



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE INGENIERÍA

**DESARROLLO DEL PORTAL WEB DEL
MÓDULO DE COMPUTACIÓN GRÁFICA
DE LA FACULTAD DE INGENIERÍA, UNAM**

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

PRESENTA:

ERICK GUSTAVO CRUZ BARRIOS



Director de tesis: Ing. Genaro Andrés Garrido Lazcano

MÉXICO D.F.

ABRIL 2012

La más grande enseñanza que me dejó la UNAM:
Todo problema tiene solución,
y si no la conozco soy capaz de encontrarla...

Agradezco a Dios y a la vida por darme cada día la fuerza para alcanzar mis metas y cumplir mis objetivos.

Muy especialmente a mis padres por todo su apoyo y su inmenso amor. Gracias por creer en mí, confiar en mis decisiones y porque siempre me han alentado a alcanzar mis sueños para llegar a ser quien ahora soy. Gracias a mi familia, mis hermanos Edwin Iván y Edgar Cesar y su familia, mi cuñada Blanca y mis sobrinos José Esteban y María Fernanda por creer en mí en todo momento y alentarme a mejorar día con día.

Gracias a Claudia Julieta Solís González por estar a mi lado en esta etapa de mi vida, gracias por su cariño, comprensión y apoyo, por creer y confiar en mí; eres una gran persona y ya alguien muy importante en mi vida.

Gracias a la Universidad Nacional Autónoma de México, especialmente a la Facultad de Ingeniería por haberme adoptado como parte de su comunidad y brindarme la oportunidad de formarme profesionalmente. Por convertirme en una persona preparada para enfrentar la vida y por contribuir de manera significativa en mi desarrollo profesional.

Gracias a mis amigos, por compartir tan buenos momentos, por estar presentes cuando más los he necesitado, gracias a todos los del rincón de la prepa 4: Bety, Beatriz, Karla, Yura, Juan, Carlos, Ricardo, Garnica, etc. (presentes y los que se nos han adelantado en el camino) algún día estaremos juntos de nuevo. Gracias a mis amigos de la facultad ya considerados parte de mi familia, Leonardo, Cristian, Iván, Iliana, Nidia, gracias en especial a Rodrigo; y todos aquellos que han llegado a ocupar un lugar especial en mi corazón y mis recuerdos.

Gracias al Ing. Genaro Andrés Garrido Lazcano por haberme brindado su apoyo y confianza en la realización de este proyecto-tesis y compartido sus conocimientos.

A todos ustedes “MUCHAS GRACIAS”

Erick Gustavo Cruz Barrios

Índice

1.- Resumen.....	- 1 -
2.- Introducción	- 3 -
3.- Objetivos	- 4 -
3.1.- Objetivo principal	- 4 -
3.2.- Objetivos particulares	- 4 -
4.- Fundamentos.....	- 5 -
4.1.- Características de las aplicaciones basadas en <i>web</i> (<i>webapp</i>)	- 6 -
4.2.- Ingeniería de <i>webapp</i>	- 10 -
4.2.1.- Proceso.....	- 10 -
4.2.2.- Métodos	- 10 -
4.3.- Ingeniería <i>web</i> (proceso).....	- 11 -
4.3.1.- <i>WORK FLOW</i> . Dinámica de trabajo.....	- 12 -
5.- Procedimiento	- 61 -
5.1.- Definición del proyecto	- 62 -
5.2.- Estructura del sitio.....	- 71 -
5.3.- Diseño para aplicaciones <i>web</i>	- 78 -
5.4.- Desarrollo	- 94 -
5.5.- <i>Pruebas</i>	- 112 -
5.6.- Entrega.....	- 115 -
6.- Resultados.....	- 117 -
7.- Conclusiones	- 127 -
8.- Recomendaciones.....	- 128 -
9.- Glosario	- 130 -
10.- Consultas realizadas.....	- 137 -

Índice de ilustraciones

Ilustración 1 Work Flow.	- 13 -
Ilustración 2 Jerarquía de usuarios.	- 20 -
Ilustración 3 Diagrama de caso de uso para cliente nuevo.	- 21 -
Ilustración 4 Ejemplos de objetos de contenido.	- 23 -
Ilustración 5 Árbol de requisitos de calidad.	- 27 -
Ilustración 6 Pirámide del diseño web.	- 29 -
Ilustración 7 Bosquejo interfaz de la webapp.	- 31 -
Ilustración 8 Diferentes tipos de estructuras.	- 35 -
Ilustración 9 HTML, XHTML y CSS.	- 41 -
Ilustración 10 Valoración cualitativa de la facilidad de uso.	- 53 -
Ilustración 11 Faces del diseño Web.	- 62 -
Ilustración 12 Ordenamiento y clasificación.	- 70 -
Ilustración 13 Jerarquía de usuarios del portal web del MCG.	- 71 -
Ilustración 14 Diferentes tipos de estructuras.	- 73 -
Ilustración 15 Estructura y navegación del portal.	- 76 -
Ilustración 16 Pirámide del diseño web.	- 80 -
Ilustración 17 Mecanismos de control de la interfaz.	- 81 -
Ilustración 18 La plantilla para la webapp.	- 82 -
Ilustración 19 Zonas calientes.	- 83 -
Ilustración 20 Grilla.	- 85 -
Ilustración 21 Encabezado.	- 86 -
Ilustración 22 Cuerpo.	- 87 -
Ilustración 23 Pie de página.	- 88 -
Ilustración 24 Boceto Página principal.	- 88 -
Ilustración 25 Bocetos portal.	- 89 -
Ilustración 26 Diseño del Contenido.	- 90 -
Ilustración 27 Navegación de la webapp.	- 92 -
Ilustración 28 Barra de navegación horizontal.	- 92 -
Ilustración 29 Pestañas.	- 92 -
Ilustración 30 Mapa del sitio.	- 92 -
Ilustración 31 Estructura jerárquica Ilustración 32 Estructura del portal.	- 93 -
Ilustración 33 Diseño componentes.	- 94 -
Ilustración 34 Estructura Básica de una página web.	- 95 -
Ilustración 35 Maquetado de un sitio web.	- 97 -
Ilustración 36 Maquetado del portal web.	- 97 -
Ilustración 37 Content.	- 100 -
Ilustración 38 Página principal de la sección Módulo.	- 101 -
Ilustración 39 Definición del Módulo.	- 101 -
Ilustración 40 Directori del Módulo.	- 102 -
Ilustración 41 Horarios del Módulo.	- 102 -
Ilustración 42 Eventos del Módulo.	- 103 -
Ilustración 43 Plan de estudios del Módulo.	- 103 -

Ilustración 44 Proyectos del Módulo	- 104 -
Ilustración 45 Materias del Módulo (primera sección)	- 104 -
Ilustración 46 Materias del Módulo (segunda sección)	- 105 -
Ilustración 47 Material didáctico (primera parte)	- 105 -
Ilustración 48 Material didáctico (tercer parte)	- 106 -
Ilustración 49 Noticias	- 106 -
Ilustración 50 Enlaces	- 107 -
Ilustración 51 Demo-reel	- 108 -
Ilustración 52 Buscador	- 108 -
Ilustración 53 Aportaciones	- 109 -
Ilustración 54 Contáctanos	- 109 -
Ilustración 55 Imágenes listas para web	- 110 -
Ilustración 56 API GoogleMaps	- 111 -
Ilustración 57 Valoración cualitativa de la facilidad de uso	- 112 -
Ilustración 58 HOME	- 118 -
Ilustración 59 Módulo	- 119 -
Ilustración 60 Material didáctico	- 120 -
Ilustración 61 Enlaces	- 121 -
Ilustración 62 Proyectos	- 121 -
Ilustración 63 Demo reel	- 122 -
Ilustración 64 Materias	- 123 -
Ilustración 65 Buscador	- 124 -
Ilustración 66 Contáctanos	- 124 -
Ilustración 67 Noticias	- 125 -
Ilustración 68 Aportaciones	- 125 -

1.- Resumen

Actualmente, no existe un único sitio que reúna información útil para los miembros de la comunidad perteneciente al Módulo de Computación Gráfica de la Facultad de Ingeniería (horarios, plan de estudios, directorio, proyectos, etc.); aunado a esto, es difícil hallar información relacionada con temas de gráficos por computadora, además de que no existe un espacio en el cual los alumnos expongan sus proyectos.

Se llegó a la conclusión que la solución más factible para resolver este problema era desarrollar un sitio *web* (portal) especializado en temas de computación gráfica y que este dirigido a los integrantes del Módulo, el cual adicionalmente servirá como punto de partida para iniciar la navegación en internet, así como directorio de sitios web de utilidad.

El desarrollo de la *webapp* (portal) se basó en el proceso *lweb*, debido a que este adopta el enfoque de desarrollo ágil el cual subraya un punto de vista de ingeniería “magro”, riguroso, que conduce a la entrega incremental del sistema que será construido.

El portal se inició mediante la recopilación de la información que se pretendía mostrar, se continuó con el diseño de los bocetos (para la parte visual), el desarrollo del código *XHTML* y las correspondientes pruebas de calidad hasta llegar a la entrega del mismo. De acuerdo con la literatura consultada se establece que son seis los pasos mínimos necesarios para desarrollar y mantener una *webapp* (definición del proyecto, estructura del sitio, diseño, desarrollo, pruebas, entrega).

Se obtuvo el portal web del Módulo de Computación Gráfica de la Facultad de Ingeniería, el cual se empleará como plataforma para el desarrollo de los usuarios y puerta de acceso a una serie de recursos y servicios de computación gráfica. Esta *webapp* cubre los objetivos iniciales además de ser estético y funcional.

Resumen

La funcionalidad del portal consiste en sus nueve secciones: Módulo, Material didáctico, Enlaces, Proyectos, *Demo-reel*, Materias, Buscador, Contáctanos y Noticias; la mayoría de estas secciones están divididas en subsecciones para que sea posible cubrir las necesidades inicialmente planteadas.

Se cubrieron las necesidades de información específica sobre el Módulo de Computación Gráfica; se proporcionó material de apoyo académico (imágenes, texto/definiciones, videos, material multimedia etc.) para obtener el máximo beneficio en el proceso de aprendizaje; se centralizaron hipervínculos especializados en tópicos de computación gráfica; y se procuró vincular a los usuarios con la industria, mediante la exposición permanente del trabajo de los usuarios (*demo-reel's*).

Se aprovechó el espacio proporcionado por la *webapp* para poner a disposición de todos los usuarios una lista de proyectos, en los cuales los alumnos se pueden incorporar para cubrir aspectos como el servicio social, la titulación, prácticas profesionales entre otras cosas.

2.- Introducción

En el año 2006 se formó la primera generación perteneciente al Módulo de Computación Gráfica de la Facultad de Ingeniería; con el paso del tiempo esta comunidad ha aumentado significativamente; sin embargo, no existe actualmente un único sitio que reúna información útil para los miembros de esta comunidad (horarios, plan de estudios, directorio, etc.).

La información relacionada con temas de gráficos por computadora está muy dispersa y, en algunos casos, es escasa y confusa, no es fácil encontrar sitios en los cuales se pueda obtener información sobre cursos o incluso estudios ya sean de animación o desarrollo. Aunado a esto, no se cuenta con un espacio determinado para que los alumnos muestren sus proyectos al concluir con éxito los estudios pertenecientes a este módulo.

Tomando en cuenta esta problemática se llega a la conclusión que es necesario desarrollar un sitio *web* (portal) dirigido a los integrantes del módulo de Computación Gráfica. En este portal, se reunirá información pertinente al módulo como directorio de profesores, plan de estudios, inventario de materias y horarios entre otras cosas. Es decir, este proyecto contendrá lo necesario para iniciarse, capacitarse y especializarse en temas de gráficos por computadora; además, de un espacio de exposición permanente de proyectos académicos.

Con la creación de este portal será posible ofrecer en un solo lugar toda la información concerniente al módulo de Computación Gráfica y se beneficiará a toda la comunidad estudiantil de la Facultad de Ingeniería con interés en temas relacionados a la graficación. Este trabajo tiene como meta facilitar la localización de noticias, información y proyectos sobre gráficos. Además, servirá como punto de partida para iniciar la navegación en internet, así como directorio de sitios web de utilidad.

3.- Objetivos

3.1.- Objetivo principal

Desarrollar un portal web correspondiente al Módulo de Computación Gráfica de la Facultad de Ingeniería, UNAM. Este portal servirá como plataforma para el desarrollo de los alumnos del módulo además de punto de partida para los usuarios que deseen tener acceso a una serie de recursos y servicios relacionados con los temas de computación gráfica.

3.2.- Objetivos particulares

- Resolver las necesidades de información específica sobre el Módulo de Computación Gráfica.
- Centralizar hipervínculos de una forma organizada que facilite la navegación web en temas relacionados con computación gráfica.
- Proporcionar material de apoyo académico para obtener el máximo beneficio en el proceso de aprendizaje de los alumnos del módulo.
- Fomentar el desarrollo de los alumnos en áreas como: visualización científica, animación y producción digital, desarrollo de videojuegos, desarrollo de interfaces y realidad virtual, CAD y GIS; entre otras.
- Crear una lista de proyectos, en los cuales, los alumnos se pueden incorporar para cubrir servicio social, titulación, prácticas profesionales y vinculación.
- Vincular a la industria con los alumnos por medio de la exposición permanente de sus trabajos.

4.- Fundamentos

Existen una gran variedad de sistemas y aplicaciones basados en tecnología *web*, llamados *webapps*, los cuales están conformados por una mezcla de contenido y funcionalidad y son ofrecidos a un elevado número de clientes (usuarios).

La ingeniería de software es quien hereda conceptos y principios fundamentales a la ingeniería *web* (*iweb*) para su buen funcionamiento; además de contar con actividades técnicas y administrativas. Esta ingeniería está enfocada en el desarrollo de sistemas creados en y para la computadora para lo cual emplea un enfoque disciplinado.

El enfoque disciplinado en cuanto al desarrollo *web* es necesario debido a que crece en importancia la necesidad de construir sistemas seguros, fiables, adaptables e intuitivos; los cuales serán empleados por una gran diversidad de empresas en sus estrategias de negocios en red. Como ejemplo; se busca una comunicación directa con los clientes para mejorar sus productos o servicios.

La ingeniería *web* aplica un enfoque genérico que se suaviza mediante estrategias, tácticas y métodos especializados en este tipo de tecnología. El proceso de la *iweb* está enfocado en resolver un problema mediante el uso de una *webapp* en particular basándose en el planteamiento de requisitos y diseño primario; este sistema se construye con métodos, herramientas y tecnologías especializadas en desarrollo *web*.

Una vez que se tiene la *webapp* que ayudará a resolver el problema planteado será entregada a los usuarios finales, para que estos evalúen si cumple con los objetivos previamente establecidos basándose en criterios técnicos y empresariales.

Las *webapps* evolucionan (crecen) continuamente por lo que es necesario establecer un cierto mecanismo para llevar a cabo el control de configuraciones, soporte técnico y aseguramiento de la calidad del sistema.

La ingeniería *web* aplica “sólidos principios científicos, de ingeniería y de administración y enfoques disciplinados y sistemáticos para el desarrollo, despliegue y mantenimiento exitoso de sistemas y aplicaciones basados en *web* de alta calidad”¹

4.1.- Características de las aplicaciones basadas en *web* (*webapp*)

El mayor debate en cuanto a las *webapps* es conformado por sus diferencias con respecto a las otras categorías del software informático. Los sistemas basados en *web* cuentan con ciertas diferencias principales:

“Involucran una mezcla entre publicación impresa y desarrollo de software, entre *marketing* e informática, entre comunicaciones internas y relaciones externas y entre arte y tecnología”²

En la gran mayoría de las *webapp* se encuentran las siguientes características:

Intensidad de red. Toda *webapp* reside en una red. Estas *webapps* puede residir en Internet (y en consecuencia, permitir una comunicación mundial abierta); alternativamente, puede colocarse en una *Intranet* (lo que implementa la comunicación en una organización) o en una *Extranet* (comunicación inter-red). Y deben satisfacer las necesidades de una variada comunidad de clientes o usuarios finales.

1.- Murugesan, S. Julio de 1999

2.- Powell, T. A., 1998

Concurrencia. En ocasiones las *webapps* serán visitadas por un grupo nutrido de usuarios al mismo tiempo. En la mayoría de los casos, no se pueden predecir los patrones de uso de los diferentes usuarios finales por las propias variaciones de estos.

Carga impredecible. El lunes se puede contar con 100 usuarios; el martes este número puede aumentar hasta 10 000; esto muestra que el número de usuarios de la *webapp* varía dependiendo el día y la temporada en la que se encuentre.

Desempeño. Si no se cuenta con un buen desempeño de la aplicación, se corre el riesgo de que los usuarios decidan abandonar la *webapp* para buscar en otro sitio, esto se puede deber a una larga espera para ingresar, en el procesamiento del lado del servidor, o para formateo y despliegue en el lado del cliente, etc.

Disponibilidad. Lo óptimo es contar con un acceso sobre la base de “24/7/365”, pero, esto es poco razonable, aun así las *webapps* deben tender a esta disponibilidad ya que los usuarios exigentes con frecuencia lo demandan de esta manera.

Por ejemplo, los usuarios del otro lado del planeta pueden querer acceder a ciertas aplicaciones durante los momentos cuando estas estén fuera de línea por mantenimiento.

Gobernada por los datos. La función primordial de muchas *webapps* es usar hipertexto para presentar contenido de texto, gráficos, audio y video al usuario final. Además, por lo general, las *webapps* se utilizan para tener acceso a información que existe en bases de datos que originalmente no eran parte integral del ambiente basado en *web*.

Sensibilidad al contenido. La calidad visual (estética) y de contenido son un importante determinante de la calidad de toda *webapp*.

Evolución continúa. Tradicionalmente el software de aplicación convencional, evoluciona a través de una serie de liberaciones espaciadas cronológicamente (versiones); a diferencia de esto, las aplicaciones *web* evolucionan de manera continua.

Por lo tanto es ciertamente normal que algunas *webapps* (en cuanto a contenido) se actualicen sobre una agenda minuto a minuto, o que el contenido sea calculado de manera independiente para cada solicitud.

Inmediatez. Generalmente las *webapps* con frecuencia cuentan con un tiempo para comercializar que suele ser cuestión de unos cuantos días o semanas. Por lo que los ingenieros *web* se ven obligados a aplicar métodos de planeación, análisis, diseño, implementación y puesta a prueba que han sido adaptados a los escasos tiempos requeridos para el desarrollo de *webapps*.

Seguridad. Puesto que las *webapps* están disponibles mediante el acceso a la red, es difícil, sino imposible, limitar la población de usuarios finales que pueden tener acceso a la aplicación.

Con la finalidad de proteger el contenido confidencial y ofrecer modos seguros de transmisión de datos, se deben implementar fuertes medidas de seguridad a lo largo de la infraestructura que sustenta una *webapp* y dentro de la misma aplicación.

Estética. Una parte fundamental para captar la atención de los usuarios es su presentación y la disposición de sus elementos. Si se cuenta con un gran contenido, pero una pobre estética difícilmente se conservaran a los usuarios.

Cuando una aplicación se diseña para comercializar o vender productos o ideas, la estética puede tener tanto que ver con el éxito como el diseño técnico.

Estas características se aplican a todas las *webapps*, en mayor o menor medida.

Se tiene una clasificación de las *webapps* con respecto a su contenido o uso, a continuación se muestran algunas de estas clasificaciones:

- **Sitio archivo:** usado para preservar contenido electrónico valioso amenazado con extinción.
- **Sitio *web log* (o *blog* o bitácora digital):** sitio usado para registrar lecturas *online* o para exponer contenidos en línea con la fecha del día de ingreso.
- **Sitio de empresa:** usado para promocionar una empresa o servicio.
- **Sitio de comercio electrónico:** para comprar o vender diferentes bienes o servicios.
- **Sitio de comunidad virtual:** un sitio o portal social donde las personas con intereses similares se comunican unos con otros, normalmente por *chat* o foros o simples mensajes.
- **Sitio de desarrollo:** un sitio con el propósito de proporcionar información y recursos relacionados con el desarrollo de software, diseño *web*, etc.
- **Sitio de descargas:** estrictamente usado para descargar contenido electrónico, como software, juegos o fondos de escritorio.
- **Sitio de juego:** un sitio que es propiamente un juego o un “patio de recreo” donde mucha gente asiste a jugar.
- **Sitio de información:** contiene contenido que pretende informar a los visitantes, pero no necesariamente de propósitos comerciales. La mayoría de los gobiernos e instituciones educativas y sin ánimo de lucro tienen un sitio de información.
- **Sitio buscador:** un sitio que proporciona información general y está pensado como entrada o búsqueda para otros sitios.
- **Sitio personal:** mantenido por una persona o un pequeño grupo (como por ejemplo la familia) que contiene información o cualquier contenido que la persona quiere incluir.
- **Sitio portal:** un sitio *web* que proporciona un punto de inicio, entrada o portal, a otros recursos en Internet o una intranet.
- **Sitio *Web 2.0*:** un sitio donde los propios usuarios, los propios visitantes, son los responsables de mantener la aplicación viva, usando tecnologías de última generación.
- **Wiki:** un sitio donde los usuarios editan colaborativamente (por ejemplo *Wikipedia*).
- **Sitios educativos:** promueven cursos presenciales y a distancia, información a profesores y estudiantes, permiten ver o descargar contenidos de asignaturas o temas.

4.2.- Ingeniería de *webapp*

Para un buen desarrollo de sistemas y aplicaciones basados en *web* (*webapps*) se deben incorporar modelos de procesos especializados, métodos de ingeniería del software adaptados a las características del desarrollo de estas y un conjunto de importantes tecnologías *web*.

4.2.1.- Proceso

Los modelos de procesos *iweb* adoptan la filosofía del desarrollo ágil. El desarrollo ágil enfatiza un enfoque de desarrollo riguroso que incorpora rápidos ciclos de desarrollo. Aoyama (noviembre-diciembre, 1998) describe la motivación para el enfoque ágil en la siguiente forma:

“Internet cambió la prioridad principal del desarrollo del software de qué a cuándo. El reducido tiempo para el mercado se ha convertido en el límite, competitivo por el que luchan las compañías líderes. En consecuencia, reducir el ciclo de desarrollo es ahora una de las misiones más importantes de la ingeniería del software”

4.2.2.- Métodos

Tenemos que los métodos de *iweb* son un conjunto de labores técnicas que permiten al ingeniero *web* comprender, caracterizar y construir una serie de *webapps* de alta calidad. Así pues los métodos de *iweb* se categorizan de la siguiente manera:

Métodos de comunicación: consisten del enfoque útil para facilitar la comunicación entre usuarios finales, clientes de negocios, expertos en problemas de dominio, etc., con los ingenieros *web*. La importancia de las técnicas de comunicación

radica en la recolección de requisitos y posteriores juntas para evaluar los incrementos en la *webapp*.

