







**UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO**

---

---

**FACULTAD DE INGENIERÍA**

**DESARROLLO DE APLICACIONES WEB PARA  
LA ADMINISTRACIÓN ESCOLAR  
USANDO SOFTWARE LIBRE  
CASO DE ESTUDIO ESCUELA PARTICULAR**

**TESIS**

**PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACIÓN**

**PRESENTA  
CAB SALINAS ANIBAL JOSE**



**DIRECTOR DE TESIS  
M.I. AURELIO SÁNCHEZ VACA**

**CIUDAD UNIVERSITARIA 2014**



## Índice

- A- Introducción**
- B- Planteamiento del problema**
- C- Objetivo general**

### **1- Software libre**

1.1- Historia del software libre.....	7
1.1.1- ¿Qué es software libre? .....	8
1.1.2- Definición de software libre.....	9
1.1.3- Código abierto.....	9
1.2- El proyecto GNU.....	9
1.2.1- Fundación software libre.....	11
1.2.2- Fundación Software Libre América Latina.....	12
1.3- Licencia Pública General de GNU.....	15
1.3.1- Varias licencias y comentarios acerca de las mismas.....	16
1.4- Administración de sistemas de información basados en software libre.....	16
1.4.1- El software libre en la educación.....	17
1.4.2- Enfoque económico empresarial.....	18
1.4.3- Mercado del software libre.....	19

### **2- World Wide Web**

2.1- El World Wide Web Consortium (W3C).....	23
2.1.1- La Web.....	23
2.1.2- Una página Web.....	23
2.1.3- El Internet.....	24
2.2- Estándares Web.....	25
2.2.1- Accesibilidad Web.....	27
2.2.2- Iniciativa (WAI).....	28
2.2.3- Internacionalización (I18N).....	29
2.2.4- Interacción Multimodal.....	30
2.3- El futuro de la Web.....	32
2.3.1- Web 2.0.....	32
2.3.2- Web 3.0.....	35
2.3.3- Web semántica.....	35
2.3.4- Páginas dinámicas.....	37
2.3.5- Aplicaciones de Internet enriquecidas (RIA).....	38
2.3.6- Páginas offline.....	39
2.3.7- DOM.....	40

### **3- Herramientas**

3.1- Base de datos.....	43
3.1.1- Historia.....	44
3.1.2- MySQL.....	45
3.2- IDE.....	47
3.2.1- Eclipse.....	48
3.3- Frameworks.....	49
3.3.1- jQuery.....	49
3.3.2- Mootools.....	52
3.3.3- CakePHP.....	53
3.4- Lenguaje de programación.....	54
3.4.1- JavaScript.....	56
3.4.2- PHP.....	57
3.5- Diseño gráfico.....	59
3.5.1- The GIMP.....	59
3.6- Desarrollo Web.....	61
3.6.1- HTML.....	62
3.6.2- XHTML.....	63
3.6.3- HTML 5.....	64
3.6.4- AJAX.....	66
3.6.5- CSS.....	66
3.6.6- Firebug.....	68
3.7- Programas manejadores de contenido.....	69
3.7.1- Joomla.....	70

### **4- Desarrollo de software**

4.1- Diseño de una página Web.....	75
4.1.1- No me hagas pensar.....	75
4.1.2- Metodologías de desarrollo de software.....	75
4.1.3- Separación entre contenido y presentación.....	76
4.1.4- Diferentes navegadores.....	76
4.2- Desarrollo rápido de aplicaciones (RAD).....	77
4.2.1- Etapas del ciclo RAD.....	79
4.2.2- Calidad contra rapidez.....	80
4.3- Modelado de gestión.....	80
4.3.1- Planificación de los requisitos.....	86
4.3.2- Flujo de información entre las funciones.....	87
4.3.3- Front-end.....	89
4.3.4- Diseño de interfaz.....	90
4.4- Modelado de datos.....	94
4.4.1- Flujo de información.....	97
4.4.2- Análisis y diseño de datos.....	98
4.4.3- Datos de entrada.....	98
4.4.4- Datos de salida.....	99

4.5- Modelado de procesos.....	100
4.5.1- Objetos del modelo de datos .....	103
4.5.2- Back-end.....	107
4.5.3 - Construcción.....	108
4.6- Pruebas y entrega.....	110
4.6.1- Pruebas en diversos exploradores.....	111
4.6.2- Entrega del programa.....	116
4.7- Análisis de resultados.....	124
4.7.1- El progreso.....	130
4.7.2- Herramientas.....	133
4.7.3- Experiencia.....	134
4.8- Conclusión.....	134

## **5- Apéndice**

A- Glosario.....	139
B- Bibliografía.....	145





## **A- Introducción**

Los sistemas de información y las tecnologías han cambiado la forma en que operan las organizaciones, a través de su uso se logran importantes mejoras, pues automatizan los procesos operativos, suministran una plataforma de información necesaria para la toma de decisiones y, lo más importante, su implementación logra ventajas competitivas o reducir la ventaja de los rivales.

Las tecnologías de la información han sido conceptualizadas como la integración y convergencia de la computación, las telecomunicaciones y la técnica para el procesamiento de datos, donde sus principales componentes son: el factor humano, los contenidos de la información, el equipamiento, la infraestructura, el software y los mecanismos de intercambio de información, los elementos de política y regulaciones, además de los recursos financieros.

Desde hace ya algunos años las organizaciones han reconocido la importancia de administrar los principales recursos haciendo uso de los sistemas de información.

La información se ha colocado en un buen lugar como uno de los principales recursos que poseen las empresas actualmente. Quienes se encargan de las tomas de decisiones han comenzado a comprender que la información no es sólo un subproducto de la conducción empresarial, si no que a la vez alimenta a los negocios y puede ser uno de los tantos factores críticos para la determinación del éxito o fracaso de éstos.

Si deseamos optimizar la utilidad que posee nuestra información, esta se debe manejar de forma correcta y eficiente, tal y como se manejan los demás recursos existentes. Los administradores deben comprender de manera general que hay costos asociados con la producción, distribución y seguridad, almacenamiento y recuperación de toda la información que es manejada en la organización.

La fácil disponibilidad que poseen las computadoras y las tecnologías de información en general, han creado una revolución informática en la sociedad y de forma particular en los negocios. El manejo de información generada por computadora difiere de manera significativa del manejo de datos producidos manualmente.

El uso de tecnología de información para la globalización y la reingeniería de procesos da como resultado el desarrollo de sistemas de información que ayudan a tener una ventaja competitiva en el mercado, utilizándolos para desarrollar productos, servicios, procesos y capacidades que dando una ventaja estratégica sobre las fuerzas competitivas a las que enfrentamos.

## **B- Planteamiento de problema**

**Problemática:** Actualmente el área de administración escolar de la escuela secundaria particular, no cuenta con un sistema de información que le permita hacer una eficiente distribución de horarios y cursos ocasionándole retraso en la planeación de sus actividades escolares, por lo que es necesario implementar una herramienta computacional, que le auxilie en la planeación adecuada de dichas actividades y poder hacer una asignación eficiente y en menor tiempo.

Por otro lado, existe una falta de comunicación entre los administradores del sistema y los usuarios (profesores), ya que existen salones con características particulares que deben ser asignados de acuerdo a la materia y disposición de horario de los usuarios.

Es importante mencionar que la asignación de salones tiene varias particularidades, y que en ocasiones necesita más trabajo complejo y tedioso si no se tiene el orden adecuado, ocasionando que no se pueda tener el máximo beneficio en la asignación de salones.

**Necesidades:** Es necesario crear un diseño de base de datos que la planeación en la asignación de horarios y salones tomando en cuenta todas sus particularidades y poder hacer una distribución eficiente.

Desarrollar una aplicación Web que permita la fácil administración en el proceso de distribución de las materias y los salones.

Administrar peticiones de los usuarios y poder atenderlas de acuerdo a sus necesidades en cuanto a equipamiento del salón y la disponibilidad de horarios.

Para hacer el desarrollo de la aplicación Web se tiene pensado diseñarlo haciendo uso del software libre y herramientas de licencia libre, como un ejemplo de su importancia para crear aplicaciones para uso y beneficio de la comunidad.

**Alcance:** Con la plantación del sistema se pretende tener una adecuada planeación semestral en la distribución de salones normales y equipados, horarios y cursos.

**Justificación:** Los sistemas de información hoy en día se hacen indispensables, sobre todo en la administración escolar, lo que permite disponer de la información en tiempo y forma, para poder tomar decisiones y satisfacer las necesidades de los usuarios.

El uso de software libre permite desarrollar sistemas confiables con un costo bastante económico.

### **Método**

- Hacer el análisis de requerimientos para el desarrollo del sistema Web.
- Diseño de la base de datos.
- Modelo conceptual del sistema.
- Análisis de requerimientos de software y hardware.
- Desarrollo de la base de datos, procesos y sitio Web.

### **C- Objetivo general**

Desarrollar una aplicación Web que permita tener una planeación semestral eficiente en la distribución de salones, materia y horarios.

Administrar peticiones de los usuarios y poder atenderlas de acuerdo a sus necesidades requeridas para la impartición de sus asignaturas, curso u otra actividad académica.

### **Resultados esperados**

Contar con un sitio Web desarrollado con software libre, que nos ayude en la tarea de distribuir los horarios de clases y tener una planificación idónea de salones.

Contar con una base de datos confiable que nos permita generar reportes de uso de salones, horarios por asignatura, nombre de usuario, etc.



# 1

# Software libre

---

---



## 1.1- Historia del software libre

La definición de Software Libre, significa que el software respeta la libertad de los usuarios y la comunidad. En términos generales, los usuarios tienen la libertad de copiar, distribuir, estudiar, modificar y mejorar el software. Con estas libertades, los usuarios (tanto individualmente como en forma colectiva) controlan el programa y lo que hace.

En enero de 1984 Richard Stallman renunció a su empleo en el Instituto Tecnológico de Massachusetts o MIT (Massachusetts Institute of Technology) por desarrollar software para GNU, con el objetivo de crear una comunidad que compartiera software. En septiembre de 1984 empezó con el desarrollo de Emacs (editor de texto) de GNU, para las tareas de edición. El proyecto ganó popularidad, se podía descargar desde un anónimo FTP del ordenador del MIT, mucha gente deseaba el software, así empezó la empresa de distribución de software libre, precursora de la que actualmente distribuye sistemas GNU basados en Linux.

Su trabajo no se destaca realmente por el software, es por sus ideales de libertad para el usuario, que muchos conocen como copyleft. El objetivo de GNU es otorgar libertad a los usuarios, no solo crear una colección completa de software denominado software libre. Curiosamente para asegurar que esas libertades sean respetadas se debió colocar varias restricciones, como marcar que todo software que los usan parcial o totalmente, deben heredar las mismas libertades del software libre. Busca preservar al software libre como tal evita que se agregue a otros proyectos que pudiera convertirse en software privativo. Estos ideales no se limitan al software, pueden ser aplicados a casi cualquier cosa, como la misma documentación del software que presenta formas más simples de este ideal. Es muy recomendado para los artistas principiantes, al permitir la distribución y venta de sus trabajos siempre que respeten datos como autor original y si fue alterado o no el producto original.

El copyleft como método de conocimiento describe los derechos de copia no de venta. Otorgar a los usuarios el permiso para ejecutar, copiarlo, modificarlo y redistribuir el programa; respetando los derechos de autor y las libertades de los usuarios. La aplicación del copyleft para el software de GNU es con la Licencia Pública General de GNU o GPL (GNU General Public License), existen muchos tipos de licencia parecidas pero con diversos cambios para circunstancias específicas, por ejemplo los manuales de software de GNU están cubiertos por una licencia parecida pero menos estrictas.

Cuando el proyecto de GNU crece se toma la decisión de crear la fundación para el software libre o FSF (Free Software Foundation), creada en 1985 por Richard Stallman principalmente se encargó de la distribución de Emacs, otros programas libres de GNU y la venta de manuales libres. La mayoría de los ingresos de la fundación provenía de:

---

- Copias del software libre.
- CD-ROM con código fuente.
- CD-ROM con los binarios.
- Manuales impresos.
- Financiación de las cuotas de socios.

La venta es siempre respetando las libertades de los usuarios de redistribuir, modificar y ejecutar la colección de software en la plataforma de su preferencia.

### **1.1.1- ¿Qué es software libre?**

La dependencia de la sociedad por la tecnología es muy grande, así que si queremos una sociedad libre el software que ejecuta debe ser libre. El software libre le da la libertad al usuario sobre la tecnología que utilizamos en nuestros hogares, escuelas y negocios, donde los equipos (computadoras) trabajan para nuestro beneficio individual y colectivo, no para los propietarios de software privativo que intentan restringir y controlar el acceso a la tecnología.

El software libre es más que un conjunto de programas, es un movimiento social que busca la libertad de los usuarios. Con este software el usuario tienen la libertad de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Significa que el usuario tiene las cuatro libertades esenciales:

- La libertad de ejecutar el programa, para cualquier propósito.
- La libertad de estudiar el programa y modificarlo su funcionamiento. El acceso al código fuente debe ser posible para el usuario.
- La libertad de redistribuir copias.
- La libertad de distribuir copias de sus propias versiones modificadas a terceros. El acceso al código fuente debe ser posible para el usuario.

Con el software libre el usuario tiene la libertad de usarlo en privado, en su trabajo o proyecto. Estas libertades resaltan que el propósito del usuario es el que importa, no el propósito de los programadores. Los programadores que quitan estas libertades al usuario buscan obtener control del mismo a través del control que ejerce su programa en el usuario. El Software Libre es una cuestión de libertad, no de precio.

---



### 1.1.2- Definición de software libre

La definición a tenido cambios para adaptarse a los cambios continuos del medio, el software libre sigue siendo aquel que respeta las cuatro libertades del usuario.

- La libertad de ejecutar el programa, para cualquier propósito.
- La libertad de estudiar el programa y mirificarlo su funcionamiento. El acceso al código fuente debe ser posible para el usuario.
- La libertad de redistribuir copias.
- La libertad de distribuir copias de sus propias versiones modificadas a terceros. El acceso al código fuente debe ser posible para el usuario.

### 1.1.3- Código abierto

En 1998, nace una campaña de nombre “código abierto“ (Open Source), formada por una parte de la comunidad del software libre que no estaban de acuerdo con todos los ideales que representaba.

La filosofía del código abierto consiste en compartir el código fuente del programa con el fin de lograr un “mejor” software, básicamente es un método para lograr crear un mejor software. En el software libre es un imperativo la ética, porque pone la libertad del usuario ante todo, ética que asegura que el usuario es el dueño del programa no al revés. La diferencia consiste en que el código abierto es una metodología de programación, el software libre es un movimiento social. Lo que significa que aunque el Open Source nos da acceso al código fuente no es software libre.

El software libre es un movimiento social para el cual la libertad es un imperativo ético porque solamente el software libre respeta la libertad de los usuarios y busca remarcar que no son partidarios del código abierto. En sus principios algunos usuarios y programadores de software libre no estaban de acuerdo con las metas del movimiento por lo que en 1998 una parte de la comunidad se bifurcó y empezó a hacer una campaña en nombre del código abierto (Open Source), el término fue propuesto originalmente para evitar malentendidos con el término software libre. El código abierto es una metodología de programación, su filosofía considera a los usuarios un medio para hacer mejor al software, en sentido práctico solamente.

## 1.2- El proyecto GNU

Proyecto iniciado por Richard Stallman con el objetivo de crear un sistema operativo completo y libre, el sistema GNU fue diseñado para ser compatible con UNIX, un sistema operativo que no es libre. Funciona bajo la Licencia General Pública de GNU (GPL), gracias a ella cualquier usuario tienen la libertad de ejecutar, copiar, modificar o distribuir

---

el sistema. El sistema operativo goza de un conjunto de aplicaciones, bibliotecas, herramientas de programación y un programa para interactuar con el hardware denominado núcleo o kernel (The Hurd), el núcleo propio de GNU está lejos de poderse usar, por lo que usa un núcleo denominado Linux, creado por Linus Torvalds. La combinación del sistema y el núcleo es el sistema operativo denominado GNU con base en Linux o GNU/Linux, GNU/Linux es llamado erróneamente Linux, la FSF lucha por establecer el término GNU/Linux. Stallman creó en 1985 la Fundación para el Software Libre o FSF (Free Software Foundation), para proveer soporte logístico, legal y financiero el proyecto GNU.



**Imagen 1.1 – Logotipo del sistema operativo GNU.**

En los ochentas la mayoría del software era privativo, lo que limitaba las libertades del usuario e impedían la cooperación entre los mismos.

En 1983 Richard Stallman concibió el proyecto para desarrollar un sistema operativo completo basado en software libre nombrado “Proyecto GNU” como una forma de devolver el espíritu cooperativo de comunidad al eliminar los obstáculos impuestos por los dueños de software privativo. El sistema operativo GNU es un sistema completo de software libre, GNU es un acrónimo recursivo que significa GNU No es Unix (GNU is Not Unix), compatible con Unix por el diseño y funcionalidad, facilitando a los usuarios de Unix cambiar a GNU.

En 1990 se contaba con los componentes principales, excepto el núcleo. Entonces Linux un núcleo similar a Unix, fue programado por Linus Torvalds en 1991 y lo liberó como software libre el 1992. La combinación de Linux (núcleo) con GNU (sistema) formó un sistema operativo completo, el sistema GNU/Linux (GNU con base en Linux) es completamente funcional y ha sido traducido a diversos idiomas.

El proyecto GNU no se limita a sistemas operativos, busca proporcionar una amplia gama de software de aplicación para cubrir las necesidades de los usuarios, desde juegos hasta ofimáticas de los cuales podrás encontrar un catálogo en la página de “La Fundación para el Software Libre” (Free Software Foundation):

[http://directory.fsf.org/wiki/Main\\_Page](http://directory.fsf.org/wiki/Main_Page)

---

### 1.2.1- Fundación software libre

La fundación para el software libre o FSF (Free Software Foundation), es una organización sin fines de lucro, creada principalmente por Stallman en 1985, busca proveer soporte logístico, legal y financiero al proyecto GNU. Tiene como misión:

*Promover la libertad de las personas usuarias de computadoras y defender los derechos de quienes usan software libre.*



**Imagen 1.2 – Fundación de software libre.**

La dependencia de la sociedad por la tecnología es muy grande, si queremos una sociedad libre el software que ejecuta debe ser libre. El software libre le da la libertad al usuario sobre la tecnología que utilizamos en nuestros hogares, escuelas y negocios, donde los equipos trabajan para nuestro beneficio individual y colectivo, a diferencia del software privativo que busca su beneficio propio controlando el acceso a la tecnología.

La FSF publica diversas licencias enfocadas a diversos casos, pero la más importante es la Licencia Pública General GNU (GNU GPL), aunque existen licencias parecidas es la única documentada con el fin de promover y preservar la libertad del software y los usuarios. Otras licencias importantes que publican incluyen:

- La GNU Lesser General Public License (GNU LGPL).
- La GNU Affero General Public License (GNU AGPL).
- La Licencia GNU Free Document (GNU FDL).

La FSF patrocina el proyecto GNU el cual se esfuerza diariamente para ofrecer un sistema operativo completo hecho en su totalidad de software libre, junto con una amplia gama de software para todo tipo de usuarios, todos licenciados bajo la licencia de software libre. Para más información visita la siguiente página Web:

[www.fsf.org/](http://www.fsf.org/)

<http://www.gnu.org/home.es.html>

También desarrolla varias campañas para mantener las libertades del software libre y facilitar su aceptación por parte del público:

DefectiveByDesign.org es una campaña anti-DRM, donde movilizamos a un gran número de comunidades a rechazar productos de empresas que insisten en usar DRM para controlar a sus clientes. Aprenda más en DefectiveByDesign.org.

<http://www.defectivebydesign.org/>

---

Los 7 Pecados de Windows, campaña en contra Microsoft y el software privativo, en respuesta al último lanzamiento de Microsoft Windows.

<http://es.windows7sins.org/>

PlayOgg, campaña que promueve el uso de formatos de audio y video libres que no tengan restricciones de patentes que por sus mismas características afectan las libertades que promueve el software libre y dificultan su progreso. Para más información visita la siguiente página Web:

<http://www.fsf.org/campaigns/playogg/en/>  
<http://www.playogg.org>

Endsoftpatents, fin a las patentes de software, campaña para acabar con las patentes en el software privativo, afecta principalmente a los usuarios comunes, los desarrolladores de nuevo software y al movimiento del software libre. Para más información visita la siguiente página Web:

<http://endsoftpatents.org/>

Campaña a favor del OpenDocument, lucha por el uso de formatos libres en documentos gubernamentales, presionando a los gobiernos para que adopten políticas que requieran que todos los documentos digitales e información pública, se almacenen y se distribuyan en formatos que sean estándar, abiertos y libres de regalías.

Hardware compatible, campaña que busca que los fabricantes de nuevo hardware que normalmente no les importa apoyar la compatibilidad con el software libre ofrezcan soporte. Ofrece un directorio de hardware compatible para identificar que hardware funciona correctamente con su sistema operativo libre. Para más información visita la siguiente página Web:

<http://www.fsf.org/resources/hw>

### **1.2.2- Fundación Software Libre América Latina**

Free Software Foundation Latin America (FSFLA) o La Fundación Software Libre América Latina es la organización latinoamericana miembro de la red internacional de FSF, cuya misión es:

*Es defender los derechos y las libertades de usuarios y desarrolladores de software, luchar por tu libertad de ejecutar software para cualquier propósito, de estudiar su código fuente si quieres y adaptarlo para que haga lo que quieras, de copiarlo, distribuirlo y publicarlo cuando quieras, con o sin mejoras, de manera que tú, nosotros y todos podamos usar computadoras en libertad.*

---



**Imagen 1.3 – La Fundación Software Libre América Latina**

Es una organización sin fin de lucro, formada por personas que apoyan el Software Libre y los ideales que representa, buscan promover, defender el uso y desarrollo de Software libre. Algunas de sus principales actividades.

- **Noticias:** Información sobre eventos y actividades del mundo del software libre.
- **Traducciones:** Traducción de documentos, páginas, boletines entre otros.
- **Campaña contra DRM:** Campaña en contra de los DRM (Gestión Digital de Restricciones).
- **Estándares Libres:** Lucha contra el monopolio de los estándares propietarios.
- **Iniciativa (GNU)^2:** Grupos enfocados en promover el contacto de FSFLA con comunidades a favor del Software Libre en América Latina.
- **Aspectos legales:** El impacto de las legislaciones, normas vigentes, tratados de libre comercio, el uso, distribución y desarrollo de Software Libre.
- **Eventos:** El desarrollo y participación en eventos para promover la filosofía del Software Libre en la comunidad.
- **Sitio Web:** La página Web de la FSFLA recibe continuas actualizaciones.



**Imagen 1.4 – Freedo, logotipo oficial de la Fundación Software Libre América Latina.**

FSFLA es una organización hermana que comparte los mismos ideales y objetivos con:

**Free Software Foundation (FSF):**

<http://www.fsf.org/>



Imagen 1.5 – Fundación de software libre.

**Free Software Foundation Europe (FSFE):**

<http://fsfe.org/>



Imagen 1.6 – Logotipo de la fundación de software libre Francia.

**Free Software Foundation France (FSFF):**

<http://www.fsf.org.in/>



Imagen 1.7 – Logotipo de la fundación de software libre Francia.

Para más información visita la siguiente página Web:

<http://www.fsfla.org/svnwiki/>

---

### 1.3- Licencia Pública General de GNU

La Licencia Pública General de GNU o GNU General Public License (GPL o GNU GPL), usada principalmente por la Fundación para el Software Libre, fue creada para el software pero puede ser aplicada a muchos otros objetos. Normalmente las licencias restringen las libertades y resaltan el control del programador sobre el usuario, GPL se enfoca en las libertades del usuario como instalar, ejecutar, modificar, estudiar y compartir, para asegurar estas libertades tiene cláusulas que impiden agregarles otras licencias que los vuelvan privativos o a las variaciones de estos, así todas sus versiones hereden la licencia perpetuando sus propiedades. La licencia para adaptarse a las nuevas necesidades ha cambiado y actualmente existe la versión 3 (GPLV3). La licencia para el software libre no es sinónimo de gratis, puede ser vendido sin ningún problema, siempre y cuando se respeten todas las libertades que lo representan.

Una de sus principales características es poder estudiarlo, modificarlo y distribuir copias modificadas completas o parciales del programa, esto alienta a la comunidad a participar en el desarrollo y asegura que aunque los desarrolladores originales no continúen alguien más lo hará. Siempre deben ser claros y visibles las siguientes 2 notas: imponer los derechos al software y las características de la licencia que le dan permiso legal para copiar, distribuir y / o modificar la misma. Cuando se crea una versión modificada se debe dejar en claro que es una versión modificada, no posee una garantía de funcionamiento y los errores no se pueden atribuir al autor original.

Esta licencia brinda todos los derechos al autor como en el copyright siempre y cuando se respeten todos los términos de la licencia, la licencia prohíbe hacer copias con derechos de autor no relacionados a usted. Esta licencia se puede aplicar sobre trabajos de terceros siempre y cuando ellos aporten solo modificaciones para usted. Prohíbe sublicenciar su trabajo. Al aceptar esta licencia renuncia a cualquier método de control sobre el producto como medio de limitación de las libertades del usuario, cosas como evitar su modificación o uso. Usted puede cobrar el precio que considere adecuado por el producto.

Características para la distribución de copias:

- **Copias literales:** Copias idénticas del programa, siempre y cuando se mantenga visible un aviso de copyright, tipo de licencia, avisos de ausencia de garantía, proporcione una copia de esta licencia y el programa.
  - **Versiones modificadas:** Es una versión modificada del programa basado en otro con esta licencia, siempre y cuando cumpla con las condiciones:
    - a) Debe poseer avisos marcando que es un trabajo modificado y su fecha de modificación.
    - b) Avisos claros sobre su licencia y las modificaciones añadidas.
-

c) El producto puede ser empaquetado por el método que usted elija, pero debe ser licenciado como un todo.

d) Toda interfaz interactiva presente en el trabajo deben mostrar avisos legales sobre la licencia, al nivel de complejidad que usted crea adecuado.

### 1.3.1- Varias licencias y comentarios acerca de las mismas

Actualmente la licencia GPL existe en la versión 3 (GPLV3), a sufrido cambios para adecuarse a las necesidades actuales y seguir cumpliendo con su función de proteger al software y a los usuarios. Existen muchos tipos de licencias diferentes a la GPL que protegen las libertades de los usuarios, pero es la única documentada explícitamente para ello, igualmente existen versiones menos estrictas pero con el mismo objetivo, dependiendo de lo esperado para tu producto.

- **Licencia de Documentación Libre de GNU (FDL GNU):** Esta licencia es la principal usada por la FSF para sus documentos.
- **Licencia Completamente Permisiva de GNU (GNUAllPermissive):** Esta basada en la licencia GPL GNU y es compatible, es recomendable para esos pequeños archivos como soporte, mensajes e instrucciones.
- **Licencia Pública General Reducida de GNU (LGPLv3):** Licencia para software libre, pero esta versión permite el enlazado con módulos privativos.
- **Licencia Pública General Affero de GNU (AGPLv3):** Es muy parecida a la licencia GPLv3 pero no es compatible, aunque pueden trabajar en conjunto para crear nuevos proyectos, permite a los usuarios que interactúan en una red recibir la fuente de tal software.
- **Educational Community License 2.0 (ECL2.0):** Licencia para software libre compatible con la GPLv3, para las organización que trabajan con esta licencia no tenga que poner bajo esta licencia todas sus demás patentes.

### 1.4- Administración de sistemas de información basados en software libre

Un mito en el que muchos creen es que “costo es igual a calidad”, por lo que consideran al software libre como ocio o un simple conjunto de herramientas, pero la verdad es que es un sistema operativo completo con los recursos necesarios para una empresa o usuario. Existe muchas versiones del sistema operativo enfocados a

---



diversos propósitos y niveles de complejidad, actualmente es posible sustentar un negocio completamente basado en software libre.

En costos, para una empresa el pago de licencias consume muchos recursos por lo que los sistemas basados en software libre son rentables, equivale a un sistema completo y de alta calidad sin pagar un centavo. Las pequeñas empresas con poco capital empiezan a adoptar estos sistemas para aumentar su competitividad, de hecho muchas empresas están cambiando al software libre por su estabilidad.

En el software libre existen herramientas enfocadas especialmente para las empresas que pueden reemplazar o superar a las versiones privativas. Para una empresa existen los Planificación de Recursos Empresariales o Enterprise Resource Planning (ERP), que consiste en un conjunto de herramientas enfocadas en cubrir todas las necesidades de un proyecto, en el software libre existe ERP GNU un subproyecto de GNU que ofrece un ERP de calidad basado completamente en software libre.

Algunas herramientas software libre útiles para una empresa:

- OpenOffice Writer, procesador de texto.
- MPlayer, multimedia.
- OpenOffice Calc, hoja de cálculo.
- Firefox Mozilla, navegador Web.
- Open Office, paquete de Ofimática.
- Mozilla Thunderbird, gestor de correo electrónico.
- Gimp, retoque fotográfico.
- 7-zip, compresor de archivos.
- OpenOffice Impress, creación de presentaciones.
- Xpdf, visor de archivos PDF.

#### **1.4.1- El software libre en la educación**

El software privativo es predominante en la educación, las empresas usan la educación como método de venta al facilitar copias gratuitas de sus programas a las escuelas para crear dependencia en los alumnos acostumbrándolos a su modelo de trabajo. El software privativo limita las posibilidades de desarrollo a otras herramientas para mantener el control sobre sus usuarios, el software libre lucha contra este método promoviendo sus ideales de comunidad y aprendizaje, las principales razones para su enseñanza son:

- **Bajos costos:** Las escuelas no pueden gastar grandes cantidades de dinero pagando licencias innecesarias, que podría ser más útil en otros ámbitos.
-

- **Promueve el apoyo a la comunidad:** Al difundir los ideales de compartir y solidaridad así la comodidad, enseñando que el software al igual que el conocimiento debe ser libre y para todos.
- **Ofrece total libertad a los usuarios:** Libertad sobre el software, facilitando el migrar a otras herramientas o utilizarlas como base para la creación de nuevas herramientas.
- **La creación de buenos programadores:** Promueven la modificación y creación de nuevo software, ayudando a los programadores novatos a practicar con grandes proyectos, ganando conocimiento para crear sus propios grandes proyectos.
- **Otras herramientas:** Facilita la migra a nuevas herramientas poniendo una amplia gama de ellas a fácil disposición, permitiendo al usuario elegir la que más le acomode.

