

UNIVERSIDAD POLITÉCNICA

SALESIANA

FACULTAD DE INGENIERÍAS

SEDE QUITO-CAMPUS SUR

CARRERA DE INGENIERÍA DE SISTEMAS

MENCIÓN EN ROBÓTICA E INTELIGENCIA ARTIFICIAL

**SOFTWARE PARA CONTROL DEL MANTENIMIENTO Y ADMINISTRACIÓN
DE MAQUINARIA UTILIZADA EN INDUSTRIAS O EMPRESAS**

**TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE
SISTEMAS**

**VERÓNICA FERNANDA HERRERA CANDO
JUAN CARLOS MÁRQUEZ NARANJO**

ING. DORIS MEZA

QUITO, AGOSTO 2010

DECLARACIÓN

Nosotros, Verónica Herrera y Juan Carlos Márquez, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Verónica Herrera

Juan Carlos Márquez

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Verónica Herrera y Juan Carlos Márquez, bajo mi dirección.

Ing. Doris Meza
Director de tesis

RESUMEN

El mantenimiento dentro de la industria se torna de suma importancia ya que con el correcto funcionamiento de los equipos involucrados en el proceso de producción sea cual fuere su naturaleza y funcionalidad conlleva a esta a obtener los mejores réditos un cuanto a ganancias y calidad.

MANPRECOR está concebido para cubrir todas las necesidades en cuanto al mantenimiento y sus procesos ya que mediante esta herramienta de software, el usuario tendrá la posibilidad de controlar y tener documentado todo el mantenimiento de los equipos ingresados en cuanto a:

- Las rutinas de mantenimiento preventivo que se deben realizar, dándoles una fecha exacta y haciendo responsable a un usuario (encargado del mantenimiento) a realizar dicho trabajo.
- Documentar fallas que se presenten en los equipos, esto servirá para llevar un histórico de las mismas, tener un mejor control y la solución exacta si se vuelven a presentar en el trabajo de la maquinaria.
- Tener conocimiento integral de la información del equipo como su ubicación física en la empresa, a qué grupo de maquinaria pertenece, documentos relacionados como garantías o manuales, imágenes del equipo e información adicional según el tipo específico de maquinaria que estemos ingresando.
- Cada vez que una rutina se tenga que cumplir MANPRECOR realiza una orden de trabajo en donde se documenta el estado de la rutina, el tiempo en el cual fue solventada y los insumos que fueron necesarios para llevarla a cabo.
- Tener un control de los insumos que se utilizan en el mantenimiento, con la posibilidad de realizar solicitudes hacia las personas encargadas de la proveeduría teniendo registrado todo el proceso.
- Consultas detalladas de toda la información que maneja el sistema, con una interfaz amigable y fácil de manejar, así como también la posibilidad de tener esa información impresa o exportarla a Excel para su posterior manipulación.

ÍNDICE

CAPÍTULO 1	1	
1.1 Introducción	1	
1.2 Antecedentes y Justificación	3	
1.2.1 Objetivos	3	
1.2.1.1 Objetivo General	3	
1.2.1.2 Objetivos Específicos	3	
1.2.2 Justificación	3	
1.2.3 Delimitación	4	
CAPÍTULO 2	6	
2.1 Mantenimiento de Maquinaria	6	
2.1.1 Objetivo del Mantenimiento	7	
2.1.2 Historia del Mantenimiento	8	
2.1.3 Tipos de Mantenimiento	10	
2.2 Herramientas del Sistema	13	
2.2.1 Arquitectura en capas		13
2.2.1.1 Ventajas	14	
2.2.1.2 Desventajas	15	
2.2.1.3 Capas	15	
2.2.2 Arquitectura Orientada a Servicios (SOA)	17	
2.2.2.1 Concepto	17	
2.2.2.2 Ventajas	18	
2.2.2.3 Desventajas	18	
2.2.3 Base de Datos	19	
2.2.3.1 Generalidades	19	
2.2.3.2 Conexión a través de Enterprise Library	21	
CAPÍTULO 3	24	
3.1. Análisis, Diseño y Desarrollo del Sistema	24	
3.1.1 Análisis y Diseño de la Base de Datos	25	
3.1.1.1 Modelo Lógico	27	

3.1.1.2 Modelo Físico	35
3.1.2 Análisis del Sistema	36
3.1.2.1 Funcionalidades del Sistema	36
3.1.2.2 Diseño UML	38
3.1.2.2.1 Diagrama de Clases	38
3.1.2.2.2 Diagrama de Casos de Uso	41
3.1.2.2.3 Diagrama de Actividades	43
3.1.2.2.4 Diagrama de Secuencia	48
3.1.2.3 Requerimientos del Sistema	55
3.1.3 Diseño del Sistema	70
3.1.3.1 Capas	70
3.1.3.1.1 Capa de Presentación	70
3.1.3.1.2 Capa de Negocio	71
3.1.3.1.3 Capa de Datos	71
3.1.4 Desarrollo del Sistema	71
3.1.4.1 Información y Localización de Equipos	71
3.1.4.2 Administración de Insumos	74
3.1.4.3 Mantenimiento	86
3.1.4.3.1 Preventivo	86
3.1.4.3.2 Correctivo	94
3.1.4.4 Análisis de información	95
3.1.4.4.1 Reportes	95
3.1.4.4.2 Seguridad	99
3.1.4.5 Capas	102
3.2 Etapa de Pruebas	109
3.2.1 Pruebas Funcionales	109
3.2.2 Pruebas de carga	138
CAPÍTULO 4	145
4.1 Conclusiones	145
4.2 Recomendaciones	147
4.3 Bibliografía	148
4.4 Anexos	149

ÍNDICE DE IMÁGENES

- Fig. #1 Procesamiento de la información
- Fig. #2 Modelo Físico de la base de datos
- Fig. #3 Diagrama de Clases de la Capa de Datos
- Fig. #4 Diagrama de Clases de Capa de Presentación
- Fig. #5 Diagrama de Clases de Capa de Procesos
- Fig. #6 Diagrama de casos de uso del módulo de Mantenimiento
- Fig. #7 Diagrama de casos de uso del módulo de Insumos
- Fig. #8 Diagrama de Actividades de Solicitud de Insumo
- Fig. #9 Diagrama de actividades de Rutinas de mantenimiento
- Fig. #10 Diagrama de actividades de Parametrización Inicial módulo de Mantenimiento
- Fig. #11 Diagrama de actividades Parametrización Inicial módulo de Insumos
- Fig. #12 Diagrama de actividades de Órdenes de Trabajo
- Fig. #13 Diagrama de actividades de Fallas
- Fig. #14 Diagrama de secuencia de Ingreso de Solicitudes
- Fig. #15 Diagrama de secuencia de Ingreso de Rutinas
- Fig. #16 Diagrama de secuencia de Ingreso de Ordenes
- Fig. #17 Diagrama de secuencia de Ingreso de Insumos
- Fig. #18 Diagrama de secuencia de Ingreso de Fallas
- Fig. #19 Diagrama de secuencia de Ingreso de Equipos
- Fig. #20 Diagrama de secuencia de Aprobación de Solicitudes
- Fig. #21 Resultados del módulo de mantenimiento
- Fig. #22 Gráfico de rendimiento del módulo de mantenimiento
- Fig. #23 Gráfico de ancho de banda del módulo de mantenimiento
- Fig. #24 Gráfico de tiempo de respuesta del módulo de mantenimiento
- Fig. #25 Resumen del módulo de insumos
- Fig. #26 Gráfico de Rendimiento del módulo de insumos
- Fig. #27 Gráfico de ancho de banda del módulo de insumos
- Fig. #28 Gráfico de tiempo de respuesta del módulo de insumos

Fig. #29	Encendido del computador
Fig. #30	Inicio de Windows
Fig. #31	Instalación
Fig. #32	Acceso al sistema
Fig. #33	Pantalla de inicio
Fig. #34	Selección de módulo
Fig. #35	Alerta de Rutinas
Fig. #36	Acceso Tipos de equipos
Fig. #37	Tipos de equipos
Fig. #38	Ingreso Tipos de equipos
Fig. #39	Edición Tipos de equipos
Fig. #40	Eliminar Tipos de equipos
Fig. #41	Ubicación de Equipos
Fig. #42	Ingreso de Ubicación de Equipos
Fig. #43	Eliminación de Ubicación de Equipos
Fig. #44	Unidades
Fig. #45	Ingreso de Unidades
Fig. #46	Edición de Unidades
Fig. #47	Eliminación de Unidades
Fig. #48	Proveedores
Fig. #49	Ingreso de Proveedores
Fig. #50	Edición de Proveedores
Fig. #51	Eliminación de Proveedores
Fig. #52	Equipos
Fig. #53	Ingreso de Equipos
Fig. #54	Edición de Equipos
Fig. #55	Eliminación de Equipos
Fig. #56	Partes de Equipos
Fig. #57	Ingreso Partes de Equipos
Fig. #58	Edición Partes de Equipos
Fig. #59	Documentación
Fig. #60	Ingreso de Imágenes

Fig. #61	Ingreso de Documentación
Fig. #62	Eliminación de Documentación
Fig. #63	Empleados
Fig. #64	Ingreso Empleados
Fig. #65	Edición Empleados
Fig. #66	Eliminación Empleados
Fig. #67	Usuarios y perfiles
Fig. #68	Ingreso Usuarios y perfiles
Fig. #69	Asignación Usuarios y perfiles
Fig. #70	Edición Usuarios y perfiles
Fig. #71	Edición Usuarios y perfiles 2
Fig. #72	Eliminación Usuarios y perfiles 2
Fig. #73	Perfiles
Fig. #74	Ingreso Perfiles
Fig. #75	Edición Perfiles
Fig. #76	Eliminación Perfiles
Fig. #77	Rutinas
Fig. #78	Ingreso Rutinas
Fig. #79	Órdenes de trabajo
Fig. #80	Ingreso Órdenes de trabajo
Fig. #81	Ingreso Insumos
Fig. #82	Cierre de orden
Fig. #83	Fallas
Fig. #84	Ingreso Fallas
Fig. #85	Registro de solución
Fig. #86	Alertas
Fig. #87	Tipos de insumos
Fig. #88	Ingreso Tipos de insumos
Fig. #89	Edición Tipos de insumos
Fig. #90	Eliminación Tipos de insumos
Fig. #91	Proveedores
Fig. #92	Ingreso Proveedores

Fig. #93	Edición Proveedores
Fig. #94	Eliminación Proveedores
Fig. #95	Insumos
Fig. #96	Ingreso Insumos
Fig. #97	Edición Insumos
Fig. #98	Eliminación Insumos
Fig. #99	Solicitud de insumo
Fig. #101	Solicitud ingresada
Fig. #102	Aprobación
Fig. #103	Compras
Fig. #104	Entrega
Fig. #105	Consultas

ÍNDICE DE ANEXOS

1	Manual explicativo de la base de datos	1
2	Manual de Instalación	10
3	Manual de Usuario	15
4	Presupuesto	74

CAPITULO 1

1.1 INTRODUCCIÓN

Desde el principio de los tiempos, el hombre ha sentido la necesidad de mantener en óptimas condiciones su equipo; aún las más rudimentarias herramientas o aparatos. La mayoría de las fallas que se experimentaban eran el resultado del abuso y esto sigue sucediendo en la actualidad. Al principio solo se hacía mantenimiento cuando ya era imposible seguir usando el equipo. A eso se le llamaba *Mantenimiento de Ruptura o Reactivo*.¹

El mantenimiento hoy en día no es considerado como una actividad de segundo plano, ya que de su correcta aplicación se obtiene bienes reales, cada vez más se potencializa esta actividad con una serie de normas que se han dado por la necesidad de mantener una armonía en el funcionamiento de la maquinaria de cualquier tipo. Por esta razón las entidades que operan con maquinaria se han dado tiempo para fortalecer la actividad del mantenimiento dentro de su organización aplicando planes y procedimientos e inclusive reclutando gente que se enfoque exclusivamente a esta actividad dejando de lado personal externo en algunos casos.

Para nadie es un secreto la exigencia que plantea una economía globalizada, mercados altamente competitivos y un entorno variable donde la velocidad de cambio sobrepasa en mucho nuestra capacidad de respuesta. En este panorama estamos inmersos y vale la pena considerar algunas posibilidades que siempre han estado pero ahora cobran mayor relevancia.

¹ http://www.tpmonline.com/articles_on_total_productive_maintenance/tpm/tpmprocess/maintenanceinhistorySpanish.htm

Particularmente, la imperativa necesidad de redimensionar la empresa implica para el mantenimiento, retos y oportunidades que merecen ser valorados.

Debido a que el ingreso siempre provino de la venta de un producto o servicio, esta visión primaria llevó a la empresa a centrar sus esfuerzos de mejora, y con ello los recursos, en la función de producción.

El mantenimiento fue "un problema" que surgió al querer producir continuamente, de ahí que fue visto como un mal necesario, una función subordinada a la producción cuya finalidad era reparar desperfectos en forma rápida y barata. Sin embargo, sabemos que la curva de mejoras incrementales después de un largo período es difícilmente sensible, a esto se une la filosofía de calidad total, y todas las tendencias que trajo consigo que evidencian que requiere la integración del compromiso y esfuerzo de todas sus unidades. Esta realidad ha volcado la atención sobre un área relegada: el mantenimiento.²

El mantenimiento juega un papel importante en el éxito o fracaso de una empresa, por estudios comprobados se sabe que incide en:

- Costos de producción.
- Calidad del producto servicio.
- Capacidad operacional (aspecto relevante dado el ligamen entre competitividad y por citar solo un ejemplo, el cumplimiento de plazos de entrega).
- Capacidad de respuesta de la empresa como un ente organizado e integrado: por ejemplo, al generar e implantar soluciones innovadoras y manejar oportuna y eficazmente situaciones de cambio.
- Seguridad e higiene industrial.
- Calidad de vida de los colaboradores de la empresa.

² www.monografias.com/mantenimiento

- Imagen y seguridad ambiental de la compañía.

Como se desprende de argumentos de tal peso, " El mantenimiento no es una función "miscelánea", produce un bien real, que puede resumirse en: capacidad de producir con calidad, seguridad y rentabilidad.³

1.2 ANTECEDENTES Y JUSTIFICACIÓN

1.2.1 OBJETIVOS

1.2.1.1 Objetivo General

Desarrollar una herramienta en línea que permita controlar de manera eficiente todas las rutinas de mantenimiento de cualquier tipo de maquinaria.

1.2.1.2 Objetivos Específicos

- 1.- Agilizar y automatizar el proceso de mantenimiento preventivo y correctivo de la maquinaria.
- 2.- Brindar al personal encargado de la manipulación de la maquinaria información correcta con resultados, estadísticas, características así como también las posibles soluciones a los errores más frecuentes.
- 3.- Desarrollar una herramienta eficaz y genérica la cual conlleve la posibilidad de acoplarse a cualquier tipo de industria.

1.2.2 JUSTIFICACIÓN

La necesidad de automatizar los procesos de mantenimiento de sus equipos así como también de documentar las falencias más frecuentes que sufren, con datos que permitan realizar un estudio de la productividad de cada una de ellas hace que

³ www.monografias.com/mantenimiento

los colaboradores del área de mantenimiento de todo tipo de empresas, requieran un software que ofrezca soporte a sus necesidades tanto de información como de disponibilidad de repuestos e insumos desde su propia área de trabajo.

Actualmente las pequeñas y medianas empresas no cuentan con un software de producción nacional que apoye a este tipo de procesos.

Por lo expuesto anteriormente, mediante la aplicación de los conocimientos de programación hacia procesos industriales, lo cual es la esencia académica de nuestra formación se desarrollará el software que lo denominaremos MANPRECOR (Mantenimiento preventivo y correctivo), que asistirán en la planificación y control de los procesos de mantenimiento y manejo de insumos.

1.2.3 DELIMITACIÓN

El presente software en línea apoyará a la industria en el mantenimiento de todo tipo de maquinaria, teniendo en su contenido las rutinas de mantenimiento que se deben tomar para evitar errores de funcionamiento; si ocurren errores o mal funcionamiento, almacenar los pasos que se siguieron para corregirlo y que a futuro sea corregido oportunamente, además de contar con un stock de piezas e insumos que se utilizan en las maquinarias en caso de que fuere necesario cambiar alguna pieza o algún fluido que sea importante en el uso

El sistema estará formado por los siguientes módulos:

Módulo de mantenimiento

Este módulo contendrá: Información y características de cada equipo ingresado, ubicación física de la maquinaria, rutinas de mantenimiento, reportes, estadísticas de funcionamiento, descripción y control de las herramientas que se utilizan en el mantenimiento, y proveedores de servicio de mantenimiento.

Módulo de Inventario de insumos

Este módulo controlará todos los insumos que se utilicen en las maquinarias tales como aceites, tuercas, bandas, cintas, tinta, lubricantes entre otros, guardando relación entre la maquinaria donde se las va a utilizar y el insumo ingresado, además contará con un proceso que permitirá realizar la solicitud de compra desde el departamento de mantenimiento la cual tendrá que pasar por niveles de aprobación para llegar a su compra definitiva y así evitar el desabastecimiento de los insumos que se vayan a utilizar, también contendrá listas de proveedores para realizar compras.

CAPITULO 2.

2.1 MANTENIMIENTO DE MAQUINARIA

El mantenimiento de maquinaria dentro de una empresa u organización se traduce en realizar tareas de control para el correcto funcionamiento de las maquinarias, tareas que van desde el cambio de una pieza ya sea por uso o por defecto en su funcionamiento, el aprovisionamiento de algún fluido del tipo que la máquina lo requiera o la simple lectura de voltaje o corriente que recibe para confirmar que es el correcto. Estas tareas llevan tiempo en realizarlas, sobre todo si no son previstas con anterioridad y se espera hasta que el funcionamiento normal se vea afectado, aplazando así las tareas de producción. Razón por la cual hoy en día todos los procesos de producción que utilizan máquinas deberían tomar muy en cuenta las tareas y operaciones de mantenimiento que se deben realizar, enfocado en todas sus formas como lo es el mantenimiento preventivo que en este caso sería el ideal, el mantenimiento correctivo que aparece con la solución adecuada cuando no se pudo evitar el error en el desenvolvimiento de la maquinaria y el mantenimiento predictivo con la misma esencia del preventivo adelantarse a los defectos de funcionamiento pero teniendo como base los errores ocurridos con anterioridad para que con los correctos pasos a seguir y adelantándose a cualquier mal funcionamiento no vuelvan a ocurrir.

Esto conlleva a que la empresa tenga en su poder una correcta forma de aplicar el mantenimiento y esto se logra a partir de crear una rutina o un plan de mantenimiento que tenga en su contenido las tareas fundamentales que se deben seguir de una forma entendible y de rápido acceso para el operario, esta es la principal necesidad que busca cubrir el sistema desarrollado, entregar una completa información de la maquinaria y todas sus partes, así como las tareas que se deben realizar en el correcto tiempo y espacio, las estadísticas del trabajo cumplido y de las averías que se presentaron, también contar con un stock de las partes que deben ser sujetas a cambio.

2.1.1 OBJETIVOS DEL MANTENIMIENTO

El diseño e implementación de cualquier sistema organizativo y su posterior informatización deben siempre tener presente que están al servicio de unos determinados objetivos. Cualquier sofisticación del sistema debe ser contemplada con gran prudencia en evitar, precisamente, que se enmascaren dichos objetivos o se dificulte su consecución.

En el caso del mantenimiento la organización e información deben estar encaminadas a la permanente consecución de los siguientes objetivos

- Optimizar la disponibilidad del equipo productivo.
- Disminuir los costos de mantenimiento.
- Optimizar los recursos humanos.
- Maximizar la vida de la máquina
- Evitar, reducir, y en su caso, reparar, las fallas sobre los bienes.
- Disminuir la gravedad de las fallas que no se lleguen a evitar.
- Evitar detenciones inútiles o para de máquinas.
- Evitar accidentes.
- Evitar incidentes y aumentar la seguridad para las personas.
- Conservar los bienes productivos en condiciones seguras y preestablecidas de operación.
- Balancear el costo de mantenimiento con el correspondiente al lucro cesante.
- Alcanzar o prolongar la vida útil de los bienes.

El mantenimiento adecuado, tiende a prolongar la vida útil de los bienes, a obtener un rendimiento aceptable de los mismos durante más tiempo y a reducir el número de fallas.

Decimos que algo falla cuando deja de brindarnos el servicio que debía darnos o cuando aparecen efectos indeseables, según las especificaciones de diseño con las que fue construido o instalado el bien en cuestión.

2.1.2 HISTORIA DEL MANTENIMIENTO

Desde hace varias décadas surgió la necesidad de organizar adecuadamente el servicio de mantenimiento con la introducción de programas que realicen un control y administración del mismo, con el objetivo de optimizar la disponibilidad de los equipos productores.

Posteriormente, la necesidad de minimizar los costos propios de mantenimiento acentúa esta necesidad de organización mediante la introducción de controles adecuados de costos. Recientemente, la exigencia a que la industria está sometida de optimizar todos sus aspectos, tanto de costos, como de calidad, como de cambio rápido de producto, conduce a la necesidad de analizar de forma sistemática las mejoras que pueden ser introducidas en la gestión, tanto técnica como económica del mantenimiento. Es la filosofía de la terotecnología. Todo ello ha llevado a la necesidad de manejar desde el mantenimiento una gran cantidad de información.

La historia de mantenimiento acompaña el desarrollo Técnico-Industrial de la humanidad. Al final del siglo XIX, con la mecanización de las industrias, surgió la necesidad de las primeras reparaciones.

Hasta 1914, el mantenimiento tenía importancia secundaria y era ejecutado por el mismo personal de operación o producción.⁴

Con el advenimiento de la primera guerra mundial y de la implantación de la producción en serie, instituida por la compañía Ford-Motor Company, fabricante de vehículos, las fábricas pasaron a establecer programas mínimos de producción, en consecuencia, sintieron la necesidad de crear equipos que pudieran efectuar el mantenimiento de las máquinas de la línea de producción en el menor tiempo posible.

⁴ <http://www.mailxmail.com/cursos/ingenieria-mantenimiento-hospitalario/historia-mantenimiento>

Así surgió un órgano subordinado a la operación, cuyo objetivo básico era la ejecución del mantenimiento, hoy conocida como mantenimiento correctivo. Esa situación se mantuvo hasta la década de los años 30, cuando en función de la segunda guerra mundial y de la necesidad de aumentar la rapidez de la producción, la alta administración industrial se preocupó, no solo en corregir fallas, sino evitar que ocurriesen, y el personal técnico de mantenimiento, pasó a desarrollar el proceso del mantenimiento preventivo, de las averías que, juntamente con la corrosión, completaban el cuadro general de mantenimiento como de la operación o producción.

Por el año de 1950, con el desarrollo de la industria para atender a los esfuerzos de la post-guerra, la evolución de la aviación comercial y de la industria electrónica, los gerentes de mantenimiento observaron que, en muchos casos, el tiempo de paro de la producción, para diagnosticar las fallas era mayor que la ejecución de la reparación; esto da lugar a seleccionar un equipo de especialistas para componer un órgano de asesoramiento a la producción que se llamó «Ingeniería de Mantenimiento» y recibió los cargos de planear y controlar el mantenimiento preventivo y analizar causas y efectos de las averías.

A partir de 1966 con el fortalecimiento de las asociaciones nacionales de mantenimiento de países industrializados, creadas al final del período anterior, y la sofisticación de los instrumentos de protección y medición, la ingeniería de mantenimiento, pasa a desarrollar criterios de predicción o previsión de fallas, visando la optimización de la actuación de los equipos de ejecución de mantenimiento.

Esos criterios, conocidos como mantenimiento predictivo o previsorio, fueron asociados a métodos de planeamiento y control de mantenimiento. Así también hay otros tipos de mantenimiento como el de precisión y el de clase mundial.

La justificación del tema se sustenta ampliamente en la necesidad de que la maquinaria que interviene en el proceso productivo debe siempre mantener una línea de funcionamiento normal sin paras abruptas, esto se logra con una correcta rutina de mantenimiento, tomando en cuenta todas la variables que se pueden presentar, con esto tenemos que hoy en día el mantenimiento es un tema fundamental para un desarrollo armónico y sustentable de cualquier proceso de producción desarrollado en todo tipo de empresas.⁵

2.1.3 TIPOS DE MANTENIMIENTO

2.1.3.1 Mantenimiento Preventivo

La programación de inspecciones, tanto de funcionamiento como de seguridad, ajustes, reparaciones, análisis, limpieza, lubricación, calibración, que deben llevarse a cabo en forma periódica en base a un plan establecido y no a una demanda del operario o usuario; también es conocida como Mantenimiento Preventivo Planificado - MPP .

Su propósito es prever las fallas manteniendo los sistemas de infraestructura, equipos e instalaciones productivas en completa operación a los niveles y eficiencia óptimos.

La característica principal de este tipo de Mantenimiento es la de inspeccionar los equipos y detectar las fallas en su fase inicial, y corregirlas en el momento oportuno. Con un buen Mantenimiento Preventivo, se obtiene experiencias en la determinación de causas de las fallas repetitivas o del tiempo de operación seguro de un equipo, así como en definir puntos débiles de instalaciones, máquinas, etc.⁶

⁵ <http://www.mailxmail.com/curso-ingenieria-mantenimiento-hospitalario/historia-mantenimiento>

⁶ www.solomantenimiento.com

2.1.3.2 Mantenimiento Predictivo

Mantenimiento basado fundamentalmente en detectar una falla antes de que suceda, para dar tiempo a corregirla sin perjuicios al servicio, ni detención de la producción, etc. Estos controles pueden llevarse a cabo de forma periódica o continua, en función de tipos de equipo, sistema productivo, etc.

Para ello se usan instrumentos de diagnóstico, aparatos y pruebas no destructivas, como análisis de lubricantes, comprobaciones de temperatura de equipos eléctricos, etc.

Ventajas del Mantenimiento Predictivo:

- Reduce los tiempos de parada.
- Permite seguir la evolución de un defecto en el tiempo.
- Optimiza la gestión del personal de mantenimiento.
- La verificación del estado de la maquinaria, tanto realizada de forma periódica como de forma accidental, permite confeccionar un archivo histórico del comportamiento mecánico.
- Conocer con exactitud el tiempo límite de actuación que no implique el desarrollo de un fallo imprevisto.
- Toma de decisiones sobre la parada de una línea de máquinas en momentos críticos.
- Confección de formas internas de funcionamiento o compra de nuevos equipos.
- Permitir el conocimiento del historial de actuaciones, para ser utilizado por el mantenimiento correctivo.
- Facilita el análisis de las averías.
- Permite el análisis estadístico del sistema.⁷

Desventajas del Mantenimiento Predictivo:

⁷ www.solomantenimiento.com

- La implantación de un sistema de este tipo requiere una inversión inicial importante, los equipos y los analizadores de vibraciones tienen un costo elevado. De la misma manera se debe destinar un personal a realizar la lectura periódica de datos.
- Se debe tener un personal que sea capaz de interpretar los datos que generan los equipos y tomar conclusiones en base a ellos, trabajo que requiere un conocimiento técnico elevado de la aplicación.

Por todo ello la implantación de este sistema se justifica en máquina o instalaciones donde los paros intempestivos ocasionan grandes pérdidas, donde las paradas innecesarias ocasionen grandes costos.⁸

2.1.3.3 Mantenimiento Correctivo

- No Planificado

Corrección de las averías o fallas, cuando éstas se presentan, y no planificadamente, al contrario del caso de Mantenimiento Preventivo.

Esta forma de Mantenimiento impide el diagnóstico fiable de las causas que provocan la falla, pues se ignora si falló por mal trato, por abandono, por desconocimiento del manejo, por desgaste natural, etc.

El ejemplo de este tipo de Mantenimiento Correctivo No Planificado es la habitual reparación urgente tras una avería que obligó a detener el equipo o máquina dañada.

- Planificado

El Mantenimiento Correctivo Planificado consiste en la reparación de un equipo o máquina cuando se dispone del personal, repuestos y documentos técnicos necesarios para efectuarlo.⁹

⁸ <http://www.monografias.com/trabajos15/mantenimiento-industrial/mantenimiento-industrial.shtml#PREDICTIVO>

2.2 HERRAMIENTAS DEL SISTEMA

Para la creación del sistema se cuenta con varias herramientas de diseño, desarrollo y arquitectura de software, cuyas especificaciones y características más importantes se van a exponer a continuación.

2.2.1 ARQUITECTURA N CAPAS

La arquitectura por capas es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, esto nos sugiere que separemos en capas todos los procesos que se van a dar en el aplicativo, como el manejo de los datos, la lógica del negocio que son todos los procesos por los cuales van a pasar los datos según las necesidades que va a cubrir la solución y la presentación de estos datos como tales para que puedan ser manipulados por el cliente. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles.

Este modelo surge como la arquitectura superior para la construcción de aplicaciones multiplataforma en la mayor parte de las empresas.¹⁰

En el diseño de sistemas informáticos actual se suele usar las arquitecturas multinivel o **Programación por capas**. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables, es decir que pueden ampliarse con facilidad en caso de que las necesidades aumenten.

⁹ www.solomantenimiento.com

¹⁰ <http://kernelerror.net/programacion/php/arquitectura-3-capas.com>

Esta arquitectura es un concepto para la construcción de un sistema, facilita el rápido desarrollo y disminuye costos.

El diseño de aplicaciones basado en n-capas considera a la red como un pool de servicios distribuidos, un concepto mucho más ambicioso que el simple acceso de un cliente a un servidor.¹¹

La separación de la presentación, lógica de negocio y datos es realizada en un número indefinido de capas lógicas, permitiendo a cada capa ser desarrollada, mejorada, gestionada y desplegada de forma independiente. Esta es precisamente la base para el modelo de informática de red en n-capas.

El diseño de aplicaciones en n-capas es ideal para la creación de sistemas adaptables, donde cada componente puede ser utilizado y reutilizado en nuevas combinaciones para satisfacer requisitos de negocio dinámicos.¹²

2.2.1.1 Ventajas

- Desarrollo en cada capa.
- El mantenimiento resulta más fácil ya que los cambios se realizan en cada capa y no es necesario cambiar todo el esquema.
- Mayor flexibilidad permitiendo añadir nuevos módulos para agregar funcionalidades al sistema; la división en componentes reduce la complejidad, permite la reutilización y acelera el proceso de ensamblaje de software.
- El software se hace cada vez más rápido, de mejor calidad y a menor costo.
- Alta escalabilidad, permitiendo manejar muchas peticiones con el mismo rendimiento solo con el aumento de hardware.

¹¹ <http://www.slideshare.net/Decimo/arquitectura-3-capas>

¹² <http://www.forosdelweb.com/f50/modelo-3-capas-arquitectura-n-capas-122898>

- Los costos de mantención del software se reducen.
- Se obtiene un repositorio de objetos probados que pueden ser utilizados en la construcción de nuevas aplicaciones.
- Las arquitecturas basadas en n-capas permiten a los componentes de negocio correr en una LAN, WAN o Internet.¹³

2.2.1.2 Desventajas

- A veces no se logra la contención del cambio y se requiere una cascada de cambios en varias capas.
- Trabajo innecesario por parte de capas más internas o redundante entre varias capas.
- Dificultad de diseñar correctamente la granularidad de las capas.¹⁴

2.2.1.1 Capas

2.2.1.1.1.- Capa de presentación

Es la que ve el usuario también denominada como "capa de usuario", es un medio a través del cual se presenta el sistema al usuario, comunica y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" es decir entendible y fácil de usar para el usuario.

¹³ http://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas

¹⁴ www.galeon.com/zuloaga/Doc/ArqCapasSI.doc

2.2.1.1.2.- Capa de negocio

Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio o incluso de lógica del negocio, porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

Se puede seleccionar almacenar la lógica de negocios sobre cada estación de cliente, u optar por ejecutar la lógica de negocios sobre un servidor de aplicaciones.

15

2.2.1.1.3.- Capa de datos

Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Esta capa permite realizar operaciones con la base de datos a través de procedimientos almacenados de una manera transparente para la capa de negocio.

16

Se ha decidido adoptar esta metodología para desarrollar este software ya que permite tener un mayor grado de organización en la aplicación, detectar y corregir errores de manera eficiente.

¹⁵ http://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas

¹⁶ <http://www.generatorfd.com/Arquitectura.aspx>

2.2.2 ARQUITECTURA ORIENTADA A SERVICIOS (SOA)

2.2.2.1 Concepto

La Arquitectura Orientada a Servicios (Service Oriented Architecture), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.¹⁷

Permite la creación de sistemas altamente escalables que reflejan el negocio de una organización, a su vez brinda una forma estándar de exposición e invocación de servicios, lo cual facilita la interacción entre diferentes sistemas propios o de terceros.¹⁷

SOA define las siguientes capas de software:

- Aplicaciones básicas.- Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad.
- De exposición de funcionalidades.- Donde las funcionalidades de la capa aplicativa son expuestas en forma de servicios (servicios web).
- De integración de servicios.- Facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración.
- De composición de procesos.- Define el proceso en términos del negocio y sus necesidades varían en función del negocio.
- De entrega.- Donde los servicios son desplegados a los usuarios finales.¹⁸

SOA proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación.

¹⁷ http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios

¹⁸ <http://www.es.capgemini.com/servicios/technology/soa/>

2.2.2.2 Ventajas

Flexibilidad, SOA es la primera arquitectura de Tecnologías de Información (TI) que asume lo que los negocios han sabido desde hace mucho tiempo. Se trata esencialmente de un paquete de servicios sueltos, donde cada uno es relativamente económico para construirlo o reemplazarlo si es necesario. Al ser independientes, el poder unirlos permite a SOA adaptar cambios, cuestión imposible para arquitecturas tradicionales.¹⁷

En la Arquitectura Orientada a Servicios, se puede reemplazar un servicio sin tener que preocuparse por la tecnología fundamental; la interfaz es lo que importa y está definida en un estándar universal en servicios Web y XML.¹⁹

Esto es flexibilidad a través de la interoperabilidad. También es la habilidad de asegurar los activos existentes, aplicaciones y bases de datos legales y hacerlos parte de las soluciones empresariales extendiéndolos al SOA en vez de reemplazarlos. El resultado en la red es la habilidad de evolucionar rápida y eficientemente, en otras palabras, adaptarse “orgánicamente” de acuerdo a la demanda del negocio.²⁰

SOA es sencillo y está basado en estándares universales, lo cual asegura que las habilidades para construir aplicaciones sean muy accesibles.

SOA también mejora en los tiempos de realización de cambios en procesos.

¹⁹ http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios

²⁰ <http://www.mastermagazine.info/articulo/3391.php>

2.2.2.3 Desventajas

SOA requiere un cambio en las organizaciones, un alto esfuerzo. No siendo sencillo, para la mayoría de las organizaciones adoptar SOA.²¹

Se requiere además profesionales con alto conocimiento para diseñar la organización, y si se implementa de manera inadecuada esto generaría grandes pérdidas.

2.2.3 BASE DE DATOS

2.2.3.1 Generalidades

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California. Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada. Desde el punto de vista de la Informática, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Una base de datos tiene mucha importancia en el ritmo de vida que llevamos en los actuales momentos, ya que acelera el ritmo al momento de realizar una búsqueda de información.²²

Los elementos claves de la organización en un ambiente de Bases de Datos son:

- Sistema de administración de base de datos
- Administración de información

²¹ <http://www.tecnologiahechapalabra.com/datos/soluciones/gerencia/articulo.asp?i=522>

²² <http://www.monografias.com/trabajos24/bases-de-datos/bases-de-datos.shtml>

- Tecnología de administración de base de datos
- Usuarios
- Planeación de información y tecnología de modelaje

Características de las bases de datos:

Una base de datos contiene entidades de información que están relacionadas vía organización y asociación. La arquitectura lógica de una base de datos se define mediante un esquema que representa las definiciones de las relaciones entre las entidades de información. La arquitectura física de una base de datos depende de la configuración del hardware residente. Sin embargo, tanto el esquema (descripción lógica) como la organización (descripción física) deben adecuarse para satisfacer los requerimientos funcionales y de comportamiento para el acceso al análisis y creación de informes.

Ventajas en el uso de bases de datos:

La utilización de bases de datos como plataforma para el desarrollo de Sistemas de Aplicación en las Organizaciones se ha incrementado notablemente en los últimos años, se debe a las ventajas que ofrece su utilización como:

- Globalización de la información: permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.
- Eliminación de información inconsistente: si existen dos o más archivos con la misma información, los cambios que se hagan a éstos deberán hacerse a todas las copias del archivo de facturas.
- Compartir información.
- Mantener la integridad en la información: la integridad de la información es una de sus cualidades altamente deseable y tiene por objetivo que sólo se almacena la información correcta.
- Independencia de datos: Es la separación entre los programas y los datos que permite realizar modificaciones sin provocar cambios significativos en la

aplicación, es quizás la que más ha ayudado a la rápida proliferación del desarrollo de Sistemas de Bases de Datos.²³

Desventajas en el uso de bases de datos:

- El Sistema de Manejo de Base de Datos (DBMS) requiere de mucho espacio en disco duro y también requiere de mucha memoria principal (RAM) para poder correr adecuadamente.
- Debido a la cantidad de operaciones y a las capacidades del DBMS, se convierte en un producto complejo de entender. Esto requiere que los programadores y los analistas deban tomar cursos que los adiestren para poder comprender las capacidades y limitaciones del DBMS
- Los productos de Bases de Datos (Oracle, DB2, Etc.) son productos caros.
- El adquirir un producto de Base de Datos, requiere a su vez adquirir equipo adicional para poder correr ese producto.
- Si un componente de la Base de Datos sufre un desperfecto, se detiene las operaciones del producto por completo.
- En caso de un accidente que corrompa la Base de datos, el proceso de recuperación y de devolver a la Base de Datos su estado anterior al problema, es mucho más complejo de ejecutar que en los sistemas tradicionales.²⁴

2.2.3.2 Conexión a través de Enterprise Library

Enterprise Library es una librería de módulos comunes de software, reutilizable y ampliable, cuyo objetivo es permitir un desarrollo más rápido y fácil de las aplicaciones en el entorno de la plataforma Microsoft .Net.²⁵

²³ <http://www.monografias.com/trabajos7/bada/bada.shtml>

²⁴ http://www.uprb.edu/profesor/ntorres/base_de_datosVentajasDesventajas.htm

²⁵ [www.microsoft.com/spain/enterprise/perspectivas/Enterprise Library la evolucion_NET.htm](http://www.microsoft.com/spain/enterprise/perspectivas/Enterprise%20Library%20la%20evolucion_NET.htm)

Está basada en la ACA.Net (Avanade Connected Architecture) y en los patrones y prácticas de los Application Blocks de Microsoft, de los que puede considerarse una evolución. Una de las grandes ventajas que ofrece es su escalabilidad; por ello, es considerada como una arquitectura prescriptiva.²⁵

Una de las mayores ventajas que encuentran los usuarios de los distintos bloques de Enterprise Library es la posibilidad de acceder a unos elementos que están contrastados por su uso en docenas de proyectos y que no plantean ningún tipo de problemas.

Esta seguridad reduce el time to market, el número de líneas de código que hay que implementar para las aplicaciones y por lo tanto, el costo de desarrollo de un sistema.²⁵

Objetivos

- Coherencia

Todos los bloques de aplicación poseen patrones de diseño y criterios de aplicación coherentes.

- Extensibilidad

Todos los bloques de aplicación incluyen puntos definidos de extensibilidad que permiten a los desarrolladores personalizar el comportamiento de los bloques de la aplicación mediante la adición de su propio código.

- Integración

Los bloques de aplicación de la Enterprise Library están diseñados para trabajar bien ya sea juntos o individualmente.

- Facilidad de uso

²⁵ [www.microsoft.com/spain/enterprise/perspectivas/Enterprise Library la evolucion_NET.htm](http://www.microsoft.com/spain/enterprise/perspectivas/Enterprise%20Library%20la%20evolucion_NET.htm)

Enterprise Library ofrece numerosas mejoras para su uso, incluyendo una herramienta de configuración gráfica, un sencillo procedimiento de instalación, documentación y muestras más claras y completas.²⁶

Enterprise library es una aplicación muy interesante por las facilidades que otorga al hacer una conexión, cambiando los valores. Esta es una herramienta de fácil uso que permite también proteger la conexión como un tipo de seguridad adicional.²⁷

Es un conjunto de proyectos que contienen código fuente que podemos compilar como dll's de .Net y utilizarlas en varios proyectos o sistemas, con la finalidad de cumplir con ciertos patrones que se recomiendan en las "buenas prácticas", de hecho, podríamos modificar estos bloques de código para ajustarlos a cada necesidad del sistema.

Enterprise library contiene los siguientes bloques de aplicación:

- Caching Application Block: Incorporar caché local en las aplicaciones.
- Cryptography Application Block: Permite incorporar técnicas de encriptación simétrica en las aplicaciones.
- Data Access Application Block: Incorpora funcionalidad estándar de base de datos a las aplicaciones.
- Exception Handling Application Block: Crear una estrategia consistente para procesar excepciones que ocurren a través de las distintas capas arquitectónicas de la aplicación.
- Logging Application Block: Puede ser usado para agregar funcionalidad de bitácora a una aplicación.
- Policy Injection Application Block: Implementa políticas de intercepción que pueden ser usadas para facilitar la implementación a lo largo de una aplicación de funcionalidades como manejo de excepciones, validación, etc.

²⁶ www.microsoft.com/en-us/library/cc467894.aspx

²⁷ www.netveloper.com/contenido2.aspx?IDC=235_0

- Security Application Block: Como su nombre lo indica, sirve para incorporar funcionalidad de autorización y cacheo de seguridad en una aplicación.
- Validation Application Block: Puede ser usado para crear “reglas de validación” para objetos de negocios a lo largo de una aplicación.²⁸

²⁸ <http://hancocchi.net/enterprise-library-31/>

CAPITULO 3.

3.1 ANÁLISIS, DISEÑO Y DESARROLLO DEL SISTEMA

Para el desarrollo del proyecto de mantenimiento se utilizarán varias herramientas de software y arquitecturas.

El lenguaje de desarrollo es C# en Visual Studio 2005, tomando como herramienta Windows forms con Clickonce que le permitirá a la aplicación manejarse como un sistema cliente servidor.

En cuanto a la arquitectura se aplicará el concepto de la programación por capas, con todos los métodos utilizados en el Sistema publicados en un servicio Web.

La metodología de desarrollo de software que se utilizará es RUP (Proceso Unificado Racional), la cual permite realizar el análisis, implementación y documentación de un sistema orientado a objetos.

Esta metodología se adapta a las necesidades de cada empresa ya que no tiene pasos firmemente establecidos sino que es adaptable a cada contenido.

Se caracteriza porque es iterativo, está centrado en la arquitectura y se guía por los casos de uso.

Las fases en la cuales se divide esta metodología son:

1) Inicio

Determina la visión del proyecto, la comprensión del problema y la tecnología, delimita el proyecto, elimina los riesgos críticos y establece una línea base de la arquitectura.

2) Elaboración

Se orienta al desarrollo de la línea base de la arquitectura, su objetivo principal es determinar la arquitectura óptima a utilizar.

3) Construcción

En esta fase se ejecuta la construcción del producto a través de una serie de iteraciones.

4) Transición

En esta fase se garantiza la calidad del producto previa a la entrega al cliente

3.1.1 ANÁLISIS Y DISEÑO DE LA BASE DE DATOS

El motor de base de datos que se empleará es SQL 2005. La conexión con la base de datos se dará por una herramienta llamada Enterprise library propia de Microsoft que permite realizar la conexión con el motor de base de datos por medio de ADO, los datos se cargarán en datasets y después de su manipulación se los mostrará en la capa de presentación, el proceso de manipulación de la información será el siguiente:

Se realiza la petición por medio de una sentencia SQL en la capa de datos.

El aplicativo se comunica con la base de datos mediante enterprise library y ejecuta la sentencia SQL en el motor de la base de datos.

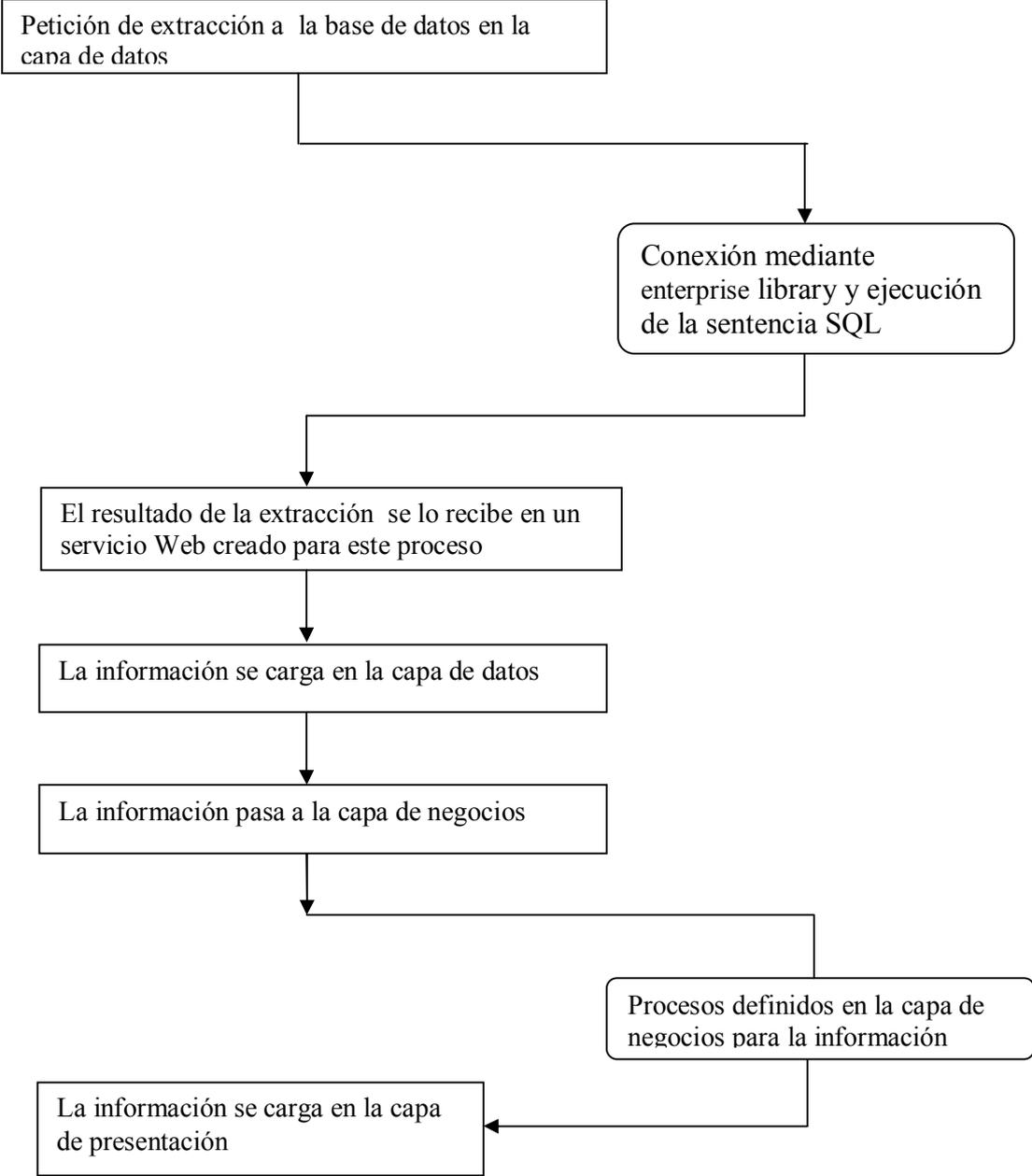
La información se recibe en un servicio Web el cual tiene los datos en formato xml.

Según las necesidades se carga la información en la capa de datos ya sea en un data set ó en una lista según sea el caso y se envía a la capa de negocio.

Ya en la capa de negocio la información pasa por los procesos que se han definido en esta capa para posteriormente enviarla a la capa de presentación.

En la capa de presentación o capa de usuario la información ya llega lista para la visualización del usuario o en su defecto tiene que pasar procesos mínimos para su posterior presentación.

Procesamiento de la información



Fig#1. Procesamiento de la información
Fuente: Los autores

3.1.1.1 Modelo Lógico

Nombre base de datos	TVJ_Manprecor					
Tabla: TVJ_descripción_estado	Tabla que contiene la descripción de los estados de una solicitud					
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_destest(PK)	código de manejo para la base de datos	Int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese un código de estado
descrip_destest	descripción del código de estado	Varchar	ingresado por usuario		s	
Tabla: TVJ_empleado	Tabla que contiene la información personal y laboral de cada empleado					
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_empleado(PK)	código de manejo para la base de datos	Int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese un nuevo empleado
ced_emp	cédula de identidad del empleado	Varchar	ingresado por usuario		s	
nom_emp	nombre del empleado	Varchar	ingresado por usuario		s	
dir_emp	dirección domiciliaria del empleado	Varchar	ingresado por usuario		s	
tel_emp	teléfono de contacto del empleado	Varchar	ingresado por usuario		s	
car_emp	tipo de cargo que desempeña el empleado	varchar	ingresado por usuario		s	
Estado	indica si el empleado se encuentra activo o inactivo a través de un número que indica el estado	int	ingresado por usuario		s	
Tabla: TVJ_equides	Tabla que contiene la descripción y las características de todos los equipos					
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_equides(PK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese un equipo

nom_equides	nombre del equipo ingresado	varchar	ingresado por usuario		s	
grp_equides	grupo al cual pertenece el equipo	varchar	combo		s	
subgrp_equides	subgrupo del equipo si lo hubiere	varchar	combo		s	
equibas_equides	equipo base al cual pertenece el equipo ingresado	varchar	combo		s	
feccom_equides	fecha de compra del equipo	varchar	ingresado por usuario	aaaa/mm/dd	s	
garan_equides	garantía en tiempo del equipo	char	ingresado por usuario	numero /anos	s	
dimen_equides	dimensiones del equipo	varchar	ingresado por usuario	numero/unidad	s	
pes_equides	peso del equipo en las unidades especificadas	varchar	ingresado por usuario	numero/unidad	s	
capa_equides	capacidad de equipo en las unidades especificadas	varchar	ingresado por usuario	numero/unidad	s	
marc_equides	marca del equipo	varchar	ingresado por usuario		s	
priori_equides	prioridad del equipo en el mantenimiento	varchar	combo		s	
identi_equides	identificador del equipo	varchar	ingresado por usuario		s	
tipoidenti_equides	tipo de identificador del equipo	varchar	combo		s	
cod_tipoequi	código del tipo de equipo	int	combo		s	
cod_proveequi	código del proveedor del equipo	int	combo		s	
cod_ubicacion	código de localización del equipo	int	combo		s	
aux1_equides	campo auxiliar según tipo de equipo	varchar	ingresado por usuario		s	
aux2_equides	campo auxiliar según tipo de equipo	varchar	ingresado por usuario		s	
aux3_equides	campo auxiliar según tipo de equipo	varchar	ingresado por usuario		s	
aux4_equides	campo auxiliar según tipo de equipo	varchar	ingresado por usuario		s	
aux5_equides	campo auxiliar según tipo de equipo	varchar	ingresado por usuario		s	
aux6_equides	campo auxiliar según tipo de equipo	varchar	ingresado por usuario		s	
aux7_equides	campo auxiliar según tipo de equipo	varchar	ingresado por usuario		s	

aux8_equides	campo auxiliar según tipo de equipo	varchar	ingresado por usuario		s	
Estado	indica el estado del equipo	varchar	ingresado por usuario		s	
Tabla: TVJ_equima Tabla que contiene las imagenes de los equipos ingresados						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_equima(PK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese una imagen
desima_equima	path de la imagen ingresada	varchar	ingresado por usuario		s	
cod_equides(FK)	código del equipo al cual pertenece la imagen	int	proporcionado por el sistema	'001	n	
Tabla: TVJ_equidoc Tabla que contiene los documentos relacionados a los equipos						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_equidoc(PK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese un documento
desdoc_equidoc	path del documento ingresado	varchar	ingresado por usuario		s	
cod_equides(FK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	
Tabla: TVJ_equifalla Tabla que contiene las relaciones entre el código de los equipos y sus correspondientes fallas.						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_equides	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	
cod_fallas	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	
Tabla: TVJ_equinsu Tabla que contiene la información de la relación de cada equipo y sus respectivos insumos.						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_equides	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	
cod_insumo	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	

Tabla: TVJ_equirut						
Tabla que contiene la información del los equipos y sus respectivas rutinas						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_equides	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	
cod_rutina	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	
Tabla: TVJ_estadosolicitud						
Tabla que contiene toda la información de una solicitud						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_solicitud(PK)	código de manejo para la base de datos	Int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese una nueva solicitud
Descripciónsolicitud	descripción de la solicitud	Varchar	ingresado por usuario		S	
cod_usuario	código del usuario que ingresa la solicitud	Int	proporcionado por el sistema	'001	S	
cod_desest	código del estado de la solicitud	Int	proporcionado por el sistema	'001	S	
fecha_solicitud	fecha de ingreso de la solicitud	Varchar	ingresado por usuario	aaaa/mm/dd	N	
Tabla: TVJ_fallas						
Tabla que contiene la información de una falla						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_fallas(PK)	código de manejo para la base de datos	Int	proporcionado por el sistema	'001	N	es el código que le dará el sistema cuando se ingrese una nueva falla
desc_fallas	descripción de la falla ingresada	Varchar	ingresado por usuario		S	
soluc_fallas	solución a la falla ingresada	varchar	ingresado por usuario		S	
Tabla: TVJ_insumo						
Tabla que contiene la información de los insumos						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_insumo(PK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese un insumo
nombre_insumo	nombre del insumo	varchar	ingresado por usuario		s	

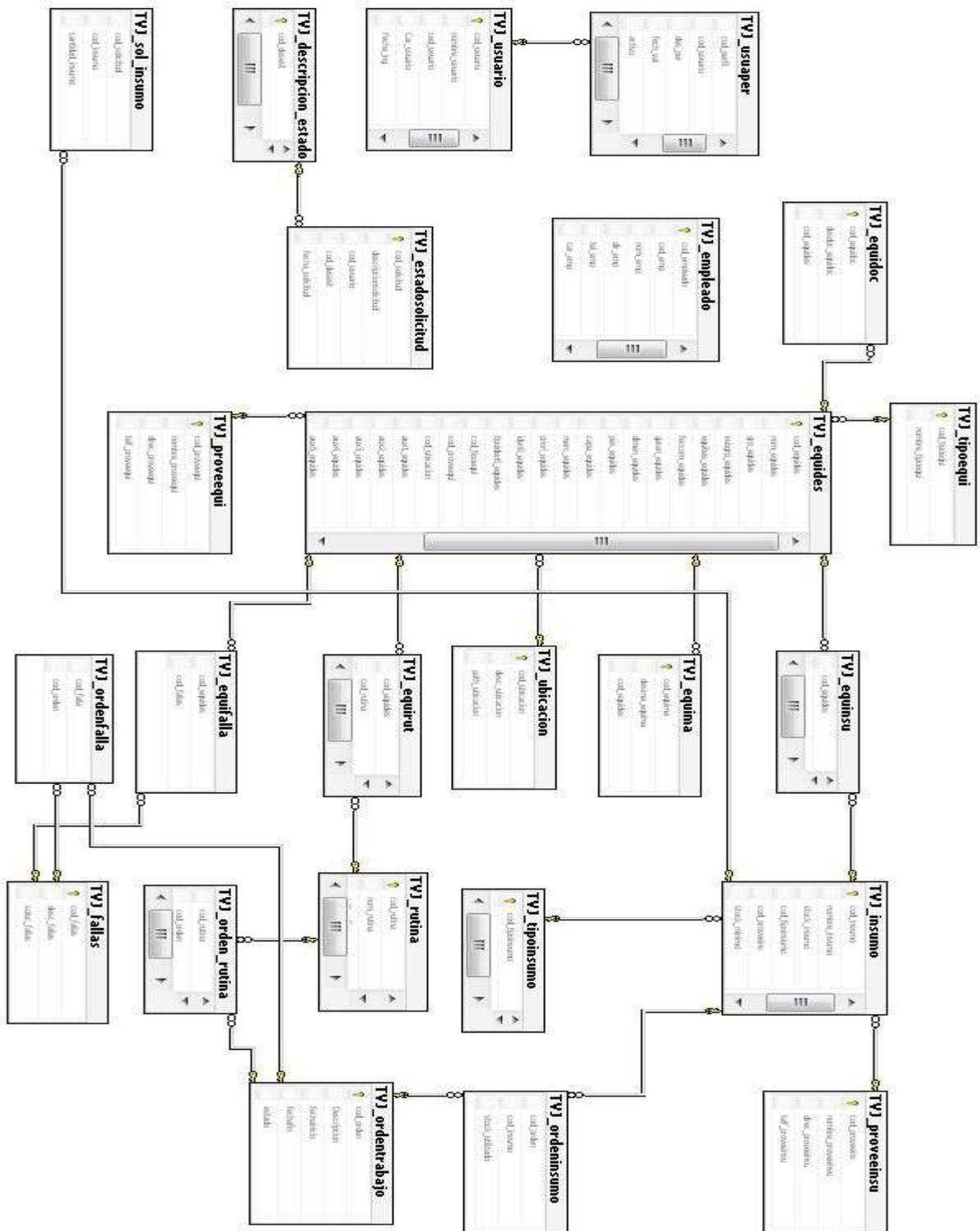
stock_insumo	cantidad en stock del insumo	numeric	ingresado por usuario		s	
cod_tipoinsumo	código del tipo de insumo	nchar	proporcionado por el sistema		n	
cod_proveeins	código del proveedor del insumo	nchar	proporcionado por el sistema		n	
stock_minimo	cantidad mínima del insumo	numeric	ingresado por usuario		s	
Parámetro	unidad de medida	varchar	ingresado por usuario		s	
Tabla: TVJ_ordeninsumo						
Tabla que contiene la información acerca de las ordenes de compra para insumos						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_orden	código de la orden	int	proporcionado por el sistema	'001	n	
cod_insumo	código del insumo	int	proporcionado por el sistema	'001	n	
stock_utilizado	cantidad de insumo utilizado	numeric	proporcionado por el sistema		n	
Tabla: TVJ_ordentrabajo						
Tabla que contiene la información de las ordenes de trabajo generadas						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_orden(PK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese una orden de trabajo
Descripción	descripción de la orden de trabajo	varchar	ingresado por usuario		s	
FechaInicio	fecha de inicio de la orden	varchar	ingresado por usuario		s	
FechaFin	fecha de fin de la orden	varchar	ingresado por usuario		s	
Estado	indica el estado en el que se encuentra la orden	varchar	ingresado por usuario		s	
Tabla: TVJ_proveeinsu						
Tabla que contiene la información de los proveedores de insumos						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_proveeins(PK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese un proveedor de insumo

nombre_proveeinsu	nombre del proveedor de insumo	varchar	ingresado por usuario		s	
direc_proveeinsu	dirección del proveedor de insumo	varchar	ingresado por usuario		s	
telf_proveeinsu	teléfono de contacto del proveedor de insumo	nchar	ingresado por usuario		s	
Tabla: TVJ_proveequi						
Tabla que contiene la información de los proveedores de los equipos						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_proveequi(PK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese un proveedor de equipos
nombre_proveequi	nombre del proveedor de equipos	varchar	ingresado por usuario		s	
direc_proveequi	dirección del proveedor de equipos	varchar	ingresado por usuario		s	
telf_proveequi	teléfono de contacto del proveedor de equipos	nchar	ingresado por usuario		s	
Tabla: TVJ_rutina						
Tabla que contiene la información de las rutinas						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_rutina(PK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese una rutina
nom_rutina	nombre de la rutina	varchar	ingresado por usuario		s	
path_rutina	path/dirección de ubicación del documento de la rutina	varchar	ingresado por usuario		s	
Tabla: TVJ_sol_insumo						
Tabla que contiene la información de los pedidos para solicitar que se compre insumos						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_solicitud	código de la solicitud	int	proporcionado por el sistema	'001	s	es el código que le dara el sistema cuando se ingrese una solicitud de insumo
cod_insumo	código del insumo	int	proporcionado por el sistema	'001	s	

cantidad_insumo	cantidad de insumo a comprar	numeric	ingresado por usuario		s	
Tabla: TVJ_tipoequi Tabla que contiene la información sobre los tipos de equipos existentes						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_tipoequi(PK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese un tipo de equipo
nombre_tipoequi	nombre del tipo de equipo	varchar	ingresado por usuario		s	
Tabla: TVJ_tipoinsumo Tabla que contiene la información sobre los tipos de insumos existentes						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_tipoinsumo(PK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese un tipo de insumo
nombre_tipoinsumo	nombre del tipo de insumo	varchar	ingresado por usuario		s	
Tabla: TVJ_ubicacion Tabla que contiene la información sobre todas las ubicaciones (areas,departamentos) que tiene la empresa						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_ubicacion(PK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese una ubicación
desc_ubicacion	descripción de la ubicación	varchar	ingresado por usuario		s	
path_ubicacion	dirección exacta donde se encuentra la ubicación ingresada	varchar	ingresado por usuario		s	
Tabla: TVJ_usuaper Tabla que contiene la información sobre los perfiles que tiene cada usuario						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_perfil	código del perfil	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese el perfil de un usuario
cod_usuario	código del usuario	int	proporcionado por el sistema	'001	n	

			sistema			
des_per	descripción del perfil	nchar	ingresado por usuario		s	
feh_val	fecha máxima de validez del perfil	nchar	ingresado por usuario	aaaa/mm/dd	s	
Activo	estado del usuario en la base de datos	numeric	ingresado por usuario		s	
Tabla: TVJ_usuario						
Tabla que contiene la información sobre los usuarios creados para acceder al sistema						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_usuario(PK)	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese un usuario
nombre_usuario	nombre del usuario	varchar	ingresado por usuario		s	
ced_usuario	cédula del usuario	varchar	ingresado por usuario		s	
Car_usuario	cargo que desempeña el usuario	varchar	ingresado por usuario		s	
Fecha_ing	fecha de ingreso del usuario	varchar	ingresado por usuario	aaaa/mm/dd	s	
Estado	estado del usuario en la base de datos	numeric	ingresado por usuario		s	
Tabla: TVJ_orden_rutina						
Tabla que contiene la información sobre la relación entre la rutina y la orden generada						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_rutina	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese un usuario
cod_orden	código de manejo para la base de datos	int	proporcionado por el sistema		n	
Tabla: TVJ_ordenfalla						
Tabla que contiene la información sobre la relación entre la falla y la orden generada						
Campos	Descripción	Tipo dato	Origen Dato	Formato	Acepta Nulos	Observaciones
cod_falla	código de manejo para la base de datos	int	proporcionado por el sistema	'001	n	es el código que le dará el sistema cuando se ingrese un usuario
cod_orden	código de manejo para la base de datos	int	proporcionado por el sistema		n	

3.1.1.2 Modelo Físico



Fig#2. Modelo Físico Base de datos

Fuente: Los autores

3.1.2 ANÁLISIS DEL SISTEMA

La arquitectura que se utilizará es el concepto de programación por capas, utilizando métodos que se publicarán en un servicio Web.