Métodos de análisis de requisitos: sirven para comprender que es lo que se espera de la *webapp*, desde el contenido que presentará, funciones que proporcionará a los usuarios hasta la manera en la que se desea que interactúe con estos mientras naveguen por la *webapp* y de esta manera conseguir su completa satisfacción.

Métodos de diseño: se asientan las revisiones técnicas de diseño para el contenido, la aplicación y la arquitectura de información, así como el diseño de interfaz (aspecto visual) y la estructura de navegación de la *webapp*.

Métodos de prueba: incluye una serie de revisiones técnicas formales (contenido, modelo de diseño y una amplia variedad de técnicas de prueba que abordan conflictos al nivel de componente y arquitectónicos), pruebas de la navegación, pruebas de facilidad de uso, pruebas de seguridad y pruebas de configuración.

4.3.- Ingeniería *web* (proceso)

El modelo de proceso ágil produce liberaciones de software a un ritmo vertiginoso; es por eso que en la mayoría de los casos es el elegido por los equipos de ingeniería *web* (*iweb*), esto debido a los principales atributos de las *webapp*: inmediatez y evolución continua.

La simple naturaleza de las *webapps* actuales nos muestran una amplia diversidad de usuarios con diferentes niveles de especialización a los cuales tenemos que satisfacer, desde el que busca un diseño simple y fácilmente manejable, hasta el que quiere una aplicación acorde a su nivel de conocimientos.

4.3.1.- *WORK FLOW*. Dinámica de trabajo

Se deben de tener conocimientos sólidos sobre los procedimientos por seguir al momento de levantar los requerimientos necesarios para el buen desarrollo del sitio *web*.

Esto ayudara a tener claras las necesidades y lineamientos del sitio en cuestión, elaborar una propuesta que cubra dichas necesidades y hacer una correcta evaluación del nivel de diseño y la complejidad de desarrollo que demandara el sitio.

Un modelo de procedimiento ideal ayuda en el desarrollo *web* a superar la complejidad y fracaso del sitio, a minimizar el riesgo de fallo del proyecto y a tener un control de los cambios durante el desarrollo; ya que al proporcionar la información desde el inicio del proyecto se pueden tomar las mejores decisiones para hacer más eficiente el desarrollo del mismo logrando entregarlo en el tiempo establecido y cumpliendo con los lineamientos y objetivos planteados.

Esta dinámica de trabajo se compone de seis etapas, las cuales son:

- Definición del proyecto.
- Estructura del sitio.
- Diseño.
- Desarrollo.
- Pruebas.
- Entrega.

Fundamentos

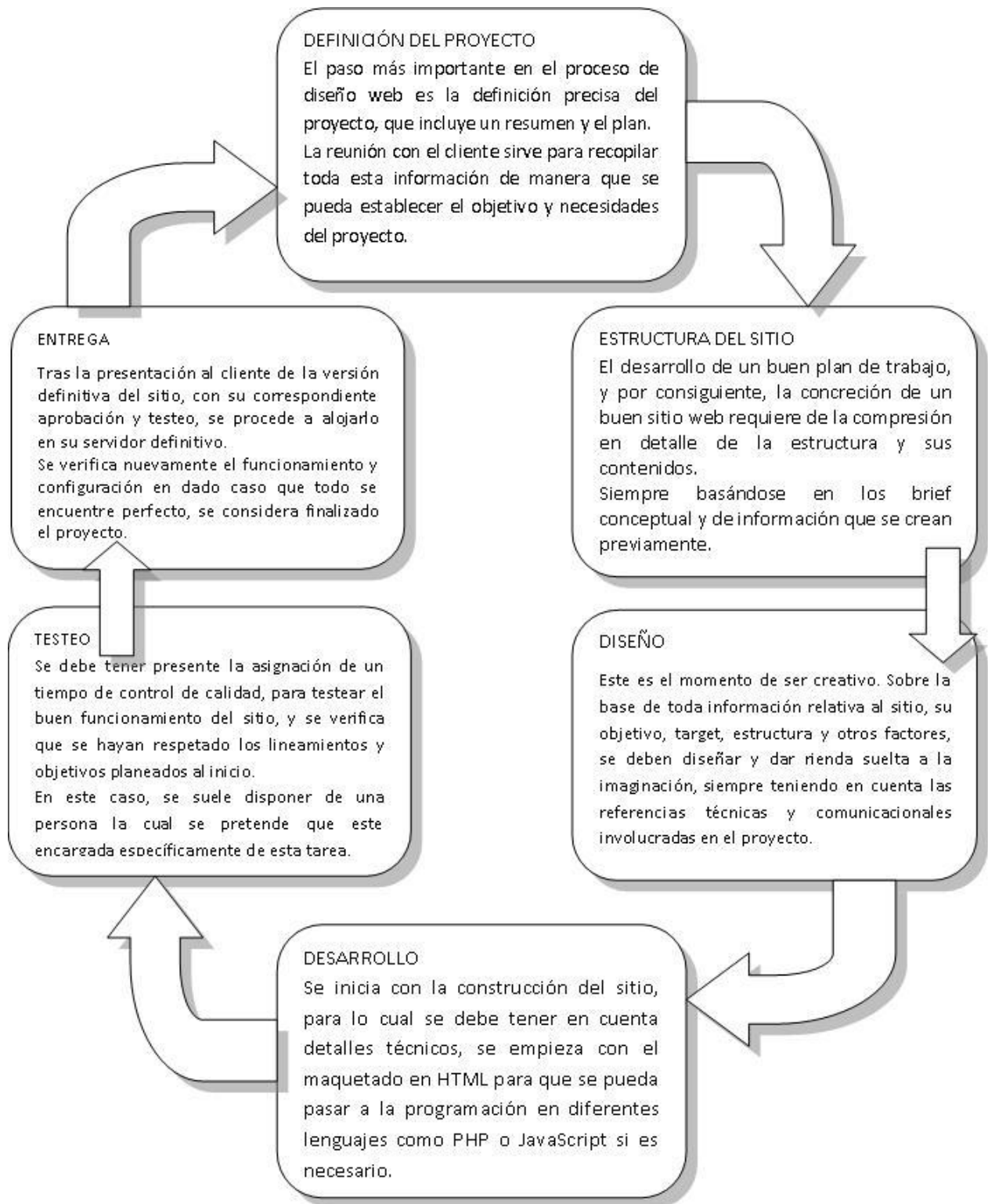


Ilustración 1 Work Flow. Dinámica de trabajo de los diseñadores y desarrolladores web. Se presentan cuales son las diferentes etapas del desarrollo web.

4.3.1.1.- Definición del proyecto

Entre los principios fundamentales de la ingeniería del software destaca el de: “*comprender el problema antes de comenzar a resolverlo y estar seguro de que la solución concebida es aquella que la gente realmente quiere*”.³ Esta es la base de la formulación, la primera gran actividad en la ingeniería *web*. Otro principio fundamental de la ingeniería de software es: planear el trabajo antes de comenzar a realizarlo.

Una buena planeación nos permite gestionar el trabajo conforme avanza el desarrollo del proyecto ya que contempla los elementos básicos necesarios para establecer un programa de actividades que nos lleven al buen flujo de trabajo. La formulación valora la creación de la *webapp*, sus características particulares y las funciones disponibles para los usuarios finales.

Al establecer la comunicación con el cliente comienza la formulación, primordialmente se plantean las razones para desarrollar la *webapp*: ¿Por qué es necesaria la *webapp*?, ¿Público objetivo?, ¿Características técnicas y funciones para los usuarios finales?, ¿Qué sistemas y/o bases de datos emplearán?, ¿Es técnicamente realizable?

Al planear el desarrollo de la *webapp* se establece un plan de trabajo, se establecen fechas tentativas de entrega, presupuestos, riesgos a considerar, se define también un programa de desarrollo y mecanismos para rastreo y control del desarrollo.

El análisis de requisitos para las *webapps* abarca tres grandes tareas: formulación, recopilación de requisitos y modelado de análisis. Durante la formulación se identifican la motivación (metas) y los objetivos básicos para la *webapp* y también se definen las categorías de usuarios.

3.- Roger S. Pressman, 2006; Ingeniería del software

Cuando comienza la recopilación de requisitos se intensifica la comunicación entre el equipo de ingeniería *web* y los accionistas. Los requisitos de contenido y funcionales se enlistan y se desarrollan los escenarios de interacción (casos de uso) escritos desde el punto de vista del usuario final. La intención es establecer una comprensión básica de por qué se construirá la *webapp*, quién la usará y qué problema resolverá a sus usuarios.

La formulación y la planeación proporcionan una guía útil a todo equipo de ingeniería *web*, esto para llegar a un buen desarrollo web.

Sistemas basados en web (formulación)

La formulación de *webapps* se basa en una secuencia de acciones propias de la ingeniería *web* siendo primero la identificación de las necesidades que dieron pie al desarrollo de la misma, continua con una descripción breve pero concisa de los objetivos de la *webapp*, define las características y funciones sobresalientes de esta además de recopilar los requisitos que nos llevan al desarrollo del modelo de análisis.

Gracias a la formulación el equipo de ingeniería *web* y los clientes pueden establecer un conjunto común de metas y objetivos para el desarrollo de la *webapp*.

Primeras preguntas

Son tres las preguntas fundamentales que se deben realizar y responder al comienzo de la etapa de formulación (Powel, T.A., *Web Site Engineering*, 1998):

- ¿Cuál es la principal motivación (necesidades del negocio) para la *webapp*?
- ¿Cuáles son los objetivos que debe satisfacer la *webapp*?
- ¿Quién usará la *webapp*?

De manera breve y concisa se han de contestar estas simples preguntas, para evitar ambigüedades.

En general, se identifican dos categorías de metas⁴:

- Metas informativas: indican una intención de proporcionar contenido e información específicos al usuario final.
- Metas aplicables: indican la habilidad para realizar alguna tarea dentro de la *webapp*.

El desarrollo de un perfil de usuario se lleva a cabo una vez ya identificadas todas las metas informativas y aplicables. El perfil de usuario captura “características relevantes relacionadas con los usuarios potenciales, que incluye sus antecedentes, escolaridad, preferencias e incluso más”⁴.

Recopilación de requisitos para *webapps*

A diferencia de la Ingeniería de Software en la cual los objetivos globales de la recopilación de requisitos propuestos permanecen inalterados; para la ingeniería *web* esta actividad puede abreviarse. Dichos objetivos adaptados para las *webapps* serían:

- Identificar requisitos de contenido.
- Identificar requisitos funcionales.
- Definir escenarios de interacción para diferentes clases de usuarios.

4.- Gnado, C. y F. Larcher, mayo de 1999

Para conseguir estos objetivos es necesario una serie de pasos de la recopilación de requisitos:

1. Pedir a los clientes que definan las categorías de usuario y describan cada categoría.
2. Comunicarse con los clientes para definir los requisitos básicos de la *webapp*.
3. Analizar la información recopilada y utilizar la información para realizar un seguimiento con los clientes.
4. Definir casos de uso que describan escenarios de interacción para cada clase de usuario.

Definición de las categorías de usuario. Para poder definir una categoría de usuario es recomendable responder un conjunto de preguntas fundamentales:

- ¿Cuál es el objetivo global del usuario cuando usa la *webapp*?
- ¿Cuáles son los antecedentes y la pericia del usuario en relación con el contenido y la funcionalidad de la *webapp*?
- ¿Cómo llegara el usuario a la *webapp*? ¿El arribo ocurrirá a través de un enlace desde otro sitio *web*, o llegara en una forma más controlada?
- ¿Qué características genéricas de las *webapps* le gustan o disgustan al usuario?

Las respuestas a estas preguntas nos ayudan a definir un pequeño conjunto de clases de usuarios, al avanzar en la recopilación de requisitos es recomendable encuestar a cada diferente clase de usuarios para de esta manera obtener datos.

Comunicación con los clientes y usuarios finales. Debido a que toda *webapp* tiene una amplia población de usuarios finales, es recomendable no emplear información recopilada sólo de una o dos personas como la base para la formulación o el análisis. La creación de categorías o clases de usuario ayuda a que una evaluación de los requisitos de usuario sea más manejable.

Algunos mecanismos que pueden lograr una buena comunicación con los clientes son:

- Grupo muestra tradicional.
- Grupo muestra electrónico.
- Entrevistas iterativas.
- Entrevistas de exploración.
- Construcción de escenarios.

Análisis de la información recopilada. Al tiempo en que se recopila información es recomendable categorizarla en clases de usuario y tipo de transacción para valorarla según su propia relevancia.

Algunos de los objetivos son: desarrollar listas de objetos de contenido, listas de operaciones aplicables a los objetos de contenido dentro de una transacción de usuario específica y listas de funciones proporcionadas a los usuarios finales.

Desarrollo de casos de uso. Un caso de uso nos ayuda a describir la manera cómo interactúa la *webapp* con determinada categoría de usuario específica, esto para lograr una acción específica.

Estas acciones pueden ir desde: entregar cierta información descriptiva a los usuarios finales hasta realizar complejos cálculos matemáticos en base a datos proporcionados por los usuarios o almacenar un cierto registro en determinada base de datos. Los casos de uso son una descripción de la interacción con la *webapp* desde un punto de vista del usuario final. Los casos de uso:

- 1) Ayudan al desarrollador a entender cómo perciben los usuarios su interacción con la *webapp*.
- 2) Proporcionan el detalle necesario para crear un modelo de análisis efectivo.
- 3) Ayudan a dividir en compartimientos el trabajo de *iweb*.
- 4) Ofrecen una guía importante para quienes deben probar la *webapp*.

4.3.1.2.- Estructura del sitio

Tres son las preguntas importantes sobre las que se enfoca el análisis de una potencial *webapp*:

- 1) ¿Qué información o contenido se presentará o manipulará?
- 2) ¿Qué funciones realizará el usuario final?
- 3) ¿Qué comportamientos exhibirá la *webapp* conforme presente contenido y realice funciones?

Las respuestas a estas preguntas se representan como un modelo de análisis que abarca una diversidad de representaciones *UML*.

Textos *UML* y un conjunto de diagramas que describen el contenido, la interacción, la función y la configuración, son los actores que integran el modelado de análisis.

Análisis de requisitos para las *webapps*

El análisis de requisitos para las *webapps* cubre tres tareas básicas:

1. Formulación.
2. Recopilación de requisitos.
3. Modelado de análisis.

Primeramente se lleva a cabo la formulación en la cual se identifican las razones y los objetivos para la *webapp*, así como definir las categorías de usuarios. Ya una vez realizado esto se comienza la recopilación de requisitos con lo que se intensifica la comunicación entre el equipo de ingeniería *web* y los clientes.

Tomando en cuenta que la intención es establecer una comprensión básica de la *webapp*: qué problema resolverá, quién la usará y el por qué se construirá; se enlistan los requisitos de contenido y funcionales y se desarrollan los escenarios de interacción (casos de uso) escritos desde el punto de vista del usuario final.

Jerarquía de usuario

Las categorías de usuarios se identifican como parte de las tareas de formulación y de recopilación de requisitos. Las categorías de usuarios son relativamente limitadas y no necesitan una representación *UML*. Cuando crece el número de categorías de usuario, es aconsejable desarrollar una jerarquía de usuarios, como se muestra en la Ilustración 2.

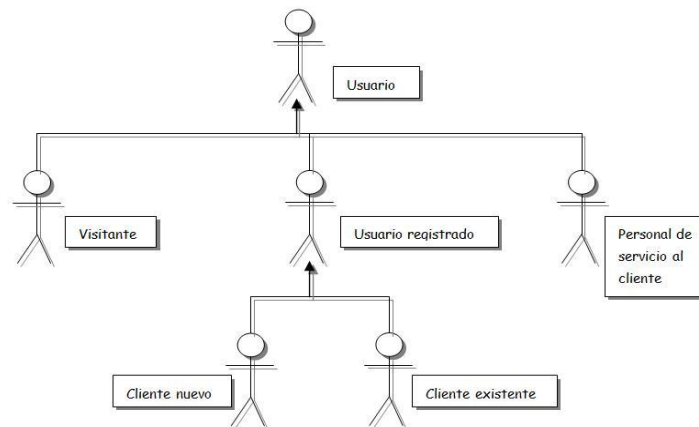


Ilustración 2 Jerarquía de usuarios.

Desarrollo de casos de uso

Franklin⁵ se refiere a los casos de uso como “haces de funcionalidad”. A cada categoría de usuario descrita en la jerarquía de usuario se le desarrolla un caso de uso. La descripción por parte de la ingeniería *web* nos dice que el caso de uso en sí mismo es relativamente informal: un párrafo narrativo que describe una interacción específica entre un usuario y la *webapp*.

5.- Franklin, S., 2001 “*Planning Your Web Site with UML*”

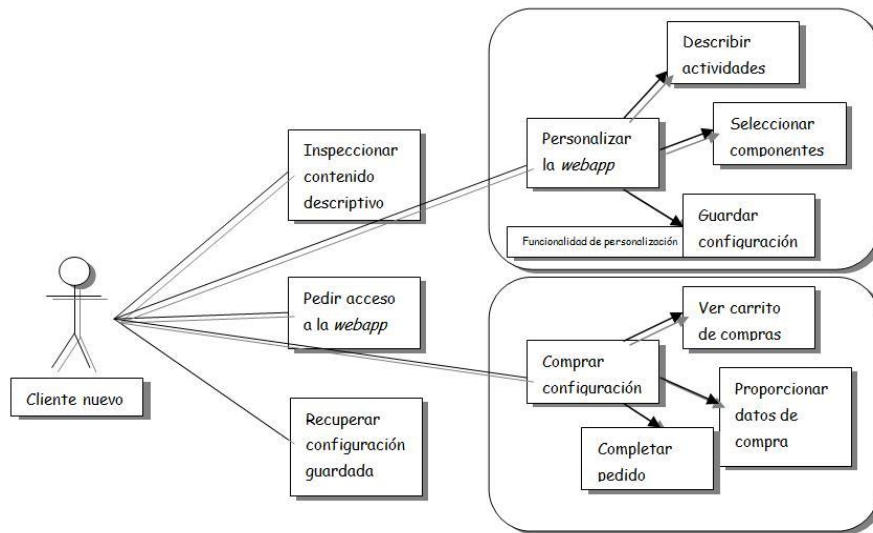


Ilustración 3 Diagrama de caso de uso para cliente nuevo

Ajustes para el modelo de caso de uso

Es necesario desarrollar una vista superior de los requisitos de la *webapp* observables de manera externa. Los casos de uso deben organizarse en paquetes funcionales que se tienen que valorar de manera que se garantice que cada uno es (Conallen, J., 2000):

- Comprensible: todos los clientes entienden el propósito del paquete.
- Cohesivo: el paquete aborda funciones relacionadas cercanamente una con otra.
- Librementemente acoplados: las funciones o clases dentro del paquete colaboran una con otra, pero la colaboración exterior del paquete se mantiene en un mínimo.
- Jerárquicamente superficial: las jerarquías funcionales profundas son difíciles de navegar y entender para los usuarios; en consecuencia, el número de niveles dentro de una jerarquía de casos de uso debe reducirse siempre que sea posible.

El análisis de requisitos y el modelado son actividades iterativas, así que es probable que se sumen nuevos casos de uso a los paquetes que se han definido, que los casos de uso existentes sean refinados y que casos de uso específicos puedan reubicarse en paquetes diferentes.

El modelado de análisis para *webapps*

Se estudian cuatro actividades de análisis, cada una con su aporte a la creación de un modelo de análisis completo:

- Análisis de contenido: identifica todo el espectro del contenido que ofrecerá la *webapp*. En contenido incluye texto, gráficas e imágenes, así como datos de video y audio.
- Análisis de interacción: describe como interactúa el usuario con la *webapp*.
- Análisis de funciones: define las operaciones que se aplicarán al contenido de la *webapp* y describe otras funciones de procesamiento, independientes del contenido pero necesarias para el usuario final.
- Análisis de configuración: describe el ambiente y la infraestructura en la que reside la *webapp*.

Durante las tareas de estos cuatro análisis se recopila información, la cual se tiene que revisar, modificar (si se requiere) y organizar en un modelo que se pueda transferir a los diseñadores *web* para que puedan emplearla de la mejor manera en el desarrollo del diseño de la *webapp*.

El modelo de contenido

Este modelo contiene los elementos estructurales que proporcionan una importante visión de los requisitos de contenido para la *webapp*. Entre estos elementos

estructurales se incluyen objetos de contenido (texto, imágenes gráficas, fotografías, imágenes de video, audio) que se pueden presentar como parte de la *webapp*.

Además, incluye todas las clases de análisis: entidades visibles para el usuario que se crean o manipulan conforme éste interactúa con la *webapp*.

Objetos de contenido

La razón principal de toda aplicación *web* es el presentar información preexistente (contenido) al usuario final. La calidad y estructura de este contenido puede ser desde lo más sencillo hasta lo más complejo.

No se tiene un tiempo estricto para el desarrollo de el contenido, puede ser antes de implementar la *webapp*, una vez que esta ya esté operando o mientras se está construyendo.

Una noticia relativa a un tema de interés, alguna imagen o fotografía, la animación de un logotipo corporativo, el video de algún evento importante, una colección de diapositivas electrónicas, o bien una descripción textual de algún producto son bien un objeto de contenido.



Ilustración 4 Ejemplos de objetos de contenido

El modelo de interacción

Este modelo se basa en la interacción entre la funcionalidad de la *webapp* con el usuario final, el contenido y el comportamiento de esta. Este modelo de interacción lo componen cuatro elementos:

- 1) Casos de uso.
- 2) Diagramas de secuencia.
- 3) Diagramas de estado.
- 4) Un prototipo de interfaz de usuario.

La interacción también se representa dentro del contexto, del modelo de navegación.

El modelo funcional

El modelo funcional aborda dos elementos de procesamiento de la *webapp* y cada uno representa un grado diferente de la abstracción de procedimiento:

- 1) Funcionalidad observable respecto al usuario y que entrega al usuario final la *webapp*.
- 2) Las operaciones dentro de las clases de análisis que implementan comportamientos asociados con la clase.

