

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR  
FACULTAD DE INGENIERIA  
ESCUELA DE SISTEMAS**

**DISERTACION PREVIA A LA OBTENCION DEL TÍTULO DE  
INGENIERO EN SISTEMAS**

**“DISEÑO, DESARROLLO E IMPLEMENTACION DE UN CATALOGO VIRTUAL  
DE SOPORTE TECNICO PARA EL DIAGNOSTICO Y RECTIFICACIÓN DE  
MOTORES PARA LA EMPRESA RECONSTRUCTURA MOTORISTA  
ECUATORIANA CIA. LTDA”**

**NOMBRE  
DANIELA CARRILLO PEÑAFIEL**

**DIRECTOR: ING.OSWALDO ESPINOZA**

**QUITO, 2010**

## AGRADECIMIENTOS

*Agradezco a Dios, a mis padres, hermanos y amigos por su apoyo incondicional, a José por su compañía y aliento en cada momento.*

## DEDICATORIA

*A todas aquellas  
personas que creyeron y  
confiaron que este desarrollo  
se lograría.*

La Empresa RME (RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA), con el objetivo de evolucionar y mejorar su calidad en el servicio que presta en la rectificación de motores , tuvo la iniciativa de crear un catalogo virtual el cual contenga toda la información necesaria relacionada con el motor esta es combustible, marca, modelo y las piezas más importantes del motor , necesarias para una rectificación , los cálculos de rectificación para las piezas Biela, Bancada, Cigüeñal son de mucha importancia para la empresa debido que estos cálculos deben ser muy precisos para evitar posibles fallos.

El catálogo virtual dentro de su Menú Principal consta de; Administración permite ingresar, modificar y visualizar a los Usuarios que tienen acceso al sistema, Admin. Motor permite la administración (agregar, actualizar, eliminar) de Combustible, Marca, Modelo y Motor, Consulta Motor permite hacer la búsqueda rápida de un motor para posteriormente ingresar los valores de las piezas para que el sistema realice los cálculos necesarios, finalmente se detalla los resultando ingresados y los recomendados para rectificar.

El catálogo virtual se desarrolló con herramientas de programación libres, estas son PHP como lenguaje de programación WEB, MySQL como motor de base de datos, además incluidas técnicas como AJAX, y como parte del diseño la inclusión de imágenes en 3D, se aplicó la metodología de Programación Extrema que utiliza una realimentación rápida y un alto ancho de banda en la comunicaciones interpersonales para maximizar el valor que se entrega al cliente.

1	INTRODUCCION .....	1
1.1	Introducción.....	3
1.2	Objetivos .....	4
1.2.1	Objetivo General.....	4
1.2.2	Objetivos Específicos .....	4
1.3	Historia de la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA.....	5
1.4	Evolución de la Empresa.....	7
1.4.1	Nacimiento.....	7
1.4.2	Fortalecimiento .....	7
1.4.3	Problemas Presentados en la Empresa .....	9
1.4.4	Misión, Visión y Políticas de Calidad .....	9
1.4.5	Proyecciones Futuras de la Empresa.....	10
2	METODOLOGÍA Y HERRAMIENTAS A UTILIZAR .....	11
2.1	METODOLOGÍA .....	11
2.1.1	Introducción Metodología .....	11
2.1.2	Clases de metodologías .....	12
2.1.3	Programación Extrema .....	15
2.1.4	Etapas de Programación Extrema .....	19
2.2	HERRAMIENTAS A UTILIZAR .....	26
2.2.1	Diseño .....	26
2.2.2	Desarrollo .....	27
2.2.3	Bases de datos .....	33
2.2.4	Tipos de compilación del servidor .....	35
3	FASE DE ANÁLISIS, DISEÑO Y CONSTRUCCIÓN .....	36
3.1	Análisis.....	36
3.1.1	Análisis de factibilidad .....	36
3.1.2	Análisis de requerimientos.....	42
3.1.3	Etapas del Sistema.....	43
4	IMPLANTACIÓN .....	47

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

---

4.1	Implantación.....	47
4.2	Análisis de la aplicación.....	47
4.2.1	Diseño de la interfaz.....	47
4.3	Garantía y mantenimiento del producto.....	48
4.4	Capacitación al usuario final de la aplicación.....	49
5	CONCLUSIONES Y RECOMENDACIONES.....	50
5.1	Conclusiones.....	50
5.2	Recomendaciones.....	51
6	Bibliografía.....	52
7	Anexos.....	53

# 1 INTRODUCCION

## 1.1 Introducción

El avance tecnológico ha ido evolucionando a pasos gigantescos, con la llegada de las computadoras los procesos que antes lo hacíamos a mano ahora son desarrollados con mayor facilidad y con el mínimo de tiempo requerido.

Debido al actual consumo masivo de tecnología y la gran demanda la empresa **RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA (RME)**, creada en 1951 para prestar servicio de rectificación de motores y sus partes; se ha visto en la obligación de implementar un catalogo virtual para acceder con mayor facilidad los datos requeridos por los empleados en la parte operativa donde su desarrollo sea mas rápido y practico.

La empresa **RME** ha ganado espacio con su gran cantidad de clientes a quienes quieren ofrecer un mejor servicio por lo cual, pretenden ampliar su asistencia mediante un catalogo virtual que cumpla con las expectativas de la empresa, además para que los operarios puedan acceder y obtener ayuda en sus cálculos, así también implementar el resultado final en la hoja de control de calidad para poder tener un mayor servicio.

Por ser **RME** una empresa prestadora de servicios donde los cálculos de la rectificadora deben ser precisos y dada la necesidad de manejar gran cantidad de datos que en la práctica difícilmente son posibles de manejar por un ser humano, se considera la necesidad de contar con un catalogo virtual.

## 1.2 Objetivos

### 1.2.1 Objetivo General

Diseñar e implementar mediante la aplicación de la metodología de programación extrema un catalogo virtual de servicio técnico para la empresa **RME** que satisfaga las necesidades de los clientes internos en cuanto al manejo de información mediante una interacción eficaz de los empleados con el catálogo.

### 1.2.2 Objetivos Específicos

- Identificar y determinar los requerimientos de los usuarios.
- Proponer alternativas de accesibilidad el catálogo para los usuarios del sistema.
- Diseñar e implementar el catálogo con imágenes en tres dimensiones para facilitar el uso del mismo.
- Automatizar el procedimiento de verificación para el control de calidad del producto antes de la entrega final al cliente.
- Diseñar e implementar el ingreso de motores nuevos en el catálogo para la utilización de la empresa.
- Agilizar, mantener, optimizar el cálculo para el proceso de diagnostico para la ejecución de los servicios de rectificación.
- Mantener, agilizar, optimizar la elaboración de la hoja de control de calidad



### **1.3 Historia de la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

La empresa RME presta sus servicios al país con una vasta experiencia de alrededor de 51 años, tiempo en el cual se ha preocupado de cumplir con las exigencias requeridas por el cliente mediante un trabajo de excelente calidad y profesionalismo en el tiempo estipulado.

Su fundación la realizó el Sr. Carlos Bonilla Campos y tuvo lugar en la ciudad de Quito en el año de 1959 en su antiguo local ubicado en la Av. Maldonado 1017, en el barrio la Villaflora, en este sitio la empresa prestó su servicio alrededor de 39 años. RME empezó con una rectificadora de cigüeñales y los accesorios necesarios para utilizarla; el local contaba con muy poco espacio, además la empresa contaba con el servicio de 3 aprendices.

Debido al desarrollo en el que se ubico la empresa se procedió a buscar un local adecuado a las nuevas necesidades de la misma con el fin de ampliar sus servicios, maquinarias y equipos, y por supuesto mejorarlos con el objetivo de satisfacer los requerimientos actuales del cliente. El nuevo local de la empresa está ubicada en la calle Crnl. Lauro Guerrero S10-292 y Av. Alonso de Angulo esquina, dirección en la que empezó a prestar sus servicios desde junio de 1994 la figura 1.1 muestran los dos frentes de la empresa en las calles antes mencionadas.



**FIGURA 1.1 RME (ACCESO PRINCIPAL)<sup>1</sup>**

Junto con el cambio de dirección de la empresa, se adquirió un número de maquinas herramientas entre las cuales se encuentra una rectificadora de ciguenales, una madrinadora de cilindros, una rectificadora de valvulas, dos tornos universales, etc; para un total de veinticinco maquinas que ayudarian en el desempeno de todos los servicios de RME. Para guiarse de mejor manera en el ambito de la calidad que la empresa cuenta en la actualidad con un total de 16 empleados comprometidos con su trabajo y con los objetivos y metas de la empresa.

---

<sup>1</sup> Imagen proporcionada por Christian Bonilla Presidente de RME

## 1.4 Evolución de la Empresa

### 1.4.1 Nacimiento

Su fundación la realizó el Sr. Carlos Bonilla Campos y tuvo lugar en la ciudad de Quito en el año de 1959 en su antiguo local ubicado en la Av. Maldonado 1017, en el barrio la Villaflora, en este sitio la empresa prestó su servicio alrededor de 39 años.

### 1.4.2 Fortalecimiento

La empresa RME pone a total disposición del cliente los servicios de reconstrucción y rectificación de motores mediante la reparación de sus partes, en este servicio se toma en cuenta la parte a ser reparada, y en la cual se divide en varios procesos que se detallan en la tabla 1.1

PARTE	SERVICIOS
<b>Motor Completo</b>	<ul style="list-style-type: none"><li>• Lavado</li><li>• Armado</li></ul>
<b>Cabezote (Culata)</b>	<ul style="list-style-type: none"><li>• Rectificación de Asientos</li><li>• Rectificación de Válvulas</li><li>• Cambio de Asientos</li><li>• Apertura de Cajeras</li><li>• Rectificación de alojamiento de Árbol</li><li>• Cepillado</li><li>• Perforación para Guías</li></ul>
<b>Bloque de Motor</b>	<ul style="list-style-type: none"><li>• Mandrinado de Cilindros</li></ul>

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

<b>(Block)</b>	<ul style="list-style-type: none"> <li>• Bruñido</li> <li>• Encamisado</li> <li>• Cepillado</li> <li>• Mandrinado de Cepos</li> <li>• Regulación de Bancada y Bielas</li> <li>• Rellenado</li> </ul>
<b>Cigüeñal</b>	<ul style="list-style-type: none"> <li>• Rectificación</li> <li>• Pulido</li> <li>• Enderezado / relleno</li> <li>• Reparación de Porta retenedores</li> </ul>
<b>Bielas</b>	<ul style="list-style-type: none"> <li>• Alineación y Chequeo</li> <li>• Rectificación</li> <li>• Cambio de bocines y Rectificado</li> <li>• Construcción de Bocines</li> </ul>
<b>Trabajos de Tornos</b>	<ul style="list-style-type: none"> <li>• Adaptación de Válvulas</li> <li>• Adaptación de Guías</li> <li>• Construcción de Asientos</li> </ul>
<b>Trabajos de Prensa</b>	<ul style="list-style-type: none"> <li>• Armado de Pistones</li> </ul>
<b>Repuestos</b>	<ul style="list-style-type: none"> <li>• Venta de Repuestos</li> </ul>

TABLA 1.1 SERVICIOS QUE OFRECE RME<sup>2</sup>

### **1.4.3 Problemas Presentados en la Empresa**

En RME se ha visto en la obligación de evolucionar debido a las exigencias actuales, y de tener la información de todos los motores existentes en el mercado ecuatoriano digitalizado; por haber pasado durante todos estos años bajo el dominio de los libros y la realización de los cálculos a mano, pero esto en los últimos años se ha dificultado por el incremento en el mercado de nuevos modelos de motores , lo cual afecta e impacta en el tiempo de la entrega del motor, esto se lograría reducir con la ayuda del catálogo virtual; los operarios no tendrían que dirigirse a los libros y hacer los cálculos para la rectificación y la respectiva transformación de medidas en el resultado.

Otro de los problemas existente en la empresa RME es que la hoja de calidad que manejan desean que sea digitalizada para no tener errores al momento de la entrega.

### **1.4.4 Misión, Visión y Políticas de Calidad**

#### **1.4.4.1 MISION<sup>3</sup>**

Prestar el servicio de reconstruccion de motores y rectificacion de precision con un recurso humano competente y alta tecnologia , cumpliendo estandares internacionales de calidad para garantizar la satisfaccion total de nuestros clientes y superar sus expectativas.

---

<sup>2</sup> Tabla proporcionada por Christian Bonilla Presidente de RME

<sup>3</sup> Información proporcionada por Christian Bonilla Presidente de RME

#### *1.4.4.2 VISION*

Consolidar a RME como líder en reconstrucción de motores y rectificación de precisión en la zona sur de la ciudad de Quito.

#### *1.4.4.3 POLITICA DE CALIDAD*

Rectificar y reconstuir motores de combustion cumpliendo los requisitos internacionales de la norma ISO 9001:2000 ademas de los legales y reglamentarios. Mejorando continuamente el Sistema de Gestión de Calidad a través del cumplimiento y seguimiento de indicadores y metas; con su respectiva comunicación y actualización dentro de la organización.

#### **1.4.5 Proyecciones Futuras de la Empresa**

La proyección futura de la empresa RME es que con este catálogo virtual pueda tener una base de datos lo suficientemente completa y robusta con la mayoría de motores que el mercado ecuatoriano va obteniendo, además que en el futuro pueda acceder a la certificación de calidad ISO 9000-2001, así como ofrecer un servicio de calidad y ser una de las pioneras brindando este servicio.

## 2 METODOLOGÍA Y HERRAMIENTAS A UTILIZAR

### 2.1 METODOLOGÍA

#### 2.1.1 Introducción Metodología

Las metodologías<sup>4</sup> de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. Es un marco de trabajo conceptual de la ingeniería de software que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto.

Hasta hace poco el proceso de desarrollo llevaba asociada un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada; este esquema tradicional para abordar el desarrollo de software ha demostrado ser efectivo y necesario en proyectos de gran tamaño, donde por lo general se exige un alto grado de ceremonia en el proceso. Una dificultad prácticamente imposible de evitar es la de atender los cambios en los requisitos de clientes y usuarios o en la tecnología a usar, o en la disponibilidad de personal, conocimiento, experiencia y talento en la organización. Se infiere que para lograr desarrollos de proyectos con buen desempeño y calidad, es deseable un estilo de administración de proyectos iterativo e incremental, con suficientes puntos de verificación como para poder tomar en cuenta dichas necesidades de cambios sin desviaciones mayores a planes realizados con sumo detalle, cuidado y dificultad.

Sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad.

---

<sup>4</sup> [http://es.wikipedia.org/wiki/Metodolog%C3%ADa\\_de\\_desarrollo\\_de\\_software](http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software)

## 2.1.2 Clases de metodologías

### 2.1.2.1 Metodologías Tradicionales

Hasta hace poco el proceso de desarrollo llevaba asociada un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada. Este esquema tradicional para abordar el desarrollo de software ha demostrado ser efectivo y necesario en proyectos de gran tamaño, donde por lo general se exige un alto grado de ceremonia en el proceso.

El método de investigación, se basa en los pasos que, se deben seguir en toda investigación científica, aunque estos pasos están basados en el método hipotético-deductivo, por su generalidad, son aplicables, a cualquier tipo de investigación.

Entre las principales metodologías tradicionales tenemos los ya tan conocidos RUP (*Rational Unified Process*) y MSF (*Microsoft Solution Framework*) entre otros, que centran su atención en llevar una documentación exhaustiva de todo el proyecto y centran su atención en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto. Otra de las características importantes dentro de este enfoque, son los altos costos al implementar un cambio y la falta de flexibilidad en proyectos donde el entorno es volátil.

A continuación se detalla RUP y MSF uno de los métodos más usados dentro de los métodos tradicionales.

#### 2.1.2.1.1 Rational Unified Process (RUP)

RUP<sup>5</sup> es un proceso formal, provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos

---

<sup>5</sup> Proceso Unificado de Rational , proviene del trabajo de UML . Ingeniería de Software ; Ian Sommerville -7 Edicion; pag 76



## Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA

de los usuarios finales fue desarrollado por Rational Software, y está integrado con toda la suite Rational de herramientas. Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de notación.

### FASES DEL RUP

Las cuatro fases del ciclo de vida son:

- Concepción
- Elaboración
- Construcción
- Transición

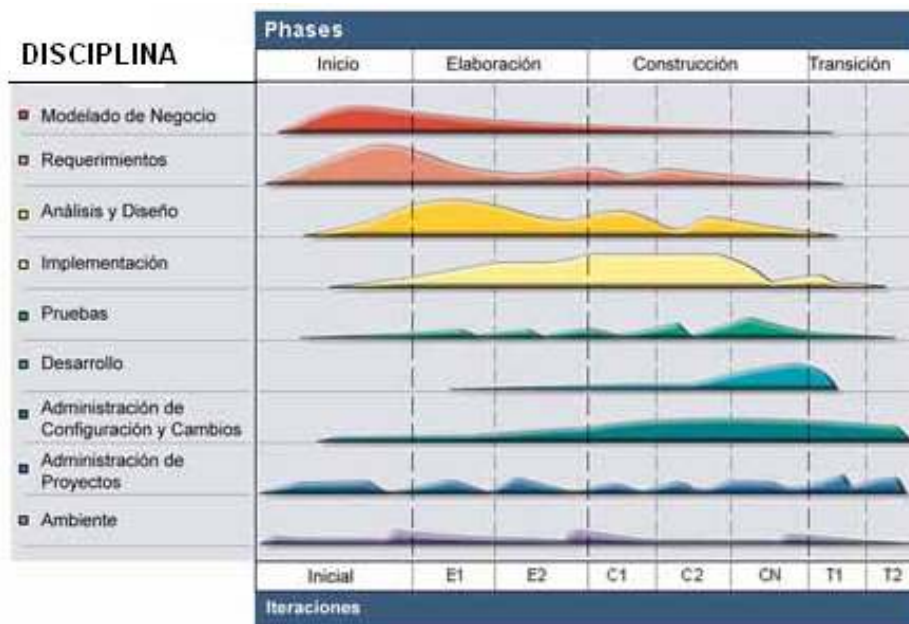


FIGURA 2.1 Fases del RUP<sup>6</sup>

6

[http://www.iteraproces.com/index.php?option=com\\_content&task=view&id=18&Itemid=42&limit=1&limitstart=1](http://www.iteraproces.com/index.php?option=com_content&task=view&id=18&Itemid=42&limit=1&limitstart=1)

### 2.1.2.1.2 Microsoft Solution Framework (MSF)

MSF<sup>7</sup> es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información.

Al usar MSF todo proyecto es separado en cinco principales fases:

- Visión y Alcances.
- Planificación.
- Desarrollo.
- Estabilización.
- Implantación.



FIGURA 2.2 Metodología de MSF<sup>8</sup>

Actualmente es imprescindible considerar los riesgos, aunque habitualmente las empresas, no han sido concienciadas de los riesgos inherentes al procesamiento de la información mediante ordenadores, a lo que han contribuido, a veces, los propios

---

<sup>7</sup> <http://www.netways.com/Methodology/Pages/MS%20Solutions%20Framework.aspx>

<sup>8</sup> <http://technet.microsoft.com/en-us/library/bb497060.aspx>

responsables de informática, que no han sabido explicar con la suficiente claridad las consecuencias de una política de seguridad insuficiente o incluso inexistente. Por otro lado, debido a una cierta deformación profesional en la aplicación de los criterios de coste/beneficio, el directivo desconocedor de la informática no acostumbra a autorizar inversiones que no lleven implícito un beneficio demostrable, tangible y mensurable.

### **2.1.3 Programación Extrema**

La programación extrema<sup>9</sup> es una metodología ágil de desarrollo de software, es un nuevo enfoque liviano al tema del desarrollo de software, usa una realimentación rápida y un alto<sup>10</sup> ancho de banda en la comunicaciones interpersonales para maximizar el valor que se entrega al cliente, mediante la presencia del cliente en el lugar de trabajo de los desarrolladores, un enfoque particular del planeamiento y las comprobaciones constantes de software.

La programación extrema se basa en una serie de reglas y principios que se han ido gestando a lo largo de toda la historia de la ingeniería del software. Usadas conjuntamente proporcionan una nueva metodología de desarrollo software que se puede englobar dentro de las metodologías ligeras, que son aquellas en la que se da prioridad a las tareas que dan resultados directos y que reducen la burocracia que hay alrededor tanto como sea posible.

Los valores de la programación extrema son:

- Comunicación:** Crear software requiere de sistemas comunicados.
- Simplicidad:** Empezar con lo necesario y requerido y trabajar desde ahí.
- Retroalimentación:** Del sistema, del cliente, y del equipo.
- Valentía:** Programa para hoy y no para mañana.
- Respeto:** El equipo debe trabajar como uno, sin hacer decisiones repentinas.

---

<sup>9</sup> Programación Extrema el método ágil más conocido. Ingeniería de Software; Ian Sommerville – 7 Edición ; pag 364

Las características fundamentales del método son:

- Desarrollo iterativo e incremental: se lleva a cabo a través de entregas pequeñas y frecuentes del sistema y por medio de un enfoque para la descripción de requerimientos basado en las historias del cliente o escenarios.
- Pruebas unitarias: frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación. Los programadores escriben pruebas para chequear el correcto funcionamiento del programa, los clientes realizan pruebas funcionales. Las pruebas del sistema en la que se ha desarrollado en un enfoque que reduce la probabilidad de producir nuevos incrementos del sistema que introduzca errores en el software existente.
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Las parejas de programadores se intercambian con frecuencia, de forma que todos acaban trabajando con todos.

El trabajo por parejas haciendo intercambios tiene las siguientes ventajas:

- Más Disciplina. Emparejando correctamente es más probable que hagan "lo que se debe hacer" en lugar de tomar largos descansos.
- Mejor código. Emparejando similares es menos probable producir malos diseños ya que su inmersión tiende a diseñar con mayor calidad.
- Flujo de trabajo constante. El emparejamiento produce un flujo de trabajo distinto al trabajar solo. En pareja el flujo de trabajo se recupera más rápidamente: un programador pregunta al otro "*¿por dónde quedamos?*". Las parejas son más resistentes a las interrupciones ya que un desarrollador se ocupa de la interrupción mientras el otro continúa trabajando.
- Múltiples desarrolladores contribuyen al diseño. Si las parejas rotan con frecuencia en el proyecto significa que más personas están involucradas con

una característica en particular. Esto ayuda a crear mejores soluciones, especialmente cuando una pareja no puede resolver un problema difícil.

- Moral mejorada. La programación en parejas es más agradable para algunos programadores, que programar solos.
- Propiedad colectiva del código. Cuando el proyecto se hace en parejas, y las parejas se rotan con frecuencia, todos tienen un conocimiento del código base.
- Enseñanza. Todos, hasta los novatos, poseen conocimientos que los otros no. La programación en pareja es una forma amena de compartir conocimientos.
- Cohesión de equipo. La gente se familiariza más rápidamente cuando programa en pareja. La programación en pareja puede animar el sentimiento de equipo.
- Pocas interrupciones. La gente es más renuente a interrumpir a una pareja que a una persona que trabaja sola.
- Menos estaciones de trabajo. Ya que dos personas van a trabajar en una estación de trabajo, se requieren menos estaciones de trabajo, y las estaciones extras pueden ser ocupadas para otros propósitos.
- Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.

El cliente junto al equipo de desarrollo definen qué es lo que se quiere hacer, para ello utilizan los históricos de usuario. Una historia de usuario en un texto de una o dos frases en las que se dice algo que debe hacer el sistema. Es más extensa que un requisito, se evalúa para cada historia de usuario el tiempo que puede llevar. Un programador puede estimar con cierta fiabilidad un trabajo que le lleve unos días, pero la estimación es menos fiable si es de un plazo superior a una semana. Si es más largo, hay que partir la

historia en otras más pequeñas, luego se ordenan en el orden en que se van a desarrollar y se establecen las mini-versiones, de forma que cada mini-versión implementa varias de las historias de usuario.

- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes, cada “entrega” o “hito” no debe durar más de una semana. Cuanto más cortos mucho mejor, pues se genera menos ruido o caos en el código general
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo; es importante refactorizar el código para aumentar la escalabilidad y el mantenimiento. Lo ideal es que cada iteración de la aplicación sufra una refactorización total, por supuesto sin modificar las funcionalidades
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La tendencia de entregar software en lapsos cada vez menores de tiempo y con exigencias de costos reducidos y altos estándares de calidad, hace que XP sea una opción a considerar. En términos generales XP parece ser una metodología adecuada para proyectos medianos y pequeños, donde los equipos de desarrollo no

pasan de 10 programadores y donde la constante es que los requerimientos cambien, a veces radicalmente, durante la etapa de desarrollo.

### 2.1.4 Etapas de Programación Extrema

La programación extrema está muy ligada o es parecida a los métodos evolutivos. Las prácticas que componen la programación extrema se pueden agrupar en cuatro grandes bloques: que conforman el ciclo de vida en la programación extrema; plan, diseño, codificación y pruebas, sin embargo, estos bloques no deben realizarse en orden, si no que cada uno consta de una serie de actividades, y todas ellas se irán realizando de manera evolutiva.