Para cada módulo tanto el de Mantenimiento como el de Insumos se utilizará 3 capas en la programación: capa de presentación, capa de negocio y la capa de datos.

A parte de las 3 capas que se utilizarán, se tiene una capa adicional y se utilizará como auxiliar para contener los datasets que son las réplicas de las tablas de la base de datos.

Se tendrá además el servicio Web que se utilizará para la publicación de los métodos utilizados en el Sistema.

3.1.2.1 Funcionalidades del Sistema

El sistema a desarrollarse deberá poseer las siguientes funcionalidades:

- Será adaptable a empresas donde exista cualquier tipo de maquinaria que esté sujeta a mantenimiento, desde computadores e impresoras que se utilizan en una oficina hasta grandes máquinas industriales.
- Su funcionamiento permitirá documentar todo lo referente a información sobre los equipos como son imágenes, planos, atributos, manuales y localizaciones físicas.

- Además permitirá calendarizar las fechas en las cuales se debe realizar el mantenimiento preventivo a la maquinaria.
- Documentará los errores que se producen en la maquinaria para que sirvan de guía de solución en futuros mantenimientos.
- Poseerá un inventario de insumos, para calcular oportunamente el abastecimiento de los mismos.

El módulo de insumos permitirá ingresar solicitudes de pedidos de insumos que no se encuentren en stock y que se necesite para realizar mantenimientos, estas solicitudes seguirán un flujo establecido, en el cual primero deberán ser analizadas y aprobadas por el personal encargado, posterior a esto se realizará la compra y se definirá fechas de entrega de la mercadería.

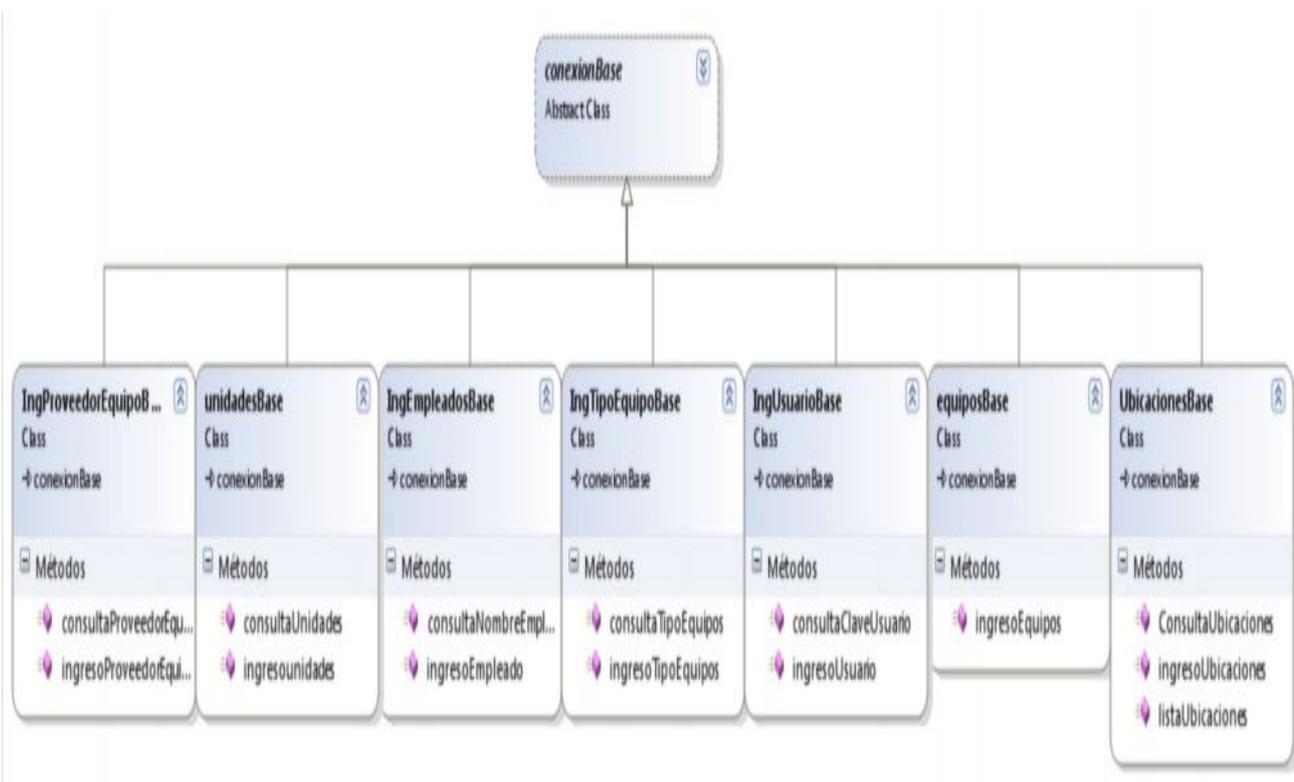
Una vez que los insumos se reciban en el departamento de Logística, se registrará la recepción de los mismos en el sistema, concluyendo así el flujo de la solicitud.

- Se podrá obtener una estadística de las fallas más frecuentes de la maquinaria a partir de la información obtenida.
- En el ingreso a la aplicación se deberá contar con un sistema de autenticación seguro para cada usuario, el cual deberá tener un método propio de encriptación para el almacenamiento de la clave personal. Únicamente se permitirá 3 intentos de clave errónea, si sobrepasa la clave automáticamente se bloqueará y solo un usuario Administrador podrá realizar el desbloqueo.

La clave por defecto al momento de la creación del usuario será el número de cédula de identidad y en el primer ingreso satisfactorio el sistema solicitará el cambio de clave al usuario.

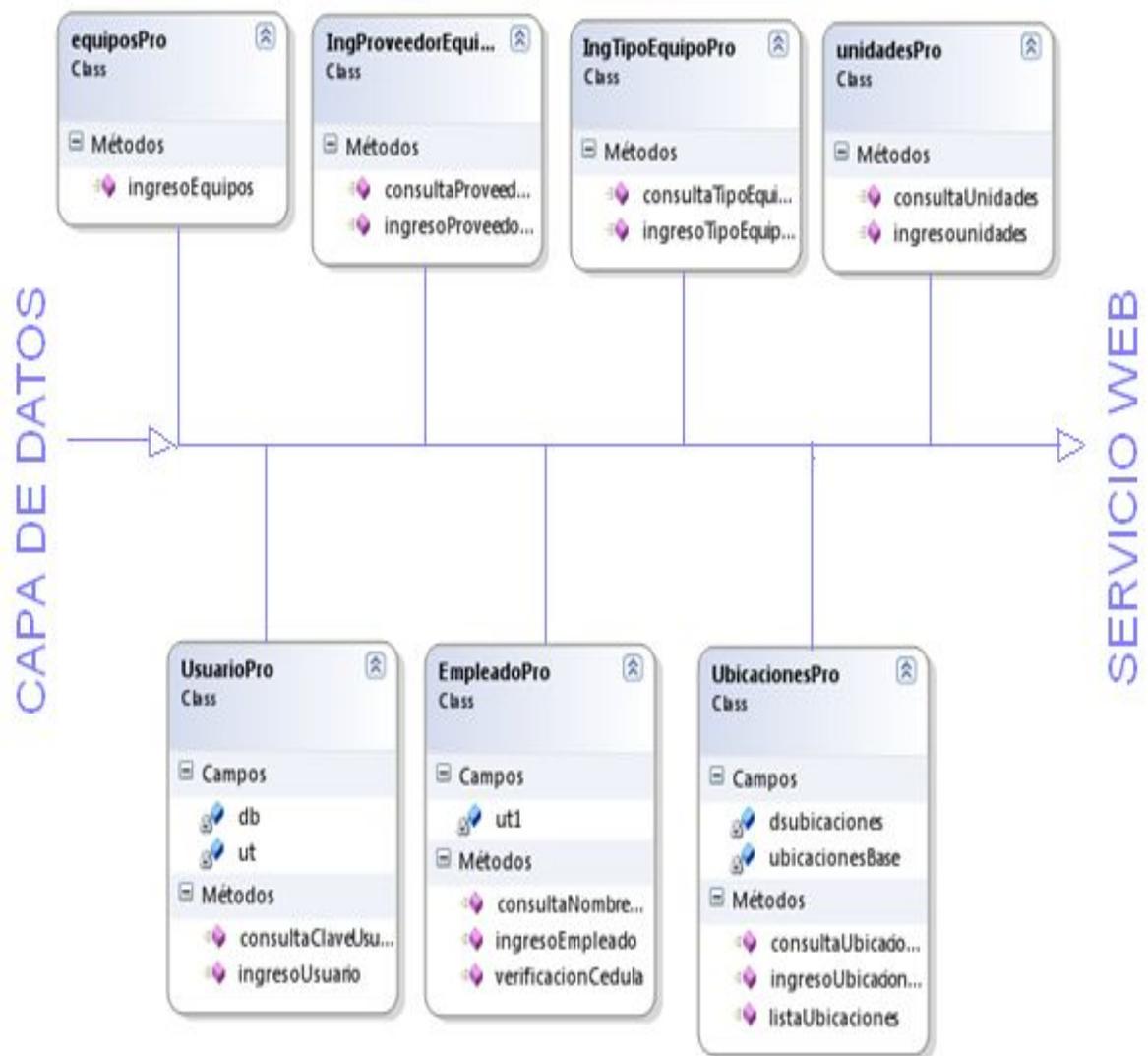
3.1.2.2 DISEÑO UML

3.1.2.2.1 Diagramas de Clases



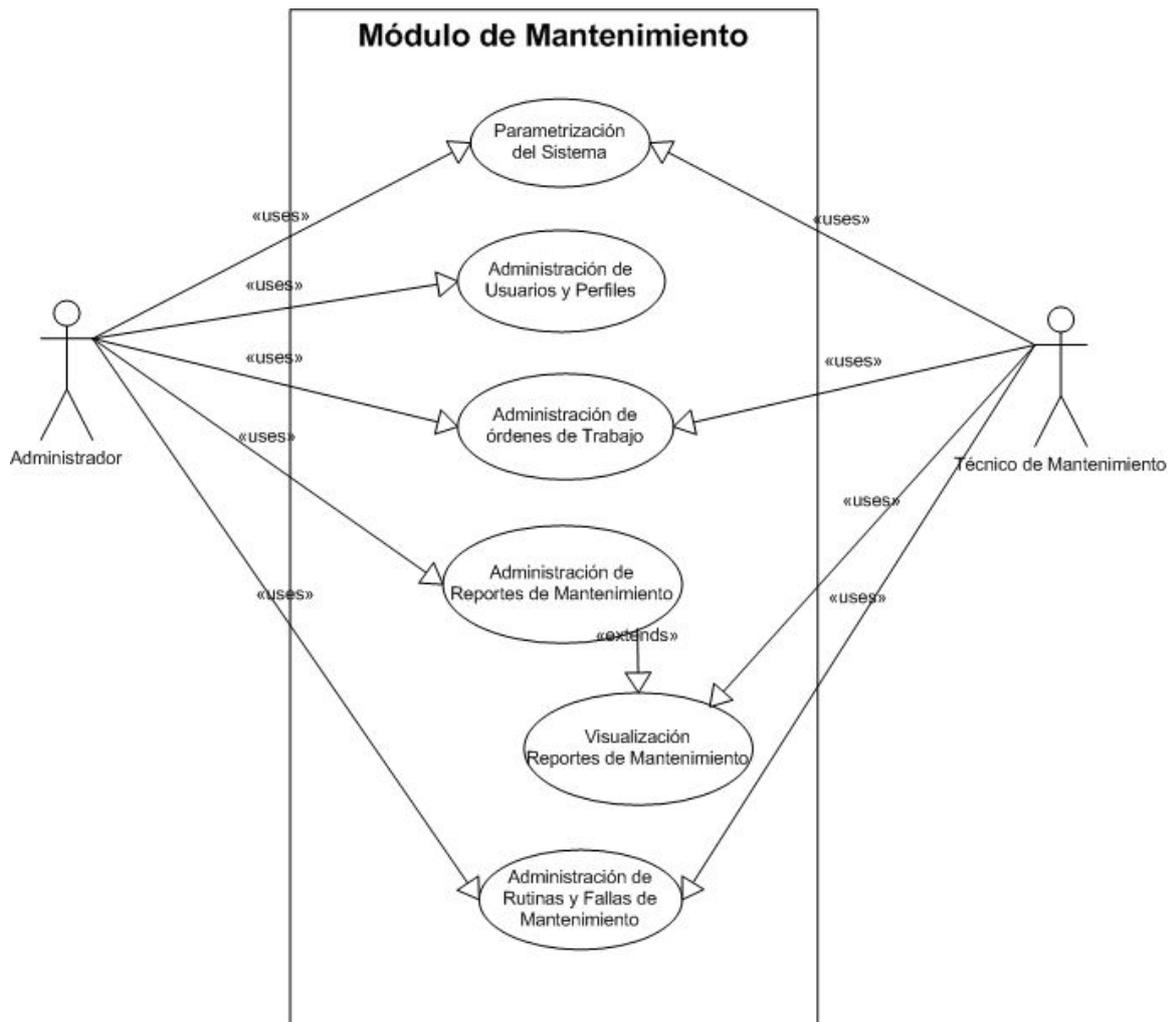
Fig#3. Diagrama de Clases de la Capa de Datos

Fuente: Los autores



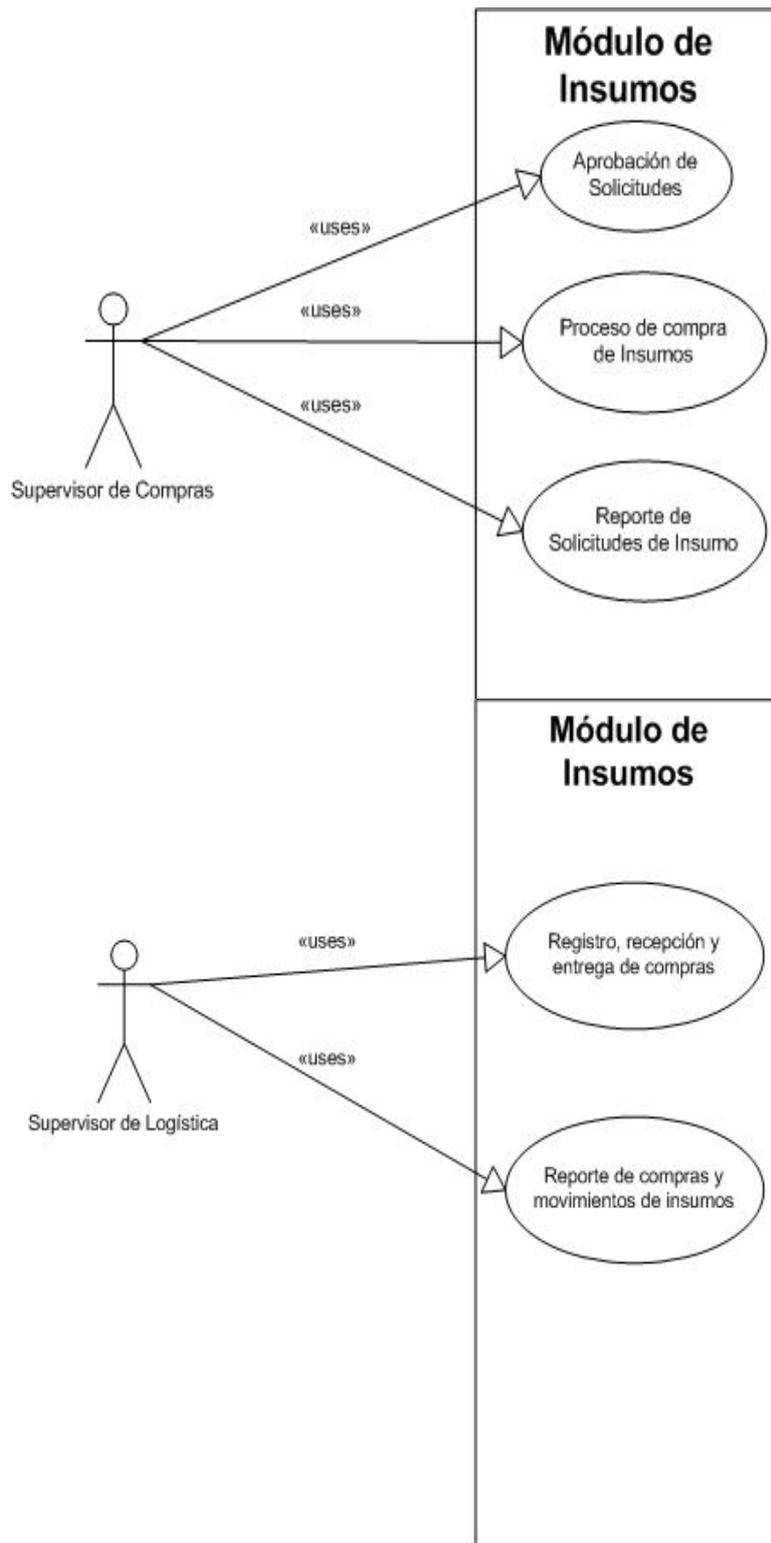
Fig#5. Diagrama de Clases de Capa de Procesos
Fuente: Los autores

3.1.2.2 Diagramas de Casos de Uso



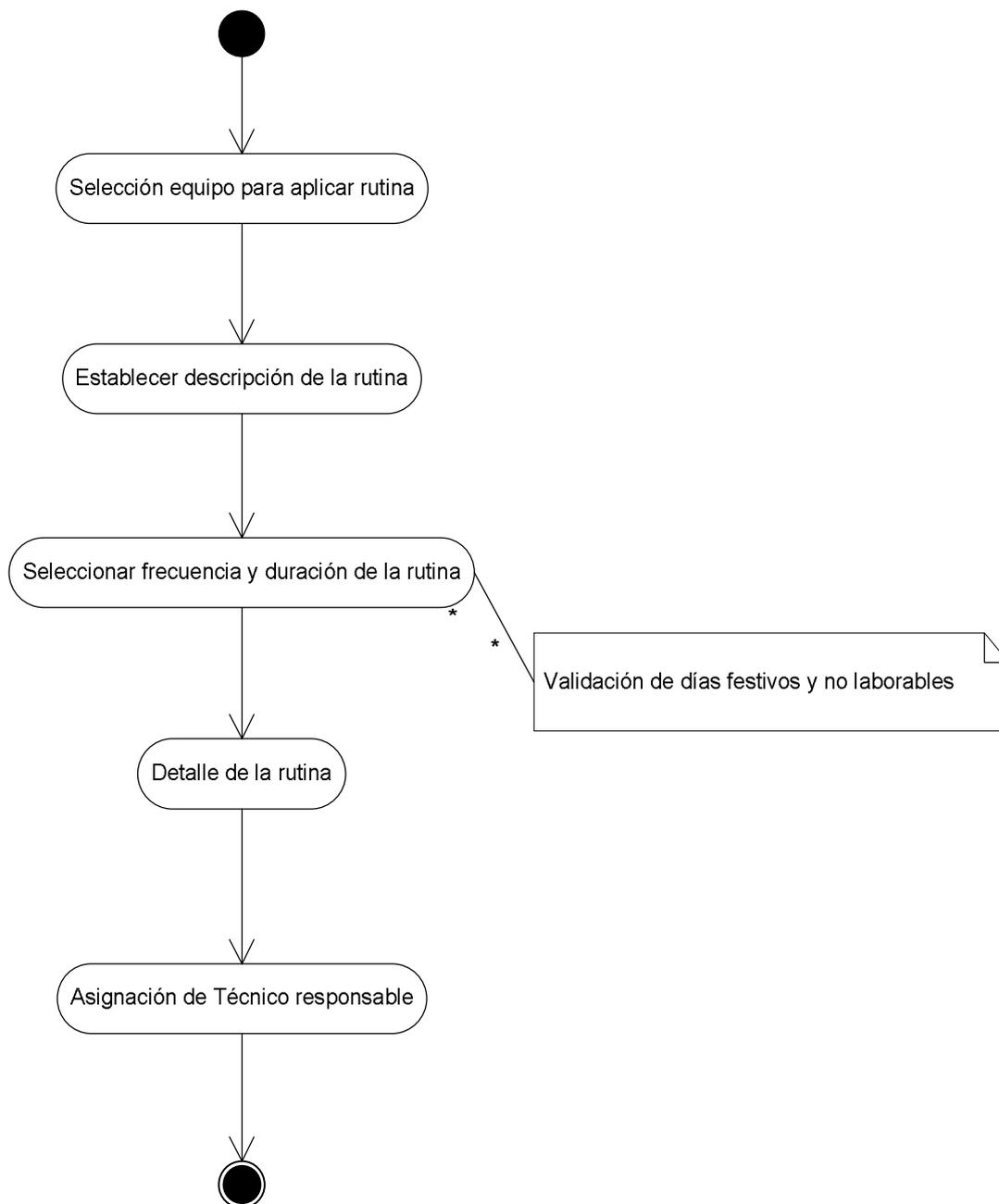
Fig#6. Diagrama de casos de uso del módulo de Mantenimiento

Fuente: Los autores



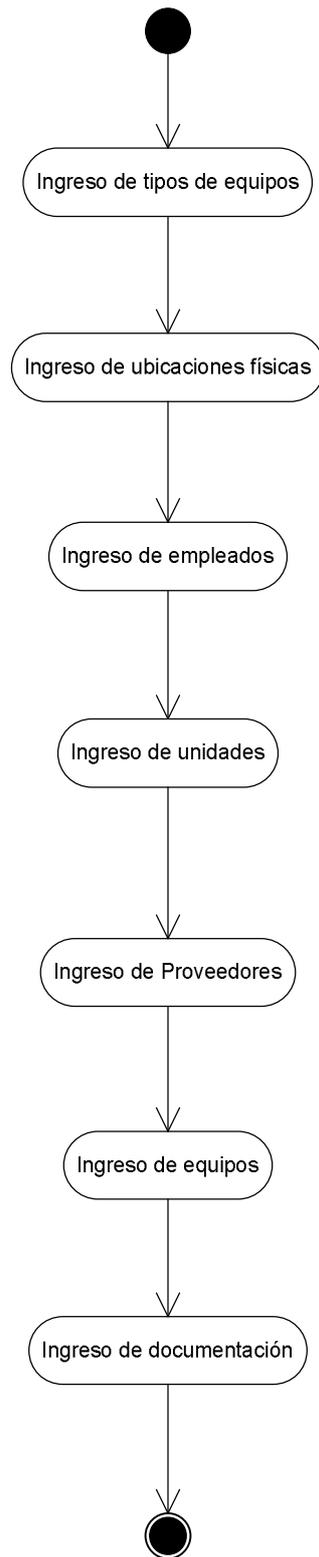
Fig#7. Diagrama de casos de uso del módulo de Insumos

Fuente: Los autores



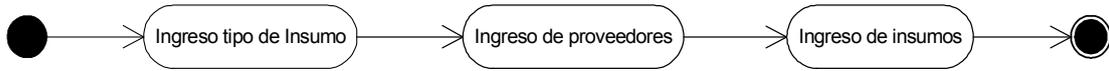
Fig#9. Diagrama de actividades de Rutinas de mantenimiento

Fuente: Los autores

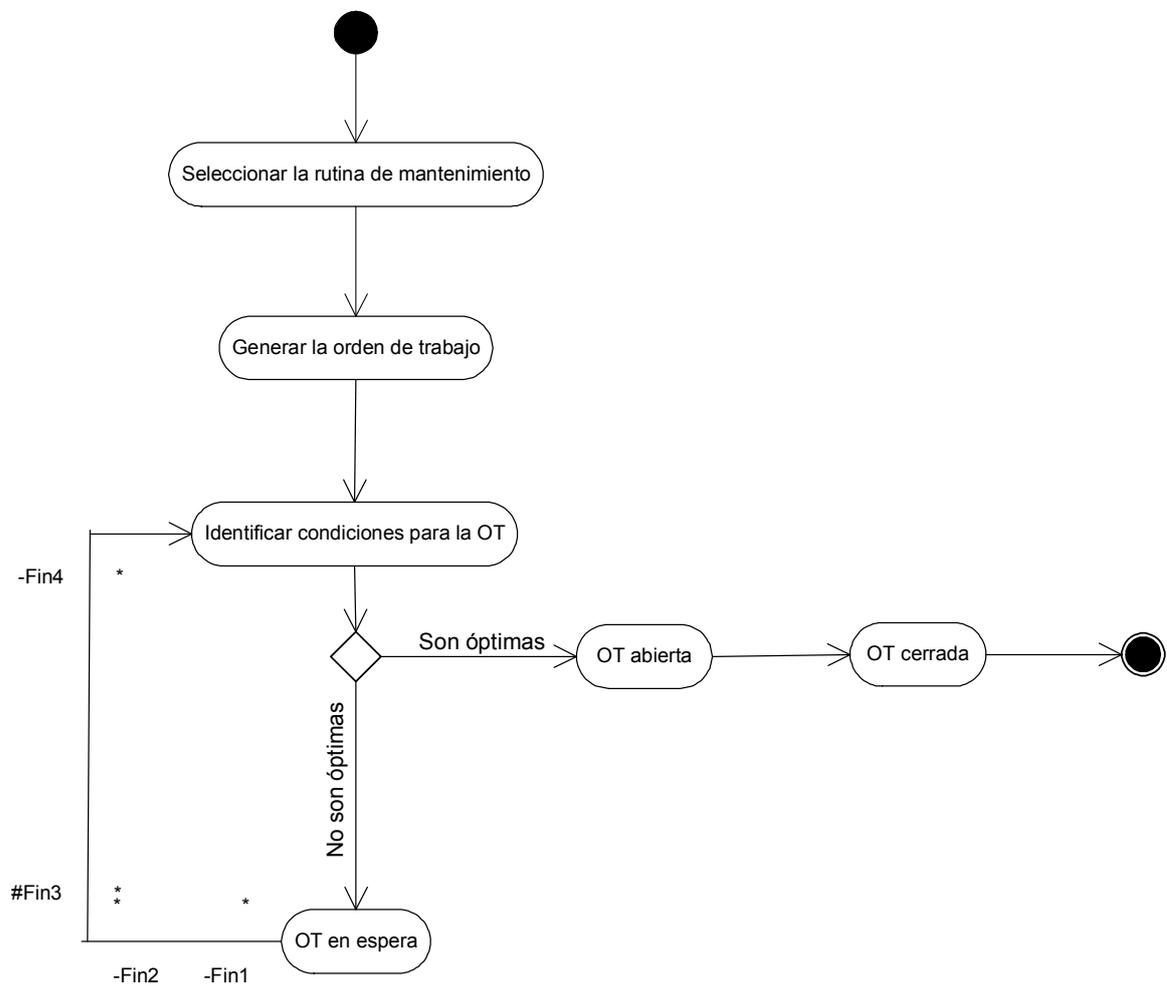


Fig#10. Diagrama de actividades de Parametrización Inicial módulo de Mantenimiento

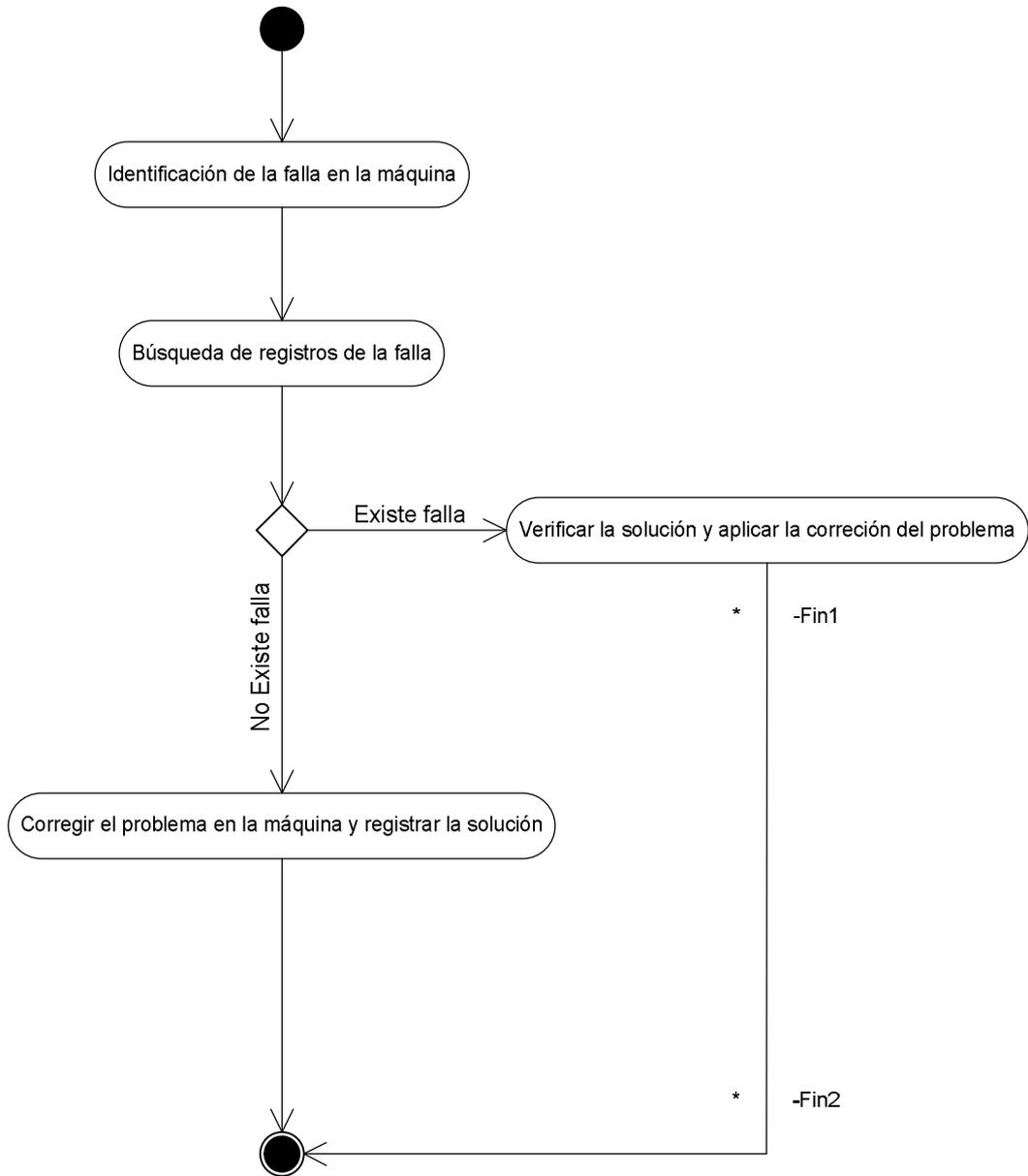
Fuente: Los autores



Fig#11. Diagrama de actividades Parametrización Inicial módulo de Insumos
Fuente: Los autores

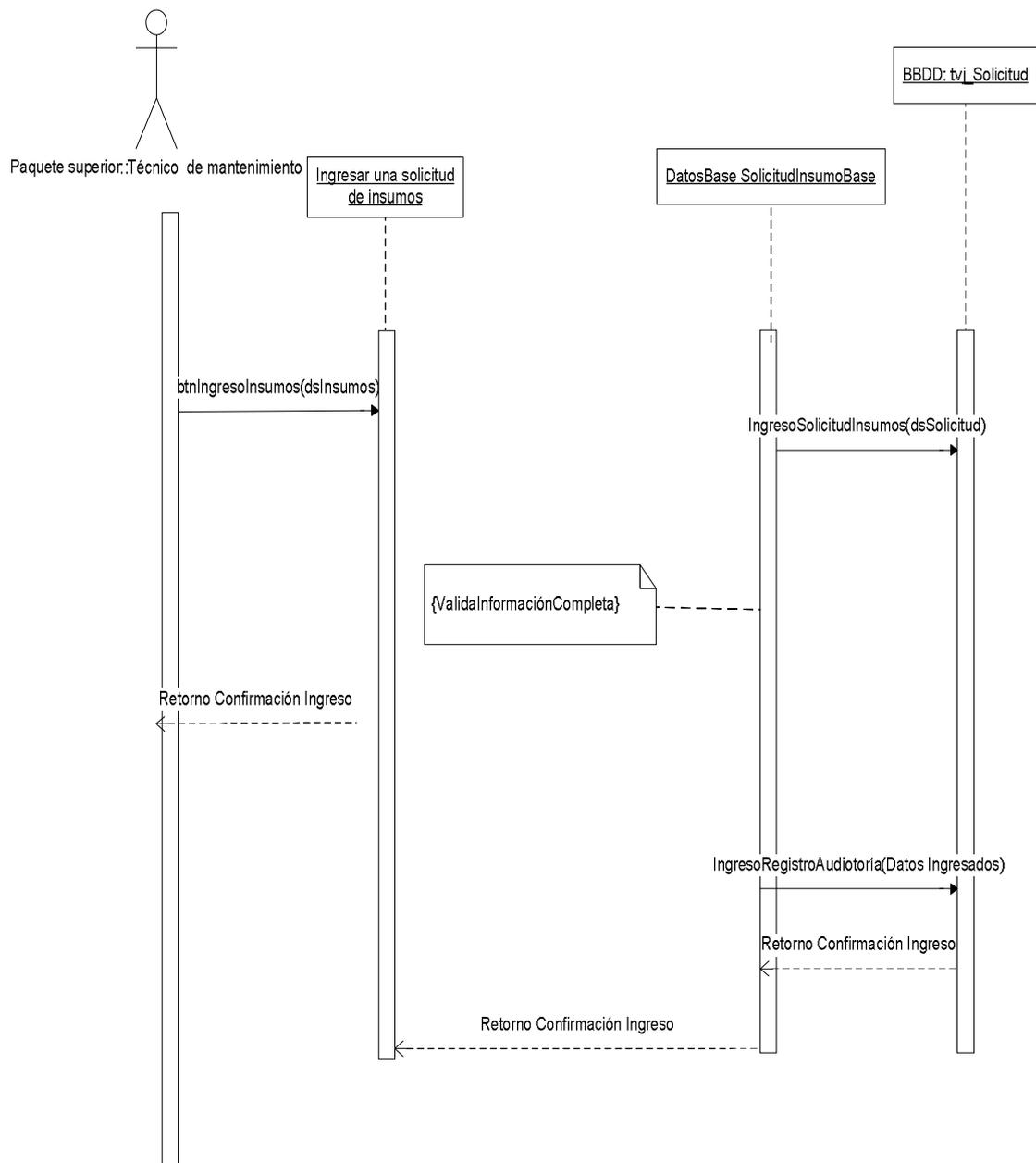


Fig#12. Diagrama de actividades de Órdenes de Trabajo
Fuente: Los autores



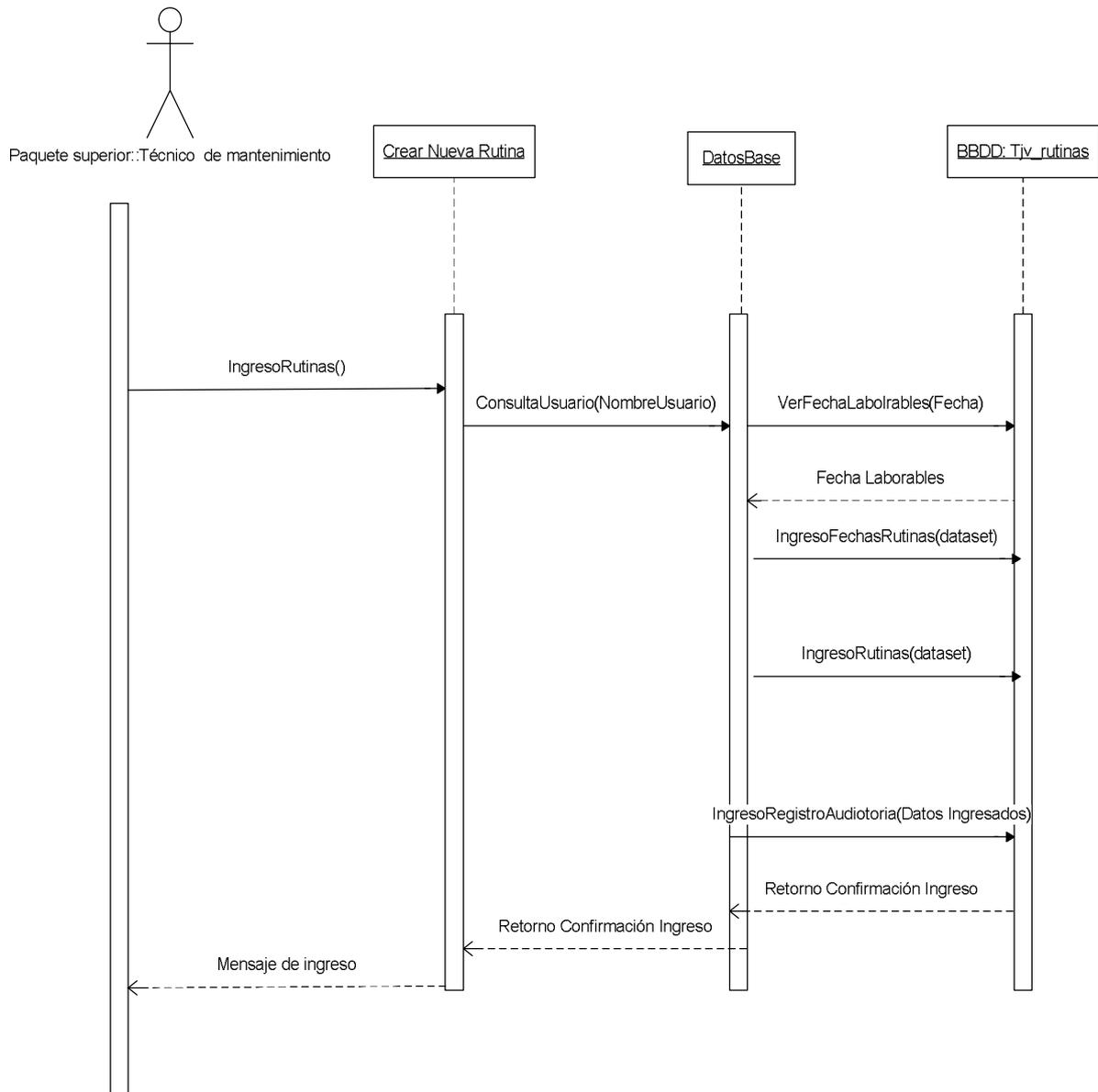
Fig#13. Diagrama de actividades de Fallas
 Fuente: Los autores

3.1.2.2.4 Diagramas de Secuencias



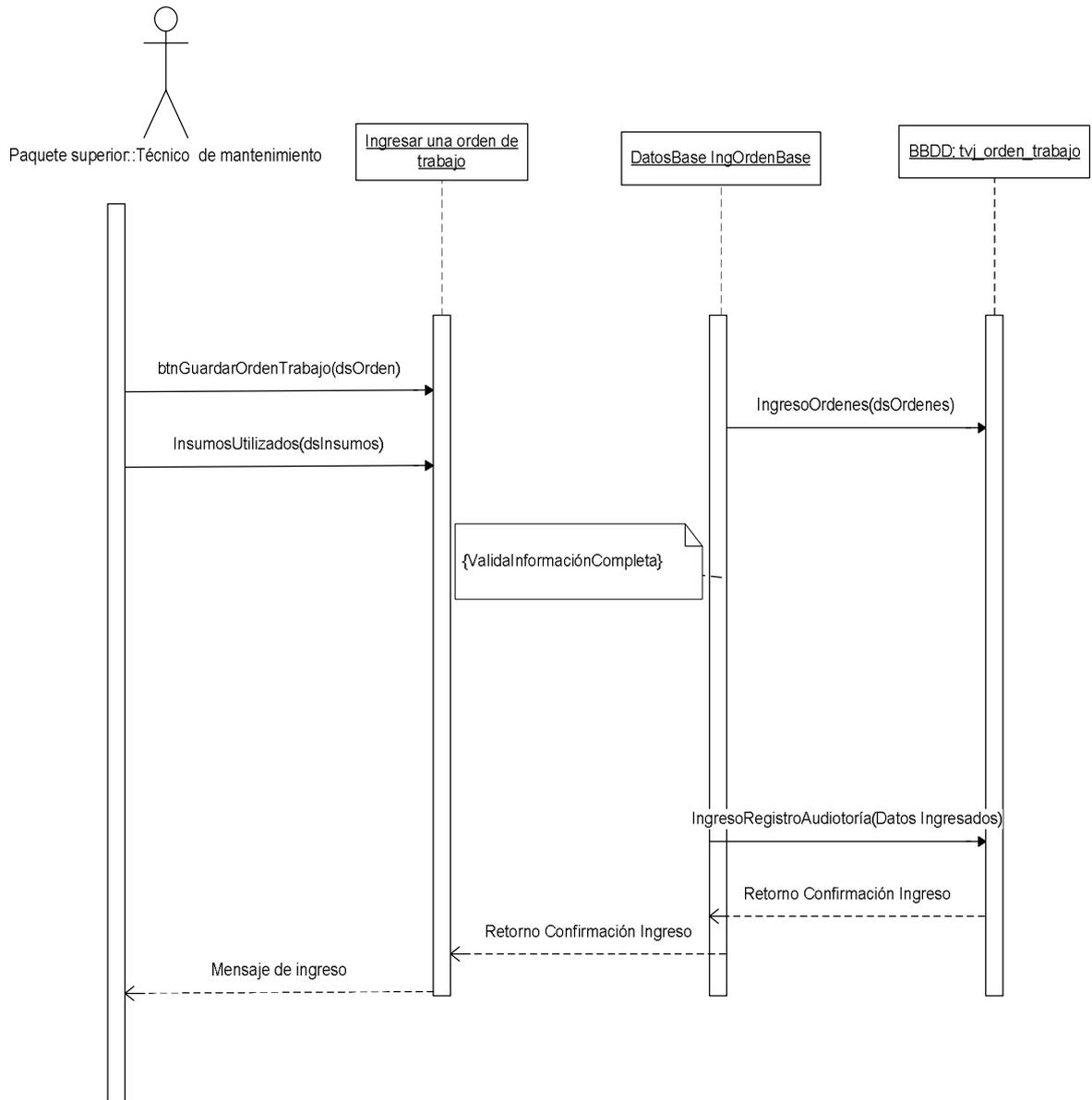
Fig#14. Diagrama de secuencia de Ingreso de Solicitudes

Fuente: Los autores

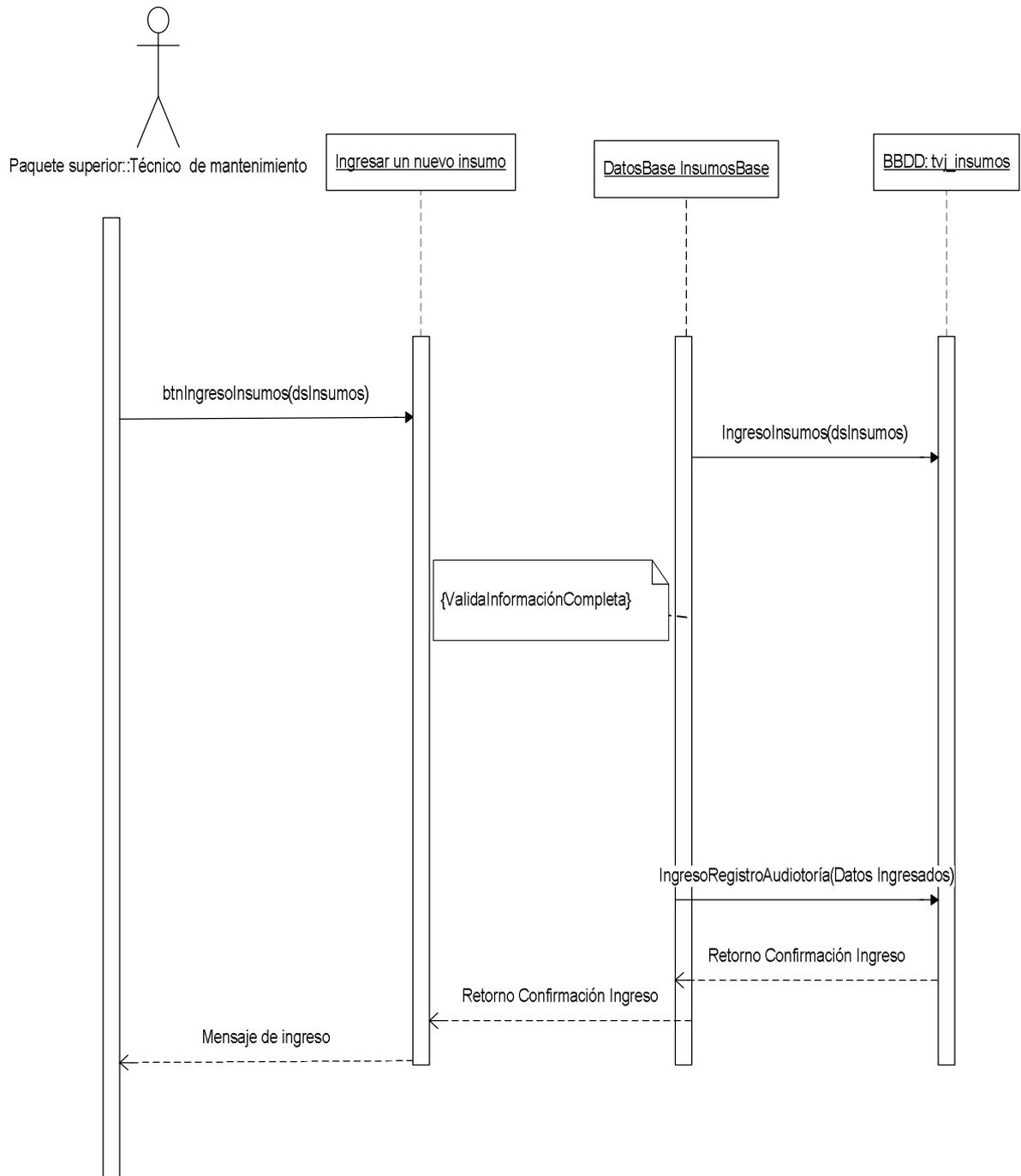


Fig#15. Diagrama de secuencia de Ingreso de Rutinas

Fuente: Los autores

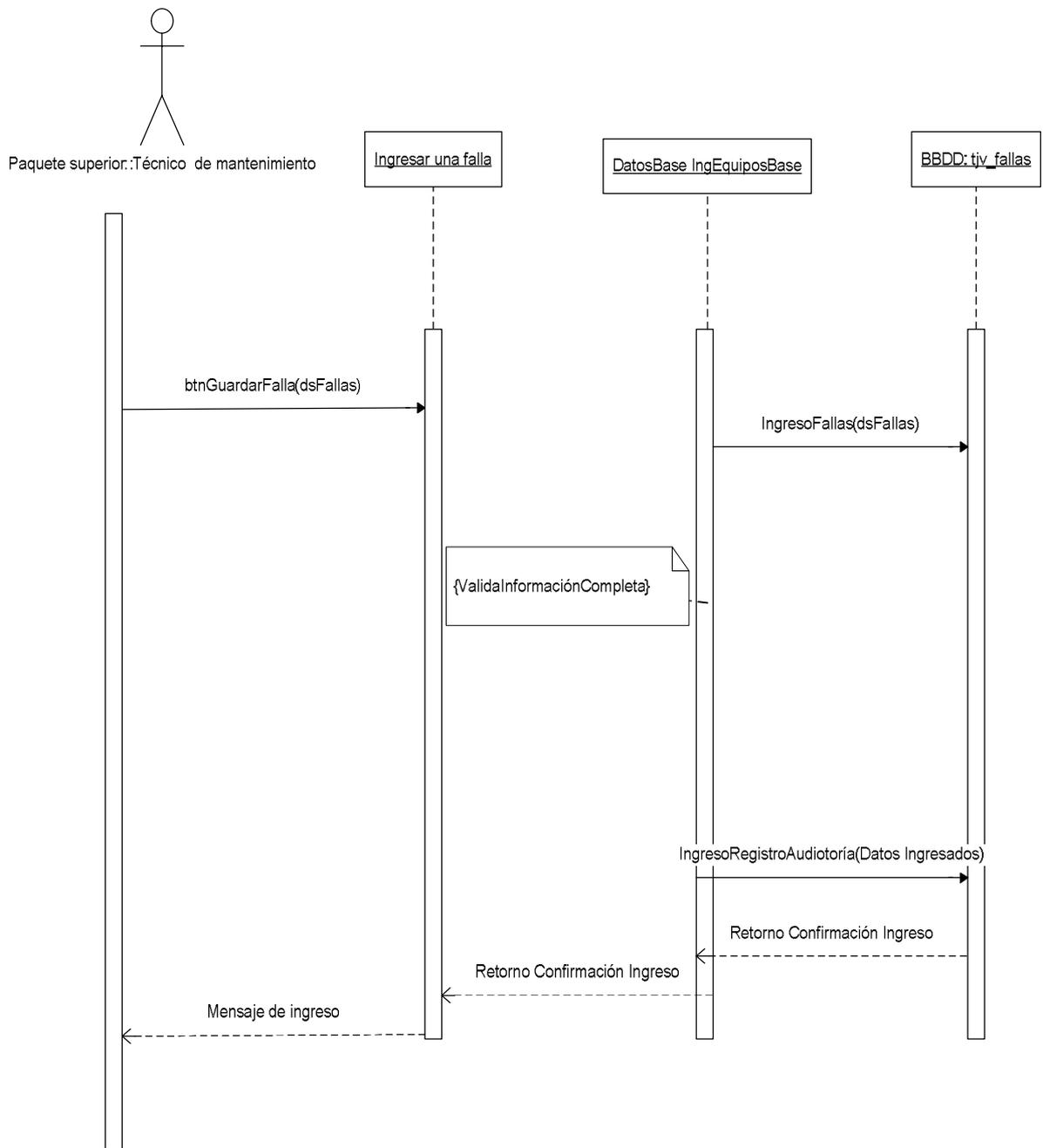


Fig#16. Diagrama de secuencia de Ingreso de Ordenes
Fuente: Los autores



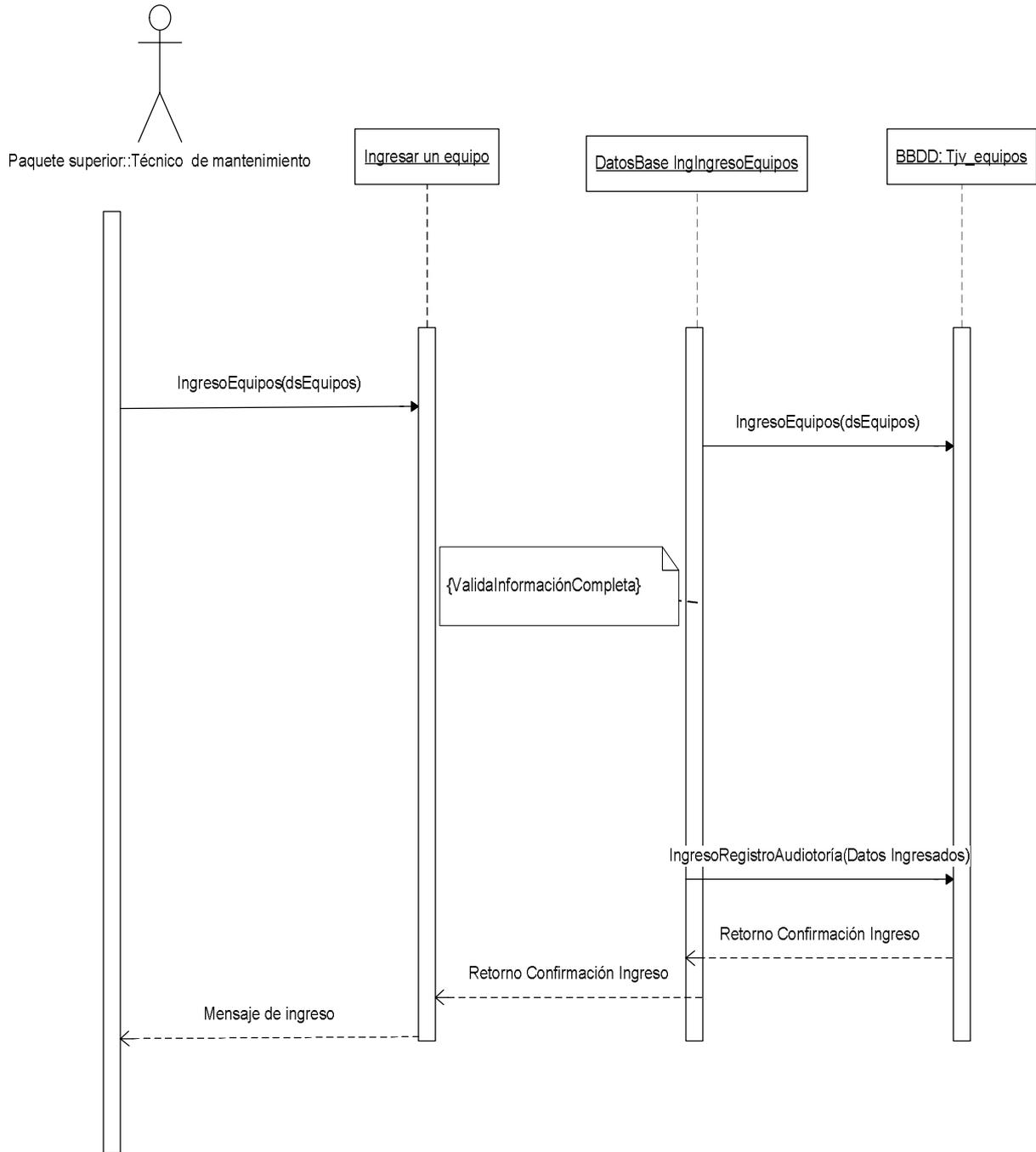
Fig#17. Diagrama de secuencia de Ingreso de Insumos

Fuente: Los autores



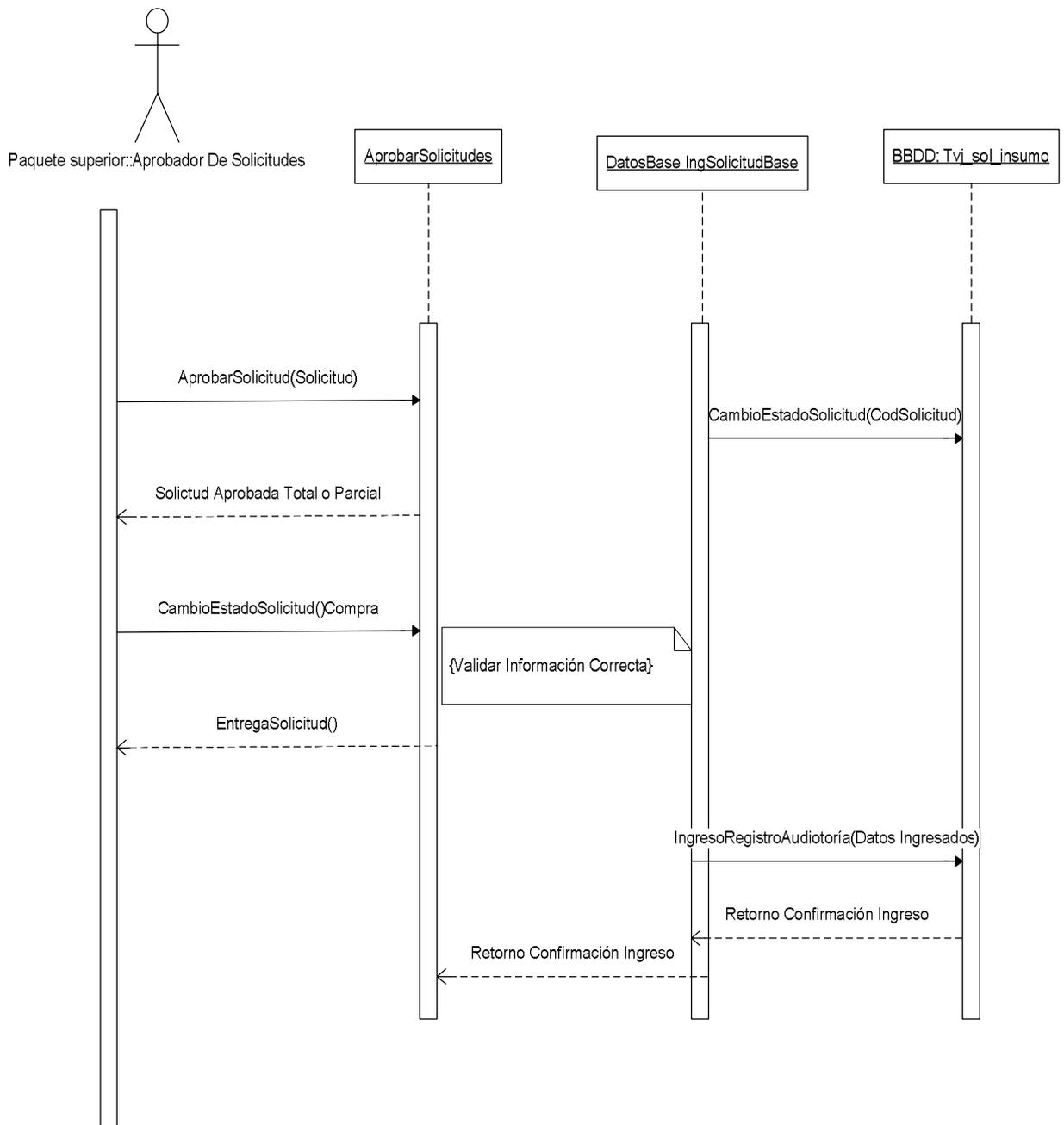
Fig#18. Diagrama de secuencia de Ingreso de Fallas

Fuente: Los autores



Fig#19. Diagrama de secuencia de Ingreso de Equipos

Fuente: Los autores



Fig#20. Diagrama de secuencia de Aprobación de Solicitudes
Fuente: Los autores

3.1.2.3 REQUERIMIENTOS DEL SISTEMA

MANPRECOR como sistema está llamado a cubrir las necesidades que tiene una empresa en cuanto al mantenimiento de sus equipos sin importar su naturaleza, calendarizando las rutinas que se deben realizar sobre los equipos así como asignando órdenes de trabajo a cada una de las rutinas para su correcta aplicación. Documentar los errores presentados en el funcionamiento con su respectiva solución si ya fue hallada y puesta en marcha. Además como prestación adicional ofrece la oportunidad de tener un stock de insumos para los equipos, realizar el proceso de solicitud de insumos si fuese necesario, teniendo en cuenta todo el flujo desde su petición, aprobación compra y posterior entrega.

