DE VERIFICACION DE SOFTWARE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lista de chequeo de estándares de presentación y funcionalidad de la aplicación para ventanas** | | | | | | |
|  | **Revisión de estándares** | | | | |  |
|  | **Actividad** | **Si** | **No** | **No Aplica** | **Información Adicional** |  |
|  | ¿Las ventanas tienen el título adecuado? |  | \* |  |  |  |
|  | ¿La ventana tiene un distintivo que lo identifique? |  | \* |  |  |  |
|  | ¿Los botones tienen una buena ubicación? | \* |  |  |  |  |
|  | ¿Los datos en las pestañas están ordenados adecuadamente? | \* |  |  |  |  |
|  | ¿El tamaño de la letra es el adecuado? | \* |  |  |  |  |
|  | ¿Los datos en las pestañas están centrados? | \* |  |  |  |  |
|  | ¿Los datos tienen enumeradas las filas? | \* |  |  |  |  |
|  | ¿La ventana tiene un logo representativo a la institución a la que pertenece? | \* |  |  |  |  |
|  | ¿Están ordenados en base a cada requerimiento? | \* |  |  |  |  |
|  | ¿El tiempo de respuesta es mínimo? | \* |  |  |  |  |
|  | ¿Cada requerimiento está en pestañas distintas? |  |  |  |  |  |
|  | ¿Las ventanas tienen el visto bueno del usuario? | \* |  |  |  |  |