El modelo de configuración

Tanto en el lado del servidor como en el del cliente existe una amplia diversidad de ambientes, por ello es que hay que diseñar e implementar las *webapps* para que éstas se acoplen a todos y cada uno de estos ambientes.

Es recomendable especificar una serie de datos para el buen funcionamiento de las *webapps*, como pueden ser si la *webapp* reside en un servidor que proporcione acceso vía *Internet*, *Intranet* o bien *Extranet*; especificar el hardware del servidor y que sistema operativo maneja.

Un punto importante a considerar es los aspectos de interoperabilidad en el lado del servidor. Si es necesario que la *webapp* tenga acceso a bases de datos, o bien interoperar con aplicaciones ya existentes en el lado del servidor; se deben especificar cuáles son las interfaces apropiadas, los protocolos de comunicación y la información complementaría necesaria.

Del lado del cliente existen diferentes aplicaciones que proporciona cierta infraestructura que permite el acceso a la *webapp* desde la ubicación del usuario final. El software de navegación (*Internet Explorer*, *Firefox*, *Opera*, etc) es el encargado de entregar el contenido y la funcionalidad de la *webapp* que se descarga del servidor.

Cada uno de estos navegadores cuenta con sus propias peculiaridades; esto a pesar de que existen estándares. Para sobreponerse a esto, la *webapp* debe someterse a una amplia prueba de configuración de navegador, esto se especifica como parte del modelo de configuración.

4.3.1.3.- Diseño para aplicaciones web

Para diseñar *webapps* es necesario abarcar actividades tanto técnicas como no técnicas; el diseño gráfico se encarga de desarrollar la visión y el sentido del contenido, el diseño de la interfaz cubre la creación de la plantilla estética de la interfaz de usuario, así como el diseño arquitectónico y de navegación se encarga de modelar la estructura técnica de la *webapp*.

Primeramente hay que crear un modelo de diseño antes de comenzar la construcción de la *webapp*, todo buen ingeniero *web* sabe que el diseño evolucionará conforme más se conozca sobre los requisitos de los participantes mientras se construye la *webapp*.

Diseño en la Ingeniería Web

El diseño dentro del contexto de la ingeniería *web* debe considerar cuestiones tanto genéricas como específicas. El diseño ofrece un modelo que sirve de guía para la construcción de la *webapp*.

El modelo de diseño debe contar con la información necesaria para interpretar los requisitos de los participantes y pasarlos a contenido y código ejecutable; debe ser específico y tomar en cuenta las características mínimas necesarias para que el ingeniero *web* pueda construir y poner en marcha la *webapp* para de esta manera ponerla a prueba de una manera efectiva y así obtener retroalimentación.

Diseño y calidad de la *webapp*

Existen ciertos criterios, los cuales se pueden tomar como base para decir si una *webapp* es buena o no; cada usuario puede tener su punto de vista y no necesariamente estos tienen que coincidir.

Lo que para algunos usuarios es agradable para la vista (gráficos en movimiento, sonidos, cambio de colores, etc.), para otros esta por demás (prefieren texto simple, colores sólidos y sin sonidos), sobre la funcionalidad de la *webapp*. Muchos usuarios optan por grandes cantidades de información sobre el mismo tema, mientras que otros prefieren extractos resumidos y concisos de la misma información.

Algunos les gusta que todo sea simple y fácil de usar y otros los motivan las herramientas sofisticadas, accesos a bases de datos, o aplicaciones que cuenten con

gran complejidad. Es posible asegurar que sobre cualquier discusión técnica de la calidad de la *webapp* está la percepción de usuario final sobre lo que es “bueno o aceptable” en cuanto al desarrollo y diseño de la *webapp*.

Facilidad de uso, funcionalidad, confiabilidad, eficiencia y facilidad de mantenimiento son las características generales más relevantes de la calidad de software que se aplican a las *webapps*, estas proporcionan una base para valorar la calidad de los sistemas *web*.

Olsina y sus colegas han preparado un “árbol de requisitos de calidad” que identifican un conjunto de atributos técnicos que conducen a *webapps* de gran calidad. La Ilustración 5 resume este trabajo, los criterios anotados en ella son de particular interés para los ingenieros *web* que deben diseñar, construir y mantener las *webapps* a largo plazo.

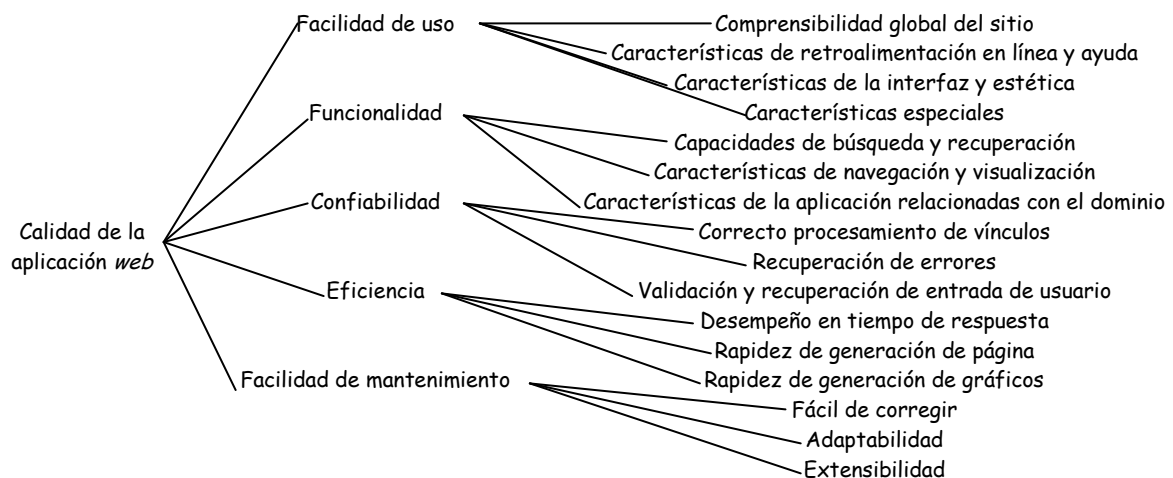


Ilustración 5 Árbol de requisitos de calidad

Offutt extiende los cinco principales atributos de calidad anotados en la Ilustración anterior al agregar:

- Seguridad.
- Disponibilidad.
- Escalabilidad.
- Tiempo de mercado.

Metas de diseño

En su columna regular acerca del diseño *web*, Jean Kaiser (Kaiser, J., 2002) sugiere las siguientes metas de diseño, que son aplicables virtualmente a toda *webapp* sin importar el dominio, tamaño y complejidad de la aplicación:

- Simplicidad.
- Consistencia.
- Identidad.
- Robustez.
- Navegabilidad.
- Apariencia visual.
- Compatibilidad.

Pirámide del diseño *web*

Una buena mezcla de estética, contenido y tecnología es el producto del diseño, esta mezcla variara dependiendo de la *webapp*, la complejidad y diseño estético de esta; de igual manera variaran las actividades de diseño.

A continuación se muestra la pirámide de diseño para ingeniería *web*:

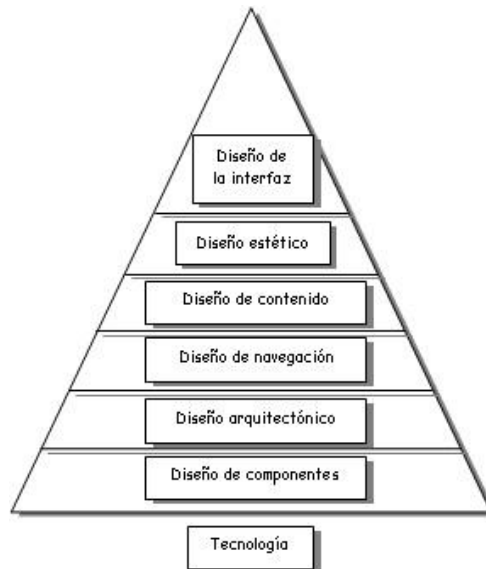


Ilustración 6 Pirámide del diseño web

Diseño de la interfaz de la *webapp*

Es recomendable que toda interfaz de usuario tome en cuenta el uso de las siguientes características: fácil de aprender, fácil de usar, fácil de navegar, intuitiva, consistente, eficiente, funcional y libre de errores. Y lo más importante es ofrecer al usuario final una experiencia satisfactoria y gratificante.

Dix (Dix, A., septiembre de 1999) argumenta que todo ingeniero *web* debe diseñar la interfaz de usuario de modo que responda tres preguntas primarias para el usuario final:

1. ¿Dónde estoy?
2. ¿Qué puedo hacer ahora?
3. ¿Dónde he estado, a dónde voy?

Mecanismos de control de la interfaz

Una *webapp* tiene los siguientes objetivos:

- 1) Establecer una ventana consistente con el contenido y la funcionalidad que proporciona.
- 2) Guiar al usuario a través de una serie de interacciones con la *webapp*.
- 3) Organizar las opciones de navegación y el contenido disponible para el usuario.

El diseñador de la interfaz debe emplear alguna técnica apropiada que permita al usuario final aprender de manera intuitiva el uso de la interfaz, esto con el fin de guiar la interacción entre la interfaz y el usuario. Existen varios mecanismos de interacción de entre los cuales el diseñador puede seleccionar el que más se acomode a *webapp*, algunos de ellos son:

- Menús de navegación.
- Iconos gráficos.
- Imágenes gráficas.

Flujo de trabajo

Para iniciar el diseño de la interfaz de usuario es necesario identificar la tarea y los requisitos ambientales. Una vez que se tengan ya identificadas las tareas del usuario, se crean y analizan los casos de uso para definir el conjunto de objetos y acciones propias de la interfaz.

A continuación se muestra una lista que contiene las tareas que representan un flujo de trabajo rudimentario para el diseño de la interfaz para una *webapp* exitosa:

1. Revisar la información contenida en el modelo de análisis y refinarla conforme se requiera.
2. Desarrollar un bosquejo aproximado de la plantilla de la interfaz de la *webapp*.

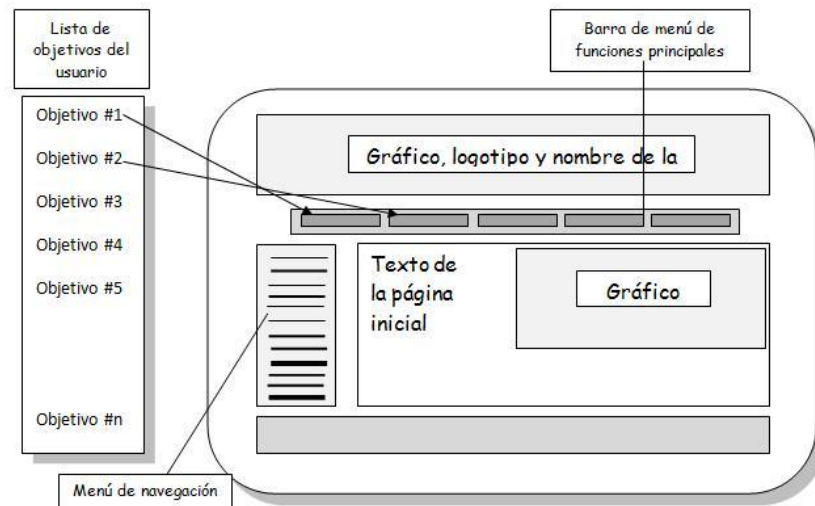


Ilustración 7 Bosquejo interfaz de la webapp

3. Correlacionar los objetivos del usuario con acciones específicas de la interfaz.
4. Definir un conjunto de tareas de usuario que estén asociadas con cada acción.
5. Elaborar bosquejos con imágenes de la pantalla para cada acción de la interfaz.
6. Refinar la plantilla de la interfaz y los bosquejos con el uso de entradas desde el diseño estético.
7. Identificar los objetos de la interfaz del usuario que se requieran para implementarla.
8. Desarrollar una representación del comportamiento de la interacción del usuario con la interfaz.
9. Desarrollar una representación del comportamiento de la interfaz.
10. Describir la plantilla de la interfaz para cada estado.
11. Refinar y revisar el modelo de diseño de la interfaz.

Diseño estético

Este diseño también llamado diseño gráfico, es un complemento a los aspectos técnicos de la ingeniería *web* desde el punto de vista de un esfuerzo artístico. Al no contar con este diseño gráfico, las *webapps* son funcionales pero sin atractivo y, al

agregarlo; la *webapp* lleva a los usuarios a un mundo que los incluye en un ámbito tanto emocional como intelectual.

La plantilla para la *webapp*

El desarrollo del “bien inmueble” se planea durante el diseño estético. El bien inmueble puede usarse para dar soporte a la estética no funcional, características de navegación contenido de información y funcionalidad dirigida al usuario final de toda *webapp*.

No existen reglas absolutas o normas básicas que seguir cuando se trata del diseño de una plantilla de pantalla; sin embargo, es recomendable considerar algunos lineamientos generales de plantilla, como los mencionados a continuación:

- No temerle al espacio vacío.
- Resaltar el contenido.
- Organizar los elementos de plantilla de arriba a la izquierda hacia abajo a la derecha.
- Agrupar navegación, contenido y función geográficamente dentro de la página
- No extender el bien inmueble con la barra de desplazamiento.
- Considerar la resolución y el tamaño de la ventana de navegador cuando se diseñe plantillas.

Diseño gráfico

Un buen diseño gráfico toma en consideración cada aspecto de la presentación y percepción de la *webapp*. El proceso de este diseño inicia con la plantilla y continúa con la consideración de esquemas de color globales (teoría del color); tipos, tamaños y estilos de fuentes además del uso de medios audiovisuales complementarios (audio, video, animaciones, etc.) y todos los elementos estéticos que hagan falta para la aplicación *web*.

Diseño del contenido

El diseño de contenido se ocupa de la representación de la información dentro de un objeto de contenido específico: actividad de diseño que dirigen los publicistas, los diseñadores gráficos y otros que generan el contenido de una *webapp*.

Objetos de contenido

En el contexto de la ingeniería *web* un objeto de contenido está alineado de manera más cercana con un objeto de dato para software convencional. Un objeto de contenido tiene atributos que incluyen información específica de contenido y atributos específicos de implementación que se especifican como parte del diseño.

Mediante una asociación *UML* y un agregado se pueden representar relaciones entre los objetos de contenido.

Diseño arquitectónico

Las metas establecidas para la *webapp* están enlazadas directamente con el diseño arquitectónico, con el contenido que se presentara, la filosofía de navegación establecida y los usuarios que se espera la visitan. El diseñador encargado de esta actividad identifica la arquitectura de contenido y la propia de la *webapp*.

La arquitectura de contenido fija su atención principalmente en la manera en la que los objetos de contenido se estructuran para su presentación y navegación. La arquitectura de la *webapp* es la encargada de gestionar la interacción entre esta y el usuario, maneja las tareas de procesamiento, presenta el contenido además de efectuar la navegación correspondiente.

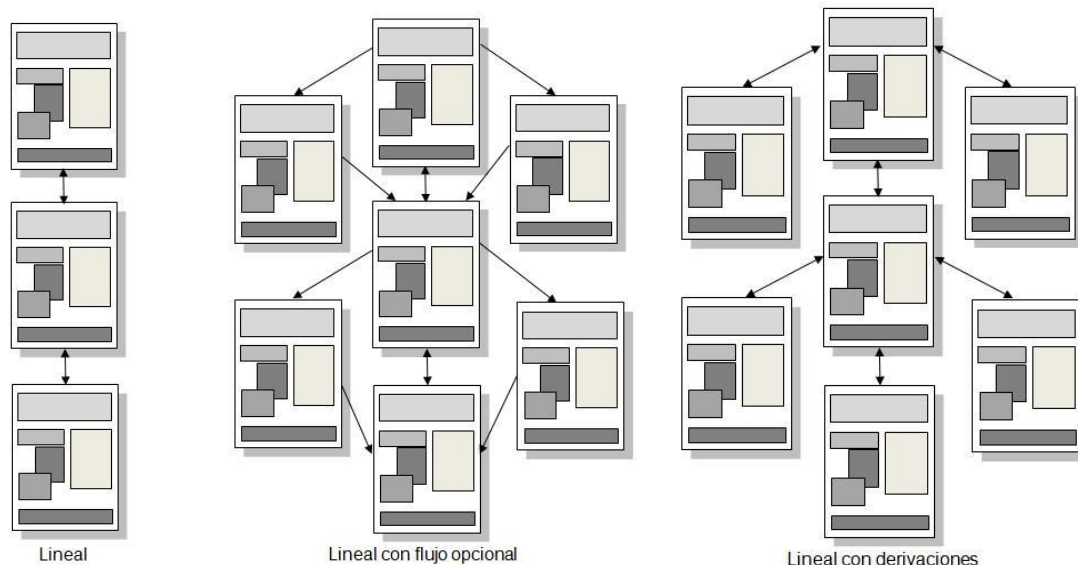
Arquitectura de contenido

El eje central del diseño de la arquitectura de contenido es la definición de la estructura hipertextual global de la *webapp*. Existen cuatro diferentes estructuras de contenido de entre las cuales el diseñador puede elegir⁸:

- Las estructuras lineales se encuentran cuando es común una secuencia predecible de interacciones.
- Las estructuras en retícula son una opción arquitectónica aplicable cuando el contenido de la *webapp* está organizado categóricamente en dos (o más) dimensiones.
- Las estructuras jerárquicas son indudablemente las arquitecturas *webapp* más comunes.
- Una estructura en red o “web pura” es similar en muchos sentidos a la arquitectura que evoluciona para los sistemas orientados a objetos.

Además de estas existen otras estructuras llamadas compuestas, las cuales son una combinación de una o más de las estructuras antes mencionadas.

8.- Powell, T., 2000



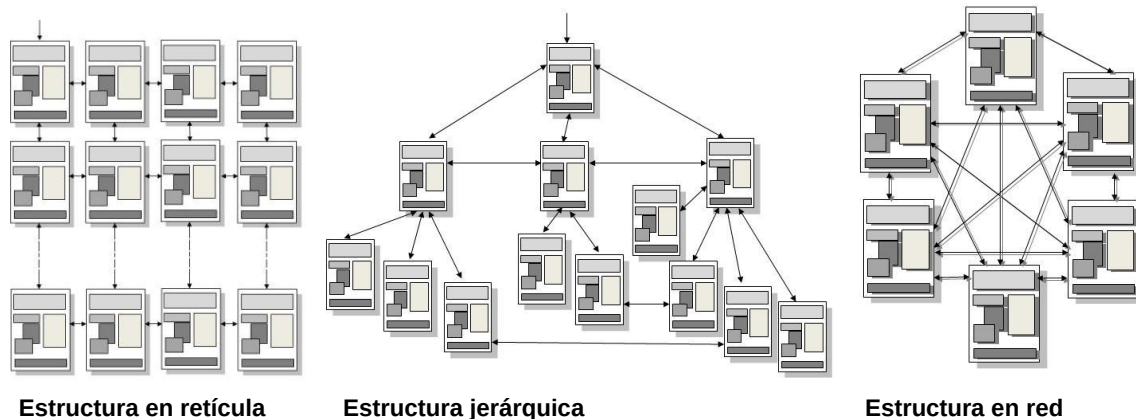


Ilustración 8 Diferentes tipos de estructuras

Diseño de navegación

Ya que se cuenta con la arquitectura de la *webapp* establecida además de la identificación de los componentes, el diseñador puede definir las rutas a habilitar que quiere establecer para que los usuarios puedan navegar a modo de llegar al contenido y a las funciones de la *webapp*. Una manera de llegar a esto es:

- 1) Identificar la semántica de navegación para diferentes usuarios del sitio.
- 2) Definir la mecánica (sintaxis) que logra la navegación.

Semántica de navegación

El diseño de navegación inicia con la consideración de la jerarquía de usuario y los casos de uso relacionados y desarrollados para cada categoría de usuario. Cada usuario tiene diferentes requisitos de navegación y por lo tanto usa la *webapp* de manera un poco diferente al resto de usuarios.

Los casos de uso desarrollados para cada categoría de usuario definen un conjunto de clases que abarcan uno o más objetos de contenido y funciones de la

webapp. Cada usuario tiene una diferente interacción con la *webapp* y conforme esta avanza se puede encontrar una serie de unidades semánticas de navegación.

Sintaxis de navegación

La mecánica de navegación es definida mientras se lleva a cabo el diseño; es posible elegir entre muchas opciones, entre las cuales se encuentran:

- Vínculo de navegación individual.
- Barra de navegación horizontal.
- Columna de navegación vertical.
- Pestañas.
- Mapas de sitio.

Entre las actividades del diseñador *web* se tiene que debe elegir los mecanismos más óptimos de navegación para la aplicación, establecer convenciones óptimas y auxiliares de navegación adecuados específicos a cada *webapp*.

Diseño a nivel de componente

Las aplicaciones *web* más actuales entregan funciones de procesamiento cada vez más elaboradas, las cuales:

1. Realizan procesamiento localizado para generar capacidad de contenido y navegación en una forma dinámica.
2. Ofrecen capacidades de computación o procesamiento de datos que son adecuadas para el dominio de negocios de la *webapp*.
3. Proporcionan cuestionamientos y accesos sofisticados a bases de datos.
4. Establecen interfaces de datos con sistemas corporativos externos.

Patrones de diseño

Los patrones de diseño aplicados en la ingeniería *web* están ubicados en dos clases:

1. Patrones de diseño genérico que son aplicables a todos los tipos de software.
2. Patrones de diseño hipermedia que son específicos de las *webapp*.

Existen pequeños problemas de diseño que se pueden adaptar a una variedad mucho más amplia de problemas específicos, los cuales pueden ser resueltos por los patrones de diseño que es un enfoque genérico para esta actividad. En el contexto de los sistemas basados en *web*, German y Cowan⁹ sugieren las siguientes categorías de patrones:

- Patrones arquitectónicos.
- Patrones de construcción de componentes.
- Patrones de navegación.
- Patrones de presentación.
- Patrones de interacción comportamiento/usuario.

4.3.1.4.- Desarrollo (Herramientas y tecnología)

En los últimos años contamos con un amplio conjunto de herramientas y tecnologías que han evolucionado a la par que las *webapps* se han vuelto más complejas y extendidas.

