

Universidad Nacional Experimental del Táchira

Vicerrectorado Académico

Decanato de Docencia

Departamento de Ingeniería Informática

Trabajo de Aplicación Profesional

Proyecto Especial de Grado

SISTEMA PARA LA GESTIÓN DE FLOTA DE VEHICULOS. CASO DE ESTUDIO: SERVICIOS GENERALES UNET

Autores:

Jacobo A. Celis A.

C.I.19.234.652

jacobo.celis@gmail.com

Marlet Y. Pereira A.

C.I. 19.778.019

[letmarariday@gmail.com](mailto:letmarariday@gmail.com)

Tutor: Msc. José Orlando Figueroa

San Cristóbal, Abril 2015

**CAPÍTULO I**

**EL PROBLEMA**

**Planteamiento del Problema**

Una empresa que utiliza vehículos para generar sus ingresos bien sea ofreciendo servicios de transporte de personas o mercancía, de forma terrestre aérea o marítima, necesita un sistema para la gestión de su flota que permita conocer de manera oportuna e inmediata el estado de funcionamiento de cada vehículo de la empresa. La gestión de flotas se basa en la administración del conjunto de [vehículos](http://es.wikipedia.org/wiki/Veh%C3%ADculo) de una [organización](http://es.wikipedia.org/wiki/Organizaci%C3%B3n) incluyendo funciones como la disminución de costos, el mantenimiento de los vehículos, seguimiento en tiempo real de la flota, diagnóstico de fallas, gestión de conductores, consumo de combustible y seguridad del personal, la gestión de flota permite minimizar o eliminar los riesgos asociados con la inversión de los vehículos y mejorar su eficiencia y productividad.

La Dirección de servicios funciona como una unidad de apoyo para toda la universidad y se encuentra adscrita al Vice-Rectorado Administrativo, ante el cual será responsable por el cumplimiento de sus funciones. El departamento se encuentra estructurado de la siguiente manera: Dirección, Departamento de Servicios Generales, Departamento de Mantenimiento y Departamento de Administración de Contrato de Servicio, los cuales funcionan de manera global y simultánea para garantizar que se cumplan todos los procesos que sean requeridos a dicha jurisdicción.

La Dirección de Servicios de la Universidad Nacional Experimental del Táchira tiene entre sus funciones regular los procesos de los diferentes servicios requeridos por las dependencias universitarias, tales como: mantenimiento y reparación de equipos, limpieza, vigilancia, transporte, mantenimiento y conservación de los activos universitarios, bienes en custodia y administración de todos los contratos de servicios.

Para dicha Organización, la falta de información acerca del estado de las unidades en la flota constituye un problema para la toma decisiones en la organización, conocer el estado físico de una unidad de acuerdo a su utilización garantiza cumplir totalmente un recorrido sin temor a que la unidad falle o se accidente, fracasando con la tarea designada. La ausencia de registros del uso o del recorrido total efectuado por cada unidad a lo largo de su vida útil, también representa un problema, puesto que no permite planificar oportunamente, cuando es necesario realizar mantenimientos que garanticen el óptimo estado de las unidades dentro de la flota, ni realizar estimaciones sobre el costo por recorrido de cada una de las unidades.

También resulta un inconveniente para la gerencia la ausencia de un inventario de las piezas que un vehículo emplea, provocando en algunos casos desinformación sobre las reparaciones de un vehículo que hacen uso de intercambio de piezas, ocasionando por parte del personal encargado procesamiento de facturas que no garantizan el pago de la reparación de la avería de la unidad, es decir, el personal no puede verificar si se realizó de manera correcta el arreglo de la unidad.

Asimismo, la solicitud de las rutas que realizan los distintos departamentos de la universidad representa un problema, puesto que no se cuenta con ningún tipo de control automático que garantice el apartado de una ruta para una fecha en específico, tampoco existen históricos que permitan llevar un control sobre la vida útil del vehículo, ni listados donde pueda apreciarse el consumo de combustible, ruedas y viajes, las reparaciones, los mantenimientos preventivos y correctivos, los siniestros ocurridos, los fallos en las reparaciones, las multas y sanciones y el listado de conductores de cada una de las unidades.

Para dar solución a los problemas anteriormente planteados, se propone a la unidad de servicios generales el desarrollo de un sistema que permita la gestión de la flota de rutas de la UNET, para así fortalecer y facilitar el proceso de control y mantenimiento de las unidades de la flota, mejorando el desarrollo de las funciones de las gerencias tanto operativas como administrativas. El sistema planteado resulta una herramienta útil para los dueños de empresas dedicadas a la gestión de flotas, puesto que para garantizar el éxito de estas empresas, se requiere conocer en tiempo real el estado de todas las unidades que conforman el conjunto total de la flota.

**Objetivos de Investigación**

***General***

Desarrollar un sistema para la gestión de flota de vehículos. Caso de estudio: servicios generales UNET.

***Específicos***

1. Definir los requerimientos y necesidades para la gestión de flota de vehículos.
2. Construir los módulos del sistema.
3. Construir los indicadores de gestión necesarios para la toma de decisiones.
4. Realizar pruebas de calidad del software desarrollado
5. Integrar el sistema a los sistemas actuales de la Universidad Nacional Experimental del Táchira.

**Justificación de la Investigación**

Hoy en día la gran mayoría de empresas y organizaciones cuentan con sistemas web adaptables que mejoran el desempeño operativo y fortalecen la capacidad y crecimiento de la misma, así como también garantizan una mejor administración sobre las diferentes actividades realizadas.

Según Bautista (2005), la gestión de flotas puede incluir una variedad de objetivos y funciones como el mantenimiento de vehículos, el seguimiento y control de vehículos, la detención remota de vehículos, el diagnóstico mecánico, la administración de conductores, la gestión de combustible, la gestión de la seguridad y, en general, todo lo referido al análisis de los datos e información disponible y a la toma de decisiones vinculados a la flota de vehículos.

De acuerdo con lo dicho anteriormente, surge la necesidad de elaborar un sistema que permita la gestión de la flota de unidades de transporte de la UNET, con la finalidad de obtener en determinado momento las características de cada vehículo perteneciente a la empresa, a fin de sustentar dicha información, tanto para ellos, como para fines organizacionales a la hora de realizar reuniones y rendir cuentas sobre su gestión.

Dicho sistema está dirigido para todos aquellos usuarios que formen parte de la unidad de servicios generales de la UNET, y se encuentren en cargos que se relacionen directamente con el control de la flota de vehículos para el transporte universitario. Su enfoque es garantizar el control y mantenimiento de las unidades de la flota para mejorar los procesos operativos y administrativos de la organización.

**Alcance y Limitaciones**

El propósito de esta investigación consiste en desarrollar un sistema que brinde apoyo a la gestión de flota de unidades de transporte de la UNET, esto se logra con la construcción de los módulos para el mantenimiento preventivo y correctivo de las unidades de transporte, administración y control de los neumáticos de cada vehículo, control de rutas, generación de los reportes e indicadores de gestión que permitan ayudar a la gerencia en la toma de decisiones y en la gestión, permitiendo la reducción de costos operativos y administrativos, mejoras en los servicios prestados y el resto de actividades asociadas con la unidad de servicios generales.

El sistema debe permitir observar en el momento actual la disponibilidad de cada unidad, el estado de cada ruta (funcionando, dañada, o en reparación), los tiempos que ha tardado en volver a funcionar, la última falla registrada y el comportamiento de éstas. Adicionalmente, el sistema también debe permitir registrar un seguimiento de cada uno de los vehículos en la flota, almacenando las rutas que han realizado cada día, el tiempo de duración de cada viaje y la cantidad de combustible consumido en cada viaje. También debe permitir la notificación de fallas recurrentes, es decir, en caso de ingresar una falla nueva en un vehículo debe mostrar si ya este vehículo había registrado ese tipo de falla y la fecha en que fue verificada. Esto puede ayudar a la gerencia a determinar la calidad de los repuestos usados en la reparación de cada ruta y poder tomar las decisiones correspondientes.

La implementación de dicho software se plantea utilizando una arquitectura web intuitiva y amigable al usuario, que resulte sencilla pero a la vez completa, generando así una solución que llegará a un mercado más amplio de clientes, por otra parte dicho sistema garantiza que no existan terceros que manipulen la información sensible de los usuarios. Se realizará la verificación de la funcionalidad del producto garantizando un software de calidad para posteriormente integrarlo a la plataforma que actualmente posee la universidad.

Sin embargo, la investigación se ve afectada por ciertas limitantes como la ausencia de dispositivos de localización satelital (GPS) para la gestión y control de las unidades, en cuanto a la ubicación y seguimiento de un vehículo en tiempo real, y el acceso instantáneo a la información importante de las unidades de la flota, así como también aspectos relacionados con el comportamiento de conducción como la velocidad, el kilometraje y el consumo de combustible.

**CAPÍTULO II**

**MARCO TEÓRICO**

**Antecedentes de Investigación**

A nivel internacional, se reseña el siguiente antecedente: Bautista (2005), en Montevideo-Uruguay, realizó una investigación titulada *Sistema de Gestión de Flota Vehicular,* el cual tenía como propósito el desarrollo e implantación de un software corporativo para la gestión de la operación, mantenimiento y supervisión de una flota de 1200 vehículos de gran variedad (vehículos livianos, vehículos pesados de carga, maquinaria vial y maquinaria de elevación) ampliamente distribuidos en el territorio nacional. Proporcionando como conclusiones más importantes de este proyecto una oportuna y correcta toma de decisiones tanto gerenciales como operativas, y que la información es accesible, suficiente y única. Los resultados obtenidos con la aplicación de esta herramienta teniendo en cuenta que el sistema documenta la forma de trabajo en lo que respecta a mantenimiento y operación de una flota, seria principalmente que estarían dadas las bases para comenzar un proceso de certificación de la “Gestión de Flota Vehicular” en la empresa.

Por otra parte, a nivel nacional Busto (2012), en la Universidad Nacional Experimental de Guayana, realizó una investigación titulada *Diseño de un sistema de información para el control de la flota de vehículos de la gerencia de transportes de C.V.G Ferrominera Orinoco.* El cual se enfoca en garantizar el cumplimiento de los programas de mantenimiento preventivo y correctivo de los equipos de transporte, así como de la coordinación de los servicios de transporte de la empresa con el objeto que sean ejecutadas oportunamente de acuerdo con lo establecido con la gerencia general de operaciones mineras. Presentando como resultados positivos que la propuesta del sistema les permitió recolectar y resumir información importante de los vehículos, en tiempo real y veraz, donde se observaron diferentes indicadores de mantenimiento como: la disponibilidad, el tiempo entre fallas, el tiempo de reparación, el estado, última falla y el comportamiento de estas, las cuales al momento de necesitarlas se obtendrán mediante la generación de reportes, ya sean diarios, semanal, mensual o anual.

A nivel nacional, se tiene a Peinado (2009), en la Universidad Nacional Experimental Politécnica Antonio José de Sucre, quién realizó una investigación titulada *Optimización de la flota de camionetas tipo pick – up asignadas a las áreas administrativas de CVG Ferrominera Orinoco,* dicha investigación propone una herramienta para dar soporte al seguimiento y evaluación de la operatividad de las camionetas tipo pick – up, de las áreas administrativas, desarrollando de esta manera un análisis y evaluación de las condiciones de la flota de vehículos livianos. La aplicaciónse basa fundamentalmente en el cumplimiento constante y preciso de procedimientos de asignación, uso y control de accidentes, mantenimiento de vehículos y desincorporación y reemplazo de unidades, a partir de la definición de los problemas actuales que se presentan en la flota actual de vehículos. Teniendo como fin el aumento de los niveles de productividad y reducción de tiempo de desincorporación por fallas o averías.

A nivel regional se encuentra a Zambrano (2010), quién realizó un estudio en la Universidad Nacional Experimental de Táchira, titulado *Sistema para el Control de la Gestión de acciones de Mantenimiento en Ambiente Web.* Este proyecto se basa en acoplar los procesos que abarca la gestión de mantenimiento, de manera automatizada, en un sistema que permita entradas de objetos a mantener, instrucciones técnicas a ejecutar y recursos materiales, repuestos, herramientas, equipos e instrumentos a utilizar. La aplicación cuenta con la programación de un objeto en específico durante un período de tiempo o rango de vida definido, en los cuales se aplican acciones de mantenimiento, llevando un control de inventario, horas de trabajo y costo de la actividad. La finalidad del proyecto es la planificación, programación, seguimiento y control de las actividades con el fin de pretender la disminución de paradas no programadas en la organización, optimizar la utilización de los recursos y así, aumentar la productividad y el beneficio del sistema productivo.

**Bases Teóricas**

Dado que las bases teóricas constituyen el corazón de todo trabajo de investigación, pues es sobre este que se construye todo el trabajo, y es aquí también donde se sitúa el problema que se está estudiando, dentro de un conjunto de conocimientos, que permita orientar la búsqueda y ofrezca una conceptualización adecuada de los términos que serán utilizados. Cabe destacar que, para que este trabajo sea fácilmente comprendido por cualquier investigador existen tres definiciones básicas antes de abordar cualquier otra conceptualización que fundamente la investigación.

***Proyecto***

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. La temporalidad indica que un proyecto tiene un principio y un final, siendo el final alcanzado cuando se logran los objetivos del mismo o se termina debido a que los objetivos no se cumplieron o no podrán ser cumplidos (Project Management Institute, Inc, 2008).

***Sistema de Información***

Un sistema de información es un organismo que recolecta, procesa, almacena y distribuye información. Son indispensables para ayudar a los gerentes a mantener ordenada su compañía, a analizar todo lo que por ella pasa y a crear nuevos productos

que coloquen en un buen lugar a la organización (Laudon y Laudon, 2008).

Esta definición es una de las únicas que manifiesta la exigencia de que un sistema de información tenga componentes, aunque no especifica cuáles deban ser, posiblemente porque intenta englobar todas las posibles variantes de este concepto. Cabe resaltar que el concepto de sistema de información suele ser utilizado como sinónimo de sistema de información informático, aunque no son lo mismo. Un sistema de información informático realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información. Dichas actividades se describen a continuación:

1. ***Entrada de Información:*** Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas.
2. ***Almacenamiento de información:***El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.
3. **Procesamiento de Información:** Es la capacidad del sistema de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.
4. ***Salida de Información:*** La salida es la capacidad de un sistema de información para sacar la información procesada o bien datos de entrada al exterior. Es importante aclarar que la salida de un sistema de información puede constituir la entrada a otro sistema de información o módulo. En este caso, también existe una interface automática de salida.

Entrada

De

Datos

Interface Automático de Entrada

Interface Automático de Salida

Reportes e

Informes

Proceso

Almacenamiento

**Figura 1**. Funcionamiento de un sistema de información informático. Peralta. 2008.

***Gestión***

Beltrán (1999), define genéricamente la gestión de la siguiente manera: “conjunto de decisiones y acciones que llevan al logro de objetivos previamente establecidos” (p. 24).

De acuerdo a Porter (1985), en las empresas es común hablar de tres niveles de gestión: a) La gestión estratégica, b) La gestión táctica, c) La gestión operativa, además de esto, dicho autor también habla acerca de medir la gestión de acuerdo a signos vitales, refiriéndose a factores claves de éxito que indican si la gestión está dando los resultados esperados, o si por el contrario, existe alguna desviación que los aparte de los objetivos establecidos. Estos signos vitales de la gestión son: la eficacia, la eficiencia, la efectividad y la productividad.

Hasta aquí se han mencionado tres definiciones que son primordiales para hacerse a una idea base sobre el trabajo que será realizado en esta investigación. Ahora bien, se necesitan conceptualizaciones más globales que sirvan como apoyo teórico y expliquen de una forma más clara cada uno de los aspectos claves para comprender el proceso de actividades del sistema que será desarrollado, es por ello que a continuación se describirá toda la información que sea necesaria para tener más claro el objetivo del proyecto.

***Gestión de Proyectos***

La Guía del PMBOK (2008) define la Gestión de Proyectos como la aplicación del conocimiento, habilidades, herramientas y técnicas a las actividades del proyecto de forma tal que se cumpla con los requerimientos del mismo. Está enmarcada en cinco grupos de procesos denominados Iniciación, Planificación, Ejecución, Control y Seguimiento, y Cierre. Del mismo modo Pressman (2002) comenta que la Gestión de Proyectos de Software se centra en las cuatro P’s Personal, Producto, Proceso y Proyecto:

El gestor que olvide que el trabajo de ingeniería del software es un esfuerzo humano intenso nunca tendrá éxito en la gestión de proyectos. Un gestor que no fomenta una minuciosa comunicación con el cliente al principio de la evolución del proyecto se arriesga a construir una elegante solución para un problema equivocado. El administrador que presta poca atención al proceso corre el riesgo de arrojar métodos técnicos y herramientas eficaces al vacío. El gestor que emprende un proyecto sin un plan sólido arriesga el éxito del producto.

1. ***Grupo del Proceso de Iniciación***

Son aquellos procesos que se realizan para definir un nuevo proyecto o una nueva fase de un proyecto ya existente, mediante la obtención de la autorización para comenzar dicho proyecto o fase. Para ello se desarrolla el acta de constitución del proyecto la cual autoriza el proyecto o una fase del mismo, también se identifican los stakeholders o personas interesadas en el proyecto, estas son las que reciben el impacto del proyecto (Project Management Institute, Inc, 2008).

1. ***Grupo del Proceso de Planificación***

Corresponde a aquellos procesos requeridos para establecer el alcance del proyecto, refinar los objetivos y definir el curso de acción necesario para alcanzar los objetivos para cuyo logro se emprendió el proyecto. Entre los principales procesos de la planificación se encuentra el desarrollo del plan, identificación de requisitos, definición del alcance, creación del edt (estructura desglosada del trabajo),definición y secuencia de las actividades, estimación de las actividades, desarrollo del cronograma, estimación de costos, creación del presupuesto, planificación de calidad, gestión de riesgos, entre otras (Project Management Institute, Inc, 2008).

1. ***Grupo del Proceso de Ejecución***

Son aquellos procesos realizados para completar el trabajo definido en el plan para la gestión del proyecto con el fin de cumplir con las especificaciones del mismo. Entre los procesos importantes de este grupo se encuentran la gestión de la ejecución del proyecto, aseguramiento de calidad, adquisición del equipo del proyecto, desarrollo del equipo del proyecto, distribución de la información y gestión de las expectativas de los interesados (Project Management Institute, Inc, 2008).