Características del usuario MANPRECOR

El usuario MANPRECOR deberá estar directa o indirectamente ligado al mantenimiento de los equipos en la empresa ya que la naturaleza del software es precisamente esa, ahora bien también tendrán que utilizarlo las personas encargadas de las solicitudes de insumos si estas funciones están destinadas a alguien fuera del departamento de mantenimiento, también usuarios MANPRECOR serán los funcionarios encargados de aprobar la compra de insumos dentro de la empresa, los que realizan las compras y las entregas de lo solicitado. Como podemos darnos cuenta MANPRECOR está orientado a una gran cantidad de personas es por eso que cada usuario está definido dentro del sistema con un rol específico y la interfaz que manipulará será totalmente intuitiva y de fácil manejo.

Requisitos para futuras versiones del sistema

La escalabilidad de MANPRECOR está garantizada por su arquitectura de capas, ya que esto permite incorporar de una manera más fácil y con muy bajo impacto

nuevas funcionalidades además los módulos del sistema se pueden utilizar independientemente el uno del otro. El tener todos los métodos en un servicio Web también provee a MANPRECOR de una gran modularidad con otros sistemas, con esto vemos que las versiones futuras estarán garantizadas de un correcto funcionamiento.

Descripción De Formularios

Módulo de Mantenimiento

* Nombre del formulario: Alerta de rutinas

Elementos de la forma:

Contiene una tabla de opciones con los campos descripción, equipo y fecha.

Funcionalidades:

Este formulario sirve para informar al usuario sobre las rutinas que tiene pendiente en los próximos 5 días mostrando su descripción el nombre del equipo y la fecha en la cual se realizará la rutina.

Datos de entrada y de salida

El formulario es solo informativo.

Usuarios que pueden utilizar:

Todos los usuarios que tengan acceso al módulo de mantenimiento y que tengan rutinas asociadas.

*Nombre del formulario: Equipos

Elementos del formulario

Equipos:

Contiene una grilla con la información de los equipos ingresados

Administración de Equipos:

Nombre del Equipo: Descripción del equipo.

Tipo de Equipo: El tipo de equipo al cual pertenece la máquina que estamos ingresando.

Ubicación del Equipo: Indica la ubicación física del equipo dentro de la empresa.

Fecha de compra: Fecha de compra del equipo

Garantía del equipo: garantía del equipo en años.

Dimensiones, peso, capacidad y marca del equipo: Información general del equipo.

Proveedor: Indica el proveedor de mantenimiento del equipo si lo tiene.

Estado: Indica el estado del equipo.

Partes Equipos:

Grilla con la información de los equipos ingresados.

Grilla con la información de las partes del equipo seleccionado.

Ingresar Parte: Describimos el nombre de la parte a ingresar, editar o eliminar.

Información Adicional:

Grilla con la información de los equipos ingresados

Espacios de texto para ingresar la información adicional sobre el equipo seleccionado.

Funcionalidades:

El formulario de equipos está direccionado para registrar, editar y eliminar la información del equipo así como sus partes relacionadas y toda la información adicional que deseemos ingresar.

Datos de entrada y de salida:

Los datos de entrada son todos los que tienen que ver con la información del equipo que se guardarán en la base de datos, los datos de salida son los de los equipos presentados en la grilla.

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento y de administración de equipos.

*Nombre del formulario: Ubicación de Equipos

Elementos del formulario:

Botones para el ingreso y eliminación de las ubicaciones

Árbol jerárquico en el cual se muestran las ubicaciones ingresadas.

Nueva Ubicación: Recibe la ubicación a ingresar o eliminar.

Funcionalidades:

El formulario de ubicaciones está concebido para ingresar las ubicaciones físicas que hayan dentro de la empresa para después separar a los equipos por sus ubicaciones físicas.

Datos de entrada y salida:

Recibe el nombre de las ubicaciones y las guarda en la base de datos. Muestra en el árbol todos los lugares ingresados según su posición y jerarquía.

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento y parametrización del sistema.

*Nombre del formulario: Documentación Equipos

Elementos del formulario:

Grilla con la información de los equipos ingresados.

Nombre de Equipo: muestra el equipo seleccionado

Visor de imágenes ingresadas según el equipo.

Documento seleccionado: Nombre del documento a ingresar.

Documentos Cargados: Documentos guardados relacionados al equipo

Funcionalidades:

El formulario de documentación fue creado para relacionar al equipo información como imágenes de su estructura ya sean planos o fotos, así como documentos que tengan datos del equipo como garantías o manuales de usuarios.

Datos de entrada y salida:

Recibe los nombres de las imágenes y de los documentos para guardarlos en la base de datos, muestra en un visor las imágenes y los documentos los abre según su extensión en el programa correspondiente.

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento.

*Nombre del formulario: Tipos de Equipos

Elementos del formulario:

Grilla con la información de los tipos de equipos ingresados

Botones para el ingreso, edición y eliminación de los tipos de equipos.

Funcionalidades:

El formulario de tipos de equipos está llamado a clasificar los equipos ingresados en grupos de estos según se hayan clasificado, para tener una mejor organización de la información.

Datos de entrada y salida:

Recibe la descripción del tipo de equipo y lo guarda en la base de datos, muestra los ya ingresados

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento y parametrización del sistema.

*Nombre del formulario: Unidades

Elementos del formulario:

Grilla con la información de las unidades ingresadas.

Opciones de tipos de unidades: para escoger el tipo de unidad a ingresar

Ingresar unidad: Tipo texto

Código: tipo Texto

Descripción: Tipo texto

Tipo Unidad: Tipo Texto

Botones para el ingreso, edición y eliminación de las unidades.

Funcionalidades:

El formulario de unidades está realizado para tener en los equipos una información más al detalle de las unidades de medida que manejan según sea la necesidad.

Datos de entrada y salida:

Recibe la descripción de la unidad y el tipo y muestra las unidades ya ingresadas.

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento y parametrización del sistema.

*Nombre del formulario: Lista de Proveedores

Elementos del formulario:

Grilla con la información de las unidades ingresadas.

Ingresar unidad: Tipo texto

Nombre del Proveedor: tipo Texto

Dirección del proveedor: Tipo texto

Teléfono del proveedor: Tipo Texto

Botones para el ingreso, edición y eliminación de las unidades.

Funcionalidades:

El formulario de proveedores permite conocer la información de los proveedores de mantenimiento para los equipos

Datos de entrada y salida:

Recibe la información de los proveedores y la guarda en la base de datos, muestra los ingresados

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento y parametrización del sistema.

*Nombre del formulario: Empleados

Elementos del formulario:

Grilla con la información de los empleados.

Cédula empleado: Tipo numérico 10 dígitos.

Nombre del Empleado: tipo Texto.

Dirección del domicilio: Tipo texto.

Teléfono: Tipo numérico.

Celular: Tipo numérico.

Cargo del empleado: tipo texto.

Correo Electrónico: tipo texto.

Estado: tipo texto.

Botones para el ingreso, edición y eliminación de las unidades.

Funcionalidades:

El formulario de empleados cubre la necesidad de tener un registro de los empleados ya sea solo del departamento de mantenimiento o de toda la empresa

Datos de entrada y salida:

Recibe la información de los empleados y la guarda en la base de datos, muestra los ya ingresados.

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento y parametrización del sistema.

*Nombre del formulario: Usuarios y Perfiles

Elementos del formulario:

Grilla con la información de los usuarios.

Cédula: Tipo numérico 10 dígitos.

Nombre de Usuario: tipo Texto.

Fecha: Tipo texto dd/mm/aa.

Contraseña: Tipo texto.

Estado: tipo texto.

Botones para el ingreso, edición y eliminación de las unidades.

Funcionalidades:

El formulario de usuarios y perfiles permite la definición de los usuarios que manejarán el sistema así como los perfiles asignados para el acceso respectivo.

Datos de entrada y salida:

Recibe la información de los usuarios y la guarda en la base de datos, envía la contraseña del usuario encriptada.

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento.

*Nombre del formulario: Perfiles

Elementos del formulario:

Nombre del perfil: tipo Texto.

Perfiles disponibles: Perfiles del sistema

Botones para el ingreso, edición y eliminación de las unidades.

Funcionalidades:

El formulario perfiles permite la creación de nuevos perfiles además de los ya dados por el sistema.

Datos de entrada y salida:

Recibe la descripción del perfil y los accesos parametrizados.

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento.

*Nombre del formulario: Fechas no Laborables

Elementos del formulario:

Grilla con la información de las fechas no laborables

Descripción: Tipo texto

Botones para el ingreso, edición y eliminación de las unidades.

Funcionalidades:

El formulario de fechas no laborables permite ingresar al sistema las fechas festivas en las cuales no se trabajará en la empresa para evitar que se realicen rutinas en estas fechas.

Datos de entrada y salida:

Recibe la descripción y la fecha guardándolas en la base de datos.

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento.

*Nombre del formulario: Rutinas de Mantenimiento

Elementos del formulario:

Grilla con la información, los equipos y sus partes

Calendario con información de las fechas de las rutinas

Seleccionar Parte: Tipo texto

Descripción Rutina: Tipo texto

Fecha de inicio: Tipo texto dd/mm/aa

Frecuencia: Tipo texto

Duración de la rutina: Tipo texto

Botones para el ingreso, edición y eliminación de las unidades.

Funcionalidades:

El formulario de rutinas de mantenimiento está concebido para registrar las fechas en las cuales se realizará el mantenimiento de los equipos así como su duración, frecuencia y fecha en la cual va a iniciar

Datos de entrada y salida:

Recibe datos para el ingreso de la rutina y expone las fechas en las cuales se va a realizar cada una de ellas

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento.

*Nombre del formulario: Órdenes de Trabajo

Elementos del formulario:

Grilla con la información de las rutinas.

Número de Orden de trabajo: tipo numérico.

Seleccionar Parte: Tipo texto

Grillas con información de los insumos disponibles y utilizados.

Estado de la orden de trabajo: Tipo texto.

Fecha de inicio: Tipo texto dd/mm/aa.

Fecha Máxima de ejecución: Tipo texto dd/mm/aa.

Fecha de finalización: Tipo texto dd/mm/aa.

Nombre de la rutina: Tipo texto.

Equipo: Tipo texto.

Parte: Tipo texto.

Observación: Tipo texto.

Funcionalidades:

El formulario de órdenes de trabajo permite conocer la realización de una rutina en cuanto tiempo fue efectivizada y que insumos fueron necesarios para llevarla a cabo así como el estado en cual terminó la rutina.

Datos de entrada y salida:

Recibe la descripción de la rutina y guarda en la base de datos el estado de su realización y los insumos que fueron utilizados.

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento.

*Nombre del formulario: Fallas

Elementos del formulario:

Grilla con la información de los equipos ingresados.

Seleccionar Parte: Tipo texto

Descripción Falla: Tipo texto.

Fecha de registro: Tipo texto dd/mm/aa.

Equipo: Tipo texto.

Parte: Tipo texto.

Solución: Tipo texto.

Funcionalidades:

El formulario de fallas permite documentar las fallas que se presenten dentro del sistema y también ingresar las soluciones presentadas

Datos de entrada y salida:

Recibe la descripción de la falla presentada y envía la descripción de las fallas según el equipo y la parte donde se presentaron.

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento.

*Nombre del formulario: Consultas

Elementos del formulario:

Parámetros de búsqueda según el tipo de consulta: Varios tipos

Grilla que presenta la información consultada.

Botones para exportar la información

Funcionalidades:

El formulario consultas permite al usuario tener la información requerida según sea su necesidad y según se ingresen los parámetros de búsqueda.

Datos de entrada y salida:

Recibe todos los parámetros de búsqueda posibles según cada consulta y envía los resultados en una grilla con toda la información requerida pudiéndola exportar fuera del aplicativo.

Usuarios que pueden utilizar:

Los usuarios que tengan el perfil de mantenimiento.

Módulo de Insumos

*Nombre del Formulario: Tipo de Insumos

Elementos del formulario:

Nombre del tipo: Tipo texto

Código: Tipo numérico

Edición del Nombre: Tipo texto

Edición del código: Tipo numérico

Nombre a eliminar: Tipo texto

Botones para el ingreso, edición y eliminación de tipos de insumo

Funcionalidades:

Ingresa, edita y elimina información de tipos de insumos.

Datos de entrada y salida:

La información que se receipta en el formulario proviene de lo ingresado por el usuario según sus necesidades. Esta información es almacenada en la base de datos

Usuarios que pueden utilizar:

Los usuarios que pueden ingresar a este formulario son: el administrador y el personal de mantenimiento que posea el perfil.

*Nombre del Formulario: Proveedores

Elementos del formulario:

Nombre: Tipo texto

Proveedor: Tipo texto

Teléfono: Tipo texto

Código: Tipo texto

Nombre a editar: Tipo texto

Dirección a editar: Tipo texto

Teléfono a editar: Tipo texto

Código del proveedor a eliminar: Tipo texto

Nombre a eliminar: Tipo texto

Dirección a eliminar: Tipo texto

Teléfono a eliminar: Tipo texto

Botones para el ingreso, edición y eliminación de proveedores.

Funcionalidades:

Ingresa, edita y elimina información de los proveedores de insumos

Datos de entrada y salida:

La información que se receipta en el formulario proviene de lo ingresado por el usuario según sus necesidades y proveedores con los cuales se tenga acuerdo. Esta información es almacenada en la base de datos.

Usuarios que pueden utilizar:

Los usuarios que pueden ingresar a este formulario son: el administrador y el personal de mantenimiento que posea el perfil.

*Nombre del Formulario: Insumos

Elementos del formulario:

Nombre del insumo: Tipo texto

Stock: Tipo numérico

Tipo de insumo: Tipo combo de selección

Tipo de proveedor: Tipo combo de selección

Cantidad mínima: Tipo numérico

Nombre del parámetro: Tipo texto

Nombre del usuario: Tipo texto

Fecha: Tipo texto

Código del insumo a editar: Tipo texto

Nombre a editar: Tipo texto

Stock a editar: Tipo numérico

Tipo de insumo a editar: Tipo combo de selección

Tipo de proveedor a editar: Tipo combo de selección

Cantidad mínima a editar: Tipo numérico

Nombre del parámetro: Tipo texto

Usuario a editar: Tipo texto

Fecha a editar: Tipo texto

Código del insumo a eliminar: Tipo texto

Nombre a eliminar: Tipo texto

Botones para el ingreso, edición y eliminación de insumos.

Funcionalidades:

Ingresa, edita y elimina información de los nuevos insumos que se registran.

Datos de entrada y salida:

La información que se receipta en el formulario proviene de lo ingresado por el usuario según sus necesidades y proveedores con los cuales se tenga acuerdo.

Usuarios que pueden utilizar:

Los usuarios que pueden ingresar a este formulario son: el administrador y el personal de mantenimiento que posea el perfil.

*Nombre del Formulario: Solicitudes

Elementos del formulario:

Fecha: Tipo texto

Nombre del usuario: Tipo texto

Tipo de insumo: Tipo combo de selección

Insumos disponibles: Tipo combo de selección

Nombre del proveedor: Tipo combo de selección

Cantidad: Tipo texto

Nombre del nuevo equipo: Tipo texto

Botones para el ingreso de solicitudes.

Funcionalidades:

Ingresa las solicitudes de compra de insumos que no existen en stock.

Datos de entrada y salida:

La información que se receipta en el formulario proviene de lo ingresado por el usuario según sus necesidades y requerimiento de insumos.

Usuarios que pueden utilizar:

Los usuarios que pueden ingresar a este formulario son: el administrador y el personal de mantenimiento que posea el perfil.

*Nombre del Formulario: Aprobación

Elementos del formulario:

Código: Tipo texto

Fecha: Tipo texto

Nombre del usuario: Tipo texto

Items ingresados en la solicitud: Tipo datagrid

Observación: Tipo texto

Botones para aceptación o rechazo de solicitudes.

Funcionalidades:

El usuario es capaz de aprobar parcial o totalmente la solicitud, así como también el rechazo total de la misma, para que siga su flujo de gestión.

Datos de entrada y salida:

La información que se receipta para la aprobación de la solicitud proviene del aprobador que realiza el análisis de la solicitud.

Usuarios que pueden utilizar:

Los usuarios que pueden ingresar a este formulario son: el administrador y jefe del personal de mantenimiento que posea perfil.

*Nombre del Formulario: MisCompras

Elementos del formulario:

Código: Tipo texto

Fecha: Tipo texto

Nombre del usuario: Tipo texto

Estado de la solicitud: Tipo texto

Fecha de entrega: Tipo calendar combo

Botón para el envío a Compras

Funcionalidades:

El personal de Compras puede aprobar las solicitudes y enviar a Comprar, según las necesidades y el análisis realizado.

Datos de entrada y salida:

La información que se receipta para la aprobación de la solicitud proviene del supervisor de mantenimiento que realiza el análisis de la solicitud.

Usuarios que pueden utilizar:

Los usuarios que pueden ingresar a este formulario son: el administrador, personal autorizado de compras y jefe del personal de mantenimiento que posea perfil.

*Nombre del Formulario: MisEntregas

Elementos del formulario:

Código: Tipo texto

Fecha de ingreso: Tipo texto

Nombre del usuario: Tipo texto

Estado de la solicitud: Tipo texto

Observación: Tipo texto

Botón para el registro de recepción de entregas

Funcionalidades:

El personal de Logística puede terminar el flujo de la solicitud una vez que ya se haya realizado la entrega de los insumos físicamente en bodega con lo cual indicaría que los insumos sí fueron entregados.

Datos de entrada y salida:

La información que se receipta para la entrega y aceptación de fin de flujo de la solicitud proviene del área de logística que realiza el análisis de la solicitud. Usuarios que pueden utilizar:

Los usuarios que pueden ingresar a este formulario son: el administrador, personal autorizado de logística y jefe del personal de mantenimiento que posea perfil.

*Nombre del Formulario: ConsultaInsumos

Elementos del formulario:

Nombre del insumo: Tipo texto

Cantidad de insumos en stock: Tipo texto

Cantidad mínima: Tipo texto

Estado del insumo: Tipo estado

Tipo de insumo: Tipo combo se selección

Tipo de proveedor: Tipo texto

Botones para la consulta, exportación e impresión.

Funcionalidades:

Permite visualizar los reportes de consultas que se ejecuta.

Datos de entrada y salida:

El personal que trabaja en la empresa puede visualizar este formulario sin problema alguno. La transacción es una consulta por la cual se extrae datos de la base de datos para indicar la información

Usuarios que pueden utilizar:

Los usuarios que pueden ingresar a este formulario son: el administrador, el jefe de personal de mantenimiento y cualquier usuario que posea el perfil de Consultas.

*Nombre del Formulario: ConsultaSolicitudes

Elementos del formulario:

Nombre: Tipo texto

Cantidad disponible: Tipo texto

Cantidad mínima: Tipo texto

Estado de la solicitud: Tipo estado

Tipo de insumos: Tipo combo se selección

Tipo de proveedor: Tipo de texto

Botones para la consulta, exportación e impresión

Funcionalidades:

Permite visualizar los reportes de consultas generados por el usuario.

Datos de entrada y salida:

El personal que trabaja en la empresa puede visualizar este formulario sin problema alguno. La transacción es una consulta por lo cual se extrae datos de la base de datos para indicar la información

Usuarios que pueden utilizar:

Los usuarios que pueden ingresar a este formulario son: el administrador, el jefe de personal de mantenimiento y cualquier usuario que posea el perfil de Consultas.

3.1.3 DISEÑO DEL SISTEMA

3.1.3.1 CAPAS

3.1.3.1.1 Capa de Presentación

La capa de presentación o de usuario es la interfaz gráfica que comunica al sistema con el usuario, aquí se almacenará las formas que serán presentadas al usuario para realizar todos los procesos del mantenimiento.

El diseño de la capa de presentación es fundamental para tener un entorno amigable con el usuario, por tal motivo se utilizará los componentes de Infragistics; con los cuales daremos un diseño agradable y fácil de utilizar a los usuarios.

Para el diseño gráfico de la aplicación en cuanto a la interfaz y sus componentes se utiliza la herramienta de Infragistics, que cuenta con un menú similar al de Office 2007, con una apariencia totalmente intuitiva e ilustrada que permite al usuario un fácil manejo de la aplicación. El color predominante dentro de la aplicación es Lavanda con varias tonalidades de azul.

3.1.3.1.2 Capa de Negocio

La capa de negocio es el enlace entre la capa de presentación y la capa de datos, va a contener todos los procesos que necesiten de cálculo interno antes de presentar al usuario; aquí se alojarán los métodos que se van a enlazar con el servicio Web.

3.1.3.1.3 Capa de Datos

La capa de datos es la encargada de contener los métodos de consultas o almacenamiento hacia las tablas de la base de datos mediante la utilización de los datasets ubicados en la capa auxiliar; ésta nos sirve de enlace y comunicación entre la base de datos y el Sistema.

3.1.4 DESARROLLO DEL SISTEMA

3.1.4.1 Información y Localización de Equipos

Ingreso de Equipos

```
private void btnIngreso_Click(object sender, EventArgs e)
{
    StringBuilder error = new StringBuilder();
    try
    {
        //Inicializamos las variables y buscamos el código la ubicación, el tipo de equipo y el
        proveedor a ingresar
        DTSEquipos dsEquipos = new DTSEquipos();
        DTSEquipos.TVJ_equidesRow fila = dsEquipos.TVJ_equides.NewTVJ_equidesRow();
        DTSUbicaciones.TVJ_ubicacionRow[] filaUbicacionEquipo = null;
        filaUbicacionEquipo = dslistaubicaciones.TVJ_ubicacion.Select("path_ubicacion=" +
        txtUbicacionEquipo.Text + "") as DTSUbicaciones.TVJ_ubicacionRow[];

        if (ucTipoEquipo.Value==null)
        {
            error.Append(" Ingresar un tipo de equipo valido, ");
        }
        if (filaUbicacionEquipo.Length == 0)
        {
            error.Append(" Ingresar una ubicacion valida, ");
        }
        if (ucProveedorEquipo.Value==null)
        {
            error.Append(" ingresar un proveedor valido, ");
        }
        DialogResult CampoAuxiliar = MessageBox.Show("Está Seguro que desea Ingresar la
        información ?", "Informacion", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Information);

        if (CampoAuxiliar == System.Windows.Forms.DialogResult.Yes &&
        txtNombreEquipo.Text!="")
        {
            //Con la confirmación de la información correcta tomamos los valores y los ingresamos
            en la BBDD
        }
    }
}
```

```

fila.nom_equides = txtNombreEquipo.Text;
fila.cod_tipoequi = Convert.ToInt32(ucTipoEquipo.Value);
fila.cod_ubicacion = filaUbicacionEquipo[0].cod_ubicacion;
fila.feccom_equides = uccFechaCompra.Text;
fila.garan_equides = txtGarantía.Text;
fila.equibas_equides = "";
fila.dimen_equides = txtDimensionEquipo.Text + ". " + ucUnidades1.Text;
fila.pes_equides = txtpesoEquipo.Text + ". " + ucUnidades2.Text;
fila.capa_equides = txtcapacidadEquipo.Text + ". " + ucUnidades3.Text;
fila.cod_proveequi = Convert.ToInt32(ucProveedorEquipo.Value);
fila.estado = ucEstado.Text;
fila.marc_equides = txtmarcaEquipo.Text;
fila.priori_equides = "";
if (txtAux1.Text != "")
{ fila.aux1_equides = lblAux1.Text + "/" + txtAux1.Text; }
if (txtAux2.Text != "")
{ fila.aux2_equides = lblAux2.Text + "/" + txtAux2.Text; }
if (txtAux3.Text != "")
{ fila.aux3_equides = lblAux3.Text + "/" + txtAux3.Text; }
if (txtAux4.Text != "")
{ fila.aux4_equides = lblAux4.Text + "/" + txtAux4.Text; }
if (txtAux5.Text != "")
{ fila.aux5_equides = lblAux5.Text + "/" + txtAux5.Text; }
if (txtAux6.Text != "")
{ fila.aux6_equides = lblAux6.Text + "/" + txtAux6.Text; }
if (txtAux7.Text != "")
{ fila.aux7_equides = lblAux7.Text + "/" + txtAux7.Text; }
if (txtAux8.Text != "")
{ fila.aux8_equides = lblAux8.Text + "/" + txtAux8.Text; }
dsEquipos.TVJ_equides.AddTVJ_equidesRow(fila);
servicioWeb.ingresoEquipos(dsEquipos);

//inicio registro Auditoría
DTSAuditoría dsAuditoría = new DTSAuditoría();
DTSAuditoría.TVJ_AuditoríaRow filaAud =
dsAuditoría.TVJ_Auditoría.NewTVJ_AuditoríaRow();
filaAud.módulo = "Mantenimiento";
filaAud.nom_usuario = usuarioRefactorizado12;
filaAud.accion = "Inserción";
filaAud.fecha = DateTime.Now;
filaAud.datos_afectados = "Tabla:TVJ_equides/Datos:" + "Equipo " +
txtNombreEquipo.Text + ";Tipo " + Convert.ToInt32(ucTipoEquipo.Value)
+ ";Ubicacion " + filaUbicacionEquipo[0].cod_ubicacion + ";Compra " +
uccFechaCompra.Text + ";Garantía " + txtGarantía.Text
+ ";Dimension " + txtDimensionEquipo.Text + ". " + ucUnidades1.Text + ";Peso " +
txtpesoEquipo.Text
+ ". " + ucUnidades2.Text + ";Capacidad " + txtcapacidadEquipo.Text + ". " +
ucUnidades3.Text + ";Proveedor " + Convert.ToInt32(ucProveedorEquipo.Value)
+ ";Estado " + ucEstado.Text + ";Marca " + txtmarcaEquipo.Text ;
dsAuditoría.TVJ_Auditoría.AddTVJ_AuditoríaRow(filaAud);
servicioWeb.ingresoRegistroAuditoría(dsAuditoría);
//fin registro Auditoría

LimpiarInformacion();
dsEquipos = servicioWeb.consultaEquipos();
MessageBox.Show("Dato Ingresado Correctamente", "Ingreso de Datos",
MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

    }

    else
    {
        MessageBox.Show("No se ingresó la información, digite el nombre del equipo");
    }

    //consulta código del equipo y envía al reporte
    DTSEquipos.TVJ_equidesRow[] filaconsulta = null;
    dsEquipos = servicioWeb.consultaEquipos();
    filaconsulta = dsEquipos.TVJ_equides.Select() as DTSEquipos.TVJ_equidesRow[];
    codEqEnvia = filaconsulta[filaconsulta.Length - 1].cod_equides;
    repEquipo.CodEquipo = codEqEnvia;
    repEquipo.Show();
}
catch (Exception ex)
{
    MessageBox.Show("Error al ingresar los datos,favor" + error.ToString(), "Ingreso de Datos",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}
}

```

3.1.4.2 Administración de Insumos

Ingreso de Insumos

```

private void btnIngreso_Click(object sender, EventArgs e)
{
    //Ingreso de un nuevo insumo
    int auxCódigo = 0;

    try
    {
        DTSInsumos dsinsumos = new DTSInsumos();
        DTSInsumos.TVJ_insumoRow fila = dsinsumos.TVJ_insumo.NewTVJ_insumoRow();

        fila.nombre_insumo = txtnombre.Text.Trim();
        fila.stock_insumo = Convert.ToInt32(txtstock.Text.Trim());
        fila.stock_minimo = Convert.ToInt32(txtminimo.Text.Trim());
        fila.parametro = txtparametro.Text.Trim();
        fila.estado = 9;

        //guarda el código del tipo de insumo
        DTSTipoInsumo.TVJ_tipoinsumoRow[] filaconsulta = null;
        dstipoInsumo = servicioWeb.consultaCódigo Insumo(cbtipo.Text);
        filaconsulta = dstipoInsumo.TVJ_tipoinsumo.Select("nombre_tipoinsumo=" +
"nombre_tipoinsumo" + "") as DTSTipoInsumo.TVJ_tipoinsumoRow[];
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error al ingresar los datos,favor" + error.ToString(), "Ingreso de Datos",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
}

```

```

fila.cod_tipoinsumo = filaconsulta[0].cod_tipoinsumo;

//guarda el código del proveedor de insumo
DTSProveedorInsumo.TVJ_proveeinsuRow[] filaconsulta2 = null;
dsproveedorInsumo = servicioWeb.consultaCódigo Proveedor(cbproveedor.Text);
filaconsulta2 = dsproveedorInsumo.TVJ_proveeinsu.Select("nombre_proveeinsu=" +
"nombre_proveeinsu" + "") as DTSProveedorInsumo.TVJ_proveeinsuRow[];
fila.cod_proveeins = filaconsulta2[0].cod_proveeins;

if ((fila.nombre_insumo == "") || (txtstock.Text == "") || (cbtipo.Text == "") ||
(cbproveedor.Text == "") || (txtminimo.Text == "") || (fila.parametro == ""))
{
    MessageBox.Show("Debe ingresar toda la información requerida");
}
else
{
    DialogResult resultado;
    resultado = MessageBox.Show("Esta seguro que desea ingresar la información",
"Confirmación", MessageBoxButtons.YesNo);

    if (resultado == System.Windows.Forms.DialogResult.Yes)
    {
        dsinsumos.TVJ_insumo.AddTVJ_insumoRow(fila);
        servicioWeb.ingresoInsumos(dsinsumos);

        //ingreso de insumo como entrada en tabla movimientos
        dsinsumos.Clear();
        DTSInsumos.TVJ_insumoRow[] filaconsulta3 = null;
        dsinsumos = servicioWeb.consultaInsumos(9);
        filaconsulta3 = dsinsumos.TVJ_insumo.Select("nombre_insumo= " + txtnombre.Text +
""") as DTSInsumos.TVJ_insumoRow[];
        auxCódigo = filaconsulta3[0].cod_insumo;

        DTSMovimientoInsumos.TVJ_movimientos_insumosRow filaMov =
dsmovimientos.TVJ_movimientos_insumos.NewTVJ_movimientos_insumosRow();
        filaMov.movimiento = "Entrada";
        filaMov.cod_insumo = Convert.ToString(auxCódigo);
        filaMov.insumo = txtnombre.Text;
        filaMov.cantidad = Convert.ToInt32(txtstock.Text);
        filaMov.usuario = txtusuario.Text;
        filaMov.fecha = txtfecha.Text;
        //filaMov.factura = txtfactura.Text;

        dsmovimientos.TVJ_movimientos_insumos.AddTVJ_movimientos_insumosRow(filaMov);
        servicioWeb.ingresoMovimientos(dsmovimientos);

        //inicio registro Auditoría
        DTSAuditoría dsAuditoría = new DTSAuditoría();
        DTSAuditoría.TVJ_AuditoríaRow filaAud =
dsAuditoría.TVJ_Auditoría.NewTVJ_AuditoríaRow();
        filaAud.módulo = "Insumos";
        filaAud.nom_usuario = usuarioRefactorizado6;
        filaAud.accion = "Inserción";
        filaAud.fecha = DateTime.Now;
        filaAud.datos_afectados = "Tabla: TVJ_insumo/Datos: " + "Nombre " + txtnombre.Text
+ ";Stock " + txtstock.Text + ";Tipo " + cbtipo.Text + ";Proveedor " + cbproveedor.Text +

```

```

";Stock_minimo " + txtminimo.Text + ";Parametro " + txtparametro.Text /*+ ";Factura " +
txtfactura.Text*/ + ";Usuario " + txtusuario.Text + ";Fecha " + txtfecha.Text;
    dsAuditoría.TVJ_Auditoría.AddTVJ_AuditoríaRow(filaAud);
    servicioWeb.ingresoRegistroAuditoría(dsAuditoría);
    //fin registro Auditoría

    MessageBox.Show("Dato Ingresado Correctamente");
    //refrescar
    cbtipo.Text = "";
    cbproveedor.Text = "";
    cargaTiposInsumo();
    cargaProveedorInsumo();
    cargaInsumos();
    txtnombre.Text = "";
    txtstock.Text = "";
    txtminimo.Text = "";
    txtparametro.Text = "";
    txtusuario.Text = "";
    txtfecha.Text = "";
}
else
{
    cbtipo.Text = "";
    cbproveedor.Text = "";
    cargaTiposInsumo();
    cargaProveedorInsumo();
    cargaInsumos();
    txtnombre.Text = "";
    txtstock.Text = "";
    txtminimo.Text = "";
    txtparametro.Text = "";
    txtnombre.Focus();
}
}
}
}
catch
{
    MessageBox.Show("Debe ingresar toda la información requerida");
}
}
}

```

Registro de Solicitud

```

private void btnIngresar_Click(object sender, EventArgs e)
{
    try
    {
        int codSolicitud,contInsumos;
        string nombreInsumo;
        int auxCant,auxCod;

        //inicialización Auditoría
        DTSAuditoría dsAuditoría = new DTSAuditoría();

        DTSSolicitudInsumo dssolinsumo = new DTSSolicitudInsumo();
    }
}

```

```

DTSSolicitudInsumo.TVJ_sol_insumoRow fila =
dssolinsumo.TVJ_sol_insumo.NewTVJ_sol_insumoRow();

fila.fecha = txtfecha.Text;
fila.usuario = txtusuario.Text;
fila.estado = 3;
fila.observacion = "Ingresado OK";

//guarda la solicitud

if (dgvDetalle.Rows.Count != 0)
{
    DialogResult resultado;
    resultado = MessageBox.Show("Está seguro que desea ingresar la información",
"Confirmación", MessageBoxButtons.YesNo);

    if (resultado == System.Windows.Forms.DialogResult.Yes)
    {
        //ingreso en tabla sol_insumo
        dssolinsumo.TVJ_sol_insumo.AddTVJ_sol_insumoRow(fila);
        servicioWeb.ingresoSolicitudInsumo(dssolinsumo);

        //consulta código de solicitud
        DTSSolicitudInsumo.TVJ_sol_insumoRow[] filaconsulta = null;
        dssolinsumo = servicioWeb.consultaSolicitudInsumo();
        filaconsulta = dssolinsumo.TVJ_sol_insumo.Select() as
DTSSolicitudInsumo.TVJ_sol_insumoRow[];
        codSolicitud = filaconsulta[filaconsulta.Length - 1].cod_solicitud;
        contInsumos = (dgvDetalle.Rows.Count);

        DTSItemSolicitud dsitemsol = new DTSItemSolicitud();
        DTSItemSolicitud.TVJ_itemSolicitudRow fila2 =
dsitemsol.TVJ_itemSolicitud.NewTVJ_itemSolicitudRow();
        DTSInsumos.TVJ_insumoRow[] filaconsulta2 = null;

        for (int i = 0; i < contInsumos; i++)
        {
            //consulta el código del insumo
            dsInsumo = servicioWeb.consultaInsumos(9);
            filaconsulta2 = dsInsumo.TVJ_insumo.Select("nombre_insumo=" +
(dgvDetalle.Rows[i].Cells[1].Value.ToString()) + """) as DTSInsumos.TVJ_insumoRow[];

            if (filaconsulta2.Length == 0)
            {
                //ingresa primero el nuevo insumo en la tabla insumo
                DTSInsumos.TVJ_insumoRow filaAux =
dsInsumo.TVJ_insumo.NewTVJ_insumoRow();
                filaAux.nombre_insumo = dgvDetalle.Rows[i].Cells[1].Value.ToString();
                filaAux.stock_insumo = 0;
                filaAux.stock_minimo = 0;
                filaAux.parametro = " ";
                filaAux.estado = 9;

                //guarda el código del tipo de insumo
                DTSTipoInsumo.TVJ_tipoinsumoRow[] filaconsultaAux = null;
                dsTipo = servicioWeb.consultaCódigo
Insumo(dgvDetalle.Rows[i].Cells[2].Value.ToString());

```

```

        filaconsultaAux = dsTipo.TVJ_tipoinsumo.Select() as
DTSTipoInsumo.TVJ_tipoinsumoRow[];
        filaAux.cod_tipoinsumo = filaconsultaAux[0].cod_tipoinsumo;

        //guarda el código del proveedor de insumo
        DTSProveedorInsumo.TVJ_proveeinsuRow[] filaconsulta2Aux = null;
        dsproveedorInsumo = servicioWeb.consultaCódigo
Proveedor(dgvDetalle.Rows[i].Cells[3].Value.ToString());
        filaconsulta2Aux = dsproveedorInsumo.TVJ_proveeinsu.Select() as
DTSProveedorInsumo.TVJ_proveeinsuRow[];
        filaAux.cod_proveeins = filaconsulta2Aux[0].cod_proveeins;

        dsInsumo.TVJ_insumo.AddTVJ_insumoRow(filaAux);
        servicioWeb.ingresoInsumos(dsInsumo);
        dsInsumo.Clear();

        //consulta el código del insumo generado anteriormente
        dsInsumo = servicioWeb.consultaInsumos(9);
        filaconsulta2 = dsInsumo.TVJ_insumo.Select("nombre_insumo=" +
(dgvDetalle.Rows[i].Cells[1].Value.ToString()) + """) as DTInsumos.TVJ_insumoRow[];
        fila2.cod_insumo = filaconsulta2[0].cod_insumo;
    }
    else
    {
        fila2.cod_insumo = filaconsulta2[0].cod_insumo;
    }
    //auxiliares para Auditoría
    auxCant = Convert.ToInt32(dgvDetalle.Rows[i].Cells[0].Value);
    auxCod = fila2.cod_insumo;

    fila2.cod_solicitud = codSolicitud;
    fila2.cantidad = Convert.ToInt32(dgvDetalle.Rows[i].Cells[0].Value);
    fila2.estado_item = 0;
    fila2.cantidad_aprobada = 0;
    dsitemsol.TVJ_itemSolicitud.AddTVJ_itemSolicitudRow(fila2);
    servicioWeb.ingresoItemSolicitud(dsitemsol);
    dsitemsol.Clear();

    //inicio registro Auditoría
    DTSAuditoría.TVJ_AuditoríaRow filaAud1 =
dsAuditoría.TVJ_Auditoría.NewTVJ_AuditoríaRow();
    filaAud1.módulo = "Insumos";
    filaAud1.nom_usuario = txtusuario.Text;
    filaAud1.accion = "Inserción";
    filaAud1.fecha = DateTime.Now;
    filaAud1.datos_afectados = "Tabla: TVJ_sol_insumo/Datos: Ingreso Item" +
";CódigoSolicitud " + codSolicitud + ";Cantidad: " + auxCant + ";CodInsumo: " + auxCod;
    dsAuditoría.TVJ_Auditoría.AddTVJ_AuditoríaRow(filaAud1);
    servicioWeb.ingresoRegistroAuditoría(dsAuditoría);
    //fin registro Auditoría

}

//inicio registro Auditoría

```

```

        DTSAuditoría.TVJ_AuditoríaRow filaAud =
dsAuditoría.TVJ_Auditoría.NewTVJ_AuditoríaRow();
        filaAud.módulo = "Insumos";
        filaAud.nom_usuario = txtusuario.Text;
        filaAud.accion = "Inserción";
        filaAud.fecha = DateTime.Now;
        filaAud.datos_afectados = "Tabla: TVJ_sol_insumo/Datos: Ingreso Solicitud" +
";CódigoSolicitud " + codSolicitud + ";InsumosIngresados " + contInsumos;
        dsAuditoría.TVJ_Auditoría.AddTVJ_AuditoríaRow(filaAud);
        servicioWeb.ingresoRegistroAuditoría(dsAuditoría);
        //fin registro Auditoría

        MessageBox.Show("Dato Ingresado Correctamente");
        MessageBox.Show("Se generó la solicitud No." + codSolicitud);
        cbtipoinsumo.Text = "";
        cbInsumo.Text = "";
        txtcantidad.Text = "";
        cbtipoinsumo.Items.Clear();
        cbInsumo.Items.Clear();
        cargaTipoinsumo();
        dgvDetalle.Rows.Clear();
        codSolicitudEnvia = codSolicitud;
        Replmprime.CodSolicitudRecibe = codSolicitudEnvia;
        Replmprime.Show();

    }
    else
    {
        cbtipoinsumo.Text = "";
        cbInsumo.Text = "";
        txtcantidad.Text = "";
        cbtipoinsumo.Items.Clear();
        cbInsumo.Items.Clear();
        cargaTipoinsumo();
    }
}
else
{
    MessageBox.Show("Debe ingresar la información de los insumos a solicitar");
}
}
catch
{
    MessageBox.Show("Error al Ingresar el Dato,Favor intente nuevamente");
}
}
}

```

Aprobación de Solicitud

```

private void btnAprobar_Click(object sender, EventArgs e)
{
    //declaración de variables
    int contadorAprob = 0;
    int fila;
    try
    {
        //se consulta la fila que se va a editar en las solicitudes ingresadas enviando el número
        que se seleccionó
    }
}

```

```

ServiceInsumos.DTSSolicitudInsumo.TVJ_sol_insumoRow[] filaEdicion = null;
dssolinsumo = servicioWeb.consultaSolicitudEstado(3);
filaEdicion = dssolinsumo.TVJ_sol_insumo.Select("cod_solicitud=" + txtcodigo .Text + "")
as ServiceInsumos.DTSSolicitudInsumo.TVJ_sol_insumoRow[];

DialogResult = MessageBox.Show(" ¿Está seguro que desea aprobar la solicitud?",
"Aprobación", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (DialogResult == DialogResult.Yes)
{
//se verifica si tiene activo 1 o mas checkbox, caso contrario quiere decir que no está
aprobando ningún item
for (int i = 0; i < dgItems.Rows.Count; i++)
{
if (Convert.ToBoolean(dgItems.Rows[i].Cells[2].Value) == true)
{
contadorAprob = contadorAprob + 1;
}
}

//si se encontró anteriormente al menos 1 aprobación edita la fila consultada
anteriormente cambiando el estado
//de la solicitud a APROBADA y registrando la fecha de aprobación
if (contadorAprob > 0)
{
filaEdicion[0].fecha = txtfecha.Text;
filaEdicion[0].usuario = txtusuario.Text;
filaEdicion[0].estado = 4;
filaEdicion[0].observacion = "Fecha aprobación:" + fecha.ToString("dd/MM/yyyy");

dssolinsumo.TVJ_sol_insumo.ImportRow(filaEdicion[0]);
servicioWeb.ingresoSolicitudInsumo(dssolinsumo);
cargarSolicitudPendiente();

//ingresa en tabla item_solicitud el estado de cada item

DTSInsumos.TVJ_insumoRow[] filaconsulta2 = null;
ServiceInsumos.DTSItemSolicitud.TVJ_itemSolicitudRow[] filaEdicion2 = null;

//recorre el datagrid
for (int i = 0; i < dgItems.Rows.Count; i++)
{
dsitemsol.Clear();
dsinsumos.Clear();

//consulta los insumos que están en estado ACTIVO
dsinsumos = servicioWeb.consultaInsumos(9);
filaconsulta2 = dsinsumos.TVJ_insumo.Select("nombre_insumo=" +
dgItems.Rows[i].Cells[1].Value + "") as DTSInsumos.TVJ_insumoRow[];

//consulta todos los insumos ingresados en la tabla con el código de la solicitud
dsitemsol = servicioWeb.consultaItemsSolicitud(Convert.ToInt32(txtcodigo .Text));
filaEdicion2 = dsitemsol.TVJ_itemSolicitud.Select("cod_insumo=" +
filaconsulta2[0].cod_insumo + "") as ServiceInsumos.DTSItemSolicitud.TVJ_itemSolicitudRow[];

if ((Convert.ToBoolean(dgItems.Rows[i].Cells[2].Value) == true) &&
(dgItems.Rows[i].Cells[3].Value != null))

```

```

        {
            filaEdicion2[0].estado_item = 8;
            filaEdicion2[0].cantidad_aprobada =
Convert.ToInt32(dglItems.Rows[i].Cells[3].Value);
        }
        else
        {
            if ((Convert.ToBoolean(dglItems.Rows[i].Cells[2].Value) == true) &&
(dglItems.Rows[i].Cells[3].Value == null))
            {
                filaEdicion2[0].estado_item = 4;
            }
            else
            {
                filaEdicion2[0].estado_item = 5;
            }
        }
        dsitemsol.TVJ_itemSolicitud.ImportRow(filaEdicion2[0]);
        servicioWeb.ingresoltemSolicitud(dsitemsol);
        dsitemsol.Clear();
    }

    //inicio registro Auditoría
    DTSAuditoría dsAuditoría = new DTSAuditoría();
    DTSAuditoría.TVJ_AuditoríaRow filaAud =
dsAuditoría.TVJ_Auditoría.NewTVJ_AuditoríaRow();
    filaAud.módulo = "Insumos";
    filaAud.nom_usuario = txtusuario.Text;
    filaAud.accion = "Inserción";
    filaAud.fecha = DateTime.Now;
    filaAud.datos_afectados = "Tabla: TVJ_sol_insumo/Datos: " + "Aprobación
Solicitud"+";CódigoSolicitud " + txtcódigo .Text;
    dsAuditoría.TVJ_Auditoría.AddTVJ_AuditoríaRow(filaAud);
    servicioWeb.ingresoRegistroAuditoría(dsAuditoría);
    //fin registro Auditoría

    MessageBox.Show("Solicitud Aprobada");

    CodSolicitudEnvia = Convert.ToInt32(txtcódigo .Text);
    txtcódigo .Text = "";
    txtfecha.Text = "";
    txtusuario.Text = "";
    dglItems.Rows.Clear();

    //imprime el reporte
    Replmprime.CodSolicitudRecibe = CodSolicitudEnvia;
    Replmprime.Show();

}
else
{
    MessageBox.Show("No se puede aprobar, la solicitud no tiene items aprobados");
}

```

```

    }
    else
    {
        txtcodigo .Text = "";
        txtfecha.Text = "";
        txtusuario.Text = "";
        dgltems.Rows.Clear();
    }
}
catch
{
    MessageBox.Show("Error al intentar aprobar");
}
}

```

Rechazo de Solicitud

```

private void btnRechazar_Click(object sender, EventArgs e)
{
    try
    {
        int contador = 0;

        if (txtobservacion.Text != "")
        {
            ServiceInsumos.DTSSolicitudInsumo.TVJ_sol_insumoRow[] filaEdicion = null;
            dssolinsumo = servicioWeb.consultaSolicitudEstado(3);
            filaEdicion = dssolinsumo.TVJ_sol_insumo.Select("cod_solicitud=" + txtcodigo .Text + "")
as ServiceInsumos.DTSSolicitudInsumo.TVJ_sol_insumoRow[];

            DialogResult = MessageBox.Show(" ¿Está seguro que desea rechazar la solicitud?",
"Rechazo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

            if (DialogResult == DialogResult.Yes)
            {
                //se verifica si tiene activo 1 o mas checkbox,caso contrario quiere decir que no está
aprobando ningún item
                for (int i = 0; i < dgltems.Rows.Count; i++)
                {
                    if (Convert.ToBoolean(dgltems.Rows[i].Cells[2].Value) == true)
                    {
                        contador = contador + 1;
                    }
                }

                // si no tiene ninguna aprobación entonces se permite rechazar la solicitud

                if (contador ==0)
                {
                    filaEdicion[0].fecha = txtfecha.Text;
                    filaEdicion[0].usuario = txtusuario.Text;
                    filaEdicion[0].estado = 5;
                    filaEdicion[0].observacion = txtobservacion.Text + " Fecha de rechazo:" +
fecha.ToString("dd/MM/yyyy");

```

```

dssolinsumo.TVJ_sol_insumo.ImportRow(filaEdicion[0]);
servicioWeb.ingresoSolicitudInsumo(dssolinsumo);
cargarSolicitudPendiente();
MessageBox.Show("Solicitud Rechazada");

txtcódigo .Text = "";
txtfecha.Text = "";
txtusuario.Text = "";
dgltems.Rows.Clear();
txtobservacion.Text = "";
}
else
{
    MessageBox.Show("No se puede rechazar, la solicitud tiene 1 o más items
aprobados");
}
}
else
{
    txtcódigo .Text = "";
    txtfecha.Text = "";
    txtusuario.Text = "";
    dgltems.Rows.Clear();
}
}
else
{
    MessageBox.Show("Debe ingresar las observaciones del rechazo de la solicitud");
}
}
catch
{
    MessageBox.Show("Error al intentar rechazar");
}
}
}

```

Envío de solicitud a Compras

```

private void btnComprar_Click(object sender, EventArgs e)
{
    try
    {
        ServiceInsumos.DTSSolicitudInsumo.TVJ_sol_insumoRow[] filaEdicion = null;
        dssolinsumo = servicioWeb.consultaSolicitudEstado(4);
        filaEdicion = dssolinsumo.TVJ_sol_insumo.Select("cod_solicitud=" +
Convert.ToInt32(txtcódigo .Text) + """) as ServiceInsumos.DTSSolicitudInsumo.TVJ_sol_insumoRow[];

        //valida que el campo entrega este lleno ya que debe ingresar la fecha de entrega
para poder dar clic en el botón Comprar
        if (txtentrega.Text != "")
        {
            DialogResult = MessageBox.Show(" ¿Está seguro que desea enviar a Compras?",
"Compras", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

            if (DialogResult == DialogResult.Yes)
            {
                filaEdicion[0].fecha = txtfecha.Text;
            }
        }
    }
    catch
    {
    }
}

```

```

        filaEdicion[0].usuario = txtusuario.Text;
        filaEdicion[0].estado = 6;
        filaEdicion[0].observacion = "Fecha de compra:" + fecha.ToString("dd/MM/yyyy") + "
Fecha de entrega: " + txtentrega.Text;

        //edita en la tabla el estado de la solicitud, además ingresa la fecha de
entrega y de compra
        dssolinsumo.TVJ_sol_insumo.ImportRow(filaEdicion[0]);
        servicioWeb.ingresoSolicitudInsumo(dssolinsumo);
        //refresca las solicitudes
        cargarSolicitudesAprob();
        MessageBox.Show("La solicitud cambió su estado a Comprado, en espera de
entrega");
        CodSolicitudEnvia = Convert.ToInt32(txtcódigo .Text);
        FechaEntrega = txtentrega.Text;
        txtcódigo .Text = "";
        txtfecha.Text = "";
        txtusuario.Text = "";
        txtestado.Text = "";
        txtobservacion.Text = "";
        dgltems.Rows.Clear();
        //imprime el reporte enviando el número de solicitud
        Replmprime.CodSolicitudRecibe = CodSolicitudEnvia;
        Replmprime.FechaEntrega = FechaEntrega;
        Replmprime.Show();

    }
    else
    {
        txtcódigo .Text = "";
        txtfecha.Text = "";
        txtusuario.Text = "";
        txtestado.Text = "";
        txtobservacion.Text = "";
        txtentrega.Text = "";
        dgltems.Rows.Clear();
    }
}
else
{
    MessageBox.Show("Debe ingresar la fecha de entrega del insumo");
}
}
catch
{
    MessageBox.Show("Error al tratar de cambiar de estado");
}
}
}

```

Registro de Logística y cierre de flujo de la solicitud

```

private void btnRecibido_Click(object sender, EventArgs e)
{
    try
    {
        int contItem = 0;
    }
}

```

```

ServiceInsumos.DTSSolicitudInsumo.TVJ_sol_insumoRow[] filaEdicion = null;
dssolinsumo = servicioWeb.consultaSolicitudEstado(6);
filaEdicion = dssolinsumo.TVJ_sol_insumo.Select("cod_solicitud=" + txtcodigo .Text + "")
as ServiceInsumos.DTSSolicitudInsumo.TVJ_sol_insumoRow[];

DialogResult = MessageBox.Show(" ¿Está seguro que desea actualizar a Entregado?",
"Logística", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

//si ya se recibió los insumos de la solicitud, Logística acepta y cierra el flujo de la
solicitud
if (DialogResult == DialogResult.Yes)
{
filaEdicion[0].fecha = txtfecha.Text;
filaEdicion[0].usuario = txtusuario.Text;
filaEdicion[0].estado = 7;
filaEdicion[0].observacion = "Solicitud Finalizada. " + "Fecha de entrega:" +
fecha.ToString("dd/MM/yyyy");
//se envía a modificar el estado de la solicitud e indica que la solicitud finalizó su
flujo
dssolinsumo.TVJ_sol_insumo.ImportRow(filaEdicion[0]);
servicioWeb.ingresoSolicitudInsumo(dssolinsumo);

//Actualiza el stock de insumos en la tabla según el detalle de la solicitud y según
los items y cantidades aprobadas
DTSItemSolicitud.TVJ_itemSolicitudRow[] filaconsulta = null;
dsitemsol = servicioWeb.consultaItemsSolicitud(Convert.ToInt32(txtcodigo .Text));
filaconsulta = dsitemsol.TVJ_itemSolicitud.Select() as
DTSItemSolicitud.TVJ_itemSolicitudRow[];

contItem = filaconsulta.Length;

ServiceInsumos.DTSInsumos.TVJ_insumoRow[] filaEdicion2 = null;

for (int i = 0; i < contItem; i++)
{
// si tiene el item estado 4 quiere decir que se aprobó la cantidad total ingresada
if (filaconsulta[i].estado_item == 4)
{
dsinsumos = servicioWeb.consultaInsumos(9);
filaEdicion2 = dsinsumos.TVJ_insumo.Select("cod_insumo=" +
filaconsulta[i].cod_insumo + "") as ServiceInsumos.DTSInsumos.TVJ_insumoRow[];
filaEdicion2[0].stock_insumo = Convert.ToInt32(filaEdicion2[0].stock_insumo +
filaconsulta[i].cantidad);
dsinsumos.TVJ_insumo.ImportRow(filaEdicion2[0]);
servicioWeb.ingresoInsumos(dsinsumos);
dsinsumos.Clear();
}
else
{
// si el item tiene estado 8 quiere decir que la solicitud se aprobó
parcialmente y debe actualizar según el campo cantidad aprobada
if (filaconsulta[i].estado_item == 8)
{
dsinsumos = servicioWeb.consultaInsumos(9);
filaEdicion2 = dsinsumos.TVJ_insumo.Select("cod_insumo=" +
filaconsulta[i].cod_insumo + "") as ServiceInsumos.DTSInsumos.TVJ_insumoRow[];

```

```

                filaEdicion2[0].stock_insumo = Convert.ToInt32(filaEdicion2[0].stock_insumo +
filaconsulta[i].cantidad_aprobada);
                dsinsumos.TVJ_insumo.ImportRow(filaEdicion2[0]);
                servicioWeb.ingresoInsumos(dsinsumos);
                dsinsumos.Clear();
            }
        }

        }
        MessageBox.Show("La solicitud ha finalizado estado actual Entregado");
        cargarSolicitudesCompradas();
        CodSolicitudEnvia = Convert.ToInt32(txtcodigo .Text);
        FechaEntrega = fecha.ToString("dd/MM/yyyy");

        txtcodigo .Text = "";
        txtfecha.Text = "";
        txtusuario.Text = "";
        txtestado.Text = "";
        txtobservacion.Text = "";
        dgltems.Rows.Clear();
        //imprime el reporte
        Replmprime.CodSolicitudRecibe = CodSolicitudEnvia;
        Replmprime.FechaEntrega = FechaEntrega;
        Replmprime.Show();
    }
    else
    {
        txtcodigo .Text = "";
        txtfecha.Text = "";
        txtusuario.Text = "";
        txtestado.Text = "";
        txtobservacion.Text = "";
        dgltems.Rows.Clear();
    }
}
}
catch
{
    MessageBox.Show("Error al intentar entregar");
}
}
}

```

3.1.4.3 Mantenimiento

En el mantenimiento preventivo Manprecor genera rutinas de mantenimiento las cuales son ejecutadas a través de órdenes de trabajo asignadas para cada fecha de las rutinas.