9.- German, D. y D. Cowan, junio de 2000

Estas tecnologías abarcan un amplio conjunto de descripción de contenido y lenguajes de modelación (*HTML*, *VRML*, *XML*, *XHTML*) lenguajes de programación (*Java*, *JavaScript*, *PHP*) recursos de desarrollo basados en componentes (*CORBA*, *COM*, *Activex*, *.NET*), navegadores, herramientas multimedia, herramientas de autoría de sitio, herramientas de conectividad de bases de datos, herramientas de seguridad, servidores y utilidades de servidor, herramientas de administración y análisis de sitio.

Existen algunas aplicaciones que podemos utilizar para redactar y generar código *CSS* y *HTML* para los sitios *web*. Inicialmente cualquier editor de texto plano, como el Bloc de notas de *Windows*, es suficiente para redactar código *HTML* y *CSS*, pero, es preferible recurrir a programas diseñados especialmente para tal fin ya que ellos cuentan con ayudas visuales (como el uso de distintos colores para el texto) que orientan en la tarea y simplifican el trabajo.

Notepad++ *Notepad++* es una aplicación que bien se podría definir como una versión mejorada del editor de texto o el Bloc de notas incluidos en todos los sistemas *Windows*.

Entre los principales puntos a favor de esta aplicación están:

- Es gratuita.
- Sintaxis coloreada.
- Numeración de línea.
- Muy liviana, consume pocos recursos del sistema.
- Código autocompletable en diferentes lenguajes.
- Es posible abrir y gestionar archivos de texto pesados sin perder estabilidad.

TopStyle pro *TopStyle Pro* es uno de los mejores editores de código *CSS* y, en general, es recomendable utilizarlo como complemento del editor *HTML* principal que se emplee para trabajar.

Entre sus principales ventajas se pueden mencionar:

- *Preview* del código escrito en tiempo real.
- Validador de estilos CSS.
- Sintaxis autocompletable del código que estamos escribiendo.

Lenguajes de ejecución del lado del usuario

Con los lenguajes de ejecución del lado del usuario se generan cierto tipo de sitios, los cuales cuando el usuario accede a una *webapp* trabajan de la siguiente manera: los archivos que la componen son descargados e interpretados directamente por el navegador (*Firefox, IE, Opera*, etc.). Pueden estar creados con *HTML* o *XHTML*, *JavaScript*, entre otros lenguajes.

HTML vs XHTML

La mayoría de los sitios *web* se crean con uno de estos dos lenguajes. Estas páginas se componen por una estructura de etiquetas que tienen significados semánticos para que los buscadores vinculen un determinado texto existente en la página al significado semántico que le da la etiqueta que lo contiene.

HTML: (*HyperText Markup Language*) Lenguaje de marcado de hipertexto. Un lenguaje de codificación utilizado para hacer páginas *web*. Es el lenguaje básico con el que se escribe la mayoría de las páginas *web*. Está compuesto por etiquetas delimitadas por paréntesis angulares (<,>), que describen la estructura y el contenido.

Cuenta con una serie de instrucciones que indican al navegador, que las interpreta, la manera en que la página debe ser representada y visualizada. Su característica principal es su simplicidad, solo es necesario el *WordPad* o Bloc de notas de *Windows* para estructurar una página *web* por medio de la inclusión de etiquetas (*tags*) propias del lenguaje.

Es un lenguaje, pero no de programación, sino de marcado. Se trata de un sistema de etiquetas que le indican al navegador que información y de qué manera, tiene que presentar al usuario final.

XHTML (Lenguaje de Marcado de Hipertexto Extensible) es una versión más estricta y limpia de *HTML*, que nace precisamente con el objetivo de remplazar a *HTML* ante su limitación de uso con las cada vez más abundantes herramientas basadas en *XML*. *XHTML* extiende *HTML* 4.0 combinando la sintaxis de *HTML*, diseñado para mostrar datos, con la de *XML*, diseñado para describir los datos.

Ante la llegada al mercado de un gran número de dispositivos, *XHTML* surge como el lenguaje cuyo etiquetado, más estricto que *HTML*, va a permitir una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella. *XHTML* puede incluir otros lenguajes como *MathML*, *SMIL* o *SVG*, al contrario que *HTML*.

XHTML, al estar orientado al uso de un etiquetado correcto, exige una serie de requisitos básicos a cumplir en lo que a código se refiere. Entre estos requisitos básicos se puede mencionar una estructuración coherente dentro del documento donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente, atributos de valores entrecomillados, etc.

Las principales ventajas del *XHTML* sobre el *HTML* son:

- Compatibilidad con navegadores antiguos.
- Reducción de errores.
- Facilidad de edición.
- Compatibilidad con estándares *web*.
- Se pueden incorporar elementos de distintos espacios de nombres *XML* (como *MathML* y *Scalable Vector Graphics*).

- Un navegador no necesita implementar heurísticas para detectar qué quiso poner el autor, por lo que el *parser* puede ser mucho más sencillo.
- Como es *XML* se pueden utilizar fácilmente herramientas creadas para procesamiento de documentos *XML* genéricos (editores, *XSLT*, etc.).

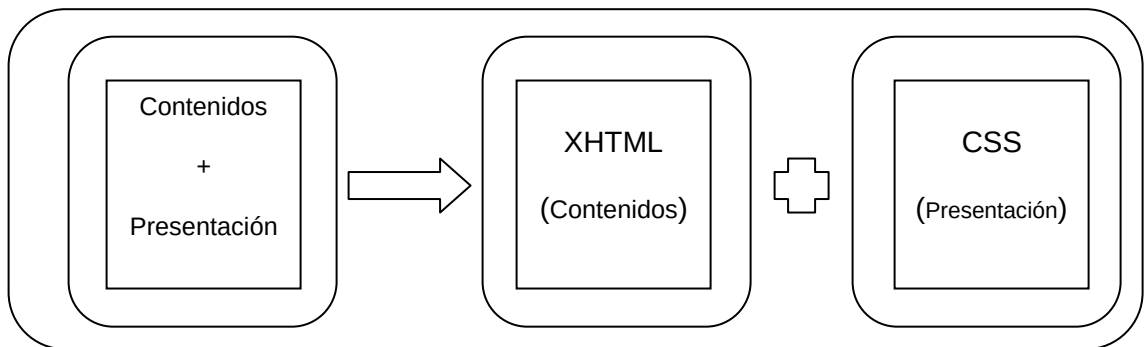


Ilustración 9 HTML incluye los contenidos y la forma de presentación. XHTML sólo trata con contenidos, que se representan según lo indicado en CSS

Lenguajes de ejecución del lado del servidor

Son lenguajes que ejecuta el servidor para crear una página cuando el usuario accede a un sitio. Es decir, buscan la información en una base de datos y, tomándola como referencia, generan archivos *HTML* que serán finalmente visualizados por el usuario.

PHP Es un lenguaje de programación al que pueden agregarse diversas librerías para otorgarle funcionalidades y usarse sin costo alguno. Es compatible con la mayoría de los servidores *web* y sistemas operativos, aunque suele implementarse en entornos *Linux* y con bases de datos *MySQL*, lo cual evita por completo los costos de licencias.

Es diseñado originalmente para la creación de páginas *web* dinámicas. Se usa principalmente para la interpretación del lado del servidor (*server-side scripting*) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la

creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas *Qt* o *GTK+*.

Hojas de estilo CSS

Hojas de Estilo en Cascada (*Cascading Style Sheets*), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS es el lenguaje creado para controlar el aspecto visual de los documentos *HTML*. Es la mejor forma de separar los contenidos y la presentación.

Se utiliza para dar estilo a documentos *HTML* y *XML*, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos *HTML* y *XML*. CSS permite a los desarrolladores *web* controlar el estilo y el formato de múltiples páginas *web* al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

Colores

Los colores que se manejen dentro de la *webapp* toman mucha importancia ya que al elegir los apropiados podemos lograr que los usuarios se interesen o no en nuestro sitio.

Es común hacer uso de una gama de colores armónicos, con variaciones de tonos de un mismo color. Con estos tonos se forma la paleta de colores del sitio *web*; hacer uso de la misma paleta es importante ya que esto evita que al final tengamos un sitio con muchos colores distintos.

Los selectores de colores, son una herramienta que permite tomar el color de una parte de la pantalla o de alguna página *web*. Existen muchas herramientas gratuitas de este tipo.

ColorPic Es un programa que permite seleccionar cualquier color de cualquiera de las ventanas, o en general de cualquier píxel de nuestro monitor. Una vez seleccionado nos muestra su valor *RGB* y nos permite almacenarlo en una paleta.

Imágenes

Es importante hacer uso de imágenes para dar una mejor perspectiva del trabajo y contenido del sitio, así como hacer una buena presentación del diseño. Para trabajar con imágenes debemos de emplear algún programa especializado que facilita esta tarea y entrega contenido de alta calidad, el cual se puede emplear para hacer más atractiva y funcional la *webapp*.

La variedad de programas que se pueden emplear para crear o editar las imágenes necesarias para la *webapp* es muy amplia, entre algunos de estos programas se pueden mencionar: *GIMP*, *Krita*, *Scribus*, *Paint.NET*, *Photoshop*, *Paint*, *SuperAlbum*, *Picozu*, *Inkscape*, *Aviary*, entre otros.

GIMP Es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Es un programa libre y gratuito. Forma parte del proyecto *GNU* y está disponible bajo la Licencia pública general de *GNU*. Es el programa de manipulación de gráficos disponible en más sistemas operativos (*Unix*, *GNU/Linux*, *FreeBSD*, *Solaris*, *Microsoft Windows* y *Mac OS X*, entre otros).

Inkscape Es un editor de gráficos vectoriales de código abierto, con capacidades similares a *Illustrator*, *Freehand*, *CorelDraw* o *Xara X*, usando el estándar de la *W3C*: el formato de archivo *Scalable Vector Graphics (SVG)*. Las características soportadas

incluyen: formas, trazos, texto, marcadores, clones, mezclas de canales alfa, transformaciones, gradientes, patrones y agrupamientos.

Inkscape también soporta *meta-datos Creative Commons*, edición de nodos, capas, operaciones complejas con trazos, “vectorización” de archivos gráficos, texto en trazos, alineación de textos, edición de *XML* directo y mucho más. Puede importar formatos como *Postscript*, *EPS*, *JPEG*, *PNG* y *TIFF* y exporta *PNG* así como muchos formatos basados en vectores.

Apis

Aunque si combinamos *HTML*, *JavaScript* y *PHP* podemos hacer muchas cosas con nuestras páginas, existen aún otras fuentes de datos que residen fuera de nuestra página y que podemos incorporar a nuestra *web* mediante las *APIs*.

Para añadir funcionalidades complejas a nuestra página *web* o para acceder a información, podemos comunicarnos con otro software, ofrecido por sitios *web* especializados, utilizando una Interfaz de Programación de Aplicaciones (API).

Lo que nos ofrece una API es una serie de procedimientos para acceder e interactuar con aplicaciones realizadas por terceros. Es decir, en vez de desarrollar esas funciones con un lenguaje de programación y almacenarlas en nuestro sitio, creamos una “*ventana*”, a través de la cual mostramos esa aplicación. Además, nos ofrece una serie de funciones, normalmente en *JavaScript* con las que podemos interactuar con esa ventana.

Internet nos ofrece una gran cantidad de añadidos que nos permiten ampliar la funcionalidad de nuestras páginas, muchos de ellos sin necesidad de tener demasiados conocimientos de programación ni de *HTML*. Podemos encontrar desde sitios que nos ofrecen un código fuente listo para copiar y pegar en nuestra *web*, hasta otros con *APIs* más avanzadas, que nos permiten una gran personalización de la aplicación.

Existen multitud de APIs con diferentes objetivos, por ejemplo, *Flickr* y *Panoramio* para usar sus amplios catálogos de imágenes, *Google Maps* para usar mapas de cualquier rincón del mundo, *Digg* y *Meneame* para comentarios y noticias, etc.

Estas y otras APIs se pueden utilizar para añadir ciertas funcionalidades a nuestra web, pero es tal la cantidad y calidad de APIs, que existen sitios web que se basan únicamente en las funciones y datos suministrados por las APIs y que son conocidas como "*Mashup*" o aplicaciones híbridas.

Google Maps Una API muy utilizada es la de *Google Maps*. Es un servidor de aplicaciones de mapas en la Web. Ofrece imágenes de mapas desplazables, así como fotos satelitales del mundo entero e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle.

JavaScript

Es un lenguaje de programación usado para generar interfaces de usuario mejoradas, con más interactividad. Las sentencias y funciones de este código pueden estar en el mismo archivo *HTML* o en un archivo externo.

Es un lenguaje interpretado, dialecto del estándar *ECMAScript*. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (*client-side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de *JavaScript* del lado del servidor (*Server-side JavaScript* o *SSJS*). Su uso en aplicaciones externas a la web, por ejemplo en documentos *PDF*, aplicaciones de escritorio (mayoritariamente *widgets*) es también significativo.

Se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación *Java*. Sin embargo *Java* y *JavaScript* no están relacionados y tienen semánticas y propósitos diferentes.

Navegadores web

Un *browser* o navegador es un programa de software usado para ver un sitio *web* y para navegar en Internet. Existen varios *browsers* disponibles gratuitamente para los usuarios, tal como *Internet Explorer*, *Mozilla Firefox* y *Ópera*. Algunos usuarios pueden también hacer uso de los navegadores de texto o de lectores de la pantalla para ver los sitios *web* que han sido diseñados específicamente para los usuarios con discapacidad visual.

	OPERA 1994. Es un navegador creado por la empresa noruega <i>Opera Software</i> . En la actualidad se encuentra en la versión 11.52.
	INTERNET EXPLORER 1995. Es el navegador desarrollado por <i>Microsoft</i> para <i>Windows</i> . Hoy en día existe la versión 9.0.
	SAFARI 2003. Es el navegador de <i>Mac</i> desarrollado por <i>Apple Inc.</i> Su versión más reciente es la 5.1.1.
	FIREFOX 2004. Navegador multiplataforma y de código abierto. Su última versión es la 8.0.
	GOOGLE CHROME 2008. Es el navegador desarrollado por <i>Google</i> . Es de código abierto y su última versión es la 5.0.375.86.

Tabla 1 Ejemplos de Navegadores

4.3.1.5.- Como probar aplicaciones web

Una vez que se tenga desarrollada la *webapp* es indispensable someterla a prueba, para ello es recomendable emplear una suma de actividades relacionadas, todas ellas con una sola meta en común: descubrir errores en el contenido, la función,

la facilidad de uso, la navegabilidad, el desempeño, la capacidad y la seguridad de la *webapp*.

Esto se puede hacer a lo largo de todo el proceso de ingeniería *web* mediante la aplicación de una estrategia de prueba que abarca tanto revisiones como pruebas ejecutables.

Pruebas de conceptos para *webapps*

Una buena consideración de las diversas dimensiones de la calidad de la *webapp* abarca la comprensión de los objetivos de las pruebas dentro del contexto de ingeniería *web*. En cualquier debate de las pruebas para el trabajo de ingeniería *web* se consideran las dimensiones de calidad que son particularmente relevantes. Estas pruebas arrojan ciertos errores de los cuales hay que considerar su naturaleza y la estrategia de poner a prueba aplicable para descubrir dichos errores.

Dimensiones de calidad

Un buen diseño proporciona como consecuencia el incorporar un alto nivel de calidad en la aplicación *web*. Esta calidad es evaluada al aplicar una serie de revisiones técnicas que valoran varios elementos del modelo de diseño y al aplicar un proceso de prueba. Tanto las revisiones como las pruebas examinan una o más de las siguientes dimensiones de calidad:

- El contenido se evalúa tanto en el ámbito sintáctico como semántico.
- La función se prueba para descubrir errores que indiquen que no hay concordancia con los requisitos del cliente.
- La estructura se valora para asegurarse de que entrega adecuadamente contenido y función de la *webapp*.
- La facilidad de uso se prueba para garantizar que a cada categoría de usuario la soporta la interfaz.

- La navegabilidad se pone a prueba para garantizar que toda la sintaxis y semántica de navegación se ejercen para descubrir cualquier error de navegación.
- El desempeño se prueba en una diversidad de condiciones operativas, configuraciones y cargas para asegurar que el sistema responde a la interacción del usuario y maneja cargas extremas sin que haya una degradación operativa inaceptable.
- La compatibilidad se prueba al ejecutar la *webapp* en varias configuraciones huésped, en los lados tanto del cliente como del servidor.
- La interoperabilidad se prueba para asegurar que la *webapp* realiza interfaces adecuadas con otras aplicaciones o bases de datos.
- La seguridad se prueba al valorar las vulnerabilidades potenciales e intentar explotar cada una de ellas.

Estrategias de pruebas

Los principios básicos que se aplican a todas las pruebas de software igualmente son aplicables como estrategia de prueba para las *webapps*. La estrategia para probar una *webapp* adopta los principios básicos para todas las pruebas de software. Los siguientes pasos resumen el enfoque:

1. Se revisa el modelo de contenido de la *webapp* para descubrir errores.
2. Se revisa el modelo de la interfaz para asegurarse que todos los casos de uso pueden acomodarse.
3. Se revisa el modelo de diseño de la *webapp* para descubrir errores de navegación.
4. Se prueba la interfaz de usuario para descubrir errores en la presentación o los mecanismos de navegación.
5. Componentes funcionales seleccionados se prueban en forma individual.
6. Se prueba la navegación a través de toda la arquitectura.
7. La *webapp* se implementa en diversas configuraciones ambientales y se prueba su compatibilidad con cada configuración.
8. Se realizan pruebas de seguridad con el objetivo de explotar vulnerabilidades en la *webapp* o dentro de su ambiente.
9. Se lleva a cabo pruebas de desempeño.
10. La *webapp* se prueba en una población controlada y monitoreada de usuarios finales.

Planeación de las pruebas

Un plan de pruebas para una *webapp* identifica:

- 1) Un conjunto de tareas que se aplicarán cuando comience la prueba.
- 2) Los productos de trabajo que se generarán conforme se ejecute cada tarea de prueba.
- 3) La forma en la que los resultados de las pruebas se evalúan, registran y reutilizan cuando se realicen pruebas de gestión.

El proceso de prueba

Las pruebas que ejercitan el contenido y la funcionalidad de la interfaz que es inmediatamente visible para los usuarios son el inicio de los procesos de prueba para la ingeniería *web*. A la par que se avanza con las pruebas, se ejercitan los aspectos de la arquitectura de diseño y la navegación.

En última instancia la atención pasa a las pruebas que ejercitan las capacidades tecnológicas que en pocas ocasiones son perceptibles para los usuarios finales: tales como la infraestructura de la *webapp* y las cuestiones de instalación/implementación.

Prueba del contenido

Este tipo de pruebas emplea una combinación tanto de revisiones como la generación de casos de prueba ejecutables. Esta revisión se aplica generalmente para encontrar errores semánticos en el contenido. Mientras que la prueba ejecutable es aprovechada para descubrir errores de contenido susceptibles de rastrear hacia contenido dinámicamente derivado, los cuales pudieron ser suministrados por los datos adquiridos de una o más bases de datos.

Objetivos de la prueba de contenido

La prueba de contenido tiene tres objetivos importantes:

- 1) Descubrir errores sintácticos en los documentos basados en texto, representaciones gráficas y otros medios audiovisuales.
- 2) Descubrir errores semánticos en cualquier objeto de contenido presentado conforme ocurra la navegación.
- 3) Hallar errores en la organización o estructura del contenido que se presenta al usuario final.

Prueba de la interfaz del usuario

Tanto la verificación como la validación de la interfaz del usuario de una *webapp* se llevan a cabo en tres puntos distintos durante el proceso de ingeniería *web*:

1. Durante la formulación y análisis de requisitos; que es cuando se revisa el modelo de la interfaz para de esta manera garantizar que se ajusta a los requisitos del cliente a la par de que lo hace con otros elementos del modelo de análisis.
2. Durante el diseño; que es cuando se revisa el modelo de diseño de la interfaz, esto para garantizar que se han alcanzado los criterios genéricos de calidad establecidos para todas las interfaces de usuario.
3. Durante las pruebas; el enfoque se cambia a la ejecución de los aspectos específicos de la aplicación de la interacción del usuario según se manifiestan mediante la sintaxis y la semántica de la interfaz.

Estrategia de prueba de la interfaz

La estrategia global para la prueba de la interfaz es:

- 1) Descubrir los errores relacionados con mecanismos específicos de la interfaz.
- 2) Descubrir los errores en la forma en que la interfaz implementa la semántica de navegación, la funcionalidad de la *webapp* o el despliegue de contenido.

El cumplimiento de esta estrategia requiere lograr los siguientes objetivos:

- Las características de la interfaz se prueban para asegurar que las reglas del diseño, la estética y el contenido visual relacionado están a disposición del usuario sin error alguno.
- Los mecanismos individuales de la interfaz se prueban en una forma análoga a la prueba unitaria.
- Cada mecanismo de la interfaz se prueba dentro del contexto de un caso de uso o USN para una categoría de usuario específica.
- La interfaz completa se prueba frente a los casos de uso y las USN seleccionadas para descubrir errores en su semántica.
- La interfaz se prueba dentro de una diversidad de ambientes (navegadores) para asegurar que será compatible.

Prueba de la semántica de la interfaz

El objetivo principal de la prueba de la semántica es: “evaluar cuán bien el diseño se ocupa de los usuarios, ofrecer una dirección clara, entregar retroalimentación y mantener consistencia de lenguaje y enfoque”

El equipo de ingeniería *web* debe mantener una lista de verificación mientras se prueba cada caso de uso, esto para asegurarse de que todo artículo del menú se ha ejercido al menos en una ocasión, además de verificar que todo vínculo anidado ha sido empleado. Se espera determinar si la *webapp* ofrece un efectivo manejo del error y recuperación.

Prueba de la facilidad de uso

Esta prueba es similar a la prueba de la semántica de la interfaz debido a que también evalúa el grado en el cual los usuarios interactúan de manera efectiva con la *webapp*, así como el grado de influencia que tiene la *webapp* para guiar las acciones de los usuarios, proporciona retroalimentación significativa y fortalece un enfoque de interacción consistente.

Las pruebas de facilidad de uso son diseñadas a modo que su resultado nos ayude a determinar el grado en el cual la interfaz de la *webapp* facilita las tareas de los usuarios.