#### 1.4.2- Enfoque económico empresarial

Cuando se piensan en software libre muchos piensan en gratis, pero se equivocan su licencia describe las libertades que posee el usuario, no su precio. El software libre representa un nuevo modelo de negocios, así que no se le puede aplicar los viejos modelos de negocios y esperar los mismos resultados. Un método reconocido es la venta de “servicios no licencias”, cuando un software se establece como una buena solución, aparece la necesidad de servicio al cliente.

Los valores del software libre no son una debilidad, pueden convertirse en el mejor atractivo del producto cuando son bien manejados, su venta es posible y el precio puede ser puesto por la persona que lo vende pero las ganancias no serán tan grandes como con otros métodos, para compensar hay que basarse en más de un estilo de venta. Muchos usuarios que gusten de tu herramienta querrán personalizarla de acuerdo a sus necesidades, ofrecer servicio al cliente modificando lo que ya programaste es un gran modelo de negocios además de que puedes ser agregado a la herramienta y aumentar así su calidad. Los paquetes de software que incluyen otro software o los documentos del mismo aumentan el atractivo del producto.

Algunos ejemplos de negocios con el software libre:

- Adecuar los sistemas operativos al nuevo hardware.
  - Los servicios de asistencia.
  - Mantenimiento de computadoras con herramientas de software libre.
  - La creación de pequeñas herramientas, pidiendo pequeños donativos a los usuarios que les agrade tu trabajo.
-

- Desarrollar nuevas versiones del software existente y ofrecer servicios de asistencia.
- Los desarrolladores pueden dejar contactos en la página de su software para ofrecer sus servicios como programador.

### **1.4.3- Mercado del software libre**

Al legalizar un software libre, establecemos las libertades del usuario no su precio, el software libre también puede ser vendido. Muchos consideran que costo equivale a calidad, mucho software libre iguala o supera a versiones privativas, recibe apoyo de la comunidad y grupos dedicados, lo que asegura estar directamente en contacto con las necesidades del usuario.

En el mercado algunas empresas empezaron su uso por la libertad que pose en comparación con el privativo, poder adecuar el software a sus necesidades es un gran ahorro de tiempo y esfuerzo. Con el software privativo están obligados a comprar y ejecutar el software bajo las condiciones del programador. Las PYMES lo eligen por la ventaja de tener una documentación libre para capacitar a su personal y la posibilidad de tener un sistema completo sin necesidad de gastar un centavo.

La publicidad afecta el cómo la gente ve nuestro producto, por lo que muchos creen que al no tener mucha publicidad el software libre no llegará muy lejos, pero recibe publicidad de la FSF y sus propias características son su mayor publicidad, el ser fácil de encontrar y usar, son la mejor publicidad. Actualmente el Internet es un gran medio de comunicación y la calidad es el mejor método de publicidad.

El negocio está en los servicios no en las licencias, es importante entender que es un producto diferente basado en libertades y no se le puede aplicar los mismos sistemas de negocios esperando los mismos resultados, en el software libre el negocio se obtiene indirectamente. La competencia en este producto es difícil, pero esto no es un pretexto para el cambio, este nuevo modelo de negocios que se basa en las libertades no en el control y restricciones, busca no solo lucrar si no también dejar un aporte a la comunidad.

---



**2**

**La**

**World Wide Web**

---

---



## 2.1- El World Wide Web Consortium (W3C)

La **The World Wide Web Consortium** (Consortio World Wide Web), o también conocida como W3C organización independiente y neutral, con el objetivo de desarrollar estándares relacionados con la Web también conocidas como “recomendaciones”, que sirven como referencia para construir una Web accesible y eficiente, en la que se puedan desarrollar aplicaciones cada vez más robustas.

<http://www.w3.org/>



Imagen 2.1 – Logotipo de World Wide Web Consortium.

Para cumplir con esto, tiene varios objetivos:

- Es desarrollar protocolos y directrices.
- Aseguren el crecimiento de la Web.
- La Web para todo el mundo.
- La Web desde cualquier dispositivo.
- La Web de los Autores y Consumidores.
- La Web de los Datos y Servicios.
- La Web de Confianza.

### 2.1.1- La Web

La W3C define la **World Wide Web** (WWW, o simplemente Web) es un espacio de información donde los elementos de interés, denominados como recursos, se identifican a través de identificadores globales llamados **Identificadores de Recurso Uniforme** (URI).

### 2.1.2- Una página Web

Una **página Web** es un documento o información adaptada para la World Wide Web el cual puede ser visualizado de manera remota. Puede estar almacenadas en un equipo local o un servidor Web remoto.

La Web tiene dos bases simples:

---

- **El protocolo HTTP:** Un sistema simple y sencillo de comunicación que permite enviar cualquier tipo de ficheros de una forma fácil, simplificando el funcionamiento del servidor y permitiendo servidores poco potentes atender miles de peticiones y reducir los costes de despliegue.
- **El lenguaje HTML:** Proporciona un mecanismo de composición de páginas enlazadas simple y fácil, altamente eficiente y de uso muy simple.

Nace con el fin de facilitar la información en la CERN, **The European Organization for Nuclear Research** (Laboratorio Europeo de Física de Partículas). Dicho sistema empleaba el hipertexto para estructurar una red de enlaces entre los documentos. Una vez obtenida la aprobación para continuar el proyecto, nació el primer navegador Web, llamado World-WideWeb.

En 1992 el sistema ya se había extendido fuera del CERN. El número de servidores “estables” había aumentado, alcanzando la cifra de veintiséis, a partir de este punto el crecimiento es enorme.

En 1993 la Web ya era merecedora de un espacio en el New York Times. En ese año el lanzamiento de Mosaic, un navegador para X-Window/Unix que con el tiempo se convertiría en Netscape y que fue un factor clave de popularización de la Web. **En 1994 se fundó el WWW Consortium** que se convertiría en el motor de desarrollo de los estándares predominantes en la Web. A partir de ese momento, el crecimiento ya fue constante, convirtiéndose hacia finales de los noventa en el servicio insignia de Internet y dando lugar al crecimiento imparable de los servicios en línea que conocemos actualmente.

### 2.1.3- El Internet

Nace a mediados de la década de los setenta, bajo los auspicios de DARPA(Defense Advanced Research Projects Agency), Agencia de Proyectos de Investigación Avanzados de Defensa para Estados Unidos. DARPA inició un programa de investigación de técnicas y tecnologías para unir diversas redes de conmutación de paquetes, permitiendo así a los ordenadores conectados a estas redes comunicarse entre sí de forma fácil y transparente.

De estos proyectos nació un protocolo de comunicaciones de datos, IP(Internet Protocol) que permitía a ordenadores diversos comunicarse a través de una red o Internet formada por la interconexión de diversas redes.

A mediados de los ochenta la Fundación Nacional para la Ciencia norteamericana, la NSF, creó una red, la NSFNET, que se convirtió en el backbone (el troncal) de Internet junto con otras redes similares creadas por la NASA (NSINet) y el U.S. DoE (Department of Energy). En Europa, la mayoría de países disponían de backbones nacionales (NORDUNET, RedIRIS, SWITCH, etc.) y de una serie de iniciativas paneuropeas (EARN y RARE). En esta época aparecen los primeros proveedores de acceso a Internet privados que ofrecen acceso pagado a Internet.

---



A mediados de los noventa se inició el boom de Internet. En esa época el número de proveedores de acceso privado se disparó, **permitiendo a millones de personas acceder a Internet que a partir de ese momento ya se empezó a conocer como la Red**, desbancado a las demás redes de comunicación existentes (Compuserve, FidoNet/BBS, etc.). El punto de inflexión vino marcado por la aparición de implementaciones de TCP/IP gratuitas (incluso de implementaciones que formaban parte del sistema operativo) así como por la popularización y abaratamiento de medios de acceso cada vez más rápidos (módems de mayor velocidad, RDSI, ADSL, cable, satélite).

A partir de esta época, gracias entre otras cosas a la amplia disponibilidad de implementaciones de la suite de protocolos TCP/IP (formada por todos los protocolos de Internet y no sólo por TCP e IP), algunas de las cuales eran ya de código libre, Internet empezó lo que posteriormente se convertiría en una de sus características fundamentales, un ritmo de crecimiento exponencial, hasta que a mediados del 2002 empieza a descender ligeramente el ritmo de crecimiento.

## 2.2- Estándares Web

Un **estándar** es un conjunto de reglas normalizadas que describen los requisitos que deben ser cumplidos por un producto, proceso o servicio, con el objetivo de establecer un mecanismo base para permitir que distintos elementos hardware o software que lo utilicen, sean compatibles entre sí.

**En la creación de los estándares(Recomendaciones)** participan el equipo del W3C, expertos, invitados y cualquier usuario de la Web que quiera mostrar su opinión. **Todos ellos trabajan conjuntamente a través de un proceso basado en el consenso, la neutralidad y la transparencia de la información.** El objetivo es lograr una Web universal, accesible, fácil de usar y en la que todo el mundo pueda confiar.

Ejemplos de algunos de los estándares Web más conocidos y ampliamente utilizados son:

- HTML (HyperText Markup Language), define la estructura de los documentos.
  - XML (eXtensible Markup Language), transmitir datos, útil para un gran número de tecnologías.
  - CSS (Cascading Style Sheets), asigna estilo para la representación de los documentos.
-



**Imagen 2.2 – Logotipo del estándar HTML 5.**

Sus principales objetivos son:

**Acceso universal:** Busca que el acceso a la Web sea posible desde cualquier punto sin importar desde el tipo de hardware o software involucrado en el proceso.

**Una Web con significado:** Anteriormente a la Web podía ser considerada como un conjunto de documentos estáticos los cuales están expresados en lenguaje natural lo que limitaba a las computadoras el momento de procesar el contenido. Pero el nuevo enfoque buscado por la W3C es que todo el contenido como texto hasta un video tenga un contexto para permitirle a la computadora manipularlo. La Web se convertirá en una gran base de datos de elementos que culminara en una organización y reestructuración con base a la semántica de sus elementos. La Web semántica fomentara el rendimiento y eficiencia de la Web.



**Imagen 2.3 – Logotipo de la Web semántica.**

**Confianza en la Web:** La Web evoluciono para convertirse en un gran medio colaboración entre usuarios: ventas, blogs, negocios, diversión, etcétera. Esto requiere de confianza entre los usuarios, se han creado diversas formas: firmas digitales de documentos que fomentan la responsabilidad de las personas que se declaran autores de estos; encriptación de los datos para la confidencialidad; mecanismos de establecimiento y declaración de las políticas de privacidad de los datos de los sitios Web.

---

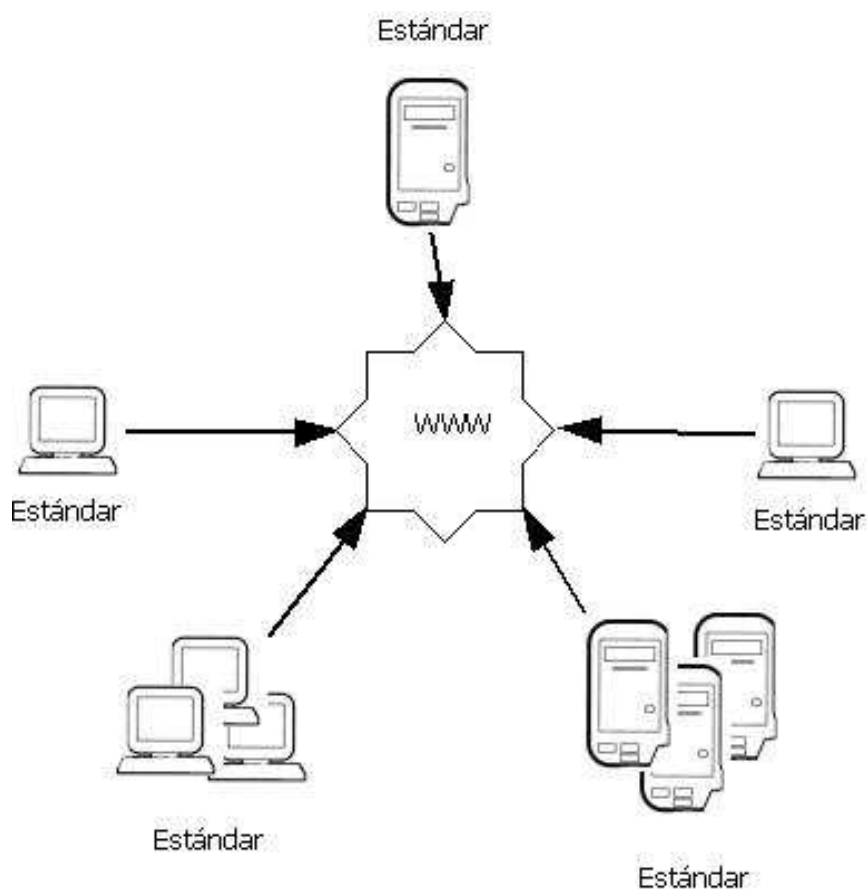


Imagen 2.4- Diversos dispositivos y aparatos comunicándose entre sí gracias al uso de estándares, ejemplificando la importancia de su desarrollo.

### 2.2.1- Accesibilidad Web

Los avances en la tecnología inalámbrica y la gran cantidad de aparatos con la opción de conexión inalámbrica: telefonía móvil, en el automóvil, en los electrodomésticos o en los televisores manifiesta **la necesidad de utilizar tecnologías y lenguajes unificados, libres y gratuitos que no se limiten por patentes comerciales.**

El acceso a **la Web debe ser posible desde cualquier punto** sin importar el tipo de hardware o software involucrado en el proceso. Además de las posibles restricciones técnicas:

- La existencia de múltiples idiomas.
- Las diversas localizaciones geográficas.
- Las diferencias culturales.
- Tradiciones.
- Limitaciones físicas, psíquicas o sensoriales de los usuarios.
- Persona con discapacidad.

Para cumplir con estos objetivos se desarrollaron las **Pautas de Accesibilidad al Contenido en la Web (WCAG)**, el cual es un conjunto de recomendaciones para el diseño de páginas Web accesible, permitiendo que la información llegue a todos los usuarios.

WCAG consiste en 14 pautas que proporcionan soluciones de diseño para evitar que una página produzca problemas de acceso a la información. Las Pautas contienen además una serie de puntos de verificación que ayudan a detectar posibles errores.

Cada punto de verificación está asignado a uno de los tres niveles de prioridad establecidos por las pautas:

- **Prioridad 1:** Son aquellos puntos que un desarrollador Web tiene que cumplir ya que, de otra manera, ciertos grupos de usuarios no podrían acceder a la información del sitio Web.
- **Prioridad 2:** Son aquellos puntos que un desarrollador Web debería cumplir ya que, si no fuese así, sería muy difícil acceder a la información para ciertos grupos de usuarios.
- **Prioridad 3:** Son aquellos puntos que un desarrollador Web debería cumplir ya que, de otra forma, algunos usuarios experimentarían ciertas dificultades para acceder a la información.

En función a estos puntos de verificación se establecen los niveles de conformidad:

- **Nivel de Conformidad "A":** Todos los puntos de verificación de prioridad 1 se satisfacen.
- **Nivel de Conformidad "Doble A":** Todos los puntos de verificación de prioridad 1 y 2 se satisfacen.
- **Nivel de Conformidad "Triple A":** Todos los puntos de verificación de prioridad 1, 2 y 3 se satisfacen.

### 2.2.2- Iniciativa (WAI)

La accesibilidad Web busca el acceso para toda persona a la Web sin importar sus limitaciones físicas, psíquicas o sensoriales. Para cumplir con los ideales de accesibilidad se creó la WAI (Web Accessibility Initiative), **Iniciativa de Accesibilidad Web** desarrollada por el W3C, **cuyo objetivo principal es facilitar el acceso a la Web a todas aquellas personas con discapacidad. Consiste en un diseño Web que va a permitir que estas personas puedan percibir, entender, navegar e interactuar con la Web** aportando a su vez contenidos. La accesibilidad Web también beneficia a otras personas, incluyendo

---

personas de edad avanzada que han visto mermadas su habilidad a consecuencia de la edad.

La WAI ofrece a aquellas personas con discapacidad una oportunidad de acceder a la información y de interactuar con ella; **pero no esta limitada a este grupo, por ejemplo, una conexión lenta.** La Web accesible es posible en su totalidad pero para eso se requiere de la cooperación de todos.

Entre sus principales actividades se encuentran:

- El desarrollo de pautas de accesibilidad.
- Crear y mejorar herramientas de evaluación y reparación de la accesibilidad.
- Labores educativas en relación con la importancia del diseño accesible de páginas Web.
- Abrir nuevos campos en accesibilidad a través de la investigación en esta área.

Puedes encontrar más información sobre la WAI en la siguiente dirección:

<http://www.w3.org/WAI/>

### **2.2.3- Internacionalización (I18N)**

La Web como herramienta de difusión de información para funcionar debe tener la capacidad procesar la información de modo sencillo y rápido para que la información sea accesible y universal, a esto se le conoce como Internacionalización, un proceso a través del cual se van a diseñar sitios Web adaptables a diferentes idiomas y regiones sin necesidad de realizar cambios en el código.

La Internacionalización también es conocida como I18N que es la abreviatura de Internacionalización porque hay 18 letras entre la i y la n. Para hablar de internacionalización en un sitio Web las partes que la integran deben funcionar bajo cualquier circunstancia, en cualquier país, con cualquier idioma y cultura, por ello el uso de formatos y protocolos que no establezcan barreras a los diferentes idiomas, sistemas de escritura, códigos u otras convenciones locales. Por lo tanto, la creación de un sitio Web internacional permite garantizar su utilización universal incluyendo todos los idiomas y culturas.

Existen identificadores basados en el idioma y lugar como referente para recoger información sobre las preferencias de los usuarios. Por ejemplo **HTML usa el atributo lang** para indicar el idioma de segmentos de contenido. **XML utiliza el atributo xml:lang** con el mismo objetivo.

---

Los idiomas se especifican a través de un código estándar. Primero se indica el idioma (en, inglés), y si se desea, el país (en-US, inglés de EEUU). Para identificar el idioma de nuestra página al inicio del documento, antes de head:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">
```

Para marcar cualquier cambio de idioma que se pueda producir a lo largo del texto en una página Web:

```
<p xml:lang="en">The quick brown fox jumps over the dog.</p>
<p xml:lang="en-GB">What colour is it?</p>
<p xml:lang="en-US">What color is it?</p>
```

Si incluimos un enlace a otra página, el idioma en el que esa página se va a mostrar, esta vez se indicará con hreflang:

```
<a href="http://www.w3.org/International/questions/" hreflang="en">
Preguntas frecuentes sobre Internacionalización</a>
```

Para más información revisar la siguiente dirección:

<http://www.w3.org/International/getting-started/#basics>

## 2.2.4- Interacción Multimodal

En la actual gran explosión de distintos dispositivos de introducción de datos los cuales nos llevan a diversos modos de interacción entre usuarios y dispositivos presentes principalmente en los dispositivos móviles los cuales sufren de distintas limitaciones como pantallas pequeñas, su posición de uso, teclados pequeños, etcétera. El uso de opciones para facilitar la navegación al usuario es obligatorio, una recomendación para eso es mezclar voz, teclado, ratón, pantallas táctiles, etcétera obteniendo diferentes mecanismos de acceso a la información, puedan interactuar conjuntamente ofreciendo una experiencia satisfactoria al usuario.

**La Interacción Multimodal o Multimodalidad** es el proceso con el cual dispositivos y personas son capaces de llevar a cabo una interacción (auditiva, visual, táctil y gestual) conjunta desde cualquier sitio, en cualquier momento, utilizando cualquier dispositivo y de forma accesible, incrementando la interacción entre personas y dispositivos.

La W3C trabaja en estándares como VoiceXML, SSML, SRGS, SISR, InkML y herramientas de integración de datos de entrada como EMMA. Pero se hace cada vez más difícil frente a un mercado en expansión, en lo que se refiere a dispositivos de acceso a la información, arquitecturas, sistemas operativos, etcétera. Por este motivo la estandarización adquiere una gran importancia.

---

Algunos de los estándares de la W3C para **la Multimodalidad** son:

- **VoiceXML**: Es un lenguaje de etiquetado que permite crear diálogos con los que se puede interactuar escuchando comandos hablados, controlables a través de entradas de voz. Se encarga de convertir habla en texto y para eso utiliza, entre otros mecanismos SRGS (Gramática de Reconocimiento del Habla).
- **SRGS**: Gramática de Reconocimiento del Habla, permitir que una aplicación de voz indique a un reconocedor qué es lo que tiene que escuchar, es decir palabras, modelos en los que estas palabras surgen, lenguaje hablado de cada palabra, etcétera.
- **SSML**: Lenguaje de marcado de Síntesis del Habla, basado en XML, forma parte del proceso de salida de información y funciona como componente de conversión ayudando a generar habla sintética. Este lenguaje tiene como principal objetivo ofrecer a los autores de contenido sintetizable un camino estándar para controlar aspectos del habla como pueden ser la pronunciación, volumen, tono y velocidad, mejorando así la calidad del contenido sintetizado.
- **SISR**: Permite describir anotaciones sobre reglas gramaticales para extraer resultados semánticos a través del reconocimiento del habla. El resultado obtenido puede expresarse en formato XML, usando el lenguaje EMMA.
- **EMMA**: Es un lenguaje utilizado para el intercambio de datos en sistemas de administración de interacción multimodal. Es una especie de lenguaje común utilizado en la comunicación entre componentes de un sistema multimodal, cuyo objetivo es integrar la entrada de datos procedente de los usuarios desde diferentes recursos, y darle forma para ser procesada en una representación única que será a su vez procesada por componentes avanzados de procesamiento de información.
- **InkML**: Convierte en texto símbolos escritos y mensajes. El componente de reconocimiento de escritura puede usar un modelo de gestos de escritura, un modelo de lenguaje y una gramática para convertir la escritura en palabras especificadas en una gramática.

Algunos dispositivos de introducción de datos (reconocimiento de entrada de datos):

- Teclado.
  - Dispositivos de señalización.
  - Visual.
  - Lenguaje de signos.
  - Sistema de marcación por tonos, también llamado sistema multifrecuencial o DTMF (Dual-Tone Multi-Frequency) o entradas táctiles.
  - Verificación de habla.
  - Identificación de escritura.
-

### 2.3- El futuro de la Web

La Web 1.0, su objetivo era facilitar el acceso a investigaciones punto del cual tuvo una gran expansión pero publicar algo seguía siendo complejo y costoso, los únicos con acceso eran las compañías que la convirtieron en un gran catalogo para promocionar productos y servicios, los usuarios no tenían ninguna clase de poder sobre su contenido.

Con la explosión de nuevas tecnologías las cuales redujeron su costo y facilitaron publicar información, dando oportunidad a cualquier usuario de plasmar sus ideas. Se convierte es un escaparate de información ya que la información se mueve en una sola dirección, los usuarios que observan pero no pueden modificarlo ni opinar sobre su contenido.

La Web 2.0, en los últimos años la Web ha sufrido un cambio sorprendente al dejar de ser un escaparate de una dirección para convertirse en una plataforma abierta, con una arquitectura basada en la participación de los usuarios. Alrededor del concepto de retroalimentación logrado por el Software social, arquitectura de la participación, contenidos generados por el usuario, aplicaciones ricas, etiquetas, intercambio de contenidos y redes sociales; son algunos de los conceptos que enriquecen este fenómeno. La Web a evolucionado asta ser el espejo de una comunidad, un lugar que no pertenece a las grandes empresas sino a todos los usuarios para compartir sus ideas, como concepto de una red social.

La Web para los usuarios ha cambiado mucho desde sus inicios, podemos englobarlo en tres fases:

- **Web 1.0:** La información era en una dirección, los usuarios solo podían observar la información en la Web pero no podían interactuar con ella.
- **Web 2.0:** En este punto evoluciono para convertirse en una Web social, donde sin ser el dueño de la página un usuario puede modificar, aportar o eliminar información de la misma.
- **Web 3.0:** Es el futuro, una Web que aproveche la semántica, inteligencia artificial, inteligencia colectiva y gestión del conocimiento. En este punto todo lo que hagamos en la red dejara de ser texto plano para convertirse en información que le será fácilmente manipulable a la Web.

#### 2.3.1- Web 2.0

La Web en sus inicios estaba limitada a una dirección (mostrar información), un lugar sólo accesible para las grandes empresas lo que la convertía en un gran comercial para servicios y productos, cuando surgió la idea de una Web social con retroalimentación por parte de los usuarios es cuando nace el concepto de Web 2.0.

---



El término Web 2.0 en 2004 momento desde el cual se ha difundido como virus. La Web como idea social Según Tim O'Reilly, principal promotor de la noción considera que sus principios constitutivos son siete:

- La World Wide Web como plataforma de trabajo.
- El fortalecimiento de la inteligencia colectiva.
- La gestión de las bases de datos como competencia básica.
- El fin del ciclo de las actualizaciones de versiones del software.
- Los modelos de programación ligera junto a la búsqueda de la simplicidad.
- El software no limitado a un solo dispositivo.
- Las experiencias enriquecedoras de los usuarios.
- Generadores de contenidos.

Este fenómeno social obtuvo su auge gracias a la exceso de generadores de contenido y miles de páginas buscando usuarios, pero sobre todo a su gran número de aplicaciones que permitían la interacción entre usuarios, permitiendo el intercambio desde comentarios y marial multimedia. Algunas de sus aplicaciones más representativas son:

- Wikipedia.
- YouTube.
- Flickr.
- WordPress.
- Blogger.
- MySpace.
- Facebook.
- OhMyNews.

Existe mucha polémica con respecto al termino Web 2.0 al punto de ser cuestionado por más de una persona en el ramo, algunos consideran que es más correcta para definir las aplicaciones Web cooperativas; otros consideran que solo es el nombre dado a la falta de ingenio tecnológico de la época al punto de no saber si la WEB 2.0 es inteligencia colectiva o un medio fast food. Un ejemplo del problema son las páginas donde cualquiera puede subir su opinión sobre cualquier tema sin moderador verdaderamente calificado, convirtiendo la páginas en un almacén de información irrelevante y dudosa. Sea cual sea el caso es claro que la Web ha cambiado.

Según O'Reilly (2005), los cambios que sufrió la Web y sus aplicaciones para convertirse en 2.0 tiene siete principios constitutivos los cuales se enumeran a continuación:

**1.- La World Wide Web como plataforma:** Desde su principio el software propietario a controlado a los usuarios con sus continuos pagos de licencia y restricciones, igual era el Internet en sus inicios siendo sólo un escaparate de productos y servicios, pero al ganar poder los usuarios de la Web obligo a cambiar a un mercado que dependía de ella, muchas empresas de software se adaptaron y permiten acceder a

---

versiones limitadas de su software en línea ahorrando al usuario la necesidad de instalarlo en su computadora, la idea tras esta estructura es que la aplicación mejore continuamente con el uso y comentarios de los usuarios. Una aplicación bajo esta premisa es YouTube el cual a crece continuamente gracias a la contribución de la comunidad.

**2.- Aprovechar la inteligencia colectiva:** Esta idea refuerza los ideales originales de la Web, la idea de una comunidad como inteligencia colectiva donde todos los usuarios pueden participar aportando sus conocimientos para crear algo más grande como es el software libre el cual crece con el apoyo de una comunidad. La inteligencia colectiva está creciendo gracias a la nueva generación de aplicaciones Web, con software en línea diseñado a partir de una interfaz simple, escalable y de acceso gratuito. El ejemplo más grande de este desarrollo es la página Wikipedia la cual permite unir el conocimiento de cientos de usuarios.

**3.- La base de datos como competencia básica:** En la actualidad la información es tan o más importante que el software que la maneja. La cantidad de información que producen los usuarios en la Web es enorme y muy valiosa; muchas empresas se han dado cuenta del potencial de la Web como recolector de información, su uso en estadísticas de mercado puede decir en que sector del público tiene mayor aceptación su producto o de que modo presentarlo, se habla de entender al público. Este es el potencial de las bases de datos, almacenar grandes cantidades de información y muchos métodos de administración.

**4.- El fin del ciclo de las actualizaciones de versiones del software:** La Web 2.0 propone el fin del ciclo de actualizaciones del software propietario; con el gran número de aplicaciones Web gratuitas y un nuevo modelo de negocios surge la pregunta ¿si remplazara al software empaquetado?, con su caducidad planificada y continuo coste de licencias. Siendo más que un modelo de ventas es un modelo de desarrollo al poner prototipos en versiones beta y colocarlos en línea, utilizando el método de ensayo y error para aprender del consumo de los usuarios; así se mejora la herramienta en forma constante sin costo adicionales para la empresa y acelerando el proceso de mejora de la aplicación, logrando tener software de mayor calidad en menos tiempo. Un ejemplo es como Google, pone a nuestra disposición un gran conjunto de aplicaciones Web, la pregunta es ¿cuántas personas pagarían por este software empaquetado pudiendo tenerlo a su alcance online?.

**5.- Modelos de programación ligera:** Este enfoque de programación busca promover la simplicidad, escalabilidad y eficiencia en las aplicaciones para reducir el tiempo de descarga. Pensando en la simplicidad las aplicaciones sólo tienen las funciones de mayor uso, comparando con el gran número de funciones que posee el software empaquetado y las que realmente utiliza la gente. La necesidad de generar formas creativas de reutilizar código para la estandarización, integración y usos bajo demanda, con partes rápidamente adaptables las aplicaciones pueden mantenerse al día de las necesidades de los usuarios siempre pensando en ellos también como productores de contenido.

---

**6.- El software no limitado a un solo dispositivo:** Con el surgimiento de los dispositivos móviles (mobile devices) como preferencia de comunicación obliga a la Web a adaptarse a las nuevas necesidades y sus limitaciones de estos dispositivos como el teclado, la resolución y tamaño de pantalla. El estilo de diseño y las nuevas ideas que eran exclusivas de estos aparatos se han transformado en un común para muchas páginas.

**7.- Experiencias enriquecedoras del usuario:** La Web está llena de aplicaciones visualmente sorprendentes para el intercambio de información, pero no alcanzan el nivel de interacción con el usuario de las aplicaciones que la convirtieron en un punto de reunión más complejo. Juegos como Second life, emuladores de la vida real en el cual miles de jugadores pueden realizar cualquier actividad cotidiana facilitando compartir experiencias e ideas, plataformas parecidas se empiezan a usar con fines didácticos como aulas virtuales para los usuarios a distancia o medio de visita a lugares muy lejanos como museos.

### **2.3.2- Web 3.0**

La expresión Web 3.0 apareció por primera vez en 2006 en un artículo de Jeffrey Zeldman, describe la evolución del uso, la interacción de los usuarios en Internet pronosticando la transformación de la red en una gran base de datos distribuida con la información en múltiples repositorios basados en XML, RDF y microformatos para lograr la conversión de cualquier dato estructurado en una página Web. Se espera convertir a las páginas actuales de texto a contenido semántico el cual pueda ser fácilmente manipulado por programas inteligentes. Esta tecnología se ha implementado en pequeña escala en compañías para conseguir una manipulación de datos más eficiente. Este modelo de evolución se apoya en tecnologías llamadas asíncronas para recibir los datos de forma independiente.

Muchos no están de acuerdo con el término y han propuesto otros términos y enfoque para la red pero entre ellos este es el que más fuerza ha tomado.

### **2.3.3- Web semántica**

La Web actual sufre de varios problemas como son: sobrecarga de información y heterogeneidad de fuentes de información lo que afecta a la interoperabilidad. Estos le dificultan a los usuarios la búsqueda de información al producir resultados redundantes o no relacionados con la búsqueda. La Web Semántica busca resolver estos problemas permitiendo a los usuarios delegar tareas en software, este es capaz de procesar su contenido y realizar deducciones lógicas para resolver problemas cotidianos automáticamente.

---

En la Web Semántica el contenido tendrá un valor de dato más definido con significado para que el manejo de información y búsqueda en la misma sean más rápidas y sencillas. Con este paso se espera dotar a la Web de:

- Más significado en su contenido.
- Más semántica.
- Mejorar la infraestructura.
- Mejorar el procesamiento de la información.
- Agilizar procesos.
- Transferir información de forma sencilla y rápida.

En la Web actual es común guardar pequeños fragmentos de información sobre los gustos del usuario como es: color, diseño, más visitados, videos favoritos, etcétera. En la Web 3.0 se espera tener una base de datos con las preferencias de los usuarios y la información de Internet, hacer búsquedas de información más exacta para los usuarios, en las búsquedas actuales los resultados contienen muchos errores o son irrelevantes, en una búsqueda semántica los resultados serían exactos.

Para lograr una definición correcta de datos y crear una infraestructura global en la que es posible compartir, reutilizar datos y documentos entre diferentes tipos de usuarios se requiere de varias herramientas:

- **Marco de Descripción de Recursos:** RDF (Resource Description Framework), proporciona información descriptiva simple sobre los recursos que se encuentran en la Web, usado comúnmente en catálogos de libros, directorios, colecciones personales de música, fotos, eventos, etcétera. Mediante RDFa se pueden representar los datos estructurados visibles en las páginas Web (eventos en calendarios, información de contacto personal, información sobre derechos de autor, etcétera), a través de unas anotaciones semánticas incluidas en el código e invisibles para el usuario, lo que permitirá a las aplicaciones interpretar esta información y utilizarla de forma eficaz.
  - **SPARQL:** SPARQL (SPARQL Protocol and RDF Query Language), lenguaje de consulta sobre RDF que permite hacer búsquedas sobre los recursos de la Web Semántica utilizando distintas fuentes de datos.
  - **Lenguaje de Ontologías Web:** OWL (Web Ontology Language), es un mecanismo para desarrollar temas o vocabularios específicos. Proporciona un lenguaje para definir ontologías estructuradas que pueden ser utilizadas a través de diferentes sistemas. Las ontologías que se encargan de definir los términos utilizados para describir y representar un área de conocimiento, son utilizadas por los usuarios, las bases de datos y las aplicaciones que necesitan compartir información específica, es decir en un campo determinado como puede ser el de las finanzas, medicina, deporte, etcétera. Las ontologías incluyen definiciones de conceptos básicos en un campo determinado y la relación entre ellos.
-

- **Espigar Descripciones de Recursos de Dialectos de Lenguajes:** GRDDL (Gleaning Resource Descriptions from Dialects of Languages), es el puente que permite convertir datos expresados en formato XML (como XHTML) en datos de Web Semántica. Con GRDDL, los desarrolladores transforman los datos que desean compartir en un formato que puede ser utilizado y transformado nuevamente para aplicaciones más rigurosas.

Las posibilidades de estas herramientas son muy grandes sobre todo en aplicaciones sociales, un ejemplo sería una aplicación de calendario que importe directamente eventos que encuentra al navegar por la Web, o se podrían especificar los datos del autor de una foto publicada, así como la licencia de documento y así permitir recopilar información rápida y eficientemente para el usuario común.

### **2.3.4- Páginas dinámicas**

Existen muchas tecnologías y métodos para crear páginas Web, actualmente se busca que una página Web interactúe lo más posible con el usuario respondiendo a sus necesidades y con el contenido a la mano usando lo menos posible de recargas que traigan nueva información del servidor, ya que es importante mantener la atención del usuario y los continuos llamados al servidor requieren de tiempo.

Las páginas Web creadas con lenguaje de marcado HTML que traen su contenido total al cargarse la página y que no lo modifican ni lo crean son conocidas como páginas estáticas. Este tipo de páginas requieren que el editor publique de nuevo toda la página cuando es necesario modificar su contenido por lo que se utilizan para temas que no requieren cambios constantes. Pueden ser comparadas con un libro, donde uno sólo puede ver el contenido pero no podemos modificarlo o interactuar con él, su principal ventaja es su creación fácil y económica.

HTML, aunque muy útil a pequeña escala, resulta bastante limitado a la hora de crear grandes sitios o portales, es necesario usar otros lenguajes (accesorio) mucho más versátiles, capaces de responder de manera inteligente a las demandas del usuario y que permiten la automatización de tareas tediosas y necesarias como las actualizaciones. Estos lenguajes capaces de recrear el contenido de la página a partir de ciertos "scripts" dan pie a las páginas dinámicas. Este tipo de páginas suele ser extensas y sus contenidos cambia rápidamente, donde el interés es la automatizar del contenido.

Las páginas dinámicas son páginas HTML que generan contenido a partir de lenguajes de programación (scripts) que son ejecutados en el servidor Web o en el cliente (computadora del usuario). Pueden ser comparadas con un televisor que responde a los comandos del usuario y las necesidades del creador. Este tipo de páginas se apoya en lenguajes interpretados como JavaScript, y en tecnologías como Ajax. El nuevo contenido puede se genera en el cliente por los lenguajes interpretados o ser solicitada al servidor.

---

### 2.3.5- Aplicaciones de Internet enriquecidas (RIA)

Las Aplicaciones de Internet Enriquecidas o RIA (Rich Internet Applications), surgen de la competencia por el desarrollo de software en la Web y la necesidad de reducir el continuo tráfico cliente-servidor. Estas aplicaciones mezclan las características de las aplicaciones Web con las características de las aplicaciones de escritorio tradicionales. Su principal propiedad es no necesitar nuevas recargas de página (peticiones al servidor), ya que desde el principio se carga toda la aplicación, y sólo se hacen peticiones al servidor por nuevos datos.

Su estructura se apoya en la idea “cliente-servidor” en vez de un desarrollo Web tradicional donde el estado se mantiene en el servidor por sesiones. Para reducir la carga de trabajo en el servidor es necesario repartir procesos entre el cliente (el usuario) y el servidor (proveedores de recursos o servicios). Al usuario se le ocultan las peticiones a servidor, ellas traen la información requerida para evento y no más, lo que evita pérdida de tiempo para el usuario en recargas innecesarias de la página para traer nueva información que en muchos casos es mínima.

La comunicación cliente-servidor se basa en varias tecnologías como AJAX (JavaScript Asíncrono y XML) en el navegador Web del cliente. El cliente se encarga la interacción entre el usuario y la interfaz de usuarios, maneja las peticiones de datos al servidor y controla como se presentan los datos. El servidor, se encarga de varias actividades:

- Manejo de datos.
- Procesar peticiones de la aplicación cliente.
- Delega las acciones en el servidor.
- Guardar datos en la base de datos.
- Extraer datos de la base de datos.
- Actualizar los archivos del sistema.
- Retornar datos al servidor.
- Procesamiento analítico.
- Determina y le da formato a los datos que son retornados al cliente.
- Prepara la interacción con otras aplicaciones del servidor.

#### Beneficios

Las aplicaciones RIA aunque es clara su limitación en comparación con las aplicaciones de escritorio tienen muchas ventajas en comparación con las mismas:

- No necesitan instalación (solo es necesario mantener actualizado el navegador Web).
  - Las actualizaciones hacia nuevas versiones son automáticas.
-

- Se pueden utilizar desde cualquier ordenador con una conexión a Internet sin depender del sistema operativo que este utilice.
- Generalmente es menos probable la infección por virus que utilizando por ejemplo programas ejecutables.
- Más capacidad de respuesta, ya que el usuario interactúa directamente con el servidor, sin necesidad de recargar la página.
- Ofrecen aplicaciones interactivas que no se pueden obtener utilizando solo HTML, incluyendo arrastrar y pegar, cálculos en el lado del cliente sin la necesidad de enviar la información al servidor.
- Evita la problemática del uso de diferentes navegadores al abstraerse de ellos a través de un Framework.
- Conectividad y despliegue instantáneo de la aplicación.
- Agilidad de acceso.
- Garantizan la desvinculación de la capa de presentación es decir acceso a la aplicación desde cualquier computador en cualquier lugar del mundo.

### **2.3.6- Páginas offline**

Las páginas Web contienen gran cantidad de funcionalidades para atraer a los usuarios pero todas ellas tienen la misma limitante, depender de una conexión a Internet para su funcionamiento o descarga por lo que se busca mantener una máxima funcionalidad en situaciones sin conexión. En un mundo que depende cada día más de las bases de datos esto parece un reto imposible pero con el surgimiento de las nuevas tecnologías HTML5 esto es posible. Aunque el Internet tiene un alcance incomparable con sus inicios no se puede depender de la conexión ya que existen situaciones y lugares donde no exista la conexión, ya sea leer un documento o un juego de PC crear páginas Web que funcionen adecuadamente en casos sin conexión a Internet es importante. En la tecnología HTML5 existe una característica llamada offline application cache que permite el diseño de páginas Web offline, así las páginas Web pueden funcionar correctamente sin conexión.

Una página Web que funciona sin necesidad de conexión a Internet es conocida como página offline. Aunque se habla de ellas como una nueva tecnología la posibilidad de una página Web sin conexión ya existía desde sus inicios, el simple hecho de descargar una página Web como documento HTML nos permite visualizarla su contenido más tarde sin necesidad de conexión ya que tenemos la página completa en nuestro poder, el problema se encuentra en las páginas que para

---

funcionar requieren continuamente información proporcionada del servidor. Consiste en almacenar en la computadora del usuario toda la información que será requerida para su correcto funcionamiento, cuando requiera nueva información tomar la almacenada en el usuario sin llamar al servidor; el almacenamiento de información no tiene que ser al descargar la página por primera vez, puede ser gradual o controlado para evitar problemas por lentitud o fallo en la conexión, así para el usuario todas las actividades serán fluidas.

### 2.3.7- DOM

El Modelo de Objetos del Documento o DOM (Document Object Model) es una interfaz de programación de aplicaciones o API (Application Programming Interface) para documentos como HTML y XML, define la estructura lógica de los documentos y el modo en que se accede y manipula. Es un modelo lógico que puede implementarse de cualquier manera conveniente, la más común es la representación en forma de árbol de un documento. Uno de sus objetivos importantes proporcionar la interfaz estándar de programación que pueda ser utilizada en una amplia variedad de entornos y aplicaciones. Se diseñó para ser utilizado en cualquier lenguaje de programación y proporcionar una especificación del DOM precisa e independiente del lenguaje. El DOM permite a los programadores y a otras aplicaciones manipular un documento, entre las más importantes se encuentran:

- Construir documentos.
- Navegar por su estructura.
- Añadir elementos y contenido.
- Modificar elementos y contenido.
- Eliminar elementos y contenido.

Permite modelar la página Web y todos sus elementos como: párrafos, divisiones, campos, etcétera. Con él se puede acceder usando JavaScript a cualquier elemento para alterar sus propiedades o invocar a sus métodos, incluso eliminar o agregar nuevos elementos.

El DOM no define que información en un documento es relevante ni estructurada, modelo de objeto que especifica aplicaciones. Este documento usa diagramas que muestran relaciones padres/hijos, son relaciones lógicas definidas por las aplicaciones de programación, no representaciones de ninguna estructura interna de datos en particular.

---



# 3

# Herramientas

---

---



### 3.1- Base de datos

Las bases de datos siempre han existido, porque no se limitan solo al modo digital; pueden ser una colección de documentos u objetos, surgen de la necesidad del almacenaje y administración de la información. En una base de datos pueden existir miles o millones de datos, para su correcto control existe una herramienta muy importante y poderosa. Las bases de datos digitales han significado un gran paso en la historia, desde las enormes bibliotecas que debían ser administradas a mano hasta hoy que con un simple comando podemos borrar o agregar todo su contenido; en un mundo en el cual la información es más valiosa que nunca y el mantenerla actualizada y accesible es constante. Cientos de empresas dependen de sus bases de datos para su correcto funcionamiento, sin ellas la sociedad no sería la misma.

**Una base de datos** puede ser definida como un conjunto de elementos ya sean físicos o virtuales con relación entre sí para su administración. Una base de datos y el software que la administra se le denomina **sistema de administración o gestor de bases de datos DBMS** (Database Management System) que ayuda al administrador a realizar las diversas tareas.

Algunas de las opciones más comunes en un DBMS:

- Modificar registros.
- Estructuras de datos.
- Buscar información.
- Registrar datos.
- Crear nuevas bases de datos.
- Agregar registros.
- Eliminar registros.
- Recuperar datos (búsqueda).
- Modificar datos.
- Eliminar datos.

Para el correcto funcionamiento de un DBMS se guarda información de dos tipos:

- Los datos de usuarios (datos usados para administrar).
- Los datos de sistema (datos para el funcionamiento del DBMS como nombres de usuario).

Debido a la gran cantidad de información a manejar en una base de datos existe una herramienta un **lenguaje de consulta estructurado** o **SQL** (Structured Query Language) un lenguaje declarativo que nos permite manipular la información en la base de datos. Algunas de sus características son:

---

- El manejo del álgebra.
- El cálculo relacional.
- Búsqueda entre grandes cantidades de datos.
- Recuperación de datos.
- Modificación de gran cantidad de datos con pocos comandos.

Algunas ventajas:

- 1.- Fácil manipulación de la información.
- 2.- Funciones rutinarias programadas y automáticas.
- 3.- Organiza la información de manera eficaz.
- 4.- Permite elaborar programas para crear sistemas, reportes, utilerías, etcétera.

Distintos tipos de bases de datos:

- Base de Datos Relacional.
- Base de Datos Jerárquica.
- Bases de Datos Red.
- Bases de datos transaccionales.
- Bases de datos relacionales.
- Bases de datos multidimensionales.
- Bases de datos orientadas a objetos.
- Bases de datos documentales.
- Bases de datos deductivas.

### **3.1.1- Historia**

Las bases de datos han existido desde que el hombre a buscado almacenar y manipular información, desde un conjunto de documentos u objetos, hasta la información de una población entera.

La evolución de las bases de datos fue de la mano con el desarrollo de la computación y la informática. A principios de los años 60 no existía un buen modo de almacenar la información, varias empresas desarrollaron programas y métodos para solucionar este problema; estos son los principios de los “administradores de bases de datos”, pero estaban muy lejos de ser las grandes bases de información que conocemos hoy en día.

En 1884 Herman Hollerith estadístico estadounidense crea una máquina tabuladora o censadora que usaba tarjetas perforadas, para facilitar la tarea la cual se realizaba a mano por lo que muchos lo consideran el primer informático de la historia.

---

En los años 50's las cintas magnéticas facilitaban el almacenaje y respaldo de la información, con la desventaja de que el acceso a la información era lento por ser un método de almacenaje secuencial y a un muy alto costo.

En los 60's las computadoras reducen su costo haciéndolas accesibles para las empresas que vieron en ellas un modo fácil y rápido de acceso a la información. Aparecen las primeras generaciones de bases de datos de red y jerárquicas, empieza el uso de estructuras de árbol y lista para representar información. Charles Bachman informático que durante sus actividades en General Electric desarrolla el IDS (Integrated Data Store) uno de los primeros sistemas de gestión de bases de datos.

En los años 70's Edgar F. Codd publica su libro "Un modelo relacional de datos para grandes bancos de datos compartidos" (A Relational Model of Data for Large Shared Data Banks) e inventa el modelo relacional, el modelo de bases de datos más usado. Define las tres primeras Formas Normales, base para la normalización de sistemas de bases de datos. Utiliza por primera vez el término "procesamiento analítico en línea" OLAP (On-Line Analytical Processing) y redactó las doce leyes del procesamiento analítico informático.

En los años 80's se desarrolla el "lenguaje de consulta estructurado" SQL (Structured Query Language) un lenguaje declarativo de acceso a bases de datos relacionales que permite efectuar consultas con el fin de recuperar información. En 1986 el ANSI adoptó SQL como estándar para los lenguajes relacionales y en 1987 se transformó en estándar ISO. Esta versión del estándar va con el nombre de SQL/86.

En los años 90's el liderazgo en el mercado se debate entre Oracle considerado como uno de los sistemas de bases de datos más completos, competencia directa de SQL Server de Microsoft y con licencia libre MySQL, PostgreSQL y Firebird. Los estudios consideran como el futuro de las bases de datos a las base orientadas a objetos.

### **3.1.2- MySQL**

MySQL es un sistema de administración de bases de datos relacional RDBMS (Relational Database Management System), una de las base de datos de código abierto más popular del mercado; con una comunidad entera apoyando su desarrollo, pero cualquier persona dispuesta contribuir puede hacer con: elementos, resolver problemas, realizar mejoras o sugerir optimizaciones. Se ha convertido en una robusta herramienta para los administradores, una opción recomendable sobretodo para pequeñas y medianas empresas.

---

Algunas de sus principales ventajas:

- Uso del lenguaje de consulta estructurado, SQL (Structured Query Language).
- Multi usuario.
- Entornos de producción críticos.
- Soporta alta carga.
- Usa GNU Automake, Autoconf, y Libtool para portabilidad
- Uso de multi hilos mediante hilos del kernel.
- Soporte para operadores y funciones en cláusulas select y where.
- Soporte para cláusulas group by y order by, soporte de funciones de agrupación
- Soporta gran cantidad de datos, cerca de los 50 millones de registros.
- Un paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL.

Para más información sobre características y licencia visitar la siguiente dirección:

<http://www.mysql.com/>



**Imagen 3.1 – Logotipo de la base de datos MySQL.**

Michael Widenius uno de los fundador de MySQL AB, creada en 1995 desarrollo MySQL en 1996, un sistemas RDBMS código abierto competencia directa de sistemas propietarios como: Oracle, SQL Server y DB2. En enero del 2008 se convierte en una subsidiaria de Sun Microsystems y en abril 2009 Sun Microsystems fue comprada por Oracle Corporation. La toma de MySQL por parte de Oracle causo preocupando al considerarse una estrategia para acabar con la competencia a su base de datos Oracle, aunque asegura dar mantenimiento y apoyo a MySQL por mucho tiempo ya perdió una de sus características más importantes, dejo de ser un software libre, por requerir la compra de una licencia para poder usarlo en proyectos comerciales.

Michael Widenius actualmente trabaja en una nueva base de datos, MariaDB que incluye los principales motores de almacenamiento abierto que caracterizaron a MySQL junto con parches y plug-in desarrollados por empresas o la comunidad. Su motor de almacenamiento "Aria", conocido como Maria del cual recibe su nombre MariaDB.

---

MariaDB busca igualar e incluso reemplazar a MySQL, esta continuamente apoyado por un conjunto de empresas que trabajan para proporcionar el software, soporte y servicios para MariaDB.

Para más información visita la siguiente página Web:

<http://askmonty.org/>

### **3.2- IDE**

Entorno de Desarrollo Integrado, IDE (Integrated Development Environment) es un programa herramienta que le ayuda al programador a crear código fuente. Su objetivo principal es facilitar y agilizar la programación.

Existen IDE's para prácticamente cualquier lenguaje, pero entre los más comunes se encuentran:

- C.
- C++.
- C#.
- Java.
- Python.
- Visual Basic.
- Delphi.
- PHP.
- JavaScript.
- JSP.

Una IDE es recomendable para todo tipo de usuario, desde novatos hasta los programadores expertos, por tener un entorno de trabajo más amigable que permite la administración de los proyectos y aumenta la eficiencia.

Unas de sus principales características son:

- Un editor de texto.
  - Un compilador.
  - Un intérprete.
  - Automatización de tareas rutinarias.
  - Un depurador.
  - Un constructor de interfaz gráfica de usuario.
  - Un sistema de control de versiones.
  - Autocompletado.
  - Un depurador.
-

- Resaltado de sintaxis.
- Compilación en tiempo real.
- Pruebas unitarias con JUnit.
- Control de versiones con CVS.
- Uso de plug-in.

Algunos de los IDE's de más populares actualmente:

- Code::Blocks.
- Eclipse.
- Lazarus.
- Kdevelop.
- Netbeans.

### 3.2.1- Eclipse

Eclipse conocida como una herramienta todo y nada, por su estructura de almacén con la principal característica de poder montar en el otras herramientas (plug-ins) para integrar diversos lenguajes sobre un mismo IDE. No requiere de instalación para ejecutarse, lo que te permite tener varias versiones de Eclipse con diferentes configuraciones para cada proyectos en una sola computadora sin que entren en conflicto. Está escrito en Java, por lo que es necesario para su ejecución tener un JRE (Java Runtime Environment) instalado previamente en el sistema. Su licencia, Licencia Pública Eclipse EPL (Licencia Pública Eclipse) considerada software libre y avalada por la FSF (Free Software Foundation), la misma que usa para sus demás programas.

El proyecto eclipse de naturaleza OpenSource iniciado por IBM, el cual fundó el Consorcio Eclipse al con el apoyo de empresas como: Rational, HP o Borland dedicados a su desarrollo y mantenimiento. En febrero del 2004, el Consorcio Eclipse es independiente de IBM ahora formado por las empresas: HP, QNX, IBM, Intel, SAP, Fujitsu, Hitachi, Novell, Oracle, Palm, Ericsson y RedHat y algunas universidades. El proyecto eclipse que actualmente es una fundación sin fin de lucro, no esta limitado a esta IDE, desarrolla también un gran variedad de herramientas para distintas necesidades.



**Imagen 3.2 – Logotipo del IDE Eclipse.**

Para más información visita la siguiente página Web:

<http://www.eclipse.org/>

---



### 3.3- Frameworks

Los programadores no quieren detener su trabajo con código trivial, su objetivo no es “reinventar la rueda”, para ello utilizan los Frameworks. Son recomendables para tareas básicas y monótonas que se deben realizar diariamente, esto permite enfocarse más en la funcionalidad del código, y menos en su redacción. Un Framework (marco de trabajo) es un conjunto de librerías de código que contienen procesos o rutinas ya listos para usar, una herramienta que facilita tareas cotidianas o repetitivas. En la actualidad el uso de un Framework de trabajo marca la diferencia entre un programador y un especialista, no solo aumenta la calidad, limpieza y rapidez del trabajo, te permite estandarizarlo. Un ejemplo es el desarrollador Web que pierde tiempo creando un código adecuado para cada navegador buscando que la página funcione correctamente en cada uno, los frameworks se encargan de eso, para que el escriba su página una sola vez.

Desventajas, el tiempo gastado en aprender a usar el mismo, la dependencia a sus actualizaciones porque continuamente surgen nuevos frameworks competitivos. La necesidad de agregar código extra para su funcionamiento, que puede causar problemas a la hora de modificar el programa. Existen frameworks para muchos lenguajes, más o menos especializados, el objetivo es siempre elegir el que más se adapte a sus necesidades. Su uso nunca debe reemplazar las habilidades del programador.

#### 3.3.1- jQuery

jQuery es una biblioteca basada en JavaScript, un conjunto de funciones o códigos para la realización de tareas habituales de un modo rápida y fácil. Su principio es “escribir menos y hacer más”, busca que un programador no pierda tiempo creando más código rutinario para enfocarse en la funcionalidad y limpieza de su código. Es un software libre bajo las licencias Instituto Tecnológico de Massachusetts MIT (Massachusetts Institute of Technology) y la Licencia Pública General de GNU, es gratuito y puede ser usado con fin personal o comercial.



Imagen 3.3 – Logotipo del Framework jQuery.

Algunas de sus características:

- Creación de aplicaciones Web.
- Programación del lado del cliente.
- Creación de interfaz de usuario.
- Efectos dinámicos.
- Muchos usos de Ajax simplificado.
- Compatibilidad entre navegadores.
- Funcionamiento con Clases.
- Manejo del DOM.
- Manejo de eventos.
- Animaciones.

Sus de ventajas a destacar:

- **Desarrollo:** Esta continuamente en desarrollo, lo que asegura nuevas soluciones a viejos problemas y compatibilidad entre los exploradores más populares.
- **Curva de aprendizaje:** Su uso es muy fácil, por lo que no se pierde tiempo con sus aprendiendo su gran número de funciones, además cuenta con una extensa documentación con ejemplos.
- **Otras herramientas:** Existe una gran comunidad de desarrollo que basan sus propios proyectos y herramientas en el para cubrir sus nuevas necesidades, creando plug-in y otras herramientas que ponen a disposición del público que son compatibles con jQuery. Al gozar de una gran popularidad en el mercado y los programadores ayuda a elegirla como una herramienta de calidad.

Para más información visita la siguiente página Web:

<http://jquery.com/>

Por la aceptación que a tenido jQuery se a expandió a nuevos ramos con el objetivo de solucionar nuevas necesidades, surgiendo dos nuevas herramientas que usan como base a jQuery, jQuery UI y jQuery Mobile. jQuery UI enfocado en el desarrollo de una interfaz de usuario gráfica y amigables. jQuery Mobile crea páginas para la nueva ola de dispositivos móviles, aplicando una mezcla características entre jQuery y jQuery UI, obteniendo páginas fáciles de crear, visualmente atractivas y con un gran número de funcionalidades; siendo lo mejor que aunque esta enfocado a desarrollo móvil no esta limitado a ellas.

jQuery UI fue desarrollado para diseñadores, programadores y principiantes por igual; gracias a su fácil aplicación y la suficiente flexibilidad para evolucionar con sus necesidades y abarcar muchos casos de uso. Con una gran cantidad de documentación, interacciones con la interfaz de usuario, efectos, widgets y temas; los usuarios podrán elegir entre una gran cantidad de combinaciones para lograr la página deseada.

---

Algunas características importantes:

- Menús de abanico animado, barra de navegación, desplegados y bloques para sistemas de menús complejos.
- El widget spinner muestra los botones para introducir valores numéricos fácilmente a través del teclado o el ratón.
- El widget de información muestra información adicional sobre cualquier elemento en vuelo estacionario o el enfoque.
- Autocompletado de palabras en formularios.
- Creación de pestañas de accesibilidad.
- API fácil e intuitiva.
- Fácil aplicación al diseño HTML5.

Para más información visita la siguiente página Web:

<http://jqueryui.com/>

jQuery Móvil toma la frase "escribir menos, hacer más" y lo lleva al siguiente nivel, al crear una amalgama entre jQuery y jQuery UI para crear páginas fácilmente y prácticamente sin necesidad escribir una sola línea de JavaScript. Anteriormente jQuery resolvía el problema de compatibilidad de nuestro código con todos los exploradores que existían, pero en este caso el reto es mayor al contar con: resolución de pantalla, exploradores y dispositivos. En lugar de escribir aplicaciones únicas para cada uno creara una sola página, la cual no se limita solo a los dispositivos móviles.

Algunas características importantes:

- **Tamaño de pantalla:** Automáticamente adapta el tamaño de tu página a la del dispositivo móvil.
  - **HTML5:** Aunque funciona correctamente en versiones anteriores del HTML, esta adaptada para trabajar con los nuevos estándares HTML5.
  - **Bases de datos:** Esta construido en base a jQuery, por lo que hereda todas las facilidades características de jQuery.
  - **Menos código:** En la mayoría de los casos no tendrás que escribir ninguna línea de código JavaScript para desarrollar una página Web.
  - **Diseño:** El diseño de la página es muy flexible.
  - **Compatibilidad:** No será necesario escribir código para cada navegador ya que te dará el mismo resultado para todos.
  - **Más fácil:** Algunas funciones como Ajax se aplicaran automáticamente si es posible sin necesidad que el programador escriba código indicándolo.
-

- **Diferentes diseños:** Podemos elegir entre varios temas gráficos ya listos para aplicar al aspecto de nuestra página o podemos crear nuestros propios temas personalizados.
- **Herramientas de diseño:** Cuenta con una herramienta adicional que nos permite crear nuestros propios diseños.
- **Varios medios:** Aunque su objetivo principal es el desarrollo móvil no está limitado a este, se puede crear con el páginas tradicionales con excelentes reuntados.
- **Eventos:** Los dispositivos móviles tienen sus propios eventos, y tiene soporte para muchos de ellos.
- **Un solo documento:** Ofrece la oportunidad de desarrollar todas tus páginas Web en un solo documento HTML, facilitando la carga y administración de las mismas.

Para más información visita la siguiente página Web:

<http://jquerymobile.com/>

### 3.3.2- Mootools

MooTools o “mis herramientas orientadas a objetos” (My Object Oriented Tools), es un Framework de JavaScript orientada a objetos para el desarrollador intermedio y avanzado. Con el se puede crear código multi-navegador que asegura que tu desarrollo se comporte igual en todos los navegadores. Cuenta con una extensa documentación y una gran variedad de ejemplos listos para aplicar. Tiene licencia MIT por lo que es gratuito y de código abierto, puede ser usado y modificarlo para cualquier caso de uso.



Imagen 3.4 – Logotipo del sistema de gestión de contenido Joomla!.

---

Algunas de sus características para desarrollo Web:

- Trabajo con capas.
- Diversos efectos de animación.
- Facilita el trabajo con Ajax.
- Es ligero, no aumentar mucho el peso de nuestra página Web.
- MooTools se compone de diversos módulos, podemos agregar solo los que usaremos para no poner mucho peso a nuestra página, que afecte su tiempo de descarga.
- Una gran cantidad de componentes adicionales creados por la comunidad ya listos para usar en nuestras páginas Web como: calendarios, animaciones, etcétera.

Algunas desventajas:

- Los ejemplos en la documentación a veces son escasos o muy complejos.
- Los componentes tienden a cambiar con la versión.
- No todos los componentes trabajan con las versiones más actuales.
- Existen algunos problemas de compatibilidad con CSS.

Para más información visita la siguiente página Web:

<http://mootools.net/>

### 3.3.3- CakePHP

CakePHP es un Framework para PHP con objetivo de facilitar la programación y desarrollo en PHP, ofrece rapidez, flexible y compatible. Libera al programador para plasmar la lógica sin preocuparse por detalles de programación. Con licencia MIT (Massachusetts Institute of Technology), es de libre uso y código abierto. Una de sus principales características es la posibilidad de trabajar con el popular modelo MVC (Modelo Vista Controlador), caracterizado por su programación limpia y ordenada.



Imagen 3.5– Logotipo del Framework CakePHP.

Algunas de sus características:

- Comunidad activa.
- Compatible versiones las más recientes de PHP.
- Rutinas para interacción con bases de datos.
- Generación automática de código.
- Compatibilidad con el modelo MVC (Modelo Vista Controlador).
- URLs personalizadas.
- Función de Validación.
- Plantillas rápidas y flexibles.
- Ayudantes para AJAX.
- Ayudantes para JavaScript.
- Componentes de Email.
- Facilidades para el manejo de seguridad y sesión.
- Facilidades para el manejo de cookies.

Para más información visita la siguiente página Web:

<http://cakephp.org/>

### **3.4- Lenguaje de programación**

Las actividades que debe realizar el hardware para cumplir sus funciones radican en una serie de ordenes (instrucciones), estos diferentes pasos (acciones) se encuentran en algoritmos que se expresan en programas o software. Para crear un programa se usan otros programas conocidos como lenguajes de programación que permitan la comunicación usuario/máquina, sin ellos la programación sería un acto muy difícil y rutinario. Unos programas especiales llamados traductores (compiladores o intérpretes) que convierten las instrucciones escritas de un lenguaje de programación a instrucciones escritas en lenguajes máquina (0 y 1, bits) que ésta pueda entender.

Algunos de los lenguajes de programación más populares:

- C.
  - Java.
  - Objective-C.
  - C++.
  - PHP.
  - C#.
  - Visual Basic.
  - Python.
  - Perl.
  - Ruby.
  - JavaScript.
-

Un programa es básicamente un algoritmo que junto con datos de entrada, que por lo general al cumplirse el algoritmo entrega datos de salida.

Los tipos de lenguajes más comunes son:

- Lenguaje de máquina.
- Lenguaje de bajo nivel (ensamblador).
- Lenguajes de alto nivel.

El lenguaje máquina está escrito en un lenguaje directamente inteligible para la máquina (computadora), está escrito en lenguaje binario (cadenas o series de caracteres, dígitos 0 y 1) o conocido como código binario. Programar en este nivel sigue siendo una tarea difícil y compleja.

Los lenguajes de bajo nivel son más fáciles de utilizar que los lenguajes máquina, aunque no se parecen al lenguaje humano, ya no son cadenas binarias. Es conocido también como ensamblador, pero sigue dependiendo del hardware para el que es creado. Las instrucciones en lenguaje ensamblador son conocidas como mnemotécnicas, un ejemplo de operaciones aritméticas típicas son: ADD, SUB, DIV, etcétera.

Los lenguajes de alto nivel, su escritura y reglas se asemejan al lenguaje humano. Su principal ventaja es una curva de aprendizaje muy pequeña en comparación con los lenguajes de menor nivel.

Principales ventajas de un lenguaje de alto nivel:

- Es ajeno al hardware de la máquina.
- Las modificaciones al programa son más fáciles.
- Reducción del coste de programación.
- Compatibilidad entre máquinas de una a otra.
- Su aprendizaje es más fácil.
- Es muy semejante al lenguaje natural.

Algunos inconvenientes:

- No aprovechan los recursos internos del hardware, como en lenguajes de menor nivel.
  - Necesita más recursos en memoria.
  - Requiere de más tiempo su ejecución.
  - Requieren ser interpretado para su ejecución.
-

Algunos ejemplos de lenguajes de alto nivel:

- C.
- C++.
- #COBOL.
- Fortran.
- Pascal.
- Visual Basic.
- Java.

### 3.4.1- JavaScript

JavaScript es un popular lenguaje de programación que no necesita ser copiado y funciona del lado del cliente reduciendo la carga de procesamiento en servidor. Permite la creación de páginas dinámicas que responden a las actividades del usuario, asegurar una experiencia más placentera de navegación. Para el desarrollo Web es prácticamente básico y hoy en día todos los navegadores soportan JavaScript como un básico.

Para su funcionamiento puede ser insertado en el documento HTML o ser llamado desde un documento externo. Permite la creación desde pequeños scripts hasta programas más complejos orientados a objetos. Su sintaxis es familiar a Java o C, por lo que los usuarios con experiencia en alguno de estos programas podrán fácilmente dar el salto, tiene algunas de las características de la programación orientados a objetos pero mucho más flexible.

Algunas de las características:

- Permite interactuar con archivos CSS, HTML.
- Puede alterar el diseño de una página: color, tamaño, capas, efectos, etcétera.
- Manejo de eventos como: presión de teclas, links, el movimiento del mouse, etcétera.
- Cálculos matemáticas.
- Modificación del DOM.
- Dibujo con nuevas herramientas como canvas.
- Recepción de información con métodos Ajax.

JavaScript agrega sorprendente efectos a nuestros proyectos pero debe ser usado con cuidado, uno de sus principales defectos es el agregar peso a nuestra página, aumentando su tiempo de descarga y afecta su rendimiento. Muchos usuarios preferirían buscar en otra página antes que esperar a que se termine de desplegar la nuestra.

---



### 3.4.2- PHP

El ciclo de vida de una página Web en el pasado era muy repetitivo y difícil, creación, escritura y subirla al servidor, en caso de necesitar modificación se descargaba modificar y subía de nuevo al servidor; la llegada de los lenguajes de programación en el servidor es posible crear páginas fáciles de modificar sin necesidad de descargar la página, con contenido dinámico. Existen muchos lenguajes del lado del servidor que permiten desarrollar páginas Web dinámicamente, en las cuales el contenido puede variar rápidamente a veces sin esfuerzo del desarrollador, estos procesos son completamente invisibles para el cliente. Los scripts se insertan en el documento HTML, pero este desaparece cuando el cliente recibe la página.



**Imagen 3.6 – Logotipo del lenguaje Hypertext Preprocessor ( PHP ).**

Existe una gran lista de lenguajes del servidor, algunos de los más conocidos:

- ASP de Microsoft.
- JSP de Sun.
- Perl.
- Python.
- PHP.

PHP originalmente llamado (Personal Home Page), creado por Rasmus Lerdorf en 1995, actualmente su nombre es un acrónimo recursivo que significa “PHP Hypertext Preprocessor”. Con licencia PHP License, la cual es considerada como software libre por la FSF (Free Software Foundation).

La historia de PHP, Rasmus Lerdorf parte del equipo de desarrollo de Apache creó PHP, inicialmente eran funciones en lenguaje C con el objetivo de lograr conexiones a bases de datos. Al ganar popularidad el proyecto siguió creciendo con el apoyo de la comunidad que utilizaba la herramienta.

En 1995 aparece una revisión llamada PHP/FI 2.0. (Form Interpreter), que ofrecía conexión con la base de datos MySQL (gestor de bases de datos). Su uso llegó a más de 50.000 páginas y contenía un conjunto de herramientas:

- Un analizador.
  - Un libro de visitas.
  - Un contador.
  - Macros para diseño.
-

En 1998 aparece PHP 3.0, ofrece la posibilidad de conexión con un número mayor de gestores de bases de datos, más protocolos y una API ampliada. Implementa la programación orientada a objetos, uso del motor Zend (Zend Engine) y nuevas herramientas que facilitan el desarrollo de páginas Web.

En el 2000P aparece HP 4.0, ofrece el manejo de archivos XML y manejo de imágenes. Se divide en dividida en 3 partes: el motor Zend, la API de servidor y los módulos de funciones.

PHP 5 se mejora el concepto de Programación Orientada a Objetos (POO) que no se logro por completo en la versión PHP 4.0, añadiendo las palabras reservadas `public`, `protected` y `private` a la definición de las propiedades y nuevos métodos para los objetos permitiendo una verdadera encapsulación. Se modifica el significado de PHP a PHP Hypertext Preprocessor. Se implementa el motor Zend 2.0.

Algunas ventajas de PHP 5:

- **Fácil de usar:** Gracias a su similitud con programas de uso común como Java y C no requiere de mucho tiempo para lograr programación básica que resolverá muchos problemas.
  - **Documentación:** Gran cantidad de documentación escrita por otro programadores, en distintos idiomas.
  - **Funciones:** Existen funciones para las actividades más comunes como conexión a bases de datos.
  - **Compilar:** Al ser un lenguaje interpretado no requiere ser copilado para probar su funcionamiento.
  - **Multiplataforma:** No esta limitado por el servidor o el sistema operativo, y es compatible con los servidores más populares.
  - **Licencia:** La licencia Open Source implica que el código fuente es libre de ser descargado, estudiado y modificado por cualquiera sin costo alguno, lo que permite correrlo a la medida de cada usuario.
  - **Costo:** Es una opcion muy potente que rivaliza con sus versiones privativas, además de que funciona sin problema con otras herramientas Open Source como Apache, y PostgreSQL, logrando un sistema completo sin costo, recomendable para pequeñas y medianas empresas.
  - **Bases de datos:** Ofrece el acceso a 15 gestores de bases de datos de uso popular.
  - **Imágenes:** Gracias a sus librerías puede crear imágenes en tiempo de ejecución.
-

- **Errores:** El manejo de errores mediante el uso de excepciones.
- **Comunidad:** Basado en el concepto de código abierto, existe toda una comunidad apoyando su desarrollo, así no es necesario tener a un usuario conocedor de todo, teniendo muchos usuarios apoyando en distintas partes.

### 3.5- Diseño gráfico

Técnica que busca transmitir información (mensaje) con un medio visual o mezcla de varios, se a expandido a muchas ares pero básicamente el objetivo sigue siendo el mismo. Con la explosión digital y el surgimiento de nuevas herramientas, las posibilidades se han multiplicado; por ejemplo un fotógrafo ya no tiene que sufrir por los pequeños defectos en las fotografía, podrá simplemente eliminarlos con las miles de herramientas de manipulación de imágenes que existen.

Para poder cumplir con su objetivo de difundir un mensaje se apoyan en diferentes medios como: visual, tipográfico, arte, cultura, emocional, etcétera buscando elegir el correcto para llegar hasta publico. La búsqueda del medio correcto no debe opacar al mensaje, ya que siempre debe ser lo más claro y directo posible, porque si el mensaje se vuelve ambiguo se pierde el objetivo principal.

La influencia que pose en la toma de decisiones de un cliente potencial es muy grande causando que muchas empresas lo tomen como base en sus campañas de marketing. Algunos lo utilizan para influir en su apariencia e imagen y así ganar reputación, para afectar en la toma de decisiones.

Todas sus características encajan en el desarrollo Web, donde funcionalidad ya no es suficiente para mantener la atención de los usuarios, es muy importante sobretodo si se trata de una página comercial, ante la competidores es importar ganar la preferencia del publico por todos los medios posibles.

#### 3.5.1- The GIMP

GIMP es un acrónimo de “programa de manipulación de imágenes de GNU” (GNU Image Manipulation Program), funciona en muchos sistemas operativos y bajo muchos idiomas. Hoy en día se encuentra en casi cualquier distribución de Linux. Aunque muchos no lo consideran competencia para los programas propietarios como PhotoShop, su gran cantidad de plug-in y su continuo crecimiento por el equipo llamado GIMP-Team, lo califican como una buena opción.

---



**Imagen 3.7 – Logotipo del estándar HTML 5.**

Esta diseñado para su fácil ampliación de acuerdo a las necesidades del usuario por la gran cantidad de plug-in que existen y su interfaz avanzada de guionado (scripting extensiones) facilita la automatización de tareas monótonas. Es una aplicación de Software Libre cubierta por la Licencia Pública General GNU o GPL, brinda a los usuarios la libertad de acceder y modificar el código.

Para más información visita la siguiente página Web:

<http://www.gimp.org/>

<http://www.gimp.org.es/>

Algunas de sus características:

- Manipulación fotográfica.
- Composición de imágenes.
- Creación de imágenes.
- Gran variedad de pinceles.
- Uso de plug-ins.
- Sistema en línea de proceso por lotes.
- Generador de imágenes para producción en masa.
- Conversor de formatos de imagen.

Algunas de sus herramientas:

- Pincel.
  - Lápiz.
  - Aerógrafo.
  - Clonado.
  - Gestión de la memoria de mosaico.
  - Muestreo de sub-píxel.
  - Canal alfa.
  - Capas y canales.
-

Los formatos que soportados:

- GIF.
- JPEG.
- PNG.
- XPM.
- TIFF.
- TGA.
- MPEG.
- PS.
- PDF.
- PCX.
- BMP.

### 3.6- Desarrollo Web

El desarrollo Web es una actividad fuente de empleo como muchas otras que surgió con el Internet y es un reflejo del mundo actual que busca comunicación constante. Un proyecto puede ser desde individual hasta grupal, requiere de mucho trabajo en conjunto con sus miembros lo que rompe con la idea popular de que solo se necesita un hombre detrás de una máquina el que hace todo el trabajo.

Algunos aspectos importantes en el diseño Web:

- **Presentación:** El usuario no confíara y atención en una página descuidada, al existir tal cantidad de opciones el preferirá pasar a la siguientes página.
- **Identidad:** Al pertenecer a una corporación debe respetar la identidad de la misma, usando aspectos que la identifiquen como sus colores y logo.
- **SEO:** La optimización de motores de búsqueda o SEO (Search Engine Optimization), habla de la posición dada a nuestra página en los principales motores de búsqueda como Google y Yahoo. Mejor posición equivale a más visitas, logrando su objetivo de comunicar. El mejor método es tener una página bien estructuradas y que respeta los estándares.

La creación de una página Web no es diferente a otros proyectos, existen pasos recomendados para tener buenos resultados y siempre mantener una documentación de los procesos y pasos realizados, para poder regresar y hacer correcciones en caso de ser necesario. Los pasos aunque recomendados no son forzoso ya que lo más importante es adecuarse a las necesidades del proyecto. Los siguientes pasos a seguir son más de carácter cíclico que una lineal.

---

**1.- Diseño:** Se crea un boceto, al desarrollar una página Web sin tener en cuenta su presentación crea muchos conflictos. De este diseño se espera:

- Objetivos y expectativas de la página Web.
- Estudio inicial de necesidades y requisitos.
- Recolección de información.
- Selección de información.
- Medir los alcances del proyecto.

**2.- Estructura y programación:** Conocer que componentes (programación o didácticos) serán necesarios y la relación jerárquica entre ellos.

**3.- Pruebas:** Uno debe asegurarse del correcto funcionamiento del proyecto, existen errores que pueden existir desde el diseño y que solo son detectables con varias pruebas.

- Pruebas finales.
- Planificación del mantenimiento de la página Web.
- Planificación del seguimiento de resultados previsto.

Estos pasos pueden ser cíclicos, porque al descubrir un error uno tendrá que regresar al inicio del mismo para corregir errores.

### 3.6.1- HTML

HTML desarrollado por Tim Berners-Lee en 1980, gana popularidad al ser utilizado por navegadores Web como Mosaic de Marc Andreessen. Su facilidad de uso y buenos resultados en el diseño de páginas Web lo convirtieron en un estándar para los desarrolladores de navegadores. Su popularidad entre 1993 y 1995 causó la aparición de las versiones HTML2.0 y HTML3.0, se realizan varios cambios para enriquecer las posibilidades del lenguaje. En 1997, cuando el grupo de trabajo del W3C responsable de los estándares Web apoya el desarrollo de la versión 3.2, toma su lugar como principal lenguaje de desarrollo Web.

HTML es una aplicación de SGML (Standard Generalized Markup Language). Es un lenguaje de marcado de hipertexto HTML (HyperText Markup Language), un documento HTML es básicamente un documento de texto plano con etiquetas de marcado (tags) que definen como se desplegará el contenido en pantalla, ejemplo:

```
<etiqueta> contenido </etiqueta>
```

Las etiquetas no son sensibles a mayúsculas y minúsculas (case sensitive). Su flexibilidad es una de sus características más apreciadas, aunque un documento estuviera mal redactado (uso incorrecto de las reglas) podría ser desplegado en pantalla, esto permite

---

a los usuarios inexpertos obtener resultados a pesar de los errores, el problema es que dio paso a muchas malas técnicas de desarrollo.

Su desventaja son las limitadas posibilidades de diseño, a lo que surgieron otras herramientas como las hojas de estilo en cascada CSS (Cascading Style Sheets), cuyo enfoque es la presentación del documento ante el usuario final. Los desarrolladores usan la política de separar presentación de contenido.

La estructura del documento es básicamente dos secciones, la cabecera (head) y el cuerpo del documento (body) por ejemplo :

```
<HTML>
  <HEAD>
    <TITLE>Título de la página Web</TITLE>
  </HEAD>
  <BODY>
    Texto del documento
  </BODY>
</HTML>
```

Algunas de las opciones de CSS para la presentación de contenido son:

- Párrafos.
- Saltos de línea.
- Bloques citados.
- Divisiones.
- Texto formateado.
- Centrado.
- Imágenes.
- Tablas.

Después de HTML 4 surgió el Lenguaje Extensible para el Formato de Documentos de Hipertexto XHTML, que buscaba una estructura más estricta para los documentos e implementar el uso de la tecnología XML. Actualmente se retomó el uso del HTML y surgió HTML 5, buscando documentos mejor estructurados y flexibles para el despliegue. Actualmente muchos aspectos de él siguen en prueba, pero lo básico ya ha sido implementado por la mayoría de los navegadores.

### **3.6.2- XHTML**

XML es un método de escritura de documentos con el uso de etiquetas para darle a todo el texto un valor semántico, al tener un valor pueden ser fácilmente manipulados por otros programas. La principal ventaja es la gran variedad de tecnologías que trabajan

---

con estos documentos. Es una forma limitada de SGML como lo es HTML, busca heredar todas sus características sin su complejidad.

Un ejemplo:

```
<?xml version="1.0" encoding="UTF-8" ?>
<receta>
  <autor>Edgar </autor>
  <titulo>Papas en 10 min</titulo>
  <ingredientes>papas</ ingredientes >
  <ingredientes>agua</ ingredientes >
  <ingredientes>Sal</ ingredientes >
  <fecha>22/02/97 </fecha>
</receta >
```

Algunas de sus características son:

- **Compatibilidad:** El gran número herramientas surgiendo continuamente que trabajan con XML.
- **XSLT:** Para la transformación de documentos XForms.
- **Validación:** la posibilidad de validar tus documentos, verificando una estructura correcta.
- **DOM:** Pueden trabajar con el Modelo del Objeto Documento de HTML o XML (DOM).

### 3.6.3- HTML 5

HTML 5 no es un lenguaje nuevo, agrego algunas características que buscan hacer del diseño más semántico con menos código, así que los viejos usuarios no tendrán que preocuparse por la migración. Se a mantenido la compatibilidad con versiones anteriores, la páginas Web actuales solo requieren agregar las nuevas reglas para funcionar como HTML 5.Las nuevas etiquetas para la creación de secciones comunes como encabezado, pie de página, barra lateral, menús, etcétera. Otros elementos y atributos fueron modificados para mejorar su eficacia. Por ejemplo, elementos <B> que fue abandonado en versiones anteriores de HTML, cambia su rol para dar más significado a los usuarios.

---





**Imagen 3.8– Logotipo del estándar HTML 5.**

El método de desarrollo fue cambiado para las tecnologías HTML5 y CSS3. Se dividió en módulos para que los desarrolladores la implementaran gradualmente y no tuvieran que esperar a que estuviera terminada. Por ejemplo bloque de diseño contiene propiedad Background, propiedad de borde, propiedades de texto y así sucesivamente. La desventaja de este método es que los problemas de compatibilidad son más comunes.

Algunas novedades:

- **Canvas:** Busca ofrecer las funciones de flash usando las especificaciones HTML y sin instalar plug-in. Puede ser comparado con un pizarrón sobre el que se escribe usando JavaScript.
  - **Nuevas etiquetas:** Etiquetas enfocadas a material cotidiano para los usuarios como: video, pie de página, canvas, audio entre otros.
  - **Viejas etiquetas:** Recupera etiquetas que fueron abandonadas en versiones anteriores como `<B>`, son retomadas en nuevos contextos para mejorar su funcionalidad.
  - **Web Workers:** Son procesos que se realizaran en segundo plano por necesitar mucho tiempo de procesamiento, permite al usuario interactuar con la página sin necesidad de esperar a que terminen.
  - **Geolocalización:** Las páginas Web podrán ser localizadas por medio de la geolocalización.
-

- **Páginas Web Offline:** Las páginas Web podrán funcionar sin necesidad de conexión, incluso las aplicaciones.
- **Uso de Bases de datos:** Parecido a las cookies pero enfocado en cantidades mas grandes de información, permitiendo que las aplicaciones Web funcionen offline.

### 3.6.4- AJAX

Al cargar una página Web se descarga del servidor junto con todos los demás archivos que necesitará para su correcto funcionamiento, lo cual puede tardar mucho tiempo y molestar a los usuario. Ajax nos permite hacer llamados al servidor y traer solo la información que necesitamos en ese momento, logrando la carga de una página Web por etapas, así el usuario podrá interactuar con ella desde el principio sin tener que esperar la carga total. Eliminar las molestas esperas y la necesidad de recargar toda la página Web para traer solo un poco de información, manteniendo la atención del usuario en todo momento.

JavaScript asíncrono y XML conocido como Ajax (Asynchronous JavaScript and XML), no es una tecnología nueva, sino la unión de varias tecnologías viejas para crear un nuevo método dinámico y eficiente de comunicación de una página Web con el servidor. Uno de los mejores ejemplos de esta tecnología es Googlemaps, que la usa para que la carga de información e imágenes sea invisible para el usuario.

El nuevo modelo de aplicaciones Web, rompe con el modelo de aplicaciones de escritorio al no necesitar de su instalación para funcionar y en muchos casos al igualar sus propiedades. El modelo funciona al responder a las acciones del usuario con llamados (requerimiento) HTTP al servidor Web, el servidor efectúa un procesos de recopilación de información, cálculos números o comunicación con otras aplicaciones del servidor que luego devuelve al cliente. Este es un adaptado del modelo original de la Web sobre manejo híper texto con el fin de ampliar sus posibilidades.

Ajax trabaja como una capa la interacción entre el cliente y el servidor, la comunicación puede ser de forma sincronía o asíncrona, normalmente es asíncrona por su eficiencia. En el modo síncrono el cliente manda un llamado (petición) al servidor y los procesos se detienen hasta recibir una respuesta, en el modo asíncrono no se espera la respuesta, se puede seguir trabajando exista respuesta o no.

### 3.6.5- CSS

HTML es una lenguaje para el despliegue de páginas Web, pero debido a sus limitaciones para el diseño resulta insuficiente. Las presentación de una páginas Web es muy importante, por lo que otras herramientas como CSS fueron creadas enfocadas únicamente en el aspecto visual del documento, logrando separar el contenido de la

---

presentación. La separación del contenido y la presentación fue un gran avance que permitió programación más limpia, anteriormente un buen diseño necesitaba de agregar más código como las tablas que dificultaban la lectura y corrección de errores.

Las hojas de estilo en cascada o CSS (Cascading Style Sheets), describen el aspecto de un documento HTML en algún medio, por ejemplo pantalla, impresión, dispositivo móvil entre otros. Su estructura consiste en sentencias (el diseño) que afectan a uno o más elementos (sobre el cual recae el diseño) en el documento, se le llama en cascada por la preferencia que se le da al orden de las sentencias, siempre tendrá preferencia la última sentencia. Las sentencias tienen dos partes: el selector y la declaración, a su vez está compuesta por una propiedad y el valor que se le asigne. Al igual que el HTML a seguido un proceso de evolución, actualmente existe una versión CSS 3.

```
p{
color: blue;
}
```

Esta sentencia indica que todos los elementos **p** (párrafo) tendrán color azul.

**p**: el selector que designa que elemento será afectado.

**color: blue;** : es la declaración que puede contener una o más propiedades especificando el diseño a ser aplicado.

Los estilos pueden ser indicados de varias formas:

1.- Un documento externo.

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

2.- En varios documentos externos.

```
<link rel="stylesheet" type="text/css" href="mystyle-1.css">
```

```
<link rel="stylesheet" type="text/css" href="mystyle-2.css">
```

```
<link rel="stylesheet" type="text/css" href="mystyle-3.css">
```

3.- En los mismos elementos.

```
<p style="color:sienna;margin-left:20px">This is a paragraph.</p>
```

4.- Incrustarlas directamente en el documento.