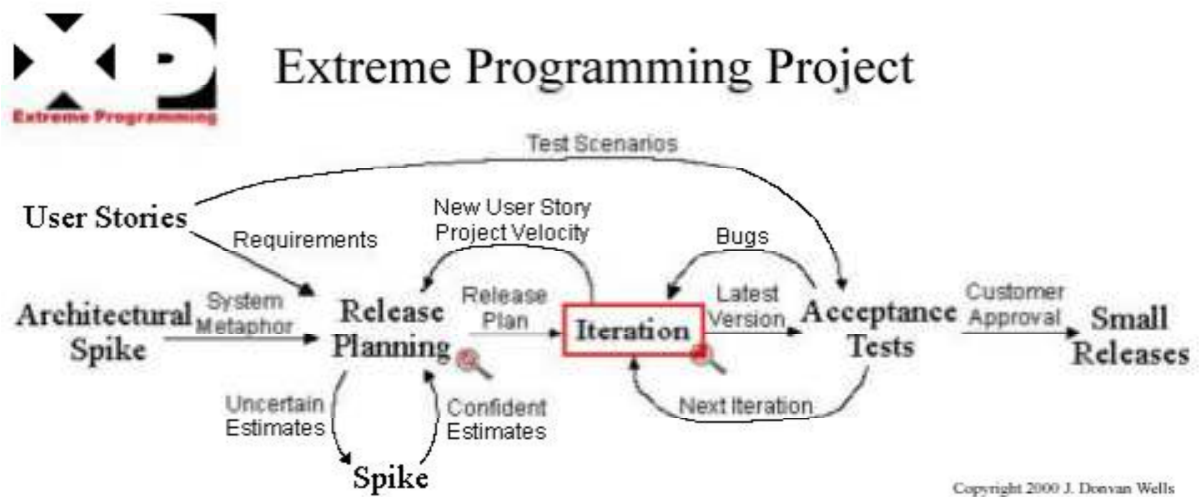


FIGURA 2.3 ETAPAS DE LA PROGRAMACION EXTREMA<sup>11</sup>

#### 2.1.4.1 Ciclo de vida

##### 2.1.4.1.1 Planificación

Planificación de entregas se realiza en base a reglas para que tanto el equipo de gestión, el equipo de desarrollo y el cliente se sientan parte de la decisión.

<sup>11</sup> Imagen extraída de [www.xprogramming.com](http://www.xprogramming.com)

Se crea un plan de lanzamiento, que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto. Se usará como base las historias de usuarios, participando el cliente en la elección de los que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.

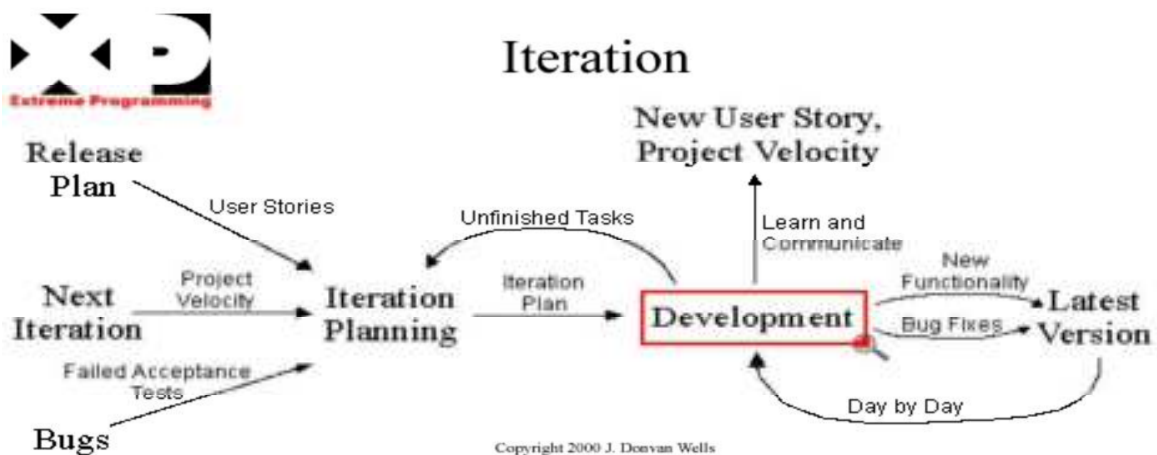


FIGURA 2.4 CICLO DE VIDA (PLANIFICACION)<sup>12</sup>

La planificación debe seguir unas ciertas premisas, lo primordial es que las entregas se hagan cuanto antes y que con cada iteración el cliente reciba una nueva versión. Cuanto más tiempo se tarde en introducir una parte esencial, menos tiempo habrá para trabajar en ella posteriormente. Se aconsejan muchas entregas y muy frecuentes. De esta forma, un error en una parte esencial del sistema se encontrará pronto y, por tanto, se podrá arreglar antes; sin embargo; la retroalimentación se puede aplicar a diferentes escalas de tiempo, cuando se requiere tener una información constante del estado del sistema, tanto de las pruebas del programador conforme implementa unidades de código, como del equipo de programación hacia el cliente. El cliente se encuentra mucho más cerca del proceso de desarrollo se

<sup>12</sup> Imagen extraída de [www.xprogramming.com](http://www.xprogramming.com)



elimina la fase inicial de recopilación de requerimientos, y se permite que éstos se vayan cogiendo a lo largo del proyecto, de una manera ordenada. De esta forma se posibilita que el cliente pueda ir cambiando de opinión sobre la marcha, pero a cambio han de estar siempre disponibles para solucionar las dudas del equipo de desarrollo. A una escala mayor de retroalimentación, de semanas o meses, se fomentan las revisiones del cliente hacia el avance general del proyecto, comparando las entregas parciales contra el plan general.

- Redactar historias de usuario: Las historias de usuario tienen el mismo propósito que los casos de uso, pero no son lo mismo. Las escriben los propios clientes, tal y como ven ellos las necesidades del sistema. Por tanto serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica.
- Crear plan de entregas: El plan de entregas se usará para crear los planes de iteración para cada iteración. Se usará como base los *user stories*<sup>313</sup>, participando el cliente en la elección de los que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.
- Controlar la velocidad del proyecto: La velocidad de proyecto se usa para determinar cuántas historias de usuario pueden ser implementadas antes de una fecha dada o cuánto tiempo es necesario para llevar a cabo un conjunto de historias. Los desarrolladores deberán hacer una estimación de cuánto tiempo, idealmente, les llevará implementar cada historia de usuario. Las condiciones ideales son aquellas en las que se codifica la historia de usuario sin otras distracciones y sabiendo exactamente qué es lo que hay que implementar.
- Dividir un proyecto en iteraciones: Cada iteración corresponde a un periodo de tiempo de desarrollo del proyecto de entre una y tres semanas. De esta forma, un proyecto, se divide en una docena de iteraciones, más o menos. Al principio de

---

<sup>13</sup> User Stories: O comúnmente llamadas Historias de Usuario, los cuales implementan directamente una serie de tareas. Ingeniería de Software; Ian Sommerville-7 Edición ; pag 364

cada iteración se debería convocar una reunión para trazar el plan de iteración correspondiente.

- Planificar cada interacción antes de comenzarla: El plan de iteración consiste en seleccionar las historias de usuario que, según el plan de entregas, corresponderían a esta iteración. También se eligen qué pruebas de aceptación fallidas se corregirán.
- Rotar al personal: Las rotaciones evitarán los cuellos de botella. Si sólo una persona de nuestro equipo es capaz de trabajar en un área concreta, existirá un riesgo enorme si esa persona nos deja por cualquier circunstancia. De esta forma, las rotaciones permitirán que todo el mundo conozca cómo funciona el sistema en general y ayudarán a realizar un reparto más equitativo del trabajo.
- Reunión de seguimiento diaria: Se recomienda que estas sean frecuentes, de poca duración y a ser posible delante de la pantalla del ordenador. Según la metodología XP, se recomienda que sean diarias; recordemos que los usuarios se considerarán parte integrante del equipo de desarrollo del proyecto. La *reunión de seguimiento* de cada mañana debe usarse para sacar a la luz los problemas, las soluciones y centrar el objetivo del equipo.
- Corregir la propia metodología: Se debe mejorar el proceso cuando éste falle. Cuando se empiece con un proyecto, seguiremos la metodología XP, pero debemos cambiar aquello que no funcione. Además los cambios que se realicen deberán ser comunicados al resto del equipo, todo el mundo debe estar al corriente de los cambios.

#### *2.1.4.1.2 Diseño*

- Se eligen los diseños más simples que funcionen. Siempre costará menos tiempo de implementar un diseño sencillo que uno complejo. Por lo que, se trata siempre de realizar las cosas de la manera más sencilla posible. Si alguna parte de la implementación resulta especialmente compleja, se debe replantearla; así,

cualquier cambio y modificación será mucho más sencillo. En ocasiones, realizar un diseño sencillo puede resultar una tarea especialmente difícil.

- Se elige una metáfora del sistema para que el nombrado de clases, siga una misma línea, facilitando la reutilización y la comprensión del código. La tarea de elegir una metáfora para el sistema permitirá mantener la coherencia de nombres de todo aquello que se va a implementar. Se debe elegir un sistema de nombres que permita que cualquiera que lo vea adivine la relación entre el objeto y aquello que representa
- Se escriben tarjetas CRC (*class-responsabilities-collaboration*) de clase-responsabilidades-colaboración para cada objeto, que permiten abstraerse el pensamiento estructurado y que el equipo de desarrollo al completo participe en el diseño. El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar.

#### *2.1.4.1.3 Codificación*

El cliente está siempre disponible, a ser posible cara a cara. La idea es que forme parte del equipo de desarrollo, y esté presente en todas las fases de XP. La idea es usar el tiempo del cliente para estas tareas en vez de para que cree una detalladísima especificación de requisitos. Durante la reunión del plan de entregas, el usuario propondrá qué historia de usuario se incluye en cada plan. También se negociararán los plazos de entrega, el usuario o cliente tomará las decisiones que le afecten para alcanzar los objetivos de su negocio; es necesario que el cliente colabore en la realización de los test, estos test comprobarán que el sistema está listo para pasar a la fase de producción, el usuario comprobará los resultados obtenidos y tomará decisiones en cuanto a la utilización o no del sistema realizado. El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código, este debe ser desarrollado siguiendo los estándares para facilitar su lectura y modificación por cualquier miembro del equipo de desarrollo.

- Las pruebas unitarias se codifican antes que el código en sí, haciendo que la codificación de este último sea más rápida, y que cuando se afronte la misma se tenga más claro qué objetivos tiene que cumplir lo que se va a codificar.
- La programación del código se realizará en parejas, para aumentar la calidad del mismo. En cada momento, sólo habrá una pareja de programadores integrando código.
- Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo y permitiendo que todo el mundo trabaje con la última versión del desarrollo. De esta manera, se evitará pasar grandes periodos de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades habrán sido detectadas enseguida.
- Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentar la contribución de ideas por parte de todo el equipo de desarrollo
- Se deja la optimización para el final
- No se hacen horas extra de trabajo.

#### *2.1.4.1.4 Pruebas*

XP pone más énfasis en el proceso de pruebas que otros métodos ágiles, las pruebas del sistemas son fundamentales, en la que se han desarrollado un enfoque que reduce la probabilidad de producir nuevos incrementos del sistema que introduzca errores en el software existente.

- Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado. Las unidades de test serán incluidas junto con el código que verifican dentro del repositorio, el código no se considerará completo si este no consta de su unidad de test correspondiente. El código será implantado cuando supere sus correspondientes unidades de test.

- Cuando se encuentra un error de codificación o *bug*, se desarrollan pruebas para evitar volver a caer en el mismo. De esta manera, la localización del mismo será mucho más fácil por parte de los programadores. Este nuevo test será empleado para aislar el fallo y depurarlo.
- Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las *user stories* elegidas para la iteración, y son "pruebas de caja negra", en las que el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se pasa la prueba de aceptación, se considera que el correspondiente *user stories* se ha completado. El cliente o usuario especifica los aspectos a probar cuando una historia de usuario ha sido correctamente implementada. Una historia de usuario puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento. Una prueba de aceptación es como una caja negra. Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones a cerca de las mismas. Una historia de usuario no se considera completa hasta que no supera sus pruebas de aceptación, esto significa que debe desarrollarse un nuevo test de aceptación para cada iteración o se considerará que el equipo de desarrollo no realiza ningún progreso. Las pruebas de aceptación deberían automatizarse ya que se deben pasar frecuentemente.

A las pruebas de aceptación también se las conoce con el nombre de pruebas de funcionalidad, y constituyen la garantía de que los requerimientos fijados por los usuarios han sido reflejados en el sistema.

## 2.2 HERRAMIENTAS A UTILIZAR

### 2.2.1 Diseño

#### 2.2.1.1 Introducción a Imágenes en 3D

El grafico tridimensional deja atrás al 2D por su forma en que fue creado. El gráfico 3D se creó por medio de un proceso de cálculos matemáticos sobre entidades geométricas 3D originadas en un ordenador, donde tiene como función principal conseguir una proyección visual en dos dimensiones para ser mostradas en una pantalla.

Una animación 3D se caracteriza por poseer las coordenadas:

x: representa la anchura del objeto.

y: representa la altura del objeto.

z: representa la profundidad del objeto.

La diferencia de una animación 3D de una 2D es fundamentalmente la forma en que ha sido generada esta animación. La animación 3D<sup>14</sup> se basa sobre gráficos generados mediante cálculos matemáticos tridimensionales en una computadora, el objetivo es obtener una proyección visual en tres dimensiones.

El proceso de creación de gráficos 3D por computadora puede ser dividido en estas tres fases básicas:

- Modelado
- Composición de la escena
- Rénder (creación de la imagen final)

#### 2.2.1.1.1 Modelado

La etapa de modelado consta de ir dando forma a objetos individuales que luego serán usados en la escena. Existen diversas técnicas de modelado;

---

<sup>14</sup> Animacion 3D;Peter Ratner;Ediciones Anaya Multimedia 2005.Modelado basico 3D; pag 57

*Constructive Solid Geometry*, modelado con NURBS y modelado poligonal son algunos ejemplos. Los procesos de modelado puede incluir la edición de la superficie del objeto o las propiedades del material (por ejemplo, color, luminosidad, difusión, especularidad, características de reflexión, transparencia u opacidad, o el índice de refracción), agregar texturas, mapas de relieve (*bump-maps*) y otras características.

El proceso de modelado puede incluir algunas actividades relacionadas con la preparación del modelo 3D para su posterior animación. A los objetos se les puede asignar un esqueleto, una estructura central con la capacidad de afectar la forma y movimientos de ese objeto. Esto ayuda al proceso de animación, en el cual el movimiento del esqueleto automáticamente afectara las porciones correspondientes del modelo.

#### *2.2.1.1.2 Composición de la escena*

Esta etapa involucra la distribución de objetos, luces, cámaras y otras entidades en una escena que será utilizada para producir una imagen estática o una animación. Si se utiliza para Animación, esta fase, en general, hace uso de una técnica llamada "*Keyframing*", que facilita la creación de movimientos complicados en la escena. Con la ayuda de la técnica de *keyframing*, en lugar de tener que corregir la posición de un objeto, su rotación o tamaño en cada cuadro de la animación, solo se necesita marcar algunos cuadros clave (*keyframes*). Los cuadros entre *keyframes* son generados automáticamente, lo que se conoce como Interpolación.

### **2.2.2 Desarrollo**

#### *2.2.2.1 Lenguaje de Programación Web*

Un lenguaje de programación<sup>15</sup> es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo, permite a uno o más programadores especificar de

---

<sup>15</sup> <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>

*manera precisa*: sobre qué datos una computadora debe operar, cómo deben ser estos almacenados, transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar *relativamente* próximo al lenguaje humano o natural, tal como sucede con el lenguaje Léxico. Una característica relevante de los lenguajes de programación es precisamente que más de un programador puedan tener un conjunto común de instrucciones que puedan ser comprendidas entre ellos para realizar la construcción del programa de forma colaborativa.

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que paso el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la web dinámica, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos.

#### 2.2.2.1.1 PHP

PHP<sup>16</sup> fue originalmente diseñado en Perl, seguidos por la escritura de un grupo de CGI binarios escritos en el lenguaje C por el programador danés-canadiense Rasmus Lerdorf en el año 1994 . El 8 de junio de 1995 fue publicado "Personal Home Page Tools" después de que Lerdorf lo combinara con su propio *Form Interpreter* para crear PHP/FI.

PHP<sup>17</sup> es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* . Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado

---

<sup>16</sup> <http://es.wikipedia.org/wiki/PHP>

<sup>17</sup> Manual Imprescindible de PHP5 Introduccion; Ediciones Anaya Multimedia 2004; pag 30



bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

La similitud con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy suave. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas.

También es posible crear aplicaciones con una interfaz gráfica para el usuario. Además es posible utilizar PHP para generar archivos PDF, Flash, así como imágenes en diferentes formatos, entre otras cosas.

Permite también la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones Web muy robustas.

Los principales usos del PHP son los siguientes:

- Programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- Programación en consola, al estilo de Perl o Shell scripting.
- Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y Qt/GTK+, lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado.

### 2.2.2.1.2 AJAX

Ajax, es un nuevo conjunto de técnicas basado en tecnologías que conviven en el mundo de la Web. AJAX incorpora:

- Presentación basada en estándares usando XHTML y CSS;
- Exhibición e interacción dinámicas usando el Document Object Model;
- Intercambio y manipulación de datos usando XML and XSLT;
- Recuperación de datos asincrónica usando XMLHttpRequest y JavaScript

Ajax<sup>18</sup> permite que las páginas Web se comuniquen con su servidor sin necesidad de actualizar por completo la página, utilizando Ajax se puede traspasar gran parte del proceso al navegador, intercambiando información con el servidor solo cuando es necesario. Es perfecto para grandes proyectos, permite obtener grandes usos como en blogs, herramientas de aprendizaje, diarios online y comunidades Web.

El modelo clásico de aplicaciones Web funciona de esta forma: La mayoría de las acciones del usuario en la interfaz disparan un requerimiento HTTP al servidor web. El servidor efectúa un proceso (recopila información, procesa números, hablando con varios sistemas propietarios), y le devuelve una página HTML al cliente.

---

<sup>18</sup> AJAX Mejores Trucos; Ediciones Anaya 2007; Bruce W. Perry; pag 29

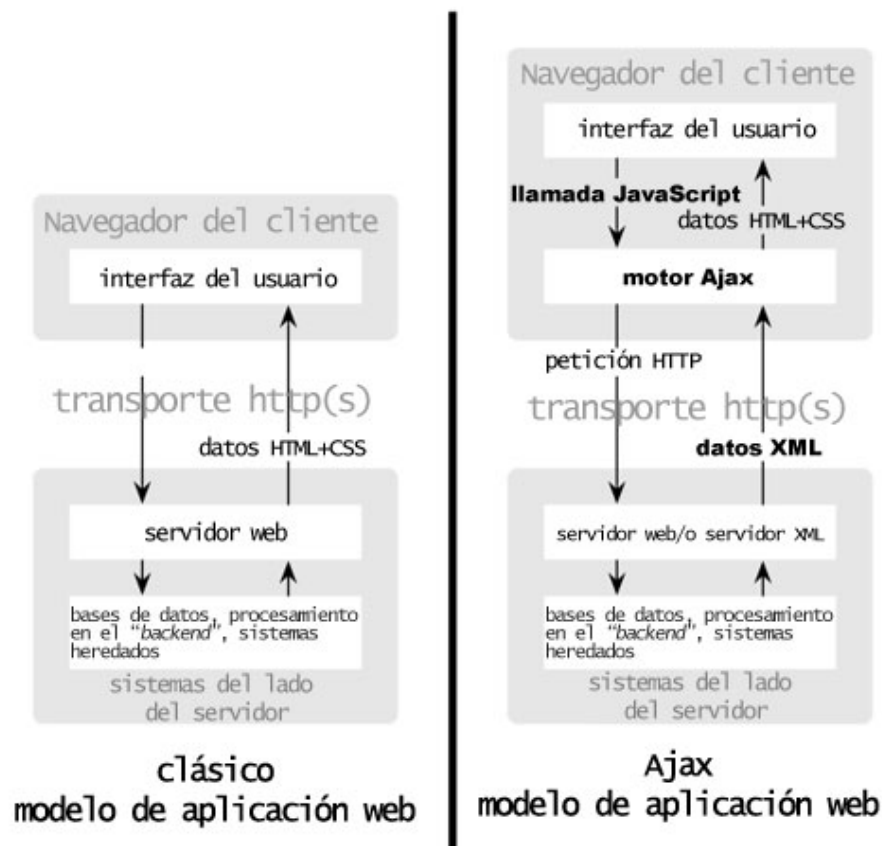


FIGURA 2.5 El modelo tradicional para las aplicaciones Web (izq.) comparado con el modelo de AJAX (der.).<sup>19</sup>

Una aplicación AJAX elimina la naturaleza “arrancar-frenar- arrancar-frenar” de la interacción en la Web introduciendo un intermediario -un motor AJAX- entre el usuario y el servidor. Parecería que sumar una capa a la aplicación la haría menos reactiva, pero la verdad es lo contrario.

En vez de cargar un pagina Web, al inicio de la sesión, el navegador carga al motor AJAX (escrito en JavaScript y usualmente “sacado” en un frame oculto). Este motor es el responsable por renderizar la interfaz que el usuario ve y por comunicarse con el servidor en nombre del usuario.

<sup>19</sup> <http://www.maestrosdelweb.com/editorial/ajax/>

El motor AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador y un icono de reloj de arena esperando a que el servidor haga algo.

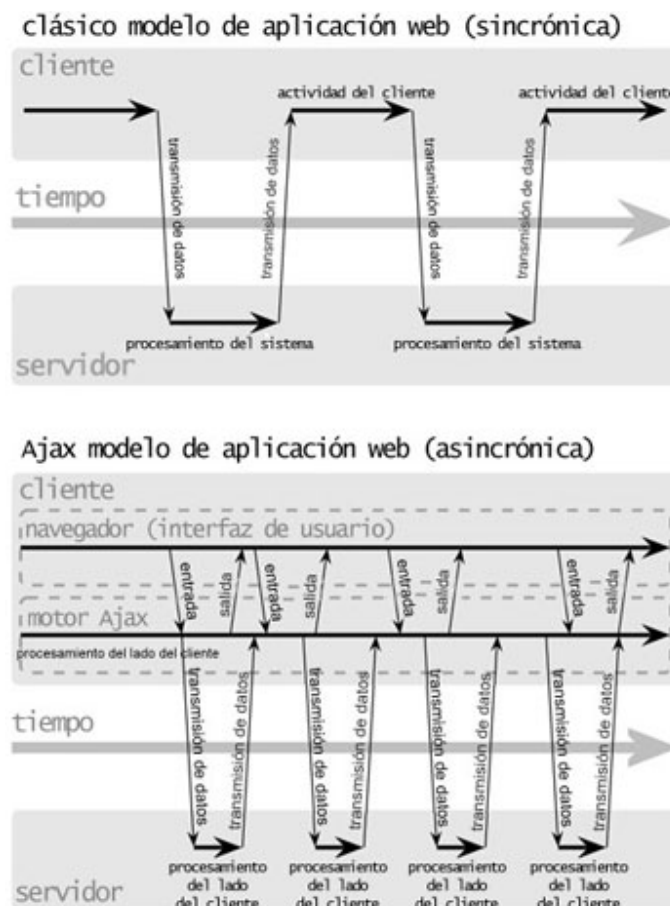


FIGURA 2.6 El patrón de interacción sincrónica de una aplicación Web tradicional (arriba) comparada con el patrón asincrónico de una aplicación AJAX (abajo)<sup>20</sup>

Cada acción de un usuario que normalmente generaría un requerimiento HTTP toma la forma de un llamado JavaScript al motor AJAX en vez de ese requerimiento. Cualquier respuesta a una acción del usuario que no requiera una viaje de vuelta al servidor es manejado por su cuenta.

<sup>20</sup> <http://www.maestrosdelweb.com/editorial/ajax/>

Si el motor necesita algo del servidor para responder hace esos pedidos asincrónicamente, usualmente usando XML, sin frenar la interacción del usuario con la aplicación.

### **2.2.3 Bases de datos**

#### *2.2.3.1 Introducción a MySql*

MySQL<sup>21</sup> es un sistema de gestión de base de datos relacional, multihilo y multiusuario, es muy utilizado en aplicaciones web como MediaWiki, Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL<sup>22</sup>, es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

Unas de las características que se destacan de la última versión son:  
Características de la versión 5.0.22

- Un amplio subconjunto de ANSI SQL 99, y varias extensiones.
- Soporte a multiplataforma
- Procedimientos almacenados
- Triggers
- Cursores
- Vistas actualizables
- Soporte a VARCHAR

---

<sup>21</sup> <http://es.wikipedia.org/wiki/MySQL>

<sup>22</sup> MySQL Avanzado; Ediciones Anaya Multimedia 2004; pag 46

- INFORMATION\_SCHEMA
- Modo Strict
- Soporte X/Open XA de transacciones distribuidas; transacción en dos fases como parte de esto, utilizando el motor InnoDB de Oracle
- Motores de almacenamiento independientes (MyISAM para lecturas rápidas, InnoDB para transacciones e integridad referencial)
- Transacciones con los motores de almacenamiento InnoDB, BDB Y Cluster; puntos de recuperación(savepoints) con InnoDB
- Soporte para SSL
- Query caching
- Sub-SELECTs (o SELECTs anidados)
- Réplica con un maestro por esclavo, varios esclavos por maestro, sin soporte automático para múltiples maestros por esclavo.
- indexing y buscando campos de texto completos usando el motor de almacenamiento MyISAM
- Embedded database library
- Soporte completo para Unicode
- Conforme a las reglas ACID usando los motores InnoDB, BDB y Cluster
- Shared-nothing clustering through MySQL Cluster.