Estas pruebas son diseñadas por un equipo de ingeniería *web*, pero son implementadas por los usuarios finales. Se aplica la siguiente secuencia de pasos¹⁰:

1. Definir un conjunto de categorías de prueba de facilidad de uso e identificar las metas para cada una.
2. Diseñar pruebas que permitirán evaluar cada meta.
3. Seleccionar los participantes que dirigirán las pruebas.
4. Instrumentar la interacción de los participantes con la *webapp* mientras se lleva a cabo la prueba.
5. Desarrollar un mecanismo para valorar la facilidad de uso de la *webapp*.

La Ilustración 10 muestra un posible conjunto de “calificaciones” de valoración que pueden seleccionar los usuarios como parte de las pruebas de la interfaz de usuario. Estas calificaciones se aplican a cada característica individualmente, a una página *web* completa o a la *webapp* como un todo dependiendo de las necesidades específicas de cada caso.

10.- Splaine, S. y S. Jaskiel, 2001

Fundamentos

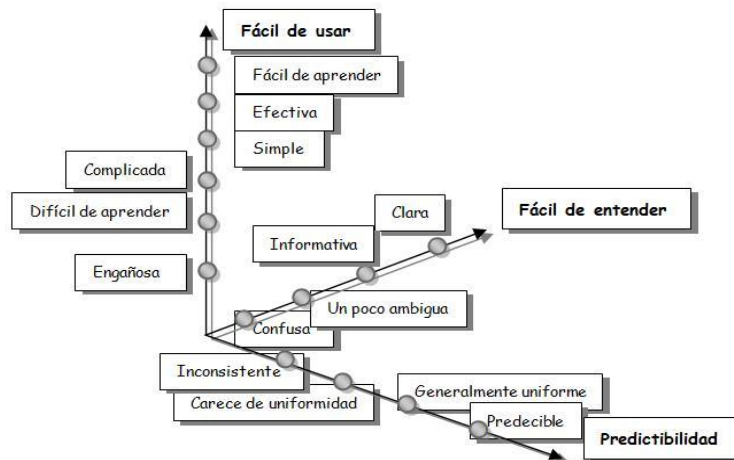


Ilustración 10 Valoración cualitativa de la facilidad de uso

Pruebas de compatibilidad

Para hacer uso de las *webapps* es necesario operarlas dentro de diferentes ambientes, cada uno con sus respectivas características. Estos ambientes pueden abarcar tanto diferentes computadoras, dispositivos de despliegue, sistemas operativos, navegadores y velocidad en las conexiones de red, estos ambientes tienen una influencia significativa en las velocidades de procesamiento en el lado del cliente, la resolución de despliegue y la velocidad de conexión.

Las diferencias que se encuentra entre los distintos ambientes pueden provocar conflictos en el procesamiento de la *webapp*. Dentro de la red de redes es normal que se perciba que los diferentes navegadores producen resultados ligeramente diferentes al recibir la misma solicitud, aun a pesar del estado de estandarización HTML dentro de la *webapp*.

Pruebas al nivel de componentes

Este conjunto de pruebas se enfoca en intentar descubrir errores en las funciones de la *webapp*.

Los casos de prueba al nivel de componentes con frecuencia se alimentan con entrada al nivel de formularios. Una vez definidos los datos de los formularios, el usuario selecciona un botón u otro mecanismo de control para iniciar la ejecución. Son comunes los siguientes métodos de diseño de casos de prueba:

- Partición de equivalencia.
- Análisis de valores límite.
- Pruebas de ruta.

Pruebas de navegación

En toda *webapp* existen varias rutas por las cuales puede navegar el usuario final, en muchos lugares se puede detener, mucha información que aprender y objetos q ver, actividades por iniciar, juegos que disfrutar, videos que criticar y decisiones que tomar.

El desarrollador espera que los usuarios sigan ciertas rutas, esto debido al diseño con el que cuenta la *webapp*, el conjunto de objetos que se les proporciona a los usuarios y los enlaces que se les proporcionan. Pero, no siempre se siguen estas rutas predeterminadas, ya que cada usuario es motivado por su propio interés de aprender más o acceder a otra información, lo cual nos da resultados diferentes al objetivo original. El trabajo de aprobar la navegación es:

- 1) Garantizar que todos los mecanismos que permiten al usuario de la *webapp* viajara a través de ella son funcionales.
- 2) Validar que cada unidad semántica de navegación (USN) pueda ser alcanzada por la categoría de usuaria adecuada.

Prueba de la sintaxis de navegación

Durante la prueba de la interfaz se inicia la primera fase de la prueba de navegación. Es necesario probar los mecanismos de navegación para asegurar que cada uno de estos realiza la función que le corresponde. Splaine y Jaskiel (Splaine, S. y S. Jaskiel, 2001) sugieren que se debe probar cada uno de los mecanismos de navegación siguientes:

- Vínculos de navegación.
- Redirigir.
- *Bookmarks*.
- Marcos y conjuntos de marcos.
- Mapas de sitio.
- Motores de búsqueda internos.

Pruebas de la configuración

Se tienen dos factores de suma importancia, los cuales logran que la ingeniería *web* sea todo un desafío; la variabilidad y la inestabilidad. Entre algunas cosas que son difíciles de predecir para cada usuario están: hardware, sistemas operativos, navegadores, capacidad de almacenamiento, rapidez de la comunicación de la red, además de una diversidad de otros factores del lado del cliente.

También se debe de considerar que la configuración para un usuario dado puede cambiar con regularidad. Todo esto puede producir como resultado, en el lado del cliente, un ambiente proclive a errores, tanto sutiles como significativos. Cada usuario tendrá una impresión diferente a la experiencia de otro usuario, en cuanto a la forma en la que interactúa con la *webapp*, si es que la configuración del lado del cliente en la que estén trabajando es diferente.

Conflictos en el lado del servidor

Trabajando en el lado del servidor; es fundamental verificar que la configuración de servidor proyectada puede soportar la *webapp* sin error, para esto se diseñan los casos de prueba de configuración. Para verificar hay que instalar la *webapp* dentro del ambiente del lado del servidor y se prueba con la intención de encontrar errores relacionados con la configuración.

Conflictos en el lado del cliente

En el lado del cliente las pruebas de configuración se centran principalmente en la compatibilidad de la *webapp* con las configuraciones que contienen una o más permutaciones de los siguientes componentes¹¹:

- *Hardware*. CPU, memoria, almacenamiento y dispositivos de impresión.
- Sistemas operativos. Linux, Macintosh, Microsoft Windows, algún sistema operativo con base móvil.
- Software de navegación. *Internet Explorer*, *Mozilla*, *Opera*, *Safari* y otros.
- Componentes de la interfaz de usuario. *AcxtiveX*, *applets Java* y otros.
- *Plug-ins*. *QuikTime*, *RealPlayer* y muchos otros.
- Conectividad. Cable, DSL, módem regular, T1.

A estos componentes se le pueden sumar algunas variables como: el software de red, las variaciones de ISP y aplicaciones que corren al mismo tiempo.

11.- Nguyen, H., Wiley, 2001

Pruebas de seguridad

Un buen paso y muy recomendable, es entender por completo la seguridad de las *webapps* antes de que se pueda lograr una prueba de seguridad efectiva ya que es una materia compleja. Los *hackers*, empleados descontentos, competidores deshonestos y cualquier otro que desee robar información sensible, modificar contenido maliciosamente, disminuir el desempeño, deshabilitar la funcionalidad o poner en apuros a una persona, organización o negocio, ven un blanco atractivo en las *webapps* y los ambientes en el lado del cliente y en el lado del servidor en los que están hospedados

La labor de quien prueba la seguridad es descubrir las debilidades que pueden explotar todas aquellas personas que tengan la intención de atacar los dominios susceptibles. El diseño de las pruebas de seguridad está enfocado en probar las vulnerabilidades en el ambiente del lado del cliente, las comunicaciones de red que ocurren mientras los datos pasan del cliente al servidor y de vuelta y el ambiente del lado del servidor.

Pruebas del desempeño

Los problemas de desempeño que se presentan debido a falta de recursos en el lado del servidor, ancho de banda de red inapropiado, capacidades inadecuadas de bases de datos, defectuosas o débiles capacidades del sistema operativo, funcionalidad *webapp* mal diseñada y otros conflictos de hardware y software que pueden conducir a un pobre desempeño cliente-servidor son detectados gracias a la aplicación de las pruebas de desempeño. Estas pruebas tienen dos finalidades:

1. Comprender como responde el sistema a la carga.
2. Recolectar métricas que conducirán a modificaciones de diseño para mejorar el desempeño.

4.3.1.6.- Entrega (Publicar)

Ya que está terminada la *webapp* el siguiente paso es publicarla en Internet para ponerla al alcance de los millones de internautas que existen en todo el mundo. Hoy en día es normal que cualquier persona del planeta pueda ver cualquier página *web* en Internet pero hasta la década de 1990 esto no era tan normal para nadie.

Tener acceso a un servidor de Internet y disponer de una forma de subir los archivos al servidor es lo mínimo básico que se necesita para publicar una *webapp*. Algunas cosas extras de las que se puede disponer son: conocimientos básicos sobre dominios y posicionamiento *web*.

Antes de llevar a cabo la publicación en internet es conveniente hacer pruebas en un servidor local instalado en la computadora en la cual se esté trabajando, esto es para comprobar las páginas dinámicas sin la necesidad de subirlas a *Internet* por ejemplo, mediante *WAMP* o *XAMPP* se puede instalar un entorno con el servidor *APACHE*, *PHP*, *MySQL* en *Windows*. Para *Linux* *XAMPP* o *LAMP*.

Servidores

Un servidor *web* es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. Si la *webapp* se encuentra almacenada en un servidor de Internet es posible que cualquier usuario de todo el mundo pueda verla.

Para tener un servidor de Internet, sólo hace falta un ordenador, una dirección *IP* fija, una conexión telefónica y un software adecuado, como el servidor *Apache*, que además es gratuito. Existen miles de servidores que comparten su información por el simple gusto de aprender y enseñar. Y muchos más que esperan hacer negocio en la red.

Crear un servidor tiene cierta complejidad, se necesita una conexión permanente a Internet y un ordenador con el servidor *web* en funcionamiento las 24 horas del día.

Existen dos alternativas para acceder a un servidor de Internet, buscar un servidor gratuito o contratar uno de pago. Dependiendo de los usos que se pretende dar a la *webapp* es como se toma la decisión de usar un servidor gratuito o uno de pago.

Servidores gratuitos. Hay que destacar que la mayoría de servidores gratuitos no admiten el uso de páginas dinámicas ni bases de datos.

Se pueden diferenciar dos alternativas gratuitas:

- Sitios especializados. Los servidores gratuitos pueden imponer restricciones en el uso de ciertas instrucciones de programación *web* y no te permiten cambiar la configuración del servidor.
- Proveedores de acceso. Cuando contratas un servicio de conexión a Internet a través de un proveedor, suelen ofrecer de forma gratuita un espacio *web* en sus servidores.

Servidores de pago. Son la opción si lo que se pretende es tener un lugar más profesional donde colocar las *webapps*; son servicios ofrecidos por empresas especializadas y su costo puede variar dependiendo de los servicios contratados.

Los servidores pueden ser de dos tipos básicos, compartidos o dedicados. Un servidor compartido (*shared hosting*) es aquel que comparte máquina física con otros servidores compartidos. En un servidor dedicado una máquina física soporta un solo servidor dedicado.

Es primordial que a la hora de contratar un servidor se tome en cuenta que es importante la velocidad con la que las páginas se cargan en *Internet* y que diferentes servidores ofrecen diferente velocidad de transferencia o ancho de banda. Algunos

servidores ofrecen "espacio ilimitado" pero en realidad quieren decir "espacio suficiente" para la mayor parte de los usuarios.

Dominios de Internet

Los dominios hay que contratarlos con un registrador autorizado por un periodo de tiempo determinado. Las empresas que ofrecen alojamiento también suelen ofrecer registro de dominios.

Los servidores gratuitos suelen ofrecer subdominios de la forma `http://www.servidor_gratuito.com/tu_subdominio`, de manera que no se necesita registrar ningún dominio ya que los subdominios no se registran.

Subir archivos al servidor

Durante el desarrollo de la *webapp* las pruebas se harán de manera local, en la misma computadora en la que se esté trabajando. Ya en la fase de entrega se utilizará un servidor, ya sea gratuito o de pago, en el cual habrá que subir todos los archivos de los cuales se compone la *webapp*.

Hay varias opciones para poder subir los archivos al servidor:

- a) Utilizar un programa específico para ello.
- b) Utilizar un editor de páginas *web* que incorpore esta funcionalidad.
- c) Utilizar las facilidades proporcionadas por la empresa de *hosting*.

Todas las empresas de *hosting* proporcionan una interfaz la cual ayuda a subir archivos al servidor. Los servidores gratuitos cuentan con una interfaz muy simple, con la cual solo permiten subir algunos archivos cada vez a diferencia de los de pago que ofrecen un mejor servicio. Algunos implementan un interfaz muy completo (*WebFTP*) que realmente puede sustituir al cliente *FTP*.

5.- Procedimiento

En la historia de la computación, uno de los sucesos más trascendentes es el impacto de los sistemas y aplicaciones basados en *web*. La importancia de las *webapps* crece con el paso de los años y esto provoca que se inicie una evolución de un enfoque *Iweb* disciplinado.

Las *webapps* cumplen con ciertas características que las hacen diferir con respecto de otras categorías de software informático; algunas de estas características: son eminentemente de red, las gobiernan los datos y se encuentran en evolución continua. Adicionalmente a estas podemos agregar: la inmediatez que dirige su desarrollo, la necesidad apremiante de seguridad en su operación y la demanda de estética, así como la entrega de contenido funcional.

Las *webapps* al igual que otros tipos de software, pueden valorarse mediante una diversidad de criterios de calidad, entre los cuales podemos mencionar: facilidad de uso, funcionalidad, confiabilidad, eficiencia, capacidad de mantenimiento, seguridad, disponibilidad, escalabilidad y tiempo para comercialización.

La *Iweb* se describe en tres estratos: procesos, métodos y herramientas/tecnología. El proceso *Iweb* adopta el enfoque de desarrollo ágil que subraya un punto de vista de ingeniería “magro”, riguroso, que conduce a la entrega incremental del sistema que será construido.

Para llevar a cabo cada proyecto de *Iweb* es necesario refinar y adaptar un conjunto de tareas provenientes de las actividades del marco de trabajo (comunicación, planeación, modelado, construcción y despliegue), las cuales se ejecutan generalmente en el desarrollo de la gran mayoría de los sistemas de información computacional.

Fases del diseño web

La vida de esta *webapp* abarca desde la recopilación de información inicial, pasando por la creación y llegando hasta el mantenimiento y actualización de la misma. Existen varios pasos en el diseño y desarrollo de los sitios *web*; según la opinión de un gran número de ingenieros *web*, los pasos mínimos necesarios para desarrollar y mantener una *webapp* son:

- Definición del proyecto.
- Estructura del sitio.
- Diseño.
- Desarrollo.
- Pruebas.
- Entrega.



Ilustración 11 Fases del diseño Web

5.1.- Definición del proyecto

Antes de comenzar, fue necesario comprender el problema que había que resolver: “... es necesario desarrollar un sitio dirigido a los integrantes del módulo de Computación Gráfica; en el cual se reunirá información pertinente al módulo como directorio de profesores, plan de estudios, inventario de materias y horarios entre otras cosas.” Para esto se elaboró un “*BRIEF CONCEPTUAL Y TÉCNICO*”.

BRIEF conceptual y técnico

Un *brief* es un documento en el que se especifican y definen todo tipo de información relacionada con el proyecto que se está por encarar. Se deben recopilar datos referidos al cliente y a sus objetivos, al mercado y al tipo de trabajo que se va a efectuar.

Acerca del cliente:

1.- ¿Cuál es el nombre de la compañía?

Módulo de Computación Gráfica de la FI, UNAM

2.- ¿A qué se dedica?

Ofrece una amplia diversidad de opciones para el desarrollo tanto profesional como académico en las áreas de:

- Visualización científica: médica, petrolera, astronómica, etc.
- Animación y producción digital.
- Desarrollo de videojuegos.
- Desarrollo de interfaces y realidad virtual.
- CAD y GIS.
- Entre otras.

3.- Describa la diferencia respecto de la competencia:

Al ser el módulo parte de la Facultad de Ingeniería y, esta a su vez de la UNAM, se ofrece a los alumnos una formación muy completa, tomando en cuenta aspectos diversos tales como el físico, cultural artístico, entre otros.

La Facultad de Ingeniería es una de las escuelas más importantes en el país y América Latina en las cuales se imparte la enseñanza de la ingeniería, por lo tanto los conocimientos generados e impartidos en ella son sólidos y bien fundamentados.

Basados en esto, el Módulo nos presenta una de las mejores opciones para aprender todo lo relacionado con la Computación Gráfica. Ya que la formación recibida aquí nos prepara como ingenieros y no como técnicos o simples usuarios.

4.- Acerca de la competencia: ¿Quiénes son? ¿Cuáles son las direcciones de sus sitios web?

Son todas aquellas escuelas privadas o de gobierno en las cuales se imparten materias de Computación Gráfica ya sean para estudios de licenciatura, especialidad, posgrado, diplomado o especialización; en estas escuelas se pueden tomar solo algunas materias o bien un conjunto de estas.

Algunos ejemplos son:

- WEFIS escuela de animación, http://www.wefis.com/es_index.htm
- Diplomado en animación, <http://www.3dpuppets.com/>
- Efectos visuales por computadora, <http://www.aztec-tech.com/>

Procedimiento

- Autodesk capacitación, <http://www.darco.com.mx/>

5.- Indique la fecha estimada en la que desee finalizar el proyecto:
Diciembre del 2011.

Funciones del sitio.

6.- ¿Ya cuenta con un sitio web? De no ser así ¿Necesita ayuda para elegir y adquirir un dominio?

No, actualmente no se cuenta con un sitio web. Por el momento solo tenemos un dominio tentativo pero ya llegado el momento se verá todo lo relacionado a él y su adquisición si es el caso.

7.- En orden de importancia ¿Cuáles son los objetivos de negocio del sitio web de la empresa?

- Poner al alcance de los usuarios la información específica sobre el Módulo de Computación Gráfica.
- Centralizar hipervínculos de forma organizada, para tener a la mano información de todas las ramas de la Computación Gráfica.
- Proporcionar material de apoyo académico a todos los visitantes.
- Fomentar el desarrollo de los alumnos en áreas de Computación Gráfica.
- Vincular a la industria con los alumnos del Módulo.

8.- ¿Para qué utilizarán los usuarios el sitio web?

Para buscar información que los ayude en su formación académica dentro de la especialidad así como para cubrir aspectos de Servicio Social, Titulación, Prácticas Profesionales, Vinculación con la industria entre otras.

Definición del público objetivo.

9.- ¿Quiénes cree que visitarán el sitio de la empresa? ¿Quiénes le gustaría que lo visitaran?

En un principio el sitio será visitado por alumnos inscritos en el Módulo de Computación Gráfica ya después se espera que llegue a ser visitado por todo el público interesado en temas de gráficos por computadora.

10.- ¿Cómo cree que los clientes perciben actualmente a la empresa?

Como una muy buena opción para desarrollar su carrera profesional y llegar a una especialización debido al gran crecimiento que ha tenido últimamente la computación generadora de gráficos.

11.- ¿Por qué la gente utilizaría el producto o servicio que ofrece la empresa?

Por ser la mejor opción en cuanto a contenido. Amplio, completo y actualizado, tanto en información directa del Módulo como de computación gráfica.

Preferencias técnicas.

12.- Describa las características técnicas necesarias para este proyecto:

Se espera que para la construcción del sitio se empleen lenguajes como HTML, XHTML, XML, PHP, SQL, JAVASCRIPT, PERL; herramientas para el desarrollo como las CSS e incluso algún editor de texto como NOTEPAD++; entre otras.

Procedimiento

Conceptos de diseño.

13.- ¿Cuenta la empresa con un desarrollo de logo o identidad realizado?

No, a la fecha aun no contamos con algún logo que nos represente.

14.- ¿Qué se quiere transmitir con el sitio web?

Principalmente la seriedad necesaria para respaldar la información proporcionada a los usuarios, tanto a la relacionada directamente con el Módulo como la de Computación Gráfica, aunado a los hipervínculos que proporcionamos para expandir la experiencia de navegación.

15.- ¿Tiene definido algún aspecto visual?

Se han trabajado en algunos borradores que servirán como base para definir un diseño visual agradable y acorde a las necesidades y requerimientos del portal.

Plan de trabajo

Ya que se identificaron las necesidades que propiciaron el desarrollo de la *webapp*, la descripción de los objetivos, las características y funciones sobresalientes de la misma; se estructuró un plan de trabajo en base al cual se llevó a cabo la definición del sitio.

- Recopilar la información que se presentará por medio de la *webapp*.
- Editar y clasificar la información recopilada en el punto anterior.
- En caso de que la información no esté en formato electrónico, habrá que digitalizarla.
- Una vez que se tenga este material listo habrá que pasar a la siguiente etapa del desarrollo (Estructura del sitio).

Material a recopilar por página

El primer paso de este plan de trabajo para hacer de la *webapp* un sitio exitoso fue la recopilación de información relevante. Se reunió todo aquel material que se quiere presentar a los usuarios finales (horarios, grupos, salones, material didáctico, información de proyectos, directorios, temarios, plan de estudios, etc.), la mayoría de esta información se obtuvo en formato electrónico y el resto tuvo un proceso de

digitalización; esta información fue evaluada y optimizada para que los usuarios finales obtengan sólo lo que realmente les sea de utilidad.

La información específica del portal como son los horarios, eventos, proyectos y directorio se tendrá que actualizar al menos cada inicio de semestre; el resto de la información se tendrá que estar actualizando constantemente para evitar tener (entre otras cosas) hipervínculos rotos o código desactualizado para no contar con un portal obsoleto.

A continuación se muestra una breve descripción del material recopilado:

Módulo

Definición: Se reunieron definiciones de computación gráfica para con base en ellas conseguir desarrollar una clara definición de lo que se realiza en el Módulo, además de esto se fundamentó en la definición que se tiene en el tríptico descriptivo del Módulo, editado por la Facultad.

Directorio: Se llenó el formulario 001, mediante entrevistas directas con cada uno de los profesores que componen la plantilla del Módulo o bien mediante información proporcionada por la misma Facultad.

Horarios: Esta información se obtuvo de la página de la Facultad de Ingeniería, UNAM; sección horarios.

Eventos: Se llenó el formulario 002 por medio de entrevistas con el coordinador del Módulo además de las mismas aportaciones de la comunidad.