```
<head>
<style type="text/css">
  h1 {
    font-family: Helvetica, Geneva, Arial, sans-serif;
  }
</style>
</head>
```

Aunque es un gran avance tiene sus problemas, como la poca congruencia entre los navegadores. No todos los navegadores la soportan del mismo modo, variaciones en el diseño de cada uno provocan resultados distintos, que un programador tenga que probar su diseño en varios exploradores con distintas versiones para asegurar su terminado. El nivel de desarrollo, no todos los exploradores están actualizados con las nuevas normas, aunque uno quisiera implementar los diseños más modernos, tiene que pensar en los usuarios con exploradores viejos o no actualizados, la preferencia del programador debe ser con la accesibilidad de su trabajo. Esto quiere decir que debemos usar las nuevas tecnologías con moderación.

### 3.6.6- Firebug

Firebug es una herramienta para el desarrollo Web, aunque es común que los navegadores contengan herramientas para el diseño como es el caso de Google Chrome y Firefox. Sus ventajas como ser fácil de instalar, ligera y muy intuitiva lo hacen resaltar. Facilita el análisis y depuración de tu código HTML, CSS, JavaScript entre otros, pudiendo ver los cambios en el código desplegarse de inmediato sin tener que volver a cargar la página.



**Imagen 3.9 – Logotipo de Firebug.**

Su enfoque es para Firefox pero también existe una versión ligera para Internet Explore. Es un software libre y de código abierto distribuido bajo la licencia BSD (Berkeley Software Distribution), la ventaja que ofrece esta licencia ante otras licencias de software libre es que te permite incluir su código en software no libre.

Para más información visita la siguiente página Web:

<http://getfirebug.com/>

---

Algunas características:

- Modificación de estilos y diseño en tiempo real.
- Inspección y modificación del código HTML.
- Depurador de JavaScript.
- Analizar del rendimiento.
- La capacidad modificar y de extender Firebug para las necesidades del usuario.
- Versión lite Puede escrita en JavaScript.
- Mucha documentación, tutoriales, videos, wiki y libros, todos accesibles para los usuarios.

### **3.7- Programas manejadores de contenido**

El sistema de gestión de contenido CMS (Content Management System) empezó como un método para automatizar tareas para las empresas que tenían la necesidad de publicar constantemente contenido en Internet. Aunque el termino se usa ampliamente para describir muchas herramientas, básicamente es una herramienta que te permite desplegar y administrar contenido en páginas Web. Los CMS son muy fáciles de usar y son recomendados para aquellas empresas o personas sin conocimiento en computación que quieren resultados rápidos y profesionales en sus páginas Web. Por muy pequeña que sean las empresas necesitan de la administración de tareas importantes y repetitivas, ya que facilita el hacer cambios pequeños a nuestra página sin ningún problema.

Gracias a su fácil y rápida gestión, esta tecnología no esta limitada solo a los negocios, en la educación a distancia permite un nivel de comunicación entre los alumnos y docentes que no es posible en las aulas. Un ejemplo muy conocido de este sistema es la Wikipedia.

Ya que no existe un sistema para clasificarlos podemos hacer referencia al tipo de contenido que manejan:

- Organizaciones privadas.
  - Genéricos.
  - Galerías.
  - Publicaciones digitales.
  - Foros.
  - Blogs.
  - Wikis.
  - Comercio digital.
  - Educación a distancia.
  - Revistas digitales.
-

Algunos de los CMS gratuitos más conocidos son:

- TextPattern.
- Drupal.
- WordPress.
- Nucleus.
- Alfresco.
- Joomla.
- Plone.
- dotCMS.
- Mambo.
- Ez Publish.
- Diferior.
- Symphony.
- Xpressionengine.
- Frog CMS.

Algunas desventajas son la poca estandarización entre los distados administradores, lo que impide una fácil comunicación y su manejo. El objetivo actual es ser más interpolables evitando programación extra innecesaria.

### 3.7.1- Joomla

Joomla! es un Sistema Gestor de Contenido o CMS (Content Management System) para la administración y creación de páginas Web con calidad profesional, difundido por su fácil uso, permite a principiantes la creación de páginas y el manejo de bases de datos a usuarios con poco o mínima experiencia. Utiliza programación del lado del servidor con PHP y base de datos MySQL. Cuenta con una Licencia Pública General (GNU/GLP) es de uso gratuito y de código abierto.



Imagen 3.10– Logotipo del sistema de gestión de contenido Joomla!.

---

Con una fácil instalación por el asistente de instalación de Joomla!, un grupo de desarrollo que lo mantiene actualizado y la posibilidad de elegir entre varios idiomas como español, le han dado un lugar entre los CMS más populares. Actualmente Joomla esta en la versión 3.0.2.

Para más información visita la siguiente página Web:

<http://www.joomla.org/>

Algunas nuevas características:

- Joomla cera completamente móvil.
  - Más adaptabilidad con nuevas herramientas.
  - Facilidad de uso.
  - Actualizaciones rápidas desde versionas anteriores.
  - Documentación abundante.
  - Generadores de formularios dinámicos.
  - Creador de directorios con enfoque a empresas u organizaciones.
  - Gestores de documentos.
  - Galerías de imágenes multimedia.
  - Motores de comercio y venta electrónica.
  - Software de foros y chats.
  - Calendarios.
  - Software para blogs.
  - Servicios de directorio.
  - Boletines de noticias.
  - Herramientas de registro de datos.
  - Sistemas de publicación de anuncios.
  - Servicios de suscripción.
-





# 4

# Desarrollo de Software

---

---



## 4.1- Diseño de una página Web

Para la creación de software existen muchos métodos de desarrollo con el objetivo de proporcionar bases que aseguren el éxito del proyecto. Se eligió el método de Desarrollo Rápido de Aplicaciones o RAD (Rapid Application Development) con base en nuestro objetivo general descrito anteriormente.

### 4.1.1- No me hagas pensar

“No Me Hagas Pensar” es el nombre de un libro perteneciente a Krug Steve, en el que se recomiendan características de diseño para la interfaz gráfica de usuario o GUI que faciliten la navegación de los usuarios por nuestra aplicación, algunas muy comunes y otras no esperadas. Una aplicación debe ser intuitiva y fácil de usar, asegurando cumplir con sus objetivos, para eso un usuario de experiencia mínima debe poder usarla sin ningún contratiempo.

Recomendaciones simples:

- **Fácil:** Una página Web debe ser fácil e intuitiva, los pasos a seguir para conseguir algo deben ser visibles, recuerda “no me hagas pensar”.
- **Rápido:** En el mundo del Internet los usuarios no quieren perder su tiempo, su atención es igual a confianza, un usuario que confía en nuestro trabajo volverá a nuestra página.
- **Ubicación:** Las páginas no siguen reglas geográficas, los usuarios deben recordar como llegaron desde otros puntos, tener un menú con los puntos principales de tu página Web ayudará a los usuarios a ubicarse y sin importar que siempre podrán encontrar lo que buscan.

### 4.1.2- Metodologías de desarrollo de software

En el pasado muchos proyectos de software fracasaban representando grandes pérdidas económicas, por lo que empezaron a surgir metodologías de desarrollo de software buscando dar base a los proyectos para lograr su objetivo, pero debido al gran número de requisitos de un software que pueden variar de un proyecto a otro surgió una gran variedad de metodologías. Para elegir la metodología más adecuada es recomendable basarse en su complejidad y el tiempo que requiere el proyecto.

---

Algunos métodos de desarrollo:

- Modelo incremental o evolutivo.
- Modelo orientado a objetos.
- Modelo entrega por etapas.
- Modelo en cascada.
- Modelo basado en prototipos.
- Modelo espiral.

### **4.1.3- Separación entre contenido y presentación**

La presentación es todo elemento que define el comportamiento visual de nuestro trabajo en algún medio, pueden ser desde CSS, etiquetas HTML u otro elemento externo como JavaScript. La Funcionalidad, son las características dinámicas que pueden responder a las acciones del usuario, como modificar contenido o cálculos. El contenido es básicamente texto plano que busca transmitir información que no altera la presentación ni la funcionalidad.

Al crecer nuestro código identificar la presentación, funcionalidad y contenido para luego seleccionarlo se vuelve vital, agilizando nuestro desarrollo, aunque es un concepto simple trae muchas ventajas:

- Permite distribuir la carga laboral.
- Evita que los problemas de una pieza afecten a las otras.
- Permite mantener un código limpio.
- Facilita su lectura.
- Simplifica la modificación y el mantenimiento.
- Reduce el tiempo de desarrollo.

### **4.1.4- Diferentes navegadores**

Todos los navegadores son diferentes con sus propias características y no siempre aplican los standards de la misma manera. Para cumplir el objetivo de llegar al mayor número de usuarios posibles es importante asegurar que nuestro diseño se mantenga al menos en los navegadores más populares, usar instrucciones estándar que funcionen en todos y no solo en uno. Las pruebas deben enfocarse no solo en la funcionalidad, si no también en el diseño.

---

## 4.2- Desarrollo rápido de aplicaciones (RAD)

El Desarrollo Rápido de Aplicaciones o RAD (Rapid Application Development), desarrollado inicialmente por James Martin en 1980, enfocado en el desarrollo rápido para proyectos cortos que se esperan realizar de 30 a 90 días. Su uso es recomendable cuando existen componentes (reutilizables) y se busca mantener la calidad a pesar del tiempo reducido. Trabajar con tiempos limitados requiere de seguir el calendario de trabajo estrictamente, sin realizar cambios drásticos por lo que se usan elementos como "Timeboxing" que se basa en la distribución de tiempos para tareas grandes y la eliminación de funcionalidades secundarias como sea necesario para cumplir con el calendario.

Características del método RAD:

- **Equipos Híbridos:** Los equipos no estarán limitados a una clase de elemento, se espera que en un equipo se encuentren todos los elementos posibles: diseñador, programador, "tester" (probador), el usuario final, el cliente, etcétera, principalmente el cliente para ayudar a enfocar el objetivo del proyecto.
- **Herramientas Especializadas:** Herramientas de Ingeniería de Software Asistida por Computadora o CASE (Computer Aided Software Engineering), son aquellas enfocadas en aumentar la productividad en los ciclos de desarrollo de software, reduciendo también tiempo y costo.
- **Timeboxing:** Método para distribuir actividades en lapsos de tiempo y dispensar de las funciones secundarias que podrían afectar al calendario de actividades.
- **Prototipos Iterativos:** Desde su primera etapa se crean modelos del proyecto que servirán como guía para su avance, el cual evolucionará hasta convertirse en el modelo terminado.

Ventajas:

- **Rápidos resultados:** Desde las primeras etapas es posible obtener un prototipo sobre el cual seguir trabajando.
  - **Flexibilidad:** Al tener equipos híbridos es posible hacer cambios rápidos al proyecto, evitando sorpresas a finales del proyecto.
  - **Usuarios:** Una de las principales características de esta metodología es el nivel de intervención por parte de los usuarios, considerados muy importantes en el diseño.
-

- **Recomendable para proyectos pequeños:** El uso de ciclos de desarrollo más pequeños favorece para localizar problemas.
- **Costos reducidos:** Al buscar máxima calidad en el menos tiempo logra la reducción en el costo del proyecto.

El desarrollo de software requiere de muchos diagramas, diagramas que representen el comportamiento correcto del programa y faciliten la comunicación entre los participantes, el uso de UML con este fin es muy recomendable. Lenguaje Unificado de Modelado conocido como UML o UML (Unified Modeling Language), aunque su creación es relativamente fácil y puede realizarse sobre el papel, el gran número de diagramas incluyendo sus modificaciones continuas nos obliga a usar herramientas que faciliten su creación.

Existe una gran cantidad de herramientas especializadas para la creación de UML en el mundo del software libre, para este proyecto se eligieron dos que pueden verse en la siguiente lista:

- **Dia:** Es una aplicación para crear diagramas técnicos. Su interfaz y funciones son similares al programa “Visio” de Windows. Algunas de sus características son la impresión de varias páginas, la exportación a muchos formatos (EPS, SVG, CGM y PNG) y la capacidad personalizar la herramienta. Algunos de sus diagramas que puede crear son diagramas UML, mapas de red, y diagramas de flujo. Con licencia GPL esta disponible para el uso público.

Para más información puedes visitar su página Web:

<http://dia-installer.de/>

- **ArgoUML:** Es una herramienta de modelado UML con soporte para todos los diagramas UML 1.4 estándar. Se ejecuta en cualquier plataforma Java y está disponible en diez idiomas. Su interfaz se divide en 4 paneles y soporta diagramas de clase, estado, casos de uso, actividad, colaboración, despliegue y de secuencia. Se distribuye bajo la licencia Eclipse Public License 1.0 (EPL).

Para más información puedes visitar su página Web:

<http://www.argouml.org>

Para medir el desarrollo del proyecto se usará un diagrama de Gantt, aplicando la misma ideología que en UML se usará el programa GanttProject. GanttProject es una herramienta multiplataforma para la gestión de proyectos por medio de un diagrama de Gantt. Se ejecuta en Windows, Linux y MacOSX, es una herramienta software libre.

---

---

Características:

- **Diagrama de Gantt:** Crear una estructura representante del diagrama de gantt.
- **Recursos:** Asignar los recursos humanos para trabajar en las tareas, ver su asignación en la tabla de carga de recursos.
- **Diagrama PERT:** Generar diagrama PERT de diagrama de Gantt.
- **Exportar:** Guardar gráficos como imágenes PNG, generar informes PDF y HTML.
- **Inter operar:** Importación de proyectos desde y exportarlos a formatos de Microsoft Project. Exportar a hoja de cálculo con CSV.
- **Colaborar:** Comparte proyectos con sus colegas mediante WebDAV.

Para más información puedes visitar su página Web:

<http://www.ganttproject.biz/>

#### 4.2.1- Etapas del ciclo RAD

La metodología RAD se divide en varios pasos, los cuales pueden o no ser cíclicos de según las necesidades del proyecto.

**Modelado de gestión:** Se plantean las bases del proyecto desde sus necesidades hasta su objetivo, en este punto los usuarios y el equipo híbrido son muy importantes para definir el proyecto. El flujo de información debe responder a las siguientes preguntas:

- ¿Qué información conduce el proceso?
- ¿Qué información genera?
- ¿Quién la genera?
- ¿A dónde va la información?
- ¿Quién la proceso?
- ¿Cuál es el objetivo del proyecto?
- ¿Quiénes interfieren?
- ¿A quienes esta dirigido?

**Modelado de datos:** El flujo de información establecido en la fase anterior “modelado de gestión” en esta fase se convertirá en un conjunto de “objetos de datos” ordenados y correctamente definidos. Se definen todas sus características (atributos) y la relación entre ellos.

**Modelado de procesos:** Los “objetos de datos” definidos en la fase anterior “modelado de datos” en este paso se definirá las relaciones entre ellos y las funciones del programa (procesos), representando el flujo de información y las transformaciones que sufre los objetos de datos. Al definir las funciones del software es importante marcar las modificaciones en los datos: resultado, añadir, modificar, suprimir o recuperar.

---

**Generación de aplicaciones:** En esta fase se promueve el uso de herramientas CASE (Ingeniería de Software Asistida por Computadora) con el objetivo de reducir costos y tiempo, a reutilizar componentes de programas ya existentes y a crear componentes reutilizables. Básicamente es el uso de herramientas y métodos que faciliten el desarrollo de software.

**Pruebas de entrega:** Basados en la fase anterior “generación de aplicaciones”, muchos de nuestro código y herramientas utilizadas no requieren de revisión, porque vienen de terceros, esto reduce el tiempo de pruebas. Para el proyecto lo más importante son las pruebas a nuestro código, individuales, pruebas de integridad del sistema, de diseño y que el proyecto cumpla con las expectativas del usuario.

#### **4.2.2- Calidad contra rapidez**

Es muy importante el cumplimiento del calendario, el cual es muy estricto, por lo que desde el principio se mantiene comunicación continua con el usuario (cliente) para evitar cambios de último momento que afecten los tiempos. Para asegurar el tiempo es importante definir el nivel de calidad (eficiencia) del proyecto enfocándose en la funcionalidad que necesita el cliente, un software con menos calidad no significa que contendrá errores, significa menos funcionalidades dependiendo del tiempo y el presupuesto. El "Timeboxing" es la elección de las funciones principales y la distribución de tareas demandantes en varios periodos de tiempo para asegurar el cumplimiento del horario, es importante recordar que este método siempre busca la máxima calidad con el menor tiempo.

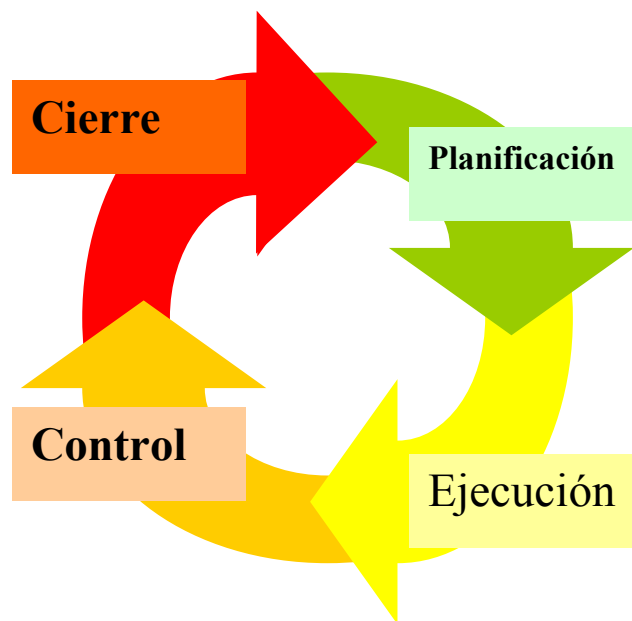
#### **4.3- Modelado de gestión**

En esta fase plantean las bases del proyecto desde sus necesidades hasta su objetivo, en este punto los usuarios y el equipo híbrido son muy importantes para definir el proyecto.

Dentro de cada fase existen procesos los cuales requieren de su propia administración, por lo cual se muestra el ciclo de “Administración de procesos”, existe una gran variedad de pasos que se pueden agregar al ciclo para su administración correcta de un proceso pero este tiene los pasos básicos recomendados.

---





**Imagen 4.1-** Administración de procesos: Muestra los pasos básicos y recomendables para la administración internas de los procesos en un proyecto.

### **Bases**

Son factores que sirven como bases o limitantes a considerar al momento de enfocar un proyecto:

- Existen 3 grados de estudio: primero, segundo y tercero.
- Existen 4 salones llamados grupos por cada grado: A, B, C y D.
- Un salón laboratorio.
- Un salón de informática.
- Material audio visual que consiste en un proyector y una computadora.
- El material audiovisual existe actualmente en los salones de laboratorio e informática.
- Los horarios se visualizan usando como referencia salón o profesor.
- Los días de clases son 5, de lunes a viernes.
- El turno es matutino, de 07:00 - 14:00.
- El horario de clases está dividido en 8 secciones de 50 minutos.
- Un sección de receso, 20 minutos de 10:20 a 10:40.

### Esquema de clases

Hora

07:00 - 07:50 --- sección 1

07:50 - 08:40 --- sección 2

08:40 - 09:30 --- sección 3