El mantenimiento correctivo se administra bajo registro de fallas para cada maquinaria si es que alguna máquina presenta problemas en su funcionamiento.

3.1.4.3.1 Preventivo

Registro de Rutinas

```
private void btnIngresarRutina_Click(object sender, EventArgs e)
{
    try
    {
        //Controlamos que toda la información esté correctamente ingresada
        if (txtdescripcion.Text.Trim() != "" && uccFechaInicio.Text.Trim() != "" &&
txtFrecuencia.Text.Trim() != "" && ucFrecuencia.Text.Trim() != "" && txtDuracion.Text.Trim() != "" &&
ucDuracion.Text.Trim() != "")
        {
            DialogResult CampoAuxiliar = MessageBox.Show("Esta Seguro que desea Ingresar la
informacion ?", "Información", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Information);

            if (CampoAuxiliar == System.Windows.Forms.DialogResult.Yes)
            {
                if (rbPartes.Checked == true)
                {
                    if (cbPartesEquipo.Text != "")
                    {
                        //LLamamos al método de ingreso de rutinas
                        ingresoRutinas();
                    }
                    else
                    {
                        MessageBox.Show("Debe seleccionar la parte del equipo", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }
                }
                else
                {
                    ingresoRutinas();
                }
            }
            else
            {
                MessageBox.Show("Debe ingresar la información completa", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                txtdescripcion.Text = "";
                txtdescripcion.Focus();
            }
        }
    }
    catch (Exception ex)
    {
    }
}
```

```

        MessageBox.Show("Error al ingresar los datos,favor" , "Ingreso de Datos",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
public void ingresoRutinas()
{
    try
    {
        string diasAux = "";
        DTSRutinas dsrutinas = new DTSRutinas();
        DTSRutinas.TVJ_rutinaRow fila = dsrutinas.TVJ_rutina.NewTVJ_rutinaRow();

        //Obtenemos el usuario para ingresarlo en la rutina
        codUsuario = servicioWeb.consultaCódigo Usuario2(usuarioRefactorizado3);
        //Ingresamos la descripción y el equipo de la rutina
        fila.nom_rutina = txtdescripción.Text;
        fila.cod_equipo = código EquipoParte;

        if (rbPartes.Checked == true)
        {
            fila.cod_parte = codParte;
        }
        else
        {
            fila.cod_parte = 0;
        }
        //Ingresamos la información de la rutina
        fila.path_rutina = " ";
        fila.fecha_inicio = Convert.ToDateTime(uccFechaInicio.Text);
        fila.frecuencia = FrecuenciaDias;
        fila.duracion = DuracionDias;
        fila.cod_usuario = codUsuario;
        //llamamos al metodo para ingresar la informacion
        dsrutinas.TVJ_rutina.AddTVJ_rutinaRow(fila);
        servicioWeb.ingresoRutinas(dsrutinas);

        //rutina
        //inicio registro Auditoría
        DTSAuditoría dsAuditoría = new DTSAuditoría();
        DTSAuditoría.TVJ_AuditoríaRow filaAud =
dsAuditoría.TVJ_Auditoría.NewTVJ_AuditoríaRow();
        filaAud.módulo = "Mantenimiento";
        filaAud.nom_usuario = usuarioRefactorizado3;
        filaAud.accion = "Inserción";
        filaAud.fecha = DateTime.Now;
        filaAud.datos_afectados = "Tabla:TVJ_rutina/Datos:" + "Descripción " + txtdescripción.Text +
";Equipo " + código EquipoParte
        + "," + codParte + ";Fecha " + uccFechaInicio.Text + ";Frecuencia " + FrecuenciaDias +
";Duración " + DuracionDias;
        dsAuditoría.TVJ_Auditoría.AddTVJ_AuditoríaRow(filaAud);
        servicioWeb.ingresoRegistroAuditoría(dsAuditoría);
        //fin registro Auditoría

        //consulta el código de la rutina ingresada anteriormente
        int codRutina;
        DTSRutinas.TVJ_rutinaRow[] filaconsulta = null;
        dsrutinas = servicioWeb.consultaRutinas();
    }
}

```

```

filaconsulta = dsrutinas.TVJ_rutina.Select() as DTSRutinas.TVJ_rutinaRow[];
codRutina = filaconsulta[filaconsulta.Length - 1].cod_rutina;

//Ingreso de fechas de rutina en tabla
DateTime fecha_prox, fechaaux;
int numero_veces = 0;
DTSFechasRutina dsfechaRutinas = new DTSFechasRutina();
DTSFechasRutina.TVJ_fechas_rutinaRow fila2 =
dsfechaRutinas.TVJ_fechas_rutina.NewTVJ_fechas_rutinaRow();

numero_veces = DuracionDias / FrecuenciaDias;
fecha_prox=Convert.ToDateTime(uccFechaInicio.Text);

for (int i = 0; (i < numero_veces) && (fecha_prox <
Convert.ToDateTime(uccFechaInicio.Text).AddDays(DuracionDias)); i++)
{
    if (i == 0)
    {
        fila2.cod_rutina = codRutina;
        fila2.fecha_ejecucion = Convert.ToDateTime(uccFechaInicio.Text);
        fila2.estado = 13; //estado: ingresado
        dsfechaRutinas.TVJ_fechas_rutina.AddTVJ_fechas_rutinaRow(fila2);
        servicioWeb.ingresoFechasRutinas(dsfechaRutinas);
        dsfechaRutinas.Clear();
    }
    fechaaux = fecha_prox.AddDays(Convert.ToDouble(FrecuenciaDias));

    fechaaux = verificaFechasNoLaborables(fechaaux);

    if (chbSabados.Checked == false)
    {
        if (Convert.ToString(fechaaux.DayOfWeek) == "Sabado" ||
Convert.ToString(fechaaux.DayOfWeek) == "Saturday")
        {
            fechaaux = fechaaux.AddDays(2);
        }
    }
    if (chbDomingos.Checked == false)
    {
        if (Convert.ToString(fechaaux.DayOfWeek) == "Domingo" ||
Convert.ToString(fechaaux.DayOfWeek) == "Sunday")
        {
            fechaaux = fechaaux.AddDays(1);
        }
    }
}

fechaaux = verificaFechasNoLaborables(fechaaux);

fila2.cod_rutina=codRutina;
fila2.fecha_ejecucion = fechaaux;
fila2.estado =13; //estado: ingresado
dsfechaRutinas.TVJ_fechas_rutina.AddTVJ_fechas_rutinaRow(fila2);
servicioWeb.ingresoFechasRutinas(dsfechaRutinas);
fecha_prox = fechaaux;
dsfechaRutinas.Clear();
//fechas rutina
//inicio registro Auditoría
DTSAuditoría dsAuditoría2 = new DTSAuditoría();

```

```

        DTSAuditoría.TVJ_AuditoríaRow filaAud2 =
dsAuditoría2.TVJ_Auditoría.NewTVJ_AuditoríaRow();
        filaAud2.módulo = "Mantenimiento";
        filaAud2.nom_usuario = usuarioRefactorizado3;
        filaAud2.accion = "Inserción";
        filaAud2.fecha = DateTime.Now;
        filaAud2.datos_afectados = "Tabla:TVJ_fechas_rutina/Datos:" + "Código " + codRutina +
";Fecha " + fechaaux + ";Estado Ingresado";
        dsAuditoría2.TVJ_Auditoría.AddTVJ_AuditoríaRow(filaAud2);
        servicioWeb.ingresoRegistroAuditoría(dsAuditoría2);
        //fin registro Auditoría      }
        //pinta las fechas en el calendario
        pintaFechasRutina(codRutina);

```

```

        MessageBox.Show("Rutina Ingresada Correctamente", "Ingreso de Rutina",
        MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

//impresión del reporte
Reportes.frmRepRutinas repRutina1 = new Reportes.frmRepRutinas();
repRutina1.Código = codRutina;
repRutina1.DescripciónRut = txtdescripción.Text;
repRutina1.Equipo = nombreEq;
repRutina1.Parte = cbPartesEquipo.Text;
repRutina1.Fechainicio = uccFechalnicio.Text;
repRutina1.Frecuencia = txtFrecuencia.Text + " " + ucFrecuencia.Text;
repRutina1.Duracion = txtDuracion.Text + " " + ucDuracion.Text;
if (chbSabados.Checked == true)
{
    diasAux = diasAux + "Sabados" + "/";
}
if (chbDomingos.Checked == true)
{
    diasAux = diasAux + "Domingos" + "/";
}
repRutina1.Dias = diasAux;
repRutina1.Show();

```

```

cbPartesEquipo.Text = "";
txtdescripción.Text = "";
uccFechalnicio.Text = DateTime.Today.ToString();
txtFrecuencia.Text = "";
txtDuracion.Text = "";
ucFrecuencia.Text = "";
ucDuracion.Text = "";
chbSabados.Checked = false;
chbDomingos.Checked = false;
rbPartes.Checked = false;
rbEquipos.Checked = false;
ugbRutinas.Enabled = false;
}
catch
{
    MessageBox.Show("Error al tratar de ingresar");
}

```

```
}
```

Verifica en la rutina las fechas no laborables

```
public DateTime verificaFechasNoLaborables(DateTime fechaaux)
{
    try
    {
        DTSFechasNoLaborables dsfechasNL = new DTSFechasNoLaborables();
        DTSFechasNoLaborables.TVJ_fechas_nolaborableRow[] filaconsulta = null;
        //Verifica si en las fechas de rutinas
        dsfechasNL = servicioWeb.consultaFechasNoLaborables();
        filaconsulta = dsfechasNL.TVJ_fechas_nolaborable.Select() as
DTSFechasNoLaborables.TVJ_fechas_nolaborableRow[];

        for (int i = 0; i < filaconsulta.Length; i++)
        {
            if (fechaaux == Convert.ToDateTime(filaconsulta[i].fecha))
            {
                fechaaux = fechaaux.AddDays(1);
            }
        }
        return fechaaux;
    }
    catch
    {
        return fechaaux;
        MessageBox.Show("Error al tratar de ingresar");
    }
}
```

Órdenes de Trabajo

```
private void btnGuardarEstado_Click(object sender, EventArgs e)
{
    try
    {
        dsOrdenesTrabajo=servicioWeb.consultaOrdenesIngresadas();

        if (txtordenEstado.Text != "")
        {
            if (cbEstado.Text == "En espera")
            {
                DTSOrdenesTrabajo.TVJ_ordentrabajoRow[] filaEditar = null;
                filaEditar = dsOrdenesTrabajo.TVJ_ordentrabajo.Select("cod_orden=" +
Convert.ToInt32(txtordenEstado.Text) + "") as DTSOrdenesTrabajo.TVJ_ordentrabajoRow[];

                DialogResult = MessageBox.Show(" ¿Está seguro que desea modificar la orden de
trabajo?", "Editar Orden", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

                if (DialogResult == DialogResult.Yes)
                {
                    filaEditar[0].estado = 17;
                }
            }
        }
    }
}
```

```

servicioWeb.ingresoOrdenes(dsOrdenesTrabajo);
cargarOrdenes();
limpiarModificacionOrden();

//inicio registro Auditoría
DTSAuditoría dsAuditoría = new DTSAuditoría();
DTSAuditoría.TVJ_AuditoríaRow filaAud =
dsAuditoría.TVJ_Auditoría.NewTVJ_AuditoríaRow();
filaAud.módulo = "Mantenimiento";
filaAud.nom_usuario = usuarioRefactorizado4;
filaAud.accion = "Edición";
filaAud.fecha = DateTime.Now;
filaAud.datos_afectados = "Tabla:TVJ_ordentrabajo/Datos:" + "CódigoOrden " +
txtordenEstado.Text + ";Estado En espera";
dsAuditoría.TVJ_Auditoría.AddTVJ_AuditoríaRow(filaAud);
servicioWeb.ingresoRegistroAuditoría(dsAuditoría);
//fin registro Auditoría

MessageBox.Show("Orden modificada correctamente");

}
else
{
if (cbEstado.Text == "Cerrada" && uccFechaFin.Text != "" && txtobservacion.Text != "")
{

DTSOrdenesTrabajo.TVJ_ordentrabajoRow[] filaEditar = null;
filaEditar = dsOrdenesTrabajo.TVJ_ordentrabajo.Select("cod_orden="" +
Convert.ToInt32(txtordenEstado.Text) + """) as DTSOrdenesTrabajo.TVJ_ordentrabajoRow[];

DialogResult = MessageBox.Show(" ¿Está seguro que desea modificar la orden de
trabajo?", "Editar Orden", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (DialogResult == DialogResult.Yes)
{
filaEditar[0].estado = 18;
filaEditar[0].fechafin = Convert.ToDateTime(uccFechaFin.Text);
filaEditar[0].observacion = txtobservacion.Text;
servicioWeb.ingresoOrdenes(dsOrdenesTrabajo);

//inicio registro Auditoría
DTSAuditoría dsAuditoría = new DTSAuditoría();
DTSAuditoría.TVJ_AuditoríaRow filaAud =
dsAuditoría.TVJ_Auditoría.NewTVJ_AuditoríaRow();
filaAud.módulo = "Mantenimiento";
filaAud.nom_usuario = usuarioRefactorizado4;
filaAud.accion = "Edición";
filaAud.fecha = DateTime.Now;
filaAud.datos_afectados = "Tabla:TVJ_ordentrabajo/Datos:" + "CódigoOrden " +
txtordenEstado.Text + ";Estado Cerrada " + ";FechaCierre " + uccFechaFin.Text + ";Observación " +
txtobservacion.Text;
dsAuditoría.TVJ_Auditoría.AddTVJ_AuditoríaRow(filaAud);
servicioWeb.ingresoRegistroAuditoría(dsAuditoría);
//fin registro Auditoría

MessageBox.Show("Orden modificada correctamente");
}
}
}
}

```

```

//impresión del reporte
Reportes.frmRepOrdenIns repOrden2 = new Reportes.frmRepOrdenIns();
repOrden2.Orden = txtordenEstado.Text;
repOrden2.Estado = cbEstado.Text;
repOrden2.Fecharegistro = txtfechainicio.Text;
repOrden2.Fechamax = txtfechamax.Text;
repOrden2.Fechafin = uccFechaFin.Text;
repOrden2.Rutina = txtnomRutina.Text;
repOrden2.Equipo = txtnomEquipo.Text;
repOrden2.Parte = txtnomParte.Text;
repOrden2.Observacion = txtobservacion.Text;

repOrden2.Show();

//actualiza y limpia registros
cargarOrdenes();
limpiarModificacionOrden();
    }
    }
else
{
    MessageBox.Show("Debe ingresar toda la información");
}
}
}
else
{
    MessageBox.Show("Debe seleccionar una orden para modificar");
}
}
}
catch
{
}
}
}

```

Registro de Insumos en la orden

```

private void btnIngreso_Click(object sender, EventArgs e)
{
    try
    {
        int aux = 0;

        //Veificamos los insumos utilizados en la orden
        if (ugInsumosUtilizados.Rows.Count != 0)
        {
            for (int i = 0; i < ugInsumosUtilizados.Rows.Count; i++)
            {
                if (Convert.ToInt32(ugInsumosUtilizados.Rows[i].Cells["Cantidad Utilizada"].Value) ==
0)
                {
                    aux++;
                }
            }
        }
    }
}

```

```

    }
}

if (aux != 0)
{
    MessageBox.Show("Debe ingresar una cantidad de stock utilizado");
}
else
{
    DialogResult = MessageBox.Show(" Está seguro que desea ingresar la información?",
    "Ingreso Insumos", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

    if (DialogResult == DialogResult.Yes)
    {
        servicioWeb.ingresoInsumosUtilizados(dt);
        dt.Clear();
        ugInsumosUtilizados.DataSource = dt;

        //inicio registro Auditoría
        DTSAuditoría dsAuditoría = new DTSAuditoría();
        DTSAuditoría.TVJ_AuditoríaRow filaAud =
dsAuditoría.TVJ_Auditoría.NewTVJ_AuditoríaRow();
        filaAud.módulo = "Mantenimiento";
        filaAud.nom_usuario = usuarioRefactorizado4;
        filaAud.accion = "Inserción";
        filaAud.fecha = DateTime.Now;
        filaAud.datos_afectados = "Tabla:TVJ_ordeninsumo/Datos:" + "CódigoOrden " +
txtorden.Text + ";CantidadInsumos " + ugInsumosUtilizados.Rows.Count;
        dsAuditoría.TVJ_Auditoría.AddTVJ_AuditoríaRow(filaAud);
        servicioWeb.ingresoRegistroAuditoría(dsAuditoría);
        //fin registro Auditoría

        MessageBox.Show("Los insumos se ingresaron correctamente");

        txtorden.Text = "";
        txtbusqueda.Text = "";
        dt.Clear();

    }

}

}
else
{
    MessageBox.Show("Debe seleccionar al menos 1 insumo");
}

}
catch (Exception)
{

```

```

        MessageBox.Show("Debe ingresar una cantidad válida de stock utilizado");
    }
}

```

3.1.4.3.2 Correctivo

Registro de Fallas

```

private void btnGuardar_Click(object sender, EventArgs e)
{
    try
    {
        //Inicializo las variables
        DTSPFallas dsfallas = new DTSPFallas();
        DTSPFallas.TVJ_fallasRow fila = dsfallas.TVJ_fallas.NewTVJ_fallasRow();
        gbSolucion.Visible = false;
        fila.desc_fallas = txtdescripcionSol.Text;
        fila.fecha_fallas = Convert.ToDateTime(txtfechaSol.Text);
        fila.soluc_fallas = txtSolucion.Text;
        fila.cod_equides= codEq ;
        fila.cod_parte = codParte;

        DialogResult resultado;

        resultado = MessageBox.Show("Está seguro que desea ingresar la información",
"Confirmación", MessageBoxButtons.YesNo);

        if (resultado == System.Windows.Forms.DialogResult.Yes &&
txtdescripcionSol.Text.Trim()!="")
        {
            //Llamamos al método de ingreso de fallas
            dsfallas.TVJ_fallas.AddTVJ_fallasRow(fila);
            servicioWeb.ingresoFallasEquipos(dsfallas);

            //inicio registro Auditoría
            DTSAuditoría dsAuditoría2 = new DTSAuditoría();
            DTSAuditoría.TVJ_AuditoríaRow filaAud2 =
dsAuditoría2.TVJ_Auditoría.NewTVJ_AuditoríaRow();
            filaAud2.módulo = "Mantenimiento";
            filaAud2.nom_usuario = usuarioRefactorizado15;
            filaAud2.accion = "Inserción";
            filaAud2.fecha = DateTime.Now;
            filaAud2.datos_afectados = "Tabla:TVJ_fallas/Datos:" + "Descripción " +
txtdescripcionSol.Text + ";Fecha " + txtfechaSol.Text
            + ";Solución " + txtSolucion.Text + ";CódigoEq " + codEq + ";CódigoParte " +
codParte;

            dsAuditoría2.TVJ_Auditoría.AddTVJ_AuditoríaRow(filaAud2);
            servicioWeb.ingresoRegistroAuditoría(dsAuditoría2);
            //fin registro Auditoría

            MessageBox.Show("Dato Ingresado Correctamente");

            //consulta código de la falla y envia al reporte

            DTSPFallas dsfallas2 = new DTSPFallas();

```

```

DTSFallas.TVJ_fallasRow[] filaconsulta = null;
dsfallas2 = servicioWeb.consultaFallas();
filaconsulta = dsfallas2.TVJ_fallas.Select() as DTSFallas.TVJ_fallasRow[];
codFallaEnvia = filaconsulta[filaconsulta.Length - 1].cod_fallas;
repFalla1.CodFalla = codFallaEnvia;
repFalla1.CodTipo = codParte;
repFalla1.Show();

//refrescar botón
cbPartesEquipo.Text = "";
txtdescripción.Text = "";
txtdescripciónSol.Text = "";
txtfechaSol.Text = "";
txtSolucion.Text = "";
txtdescripción.Text = "";
rbPartes.Checked = false;
rbEquipos.Checked = false;

}
else
{
    MessageBox.Show("Debe ingresar la información completa");
    txtdescripciónSol.Text = "";
    txtfechaSol.Text = "";
    txtSolucion.Text = "";
    cbPartesEquipo.Text = "";
    txtdescripción.Text = "";
    rbPartes.Checked = false;
    rbEquipos.Checked = false;
}
}
catch
{
    MessageBox.Show("Error al Ingresar el Dato,Favor intente nuevamente");
}
}

```

3.1.4.4 Análisis de información

3.1.4.4.1 Reportes

Consultas Mantenimiento

```

private void btnBuscar_Click(object sender, EventArgs e)
{
    try
    {
        //Setemos la variable tipo vector de string para ingresar los parámetros

        string[] parametros = new string[5];

        parametros[0] = txtEquipos.Text;
    }
}

```

```

parametros[1] = ucUbicaciones.Text;
parametros[2] = ucTiposEquipo.Text;
parametros[3] = cbEstado.Text;
parametros[4] = ucProveedorEquipo.Text;

//Recorremos el vector de los parámetros para conocer que parámetros fueron enviados
for (int i = 0; i < parametros.Length; i++)
{
    if (parametros[i] == "")
    {
        parametros[i] = null;
    }
}

//Enviamos los parámetros al método de consulta
consultaEquipos = servicioWeb.EquiposConsulta(parametros);
ugEquipos.DataSource = consultaEquipos;

}
catch (Exception ex)
{

}

}
}

```

Exportar Consultas a Excel

```

private void btnExportarExcelEquipos_Click(object sender, EventArgs e)
{
    try
    {
        //Seteamos la fecha y hora actual
        string fecha = DateTime.Now.ToString();
        string fecha1 = fecha.Replace('/', '-').Replace(':', '-').Replace(' ', '-');
        //Creamos una variable tipo string para crear el directorio donde se guardaran los archivos
        excel
        string directorio = "";
        string path = "\\DocumentosExcel\\ConsultaEquipos" + fecha1 + ".xls";
        directorio = "C:\\Manprecor_Archivos" + path;
        //Si no existe el directorio lo creamos
        DirectoryInfo di = new DirectoryInfo("C:\\Manprecor_Archivos" + "\\DocumentosExcel");
        if (di.Exists == false)
        {
            di.Create();
        }

        //Creamos una variable de tipo Infragistics.Excel.Workbook para ingresar los datos de la
        consulta
        Infragistics.Excel.Workbook libroExcel1 = new Infragistics.Excel.Workbook();
        libroExcel1.Worksheets.Add("hoja1");
        libroExcel1.Worksheets.Add("hoja2");

        libroExcel1.ActiveWorksheet.Rows[0].Cells[0].CellFormat.Font.Bold =
        Infragistics.Excel.ExcelDefaultableBoolean.True;
        libroExcel1.ActiveWorksheet.Rows[0].Cells[0].CellFormat.Font.Height = 350;
        libroExcel1.ActiveWorksheet.Rows[2].Cells[0].Value = "Fecha: ";
    }
}

```

```

        libroExcel1.ActiveWorksheet.Rows[2].Cells[2].Value = DateTime.Today.Year.ToString() + "/"
+ DateTime.Today.Month.ToString() + "/" + DateTime.Today.Day.ToString();

        //ugExcelExporter.Export(ugrvDetalleCompromiso, libroExcel1, 8, 0);

        //Exportamos toda la informacion de la grilla a excel
        ExportarExcel.Export(ugEquipos, libroExcel1, 0, 0);
        Infragistics.Excel.BIFF8Writer.WriteWorkbookToFile(libroExcel1, directorio);
        System.Diagnostics.Process process = new System.Diagnostics.Process(); //creamos un
proceso
        process.EnableRaisingEvents = false;
        process.StartInfo.FileName = "excel"; // aqui ponemos el programa que queremos ejecutar
        process.StartInfo.Arguments = directorio; //parametros
        process.Start(); //ejecutamos el programa
        process.WaitForExit(); // esperamos a que termine
    }
    catch (Exception ex)
    {

    }
}
}

```

Consulta Insumos

```

private void btnBuscar_Click(object sender, EventArgs e)
{
    try
    {
        //Seteamos la variable tipo vector de string para ingresar los parámetros
        string[] parametros = new string[6];

        parametros[0] = txtnombre.Text;
        parametros[1] = txtstock.Text;
        parametros[2] = txtminimo.Text;
        parametros[3] = txtEstado.Text;
        parametros[4] = cbtipo.Text;
        parametros[5] = cbproveedor.Text;
        //recorre el vector
        for (int i = 0; i < 6; i++)
        {
            if (parametros[i] == "")
            {
                parametros[i] = null;
            }
        }

        //se envia los parámetros al método de consulta
        dsInsumos =servicioWeb.consultaReporteInsumos(parametros);
        grConsultaInsumos.DataSource = dsInsumos;

    }
    catch
    {
    }
}

```

```
        MessageBox.Show("Error al cargar los datos ", "Aviso", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
```

```
    }
}
```

Exportar consultas a Excel

```
private void btnExportarExcel_Click_1(object sender, EventArgs e)
{
    try
    {
        //Seteamos la fecha y hora actual
        string fecha = DateTime.Now.ToString();
        string fecha1 = fecha.Replace('/', '-').Replace(':', '-').Replace(' ', '-');
        //Creamos una variable tipo string para crear el directorio donde se guardaran los
archivos excel
        string directorio = "";
        string path = "\\DocumentosExcel\\ConsultaInsumos" + fecha1 + ".xls";
        directorio = "C:\\Manprecor_Archivos" + path;
        //Si no existe el directorio lo creamos
        DirectoryInfo di = new DirectoryInfo("C:\\Manprecor_Archivos" + "\\DocumentosExcel");
        if (di.Exists == false)
        {
            di.Create();
        }
        //Creamos una variable de tipo Infragistics.Excel.Workbook para ingresar los datos
de la consulta
        Infragistics.Excel.Workbook libroExcel1 = new Infragistics.Excel.Workbook();
        libroExcel1.Worksheets.Add("hoja1");
        libroExcel1.Worksheets.Add("hoja2");

        libroExcel1.ActiveWorksheet.Rows[0].Cells[0].CellFormat.Font.Bold =
Infragistics.Excel.ExcelDefaultableBoolean.True;
        libroExcel1.ActiveWorksheet.Rows[0].Cells[0].CellFormat.Font.Height = 350;
        libroExcel1.ActiveWorksheet.Rows[2].Cells[0].Value = "Fecha: ";
        libroExcel1.ActiveWorksheet.Rows[2].Cells[2].Value = DateTime.Today.Year.ToString() + "/"
+ DateTime.Today.Month.ToString() + "/" + DateTime.Today.Day.ToString();

        //ugExcelExporter.Export(ugrvDetalleCompromiso, libroExcel1, 8, 0);
        //Exportamos toda la información de la grilla a excel
        ExportarExcel.Export(grConsultaInsumos, libroExcel1, 0, 0);
        Infragistics.Excel.BIFF8Writer.WriteWorkbookToFile(libroExcel1, directorio);
        System.Diagnostics.Process process = new System.Diagnostics.Process(); //creamos un
proceso
        process.EnableRaisingEvents = false;
        process.StartInfo.FileName = "excel"; // aquí ponemos el programa que queremos ejecutar
        process.StartInfo.Arguments = directorio; //parámetros
        process.Start(); //ejecutamos el programa
        //process.WaitForExit(); // esperamos a que termine
    }
    catch (Exception ex)
    {}
}
```

3.1.4.4.2 Seguridad

La seguridad de la información es un tema sumamente importante debido a que se maneja datos confidenciales que deben ser almacenados con seguridad para evitar que sean mal utilizados, robados o alterados.

Un sistema debe estar en la capacidad de proteger la confidencialidad, integridad y disponibilidad de la información que se maneja a través de métodos ó estrategias que eviten los riesgos de mala utilización de la información almacenada en el sistema, por esta razón se desarrolló un algoritmo de encriptación propio de la aplicación Manprecor, cuyo proceso se detalla a continuación:

Para realizar la encriptación se creó un algoritmo que cifra cada letra de la frase, al recibirla el algoritmo descompone las palabras en letras y a cada letra asigna una codificación que se forma de la siguiente manera:

Existen 10 posiciones para cada letra de las cuales seis se forman de códigos alfanuméricos asignados aleatoriamente, una posición pertenece a la letra real que estamos cifrando y las últimas 3 posiciones se forman de letras tomadas de una tabla de equivalencias previamente definida, cuya combinación revela la posición en la cual se encuentra la letra de la palabra.

Este proceso se realiza para todas las letras de la frase, y al final se entrega una cadena alfanumérica que es utilizada como información para guardar en la base de datos.

Para la desencriptación de la frase se realiza el proceso inverso.

```
public string Llamar_encriptacion (string cadena_encripta){  
    cadenaEncriptar = cadena_encripta;  
    Encriptacion1(cadenaEncriptar);  
  
    resultado_encripta = cadena_total;  
    return resultado_encripta;  
}  
  
public void Encriptacion1(string cadena)  
{  
    for (int i = 0; i < cadena.Length; i++)  
    {
```

```

        aux = cadena.Substring(i, 1);
        EncriptaLetra(aux);
        aux = "";
    }
}

public void EncriptaLetra(string letra) {
    string[] código = new string[10];
    int posicion = ran.Next(0, 6);
    Posiciones_encripta(posicion,código );
    for (int i = 0; i < 7; i = i + 1) {
        if (i == posicion)
        {
            código [i] = letra;
        }
        else {
            //int randomEntero = ran.Next(0, 125);
            int randomEntero = ran.Next(65, 90);
            código [i] = (Convert.ToChar(randomEntero)).ToString();
            randomEntero = 0;
        }
    }
    cadena_total += código [0] + código [1] + código [2] + código [3] + código [4] + código [5] +
código [6] + código [7] + código [8] + código [9];
    posicion = 0;
}

}

public void Posiciones_encripta(int posicion, string[] pos_encriptada)
{
    switch (posicion) {
        case 0:
            pos_encriptada[7] = "y";
            pos_encriptada[8] = "a";
            pos_encriptada[9] = "l";
            break;
        case 1:
            pos_encriptada[7] = "x";
            pos_encriptada[8] = "i";
            pos_encriptada[9] = "s";
            break;
        case 2:
            pos_encriptada[7] = "p";
            pos_encriptada[8] = "c";
            pos_encriptada[9] = "w";
            break;
        case 3:
            pos_encriptada[7] = "h";
            pos_encriptada[8] = "r";
            pos_encriptada[9] = "m";
            break;
        case 4:
            pos_encriptada[7] = "u";
            pos_encriptada[8] = "v";
            pos_encriptada[9] = "k";
    }
}

```

```

        break;
    case 5:
        pos_encryptada[7] = "f";
        pos_encryptada[8] = "o";
        pos_encryptada[9] = "d";
        break;
    case 6:
        pos_encryptada[7] = "t";
        pos_encryptada[8] = "n";
        pos_encryptada[9] = "j";
        break;
    }
}

```

Proceso de descriptación

```

public string Llamar_descriptacion(string cadena){
    cadenaDescriptar = cadena;

    for (int i = 0; i < ((cadenaDescriptar.Length)/10); i++)
    {
        aux2 = cadenaDescriptar.Substring((i*10), 10);
        DescriptaLetra(aux2);
        aux2 = "";
    }

    resultado_descripta=palabra;

    return resultado_descripta;
}

public void DescriptaLetra(string letra)
{
    int posicion = 0;
    posicion=Posiciones_descripta(letra);

    for (int i = 0; i < 7; i = i + 1)
    {
        if (i == posicion)
        {
            palabra +=letra.Substring(i, 1);
        }
    }
}

public int Posiciones_descripta(string letra)
{

```

```

int posicion = 0;
string lugar = "";
lugar = letra.Substring(7,3);

switch (lugar)
{
    case "yal":
        posicion = 0;
        break;
    case "xis":
        posicion = 1;
        break;

    case "pcw":
        posicion = 2;
        break;

    case "hrm":
        posicion = 3;
        break;

    case "uvk":
        posicion = 4;
        break;

    case "fod":
        posicion = 5;
        break;

    case "tnj":
        posicion = 6;
        break;
}
return posicion;
}
}

```

3.1.4.5 Capas

Capa de Negocio

//Perfiles

```

public class PerfilesPro
{
    DatosBase.IngPerfilesBase db = new IngPerfilesBase();
    public DTSPerfiles consultaPefiles()
    {
        return new IngPerfilesBase().consultaPefiles();
    }
    //consulta codigo del perfil
    public string consultaCodigoPerfil(string nombre)
    {
        string codperfil;
        DTSPerfiles dscodigoPerfil = new DTSPerfiles();
        dscodigoPerfil = db.consultaCodigoPerfil(nombre);

        try
        {
            DTSPerfiles.TVJ_perfilesRow[] filaconsulta = null;
            filaconsulta = dscodigoPerfil.TVJ_perfiles.Select("nombre_perfil=" + "nombre_perfil" + "") as
DTSPerfiles.TVJ_perfilesRow[];
            codperfil = Convert.ToString(filaconsulta[0].cod_perfil);
            return codperfil;
        }
        catch
        {
            codperfil = "";
            return codperfil;
        }
    }
    public int ingresoPerfil(DTSPerfiles dsperfiles)
    {
        return new IngPerfilesBase().ingresoPerfil(dsperfiles);
    }

    public DTSPerfiles consultaAccesos(string nombre)
    {
        return new IngPerfilesBase().consultaCodigoPerfil(nombre);
    }
}

//Rutinas
public class RutinasPro
{
    public int ingresoRutinas(DTSRutinas dsRutinas)
    {
        return new IngRutinasBase().ingresoRutinas(dsRutinas);
    }
    public DTSRutinas consultaRutinas()
    {
        return new IngRutinasBase().consultaRutinas();
    }
    public DataSet consultaEquiposRutinas()
    {
        return new IngRutinasBase().consultaEquiposRutinas();
    }
    public DTSRutinas consultaRutinasEqParte(int cod_equipo, int cod_parte)

```

```

{
    return new IngRutinasBase().consultaRutinasEqParte(cod_equipo, cod_parte);
}
public DataSet RutinasConsulta(string[] Parametros)
{

    string NombreEquipo = null;
    string NombreRutina = null;
    string Fecha = null;
    string Usuario = null;

    DataSet Rutinas = new DataSet();

    IngRutinasBase Rutinas1 = new IngRutinasBase();

    if (Parametros[0] != null)
    { NombreEquipo = Parametros[0].ToString(); }
    if (Parametros[1] != null)
    { NombreRutina = Parametros[1].ToString(); }
    if (Parametros[2] != null)
    { Fecha = Parametros[2].ToString(); }
    if (Parametros[3] != null)
    { Usuario = Parametros[3].ToString(); }

    Rutinas = Rutinas1.RutinasConsulta(NombreEquipo, NombreRutina, Fecha, Usuario);

    return Rutinas; }

//Usuarios

public class UsuarioPro
{
    Utilitario.Encryptacion ut = new Encryptacion();
    DatosBase.IngUsuarioBase db = new IngUsuarioBase();

    //consulta si la clave que el usuario ingreso esta correcta
    public Boolean consultaClaveUsuario(string usuario, string clave)
    {

        string clave_base, clave_des;
        bool resultado=false;
        DTSUsuario dsusuario = new DTSUsuario();
        dsusuario=db.consultaClaveUsuario(usuario, clave);

        try
        {
            DTSUsuario.TVJ_usuarioRow[] filaconsulta = null;
            filaconsulta = dsusuario.TVJ_usuario.Select("nombre_usuario=" + "nombre_usuario" + "") as
DTSUsuario.TVJ_usuarioRow[];
            clave_base = filaconsulta[0].clave_usuario;
            clave_des = ut.Llamar_desencryptacion(clave_base);

            if (clave_des == clave)
            {
                resultado = true;
            }
            else
            {

```

```

        resultado = false;
    }
}
catch {
}
return resultado;
}

```

Capa de Datos

//Conexión Base

```

public abstract class conexionBase
{
    public Database baseDatos;

    public conexionBase()
    {
        baseDatos = DatabaseFactory.CreateDatabase("ConectaSQL");
    }

    protected virtual Database BaseDatos
    {
        get
        {
            return this.baseDatos;
        }
    }
}

```

//Ordenes

```

public class IngOrdenesBase:conexionBase
{
    public DataSet consultaOrdenes(int codUsuario, int tipoRutina)
    {
        DataSet dsOrdenes = new DataSet();
        string sql = "exec sp_tvj_consultar_ordenes @codUsuario,@tipoRutina";
        DbCommand consulta = baseDatos.GetSqlStringCommand(sql);
        baseDatos.AddInParameter(consulta, "@codUsuario", DbType.Int32, codUsuario);
        baseDatos.AddInParameter(consulta, "@tipoRutina", DbType.Int32, tipoRutina);
        baseDatos.LoadDataSet(consulta, dsOrdenes, "TVJ_ordenes");
        return dsOrdenes;
    }

    public DataSet consultaInsumos(string nombre)
    {

```

```

DataSet dsInsumos = new DataSet();
string sql = "exec sp_consultar_insumo_orden @nombre";
DbCommand consulta = baseDatos.GetSqlStringCommand(sql);
baseDatos.AddInParameter(consulta, "@nombre", DbType.StringFixedLength, nombre);
baseDatos.LoadDataSet(consulta, dsInsumos, "TVJ_insumos");
return dsInsumos;
}

public int ingresoOrdenes(DTSOrdenesTrabajo dsOrdenes)
{
    string sql = "INSERT INTO ";
    sql += "dbo.TVJ_ordentrabajo ";
    sql += "VALUES (";
    sql += "@cod_rutina,@descripcion,@fechainicio,@fechafin,@fechamaxima,@estado) ";

    DbCommand insert = baseDatos.GetSqlStringCommand(sql);

    baseDatos.AddInParameter(insert, "@cod_rutina", DbType.Int32, "cod_rutina",
DataRowVersion.Current);
    baseDatos.AddInParameter(insert, "@descripcion", DbType.StringFixedLength, "descripcion",
DataRowVersion.Current);
    baseDatos.AddInParameter(insert, "@fechainicio", DbType.DateTime, "fechainicio",
DataRowVersion.Current);
    baseDatos.AddInParameter(insert, "@fechafin", DbType.DateTime, "fechafin",
DataRowVersion.Current);
    baseDatos.AddInParameter(insert, "@fechamaxima", DbType.DateTime, "fechamaxima",
DataRowVersion.Current);
    baseDatos.AddInParameter(insert, "@estado", DbType.Int32, "estado",
DataRowVersion.Current);
    baseDatos.AddInParameter(insert, "@observacion", DbType.StringFixedLength,
"observacion", DataRowVersion.Current);

    sql = "UPDATE ";
    sql += "dbo.TVJ_ordentrabajo ";
    sql += "SET "
        + "estado = @estado,fechafin=@fechafin,observacion=@observacion "
        + "WHERE "
        + "cod_orden = @cod_orden";

    DbCommand update = baseDatos.GetSqlStringCommand(sql);
    baseDatos.AddInParameter(update, "@cod_orden", DbType.Int32, "cod_orden",
DataRowVersion.Current);
    baseDatos.AddInParameter(update, "@estado", DbType.Int32, "estado",
DataRowVersion.Current);
    baseDatos.AddInParameter(update, "@fechafin", DbType.DateTime, "fechafin",
DataRowVersion.Current);
    baseDatos.AddInParameter(update, "@observacion", DbType.StringFixedLength,
"observacion", DataRowVersion.Current);

    DbCommand delete = baseDatos.GetSqlStringCommand(sql);

```

```

        return baseDatos.UpdateDataSet(dsOrdenes, "TVJ_ordentrabajo", insert, update, delete,
UpdateBehavior.Standard);
    }

    public DTSOrdenesTrabajo consultaOrdenesIngresadas()
    {
        DTSOrdenesTrabajo dsordenes = new DTSOrdenesTrabajo();
        string sql = "select * from TVJ_ordentrabajo;";
        DbCommand consulta = baseDatos.GetSqlStringCommand(sql);
        baseDatos.LoadDataSet(consulta, dsordenes, "TVJ_ordentrabajo");
        return dsordenes;
    }

    public int ingresoInsumosUtilizados(int CodOrden,int CodInsumo,int StockUtilizado)
    {
        string sql = "exec sp_ingreso_insumo_orden @cod_orden,@cod_insumo,@stock_utilizado";

        DbCommand insert = baseDatos.GetSqlStringCommand(sql);
        baseDatos.AddInParameter(insert, "@cod_orden", DbType.Int32, CodOrden);
        baseDatos.AddInParameter(insert, "@cod_insumo", DbType.Int32, CodInsumo);
        baseDatos.AddInParameter(insert, "@stock_utilizado", DbType.Int32, StockUtilizado);

        baseDatos.ExecuteNonQuery(insert);

        string sql1 = "exec sp_actualizar_stock_insumo @cod_insumo,@stock_utilizado";
        DbCommand consulta = baseDatos.GetSqlStringCommand(sql1);
        baseDatos.AddInParameter(consulta, "@cod_insumo", DbType.Int32, CodInsumo);
        baseDatos.AddInParameter(consulta, "@stock_utilizado", DbType.Int32, StockUtilizado);

        return baseDatos.ExecuteNonQuery(consulta);
    }

    public DataSet consultaOrdenesAbiertas(int codUsuario)
    {
        DataSet dsOrdenesAbiertas = new DataSet();
        string sql = "exec sp_tvj_consulta_ordenes @codUsuario";
        DbCommand consulta = baseDatos.GetSqlStringCommand(sql);
        baseDatos.AddInParameter(consulta, "@codUsuario", DbType.Int32, codUsuario);
        baseDatos.LoadDataSet(consulta, dsOrdenesAbiertas, "TVJ_ordenes");
        return dsOrdenesAbiertas;
    }

    public DataSet consultaInsumosOrdenes(int codOrden)
    {
        DataSet dsInsumosOrdenes = new DataSet();
        string sql = "select i.nombre_insumo as Insumo,stock_utilizado as Cantidad from
tvj_ordeninsumo o,tvj_insumo i where o.cod_insumo=i.cod_insumo and cod_orden=@cod_orden ";
        DbCommand consulta = baseDatos.GetSqlStringCommand(sql);
        baseDatos.AddInParameter(consulta, "@cod_orden", DbType.Int32, codOrden);
        baseDatos.LoadDataSet(consulta, dsInsumosOrdenes, "Tabla");
        return dsInsumosOrdenes;
    }

```

```

}

public int anulacionOrdenesCaducadas()
{
    string sql = "exec sp_anulacion_orden";
    DbCommand update = baseDatos.GetSqlCommand(sql);
    baseDatos.ExecuteNonQuery(update);

    return baseDatos.ExecuteNonQuery(update);
}

public DataSet OrdenesConsulta(string NombreEquipo, string NombreRutina, string FechaMin,
string FechaMax, string Estado)
{
    DataSet dsOrdenes = new DataSet();
    string sql = "exec sp_tvj_consulta_ordenes_trabajo
    @i_NombreEquipo,@i_NombreRutina,@i_FechaFallaMin,@i_FechaFallaMax,@i_estado ";
    DbCommand ejecutar = baseDatos.GetSqlCommand(sql);

    if (FechaMin != null)
    {
        Convert.ToDateTime(FechaMin);
    }

    if (FechaMax != null)
    {
        Convert.ToDateTime(FechaMax);
    }
    baseDatos.AddInParameter(ejecutar, "@i_NombreEquipo", DbType.StringFixedLength,
NombreEquipo);
    baseDatos.AddInParameter(ejecutar, "@i_NombreRutina", DbType.StringFixedLength,
NombreRutina);
    baseDatos.AddInParameter(ejecutar, "@i_FechaFallaMin", DbType.DateTime, FechaMin);
    baseDatos.AddInParameter(ejecutar, "@i_FechaFallaMax", DbType.DateTime, FechaMax);
    baseDatos.AddInParameter(ejecutar, "@i_estado", DbType.StringFixedLength, Estado);

    baseDatos.LoadDataSet(ejecutar, dsOrdenes, "Consultas");
    return dsOrdenes;
}
}

```

3.2 ETAPA DE PRUEBAS

3.2.1 PRUEBAS FUNCIONALES

Las pruebas de funcionalidad están orientadas a pulir las prestaciones que ofrece MANPRECOR a sus usuarios para entregar un producto final de calidad con la menor cantidad de errores en su entorno.

Las pruebas tienen en su estructura información como: la forma o el formulario en el cual se está interactuando así como la funcionalidad que se está probando, podemos documentar los valores que ingresamos de ser necesario y el resultado obtenido luego del proceso, si fue óptimo o no, de no ser así se toma los correctivos necesarios y se vuelve a realizar la prueba hasta que el resultado sea el esperado.

Con toda esta información podemos garantizar que las pruebas de funcionalidad nos ayudarán a velar por el correcto funcionamiento de nuestro sistema MANPRECOR.

Módulo de Insumos

Nº	1
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	FrmTipoInsumo.cs
Descripción de la funcionalidad	Ingreso inicial de los tipos de insumos que se maneja en el módulo
Datos ingresados	Lubricantes
Resultado esperado	Ingreso no satisfactorio
Resultado Obtenido	Ingreso satisfactorio
Observación	Se ingresó un dato duplicado, no existe validación para duplicidad de información

Nº	2
Fecha	27/06/2010

Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	FrmTipoInsumo.cs
Descripción de la funcionalidad	Ingreso inicial de los tipos de insumos que se maneja en el módulo
Datos ingresados	NULL
Resultado esperado	Ingreso no satisfactorio
Resultado Obtenido	Ingreso no satisfactorio
Observación	Si valida que no ingresen datos nulos

Nº	3
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	FrmTipoInsumo.cs
Descripción de la funcionalidad	Ingreso inicial de los tipos de insumos que se maneja en el módulo
Datos ingresados	Aditivos
Resultado esperado	Ingreso satisfactorio
Resultado Obtenido	Ingreso satisfactorio
Observación	OK

Nº	4
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	FrmTipoInsumo.cs
Descripción de la funcionalidad	Ingreso inicial de los tipos de insumos que se maneja en el módulo
Datos ingresados	(espacio en blanco)

Resultado esperado	Ingreso no satisfactorio
Resultado Obtenido	Ingreso satisfactorio
Observación	Se ingresó datos en blanco, no valida que no ingrese espacios en blanco

Nº	5
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	FrmTipoInsumo.cs
Descripción de la funcionalidad	Modificación de la información de un tipo de insumo
Datos ingresados	NULL
Resultado esperado	Edición no satisfactoria
Resultado Obtenido	Edición no satisfactoria
Observación	Si valida que no edite si no existen datos

Nº	6
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	FrmTipoInsumo.cs
Descripción de la funcionalidad	Modificación de la información de un tipo de insumo
Datos ingresados	Utilería
Resultado esperado	Edición satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	OK

Nº	7
Fecha	27/06/2010
Nombre del	Insumos

módulo	
Capa	Presentación
Clase	Insumos.dll
Nombre del form	FrmTipoInsumo.cs
Descripción de la funcionalidad	Modificación de la información de un tipo de insumo
Datos ingresados	Aditivos
Resultado esperado	Edición no satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	Se editó un dato con nombre ya existente, no existe validación para duplicidad de información

Nº	8
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	FrmTipoInsumo.cs
Descripción de la funcionalidad	Eliminación de la información de un tipo de insumo
Datos ingresados	NULL
Resultado esperado	Eliminación no satisfactoria
Resultado Obtenido	Eliminación no satisfactoria
Observación	Si valida que exista información a eliminar

Nº	9
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	FrmTipoInsumo.cs
Descripción de la funcionalidad	Eliminación de la información de un tipo de insumo

Datos ingresados	Aditivos
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	OK

Nº	10
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	FrmTipoinsumo.cs
Descripción de la funcionalidad	Eliminación de la información de un tipo de insumo
Datos ingresados	Carpintería
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	OK

Nº	11
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	Proveedores
Descripción de la funcionalidad	Ingreso inicial de los proveedores de insumos
Datos ingresados	NULL
Resultado esperado	Ingreso no satisfactorio
Resultado Obtenido	Ingreso no satisfactorio
Observación	Si valida que no ingresen datos nulos

Nº	12
Fecha	27/06/2010
Nombre del módulo	Insumos

Capa	Presentación
Clase	Insumos.dll
Nombre del form	Proveedores
Descripción de la funcionalidad	Ingreso inicial de los proveedores de insumos
Datos ingresados	(espacio en blanco)
Resultado esperado	Ingreso no satisfactorio
Resultado Obtenido	Ingreso satisfactorio
Observación	Se ingresó datos en blanco, no valida que no ingrese espacios en blanco

Nº	13
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	Proveedores
Descripción de la funcionalidad	Ingreso inicial de los proveedores de insumos
Datos ingresados	Agipac;Granda centeno OE 4123 y Carondelet;02234567
Resultado esperado	Ingreso satisfactorio
Resultado Obtenido	Ingreso satisfactorio
Observación	OK

Nº	14
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	Proveedores
Descripción de la funcionalidad	Ingreso inicial de los proveedores de insumos
Datos ingresados	Agipac;Granda centeno OE 4123 y Carondelet;02234567
Resultado	Ingreso no satisfactorio

esperado	
Resultado Obtenido	Ingreso satisfactorio
Observación	Se ingresaron datos duplicados, no existe validación para duplicidad de información

Nº	15
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	Proveedores
Descripción de la funcionalidad	Modificación de la información de un proveedor de insumo
Datos ingresados	NULL
Resultado esperado	Edición no satisfactoria
Resultado Obtenido	Edición no satisfactoria
Observación	OK

Nº	16
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	Proveedores
Descripción de la funcionalidad	Modificación de la información de un proveedor de insumo
Datos ingresados	(espacio en blanco)
Resultado esperado	Edición no satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	Se editó con datos en blanco, no valida que no ingrese espacios en blanco

Nº	17
Fecha	27/06/2010
Nombre del	Insumos

módulo	
Capa	Presentación
Clase	Insumos.dll
Nombre del form	Proveedores
Descripción de la funcionalidad	Modificación de la información de un proveedor de insumo
Datos ingresados	Agipac;Granda centeno OE 4123 y Carondelet;02234567
Resultado esperado	Edición no satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	Se editó con datos duplicados, no existe validación para duplicidad de información

Nº	18
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	Proveedores
Descripción de la funcionalidad	Modificación de la información de un proveedor de insumo
Datos ingresados	Induval;Av.6 de diciembre y Patria;02245099
Resultado esperado	Edición satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	OK

Nº	19
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	Proveedores
Descripción de la funcionalidad	Eliminación de la información de los proveedores de insumos
Datos ingresados	NULL

Resultado esperado	Eliminación no satisfactoria
Resultado Obtenido	Eliminación no satisfactoria
Observación	Si valida que exista información a eliminar

Nº	20
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	Proveedores
Descripción de la funcionalidad	Eliminación de la información de los proveedores de insumos
Datos ingresados	Induval;Av.6 de diciembre y Patria;02245099
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	OK

Nº	21
Fecha	27/06/2010
Nombre del módulo	Insumos
Capa	Presentación
Clase	Insumos.dll
Nombre del form	frmInsumos.cs
Descripción de la funcionalidad	Ingreso de datos en blanco
Datos ingresados	Ninguno
Resultado esperado	Mensaje De Error
Resultado Obtenido	Satisfactorio
Observación	OK (Cambiar el texto del mensaje de error)

Nº	22
Fecha	27/06/2010
Nombre del módulo	Insumos

Capa	Presentación
Clase	Insumos.dll
Nombre del form	frmInsumos.cs
Descripción de la funcionalidad	Ingreso de datos
Datos ingresados	Insumo de prueba,32,lubricantes,Agipag,67,10,unidades,VeronicaHerrera,12/07/2010
Resultado esperado	Ingreso exitoso
Resultado Obtenido	Ingreso exitoso
Observación	OK

Módulo de Mantenimiento

Nº	1
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmEquipos.cs
Descripción de la funcionalidad	Inserción de la información de equipos
Datos ingresados	Impresora;Computadores; Ubicación: Ubicaciones\edificio\piso uno;Fecha de compra:06/04/2010;Tiempo:18 meses;Dimensión:20 cm;Peso:3 kilos;Capacidad:4 libras;Marca:HP;Prioridad:Alta;Proveedor:Brighstar;Estado:activo
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	Debe pasar con TAB; no valida que la fecha de compra no sea mayor a la actual

Nº	2
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación

Clase	Mantenimiento
Nombre del form	frmEquipos.cs
Descripción de la funcionalidad	Edición de la información de equipos
Datos ingresados	Impresora;Computadores; Ubicación: Ubicaciones\edificio\pisos; Fecha de compra:06/04/2010;Tiempo:18 meses;Dimensión:20 cm;Peso:3 kilos;Capacidad:4 libras;Marca:HP;Prioridad:Alta;Proveedor:Brighstar;Estado:activo
Resultado esperado	Edición satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	OK

Nº	3
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmEquipos.cs
Descripción de la funcionalidad	Eliminación de la información de equipos
Datos ingresados	Impresora;Computadores; Ubicación: Ubicaciones\edificio\pisos; Fecha de compra:06/04/2010;Tiempo:18 meses;Dimensión:20 cm;Peso:3 kilos;Capacidad:4 libras;Marca:HP;Prioridad:Alta;Proveedor:Brighstar;Estado:activo
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	Ok

Nº	4
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación

Clase	Mantenimiento
Nombre del form	frmEquipos.cs
Descripción de la funcionalidad	Inserción de la partes del equipo
Datos ingresados	Cartucho
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	OK

Nº	5
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmEquipos.cs
Descripción de la funcionalidad	Edición de la partes del equipo
Datos ingresados	Toner
Resultado esperado	Edición satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	OK

Nº	6
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmEquipos.cs
Descripción de la funcionalidad	Eliminación de la partes del equipo

Datos ingresados	Toner
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	OK

Nº	7
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmEquipos.cs
Descripción de la funcionalidad	Información adicional
Datos ingresados	Documento1
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	OK

Nº	8
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmEquipos.cs
Descripción de la funcionalidad	Ubicaciones
Datos ingresados	Edificio
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria

Observación	OK
--------------------	----

Nº	9
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmDocumentacionEquipo.cs
Descripción de la funcionalidad	Inserción de imágenes de los equipos
Datos ingresados	IMA1.jpg
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	No refresca el panel de imagen después de la inserción

Nº	10
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmDocumentacionEquipo.cs
Descripción de la funcionalidad	Eliminación de imágenes de los equipos
Datos ingresados	IMA1.jpg
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	No valida espacios en blanco

Nº	11
Fecha	07/10/2010
Nombre del	Mantenimiento

módulo	
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmDocumentacionEquipo.cs
Descripción de la funcionalidad	Inserción de documentos de los equipos
Datos ingresados	doc1.pdf
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	OK

Nº	12
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmDocumentacionEquipo.cs
Descripción de la funcionalidad	Eliminación de documentos de los equipos
Datos ingresados	doc1.pdf
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	OK

Nº	13
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento

Nombre del form	frmTipoEquipos.cs
Descripción de la funcionalidad	Inserción de los tipos de equipos
Datos ingresados	Motores
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	OK

Nº	14
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmTipoEquipos.cs
Descripción de la funcionalidad	Inserción de los tipos de equipos
Datos ingresados	(espacio en blanco)
Resultado esperado	Inserción no satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	No verifica duplicidad de información y espacios en blanco

Nº	15
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmTipoEquipos.cs
Descripción de la funcionalidad	Edición de los tipos de equipos

Datos ingresados	Motor
Resultado esperado	Edición satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	OK

Nº	16
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmTipoEquipos.cs
Descripción de la funcionalidad	Eliminación de los tipos de equipos
Datos ingresados	Motor
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	OK

Nº	17
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmUnidades.cs
Descripción de la funcionalidad	Inserción de unidades
Datos ingresados	Metros;longitud
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria

Observación	OK
--------------------	----

Nº	18
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmUnidades.cs
Descripción de la funcionalidad	Inserción de unidades
Datos ingresados	Metros;longitud
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	OK

Nº	19
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmUnidades.cs
Descripción de la funcionalidad	Inserción de unidades
Datos ingresados	(espacio en blanco)
Resultado esperado	Inserción no satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	No verifica espacios en blanco

Nº	20
Fecha	07/10/2010

Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmUnidades.cs
Descripción de la funcionalidad	Edición de unidades
Datos ingresados	(espacio en blanco)
Resultado esperado	Edición no satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	No verifica espacios en blanco

Nº	21
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmUnidades.cs
Descripción de la funcionalidad	Eliminación de unidades
Datos ingresados	Decímetros
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	OK

Nº	22
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento

Nombre del form	frmListaProveedores.cs
Descripción de la funcionalidad	Inserción de proveedores
Datos ingresados	Brighstar;República y Pradera;2222333
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	OK

Nº	23
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmListaProveedores.cs
Descripción de la funcionalidad	Inserción de proveedores
Datos ingresados	(espacio en blanco)
Resultado esperado	Inserción no satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	No verifica espacios en blanco

Nº	24
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmListaProveedores.cs
Descripción de la funcionalidad	Edición de proveedores

Datos ingresados	Brighstar;República y Pradera;2222444
Resultado esperado	Edición satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	OK

Nº	25
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmListaProveedores.cs
Descripción de la funcionalidad	Eliminación de proveedores
Datos ingresados	Brighstar;República y Pradera;2222444
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	OK

Nº	26
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmIngEmpleados.cs
Descripción de la funcionalidad	Inserción de empleados
Datos ingresados	1715635148;Doris Herrera;Mitad del mundo;2225556;095051645;Asistente;doris@hotmail.com;Activo
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción no satisfactoria

Observación	No valida cuando la CI está incompleta, no permite borrar números, no valida campo correo, no ingresa nuevo registro
--------------------	--

Nº	27
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmIngEmpleados.cs
Descripción de la funcionalidad	Edición de empleados
Datos ingresados	1715635148;Doris Herrera;12 de Octubre y Colón;2225556;095051645;Asistente;doris@hotmail.com;Activo
Resultado esperado	Edición satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	Ok

Nº	28
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmIngEmpleados.cs
Descripción de la funcionalidad	Eliminación de empleados
Datos ingresados	1715635148;Doris Herrera;12 de Octubre y Colon;2225556;095051645;Asistente;doris@hotmail.com;Activo
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	Ok

Nº	29
Fecha	07/10/2010

Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmIngUsuario.cs
Descripción de la funcionalidad	Inserción de usuarios y asignación de perfiles
Datos ingresados	1415635148;Doris Herrera;(clave);11/07/2010;Activo
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción no satisfactoria
Observación	OK

Nº	30
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmIngUsuario.cs
Descripción de la funcionalidad	Edición de usuarios y perfiles
Datos ingresados	1415635148;Doris Herrera;(clave);11/07/2010;Bloqueado
Resultado esperado	Edición satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	OK

Nº	31
Fecha	07/10/2010
Nombre del módulo	Mantenimiento

Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmIngUsuario.cs
Descripción de la funcionalidad	Edición de usuarios y perfiles
Datos ingresados	1415635148;(espacio en blanco);(clave);11/07/2010;Bloqueado
Resultado esperado	Edición no satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	No valida espacios en blanco

Nº	32
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmIngUsuario.cs
Descripción de la funcionalidad	Edición de usuarios y perfiles
Datos ingresados	Perfil: consulta
Resultado esperado	Edición satisfactoria
Resultado Obtenido	Edición no satisfactoria
Observación	Al eliminar un perfil borra todos los de la tabla con ese cod de perfil

Nº	33
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmIngUsuario.cs

Descripción de la funcionalidad	Eliminación de usuarios
Datos ingresados	Usuario:1715635148
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	Al eliminar un usuario únicamente se desactiva y no se elimina los perfiles;

Nº	34
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmPerfiles.cs
Descripción de la funcionalidad	Inserción de perfiles
Datos ingresados	Temporal; Proveedores
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	No refresca perfiles de insumos después del ingreso

Nº	35
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmPerfiles.cs
Descripción de la funcionalidad	Edición de perfiles
Datos ingresados	Temporal2;Consultas

Resultado esperado	Edición satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	OK

Nº	36
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmPerfiles.cs
Descripción de la funcionalidad	Eliminación de perfiles
Datos ingresados	Temporal2;Consultas
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	OK

Nº	37
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmFechasNoLaborable.cs
Descripción de la funcionalidad	Inserción de fechas no laborables
Datos ingresados	24/12/2010; Noche buena
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	No valida espacios en blanco

Nº	38
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmFechasNoLaborable.cs
Descripción de la funcionalidad	Edición de fechas no laborables
Datos ingresados	25/12/2010;Navidad
Resultado esperado	Edición satisfactoria
Resultado Obtenido	Edición satisfactoria
Observación	No valida espacios en blanco

Nº	39
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmFechasNoLaborable.cs
Descripción de la funcionalidad	Eliminación de fechas no laborables
Datos ingresados	25/12/2010;Navidad
Resultado esperado	Eliminación satisfactoria
Resultado Obtenido	Eliminación satisfactoria
Observación	OK

Nº	40
Fecha	07/10/2010
Nombre del	Mantenimiento

módulo	
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmRutinas.cs
Descripción de la funcionalidad	Inserción de Rutinas de mantenimiento
Datos ingresados	Computadora; Limpieza general; 5 días;6 meses;Sabados
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	OK

Nº	41
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmRutinas.cs
Descripción de la funcionalidad	Inserción de Rutinas de mantenimiento
Datos ingresados	Computadora; (espacio en blanco); 5 días;6 meses; Sábados
Resultado esperado	Inserción no satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	No valida espacios en blanco

Nº	42
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento

Nombre del form	frmRutinas.cs
Descripción de la funcionalidad	Inserción de Rutinas de mantenimiento
Datos ingresados	Computadora; Monitor; Limpieza general; 5 días; 6 meses; Sábados
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	Se corrige problema con dato enviado al reporte

Nº	43
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmOrdenes.cs
Descripción de la funcionalidad	Inserción de Ordenes de trabajo
Datos ingresados	Orden de trabajo para limpieza general de equipo
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	No valida espacios en blanco

Nº	44
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmOrdenes.cs

Descripción de la funcionalidad	Inserción de Insumos en las Ordenes de trabajo
Datos ingresados	Insumos: 3 Aceites
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	Validar que se borre la tabla que compara duplicidad de insumos.