#### 2.2.4 Tipos de compilación del servidor <sup>23</sup>

Hay tres tipos de compilación del servidor MySQL:

- Estándar: Los binarios estándares de MySQL son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento InnoDB.
- Max: Los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.
- MySQL-Debug: Son binarios que han sido compilados con información de depuración extra. No debe ser usada en sistemas en producción porque el código de depuración puede reducir el rendimiento.

---

<sup>23</sup> <http://es.kioskea.net/faq/3125-mysql-optimizacion>

### 3 FASE DE ANÁLISIS, DISEÑO Y CONSTRUCCIÓN

#### 3.1 Análisis

##### 3.1.1 Análisis de factibilidad

Los estudios de factibilidad consideran la factibilidad operacional, técnica y económica de cada alternativa, así como si el proyecto es o no apropiado dados los factores políticos y otros del contexto institucional.

##### 3.1.1.1 Factibilidad operacional

Esta factibilidad comprende una determinación de la probabilidad de que un nuevo sistema se use como se supone. El proyecto propuesto únicamente tiene beneficios cuando el grupo operativo satisfacen los requerimientos de la empresa RME, por tal motivo se ha realizado el siguiente estudio de los posibles riesgos que pueden afectar:

Nivel	Descripción
4	Ignominioso
3	Critico
2	Nocivo
1	Insignificante

Tabla 3.1 Nivel de Impacto de Riesgos

Elaborado por: Daniela Carrillo<sup>24</sup>

---

<sup>24</sup> Tabla tomada de referencia de Ingeniería de Software; Ian Sommerville-7 Edicion



Descripción	Impacto
Complejidad de uso para los usuarios de la organización o los operadores del sistema.	3
Falta de adaptación y aceptación del sistema	3
Fallas en el análisis o diseño del sistema	3
Surgen cambios a los requisitos después de la entrega	2
El cliente pide nuevos requisitos	1

Tabla 3.2 Nivel de Impacto de Riesgos

Elaborado por: Daniela Carrillo<sup>25</sup>

Elaborada la Tabla de Riesgos se analiza detalladamente los riesgos de mayor impacto operativo para determinar problemas y resultados.

- Complejidad de uso para los usuarios de la organización o los operadores del sistema: Ocasionara que los usuarios pueden ignorar el sistema o bien usarlo en tal forma que cause errores o fallas en el sistema.
- Falta de adaptación y aceptación del sistema: Esto puede ocasionar resistencia sin importar qué tan atractivo pueda ser el sistema si la factibilidad operacional indica que tal vez los usuarios no aceptarán el sistema o que el uso no adecuado generara muchos errores, provocando una baja en la moral.

---

<sup>25</sup> Tabla tomada de referencia de Ingeniería de Software; Ian Sommerville-7 Edicion

- Fallas en el análisis o diseño del sistema: Este riesgo puede ocasionar el desarrollo de un producto de mala calidad que no cubrirá todas las necesidades aspiradas por el cliente.
- Surgen cambios a los requisitos después de la entrega: Produce una reformulación de requisitos lo que dará como resultado a un retroceso del ciclo de vida aumentando el tiempo de desarrollo y los costos del proyecto.

Analizado los posibles riesgos de la factibilidad operacional se concluye que la empresa no cruzara por los riesgos mencionados debido a la buena aceptación por la parte del personal operativo de la empresa a esta propuesta ya que les facilitara su trabajo diario.

La empresa RME ha brindado todo el apoyo necesario para la realización de este proyecto, proporcionando los elementos necesarios además de anticipar en las prácticas y/o políticas administrativas, operativas de una futura integración de un nuevo sistema logrando así una correcta gestión.

Después de realizar los estudios de los riesgos se puede llegar a la conclusión que el catalogo virtual es operacionalmente viable.

#### *3.1.1.2 Factibilidad Técnica*

El análisis de factibilidad técnica evalúa si el hardware y software están disponibles y si tienen las capacidades técnicas requeridas por cada alternativa del diseño que se esté considerando. Los estudios de factibilidad técnica también consideran las interfaces del sistema.

A continuación se realiza un estudio de las herramientas de programación que se usarán en el desarrollo de este sistema, se elabora un cuadro comparativo basado en el aprendizaje, conocimiento, implantación, mantenimiento, costo y

funcionalidad de la herramienta, en el cual se califica de acuerdo a la tabla que hace referencia a los niveles de puntuación que se indica a continuación:

PUNTAJE	DESCRIPCION
1	Excelente
2	Muy Bueno
3	Regular
4	Malo
5	Deplorable

Tabla 3.3 Nivel de Puntuación  
Elaborado por: Daniela Carrillo<sup>26</sup>

	Peso General	ASP.Net/SQL Server		ASP.Net/MySQL		PHP/SQL Server		PHP/MySQL	
		Ptos	Peso	Ptos	Peso	Ptos	Peso	Ptos	Peso
<b>Aprendizaje</b>	0.3	2	0.25	1	0.25	2	0.25	1	0.27
<b>Conocimiento Herramienta</b>	0.2	2	0.2	4	0.2	2	0.17	3	0.18
<b>Mantenimiento</b>	0.2	3	0.1	3	0.16	3	0.15	2	0.18
<b>Implantación</b>	0.1	3	0.05	3	0.08	3	0.08	1	0.09
<b>Costo</b>	0.1	2	0.08	3	0.05	2	0.06	1	0.1
<b>Funcionalidad</b>	0.1	2	0.07	2	0.07	1	0.06	1	0.08
	<b>1.00</b>	<b>0.75</b>			<b>0.81</b>		<b>0.77</b>		<b>0.90</b>

Tabla 3.4 Modelo de Puntuación  
Elaborado por: Daniela Carrillo<sup>27</sup>

<sup>26</sup> Tabla tomada de referencia de Ingeniería de Software; Ian Sommerville-7 Edición

<sup>27</sup> Tabla tomada de referencia de Ingeniería de Software; Ian Sommerville-7 Edición

Una vez realizada esta tabla podemos observar que de las cuatro opciones disponibles para la realización de este proyecto, la que posee la mejor puntuación como se puede observar es la correspondiente a utilizar PHP como Lenguaje de Programación, y My SQL como motor de Base de Datos, porque el aprendizaje es rápido; se tiene conocimiento de la herramienta, el mantenimiento no es complejo, su implementación es rápida, su costo no tiene impacto por ser open source, y su funcionalidad satisface las necesidades de este proyecto.

Una vez realizado los estudios respectivos que intervienen dentro del proyecto, se puede concluir que es técnicamente viable.

### *3.1.1.3 Factibilidad Económica*

Este estudio de factibilidad económica incluye el análisis de costos y beneficios asociados con cada alternativa del proyecto. Mediante un análisis de costos/beneficio, y de adquirir, operar cada sistema alternativo se identifican y se hace una comparación de ellos. La empresa considera un presupuesto de \$10.000 para la totalidad del proyecto.

#### Recurso Técnico

Los costos de la Tabla fueron tomados a partir del 25 de Enero del 2010

<b>CANT. ITEM</b>	<b>DESCRIPCION</b>	<b>COSTO UNITARIO</b>	<b>% PRORRATEO</b>
2	Computador OPTIPLEX 755 C2D 2GB 160 GB	642,88	45
2	Monitor 17 " LCD	159,82	25
2	Mouse y Teclado	17,36	20
1	MULTIFUNCION LEXMARK	128,00	10

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

<b>COSTO PROYECTO</b>		<b>948,06</b>	100
-----------------------	--	---------------	-----

**Tabla 3.5 Planificación de Hardware**

Elaborado por: Daniela Carrillo<sup>28</sup>

ITEM	DESCRIPCION	COSTO UNITARIO	% PRORRATEO
1	ADOBE Dreamweaver CS4	399	20
2	NetBeans IDE 6.1 (freeware)	0.0	15
3	PHP (freeware)	0.0	30
4	MySql (freeware)	0.0	30
<b>COSTO PROYECTO</b>		<b>399</b>	100

**Tabla 3.6 Planificación de Software**

Elaborado por: Daniela Carrillo<sup>29</sup>

**Recursos Humanos**

Los costos de la Tabla fueron tomados a partir del 25 de Enero del 2010

ITEM	DESCRIPCION	COSTO UNITARIO	MESES DE TRABAJO	COSTO PROYECTO
1	Líder del proyecto- Desarrollador	\$600	6	3600
2	Diseñador	\$600	4	2400
<b>TOTAL</b>				<b>6000</b>

**Tabla 3.7 Planificación de Recursos Humanos**

Elaborado por: Daniela Carrillo

<sup>28</sup> Tabla tomada de referencia de Ingeniería de Software; Ian Sommerville-7 Edicion

<sup>29</sup> Tabla tomada de referencia de Ingeniería de Software; Ian Sommerville-7 Edicion

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

---

Costo del Proyecto

Los costos de la Tabla fueron tomados a partir del 25 de Enero del 2010

ITEM	DESCRIPCION	COSTO PROYECTO
1	Recursos Técnicos	\$1347,06
2	Recursos Humanos	\$ 6000
<b>TOTAL</b>		<b>\$7347,06</b>

Tabla 3.8 Planificación de Costos del Proyecto

Elaborado por: Daniela Carrillo

Con el análisis de las tablas se puede considerar que la empresa RME al implementar el catalogo virtual podrá aumentar la calidad de su servicio dando como resultado una mayor conformidad con su clientela aumentando así sus ingresos cubriendo y recuperando la totalidad de la inversión en el proyecto del catalogo virtual. Según el análisis se concluye el proyecto es económicamente factible dado que la inversión en hardware, software y recursos humanos que tiene un costo de \$ 7347,06 lo cual está dentro del presupuesto definido por la empresa RME.

### **3.1.2 Análisis de requerimientos**

#### *3.1.2.1 Requerimientos del Sistema*

##### *3.1.2.1.1 Requerimientos de Hardware*

En consideración los requerimientos recomendados para que el catalogo virtual son:

- Procesador: Mínimo Intel® Pentium® 4, Intel Centrino®, Intel Xeon®, o Intel Core™ Duo (o compatible)
- Disco Duro: Mínimo 1 GB espacio libre
- RAM: Mínimo 2 GB
- Pantalla de 1.280 x 800 con tarjeta de vídeo de 16 bits.

#### *3.1.2.1.2 Requerimientos de Software*

- Microsoft® Windows® XP con Service Pack 2 (se recomienda Service Pack 3) o Windows Vista® Home Premium, Business, Ultimate o Enterprise con Service Pack 1 (certificado para ediciones de Windows XP y Windows Vista de 32 bits).
- Servidor Web que se utilizara será el XAMPP

### **3.1.3 Etapas del Sistema**

#### *3.1.3.1 Planificación del Sistema*

En la etapa de planificación se realizan user stories escritas directamente por el cliente, cada historia de usuario fue recogida de un grupo de funcionalidades relacionadas entre sí, que a su vez mantuvieron una idea lógica. El ámbito de cada historia de usuario se definió estimando que el tiempo de codificación de la misma sería de 2 meses.

Se obtuvieron mini-versiones o tareas completamente funcionales que se manejan dentro de las historias de usuario.

Todas las tareas descritas forman parte de la práctica de planificación que además ayuda a describir el orden a seguir en la etapa de diseño.

Todas las historias de usuario escritas por el cliente se encuentran añadidas en la parte de Anexos.

#### *3.1.3.2 Diseño del Sistema*

Apoyado en los user stories, se pudo diseñar una aplicación que se la fue concibiendo en varias mini-versiones; se fue diseñando el sistema de acuerdo a lo que el cliente iba necesitando y resolviendo requerimientos importantes para el cliente.

### *3.1.3.2.1 Mini-Versiones*

Dentro de las mini-versiones se define el fondo de pantalla, ubicación del logo, estilos que se utilizarán en estas mini-versiones se tiene un relación muy directa con el cliente.

Se divide el sistema en mini-programas presentados al cliente.

Los módulos o las divisiones que tiene el sistema son:

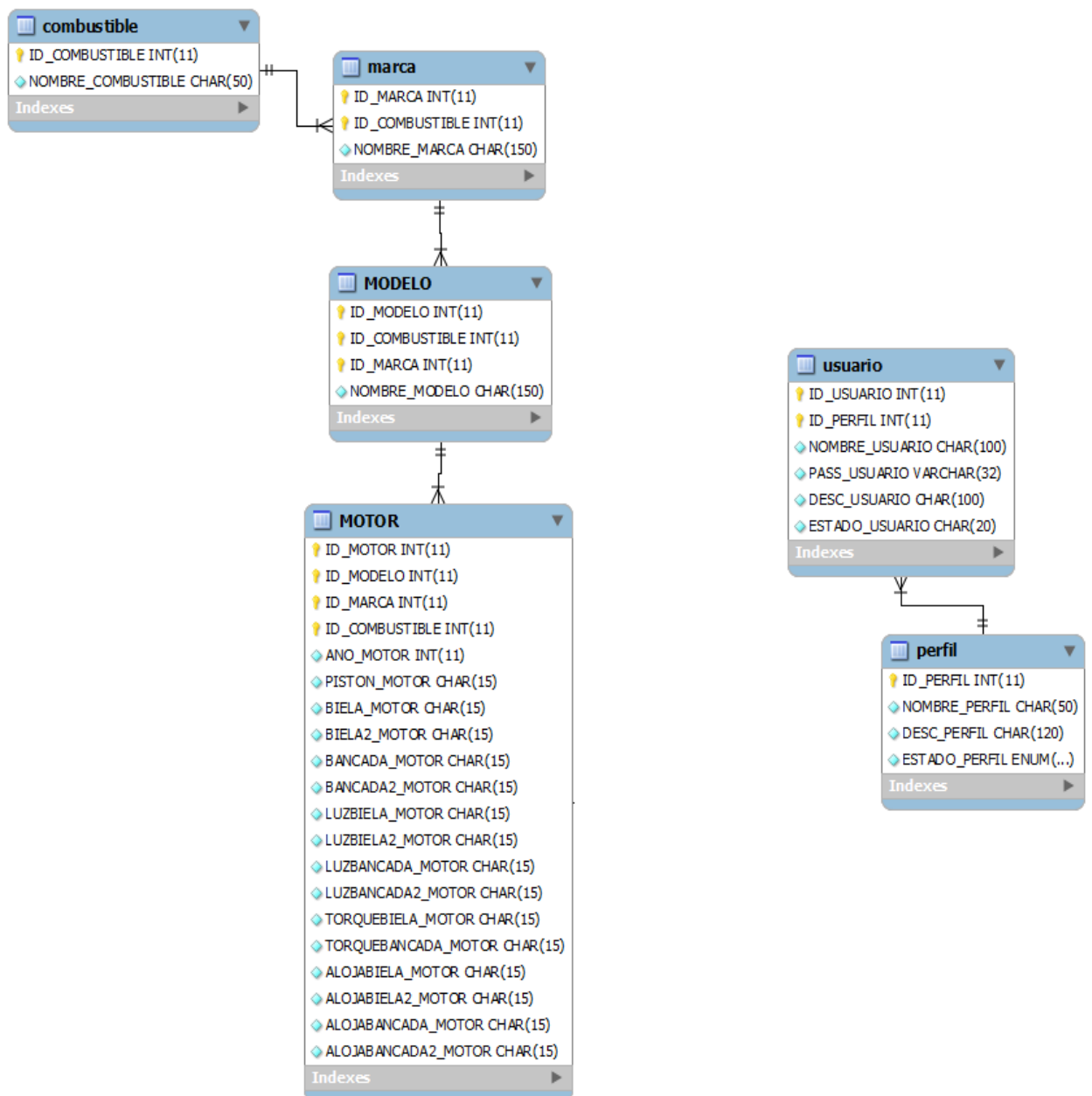
- Administración de Usuarios
  - Usuarios
  - Lista de Usuarios
- Administración de Motor
  - Parámetros
    - Combustible
    - Marca
    - Modelo
  - Motor
- Consulta Motor
  - Consulta
- Cambiar Clave

Las tareas también escritas y controladas por el cliente son añadidas en la parte de Anexos.

El código fuente se encuentra añadido en la parte de Anexos.



### 3.1.3.2.2 Modelo Relacional



### *3.1.3.3 Codificación del Sistema*

El cliente siempre estuvo a disposición el tiempo que fuera necesario para cumplir con las necesidades del equipo de trabajo, formando parte del mismo como una pieza fundamental, lo que permitió codificar adecuadamente el sistema.

### *3.1.3.4 Pruebas del Sistema*

Una vez concluido el proceso de codificación se pasa a la realización de pruebas unitarias, las pruebas unitarias se realizan concluido cada historia de usuario antes de ser lanzado a producción, en las pruebas se puede comprobar donde existen un error y poder corregirlo a tiempo antes de tener complicaciones posteriores.

Los casos de prueba comprobados por el cliente están añadidos en la parte de Anexos.

## 4 IMPLANTACIÓN

### 4.1 Implantación

Para la instalación del catalogo virtual para la empresa RME que posee administración de archivos y comunicación entre empleados, se requiere el siguiente software base para su correcto funcionamiento.

- Servidor Apache
- PHP (4.3.1 - 5.0)
- MySQL (4.1.9)

Se utiliza como unificación la herramienta XAMPP versión que unifica los tres elementos antes mencionados para un mejor rendimiento del catalogo virtual.

Para la instalación de la herramienta XAMPP es necesario cumplir con los requerimientos de hardware y software antes mencionados en el Capitulo 3.

### 4.2 Análisis de la aplicación

La aplicación realizada es un sistema de gestión Web, por lo que se ha considerado un análisis bajo 3 aspectos los más importantes son:

#### 4.2.1 Diseño de la interfaz

- La aplicación ha sido realizada bajo la estricta supervisión de la empresa RME debido a que el modulo de consulta es lo más importante dentro del catalogo virtual para el desarrollo de sus cálculos que son primordiales en su trabajo diario.
- El menú despliega las diferentes opciones de acuerdo al perfil de usuario ingresado.
- El catálogo virtual permite tener una administración de cada unos de los módulos antes mencionados.

- **Servicios proporcionados**

- ENTRADA DEL USUARIO.- Los usuarios que tienen el perfil de Administrador dentro de la base de datos pueden tener un acceso total de la información dentro de la misma. Mientras que los usuarios con el perfil de Operario solo pueden acceder al módulo de consulta respectivo para su uso de rectificación de partes.

- **Atributos de calidad**

- USABILIDAD.- La aplicación cumple con la capacidad de comprensión del sitio de manera global.
- FUNCIONALIDAD.- La aplicación posee la capacidad de consulta esperada por los administradores y operarios de la empresa, proporcionando la información necesaria para la rectificación del motor y sus partes.
- FIABILIDAD.- Existe un correcto enlace entre cada uno de los links proporcionando una navegación segura y sin errores.
- EFICIENCIA.- Los tiempos de respuesta de las consultas están dentro del promedio aceptable. Cabe resaltar que este parámetro depende del tipo de conexión con la cual se conecte al servidor de internet el usuario final.

### **4.3 Garantía y mantenimiento del producto**

Es necesario resaltar que el autor intelectual de este proyecto es Daniela Carrillo; y que la titularidad del sistema está a cargo de la institución educativa superior Pontificia Universidad Católica del Ecuador. Una vez aclarado esto hay que tomar en cuenta que la documentación de usuario, así como el código fuente quedarán a cargo del titular de este sistema de acuerdo a la ley y estatutos de la Pontificia Universidad Católica del Ecuador.

Se garantiza el correcto funcionamiento del sistema bajo las siguientes condiciones:

- La aplicación localmente se encuentra dentro del servidor donde se podrá tener acceso mediante un red interna proporcionada por la empresa RME.
- No se realicen cambios de ninguna índole a la aplicación.
- En el caso de que sea requerido realizar algún cambio, se notificará al titular del sistema para hacerlo o coordinarlo.

#### **4.4 Capacitación al usuario final de la aplicación**

La capacitación al usuario final, está dirigida a los administradores de la empresa RME: Christian Bonilla, Oscar Bonilla.

Dicha capacitación está enfocada al manejo del sistema como administradores, lo que implica el manejo de la información. Esta capacitación será realizada una vez terminada la aplicación y previo acuerdo con los administradores y el titular del sistema.

La capacitación del modulo de consulta está enfocada directamente con los operarios, lo que implica el ingreso del valor a ser rectificado con su respectivo calculo para su correcta funcionalidad. Esta capacitación será realizada por los administradores de la empresa RME.

Para facilidad de uso del sistema se genero el Manual de Usuario el cual será entregado a la empresa RME y as u vez forma parte de los anexos.

## 5 CONCLUSIONES Y RECOMENDACIONES

### 5.1 Conclusiones

- Uno de los objetivos del proyecto era identificar los requerimientos del cliente para la rectificación del motor y sus piezas, debido que en el principio se tenía ideas muy generales, para esto se hizo uso de una de las Metodologías Agiles como es Programación Extrema debido que a medida que se avanzó en el proyecto se podía ir tomando de mejor manera los requerimientos para lograr alcanzar el resultado esperado.
- La programación extrema como metodología es de gran utilidad debido que se obtuvo un contacto permanente con el cliente permitiendo levantar historias de usuarios, mini versiones, tareas y pruebas unitarias durante el desarrollo completo del sistema para la empresa RME.
- Las herramientas como PHP, MySQL y AJAX que fueron utilizadas en el desarrollo permitieron un adecuado moldeamiento con los requerimientos del cliente, al ser herramientas libres, la empresa RME invirtió lo necesario para que el catalogo virtual este con la nuevas tendencias utilizadas para un uso fácil y práctico para los usuarios de la empresa,
- Otra idea que se tomo fue implementar imágenes en 3D para darle una mejor apariencia al catálogo virtual y sea amigable para los usuarios de la empresa RME.
- Se logró obtener los cálculos de las piezas más importantes del motor como son biela, bancada y cigüeñal de una manera más fácil, rápida y precisa para los usuarios minimizando el tiempo de respuesta al momento de la entrega del servicio que brinda la empresa RME.

- Se implementó una manera rápida de mostrar los resultados de los datos iniciales, y los datos esperados al momento de la rectificación de la pieza del motor, permitiendo entregar al cliente el control del servicio realizado.

## **5.2 Recomendaciones**

- Cuando es tomada la Metodología Programación Extrema para el desarrollo de un proyecto es necesario al inicio definir las limitaciones debido que esta metodología al usar historias de usuario, el cliente puede tener requerimientos que se van incrementando a medida que se avanza con el desarrollo del proyecto.
- Es recomendable usar herramientas libres para desarrollar de un proyecto interactivo con el usuario, además que brinda la posibilidad de obtener un gran resultado con un presupuesto medio para su desarrollo.
- Es recomendable usar un diseño de navegación fácil y sencillo para el usuario, debido que necesitan obtener los resultados de una manera eficaz para lograr su objeto de brindar un servicio de calidad.
- Las herramientas utilizadas en el catalogo virtual tienen mayor flexibilidad al momento de instalar así que es recomendable utilizar en todas las sucursales de la empresa RME como forma estándar en realizar sus trabajos así se podrá brindar un servicio de calidad.
- Realizar una pequeña campaña de marketing donde se haga conocer a los clientes que se cuenta con sistemas modernos que garantizan la calidad del servicio.

- La empresa RME con la implementación del catálogo virtual como valor agregado a la calidad de su servicio debe realizar el esfuerzo necesario para conseguir la certificación de calidad ISO 9001.

## **6 Bibliografía**

- <http://www.willydev.net/descargas/prev/TodoAgil.pdf>
- <http://www.sel.unsl.edu.ar/ApuntesMaes/2004/Metodologias%20Agiles.doc>
- <http://www.rosenblueth.mx/InterFAR/Vol1Num3/doc/Vol1Num3-51.htm>
- <http://materias.fi.uba.ar/7500/schenone-tesisdegradoingenieriainformatica.pdf>
- [http://www.agileshift.cl/Tutorial/DesarrolloAgilParte1.pdf\(graficos\)](http://www.agileshift.cl/Tutorial/DesarrolloAgilParte1.pdf(graficos))
- <http://www.willydev.net/InsiteCreation/v1.0/descargas/prev/metodologiasagiles.pdf>
- <http://www.willydev.net/InsiteCreation/v1.0/descargas/articulos/general/xplibreap.aspx>
- <http://www.chuidiang.com/ood/metodologia/extrema.php>
- <http://www.willydev.net/InsiteCreation/v1.0/descargas/articulos/general/xplibreap.aspx>
- <http://www.uacj.mx/IIT/CULCYT/mayo-agosto2006/8ArtProg.pdf>
- [http://www.mygnet.net/manuales/uml/programacion\\_extrema.618](http://www.mygnet.net/manuales/uml/programacion_extrema.618)
- <http://www.maestrosdelweb.com/editorial/ajax/>



## 7 Anexos

# ANEXOS

# CATALOGO VIRTUAL DE SOPORTE TECNICO PARA EL DIAGNOSTICO Y RECTIFICACION DE MOTORES

## MANUAL DE USUARIO

“RECONSTRUCTORA MOTORISTA  
ECUATORIANA CIA. LTDA “



Quito, 2010

## INGRESO DE SISTEMA

El Ingreso del Sistema se debe realizar ingresando el usuario y la contraseña.