**NOTA: El Software se encuentra en desarrollo y aún falta completar la interfaz de Reportes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lista de chequeo de estándares de presentación y funcionalidad de la aplicación para reportes** | | | | | | |
|  | **Revisión de estándares** | | | | |  |
|  | **Actividad** | **Si** | **No** | **No Aplica** | **Información Adicional** |  |
|  | ¿El reporte tiene el título adecuado? |  | \* |  |  |  |
|  | ¿El reporte cuenta con la fecha de generación? |  | \* |  |  |  |
|  | ¿El reporte tiene el número de páginas y el total de páginas? |  | \* |  |  |  |
|  | ¿Los botones de acción tienen una buena ubicación? |  | \* |  |  |  |
|  | ¿Los datos en los reportes están ordenados adecuadamente? |  | \* |  |  |  |
|  | ¿El tamaño de la letra es el adecuado? |  | \* |  |  |  |
|  | ¿Los datos en los reportes están centrados? |  | \* |  |  |  |
|  | ¿El reporte tiene enumeradas las filas? |  | \* |  |  |  |
|  | ¿El reporte tiene un logo representativo a la institución a la que pertenece? |  | \* |  |  |  |
|  | ¿Están ordenados en base a los productos que compro? |  | \* |  |  |  |
|  | ¿Genera reportes dentro el tiempo establecido en las normas? |  | \* |  |  |  |
|  | ¿Cada reporte es generado en ventanas distintas? |  | \* |  |  |  |
|  | ¿El reporte tiene el visto bueno del usuario? |  | \* |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lista de chequeo de estándares de la base de datos** | | | | | | |
|  | **Revisión de estándares** | | | | |  |
|  | **Actividad** | **Si** | **No** | **No**  **Aplica** | **Información**  **Adicional** |  |
|  | ¿Tiene las descripciones de la columna en la base de datos? | \* |  |  |  |  |
|  | ¿Las tablas cuentan con una llave primaria? | \* |  |  |  |  |
|  | ¿Los nombres de las tablas están en plural? |  | \* |  |  |  |
|  | ¿Es correcto el uso de cada campo? | \* |  |  |  |  |
|  | ¿Las tablas tienen todos los atributos necesarios? | \* |  |  |  |  |
|  | ¿Las consultas a tablas son las adecuadas? | \* |  |  |  |  |
|  | ¿Los campos de una tabla son las necesarias de usar? | \* |  |  |  |  |
|  | ¿Se puede hacer un respaldo a la base de datos? |  | \* |  |  |  |
|  | ¿Se puede borrar toda la información de una tabla? | \* |  |  |  |  |
|  | ¿Existe consultas hechas con funciones? |  | \* |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lista de chequeo de estándares de CALIDAD** | | | | | | |
|  | **Revisión de estándares** | | | | |  |
|  | Actividad | Si | No | No Aplica | Información Adicional |  |
|  | ¿El software es multiplataforma? |  | \* |  |  |  |
|  | ¿El Software es fácil de operar? | \* |  |  |  |  |
|  | ¿El Software se adapta a las necesidades del usuario? | \* |  |  |  |  |
|  | ¿El Software guía al usuario para este no cometa errores? | \* |  |  |  |  |
|  | ¿El Software está disponible cuando se necesita? | \* |  |  |  |  |
|  | ¿El Software entrega funcionalidad libre de errores? |  | \* |  |  |  |
|  | ¿El Software es de fácil comprensión? | \* |  |  |  |  |
|  | ¿El Software está enfocado al cliente? | \* |  |  |  |  |
|  | ¿El Software satisface las necesidades del cliente? | \* |  |  |  |  |
|  | ¿Los campos de una tabla son las necesarias de usar? | \* |  |  |  |  |
|  | ¿El Software es escalable? | \* |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lista de verificación de garantía de calidad del Software basado en la norma ISO 9001** | | | | | | |
|  | **Preparación inicial** | | | | |  |
|  | Actividad | Si | No | No Aplica | Información Adicional |  |
|  | ¿Se ha establecido una política organizacional para las actividades de control de calidad? | \* |  |  |  |  |
|  | ¿Todos los recursos y fondos asegurados para realizar actividades de control de calidad? | \* |  |  |  |  |
|  | ¿El equipo de control de calidad ha recibido la capacitación adecuada y las calificaciones apropiadas? |  | \* |  |  |  |
|  | ¿El equipo de control de calidad se ha orientado sobre los roles, responsabilidades, autoridad y valores del control de calidad? | \* |  |  |  |  |
|  | **Estándar de Procedimientos Operativos** | | | | |  |
|  | Actividad | Si | No | No Aplica | Información Adicional |  |
|  | ¿Se ha establecido un cronograma de revisión por pares y se está siguiendo? | \* |  |  |  |  |
|  | ¿Se han establecido revisiones de puntos de control y se están siguiendo? | \* |  |  |  |  |
|  | ¿Las revisiones e informes se documentan de acuerdo con el procedimiento establecido? | \* |  |  |  |  |
|  | **Control de calidad del software: funcionalidad** | | | | |  |
|  | Actividad | Si | No | No Aplica | Información Adicional |  |
|  | ¿La función principal del software resuelve directamente la necesidad principal del usuario? | \* |  |  |  |  |
|  | ¿Es lo suficientemente consistente como para atender las necesidades del usuario con precisión de forma regular? | \* |  |  |  |  |
|  | **Control de calidad del software - Eficiencia** | | | | |  |
|  | Actividad | Si | No | No Aplica | Información Adicional |  |
|  | ¿Las funciones del software responden con una velocidad razonable considerando la tarea que se solicita? | \* |  |  |  |  |
|  | ¿El software utiliza activamente sus recursos económicamente cuando se le asigna la tarea de cumplir funciones? | \* |  |  |  |  |
|  | **Control de calidad del software: usabilidad** | | | | |  |
|  | Actividad | Si | No | No Aplica | Información Adicional |  |
|  | ¿El software se define lo suficientemente bien como para que los usuarios reconozcan fácilmente si es apropiado para sus necesidades? | \* |  |  |  |  |
|  | ¿Es razonable la dificultad de aprender a usar el software teniendo en cuenta sus funciones previstas y las necesidades del usuario? | \* |  |  |  |  |
|  | **Control de calidad del software - Mantenibilidad** | | | | |  |
|  | Actividad | Si | No | No Aplica | Información Adicional |  |
|  | ¿El software está compuesto por un número suficiente de componentes separados que no se ven afectados en caso de que se cambie o altere otro componente? | \* |  |  |  |  |
|  | ¿Puede el software manejar modificaciones razonables sin sufrir defectos, mal funcionamiento y una caída en la calidad general? | \* |  |  |  |  |
|  | ¿Es el software lo suficientemente consistente y confiable para que se puedan establecer los criterios de prueba y se puedan obtener resultados precisos? | \* |  |  |  |  |

1. **PRUEBAS DE SOFTWARE**
   1. **PRUEBAS DE CAJA BLANCA**

public usuario validarUsuario(string usuario, string pass)

{

using (DBTallerEntities db = new DBTallerEntities())