Plan de estudios: Se descargó el documento Plan de Estudios de la página de la Facultad de Ingeniería, al cual se le hicieron modificaciones mediante *HTML* (empleando un mapa de imágenes) para agregar los botones correspondientes a las

materias pertenecientes al módulo, además de que éstas se agregaron en el orden recomendado para tomarlas según los administradores del Módulo.

Material didáctico

En una primera etapa se descargaron los archivos *PDF* (temarios) de cada una de las seis materias que componen el Módulo; con éstos se trabajaron las páginas *HTML* de la sección “Material didáctico”.

En una segunda etapa se ingresaron a las páginas *HTML*: definiciones, imágenes, videos y en los casos que sea necesario código. Esta información puede ser modificada (acrecentándola o rectificándola) por los integrantes del Módulo ya sean académicos o estudiantes y el público en general mediante la sección aportaciones.

Enlaces

Se recopilaron un total de 250 enlaces pertenecientes a igual número de sitios *web* cuya información está directamente relacionada con los temas tratados en el Módulo de Computación Gráfica; cada uno con imagen, dirección *web* y una breve descripción del contenido de cada sitio, estos se dividen en 19 temas que son:

Aplicaciones de Computación Gráfica, Geometría Computacional, Videojuegos, Empresas de Videojuegos, Diseño Asistido por Computadora, Estudios de Animación, Fotorrealismo, Granjas de Rendering, Algoritmos Gráficos, Escuelas de Animación, Modelado Geométrico, Animación, Procesamiento Digital de Imágenes, Diseño de Interfaces, Escuelas de Videojuegos, Ambientes Gráficos Interactivos, Realidad Virtual, Multimedia.

Proyectos

Contando con el apoyo del coordinador y los profesores del Módulo se llenó el formulario 003 para recopilar información para esta sección.

Demo reel

Se creó un *demo-reel* para ingresarlo a esta sección a modo de prueba (demostrativo), además de llenar el formulario 004 el cual tendrá que ser llenado por cada usuario que pretenda participar en esta sección, con esta actividad se podrá dar inicio a el proceso de convocatoria necesario para recopilar material perteneciente a esta sección.

Materias

Se recopilaron definiciones de cada una de las ramas de la computación gráfica que componen nuestro Módulo, con esto se generó una definición propia clara y concisa, además se almacenaron imágenes y videos relacionados con estas materias.

Buscador

Se creó una cuenta en *G-mail*, la cual será manejada por el administrador del sitio, esto para poder hacer uso de la función *Google Custom Search*, además de emplear la lista de 250 hipervínculos recopilados para la sección enlaces; con esta información y la *api* se creó el buscador personalizado para el portal del Módulo.

Contáctanos

Se necesitan los datos de la persona que quedará como administrador del portal, estos datos son: nombre completo, cargo o nombramiento, correo electrónico, número de teléfono fijo (oficina o cubículo) y ubicación dentro de Ciudad Universitaria. Esto para que sea a esta persona a quien le lleguen los correos provenientes de esta sección para que así pueda darles pronta respuesta o seguimiento según sea el caso.

Noticias

Se recopilaron las noticias más relevantes en cuanto a temas relacionados con la computación gráfica para tener preparada esta sección a modo de ser mostrada al público objetivo, esta información (noticias) en adelante se estará actualizando con cierta periodicidad.

Aportaciones

Emplearemos los mismos datos que en la sección contáctanos, con la diferencia, que la información que provenga de esta sección tendrá que ser evaluada por el administrador apoyado por el coordinador y los mismos profesores del Módulo, esto con el fin de que la información pase por un filtro y sea agregada solo al aprobar el control de calidad y así asegurar que se ofrece lo mejor a los usuarios finales.

Formularios:

001

Nivel académico: _____
Nombre completo: _____
Cargo: _____
Correo electrónico: _____
Página web: _____
Especialidad: _____
Teléfono: _____
Ubicación en CU: _____

002

Título: _____
Fecha: _____
Horario: _____
Lugar: _____

003

Título: _____
Encargado: _____
Área: _____
Informes: _____

004

Nombre: _____
Especialidad: _____
Correo electrónico: _____
Teléfono: _____

Editar y clasificar la información recopilada

El material que se encontraba ya en formato electrónico se clasificó y almacenó según las necesidades del portal web, para su posterior uso; dependiendo de que se tratara, por ejemplo: imágenes, botones, archivos *PDF*, folletos electrónicos, etc.

Con toda esta información se generaron bloques los cuales fueron depurados para sólo trabajar con información valiosa para no tener excesos y evitar trabajar sin sentido; por ejemplo, en la sección de enlaces se obtuvieron un promedio de treinta enlaces por cada uno de los tópicos, se verificó que realmente la información contenida (contenido de la página) fuera del tema en el que se encontraba clasificada, de ser así pasó el filtro y de lo contrario era desechada.

En el caso de la sección Módulo el material son los horarios, directorios, grupos y eventos; se confirmó que esta información no fuera falsa, pero con cada inicio de semestre se tiene que actualizar a diferencia de la definición del Módulo y los planes de estudio ya que no tendrá cambios en el mismo periodo por lo que su actualización no será tan seguido.

Además se crearon doce carpetas para separar y ordenar esta información:

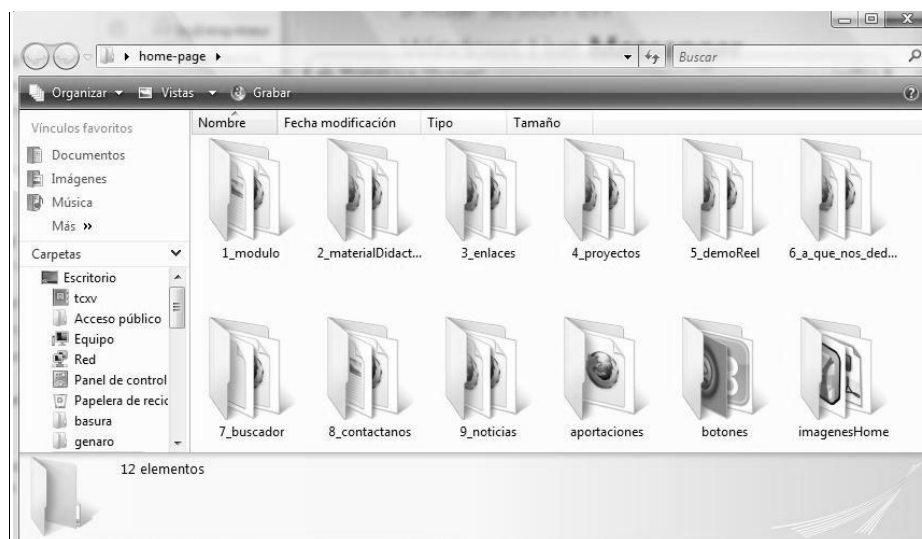


Ilustración 12 Ordenamiento y clasificación

En cada una de estas carpetas se ingreso el material (imágenes, videos, folletos, etc.) perteneciente a las diferentes secciones que componen el portal *web*.

Digitalizar material

Fue necesario convertir datos y documentos existentes en formato *web*. Esto para poder disponer de esta información con mayor facilidad durante el desarrollo del portal. Se transcribieron datos haciendo uso de un editor de textos planos para evitar incluir metadatos que pudieran causar algún contratiempo en cuanto al manejo de los mismos.

5.2.- Estructura del sitio

Jerarquía de usuario

Las categorías de usuarios reconocidas para el portal *web* son: usuario, visitante y administrador. Usuario es toda aquella persona la cual hace uso de la *webapp*; visitante son todos aquellos que visitan la *webapp* para obtener información concerniente al Módulo o las materias que lo integran y por último el administrador es aquel usuario que se encarga de tener actualizada la información contenida en el portal y de dar mantenimiento a este.

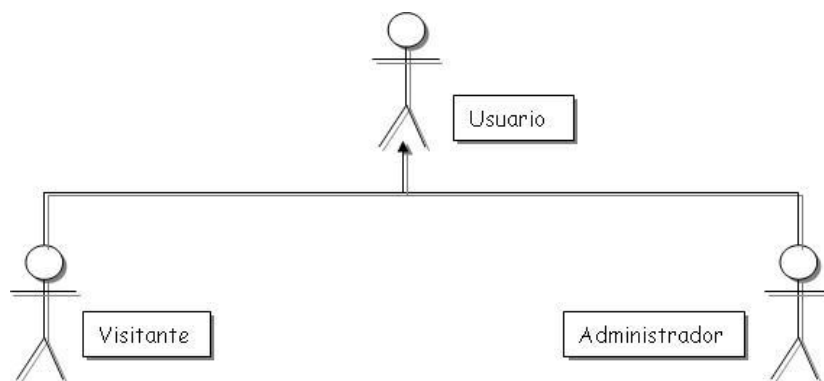


Ilustración 13 Jerarquía de usuarios del portal web del MCG

Casos de uso

Se desarrolló un caso de uso por cada categoría de usuario planteada en la jerarquía de usuarios. Se redactó un párrafo narrativo que describe una interacción específica entre cada usuario y la *webapp*. Por lo tanto se obtuvieron tres casos de uso, una para usuario otro para visitante y el ultimo para administrador.

Usuario: Son todos aquellos que despliegan la webapp en sus exploradores ya sea sólo para buscar algún tipo particular de información (haciendo funciones de visitante) o bien para administrar (editar, modificar, agregar) el contenido del portal.

Visitante: Es aquel usuario que entra al portal en busca de información ya sea del Módulo, de las materias con las que cuenta este (material didáctico), o los tópicos de interés (tema central, la Computación Grafica). Puede inspeccionar todo el contenido de la *webapp*.

Administrador: Se le nombra así a aquel usuario que al interactuar con la *webapp* modifica su contenido ya sea con actualizaciones de información como son los horarios, directorio, eventos, noticias, etc., modificaciones en cuanto al diseño gráfico con el que cuenta el portal e incluso su estructura.

Modelado de análisis para *webapps*

Al finalizar estas tareas se cuenta con cierta información, la cual se tiene que revisar, modificar (si se requiere) y organizar en un modelo a modo que pueda ser empleada de la mejor manera en el desarrollo del diseño de la *webapp*.

El modelo de contenido

Se verificó que se tiene el material necesario para presentar la información contenida en la *webapp*. Este material se incrementará conforme vaya creciendo el

portal, cada vez se tendrán más imágenes, videos, códigos, incluso de ser posible en un futuro se podrá incrementar el número de secciones que componen el portal.

Constantemente se monitoreará la *web* para encontrar noticias relativas a los tema de interés, algunas imágenes o fotografías, animaciones, videos, presentaciones; estos objetos de contenido también podrán ser proporcionados por los alumnos de la facultad o el público en general, todo aquel usuario que tenga interés en aportar algo para que la *webapp* siga creciendo.

El modelo de interacción

Una manera de contemplar la interacción se representa dentro del contexto del modelo de navegación. Existen seis diferentes estructuras de contenido de entre las cuales se puede elegir:

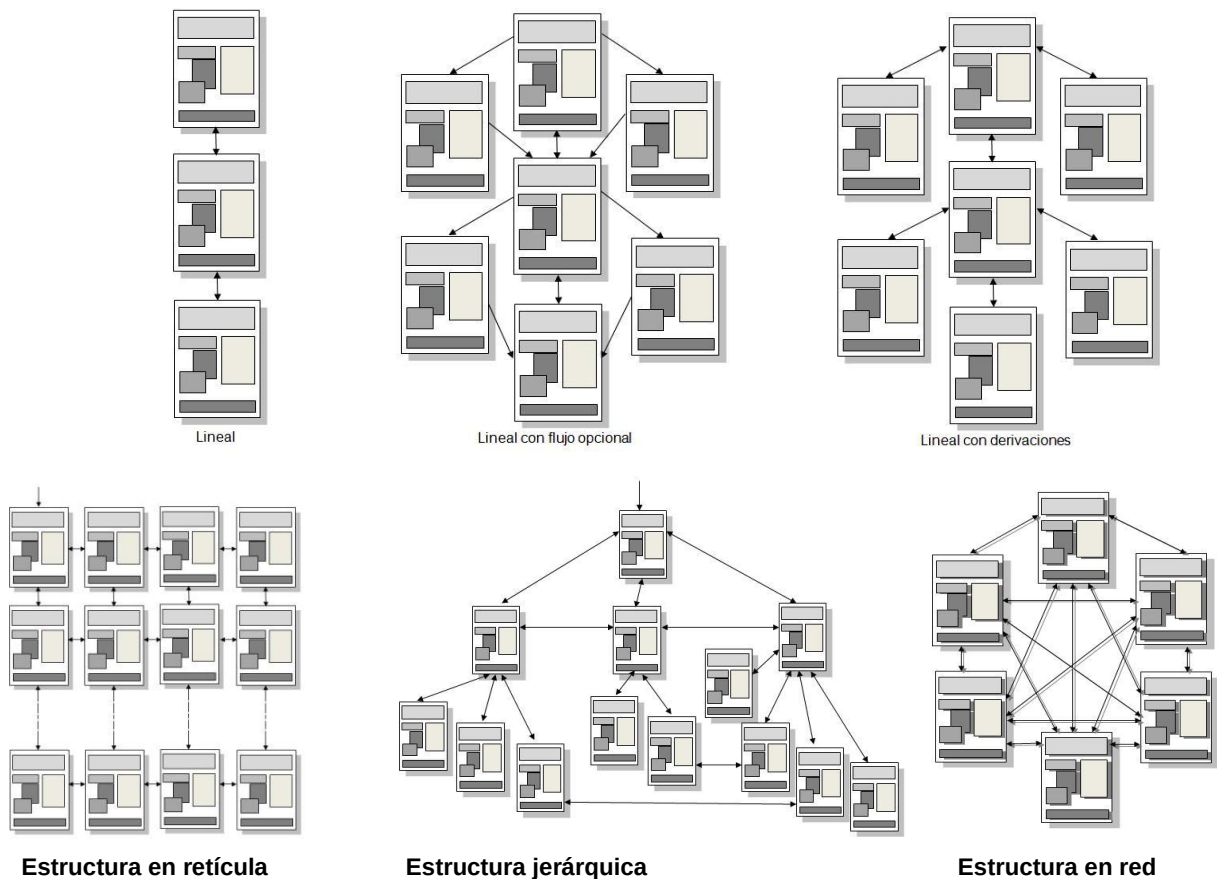


Ilustración 14 Diferentes tipos de estructuras

Con base en esto se presenta la manera en la cual los usuarios interactuarán directamente con el contenido de la *webapp*, la navegación difícilmente podrá ser lineal, ya que en un mismo caso de uso el usuario puede navegar por diferentes secciones, así que la estructura que se acomoda a las necesidades de este portal tiene que ser una de retícula para facilitar la navegación.

Estructura del portal (*HOME-PAGE*)

Para estructurar el portal primero habría que conocerlo, si bien no se contaba con el número exacto de páginas que lo integrarían si se tenía una noción o idea general del número total de éstas.

Para esto se realizó una lista en forma jerárquica de las secciones que lo componen:

Home-Page

Módulo

Definición

Directorio

Horarios

Eventos

Plan de estudios

Material didáctico

Computación Gráfica¹

Computación Gráfica Avanzada¹

Diseño de Interfaces, Multimedia y Realidad Virtual¹

Diseño asistido por Computadora¹

Procesamiento Digital de Imágenes¹

Temas Selectos de Graficación¹

Enlaces

1.- En estas páginas se podrá descargar el archivo PDF que contiene cada uno de los temarios de las materias; cada uno de los temas será un hipervínculo a una página en la que encontraremos las secciones:
Definición-Concepto, Imágenes, Videos, Código fuente

Procedimiento

Aplicaciones de Computación Gráfica

Geometría Computacional

Videojuegos.

Empresas de Videojuegos

Diseño Asistido por Computadora

Estudios de Animación

Fotorrealismo

Granjas de Rendering

Algoritmos Gráficos

Escuelas de Animación

Modelado Geométrico

Animación

Procesamiento Digital de Imágenes

Diseño de Interfaces

Escuelas de Videojuegos

Ambientes Gráficos Interactivos

Realidad Virtual

Multimedia Proyectos

Demo-reel

Materias

Computación Gráfica

Computación Gráfica Avanzada

Diseño de Interfaces, Multimedia y Realidad Virtual

Diseño asistido por Computadora

Procesamiento Digital de Imágenes

Temas Selectos de Graficación

Buscador

Contáctanos

Noticias

Aportaciones

Con base en lo anterior se pudo asegurar que el portal constaría con al menos 80 páginas en la primera versión y se procurará estar en continuo crecimiento.

Estructura y navegación

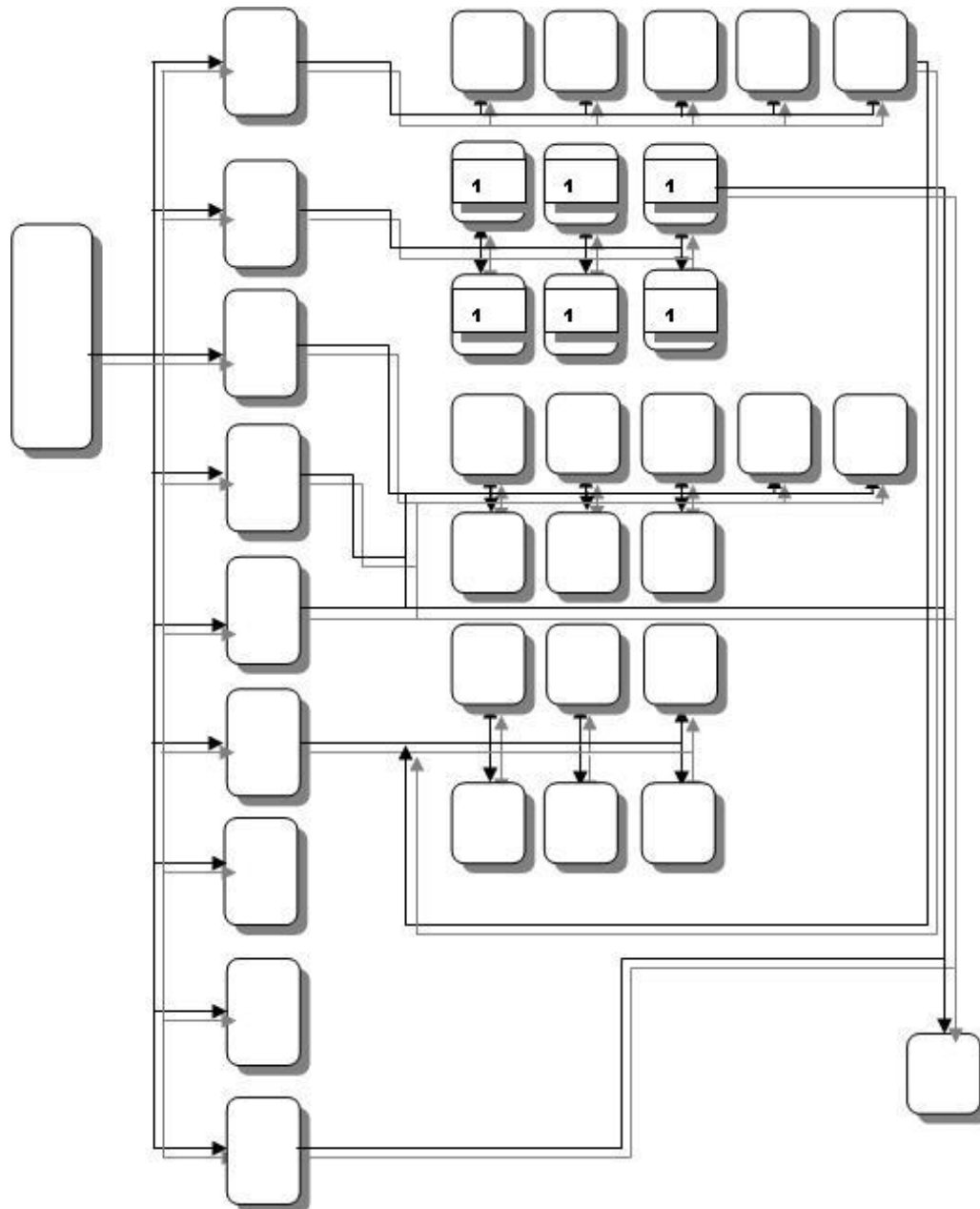


Ilustración 15 Estructura y navegación del portal

1.- En estas páginas se podrá descargar el archivo PDF que contiene cada uno de los temarios de las materias; cada uno de los temas será un hipervínculo a una página en la que encontraremos las secciones:

Definición-Concepto, Imágenes, Videos, Código fuente.

El modelo funcional

Entre las operaciones que se aplicarán al contenido de la *webapp* solo se menciona el exponer (mostrar) la información que se tiene almacenada, bien se puede asegurar que la *webapp* es meramente demostrativa, esto al menos en esta primera etapa del desarrollo.

Existe un grupo de funcionalidades propias del portal que no tienen relación directa con la información mostrada a los usuarios (contenido), pero no por ello son menos importantes, ya que su inclusión en la *webapp* proporciona una diferencia ya sea en su diseño o en su función.

Entre estas funciones están las aplicaciones para mostrar el contenido (código CSS), aplicaciones para establecer la comunicación entre los usuarios y el administrador del sitio (código PHP), las aplicaciones para mostrar el reloj y el calendario (código JavaScript), entre otras.

El modelo de configuración

La configuración de la *webapp* esta apta para residir en una amplia diversidad de ambientes, tanto del lado del cliente como del servidor; esto a pesar que aun con los intentos por estandarizar los ambientes del lado del cliente y del servidor, no se ha llegado a conseguir la estandarización en su totalidad.

La *webapp* está preparada para residir en un servidor que proporcione acceso vía *Internet*, *Intranet* o bien *Extranet*, pero esta optimizada para los servidores con acceso vía *Internet*.

En una segunda etapa de desarrollo/evolución de la *webapp*, se manejará un nivel de interoperabilidad mayor del lado del servidor ya que se establecerá la

comunicación con bases de datos, e incluso se puede esperar interoperar con aplicaciones ya existentes del lado del servidor.

Al desarrollar la aplicación *web* se tomó en cuenta que los usuarios finales manejan diferentes aplicaciones para mostrar el contenido del portal; cada una de estas tienen su propia infraestructura la cual permite el acceso de los usuarios a la *webapp* desde cualquier ubicación con conexión a *Internet*.