Nº	45
Fecha	07/10/2010
Nombre del módulo	Mantenimiento
Capa	Presentación
Clase	Mantenimiento
Nombre del form	frmFallas.cs
Descripción de la funcionalidad	Inserción de Fallas
Datos ingresados	Corrección de banda en máquina 3
Resultado esperado	Inserción satisfactoria
Resultado Obtenido	Inserción satisfactoria
Observación	OK

3.2.2 PRUEBAS DE CARGA

La realización de las pruebas de carga para el sistema MANPRECOR fueron realizadas con el software WAPT 7.0 que tiene la funcionalidad de emular la carga de usuarios que accedan simultáneamente al sistema pudiendo valorar parámetros como ancho de banda utilizado así como también los datos transmitidos satisfactoriamente, como nuestro sistema está compuesto de dos módulos, vamos a realizar la prueba en los servicios Web de cada uno de ellos.

El servidor que vamos a utilizar para las pruebas será la máquina en donde se desarrolló el software la cual tiene las siguientes características:

HARDWARE	CAPACIDAD
Disco Duro	160 Gbytes
Procesador	AMD Turion x2 de 1.9 GHz
Memoria RAM	2 Gbytes

Módulo de Mantenimiento

Para el módulo de mantenimiento estos fueron los resultados obtenidos con 10 usuarios simultáneamente, la página a la cual se accedió fue <http://localhost:1064/ServiceManprecor/Service.asmx>

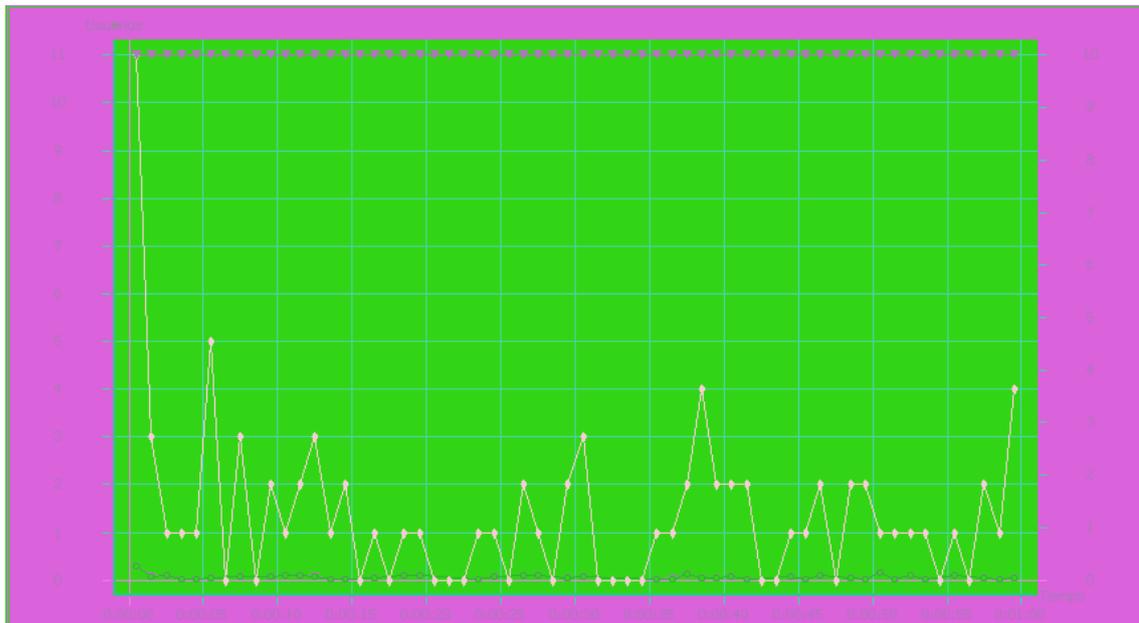
Perfil	Sesiones exitosas	Sesiones fallidas	Páginas exitosas	Páginas fallidas	Accesos Exitosos	Accesos Fallidos	Total KBytes enviados	Total KBytes recibidos	Teiempo de respuesta promedio, seg (con elementos de la página)
Manprecor Pruebas	21	0	83	0	83	0	27.7	1,119	0.1 (0.1)
Total	21	0	83	0	83	0	27.7	1,119	0.1 (0.1)

Fig#21. Resultados del módulo de mantenimiento

Fuente: Los autores

Gráficamente representado sería:

Rendimiento



Fig#22. Gráfico de rendimiento del módulo de mantenimiento

Fuente: Los autores

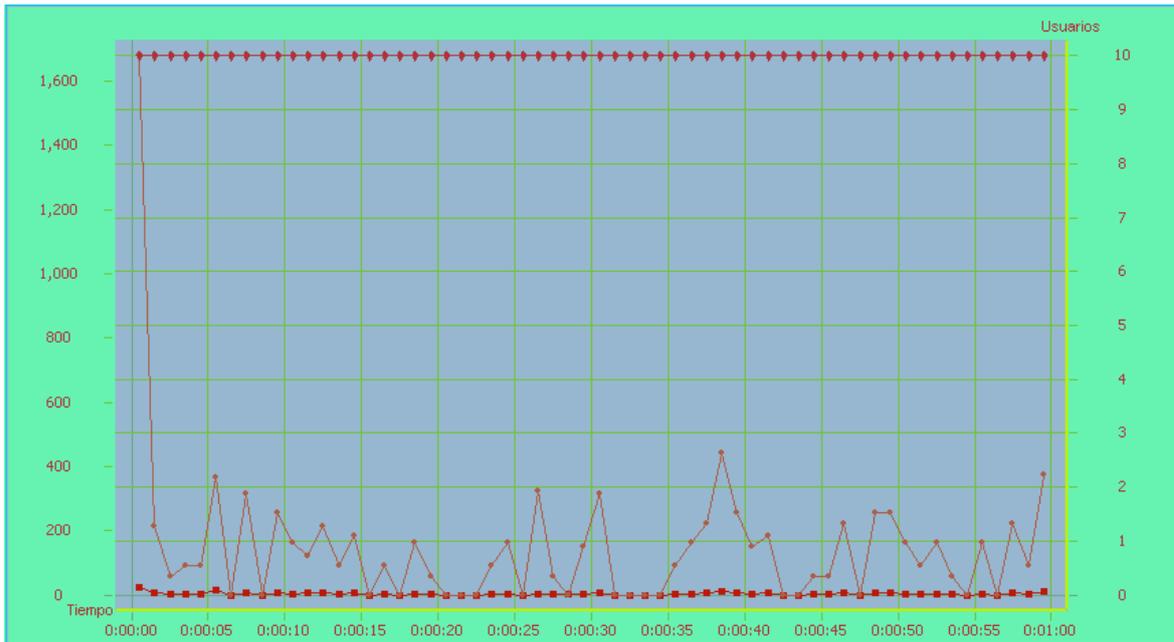
Línea Negra: Usuarios conectados.

Línea anaranjada: Muestreo por segundo del sistema.

Línea celeste: Datos que se cargaron en la aplicación en la unidad de tiempo.

Eje X= Tiempo en el cual se hizo la prueba
Eje Y = Usuarios que utilizan el sistema

Ancho de Banda utilizado



Fig#23. Gráfico de ancho de banda del módulo de mantenimiento
Fuente: Los autores

Línea Negra: Usuarios conectados.

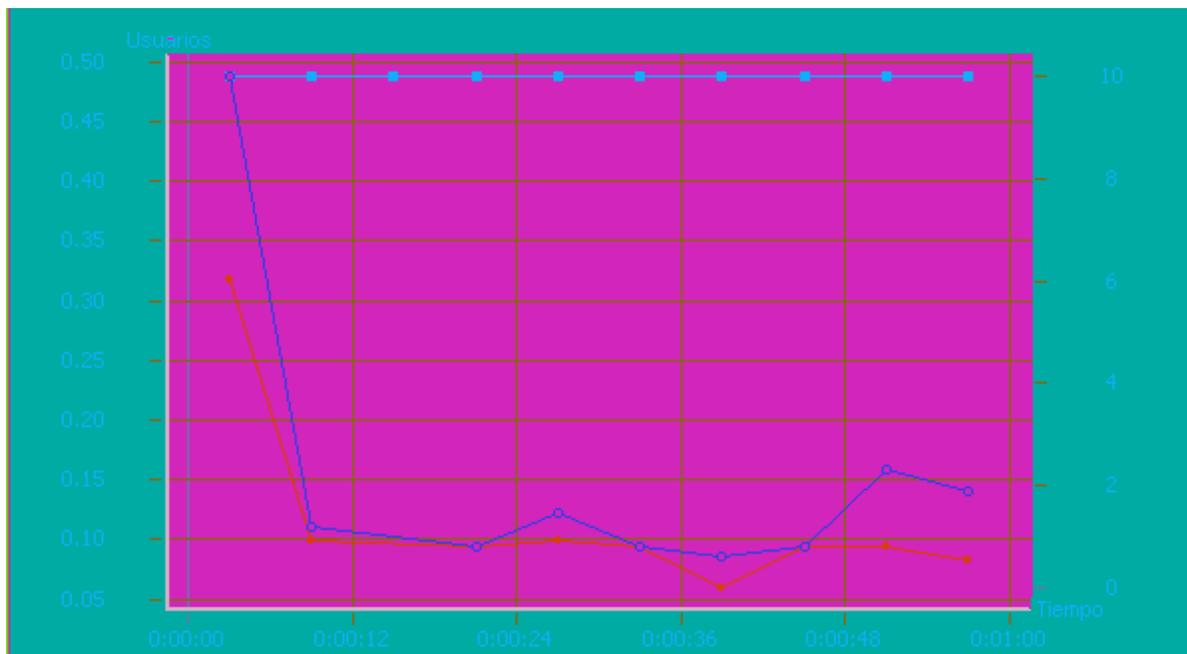
Línea Verde: Kbits enviados.

Línea Crema: Kbits recibidos.

Eje X = Tiempo en el cual se hizo la prueba.

Eje Y= Usuarios que utilizan el sistema.

Tiempo De Respuesta



Fig#24. Gráfico de tiempo de respuesta del módulo de mantenimiento
Fuente: Los autores

Línea Negra: Usuarios conectados.

Línea Rosada: Tiempo de respuesta.

Línea Roja: Promedio de tiempo respuesta.

Eje X = Tiempo en el cual se hizo la prueba.

Eje Y= Usuarios que utilizan el sistema.

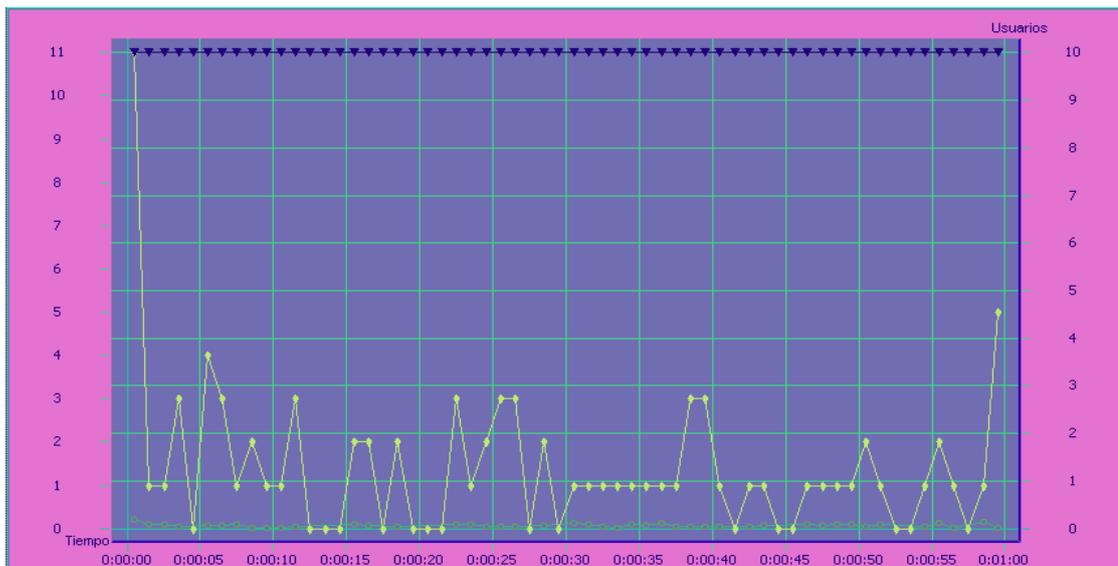
Módulo de Insumos

Resumen

Perfil	Sesiones exitosas	Sesiones fallidas	Páginas exitosas	Páginas fallidas	Accesos Exitosos	Accesos Fallidos	Total KBytes enviados	Total KBytes recibidos	Teimpo de respuesta promedio, seg (con elementos de la página)
Manprecor Pruebas	21	0	85	0	85	0	28.5	1,142	0.09 (0.09)
Total	21	0	85	0	85	0	28.5	1,142	0.09 (0.09)

Fig#25. Resumen del módulo de insumos
Fuente: Los autores

Rendimiento



Fig#26. Gráfico de Rendimiento del módulo de insumos
Fuente: Los autores

Línea Negra: Usuarios conectados.

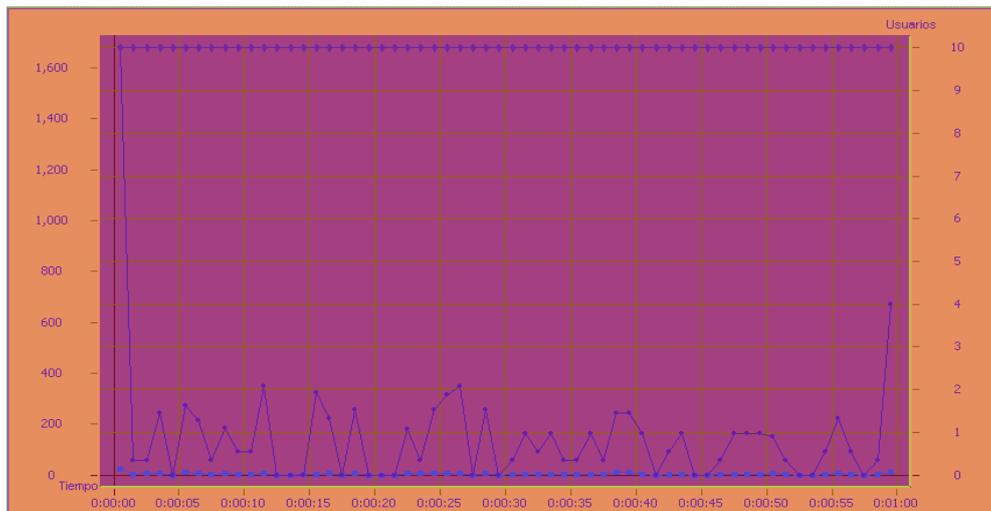
Línea anaranjada: Muestreo por segundo del sistema.

Línea celeste: Datos que se cargaron en la aplicación en la unidad de tiempo.

Eje X = Tiempo en el cual se hizo la prueba

Eje Y= Usuarios que utilizan el sistema

Ancho de Banda Utilizado



Fig#27. Gráfico de ancho de banda del módulo de insumos
Fuente: Los autores

Línea Negra: Usuarios conectados.

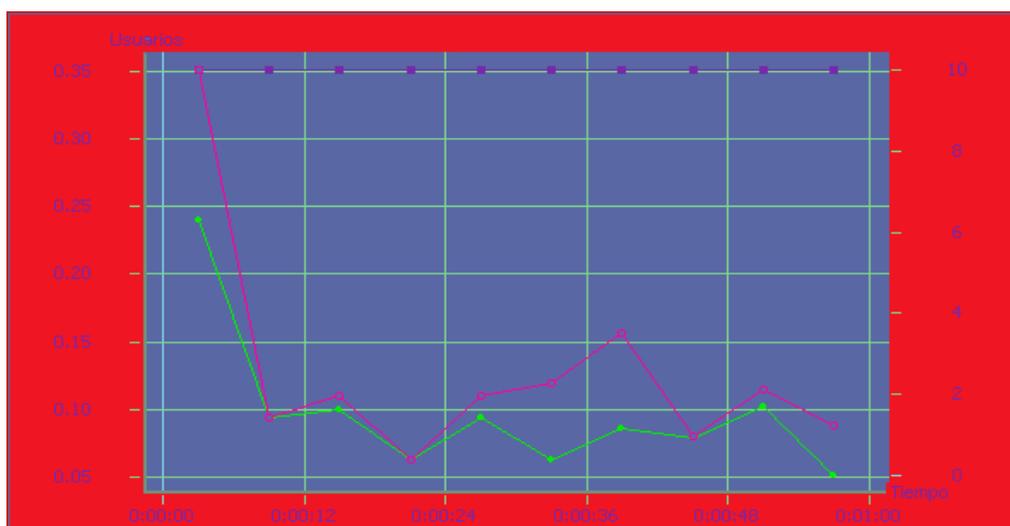
Línea Verde: Kbits enviados.

Línea Crema: Kbits recibidos.

Eje X = Tiempo en el cual se hizo la prueba.

Eje Y= Usuarios que utilizan el sistema.

Tiempo De Respuesta



Fig#28. Gráfico de tiempo de respuesta del módulo de insumos
Fuente: Los autores

Línea Negra: Usuarios conectados.

Línea Rosada: Tiempo de respuesta.

Línea Roja: Promedio de tiempo respuesta.

Eje X = Tiempo en el cual se hizo la prueba.

Eje Y= Usuarios que utilizan el sistema.

CAPITULO 4.

4.1 CONCLUSIONES

- Realizar las consultas del sistema mediante procedimientos almacenados en base de datos resulta una forma más fácil de aumentar o disminuir los parámetros de búsqueda y los campos que se requiere que se muestren en la búsqueda ya que no hay que realizar cambios dentro de las capas del programa sino solo en el procedimiento almacenado.
- Utilizar componentes Infragistics en el diseño del sistema ayudó a mejorar la apariencia de la interfaz, además de brindar más prestaciones de funcionalidad que los componentes normales que Visual Studio ofrece.
- Colocar el mismo nombre en los métodos creados en las diferentes capas ayuda a reconocer fácilmente el flujo de la información así como disminuir el tiempo en la corrección de errores
- Tener dentro de una empresa una herramienta que permita controlar el mantenimiento de maquinaria es de suma importancia ya que nos permitiría tener un mayor control sobre el equipo, alarga la vida útil de la maquinaria y minimiza los tiempos muertos por fallas.
- Integrar el mantenimiento con la utilización de insumos más la posibilidad de realizar solicitudes desde el propio departamento de mantenimiento nos permitiría disminuir los tiempos de obtención del insumo que se necesita para el funcionamiento de la maquinaria.
- Realizar un sistema con métodos publicados en un servicio Web permite que la herramienta sea modular ya que se facilita compartir información con otros sistemas, además de tener toda la información centralizada dentro de un mismo sitio.
- Tener toda la estructura en capas ayuda a integrar de manera más fácil y óptima cambios y actualizaciones sobre el sistema.

- La herramienta desarrollada permitiría administrar las rutinas de mantenimiento de una manera eficaz reduciendo los tiempos muertos y maximizando la productividad.
- Manprecor puede ser utilizado en industrias que manejen maquinaria pesada así como también en pequeñas empresas que deseen mantener sus equipos funcionando correctamente, ya que el sistema es aplicable para cualquier tipo de maquinaria.

4.2 RECOMENDACIONES

- Se recomienda tomar en cuenta todas las especificaciones técnicas del equipo el cual vamos a ingresar en el proceso de mantenimiento para poder conocer a fondo todas sus partes y funcionalidades y así optimizar su funcionamiento.
- Es importante verificar las especificaciones técnicas en el momento de la instalación para no tener inconvenientes a futuro por computadores no adecuados.
- El buen desempeño del Sistema depende del correcto ingreso de la parametrización inicial tanto del módulo de mantenimiento como del módulo de insumos; esto permitirá un óptimo funcionamiento.
- Es recomendable en futuras actualizaciones al sistema que las notificaciones de las rutinas a realizarse se envíen a un dispositivo móvil de la persona a quien está destinada la realización de la rutina.
- A la herramienta se le puede agregar la funcionalidad de multilenguaje para que pueda ser utilizado en el mercado extranjero.

BIBLIOGRAFÍA

- Herramienta Clickonces:
<http://www.microsoft.com/spanish/msdn/vstudio/Express/Docs/ClickOnce/ClickOnce.msp>
- EnterpriseLibrary:
msdn.microsoft.com/en-us/library/cc467894.aspx
- Conceptos de manteminiento correctivo y preventivo:
www.monografias.com/trabajos15/mantenimiento-industrial/mantenimiento-industrial.shtml

www.agapea.com/.../TECNICAS-PARA-EL-MANTENIMIENTO-Y-DIAGNOSTICO-DE-MÁQUINAS
<http://industrial.timberframe-house.com/mantenimiento.htm>
- Arquitectura SOA:
www.oracle.com/global/lad/technologies/soa/index.html
- Bases de datos
Mc Graw Hill, Fundamentos de bases de datos, España 2002,Cuarta Edición, ISBN=84-481-3654-3.
- Lenguaje C#
Mc Graw Hill, Superutilidades para C# , España 2002,Segúnda Edición, ISBN=0-07-219379-4.
- Lenguaje de Modelado Unificado
Forwel,Martin ,UML Gota a Gota ,México 1999,Segúnda Edición ,Páginas 1,2, ISBN: 968-444-364-1.
- Ingeniería de software
SommerVille Ian, Ingeniería de software, México 2002, Sexta Edición ISBN=970-26-0206-8.

ANEXOS

MANUAL EXPLICATIVO DE LA BASE DE DATOS

TABLA DE EQUIPOS (TVJ_EQUIDES)

Se detalla todas las características propias y técnicas del equipo como son: color, la marca, fecha de compra, garantía, tipo de máquina, peso, dimensión, etc. Con esta información se tiene un inventario de la maquinaria existente en la empresa o industria.

Con la fecha de compra se obtiene un dato muy importante que es la antigüedad del equipo mediante el cual se puede incluso conocer si el equipo cumplió o no el tiempo de la garantía.

Según la marca del equipo se puede conocer a que proveedor se debe acudir en caso de un problema.

Un requerimiento bastante útil e importante es la ubicación del equipo, esto facilita la localización en el momento que se busca el equipo para el mantenimiento.

Según el tipo de equipo se tiene asociado un código que pertenece al nombre del proveedor que se ubicaría en caso de mantenimiento.

Según la empresa o industria se crea una jerarquía de acuerdo al tipo de cada máquina para conocer si es primaria o si es parte complementaria de algún equipo.

Cada equipo tiene un ID que identifica a que tipo de maquinaria pertenece y cual es su lugar de acuerdo a la jerarquía previamente establecida.

En esta tabla se almacenan todos los equipos que posee la empresa o industria.

Se cuenta además con campos auxiliares para almacenamiento de información adicional de darse el caso.

Además cada equipo ingresa con un grado de prioridad, información que será de utilidad a la hora de realizar el mantenimiento.

TABLA DE IMÁGENES (TVJ_EQUIMA)

En esta tabla se almacena una ó varias imágenes que pertenecen a cada equipo ingresado en la base de datos.

Las imágenes se pueden guardar en cualquier lugar (path) del computador.

No se tiene restricción en cuanto al número de imágenes que se pueden ingresar en el sistema por cada equipo.

TABLA DE DOCUMENTOS (TVJ_EQUIDOC)

La tabla de documentos así como la tabla de imágenes, almacena paths de los documentos que posee cada equipo, estos archivos pueden ser manuales, diagramas, planos, documentos técnicos, etc.

Se puede guardar uno o varios documentos por cada equipo y en cualquier tipo de formato.

TABLA DE RUTINAS (TVJ_RUTINA)

Cada rutina es almacenada en una tabla con un identificador, una breve descripción del tipo de proceso que se realiza y con el path de localización del documento de la rutina, en el cual se explica todo el procedimiento de dicha rutina

Se puede ingresar una ó varias rutinas.

TABLA DE EQUIPOS CON RUTINAS (TVJ_EQUIRUT)

Se ingresa el código de la rutina con el código del equipo al cual corresponde dicha rutina, formándose así la relación que existe entre los dos.

Las rutinas pueden aplicarse a varios tipos de equipos, cada equipo puede tener varias rutinas asociadas a él, y cada rutina puede asociarse a varios tipos de equipos.

Cada rutina asociada a un equipo tiene una orden de trabajo, aquí se indica el código de la orden generada.

TABLA PROVEEDORES DE EQUIPOS (TVJ_PROVEEQUI)

Se tiene información de los proveedores de mantenimiento para los equipos.

Toda la información importante que sirve para contactar al proveedor es almacenada en esta tabla, como es: el nombre, dirección, teléfonos, etc.

A cada proveedor se asocia un código que lo identificará en las demás tablas asociadas a él.

TABLA FALLAS (TVJ_FALLAS)

Cada equipo puede presentar fallas en su funcionamiento, estas fallas son resueltas y deben ser almacenadas en la base de datos como soluciones para futuros inconvenientes en los que se presente el mismo escenario.

El mismo tipo de fallas puede presentarse en varios equipos aunque no sean del mismo tipo.

Cada falla identificada se almacena con una breve descripción y su respectiva solución, la cual se asocia a un código identificador.

TABLA EQUIPOS FALLA (TVJ_EQUIFALLA)

Se almacena por cada equipo el tipo de falla que se solucionó, obteniendo así un registro de todos los problemas resueltos asociados a ese tipo de equipo que servirían de solución en futuros casos.

Un equipo puede estar asociado con varios tipos de fallas y de igual manera una falla puede asociarse a varios tipos de equipos.

TABLA UBICACIÓN (TVJ_UBICACION)

Dependiendo de la estructura de la empresa ó industria, se crea el árbol de ubicación donde se puede encontrar un equipo.

Una ubicación es ingresada en el sistema tomando en cuenta primero el nombre del lugar físico más extenso y posteriormente las áreas puntuales, como por ejemplo: Piso1/ Depto Cobranzas/ Oficina 501.

De esta manera se almacena la descripción de la ubicación y el path donde se encuentra, que sería la dirección de localización.

Cada equipo tiene asociada una dirección de la ubicación; lo que se obtiene con esto es que al momento de realizar el mantenimiento un equipo sea fácilmente localizado dentro de la empresa.

TABLA TIPO MAQUINARIA (TVJ_TIPOEQUI)

Existen varios tipos de maquinaria ó equipos, por lo que algunos se pueden considerar como primarios y otros como secundarios.

Para reconocer a que tipo pertenece cada equipo se almacena la información de los diversos tipos de equipos que se pueden encontrar en una empresa o industria, clasificando así de mejor manera la maquinaria existente.

TABLA INSUMO (TVJ_INSUMO)

Se almacena la información de todos los insumos que se tiene en la empresa o industria según cada equipo.

Se lleva un control sobre el stock de cada insumo ingresado en la base de datos, con lo cual se puede vigilar para que nunca falte dichos elementos.

Cada insumo tiene un identificador del tipo al que pertenece y lleva asociado un campo que indica su unidad de medida.

Además se tiene la información acerca de los proveedores según el tipo de insumo al que pertenece

TABLA PROVEEDORES INSUMOS (TVJ_PROVEEINSU)

Toda la información necesaria y fundamental para contactar a los diferentes tipos de proveedores que se tiene para cada insumo se almacena en esta tabla.

TABLA EQUIPOS INSUMO (TVJ_EQUINSU)

Se registra la información de los insumos que posee cada equipo.

Cada equipo tiene asociado uno o varios tipos de insumos y de igual manera cada insumo puede relacionarse a uno o varios tipos de equipos.

TABLA TIPO INSUMO (TVJ_TIPOINSUMO)

Se almacena información de los tipos de insumos que se tiene para cada equipo, esta información es importante ya que se va a relacionar con cada insumo.

Es la clasificación de los diferentes tipos de insumos a los cuales pueden pertenecer los insumos.

TABLA USUARIOS (TVJ_USUARIO)

Para la seguridad y protección de la información ingresada y que se maneja, se crea un identificador con contraseña para cada persona que acceda al sistema, de esta manera se puede almacenar información importante asociada a cada usuario como son: código, nombres, cédula, cargo y fecha de ingreso a la empresa.

Se registra además un campo fundamental y que servirá como control para conocer que usuarios se encuentran activos y cuales inactivos, es decir se debe controlar que un usuario se encuentre en estado activo para que pueda acceder, caso contrario si deja la empresa entonces se restringirá el acceso con ese usuario.

TABLA ESTADO DE LA SOLICITUD (TVJ_ESTADOSOLICITUD)

Cada proforma de los artículos que se planifica comprar es almacenada en esta tabla, en la cual se indican todos los aspectos necesarios y fundamentales para que esta compra sea aprobada o no

Cada compra está ligada a un identificador del estado de la solicitud, el cual permite conocer si la solicitud fue o no aprobada o en que proceso se encuentra.

Adicional también está asociado a cada solicitud el nombre de usuario que ha ingresado un requerimiento, quien será la persona que verifique en que proceso se encuentra su solicitud.

TABLA DESCRIPCIÓN ESTADO (TVJ_DESCRIPCIÓN_ESTADO)

Se indica la descripción de los diferentes estados en los cuales puede estar una solicitud, aquí se asocia para cada estado un código y una breve descripción del significado del estado.

TABLA EMPLEADO (TVJ_EMPLEADO)

Se registra toda la información personal de los empleados que trabajan en el departamento de mantenimiento.

Así como también el cargo que ocupa cada empleado en la empresa o industria; además se almacena información sobre el estado de un empleado en la empresa es decir si se encuentra activo o inactivo.

TABLA USUARIO PERFIL (TVJ_USUAPER)

Contiene la descripción de los diferentes perfiles que puede tener un usuario, con lo cual se controla el nivel de permisos de un usuario.

Existe un campo en el cual se ingresa el código del usuario al que se le va a asignar un perfil.

Cada perfil tiene fecha de caducidad e indican si se encuentra activo o no dicho perfil, con lo cual se puede validar la autenticación de un usuario y controlar los accesos al sistema otorgando según las necesidades de cada empleado un diferente nivel de acceso, controlando de esta manera la seguridad en el sistema.

TABLA ORDEN DE INSUMO (TVJ_ORDENINSUMO)

En esta tabla va a almacenarse los insumos que se utilizarán en una orden de trabajo, se asocia el código de la orden de trabajo, el insumo y la cantidad que se utilizará.

Esta información se encuentra a parte de la tabla de órdenes de trabajo debido a que una misma orden de trabajo puede tener varios artículos de insumos a utilizar.

La cantidad de insumo ingresada en la orden se restará del stock de la tabla de insumos, con esto se puede controlar el stock y verificar lo que ya fue utilizado.

TABLA ORDEN DE TRABAJO (TVJ_ORDENTRABAJO)

Contiene la información de las órdenes de trabajo generadas, aquí se asigna un código a cada orden de trabajo, una breve descripción de lo que conlleva la orden, la fecha cuando va a comenzar a ejecutarse la orden, la fecha máxima en la cual debe estar finalizada la orden y el estado de la orden mediante la cual se indicará todos los procesos por los cuales pasará la orden.

TABLA SOLICITUD DE INSUMO (TVJ_SOL_INSUMO)

Se utiliza para almacenar la información de los pedidos que se realizan para la compra de insumos, una vez que el stock de un insumo esté por agotarse, se ingresará una solicitud de compra de insumos.

En esta tabla se asigna un código a la solicitud de compra, el código del insumo que se va a comprar y la cantidad de insumo.

TABLA ORDEN DE LA RUTINA (TVJ_ORDEN_RUTINA)

Se almacena la información acerca de la relación que existe entre una rutina y una orden generada.

El código de la rutina debe necesariamente asociarse al código de una orden de trabajo.

TABLA EQUIPO FALLA (TVJ_EQUIFALLA)

Esta tabla contiene la información de la relación entre un equipo y una falla encontrada.

Se asocian mediante el código de la falla y el código del equipo.

MANUAL DE INSTALACIÓN

INTRODUCCION

Este manual tiene como finalidad mostrar la correcta utilización y el funcionamiento del software MANPRECOR, para aprovechar toda su funcionalidad.

MANPRECOR fue creado para contribuir en el desarrollo del mantenimiento de cualquier tipo de maquinaria, registrando las rutinas de mantenimiento, así como las fechas en las que se deben realizar, como también un módulo para el consumo y registro de insumos.

Es de mucha importancia seguir el presente manual ya que con el fin de facilitar su uso y entendimiento se han incluido gráficos explicativos en cada uno de los temas tratados.

OBJETIVO

El objetivo primordial del presente manual es guiar al usuario que utilice el sistema MANPRECOR tanto en su instalación como en su uso, en el alcance que tenemos en las diferentes funcionalidades, así como el acceso al mismo.

CONOCIMIENTOS MINIMOS DEL USUARIO MANPRECOR

Para la correcta utilización del software MANPRECOR el usuario deberá tener los siguientes conocimientos mínimos:

- Conocimientos básicos de Windows.
- Conocimientos básicos de programas utilitarios.

ESPECIFICACIONES TÉCNICAS

HARDWARE

Para el correcto funcionamiento del sistema se deberá tener la siguiente especificación mínima de hardware:

Ciente

- Procesador Pentium 4 de 1.0 Mhz
- Monitor SVGA
- Teclado
- Mouse
- Espacio en disco mínimo de 1 giga libre.

Servidor

- Servidor Windows Pentium 4 de 2.0 o mas

SOFTWARE

Las especificaciones de software son las siguientes:

Ciente

- Windows XP Service pack 2 , Vista o Windows 7
- .NET Framework 2.0
- Microsoft Office 2003 en adelante

Servidor

- Servidor Windows NT ,2003
- .NET Framework 2.0
- Visual Studio 2005 en adelante
- SQL 2005 en adelante

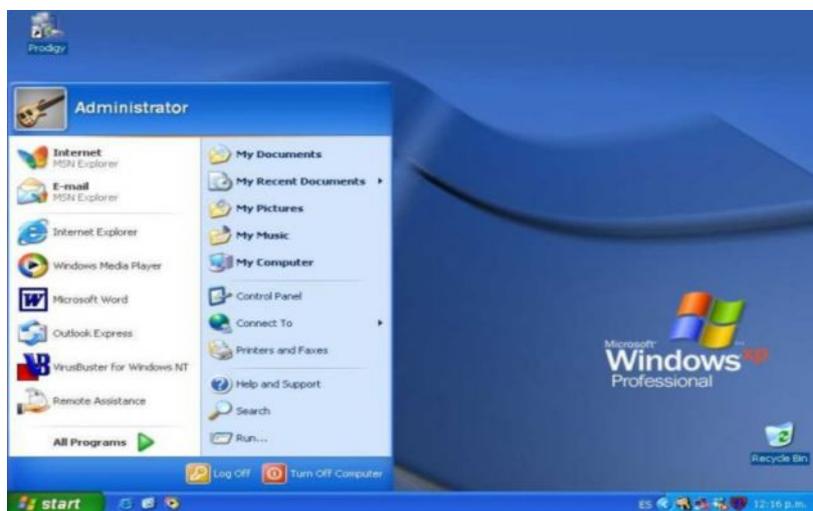
INGRESO AL SISTEMA MANPRECOR

Para el acceso al sistema MANPRECOR se debe realizar los siguientes pasos:
Encender el computador primero desde el botón del CPU continuando con el botón de encendido del monitor



A continuación esperar que el sistema operativo Windows se cargue.

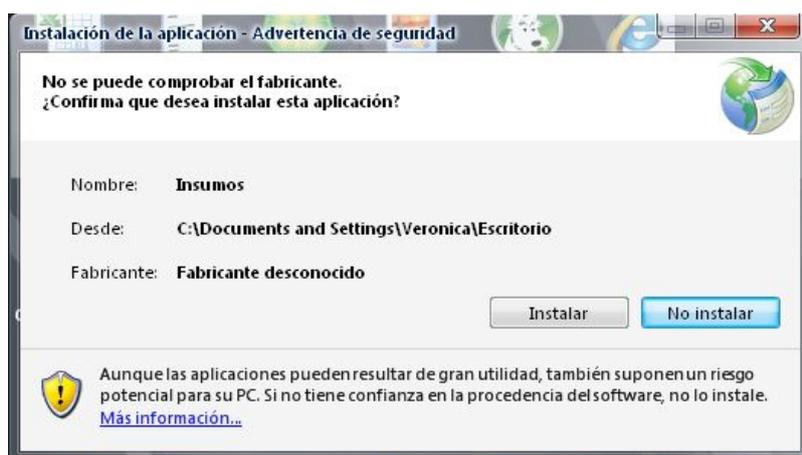
Una vez que se haya encendido correctamente Windows aparecerá el escritorio (esto varía según la versión de Windows que se tenga instalado).



Fig#30. Inicio de Windows

Fuente: Los Autores

Se debe tener especificada por el departamento de sistemas o la persona encargada de la administración del programa cual es el sitio de donde se obtiene la versión instalada en el servidor del sistema MANPRECOR.



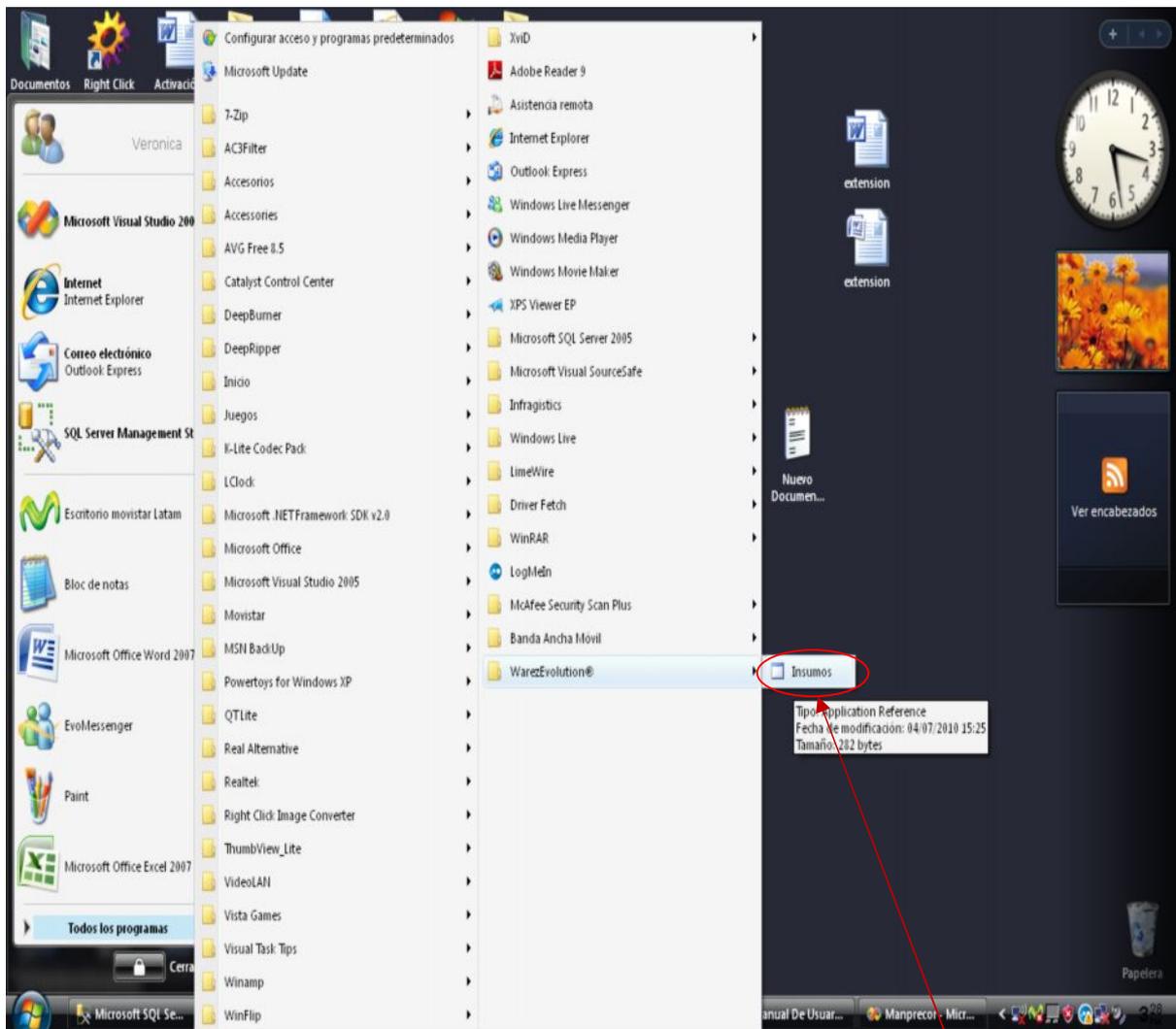
Fig#31. Instalación

Fuente: Los Autores

Aquí se tiene una ubicación en el escritorio local de ejemplo en el cual se puede instalar el sistema MANPRECOR.

Dar Clic en Instalar y esperar que el sistema se instale en la máquina local.

Ahora dirigirse al botón de INICIO de Windows, en Programas buscamos el sistema que se instaló.



Ingreso a la aplicación

Fig#32. Acceso al sistema
Fuente: Los Autores

MANUAL DE USUARIO

Acceso al Sistema

Una vez en la aplicación se desplegará la forma de acceso donde se debe colocar el usuario y la clave para poder acceder al sistema.

Es necesario acotar que el administrador del sistema será el que proporcione los usuarios y claves de accesos.



Fig#33.Pantalla de inicio

Fuente: Los Autores

Es importante saber que el sistema permite solamente tres intentos fallidos de autenticación de usuario o clave, una vez excedido este número de intentos el

usuario si ya fue creado anteriormente se bloqueará y solo podrá ser desbloqueado por el administrador del sistema.

Una vez que sean proporcionados el usuario y clave de acceso se procederá a ingresar al sistema.

Seleccionar Módulo.-



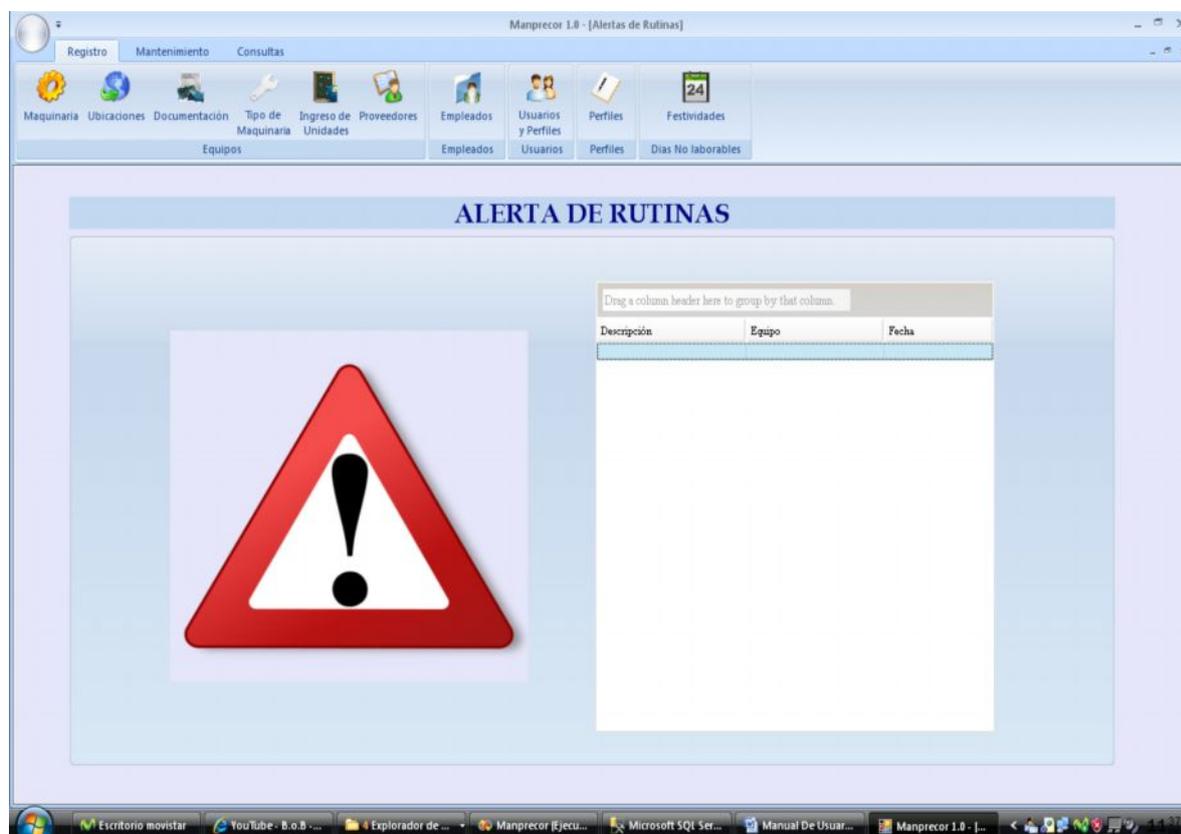
Fig#34. Selección de módulo

Fuente: Los Autores

Esta es la pantalla que se presenta una vez que el usuario ha iniciado su sesión, se puede escoger el módulo en el cual se desea trabajar, ya sea en el módulo de mantenimiento o el módulo de control de insumos.

El usuario que desee ingresar a cualquiera de los dos módulos existentes deberá tener el perfil indicado y activo para acceder al mismo.

Módulo De Mantenimiento.-



Fig#35.Alerta de Rutinas

Fuente: Los Autores

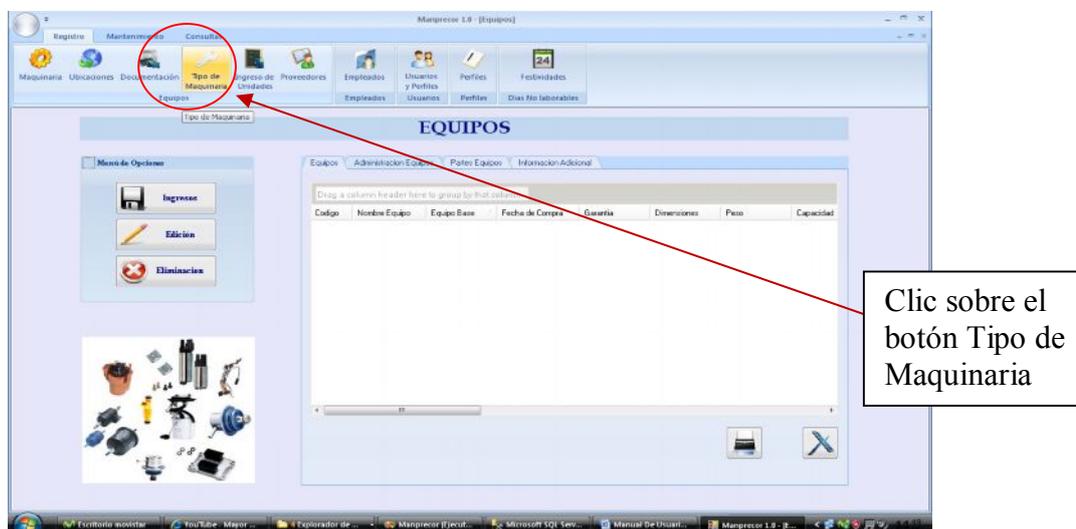
Una vez ingresado al módulo de mantenimiento se tiene la siguiente pantalla donde en la parte superior se muestra un menú de opciones con las cuales se puede interactuar, además en la parte central se muestra alertas en cuanto a las rutinas que se deben realizar, tema que se topará más adelante.

Parametrización Inicial.-

Una vez ingresado al módulo de mantenimiento el primer paso que se debe dar es acceder a la parametrización inicial que consiste en ingresar información como tipos de equipos, ubicaciones físicas, unidades de medida y proveedores de mantenimiento para cada uno de los equipos.