{

usuario oUsuario;//1. funcion

var lst = from d in db.usuario

where d.cuenta == usuario && d.pass == pass && d.activo == 1

select d;

if (lst.Count() > 0)//2. Condicional

{

oUsuario = lst.First();//3. si cumple

}

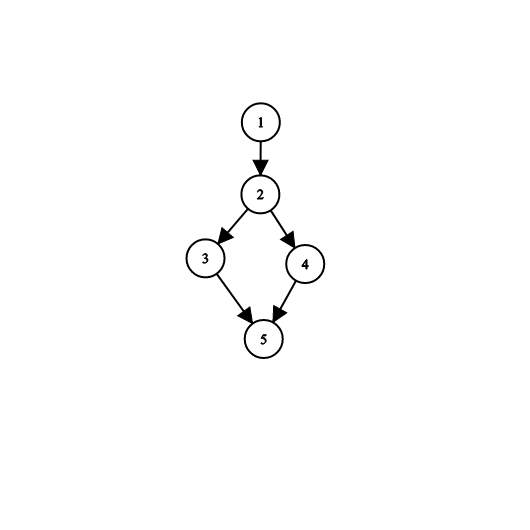
else

oUsuario = null; //4. si no cumple

return oUsuario;//5 Final

}

}



**Número de Nodos**: 5

**Número de aristas:** 5

**Complejidad ciclomática:** 2

**Caminos**

1. 1 2 3 5
2. 1 2 4 5

|  |  |  |  |
| --- | --- | --- | --- |
| **Entrada** | **Camino** | **Resultado Esperado** | **Observación** |
| “Administrador”, 123 | 1 | Ingreso exitoso | Éxito |
| “Adminisstador” , 123 | 2 | Ingreso Denegado | No se ingresa al sistema |

//Funcion para cambiar la contraseña.

private void btnAceptar\_Click(object sender, EventArgs e)

{

if(txtAntigua.Text == cuentaUsuario.pass )//1. condicional

{

if(txtNueva.Text==txtConfirmar.Text)//2. Condicional 2 confirmar txt

{

cuentaUsuario.pass = txtNueva.Text;

controladorUsuario.editar(cuentaUsuario);

this.Close(); //3. Si cumple hacer esto

}

else

{

MessageBox.Show("Las contraseñas no coinciden", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);//4. Mensaje de error si no cumple 2

}

}

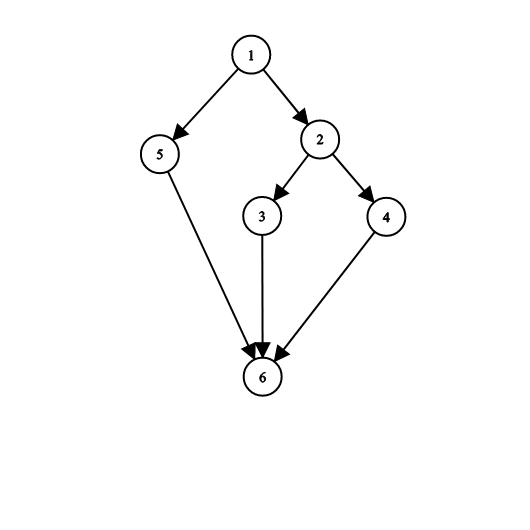
else

{

MessageBox.Show("Contraseña Incorrecta", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); //5. Mensaje de error si no cumple 1

}

}//6.Final del programa



**Número de Nodos**: 6

**Número de Aristas:** 7

**Complejidad:** 7 – 6 + 2 = 3

**Caminos:**

1: 1 – 2 – 3 – 6

2: 1 – 2 – 4 – 6

3: 1 – 5 – 6

|  |  |  |  |
| --- | --- | --- | --- |
| **Entrada** | **Camino** | **Resultado Esperado** | **Observación** |
| 123, 159 , 159 | 1 | Cambio exitoso | éxito |
| 123, 158, 159 | 2 | Error | Las nuevas contraseñas no coinciden |
| 124 , \* , \* | 3 | Error | La contraseña anterior no es la correcta |

private void btnGuardar\_Click(object sender, EventArgs e)

{

BorrarMensajeError();//1 Borra los errores en los campos

if (ValidarCampos())//2. Condicional para saber si los campos estas rellenados

{

if (id == null)//3. Condicional para saber si el id existe

{

try//4. try catch para agregar productos nuevos

{

oProducto = new producto();

oProducto.nombre = txtNombre.Text;

oProducto.precio\_compra = float.Parse(txtCompra.Text);

oProducto.precio\_venta = float.Parse(txtVenta.Text);

oProducto.tipo = Convert.ToString(cBoxTipo.SelectedItem);

oProducto.stock = Convert.ToInt32(txtStock.Text);

oProducto.marca = txtMarca.Text;

oProducto.modelo = txtModelo.Text;

controladorProducto.nuevo(oProducto);

MessageBox.Show(this, "Producto Agregado Exitosamente", "", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

this.Close();

}

catch (Exception)