09:30 - 10:20 --- sección 4

10:20 – 10:40 --- receso

10:40 - 11:30 --- sección 5

11:30 - 12:20 --- sección 6

12:20 - 13:10 --- sección 7

13:10 – 14:00 --- sección 8

### Flujo de información

El flujo de información entre los integrantes del equipo híbrido ayudará a responder las siguientes preguntas:

1. ¿Cuál es el objetivo del proyecto?

Crear una herramienta para facilitar la administración de materias.

2. ¿Qué se espera del proyecto?

Facilitar la administración de:

- Materias.
- Horarios.
- Salones.
- Multimedia.
- Necesidades de los docentes.
- La comunicación entre servicios escolares y los docentes.

3. ¿Quiénes lo manejarán?

Servicios escolares, docentes y los administradores del sitio.

4. ¿Cuál será su uso?

La administración de salones y un medio de comunicación entre los profesores y la administración con el fin de recibir comentarios sobre las actividades administrativas.

5. ¿Quién es el usuario final?

Los docentes y servicios escolares.

---

6. ¿Desde dónde será usada?

Puede ser usado remotamente o en el mismo lugar de trabajo.

7. ¿Qué tecnologías interfieren?

- Diseño, CSS.
- Presentación, Web HTML.
- Comportamiento dinámico, JavaScript.
- Programación del lado del servidor, PHP.
- Base de datos para la información, MySQL.
- Un servidor que puede ser interno o externo.

8. ¿Quién se beneficia?

Los docentes y servicios escolares, al facilitar la comunicación entre los dos. Mejorar la selección de salones para los grupos y la repartición dependiendo de las necesidades de los profesores.

9. ¿Qué información se genera?

- Administración de salones.
- Administración de comentarios por parte de los docentes.

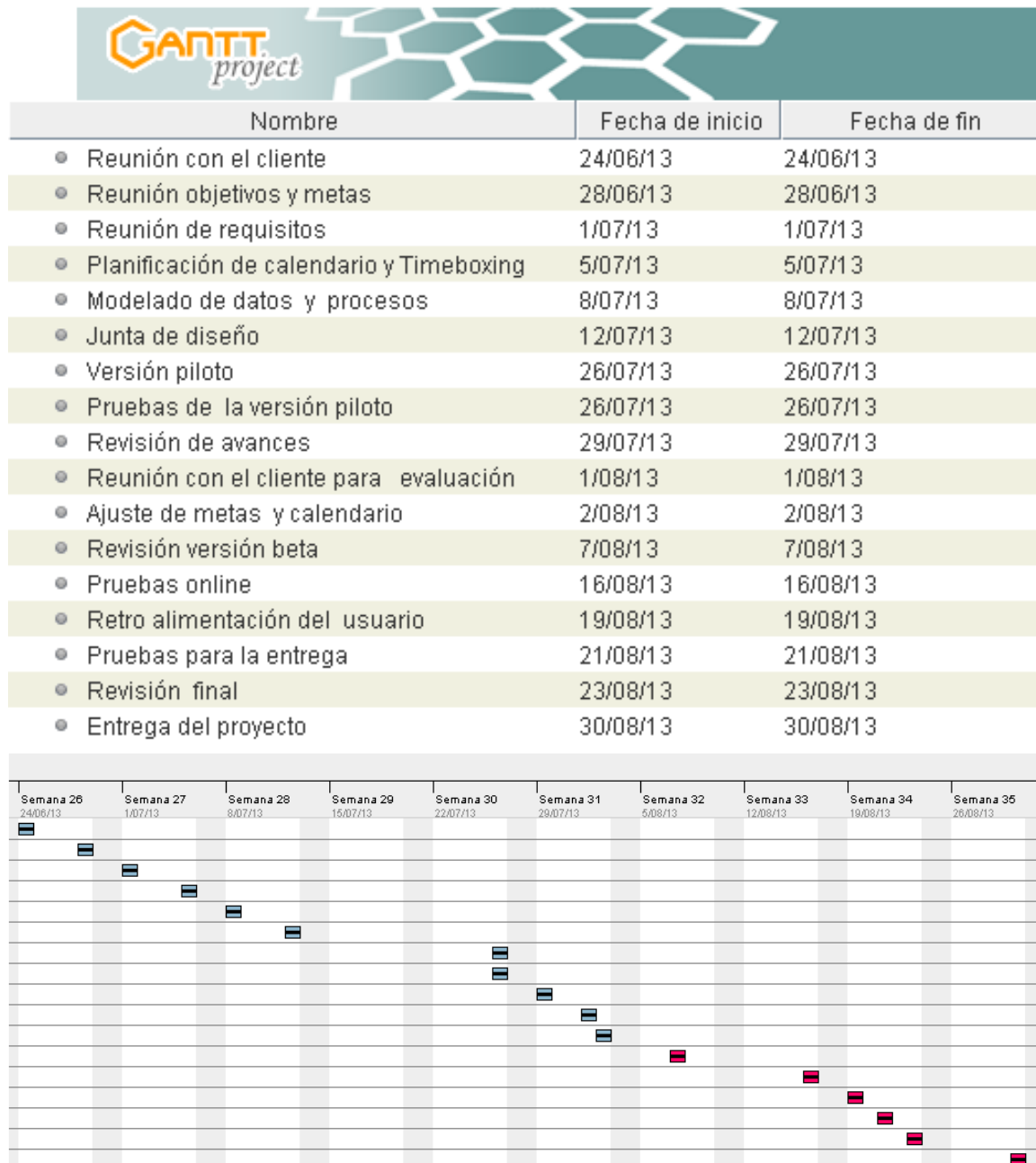
10. ¿A dónde va la información?

A una base de datos, desarrollada para este propósito.

Una vez establecidos estos parámetros debemos aplicar el Timeboxing para la creación del diagrama de Gantt (actividades) con el cual monitorear los avances en las actividades. El diagrama de Gantt es un popular método de administración gráfica de actividades relacionado con el tiempo del proyecto. Para el diagrama de Gantt se usará GanttProject, una herramienta software libre y multiplataforma para su diseño.

Al iniciar las actividades es importante realizar este diagrama para medir el tiempo estimado y el desarrollo del proyecto.

---



**Imagen 4.2 – Diagrama de Gantt inicial, con la repartición de tiempos para el proyecto a futuro.**

Se realizaron dos diagramas, uno a principios y otro a mediados porque sin importar las precauciones tomadas siempre existen altercados que afectan el tiempo y las actividades. Las actividades afectadas se muestran en rojo en ambos diagramas para su fácil comparación; en el segundo destacan los horarios recorridos.

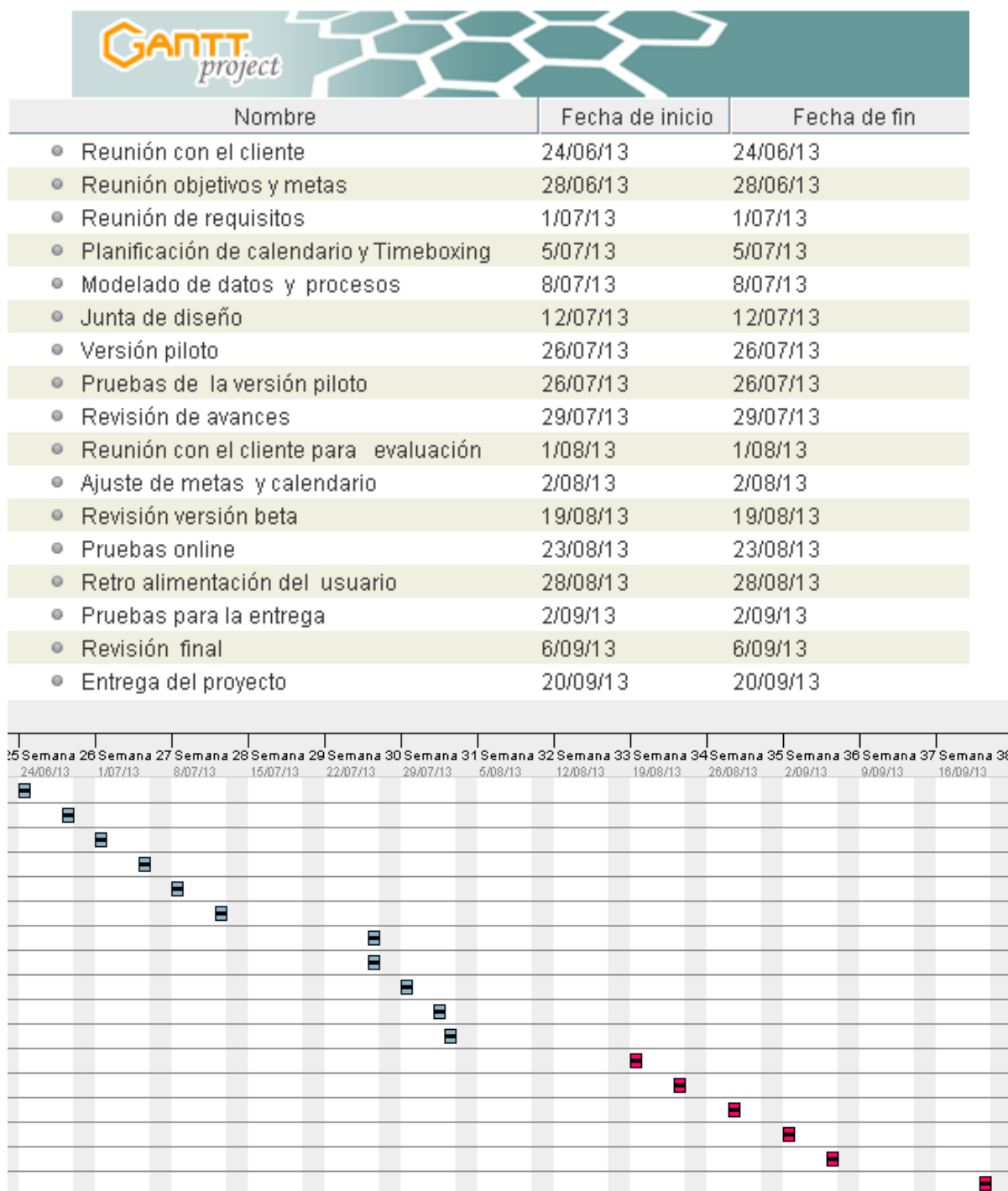


Imagen 4.3 – Diagrama de Gantt final, con las modificación en los tiempos después de la revisión de avances.

### 4.3.1- Planificación de los requisitos

Se describen las características que debe cumplir el desarrollo con el fin de garantizar la satisfacción del cliente en la entrega final. Los requisitos son una parte del trabajo que deben ponerse bien en claro antes de llegar a cualquier otro acuerdo. La redacción requiere una profunda reflexión sobre los objetivos que pretende alcanzar el proyecto.

Requisitos en el diseño:

- La página Web debe mantener una estructura correcta que permita su lectura en caso de no aplicar el diseño CSS.
- El contenido debe ser coherente y organizado.
- Un menú de navegación que permita navegar por los puntos principales debe estar presente en todo momento.
- Todo el contenido posible debe estar en español.
- El diseño debe ser el mismo en los principales navegadores en este caso Chrome, Firefox, Safari e Internet Explorer.
- La funcionalidad de la página debe mantenerse en los principales navegadores en este caso Chrome, Firefox, Safari e Internet Explorer.
- Debe mantener su funcionalidad por lo menos en Internet Explorer 9.
- Debe hacer uso de la tecnología HTML5.
- Botón de información sobre el manejo de la página Web disponible en todo momento.

Requisitos de funcionalidad:

- Posibilidad de agregar elementos multimedia posteriormente de ser necesario.
- Validación del código fuente HTML5, CSS3.
- La funcionalidad de los formularios en diferentes navegadores.

Requisitos en el software:

- El uso de lenguaje HTML5.
  - El uso del lenguaje JavaScript para procesos del lado del usuario.
  - El uso de hojas de estilo CSS3 para el diseño de la página.
  - El uso del lenguaje PHP para procesos de parte del servidor.
  - Uso de bases de datos con MySQL.
-

#### 4.3.2- Flujo de información entre las funciones

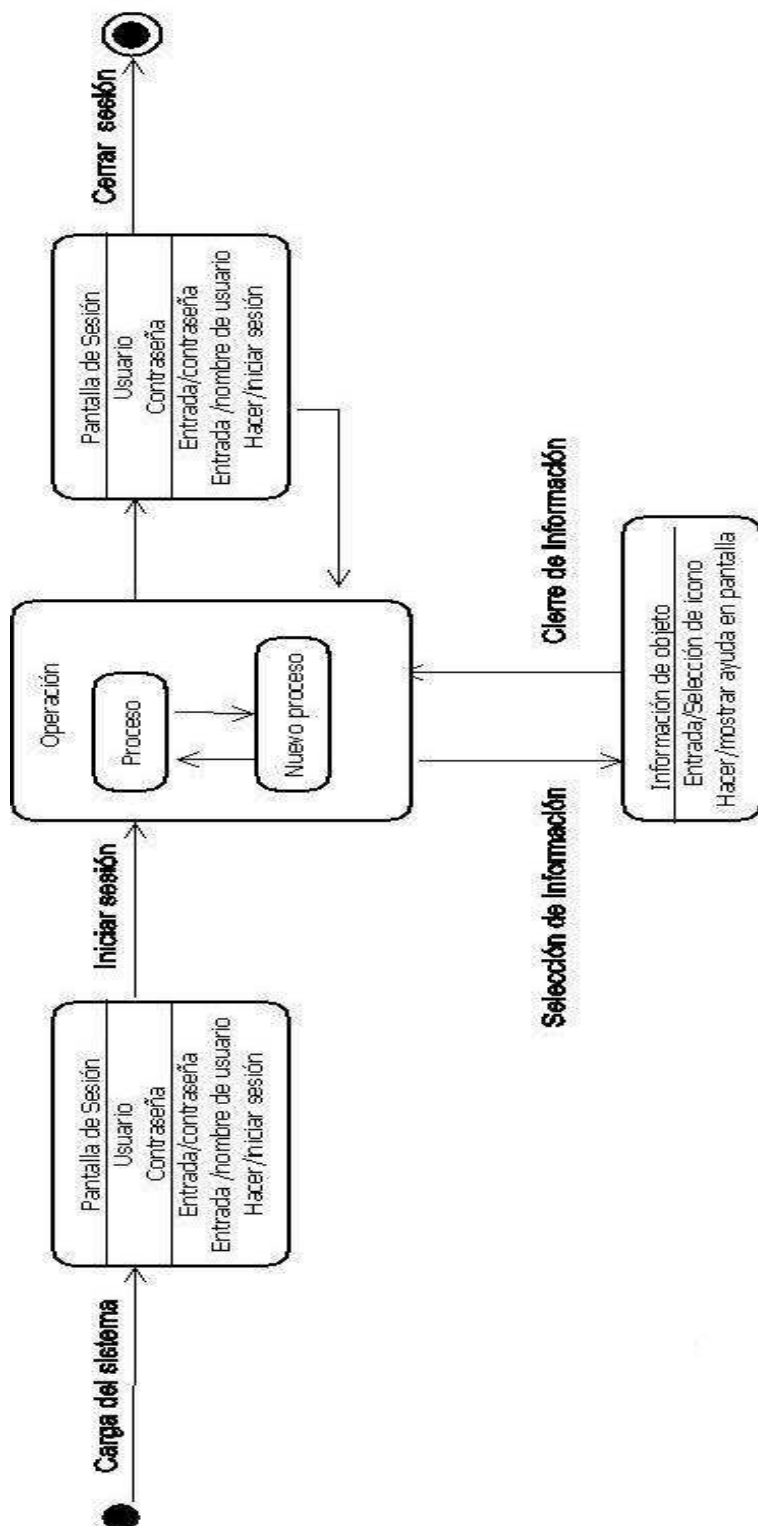


Imagen 4.4 – Diagramas de estados para nuestro sistema.

Los diagramas de estados también conocidos como motor de estados muestran el comportamiento de un sistema con respecto a los objetos que lo conforman, modificando su “estado” con respecto a sucesos y tiempo. Un diagrama de estados presenta los estados en los que puede encontrarse un objeto junto con las transiciones entre ellos; muestra el punto inicial y final de una secuencia de estados.

En el diagrama superior se observa como se siguen los estados del sistema desde la carga hasta el cierre donde el aspecto principal es el “proceso” que representa a una actividad que una vez realizada puede dar paso a otro proceso, el cual puede ser el mismo en repetición o uno distinto a partir del cual se puede pasar al cierre de sesión y el apagado.

Estos son los objetos representados por el objeto “proceso” en el diagrama superior.

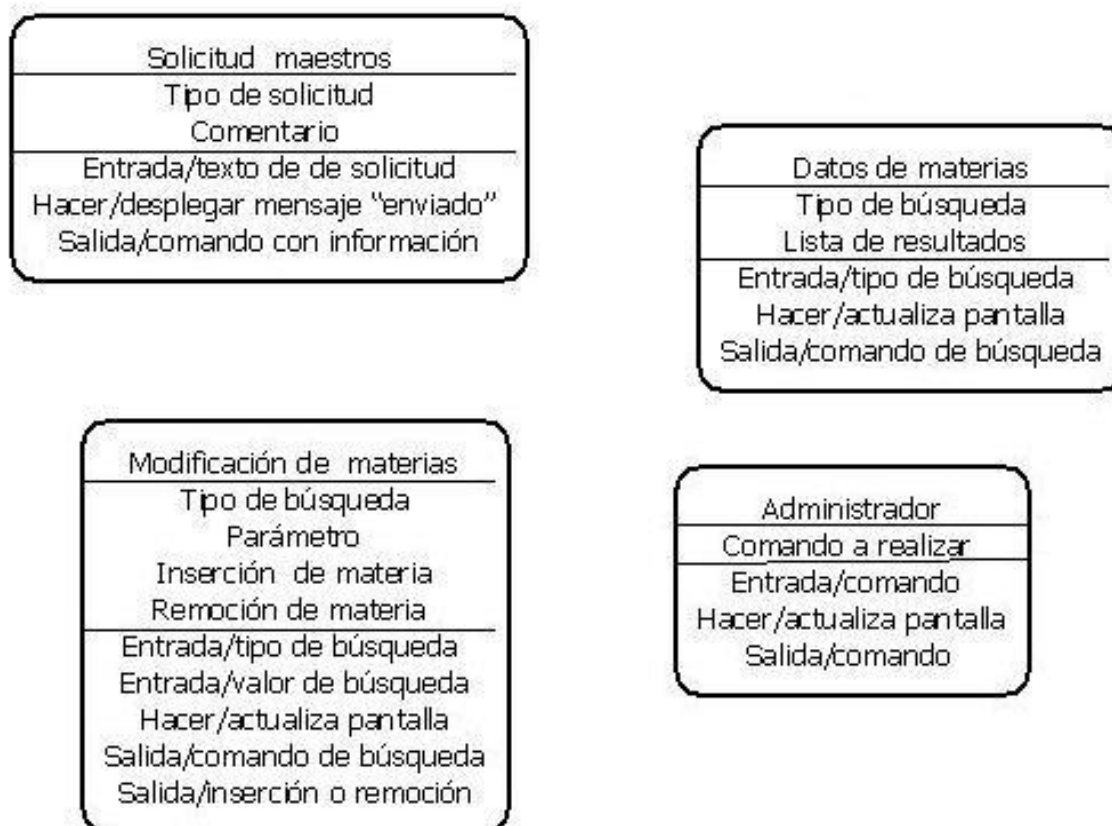


Imagen 4.5 – Objetos complementarios al diagrama de estados.



### 4.3.3- Front-end

El “front-end” es la parte del software que interactúa con el usuario. En esta etapa se desarrolla la interfaz gráfica de usuaria. Se creará un diseño base que será usado como referencia para luego crear un diseño específico para todos los demás casos de uso.

Primero se crea el diseño base para la selección de colores y ubicación de los objetos principales, el cual será usado como referencia para los demás diseños.



**Imagen 4.6 – Estructura global de nuestro documento.**

#### 4.3.4- Diseño de interfaz

##### Menú principal



Imagen 4.7- Estructura para la página Web de inicio.

##### Alertas y descripción emergente



Imagen 4.8- Estructura para su comportamiento dinámico.

## Solicitud maestros



Imagen 4.9– Estructura para el manejo de solicitudes.

## Datos de materias versión de profesores

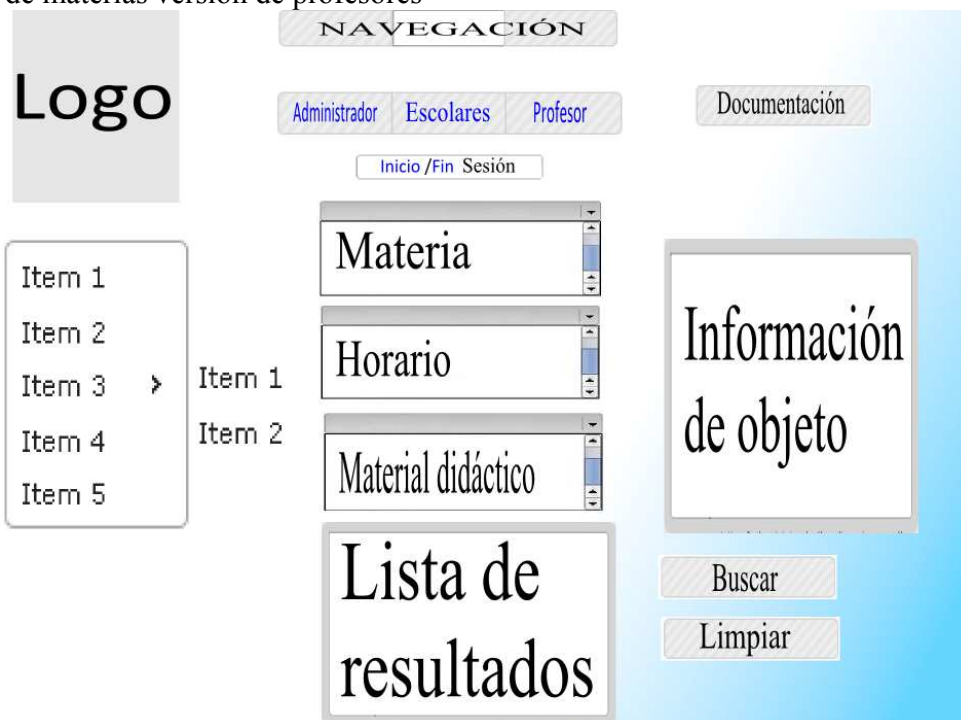


Imagen 4.10– Estructura para la búsqueda de horarios para docentes.

Datos de materias versión de servicios escolares



Imagen 4.11– Estructura para la búsqueda de horarios materias para escolares.

Administración de materias



Imagen 4.12– Estructura para modificar materias para escolares.

## Administrador



**Imagen 4.13– Estructura para el administrador de la base de datos.**

### Descripciones de pantalla:

- **Menú principal:** Es la pantalla de identificación de usuario. Entre sus características está el logo, botón de documentación con información relevante sobre su uso, dos cuadros de texto para introducir los datos del usuario, cuadro de navegación y de selección de tipo de usuario.
- **Alertas y descripción emergente:** Despliega las ventanas alerta y descripción emergente con el objetivo de mostrar información rápida y relevante para el usuario.
- **Solicitud maestros:** Presenta dos ventanas principales donde el usuario podrá elegir el tipo de comentario y un cuadro de texto donde ingresar su comentario.
- **Datos de materias versión de profesores:** Permitirá la búsqueda y su información. Se presentan tres ventanas de selección sobre materia, horario y material didáctico delimitando la búsqueda con una ventana donde se listaran los resultados. Con un botón de limpiar para facilitar una nueva búsqueda.

- **Datos de materias versión de servicios escolares:** Parecida a la versión de profesores, con diferencia de una ventana de información que muestra la solicitud del profesor para esa materia.
- **Administración de materias:** Enfocada para el uso de servicios escolares contiene una ventana de selección para elegir el tipo de búsqueda y otra para parámetros específicos; dos ventanas que muestran los resultados de una búsqueda con el propósito comparar resultados; botones de insertar o eliminar una materia elegida con la búsqueda.
- **Administrador:** Está enfocada en la administración de la base de datos por lo que posee dos ventanas principales, una para la inserción de comandos y otra para desplegar los resultados.

#### 4.4- Modelado de datos

Para continuar es importante definir los valores a utilizar y sus características: nombre, tipo, tamaño, contenido, valor predeterminado, posible nulo y comentario.

**tab\_dia:** Contiene la lista de días permitidos.

Campo	Tipo	Nulo	Predeterminado	Comentarios
id_dia	int	No	No	No
dia	varchar	No	No	No

Ejemplo.

id_dia	dia
1	lunes
2	martes

**tab\_horario:** Contiene los tipos de turno y sus horas respectivas.

Campo	Tipo	Nulo	Predeterminado	Comentarios
id_horario	int	No	No	No
turno	varchar	No	No	No
hora	varchar	No	No	No

Ejemplo.

id_horario	turno	hora
1	matutino	7:00-7:50
2	matutino	7:50-8:40

**tab\_materia:** Almacena todas las materias creadas, nombre y clave.

Campo	Tipo	Nulo	Predeterminado	Comentarios
id_materia	int	No	No	No
clave	int	No	No	No
nombre_materia	varchar	No	No	No

Ejemplo.

id_materia	clave	nombre_materia
1	1011	arte
2	2005	arte

**tab\_profesor:** Almacena los datos de los profesores que los identifican como usuarios.

Campo	Tipo	Nulo	Predeterminado	Comentarios
id_profesor	int	No	No	No
rfc	varchar	No	No	No
Contrasena	varchar	No	No	No
nombre_profesor	varchar	No	No	No
paterno	varchar	No	No	No
materno	varchar	No	No	No

Ejemplo.

id_profesor	rfc	Contrasena	nombre_profesor	paterno	materno
1	xxxxxxxxxxxxxx	contrasena	nombre	apellido	apellido
2	xxxxxxxxxxxxxx	contrasena	nombre	apellido	apellido

**tab\_salon:** Guarda las características de los salones.

Campo	Tipo	Nulo	Predeterminado	Comentarios
id_salon	int	No	No	No
ubicacion	varchar	No	No	No
material_didactico	varchar	No	No	No

Ejemplo.

id_salon	ubicacion	multimedia
1	1a	no
2	1b	no

Multimedia: apoyo visual en los cursos con un proyector.

**tab\_solicitud:** Las solicitudes echas por los usuarios junto con su clasificación para fácil manejo.

Campo	Tipo	Nulo	Predeterminado	Comentarios
id_profesor	int	No	No	No
tipo_solicitud	varchar	No	No	No
comentario	varchar	No	No	No

Ejemplo.

id_profesor	tipo_solicitud	comentario
1	administrativa	solicitud
2	docente	solicitud
3	general	solicitud

**tab\_curso:** Contiene las relaciones que conforman los cursos, desde el profesor hasta el grupo.

Campo	Tipo	Nulo	Predeterminado	Comentarios
id_curso	int	No	No	No
grupo	int	No	No	No
id_materia	int	No	No	No
id_profesor	int	No	No	No
id_salon	int	No	No	No

Ejemplo.

id_curso	grupo	id_materia	id_profesor	id_salon
1	1	1	1	1
2	2	2	2	2

**tab\_bd\_horario:** Permite la unión entre los horarios y los cursos.

Campo	Tipo	Nulo	Predeterminado	Comentarios
id_curso	int	No	No	No
id_dia	int	No	No	No
id_horario	int	No	No	No

Ejemplo.

id_curso	id_dia	id_horario
1	1	1
2	2	2



## 4.4.1- Flujo de información

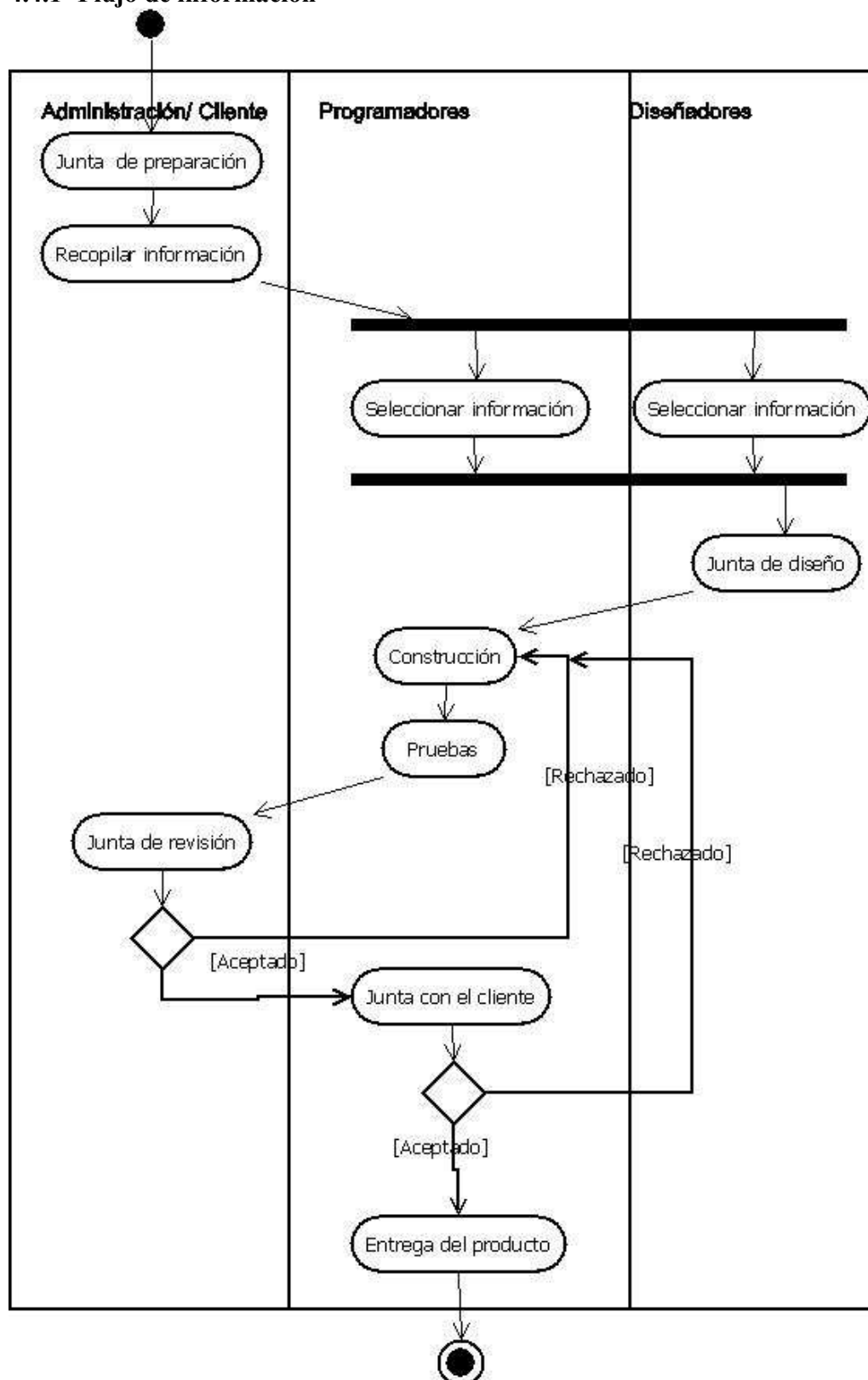


Imagen 4.14– Diagrama de flujo de información para los grupos de trabajo.

En un proyecto es importante saber cómo se comporta el flujo de información dentro de los grupos de trabajo, un buen método para lograrlo es crear un diagrama de actividades. Un diagrama de actividades muestra los pasos (actividades) de una operación o proceso de una manera simplificada. Una de sus características más útiles es la capacidad de expandirse mostrando quien tiene la responsabilidad por cada paso, esto se logra separando el diagrama en segmentos conocidos como marcos de responsabilidad que en su parte superior tienen marcado quien es el responsable.

El diagrama anterior muestra los pasos a seguir y el flujo de la información de los principales grupos de trabajo de un modo global. Empieza por la junta con el cliente hasta la entrega del mismo, se debe destacar como la junta de revisión y la junta con el cliente crean bucles que repiten el proceso de construcción con el fin de refinar el producto final.

#### 4.4.2- Análisis y diseño de datos

Existen cientos de datos que pueden ser agregados a un trabajo, siempre hay que elegir los relevantes, para eso nos basaremos en nuestros 3 “actores”.

Datos de actores:

- **Profesor:** Información de materia, horario, clave, ubicación, tipo de material didáctico, solicitud, nombre de usuario, tipo de usuario, clave de materia y profesor actual.
- **Administrador:** Información de la base de datos, claves, código fuente, contenido de la página, imágenes, nombre de usuario, lista de resultados y resultados de comandos.
- **Servicios escolares:** Horario, solicitudes, clave de la materia, ubicación, material didáctico, profesores, altas, bajas, espacios vacíos, nuevas materias, nuevos salones, nuevo material didáctico, nombre de usuario, tipo de usuario y cambios recientes.

#### 4.4.3- Datos de entrada

Separar los datos en valores de entrada y valores de salida nos ayuda a tener un panorama definido para cada una de las pantallas de nuestra interfaz gráfica, desde el punto de vista del usuario.

---

Descripciones de pantalla para valores de entrada:

- **Menú principal:** Contraseña y usuario.
- **Alertas y descripción emergente:** No hay datos de entrada para esta pantalla.
- **Solicitud maestros:** Tipo de solicitud, comentario.
- **Datos de materias versión de profesores:** Selección de materia, horario, material didáctico.
- **Datos de materias versión de servicios escolares:** Selección de materia, horario, material didáctico.
- **Administración de materias:** Tipo de búsqueda, parámetro.
- **Administrador:** Comando como texto.

#### 4.4.4- Datos de salida

En este caso definiremos los valores de salida para nuestra interfaz gráfica desde el punto de vista del usuario.

Descripciones de pantalla para valores de salida:

- **Menú principal:** Información de objetos, descripción emergente, documentación.
  - **Alertas y descripción emergente:** Descripción emergente y ventana de dialogo.
  - **Solicitud maestros:** Información de objetos, descripción emergente, y documentación.
  - **Datos de materias versión de profesores:** Información de objetos, descripción emergente, documentación y lista de resultados.
  - **Datos de materias versión de servicios escolares:** Descripción emergente, documentación, solicitud de profesor y lista de resultados.
  - **Administración de materias:** Descripción emergente, documentación, lista de resultados 1 y lista de resultados 2.
  - **Administrador:** Resultados.
-

#### 4.5- Modelado de procesos

##### Caso de uso profesor

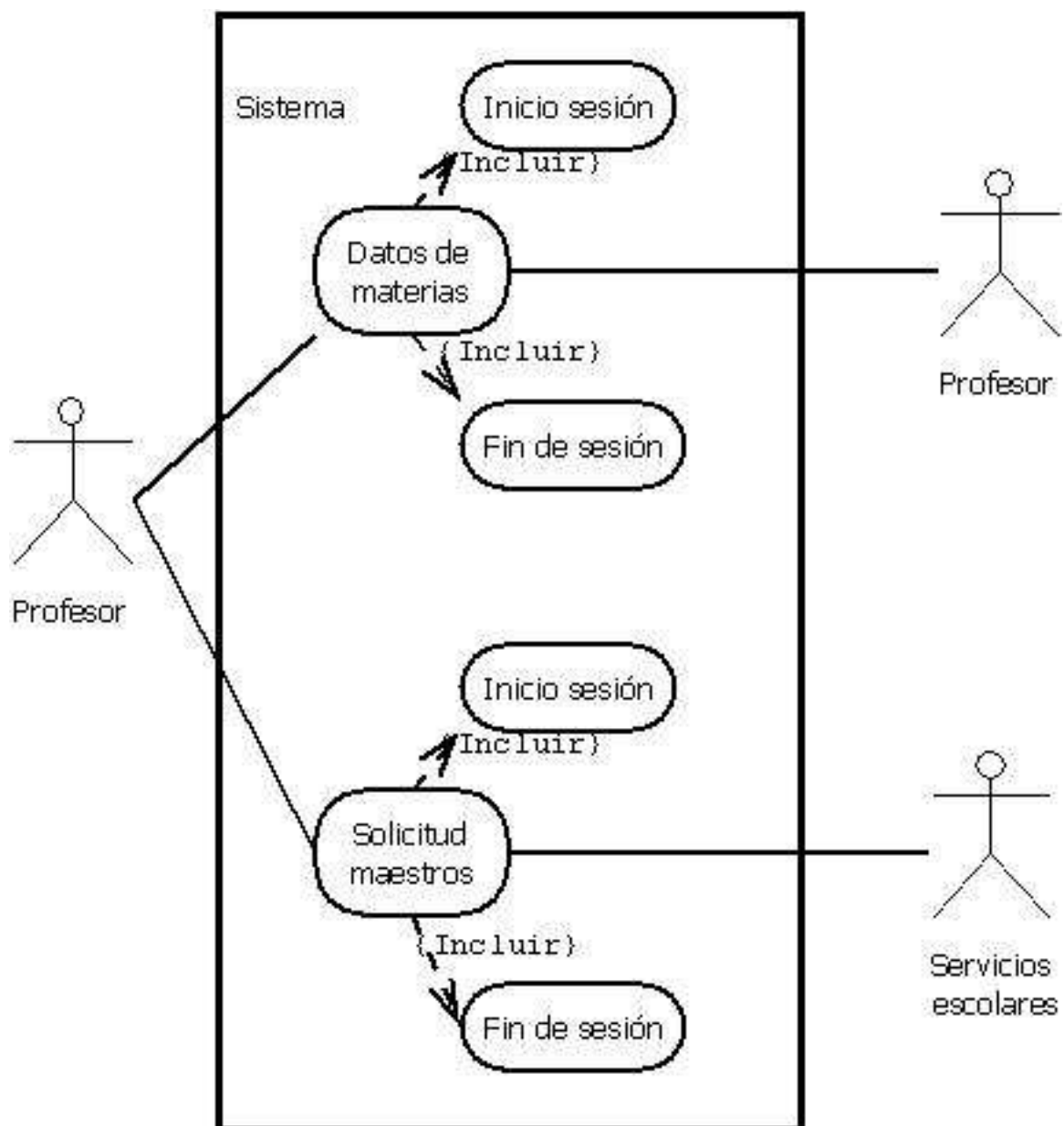
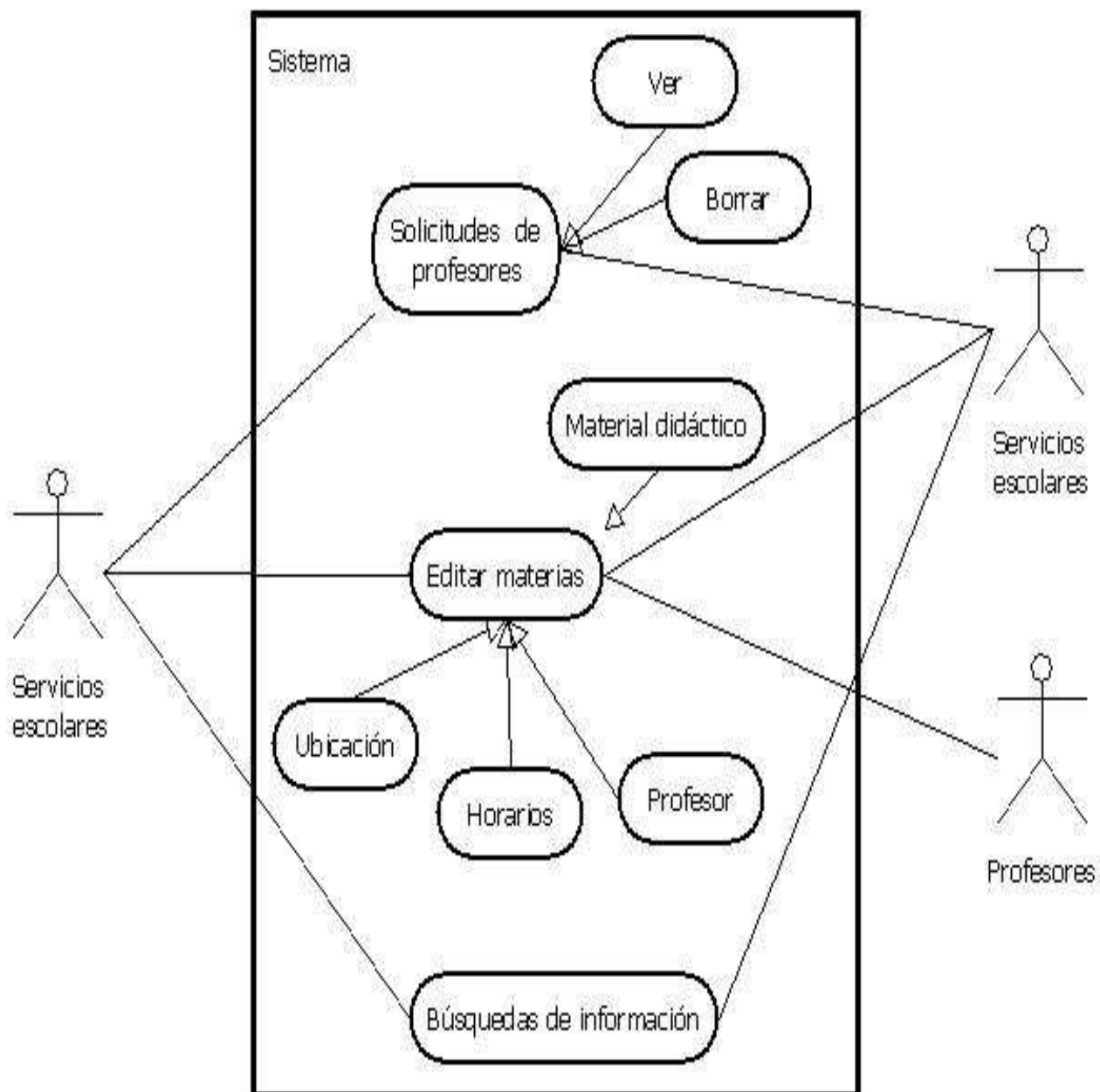


Imagen 4.15– Diagrama casos de uso para profesores, muestra su interacción con el sistema.

**Caso de uso servicios escolares**

**Imagen 4.16–** Diagrama casos de uso para servicios escolares, muestra su interacción con el sistema.

### Caso de uso administrador

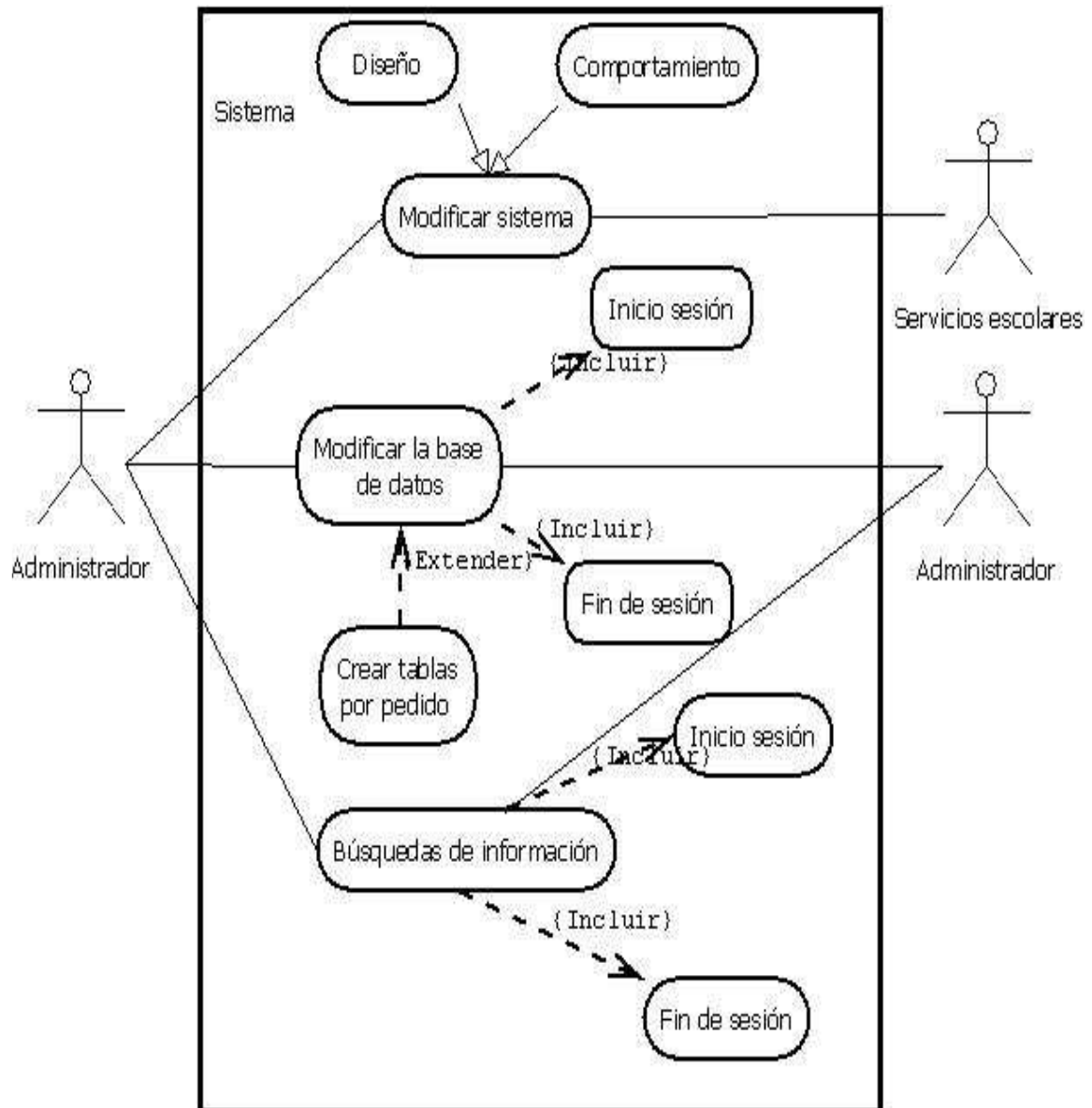


Imagen 4.17– Diagrama casos de uso para el administrador, muestra su interacción con el sistema.

La relación entre el usuario y la herramienta es muy importante, para eso se pueden usar los diagramas de casos de uso que muestran su interacción con el usuario. Esta clase de diagramas son buenos para promover la comunicación entre el cliente y el programador al explicar el funcionamiento de la herramienta de un modo simple y ameno que lo anime a dar su opinión sobre los resultados que espera. Las ventajas de este método son un producto de mejor calidad y ahorro de tiempo al evitar futuras juntas para corregir detalles causados por falta de comunicación.

Tomar en cuenta al usuario es algo que se puede lograr usando diagramas de casos de uso porque modela el sistema desde el punto de vista del usuario. Los casos de uso describen una colección de escenarios iniciados por una entidad llamada “actor”.

Nuestros casos de uso se dividen en tres actores potenciales:

- **Profesor:** Muestra los datos de materia y solicitud de maestro, los cuales llegan a los actores profesor y servicios escolares; con las características básicas de iniciar y cerrar sesión como inclusión.
- **Servicios escolares:** Muestra las solicitudes de los profesores, editar las materias y buscar información lo cual llega a servicios escolares y a los profesores.
- **Administrador:** Sus usos son más complejos ya que afectan la estructura de la herramienta, pueden modificar el diseño, la base de datos y la búsqueda de información. Los resultados de esto se reflejan en servicios escolares y en el mismo administrador.

#### 4.5.1- Objetos del modelo de datos

Se debe definir el comportamiento de nuestro programa, así poder separarlo en sus componentes básicos para evitar repetir actividades (objetos), evitando trabajo innecesario. Sus objetos deben ser definidos por su grado de complejidad en simples y complejos.

El objetivo de este método es la organización, agrupar objetos simples para crear objetos complejos. En más de una ocasión será necesario modificar el comportamiento de los objetos complejos, lo que se puede lograr modificando los objetos más simples, cambiando drásticamente el comportamiento de los objetos complejos sin necesidad de mucho trabajo o cambios drásticos.

---

## Objetos simples

Estos objetos son los más básicos y repetitivos, pueden ser de uso muy genérico o especializado a ciertas necesidades.

### Conexión

- **Variables de conexión:** Almacena las variables para lograr la conexión con la base de datos.
- **Crear conexión:** Conecta a PHP con la base de datos.
- **Cerrar conexión:** Termina la conexión de PHP con la base de datos.

### Sesión

- **Iniciar sesión:** Inicia una sesión de usuario para PHP.
- **Verificar sesión:** Verifica que actualmente exista una sesión de usuario para PHP.
- **Cerrar sesión:** Destruye una sesión de usuario para PHP.

### Base de datos

- **Búsqueda de datos:** Busca datos en una base de datos.
- **Búsqueda de datos en curso:** Busca datos especificados de un curso.
- **Insertar datos:** Inserta datos en una base de datos.
- **Insertar datos de curso:** Inserta datos específicos en un curso.
- **Borrar datos:** Borra datos seleccionados en una base de datos.
- **Actualizar datos:** Modifica datos ya existentes en una base de datos.

## Objetos complejos

Son objetos formados por uno o varios objetos simples y sus propias características distintivas.

**Crear comentario:** Recibe el comentario del usuario, verifica una existencia previa y actualiza o lo inserta.