Tipos de equipos.-

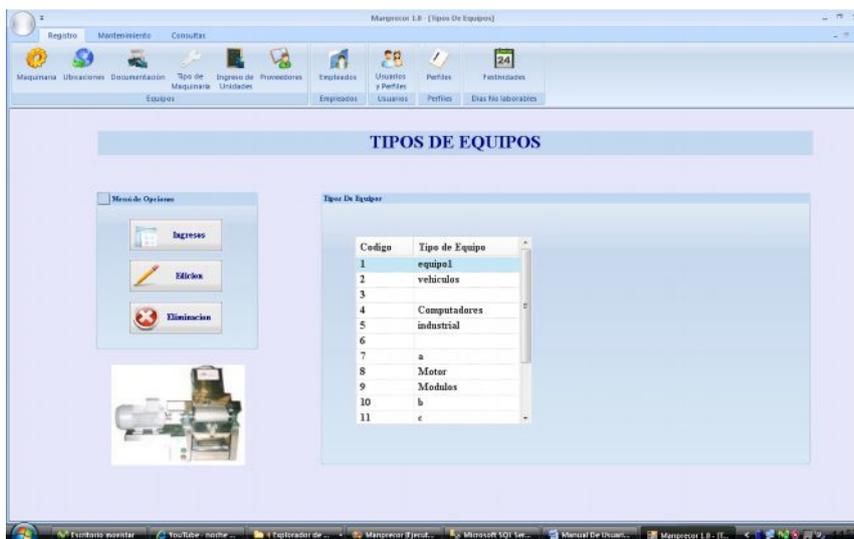
Para acceder a la administración de tipos de equipos dentro del sistema debemos proceder de la siguiente manera:



Fig#36. Acceso Tipos de equipos

Fuente: Los Autores

Una vez en la pantalla de Tipos de equipos tenemos opciones tales como ingresar un tipo nuevo, editarlo o borrarlo, además tenemos una tabla cargada con todos los tipos de equipos ya registrados anteriormente, a continuación veremos cómo administrar la información de tipos de equipos:

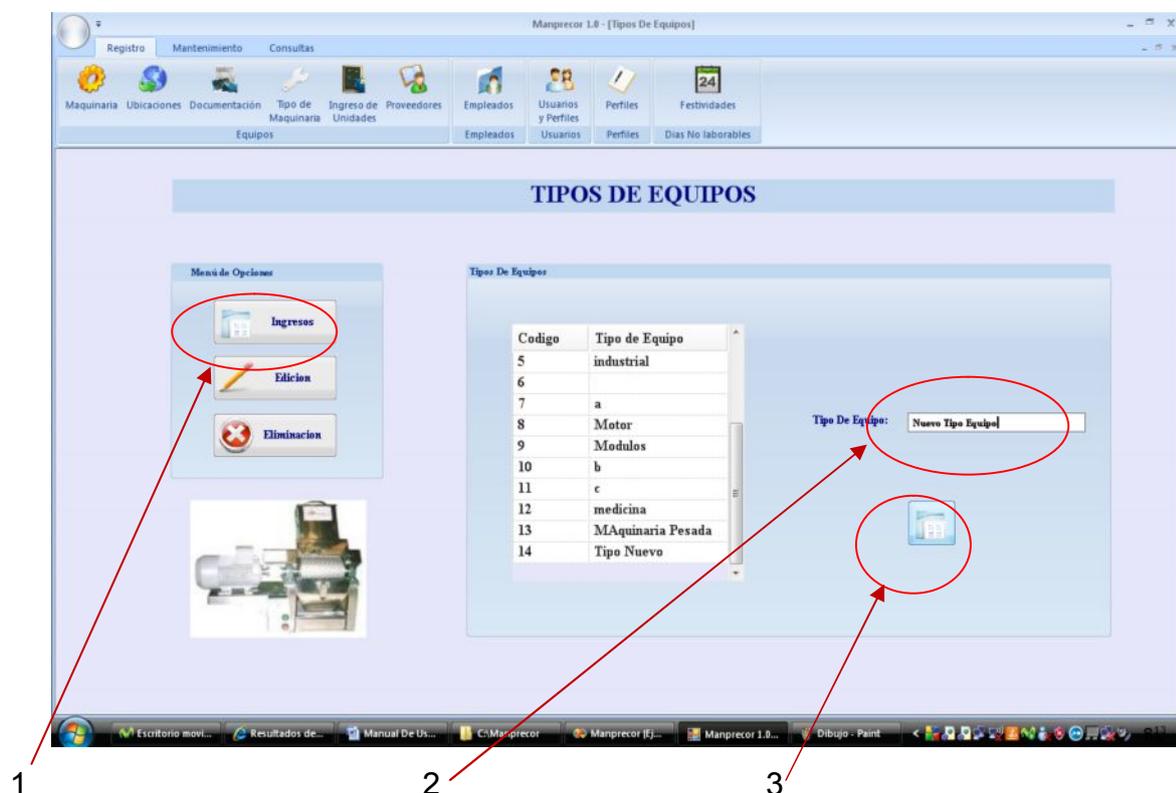


Fig#37. Tipos de equipo

Fuente: Los Autores

Ingreso.-

Con un clic en el botón  (1) se habilita el espacio de texto (2) para ingresar la descripción del tipo de maquinaria que se va a registrar en el sistema, una vez ingresado dar clic en el botón  (3) confirmar si la información es correcta e ingresar, el nuevo tipo ingresado se debe reflejar en la tabla de los tipos de equipos y se tiene un nuevo tipo de equipo almacenado.



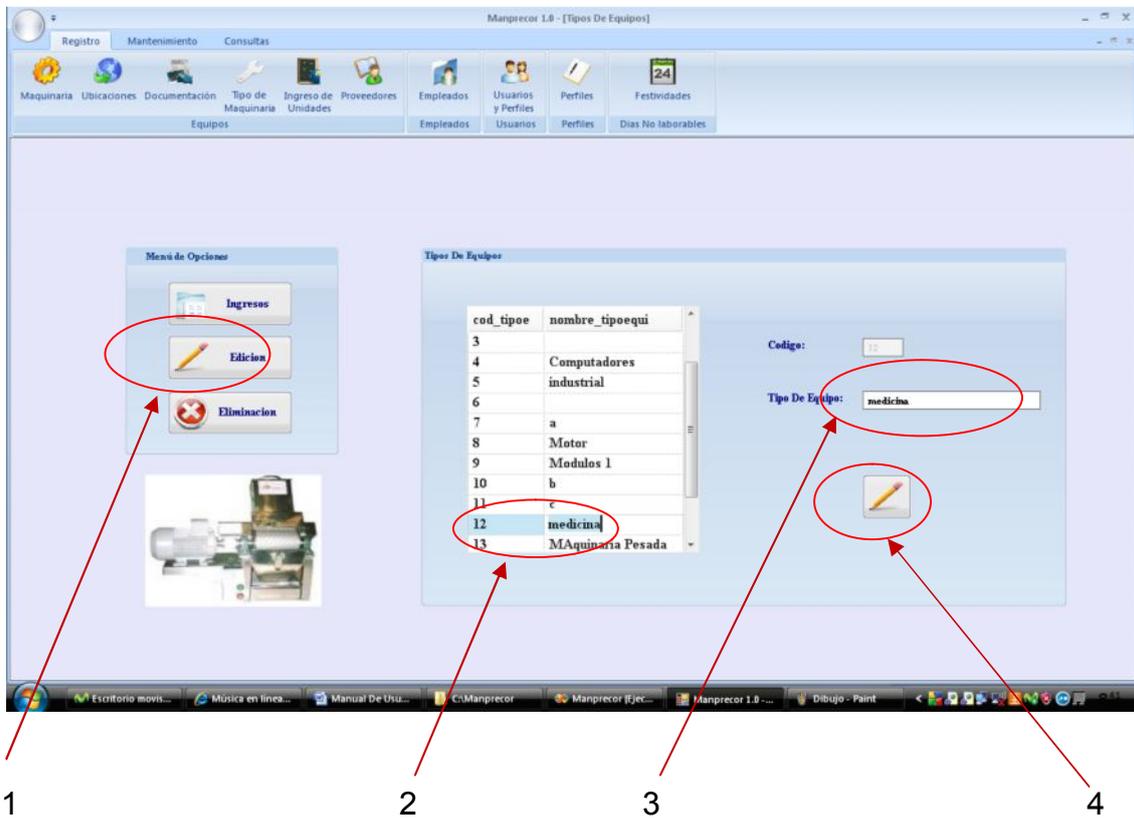
Fig#38.Ingreso Tipos de equipo

Fuente: Los Autores

Edición.-

Para la edición dar un clic en el botón  (1) ubicado en el menú de opciones, en la tabla con doble clic se selecciona el tipo de equipo que se quiere editar(2), esta acción debe cargar en el espacio de texto el nombre del tipo de equipo seleccionado (3), es ahí donde se puede editar el nombre del tipo de equipo

para su posterior utilización, dar clic en el botón  (4) confirmar que se desea editar la información y se puede visualizar en la tabla de tipos de equipos el cambio realizado.

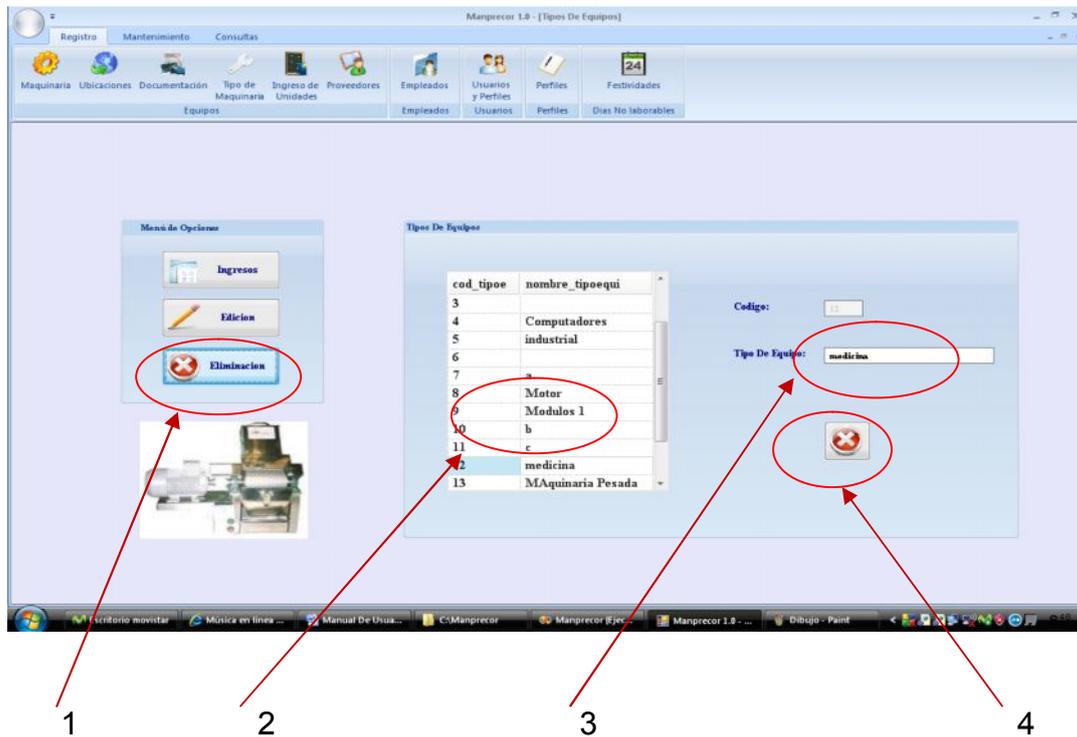


Fig#39.Edición Tipos de equipo

Fuente: Los Autores

Eliminación.-

En la eliminación de tipos de equipos dar clic en el botón  (1) ubicado en el menú de opciones, escoger en la tabla de opciones el tipo de equipo dando doble clic (2) sobre éste e inmediatamente se debe cargar en el espacio de texto (3) con la restricción que no se puede modificar el nombre del tipo que se quiere eliminar, una vez hecho esto dar clic en el botón  (4) eliminando el tipo de equipo seleccionado.



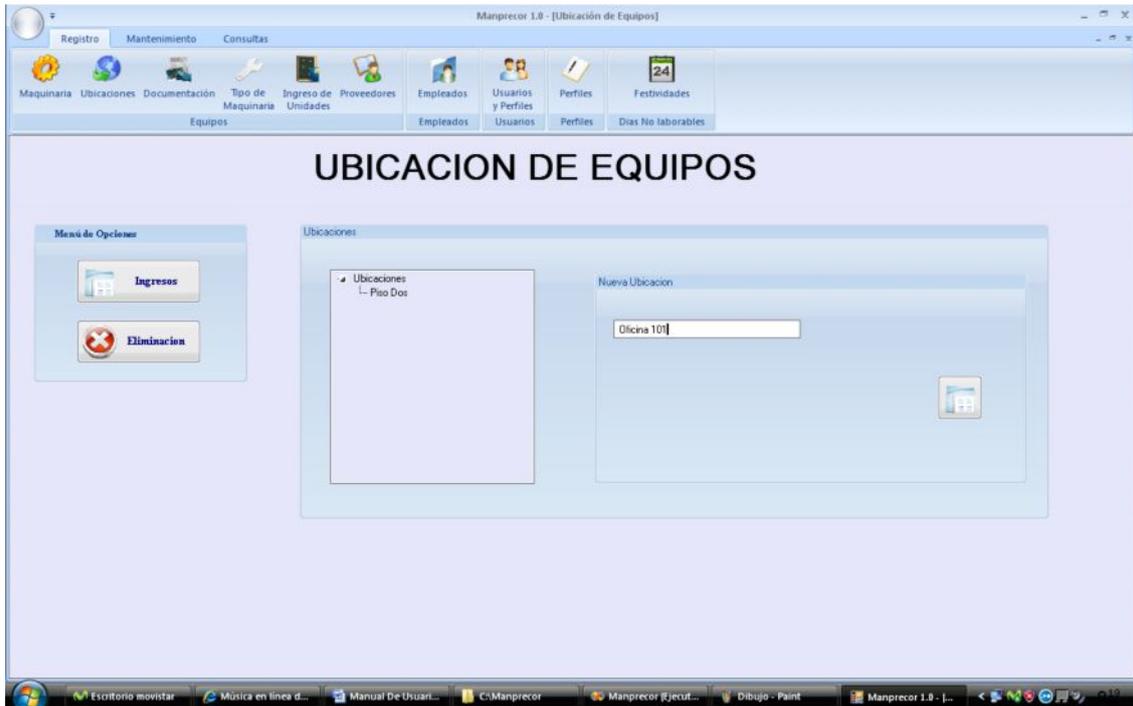
Fig#40.Eliminar Tipos de equipo

Fuente: Los Autores

Ubicaciones Físicas De los Equipos

Una de las parametrizaciones que se debe tomar en cuenta es las ubicaciones físicas del equipo, MANPRECOR proporciona la opción de ingresar las ubicaciones que se tiene en la empresa donde se esté utilizando el sistema, para acceder a esta opción dirigirse a UBICACIONES en el menú principal de mantenimiento.

Dentro de esta funcionalidad se tiene opciones como ingresar y eliminar las ubicaciones, además de visualizar en forma de árbol jerárquico las ubicaciones ya parametrizadas. En esta pantalla no existe la opción de edición de ubicaciones ya que se debe tomar en cuenta no solo el nombre de la ubicación sino también el nivel de la misma, así que es más sencillo eliminar la ubicación no deseada y crearla donde se requiera.



Fig#41.Ubicación de equipos

Fuente: Los Autores

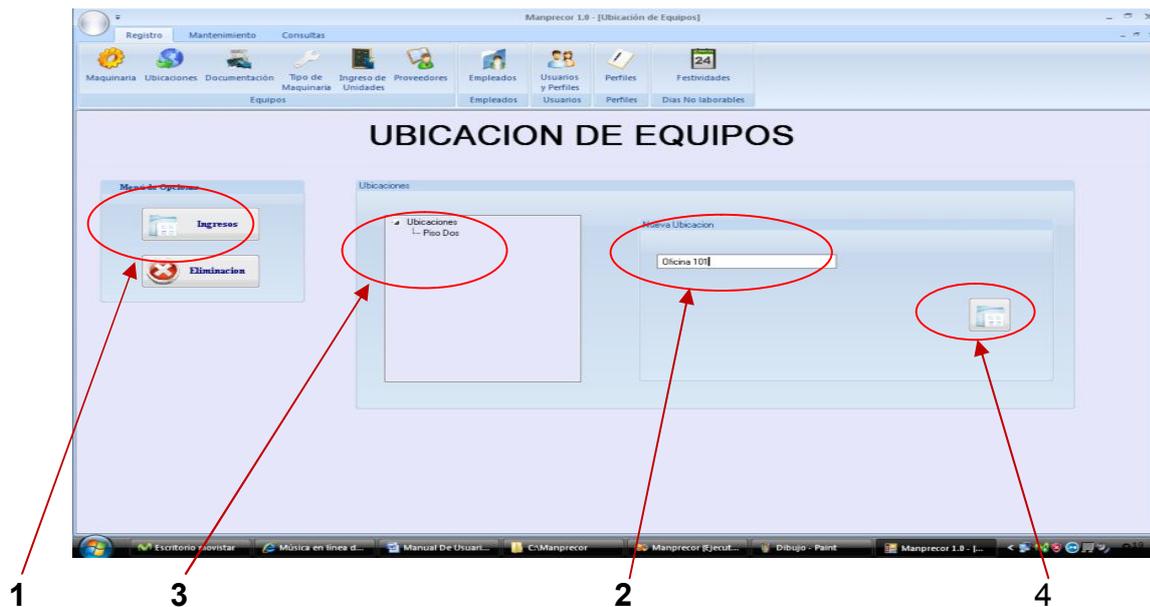
Ingresar Ubicación

Para realizar el ingreso de una nueva ubicación dirigirse al botón de



(1) en el menú de opciones, se visualiza un espacio de texto para ingresar el nombre de la ubicación(2), ahora bien se debe especificar el nivel donde se quiere ingresar la ubicación, por ejemplo: se va a ingresar la ubicación oficina 101 que se encuentra en el piso 1 del edificio de trabajo, primero se debe ingresar el edificio luego con un clic sobre el árbol(3) en la ubicación edificio se especifica que la siguiente ubicación ingresada se encuentra dentro del edificio, y se ingresa la descripción Piso 1, para el registro de la ubicación oficina 101 se procede de la misma manera, el árbol soporta hasta tres niveles de profundidad.

Una vez definido el nombre y el nivel de la ubicación dar clic en el botón  (4) confirmar la información e ingresar la ubicación nueva.

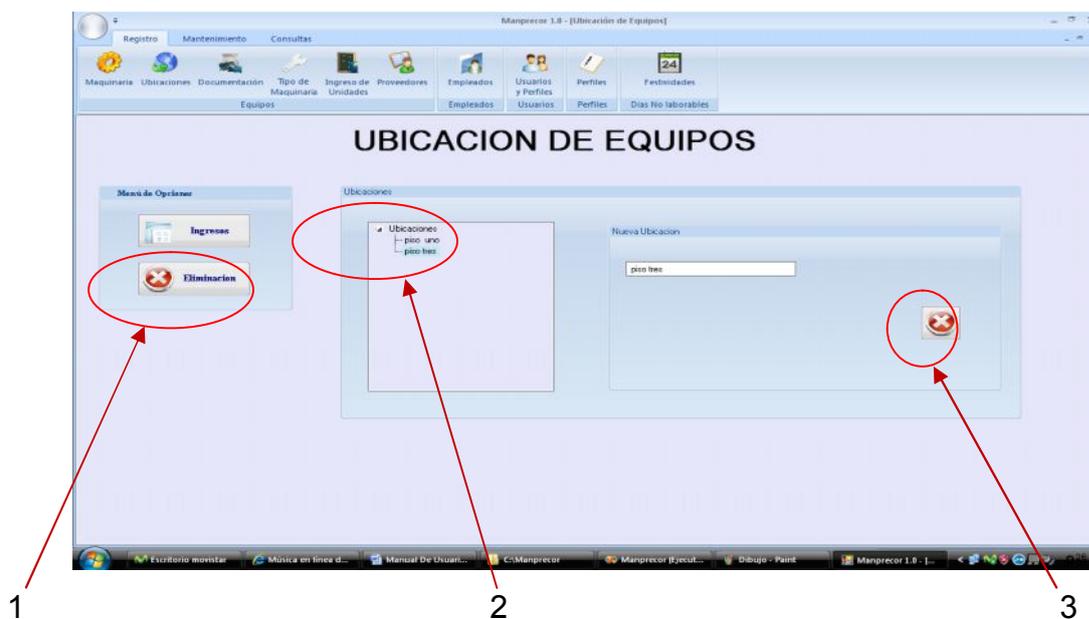


Fig#42.Ingreso de Ubicación de equipos

Fuente: Los Autores

Eliminar Ubicación

Para la eliminación de las ubicaciones se debe seleccionar el botón  (1) después escoger la ubicación que se desea borrar (2), esta se cargará en el espacio de texto sin opción a edición, enseguida seleccionar el botón  (3) eliminando la ubicación deseada.



Fig#43.Eliminación de Ubicación de equipos

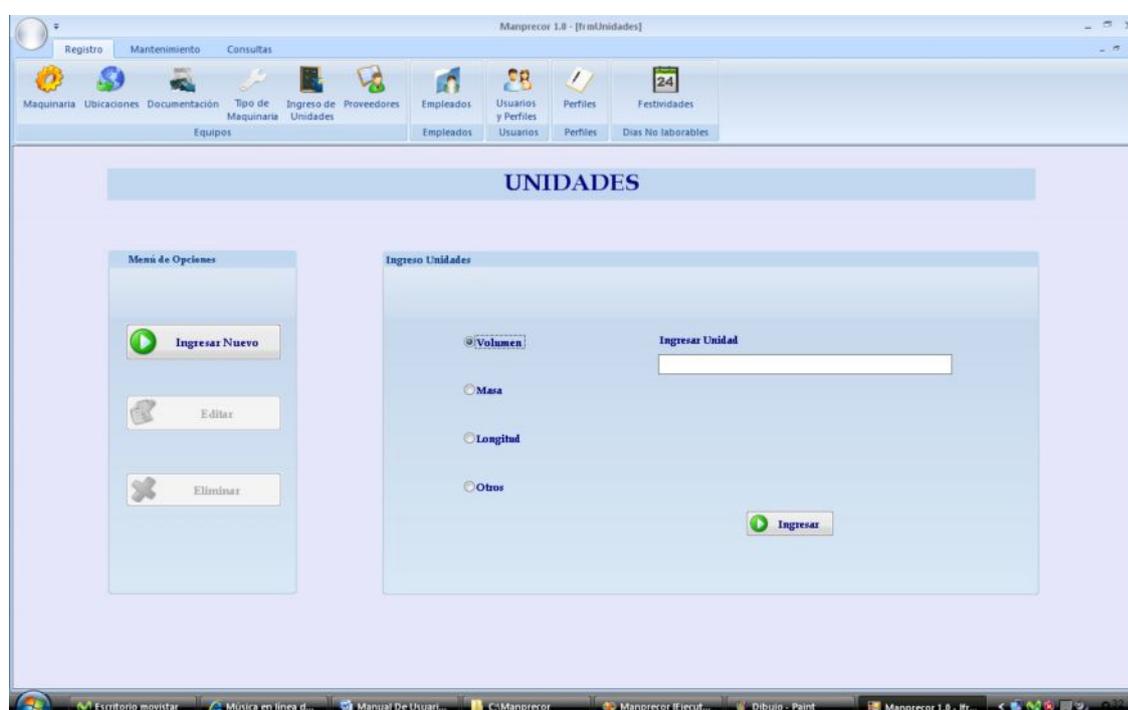
Fuente: Los Autores

Unidades De Medida

MANPRECOR da la posibilidad de ingresar las unidades de medida que se van a utilizar en la definición de los equipos tales como volumen, longitud, masa, etc.

Para acceder a la pantalla de unidades escoger la opción “Ingreso de unidades” del menú principal de MANPRECOR.

En la pantalla de unidades se tiene la opción de ingresar, editar y eliminar unidades de medida para utilizarlas en el registro de la maquinaria, además de visualizar en una tabla de opciones las unidades ya almacenadas.

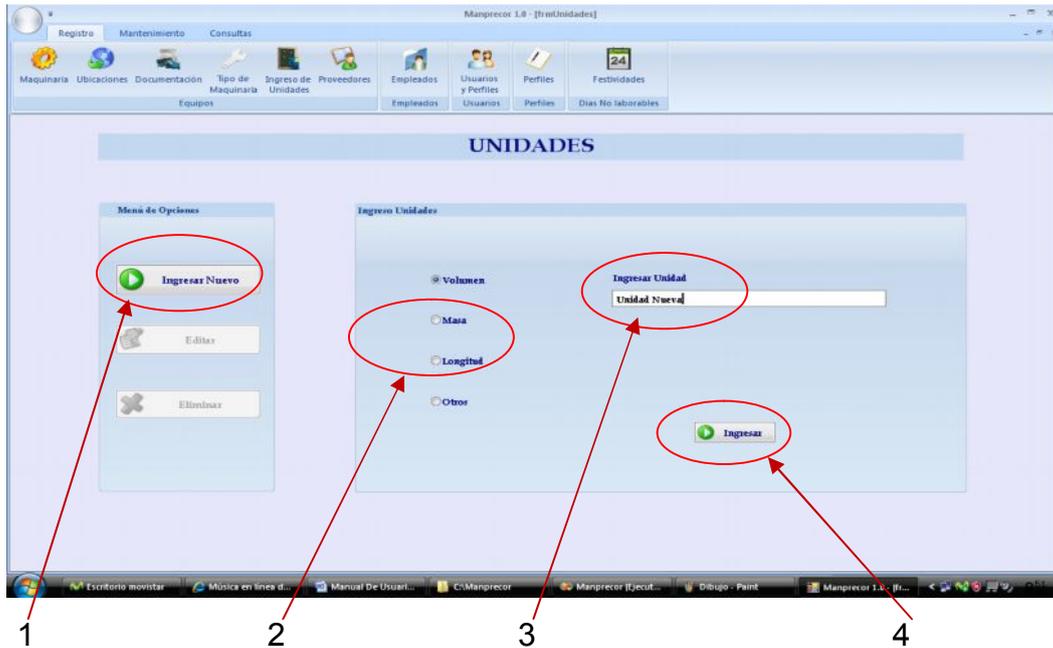


Fig#44.Unidades

Fuente: Los Autores

Ingresos.-

Para ingresar una nueva unidad seleccionar el botón  (1), después especificar el tipo de unidad que se desea almacenar (2), ingresar en el espacio de texto el nombre de la unidad (3), seleccionar el botón  para ingresar la información en el sistema.

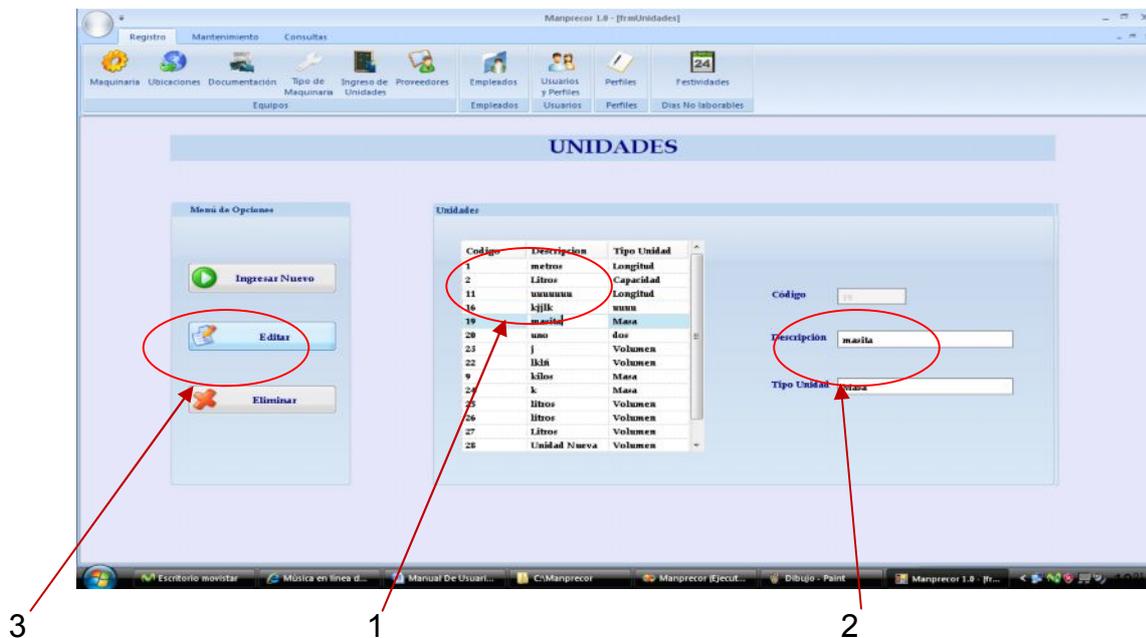


Fig#45.Ingreso de Unidades

Fuente: Los Autores

Edición.-

Para la edición seleccionar con doble clic la unidad que se desea editar en la tabla de opciones (1) se carga en los espacios de texto los datos de la unidad que se puede modificar (2), dar clic en el botón  (3) editando la información.

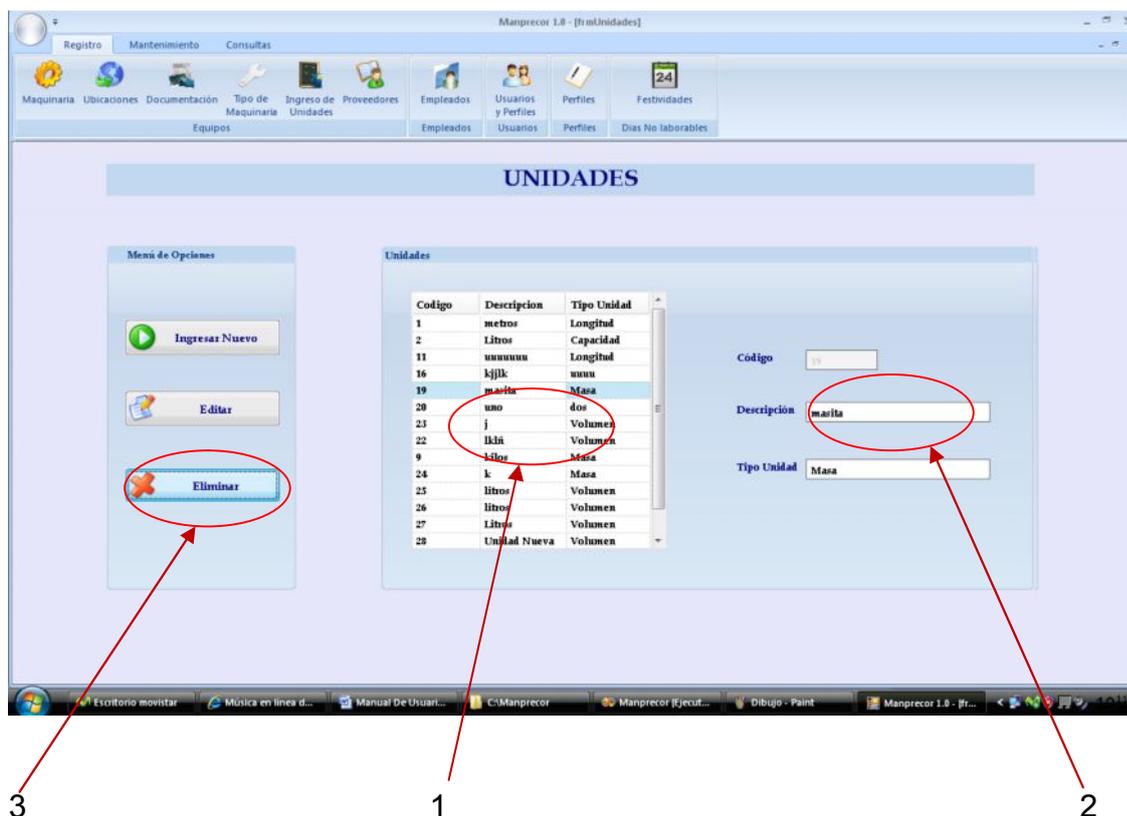


Fig#46.Edición de Unidades

Fuente: Los Autores

Eliminación.-

Para la eliminación seleccionar la unidad que se desea remover de la tabla de opciones (1) se carga la información en el espacio de texto indicado (2) y por último seleccionar el botón  (3) para eliminar del sistema la unidad indicada.



Fig#47.Eliminación de Unidades

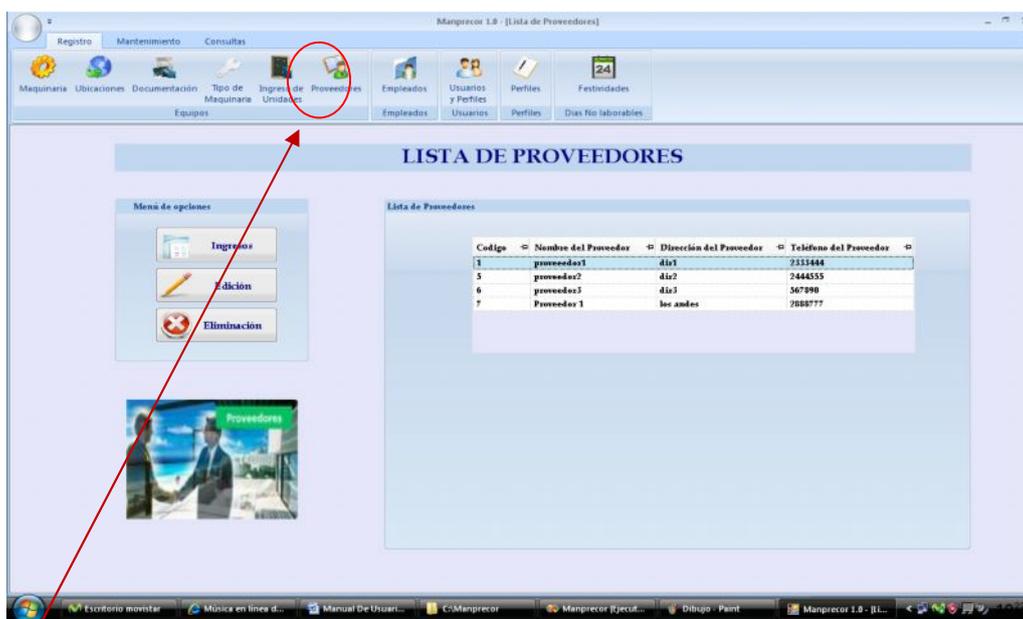
Fuente: Los Autores

Proveedores De Mantenimiento.-

El sistema MANPRECOR permite registrar información sobre las personas que se encuentran relacionadas con el mantenimiento de la maquinaria, aquí se puede ingresar información básica para su localización en caso de ser necesarios sus servicios.

En esta pantalla tenemos la opción de ingreso, edición y eliminación de los proveedores y su información relacionada. Para Acceder a la pantalla dar clic en el

botón  (1) del menú principal.



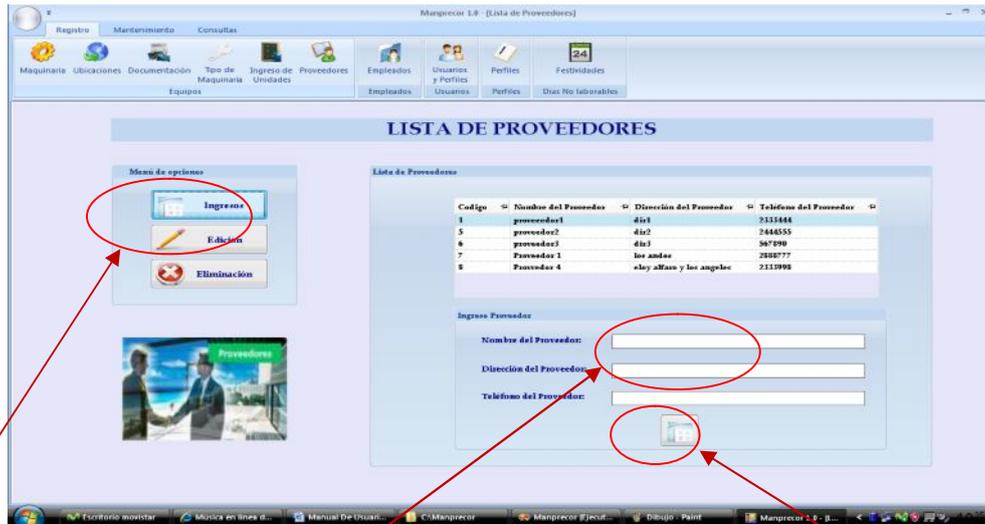
1

Fig#48.Proveedores
Fuente: Los Autores

Ingresos.-

Para el ingreso de un nuevo proveedor dar clic en el botón  (1), aparecen espacios de texto para ingresar la información como el nombre del proveedor, la dirección y el teléfono de contacto (2) con la información

correctamente digitada dar clic en el botón  (3) e ingresar la información al sistema.



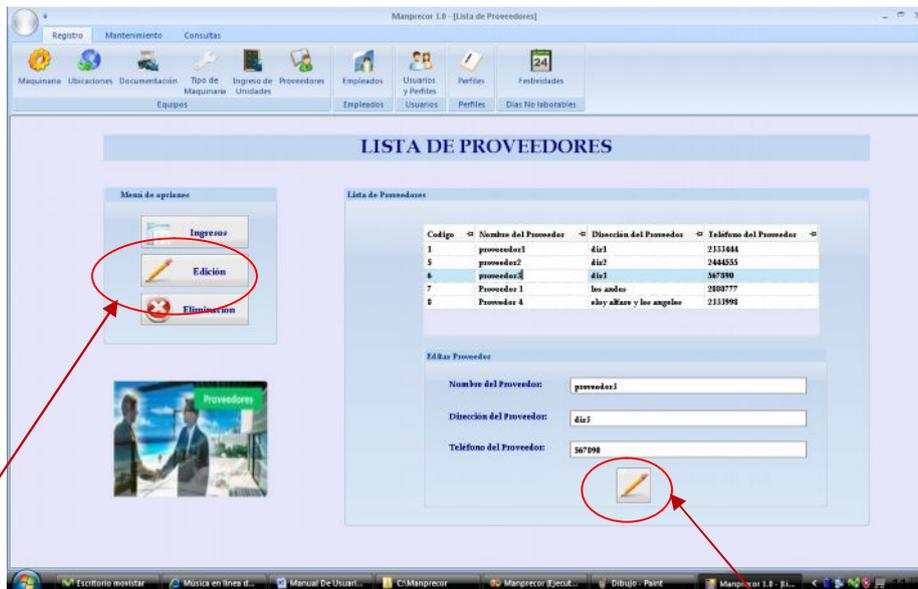
1 2 3

Fig#49.Ingreso de Proveedores

Fuente: Los Autores

Edición.-

Para la edición de información de proveedores dar clic en el botón  (1). Con doble clic en la tabla escoger el proveedor que se desea editar y la información se cargará en los espacios de texto, finalmente dar clic en el botón  (2) editando la información en el sistema.



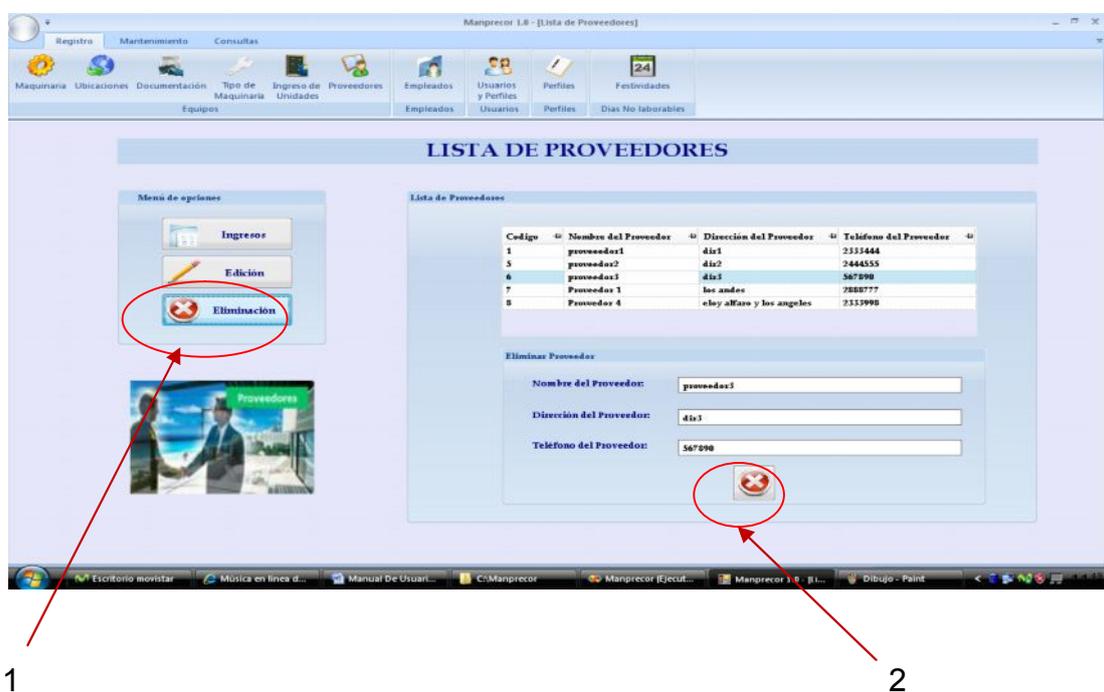
1 2

Fig#50.Edición de Proveedores

Fuente: Los Autores

Eliminación.-

Para la eliminación de información de proveedores dar clic en el botón  (1). Una vez hecho esto escoger en la tabla de opciones el proveedor que se desea eliminar del sistema, la información se carga en los espacios de texto, si tenemos certeza de eliminar se procede a dar un clic sobre el botón  (2) eliminando la información del sistema.



Fig#51.Eliminación de Proveedores

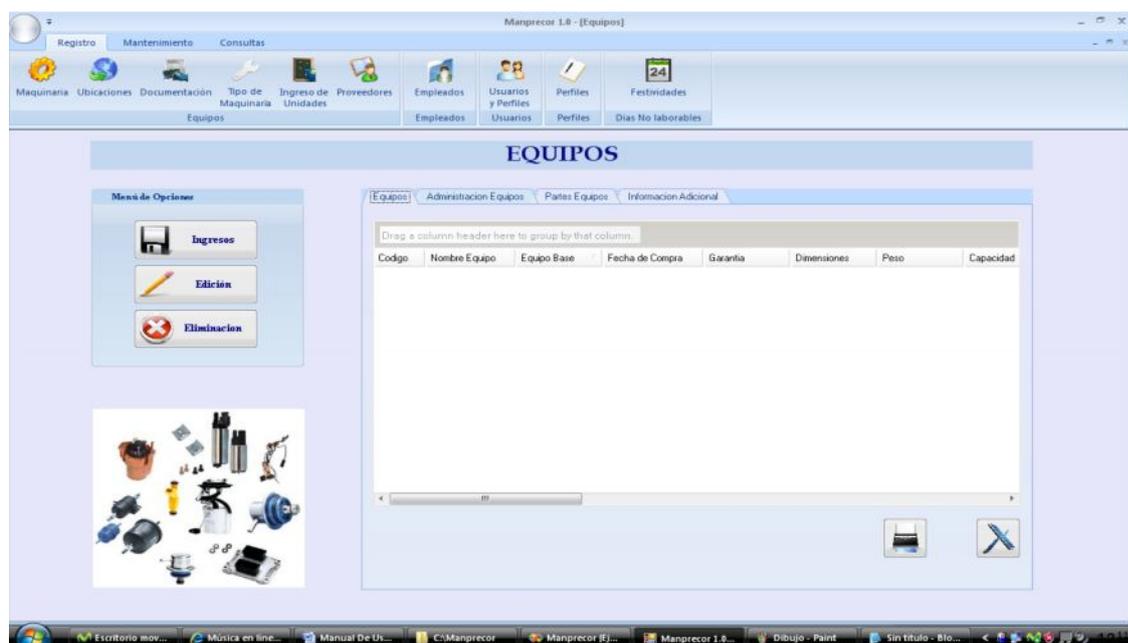
Fuente: Los Autores

Administración de Equipos.-

Esta es una sección muy importante dentro del sistema MANPRECOR ya que es aquí donde se va a registrar la información de los equipos, sus partes asociadas y todo lo necesario para que el proceso de mantenimiento se realice de una forma satisfactoria.



Para acceder a esta funcionalidad ubicarse sobre el botón **Maquinaria** en el menú principal del sistema e ingresar en la pantalla, aparecerá una tabla de opciones con todos los equipos ingresados anteriormente, junto con botones de opciones para el ingreso, edición y la eliminación de maquinaria.



Fig#52.Equipos

Fuente: Los Autores

Ingresos.-

Para el ingreso de un nuevo equipo en el sistema MANPRECOR en primer lugar se debe ubicar en la pestaña **Administración Equipos** (1) en la parte superior, una vez hecho eso dirigirse al botón **Ingresos** (2), aquí se tiene una gran cantidad de espacios de texto para ingresar cada uno con su descripción, como por ejemplo se tiene el nombre del equipo, el tipo al cual pertenece, la ubicación física en la empresa, ésta información como la del proveedor de mantenimiento se despliegan en forma de lista y ahí se reflejará toda la información de la parametrización

ingresada anteriormente, además se tiene otro tipo de información que se detalla a continuación:

Fecha de Compra: Se ingresa la fecha de compra del equipo que se va a registrar.

Tiempo De Garantía: Aquí se ingresa la garantía en años que el fabricante otorga para el equipo.

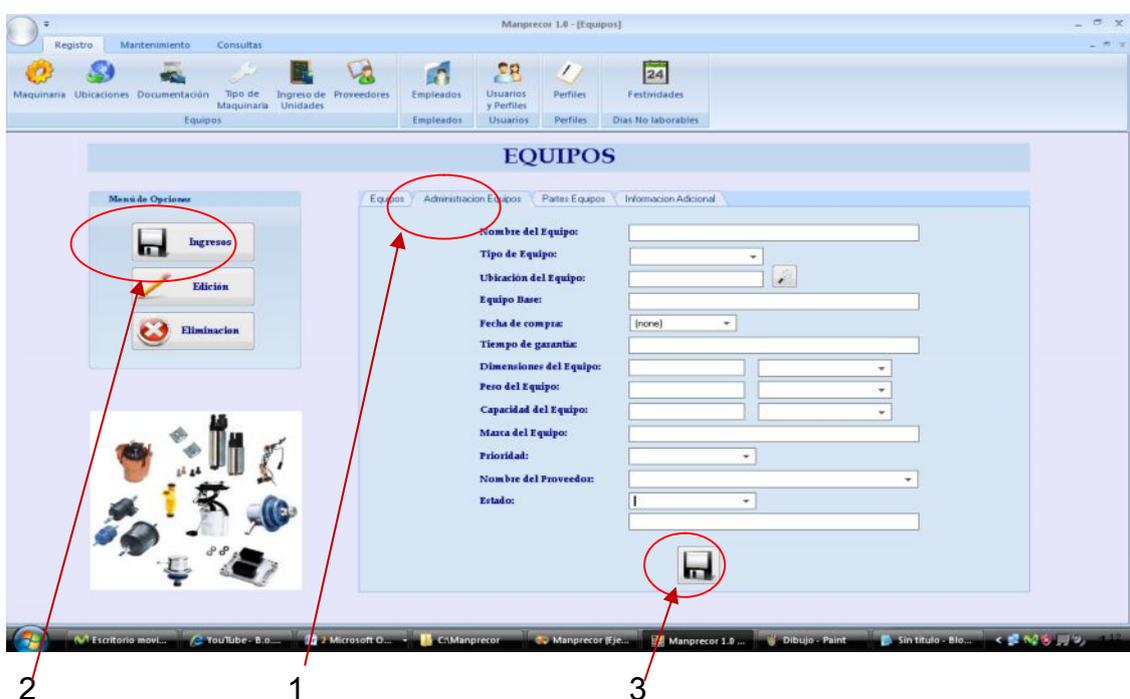
Dimensiones, peso y capacidad del equipo: En estos espacios de texto se ingresa toda la información que tiene que ver con la capacidad, el peso y las dimensiones del equipo, ahora bien esto depende del equipo que se ingresa y si es o no necesaria para el trabajo este tipo de información.

Marca del Equipo: En este campo se ingresa la marca relacionada con el equipo que se va a ingresar.

Estado: Aquí se ingresa el estado en el que se encuentra el equipo, las opciones que tenemos son: activo e inactivo, por defecto y para que el equipo pueda ser tomado en cuenta en el mantenimiento debe tener el estado Activo.

Una vez ingresada la información requerida para registrar el equipo en el sistema

dar clic en el botón  (3) para registrar el equipo en el sistema.

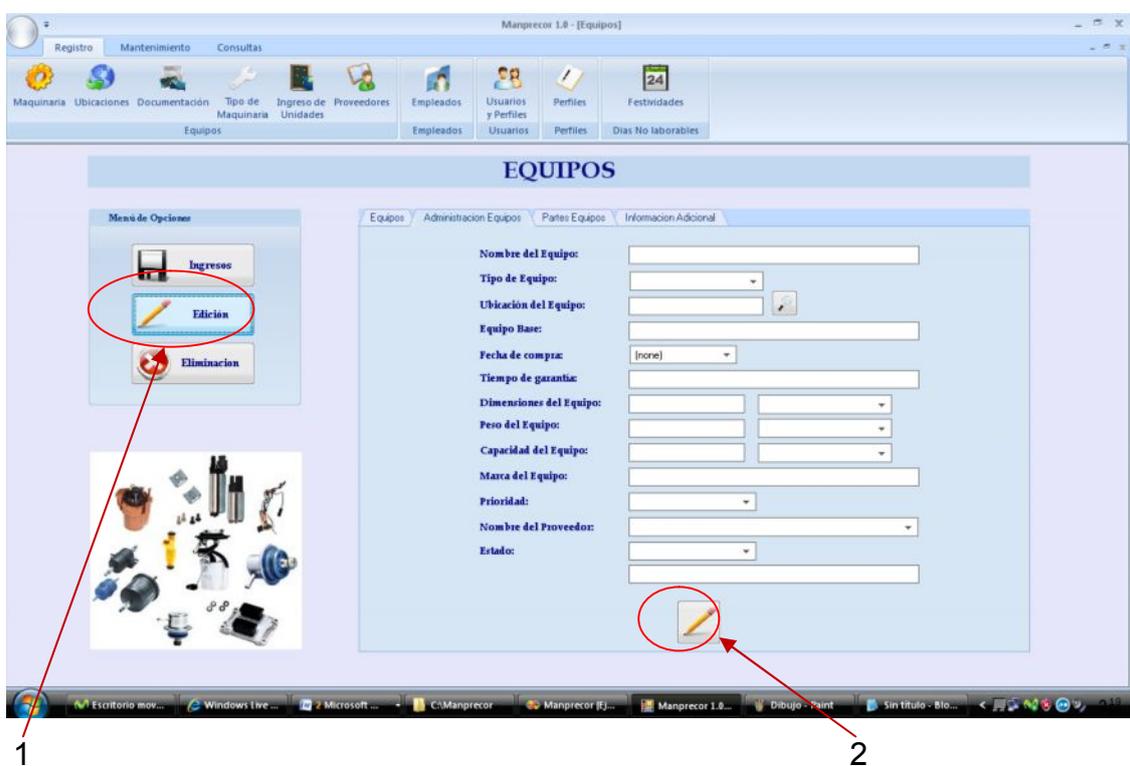


Fig#53.Ingreso de Equipos

Fuente: Los Autores

Edición.-

Para la modificación de los datos de un equipo, ubicarse en la tabla de equipos y seleccionar con doble clic el equipo del que se desea editar la información, luego se carga la información en los espacios de texto, dar clic en el botón  (1), editar la información deseada en los espacios de texto y presionar el botón  (2) guardando la información modificada.



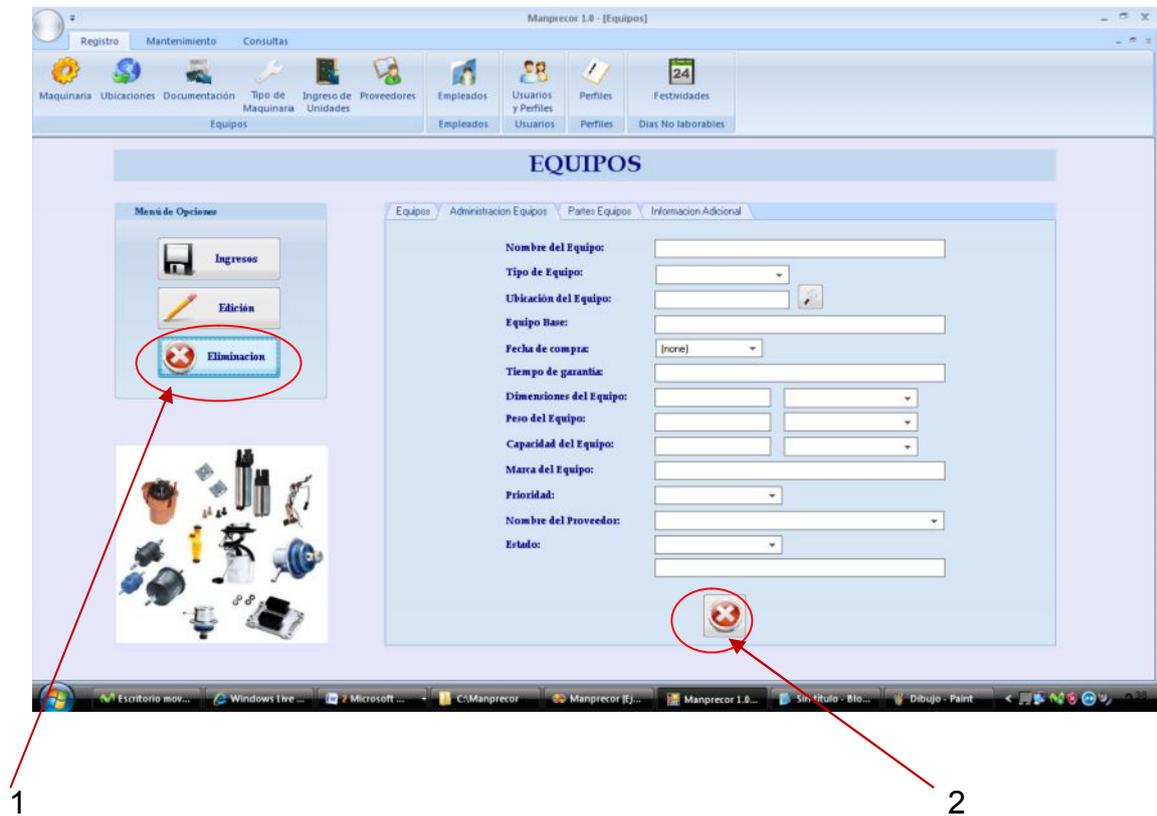
Fig#54.Edición de Equipos

Fuente: Los Autores

Eliminación.-

Para la eliminación de equipos se procede de la siguiente manera: dar doble clic sobre el equipo que se quiere eliminar en la tabla de equipos, la información del equipo se carga en los espacios de texto, luego presionar el botón  (1)

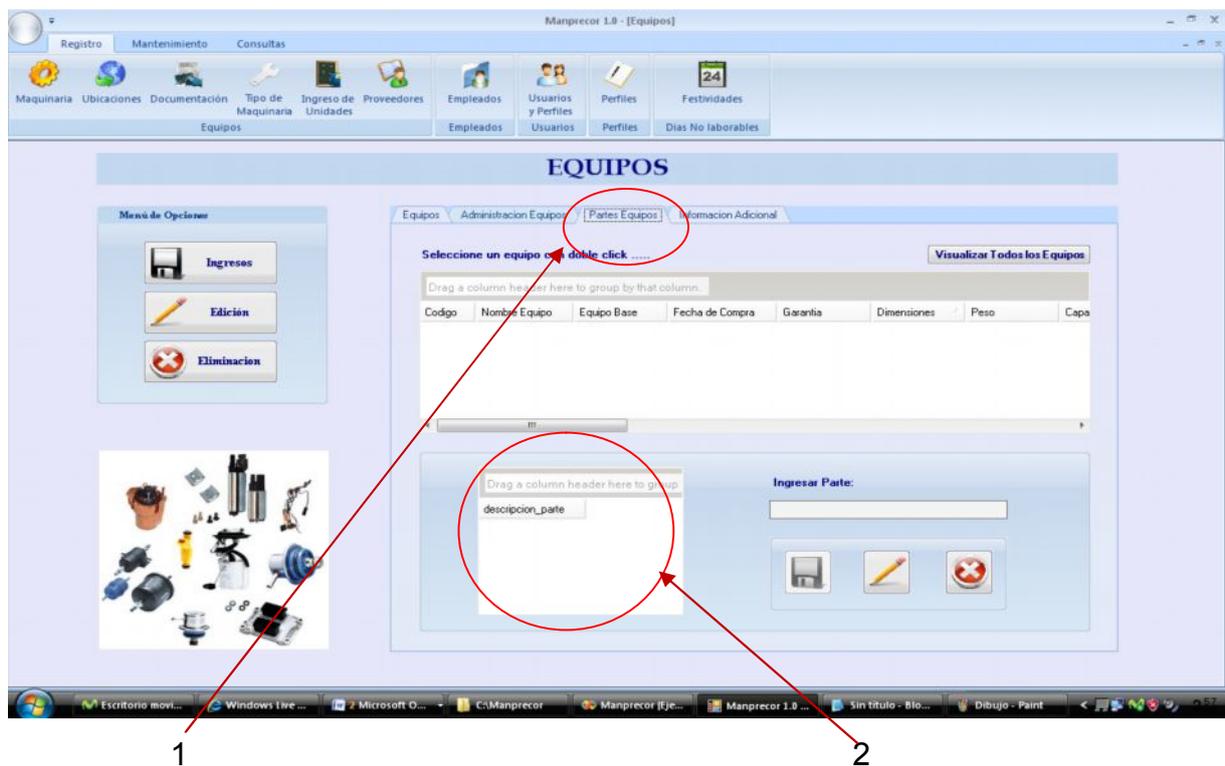
una vez que se decide eliminar la información del equipo dar clic en el botón (2) para que la información se elimine del registro de equipos.



Fig#55.Eliminación de Equipos
Fuente: Los Autores

Partes De Equipos.-

Las partes de los equipos se definen en la pantalla de equipos en la pestaña **Partes Equipos** (1), aquí se presenta una tabla con los registros almacenados, con doble clic en el equipo se mostrará en la tabla inferior (2) la información de las partes registradas, en las opciones de partes se tiene el ingreso, la edición y la eliminación de partes.



Fig#56.Partes de Equipos

Fuente: Los Autores

Ingresos.-

Para el ingreso de partes en primera instancia se debe escoger en la tabla superior el equipo al cual se va a registrar una parte en el sistema con doble clic como indica el mensaje superior, una vez realizada esta acción se cargará información en la tabla de partes si ya existen partes asociadas anteriormente, para el ingreso de una

nueva parte dar clic sobre el botón  (1) una vez hecho esto ingresar el nombre de la parte que se quiere registrar y presionar el botón  (2) guardando la información.

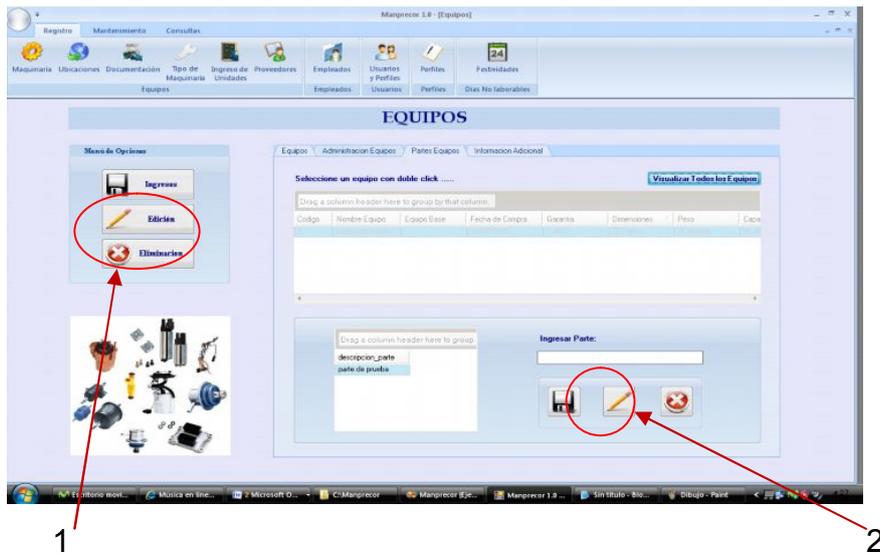


Fig#57.Ingreso Partes de Equipos

Fuente: Los Autores

Edición.-

Para la edición de partes de equipos dar doble clic sobre la tabla de opciones en el nombre de la parte que se requiere cambiar, presionar sobre el botón  (1), después aparece en el espacio de texto dando la posibilidad de editar el nombre de la parte, una vez hecho esto dar clic en el botón  (2) confirmar el cambio y se registra la información editada.