## Menu Principal

El Menu Principal va a depender del tipo de perfil del usuario que haya ingresado al sistema, existen dos usuarios que son:

**Administrador:** Tiene el acceso a todas las opciones del menú.

**Operador:** Tiene acceso únicamente a la opción Consulta Motor y Cambiar Clave.

En las opciones del Menú Principal son:

- Administración
  - Usuarios
  - Lista Usuarios
- Amín. Motor
  - Parámetros
    - Combustible
    - Marca
    - Modelo
  - Motor
- Consulta

Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA

---

- Cambiar Clave



## ADMINISTRACION

Permite la administración de Usuarios que contiene dos sub-opciones



**Usuarios:** permite el ingreso de Usuarios.

Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA

**REGISTRO EMPLEADOS**

Los campos con \* son de ingreso obligatorio

* Nombre:	<input type="text" value="Leonardo"/>
* Apellido:	<input type="text" value="Duran"/>
* Cedula de Indentidad:	<input type="text" value="0920184546"/>
* Usuario:	<input type="text" value="LDURAN"/>
* Perfil:	<input type="text" value="Operador"/>
* Fecha de Ingreso:	<input type="text" value="2010/10/04"/>
Fecha de Caducidad:	<input type="text" value="2011/10/04"/>
* Email:	<input type="text" value="lduran@rme.com"/>
* Usuario Activo:	Si <input checked="" type="radio"/> No <input type="radio"/>
* Area / Departamento:	<input type="text" value="Tecnico"/>
Telefono Movil:	<input type="text"/>
Telefono Fijo:	<input type="text"/>
* Acceso Web:	Si <input checked="" type="radio"/> No <input type="radio"/>
* Tiempo Sesion (minutos):	<input type="text" value="30"/>

**Lista de usuarios:** permite listar los Usuarios registrados en el sistema por diferentes filtros por ejemplo Nombre, Apellido, Cedula y Todos los registros.

**FILTRO DE BUSQUEDA :: EMPLEADOS**

Filtrar por:

-- Seleccione una Opcion --

**USUARIOS REGISTRADOS EN EL SISTEMA**

Su consulta genero 2 registros de resultado

Nombre	Apellido	Cédula	Telefono
Daniela	Carrillo	1715064471	222222
Christian	Bonilla	1234567890	2459622

Se presenta una pantalla con todos los datos del usuario haciendo click en cualquiera de los Usuarios que se muestran en la lista.

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

The screenshot shows a window titled 'Ficha Empleado' with a purple header bar containing the text 'FICHA EMPLEADO'. Below the header is a table with two columns of employee information. At the bottom of the window, there are two buttons: '[ Modificar ]' and '[ Cerrar Ventana ]'.

FICHA EMPLEADO			
Nombre	Christian Bonilla	Cedula de Identidad	1234567890
Empresa	RME S.A.	Departamento	Administrativo
Perfil	Administrador	Fecha Registro	2010-07-20
E-Mail	cbonilla@rme.com	Estado	ACTIVO
Celular	087422873	Telefono	2459622

[ Modificar ] [ Cerrar Ventana ]

Permite modificar los datos del Usuario antes ingresados en la sub-opción Usuarios.

Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA

USUARIOS REGISTRADOS EN EL SISTEMA

Su consulta genero 2 registros de resultado

Nombre	Apellido	Cédula	Telefono
Daniela	Carrillo	1715064471	222222
Christian	Bonilla	1234567890	2459622

Los campos con \* son de ingreso obligatorio

\* Nombre:

\* Apellido:

\* Cedula de Indentidad:

\* Usuario:

\* Perfil:

\* Fecha de Ingreso:

Fecha de Caducidad:

\* Email:

\* Usuario Activo: Si  No

\* Area / Departamento:

Telefono Movil:

Telefono Fijo:

\* Acceso Web: Si  No

\* Tiempo Sesion (minutos):

**ADMIN MOTOR**

Permite la administración del motor y sus partes, contiene dos sub-opciones Parámetros y Motor.

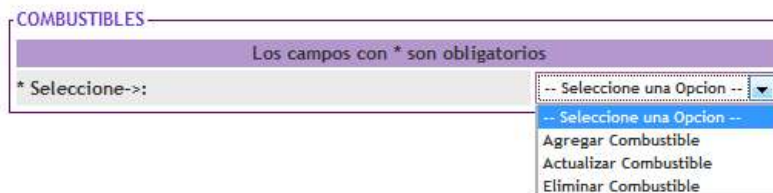


**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

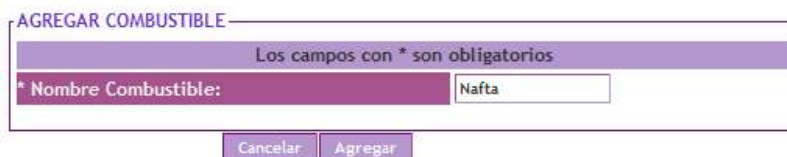
Parámetros: permite la administración de los parámetros iniciales del motor que son Combustible, Marca, Modelo



Combustible: permite la administración de combustible, esto significa Agregar, Actualizar y Eliminar.

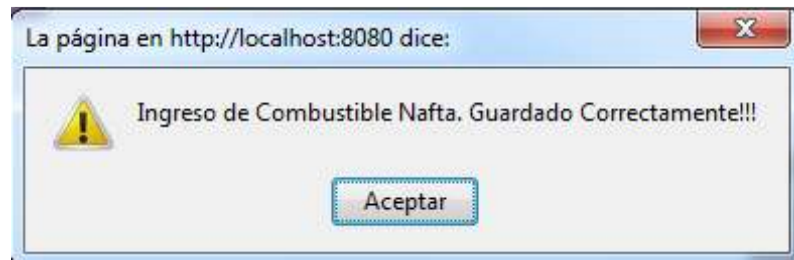


Agregar Combustible: Permite ingresar un nuevo combustible.



Si hemos ingresado correctamente aparece el siguiente mensaje de confirmación.

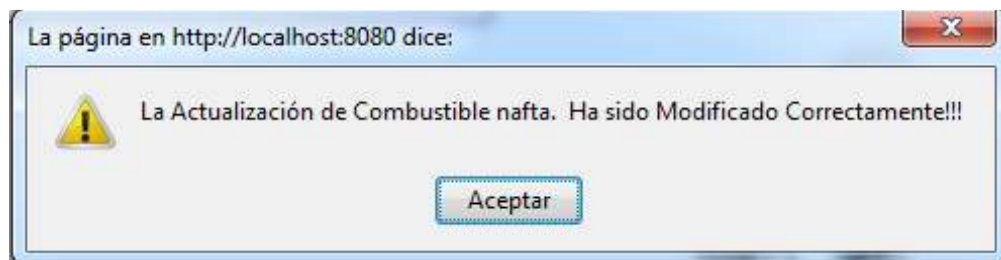




Actualizar Combustible: Permite actualizar un combustible.

A screenshot of a web form titled "ACTUALIZAR COMBUSTIBLES". Below the title is a purple header bar with the text "Los campos con \* son obligatorios". The form contains two fields: "\* Combustible:" with a dropdown menu showing "Nafta" and a downward arrow, and "\* Nombre Combustible:" with a text input field containing "nafta". At the bottom of the form are two buttons: "Cancelar" and "Editar".

Si hemos actualizado correctamente aparece el siguiente mensaje de confirmación.



Eliminar Combustible: Permite eliminar un combustible ingresado.

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

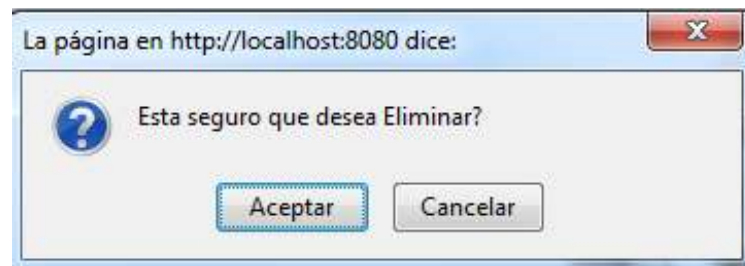
---

ELIMINAR COMBUSTIBLES

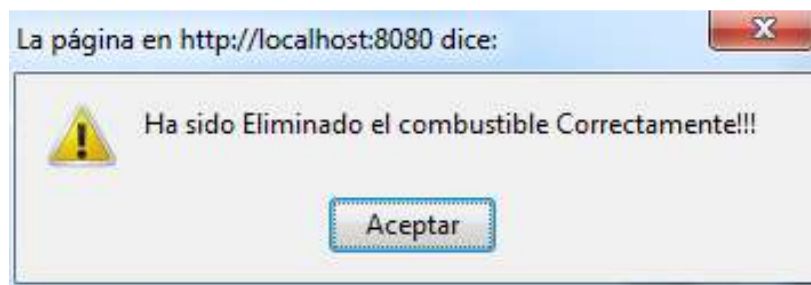
\* Combustible:



Indica un mensaje de confirmación antes de eliminar el combustible.



Muestra el siguiente mensaje de confirmación cuando la eliminación se ha realizado.



Marca: permite la administración de marca, esto significa Agregar, Actualizar, y Eliminar.

## Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA



MARCA  
\* Seleccione -> [Dropdown menu with options: Seleccione una Opción, Seleccione una Opción, Agregar Marca, Actualizar Marca, Eliminar Marca]



Agregar Marca: Permite insertar una nueva marca dependiendo del combustible.



AGREGAR MARCA  
Los campos con \* son obligatorios  
\* Combustible: Gasolina  
\* Nombre Marca: Renault  
[Cancelar] [Agregar]



Se presenta el siguiente mensaje de confirmación.



Actualizar Marca: Permite actualizar la información de la marca antes ingresada.

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

EDITAR MARCA

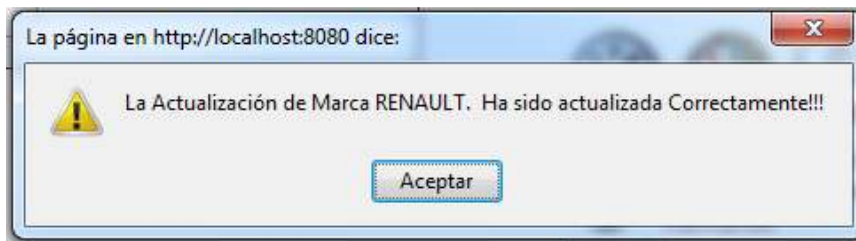
Los campos con \* son obligatorios

* Combustible:	Gasolina
* Marca:	Renault
* Nombre Marca:	RENAULT

Cancelar Editar



Se presenta el siguiente mensaje de confirmación.



Eliminar Marca: Permite eliminar una marca antes ingresada.

BORRAR MARCA

Los campos con \* son obligatorios

* Combustible:	Gasolina
* Marca:	RENAULT

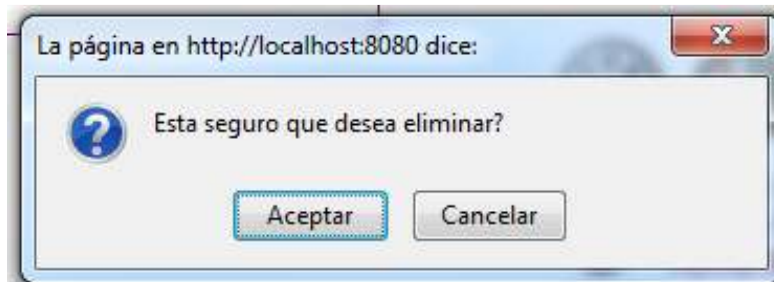
Cancelar Eliminar



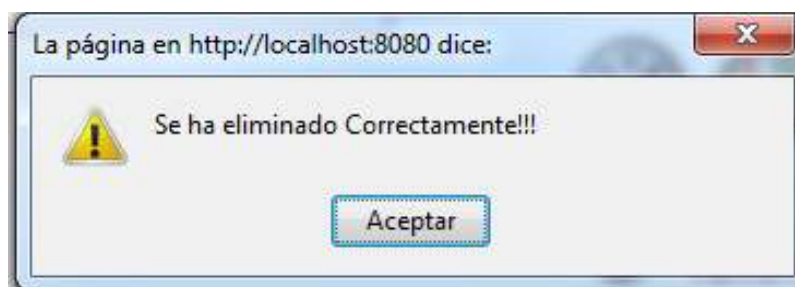
Antes de eliminar se presenta el siguiente mensaje de confirmación.

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

---



Muestra el siguiente mensaje cuando se ha eliminado la marca.



Modelo: permite la administración de combustible, esto significa Agregar, Actualizar, Eliminar modelo.



Agregar Modelo: Permite insertar un modelo dependiente del combustible y de la marca.

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

AGREGAR MODELO

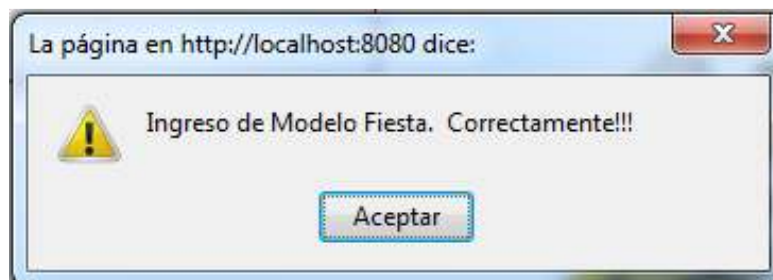
Los campos con \* son obligatorios

* Combustible:	Gasolina
* Marca:	Ford
* Nombre Modelo:	Fiesta

Cancelar Agregar



Se indica el siguiente mensaje de confirmación.



Actualizar Modelo: Permite actualizar la información del modelo antes ingresado.

EDITAR MODELO

Los campos con \* son obligatorios

* Combustible:	Gasolina
* Marca:	Ford
* Modelo:	Fiesta
* Nombre Modelo:	Fiesta

Cancelar Editar

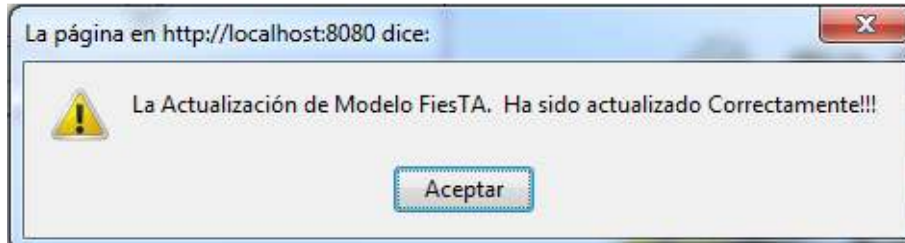





**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

---

Se indica el siguiente mensaje de confirmación.



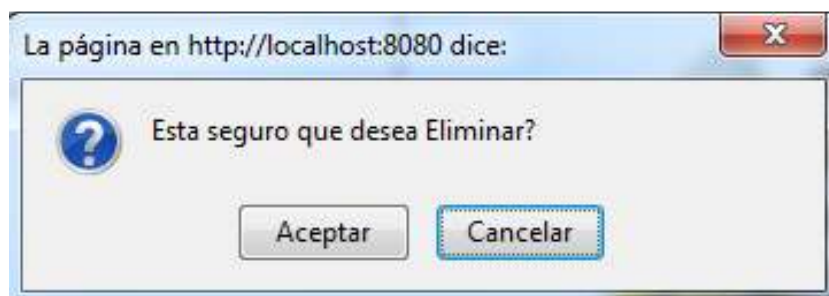
Eliminar Modelo: Permite eliminar un modelo antes ingresado.



A screenshot of a web form titled "BORRAR MODELO". The form has a header "Este campo con \* son obligatorios". It contains three fields: "Combustible" with a dropdown menu showing "Gasolina", "Marca:" with a dropdown menu showing "Ford", and "Modelo:" with a dropdown menu showing "FiesTA". At the bottom of the form are two buttons: "Cancelar" and "Eliminar".



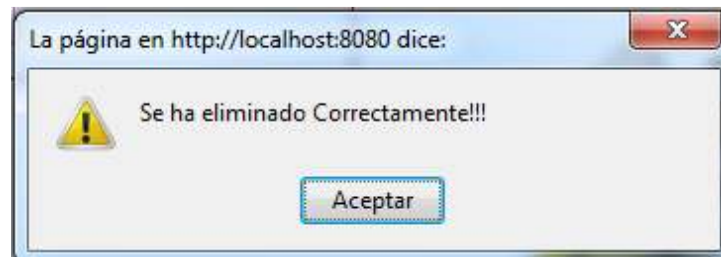
Se indica el siguiente mensaje de confirmación.



Se indica el siguiente mensaje después de ser eliminado.

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

---



Motor: permite la administración de motor, esto significa Agregar, Actualizar, Eliminar el motor con sus partes respectivamente.



Agregar Motor: Permite insertar un motor con tus partes dependiente del combustible, marca y modelo.



**Diseño, desarrollo e implementación de un catálogo virtual de soporte técnico para el diagnóstico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

Los campos con \* son obligatorios

Combustible: Diesel

\* Marca: KIA

\* Modelo: PREGGIO

\* Nombre: A350      \* Año: 2008

Medida: Pulgadas

\* Cilindraje: 1.2

\* Pistón: 2.3

\* Brazo de Biela:      Sup      Inf

3.2      3.4

Alojamiento de Biela:      Sup      Inf

0      0

\* Torque Biela: 3.1

\* Stroke Diagonal: 3.6

\* Luz Biela:      Sup      Inf

6.2      6.4

\* Luz Bancada:      Sup      Inf

6.6      6.8

\* Castignum Diagonal: 36.89

\* Comments Diagonal: Motor A350 del 100

Biela	Std	S1	S2	S3	S4	S5	S6
Sup	44.10	44.30	44.90	45.10	45.30	45.90	46.10
Inf	44.30	44.70	45.00	45.30	45.70	46.00	46.30

Bancada	Std	S1	S2	S3	S4	S5	S6
Sup	47.10	47.30	47.90	48.10	48.30	48.90	49.10
Inf	47.30	47.70	48.00	48.30	48.70	49.00	49.30

\* Torque Bancada: 30.1

Alojamiento Bancada:      Sup      Inf

32.10      32.30

Cilindro	Std	S1	S2	S3	S4	S5	S6
Sup	35.10	35.30	35.90	36.10	36.30	36.90	37.10
Inf	35.30	35.70	36.00	36.30	36.70	37.00	37.30

\* Castignum Cilindro: 38.1

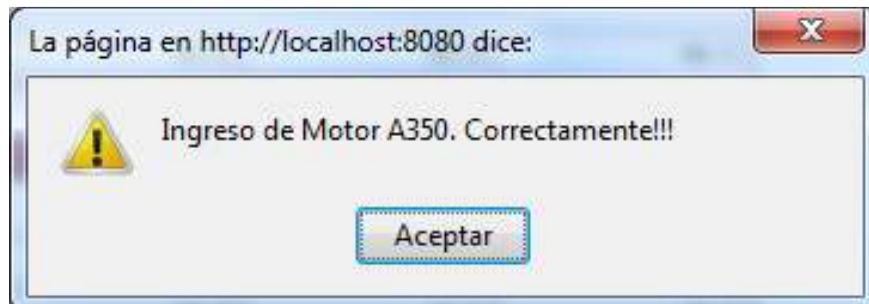
\* Comments Cilindro: Motor A350

Cancelar    Borrar

La mayoría de datos a ingresar solo permiten ingresar números, si estos son de orden decimal se separa la parte decimal con el signo de PUNTO.

Se indica el siguiente mensaje de confirmación.

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**



Actualizar Motor: Permite actualizar los datos de las partes de un motor antes ingresado.

**EDITAR MOTOR**

Los campos con \* son obligatorios

\* Combustible: Diesel

\* Marca: KIA

\* Modelo: PREGGIO

\* Nombre: A350 \* Año: 2006

Medida: Pulgadas

\* Cilindraje: 1.2

\* Pistón: 2.3

\* Brazo de Biela: Sup Inf

3.2 3.4

Alojamiento de Biela: Sup Inf

3.5 3.7

\* Torque Biela: 3.1

\* Stroke Ciguala: 3.6

\* Luz Biela: Sup Inf

6.2 6.4

\* Luz Bancada: Sup Inf

6.6 6.8

\* Castingnum Ciguala: 56.39

\* Comments Ciguala: Motor A350 del 200

Biela	Std	S1	S2	S3	S4	S5	S6
Sup	44.10	44.30	44.90	45.10	45.50	45.90	46.10
Inf	44.30	44.70	45.00	45.30	45.70	46.00	46.30

Bancada	Std	S1	S2	S3	S4	S5	S6
Sup	47.10	47.30	47.90	48.10	48.50	48.90	49.10
Inf	47.30	47.70	48.00	48.30	48.70	49.00	49.30

\* Torque Bancada: 30.1

Alojamiento Bancada: Sup Inf

32.10 32.30

Cilindro	Std	S1	S2	S3	S4	S5	S6
Sup	35.10	35.30	35.90	36.10	36.50	36.90	37.10
Inf	35.30	35.70	36.00	36.30	36.70	37.00	37.30

\* Castingnum Cilindro: 35.1

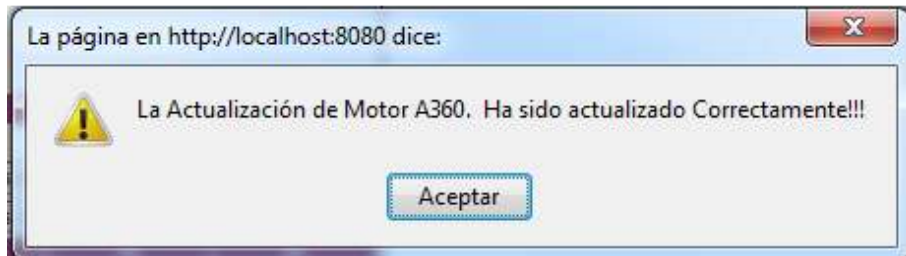
\* Comments Cilindro: Motor A350

Cancelar Editar

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

---

Se presenta el siguiente mensaje de confirmación.



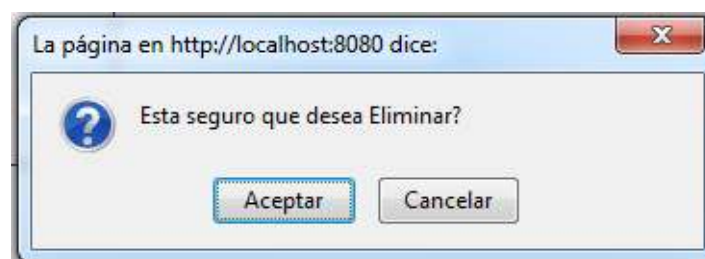
Eliminar Motor: Permite eliminar un motor antes ingresado.



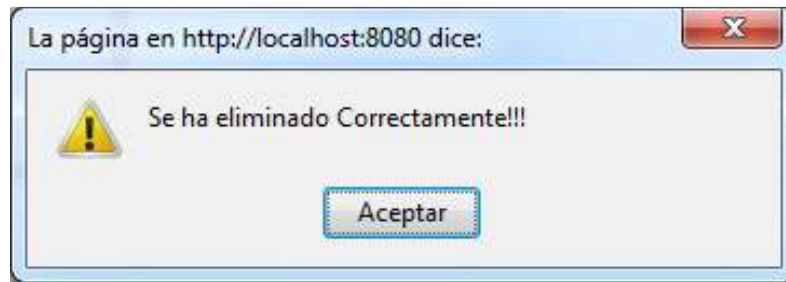
A screenshot of a web form titled "ELIMINAR MOTOR". The form has a header "Los campos con \* son obligatorios". It contains four fields, each with a red asterisk: "Combustible:" with a dropdown menu showing "Diesel"; "Marca:" with a dropdown menu showing "KIA"; "Modelo:" with a dropdown menu showing "PREGGIO"; and "Motor:" with a dropdown menu showing "A360". At the bottom of the form are two buttons: "Cancelar" and "Eliminar".



Se presenta el siguiente mensaje de confirmación.



Una vez eliminado el motor seleccionado se muestra el siguiente mensaje de confirmación.



### CONSULTA MOTOR

Contiene una sub-opción llamada Hacer Consulta.



Permite la búsqueda rápida del motor que antes fue ingresado en la opción Admin. Motor.

A screenshot of a search form. The form has a title "BUSCAR" in blue. Below the title is a text input field with the placeholder text "Buscar:". To the right of the input field is a button labeled "Buscar".

Se selecciona el motor buscado indicando toda la información relacionada al mismo.

**Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA**

**CONSULTA MOTOR**

Combustible:	Gasolina	Marca:	Toyota	Modelo:	Corona			
Nombre Motor:	2R	Año:	1964					
Cilindraje:	1490	MM	37846	PLG				
Piston:	78x78	MM	1981.2	PLG				
Brazo de Biela:	Sup		Inf					
	48.9857	mm	1244.237	plg	48.009 mm 1219.429 plg			
Luz Biela:	Sup		Inf					
	0.020	mm	0.508	plg	0.067 mm 1.702 plg			
Luz Bancada:	Sup		Inf					
	0.020	mm	0.508	plg	0.067 mm 1.702 plg			
Biela	Std	S1	S2	S3	S4	S5	S6	
	MM	PLG	MM	PLG	MM	PLG	MM	PLG
	Sup	45.01 1143.254	44.95 1141.73	44.91 1140.714	44.87 1139.698	44.81 1138.174	44.77 1137.158	44.73 1136.142
Inf	44.99 1142.746	44.93 1141.222	44.89 1140.206	44.85 1139.19	44.79 1137.666	44.75 1136.65	44.71 1135.634	
Bancada	Std	S1	S2	S3	S4	S5	S6	
	MM	PLG	MM	PLG	MM	PLG	MM	PLG
	Sup	44.01 1117.854	43.95 1116.33	43.91 1115.314	43.87 1114.298	43.81 1112.774	43.77 1111.758	43.73 1110.742
Inf	43.99 1117.346	43.93 1115.822	43.89 1114.806	43.85 1113.79	43.79 1112.266	43.75 1111.25	43.71 1110.234	
Cilindro	Std	S1	S2	S3	S4	S5	S6	
	MM	PLG	MM	PLG	MM	PLG	MM	PLG
	Sup	43.01 1092.454	42.95 1090.93	42.91 1089.914	42.87 1088.898	42.81 1087.374	42.77 1086.358	42.73 1085.342
Inf	42.99 1091.946	42.93 1090.422	42.89 1089.406	42.85 1088.39	42.79 1086.866	42.75 1085.85	42.71 1084.834	
Comments Cilindro:	2r							

Una vez mostrada toda la información presionar el botón “Siguiente” que permite el cálculo de las piezas a rectificar que son Biela, Bancada, Cigüeñal, permite seleccionar Pulgadas o Milímetros para ingresar los datos dependiendo de la selección.

**CALCULO**

**Condiciones Iniciales**

Los campos con \* son obligatorios

\* Medida:  Pulgadas  Milímetros

\* Biela:

\* Bancada:

\* Cilindro:

Se muestra en pantalla los datos Iniciales, los datos sugeridos para la rectificación de las piezas del motor.

**CALCULO**

Condiciones Iniciales

Los campos con \* son obligatorios

\* Medida: Pulgadas  Milímetros

\* Biela:

\* Bancada:

\* Cilindro:

Cancelar Calcular

**CONTROL CALIDAD MOTOR**

Ciguenal	
Valores Iniciales Biela	1
Valores Finales Biela	El valor ingresado 1 no esta en los rangos de valores de Biela
Verificacion Biela	1
Valores Iniciales Bancada	2
Valores Finales Bancada	El valor ingresado 2 no esta en los rangos de valores de Bancada
Verificacion Bancada	1
Cilindro	
Valores Iniciales Cilindro	3
Valores Finales Cilindro	El valor ingresado 3 no esta en los rangos de valores del Cilindro
Verificacion Cilindro	3

Cancelar

### CAMBIAR CLAVE

Permite el cambio de clave del usuario que ingreso al sistema.

**CAMBIO DE CLAVE**

Los campos con \* son de ingreso obligatorio

\* Usuario:

\* Clave Actual:

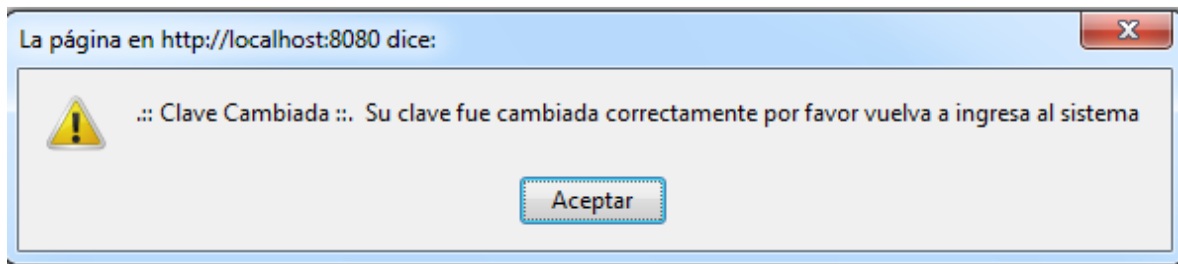
\* Ingrese la Clave Nueva:

\* Re Ingrese la Clave Nueva:

Guardar Limpiar

Se indica el siguiente mensaje de confirmación.

Diseño, desarrollo e implementación de un catalogo virtual de soporte técnico para el diagnostico y rectificación de motores para la empresa RECONSTRUCTORA MOTORISTA ECUATORIANA CIA. LTDA



## SALIR DEL SISTEMA

Permite salir del sistema.



## CODIGO

```
<?php
/*****
*****/
/*.....*/
/*
@    cob03 :: genera_cabecera_ruta
@    Genera la cabecera de la creacion de rutas
    para los clientes.
*/
/*****/
/* FCA01 :: GENERA INGRESO TABLA MOTOR */
/*****/
function genera_motor($aFrm = array(),$accion=0,$tipo=0) {
    //Definiciones
    global $DSN;

    session_start ();

    $fu = new Formulario ( );
    $fu->DSN = $DSN;

    $oCon = new Dbo ( );
    $oCon->DSN = $DSN;
    $oCon->Conectar ();

    $sHtml = "";

    $oReturn = new xajaxResponse ( );

    switch ($accion)
    {
        case 0: //por default
            $fu->AgregarCampoLista('accion','Seleccione->',true,auto);
            $fu->AgregarOpcionCampoLista('accion','Agregar Motor',1);
            $fu->AgregarOpcionCampoLista('accion','Actualizar Motor',2);
            $fu->AgregarOpcionCampoLista('accion','Eliminar Motor',3);
            $fu-
>AgregarComandoAlCambiarValor('accion',"xajax_genera_motor(xajax.getFor
mValues('form1'),document.getElementById('accion').value)");

```

ruta\_crea.php



```

        $sHtml .= '<fieldset>
                <legend class="Titulo">Motor</legend>;
                $sHtml .= '<table align="center" cellpadding="0"
cellspacing="2" width="100%" border="0">
                <tr
                class="msgFrm"><td
                colspan="4"
align="center"></td></tr>;
                $sHtml .= '<tr>
                <td class="labelFrm">' . $fu->ObjetoHtmlLBL ( 'accion' ) .
'</td>
                <td colspan="3">' . $fu->ObjetoHtml ( 'accion' ) . '</td>
                </tr>;
                break;
    case 1: //Agregar Motor
        switch($tipo)
        {
            case 0:
                $sql = 'SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';
                $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$accion,1)");

                $sHtml .= '<fieldset>
                <legend
                class="Titulo">Agregar
Motor</legend>;
                $sHtml .= '<form id="DataGrid">;
                $sHtml .= '<table align="center" cellpadding="0"
cellspacing="2" width="100%" border="0">
                <tr class="msgFrm"><td colspan="4"
align="center">Los campos con * son obligatorios</td></tr>;
                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
                <td colspan="3">' . $fu->ObjetoHtml (
'combustible' ) . '</td>
                </tr>;
                $sHtml .= '</table>;
                $sHtml .= '</form>;
                $sHtml .= '</fieldset>;
                break;
    case 1:

```

```

                                $sql                                ='SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';
                                $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );
                                $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$accion,1)");
                                $fu->cCampos['combustible']-
>xValor=$aFrm['combustible'];

                                $sql ='SELECT ID_MARCA,NOMBRE_MARCA
FROM marca WHERE ID_COMBUSTIBLE ='.$aFrm['combustible']. ' and estado =
1 ORDER BY NOMBRE_MARCA ASC';
                                $fu->AgregarCampoListaSQL ( 'marca', 'Marca',
$sql, true, 'auto' );
                                $fu-
>AgregarComandoAlCambiarValor('marca',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$accion,2)");
                                $fu->cCampos['marca']-
>xValor=$aFrm['marca'];

                                $sHtml .= '<fieldset>
                                <legend                                class="Titulo">Agregar
Motor</legend>';
                                $sHtml .= '<form id="DataGrid">';
                                $sHtml .= '<table align="center" cellpadding="0"
cellspacing="2" width="100%" border="0">
                                <tr class="msgFrm"><td colspan="4"
align="center">Los campos con * son obligatorios</td></tr>';
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
                                <td colspan="3">' . $fu->ObjetoHtml (
'combustible' ) . '</td>
                                </tr>';
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL ( 'marca' ) .
'</th>
                                <td colspan="3">' . $fu->ObjetoHtml (
'marca' ) . '</td>
                                </tr>';
                                $sHtml .= '</table>';
                                $sHtml .= '</form>';
                                $sHtml .= '</fieldset>';

```

```

                break;
            case 2:
                $sql = 'SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';
                $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$accion,1)");
                $fu->cCampos['combustible']-
>xValor=$aFrm['combustible'];

                $sql = 'SELECT ID_MARCA,NOMBRE_MARCA
FROM marca WHERE ID_COMBUSTIBLE ='.$aFrm['combustible'].' and estado =
1 ORDER BY NOMBRE_MARCA ASC';
                $fu->AgregarCampoListaSQL ( 'marca', 'Marca',
$sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('marca',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$accion,2)");
                $fu->cCampos['marca']-
>xValor=$aFrm['marca'];

                $sql = 'SELECT ID_MODELO,NOMBRE_MODELO
FROM modelo WHERE ID_MARCA ='.$aFrm['marca'].' and estado = 1 ORDER
BY NOMBRE_MODELO ASC';
                $fu->AgregarCampoListaSQL ( 'modelo',
'Modelo', $sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('modelo',"xajax_genera_motor(xajax.getFor
mValues('DataGrid'),$accion,3)");
                $fu->cCampos['modelo']-
>xValor=$aFrm['modelo'];

                $sHtml .= '<fieldset>
<legend class="Titulo">Agregar
Motor</legend>';
                $sHtml .= '<form id="DataGrid">';
                $sHtml .= '<table align="center" cellpadding="0"
cellspacing="2" width="100%" border="0">
<tr class="msgFrm"><td colspan="4"
align="center">Los campos con * son obligatorios</td></tr>';
                $sHtml .= '<tr>

```

```

        <th> . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
        <td colspan="3"> . $fu->ObjetoHtml (
'combustible' ) . '</td>
        </tr>;
        $sHtml .= '<tr>
        <th> . $fu->ObjetoHtmlLBL ( 'marca' ) .
'</th>
        <td colspan="3"> . $fu->ObjetoHtml (
'marca' ) . '</td>
        </tr>;
        $sHtml .= '<tr>
        <th> . $fu->ObjetoHtmlLBL ( 'modelo' ) .
'</th>
        <td colspan="3"> . $fu->ObjetoHtml (
'modelo' ) . '</td>
        </tr>;

        $sHtml .= '</table>;
        $sHtml .= '</form>;
        $sHtml .= '</fieldset>;
        break;
    case 3:
        $sql = 'SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';
        $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );
        $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$accion,1)");
        $fu->cCampos['combustible']-
>xValor=$aFrm['combustible'];

        $sql = 'SELECT ID_MARCA,NOMBRE_MARCA
FROM marca WHERE ID_COMBUSTIBLE ='.$aFrm['combustible'].' and estado =
1 ORDER BY NOMBRE_MARCA ASC';
        $fu->AgregarCampoListaSQL ( 'marca', 'Marca',
$sql, true, 'auto' );
        $fu-
>AgregarComandoAlCambiarValor('marca',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$accion,2)");
        $fu->cCampos['marca']-
>xValor=$aFrm['marca'];

```

```

                                $sql ='SELECT ID_MODELO,NOMBRE_MODELO
FROM modelo WHERE ID_MARCA ='.$aFrm['marca'].' and estado = 1 ORDER
BY NOMBRE_MODELO ASC';
                                $fu->AgregarCampoListaSQL      (      'modelo',
'Modelo', $sql, true, 'auto' );
                                $fu-
>AgregarComandoAlCambiarValor('modelo',"xajax_genera_motor(xajax.getFor
mValues('DataGrid'),$accion,3)");
                                $fu->cCampos['modelo']-
>xValor=$aFrm['modelo'];

                                $fu-
>AgregarCampoTexto('nombre_motor','Nombre',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('ano_motor','Año',true,"",100,50);
                                $fu-
>AgregarCampoMm_Plg('medida','Medida',0);
                                $fu-
>AgregarCampoTexto('cilindraje','Cilindraje',false,"",100,50);
                                $fu-
>AgregarCampoTexto('piston_motor','Pistón',false,"",100,50);
                                $fu-
>AgregarCampoTexto('brazo_biela_motor','Brazo de Biela',false,"",100,50);
                                $fu-
>AgregarCampoTexto('brazo_biela2_motor',"",false,"",100,10);
                                $fu-
>AgregarCampoNumerico('std_inf_alojabiela_motor','Std Inferior
',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('std_sup_alojabiela_motor','Std
Superior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('torquebiela_motor','Torque Biela',false,"",100,50);
                                $fu-
>AgregarCampoNumerico('stroke_ciguenal','Stroke Ciguenal',false,"",100,50);
                                $fu-
>AgregarCampoNumerico('luzbiela_motor','Luz Biela',false,"",100,50);
                                $fu-
>AgregarCampoNumerico('luzbiela2_motor',"",false,"",100,10);
                                $fu-
>AgregarCampoNumerico('luzbancada_motor','Luz Bancada',false,"",100,50);
                                $fu-
>AgregarCampoNumerico('luzbancada2_motor',"",false,"",100,10);

```

```

                                $fu-
>AgregarCampoTexto('castingnum_ciguenal','Castingnum
Ciguenal',false,"",100,50);
                                $fu-
>AgregarCampoTexto('comments_ciguenal','Comments
Ciguenal',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('std_sup_biela','Standar Superior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('std_inf_biela','Standar Inferior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('sobre1_sup_biela','Medida
Superior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('sobre1_inf_biela','Medida 1 Inferior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('sobre2_sup_biela','Medida
Superior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('sobre2_inf_biela','Medida 2 Inferior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('sobre3_sup_biela','Medida
Superior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('sobre3_inf_biela','Medida 3 Inferior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('sobre4_sup_biela','Medida
Superior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('sobre4_inf_biela','Medida 4 Inferior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('sobre5_sup_biela','Medida
Superior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('sobre5_inf_biela','Medida 5 Inferior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('sobre6_sup_biela','Medida
Superior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('sobre6_inf_biela','Medida 6 Inferior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('std_sup_bancada','Std Superior',true,"",100,50);
                                $fu-
>AgregarCampoNumerico('std_inf_bancada','Std Inferior',true,"",100,50);
```

\$fu- >AgregarCampoNumerico('sobre1_sup_bancada','Medida Superior',true,"",100,50);	1
\$fu- >AgregarCampoNumerico('sobre1_inf_bancada','Medida Inferior',true,"",100,50);	1
\$fu- >AgregarCampoNumerico('sobre2_sup_bancada','Medida Superior',true,"",100,50);	2
\$fu- >AgregarCampoNumerico('sobre2_inf_bancada','Medida Inferior',true,"",100,50);	2
\$fu- >AgregarCampoNumerico('sobre3_sup_bancada','Medida Superior',true,"",100,50);	3
\$fu- >AgregarCampoNumerico('sobre3_inf_bancada','Medida Inferior',true,"",100,50);	3
\$fu- >AgregarCampoNumerico('sobre4_sup_bancada','Medida Superior',true,"",100,50);	4
\$fu- >AgregarCampoNumerico('sobre4_inf_bancada','Medida Inferior',true,"",100,50);	4
\$fu- >AgregarCampoNumerico('sobre5_sup_bancada','Medida Superior',true,"",100,50);	5
\$fu- >AgregarCampoNumerico('sobre5_inf_bancada','Medida Inferior',true,"",100,50);	5
\$fu- >AgregarCampoNumerico('sobre6_sup_bancada','Medida Superior',true,"",100,50);	6
\$fu- >AgregarCampoNumerico('sobre6_inf_bancada','Medida Inferior',true,"",100,50);	6
\$fu- >AgregarCampoNumerico('torquebancada_motor','Torque Bancada',false,"",100,50);	
\$fu- >AgregarCampoNumerico('std_inf_alojabancada_motor','Std Superior',true,"",100,50);	

```

    $fu-
>AgregarCampoNumerico('std_sup_alojabancada2_motor','Std
Inferior',true,"",100,50);
    $fu-
>AgregarCampoTexto('castingnum_cilindro','Castingnum
Cilindro',false,"",100,50);
    $fu-
>AgregarCampoTexto('comments_cilindro','Comments Cilindro',true,"",100,50);
    $fu-
>AgregarCampoNumerico('std_sup_cilindro','Std Superior',true,"",100,50);
    $fu-
>AgregarCampoNumerico('std_inf_cilindro','Std Inferior',true,"",100,50);
    $fu-
>AgregarCampoNumerico('sobre1_sup_cilindro','Medida
Superior',true,"",100,50);
    $fu-
>AgregarCampoNumerico('sobre1_inf_cilindro','Medida
Inferior',true,"",100,50);
    $fu-
>AgregarCampoNumerico('sobre2_sup_cilindro','Medida
Superior',true,"",100,50);
    $fu-
>AgregarCampoNumerico('sobre2_inf_cilindro','Medida
Inferior',true,"",100,50);
    $fu-
>AgregarCampoNumerico('sobre3_sup_cilindro','Medida
Superior',true,"",100,50);
    $fu-
>AgregarCampoNumerico('sobre3_inf_cilindro','Medida
Inferior',true,"",100,50);
    $fu-
>AgregarCampoNumerico('sobre4_sup_cilindro','Medida
Superior',true,"",100,50);
    $fu-
>AgregarCampoNumerico('sobre4_inf_cilindro','Medida
Inferior',true,"",100,50);
    $fu-
>AgregarCampoNumerico('sobre5_sup_cilindro','Medida
Superior',true,"",100,50);
    $fu-
>AgregarCampoNumerico('sobre5_inf_cilindro','Medida
Inferior',true,"",100,50);
```



```

                $fu-
>AgregarCampoNumerico('sobre6_sup_cilindro','Medida Superior',true,"",100,50);
    
```

6

```

                $fu-
>AgregarCampoNumerico('sobre6_inf_cilindro','Medida Inferior',true,"",100,50);
    
```

6

```

                $sHtml .= '<fieldset>
                <legend class="Titulo">Agregar
Motor</legend>';
    
```

```

                $sHtml .= '<form id="DataGrid">';
                $sHtml .= '<table width="600" border="0"
cellspacing="2" cellpadding="0" class="tabla">
                <tr class="msgFrm">
                <td colspan="4" align="center">Los
campos con * son obligatorios</td>
    
```

```

                </tr>';
    
```

```

                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
    
```

```

                <td colspan="2">' . $fu->ObjetoHtml (
'combustible' ) . '</td>
    
```

```

                </tr>';
    
```

```

                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL ( 'marca' ) .
'</th>
    
```

```

                <td colspan="2">' . $fu->ObjetoHtml (
'marca' ) . '</td>
    
```

```

                </tr>';
    
```

```

                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL ( 'modelo' ) .
'</th>
    
```

```

                <td colspan="2">' . $fu->ObjetoHtml (
'modelo' ) . '</td>
    
```

```

                </tr>';
    
```

```

                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL (
'nombre_motor' ) . '</th>
    
```

```

                <td colspan="2">' . $fu->ObjetoHtml (
'nombre_motor' ) . '</td>
    
```

```

                <th>' . $fu->ObjetoHtmlLBL (
'ano_motor' ) . '</th>
    
```

```

                <td colspan="2">' . $fu->ObjetoHtml (
'ano_motor' ) . '</td>
    
```

```

'ano_motor' ) . '</td>
                                <td colspan="2">' . $fu->ObjetoHtml (
                                </tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL ( 'medida' ) .
'ano_motor' ) . '</th>
                                <td colspan="2">' . $fu->ObjetoHtml (
'ano_motor' ) . '</td>
                                </tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL ( 'cilindraje'
'ano_motor' ) . '</th>
                                <td colspan="2">' . $fu->ObjetoHtml (
'ano_motor' ) . '</td>
                                </tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL (
'ano_motor' ) . '</th>
                                <td colspan="2">' . $fu->ObjetoHtml (
'ano_motor' ) . '</td>
                                </tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL (
'ano_motor' ) . '</th>
                                <th colspan="2">Sup</th>
                                <th colspan="2">Inf</th>
                                </tr>;
                                $sHtml .= '<tr>
                                <td>&nbsp;</td>
                                <td colspan="2">' . $fu->ObjetoHtml (
'ano_motor' ) . '</td>;
                                $sHtml .= '
                                <td colspan="2">' . $fu->ObjetoHtml (
'ano_motor' ) . '</td>
                                </tr>;
                                $sHtml .= '<tr>
                                <th> Alojamiento de Biela </th>
                                <th colspan="2">Sup</th>
                                <th colspan="2">Inf</th>
                                </tr>;
                                $sHtml .= '<tr>
                                <td>&nbsp;</td>
                                <td colspan="2">' . $fu->ObjetoHtml (
'ano_motor' ) . '</td>;

```

```

                $sHtml .= '
                    <td colspan = "2">' . $fu->ObjetoHtml (
'std_sup_alojabiela_motor' ) . '</td>';
                $sHtml .= '<tr>
                    <th>' . $fu->ObjetoHtmlLBL (
'torquebiela_motor' ) . '</th>
                    <td colspan="2">' . $fu->ObjetoHtml (
'torquebiela_motor' ) . '</td>
                    ';
                $sHtml .= '<tr>
                    <th>' . $fu->ObjetoHtmlLBL (
'stroke_ciguenal' ) . '</th>
                    <td colspan="2">' . $fu->ObjetoHtml (
'stroke_ciguenal' ) . '</td>
                    </tr>';
                $sHtml .= '<tr>
                    <th>' . $fu->ObjetoHtmlLBL (
'luzbiela_motor' ) . '</th>
                    <th colspan="2">Sup</th>
                    <th colspan="2">Inf</th>
                    </tr>';
                $sHtml .= '<tr>
                    <td>&nbsp;</td>
                    <td colspan="2">' . $fu->ObjetoHtml (
'luzbiela_motor' ) . '</td>
                    ';
                $sHtml .= '
                    <td colspan = "2">' . $fu->ObjetoHtml (
'luzbiela2_motor' ) . '</td>
                    ';
                $sHtml .= '<tr>
                    <th>' . $fu->ObjetoHtmlLBL (
'luzbancada_motor' ) . '</th>
                    <th colspan="2">Sup</th>
                    <th colspan="2">Inf</th>
                    </tr>';
                $sHtml .= '<tr>
                    <td>&nbsp;</td>
                    <td colspan="2">' . $fu->ObjetoHtml (
'luzbancada_motor' ) . '</td>
                    ';
                $sHtml .= '
                    <td colspan="2">' . $fu->ObjetoHtml (
'luzbancada2_motor' ) . '</td>

```

```

';
    $sHtml .= '<tr>
        <th>' . $fu->ObjetoHtmlLBL (
'castingnum_ciguenal' ) . '</th>
        <td>' . $fu->ObjetoHtml (
'castingnum_ciguenal' ) . '</td>
';
    $sHtml .= '<tr>
        <th>' . $fu->ObjetoHtmlLBL (
'comments_ciguenal' ) . '</th>
        <td>' . $fu->ObjetoHtml (
'comments_ciguenal' ) . '</td>
';
    $sHtml .= '<tr>
        <td rowspan="2"> Biela </td>
        <th>Std</th>
        <th>S1</th>
        <th>S2</th>
        <th>S3</th>
        <th>S4</th>
        <th>S5</th>
        <th>S6</th>
    </tr>';
    $sHtml .= '<tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>';
    $sHtml .= '<tr>
        <th>Sup</th>
        <td>' . $fu->ObjetoHtml ( 'std_sup_biela'
) . '</td>
';
    $sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre1_sup_biela' ) . '</td>
';
    $sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre2_sup_biela' ) . '</td>

```

```

';
$sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre3_sup_biela' ) . '</td>
';
$sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre4_sup_biela' ) . '</td>
';
$sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre5_sup_biela' ) . '</td>
';
$sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre6_sup_biela' ) . '</td>
';
$sHtml .= '<tr>
        <th>Inf</th>
        <td> . $fu->ObjetoHtml ( 'std_inf_biela' )
. '</td>
';
$sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre1_inf_biela' ) . '</td>
';
$sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre2_inf_biela' ) . '</td>
';
$sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre3_inf_biela' ) . '</td>
';
$sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre4_inf_biela' ) . '</td>
';
$sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre5_inf_biela' ) . '</td>
';
$sHtml .= '