Entre las aplicaciones útiles para mostrar la *webapp* encontramos: *Internet Explorer, Firefox, Opera, Google Chrome, Safari, etc.*; aunque en su mayoría estos navegadores manejan ciertas características que establecen los estándares necesarios para unificar el desarrollo *web*, estas no se cumplen por completo provocando que la misma aplicación se vea diferente en los diferentes navegadores.

Se manejan pruebas aplicadas a la *webapp* para proporcionar la información necesaria con la cual se verá el portal casi igual en todos los navegadores sin importar con cual decida trabajar el usuario final; estas pruebas solo se pueden manejar mediante prueba y error, por lo cual fue necesario tener instalados varios navegadores en la computadora de desarrollo.

5.3.- Diseño para aplicaciones *web*

Para el desarrollo del diseño de la *webapp* se tomó en cuenta las nuevas características de los navegadores, la evolución en la velocidad de las conexiones y el desarrollo de nuevos dispositivos de navegación (tabletas, celulares, *PDA* y consolas de videojuegos, etc.).

En el diseño *web* propuesto para el portal fue necesario contemplar el desarrollarlo en base al lenguaje *XHTML* junto con *CSS* para separar el diseño del

contenido, esto debido a la necesidad de reproducir los contenidos en los nuevos dispositivos y los avances constantes en los estándares *web*.

El uso de *Flash* ha sido relegado a sitios de animación y al uso en elementos como reproductores de video o *banners*, por lo que parte del desarrollo de este portal fue hecho en *JavaScript*.

Lo primero que se hizo fue crear un modelo de diseño para poder iniciar la construcción de la *webapp*, no se tomó este diseño como final ya que la idea fundamental es que la *webapp* irá evolucionando conforme se avanza en el desarrollo de la aplicación y se aclaran dudas sobre los requerimientos de esta.

El diseño no contempló solamente actividades artísticas, sino también técnicas. Las actividades artísticas (diseño gráfico) son las que se encargaron del aspecto visual y la plantilla estética de la interfaz de usuario quedó a cargo del diseño de la interfaz, así como el modelado de la estructura técnica de la *webapp* fue por parte del diseño arquitectónico y de navegación.

El diseño gráfico fue de suma importancia ya que ofreció un modelo que sirvió de guía para la construcción de la *webapp* (maquetado). El modelo de diseño contó con la información necesaria para interpretar los requisitos de los usuarios, con esto se procuró construir y poner en marcha la *webapp* para de esta manera poder ponerla a prueba y así obtener retroalimentación.

Se contemplan algunas características mínimas necesarias con las que el portal tiene que cumplir (metas del diseño), las cuales se tomaron en consideración para cumplir con los estándares establecidos por la 3WC:

- Simplicidad.
- Consistencia.

Procedimiento

- Identidad.
- Robustez.
- Navegabilidad.
- Apariencia visual.
- Compatibilidad.

Para el buen desarrollo de este portal, el diseño fue basado en el esquema de la pirámide de diseño para ingeniería *web* que se contempló en el capítulo anterior.

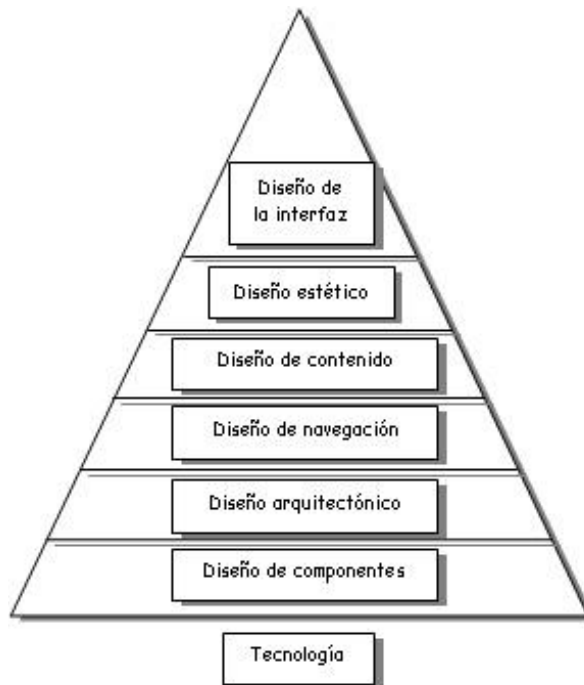


Ilustración 16 Pirámide del diseño web

Diseño de la interfaz

Se diseñó la interfaz a modo que en todas las posibles pantallas aparecieran los botones (enlaces) en las mismas posiciones, procurando que con esto se obtenga una *webapp* consistente, la cual sea fácil de usar (intuitiva), clara en su manera de navegar y funcional ya que con sólo unos *clicks* se llega a la información buscada.

Mecanismos de control de la interfaz

Esta *webapp* cumple los objetivos ya pactados; puesto la ventana que se establece como modelo es consistente con el contenido del portal, ayuda a los usuarios para que interactúen con la aplicación, organiza y proporciona el contenido disponible además de organizar las opciones de navegación.

Con el fin de propiciar la interacción entre el usuario y la interfaz se emplearon tres técnicas que permiten aprender el uso de la interfaz:



Ilustración 17 Mecanismos de control de la interfaz

Diseño estético

El diseño *web* se compone de ciertas características particulares, al tomarlas en consideración se pudo obtener como resultado una interfaz estéticamente agradable y funcional.

La primer forma en la que se pretende atraer y conservar a usuarios nuevos, es mediante el diseño gráfico que maneja el portal (ya después será por el contenido y funcionalidad del mismo). Para tener un buen diseño gráfico fue necesario considerar aspectos como presentación y percepción de la *webapp*.

Se inició el proceso de diseño con la creación de la plantilla, se continuó con la toma de decisiones considerables para este diseño como son los esquemas de color (teoría del color); tipos, tamaños y estilos de fuentes además de los medios audiovisuales.

La plantilla para la *webapp*

Se diseñó y desarrolló una plantilla para la *webapp*, esto para tener el aspecto estético y atractivo de la misma, además de contemplar las características de navegación y funcionalidad.

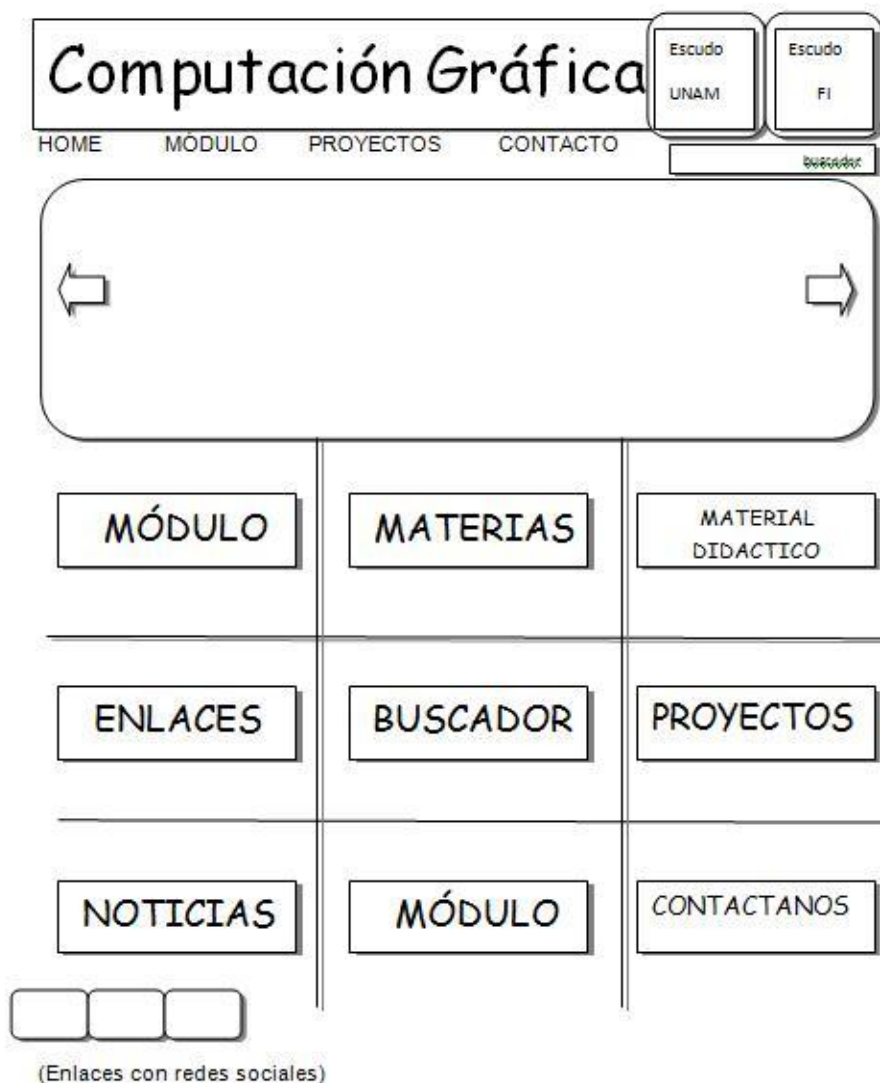


Ilustración 18 La plantilla para la webapp

Al momento de crear el diseño del portal, fue conveniente tener presentes algunas recomendaciones (lineamientos generales de plantilla) que harán que la *webapp* cumpla su objetivo:

No temerle al espacio vacío

Un buen consejo para el diseño visual es: “evitar llenar todos los espacios.” El ojo necesita tener un descanso y no se debe bombardear con un sitio que tenga imágenes en todos lados ya que el usuario no sabría por dónde comenzar a mirar.

“Los espacios con pocas cosas aparentan ser más elegantes, llaman más la atención.” Basándose en esta frase se consideró utilizar sólo contenidos relevantes para que los usuarios hagan foco en ellos.

Zonas calientes

Jakob Nielsen, Ingeniero de interfaces, realizó investigaciones sobre los contenidos más vistos dentro de un sitio web. Concluyó en denominar como forma térmica a la presentación de los contenidos, marcando las zonas más vistas como zonas calientes. Utilizó colores rojo y amarillo para indicar lo más vistoso y azul y gris para lo menos visto. Como resultado, las zonas más vistas forman una especie de letra F o triángulo sobre la estructura general del sitio, lo que confirma que las mejores ubicaciones para los contenidos importantes son esas.



Ilustración 19 Zonas calientes

Comprensión inmediata

Se procuró proporcionar la información que ofrece el portal de forma sencilla y rápida, esto con el objetivo de cautivar al usuario a manera que permanezca en la *webapp*; al no hacerlo se podría correr el riesgo de que el usuario cierre el navegador y no experimente ni conozca el contenido de la *webapp*.

El punto de tensión

Según estudios de comunicación visual: “toda pieza comunicacional presenta diversos elementos, niveles de lectura y jerarquías.” Así que aprovechando esto, se generaron relaciones entre estos componentes para guiar al usuario y llevarlo por la información que se le quiere mostrar, proporcionando puntos de tensión para hacer foco en estos elementos. Algunos de los recursos que se utilizaron para trazar el recorrido son:

- Contraste.
- Color.
- Elementos visuales.
- Posición.

Una forma de indicarle al usuario el recorrido que debe hacer dentro del portal fue empleando los elementos gráficos.

Grilla

Es un elemento que está cobrando importancia en el desarrollo *web*, todo diseño presupone una grilla constructiva. Este tipo de grilla es muy usado en medios gráficos en donde es necesario organizar la información contenida.

Procedimiento

La grilla constructiva permitió organizar la información que se proporciona mediante la *webapp*.



Ilustración 20 Grilla

Colores

Los diseñadores gráficos profesionales emplean el color como un arma muy poderosa que provoca sensaciones y tiene un efecto psicológico en la conducta de la gente. Así bien tomando en consideración esto para el diseño de este portal se emplearon colores que refuerzan la sensación de pertenencia, puesto los colores base que se utilizaron son el rojo y blanco, colores propios de la Facultad de Ingeniería.

Estética, temática y target

Al establecer que el portal es del módulo de Computación Gráfica se tiene ya el target de usuarios a los que va dirigido; por lo que el diseño fue enfocado en esta temática, así mismo se emplearon videos y gráficas generadas por computadora, a modo que todo el contenido quede acorde con el eje principal de la temática, la Computación Gráfica.

El diseño (estético) es subjetivo, el concepto “belleza” es diferente para cada persona, lo que para unos es hermoso puede que para otros no lo sea y viceversa; pero existen algunas pautas que están muy arraigadas en la sociedad y que ayudaron a tomar algunas decisiones en cuanto al diseño.

La estructura de las páginas

Se estableció cierta disposición de los elementos mínimos necesarios que aparecerán en todas las páginas del portal, para de esta manera dar al usuario la coherencia necesaria para que busque siempre estos elementos en el mismo lugar dentro de las diferentes secciones.

A fin de mantener cierta coherencia visual a la distribución del contenido, se diseñó un estándar de estructura el cual se cumple para todas las secciones que componen la *webapp*. Este estándar está basado en el estándar *web* de la estructura, el cual cuenta con: encabezado o *header*, contenido, barra lateral o *sidebar* y pie de página o *footer*.

Encabezado o *header*

El encabezado es el aspecto principal que se colocó en el sitio web ya que es lo primero que se espera verán los usuarios. Se colocó a todo lo ancho de la parte superior del sitio y se mantiene sin ninguna variación en todas las secciones, contiene: logo, menú, reloj, fecha, *links* a las páginas UNAM y FI, además del buscador interno.



Ilustración 21 Encabezado

Contenido

Justo aquí se colocó el contenido principal, el cual estará cambiando dependiendo de la sección en la que se encuentre el usuario. Esta parte de la *webapp* no se mantiene igual a lo largo de todo el portal por lo que no habrá dos iguales; es por esto que se hicieron bocetos de todas las páginas que forman parte del portal.

Se jerarquizaron los diferentes tipos de información. Se crearon puntos de tensión basados en el uso de destacados, títulos (resaltados mediante el uso de fondos), iconos, *API's* e imágenes y se complementó la información con texto y *links* siempre buscando entregar la información justa que el usuario está buscando.

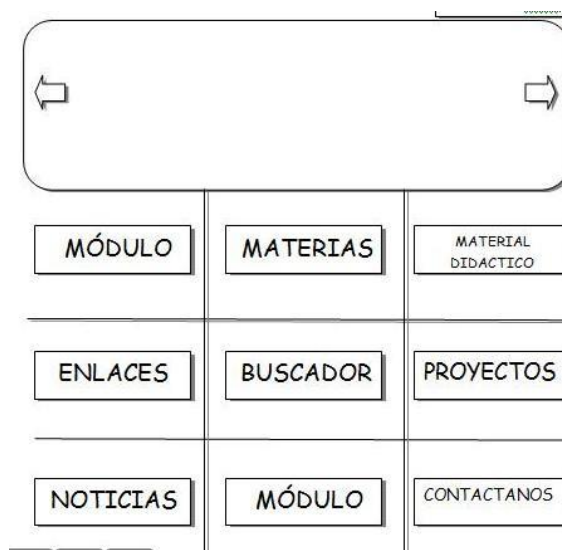


Ilustración 22 Cuerpo

Pie de página o footer

El diseño se realizó de manera austera buscando la simplicidad para esta sección, conforme se avanzó en el desarrollo se hicieron adaptaciones necesarias para aprovechar al máximo este espacio libre.

En un principio se agregaron sólo los botones necesarios para tener vínculos con las principales redes sociales: *Twitter*, *facebook*, *Youtube* y *Reevon*. Se reservó el

Procedimiento

espacio necesario para agregar enlaces destacados en cuanto al desarrollo y mapas de sitio (para ayudar a la indexación en Google), además de información como políticas de privacidad de uso y derechos de autor.



Ilustración 23 Pie de página

Estos elementos son los componentes necesarios de una interfaz web. De igual manera los contenidos, la forma de navegación, los elementos de identificación y las acciones que se pueden generar dentro de un sitio son parte de su interfaz. Al crear una interfaz se deben respetar las convenciones de estructura.

Bocetos del portal

Se crearon los bocetos de las posibles páginas que forman parte de la *webapp*:

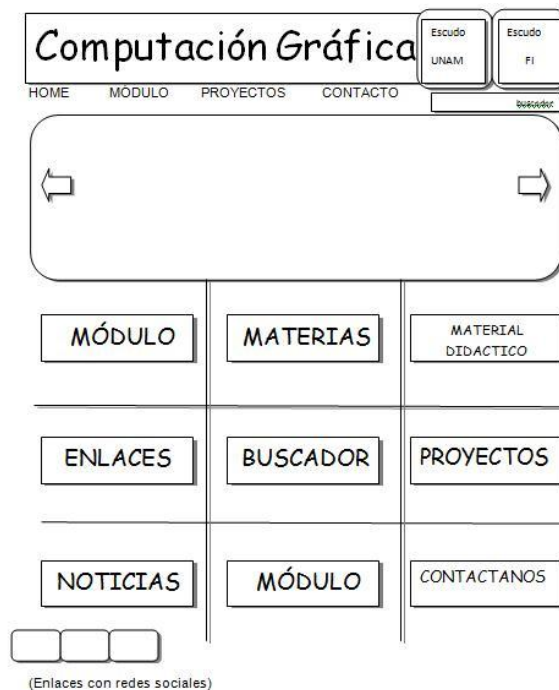


Ilustración 24 Boceto Página principal

Procedimiento

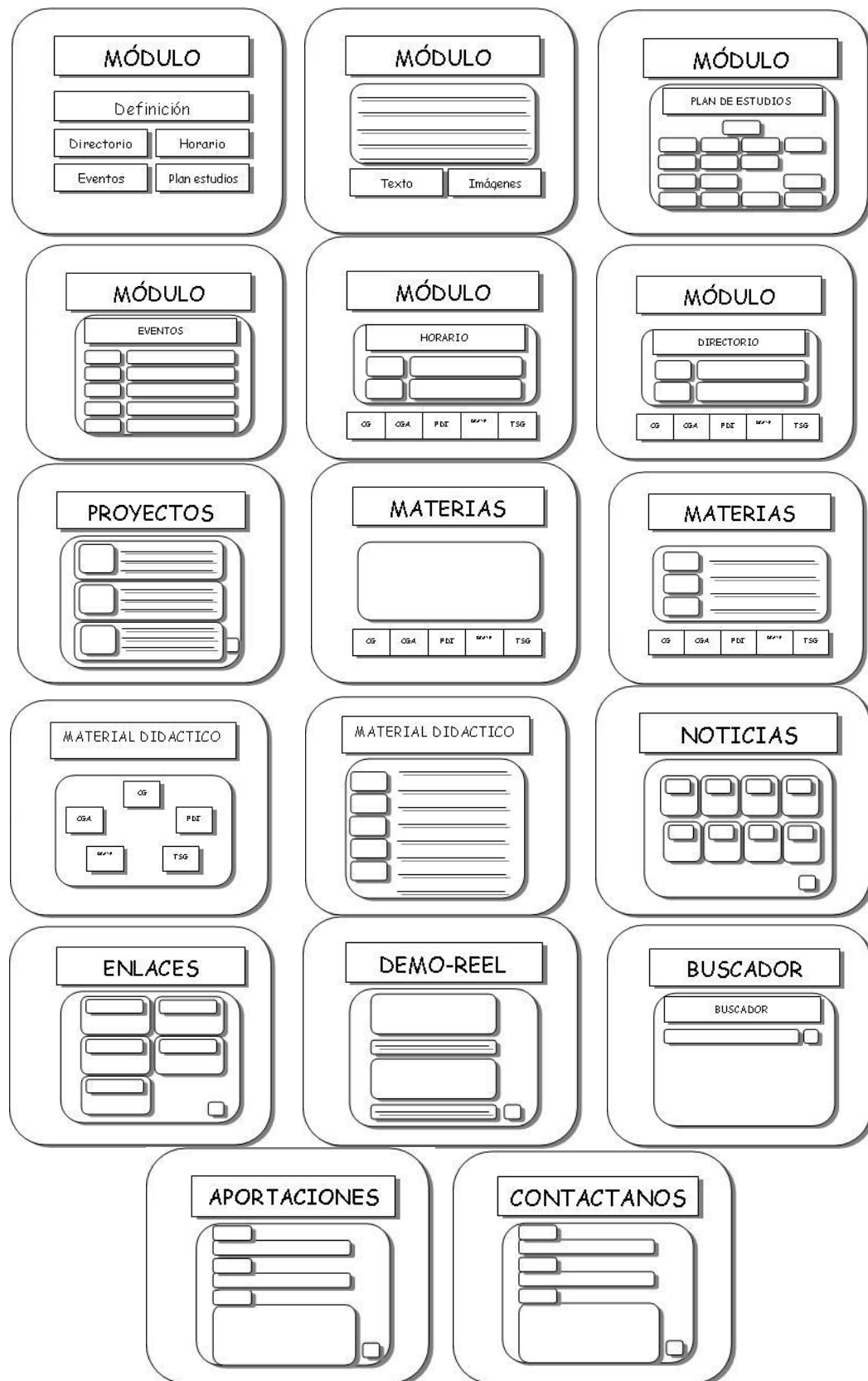


Ilustración 25 Bocetos portal

Diseño del contenido

Se diseñó la *webapp* a manera que el contenido con el que cuenta esta, se pueda proporcionar y compartir de manera fácil con todos los usuarios que visiten el portal.

Se trabajó sobre los componentes del sitio para que quedaran con ciertas características (estándares) y así proporcionar contenido similar entre páginas de la misma sección.

La *webapp* fue diseñada para que este en constante crecimiento y no llegue a estancarse, tampoco tiene un determinado número de componentes permitidos. Por ejemplo en la sección enlaces, hay ciertas pestañas en las cuales solo se cuenta con tres o cuatros enlaces a diferencia de otras cuyo promedio es de veinte elementos; la sección *Demo-reel* comenzó tan solo con tres videos (demostrativos), pero gracias a la convocatoria se espera que este número crezca conforme más compañeros hagan sus aportaciones.