{

MessageBox.Show(this, "Error en el llenado del Formulario", "", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

try// 5. try catch para modificar productos existentes

{

oProducto.nombre = txtNombre.Text;

oProducto.precio\_compra = float.Parse(txtCompra.Text);

oProducto.precio\_venta = float.Parse(txtVenta.Text);

oProducto.tipo = Convert.ToString(cBoxTipo.SelectedItem);

oProducto.stock = Convert.ToInt32(txtStock.Text);

oProducto.marca = txtMarca.Text;

oProducto.modelo = txtModelo.Text;

controladorProducto.editar(oProducto);

MessageBox.Show(this, "Producto Modificado Exitosamente", "", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

this.Close();

}

catch (Exception)

{

MessageBox.Show(this, "Error en el llenado del Formulario", "", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

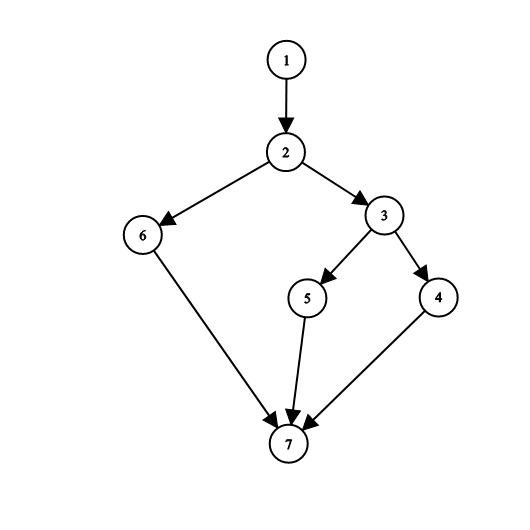
else

{

MessageBox.Show(this, "Error en el llenado del Formulario", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Error);//6. Mensaje de error en el formulario

}

}//7.Final.



**Número de Nodos:** 7

**Número de Aristas:** 8

**Complejidad:** 8 – 7 + 2 = 3

**Caminos:**

1: 1 – 2 – 3 – 4 – 7

2: 1 – 2 – 3 – 5 – 7

3: 1 – 2 – 6 – 7

|  |  |  |  |
| --- | --- | --- | --- |
| **Entrada** | **Camino** | **Resultado Esperado** | **Observación** |
| \*,nombre,$,$,tipo,stock,marca,modelo | 1 | Éxito | Producto Agregado exitosamente |
| \*,nombre,$,$,tipo,stock,marca,modelo | 2 | Éxito | Producto Modificado exitosamente |
| \*,nombre,$,$,tipo,stock, ,modelo | 3 | Error | Faltan datos |

private void btnRealizarVenta\_Click(object sender, EventArgs e)

{

det\_servicio oDetalleServicio = null;

det\_producto oDetalleProducto = null;

factura oFactura = new factura();

cliente oCliente = null;

usuario oUsuario = this.cuentaUsuario;

string ci;

ci = txtCI.Text;

double total = 0; //1. Datos

if (txtCliente.Text == "") //2. condicional 1

MessageBox.Show("NECESARIO CI DEL CLIENTE PARA REALIZAR VENTA"); //3. si cumple

else

{

oCliente = controladorCliente.buscarCI(ci);

oFactura.nro = 99999;

oFactura.id\_cliente = oCliente.id;

oFactura.id\_usuario = oUsuario.id;

oFactura.fecha = DateTime.Today;

oFactura = controladorFactura.nuevo(oFactura);//4. si no cumple

for (int i = 0; i < dgDetalle.RowCount - 1; i++)// ciclo for

{

string t = dgDetalle.Rows[i].Cells[5].Value.ToString();//5.

int id = System.Convert.ToInt16(dgDetalle.CurrentRow.Cells[0].Value);

total += Convert.ToDouble(dgDetalle.Rows[i].Cells[4].Value.ToString());//5. datos

if (string.Equals(t, "Servicio"))//6. condicional 2

{

oDetalleServicio = new det\_servicio();

oDetalleServicio.id\_servicio = System.Convert.ToInt16(dgDetalle.Rows[i].Cells[0].Value);

oDetalleServicio.id\_factura = oFactura.id;

oDetalleServicio.cantidad = System.Convert.ToInt16(dgDetalle.Rows[i].Cells[2].Value);

oDetalleServicio.precio\_unitario = System.Convert.ToDouble(dgDetalle.Rows[i].Cells[3].Value);

oDetalleServicio.subtotal = System.Convert.ToDouble(dgDetalle.Rows[i].Cells[4].Value);

oDetalleServicio.id\_tecnico = System.Convert.ToInt16(dgDetalle.Rows[i].Cells[6].Value);