```
Crear comentario{  
    Variables de conexión  
    Crear conexión  
    Verificar comentario{  
    Insertar datos  
    Cerrar conexión  
}
```

---



**Verificar comentario:** Recibe un comentario del usuario, busca un comentario existente y los une para luego realizar una actualización..

```
Verificar comentario{  
    Variables de conexión.  
    Crear conexión.  
    Búsqueda de datos  
    Actualizar datos.  
    Cerrar conexión.  
}
```

**Borrar comentario:** Utiliza los datos de usuario y su selección para borrar un comentario existente.

```
Borrar comentario{  
    Variables de conexión  
    Crear conexión  
    Búsqueda de datos  
    Borrar datos  
    Cerrar conexión  
}
```

**Crear curso:** Un método parecido a “Insertar datos” pero especializado para la tabla de curso facilitando su manejo.

```
Crear curso {  
    Variables de conexión  
    Crear conexión  
    Borrar datos  
    Insertar datos de curso  
    Cerrar conexión  
}
```

**Modificar curso:** Selecciona un curso en específico y actualiza los datos elegidos por el usuario.

```
Modificar curso{  
    Variables de conexión  
    Crear conexión  
    Búsqueda de datos  
    Actualizar datos  
    Cerrar conexión  
}
```

---

**Crear horario:** Método de “Insertar datos” especializado para insertar los horarios a cursos ya existentes. Busca los datos del curso elegido y agrega sus datos de horario.

```
Crear horario {  
    Variables de conexión  
    Crear conexión  
    Búsqueda de datos  
    Insertar datos  
    Cerrar conexión  
}
```

### Contenido dinámico

Muchas de las opciones y valores presentes en nuestra página Web dependen de los valores en la base de datos, para evitar actualizaciones constantes en el diseño HTML es necesario su creación dinámicamente.

**Crear tabla:** Inserta en el documento HTML una tabla, despliega los resultados de una búsqueda de materias como lista.

```
Crear tabla{  
    Variables de conexión  
    Crear conexión  
    Crear etiquetas HTML  
    Insertar datos de curso  
    Cerrar conexión  
}
```

**Crear tabla2:** Crea una tabla con los resultados de una búsqueda de materias como tira de materias (tradicional).

```
Crear tabla2{  
    Variables de conexión  
    Crear conexión  
    Cerrar conexión  
    Iniciar sesión  
    Verificar sesión  
    Cerrar sesión  
    Insertar datos  
    Borrar datos  
    Actualizar datos  
    Insertar datos de curso  
}
```

---

**Crear selectores:** Crea un selector con los valores existentes en la base de datos para una categoría elegida.