1. ***Grupo del Proceso de Control y Seguimiento***

Corresponde a aquellos procesos requeridos para dar seguimiento, analizar y regular el progreso y el desempeño del proyecto, para identificar áreas en las que el plan requiera cambios y para iniciar los cambios correspondientes. Entre los procesos de este grupo se encuentran el seguimiento y control del proyecto, control integrado de cambios, verificación del alcance, control del alcance, control del cronograma, control de costos, control de calidad, información del desempeño, seguimiento y control del riesgo, y administración de las adquisiciones (Project Management Institute, Inc, 2008).

1. ***Grupo del Proceso de Cierre***

Son aquellos procesos realizados para finalizar todas las actividades a través de todos los grupos de procesos, con el fin de cerrar formalmente el proyecto o una fase del mismo. Este grupo lo conforma los procesos de cierre de proyecto o fase y el cierre de adquisiciones (Project Management Institute, Inc, 2008).

***Gestión de flotas***

La gestión de una flota de vehículos se basa en dos aspectos diferenciados y coordinados: el mantenimiento de los vehículos y el tráfico de la flota. La gestión del mantenimiento de grandes flotas de vehículos es una tarea compleja, que debe tener en cuenta diversos factores. El tipo de vehículos que la forman (turismos, furgonetas, vehículos industriales, etc.), su uso (urbano, carretera, intensivo, etc.), los planes de mantenimiento establecidos por los fabricantes de los vehículos y la realización, por medios propios o ajenos, de las operaciones en ellos incluidas son aspectos claves para gestionar con eficiencia una flota de vehículos. (González, 2008).

***Mantenimiento***

En Venezuela, este término está definido por la Norma Venezolana COVENIN 3049-93 en el literal 3.1.2 como: “Es el conjunto de acciones que permite conservar o reestablecer un sistema productivo a un estado específico para que pueda cumplir un servicio determinado” (p. 1). A su vez, el objetivo del mantenimiento se encuentra definido en su literal 3.1.4. como “… mantener un sistema productivo en forma adecuada de manera que pueda cumplir su misión, para lograr una producción esperada en las empresas de producción y una calidad de servicio exigida, en las empresas de servicio, a un costo global óptimo” (p. 1).

Se pueden encontrar infinidad de definiciones diferentes para el concepto de mantenimiento según los criterios de cada autor. Intentando homogeneizar diferentes criterios, podemos definir el mantenimiento como el conjunto de actividades que se realizan sobre un componente, equipo o sistema para asegurar que continúe desempeñando las funciones que se esperan de él, dentro de su contexto operacional. O en otras palabras, como la responsabilidad de mantener en buenas condiciones la maquinaria y herramientas del equipo de trabajo, lo cual permite un mejor desenvolvimiento y seguridad evitando conflictos en el área laboral.

El objetivo fundamental del mantenimiento, por tanto, es ejecutar distintas actividades tales como, pruebas inspección reemplazo, reinstalación, detección y análisis de fallas, calibración, reparación, modificación reconstrucción, lubricación y mejoramiento, a través de optimización de los recursos humanos y económicos, con el fin de mantener las condiciones de servicios establecidas según el diseño de los equipos y de lograr el alcance de vida útil de los mismos.

Es así que la organización y el departamento de mantenimiento debe levantar información permanente para tener puntualmente los objetivos de:

* Optimización de la disponibilidad del equipo productivo.
* Disminución de los costos de mantenimiento.
* Optimización de los recursos humanos.
* Maximización de la vida de la unidad.

El mantenimiento adecuado, tiende a prolongar la vida útil de los bienes, a obtener un rendimiento aceptable de los mismos durante más tiempo y a reducir el número de fallas. Es así que tenemos una confiabilidad de las unidades en 100% aproximadamente.

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

1. ***Fallas Tempranas:*** Ocurren al principio de la vida útil y constituyen un porcentaje pequeño del total de fallas. Pueden ser causadas por problemas de materiales, de diseño o de montaje.
2. ***Fallas Adultas:*** Son las fallas que presentan mayor frecuencia durante la vida útil. Son derivadas de las condiciones de operación y se presentan más lentamente que las anteriores (suciedad en un filtro de aire, cambios de rodamientos de un vehículo, etc.).
3. ***Fallas Tardías:*** Representan una pequeña fracción de las fallas totales, aparecen en forma lenta y ocurren en la etapa final de la vida del bien (envejecimiento de la aislación de un pequeño motor eléctrico, perdida de flujo luminoso de una lámpara, etc.)

En la actualidad el reto de la industria venezolana, es la mejor utilización de sus recursos para obtener la mayor calidad del producto con una cantidad exigida en una empresa de producción. Para ello se hace necesario asegurar a través del mantenimiento disponibilidad de los equipos teniendo presentes los nuevos avances tecnológicos, los cuales facilitan ideas propuestas.

La clasificación básica del mantenimiento se divide en principalmente dos grandes actividades el Preventivo y el Correctivo. Pero, las diferentes tendencias a confundir los límites que separan dichas actividades, suponen una clasificación más completa, subdividiendo estas grandes actividades en seis tipos:

• Mantenimiento Rutinario.

• Mantenimiento Programado.

• Mantenimiento por Avería o Reparación.

• Mantenimiento Correctivo.

• Mantenimiento Circunstancial

• Mantenimiento Preventivo.

***Mantenimiento Rutinario***

Según la Norma COVENIN 3049-93 en el apartado de Tipos de Mantenimiento específicamente en la definición dada en el literal 3.1.10.1 se define como:

Es el que comprende actividades tales como: Lubricación, limpieza, protección, ajustes, calibración u otras. Su frecuencia de ejecución es hasta períodos semanales; generalmente es ejecutado por los mismos operarios de los sistemas productivos y su objetivo es mantener y alargar la vida útil de dichos sistemas productivos evitando su desgaste. (p. 1).

***Mantenimiento Programado***

La Norma COVENIN 3049-93 en el literal 3.1.10.2 describe que el mantenimiento programado:

Toma como basamento las instrucciones técnicas recomendadas por los fabricantes, constructores, diseñadores, usuarios y experiencias conocidas, para obtener ciclos de revisión y sustitución para los elementos más importantes de un sistema productivo a objeto de determinar la carga de trabajo que es necesaria programar. Su frecuencia de ejecución cubre desde quincenal hasta generalmente períodos de un año. Es ejecutado por las cuadrillas de la organización de mantenimiento que se dirigen al sitio para realizar las labores incorporadas en un calendario anual. (p. 1).

**Mantenimiento por Avería o Reparación**

La Norma COVENIN 3049-93 en el literal 3.1.10.3 define el mantenimiento poravería o reparación como:

La atención a un sistema productivo cuando aparece una falla. Su objeto es mantener en servicio adecuadamente dicho sistema minimizando sus tiempos de parada. Es ejecutado por el personal de la Organización Mantenimiento. La atención a las fallas debe ser inmediata, por tanto no da tiempo a ser programada pues implica el aumento en costos y de paradas innecesarias de personal y equipos (p. 2).

***Mantenimiento Correctivo***

Cuando se hace mantenimiento correctivo en un sistema preventivo, se le llama corrección de falla. Por otra parte, la definición presentada por la norma antes mencionada en su literal 3.1.10.4 para el Mantenimiento correctivo es la siguiente:

Comprende las actividades de todo tipo encaminadas a tratar de eliminar la necesidad de mantenimiento corrigiendo las fallas de una manera integral a mediano plazo. Las acciones más comunes que se realiza son: modificación de elementos de máquinas, modificación de alternativas de procesos, cambio de especificaciones, ampliaciones, revisión de elementos básicos de mantenimiento y conservación. Este tipo de actividades es ejecutado por el personal de la organización mantenimiento y por entes foráneos dependiendo de la magnitud, costos, especialización necesaria y otros; su intervención tiene que ser planificada y programada en el tiempo para que su ataque evite paradas injustificadas (p. 2).

Smith (1993) define al mantenimiento correctivo como “la realización de tareas de mantenimiento no planeadas para restaurar las capacidades funcionales del equipo que falla o que trabaja mal y de los sistemas”. (p. 10)

***Mantenimiento Circunstancial***

La Norma COVENIN 3049-93 en el literal 3.1.10.5 describe el mantenimiento circunstancial como:

Es una mezcla entre rutinario, programado, avería y correctivo ya que por su intermedio se ejecutan acciones de rutina pero no tienen un punto fijo en el tiempo para iniciar su ejecución, porque los sistemas atendidos funcionan de manera alterna; se ejecutan acciones que están programadas en un calendario anual pero que tampoco tienen un punto fijo de inicio por la razón anterior; se detienen averías cuando el sistema se detiene, existiendo por supuesto otro sistema que cumpla su función, y el estudio de la falla permite la programación de su corrección eliminando dicha avería a mediano plazo. La atención de los equipos bajo este tipo de mantenimiento depende no de la organización del mantenimiento que tiene a dichos equipos dentro de sus planes y programas, sino de otros entes de la organización, los cuales sugieren aumento en la capacidad de producción, cambios de procesos, disminución de ventas, reducción de personal y/o turnos de trabajo (p. 2).

***Mantenimiento Preventivo***

Cuando se hace mantenimiento preventivo dentro de un sistema correctivo,

se le llama mantenimiento rutinario. La Norma COVENIN 3049-93 en el literal 3.1.10.6 expresa que en el mantenimiento preventivo:

El estudio de fallas de un equipo deriva dos tipos de averías; aquellas que generan resultados que obliguen a la atención de los equipos mediante Mantenimiento Correctivo y las que se presentan con cierta regularidad y que ameriten su prevención. El Mantenimiento Preventivo es el que utiliza todos los medios disponibles, incluso los estadísticos, para determinar la frecuencia de las inspecciones, revisiones, sustitución de piezas claves, probabilidad de aparición de averías, vida útil, y otras. Su objetivo es adelantarse a la aparición o predecir la presencia de fallas (p. 2).

Este tipo de mantenimiento busca garantizar que las condiciones normales de operación de un equipo o sistema sean respetadas, es decir, que el equipo esté libre de polvo, sus lubricantes conserven sus características y sus elementos consumibles tales como filtros, mangueras, correas, sean sustituidas dentro de su vida útil.

El Mantenimiento Preventivo clásico prevé fallas a través de sus cuatro áreas básicas:

**a) Limpieza:** las máquinas limpias son más fáciles de mantener operan mejor y reducen la contaminación. La limpieza constituye la actividad más sencilla y eficaz para reducir desgastes, deterioros y roturas.

**b) Inspección:** se realizan para verificar el funcionamiento seguro, eficiente y económico de la maquinaria y equipo. El personal de mantenimiento deberá reconocer la importancia de una inspección objetiva para determinar las condiciones del equipo. Con las informaciones obtenidas por medio de las inspecciones, se toman las decisiones a fin de llevar a cabo el mantenimiento adecuado y oportuno.

**c) Lubricación:** un lubricante es toda sustancia que al ser introducida entre dos partes móviles, reduce el frotamiento calentamiento y desgaste, debido a la formación de una capa resbalante entre ellas. La lubricación es la acción realizada por el lubricante. Aunque esta operación es normalmente realizada de acuerdo con las especificaciones del fabricante, la ubicación física y geográfica del equipo y maquinaria; además de la experiencia, puede alterar las recomendaciones.

**d) Ajuste:** Es una consecuencia directa de la inspección; ya que es a través de ellas que se detectan las condiciones inadecuadas de los equipos y maquinarias, evitándose así posibles fallas.

El mantenimiento preventivo se realiza normalmente a través de inspecciones y operaciones sistemáticas. Estas se pueden realizar con el equipo en marcha, inmovilizado pero sin necesidad de desmontaje, inmovilizado con desmontaje. Puede asumir también la forma de sustituciones sistemáticas de componentes, órganos o equipos completos, que busquen prolongar la vida útil del sistema, disminuyendo la probabilidad de ocurrencia de fallas de estos elementos, normalmente en su etapa de desgaste.

Cuando la falla se presenta de manera progresiva, pueden monitorearse ciertos parámetros físicos que permiten decidir la intervención del equipo antes de la ocurrencia de la falla. Este tipo de mantenimiento no contemplado en la clasificación general del mantenimiento por ser una subdivisión del Mantenimiento Preventivo se conoce como **Mantenimiento por Condición o Predictivo** ya que busca efectuar la reparación del equipo en el umbral de ocurrencia de la falla, es decir, en el preciso momento de su aparición; bajo condiciones programadas, minimizando así los costos globales de mantenimiento. Este mantenimiento se lleva acabo usando herramientas de predicción física (ultrasonidos, rayos X, termografía, vibrometría, análisis espectográficos de lubricantes o estadísticos uso de técnicas de confiabilidad).

Cuando el mantenimiento preventivo implica la puesta a nuevo de un equipo a través de la sustitución sistemática de todos sus componentes que muestran desgaste, se habla también de un Overhaul, gran revisión o revisión mayor, que pueden ser hechos por etapas (cambiando un componente a la vez) o de manera global, como es el caso de las paradas de planta.

El mantenimiento preventivo según Smith (1993) es “el cumplimiento de las tareas de inspección y/o de servicio que han sido planeadas para mantener las capacidades funcionales del equipo operativo y de los sistemas en un tiempo específico”. (p. 10). Según Smith, existen tres razones principales por las que se debe hacer el mantenimiento preventivo:

* Prevenir fallas.
* Detectar la aparición de fallas.
* Descubrir fallas escondidas.

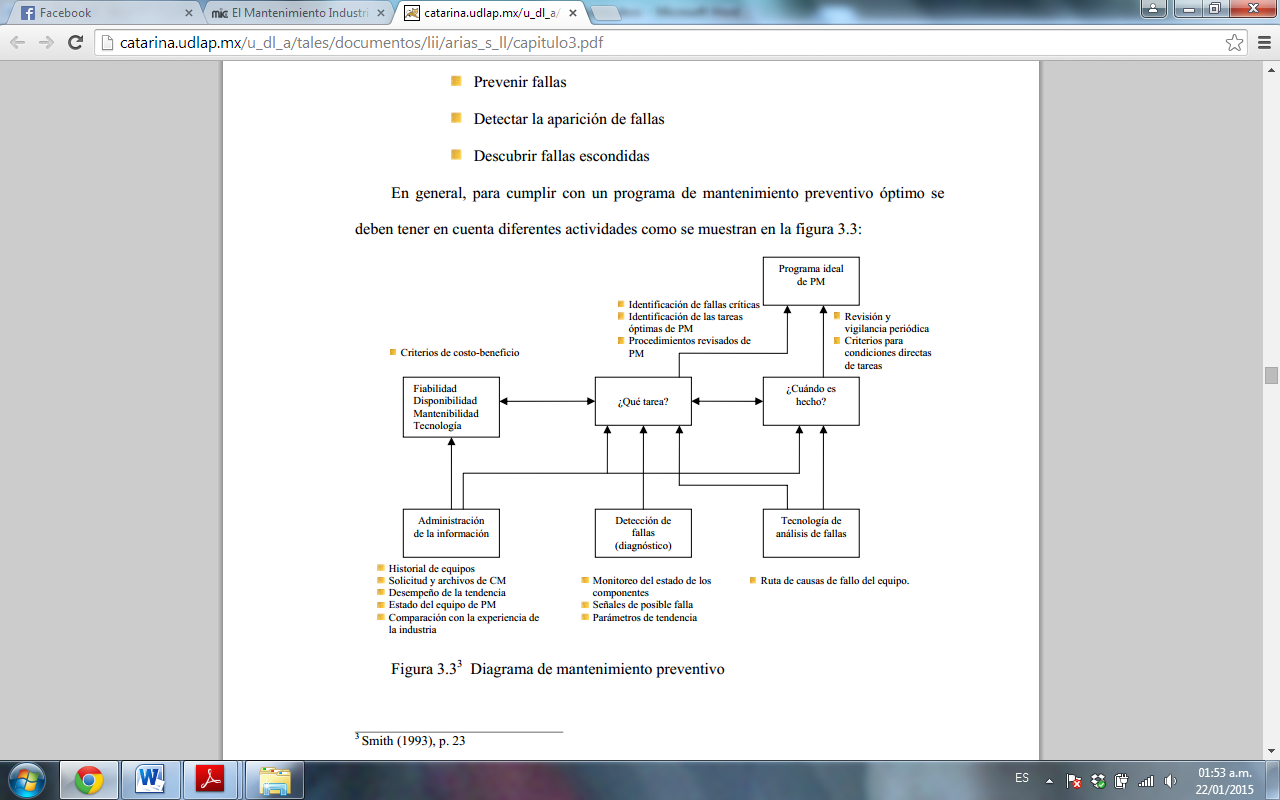
En general, para cumplir con un programa de mantenimiento preventivo optimo se debe tener en cuenta diferentes actividades como se muestra en la optimo se debe tener en cuenta diferentes actividades como se muestra en lafigura 2.

Figura 2.