```

```

        <td>' . $fu->ObjetoHtml (
'sobre6_inf_biela' ) . '</td>
';
$sHtml .= '<tr>
        <td rowspan="2"> Bancada </td>
        <th>Std</th>
        <th>S1</th>
        <th>S2</th>
        <th>S3</th>
        <th>S4</th>
        <th>S5</th>
        <th>S6</th>
</tr>';
$sHtml .= '<tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
</tr>';
$sHtml .= '<tr>
        <th>Sup</th>
        <td>' . $fu->ObjetoHtml (
'std_sup_bancada' ) . '</td>
';
$sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre1_sup_bancada' ) . '</td>
';
$sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre2_sup_bancada' ) . '</td>
';
$sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre3_sup_bancada' ) . '</td>
';
$sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre4_sup_bancada' ) . '</td>
';
$sHtml .= '

```

```

        <td>'
'sobre5_sup_bancada' ) . '</td>
        $sHtml .= '
        <td>'
'sobre6_sup_bancada' ) . '</td>
        $sHtml .= '<tr>
        <th>Inf</th>
        <td>'
'std_inf_bancada' ) . '</td>
        $sHtml .= '
        <td>'
'sobre1_inf_bancada' ) . '</td>
        $sHtml .= '
        <td>'
'sobre2_inf_bancada' ) . '</td>
        $sHtml .= '
        <td>'
'sobre3_inf_bancada' ) . '</td>
        $sHtml .= '
        <td>'
'sobre4_inf_bancada' ) . '</td>
        $sHtml .= '
        <td>'
'sobre5_inf_bancada' ) . '</td>
        $sHtml .= '
        <td>'
'sobre6_inf_bancada' ) . '</td>
        $sHtml .= '<tr>
        <th>'
'torquebancada_motor' ) . '</th>
        <td>'
'torquebancada_motor' ) . '</td>
        $sHtml .= '<tr>
        <th> Alojamiento Bancada </th>
    
```

```

                <th colspan="2">Sup</th>
                <th colspan="2">Inf</th>
            </tr>;
            $sHtml .= '<tr>
                <td>&nbsp;&nbsp;&nbsp;</td>
                <td colspan="2">' . $fu->ObjetoHtml (
'std_inf_alojabancada_motor' ) . '</td>
            ';
            $sHtml .= '
                <td colspan="2">' . $fu->ObjetoHtml (
'std_sup_alojabancada2_motor' ) . '</td>
            ';
            $sHtml .= '<tr>
                <td rowspan = "2"> Cilindro </td>
                <th>Std</th>
                <th>S1</th>
                <th>S2</th>
                <th>S3</th>
                <th>S4</th>
                <th>S5</th>
                <th>S6</th>
            </tr>;
            $sHtml .= '<tr>
                <td>&nbsp;&nbsp;&nbsp;</td>
                <td>&nbsp;&nbsp;&nbsp;</td>
                <td>&nbsp;&nbsp;&nbsp;</td>
                <td>&nbsp;&nbsp;&nbsp;</td>
                <td>&nbsp;&nbsp;&nbsp;</td>
                <td>&nbsp;&nbsp;&nbsp;</td>
                <td>&nbsp;&nbsp;&nbsp;</td>
            </tr>;
            $sHtml .= '<tr>
                <th>Sup</th>
                <td>' . $fu->ObjetoHtml (
'std_sup_cilindro' ) . '</td>
            ';
            $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre1_sup_cilindro' ) . '</td>
            ';
            $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre2_sup_cilindro' ) . '</td>
            ';

```



```

                $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre3_sup_cilindro' ) . '</td>'
                ';
                $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre4_sup_cilindro' ) . '</td>'
                ';
                $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre5_sup_cilindro' ) . '</td>'
                ';
                $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre6_sup_cilindro' ) . '</td>'
                ';
                $sHtml .= '<tr>
                <th>Inf</th>
                <td>' . $fu->ObjetoHtml (
'std_inf_cilindro' ) . '</td>'
                ';
                $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre1_inf_cilindro' ) . '</td>'
                ';
                $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre2_inf_cilindro' ) . '</td>'
                ';
                $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre3_inf_cilindro' ) . '</td>'
                ';
                $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre4_inf_cilindro' ) . '</td>'
                ';
                $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre5_inf_cilindro' ) . '</td>'
                ';
                $sHtml .= '

```

```

        <td>' . $fu->ObjetoHtml (
'sobre6_inf_cilindro' ) . '</td>
        ';
        $sHtml .= '<tr>
        <th >' . $fu->ObjetoHtmlLBL (
'castingnum_cilindro' ) . '</th>
        <td>' . $fu->ObjetoHtml (
'castingnum_cilindro' ) . '</td>
        ';
        $sHtml .= '<tr>
        <th>' . $fu->ObjetoHtmlLBL (
'comments_cilindro' ) . '</th>
        <td>' . $fu->ObjetoHtml (
'comments_cilindro' ) . '</td>
        ';

        $sHtml .= '</table>';
        $sHtml .= '</form>';
        $sHtml .= '</fieldset>';
        $aBoton[0]['tipo']='button';
        $aBoton[0]['value']='Cancelar';
        $aBoton[0]['id']='cancelar';
        $aBoton[0]['accion']='xajax_genera_motor()';

        $aBoton[1]['tipo']='button';
        $aBoton[1]['value']='Agregar';
        $aBoton[1]['id']='agregar';

        $aBoton[1]['accion']=validar_accion('xajax_procesa_motor(xajax.getForm
Values('\DataGrid\'),1),$aBoton[1]['id']);
        $sHtml .= genera_boton($aBoton);
        break;
    }
    break;
    case 2: //Editar Motor
        switch($tipo)
        {
            case 0:
                $sql = 'SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';
                $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );

```

```

                $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$accion,1)");

                $sHtml .= '<fieldset>
                <legend                                class="Titulo">Editar
Motor</legend>';

                $sHtml .= '<form id="DataGrid">';
                $sHtml .= '<table align="center" cellpadding="0"
cellspacing="2" width="100%" border="0">
                <tr class="msgFrm"><td colspan="4"
align="center">Los campos con * son obligatorios</td></tr>';
                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
                <td colspan="3">' . $fu->ObjetoHtml (
'combustible' ) . '</td>
                </tr>';
                $sHtml .= '</table>';
                $sHtml .= '</form>';
                $sHtml .= '</fieldset>';
                break;//fin case
                case 1:
                $sql = 'SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';
                $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$accion,1)");
                $fu->cCampos['combustible']-
>xValor=$aFrm['combustible'];

                $sql = 'SELECT ID_MARCA,NOMBRE_MARCA
FROM marca WHERE ID_COMBUSTIBLE ='.$aFrm['combustible'].' and estado =
1 ORDER BY NOMBRE_MARCA ASC';
                $fu->AgregarCampoListaSQL ( 'marca', 'Marca',
$sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('marca',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$accion,2)");
                $fu->cCampos['marca']-
>xValor=$aFrm['marca'];

```

```

                $sHtml .= '<fieldset>
                    <legend                                class="Titulo">Editar
Motor</legend>;
                $sHtml .= '<form id="DataGrid">;
                $sHtml .= '<table align="center" cellpadding="0"
cellspacing="2" width="100%" border="0">
                    <tr class="msgFrm"><td colspan="4"
align="center">Los campos con * son obligatorios</td></tr>;
                $sHtml .= '<tr>
                    <th>' . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
                    <td colspan="3">' . $fu->ObjetoHtml (
'combustible' ) . '</td>
                    </tr>;
                $sHtml .= '<tr>
                    <th>' . $fu->ObjetoHtmlLBL ( 'marca' ) .
'</th>
                    <td colspan="3">' . $fu->ObjetoHtml (
'marca' ) . '</td>
                    </tr>;
                $sHtml .= '</table>;
                $sHtml .= '</form>;
                $sHtml .= '</fieldset>;
                break;//fin case
            case 2:
                $sql = 'SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';
                $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$accion,1)");
                $fu->cCampos['combustible']-
>xValor=$aFrm['combustible'];

                $sql = 'SELECT ID_MARCA,NOMBRE_MARCA
FROM marca WHERE ID_COMBUSTIBLE =' . $aFrm['combustible'] . ' and estado =
1 ORDER BY NOMBRE_MARCA ASC';
                $fu->AgregarCampoListaSQL ( 'marca', 'Marca',
$sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('marca',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$accion,2)");

```

```

                                $fu->cCampos['marca']-
>xValor=$aFrm['marca'];

                                $sql ='SELECT ID_MODELO,NOMBRE_MODELO
FROM modelo WHERE ID_MARCA =' . $aFrm['marca'] . ' and estado = 1 ORDER
BY NOMBRE_MODELO ASC';
                                $fu->AgregarCampoListaSQL ( 'modelo',
'Modelo', $sql, true, 'auto' );
                                $fu-
>AgregarComandoAlCambiarValor('modelo',"xajax_genera_motor(xajax.getFor
mValues('DataGrid'),$accion,3)");
                                $fu->cCampos['modelo']-
>xValor=$aFrm['modelo'];

                                $sHtml .= '<fieldset>
                                <legend class="Titulo">Editar
Motor</legend>;
                                $sHtml .= '<form id="DataGrid">;
                                $sHtml .= '<table align="center" cellpadding="0"
cellspacing="2" width="100%" border="0">
                                <tr class="msgFrm"><td colspan="4"
align="center">Los campos con * son obligatorios</td></tr>;
                                $sHtml .= '<tr>
                                <th> . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
                                <td colspan="3">' . $fu->ObjetoHtml (
'combustible' ) . '</td>
                                </tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL ( 'marca' ) .
'</th>
                                <td colspan="3">' . $fu->ObjetoHtml (
'marca' ) . '</td>
                                </tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL ( 'modelo' ) .
'</th>
                                <td colspan="3">' . $fu->ObjetoHtml (
'modelo' ) . '</td>
                                </tr>;

                                $sHtml .= '</table>;
                                $sHtml .= '</form>;
                                $sHtml .= '</fieldset>;

```

```
break;//fin case
case 3:
    $sql = 'SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';
    $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );
    $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$accion,1)");
    $fu->cCampos['combustible']-
>xValor=$aFrm['combustible'];

    $sql = 'SELECT ID_MARCA,NOMBRE_MARCA
FROM marca WHERE ID_COMBUSTIBLE ='. $aFrm['combustible'].' and estado =
1 ORDER BY NOMBRE_MARCA ASC';
    $fu->AgregarCampoListaSQL ( 'marca', 'Marca',
$sql, true, 'auto' );
    $fu-
>AgregarComandoAlCambiarValor('marca',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$accion,2)");
    $fu->cCampos['marca']-
>xValor=$aFrm['marca'];

    $sql = 'SELECT ID_MODELO,NOMBRE_MODELO
FROM modelo WHERE ID_MARCA ='. $aFrm['marca'].' and estado = 1 ORDER
BY NOMBRE_MODELO ASC';
    $fu->AgregarCampoListaSQL ( 'modelo',
'Modelo', $sql, true, 'auto' );
    $fu-
>AgregarComandoAlCambiarValor('modelo',"xajax_genera_motor(xajax.getFor
mValues('DataGrid'),$accion,3)");
    $fu->cCampos['modelo']-
>xValor=$aFrm['modelo'];

    $sql = 'SELECT ID_MOTOR,NOMBRE_MOTOR
FROM motor WHERE ID_MODELO ='. $aFrm['modelo'].' and estado = 1 ORDER
BY NOMBRE_MOTOR ASC';
    $fu->AgregarCampoListaSQL ( 'motor', 'Motor',
$sql, true, 'auto' );
    $fu-
>AgregarComandoAlCambiarValor('motor',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$accion,4)");
```

```

                $fu->cCampos['motor']-
>xValor=$aFrm['motor'];

                $sHtml .= '<fieldset>
                <legend                                class="Titulo">Editar
Motor</legend>';

                $sHtml .= '<form id="DataGrid">';
                $sHtml .= '<table align="center" cellpadding="0"
cellspacing="2" width="100%" border="0">
                <tr class="msgFrm"><td colspan="4"
align="center">Los campos con * son obligatorios</td></tr>';
                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
                <td colspan="3">' . $fu->ObjetoHtml (
'combustible' ) . '</td>
                </tr>';
                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL ( 'marca' ) .
'</th>
                <td colspan="3">' . $fu->ObjetoHtml (
'marca' ) . '</td>
                </tr>';
                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL ( 'modelo' ) .
'</th>
                <td colspan="3">' . $fu->ObjetoHtml (
'modelo' ) . '</td>
                </tr>';
                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL ( 'motor' ) .
'</th>
                <td colspan="3">' . $fu->ObjetoHtml (
'motor' ) . '</td>
                </tr>';

                $sHtml .= '</table>';
                $sHtml .= '</form>';
                $sHtml .= '</fieldset>';
                break;//fin case
                case 4:
                $sql                                ='SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';

```

```

                $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$accion,1)");
                $fu->cCampos['combustible']-
>xValor=$aFrm['combustible'];
                $sql ='SELECT ID_MARCA,NOMBRE_MARCA
FROM marca WHERE ID_COMBUSTIBLE ='.$aFrm['combustible'].' and estado =
1 ORDER BY NOMBRE_MARCA ASC';
                $fu->AgregarCampoListaSQL ( 'marca', 'Marca',
$sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('marca',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$accion,2)");
                $fu->cCampos['marca']-
>xValor=$aFrm['marca'];
                $sql ='SELECT ID_MODELO,NOMBRE_MODELO
FROM modelo WHERE ID_MARCA ='.$aFrm['marca'].' and estado = 1 ORDER
BY NOMBRE_MODELO ASC';
                $fu->AgregarCampoListaSQL ( 'modelo',
'Modelo', $sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('modelo',"xajax_genera_motor(xajax.getFor
mValues('DataGrid'),$accion,3)");
                $fu->cCampos['modelo']-
>xValor=$aFrm['modelo'];
                $sql ='SELECT ID_MOTOR,NOMBRE_MOTOR
FROM motor WHERE ID_MODELO ='.$aFrm['modelo'].' and estado = 1 ORDER
BY NOMBRE_MOTOR ASC';
                $fu->AgregarCampoListaSQL ( 'motor', 'Motor',
$sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('motor',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$accion,4)");
                $fu->cCampos['motor']-
>xValor=$aFrm['motor'];
                $sql ='SELECT * FROM motor WHERE
ID_MOTOR ='.$aFrm['motor'].' and estado = 1 ORDER BY ano_MOTOR ASC';
                if($oCon->Query($sql))
                {
                    $fu-
>AgregarCampoOculto('motor',$aFrm['motor']);

```



```

                $fu-
>AgregarCampoOculto('medida',$oCon->f('medida'));
                $fu-
>AgregarCampoTexto('nombre_motor','Nombre',true,$oCon-
>f('nombre_motor'), 100,50 );
                $fu-
>AgregarCampoNumerico('ano_motor','Año',true,$oCon->f('ano_motor'), 100,50
);
                $fu-
>AgregarCampoTexto('cilindraje','Cilindraje',true,$oCon->f('cilindraje'), 100,50 );
                $fu-
>AgregarCampoTexto('piston_motor','Pistón',true,$oCon->f('piston_motor'),
100,50 );
                $fu-
>AgregarCampoTexto('brazo_biela_motor','Brazo de Biela',true,$oCon-
>f('brazo_biela_motor'), 100,50 );
                $fu-
>AgregarCampoTexto('brazo_biela2_motor','',true,$oCon-
>f('brazo_biela2_motor'), 100,50 );
                $fu-
>AgregarCampoNumerico('std_inf_alojabiela','Std Inferior ',true,$oCon-
>f('std_inf_alojabiela'), 100,50 );
                $fu-
>AgregarCampoNumerico('std_sup_alojabiela','Std Superior',true,$oCon-
>f('std_sup_alojabiela'), 100,50 );
                $fu-
>AgregarCampoNumerico('torquebiela_motor','Torque Biela',true,$oCon-
>f('torquebiela_motor'), 100,50 );
                $fu-
>AgregarCampoNumerico('stroke_ciguenal','Stroke Ciguenal',true,$oCon-
>f('stroke_ciguenal'), 100,50 );
                $fu-
>AgregarCampoNumerico('luzbiela_motor','Luz Biela',true,$oCon-
>f('luzbiela_motor'), 100,50 );
                $fu-
>AgregarCampoNumerico('luzbiela2_motor','',true,$oCon->f('luzbiela2_motor'),
100,50 );
                $fu-
>AgregarCampoNumerico('luzbancada_motor','Luz Bancada',true,$oCon-
>f('luzbancada_motor'), 100,50 );
                $fu-
>AgregarCampoNumerico('luzbancada2_motor','',true,$oCon-
>f('luzbancada2_motor'), 100,50 );
```

```

                                $fu-
>AgregarCampoTexto('castingnum_ciguenal','Castingnum
Ciguenal',true,$oCon->f('castingnum_ciguenal'), 100,50 );
                                $fu-
>AgregarCampoTexto('comments_ciguenal','Comments Ciguenal',true,$oCon-
>f('comments_ciguenal'), 100,50 );
                                $fu-
>AgregarCampoNumerico('std_sup_biela','Standar Superior',true,$oCon-
>f('std_sup_biela'), 100,50 );
                                $fu-
>AgregarCampoNumerico('std_inf_biela','Standar Inferior',true,$oCon-
>f('std_inf_biela'), 100,50 );
                                $fu-
>AgregarCampoNumerico('sobre1_sup_biela','Medida 1 Superior',true,$oCon-
>f('sobre1_sup_biela'), 100,50 );
                                $fu-
>AgregarCampoNumerico('sobre1_inf_biela','Medida 1 Inferior',true,$oCon-
>f('sobre1_inf_biela'), 100,50 );
                                $fu-
>AgregarCampoNumerico('sobre2_sup_biela','Medida 2 Superior',true,$oCon-
>f('sobre2_sup_biela'), 100,50 );
                                $fu-
>AgregarCampoNumerico('sobre2_inf_biela','Medida 2 Inferior',true,$oCon-
>f('sobre2_inf_biela'), 100,50 );
                                $fu-
>AgregarCampoNumerico('sobre3_sup_biela','Medida 3 Superior',true,$oCon-
>f('sobre3_sup_biela'), 100,50 );
                                $fu-
>AgregarCampoNumerico('sobre3_inf_biela','Medida 3 Inferior',true,$oCon-
>f('sobre3_inf_biela'), 100,50 );
                                $fu-
>AgregarCampoNumerico('sobre4_sup_biela','Medida 4 Superior',true,$oCon-
>f('sobre4_sup_biela'), 100,50 );
                                $fu-
>AgregarCampoNumerico('sobre4_inf_biela','Medida 4 Inferior',true,$oCon-
>f('sobre4_inf_biela'), 100,50 );
                                $fu-
>AgregarCampoNumerico('sobre5_sup_biela','Medida 5 Superior',true,$oCon-
>f('sobre5_sup_biela'), 100,50 );
                                $fu-
>AgregarCampoNumerico('sobre5_inf_biela','Medida 5 Inferior',true,$oCon-
>f('sobre5_inf_biela'), 100,50 );
```

```

                $fu-
>AgregarCampoNumerico('sobre6_sup_biela','Medida 6 Superior',true,$oCon-
>f('sobre6_sup_biela'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre6_inf_biela','Medida 6 Inferior',true,$oCon-
>f('sobre6_inf_biela'), 100,50 );
                $fu-
>AgregarCampoNumerico('std_sup_bancada','Std Superior',true,$oCon-
>f('std_sup_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('std_inf_bancada','Std Inferior',true,$oCon-
>f('std_inf_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre1_sup_bancada','Medida Superior',true,$oCon->f('sobre1_sup_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre1_inf_bancada','Medida 1 Inferior',true,$oCon-
>f('sobre1_inf_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre2_sup_bancada','Medida Superior',true,$oCon->f('sobre2_sup_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre2_inf_bancada','Medida 2 Inferior',true,$oCon-
>f('sobre2_inf_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre3_sup_bancada','Medida Superior',true,$oCon->f('sobre3_sup_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre3_inf_bancada','Medida 3 Inferior',true,$oCon-
>f('sobre3_inf_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre4_sup_bancada','Medida Superior',true,$oCon->f('sobre4_sup_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre4_inf_bancada','Medida 4 Inferior',true,$oCon-
>f('sobre4_inf_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre5_sup_bancada','Medida Superior',true,$oCon->f('sobre5_sup_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre5_inf_bancada','Medida 5 Inferior',true,$oCon-
>f('sobre5_inf_bancada'), 100,50 );
```

```

                $fu-
>AgregarCampoNumerico('sobre6_sup_bancada','Medida
Superior',true,$oCon->f('sobre6_sup_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre6_inf_bancada','Medida 6 Inferior',true,$oCon-
>f('sobre6_inf_bancada'), 100,50 );
                $fu-
>AgregarCampoNumerico('torquebancada_motor','Torque
Bancada',true,$oCon->f('torquebancada_motor'), 100,50 );
                $fu-
>AgregarCampoNumerico('std_inf_alojabancada_motor','Std
Superior',true,$oCon->f('std_inf_alojabancada_motor'), 100,50 );
                $fu-
>AgregarCampoNumerico('std_sup_alojabancada2_motor','Std
Inferior',true,$oCon->f('std_sup_alojabancada2_motor'), 100,50 );
                $fu-
>AgregarCampoTexto('castingnum_cilindro','Castingnum Cilindro',true,$oCon-
>f('castingnum_cilindro'), 100,50 );
                $fu-
>AgregarCampoTexto('comments_cilindro','Comments Cilindro',true,$oCon-
>f('comments_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('std_sup_cilindro','Std Superior',true,$oCon-
>f('std_sup_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('std_inf_cilindro','Std Inferior',true,$oCon-
>f('std_inf_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre1_sup_cilindro','Medida
Superior',true,$oCon->f('sobre1_sup_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre1_inf_cilindro','Medida 1 Inferior',true,$oCon-
>f('sobre1_inf_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre2_sup_cilindro','Medida
Superior',true,$oCon->f('sobre2_sup_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre2_inf_cilindro','Medida 2 Inferior',true,$oCon-
>f('sobre2_inf_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre3_sup_cilindro','Medida
Superior',true,$oCon->f('sobre3_sup_cilindro'), 100,50 );

```

```

                $fu-
>AgregarCampoNumerico('sobre3_inf_cilindro','Medida 3 Inferior',true,$oCon-
>f('sobre3_inf_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre4_sup_cilindro','Medida 4 Superior',true,$oCon->f('sobre4_sup_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre4_inf_cilindro','Medida 4 Inferior',true,$oCon-
>f('sobre4_inf_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre5_sup_cilindro','Medida 5 Superior',true,$oCon->f('sobre5_sup_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre5_inf_cilindro','Medida 5 Inferior',true,$oCon-
>f('sobre5_inf_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre6_sup_cilindro','Medida 6 Superior',true,$oCon->f('sobre6_sup_cilindro'), 100,50 );
                $fu-
>AgregarCampoNumerico('sobre6_inf_cilindro','Medida 6 Inferior',true,$oCon-
>f('sobre6_inf_cilindro'), 100,50 );

                $$Html .= '<fieldset>
                <legend class="Titulo">Editar
Motor</legend>';

                $$Html .= '<form id="DataGrid">';
                $$Html .= $fu->ObjetoHtml ( 'motor' );
                $$Html .= $fu->ObjetoHtml ( 'medida' );
                $$Html .= '<table width="600" border="0"
cellspacing="2" cellpadding="0" class="tabla">
                <tr class="msgFrm">
                <td colspan="4" align="center">Los
campos con * son obligatorios</td>
                </tr>';
                $$Html .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
                <td colspan="2">' . $fu->ObjetoHtml (
'combustible' ) . '</td>
                </tr>';
                $$Html .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL ( 'marca' ) .
'</th>

```

```

'marca' ) . '</td>
                                <td colspan="2">' . $fu->ObjetoHtml (
                                </tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL ( 'modelo' ) .
'</th>
                                <td colspan="2">' . $fu->ObjetoHtml (
'modelo' ) . '</td>
                                </tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL (
'nombre_motor' ) . '</th>
                                <td colspan="2">' . $fu->ObjetoHtml (
'nombre_motor' ) . '</td>
                                <th>' . $fu->ObjetoHtmlLBL (
'ano_motor' ) . '</th>
                                <td colspan="2">' . $fu->ObjetoHtml (
'ano_motor' ) . '</td>
                                </tr>;
                                if ($fu->ObjetoHtml ( 'medida' ) == 0){
                                    $valormedida ='Pulgadas';
                                }elseif ($fu->ObjetoHtml ( 'medida' ) ==
1){
                                    $valormedida ='Milimetros';
                                }
                                $sHtml .= '<tr>
                                <th> Medida :</th>
                                <td colspan="2">' . $valormedida . '</td>
                                </tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL ( 'cilindraje'
) . '</th>
                                <td colspan="2">' . $fu->ObjetoHtml (
'cilindraje' ) . '</td>
                                </tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL (
'piston_motor' ) . '</th>
                                <td colspan="2">' . $fu->ObjetoHtml (
'piston_motor' ) . '</td>
                                </tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL (
'brazo_biela_motor' ) . '</th>