```
Crear selectores{  
    Variables de conexión  
    Crear conexión  
    Cerrar conexión  
    Iniciar sesión  
    Verificar sesión  
    Cerrar sesión  
    Insertar datos  
    Borrar datos  
    Actualizar datos  
    Insertar datos de curso  
}
```

#### 4.5.2- Back-end

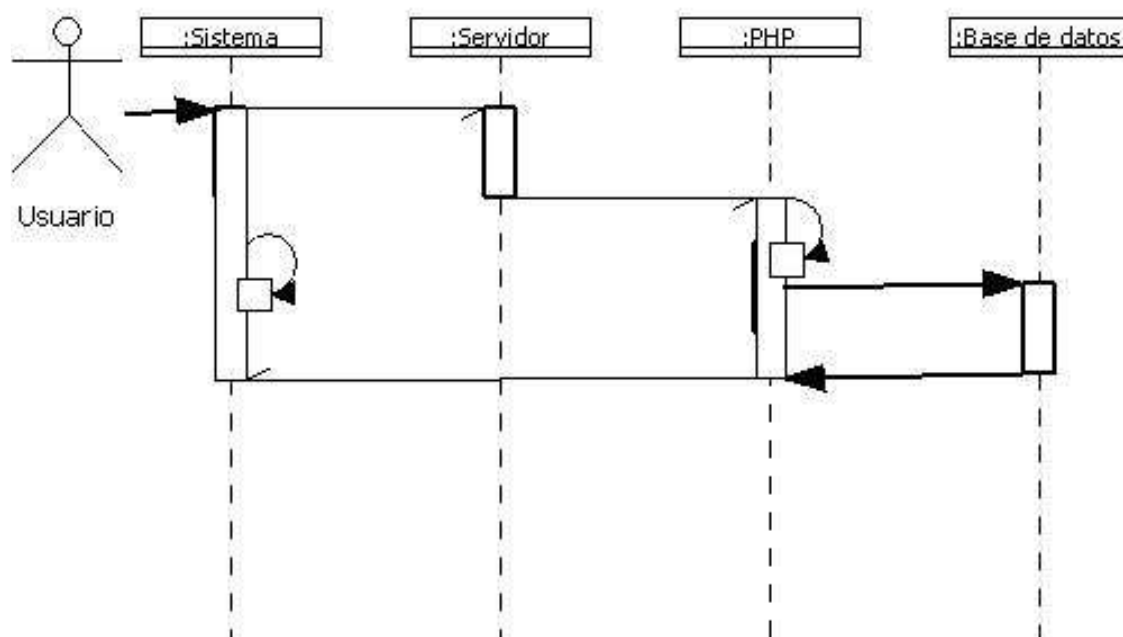


Imagen 4.18– Diagrama casos de secuencia, muestra los tiempos de interacción entre cada uno de nuestras herramientas.

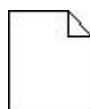
En este punto necesitamos una vista global del sistema con otros elementos, un buen modo es el diagrama de secuencia. El diagrama de secuencia muestra como los objetos se comunican entre sí al transcurrir el tiempo, en estos diagramas es importante entender que entre cada paso transcurre el tiempo. Su diseño consta de objetos representados por rectángulos con nombre (subrayado) y mensajes representados por líneas continuas con puntas de flecha y el tiempo representado por una progresión vertical.

En nuestro diagrama podemos ver un usuario mandando un mensaje al sistema luego al servidor, se comunica con PHP que se comunica con la base de datos retornando los resultado a PHP y regresando al sistema creando un ciclo dependiendo del caso. El comportamiento del sistema y PHP dependiendo del caso puede ser recursivo lo que es representado con una flecha apuntando a sí mismo con un cuadro.

### 4.5.3 –Construcción

#### La navegación

Un mapa de navegación representa los rumbos posibles a seguir para cada tipo de usuario. Los mapas ayudan a mantener la coherencia en nuestro contenido y darle preferencia a las secciones importantes, siempre hay que evitar que se formen caminos sin salida de los cuales no se pueda regresar a la página Web anterior o al índice.



**Imagen 4.19– Documento:** Este símbolo representa a cada documento HTML como un punto de elección para un nuevo rumbos de navegación en nuestras páginas Web.

#### Claves

**Índice:** Página que sirve como punto de partida para las demás.

**p-so-01:** Página Web donde los docentes pueden crear una solicitud.

**p-ho-02:** Página Web para ver en lista todas las materias asignadas para los docentes.

**p-bu-03:** Página Web para realizar búsquedas más detalladas de horarios o salones específicos del docente. .

---

**e-bx-01:** Servicios escolares tendrá acceso a la información de todos los cursos de un modo más selectivo.

**e-ms-02:** Servicios escolares podrá administrara todos los comentarios de los docentes.

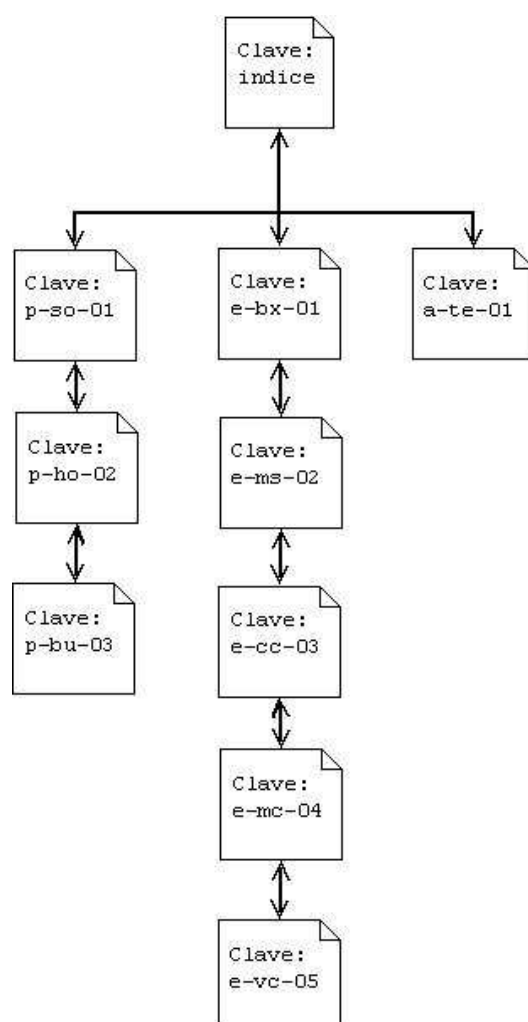
**e-cc-03:** Servicios escolares puede crear y eliminar cursos.

**e-mc-04:** Servicios escolares podrá modificar toda característica de los cursos.

**e-vc-05:** Servicios escolares tendrá una vista clásica de los horarios para un docente o un salón en específico.

**a-te-01:** Terminal para fácil inserción de comando para la base de datos.

### Estructura de la página Web



**Imagen 4.20–** Mapa de navegación, muestra de modo fácil y agradable los puntos de bifurcación para nuestra página Web.

### 4.6- Pruebas y entrega

Es importante las pruebas de nuestro producto asegurando su calidad, gracias al uso de herramientas CASE sólo son necesarias las pruebas de nuestro trabajo, sin preocuparnos en probar las herramientas de terceros que demostraron ya su funcionalidad.

La gran variedad de navegadores y sus diversas características incluso entre sus propias versiones implica que nuestro trabajo podría perder consistencia en alguno. Las pruebas visuales y funcionalidad en diferentes navegadores son básicas, pero hay que delimitar las opciones ya que existen muchas versiones y navegadores en desuso, para una mejor elección de pruebas es necesario investigar cuales son los más populares y sus versiones más recientes.

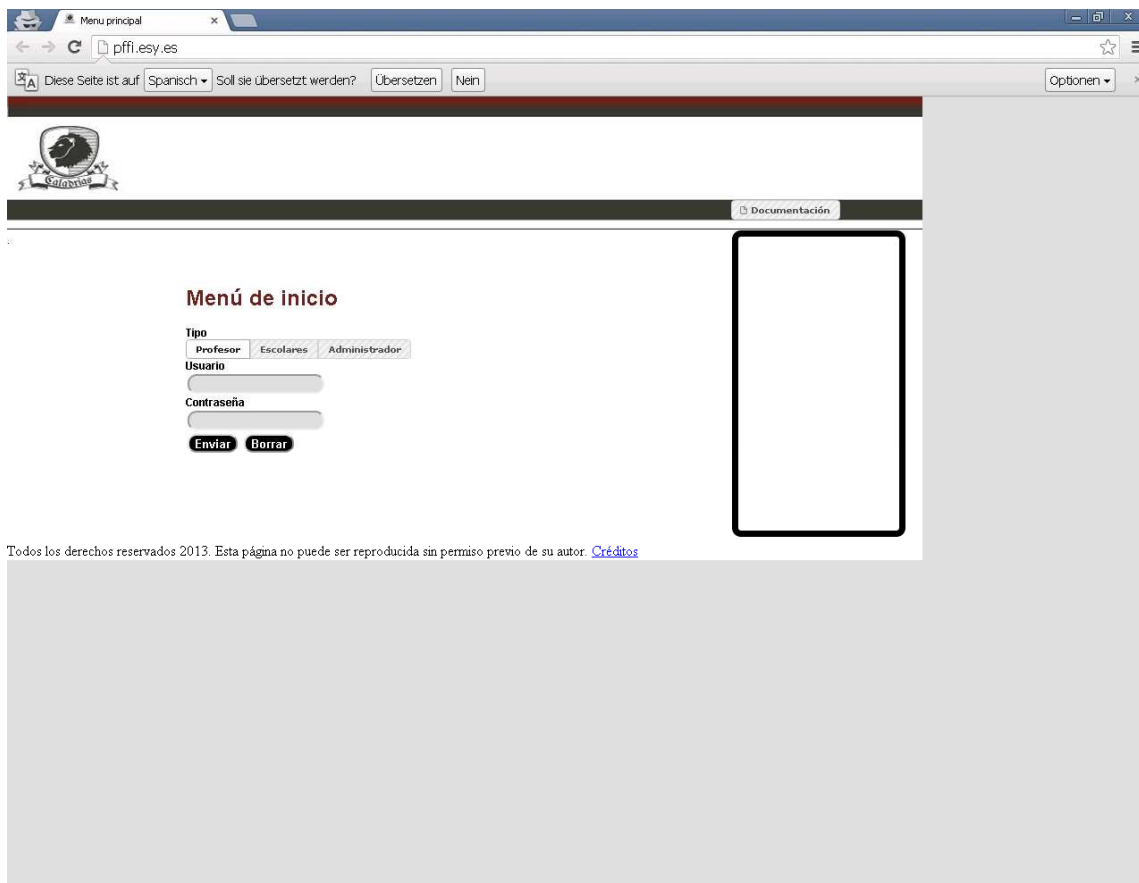
### **Herramientas de prueba**

Actual mente no existen muchas herramientas que te permitan automatizar este proceso y las que existen están limitadas en funcionalidad. IE collection es una aplicación gratuita para instalar varias versiones de Internet Explorer en un mismo sistema operativo, pudiendo cambiar entre una y otra con un sólo clic, para probar aspecto y funcionalidad de tu página Web.

---

#### 4.6.1- Pruebas en diversos exploradores

Prueba en el navegador Chrome.



**Imagen 4.21– Pantalla del aspecto de la página Web en el navegador Chrome.**

Chrome V 31.0	
Características	Cumple
Funcionalidad	Sí
Diseño	Sí

Chrome V 33.0	
Características	Cumple
Funcionalidad	Sí
Diseño	Sí

Prueba en el navegador Firefox.

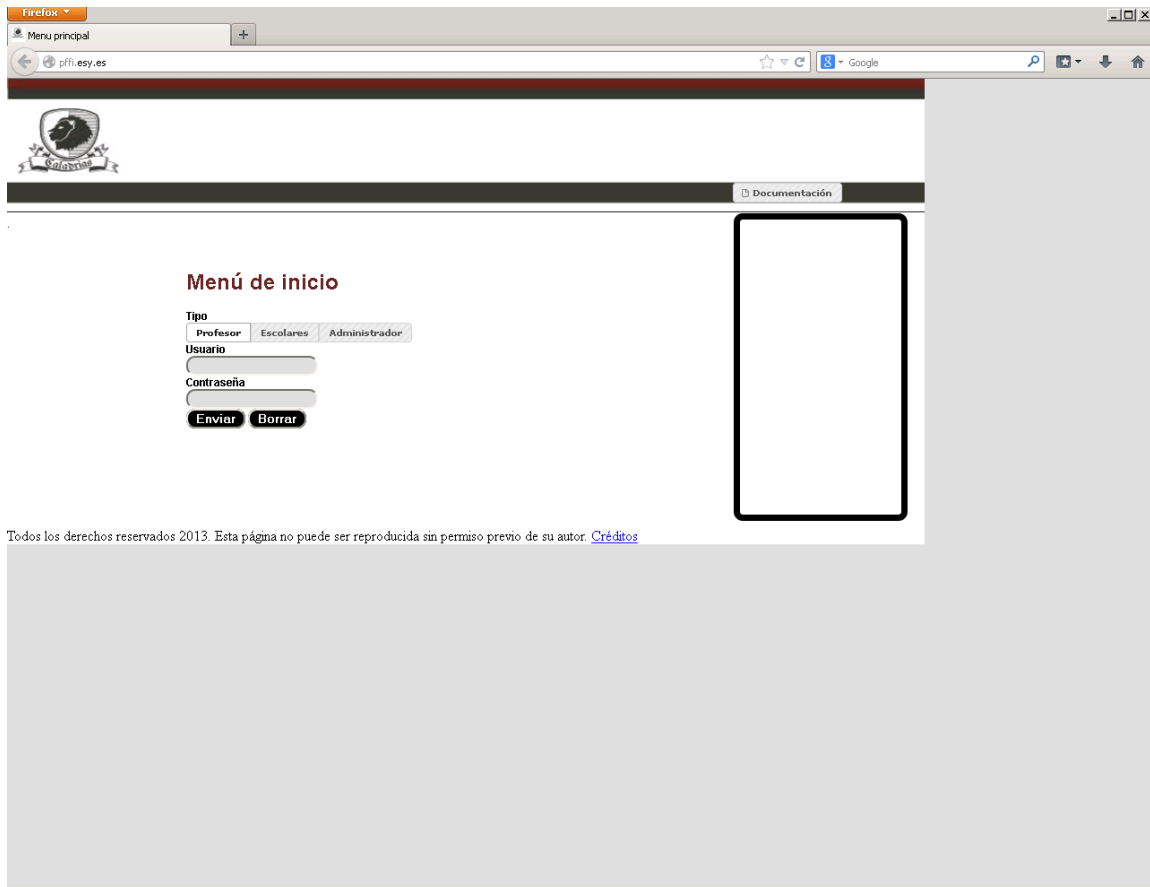


Imagen 4.22– Pantalla del aspecto de la página Web en el navegador Firefox.

Firefox V26	
Características	Cumple
Funcionalidad	Sí
Diseño	Sí

Firefox V27	
Características	Cumple
Funcionalidad	Sí
Diseño	Sí



Prueba en el navegador Opera.



**Imagen 4.23– Pantalla del aspecto de la página Web en el navegador Opera.**

Opera V11	
Características	Cumple
Funcionalidad	Sí
Diseño	Sí

Opera V12	
Características	Cumple
Funcionalidad	Sí
Diseño	Sí

Firefox V28	
Características	Cumple
Funcionalidad	Sí
Diseño	Sí

Prueba en el navegador SeaMonkey.



**Imagen 4.24–** Pantalla del aspecto de la página Web en el navegador SeaMonkey.

SeaMonkey V2.7.2	
Características	Cumple
Funcionalidad	Sí
Diseño	Sí

SeaMonkey V2.8	
Características	Cumple
Funcionalidad	Sí
Diseño	Sí

Prueba en el navegador Internet Explore.



**Imagen 4.25– Pantalla del aspecto de la página Web en el navegador Internet Explore.**

Internet Explore V7	
Características	Cumple
Funcionalidad	Sí
Diseño	Sí

Internet Explore V8	
Características	Cumple
Funcionalidad	Sí
Diseño	Sí

### 4.6.2- Entrega del programa

Una vez realizadas las pruebas se realiza una última reunión con el cliente para la revisión de resultados. La revisión de resultados busca el acuerdo de las partes sobre los resultados obtenidos y se procede a la instalación del sistema en su ubicación final, en caso contrario tomara una lista de los cambios necesarios y se realizara una segunda cita esperando un intervalo menor a una semana.

A continuación se muestran los resultados basados en las claves de navegación.

#### **Función**

**Índice:** Página que sirve como punto de partida para las demás.

Formulario que te permite acceder como usuario.

- 1.- Elige el tipo de usuario.
- 2.- Introduce tu nombre de usuario.
- 3.- Introduce tu contraseña.
- 4.- Presiona enviar.

Puedes borrar los datos elegidos con el botón borrar.

**p-so-01:** Página Web donde los docentes pueden crear una solicitud.

Formulario de solicitud, podrás comunicarte con los administradores y resolver dudas o necesidades con respecto a las materias o servicios. Elegir entre cargar una solicitud existente para ver su respuesta o crear una nueva.

Cargar:

- 1.- Elige el tipo de petición.
- 2.- Elige cargar.
- 3.- Presiona aceptar.

Crear:

- 1.- Elige el tipo de petición.
  - 2.- Elige crear.
  - 3.- Escribe tu petición.
  - 4.- Presiona aceptar.
-

**p-ho-02:** Página Web para ver en lista todas las materias asignadas para los docentes.

Tabla, muestra todos los resultados existentes de los horarios según las limitaciones indicadas por el usuario.

**p-bu-03:** Página Web para realizar búsquedas más detalladas de horarios o salones específicos del docente.

Formulario para delimitar los parámetros de búsqueda, logrando mejores reuntados.

- 1.- Elige los parámetros deseados, puedes dejar vacíos los parámetros de los que quieres el rango completo.
- 2.- Presiona buscar.

Los resultados serán visibles en la lista inferior. Puedes borrar lo elegido presionando el botón borrar.

**e-bx-01:** Servicios escolares tendrá acceso a la información de todos los cursos de un modo más selectivo.

Formulario para delimitar los parámetros de búsqueda, logrando mejores reuntados.

- 1.- Elige los parámetros deseados, puedes dejar vacíos los parámetros de los que quieres el rango completo.
- 2.- Presiona buscar.

Los resultados serán visibles en la lista inferior. Puedes borrar lo elegido presionando el botón borrar.

**e-ms-02:** Servicios escolares podrá administrara todos los comentarios de los docentes.

Formulario de solicitud, podrás comunicarte con los profesores y resolver dudas o necesidades con respecto a las materias o servicios. Puedes elegir entre:

\*Cargar: Recupera un comentario existente.

\*Modificar: Reemplaza una solicitud existente con el actual en el área de texto.

\*Crear: Crea un comentario con el tipo de solicitud y profesor que ayas elegido para que el profesor lo lea y responda a el.

\*Borrar: elimina una solicitud existente.

\*Ver todos: despliega en lista todas las solicitudes existentes facilitando la elección.

Los pasos para cada opción son parecidos:

- 1.- Elige el profesor.
  - 2.- Elige el tipo de petición.
  - 3.- Elige la acción a realiza.
-

- 4.- Modifica o crea el comentario en el área de texto.
- 5.- Presiona aceptar.

**e-cc-03:** Servicios escolares puede crear y eliminar cursos.

Formulario de curso con el cual podrás crear o eliminar cursos sin sus horarios, ellos pueden ser agregados posteriormente.

Crear:

- 1.- Elige los datos del curso empezando por el profesor.
- 2.- Presiona el botón CREAR.
- 3.- Debes verificar la conclusión del proceso en las listas inferiores. En materia original no debe aparecer nada y en materia modificada debe aparecer el curso creado con tus especificaciones.

Eliminar:

- 1.- Necesitas conocer el ID del curso a eliminar, puedes buscarlo en las secciones anteriores.
- 2.- Elige el ID del curso a eliminar.
- 3.- Presiona el botón ELIMINAR.
- 4.- Verifica tus resultados en las listas inferiores. En materia original debe aparecer el curso que elegiste y en materia modificada no debe aparecer nada, demostrando que el curso ya no existe.

Las listas inferiores muestran el antes y después del curso para verificar la conclusión del proceso.

**e-mc-04:** Servicios escolares podrá modificar toda característica de los cursos.

Formulario para modificar cursos, posee todas las características posibles de un curso. Solo puede modificar cursos ya existentes, no crearlos, eso se realiza en secciones anteriores.

Datos a modificar:

- 1.- Necesitas el ID del curso.
- 2.- Elige el ID del curso que deseas modificar.
- 3.- Elige los valores que quieres modificar del curso, los valores que dejes VACIO no serán modificados.
- 4.- Elige 1 nuevo horario en la sección de crear horario.
- 5.- Presiona MODIFICAR para aplicar los cambios.
- 6.- Verifica los cambios en las dos listas inferiores que te muestran el antes y después de la materia.
- 7.- Se desplegará una lista de los horarios existentes en la sección horario eliminar, elige alguno que desees eliminar y presiona modificar.
- 8.- Verifica el resultado en las listas de resultados, el antes y después.

No es necesario modificar todos los valores de un curso, puedes sólo agregar un nuevo horario, modificar el profesor o eliminar un horario a la vez.

---

Horario eliminar:

- 1.- Elige el ID del curso a modificar.
- 2.- Presiona VER HORARIOS.
- 3.- Se desplegaran en la sección horario eliminar.
- 4.- Elige uno.
- 5.- Presiona modificar.
- 6.- Verifica tu resultado en las lista inferiores del original y modificado.

**e-vc-05:** Servicios escolares tendrá una vista clásica de los horarios para un docente o un salón en específico.

Es un método de búsqueda restringido con el objetivo de mostrar horarios tradicionales para un profesor o salón en particular. Se mostraran en ventanas externas para abrir cuantos sean necesarios y ayudar a la selección más eficiente de nuevos horarios.

**a-te-01:** Terminal para fácil inserción de comandos para la base de datos.

Pequeño cuadro de dialogo donde podrás insertar comandos simples de SQL y ver sus resultados.

## Vista

### Indice.



The screenshot displays a web application interface. At the top left is a logo for 'Galaxia'. A navigation bar contains a 'Documentación' link. The main content area features a 'Menú de inicio' section with a 'Tipo' dropdown menu (options: Profesor, Escolares, Administrador), 'Usuario' and 'Contraseña' input fields, and 'Enviar' and 'Borrar' buttons. To the right, a 'Ayuda' box provides instructions: 'Formulario que te permite acceder como usuario.', a numbered list (1.-Elige el tipo de usuario, 2.-Introduce tu nombre de usuario, 3.-Introduce tu contraseña, 4.-Presiona enviar.), and a note: 'Puedes borrar los datos elegidos con el botón borrar.' At the bottom, a footer states: 'Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)'

## p-so-01.



[Cerrar Sesión CARLOS](#) [Documentación](#)

### Profesor solicitud

- Solicitud
- Horarios
- Búsqueda

Administrativa

**Acción**

Cargar
Crear

Ultimo proceso :-

**Ayuda**


Formulario de solicitud, podrás comunicarte con los administradores y resolver dudas o necesidades con respecto a las materias o servicios. Elegir entre cargar una solicitud existente para ver su respuesta o crear una nueva.

Cargar:  
1.- Elige el tipo de

Aceptar
Borrar

Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)

## p-ho-02.



[Cerrar Sesión CARLOS](#) [Documentación](#)

### Profesor horarios

- Solicitud
- Horarios
- Búsqueda

id	profesor	materia	clave	grupo	ubicación	hora	
10	Martinez Royo Carlos	educación física	1010	1	1a	miércoles 13:10-14:00	viernes 8:40-9:30
22	Martinez Royo Carlos	educación física	1010	2	1b	jueves 13:10-14:00	lunes 8:40-9:30
46	Martinez Royo Carlos	educación física	1010	4	1d	lunes 13:10-14:00	miércoles 8:40-9:30
52	Martinez Royo Carlos	educación física	2004	1	2a	jueves 7:50-8:40	lunes 9:30-10:20
76	Martinez Royo Carlos	educación física	2004	3	2c	lunes 7:50-8:40	miércoles 9:30-10:20
88	Martinez Royo Carlos	educación física	2004	4	2d	jueves 9:30-10:20	martes 7:50-8:40
114	Martinez Royo Carlos	educación física	3008	2	3b	miércoles 10:40-11:30	viernes 13:10-14:00
134	Martinez Royo Carlos	educación física	3008	4	3d	martes 13:10-14:00	viernes 10:40-11:30

**Ayuda**

Menú con el cual podrás desplazarte entre las funciones permitidas por la herramienta.

Imprimir

Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)



## p-bu-03.



Cerrar Sesión CARLOS Documentación

Solicitud

Horarios

Búsqueda

### Profesor búsqueda

Materia

Clave

Grupo

Salon

Día

Horario

**Buscar**

id profesor materia clave grupo ubicación hora

**Imprimir**

### Horario simple

Profesor

Salon

**Ver** **Borrar**

### Ayuda

Formulario para delimitar los parámetros de búsqueda, logrando mejores reuntados.


1.- Elige los parámetros deseados, puedes dejar vacíos los parámetros de los que quieres el rango completo.

2.- Presiona buscar.

Los resultados serán

Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)

## e-bx-01.



Cerrar Sesión ROOT1 Documentación

Búsqueda

Modificar solicitud

Crear curso

Modificar curso

Vista completa

### Escolares búsqueda

Profesor

Materia

Clave

Grupo

Salon

Día

Horario

**Buscar** **Borrar**

id profesor materia clave grupo ubicación hora

**Imprimir**

### Horario simple

Profesor

Salon

**Ver** **Borrar**

### Ayuda

Formulario para delimitar los parámetros de búsqueda, logrando mejores reuntados.

1.- Elige los parámetros deseados, puedes dejar vacíos los parámetros de los que quieres el rango completo.

2.- Presiona buscar.

Los resultados serán

Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)

## e-ms-02.



[Cerrar Sesión ROOT1](#) [Documentación](#)

### Escolares solicitud

? Búsqueda

- Modificar solicitud
- Crear curso
- Modificar curso
- Vista completa

**Profesor**

**Tipo**

**Acción**

Ultimo proceso: - -


**Ayuda**

Formulario de solicitud, podrás comunicarte con los profesores y resolver dudas o necesidades con respecto a las materias o servicios. Puedes elegir entre:

\*Cargar: Recupera un comentario existente.  
 \*Modificar: Reemplaza una solicitud existente con el actual en el área de texto.

Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)

## e-cc-03.



[Cerrar Sesión ROOT1](#) [Documentación](#)

? Búsqueda

- Modificar solicitud
- Crear curso
- Modificar curso
- Vista completa

### Escolares búsqueda

**Crear curso**

**Profesor**

**Materia**

**Grupo**

**Salon**

**Eliminar curso**

**Id del curso**

**Materia original**

**Materia modificada**


**Horario simple**

**Profesor**

**Salon**

Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)

## e-mc-04.



[Cerrar Sesión ROOT1](#) [Documentación](#)

### Escolares modificar curso

? Búsqueda

- ▣ Modificar solicitud
- Crear curso
- ⌘ Modificar curso
- 🖨 Vista completa

**Elige curso**

Id del curso

**Datos a modificar**

Profesor

Materia

Grupo

Salon

**Horario crear**

Día

Horario

**Horario a eliminar**

VACIO

**Materia original**

id profesor materia clave grupo ubicación día hora

**Materia modificada**

id profesor materia clave grupo ubicación día hora

#### Ayuda

Formulario para modificar cursos, pose todas las características posible de un curso. Solo puede modificar cursos ya existentes, no crearlos, eso se realiza en secciones anteriores.

Datos a modificar:

- 1.- Necesitas el ID del curso.
- 2.- Elige el ID del


Horario simple

Profesor

Salon

Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)

## e-vc-05.



[Cerrar Sesión ROOT1](#) [Documentación](#)

### Escolares búsqueda

? Búsqueda

- ▣ Modificar solicitud
- Crear curso
- ⌘ Modificar curso
- 🖨 Vista completa

Profesor

Salon

#### Ayuda

Es un método de búsqueda restringido con el objetivo de mostrar horarios tradicionales para un profesor o salón en particular. Se mostraran en ventanas externas para abrir cuantos sean necesarios y ayudar a la selección mas eficiente de nuevos horarios.

Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)

**a-te-01.**[Fin Sesión ROOT1](#)

Ver  Comando

COMANDO

Resultado

---RESULTADO

Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)

**4.7- Análisis de resultados****Cambios en el diseño final**

Un proyecto sin importar su nivel de plantación sufrirá cambios inesperados, desde el principio hasta el fin. El diseño fue cambiado por un alternativo, una versión más simplificada.

**Diseño final**

Imagen 4.26– Diseño alternativo para la página Web que se convirtió en el diseño final.

Una breve comparación entre el diseños inicial y el final, los cambios fueron solo estéticos respetando la funcionalidad de la página Web.



Imagen 4.27– Pantalla del índice con el diseño original.



Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)

Imagen 4.28– Pantalla del índice con el diseño alternativo que remplazo al original.



Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)

**Imagen 4.29– Comparación entre el diseño original y el final, respetando el objetivo y funcionalidad final.**



Imagen 4.30– Comparación, aunque la funcionalidad se respetó para todas las pantallas algunas tuvieron que ser modificadas por la gran cantidad de información que desplegaban.

Los cambios en el diseño fueron en general estéticos, en la funcionalidad aunque mínimos también hubo cambios, como se muestra en la imagen anterior se cambió el método para desplegar resultado debido a la cantidad de información. Se agregó la sección “horario simple” que muestra los resultados en una ventana aparte con el formato clásico de horarios limitado por salón y profesor; es un cambio en comportamiento muy importante, al crecer la cantidad se eligió darle su propia pantalla a los resultados para facilitar las comparaciones porque en más de una ocasión se tendrá 2 o 3 ventanas abiertas al mismo tiempo para facilitar las modificaciones. La siguiente imagen muestra este agregado.

## Escolares Horario

### salon - Martinez Royo Carlos

Horario	Lunes	Martes	Miércoles	Jueves	Viernes
7:00-7:50					
7:50-8:40					
8:40-9:30					Martinez Royo Carlos educación física <b>1010-1-1a</b>
9:30-10:20					
10:40-11:30					
11:30-12:20					
12:20-13:10					
13:10-14:00			Martinez Royo Carlos educación física <b>1010-1-1a</b>		

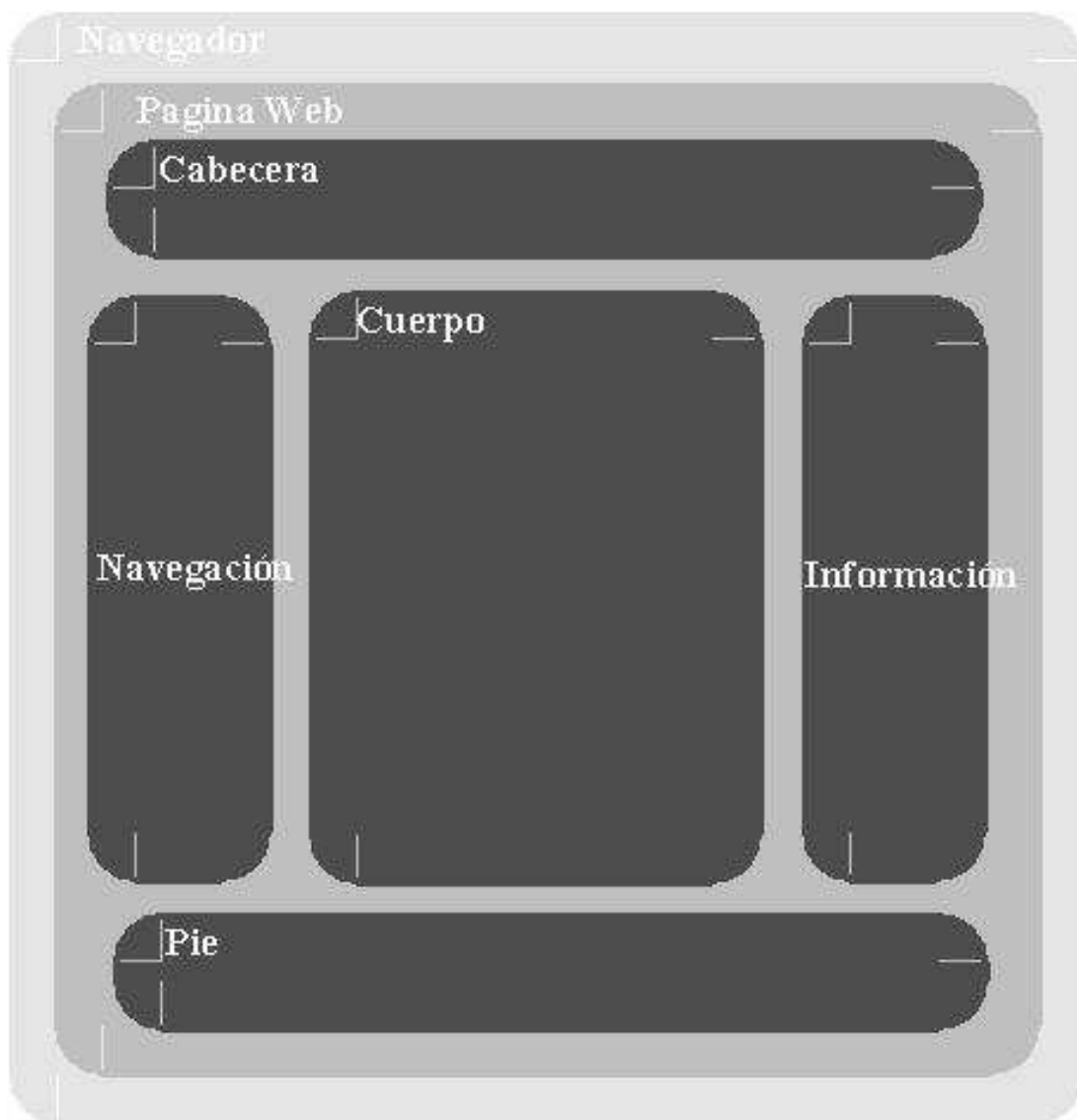
**Imprimir**

Todos los derechos reservados 2013. Esta página no puede ser reproducida sin permiso previo de su autor. [Créditos](#)

**Imagen 4.31– Pantalla agregada para desplegar los horarios en modo tradicional, por salones y profesor.**



Aunque el diseño fue modificado y se agregaron funciones, el formato para el trabajo fue consistente, formato que con el tiempo se a vuelto muy común en la mayoría de las páginas actuales y sobre todo es muy recomendable para páginas Web con HTML5.



**Imagen 4.32–Funcionalidad que fue respetada sin importar los cambios en el diseño.**

### 4.7.1- El progreso

#### Diagrama de Gantt

Al iniciar el proyecto se realizó un diagrama de Gantt, distribuyendo el tiempo esperado entre las principales actividades a realizar. Diagrama de Gantt inicial:

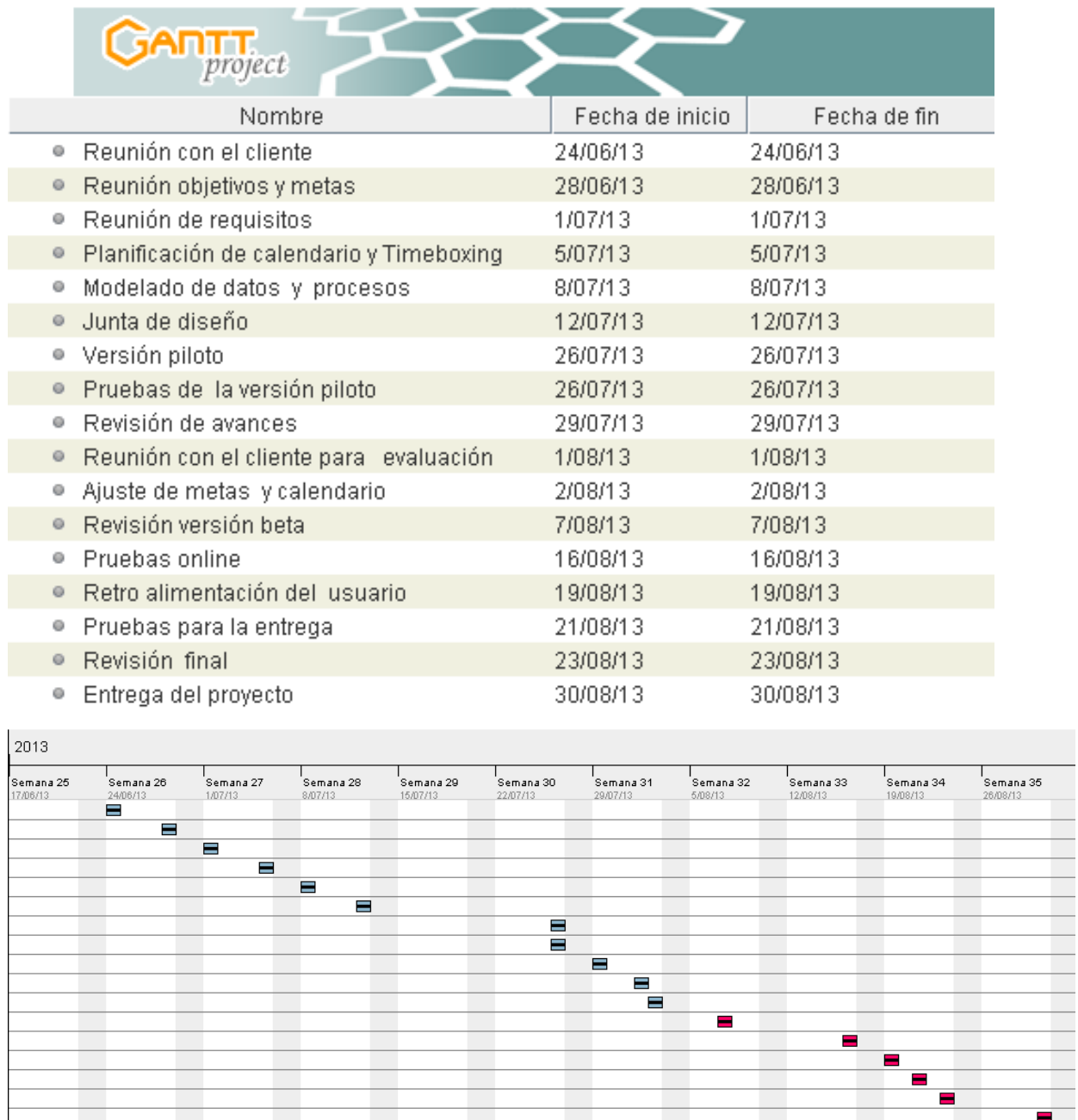


Imagen 4.33– Diagrama de Gantt inicial, con la repartición de tiempos para el proyecto a futuro.

Todo proyecto durante su desarrollo requiere ajustes de actividades lo que afecta al tiempo. En la reunión de “revisión avances“ se observó el camino recorrido con respecto al tiempo, basado en los resultados en la reunión “ajuste de metas y calendario” tubo que reajustarse los tiempos produciendo un nuevo diagrama de Gantt, y la fecha final del proyecto se recorrió un par de semanas, tiempo estimado para cumplir con el objetivo. Diagrama de Gantt final de “ajuste de metas y calendario”:

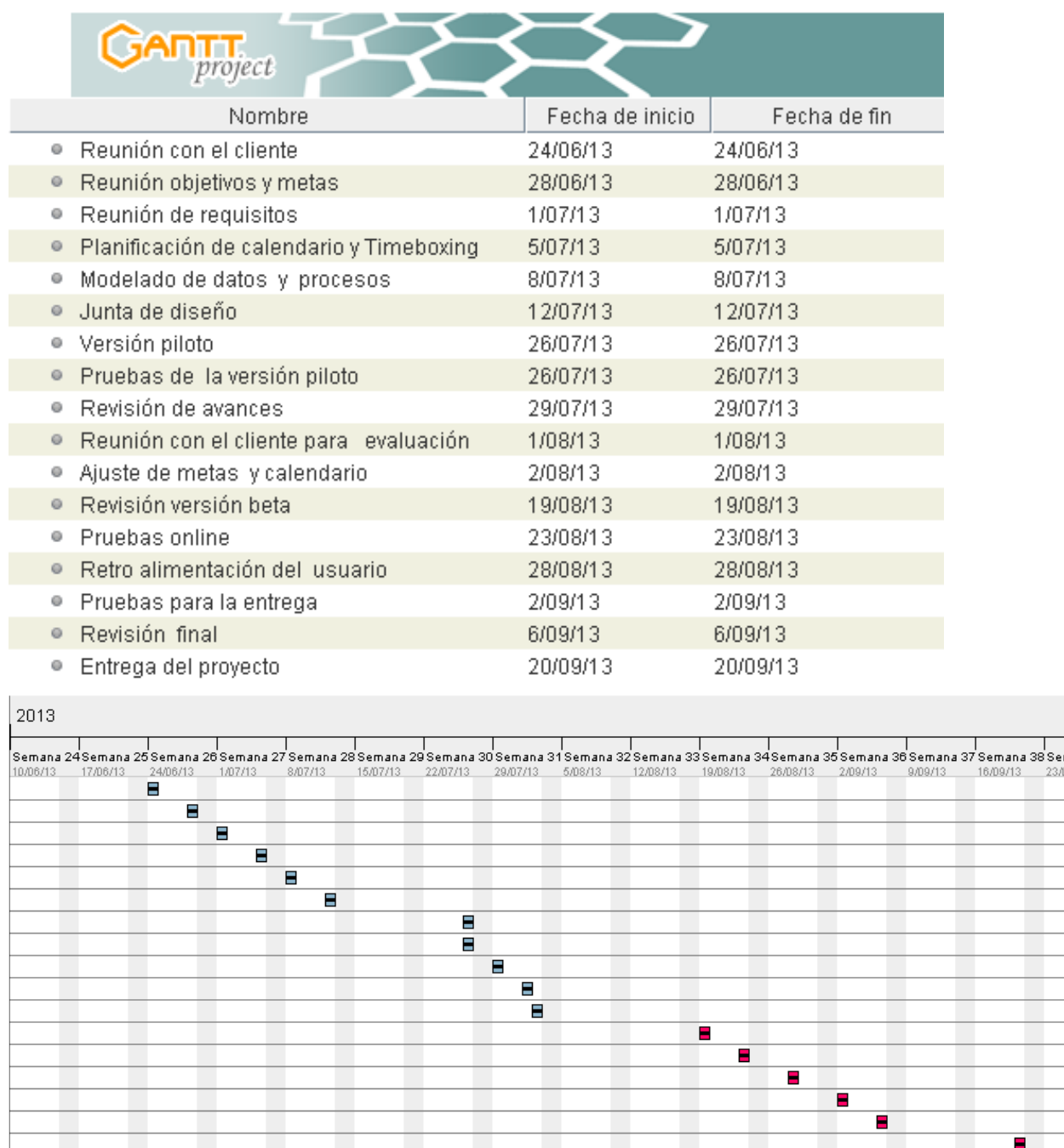


Imagen 4.34– Diagrama de Gantt final, con las modificación en los tiempos después de la revisión de avances.

Los tiempos estimados en el segundo diagrama se cumplieron, por lo que no se requirió de un tercer diagrama que hubiera necesitado de una nueva reunión de “ajuste de metas y calendario”. En conclusión cualquier proyecto sin importar la planeación siempre existirán imprevistos que afecten los tiempos y la capacidad de adaptación del proyecto hace la diferencia, métodos como el “time boxing” son muy recomendables.

### Curva de aprendizaje

Es un valor predominante en los tiempos al crear un calendario de actividades, por desgracia no es fácil calcular cuanto trabajo le costara al equipo de trabajo adaptarse a las nuevas herramientas.

Al principio del proyecto la curva de aprendizaje es mayor, significa que al principio las herramientas y conocimientos que deben adquirir son mayores afectando los tiempos, en 02/08/13 reunión de “ajuste de metas y calendario” los conocimientos necesarios ya fueron adquiridos, a partir de este punto la curva de aprendizaje fue menor y dejo de afectar al calendario.

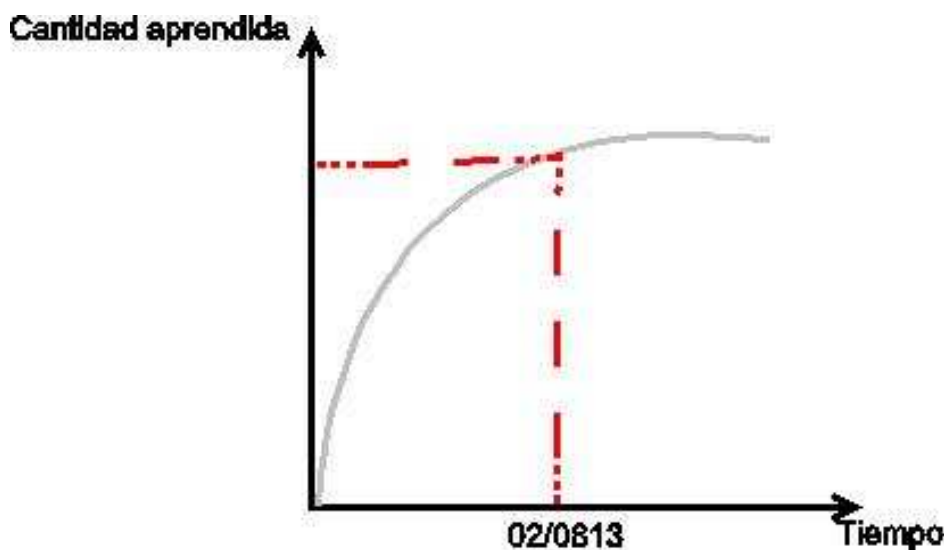


Imagen 4.35– Curva de aprendizaje, muestra la relación entre los conocimientos nuevos y el tiempo para aprenderlos.

---

#### 4.7.2- Herramientas

El uso de diversas herramientas entre ellas: herramientas CASE, Framework, IDE, etcétera pueden reducir el trabajo hasta la mitad. La gran variedad de herramientas existentes e incluso entre sus propias versiones crea conflictos al momento de elegir la adecuada para nuestro proyecto, pues la ignorancia al elegir requiere el gasto de tiempo en aprender su uso, en caso de no elegir la correcta se tendrá que empezar de nuevo con otra herramienta y es tiempo que un proyecto no puede desperdiciar; hacer una investigación previa para elegir la adecuada pero esto también implica un costo en tiempo, es uno de los grandes obstáculos que uno se puede evitar al desarrollar un proyecto.

#### Herramientas y atajos

Sin importar el número de herramientas que manejes varias con el tiempo se vuelven vitales para el proyecto y el calendario. En un proyecto de esta clase existen varias herramientas que son irremplazables y que logran aumentar la calidad y productividad en el proyecto.

- **Administrador de bases de datos:** phpMyAdmin, permite administrar en sus principales aspectos una base de datos en MySQL. En base de datos cuando los datos llegan a centenares o miles simplificar tareas monótonas y tardadas evitara la aparición de errores y tiempo perdido. Algunas de sus características más útiles: automatizar procesos, desplegar acciones en comandos SQL, desplegar acciones en código PHP, interfaz gráfica, fácil manejo, opción para introducir comandos SQL, corrección de comandos, importar o exportar base de datos en diversos formatos.
  - **Entorno de desarrollo integrado:** Aptana, al pasara hora escribiendo código con muchos comandos pertenecientes a distintos lenguajes o herramientas, memorizar todas es difícil además de innecesario porque un entorno de desarrollo integrado ayuda con tareas comunes y monótonas con: atajos, ayuda a la escritura, correcciones y sugerencias que evitara esos pequeños errores difíciles de detectar que quitan horas. Algunas de sus mejores características: notificación de errores en tiempo real, corrige, reemplaza y sugiere código.
-

### 4.7.3- Experiencia

#### Delimitar usuarios

Durante el desarrollo del proyecto una de las características de las que menos se esperaba conflictos fueron delimitar los usuarios, crear una interfaz que mezcle: facilidad, rapidez y eficiencia; elegir el nivel de libertad para modificar la interfaz y sus datos sin limitar su capacidad para cumplir su trabajo o afectar la calidad de la herramienta. Un ejemplo, los profesores pueden: leer, crear o agregar comentarios pero no pueden borrar comentarios, sólo servicios escolares pueden considerarlos terminados y eliminarlo.

#### En objetivo

El objetivo del proyecto, desarrollar una aplicación Web que permita tener una planeación semestral eficiente en la distribución de salones, materia y horarios; fue alcanzado en su totalidad y con un mínimo de cambios en el calendario.

Por parte de la interfaz tiene un diseño minimalista para el fácil manejo, donde incluso los usuarios con poca experiencia puedan manejarla. Los datos en pantalla por su cantidad son mostrados en varias pantallas para ayudar en la toma de decisiones, buscando ser simple y agradable. La información básica de la herramienta se encuentra visible en todo momento y en otra página documentación completa para su mejor manejo. La página fue reducida en peso para su fácil descarga y su comportamiento a sido probado en varios navegadores asegurando su funcionamiento y diseño para todos los usuarios.

### 4.8- Conclusión

El proyecto logro cumplir con el objetivo establecido al principio. Desarrollar una aplicación Web que permite tener una planeación semestral eficiente en la distribución de salones, materia y horarios. Al reducir procesos las ganancias son varias:

- **Personas:** Se redujo el número de personas necesarias para la actividad.
  - **Tiempo:** La principal ganancia es el tiempo, se redujo su consumo, beneficiando a las actividades relacionadas y cualquier otra que no recibiera suficiente atención por el tiempo robado.
-

- **Costo laboral:** Basado en las conclusiones anteriores. Reducir el tiempo de trabajo y el número de personas relacionadas con el mismo influye directamente en los costos reduciéndolos también.
- **Costo de desarrollo:** El implementar una herramienta o sistema puede ser muy costoso, el software libre por sus características permitió implementar la herramienta con un costo nulo, el costo total del desarrollo fue de 0\$. El dinero ahorrado podrá ser usado en cualquier otro material o en otros proyectos.
- **Papel:** Se reduce el uso de documentación impresa (físico) que afecta en costos de material y en espacio de almacenaje.
- **Correcciones:** Al mostrar los resultados en tres distintos métodos: salón, profesor y lista; y en distintas pantallas es más fácil detectar errores en los cursos y horarios.

Para cumplir con los resultados esperados se mezclaron diversos conocimientos obtenidos durante el desarrollo del proyecto: de libros, los métodos de desarrollo de software y las páginas oficiales de los lenguajes de programación. En el desarrollo se buscaron: ejemplos relacionados con el proyecto, investigar cuales eran los mejores lenguajes o herramientas para elegir los mas populares; luego elegir los más útiles, porque no es la potencia de la herramienta, es lo cómoda para uno mismo.

Al iniciar un proyecto la elección del método de trabajo para lograr un objetivo es lo más importante, en este caso RAD entre todos fue el adecuado, para un proyecto pequeño con tiempo limitado para conservar la calidad al máximo. Este método y sus enseñanza serán vuelto a utilizar en próximos proyectos por sus resultados confirmados. Una de las características más importantes del método RAD es el reuso, esta idea será aplicada en futuros proyectos al igual que el conocimiento, herramientas y funciones que fueron creadas con el fin de poder ser reutilizadas y editadas fácilmente. El reciclaje de trabajo reduce considerablemente la curva de aprendizaje consecuentemente el esfuerzo y costos.

En calendario, cualquier proyecto sin importar la planeación siempre existirán imprevistos que afecten las actividades que repercute en los tiempos, la capacidad de adaptación del proyecto hace la diferencia al momento de cumplir con su objetivo, en este caso el mismo método RAD tiene como recomendación el uso de la técnica “time boxing” la cual puede ser aplicada a cualquier proyecto futuro parecido o no.

---





# 5

## Apéndice

---

---



**A**

# **Glosario**

---

---



---

## Glosario

**Análisis:** Proceso por el cual un documento es leído y la información en él contenida se traduce en el contexto de elementos en que esta información está estructurada.

**Aplicación:** Una aplicación es una sistema que posee una colección de recursos y servicios que admite esta especificación.

**CMS:** Sistema de gestión de contenidos con siglas CMS, del inglés Content Management System, utilizado para organizar la información y facilitar la colaboración para la creación de contenido.

**Cookie:** Información mandada por un servidor a un navegador Web, que puede ser recuperada posteriormente por un programa.

**Documento:** Cadena de datos que, tras ser combinado con cualquier otra cadena a la que referencia, queda estructurado de tal manera que porta información contenida en elementos que se organizan tal y como está especificado en la correspondiente DTD.

**Elemento:** Es una unidad estructural de un documento que ha sido declarada en la DTD.

**FTP:** Protocolo de Transferencia de Archivos (File Transfer Protocol), parte del conjunto de protocolos para Internet es capaz de transmitir ficheros entre dos máquinas con distintos sistemas operativos.

**GNU:** Acrónimo de GNU no es Unix (GNU's Not Unix!), el proyecto iniciado en 1983 por Richard Stallman con objetivo de crear un sistema operativo libre.

**Gráfico:** Representación visual de un conjunto de datos que se puede ver en la pantalla de un ordenador, la mayoría de los programas de hoja de cálculo poseen un generador de gráficos.

**Gramaticalidad(Documento "bien formado"):** Un documento se dice "bien formado" o "gramaticalmente correcto" cuando está estructurado de acuerdo a las reglas definidas en algún estándar. Básicamente, en dicha definición implica que los elementos delimitados por sus etiquetas de inicio y fin, estén convenientemente anidados.

**HTML:** Lenguaje de marcado de hipertexto (Hyper Text Markup Language), basado en el uso etiquetas y atributos que le dan formato a nuestro documento para que sea visible en navegador.

---

**Interactivo:** Sistema de multimedia en el que el usuario puede controlar las acciones y la forma de trabajo del programa, o bien aquel en el que el usuario introduce comandos y el programa le responde. Todo aquel sistema o programa que posibilita el dialogo entre el ordenador y el usuario es considerado interactivo.

**Internet:** Red internacional que utiliza los protocolos TCP/IP y que posee más de diez mil redes enlazadas. Está compuesto, por tanto, por un conjunto de redes locales conectadas entre sí por medio de un ordenador llamado gateway que se encuentra en cada red. Los diferentes gateway se encuentran interconectados entre sí por diferentes medios (fibra óptica, línea telefónica, etc.).

**JavaScript:** Lenguaje script orientado a objetos de uso común en las páginas Web para agregar comportamiento dinámico a las mismas.

**Multimedia:** La combinación de sonidos, gráficas, animación, video y textos dentro de una misma aplicación. Dentro de estas aplicaciones se pueden encontrar juegos, programas de aprendizaje y material de referencia.

**Navegador:** Herramienta que despliega documentos en formato HTML y XML (páginas Web).Ejemplos: Internet Explorer, Netscape, Opera.

**Página Web:** Documento HTML situado en el World Wide Web, en el que figuran diversos enlaces de hipertexto con otros documentos del Web situados en servidores diferentes.

**Presentación:** La presentación es el proceso por el cual la información contenida en un documento se muestra al usuario. Esto se lleva a cabo de la forma más apropiada al entorno que utilice el usuario.

**Puede:** Con respecto a las aplicaciones, la palabra "puede" debe entenderse como introductoria a una característica opcional que no es obligatoria para hacer conforme el documento a esta especificación pero que puede indicarse. Con respecto a los Requisitos de Conformidad para Documentos, la palabra "puede" significa que la característica opcional que introduce no debe utilizarse. El término "opcional" tiene la misma definición que "puede".

**Reservado:** Denota un valor o comportamiento que no está especificado y cuyo uso tampoco está permitido en Documentos Conformes ni debe ser aceptado por Aplicaciones de Usuario Conformes.

**Servidor:** Aplicación software que permite trabajar a muchos usuarios con tareas determinadas.

---

**Sistema:** Es el Conjunto formado por un ordenador y todos sus periféricos, cualquier colección o combinación de programas, procedimientos, datos y equipamiento utilizados en el proceso de la información.

**Usuario:** Se define usuario como cualquier persona que esté utilizando un sistema informático, bien sea de manera individual o mediante la conexión a una red.

**Usuario final:** Desde el punto de vista de la programación, es la persona que va a utilizar el programa regularmente.

**Variable:** Representa un lugar en memoria donde se puede almacenar un dato.

---





**B**

**Bibliografía**

---

---



---

## Bibliografía

### Libros

Stallman, R. (2004). *Software libre para una sociedad libre*. Madrid: Traficantes de Sueños.

Mateu, C. (2004). *Software libre*. Barcelona: Eureka Media.

Schmuller, J. (2001). *Aprendiendo UML en 24 horas*. Edo. México: Prentice Hall.

Fowler Martin, S. K. (1999). *UML gota a gota*. México: Pearson educación.

Krug, S. (2005). *No me hagas pensar: una aproximación a la usabilidad en la Web*. México: Prentice Hall.

Welling, L. y Thomson, L. (2005). *Desarrollo Web con PHP y MySQL*. Madrid: Anaya multimedia.

DATE, C. J. (2001). *Introducción a los sistemas de bases de datos*. México: Pearson educación.

### Páginas Web

Richard, S. (2014). El Proyecto GNU. *El sistema operativo GNU*. Recuperado de <https://www.gnu.org/gnu/thegnuproject.html>

Free Software Foundation. (2014). Licencias. *El sistema operativo GNU*. Recuperado de <https://www.gnu.org/licenses/licenses.html>

Free Software Foundation. (2014). Filosofía del Proyecto GNU. *El sistema operativo GNU*. Recuperado de <https://www.gnu.org/philosophy/philosophy.html>

The World Wide Web Consortium. (2014). W3C Mission. *w3.org*. Recuperado de <http://www.w3.org/Consortium/mission.html>

The World Wide Web Consortium. (2014). Semantic Web. *w3.org*. Recuperado de <http://www.w3.org/standards/semanticweb/>

---

The World Wide Web Consortium. (2014). Tutorials and Courses. *w3.org*. Recuperado de <http://www.w3.org/2002/03/tutorials.html>

The Free Software Foundation. (2014). The Free Software Foundation (FSF) is a nonprofit with a worldwide mission to promote computer user freedom and to defend the rights of all free software users. *Free Software Foundation*. Recuperado de <http://www.fsf.org/about/>

Oracle and/or its affiliates. (2014). MySQL Documentation: Topic Guides. *MySQL: The world's most popular open source database*. Recuperado de <http://dev.mysql.com/doc/index-topic.html>

PHP Documentation Group. (2014). Manual de PHP. *PHP: Hypertext Preprocessor*. Recuperado de <http://www.php.net/manual/es/>

---