**Figura 2**. Diagrama de Mantenimiento Preventivo. Smith. 1993. p.23

***Ventajas:***

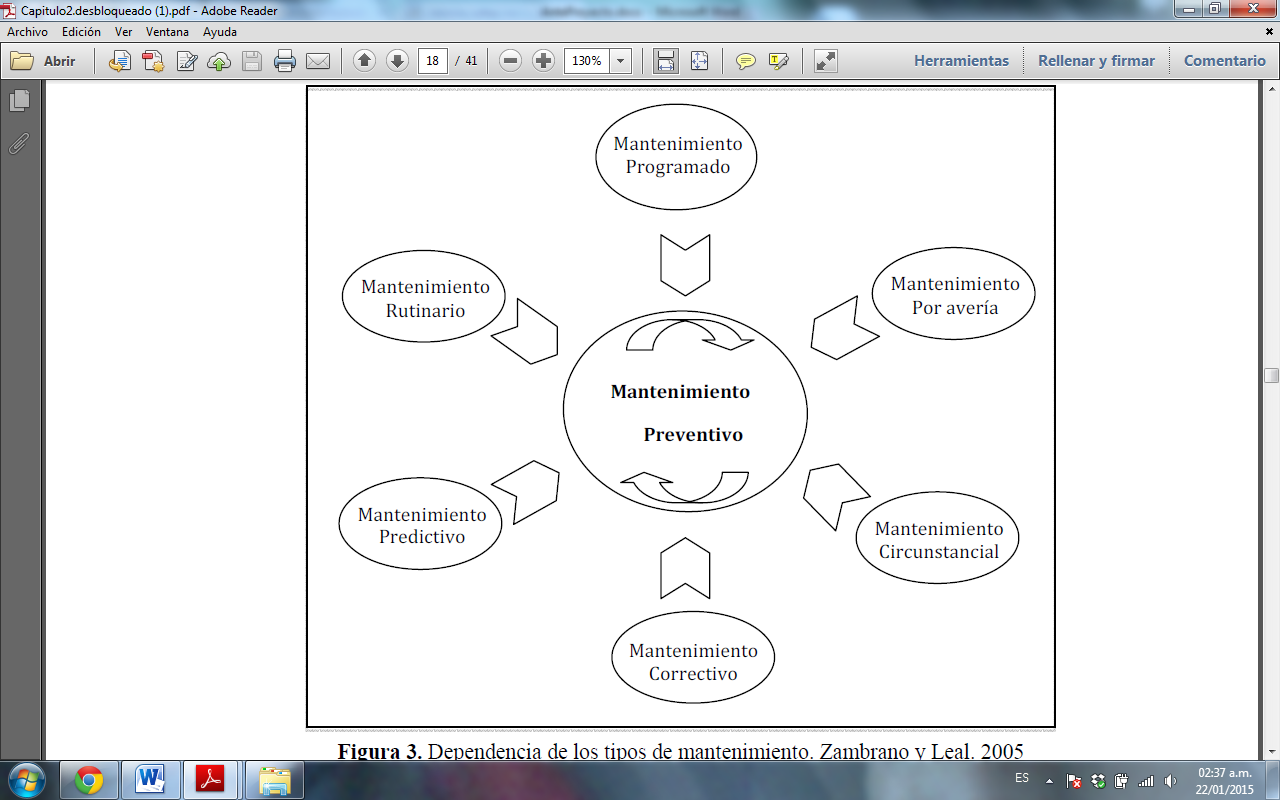
1. Si se hace correctamente, exige un conocimiento de las máquinas y un tratamiento de los históricos que ayudará en gran medida a controlar la maquinaria e instalaciones.
2. El cuidado periódico conlleva un estudio óptimo de conservación con
3. la que es indispensable una aplicación eficaz para contribuir a un correcto sistema de calidad y a la mejora de los continuos.
4. Reducción del correctivo representará una reducción de costos de producción y un aumento de la disponibilidad, esto posibilita una planificación de los trabajos del departamento de mantenimiento, así como una previsión de los recambios o medios necesarios.
5. Se concreta de mutuo acuerdo el mejor momento para realizar el paro de las instalaciones con producción.

***Desventajas:***

1. Representa una inversión inicial en infraestructura y mano de obra. El desarrollo de planes de mantenimiento se debe realizar por técnicos especializados.
2. Si no se hace un correcto análisis del nivel de mantenimiento
3. preventivo, se puede sobrecargar el costo de mantenimiento sin mejoras sustanciales en la disponibilidad.
4. Los trabajos rutinarios cuando se prolongan en el tiempo produce falta de motivación en el personal, por lo que se deberán crear sistemas
5. imaginativos para convertir un trabajo repetitivo en un trabajo que genere satisfacción y compromiso, la implicación de los operarios de preventivo es indispensable para el éxito del plan.

Por último es importante señalar que existe dependencia entre todos los tipos de mantenimiento mencionados anteriormente como se puede observar en la figura 3. Según Zambrano (2010):

Se evidencia la dependencia de todos los tipos de mantenimiento necesarios para el logro de los objetivos del mismo, es por esta razón que el mantenimiento rutinario, programado, por avería, circunstancial, predictivo, correctivo y productivo, realizan sus actividades en pro del mantenimiento preventivo (p.25).



**Figura 3.** Dependencia de los tipos de mantenimiento. Zambrano y Leal. 2005

***Gestión de Mantenimiento***

Según la Norma COVENIN 3049-93 en el apartado 3.1.3 define la gestión de mantenimiento como “la efectiva y eficiente utilización de los recursos materiales, económicos, humanos y de tiempo para alcanzar los objetivos de mantenimiento” (p. 1).

En lo que respecta, la concepción del mantenimiento bajo una visión actual no implica reparar un equipo dañado tan pronto como se pueda sino mantener el equipo en operación a los niveles especificados. En consecuencia, un buen mantenimiento no consiste en realizar el trabajo equivocado en la forma más eficiente; su prioridad es prevenir fallas y, de este modo reducir los riesgos de paradas imprevistas (Zambrano, 2010).

Además según Zambrano el mantenimiento no empieza cuando los equipos e instalaciones son recibidos y montados, sino en la etapa inicial de todo proyecto, y continúa cuando se formaliza la compra de aquellos y su montaje correspondiente. De esta forma se describe el proceso de gestión de mantenimiento donde no es atacar una falla, sino hacer seguimiento a los bienes pertenecientes a la empresa utilizando de la manera más eficiente los recursos con los que se cuenta dentro de la organización. La gestión de mantenimiento según Zambrano y Leal (2005) se define como:

Un proceso sistemático donde, a través de una serie de medidas organizativas se pueden planear las acciones de mantenimiento, de acuerdo a procedimientos que lleven un orden o secuencia lógica a fin de lograr alta confiabilidad y disponibilidad de los objetos a mantener. (p. 11)

El mantenimiento de equipos, infraestructuras, herramientas, maquinarias, representa una inversión que a mediano y largo plazo acarreará ganancias no sólo para el empresario, a quien esta inversión se le revertirá en mejoras en su producción, sino también el ahorro que representa tener trabajadores sanos e índices de accidentalidad bajos.

El mantenimiento no solo debe ser realizado por el departamento encargado de esto. El trabajador debe ser concientizado a mantener en buenas condiciones las unidades, esto permitirá mayor responsabilidad del trabajador y prevención de accidentes.

***Organización del Mantenimiento***

Una organización de mantenimiento puede ser de diversos tipos, pero en todos ellos aparecen los tres componentes siguientes:

1. ***Recursos:*** Comprende personal, repuestos y herramientas, con un tamaño, composición, localización y movimientos determinados.
2. ***Administración:*** Una estructura jerárquica con autoridad y responsabilidad que decida que trabajo se harán, y cuando y como debe llevarse a cabo.
3. ***Planificación del trabajo y sistema de control:*** Un mecanismo para planificar y programar el trabajo, y garantizar la recuperación de la información necesaria para que el esfuerzo de mantenimiento se dirija correctamente hacia el objetivo definido.

La totalidad del sistema de mantenimiento es un organismo en continua evolución, cuya organización necesitara una modificación continua como respuesta a unos requisitos cambiantes. Como el objetivo principal de la organización es hacer corresponder los recursos con la carga de trabajo, es preciso considerar estas características antes de detallar los tres componentes básicos mencionados.(Peinado, 2009)

***Objetivos Fundamentales del Mantenimiento***

Los principales objetivos del mantenimiento, manejados con criterio económico y encausado a un ahorro en los costos generales de producción son:

1. Llevar a cabo una inspección sistemática de todas las instalaciones, con intervalos de control para detectar oportunamente cualquier desgaste o rotura, manteniendo los registros adecuados.
2. Mantener permanentemente los equipos e instalaciones, en su mejor estado para evitar los tiempos de parada que aumentan los costos.
3. Efectuar las reparaciones de emergencia lo más pronto, empleando métodos más fáciles de reparación.
4. Prolongar la vida útil de los equipos e instalaciones al máximo.
5. Sugerir y proyectar mejoras en la maquinaria y equipos para disminuir las posibilidades de daño y rotura.
6. Controlar el costo directo del mantenimiento mediante el uso correcto y eficiencia del tiempo, materiales, hombres y servicio.
7. Disminuir los accidentes laborales. (Peinado, 2009)

***Indicadores de Gestión***

Un indicador se define como la relación entre las variables cuantitativas o cualitativas, que permiten observar la situación y las tendencias de cambio generadas en el objeto o fenómeno observado, respecto de objetivos y metas previstos, influencias esperadas. (Beltrán, 2000).

De tal manera se entiende que los indicadores de gestión pueden ser valores, unidades, índices, series o estadísticas, es decir, que es como la expresión cuantitativa del comportamiento o el desempeño de toda una organización o una de sus partes, cuya magnitud al ser comparada con algún nivel de referencia, puede estar señalando una desviación sobre la cual se tomaran acciones correctivas o preventivas según el caso.

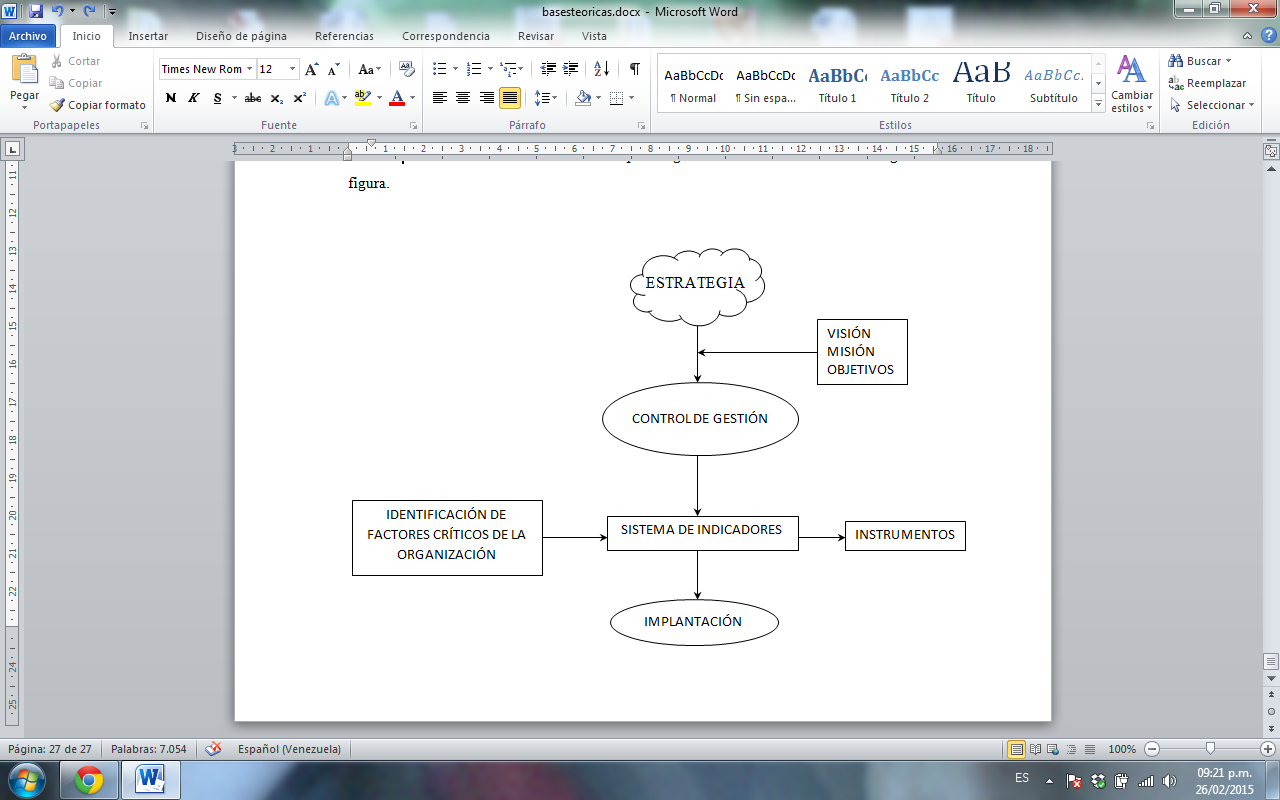
Es importante que los [indicadores de gestión](http://www.gestiopolis.com/administracion-estrategia-2/gestion-del-conocimiento-e-indicadores-de-gestion.htm) reflejen datos veraces y fiables, ya que el análisis de la situación, de otra manera, no será correcto. Por otra parte, si los indicadores son ambiguos, la interpretación será complicada. Lo que permite un indicador de gestión es determinar si un proyecto o una organización están siendo exitosos o si están cumpliendo con los objetivos. El líder de la organización es quien suele establecer los indicadores de gestión, que son utilizados de manera frecuente para evaluar desempeño y resultados.

De igual modo hay que tener presente que los indicadores de gestión son un medio y no un fin, ya que el indicador es un apoyo para saber cómo se encuentra la organización. Es un indicio expresado numéricamente o en forma de concepto, sobre el grado de eficiencia o eficacia de las operaciones de la entidad, una dependencia o un área. El indicador compara dos cifras o datos. Con base en su interpretación se puede cualificar una acción y orientar análisis más detallados en los aspectos en los que se presume desviaciones. El indicador facilita el control y el autocontrol y por consiguiente la toma de decisiones, en la medida en que sea posible relacionarlos con cantidad, calidad, costos, oportunidad y productividad.

Los indicadores de gestión son por encima de todo una información, por lo cual no se puede indicar que simplemente es un dato determinado de calificación de la empresa; teniendo en cuenta que es una información, los indicadores de gestión deben tener los atributos de la información, tanto en forma individual como grupal. Según Senn (1990) los indicadores de gestión poseen los siguientes atributos de la información:

* ***Exactitud:*** La información debe representar la situación o el estado como realmente es.
* ***Forma:*** Existen diversas formas de presentación de la información que puede ser cuantitativa o cualitativa, numérica o gráfica, impresa o visualizada, resumida y detallada. Realmente la forma debe ser elegida según la situación, necesidades y habilidades de quien recibe y procesa.
* ***Frecuencia:*** Es la medida de cuán a menudo se requiere, se recaba, se produce o se analiza.
* ***Extensión:*** Se refiere al alcance en términos de cobertura del área de interés. Además tiene que ver con la brevedad requerida, según el tópico de que se trate. La calidad de la información no es directamente proporcional con su extensión.
* ***Origen:*** Puede originarse dentro o fuera de la organización. Lo fundamental es que la fuente que la genera sea la fuente correcta.
* ***Temporalidad:*** La información puede hablarnos del pasado, de los sucesos actuales o de las actividades o sucesos futuros.
* ***Relevancia:*** La información es relevante si es importante si es necesaria para una situación particular.
* ***Integridad:*** Una información completa proporciona al usuario el panorama integral de lo que necesita saber acerca de una situación determinada.
* ***Oportunidad:*** Para ser considerada oportuna, una información debe estar disponible y actualizada cuando se la necesite.

El diseño del sistema de indicadores puede gráficamente visualizarse en la siguiente figura.



**Figura 4.** Diseño de Indicadores. Elaborado por Autores. 2015.

Dentro de la concepción sistémica del modelo integral de gestión, según Serna (2003) los indicadores pueden clasificarse por sus características en dos grandes grupos:

* ***Indicadores gerenciales o de desempeño:*** Son aquellos que miden resultados de objetivos a corto plazo sobre cuyos resultados puede modificarse muy poco. Tal es el caso de los indicadores financieros. Una vez se obtienen, muy poco puede hacerse para hacer cambios si no se afectan los factores que causaron el resultado.
* ***Indicadores impulsores o motores de desempeño:*** Conjunto de indicadores que impulsan o lideran acciones para generar resultados en el largo plazo. Tal es el caso de indicadores e índices relacionados con el desempeño de la organización frente al cliente y al mercado. Índices relacionados con el mejoramiento en productividad de la organización y los relacionados con la generación de conocimiento de la organización. Los impulsores de desempeño generan resultados en el largo plazo, pero exigen acciones de intervención en el corto plazo.

En resumen los indicadores de gestión:

* Son medios, instrumentos o mecanismos para evaluar hasta qué punto o en qué medida se están logrando los objetivos estratégicos.
* Representan una unidad de medida gerencial que permite evaluar el desempeño de una organización frente a sus metas, objetivos y responsabilidades con los [grupos](http://www.monografias.com/trabajos11/grupo/grupo.shtml) de referencia.
* Producen información para analizar el desempeño de cualquier área de la organización y verifican el cumplimiento de los objetivos en términos de resultados.
* Detectan y prevén desviaciones en el logro de los objetivos.
* El [análisis](http://www.monografias.com/trabajos11/metods/metods.shtml#ANALIT) de los indicadores conlleva a generar ALERTAS SOBRE LA ACCIÓN, no perder la dirección, bajo el supuesto de que la organización está perfectamente alineada con el plan.

***¿Qué se debe esperar de un sistema con indicadores?***

* Que se convierta en un sistema de alertas tempranas "Pre-alarmas"
* Que determine las tendencias y la causa raíz del comportamiento productivo.
* Que establezca la relación entre el valor agregado y el [costo](http://www.monografias.com/trabajos7/coad/coad.shtml#costo) [laboral](http://www.monografias.com/trabajos13/renla/renla.shtml) para definir el tamaño y el valor óptimo del equipo humano.
* Que relacione la productividad del [capital](http://www.monografias.com/trabajos13/capintel/capintel.shtml) humano, la del capital físico, la rentabilidad, el endeudamiento y la liquidez con el fin de garantizar [equilibrio](http://www.monografias.com/trabajos/tomadecisiones/tomadecisiones.shtml).
* Que facilite la toma de decisiones, que permita construir [conocimiento](http://www.monografias.com/trabajos/epistemologia2/epistemologia2.shtml), que oriente a las personas, que alimente las [políticas](http://www.monografias.com/trabajos10/poli/poli.shtml), que permita operar procesos productivos.

***Sistemas de Información para la Gestión de Mantenimiento***

Según Zambrano (2010), el sistema dedicado al mantenimiento, es un elemento vital para proporcionar a producción su máxima disponibilidad de equipo. Un sistema de mantenimiento bien cimentado no solo proporciona disponibilidad de máquinas, equipos herramentales, y recursos, sino que, incrementa su tiempo de vida y reduce su costo operativo (p. 30).

Es por eso que, la conjunción de un sistema de información automatizado con la teoría aplicada en la Gestión de Mantenimiento, asegura que por lo menos las actividades sean planificadas de manera correcta. Generando avisos que mantengan alerta a los administradores de la ejecución de actividades venideras. Actualmente, un exitoso sistema de mantenimiento requiere un fuerte compromiso gerencial, participación y soporte. Todos los niveles gerenciales de los departamentos deben estar dedicados a sanear las prácticas del mantenimiento (Zambrano, 2010).

Ante esto, el mantenimiento preventivo debe ser una parte integral de las actividades de cada departamento. El equipo y los recursos deben estar disponibles para el mantenimiento programado y las inspecciones, si se desea que el sistema de mantenimiento tenga éxito (Becerra, 2001).