```

```

                <th colspan="2">Sup</th>
                <th colspan="2">Inf</th>
            </tr>;
            $sHtml .= '<tr>
                <td>&nbsp;</td>
                <td colspan="2">' . $fu->ObjetoHtml (
' brazo_biela_motor' ) . '</td>';
            $sHtml .= '
                <td colspan="2">' . $fu->ObjetoHtml (
' brazo_biela2_motor' ) . '</td>
            </tr>';
            $sHtml .= '<tr>
                <th> Alojamiento de Biela </th>
                <th colspan="2">Sup</th>
                <th colspan="2">Inf</th>
            </tr>';
            $sHtml .= '<tr>
                <td>&nbsp;</td>
                <td colspan = "2">' . $fu->ObjetoHtml (
' std_inf_alojabiela' ) . '</td>';
            $sHtml .= '
                <td colspan = "2">' . $fu->ObjetoHtml (
' std_sup_alojabiela' ) . '</td>';

            $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL (
' torquebiela_motor' ) . '</th>
                <td colspan="2">' . $fu->ObjetoHtml (
' torquebiela_motor' ) . '</td>
            ';
            $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL (
' stroke_ciguenal' ) . '</th>
                <td colspan="2">' . $fu->ObjetoHtml (
' stroke_ciguenal' ) . '</td>
            </tr>';
            $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL (
' luzbiela_motor' ) . '</th>
                <th colspan="2">Sup</th>
                <th colspan="2">Inf</th>
            </tr>';
            $sHtml .= '<tr>
                <td>&nbsp;</td>

```

```

'luzbiela_motor' ) . '</td>
                                <td colspan="2">' . $fu->ObjetoHtml (
                                ';
                                $sHtml .= '
                                <td colspan = "2">' . $fu->ObjetoHtml (
'luzbiela2_motor' ) . '</td>
                                ';
                                $sHtml .= '<tr>
                                <th> ' . $fu->ObjetoHtmlLBL (
'luzbancada_motor' ) . ' </th>
                                <th colspan="2">Sup</th>
                                <th colspan="2">Inf</th>
                                </tr>';
                                $sHtml .= '<tr>
                                <td>&nbsp;</td>
                                <td colspan="2">' . $fu->ObjetoHtml (
'luzbancada_motor' ) . '</td>
                                ';
                                $sHtml .= '
                                <td colspan="2">' . $fu->ObjetoHtml (
'luzbancada2_motor' ) . '</td>
                                ';
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL (
'castingnum_ciguenal' ) . '</th>
                                <td>' . $fu->ObjetoHtml (
'castingnum_ciguenal' ) . '</td>
                                ';
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL (
'comments_ciguenal' ) . '</th>
                                <td>' . $fu->ObjetoHtml (
'comments_ciguenal' ) . '</td>
                                ';
                                $sHtml .= '<tr>
                                <td rowspan="2"> Biela </td>
                                <th>Std</th>
                                <th>S1</th>
                                <th>S2</th>
                                <th>S3</th>
                                <th>S4</th>
                                <th>S5</th>
                                <th>S6</th>
                                </tr>';

```



```

                $sHtml .= '<tr>
                    <td>&nbsp;</td>
                    <td>&nbsp;</td>
                    <td>&nbsp;</td>
                    <td>&nbsp;</td>
                    <td>&nbsp;</td>
                    <td>&nbsp;</td>
                    <td>&nbsp;</td>
                </tr>';
                $sHtml .= '<tr>
                    <th>Sup</th>
                    <td>' . $fu->ObjetoHtml ( 'std_sup_biela'
) . '</td>
                ';
                $sHtml .= '
                    <td>' . $fu->ObjetoHtml (
'sobre1_sup_biela' ) . '</td>
                ';
                $sHtml .= '
                    <td>' . $fu->ObjetoHtml (
'sobre2_sup_biela' ) . '</td>
                ';
                $sHtml .= '
                    <td>' . $fu->ObjetoHtml (
'sobre3_sup_biela' ) . '</td>
                ';
                $sHtml .= '
                    <td>' . $fu->ObjetoHtml (
'sobre4_sup_biela' ) . '</td>
                ';
                $sHtml .= '
                    <td>' . $fu->ObjetoHtml (
'sobre5_sup_biela' ) . '</td>
                ';
                $sHtml .= '
                    <td>' . $fu->ObjetoHtml (
'sobre6_sup_biela' ) . '</td>
                ';
                $sHtml .= '<tr>
                    <th>Inf</th>
                    <td>' . $fu->ObjetoHtml ( 'std_inf_biela' )
                . '</td>
                ';
                $sHtml .= '

```

```

        <td>' . $fu->ObjetoHtml (
'sobre1_inf_biela' ) . '</td>
';
$sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre2_inf_biela' ) . '</td>
';
$sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre3_inf_biela' ) . '</td>
';
$sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre4_inf_biela' ) . '</td>
';
$sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre5_inf_biela' ) . '</td>
';
$sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre6_inf_biela' ) . '</td>
';
$sHtml .= '<tr>
        <td rowspan="2"> Bancada </td>
        <th>Std</th>
        <th>S1</th>
        <th>S2</th>
        <th>S3</th>
        <th>S4</th>
        <th>S5</th>
        <th>S6</th>
</tr>';
$sHtml .= '<tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
</tr>';
$sHtml .= '<tr>
        <th>Sup</th>

```

```

                <td>'
'std_sup_bancada' ) . '</td>
                '
                $sHtml .= '
                <td>'
'sobre1_sup_bancada' ) . '</td>
                '
                $sHtml .= '
                <td>'
'sobre2_sup_bancada' ) . '</td>
                '
                $sHtml .= '
                <td>'
'sobre3_sup_bancada' ) . '</td>
                '
                $sHtml .= '
                <td>'
'sobre4_sup_bancada' ) . '</td>
                '
                $sHtml .= '
                <td>'
'sobre5_sup_bancada' ) . '</td>
                '
                $sHtml .= '
                <td>'
'sobre6_sup_bancada' ) . '</td>
                '
                $sHtml .= '<tr>
                <th>Inf</th>
                <td>'
'std_inf_bancada' ) . '</td>
                '
                $sHtml .= '
                <td>'
'sobre1_inf_bancada' ) . '</td>
                '
                $sHtml .= '
                <td>'
'sobre2_inf_bancada' ) . '</td>
                '
                $sHtml .= '
                <td>'
'sobre3_inf_bancada' ) . '</td>
                '
                '
    
```

```

                $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre4_inf_bancada' ) . '</td>'
                ';
                $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre5_inf_bancada' ) . '</td>'
                ';
                $sHtml .= '
                <td>' . $fu->ObjetoHtml (
'sobre6_inf_bancada' ) . '</td>'
                ';
                $sHtml .= '<tr>'
                <th>' . $fu->ObjetoHtmlLBL (
'torquebancada_motor' ) . '</th>'
                <td>' . $fu->ObjetoHtml (
'torquebancada_motor' ) . '</td>'
                ';
                $sHtml .= '<tr>'
                <th> Alojamiento Bancada </th>
                <th colspan="2">Sup</th>
                <th colspan="2">Inf</th>
                </tr>';
                $sHtml .= '<tr>'
                <td>&nbsp;</td>
                <td colspan="2">' . $fu->ObjetoHtml (
'std_inf_alojabancada_motor' ) . '</td>'
                ';
                $sHtml .= '
                <td colspan="2">' . $fu->ObjetoHtml (
'std_sup_alojabancada2_motor' ) . '</td>'
                ';
                $sHtml .= '<tr>'
                <td rowspan = "2"> Cilindro </td>
                <th>Std</th>
                <th>S1</th>
                <th>S2</th>
                <th>S3</th>
                <th>S4</th>
                <th>S5</th>
                <th>S6</th>
                </tr>';
                $sHtml .= '<tr>'
                <td>&nbsp;</td>

```

```

        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>;
    $sHtml .= '<tr>
        <th>Sup</th>
        <td>' . $fu->ObjetoHtml (
'std_sup_cilindro' ) . '</td>
';
    $sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre1_sup_cilindro' ) . '</td>
';
    $sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre2_sup_cilindro' ) . '</td>
';
    $sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre3_sup_cilindro' ) . '</td>
';
    $sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre4_sup_cilindro' ) . '</td>
';
    $sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre5_sup_cilindro' ) . '</td>
';
    $sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre6_sup_cilindro' ) . '</td>
';
    $sHtml .= '<tr>
        <th>Inf</th>
        <td>' . $fu->ObjetoHtml (
'std_inf_cilindro' ) . '</td>
';
    $sHtml .= '
        <td>' . $fu->ObjetoHtml (
'sobre1_inf_cilindro' ) . '</td>

```

```

';
    $sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre2_inf_cilindro' ) . '</td>
';
    $sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre3_inf_cilindro' ) . '</td>
';
    $sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre4_inf_cilindro' ) . '</td>
';
    $sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre5_inf_cilindro' ) . '</td>
';
    $sHtml .= '
        <td> . $fu->ObjetoHtml (
'sobre6_inf_cilindro' ) . '</td>
';
    $sHtml .= '<tr>
        <th >' . $fu->ObjetoHtmlLBL (
        <td> . $fu->ObjetoHtml (
'scastingnum_cilindro' ) . '</th>
        <td> . $fu->ObjetoHtml (
'scastingnum_cilindro' ) . '</td>
';
    $sHtml .= '<tr>
        <th>' . $fu->ObjetoHtmlLBL (
'scomments_cilindro' ) . '</th>
        <td> . $fu->ObjetoHtml (
'scomments_cilindro' ) . '</td>
';
    $sHtml .= '</table>';
    $sHtml .= '</form>';
    $sHtml .= '</fieldset>';
} //fin IF
$oCon->Free();
$aBoton[0]['tipo']='button';
$aBoton[0]['value']='Cancelar';
$aBoton[0]['id']='cancelar';
$aBoton[0]['accion']='xajax_genera_motor()';

```

```

                $aBoton[1]['tipo']='button';
                $aBoton[1]['value']='Editar';
                $aBoton[1]['id']='editar';

                $aBoton[1]['accion']=validar_accion('xajax_procesa_motor(xajax.getForm
Values(\'DataGrid\'),2),$aBoton[1]['id']);
                $sHtml .= genera_boton($aBoton);
                break;//fin case
            }
            break;
            case 3: //Eliminar Motor
            switch($tipo)
            {
                case 0:
                    $sql = 'SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';
                    $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );
                    $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$accion,1)");

                    $sHtml .= '<fieldset>
<legend class="Titulo">Eliminar
Motor</legend>';
                    $sHtml .= '<form id="DataGrid">';
                    $sHtml .= '<table align="center" cellpadding="0"
cellspacing="2" width="100%" border="0">
<tr class="msgFrm"><td colspan="4"
align="center">Los campos con * son obligatorios</td></tr>';
                    $sHtml .= '<tr>
<th> . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
<td colspan="3">' . $fu->ObjetoHtml (
'combustible' ) . '</td>
</tr>';
                    $sHtml .= '</table>';
                    $sHtml .= '</form>';
                    $sHtml .= '</fieldset>';
                break;//fin case
            case 1:

```

```

                                $sql                                ='SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';
                                $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );
                                $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$accion,1)");
                                $fu->cCampos['combustible']-
>xValor=$aFrm['combustible'];

                                $sql ='SELECT ID_MARCA,NOMBRE_MARCA
FROM marca WHERE ID_COMBUSTIBLE =' . $aFrm['combustible'] . ' and estado =
1 ORDER BY NOMBRE_MARCA ASC';
                                $fu->AgregarCampoListaSQL ( 'marca', 'Marca',
$sql, true, 'auto' );
                                $fu-
>AgregarComandoAlCambiarValor('marca',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$accion,2)");
                                $fu->cCampos['marca']-
>xValor=$aFrm['marca'];

                                $sHtml .= '<fieldset>
                                <legend                                class="Titulo">Eliminar
Motor</legend>;
                                $sHtml .= '<form id="DataGrid">;
                                $sHtml .= '<table align="center" cellpadding="0"
cellspacing="2" width="100%" border="0">
                                <tr class="msgFrm"><td colspan="4"
align="center">Los campos con * son obligatorios</td></tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
                                <td colspan="3">' . $fu->ObjetoHtml (
'combustible' ) . '</td>
                                </tr>;
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL ( 'marca' ) .
'</th>
                                <td colspan="3">' . $fu->ObjetoHtml (
'marca' ) . '</td>
                                </tr>;
                                $sHtml .= '</table>;
                                $sHtml .= '</form>;
                                $sHtml .= '</fieldset>;

```



```

                break;//fin case
            case 2:
                $sql = 'SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';
                $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$accion,1)");
                $fu->cCampos['combustible']-
>xValor=$aFrm['combustible'];

                $sql = 'SELECT ID_MARCA,NOMBRE_MARCA
FROM marca WHERE ID_COMBUSTIBLE ='.$aFrm['combustible'].' and estado =
1 ORDER BY NOMBRE_MARCA ASC';
                $fu->AgregarCampoListaSQL ( 'marca', 'Marca',
$sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('marca',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$accion,2)");
                $fu->cCampos['marca']-
>xValor=$aFrm['marca'];

                $sql = 'SELECT ID_MODELO,NOMBRE_MODELO
FROM modelo WHERE ID_MARCA ='.$aFrm['marca'].' and estado = 1 ORDER
BY NOMBRE_MODELO ASC';
                $fu->AgregarCampoListaSQL ( 'modelo',
'Modelo', $sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('modelo',"xajax_genera_motor(xajax.getFor
mValues('DataGrid'),$accion,3)");
                $fu->cCampos['modelo']-
>xValor=$aFrm['modelo'];

                $sHtml .= '<fieldset>
<legend class="Titulo">Eliminar
Motor</legend>';
                $sHtml .= '<form id="DataGrid">';
                $sHtml .= '<table align="center" cellpadding="0"
cellspacing="2" width="100%" border="0">
<tr class="msgFrm"><td colspan="4"
align="center">Los campos con * son obligatorios</td></tr>';
                $sHtml .= '<tr>

```



```

                                $fu->cCampos['marca']-
>xValor=$aFrm['marca'];

                                $sql ='SELECT ID_MODELO,NOMBRE_MODELO
FROM modelo WHERE ID_MARCA ='.$aFrm['marca'].' and estado = 1 ORDER
BY NOMBRE_MODELO ASC';
                                $fu->AgregarCampoListaSQL ( 'modelo',
'Modelo', $sql, true, 'auto' );
                                $fu-
>AgregarComandoAlCambiarValor('modelo',"xajax_genera_motor(xajax.getForm
mValues('DataGrid'),$accion,3)");
                                $fu->cCampos['modelo']-
>xValor=$aFrm['modelo'];

                                $sql ='SELECT ID_MOTOR,NOMBRE_MOTOR
FROM motor WHERE ID_MODELO ='.$aFrm['modelo'].' and estado = 1 ORDER
BY NOMBRE_MOTOR ASC';
                                $fu->AgregarCampoListaSQL ( 'motor', 'Motor',
$sql, true, 'auto' );
                                $fu-
>AgregarComandoAlCambiarValor('motor',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$accion,4)");
                                $fu->cCampos['motor']-
>xValor=$aFrm['motor'];

                                $sHtml .= '<fieldset>
                                <legend class="Titulo">Eliminar
Motor</legend>';
                                $sHtml .= '<form id="DataGrid">';
                                $sHtml .= '<table align="center" cellpadding="0"
cellspacing="2" width="100%" border="0">
                                <tr class="msgFrm"><td colspan="4"
align="center">Los campos con * son obligatorios</td></tr>';
                                $sHtml .= '<tr>
                                <th> . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
                                <td colspan="3">' . $fu->ObjetoHtml (
'combustible' ) . '</td>
                                </tr>';
                                $sHtml .= '<tr>
                                <th>' . $fu->ObjetoHtmlLBL ( 'marca' ) .
'</th>
                                <td colspan="3">' . $fu->ObjetoHtml (
'marca' ) . '</td>

```

```

        </tr>;
        $sHtml .= '<tr>
        <th>' . $fu->ObjetoHtmlLBL ( 'modelo' ) .
'</th>
        <td colspan="3">' . $fu->ObjetoHtml (
'modelo' ) . '</td>
        </tr>;
        $sHtml .= '<tr>
        <th>' . $fu->ObjetoHtmlLBL ( 'motor' ) .
'</th>
        <td colspan="3">' . $fu->ObjetoHtml (
'motor' ) . '</td>
        </tr>;

        $sHtml .= '</table>';
        $sHtml .= '</form>';
        $sHtml .= '</fieldset>';
        break;//fin case
        case 4:
            $sql = 'SELECT
ID_COMBUSTIBLE,NOMBRE_COMBUSTIBLE FROM combustible where estado
= 1 ORDER BY NOMBRE_COMBUSTIBLE ASC';
            $fu->AgregarCampoListaSQL ( 'combustible',
'Combustible', $sql, true, 'auto' );
            $fu-
>AgregarComandoAlCambiarValor('combustible',"xajax_genera_motor(xajax.ge
tFormValues('DataGrid'),$saccion,1)");
            $fu->cCampos['combustible']-
>xValor=$aFrm['combustible'];

            $sql = 'SELECT ID_MARCA,NOMBRE_MARCA
FROM marca WHERE ID_COMBUSTIBLE =' . $aFrm['combustible'] . ' and estado =
1 ORDER BY NOMBRE_MARCA ASC';
            $fu->AgregarCampoListaSQL ( 'marca', 'Marca',
$sql, true, 'auto' );
            $fu-
>AgregarComandoAlCambiarValor('marca',"xajax_genera_motor(xajax.getForm
Values('DataGrid'),$saccion,2)");
            $fu->cCampos['marca']-
>xValor=$aFrm['marca'];

            $sql = 'SELECT ID_MODELO,NOMBRE_MODELO
FROM modelo WHERE ID_MARCA =' . $aFrm['marca'] . ' and estado = 1 ORDER
BY NOMBRE_MODELO ASC';

```

```

                $fu->AgregarCampoListaSQL ( 'modelo',
'Modelo', $sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('modelo',"xajax_genera_motor(xajax.getFormValues('DataGrid'),$accion,3)");
                $fu->cCampos['modelo']-
>xValor=$aFrm['modelo'];

                $sql ='SELECT ID_MOTOR,NOMBRE_MOTOR
FROM motor WHERE ID_MODELO =' . $aFrm['modelo'] . ' and estado = 1 ORDER
BY NOMBRE_MOTOR ASC';
                $fu->AgregarCampoListaSQL ( 'motor', 'Motor',
$sql, true, 'auto' );
                $fu-
>AgregarComandoAlCambiarValor('motor',"xajax_genera_motor(xajax.getFormValues('DataGrid'),$accion,4)");
                $fu->cCampos['motor']-
>xValor=$aFrm['motor'];

                $sHtml .= '<fieldset>
                <legend class="Titulo">Consultar
Motor</legend>';

                $sHtml .= '<form id="DataGrid">';
                $sHtml .= '<table align="center"
cellpadding="0" cellspacing="2" width="100%" border="0">
                <tr class="msgFrm"><td
colspan="4" align="center">Los campos con * son obligatorios</td></tr>';
                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL (
'combustible' ) . '</th>
                <td colspan="3">' . $fu-
>ObjetoHtml ( 'combustible' ) . '</td>
                </tr>';
                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL (
'marca' ) . '</th>
                <td colspan="3">' . $fu-
>ObjetoHtml ( 'marca' ) . '</td>
                </tr>';
                $sHtml .= '<tr>
                <th>' . $fu->ObjetoHtmlLBL (
'modelo' ) . '</th>
                <td colspan="3">' . $fu-
>ObjetoHtml ( 'modelo' ) . '</td>

```

```

        </tr>;
        $sHtml .= '<tr>
        <th>' . $fu->ObjetoHtmlLBL (
'motor' ) . '</th>
        <td colspan="3">' . $fu-
>ObjetoHtml ( 'motor' ) . '</td>
        </tr>;

        $sHtml .= '</table>;
        $sHtml .= '</form>;
        $sHtml .= '</fieldset>;
        $aBoton[0]['tipo']='button';
        $aBoton[0]['value']='Cancelar';
        $aBoton[0]['id']='cancelar';
        $aBoton[0]['accion']='xajax_genera_motor()';

        $aBoton[1]['tipo']='button';
        $aBoton[1]['value']='Eliminar';
        $aBoton[1]['id']='eliminar';
        $aBoton[1]['accion']='confirma_accion('Esta
seguro que desea
Eliminar?','xajax_procesa_motor(xajax.getFormValues('\DataGrid'),3);',$aBoton
[1]['id'],true);
        $sHtml .= genera_boton($aBoton);
        break;//fin case
    }
    break;//fin case
}
}
$oReturn->assign ( "General", "innerHTML", $sHtml );
return $oReturn;
}
/*.....*/
/*
@ cob05 :: procesa_ruta
@ Procesa la informacion generada por la func.ion grid
y tambien de la cabecera de ese datagrid
*/
function procesa_motor($aFrm = "", $bandera){
    global $DSN;
    session_start();
    $aError = array('No se pudo generar el # de Ruta, por favor vuelva a
Intentarlo',
        'No se pudo guardar el detalle, por favor vuelva a Intentarlo',

```

```
'No se pudo obtener el Id de la Cabecera, por favor vuelva a
Intentarlo',
'El # de Ruta esta duplicado, por favor vuelva a Intentarlo',
'La Fecha del Diario no puede ser menor a la de Hoy, por
favor cambiala y vuelva a Intentarlo',
'Los datos para el ingreso estan vacios!!!, por favor
verifiquelos y vuelva a Intentarlo');
$my = new Dbo;
$my->DSN = $DSN;
$return = new xajaxResponse();
$my->Conectar();
switch ($bandera){
    case 1:
        if (verifica_motor($aFrm['nombre_motor'])){
            $return->alert('El motor: '.$aFrm['nombre_motor'].'
ya existe!!!');
        }else{
            $sql = 'INSERT INTO MOTOR
(NOMBRE_MOTOR,CILINDRAJE,ANO_MOTOR,PISTON_MOTOR,BRAZO_BIELA
_MOTOR,BRAZO_BIELA2_MOTOR,STD_INF_ALOJABIELA,
STD_SUP_ALOJABIELA,TORQUEBIELA_MOTOR,STROKE_CIGUENAL,L
UZBIELA_MOTOR,LUZBIELA2_MOTOR,LUZBANCADA_MOTOR,LUZBANCADA
2_MOTOR,
CASTINGNUM_CIGUENAL,COMMENTS_CIGUENAL,STD_SUP_BIELA,ST
D_INF_BIELA,SOBRE1_SUP_BIELA,SOBRE1_INF_BIELA,SOBRE2_SUP_BIELA,
SOBRE2_INF_BIELA,SOBRE3_SUP_BIELA,SOBRE3_INF_BIELA,SOBRE4
_SUP_BIELA,SOBRE4_INF_BIELA,SOBRE5_SUP_BIELA,SOBRE5_INF_BIELA,
SOBRE6_SUP_BIELA,SOBRE6_INF_BIELA,STD_SUP_BANCADA,STD_IN
F_BANCADA,SOBRE1_SUP_BANCADA,SOBRE1_INF_BANCADA,SOBRE2_SU
P_BANCADA,
SOBRE2_INF_BANCADA,SOBRE3_SUP_BANCADA,SOBRE3_INF_BANC
ADA,SOBRE4_SUP_BANCADA,SOBRE4_INF_BANCADA,SOBRE5_SUP_BANC
ADA,
SOBRE5_INF_BANCADA,SOBRE6_SUP_BANCADA,SOBRE6_INF_BANC
ADA,TORQUEBANCADA_MOTOR,STD_INF_ALOJABANCADA_MOTOR,STD_SU
P_ALOJABANCADA2_MOTOR,
CASTINGNUM_CILINDRO,COMMENTS_CILINDRO,STD_SUP_CILINDRO,S
```

TD\_INF\_CILINDRO,SOBRE1\_SUP\_CILINDRO,SOBRE1\_INF\_CILINDRO,SOBRE2\_SUP\_CILINDRO,

SOBRE2\_INF\_CILINDRO,SOBRE3\_SUP\_CILINDRO,SOBRE3\_INF\_CILINDRO,SOBRE4\_SUP\_CILINDRO,SOBRE4\_INF\_CILINDRO,SOBRE5\_SUP\_CILINDRO,