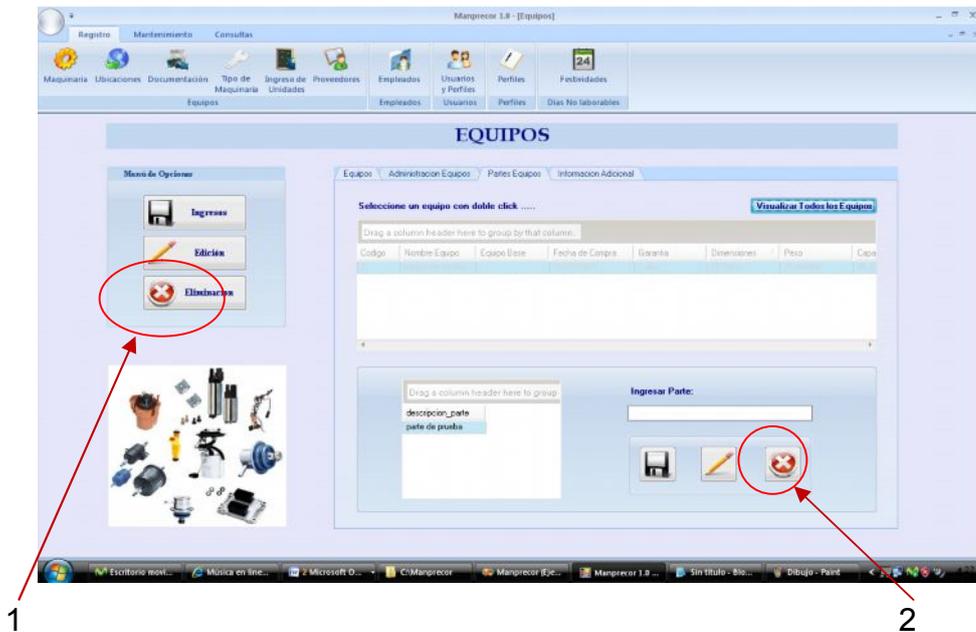


Fig#58.Edición Partes de Equipos

Fuente: Los Autores

Eliminación.-

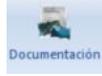
Para la eliminación de las partes de los equipos escoger la parte que se quiere eliminar en la tabla, acto seguido presiona el botón  (1), después con un clic en el botón  (2) se eliminará la información deseada.

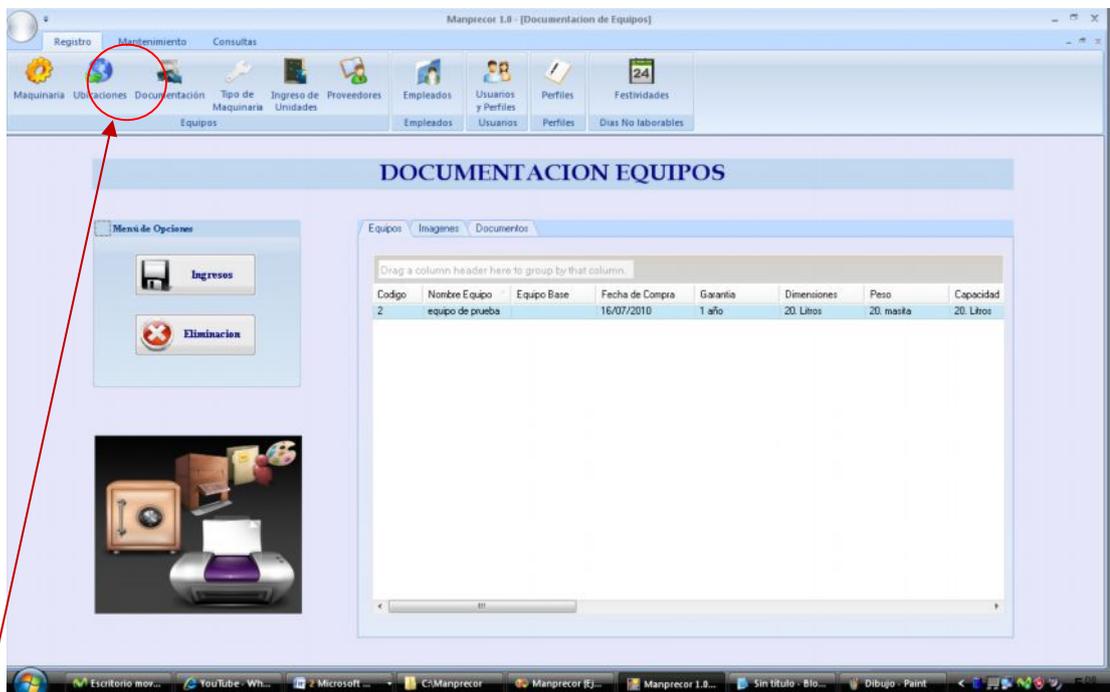


Fig#59.Eliminación Partes de Equipos

Fuente: Los Autores

Documentación De Equipos.-

Además de registrar la información básica del equipo el sistema MANPRECOR permite ingresar información adicional del equipo tal como documentación que este relacionada con el equipo como garantías, planos y cualquier documento importante, también permite ingresar imágenes de los equipos para tener una información integral acerca del equipo, en esta funcionalidad como en las demás existe la posibilidad de ingreso y eliminación tanto de imágenes como de documentos. Para acceder a esta funcionalidad dar clic en el botón  (1) que se encuentra en el menú principal MANPRECOR.



1

Fig#60.Documentación
Fuente: Los Autores

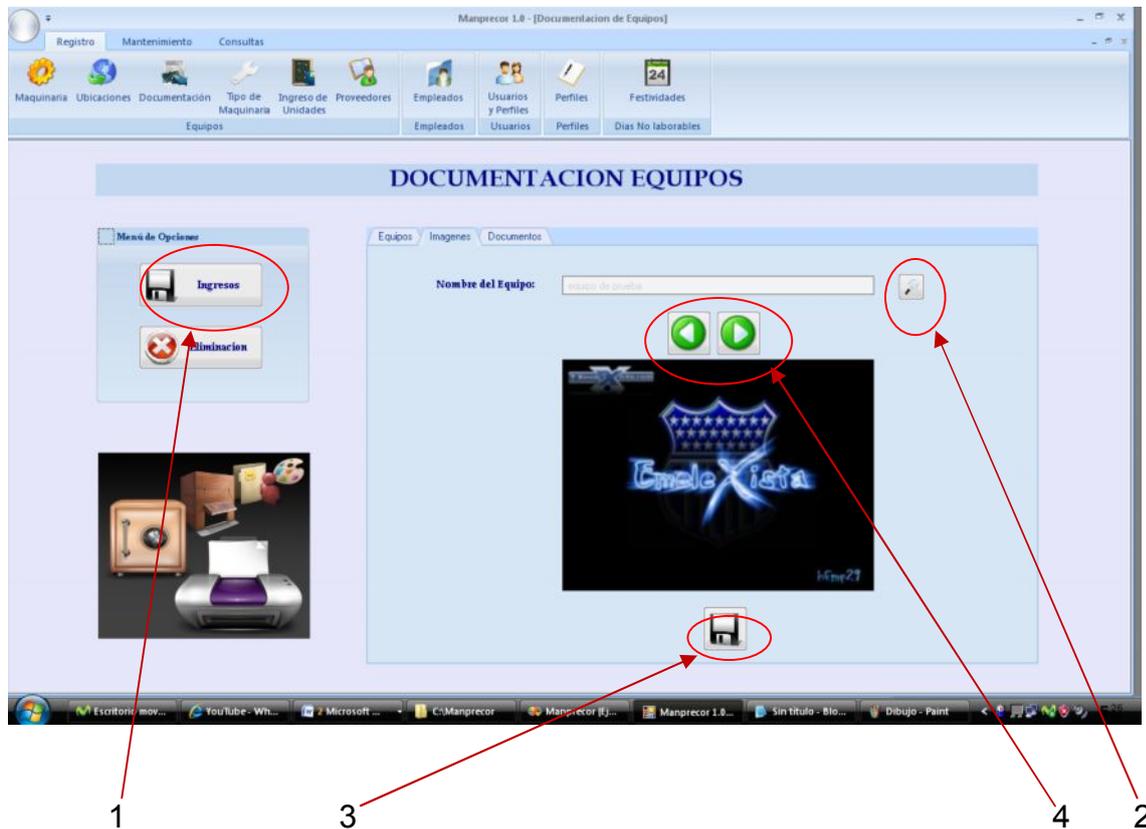
Ingresos.-

Para el ingreso de documentación escoger con doble clic en la tabla el equipo al cual se quiere asociar la documentación tanto de imágenes como de archivos, en las pestañas **Imágenes** **Documentos** se puede acceder para ingresar la información.

Imágenes

Para el ingreso de imágenes presionar el botón  (1) acto seguido en el botón  (2) se puede buscar el archivo tipo imagen en el disco local, una vez identificado se debe abrir en el visor de imágenes, para guardarla dar clic en el botón

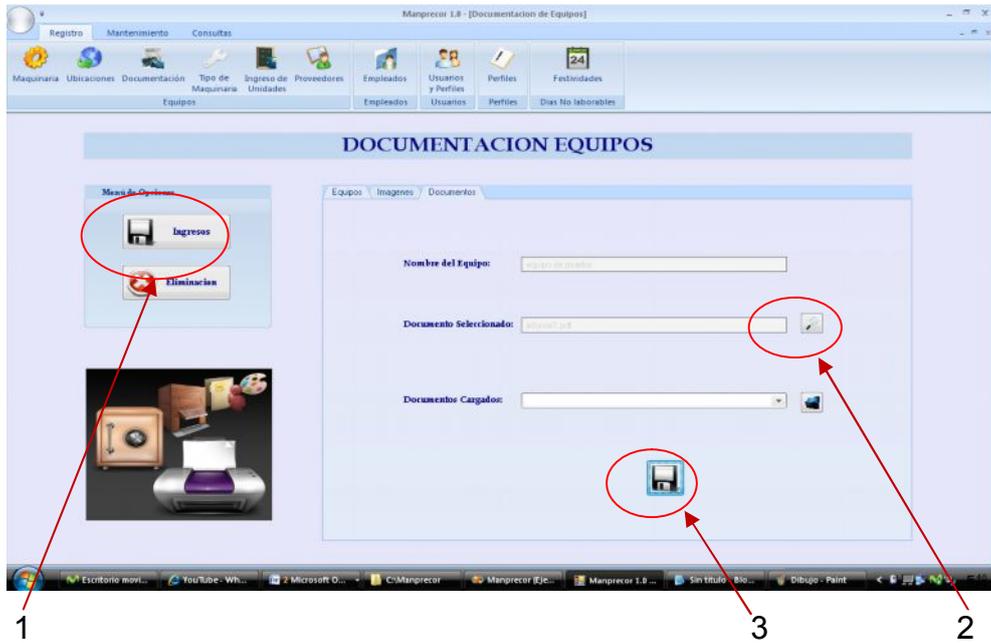
 (3) y la imagen se guardará; para visualizar las imágenes guardadas anteriormente existe la posibilidad de mostrarlas en el visor e ir las recorriendo con los botones   (4).



Fig#61.Ingreso Imágenes
Fuente: Los Autores

Documentos

Para el ingreso de documentos dar clic en la pestaña **Documentos** (1) en el botón  (2) se puede buscar el documento relacionado con el equipo en el disco local, es importante mencionar que se puede guardar documentos de Excel, Word y Acrobat PDF, cuando haya elegido el documento indicado dar clic en el botón  (3) con esto se tiene ingresado un documento relacionado al equipo.

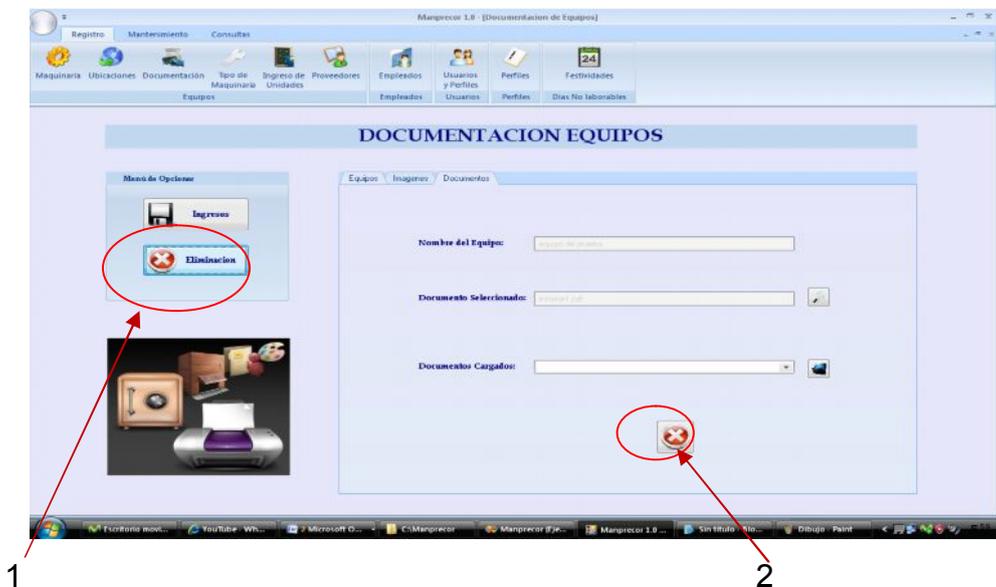


Fig#62.Ingreso Documentación

Fuente: Los Autores

Eliminación.-

Para la eliminación de documentación dar clic en el botón  (1) se escoge el documento o la imagen la cual se quiere eliminar y con clic en el botón  (2) se elimina el archivo o imagen seleccionado.



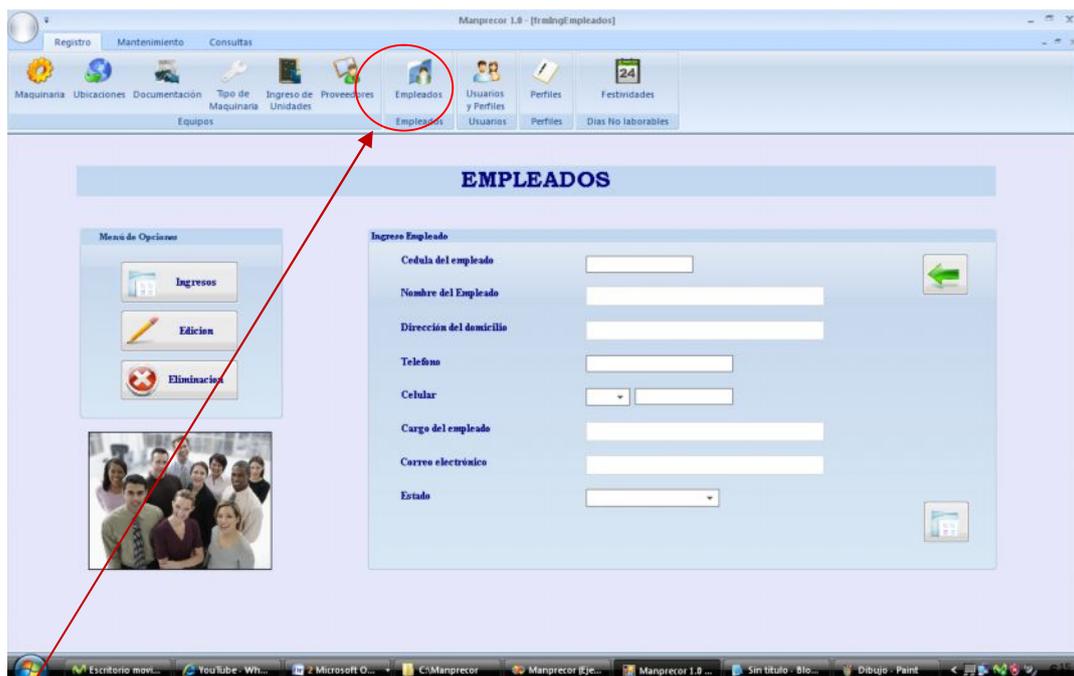
Fig#63.Eliminación Documentación

Fuente: Los Autores

Empleados.-

En el sistema MANPRECOR se tiene la opción de registrar a los empleados que trabajan en la empresa en general, para el acceso en el menú principal

MANPRECOR dar clic en el botón  (1).



1

Fig#64.Empleados

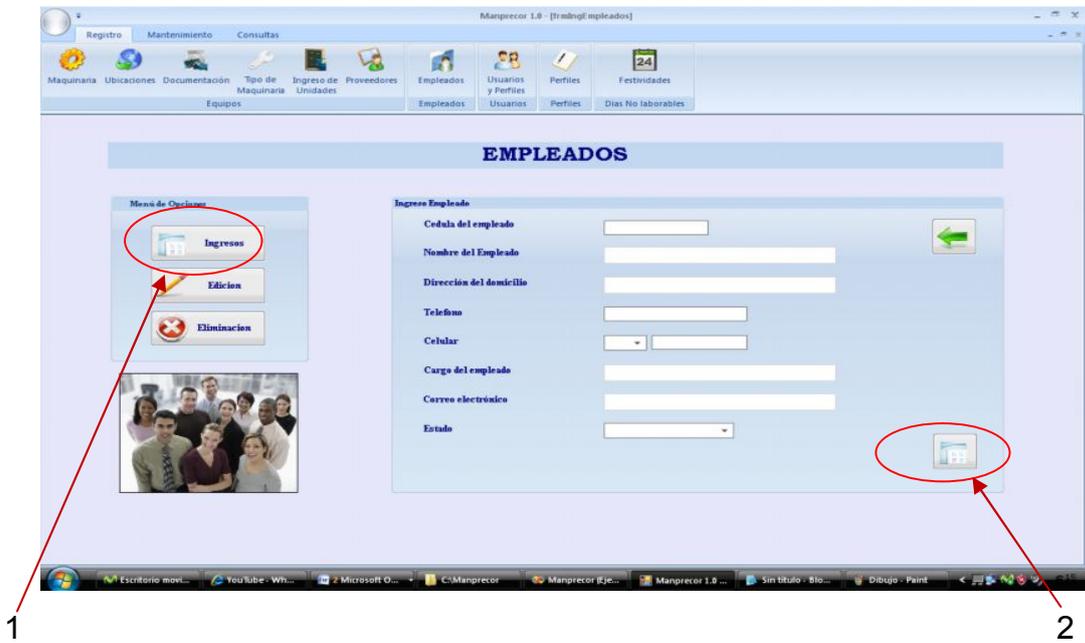
Fuente: Los Autores

Ingresos.-

Para el ingreso de un nuevo empleado dar clic en el botón  (1) ingresar

toda la información requerida en los espacios de texto y presionar el botón  (2)

una vez hecho esto se tiene ingresado un nuevo empleado.

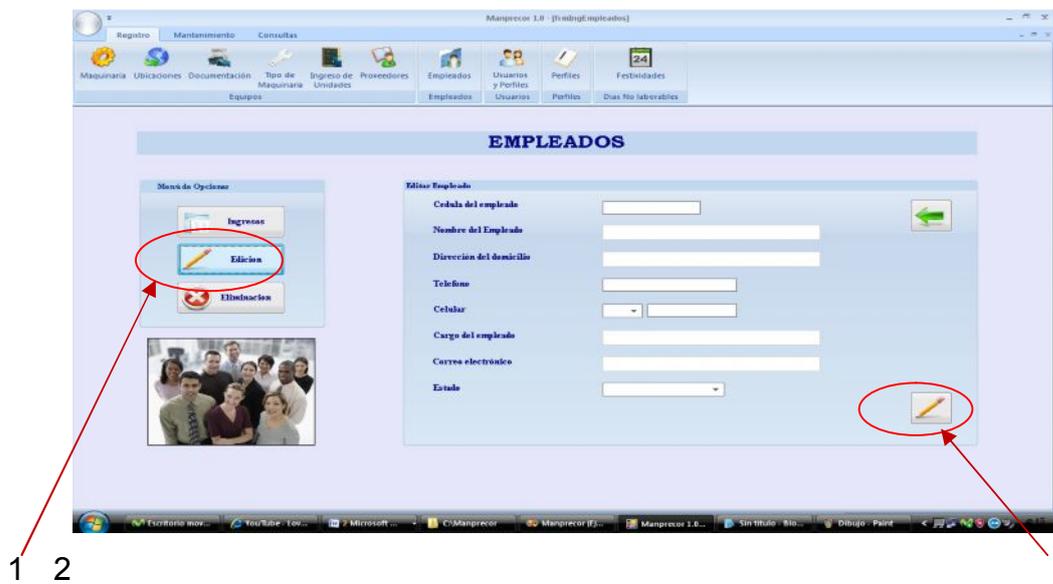


Fig#65.Ingreso Empleados

Fuente: Los Autores

Edición.-

Para la edición seleccionar de la tabla de empleados el nombre del empleado, presionar el botón  (1), editar la información deseada y luego dar clic en el botón  (2) para guardar los cambios realizados.

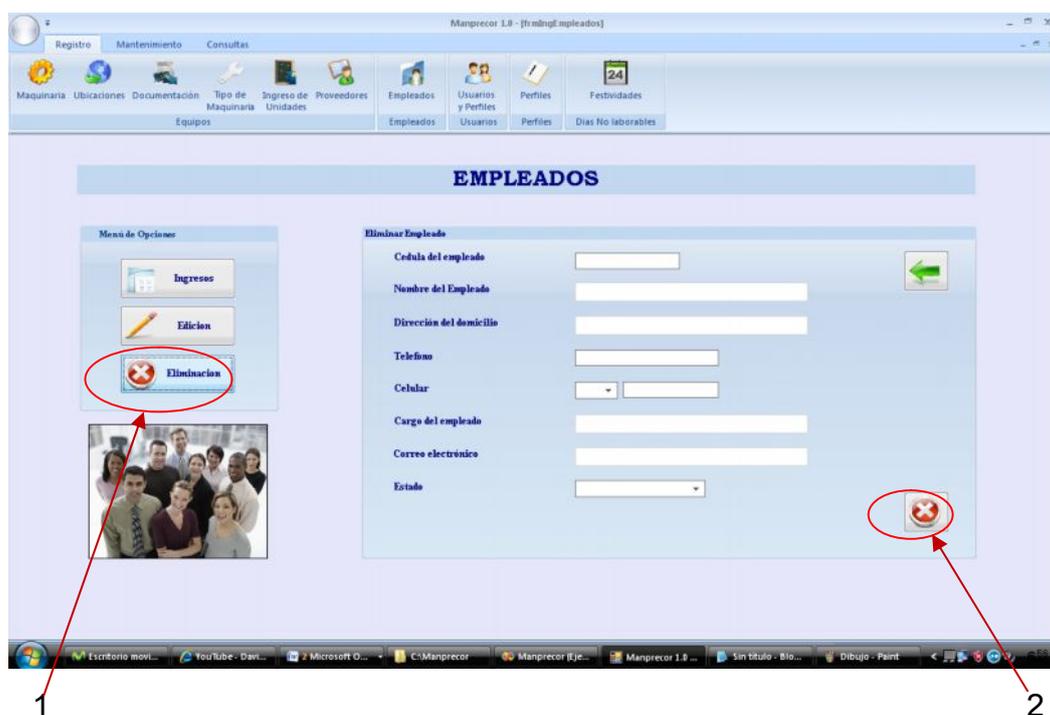


Fig#66.Edición Empleados

Fuente: Los Autores

Eliminación.-

Para la eliminación de empleados escoger en la tabla el empleado que se desea eliminar, dar clic en el botón  (1), ahora proceder a eliminar la información dando clic en el botón  (2).



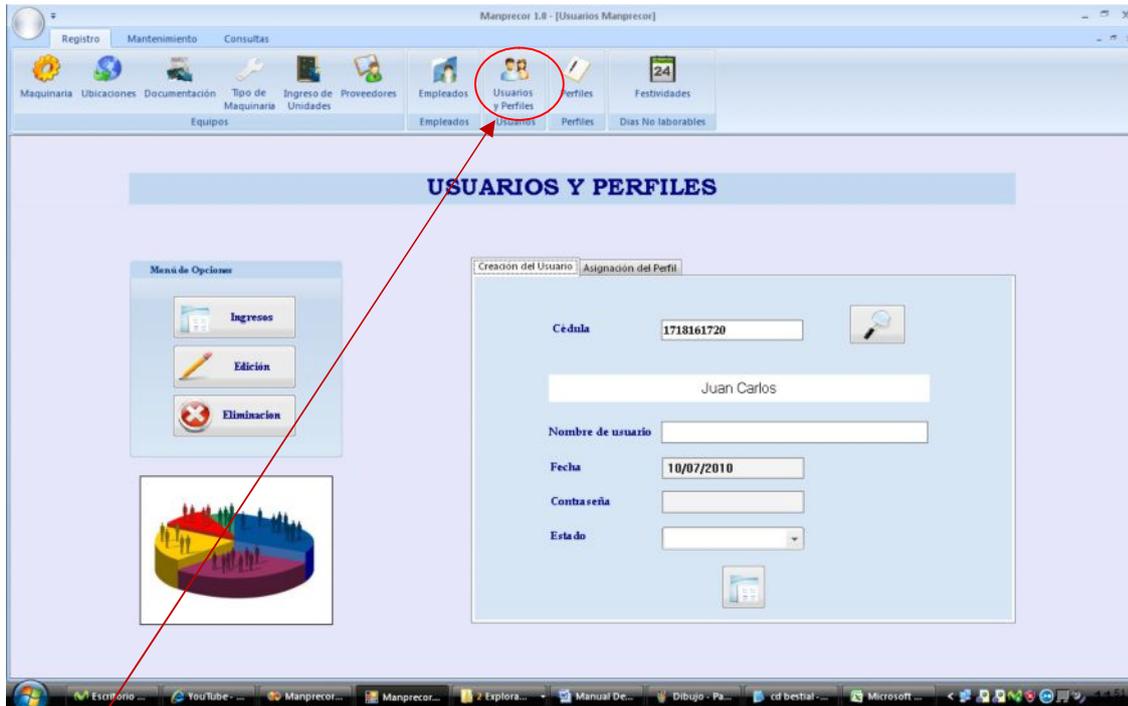
Fig#67.Eliminación Empleados

Fuente: Los Autores

Usuarios y Perfiles.-

MANPRECOR en su módulo de mantenimiento tiene la funcionalidad para el manejo de usuarios y perfiles para la administración y seguridad del sistema.

Para el acceso a Usuarios y perfiles dar clic en el botón  (1) del menú principal MANPRECOR con esto se despliega la pantalla de usuarios y perfiles con las funcionalidades de creación, edición y eliminación de usuarios que previamente debieron ser ingresados como empleados así como su perfil asociado.



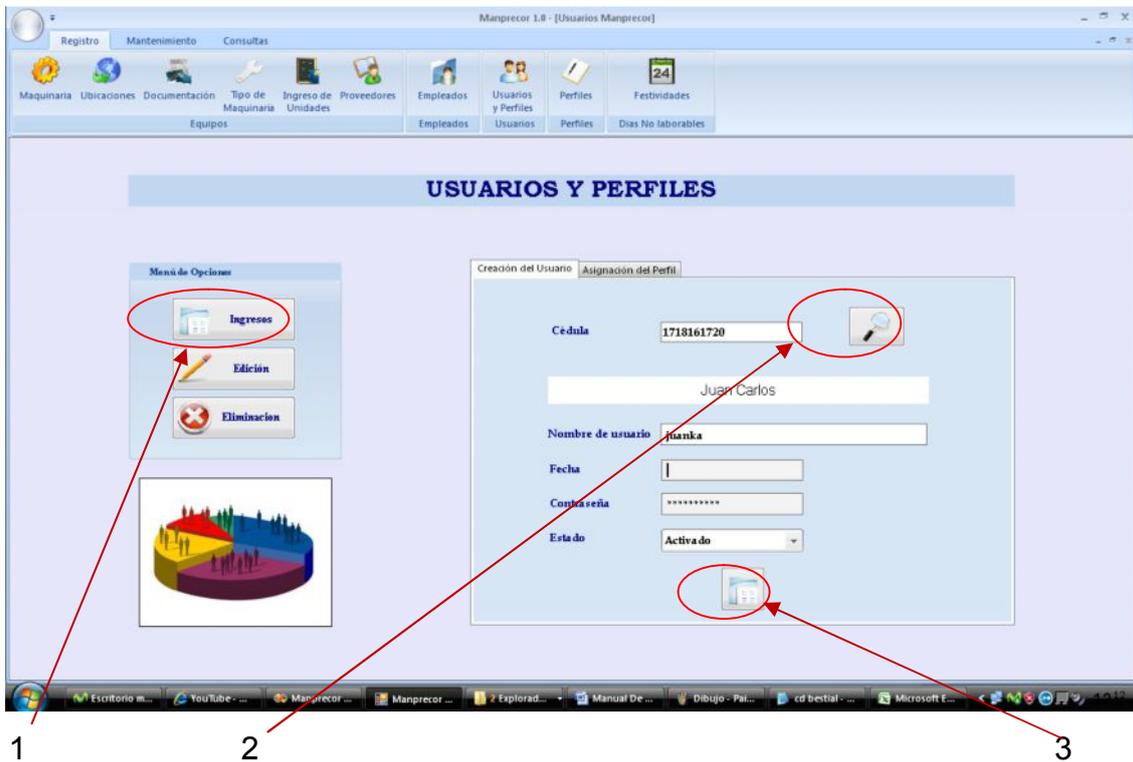
1

Fig#68.-Usuarios y perfiles

Fuente: Los Autores

Ingreso de Usuarios.-

Para el ingreso de un nuevo usuario MANPRECOR como se citó anteriormente debe estar previamente ingresado como empleado, una vez cumplido este requerimiento dar clic en el botón  (1) con esto se ingresa en el espacio de texto la cédula del empleado que se desea ingresar como usuario y dar clic en el botón  (2) se cargará la información del empleado en los espacios de texto, se registra el nombre de usuario con el cual el empleado ingresará al sistema así como la contraseña que por defecto será la cédula de identidad y el usuario deberá cambiarla una vez que ingrese con su cuenta al sistema, después presionar el botón  (3) para que el usuario quede registrado en el sistema.



Fig#69.-Ingreso Usuarios y perfiles

Fuente: Los Autores

Asignación de perfiles.-

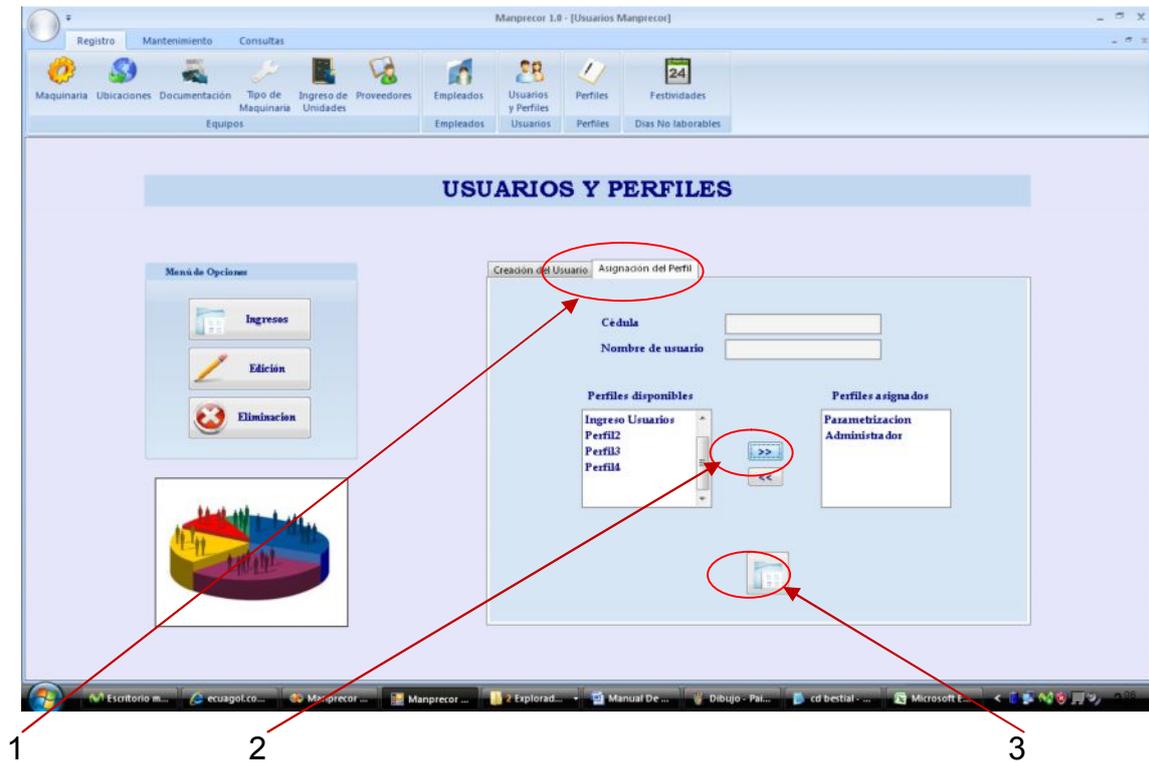
Una vez ingresado el usuario se debe asignar uno o más perfiles de acceso a las diferentes funcionalidades de MANPRECOR, para otorgar los permisos dar clic en la pestaña **Asignación del Perfil** (1) en la ventana de usuarios y aparecerá un espacio de texto para colocar la cédula del usuario al cual vamos a asignar el perfil, en la lista de la parte derecha están cargados los perfiles por defecto que se encuentran en el sistema tales como:

- Parametrización del sistema: Permite el ingreso, edición y eliminación de los registros iniciales del sistema tanto en mantenimiento como en insumos.
- Solicitante De Ordenes: Permite al usuario hacer solicitudes de insumos en el módulo correspondiente.

- Supervisor de solicitudes: Permite al usuario administrar las solicitudes de órdenes tales como la aprobación total, parcial o la negación total de las solicitudes.
- Consultas: Tiene únicamente acceso a las consultas de datos en los dos módulos del sistema.
- Ingreso Usuarios: Permite al usuario con este perfil poder ingresar, eliminar y editar los usuarios del sistema.
- Administrador: Es el super usuario del sistema y tiene acceso a todas las funcionalidades del sistema.
- Supervisor Mantenimiento: El supervisor de mantenimiento es el que tiene la posibilidad de administrar las rutinas, las órdenes y las fallas presentes en los equipos ingresados.
- Además cabe indicar que en el sistema existe la posibilidad para el usuario de crear sus propios perfiles (ver la parte de PERFILES del usuario en este manual).

Estas son las opciones que el sistema ofrece en cuanto a perfiles por defecto.

Para la asignación de un perfil al usuario presionar clic sobre el perfil que se desea escoger en la lista, una vez hecho esto dar clic en el botón  (2) con esto se carga en la lista de la derecha el perfil asignado al usuario, el usuario puede tener tantos perfiles como sean necesarios, una vez definidos todos los perfiles presionar el botón  (3) para registrar la información en el sistema.



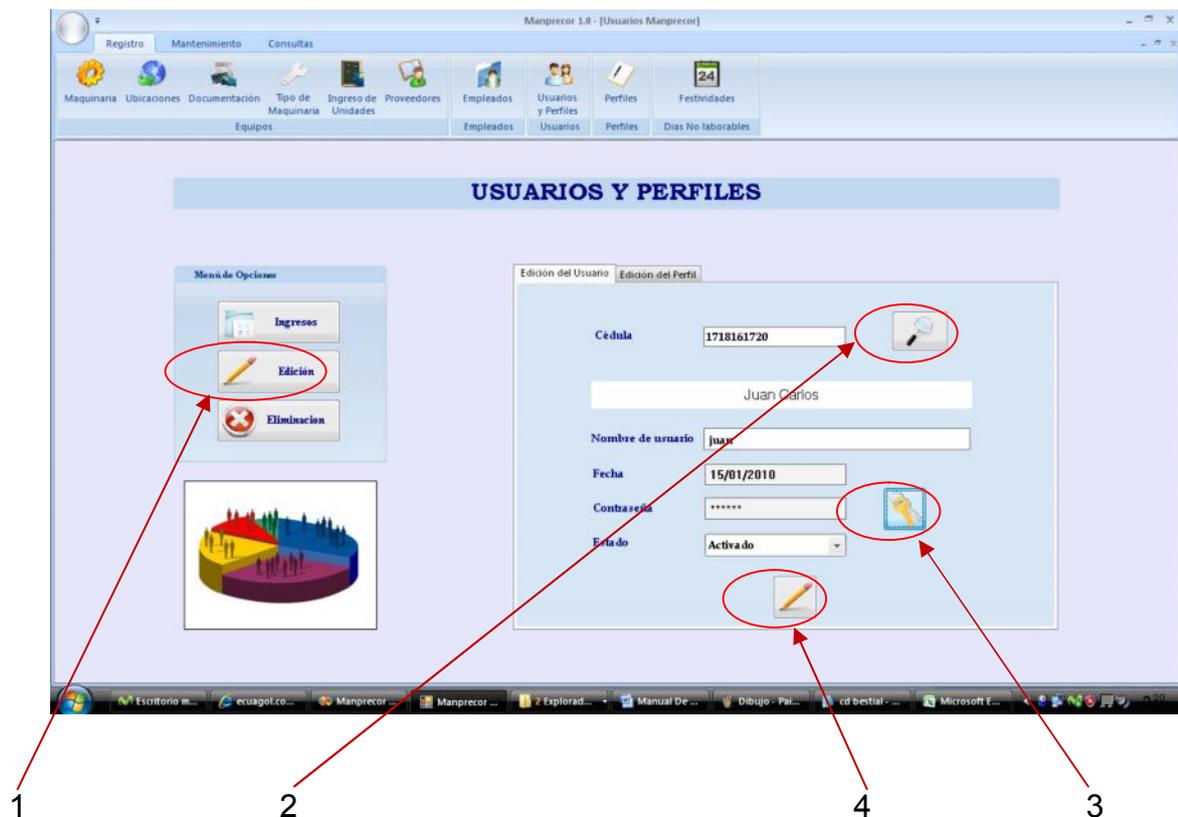
Fig#70.- Asignación Usuarios y perfiles

Fuente: Los Autores

Edición de Usuarios.-

Para la edición de usuarios se tiene que dar clic sobre el botón  (1) aparece en la parte derecha una pantalla para la edición de los datos del usuario, se digita la cédula del usuario a editar y se presiona el botón  (2) , aquí se puede editar el nombre del usuario y el estado del mismo, en cuanto a la edición de la contraseña por seguridad el sistema otorga la posibilidad de cambiarla, es decir volver a la contraseña por defecto del usuario (cédula de identidad) dando clic en el botón  (3) y se completa la edición de la contraseña.

Dar clic en el botón  (4) para guardar la edición de la información.

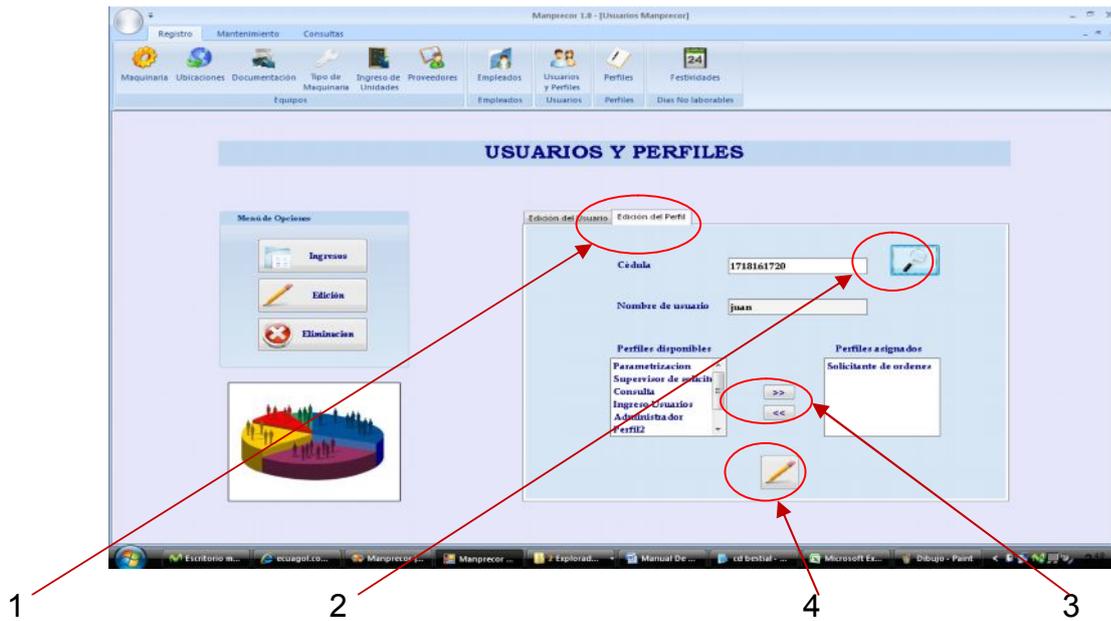


Fig#71.- Edición Usuarios y perfiles

Fuente: Los Autores

Edición de Perfiles.-

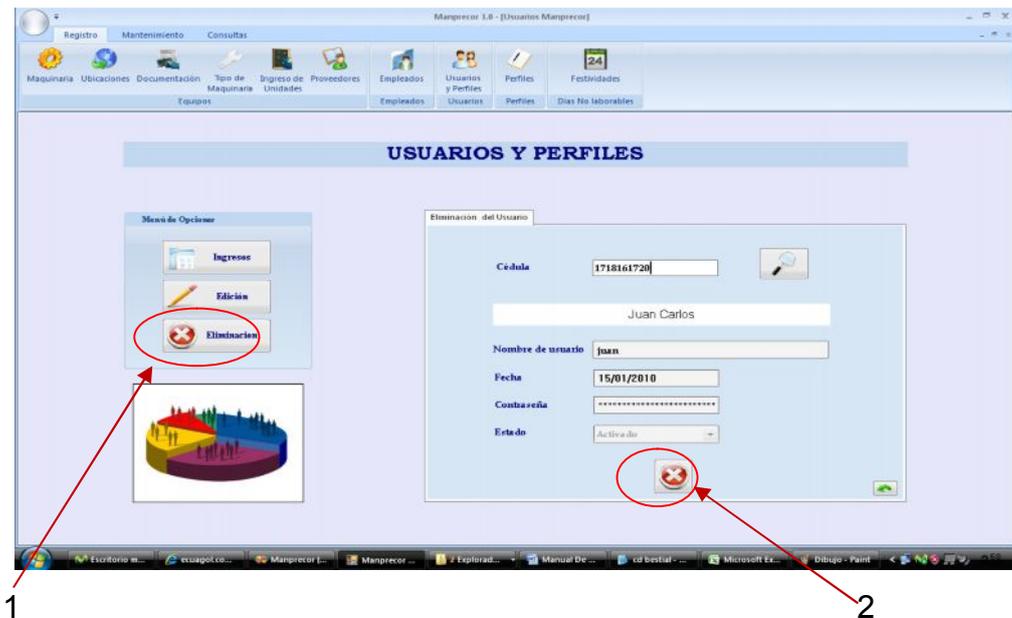
Para la edición de los perfiles dar clic en la pestaña **Edición del Perfil** (1) digitar la cédula del usuario y dar clic en el botón  (2) en la lista de la izquierda se cargarán los perfiles asignados y en la lista derecha se cargarán los perfiles que se pueden asignar al usuario, con los botones   (3) se puede tanto agregar como quitar perfiles al usuario, dar clic en el botón  (4) para registrar la edición de los perfiles.



Fig#72.- Edición Usuarios y perfiles2
Fuente: Los Autores

Eliminación de Usuarios.-

Para la eliminación de usuarios dar clic en el botón  (1) en el espacio de texto digitar la cédula del usuario que se desea eliminar, una vez realizado se carga la información del usuario para proceder a eliminarlo  (2) y el usuario queda eliminado del sistema.

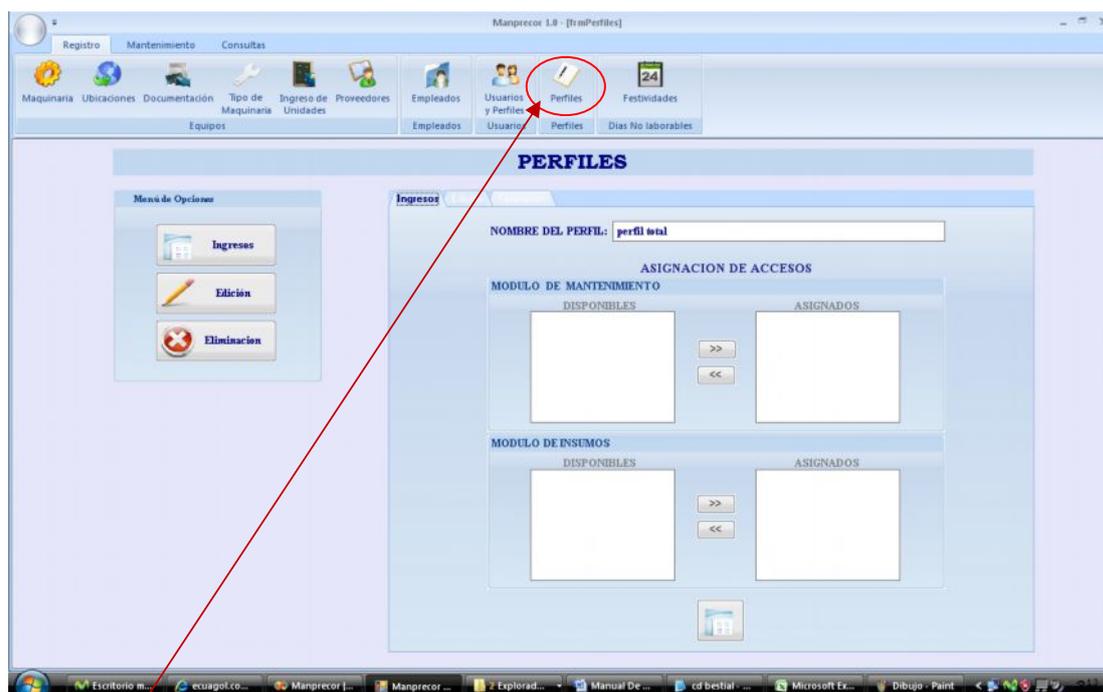


Fig#73.- Eliminación Usuarios y perfiles
Fuente: Los Autores

Perfiles.-

MANPRECOR otorga la opción al usuario de crear sus propios perfiles para que el control de acceso sea exhaustivo. Se accede a esta opción dando clic en el

botón  (1) en el menú principal MANPRECOR.



1

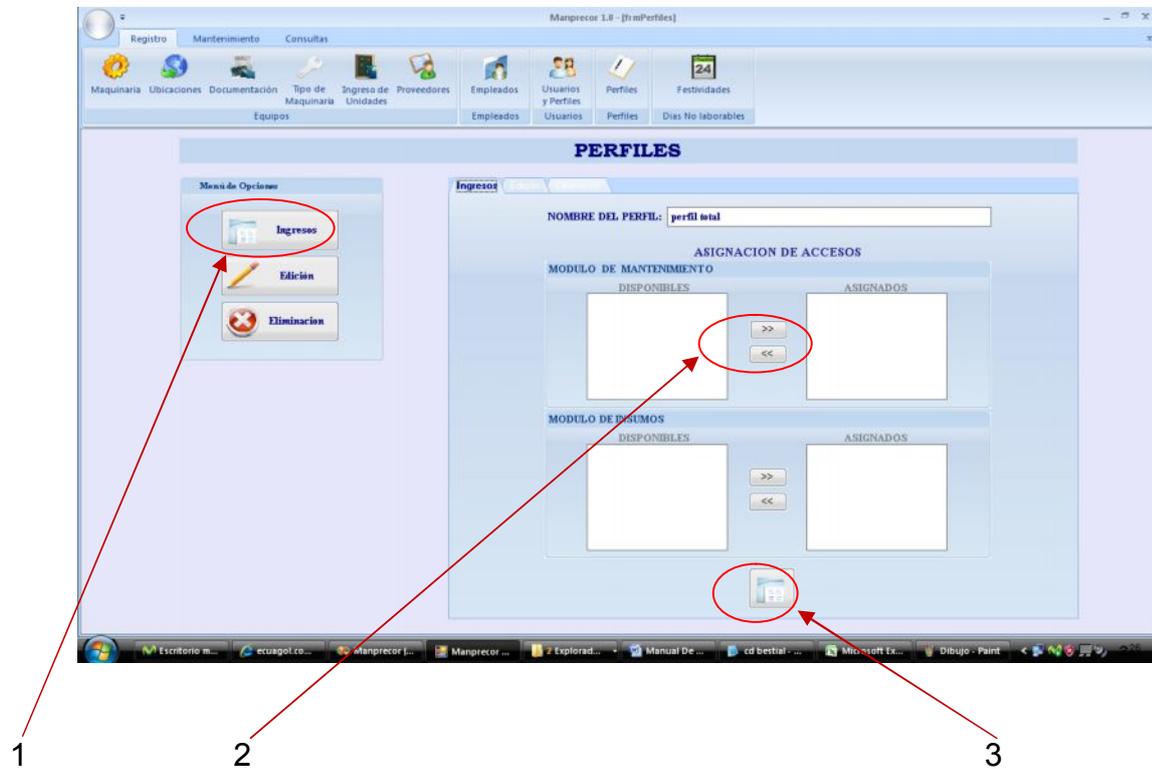
Fig#74.- Perfiles

Fuente: Los Autores

Ingreso De Perfiles.-

Para el ingreso de un nuevo perfil dar clic en el botón  (1) en el espacio de texto Nombre de perfil se indica el nombre con el cual se va a reconocer el nuevo perfil, en esta pantalla se cuenta en la parte izquierda con todas las funcionalidades tanto del módulo de mantenimiento como del módulo de insumos, se carga en las listas de la parte derecha todos los accesos que se desea tener en el nuevo perfil

con los botones   (2) y luego de que se tiene todos los accesos requeridos se presiona el botón  (3) y la información del perfil queda registrada en el sistema.

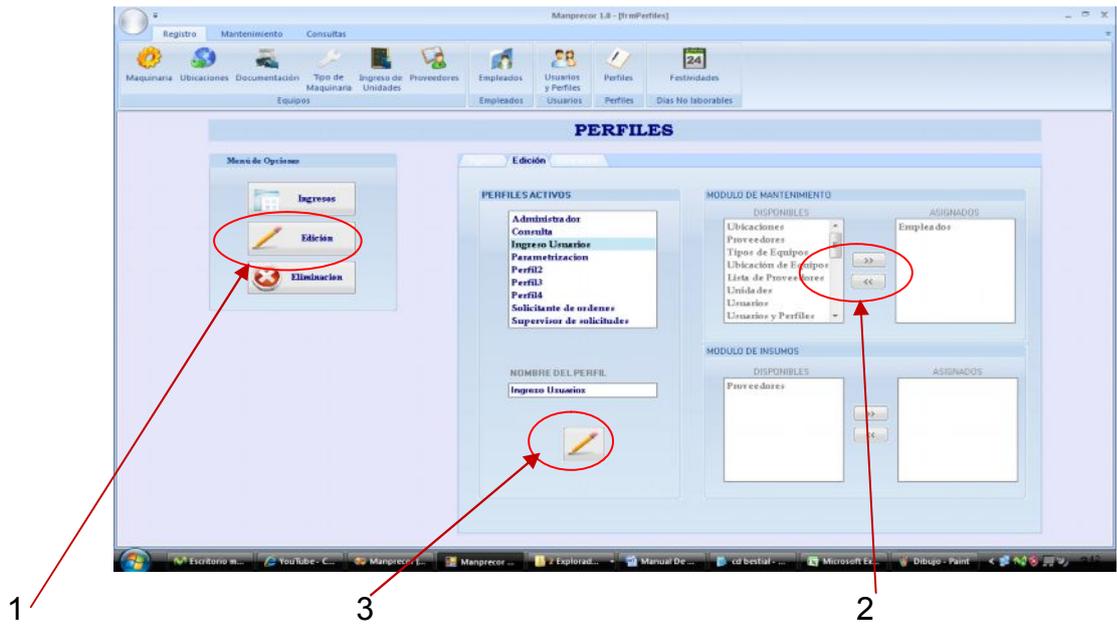


Fig#75.- Ingreso Perfiles

Fuente: Los Autores

Edición de Perfiles.-

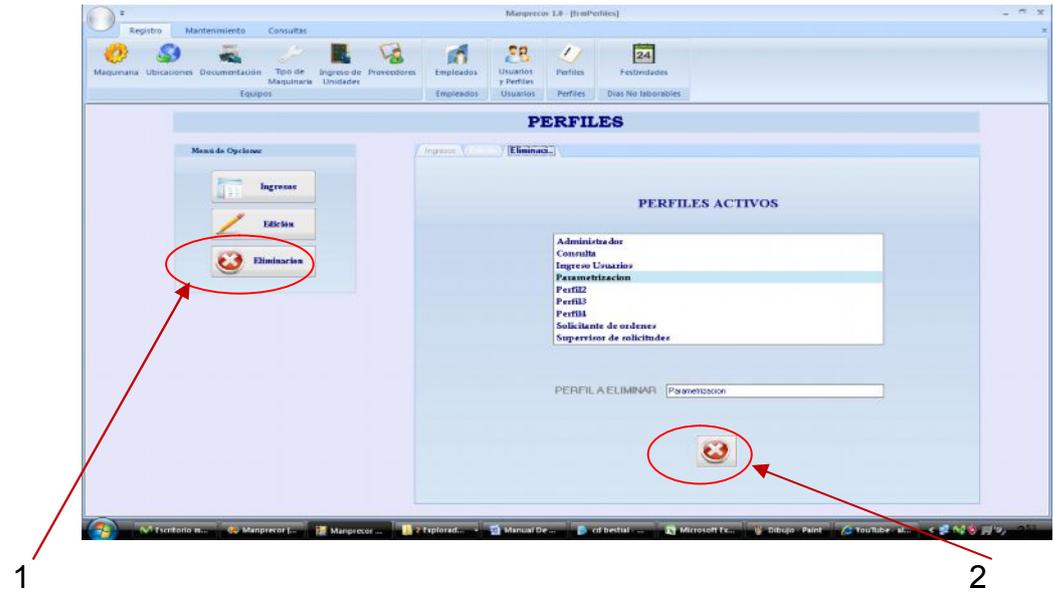
Para la edición de perfiles dar clic en el botón  (1), en la parte superior izquierda se tiene una lista con los perfiles activos del sistema con un clic sobre el nombre de alguno de estos se cargará la información de sus accesos en las listas de la parte derecha según el módulo al que pertenezcan, una vez realizado esto con los botones   (2) se puede agregar o quitar accesos a los perfiles en los diferentes módulos, cuando se haya completado la edición dar clic en el botón  (3) y se registra la edición del perfil.



Fig#76.- Edición Perfiles
Fuente: Los Autores

Eliminación de perfiles.-

Para la eliminación de los perfiles dar un clic sobre el botón  (1) en lista de los perfiles activos dar un clic sobre el que se desea eliminar y luego presionar el botón  (2) y el perfil se elimina del sistema.

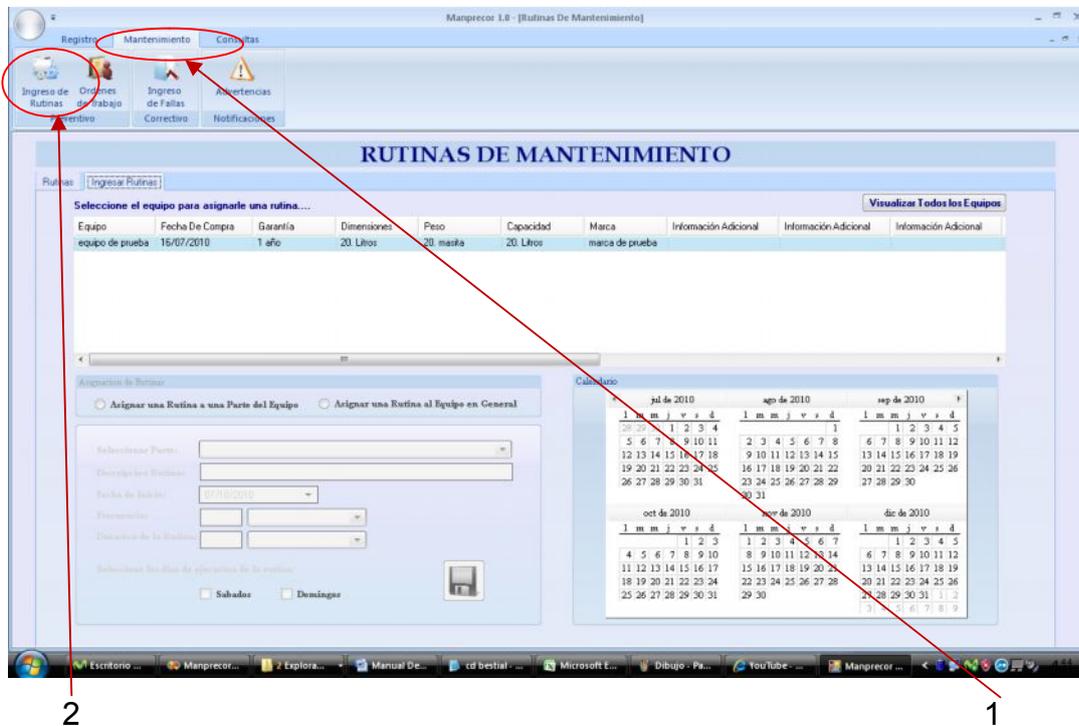


Fig#77.-Eliminación Perfiles
Fuente: Los Autores

Rutinas De Equipos.-

Dentro del sistema MANPRECOR la parte que se refiere a las rutinas de mantenimiento de equipos es una pieza fundamental, ya que aquí es donde se registran las fechas en las que se tienen que llevar a cabo el mantenimiento de cada equipo según las especificaciones entregadas al sistema. Para el acceso a la pantalla de rutinas dar un clic en la pestaña **Mantenimiento** (1) en la parte superior del

menú principal MANPRECOR, presionar el botón **Ingreso de Rutinas** (2) e ingresar a la pantalla de administración de rutinas.