Cuando los recursos y el equipo se han dispuesto, el nivel gerencial debe asegurar el cumplimiento de la programación de mantenimiento e inspecciones a través de una coordinación eficaz de las actividades productivas y de mantenimiento.

Becerra (2001) afirma que las operaciones de mantenimiento en la organización deben tener una fuerte estructura para asegurar el éxito del mantenimiento preventivo. Un verdadero entendimiento y compromiso hacia las metas y objetivos deben ser evidentes en toda la organización.

***Programación Orientada a Objetos (POO)***

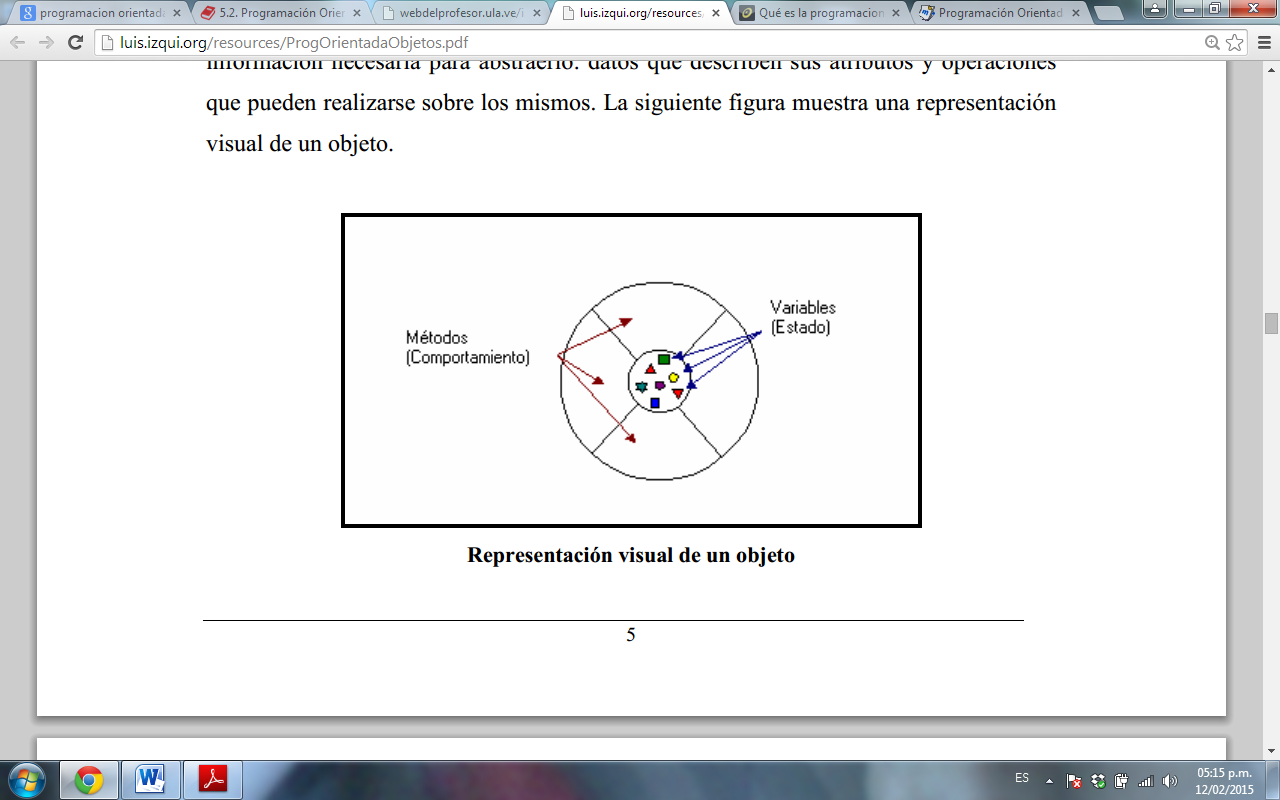
La herramienta empleada para el desarrollo de este trabajo de investigación será la programación orientado a objetos, dado que a pesar que tiende a demorar el desarrollo en etapas tempranas, esta tecnología hace que el mantenimiento se transforme en una actividad más simple, así como también la reutilización de código empleado en futuras aplicaciones web.

Según Carballo (2007) la programación Orientada a Objetos es una metodología que basa la estructura de los programas en torno a los objetos. Los lenguajes de POO ofrecen medios y herramientas para describir los objetos manipulados por un programa. Más que describir cada objeto individualmente, estos lenguajes proveen una construcción (clase) que describe a un conjunto de objetos que poseen las mismas propiedades.

La POO tiene tres características básicas: debe estar basado en objetos, basado en clases y capaz de tener [herencia](http://www.monografias.com/trabajos13/heren/heren.shtml) de clases, además el enfoque orientado a objeto se basa en cuatro principios que constituyen la base de todo desarrollo orientado a objetos, estos principios son: la abstracción, el encapsulamiento, la modularidad y la herencia. Otros elementos a destacar, aunque no fundamentales son: polimorfismo, enlace dinámico, concurrencia y persistencia.

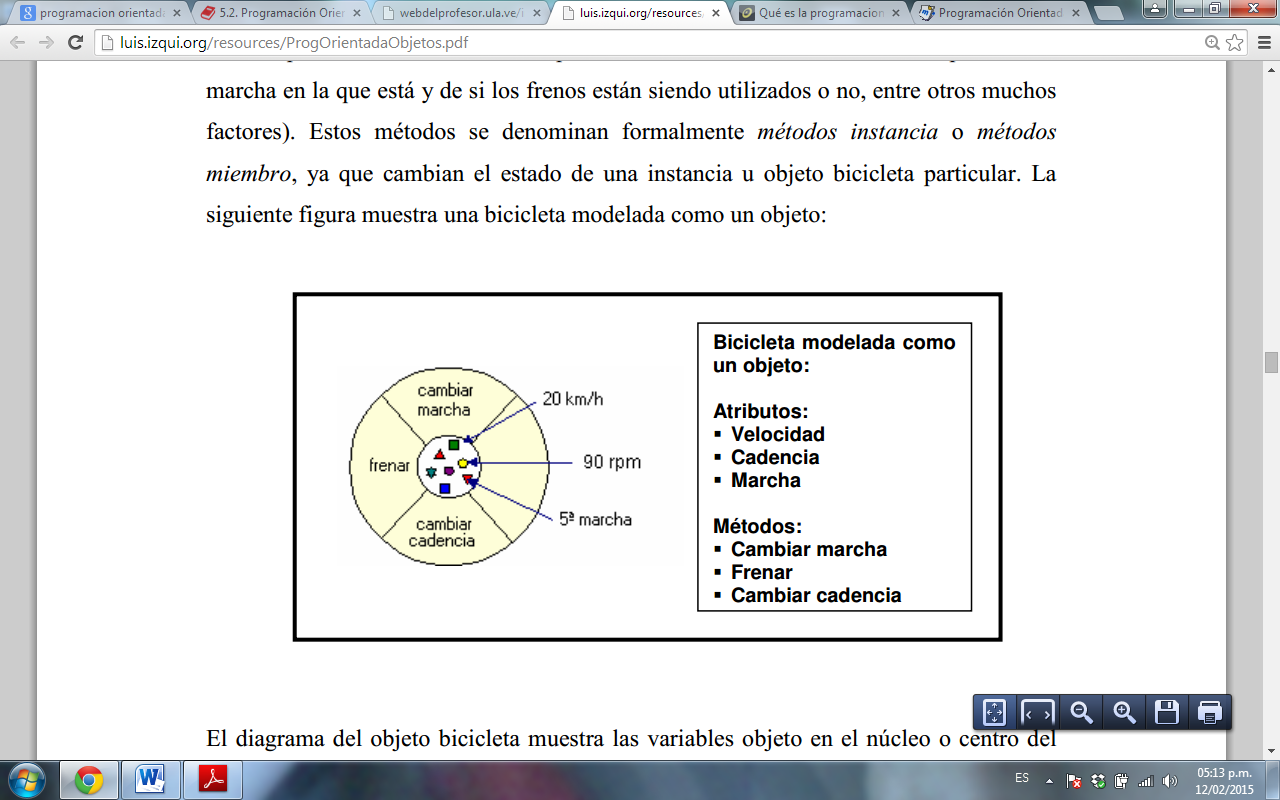
Ahora bien, el elemento fundamental de la POO es, como su nombre lo indica, el objeto. Por ello definimos un objeto como una entidad que encapsula datos (atributos) y acciones o funciones que los manejan (métodos). También se puede definir como una instancia o particularización de una clase.

De acuerdo a lo anterior¸ un objeto no es más que un conjunto de variables o datos, y métodos o funciones relacionados entre sí. Los objetos en programación se usan para modelar objetos o entidades del mundo real. Un objeto es, por tanto, la representación en un programa de un concepto, y contiene toda la información necesaria para abstraerlo: datos que describen sus atributos y operaciones que pueden realizarse sobre los mismos. La siguiente figura muestra una representación visual de un objeto.

  
**Figura 5.** Representación visual de un objeto.Izquierdo.2001

Los atributos del objeto (estado) y lo que el objeto puede hacer (comportamiento) están expresados por las variables y los métodos que componen el objeto respectivamente. Por ejemplo, un objeto que modelase una bicicleta en el mundo real tendría variables que indicarían el estado actual de la bicicleta: su velocidad es de 20 km/h, su cadencia de pedaleo 90 r.p.m. y su marcha actual es la 5ª. Estas variables se conocen formalmente como variables instancia o variable miembro porque contienen el estado de un objeto bicicleta particular y, en programación orientada a objetos, un objeto particular se denomina una instancia.

Además de estas variables, el objeto bicicleta podría tener métodos para frenar, cambiar la cadencia de pedaleo, y cambiar de marcha (la bicicleta no tendría que tener un método para cambiar su velocidad pues ésta es función de la cadencia de pedaleo, la marcha en la que está y de si los frenos están siendo utilizados o no, entre otros muchos factores). Estos métodos se denominan formalmente métodos instancia o métodos miembro, ya que cambian el estado de una instancia u objeto bicicleta particular. La siguiente figura muestra una bicicleta modelada como un objeto:



**Figura 6.** Bicicleta modelada como un objeto.Izquierdo.2001

El diagrama del objeto bicicleta muestra las variables objeto en el núcleo o centro del objeto y los métodos rodeando el núcleo y protegiéndolo de otros objetos del programa. Este hecho de empaquetar o proteger las variables miembro con los métodos miembro se denomina encapsulación. Este dibujo conceptual que muestra el núcleo de variables miembro del objeto protegido por una membrana protectora de métodos o funciones miembro es la representación ideal de un objeto y es el ideal que los programadores de objetos suelen buscar. Sin embargo, debemos matizarlo. A menudo, por razones prácticas, es posible que un objeto desee exponer alguna de sus variables miembro, o proteger otras de sus propios métodos o función miembro. Por ejemplo, Java permite establecer cuatro niveles de protección de las variables y de la función miembro para casos como éste. Los niveles de protección determinan qué objetos y clases pueden acceder a qué variables o a qué métodos.

De cualquier forma, el hecho de encapsular las variables y las funciones miembro relacionadas proporciona dos importantes beneficios a los programadores de aplicaciones:

1. ***Capacidad de crear módulos:*** El código fuente de un objeto puede escribirse y mantenerse independiente del código fuente del resto de los objetos. De esta forma, un objeto puede pasarse fácilmente de una parte a otra del programa. Podemos dejar nuestra bicicleta a un amigo, y ésta seguirá funcionando.
2. ***Protección de información:*** Un objeto tendrá una interfaz pública perfectamente definida que otros objetos podrán usar para comunicarse con él. De esta forma, los objetos pueden mantener información privada y pueden cambiar el modo de operar de sus funciones miembros sin que esto afecte a otros objetos que usen estas funciones miembro. Es decir, no necesitamos entender cómo funciona el mecanismo de cambio de marcha para hacer uso de él.

Los objetos de interés durante el desarrollo de software no sólo son tomados de la vida real (objetos visibles o tangibles), también pueden ser abstractos. En general son entidades que juegan un rol bien definido en el dominio del problema. Cada objeto puede ser considerado como un proveedor de servicios utilizados por otros objetos que son sus clientes. Cada objeto puede ser a la vez proveedor y cliente. De allí que un programa pueda ser visto como un conjunto de relaciones entre proveedores y clientes.

La programación orientada a objetos, presenta una gran ventaja para el desarrollo de sistemas informáticos, especialmente por su mantenimiento y adaptabilidad de los cambios de procesos. Dado que los objetos son portables (teóricamente) mientras que la herencia permite la reusabilidad del código orientado a objetos, es más sencillo modificar código existente, porque los objetos no interaccionan excepto a través de mensajes; en consecuencia un [cambio](http://www.monografias.com/trabajos2/mercambiario/mercambiario.shtml) en la [codificación](http://www.monografias.com/trabajos37/codificacion/codificacion.shtml) de un objeto no afectará la operación con otro objeto siempre que los métodos respectivos permanezcan intactos.

La [introducción](http://www.monografias.com/trabajos13/discurso/discurso.shtml) de [tecnología](http://www.monografias.com/Tecnologia/index.shtml) de objetos como una herramienta conceptual para analizar, diseñar e implementar aplicaciones permite obtener productos más modificables, fácilmente extensibles y a partir de componentes reusables. Esta reusabilidad del código disminuye el [tiempo](http://www.monografias.com/trabajos901/evolucion-historica-concepciones-tiempo/evolucion-historica-concepciones-tiempo.shtml) que se utiliza en el desarrollo, y hace que el desarrollo del software sea más intuitivo porque la gente piensa naturalmente en términos de objetos, más que en términos de [algoritmos](http://www.monografias.com/trabajos15/algoritmos/algoritmos.shtml) de software.

***Lenguaje de Modelado Unificado (UML)***

Tal como indica su nombre, UML es un lenguaje de modelado. Un modelo es una simplificación de la realidad. El objetivo del modelado de un sistema es capturar las partes esenciales del sistema. Para facilitar este modelado, se realiza una abstracción y se plasma en una notación gráfica. Esto se conoce como modelado visual. El modelado visual permite manejar la complejidad de los sistemas a analizar o diseñar.

UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten. Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).

UML es además un método formal de modelado. Esto aporta las siguientes ventajas:

* Mayor rigor en la especificación.
* Permite realizar una verificación y validación del modelo realizado.
* Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

1. ***Visualizar:*** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
2. ***Especificar:*** UML permite especificar cuáles son las características de un sistema antes de su construcción.
3. ***Construir:*** A partir de los modelos especificados se pueden construir los sistemas diseñados.
4. ***Documentar:*** Los propios elementos gráficos sirven como documentación del

sistema desarrollado que pueden servir para su futura revisión.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es los suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (workflow) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware.  
Un modelo UML está compuesto por tres clases de bloques de construcción:

1. ***Elementos:*** Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
2. ***Relaciones:*** relacionan los elementos entre sí.
3. ***Diagramas:*** Son colecciones de elementos con sus relaciones.

Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. UML incluye los siguientes diagramas:

* Diagrama de casos de uso.
* Diagrama de clases.
* Diagrama de objetos.
* Diagrama de secuencia.
* Diagrama de colaboración.
* Diagrama de estados
* Diagrama de actividades.
* Diagrama de componentes.
* Diagrama de despliegue.

El diagrama de casos de usos representa gráficamente los casos de uso que tiene un sistema. Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir, se está diciendo lo que tiene que hacer un sistema y cómo. El diagrama de clases muestra un conjunto de clases, interfaces y sus relaciones. Éste es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos. En el diagrama de secuencia se muestra la interacción de los objetos que componen un sistema de forma temporal.

El resto de diagramas muestran distintos aspectos del sistema a modelar. Para modelar el comportamiento dinámico del sistema están los de interacción, colaboración, estados y actividades. Los diagramas de componentes y despliegue están enfocados a la implementación del sistema.

Aunque UML es bastante independiente del proceso de desarrollo que se siga, los mismos creadores de UML han propuesto su propia metodología de desarrollo, denominada el Proceso Unificado de Desarrollo.

Según Hernández y compañeros (2001), el Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidos. Además, el proceso unificado utiliza el UML para expresar gráficamente todos los esquemas de un sistema software. Pero, realmente, los aspectos que definen este proceso unificado son tres: es iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura:

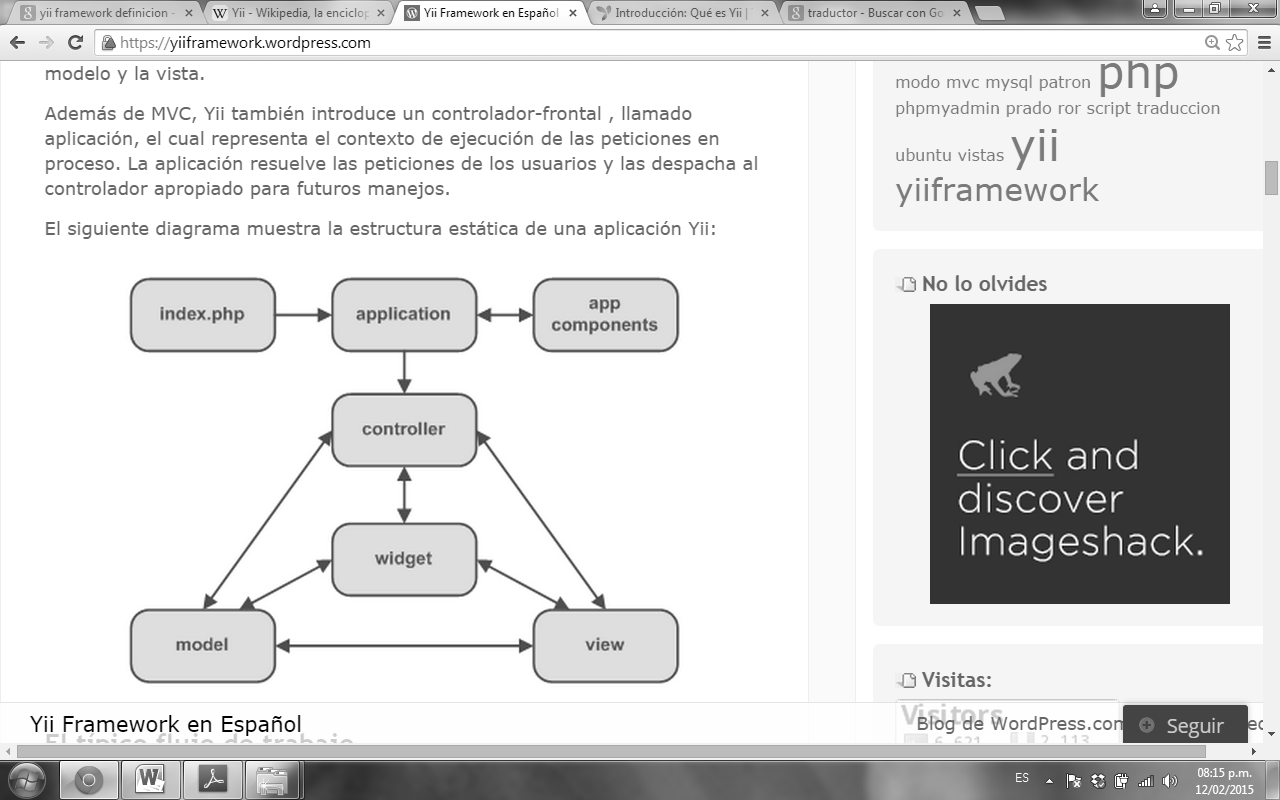
1. ***Dirigido por casos de uso:*** Basándose en los casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que los llevan a cabo. Además, estos modelos se validan para que sean conformes a los casos de uso. Finalmente, los casos de uso también sirven para realizar las pruebas sobre los componentes desarrollados.
2. ***Centrado en la arquitectura:*** En la arquitectura de la construcción, antes de construir un edificio éste se contempla desde varios puntos de vista: estructura, conducciones eléctricas, fontanería, etc. Cada uno de estos aspectos está representado por un gráfico con su notación correspondiente. Siguiendo este ejemplo, el concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema.
3. ***Iterativo e incremental:*** Todo sistema informático complejo supone un gran esfuerzo que puede durar desde varios meses hasta años. Por lo tanto, lo más práctico es dividir un proyecto en varias fases. Actualmente se suele hablar de ciclos de vida en los que se realizan varios recorridos por todas las fases. Cada recorrido por las fases se denomina iteración en el proyecto en la que se realizan varios tipos de trabajo (denominados flujos). Además, cada iteración parte de la anterior incrementado o revisando la funcionalidad implementada. Se suele denominar proceso.