SOBRE5\_INF\_CILINDRO,SOBRE6\_SUP\_CILINDRO,SOBRE6\_INF\_CILINDRO,MEDIDA,ID\_COMBUSTIBLE,ID\_MARCA,ID\_MODELO,ESTADO)

```
VALUES ( \".${Frm['nombre_motor']}.\",
\".${Frm['cilindraje']}.\", \".${Frm['ano_motor']}.\", \".${Frm['piston_motor']}.\",
\".${Frm['brazo_biela_motor']}.\",
\".${Frm['brazo_biela2_motor']}.\", \".${Frm['std_inf_alojabiela']}.\",
\".${Frm['std_sup_alojabiela']}.\",
\".${Frm['torquebiela_motor']}.\",
\".${Frm['stroke_ciguenal']}.\", \".${Frm['luzbiela_motor']}.\",
\".${Frm['luzbiela2_motor']}.\",
\".${Frm['luzbancada_motor']}.\",
\".${Frm['luzbancada2_motor']}.\", \".${Frm['castingnum_ciguenal']}.\",
\".${Frm['comments_ciguenal']}.\",
\".${Frm['std_sup_biela']}.\",\".${Frm['std_inf_biela']}.\",
\".${Frm['sobre1_sup_biela']}.\",\".${Frm['sobre1_inf_biela']}.\",
\".${Frm['sobre2_sup_biela']}.\",
\".${Frm['sobre2_inf_biela']}.\", \".${Frm['sobre3_sup_biela']}.\",
\".${Frm['sobre3_inf_biela']}.\",
\".${Frm['sobre4_sup_biela']}.\",
\".${Frm['sobre4_inf_biela']}.\", \".${Frm['sobre5_sup_biela']}.\",
\".${Frm['sobre5_inf_biela']}.\",
\".${Frm['sobre6_sup_biela']}.\",
\".${Frm['sobre6_inf_biela']}.\",\".${Frm['std_sup_bancada']}.\",\".${Frm['std_inf_bancada']}.\",
\".${Frm['sobre1_sup_bancada']}.\",
\".${Frm['sobre1_inf_bancada']}.\",\".${Frm['sobre2_sup_bancada']}.\",
\".${Frm['sobre2_inf_bancada']}.\",
\".${Frm['sobre3_sup_bancada']}.\",
\".${Frm['sobre3_inf_bancada']}.\",\".${Frm['sobre4_sup_bancada']}.\",
\".${Frm['sobre4_inf_bancada']}.\",
\".${Frm['sobre5_sup_bancada']}.\",
\".${Frm['sobre5_inf_bancada']}.\",\".${Frm['sobre6_sup_bancada']}.\",
\".${Frm['sobre6_inf_bancada']}.\",

\".${Frm['torquebancada_motor']}.\",\".${Frm['std_inf_alojabancada_motor']}.\",
\".${Frm['std_sup_alojabancada2_motor']}.\",
```



```

        \".$aFrm['castingnum_cilindro'].\", \".$aFrm['comments_cilindro'].\", \".$aF
rm['std_sup_cilindro'].\", \".$aFrm['std_inf_cilindro'].\",
        \".$aFrm['sobre1_sup_cilindro'].\",
\".$aFrm['sobre1_inf_cilindro'].\", \".$aFrm['sobre2_sup_cilindro'].\",
\".$aFrm['sobre2_inf_cilindro'].\",
        \".$aFrm['sobre3_sup_cilindro'].\",
\".$aFrm['sobre3_inf_cilindro'].\", \".$aFrm['sobre4_sup_cilindro'].\",
\".$aFrm['sobre4_inf_cilindro'].\",
        \".$aFrm['sobre5_sup_cilindro'].\",
\".$aFrm['sobre5_inf_cilindro'].\", \".$aFrm['sobre6_sup_cilindro'].\",
\".$aFrm['sobre6_inf_cilindro'].\",

        \".$aFrm['medida'].\", \".$aFrm['combustible'].\", \".$aFrm['marca'].\", \".$aFrm['m
odelo'].\", 1);
                if($my->QueryT($sql)){
                    $oReturn->alert('Ingreso de Motor
\".$aFrm['nombre_motor'].\" Correctamente!!!');
                    $oReturn->script('xajax_genera_motor();');// aqui se
cierra la ventana y se dirige al combo de combustible
                }else $oReturn->alert($aError[1]);
            }
            break;
        case 2:
            if (verifica_motor($aFrm['nombre_motor'])){
                $oReturn->alert('El motor: \".$aFrm['nombre_motor'].\"
ya existe!!!');
            }else{
                $sql = 'UPDATE motor SET NOMBRE_MOTOR =
\".$aFrm['nombre_motor'].\" ,CILINDRAJE =
\".$aFrm['cilindraje'].\" ,ANO_MOTOR = \".$aFrm['ano_motor'].\" ,
                PISTON_MOTOR =
\".$aFrm['piston_motor'].\" ,BRAZO_BIELA_MOTOR =
\".$aFrm['brazo_biela_motor'].\" ,BRAZO_BIELA2_MOTOR =
\".$aFrm['brazo_biela2_motor'].\" ,
                STD_INF_ALOJABIELA =
\".$aFrm['std_inf_alojabiela'].\" ,STD_SUP_ALOJABIELA =
\".$aFrm['std_sup_alojabiela'].\" ,TORQUEBIELA_MOTOR =
\".$aFrm['torquebiela_motor'].\" ,
                STROKE_CIGUENAL =
\".$aFrm['stroke_ciguenal'].\" ,LUZBIELA_MOTOR =
\".$aFrm['luzbiela_motor'].\" ,LUZBIELA2_MOTOR =
\".$aFrm['luzbiela2_motor'].\" ,

```

```

LUZBANCADA_MOTOR =
"".$aFrm['luzbancada_motor'].",LUZBANCADA2_MOTOR =
"".$aFrm['luzbancada2_motor'].",CASTINGNUM_CIGUENAL =
"".$aFrm['castingnum_ciguenal'].",
COMMENTS_CIGUENAL =
"".$aFrm['comments_ciguenal'].",STD_SUP_BIELA =
"".$aFrm['std_sup_biela'].",
STD_INF_BIELA =
"".$aFrm['std_inf_biela'].",SOBRE1_SUP_BIELA =
"".$aFrm['sobre1_sup_biela'].",SOBRE1_INF_BIELA =
"".$aFrm['sobre1_inf_biela'].",
SOBRE2_SUP_BIELA =
"".$aFrm['sobre2_sup_biela'].",SOBRE2_INF_BIELA =
"".$aFrm['sobre2_inf_biela'].",
SOBRE3_SUP_BIELA =
"".$aFrm['sobre3_sup_biela'].",SOBRE3_INF_BIELA =
"".$aFrm['sobre3_inf_biela'].",
SOBRE4_SUP_BIELA =
"".$aFrm['sobre4_sup_biela'].",SOBRE4_INF_BIELA =
"".$aFrm['sobre4_inf_biela'].",SOBRE5_SUP_BIELA =
"".$aFrm['sobre5_sup_biela'].",
SOBRE5_INF_BIELA =
"".$aFrm['sobre5_inf_biela'].",SOBRE6_SUP_BIELA =
"".$aFrm['sobre6_sup_biela'].",SOBRE6_INF_BIELA =
"".$aFrm['sobre6_inf_biela'].",
STD_SUP_BANCADA =
"".$aFrm['std_sup_bancada'].",STD_INF_BANCADA =
"".$aFrm['std_inf_bancada'].",
SOBRE1_SUP_BANCADA =
"".$aFrm['sobre1_sup_bancada'].",SOBRE1_INF_BANCADA =
"".$aFrm['sobre1_inf_bancada'].",SOBRE2_SUP_BANCADA =
"".$aFrm['sobre2_sup_bancada'].",
SOBRE2_INF_BANCADA =
"".$aFrm['sobre2_inf_bancada'].",SOBRE3_SUP_BANCADA =
"".$aFrm['sobre3_sup_bancada'].",
SOBRE3_INF_BANCADA =
"".$aFrm['sobre3_inf_bancada'].",SOBRE4_SUP_BANCADA =
"".$aFrm['sobre4_sup_bancada'].",SOBRE4_INF_BANCADA =
"".$aFrm['sobre4_inf_bancada'].",
SOBRE5_SUP_BANCADA =
"".$aFrm['sobre5_sup_bancada'].",SOBRE5_INF_BANCADA =
"".$aFrm['sobre5_inf_bancada'].",
SOBRE6_SUP_BANCADA =
"".$aFrm['sobre6_sup_bancada'].",SOBRE6_INF_BANCADA =

```

```

'''.$aFrm['sobre6_inf_bancada'].''',TORQUEBANCADA_MOTOR           =
'''.$aFrm['torquebancada_motor'].''',
                                STD_INF_ALOJABANCADA_MOTOR       =
'''.$aFrm['std_inf_alojabancada_motor'].''',
                                STD_SUP_ALOJABANCADA2_MOTOR      =
'''.$aFrm['std_sup_alojabancada2_motor'].''',CASTINGNUM_CILINDRO  =
'''.$aFrm['castingnum_cilindro'].''',
                                COMMENTS_CILINDRO               =
'''.$aFrm['comments_cilindro'].''',STD_SUP_CILINDRO             =
'''.$aFrm['std_sup_cilindro'].''',
                                STD_INF_CILINDRO                 =
'''.$aFrm['std_inf_cilindro'].''',SOBRE1_SUP_CILINDRO           =
'''.$aFrm['sobre1_sup_cilindro'].''',
                                SOBRE1_INF_CILINDRO              =
'''.$aFrm['sobre1_inf_cilindro'].''',SOBRE2_SUP_CILINDRO        =
'''.$aFrm['sobre2_sup_cilindro'].''',SOBRE2_INF_CILINDRO        =
'''.$aFrm['sobre2_inf_cilindro'].''',
                                SOBRE3_SUP_CILINDRO              =
'''.$aFrm['sobre3_sup_cilindro'].''',SOBRE3_INF_CILINDRO        =
'''.$aFrm['sobre3_inf_cilindro'].''',SOBRE4_SUP_CILINDRO        =
'''.$aFrm['sobre4_sup_cilindro'].''',
                                SOBRE4_INF_CILINDRO              =
'''.$aFrm['sobre4_inf_cilindro'].''',SOBRE5_SUP_CILINDRO        =
'''.$aFrm['sobre5_sup_cilindro'].''',
                                SOBRE5_INF_CILINDRO              =
'''.$aFrm['sobre5_inf_cilindro'].''',SOBRE6_SUP_CILINDRO        =
'''.$aFrm['sobre6_sup_cilindro'].''',SOBRE6_INF_CILINDRO        =
'''.$aFrm['sobre6_inf_cilindro'].''',
                                MEDIDA = '''.$aFrm['medida'].'''' WHERE
ID_MOTOR = '.$aFrm['motor'];
                                //echo $sql;
                                if($my->QueryT($sql)){
                                    $oReturn->alert('La Actualización de Motor
'. $aFrm['nombre_motor'].'. Ha sido actualizado Correctamente!!!');
                                    $oReturn->script('xajax_genera_motor());// aqui se
cierra la ventana y se dirige al combo de combustible
                                    }else $oReturn->alert($aError[1]);
                                }
                                break;
                                case 3:
                                    $sql = 'UPDATE motor SET ESTADO = 0 WHERE ID_MOTOR
= '.$aFrm['motor'].'';
                                    if($my->QueryT($sql)){
                                        $oReturn->alert('Se ha eliminado Correctamente!!!');

```

```
        $oReturn->script('xajax_genera_motor()'); // aqui se cierra
la ventana y se dirige al combo de combustible
        }else $oReturn->alert($aError[1]);
        break;
    }
    $my->Desconectar();
    $oReturn->assign("guardar","disabled",false);
    return $oReturn;
}
//Validacion Motor
function verifica_motor($motor){
    global $DSN;
    $oCon = new Dbo;
    $oCon -> DSN = $DSN;
    $$Sql="SELECT * FROM motor WHERE NOMBRE_MOTOR='$motor' and
estado = 1";
    $oCon->Conectar();
    if ($oCon->Query($$Sql))
        if ($oCon->f('nombre_motor')===$motor){
            return true;
        }else return false;
}
?>
```

**HISTORIA DE USUARIO****NUMERO:** 001**NOMBRE:** Inicio Sistema**USUARIO:** Administrador**MODIFICACION DE HISTORIA DE USUARIO:****ITERACION:****PRIORIDAD DE NEGOCIO:** Alta**PUNTOS ESTIMADOS:****RIESGO EN DESARROLLO:****PUNTOS REALES:****DESCRIPCION:**

Se requiere el ingreso con usuario y contraseña con un diseño con el logotipo de la empresa

**OBSERVACIONES:**

Considerar derechos en la pagina con pie de pagina

TAREA

NUMERO TAREA:

001

NUMERO HISTORIA:

001

NOMBRE TAREA:

Ingreso Sistema

TIPO TAREA:

Modificar

PUNTOS ESTIMADOS:

FECHA INICIO:

FECHA FIN:

PROGRAMADOR RESPONSABLE:

DESCRIPCION:

Aumentar tipo de letra y tamaño de los botones.

Controlar q la contraseña sea incorrecta y q pertenezca al usuario

### HISTORIA DE USUARIO

NUMERO: 002

NOMBRE: Administración Combustible

USUARIO: Administrador

MODIFICACION DE HISTORIA DE USUARIO:

ITERACION:

PRIORIDAD DE NEGOCIO:

Alta

PUNTOS ESTIMADOS:

RIESGO EN DESARROLLO:

PUNTOS REALES:

#### DESCRIPCION:

Permite insertar, eliminar, actualizar combustible con la siguiente información

Nombre Combustible

#### OBSERVACIONES:

No permite ingresar duplicados

TAREA

NUMERO TAREA: 002

NUMERO HISTORIA: 002

NOMBRE TAREA: Administracion Combustible

TIPO TAREA: Modificar / Adicionar

PUNTOS ESTIMADOS:

FECHA INICIO:

FECHA FIN:

PROGRAMADOR RESPONSABLE:

DESCRIPCION:

Adicionar imagen relacionada con la opción,  
Mensaje de advertencia cuando se quiera  
eliminar un combustible

Mensaje de advertencia cuando se ingresa un  
combustible repetido.



### HISTORIA DE USUARIO

NUMERO: 003 NOMBRE: Administracion Marca

USUARIO: Administrador

MODIFICACION DE HISTORIA DE USUARIO: ITERACION:

PRIORIDAD DE NEGOCIO: Alta PUNTOS ESTIMADOS:

RIESGO EN DESARROLLO: PUNTOS REALES:

#### DESCRIPCION:

Permite ingresar, actualizar, eliminar marca; donde el combustible este relacionado con la marca  
Todo combustible tiene una marca de motor,  
permite ingresar la siguiente información  
Nombre Marca

#### OBSERVACIONES:

No permite ingreso de marcas duplicadas

**TAREA****NUMERO TAREA:**

003

**NUMERO HISTORIA:**

003

**NOMBRE TAREA:**

Administracion marca

**TIPO TAREA:**

Adicionar / Modificar

**PUNTOS ESTIMADOS:****FECHA INICIO:****FECHA FIN:****PROGRAMADOR RESPONSABLE:****DESCRIPCION:**

Mostrar mensaje de advertencia cuando se ingrese duplicados o repetidos, q no permita ingresar vacios.  
Mostrar un mensaje cuando se desee eliminar una marca ~~etc~~ de confirmación.  
Adicionar una imagen relacionada con la opción.

**HISTORIA DE USUARIO****NUMERO:** 004**NOMBRE:** Administración Modelo**USUARIO:** Administrador**MODIFICACION DE HISTORIA DE USUARIO:****ITERACION:****PRIORIDAD DE NEGOCIO:**

Alta

**PUNTOS ESTIMADOS:****RIESGO EN DESARROLLO:****PUNTOS REALES:****DESCRIPCION:**

Permite ingresar, actualizar, eliminar modelo, un modelo esta relacionado con una marca y un combustible.

Permite tener la siguiente información

Nombre Marca

**OBSERVACIONES:**

No permite ingresar duplicados

TAREA	
NUMERO TAREA: 004	NUMERO HISTORIA: 004
NOMBRE TAREA: Administración Modelo	
TIPO TAREA: Adicionar / Modificar	PUNTOS ESTIMADOS:
FECHA INICIO:	FECHA FIN:
PROGRAMADOR RESPONSABLE:	
DESCRIPCION:	
<p>Modificar el tamaño de la letra, adicionar una imagen relacionada a la opción.</p> <p>Adicionar mensajes de advertencia.</p>	

### HISTORIA DE USUARIO

NUMERO: 005

NOMBRE: ~~005~~ Administración Motor

USUARIO: Administrador

MODIFICACION DE HISTORIA DE USUARIO:

ITERACION:

PRIORIDAD DE NEGOCIO:

PUNTOS ESTIMADOS:

RIESGO EN DESARROLLO:

PUNTOS REALES:

#### DESCRIPCION:

Con los parametros antes ingresados q' son Combustible, Marca, Modelo, ingresar, actualizar y eliminar motor con la siguiente informacion

año, nombre motor, biela, alojamiento de biela, alojamiento

bancada, std, sobre-medida 1p, sobre-medida 2 inf, ... sobre-medida 6

de cigueñal, std, sobre-medida 1 sup, sobre-medida 2 inf... sobre-medida 6 de biela.

Combustible

Marca

Modelo

#### OBSERVACIONES:

No permitir duplicados

TAREA	
NUMERO TAREA: 005	NUMERO HISTORIA: 005
NOMBRE TAREA: Administracion Motor	
TIPO TAREA: Modificar / Adicionar	PUNTOS ESTIMADOS:
FECHA INICIO:	FECHA FIN:
PROGRAMADOR RESPONSABLE:	
DESCRIPCION: Adicionar el check para q' se pueda seleccionar el ingreso en milimetros o pulgadas. Adicionar las 6 sobre medidas para cigueñal, cilindro, biela, bancada. Aumentar tamaño de letra.	

TAREA	
NUMERO TAREA: 006	NUMERO HISTORIA: 006
NOMBRE TAREA: Consulta Motor	
TIPO TAREA: Modificar/	PUNTOS ESTIMADOS:
FECHA INICIO:	FECHA FIN:
PROGRAMADOR RESPONSABLE:	
DESCRIPCION: Aumentar tamaño de letra, presentar el despliegue de la búsqueda en la misma pantalla. Presentar Alertas si no se ingresa valores en la pantalla de ingreso de los calculos	

HISTORIA DE USUARIO	
NUMERO: 006	NOMBRE: Consulta Motores
USUARIO: Administrador / Operador	
MODIFICACION DE HISTORIA DE USUARIO:	ITERACION:
PRIORIDAD DE NEGOCIO:	PUNTOS ESTIMADOS:
RIESGO EN DESARROLLO:	PUNTOS REALES:
<b>DESCRIPCION:</b> Se realiza un buscador, ingresando una palabra clave de combustible, marca, modelo, motor. buscar <input type="text"/> Despliega una lista con todos los motores encontrados de la búsqueda; al hacer click en uno de ellos muestra la información relacionada. Después permite el ingreso de los valores de las piezas a rectificar y los resultados del cálculo realizado.	
<b>OBSERVACIONES:</b>	



HISTORIA DE USUARIO

<b>NUMERO:</b> 007	<b>NOMBRE:</b> Cambio clave
<b>USUARIO:</b> Administrador / Operador	
<b>MODIFICACION DE HISTORIA DE USUARIO:</b>	<b>ITERACION:</b>
<b>PRIORIDAD DE NEGOCIO:</b>	<b>PUNTOS ESTIMADOS:</b>
<b>RIESGO EN DESARROLLO:</b>	<b>PUNTOS REALES:</b>
<b>DESCRIPCION:</b>  Permitir el cambio de clave de un usuario.	
<b>OBSERVACIONES:</b>  Salir del sistema e ingresar con la nueva clave	

TAREA	
NUMERO TAREA: 007	NUMERO HISTORIA: 007
NOMBRE TAREA: Cambio Clave	
TIPO TAREA: Modificar	PUNTOS ESTIMADOS:
FECHA INICIO:	FECHA FIN:
PROGRAMADOR RESPONSABLE:	
DESCRIPCION: Permitir la confirmación de la clave nueva	

PRUEBA DE ACEPTACION

CASO DE PRUEBA: 001

NUMERO CASO PRUEBA: 1

NUMERO HISTORIA DE USUARIO: 001

NOMBRE CASO PRUEBA:

Ingreso de Sistema

DESCRIPCION:

Se proba la conformidad del cliente con el requerimiento

CONDICION DE EJECUCION:

Servicios estables y corriendo

ENTRADAS:

~~N/A~~ Usuario  
Contraseña

RESULTADO ESPERADO:

Se ~~can~~ ingresa al sistema con la contraseña correspondiente para el usuario.

EVALUACION:

N/A

PRUEBA DE ACEPTACION

CASO DE PRUEBA: 002

NUMERO CASO PRUEBA: 2

NUMERO HISTORIA DE USUARIO: 002

NOMBRE CASO PRUEBA:

Administración Combustible

DESCRIPCION:

Se probará el ingreso, actualización de combustible así como su eliminación.

Se probará q' no se ingrese duplicados e indique los mensajes de advertencia.

CONDICION DE EJECUCION:

Sin condiciones

ENTRADAS:

Nombre de Combustible

RESULTADO ESPERADO:

Ingreso de Combustible fue exitoso

Actualización Combustible se realizó exitosamente

Eliminación Combustible presentó exitosamente los mensajes de advertencia se realizó con satisfacción

EVALUACION:

N/A

PRUEBA DE ACEPTACION

CASO DE PRUEBA: 003

NUMERO CASO PRUEBA: 3

NUMERO HISTORIA DE USUARIO: 003

NOMBRE CASO PRUEBA:  
Administración Marca

DESCRIPCION:

- Probar el ingreso, actualización y eliminación de Marca
- Probar mensajes de advertencia cuando se quiera ingresar una marca ya existente o cuando se requiera eliminar.

CONDICION DE EJECUCION:

Previo ingreso de un combustible

ENTRADAS:

Nombre ~~de~~ Marca

RESULTADO ESPERADO:

Ingreso, Actualización, Eliminación de Marca dependiendo del combustible se realizó exitosamente

EVALUACION:

N/A

PRUEBA DE ACEPTACION

CASO DE PRUEBA: 004

NUMERO CASO PRUEBA: 4

NUMERO HISTORIA DE USUARIO: 004

NOMBRE CASO PRUEBA:

Administración Modelo

DESCRIPCION:

Probar el ingreso, actualización y eliminación de modelo, con sus respectivas mensajes de advertencia cuando se ingresa duplicados o cuando se requiera eliminar

CONDICION DE EJECUCION:

Selección de Combustible y de Marca

ENTRADAS:

Nombre Modelo

RESULTADO ESPERADO:

Ingreso, actualización, eliminación de combustible fue exitoso

EVALUACION:

N/A

PRUEBA DE ACEPTACION

CASO DE PRUEBA: 005

NUMERO CASO PRUEBA: 5

NUMERO HISTORIA DE USUARIO: 005

NOMBRE CASO PRUEBA:

Administración Motor

DESCRIPCION:

- Probar el ingreso, actualización, eliminación de motor.
- Probar mensajes de advertencia para duplicados y para eliminar un motor.
- Probar q' permite solo números en los campos de sobre-medida para cilindro, bancada, cigüeñal.

CONDICION DE EJECUCION:

Selección Combustible, Marca, Modelo

ENTRADAS:

Todos los campos q' sean necesarios para el motor dependiendo de la selección de combustible, marca, modelo

RESULTADO ESPERADO:

Ingreso, Actualización, eliminación de motor fueron exitosos

EVALUACION:

N/A

PRUEBA DE ACEPTACION

CASO DE PRUEBA: 006

NUMERO CASO PRUEBA: 1

NUMERO HISTORIA DE USUARIO: 006

NOMBRE CASO PRUEBA:  
Consulta Motor

DESCRIPCION:

Se probará la búsqueda con ingreso de palabra clave de combustible, marca, modelo o motor

CONDICION DE EJECUCION:

Exista datos de combustible, marca, modelo relacionado con un motor

ENTRADAS:

Palabra de búsqueda

RESULTADO ESPERADO:

Listado de motor resultado del ingreso de la palabra ingresada.

EVALUACION:

N/A



PRUEBA DE ACEPTACION

CASO DE PRUEBA: 007

NUMERO CASO PRUEBA: 2

NUMERO HISTORIA DE USUARIO: 006

NOMBRE CASO PRUEBA:  
Consulta Motor

DESCRIPCION:

Se probará la muestra de los datos del motor seleccionado después de hacer la búsqueda

CONDICION DE EJECUCION:

Ingreso de la palabra clave dentro del buscador

ENTRADAS:

Palabra clave, selección del motor deseado.

RESULTADO ESPERADO:

Información de todas los datos relacionados del motor seleccionado, los resultados serán mostrados en ambas medidas pulgadas / milímetros

EVALUACION:

N/A

PRUEBA DE ACEPTACION

CASO DE PRUEBA: 008

NUMERO CASO PRUEBA: 3

NUMERO HISTORIA DE USUARIO: 006

NOMBRE CASO PRUEBA: Consulta Motor

DESCRIPCION:

Se probará el ingreso de los valores de las piezas a rectificar

CONDICION DE EJECUCION:

Selección del motor ,despues de la busqueda

ENTRADAS:

Valores de Ciguenal , Cilindro, Biela  
Selección de pulgadas /milímetros

RESULTADO ESPERADO:

Listado de los valores de las condiciones iniciales y el resultado esperado para las piezas a ser rectificadas.

EVALUACION:

N/A

PRUEBA DE ACEPTACION

CASO DE PRUEBA: 009

NUMERO CASO PRUEBA: 1

NUMERO HISTORIA DE USUARIO: 007

NOMBRE CASO PRUEBA: Cambio Clave

DESCRIPCION:

Se probara el cambio de clave del usuario ingresado al sistema

CONDICION DE EJECUCION:

Usuario , contraseña

ENTRADAS: Usuario , contraseña , confirmación contraseña nueva

RESULTADO ESPERADO:

Reingreso al sistema con la clave cambiada

EVALUACION:

N/A