Fig#78.-Rutinas
Fuente: Los Autores

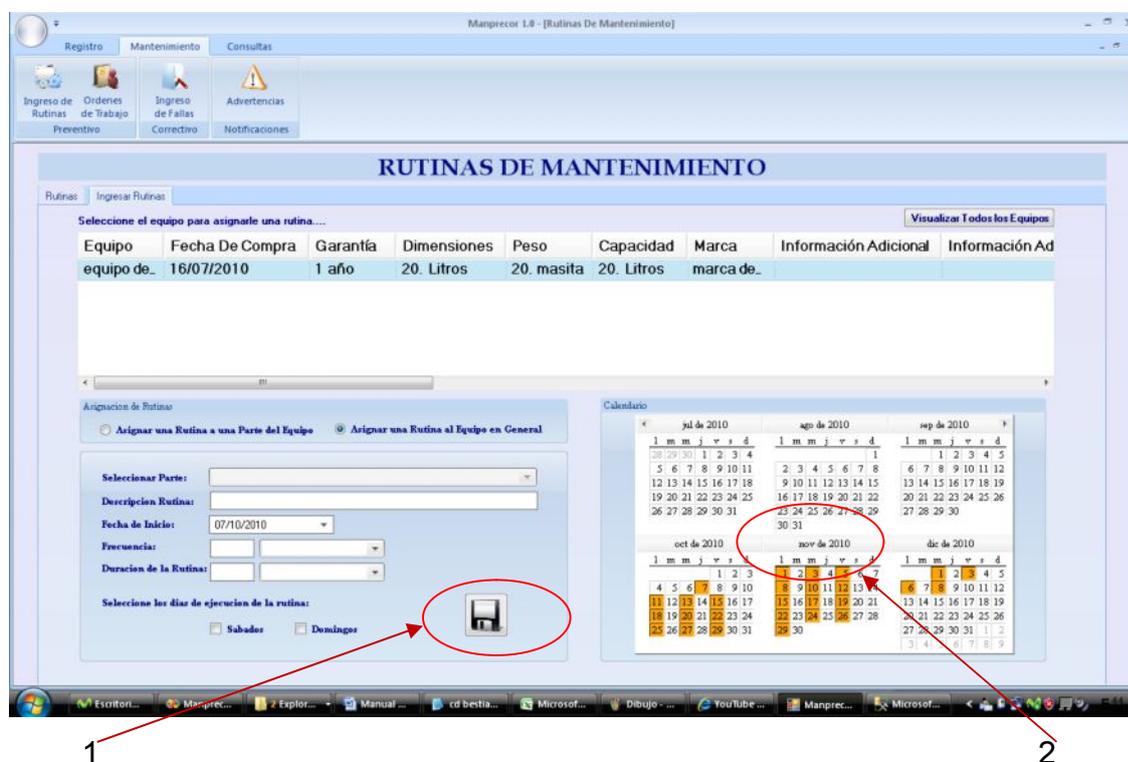
Ingreso de una Nueva Rutina.-

Para ingresar una nueva rutina el primer paso es escoger el equipo al cual afectará la rutina dando doble clic sobre la tabla de opciones de equipos.

Una vez realizado esto en la parte inferior de la tabla de información escoger si la rutina es para el equipo en general ó para una parte específica ya ingresada en el equipo, en los espacios de texto que ahora están habilitados para ingresar información tal como la fecha de inicio, la frecuencia de la rutina que indicará con qué frecuencia en días o meses se volverá a realizar la rutina si fuese necesario. Duración de la rutina contiene el tiempo en que estará activa y se realizará la rutina. En la Opción de selección de los días de ejecución nos da la posibilidad de indicar si se requiere que la rutina se ejecute los días sábados o domingos.

Una vez ingresada toda esta información dar clic en el botón  (1) y guardar la información de la rutina.

En la parte derecha de la pantalla de rutinas se tiene un calendario donde se puede visualizar todas las fechas a futuro donde se realizará el mantenimiento de la maquinaria (2).



Fig#79.-Ingreso Rutinas
Fuente: Los Autores

En esta pantalla se tiene la posibilidad de visualizar las rutinas ingresadas anteriormente dando clic en la pestaña **Rutinas** en la parte superior de la pantalla, aquí se puede ver todas las rutinas ingresadas e inclusive las fechas en las cuales se deben realizar las órdenes de trabajo.

Órdenes de Trabajo.-

Las órdenes de trabajo son acciones que se realizan a partir de las rutinas ingresadas en el sistema, en una fecha indicada se debe cumplir una rutina de mantenimiento es aquí cuando se debe generar una orden de trabajo atada a esta rutina y que afectará al equipo indicado.

En las órdenes de trabajo de MANPRECOR se ingresará información como una descripción breve de la orden, a que rutina pertenece la orden, a que equipo y a que parte se va a afectar con la orden, la fecha de inicio, la fecha máxima de realización así como el estado que puede variar entre: abierta, cerrada, en espera de insumos o anulada. Para el ingreso a la pantalla de órdenes de trabajo presionar el botón



(1) en el menú principal de MANPRECOR.



1

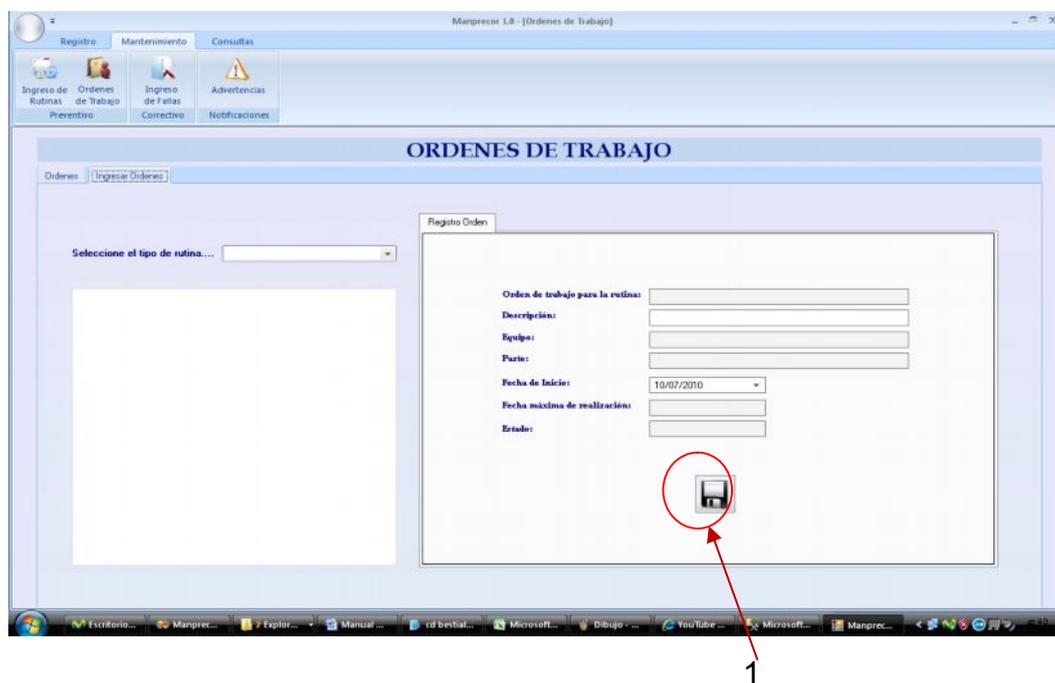
Fig#80.-Órdenes de trabajo

Fuente: Los Autores

Ingreso de Órdenes de trabajo.-

Para el ingreso de una nueva orden de trabajo ya sea una rutina de equipo en general o de una parte específica, se debe seleccionar la opción “Órdenes de Trabajo” en la lista de información que se despliega en la parte superior.

Una vez hecho esto se carga en la parte derecha toda la información relacionada con la rutina, ingresar una descripción para la orden y dar clic en el botón  (1) y se ingresa la orden de trabajo.



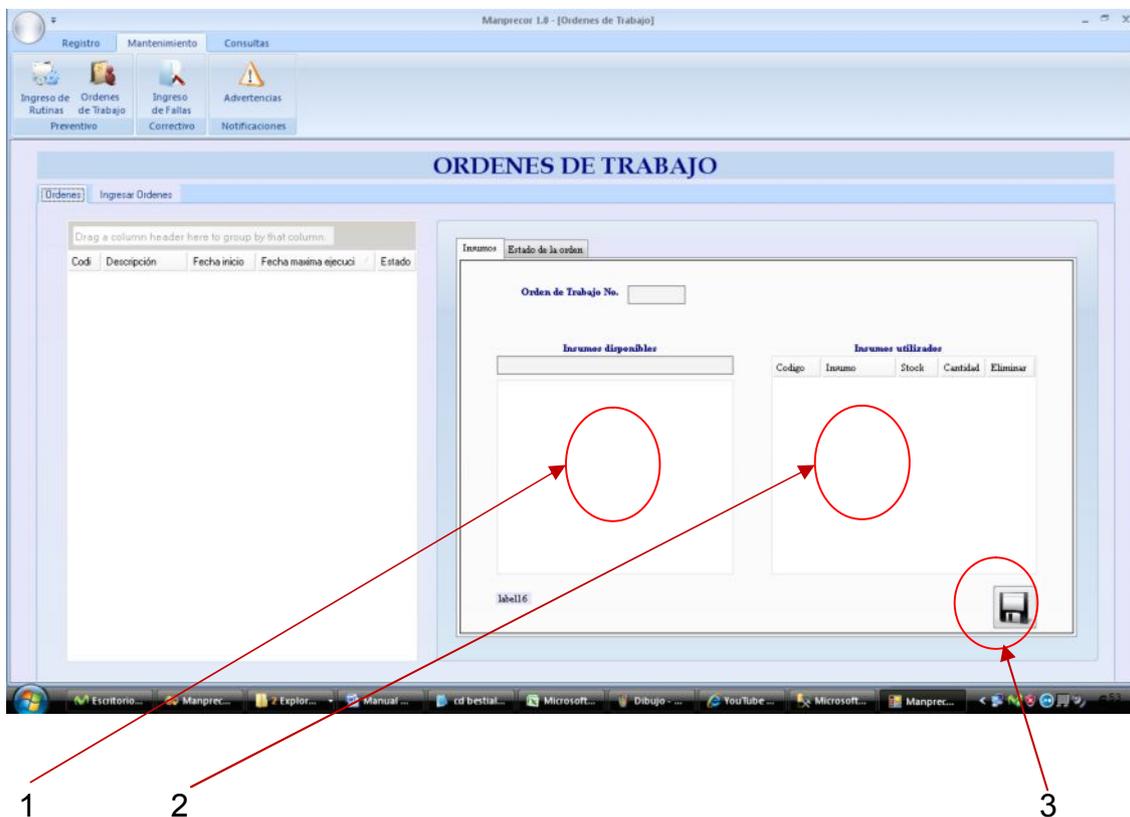
Fig#81.-Ingreso Ordenes de trabajo

Fuente: Los Autores

Ingreso de Insumos en la orden de trabajo.-

Para el ingreso de los insumos utilizados en la orden de trabajo dar doble clic en la tabla de órdenes sobre el registro de la orden que se desea abrir, cuando se realiza esto en la parte derecha se carga el número de la orden escogida, en el espacio de texto Insumos disponibles se tiene una consulta de los insumos que existen en stock e ingresados en el sistema, en la tabla inferior se cargan los insumos

disponibles(1), con doble clic sobre el insumo se carga en la tabla de opciones derecha con título insumos utilizados(2), en esta se debe definir cuál fue la cantidad que se utilizó del insumo, una vez definidos los insumos utilizados dar clic en el botón  (3) e ingresar el stock utilizado dentro de la orden.



Fig#82.-Ingreso Insumos

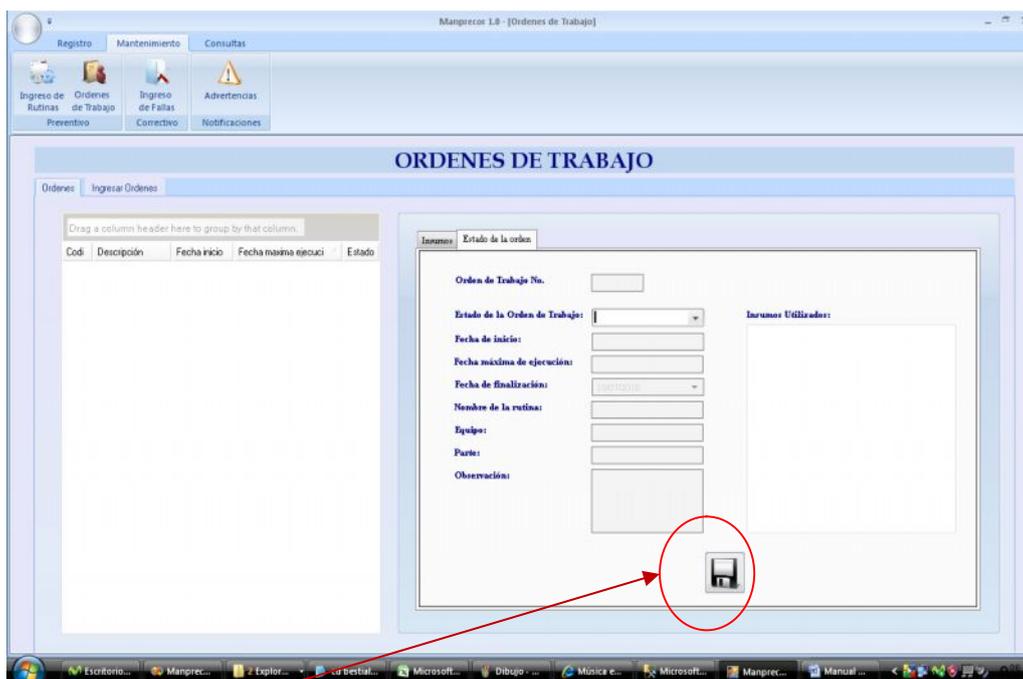
Fuente: Los Autores

Cierre de las órdenes de trabajo.-

Una vez que el encargado del mantenimiento del equipo y de una orden en general termina su trabajo deberá cerrar en el sistema la orden de trabajo que en este momento se encuentra en estado en proceso. Para esto dirigirse a la pestaña estado de la orden que se ubica en la parte superior de la pantalla, dar doble clic sobre la orden que se desea cerrar, se cargará el número de la orden en la parte derecha de la pantalla, ahora bien aquí se tiene la posibilidad de cambiar el estado de la orden de trabajo a cerrada o en proceso, en la tabla de insumos utilizados, se carga la

información ya ingresada de los insumos y en el espacio de texto de observaciones ingresamos una descripción breve del porque del estado que estamos ingresando a

la orden, dar clic en el botón  (1) con esto se ingresa el nuevo estado de la orden.



1

Fig#83.-Cierre de orden

Fuente: Los Autores

Fallas en los equipos.-

En el proceso de mantenimiento de maquinarias se pueden presentar fallas en el funcionamiento, conscientes de esto MANPRECOR tiene en su estructura una funcionalidad para almacenar las fallas que se presenten en el trabajo diario de un equipo.

Para acceder a la funcionalidad de Fallas dar clic en el botón  (1) ubicado en el menú principal MANPRECOR.



1

Fig#84.-Fallas

Fuente: Los Autores

Ingreso De fallas.-

Para el ingreso de una falla en el sistema dar doble clic sobre el equipo que se va a registrar la falla en su funcionamiento, una vez realizado esto se selecciona en que lugar se presentó la falla si fue en el equipo en general o fue en una parte del equipo, con esto se habilita el espacio de texto en la parte inferior, si se selecciona que la falla se presentó en una parte del equipo se tiene que escoger en que parte se presentó. Ingresamos una descripción de la falla y la fecha en la cual se registró.



Fig#85.-Ingreso Fallas

Fuente: Los Autores

Registro de Solución.-

Para el registro de la solución de la falla ingresada se debe dar clic en el botón



(1) Con esto se despliega una pantalla auxiliar en la cual se puede registrar en el espacio de texto una descripción de la solución de la falla que se presentó en el equipo indicado.



Fig#86.-Registro de solución

Fuente: Los Autores

Días no Laborables.-

En el registro de las rutinas cuando se generan fechas para la realización de las mismas una de estas fechas puede coincidir con un día no laborable y la rutina no se lleve a cabo ya que en la empresa no existen personas que tomen acción sobre el sistema, es por eso que MAMPRECOR ofrece la funcionalidad de ingresar al sistema las fechas no laborables durante el año para que cuando coincida una fecha de rutina con una fecha no laborable entonces la rutina se realice en un día anterior o posterior de la fecha no laborable.

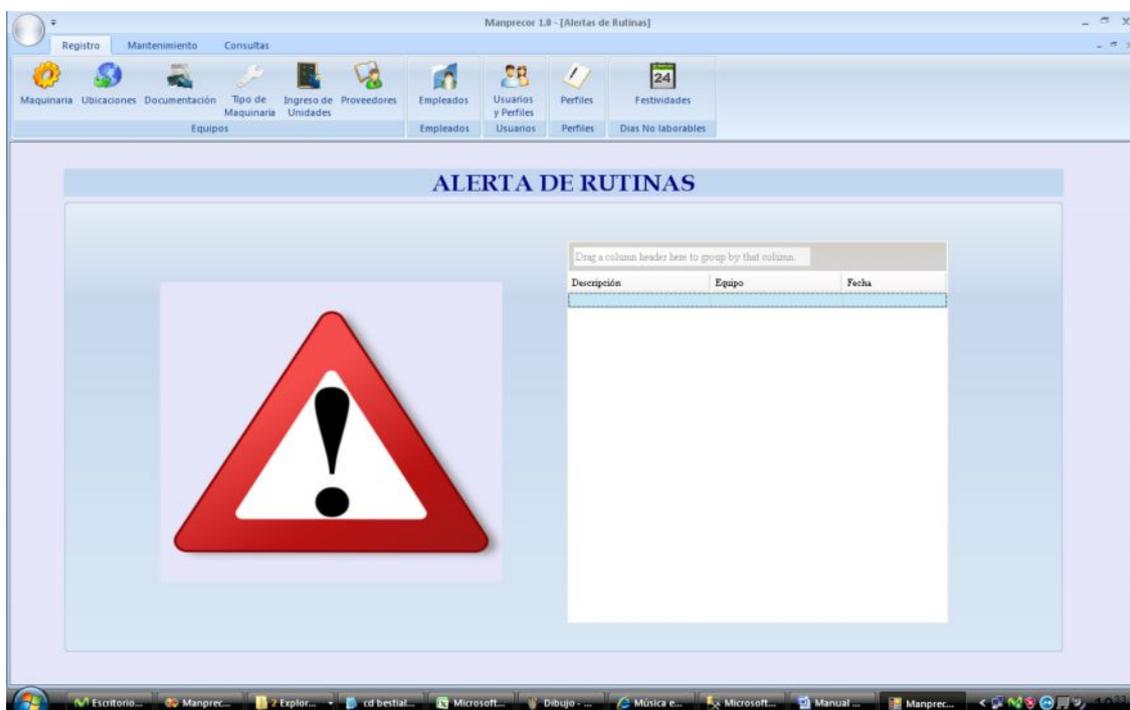
Para acceder a la pantalla de fechas no laborables se debe seleccionar la opción “Festividades” del menú principal MANPRECOR.

Ingreso De Fechas No laborables.-

Para el ingreso de la fecha no laborable escoger en el calendario de la parte izquierda la fecha en la cual no se trabajará, se ingresa con una descripción de la fecha y dar clic en el botón Grabar para registrar la información.

Alerta De Rutinas.-

En la pantalla de alerta de rutinas la cual se carga automáticamente en el ingreso al módulo de mantenimiento se presenta la información de las rutinas que se deben realizar en los próximos 5 días según el usuario que ingresó al sistema y si tiene a su cargo alguna rutina.

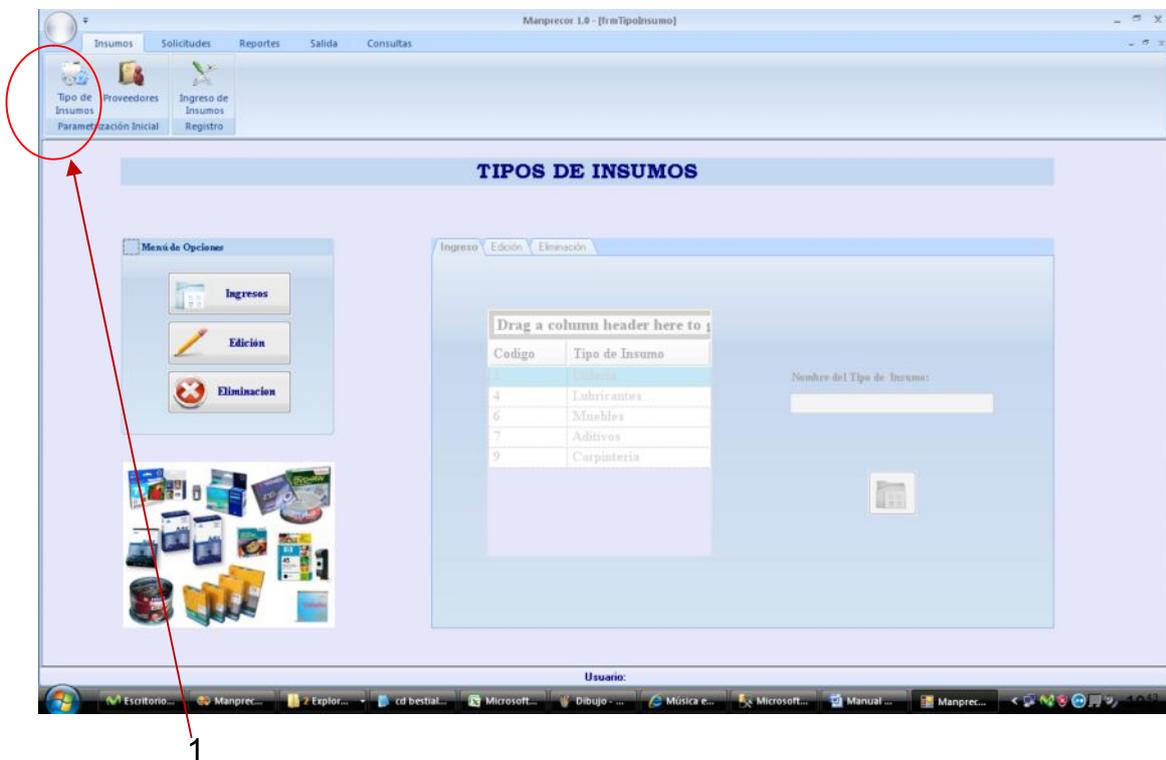


Fig#87.-Alertas
Fuente: Los Autores

MÓDULO DE INSUMOS

Tipos de Insumos.-

En la parametrización inicial del módulo de insumos se cuenta con el registro de los tipos de insumos, para el ingreso dar clic en el botón  (1) en el menú principal MANPRECOR.



Fig#88.-Tipos de insumos

Fuente: Los Autores

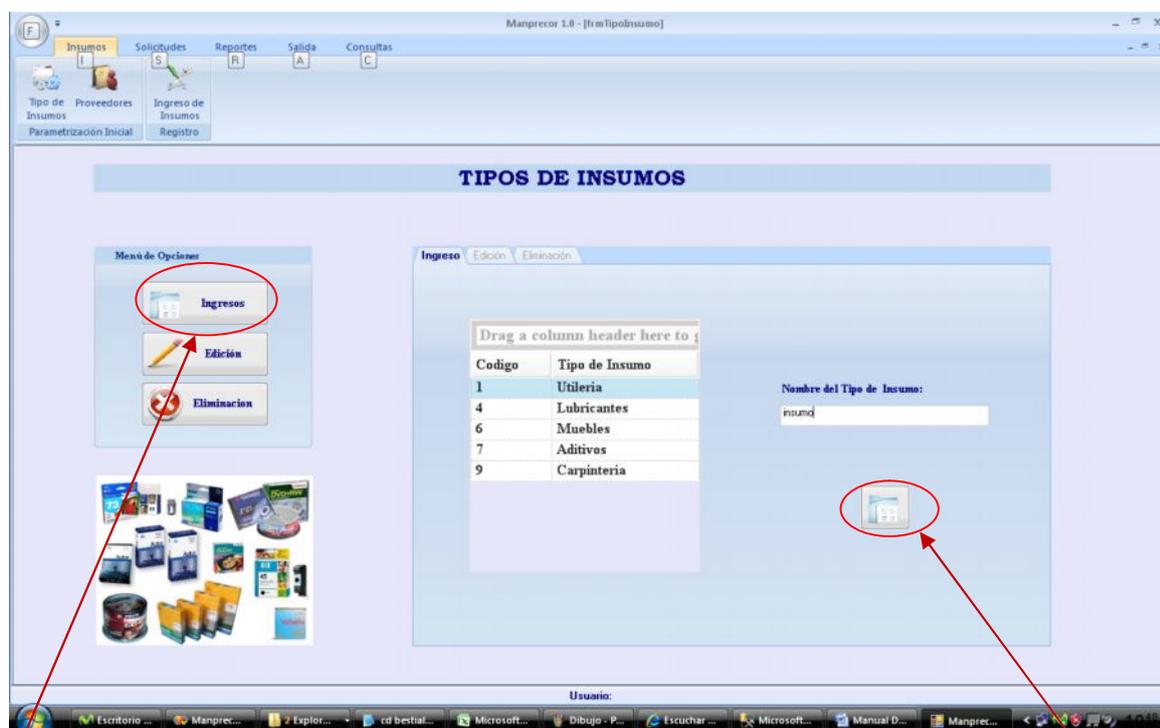
Ingreso de Tipos De Insumos.-

Para el ingreso de tipos de insumos en el sistema dar clic en el botón  (1) en el menú de opciones en la parte derecha se tiene una tabla de opciones donde se mostrarán los tipos de insumos ya ingresados y los que se sigue ingresando, en

el espacio de texto se digita el nombre del tipo de insumo y con un clic en el botón



(2) se ingresa el tipo de insumo.



1

2

Fig#89.-Ingreso Tipos de insumos

Fuente: Los Autores

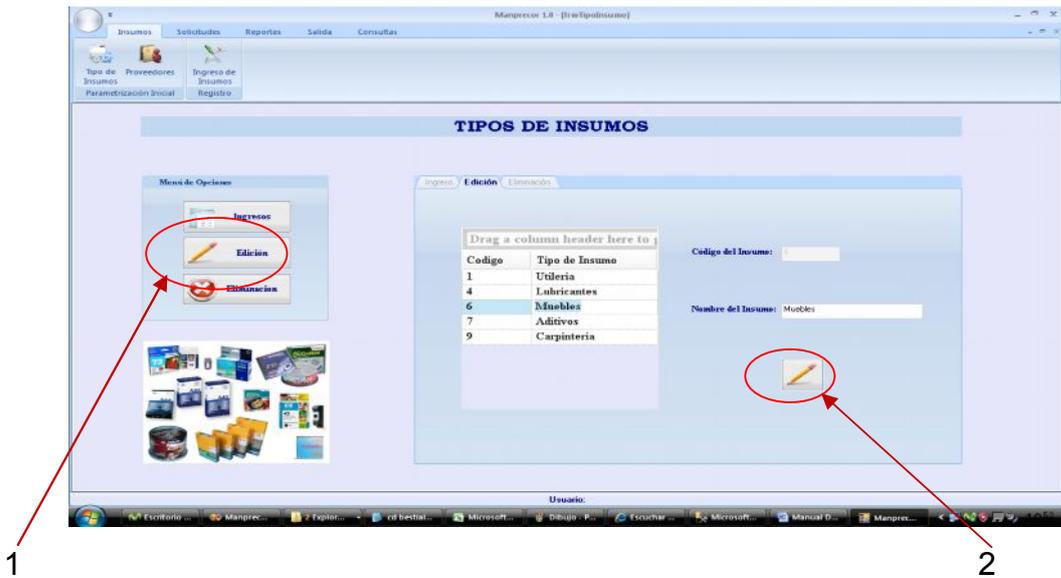
Edición de tipos de Insumos.-

Para la edición de los tipos de insumos ingresados dar un clic en el botón



(1), luego con doble clic sobre la tabla de los tipos de insumos ingresados

se selecciona, modificar la información deseada y dar un clic en el botón  (2) modificando la información.

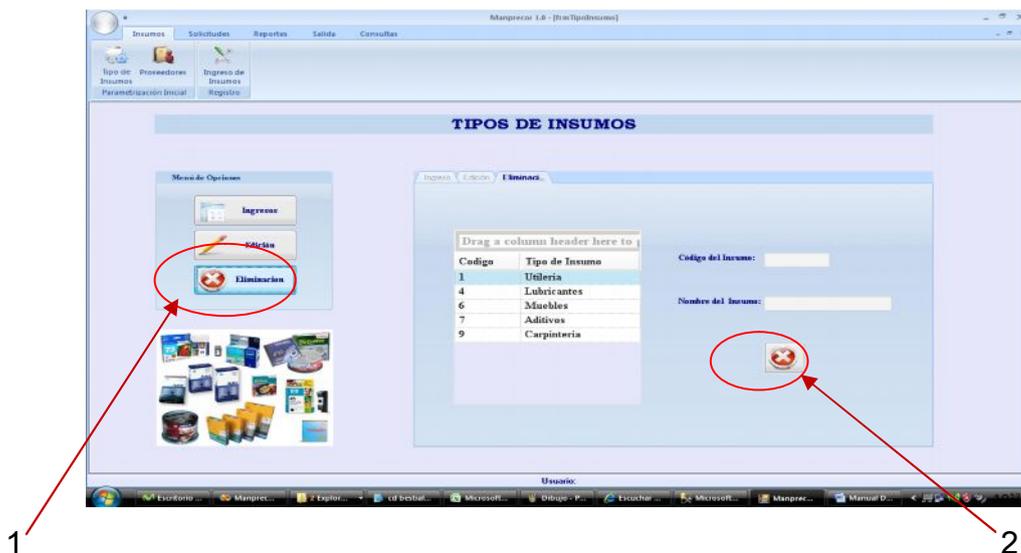


Fig#90.-Edición Tipos de insumos

Fuente: Los Autores

Eliminación de Tipos De Insumos.-

Para la eliminación de tipos de insumos dar clic sobre el botón  Eliminación (1) escoger con doble clic en la tabla el tipo de insumo que se desea eliminar y presionar el botón  (2) y la información se eliminará.



Fig#91.-Eliminación Tipos de insumos

Fuente: Los Autores

Proveedores de Insumos.-

En la pantalla de proveedores de insumos se almacena toda la información relacionada con las personas o empresas que proveen a la empresa de insumos para el mantenimiento. Para acceder a la pantalla dar clic sobre el botón  (1) en el menú principal MANPRECOR.



1

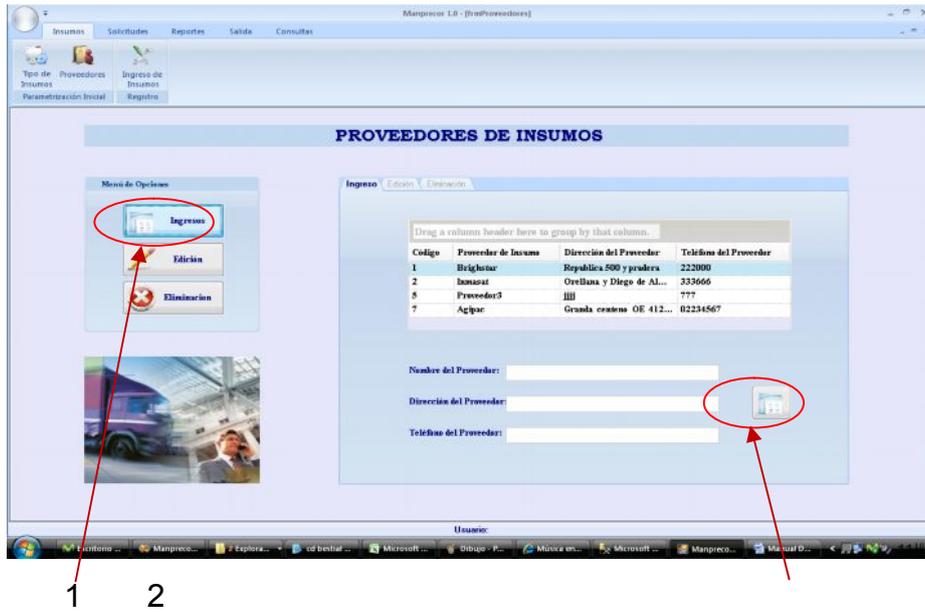
Fig#92.-Proveedores

Fuente: Los Autores

Ingreso de Proveedores.-

Para el ingreso de proveedores dar clic sobre el botón  (1) en el menú de opciones.

Una vez hecho esto ingresar la información requerida por el sistema y presionar el botón  (2) e ingresar la información.



Fig#93.-Ingreso Proveedores

Fuente: Los Autores

Edición De Proveedores.-

Para la edición de proveedores presionar el botón  (1) en el menú de opciones, dar doble clic en la tabla escogiendo el proveedor a editar y luego presionar el botón  (2) editando la información deseada.

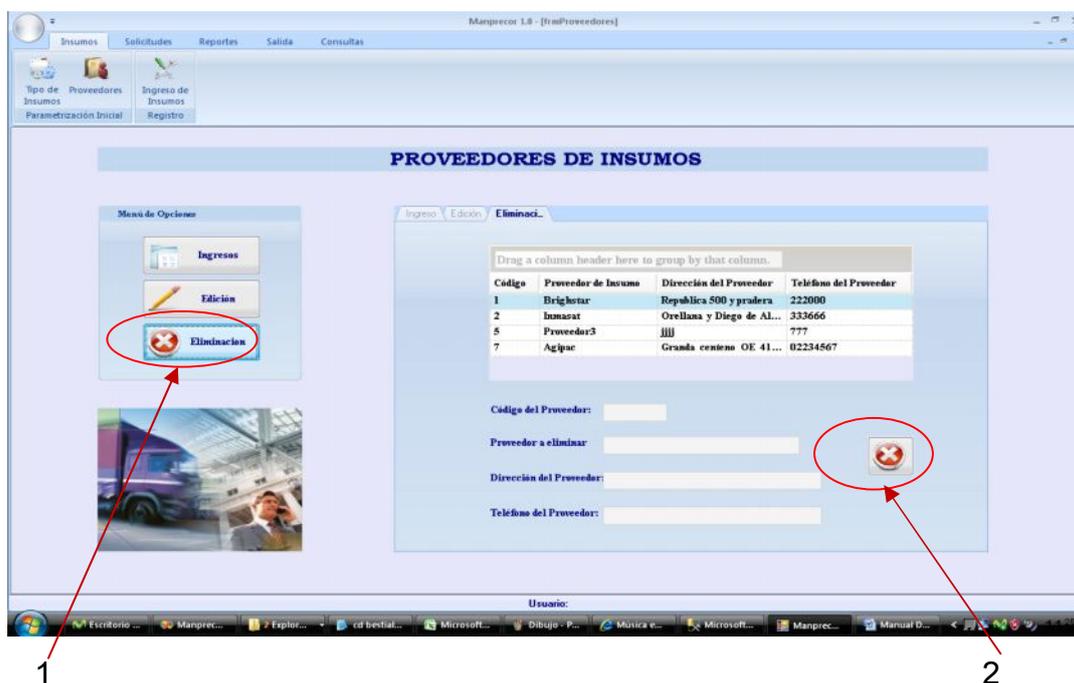


Fig#94.-Edición Proveedores

Fuente: Los Autores

Eliminación De Proveedores.-

Para la eliminación de proveedores dar clic sobre el botón  (1) en el menú de opciones, escoger en la tabla de información el proveedor que se desea eliminar y presionar el botón  (2) y eliminar al proveedor del sistema.

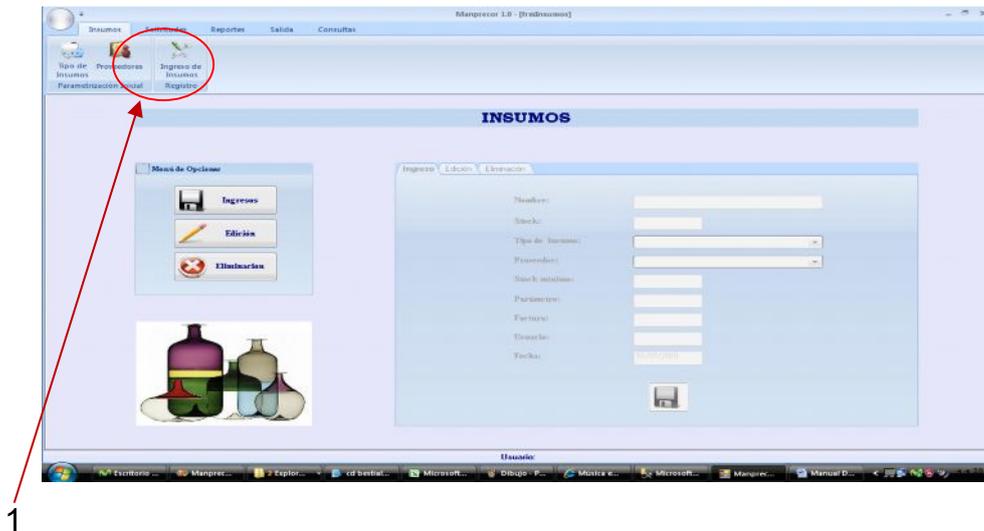


Fig#95.-Eliminación Proveedores

Fuente: Los Autores

Insumos.-

Para el registro de los insumos que ingresan a la empresa, MANPRECOR cuenta con una pantalla a la cual se accede con un clic en el botón  (1) ubicado en el menú principal MANPRECOR.

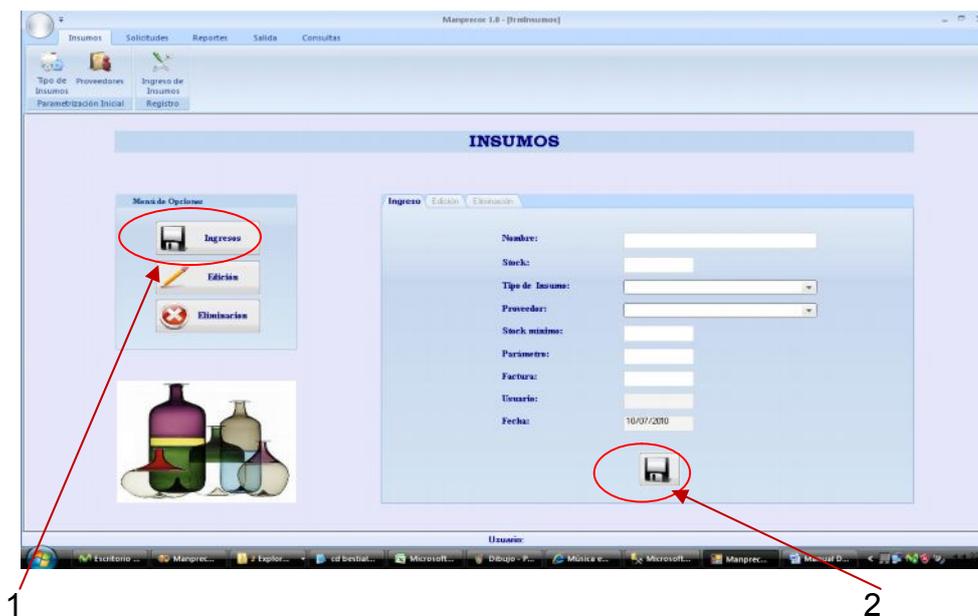


Fig#96.-Insumos

Fuente: Los Autores

Ingreso De Insumos.-

Para el ingreso de un nuevo insumo dar clic sobre el botón  (1) en el menú de opciones, una vez hecho esto ingresar toda la información solicitada por el sistema y dar clic en el botón  (2) para guardar el nuevo insumo y sus existencias dentro del sistema.

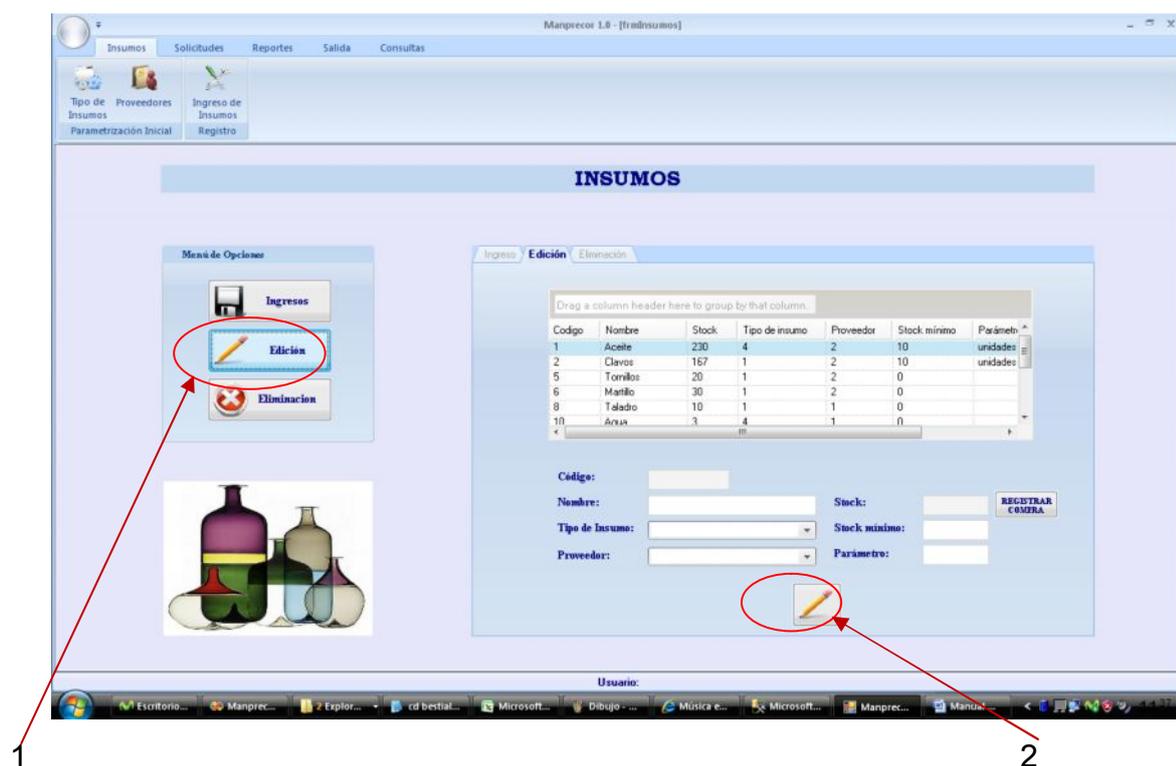


Fig#97.-Ingreso Insumos

Fuente: Los Autores

Edición de Insumos.-

Para la edición de insumos dar un clic sobre el botón  (1) en la tabla escoger el insumo que se va a editar y presionar el botón  (2) editando información deseada.

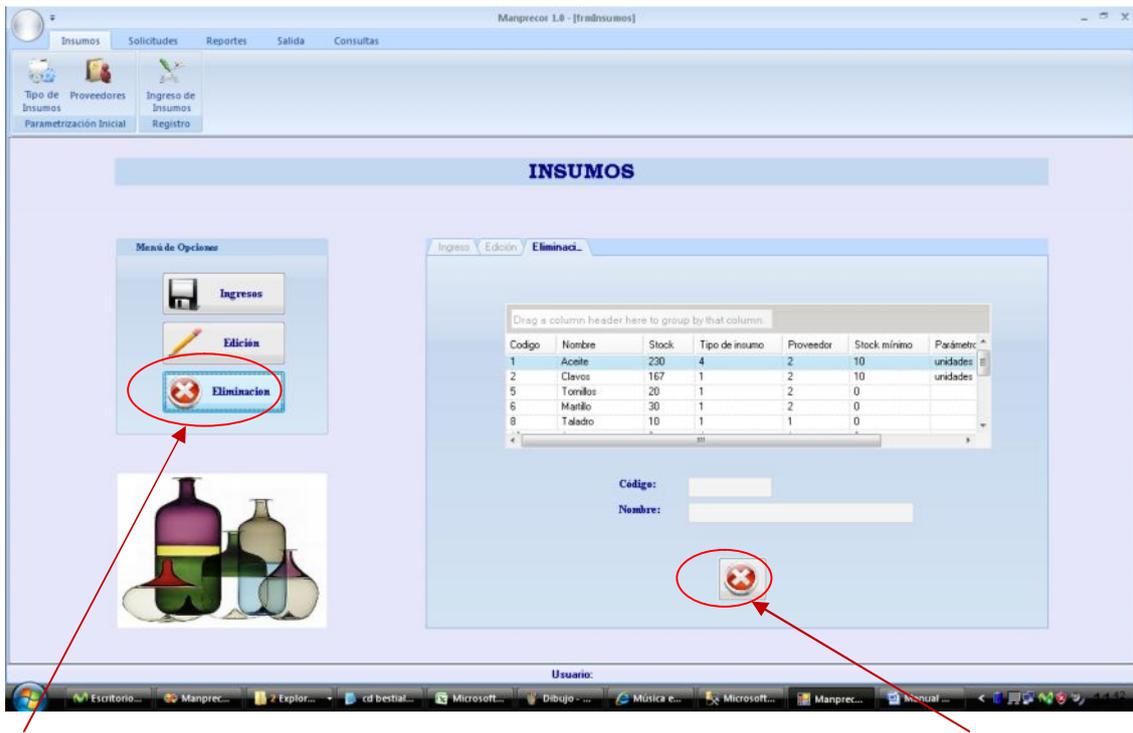


Fig#98.-Edición Insumos

Fuente: Los Autores

Eliminación De Insumos.-

Para la eliminación de insumos dar un clic sobre el botón  (1) y escoger en la tabla de insumos el insumo que se desea eliminar del sistema, una vez realizado esto presionar el botón  (2) y eliminar la información.



1 2

Fig#99.-Eliminación Insumos

Fuente: Los Autores

Solicitud De Insumo.-

En esta pantalla el usuario podrá realizar una solicitud de insumos si así fuese necesario, esta funcionalidad la encontramos presionando el botón  (1) para la realización del proceso el usuario debe tener el perfil indicado y deberá ingresar la información necesaria en la tabla de información de la parte derecha tal como el insumo solicitado y la cantidad que se debe adquirir, para registrarlo en la tabla presionar el botón  (2) y para generar la solicitud como tal dar clic en el botón  (3) ,cabe indicar que si se va a solicitar un insumo que no esté ingresado en el sistema se selecciona en la lista de insumos la opción “nuevo insumo” con esto se registra en el sistema y se realiza la solicitud al mismo tiempo.

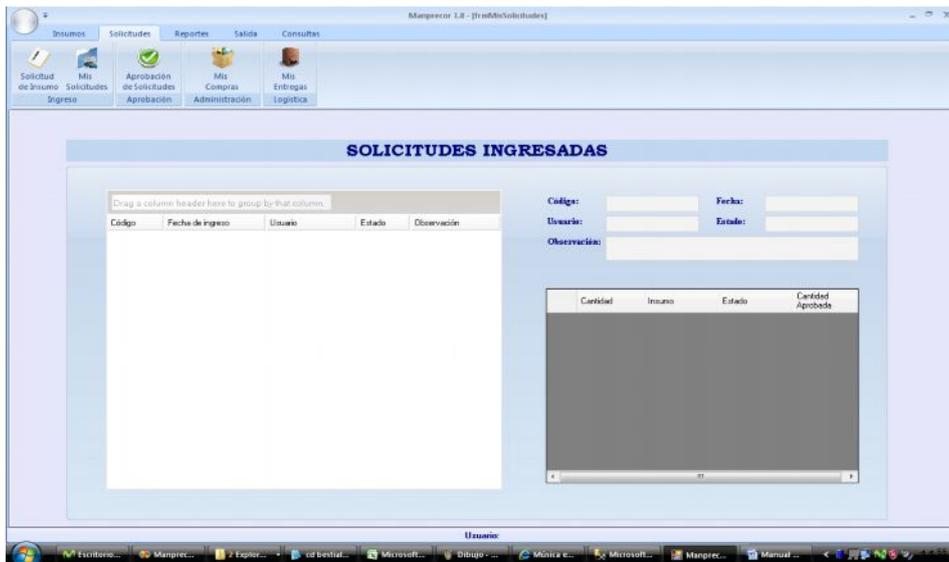


Fig#100.-Solicitud de insumo

Fuente: Los Autores

Solicitudes Ingresadas.-

En la pantalla de solicitudes ingresadas el usuario podrá tener el control y el avance de las solicitudes que tiene registradas y saber en qué estado se encuentran.

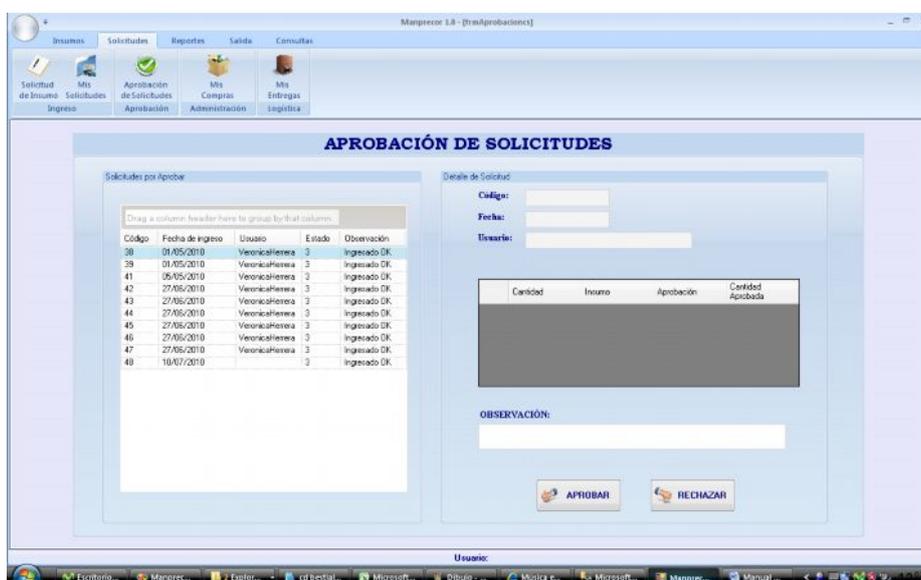


Fig#101.-Solicitud ingresada

Fuente: Los Autores

Aprobación De Solicitudes.-

Una vez ingresada una solicitud de insumos esta pasa por un filtro de aprobación que lo realiza una persona en la empresa con perfil de aprobación de solicitudes. En esta pantalla se cuenta con una tabla cargada con todas las solicitudes que se encuentran pendientes por aprobar, ahora bien como una solicitud tiene varios artículos dentro de su constitución MANPRECOR da la posibilidad de aprobar la solicitud total, parcialmente o rechazarla por completo. Con doble clic en la tabla de información escoger la solicitud en la cual se va a trabajar. En la grilla de la parte derecha se carga la solicitud con todos sus artículos y el usuario va aprobando o rechazándolos según sea su criterio.

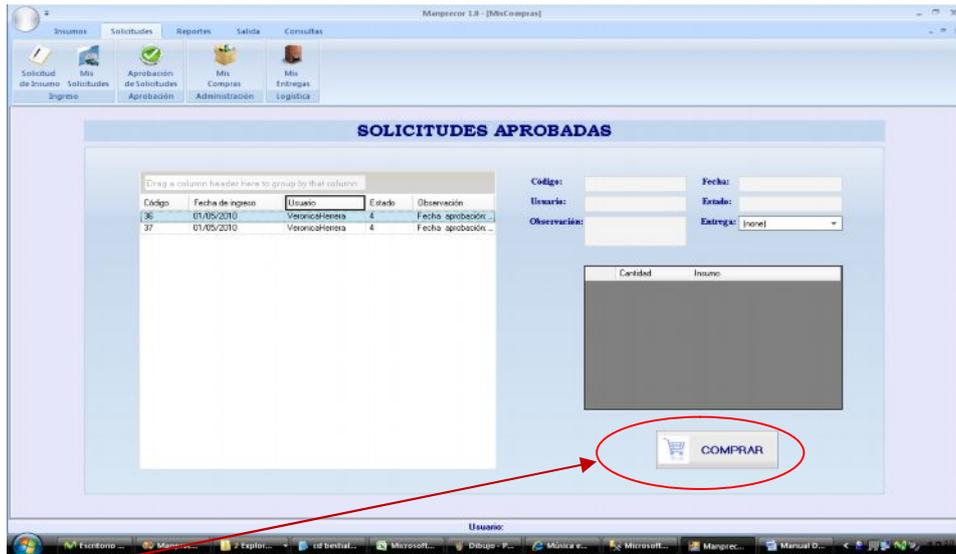


Fig#102.-Aprobación

Fuente: Los Autores

Compras.-

El usuario puede controlar que las solicitudes ya aprobadas pasen por el proceso de compras. En la tabla se tiene las solicitudes aprobadas y con doble clic escoger la que se desea comprar, ingresar una descripción y la fecha en la cual se quiere realizar la compra y dar clic en el botón  (1).



1

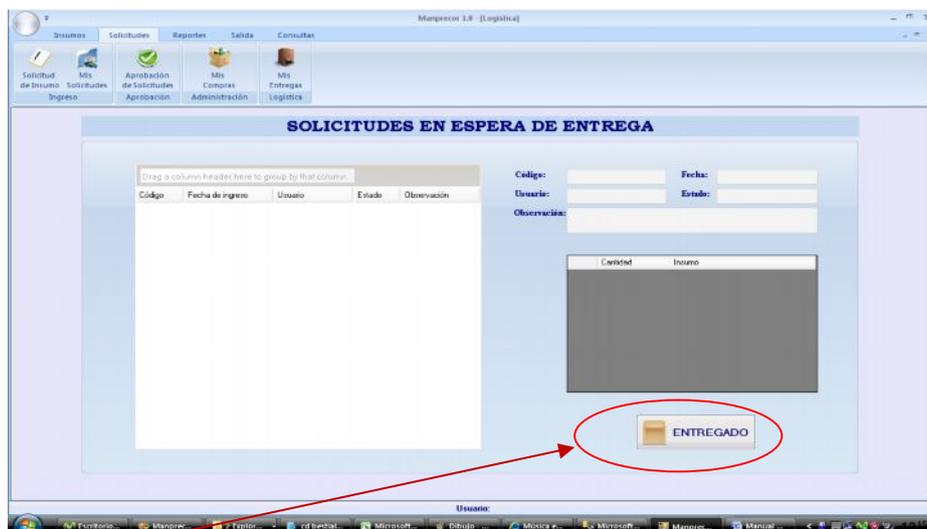
Fig#103.Compras
Fuente: Los Autores

Entrega De Insumos.-

Una vez realizada la compra de los insumos estos deben ser entregados a sus solicitantes para que hagan uso de ellos, en la pantalla se cargan las solicitudes que ya fueron compradas y están en espera de entrega, aquí se registra a la persona solicitante seleccionando la solicitud a entregar y presionando el botón



(1).



1

Fig#104.-Entrega
Fuente: Los Autores

CONSULTAS.-

La parte de consultas de MANPRECOR está estandarizada en una pantalla común que tiene una gran cantidad de prestaciones, las consultas están repartidas en los dos módulos y son muy amigables al usuario, una vez que se obtiene la información deseada en la tabla de información el usuario podrá exportar la información ya sea a Excel o imprimirla tal como se encuentra en la tabla utilizando los

botones   (1) , ahora bien si lo que desea es una información mejor presentada en la pantalla el usuario puede hacer uso del color de la tabla y de los filtros de información   (2) los cuales los puede activar o desactivar según su necesidad, también tiene la posibilidad de quitar o aumentar las columnas de la tabla según sea su gusto y necesidad.

NombreInsumo	Stock	TipoInsumo	Proveedor	StockMínimo	Cantidad	Estado
Aceite	230	Lubricantes	Inmasat	10	unidades	Activo
Clavos	167	Utileria	Inmasat	10	unidades	Activo
Tornillos	20	Utileria	Inmasat	0		Activo
Martillo	30	Utileria	Inmasat	0		Activo
Tachuela	0	Utileria	Bingstar	0		Eliminado
Taladro	10	Utileria	Bingstar	0		Activo
Grapas	0	Utileria	Proveedor3	0		Eliminado
Agua	3	Lubricantes	Bingstar	0		Activo
Mesa	0	Utileria	Bingstar	0		Activo
Playos	0	Utileria	Bingstar	0		Activo
Aceite2	0	Lubricantes	Inmasat	0		Eliminado

Fig#105.-Consultas
Fuente: Los Autores

PRESUPUESTO

<u>DESCRIPCION</u>	<u>VALOR</u>
Gastos Operativos (Internet, Libros, Aplicativos, Software)	\$ 600
Otros	\$ 100
Movilización	\$ 200
Licencias de programas a usar	\$800
Empastados	\$40
Dirección de tesis	\$300
Derecho de grado y especies	\$185
Delegado por concejo de carrera	\$28
Horas de trabajo	\$3650
TOTAL GASTOS	\$ 5903