***Yii Framework***

Yii es un framework PHP basado en componentes de alto rendimiento para desarrollar aplicaciones Web de gran escala. El mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo. Además es importante resaltar que Yii es un framework totalmente basado en la programación orientada a objetos (POO).

Como la mayoría de los frameworks PHP, Yii es un framework MVC (modelo-vista-controlador). MVC apunta a separar la lógica de negocio de la interfaz de usuario, así los programadores pueden fácilmente cambiar cada parte sin afectar unas a otras. En MVC, el modelo representa la información(los datos) y las reglas de negocio; las vistas contienen los elementos de interfaz de usuario, como texto, colores, imágenes; y los controladores manejan la comunicación entre el modelo y la vista.

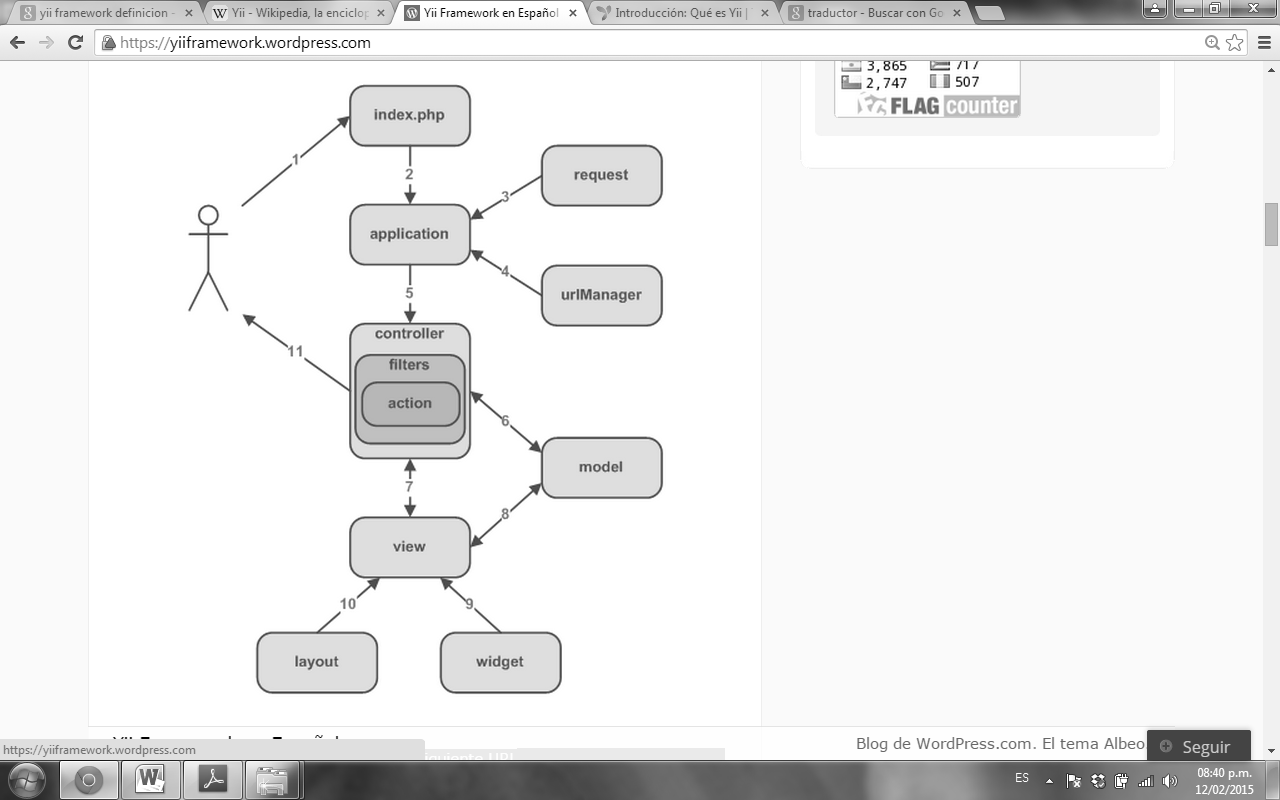
Además de MVC, Yii también introduce un controlador-frontal, llamado aplicación, el cual representa el contexto de ejecución de las peticiones en proceso. La aplicación resuelve las peticiones de los usuarios y las despacha al controlador apropiado para futuros manejos. El siguiente diagrama muestra la estructura estática de una aplicación Yii:



**Figura 7.** Estructura Estática de una Aplicación Yii. Disponible en: <https://yiiframework.wordpress.com/>

El típico flujo de trabajo de una aplicación Yii cuando está manejando la petición de un usuario puede describirse de acuerdo al siguiente orden:

1. Un usuario hace una petición en la siguiente url <http://www.example.com/index.php?r=post/show&id=1> y el servidor web maneja la petición ejecutando el bootstrap script index.php.
2. El bootstrap script crea una instancia de la aplicación y la ejecuta.
3. La aplicación obtiene la información detallada de la petición de los usuarios desde un componente de la aplicación llamado request (petición).
4. La aplicación determina la petición del controlador y actúa con la ayuda de un componente de la aplicación llamado urlManager(manejador de url’s).  
   Para este ejemplo, el controlador es Post que hace referencia a la clase PostController; y la acción es show(mostrar) para el cuál su significado es determinado por el controlador
5. La aplicación crea una instancia de la petición del controlador para más adelante manejar la petición del usuario. El controlador determina que la acción show (mostrar) se refiere a un método llamado actionShow en la clase del controlador.Entonces crea y ejecuta filtros (por ej: controles de acceso) asociados a esta acción. La acción es ejecutada si los filtros lo permiten.
6. La acción lee un modelo Post el cuál si ID es 1 en la base de datos.
7. La acción asocia una vista llamada show(mostrar) con el modelo Post.
8. La vista lee y muestra los atributos del modelo Post.
9. La vista ejecuta algunos artificios.
10. Los resultados de la vista son embebidos en un diseño(layout)
11. La acción completa la prestación de la vista y muestra el resultado al usuario. Para mayor compresión ver figura 8.



**Figura 8.** Típico Flujo de Trabajo de una Aplicación Yii. Disponible en: <https://yiiframework.wordpress.com/>

Algunas características de Yii incluyen:

* Patrón de diseño [Modelo Vista Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador) (MVC).
* [Database Access Objects](http://es.wikipedia.org/w/index.php?title=Data_access_object&action=edit&redlink=1) (DAO), query builder, [Active Record](http://es.wikipedia.org/w/index.php?title=Active_record_pattern&action=edit&redlink=1) y migración de base de datos.
* Integración con [jQuery](http://es.wikipedia.org/wiki/JQuery" \o "JQuery).
* Entradas de [Formulario](http://es.wikipedia.org/wiki/Formulario_(web)) y [validación](http://es.wikipedia.org/w/index.php?title=Validaci%C3%B3n_de_datos&action=edit&redlink=1).
* Widgets de [Ajax](http://es.wikipedia.org/w/index.php?title=Ajax_(programaci%C3%B3n)&action=edit&redlink=1), como autocompletado de campos de texto y demás.
* Soporte de [Autenticación](http://es.wikipedia.org/wiki/Autenticaci%C3%B3n) incorporado. Además soporta [autorización](http://es.wikipedia.org/wiki/Autorizaci%C3%B3n) vía [role-based access control](http://es.wikipedia.org/w/index.php?title=Role-based_access_control&action=edit&redlink=1) (RBAC) jerárquico.
* Personalización de aspectos y temas.
* Generación compleja automática de [WSDL](http://es.wikipedia.org/wiki/Web_Services_Description_Language), especificaciones y administración de peticiones [Web service](http://es.wikipedia.org/wiki/Web_service).
* [Internacionalización y localización](http://es.wikipedia.org/wiki/Internacionalizaci%C3%B3n_y_localizaci%C3%B3n) (I18N and L10N). Soporta traducciones, formato de fecha y hora, formato de números, y localización de la vista.
* Esquema de [caching](http://es.wikipedia.org/wiki/Cache" \o "Cache) por capas. Soporta el cache de datos, cache de páginas, cache por fragmentos y contenido dinámico. El medio de almacenamiento del cache puede ser cambiado.
* El [manejo](http://es.wikipedia.org/wiki/Manejo_de_excepciones) de errores y [logging](http://es.wikipedia.org/w/index.php?title=Data_logger&action=edit&redlink=1" \o "Data logger (aún no redactado)). Los errores son manejados y personalizados, y los log de mensajes pueden ser categorizados, filtrados y movidos a diferentes destinos.
* Las medidas de seguridad incluyen la prevención [cross-site scripting](http://es.wikipedia.org/wiki/Cross-site_scripting" \o "Cross-site scripting) (XSS), prevención [cross-site request forgery](http://es.wikipedia.org/w/index.php?title=Cross-site_request_forgery&action=edit&redlink=1" \o "Cross-site request forgery (aún no redactado)) (CSRF), prevención de la manipulación de [cookies](http://es.wikipedia.org/w/index.php?title=HTTP_cookie&action=edit&redlink=1), etc.
* Herramientas para pruebas unitarias y funcionales basados en [PHPUnit](http://es.wikipedia.org/wiki/PHPUnit" \o "PHPUnit) y [Selenium](http://es.wikipedia.org/w/index.php?title=Selenium_(software)&action=edit&redlink=1" \o "Selenium (software) (aún no redactado)).
* Generación automática de código para el esqueleto de la aplicación, aplicaciones [CRUD](http://es.wikipedia.org/w/index.php?title=Crear,_leer,_actualizar_y_borrar&action=edit&redlink=1), etc.
* Generación de código por componentes de Yii y la herramienta por línea de comandos cumple con los estándares de [XHTML](http://es.wikipedia.org/wiki/XHTML).
* Cuidadosamente diseñado para trabajar bien con código de terceros. Por ejemplo, es posible usar el código de [PHP](http://es.wikipedia.org/wiki/PHP) o [Zend Framework](http://es.wikipedia.org/wiki/Zend_Framework" \o "Zend Framework) en una aplicación Yii.

Yii sobresale frente a frameworks PHP por su eficiencia, su gran cantidad de características y su clara documentación. Yii ha sido diseñado cuidadosamente desde el principio para el desarrollo de aplicaciones Web. No es ni un subproducto de un proyecto ni un conglomerado de trabajo de terceros. Es el resultado de la vasta experiencia de los autores en desarrollo de aplicaciones web y de la investigación y la reflexión de los más populares los frameworks de programación web y aplicaciones.

Ahora bien si se detalla toda la información anterior, se puede observar que yii es un framework muy completo que usa muchas tecnologías consigo como javascript, Ajax, [jQuery](http://es.wikipedia.org/wiki/JQuery) y php; es por ello que para comprender un poco más cada una de esas tecnologías serán descritas continuamente debido a que son fundamentales para el desarrollo de la aplicación que será desarrollada en esta investigación.

***Pre-Procesador De Hipertexto (PHP)***

PHP (Hypertext Preprocessor) es un lenguaje script (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código), para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es de Open Source (código abierto), es el más popular y extendido en la web. (Álvarez, 2001)

PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. También puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas existentes.

PHP es capaz de realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados en un lenguaje distinto al HTML. Esto se debe a que PHP ofrece un extenso conjunto de funciones para la explotación de bases de datos sin complicaciones. Es por esto, que levanta un mayor interés con respecto a los lenguajes pensados para los CGI.

PHP fue desarrollado originalmente por Rasmus Ledford en 1994 como un CGI escrito en Perl que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y consiguió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas. Cuando Rasmus tuvo la necesidad de crear páginas dinámicas que trabajasen con formularios, creó una serie de etiquetas a las que denominó “Form Interpreters”, y lo sacó al público con el nombre de PHP/FI en 1995. Luego salió la versión mejorada, llamada PHP/FI 2.0.

Zeev Suraski y Andi Gutmans programaron el analizador sintáctico incluyendo nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales, como MySQL y Postgre SQL, así como un módulo para Apache. Con estas mejoras surgió PHP3 en 1997. Este analizador define la sintaxis y semántica de la versión PHP3 y la siguiente: PHP4.

PHP3 carecía del uso de sesiones, algo muy común en las páginas web de cierta complejidad. En el año 2000, PHP3 evolucionó a PHP4, que utiliza el motor Zend (desarrollado por Zeev y Andi encargado de interpretar el código fuente de los scripts de PHP), desarrollado para cubrir las necesidades actuales y solucionar algunos inconvenientes de la anterior versión. Algunas mejoras de esta nueva versión son su mayor independencia del servidor web y su rapidez, ya que primero se compila y luego se ejecuta, mientras que antes se ejecutaba a la vez que se interpretaba el código.

La última versión es PHP5, que utiliza el motor Zend-2 y presenta mejoras significativas y un entorno de programación orientado a objetos mucho más completo, que permite que el PHP proporcione un alto rendimiento a las aplicaciones Web empresariales a nivel de las plataformas J2EE y .NET. Otro lenguaje de scripting para la generación dinámica de contenidos en el servidor es ASP. Aunque se parece a PHP en cuanto a potencia y dificultad, su sintaxis llega a diferir notablemente.

Las ventajas más relevantes de este lenguaje de programación son:

* Lenguaje multiplataforma.
* Orientado completamente a la web.
* Capacidad de conexión con la mayoría de motores de bases de datos manejados en la actualidad.
* Permite la utilización de técnicas de Programación Orientada a Objetos.
* Soporte para gran cantidad de bases de datos.

***JavaScript***

JavaScript es un lenguaje de script multiplataforma [cross-platform] orientado a objetos. Es un lenguaje pequeño y ligero; no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores web. Dentro de un entorno anfitrión, JavaScript puede ser conectado a los objetos de su entorno para proveer un control programable sobre éstos.

De acuerdo al tutorial de JavaScript, fue desarrollado originalmente por [Brendan Eich](http://es.wikipedia.org/wiki/Brendan_Eich" \o "Brendan Eich) de [Netscape](http://es.wikipedia.org/wiki/Netscape_Communications_Corporation) con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó compatibilidad con la tecnología Java en su [navegador web Netscape Navigator](http://es.wikipedia.org/wiki/Netscape_Navigator) en la versión 2.002 en diciembre de 1995. La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en lo que eran los nuevos lenguajes de programación web.

Aunque comparte muchas de las características y de las estructuras del lenguaje java, fue desarrollado independientemente, por ello se puede decir que  [Java](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java) y JavaScript no están relacionados y tienen semánticas y propósitos diferentes. El lenguaje JavaScript puede interactuar con el código HTML, permitiendo a los programadores web utilizar contenido dinámico. Por ejemplo, hace fácil responder a los acontecimientos iniciados por usuarios (como introducción de datos en formularios) sin tener que utilizar CGI. JavaScript es un lenguaje basado en código abierto, por lo tanto cualquier persona puede utilizarlo sin comprar una licencia.

El núcleo de JavaScript contiene un conjunto central de objetos, tales como  arreglos, fechas y  objetos matemáticos, además de un conjunto central de elementos del lenguaje tales como los operadores, estructuras de control y sentencias. El núcleo de JavaScript puede ser extendido para una variedad de propósitos complementándolo con objetos adicionales; por ejemplo:

* JavaScript del lado Cliente extiende el núcleo del lenguaje proporcionando objetos para el control del navegador y su Modelo de Objetos del Documento (DOM). Por ejemplo, las extensiones del lado del cliente permiten a una aplicación ubicar elementos en un formulario HTML y responder a los eventos de usuario tales como los clicks del mouse, entradas del formulario y navegación de páginas.
* JavaScript del lado Servidor extiende el núcleo del lenguaje proporcionando objetos relevantes para la ejecución de JavaScript en un servidor. Por ejemplo, las extensiones del lado del servidor permiten que una aplicación se comunique con una base de datos relacional, proporcionar continuidad de la información desde una invocación de la aplicación a otra o efectuar la manipulación de archivos en un servidor.