oDetalleServicio.id\_rampa = System.Convert.ToInt16(dgDetalle.Rows[i].Cells[7].Value);

controladorDetalleServicio.nuevo(oDetalleServicio); // 7. Si cumple 2

}

if (string.Equals(t, "Producto")) //8. Condicional 3

{

oDetalleProducto = new det\_producto();

oDetalleProducto.id\_producto = System.Convert.ToInt16(dgDetalle.Rows[i].Cells[0].Value);

oDetalleProducto.id\_factura = oFactura.id;

oDetalleProducto.cantidad = System.Convert.ToInt16(dgDetalle.Rows[i].Cells[2].Value);

oDetalleProducto.precio\_unitario = System.Convert.ToDouble (dgDetalle.Rows[i].Cells[3].Value);

oDetalleProducto.subtotal = System.Conver

t.ToDouble(dgDetalle.Rows[i].Cells[4].Value);

controladorDetalleProducto.nuevo(oDetalleProducto);//9. Si cumple

}

}//10. Ciclo for

oFactura.total = total;

controladorFactura.editar(oFactura);

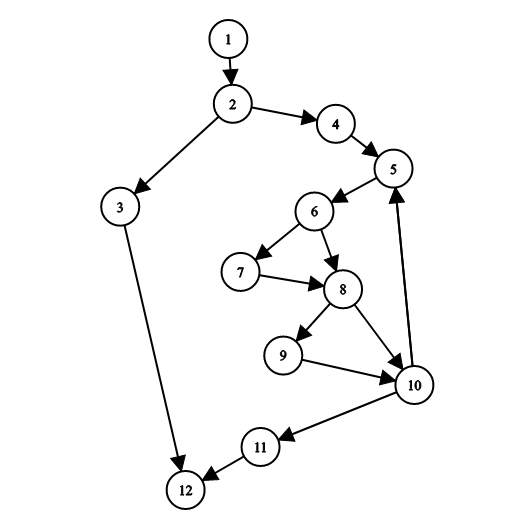
FrmFactura frmFactura = new FrmFactura();

frmFactura.oFactura = oFactura;

frmFactura.ShowDialog();//11.

}

}//12. Final



**Número de Nodos:** 12

**Número de Aristas:** 15

**Complejidad:** 15 -12 + 2 = 5

**Caminos:**

1: 1 – 2 – 3 – 12

2: 1 – 2 – 4 – 5 -6 – 8 -10 -11 - 12

3: 1 -2 – 4 - 5 – 6 - 7 – 8 - 10 – 11 – 12

4: 1 -2 – 4 - 5 – 6 – 8 – 9 - 10 – 11 – 12

5: 1 -2 – 4 - 5 – 6 - 7 – 8 - 9 – 10 -11 – 12

|  |  |  |  |
| --- | --- | --- | --- |
| Entrada | Camino | Resultado Esperado | Observación |
| “ ” | 1 | Mensaje de que el cliente no existe | El sistema indica que el ci ingresado no está registrado |
| “5772706” | 2 | No se realiza la venta ya que no existe ningún producto o servicio | El ci existe pero no se está solicitando ningún servicio. |
| “5772706” | 3 | Se realiza la venta de uno o más servicios | El sistema registra los servicios realizados |
| “5772706” | 4 | Se realiza la venta de uno o más productos | El sistema registra los productos vendidos |
| “5772706” | 5 | Se realiza la venta de productos y servicios | El sistema registra los productos y servicios vendidos |

* 1. **PRUEBAS DE CAJA NEGRA**

1. **CONCLUSIONES Y RECOMENDACIONES**
   1. **CONCLUSIONES**

**•** Se logró diseñar y crear interfaces para tener a disposición los menús requeridos.

• Se consiguió generar una plataforma para lograr el ingreso de los usuarios hacía el sistema.

• Durante el desarrollo del software conseguimos crear una Base de Datos funcional para el taller mecánico “LA RESISTENCIA”, donde podrá manipular y mantener sus datos correctamente actualizados.

• En conclusión y para finalizar el proceso de desarrollo alcanzamos el objetivo principal de desarrollar un sistema de información para el taller mecánico “LA RESISTENCIA”.

* 1. **RECOMENDACIONES**

**•** Se recomienda que el personal que usara el sistema del taller mecánico se encuentre capacitado con conocimiento básico en el área de la informática evitando la posible pérdida y/o daño de datos en su manejo.

• Se recomienda que el negocio tenga políticas establecidas dictando que las cuentas de usuario deben ser únicas y protegidas por contraseñas que solo deben ser de conocimiento del administrador.