El uso más común de JavaScript es escribir funciones embebidas o incluidas en páginas HTML y que interactúan con el Modelo de Objetos del Documento de la página. Algunos ejemplos sencillos de este uso son:

* Cargar nuevo contenido para la página o enviar datos al servidor a través de AJAX sin necesidad de recargar la página (por ejemplo, una red social puede permitir al usuario enviar actualizaciones de estado sin salir de la página).
* Animación de los elementos de página, hacerlos desaparecer, cambiar su tamaño, moverlos, etc.
* Contenido interactivo, por ejemplo, juegos y reproducción de audio y vídeo.
* Validación de los valores de entrada de un formulario web para asegurarse de que son aceptables antes de ser enviado al servidor.
* Transmisión de información sobre los hábitos de lectura de los usuarios y las actividades de navegación a varios sitios web. Las páginas Web con frecuencia lo hacen para hacer análisis web, seguimiento de anuncios, la personalización o para otros fines

Debido a que JavaScript es el único lenguaje por el que los más populares navegadores comparten su apoyo, se ha convertido en un lenguaje para la compilación factible de muchos otros frameworks entre los cuales cabe resaltar a Yii puesto que es el framework utilizado para el desarrollo de la aplicación planteada en esta investigación.

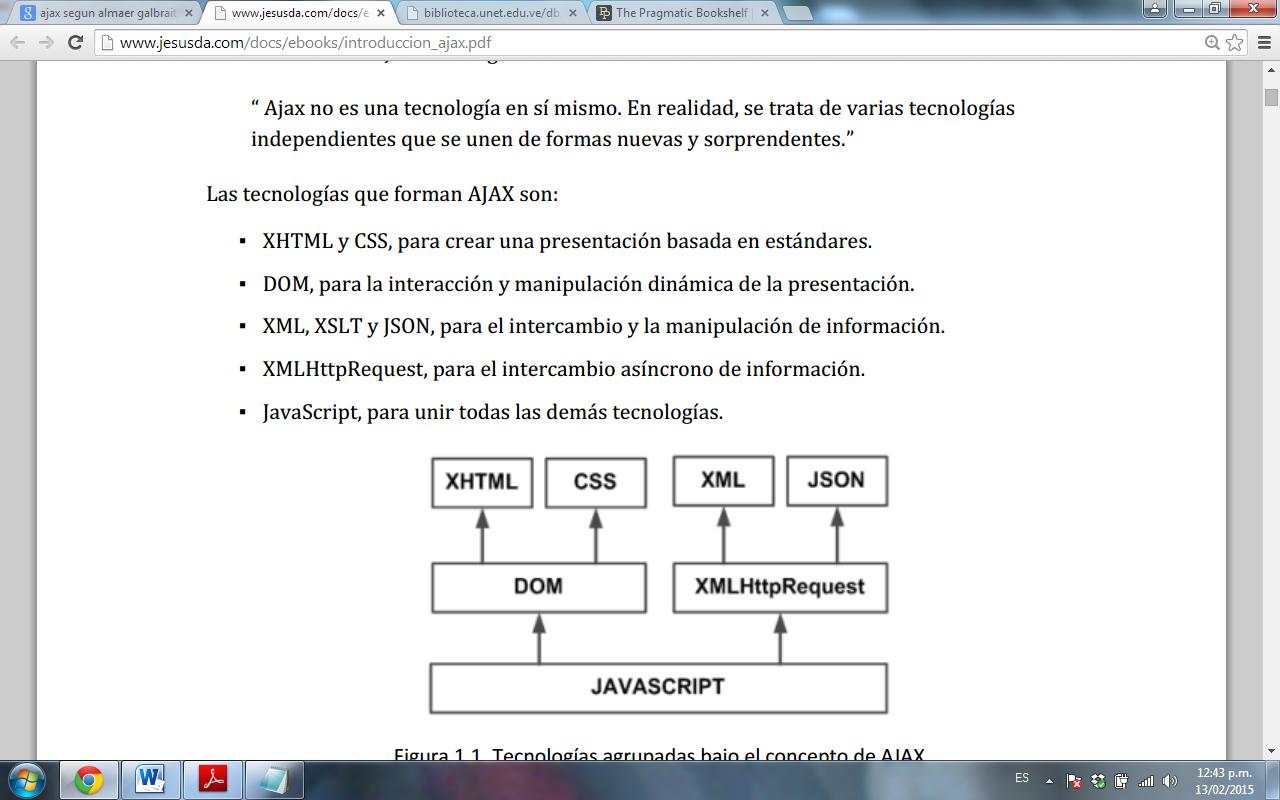
***Ajax***

El término AJAX se presentó por primera vez en el artículo “Ajax: A New Approach to Web Applications” publicado por Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo.

En realidad, el término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML". El artículo define AJAX de la siguiente manera: “Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.”

Las tecnologías que forman AJAX son:

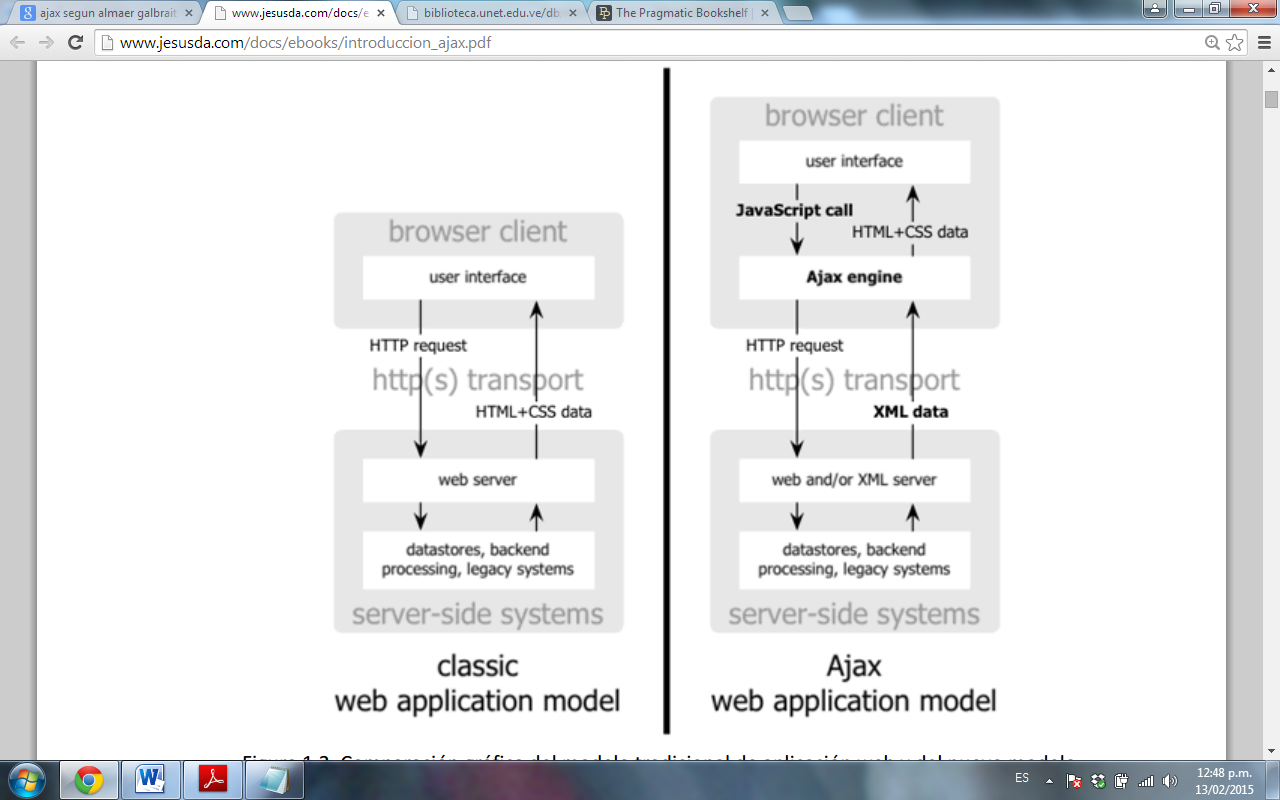
* XHTML y CSS, para crear una presentación basada en estándares.
* DOM, para la interacción y manipulación dinámica de la presentación.
* XML, XSLT y JSON, para el intercambio y la manipulación de información.
* XMLHttpRequest, para el intercambio asíncrono de información.
* JavaScript, para unir todas las demás tecnologías.



**Figura 9.** Tecnologías agrupadas bajo el concepto de AJAX.Garrett.2005

Desarrollar aplicaciones AJAX requiere un conocimiento avanzado de todas y cada una de las tecnologías anteriores. En las aplicaciones web tradicionales, las acciones del usuario en la página (pinchar en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario.

En el siguiente esquema, la imagen de la izquierda muestra el modelo tradicional de las aplicaciones web. La imagen de la derecha muestra el nuevo modelo propuesto por AJAX.

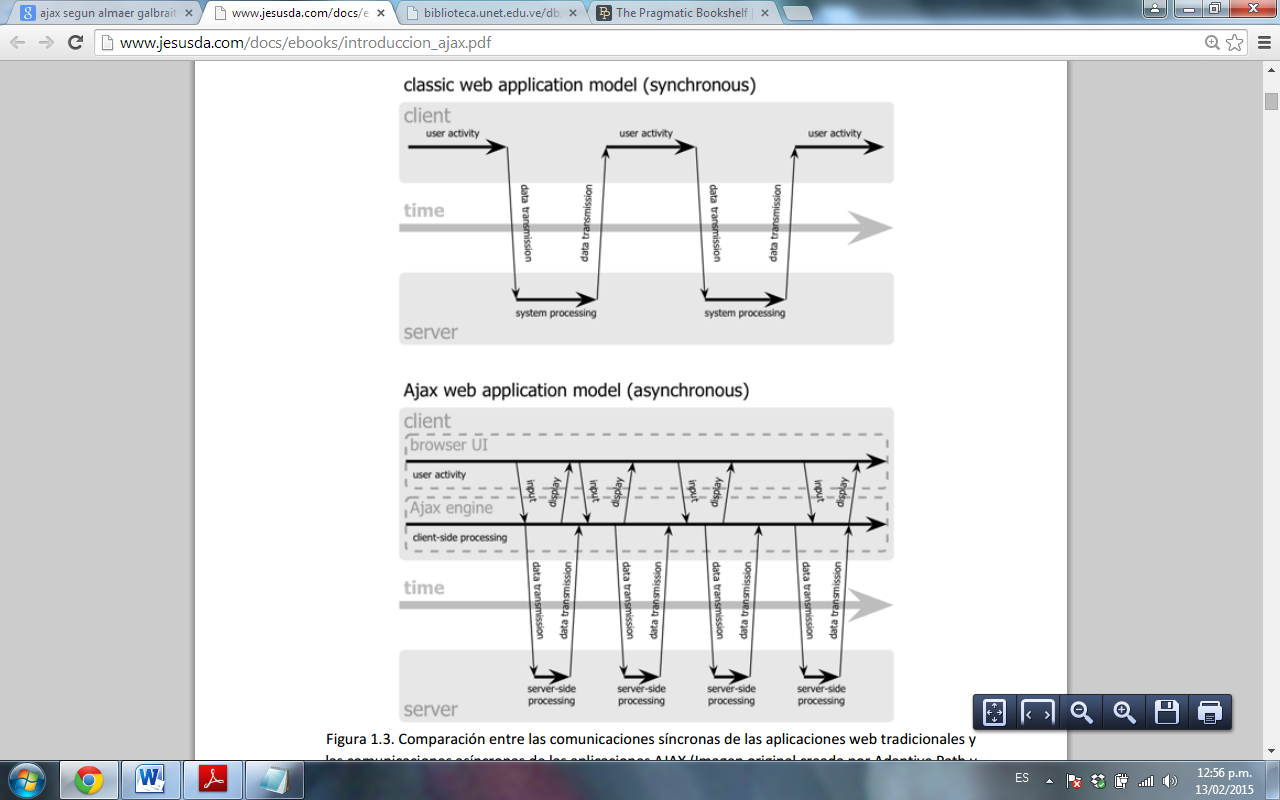
****

**Figura 10.** Comparación gráfica del modelo tradicional de aplicación web y el nuevo modelo propuesto por AJAX. Garrett.2005

Esta técnica tradicional para crear aplicaciones web funciona correctamente, pero no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, su uso se convierte en algo molesto, Mientras AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

El siguiente esquema muestra la diferencia más importante entre una aplicación web tradicional y una aplicación web creada con AJAX. La imagen superior muestra la interacción síncrona propia de las aplicaciones web tradicionales. La imagen inferior muestra la comunicación asíncrona de las aplicaciones creadas con AJAX.



**Figura 11.** Comparación entre las comunicaciones síncronas de las aplicaciones web tradicionales y las comunicaciones asíncronas de las aplicaciones AJAX. Garrett. 2005.

Las peticiones HTTP al servidor se sustituyen por peticiones JavaScript que se realizan al elemento encargado de AJAX. Las peticiones más simples no requieren intervención del servidor, por lo que la respuesta es inmediata. Si la interacción requiere una respuesta del servidor, la petición se realiza de forma asíncrona mediante AJAX. En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor.

Desde su aparición, se han creado cientos de aplicaciones web basadas en AJAX. En la mayoría de casos, AJAX puede sustituir completamente a otras técnicas como Flash. Además, en el caso de las aplicaciones web más avanzadas, pueden llegar a sustituir a las aplicaciones de escritorio.

***JQuery***

jQuery es una [biblioteca](http://es.wikipedia.org/wiki/Biblioteca_(inform%C3%A1tica)) de [JavaScript](http://es.wikipedia.org/wiki/JavaScript), creada inicialmente por [John Resig](http://es.wikipedia.org/wiki/John_Resig), es uno de los complementos más esenciales para el desarrollo web, dado que facilita el proceso de aplicaciones enriquecidas del lado del cliente en JavaScript, compatibles con todos los navegadores, además permite simplificar la manera de interactuar con los documentos  [HTML](http://es.wikipedia.org/wiki/HTML), manipular el árbol [DOM](http://es.wikipedia.org/wiki/Document_Object_Model), manejar eventos, desarrollar animaciones y agregar interacción con la técnica [AJAX](http://es.wikipedia.org/wiki/AJAX) a páginas web. Fue presentada el 14 de enero de 2006 en el [BarCamp](http://es.wikipedia.org/wiki/BarCamp" \o "BarCamp)NYC.

Algunas de las características de esta herramienta son:

* Selección de elementos [DOM](http://es.wikipedia.org/wiki/Document_Object_Model).
* Interactividad y modificaciones del árbol DOM, incluyendo soporte para [CSS 1-3](http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada) y un [plugin](http://es.wikipedia.org/wiki/Complemento_(inform%C3%A1tica)" \o "Complemento (informática)) básico de [XPath](http://es.wikipedia.org/wiki/XPath" \o "XPath).
* Eventos.
* Manipulación de la hoja de estilos CSS.
* Efectos y animaciones.
* Animaciones personalizadas.
* [AJAX](http://es.wikipedia.org/wiki/AJAX).
* Soporta [extensiones](http://es.wikipedia.org/wiki/Complemento_(inform%C3%A1tica)).
* Utilidades varias como obtener información del [navegador](http://es.wikipedia.org/wiki/Navegador_web), operar con [objetos](http://es.wikipedia.org/wiki/Objeto_(programaci%C3%B3n)) y [vectores](http://es.wikipedia.org/wiki/Vector_(inform%C3%A1tica)), funciones para rutinas comunes, etc.
* Compatible con los navegadores [Mozilla Firefox](http://es.wikipedia.org/wiki/Mozilla_Firefox) 2.0+, [Internet Explorer](http://es.wikipedia.org/wiki/Internet_Explorer) 6+, [Safari](http://es.wikipedia.org/wiki/Safari_(navegador)) 3+, [Opera](http://es.wikipedia.org/wiki/Opera_(navegador)) 10.6+ y [Google Chrome](http://es.wikipedia.org/wiki/Google_Chrome) 8+.

Conviene aclarar que jQuery no es un lenguaje, sino una serie de funciones y métodos de JavaScript. Por tanto, JavaScript es el lenguaje y jQuery es una librería que se puede usar opcionalmente para facilitar el trabajo cuando se programa en JavaScript. A veces suele referirse a jQuery como un framework o incluso como un API de funciones, útiles en la mayoría de proyectos web.

JQuery es [software libre y de código abierto](http://es.wikipedia.org/wiki/Software_libre_y_de_c%C3%B3digo_abierto), posee un doble licenciamiento bajo la [Licencia MIT](http://es.wikipedia.org/wiki/MIT_License) y la [Licencia Pública General de GNU](http://es.wikipedia.org/wiki/GNU_General_Public_License) v2, permitiendo su uso en proyectos [libres](http://es.wikipedia.org/wiki/Software_libre) y [privados](http://es.wikipedia.org/wiki/Software_propietario). Con la llegada de jQuery la ventaja principal es que no existe inquietud sobre si el navegador del usuario es explorer, chrome, firefox, etc. sino que la propia librería hará el trabajo y ejecutará el código que sea compatible con el software del cliente que está accediendo al sitio web.

Como es normal, todo framework tiene sus ventajas e inconvenientes, pero jQuery es un producto con una aceptación por parte de los programadores muy buena y un grado de penetración en el mercado muy amplio, lo que hace suponer que es una de las mejores opciones. Además, es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework. Otra cosa muy interesante es la dilatada comunidad de creadores de plugins o componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos, etc.

***MySQL***

Cyril Thibaud, define MySQL como un sistema de administración de bases de datos relacionales (SGBDR) rápido, robusto y fácil de usar. Se adapta bien a la administración de datos en un entorno de red, especialmente en arquitecturas cliente/servidor. Se proporciona con muchas herramientas y es compatible con muchos lenguajes de programación. Es el más célebre Sistema de Gestión de Base de Datos (SGBDR) del mundo Open Source, en particular gracias a su compatibilidad con el servidor de páginas Web Apache y el lenguaje de páginas Web dinámicas Hypertext Pre-Processor (PHP).

MySQL es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet.

SQL o Structured Query Lenguaje (Lenguaje de Consulta Estructurado) fue comercializado por primera vez en 1981 por IBM, el cual fue presentado a ANSI y desde ese entonces ha sido considerado como un estándar para las bases de datos relacionales. Desde 1986, el estándar SQL ha aparecido en diferentes versiones como por ejemplo: SQL:92, SQL:99, SQL:2003. MySQL es una idea originaria de la empresa opensource MySQL AB establecida inicialmente en Suecia en 1995 y cuyos fundadores son David Axmark, Allan Larsson, y Michael "Monty" Widenius. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad.

Michael Widenius en la década de los 90 trató de usar mSQL para conectar las tablas usando rutinas de bajo nivel ISAM, sin embargo, mSQL no era rápido y flexible para sus necesidades. Esto lo conllevó a crear una API SQL denominada MySQL para bases de datos muy similar a la de mSQL pero más portable.

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (vía dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, REALbasic (Mac), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

MySQL es muy utilizado en aplicaciones Web como MediaWiki o Drupal, 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. MySQL 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.

***Calidad de software***

La calidad del software es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente. (Pressman, 2002)

La ausencia de defectos, la aptitud para el uso, la seguridad, la confiabilidad y la reunión de especificaciones son elementos que están involucrados en el concepto de calidad del software. Sin embargo, la calidad del software debe ser construida desde el comienzo, no es algo que puede ser añadido después. (Humphrey, 1997)

***Modelo sistémico de calidad del software***

Callaos y Callaos (1996) proponen un concepto de calidad sistémica del software en el cual están involucradas tanto las características internas como el contexto organizacional, lo que genera un enfoque sistémico del concepto de calidad del software. Afirma también que la calidad global no es la suma de las calidades parciales, sino el compromiso entre todo el conjunto de calidades que conlleve a un óptimo global con cierto sacrificio de los óptimos parciales.

Los componentes que son tomados en cuenta en el modelo de calidad del producto son los siguientes:

* Los aspectos internos y contextuales del producto como calidad parcial del modelo de Calidad Sistémica de Callaos.
* Las características de calidad del modelo de Dromey (1996) y el estándar internacional ISO/IEC 9126 (JTC 1/SC 7, 1991): Eficiencia, Fiabilidad, Funcionalidad, Mantenibilidad, Portabilidad y Usabilidad.
* La relación utilizada en el modelo de McCall (Gillies, 1977) entre los atributos y calidad de las métricas.

***Formulación del Modelo Sistémico de Calidad (MOSCA)***

Fundamentalmente, tal como indica Humphrey, la calidad del proceso garantiza la calidad del producto y consecuentemente no se pueden desligar estas dos calidades; tener modelos separados capaces de medir individualmente la calidad de un producto o de un proceso de software no garantiza la relación sistémica que debe estar presente entre ellos. Esto se debe a que la naturaleza de los sistemas no puede ser divida en partes, sino que debe existir una interdependencia y colaboración entre las partes (proceso y producto) para que el mismo sea visto como un todo.

MOSCA está constituida por cuatro (4) niveles, los cuales son explicados a continuación:

***Nivel 0:*** Conformado por tres categorías. Mendoza y otros, indican que este nivel debe contener la categoría funcionalidad, y las otras dos, deben ser seleccionadas entre las cinco restantes propuestas por MOSCA. Expertos en simulación consideraron que las categorías usabilidad y eficiencia son las de mayor interés y que debían estar presentes en el software a ser evaluado. Las categorías se definen a continuación:

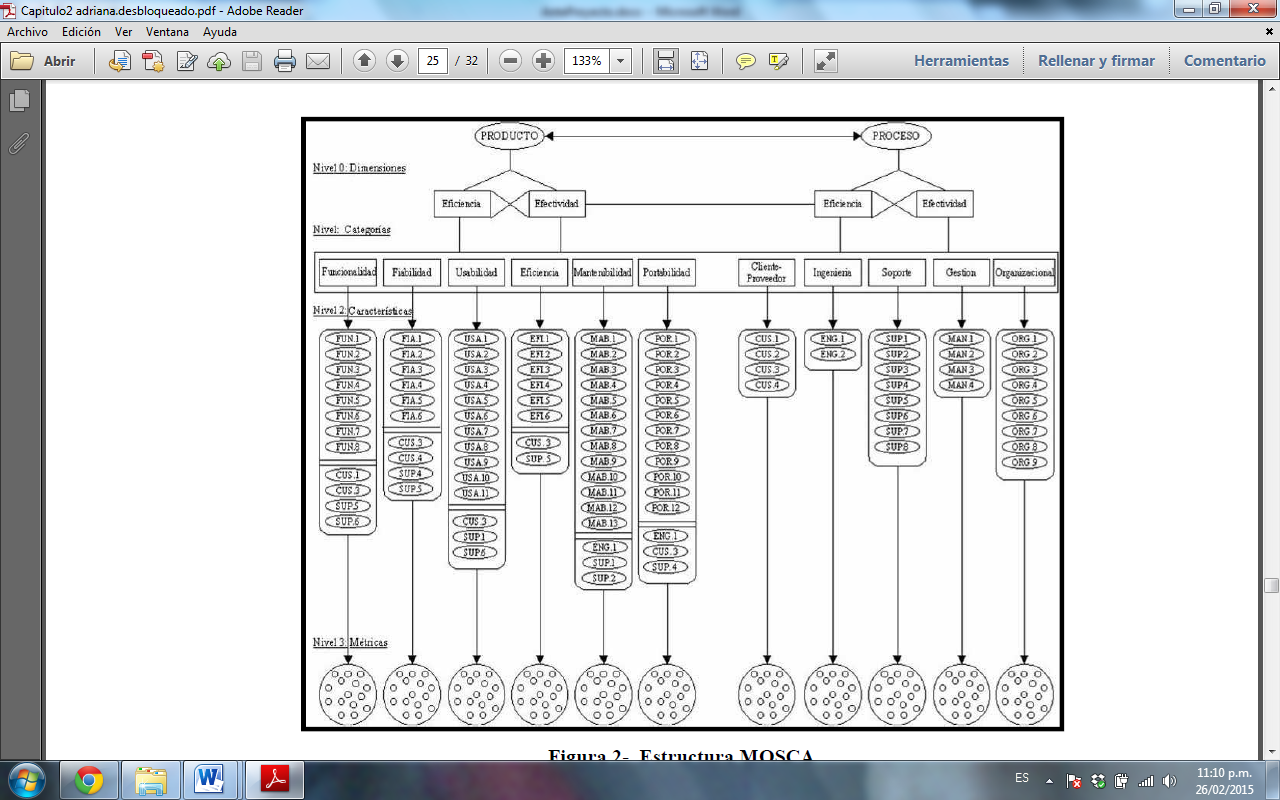
* ***Funcionalidad:*** Capacidad del software para proveer funciones que cumplan con necesidades específicas o implícitas, cuando es utilizada bajo ciertas condiciones.
* ***Usabilidad***: Capacidad del producto para ser atractivo, entendido, aprendido y utilizado por el usuario bajo condiciones específicas.
* ***Eficiencia***: Capacidad del producto para proveer un rendimiento apropiado, relativo a la cantidad de recursos utilizados, bajo condiciones específicas.

***Nivel 1:*** **Categorías.** Se contemplan once categorías: seis pertenecientes al producto y las otras cinco al proceso de desarrollo. Esta división no implica un desligamiento entre ellas, simplemente se realiza para identificar a que sector o sub-modelo pertenecen.

***Nivel 2:* Características**. Cada categoría tiene asociado un conjunto de características, las cuales definen las áreas claves a satisfacer para lograr, asegurar y controlar la calidad tanto en el producto como en el proceso. Entre las características asociadas a cada categoría del producto, se proponen una serie de características del proceso. Esto se debe, a que algunas características de la calidad del proceso, impactan directamente en las categorías del producto al igual que ciertas características de la calidad del producto definen categorías del proceso. Esto ayuda a precisar que si una vez medidas las características asociadas a una categoría en particular del producto, arroja resultados no deseados, se pueden analizar las características de la calidad del proceso asociadas a esa categoría del producto para encontrar las posibles causas.

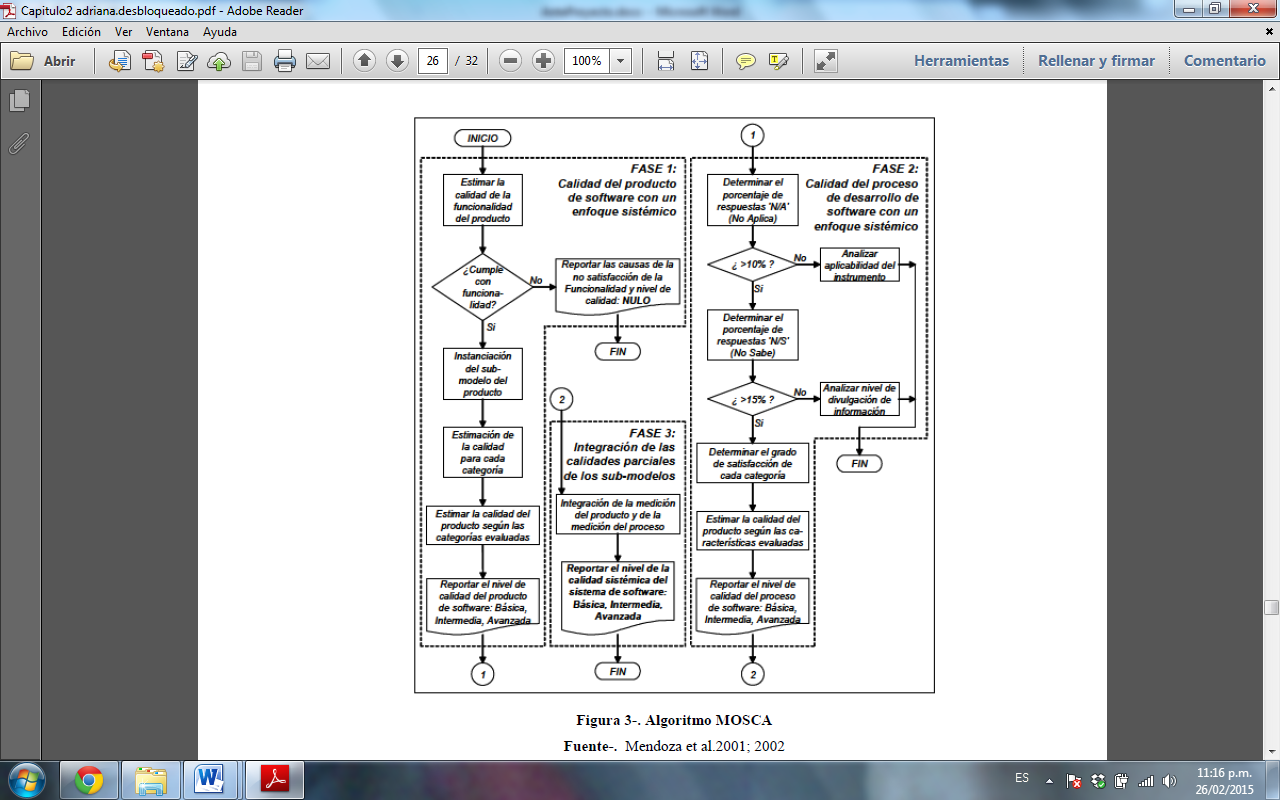
***Nivel 3:*** **Métricas.** A cada sub-característica se le asocian métricas que están relacionadas con las cualidades o atributos del software que se desean evaluar. MOSCA está conformada por 133 métricas que estiman la calidad de las 40 subcaracterísticas.

Cada uno de estos niveles es fundamental en el momento de aplicar la evaluación de calidad de un software, es por ello que a continuación se muestra gráficamente la estructura MOSCA para percibir de forma más precisa y detallada su funcionamiento.

**Figura 12.** Estructura MOSCA.Mendoza.2001

***Algoritmo de Aplicación del Modelo MOSCA***

Para medir la calidad sistémica a través de la aplicación del modelo MOSCA, se deberá medir primero la calidad del producto de software y luego la calidad del proceso de desarrollo del mismo. Gráficamente los pasos a seguir son los que se observan en la Figura 14:



**Figura 13.** Algoritmo MOSCA.Mendoza.2001

***Aspectos Legales***

El marco legal de esta investigación la comprende Norma COVENIN 3049-93 Mantenimiento definiciones, Reglamento de la Ley Orgánica de Prevención, Condiciones y Medio Ambiente de Trabajo.

**Definición de Términos Básicos**

***Aplicación Web:*** Se denomina aplicación web a aquellas [herramientas](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica) que los usuarios pueden utilizar accediendo a un [servidor web](http://es.wikipedia.org/wiki/Servidor_web) a través de [Internet](http://es.wikipedia.org/wiki/Internet) o de una [intranet](http://es.wikipedia.org/wiki/Intranet) mediante un [navegador](http://es.wikipedia.org/wiki/Navegador_web). En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

***Calidad de software:*** Conjunto de características propias de un determinado software que cumple con estándares preestablecidos.

***Evaluación de producto de software:*** Conjunto de actividades que se llevan a cabo durante y después del proceso de construcción de un software con el fin de certificar su calidad.

***Flota:*** Conjunto de vehículos destinados a prestar un servicio bien sea al transportar mercancía o personas por diferentes medios y que dependen económicamente de la misma empresa.

***Framework:*** Un framework es una colección organizada de clases que constituyen un diseño reutilizable para un dominio específico de software. Contiene un conjunto de librerías, componentes de software y directrices arquitectónicas que ofrece al desarrollador un kit de herramientas completo para construir una aplicación de principio a fin, siempre teniendo en cuenta que es necesario adaptarlo a cada aplicación en particular.

***Gestión de flota:*** Gestión del conjunto de vehículos de una organización manejando aspectos como reducción de costos, consumo de combustible, mantenimiento y seguimiento de vehículos seguridad eficiencia y productividad.

***Gestión de mantenimiento:*** Es la efectiva y eficiente utilización de los recursos materiales, económicos, humanos y de tiempo para alcanzar los objetivos de mantenimiento.

***Indicadores de Gestión:*** Son medios, instrumentos o mecanismos para evaluar hasta qué punto o en qué medida se están logrando los [objetivos estratégicos](http://www.gestiopolis.com/canales5/emp/pymecommx/45.htm) de una organización o empresa.

***Mantenimiento:*** Es el conjunto de acciones que permite conservar o restablecer un sistema productivo a un estado específico, para que pueda cumplir un servicio determinado.

***Mantenimiento correctivo:*** Conjunto de actividades que se llevan a cabo luego de que ha ocurrido una falla en un vehículo, su objetivo es corregir la falla logrando integrar la unidad de nuevo al proceso normal de la empresa.

***Mantenimiento preventivo:*** Conjunto de actividades que se llevan a cabo en un vehículo con el propósito de que opere a su máxima eficiencia evitando que se produzcan paradas forzadas o imprevistas, su objetivo es prevenir fallas antes de que éstas se produzcan interfiriendo con el proceso productivo normal de la empresa.

***Modelo de calidad:*** Conjunto de prácticas vinculadas a los procesos de gestión y el desarrollo de proyectos, supone una planificación para alcanzar un impacto estratégico, cumpliendo con los objetivos fijados en lo referente a la calidad del producto o servicio.

***Mysql:*** es un sistema de gestión de base de datos multiusuario, multiplataforma y de código abierto. Por un lado se ofrece bajo la [GNU GPL](http://es.wikipedia.org/wiki/Licencia_p%C3%BAblica_general_de_GNU) para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos [privativos](http://es.wikipedia.org/wiki/Software_propietario) deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en [ANSI C](http://es.wikipedia.org/wiki/ANSI_C).

***Operatividad:*** Capacidad que tiene un objeto para realizar una función.

***Php:*** Es un lenguaje de programación [de uso general](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_de_prop%C3%B3sito_general) de [código del lado del servidor](http://es.wikipedia.org/wiki/Script_del_lado_del_servidor) originalmente diseñado para el [desarrollo web](http://es.wikipedia.org/wiki/Desarrollo_web) de [contenido dinámico](http://es.wikipedia.org/wiki/Contenido_din%C3%A1mico). Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento [HTML](http://es.wikipedia.org/wiki/HTML) en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de [línea de comandos](http://es.wikipedia.org/wiki/L%C3%ADnea_de_comandos) que puede ser usada en [aplicaciones gráficas](http://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario) independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

***Plataforma:*** Determinado software o hardware con el cual una aplicación es compatible y permite ejecutarla.

***Proyecto:*** Conjunto de las actividades que desarrolla una [persona](http://definicion.de/persona) o una entidad para alcanzar un determinado objetivo. Estas actividades se encuentran interrelacionadas y

se desarrollan de manera coordinada.

***Servidor:*** Es un ordenador remoto que provee los datos solicitados por parte de los navegadores de otras computadoras.

***Sistem*a:** Un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben como entrada datos y proveen como salida información.

***Software:*** Es un conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora. El software es un ingrediente indispensable para el funcionamiento del computador. Un computador en sí, es sólo un conglomerado de componentes electrónicos; el software le da vida al computador, haciendo que sus componentes funcionen de forma ordenada.

***UML:*** Es un popular lenguaje de modelado de sistemas de software. Se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software. Posee la riqueza suficiente como para crear un modelo del sistema, pudiendo modelar los procesos de negocios, funciones, esquemas de bases de datos, expresiones de lenguajes de programación utilizando diferentes tipos de diagramas.

***Vehículo:*** Todo aquel medio de transporte a motor para transportar pasajeros y/o carga.

**CAPÍTULO III**

**MARCO METODOLÓGICO**

**Nivel de Investigación**

Para Arias (2006), el nivel de investigación se refiere “al grado de profundidad con que se aborda un fenómeno u objeto de estudio” (p.23). En el presente caso, el estudio se enmarcó dentro de una investigación de carácter proyectiva y aplicativa.

Es una investigación proyectiva ya que se propone una solución a una situación determinada partiendo de un proceso sistemático de búsqueda e investigación, para describir y definir los requerimientos y necesidades de la empresa. Este tipo de investigación, consiste en la elaboración de una propuesta, un plan, un programa o un modelo, como solución a un problema o necesidad de tipo práctico, ya sea de un grupo social, o de una institución, o de una región geográfica, en un área particular del conocimiento, a partir de un diagnóstico preciso de las necesidades del momento, los procesos explicativos o generadores involucrados y de las tendencias futuras, es decir, con base en los resultados de un proceso investigativo.(Hurtado,2008).

Y a su vez, es una investigación aplicativa puesto que se plantea como un objetivo, integrar el sistema a los sistemas actuales de la Universidad Nacional Experimental del Táchira. Según Padrón (1998):

Las investigaciones aplicativas tienen como objetivo central proveer tecnologías o esquemas de acción derivados de los conocimientos teóricos construidos dentro de la secuencia de la Línea. Estas investigaciones carecen, propiamente hablando, de preguntas. Más bien tienden a establecer una relación productiva, ingeniosa y creativa, entre las posibilidades de un modelo teórico, por un lado, y las dificultades o necesidades que se confrontan en el terreno de la práctica, por otro lado. (s.p).

De acuerdo con el autor, como alternativa de solución al problema identificado, se plantea la construcción de un software para la gestión de flota de vehículos que permita agilizar las tareas de una empresa dedicada a estas actividades. Como complemento a lo dicho anteriormente, expresa:

Las operaciones estandarizadas de las investigaciones aplicativas son las siguientes: descripción de la situación deficitaria, exposición del modelo teórico que resulta aplicable a esa situación, construcción del prototipo de control situacional (definición de la propuesta de solución) mediante derivación del modelo teórico, prueba del prototipo, determinación de las opciones de producción e implementación del prototipo. La relación fundamental que estas investigaciones mantienen con los enfoques epistemológicos está en que cada investigador aplicativo elige aquellas teorías que fueron construidas dentro de su propio enfoque. (s.p).

**Diseño de Investigación**

Se refiere a las estrategias que aplica el investigador para obtener la información necesaria y de esta manera llevar a cabo el desarrollo de la investigación. De acuerdo con Balestrini (2001):

El diseño de investigación se define como el plan global que integra de un modo coherente y adecuadamente correcto técnica de recogida de datos a utilizar, análisis previstos y objetivos. El diseño intenta dar de una manera clara y no ambigua respuesta a las preguntas de investigación. (pág. 33).

En esta investigación se aplica un diseño de tipo documental ya que la información que se amerita para el desarrollo de las actividades involucradas en la construcción del sistema, fue recopilada de fuentes formales (libros digitales). Además fueron de gran utilidad aquellos trabajos previos enfocados en el diseño de aplicaciones similares relacionados con los campos de gestión .Al mismo tiempo, se aplicará un diseño de campo, para realizar la identificación y clasificación y recolección de los datos de entrada que van a ser utilizados para la construcción del software.

**Técnicas e Instrumentos para la Recolección de Datos**

Hurtado (2000) señala que “las técnicas de recolección de datos comprenden procedimientos y actividades que le permiten al investigador obtener la información necesaria para dar respuesta a su pregunta de investigación”. (pág.427).Para esta investigación se hará uso de la revisión documental, la observación directa y la entrevista no estructurada.

Se usará la revisión documental para comprender y analizar ciertos procesos que deben realizarse en la investigación. Según Montaña (2008) “La revisión documental es el proceso mediante el cual el investigador recopila, revisa, analiza, selecciona y extrae información de diversas fuentes, acerca de un tema particular, con el propósito de llegar al conocimiento y compresión del mismo”. (s.p).

La revisión documental se sitúa en el análisis de las fuentes documentales, que permitieron abordar y desarrollar los requisitos del marco teórico de la investigación, la observación documental, mediante la lectura general de texto, se comenzó con la búsqueda y observación de hechos presentes en los materiales consultados de interés en la investigación, seguida de lecturas más detenidas y rigurosas de los textos, para extraer los puntos esenciales, lógicos y propuestos para el estudio que se realiza.

Esto se realiza de forma resumida y sintetizada, de ideas básicas que contienen las investigaciones consultadas, cabe destacar que también los resultados de otras investigaciones que se han realizado con relación al tema y los antecedentes del mismo, aportan a la construcción del contenido teórico de la investigación.

Se utilizará la observación directa, según Arias (1999), indica que la observación directa consiste “en visualizar o captar mediante la vista, en forma sistemática, cualquier hecho, fenómeno o situación que se produzca en la naturaleza o en la sociedad, en función de unos objetivos de investigación preestablecidos”. (pág.67). También señala que:

la entrevista no estructurada, más que un simple interrogatorio es una técnica basada en un diálogo o conversación “cara a cara”, entre el entrevistador y el entrevistado acerca de un tema previamente determinado, de tal manera que el entrevistador pueda obtener la información requerida. (p. 73)

**Metodología para el Desarrollo de Software**

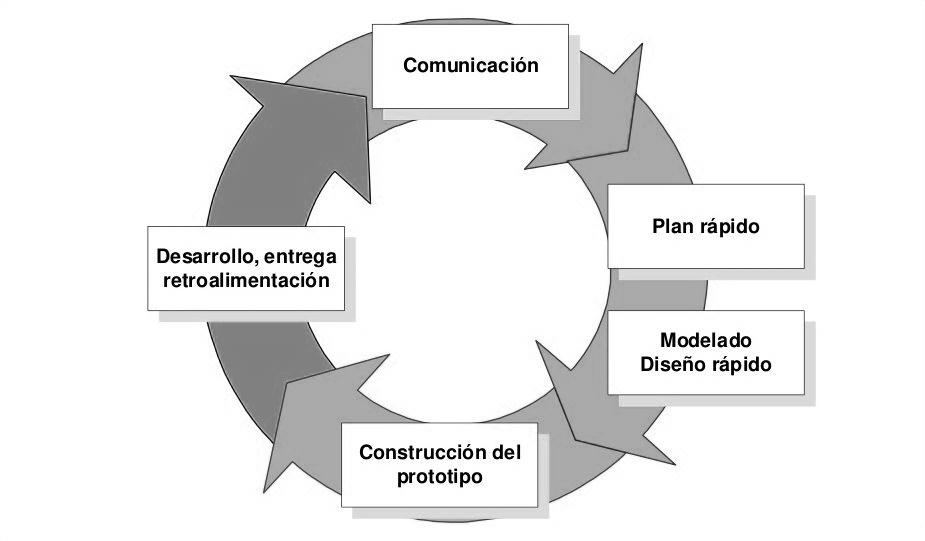
La metodología que se usará durante el desarrollo del sistema será Prototipos Evolutivos también conocida como desarrollo con prototipación o modelo de desarrollo evolutivo, la cual es definida como:

un prototipo en papel o un modelo basado en PC que describa la interacción hombre-máquina, de forma que facilite al usuario la comprensión de cómo se producirá tal interacción; un prototipo que implemente algunos subconjuntos de la función requerida del programa deseado, o un programa existente que ejecute parte o toda la función deseada. (Pressman, 2002)

El modelo de desarrollo evolutivo construye una serie de versiones sucesivas de un producto. Sin embargo, mientras que la aproximación incremental presupone que el conjunto completo de requerimientos es conocido al comenzar, el modelo evolutivo asume que los requerimientos no son completamente conocidos al inicio del proyecto.

Es por ello, que los requerimientos son cuidadosamente examinados, y sólo esos que son bien comprendidos son seleccionados para el primer incremento. A partir de allí se construye una implementación parcial del sistema que recibe sólo estos requerimientos.

Una vez construido el prototipo parcial del sistema, el usuario lo observa, y provee retroalimentación a los desarrolladores. Basada en esta retroalimentación, la especificación de requerimientos es actualizada, y una segunda versión del producto es desarrollada y desplegada. En la figura 14 puede observase el proceso como un ciclo que se repite indefinidamente hasta obtener el producto deseado.



**Figura 14.** Diagrama de Creación de Prototipos.  
Fuente: Pressman Roger. Ingeniería del Software. Un enfoque práctico (p. 128)

Este modelo se utiliza mayoritariamente en desarrollos de productos con innovaciones importantes, o en el uso de tecnologías nuevas o poco probadas, en las que la incertidumbre sobre los resultados a obtener, o la ignorancia sobre el comportamiento, impiden iniciar un proyecto secuencial. Su ventaja principal se basa en que es el único apto para desarrollos en los que no se conoce a priori sus especificaciones o la tecnología a utilizar.

Al igual que cualquier otra metodología, el modelo de Prototipos Evolutivos puede tener ventajas como desventajas, pero para el caso de esta aplicación, y, tomando en cuenta que dicho software será evaluado en cuanto a mejora de calidad, simplicidad de diseño y mejora de productividad, se mencionan algunas de las prerrogativas por las cuales se decide hacer uso de dicha metodología:

1. ***Mayor calidad:*** Al construir el prototipo se asegura que el software sea de mejor calidad, además de que su interfaz sea de agrado para el usuario. Un prototipo podrá ser construido solo si con el software es posible experimentar.
2. ***Diseño enfocado en las necesidades:*** Este modelo es básicamente prueba y error ya que si al usuario no le gusta una parte del prototipo significa que la prueba fallo por lo cual se debe corregir el error que se tenga hasta que el usuario quede satisfecho. Como resultado, habrá menos trabajo después.
3. ***El diseño se va adaptando al entendimiento del problema*:** A medida que se realizan iteraciones de retroalimentación o refinamiento de los prototipos, el entendimiento del problema se incrementa, de esta forma, se reducen los malos entendidos de la funcionalidad al final del desarrollo, resultando en menos trabajo.
4. ***Mayor productividad:*** Como tanto los desarrolladores como el cliente trabajan juntos durante el desarrollo de los prototipos es más probable que las especificaciones de los mismos se acerquen más a la realidad. Los conocimientos que se adquieren durante el desarrollo del prototipo pueden reducir el costo del desarrollo del software más adelante.

**Fases de Desarrollo del Modelo de Prototipos Evolutivos**

Según (Pressman, 2002), existen 5 fases o etapas las cuales se describen a continuación:

***Identificación de requerimientos:*** La determinación de los requerimientos de una aplicación es tan importante para el método de desarrollo de prototipos como lo es para el ciclo de desarrollo de sistemas o análisis estructurado. Por consiguiente, antes de crear un prototipo, los analistas y el usuario deben de trabajar juntos para identificar los requerimientos conocidos que tienen que satisfacer.

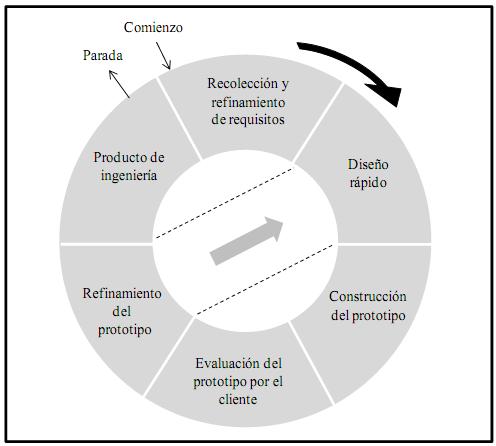
***Desarrollo de un modelo de trabajo:*** Es fácil comenzar el proceso de construcción del prototipo con el desarrollo de un plan general que permita a los usuarios conocer lo que se espera de ellos y del proceso de desarrollo. Un cronograma para el inicio y el fin de la primera interacción es de gran ayuda. En el desarrollo del prototipo se preparan los siguientes componentes:

* El lenguaje para el dialogo o conversación entre el usuario y el sistema.
* Pantallas y formatos para la entrada de datos.
* Módulos esenciales de procesamiento. Salida del sistema.

***Construcción del prototipo:*** Es responsabilidad del usuario trabajar con el prototipo y evaluar sus características y operación. La experiencia del sistema bajo condiciones reales permite obtener la familiaridad indispensable para determinar los cambios o mejoras que sean necesarios, así como las características inadecuadas.

***Revisión del prototipo:*** Durante la evaluación los analistas de sistemas desean capturar información sobre los que les gusta y lo que les desagrada a los usuarios. Los cambios al prototipo son planificados con los usuarios antes de llevarlos a cabo, sin embargo es el analista el responsable de tales modificaciones.

**Refinamiento del prototipo:** El proceso antes descrito se repite varias veces, el proceso finaliza cuando los usuarios y analistas están de acuerdo en que el sistema ha evolucionado lo suficiente como para incluir todas las características necesarias. En la figura 15 se puede observar las fases de desarrollo correspondientes al Modelo de Prototipos Evolutivos.



**Figura 15.** Fases para la Construcción de Prototipos.  
Fuente: Pressman Roger. Ingeniería del Software. Un enfoque práctico. (p. 129)

**REFERENCIAS**

Álvarez M. (2001). *Introducción, Definición y Evolución de PHP.* [Documento en línea]. Disponible: http://php.ciberaula.com/articulo/introduccion\_php/ [Consulta 2015, Febrero 12].

Arias, F. (1999). *El proyecto de investigación guía para su elaboración.* Caracas:

Episteme.

Arias, F. (2006). *El proyecto de investigación. Introducción A La Metodología Científica.* Caracas: Episteme.

Balestrini, M. (2001). *Procedimientos Técnicos de la Investigación Documental.* Caracas: PANAPO.

Bautista, G. (2005). *Sistema de Gestión de Flota Vehicular* [Documento en línea]. Disponible: http://cdn.ly.tl/publications/text-based-captcha-strengths-and-weaknesses.pdf/ [Consulta 2013, Diciembre 10].

BECERRA, F. (2001) *Estudio de los elementos de la Gestión de Mantenimiento*

*para alcanzar la Excelencia* [Documento en línea]. Disponible en Web:

[www.mantenimientomundial.com/sites/mmnew/bib/notas/excelencia.pdf](http://www.mantenimientomundial.com/sites/mmnew/bib/notas/excelencia.pdf)

Beltrán, J. (2000) Indicadores de gestión. (2da. Edición) Bogota (Colombia). 3R Editores.

Busto, D. (2012). *Diseño de un sistema de información para el control de la flota de vehículos de la gerencia de transportes de C.V.G Ferrominera Orinoco* [Documento en línea]. Disponible: http://www.cidar.uneg.edu.ve/DB/bcuneg/EDOCS/TESIS/TESIS\_PREGRADO/INFORMES%20DE%20PASANTIAS/IP100652012CDBustosDiana.pdf [Consulta 2013, Diciembre 10].

Callaos, N., Callaos, B. (1996) *Designing with Systemic Total Quality*, in International Conference on Information Systems, Orlando, Florida, pp. 548-560.

Carballo, Y. (2007). *Programación Orientada a Objetos* [Documento en línea]. Disponible:http://webdelprofesor.ula.ve/ingenieria/hyelitza/materias/programacion2/oxo/ProfaYusneyi\_Tema8\_POOClasesyObjetos.pdf [Consulta 2015, Febrero 12].

Dromey, R., “Cornering the Chimera”, IEEE Software, Vol. 13, No. 1, 1996, pp. 33-43.

Gillies, A.C., Software Quality: Theory and Management, Second Edition, Thomson Computer Press, New York, 1998.

Garrett, J. (2005). *Ajax: A New Approach to Web Applications*, [Artículo en línea]. Disponible:http://www.adaptivepath.com/publications/essays/archives/000385.php [Consulta 2013, Diciembre 10].

González, F. (2008).*Gestión de flotas. Planes de mantenimiento de vehículos y organización del tráfico*, [Revista en línea]. Disponible: http://www.revistacesvimap.com/revista77/pdfs/Ingenieria.pdf [Consulta 2013, Diciembre 10].

Hernández, E., Hernández, J., Lizandra, C. (2001). *C++ Estandar*.

ITP Paraninfo.

Humphrey, W.(1997), *Introduction to the Personal Software Process*, Addison Wesley Longman, Inc., Massachusetts.

.

Hurtado, J. (2000). *La revisión bibliográfica y la fundamentación de la investigación* [Documento en línea]. Disponible:

<http://investigacionholistica.blogspot.com/2008/02/fundamentacin-terica->yconceoptual. html [Consulta: 2013,Diciembre].

Hurtado. (2008). *La investigación proyectiva.* [Documento en línea]. Disponible:

http://investigacionholistica.blogspot.com/2008/02/la-investigacinproyectiva.

html [Consultado: 2013, mayo 26].

Izquierdo, L. (2001). *Introducción a la Programación Orientada a Objetos.*

[Documento en línea]. Disponible:

http://luis.izqui.org/resources/ProgOrientadaObjetos.pdf [Consultado: 2015, febrero 12].

KOCH, N. (2006). Comparación de métodos de desarrollo de Aplicaciones Web.

Ludwig-Maximilians-University, Munich, Alemania: Instituto de la Ciencia de la Computación.

Laudon, K. y Laudon, J.  (2008). *Sistemas de Información Gerencial*. México:

Editorial Pearson. 10ma Edición.

Mendoza, L., Pérez, M., Grimán, A., y Rojas, T. (2010). *Algoritmo para la evaluación de la calidad sistémica del software*. [Documento en línea]. Disponible:http://www.lisi.usb.ve/publicaciones/02%20calidad%20sistemica/calidad\_44.pdfVenezuela. [Consulta 2015, Febrero 12].

Montaña, M. (2008, Junio). Las Fases: Comparativa, Analítica y Explicativa del Proceso Metodológico [Blog en línea]. Disponible: <http://martaglezm.blogspot.com/2008/06/esquema-del-tema-5-las-fases.html> [Consulta: 2013, Diciembre].

Navarro, Wendys. (2011). *La tecnología avanza a grandes pasos.* [Documento

en línea] Disponible: http://wendysnavarro.blogspot.com/2011/06/latecnologia-

avanza-grandes-pasos.html[Consulta: 2013, mayo 26]

NORMA VENEZOLANA COVENIN 3049-93. Mantenimiento Definiciones

Padrón, J. (1998). *La Estructura de los Procesos de Investigación.* [Documento en línea] Disponible en: http://padron.entretemas.com/Estr\_Proc\_Inv.htm [Consulta: 2013, Diciembre].

Peinado, M. (2009). *Optimización de la flota de camionetas tipo pick – up asignadas a las áreas administrativas de CVG Ferrominera Orinoco,* [Documento en línea]. Disponible: <http://www.monografias.com/trabajos-pdf4/optimizacion-flota-camionetas-tipo-pick-up-cvg-fmo-ca/optimizacion-flota-camionetas-tipo-pick-up-cvg-fmo-ca.pdf> [Consulta 2013, Diciembre 10].

Pressman, R. (2002) *Ingeniería del Software. Un enfoque práctico*. España: McGraw-Hill Interamericana.

Senn, J. (1990) Sistema de Información para la administración. Ciudad de México (México). Grupo editorial Iberoamérica.

Serna, H. (2000) Gerencia Estratégica, (8va. Edición), Bogotá (Colombia). 3R Editores.

Thibaud, C. (2006). Mysql 5: instalación, implementación, administración, programación. Ediciones ENI.

Zambrano, S., y leal, S. (2005). *Fundamentos Básicos de Mantenimiento.*

San Cristóbal, Táchira, Venezuela: Fondo editorial UNET.

Zambrano, S. (2010). *Sistema para el Control de la Gestión de Acciones de Mantenimiento en Ambiente Web.*

San Cristóbal, Táchira, Venezuela: Fondo editorial UNET.