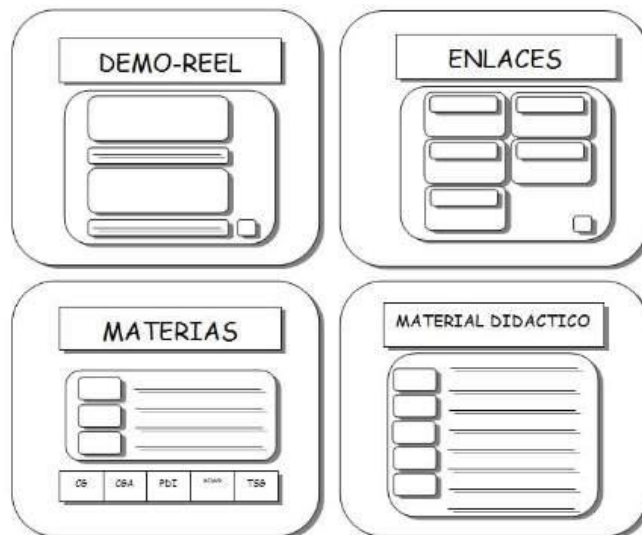


Ilustración 26 Diseño del Contenido

Diseño de navegación

Ya que se tuvieron identificados los componentes y establecida la arquitectura de la *webapp*, se habilitaron las rutas con las cuales los usuarios pueden navegar por toda la *webapp* a modo de llegar al contenido y a las funciones de su interés.

Cada usuario se maneja de diferente manera al ingresar al portal, incluso el mismo usuario puede tener un diferente comportamiento de una visita a otra; por esto fue necesario que el diseño de la navegación contemplara la mayoría de los casos de uso posibles.

A la par que se llevó a cabo el diseño de la interfaz se hizo el diseño de navegación, para que de esta manera no se intentaran poner enlaces donde estéticamente no queden o bien para que estos queden acorde al entorno en el cual se están ingresando.

Es necesario proporcionar al usuario un mecanismo por el cual pueda encontrar de forma rápida y sencilla lo que está buscando, la solución fue entregar un menú de navegación claro y conciso; un objetivo del diseño web es proporcionar al usuario final toda la información sin que este tenga que hacer demasiados *clicks*.

Los *links* del menú de navegación son representados gráficamente por botones, por lo que se procuró que fueran fáciles de localizar e interpretar bien puede ser por gráficos que denoten fácilmente de que se trata el mencionado botón, a donde nos puede llevar o que información nos proporcionará; en estos casos lo obvio y predecible fue lo más recomendado.

De entre la lista de mecanismos de navegación se eligieron los que mejor se acoplaban a la *webapp* para ser implementados y proporcionar un buen nivel de navegación:

DESARROLLO DEL PORTAL WEB DEL MÓDULO DE COMPUTACIÓN GRÁFICA DE LA FACULTAD DE INGENIERÍA, UNAM

Procedimiento

- Vínculo de navegación individual.
- Barra de navegación horizontal.
- Columna de navegación vertical.
- Pestañas.
- Mapas de sitio.

Vínculo de navegación individual

MÓDULO	MATERIAS	MATERIAL DIDACTICO
ENLACES	BUSCADOR	PROYECTOS
NOTICIAS	MÓDULO	CONTACTANOS

Ilustración 27 Navegación de la webapp

Barra de navegación horizontal

CG	CGA	PDI	DIMYR	TSG
----	-----	-----	-------	-----

Ilustración 28 Barra de navegación horizontal

Pestañas



Ilustración 29 Pestañas

Mapa del sitio

Módulo	Proyectos
Definición	Demo real
Texto	Maratón
Imágenes	CG
Videos	Texto
Horarios	Imágenes
Horario CG	Videos
Horario COA	COA
Horario PDI	Texto
Horario DAC	Imágenes
Horario DDMyRV	Videos
Horario TSG	PDI
Directorios	Texto
Directorio CG	Imágenes
Directorio COA	Videos
Directorio PDI	DAC
Directorio DAC	Texto
Directorio DDMyRV	Imágenes
Directorio TSG	Videos
Eventos	DDMyRV
Plan de estudios	Texto
Material didáctico	Imágenes
Índice CG	Videos
Índice COA	TSG
Índice PDI	Texto
Índice DAC	Imágenes
Índice DDMyRV	Videos
Índice TSG	Buscador
Enlaces	Contactanos
Aplicaciones de computación	Noticias
gráfica	Ingeniería
Geometría computacional	UNAM
Videojuegos	Redes sociales
Empresas de videojuegos	Twitter
Diseño asistido por computadora	Facebook
Estudios de animación	Youtube
Fotorealismo	Razon
Oranjas de rendering	Aviso privacidad
Algoritmos gráficos	Mapa del sitio
Escuelas de animación	XHTML 1.0
Modelado geométrico	CSS 2
Animación	Mapa del sitio
Procesamiento digital de imágenes	
Diseño de interfaces	
Escuelas de videojuegos	
Ambientes gráficos interactivos	
Realidad virtual	
Multimedia	
Otros	

Ilustración 30 Mapa del sitio

Diseño a nivel de componentes

Tomando en cuenta que en la actualidad las *webapp* entregan funciones de procesamiento cada vez más elaboradas, se diseñó este portal a modo que pudiera cumplir con el hecho de realizar procesamiento localizado para generar capacidad de contenido y navegación en una forma dinámica.

Esto se puede ver en algunas de las pestañas como: Módulo, *DemoReel*, Noticias, Enlaces, Enlaces, etc.:



Ilustración 33 Diseño componentes

5.4.- Desarrollo

La mayoría de los sitios *web* son creados ya sea con *HTML* o *XHTML*. Se eligió escribir este portal en *XHTML* por las ventajas que presenta ante *HTML* además de cumplir con la estandarización y permitir una correcta interpretación de la información independientemente del dispositivo desde el que se accede a la *webapp*.

Aunado a este lenguaje de marcado se hizo uso de las Hojas de Estilo en Cascada (*Cascading Style Sheets*), para describir cómo se va a mostrar la *webapp* en la pantalla del usuario final. Para esto fue necesario separar los contenidos de la presentación.

El desarrollo se inició con la estructura básica de la página web, enseguida se procedió al maquetado de la *webapp*, para esto se escribió el código de la plantilla (que

servirá como base para el resto de las páginas del portal), tomando en cuenta el boceto de la página principal del sitio.

Estructura básica de la página: cabecera (*head*) y cuerpo (*body*)

Según su estructura, las páginas web se dividen en dos partes: la cabecera (*head*) y el cuerpo (*body*). En la cabecera se determina el tipo de documento y sus propiedades además de algunas características no visibles para el usuario final, en el cuerpo se entregan a la página los elementos que el usuario puede ver en pantalla (texto, imágenes, videos, animaciones, etc.).

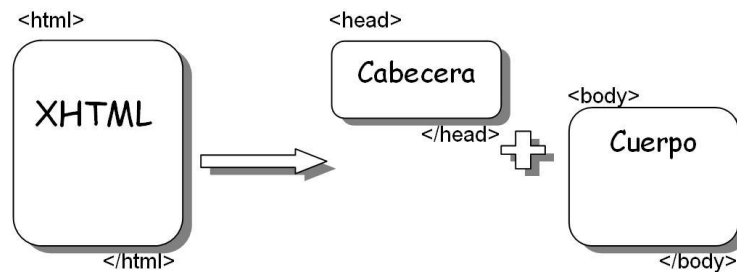


Ilustración 34 Estructura Básica de una página web

Para escribir el código de la cabecera se necesitaron siete etiquetas principales, presentes en casi todas las páginas web: *DOCTYPE*, *html*, *title*, *meta DESCRIPTION*, *meta KEYWORDS*, *link* y *link externo*.

Código de un encabezado básico:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.tdt">
<html>
<head>
<title> Título de la Página Web </title>
<meta name="DESCRIPTION" content="Descripción mínima de la Página Web que debe
contener preferentemente alrededor de 70 caracteres.">
<meta name="KEYWORDS" content="palabras clave que identifican a la webapp, separadas
por comas">
<link href="estilos.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="script.js"></script></head>
```

Código de un cuerpo básico:

```
<body>  
    Aquí se le indica al navegador que elementos debe incluir (texto, imágenes, videos,  
    animaciones, etc.) y de qué manera hacerlo (organización en el espacio de la página).  
</body>
```

Este código fue la base para programar la plantilla utilizada en el desarrollo del portal.

Maquetado del sitio web

Se estructuró el sitio de manera que se pudiera llevar a cabo el maquetado XHTML y la aplicación de estilos mediante CSS. Se muestra una estructura básica de un sitio estándar, la cual fue empleada para desarrollar la plantilla del portal, las variaciones fueron mínimas y se muestran más adelante.

Estructura básica de un sitio:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html>  
<head>  
    <title> Título de la Página Web </title>  
    <meta name="DESCRIPTION" content="Descripción mínima de la Página Web que debe contener  
    preferentemente alrededor de 70 caracteres.">  
    <meta name="KEYWORDS" content="palabras clave que identifican a la webapp, separadas por comas">  
    <link href="estilos.css" rel="stylesheet" type="text/css" />  
    <script type="text/javascript" src="script.js"></script>  
</head>  
<body>  
    <div id="wrapper">  
        <div id="header" >        header    </div>  
        <div id="nav" >          nav        </div>  
        <div id="menu" >         menu       </div>  
        <div id="content" >      content    </div>  
        <div id="sidebar" >      sidebar    </div>  
        <div id="footer" >      footer     </div>  
    </div>  
</body>  
</html>
```

Procedimiento

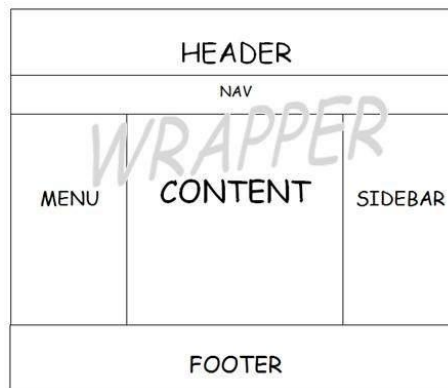


Ilustración 35 Maquetado de un sitio web

Para iniciar con el maquetado de la webapp se generó un contenedor general para todo el sitio. A este contenedor se le nombró *wrapper* y se le dieron ciertos parámetros, esto para facilitar el manejo del diseño mediante CSS; tomando en cuenta el boceto de la página principal del portal *web* se dividió el *wrapper* en tres *layouts*: *header*, *content* y *footer*.

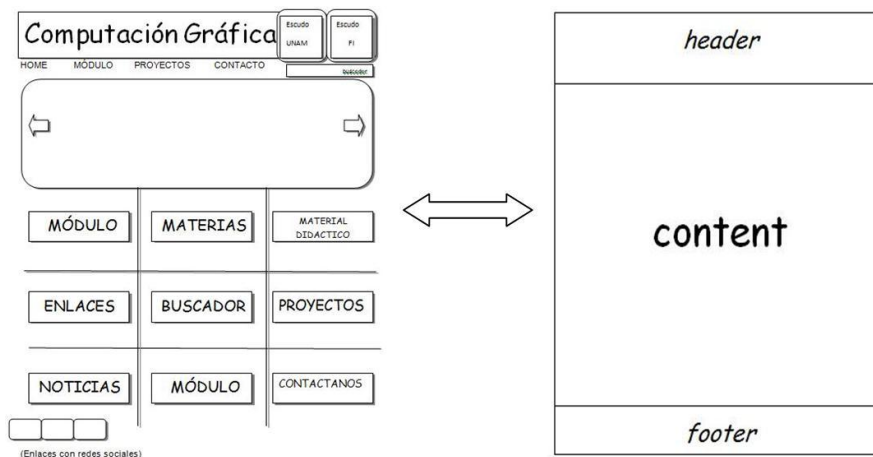


Ilustración 36 Maquetado del portal web

Aplicando las variaciones antes mencionadas el código de la plantilla quedó:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
    <title> Módulo de Computación Gráfica de la FI </title>
```

Procedimiento

```
<meta name="DESCRIPTION" content="Portal Educativo de computacion grafica, puerta de entrada al mundo del 3d, Tutoriales, apuntes de Animaci?n ">
<meta name="KEYWORDS" content="UNAM, Universidad Nacional Autónoma de México, Animacion, Animacion de personajes, character animation, educaci?n, ense?anza ">
<link href="estilos.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="script.js"></script>

</head>

<body>
  <div id="wrapper">
    <div id="header" >    header    </div>
    <div id="content" >    content    </div>
    <div id="footer" >    footer    </div>
  </div>
</body>

</html>
```

El *header* se ubica en la parte superior del sitio, en este caso particular contiene el logotipo del Módulo de Computación Gráfica, un par de *links* externos (uno para la página principal de la UNAM y el otro para la de la Facultad de Ingeniería), el buscador interno del portal, el menú principal además de que se agregó la fecha y hora mediante el uso de *JavaScript*.

Parte del código del *header* quedó:

```
<div id="header">
  <ul class="logo">
    <li class="logo_00"></li>
    <li class="logo_01"><a href="http://www.unam.mx" target="_blank">    </a></li>
    <li class="logo_02"><a href="http://www.ingenieria.unam.mx" target="_blank">
          </a></li>
  </ul>
  <ul class="second_nav">
    <li><span id="spanreloj"></span></li>
    <li><a href="javascript:agregar()">Agregar a Favoritos</a></li>
  </ul>
  <!-- buscador -->
  <ul class="main_nav">
    <li class="bt1 selected"><a href="index.html">Inicio</a></li>
    <li class="bt2"><a href="modulo.html#titulo">Módulo</a></li>
    <li class="bt3"><a href="proyectos.html#titulo">Proyectos</a></li>
    <li class="bt4"><a href="contacto.html#titulo">Contacto</a></li>
  </ul>
</div>
```


Procedimiento

El *footer* se encuentra en la parte inferior, contiene los *links* a las cuatro redes sociales más populares en las cuales se tendrán cuentas para compartir información sobre el Módulo: *Twitter*, *Facebook*, *Youtube* y *Reevon*. También se muestran los datos del *copyright* y el acceso al “mapa de sitio” del portal además de que se agregaron enlaces a las páginas oficiales del *XHTML* y *CSS* por parte de la *W3C*.

Parte del código del *footer* quedó:

```
<div id="footer">
  <div class="cont_footer">
    <div class="social_box">
      <table>
        <tr><td><div class="more"><a href="http://www.twitter.com" >
          
        </a></div></td><td></td></tr><tr><td></td>
        <td><div class="more"><a href="http://www.facebook.com" >
          
        </a></div></td></tr><tr><td><div class="more">
          <a href="http://www.youtube.com" >
          
        </a></div></td><td></td></tr><tr><td><td><div class="more">
          <a href="http://portal.reevon.com/" >
          </a></div></td></tr>
      </table>
    </div><div class="img_footer">
  </div>
</div>
```

Estos dos elementos (*header* y *footer*) permanecen constantes en todas las páginas de las que se compone el portal *web*, a diferencia del *content* el cual cambia dependiendo de la sección en la que se encuentre el usuario; no tienen variación alguna por lo que el código se aprovecha reutilizándolo y de esta forma se optimiza parte del desarrollo.

El objetivo principal de *Content* es albergar el contenido principal del sitio ya sea texto, imágenes o ambos. Así que basando el desarrollo de esta parte del *body* en el boceto de la página principal de la *webapp*, se agregó para esta sección: un *slider* con el cual se exponen imágenes del Módulo y nueve botones los cuales serán *links* a las diferentes secciones del portal (módulo, materias, material didáctico, enlaces, buscador, proyectos, noticias y contáctanos).

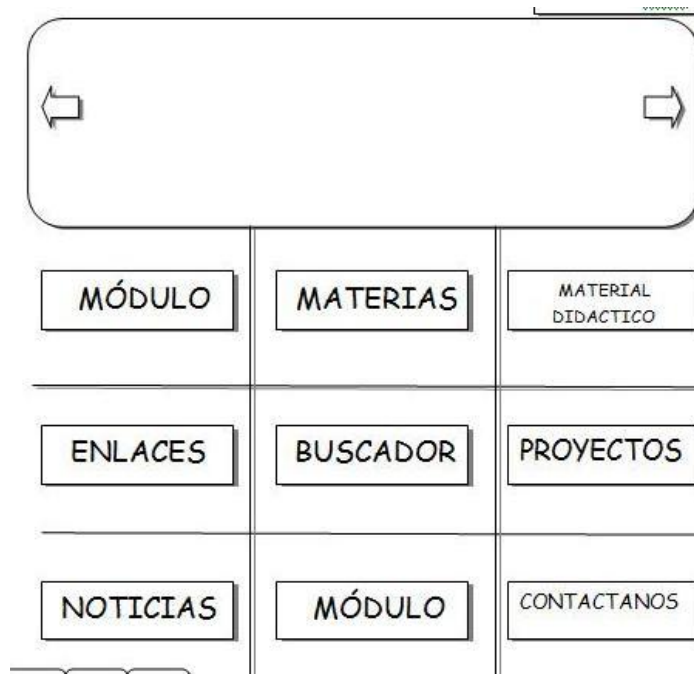


Ilustración 37 Content

Parte del código del *content* quedó:

```
<div id="content">
  <div class="cont_carousel">
    <div class="jcarousel-skin-tango" id="mycarousel"></div>
  </div>

  <h2>MÓDULO</h2></div><div class="mini_box">
    <h2>MATERIAL DIDÁCTICO</h2></div>
    <h2>ENLACES</h2></div><div class="mini_box">
    <h2>PROYECTOS</h2></div><div class="mini_box">
    <h2>DEMO REEL</h2></div><div class="mini_box">
    <h2>MATERIAS</h2></div><div class="mini_box">
    <h2>BUSCADOR</h2></div><div class="mini_box">
    <h2>CONTACTANOS</h2></div><div class="mini_box">
    <h2>NOTICIAS</h2></div><div class="clearfix"></div>
</div>
```

Para las diferentes secciones de la *webapp* se generó el código correspondiente al *content*, basándolo en los bocetos ya desarrollados; a continuación se presenta una breve descripción de estos acompañada de la imagen del boceto y un fragmento de su correspondiente código.

Procedimiento

La página principal de la sección Módulo muestra una serie de botones que llevarán al usuario a las diferentes subsecciones que lo componen, por lo que para este *body* se manejan cinco imágenes (botones/*links*).



Ilustración 38 Página principal de la sección Módulo

```
<div class="modulo_box">
  <table>
    <tr><td><div class="more_modulo"><a href="modulo_directorio" >
      </a></div></td>
    </tr><tr><td><div class="more_modulo"><a href="modulo_horario">
      
    </tr><tr><td><div class="more_modulo"><a href="modulo_def_txt">
      
    </tr><tr><td><div class="more_modulo"><a href="modulo_eventos">
      </a></div></td>
    </tr><tr><td><div class="more_modulo"><a href="modulo_planEstuos">
      
    </tr></td></table>
</div>
```

La primer sub-sección del Módulo es “DEFINICIÓN” se propusieron tres páginas que la componen: texto, imágenes y videos; dependiendo la página en la que se encuentre el usuario es el contenido que tiene, puede ser texto, imágenes o bien videos todos representativos del Módulo, además se agregaron los tres botones para navegar entre las tres páginas.

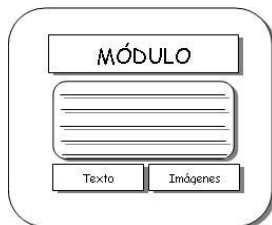


Ilustración 39 Definición del Módulo

```
<div id="content">
  <h2>MÓDULO DE COMPUTACIÓN GRÁFICA. </br> </br>
  FI, UNAM.</h2><h3>Submódulo de COMPUTACIÓN
  GRÁFICA </br>del Módulo SISTEMAS INTELIGENTES Y
  COMPUTACIÓN GRÁFICA</h3>
  <p>Considerado uno de los campos de mayor crecimiento y de mayor valor
  agregado dentro del Crecimiento,
  Informáticas y Tecnologías de la
  Información. Computación Gráfica ofrece
  una amplia diversidad de opciones para el desarrollo tanto
  profesional como académico.</p>
  <p>Los egresados de este submódulo pueden
  desarrollarse en las áreas
  <p>Las asignaturas de este módulo involucran un uso
  extensivo de conceptos de Álgebra Lineal y
  Geometría Analítica,
  así como de habilidades desarrolladas y conocimientos de
  programación, principalmente en lenguaje C, C++ y C#,
  entre otros.</p>
</div>
```

Para la sub-sección “DIRECTORIO” se manejo una plantilla que se empleó para crear las seis páginas, una por cada materia que compone el Módulo, en

Procedimiento

cada página se incluyen los datos de los profesores que imparten las materias, como son: nombre, teléfono, correo electrónico, etc., esto en texto, además se agregaron botones para intercambiar las páginas de las diferentes materias.

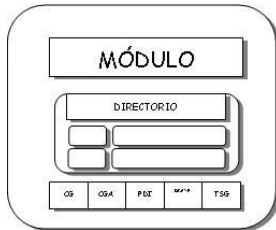


Ilustración 40 Directorio de los profesores del Módulo

```
<div id="content">
  <div class="clearfix"></div>
  <div class="botoneraMaterias">
    <div class="btnM_left">
      
    </div><div class="btnM_right">
      <table>
        <tr><td><div class="btnM">
          <a href="modulo_directorio_cg.html#palanca" >
            <p>CG...
          </a></div></td><div class="btnM">
            <a href="modulo_directorio_cga.html#palanca" >
              <p>CGA...
            </a></div></td><div class="btnM">
              <a href="modulo_directorio_dimyrv.html#palanca" >
                <p>DIMyRV...
              </a></div></td><div class="btnM">
                <a href="modulo_directorio_tsg.html#palanca" >
                  <p>TSG...
                </a></div></td></tr></table>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
```

Para escribir el código de la sub-sección horarios solo se emplearon etiquetas `<p></p>`, y `<a href= ></a>`, para mostrar los horarios de las seis materias del Módulo y poder navegar entre las diferentes páginas.

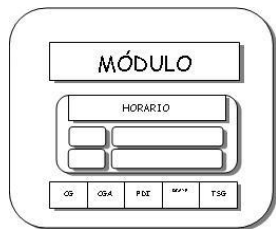


Ilustración 41 Horarios de las materias del Módulo

```
<div id="content">
  <div class="btnM_right">
    <table>
      <tr><td><div class="btnM">
        <a href="modulo_horario_cg.html#palanca" >
          <p>CG...
        </a></div></td><div class="btnM">
          <a href="modulo_horario_cga.html#palanca" >
            <p>CGA...
          </a></div></td><div class="btnM">
            <a href="modulo_horario_pdi.html#palanca" >
              <p>PDI...
            </a></div></td><div class="btnM">
              <a href="modulo_horario_dimyrv.html#palanca" >
                <p>DIMyRV...
              </a></div></td><div class="btnM">
                <a href="modulo_horario_dac.html#palanca" >
                  <p>DAC...
                </a></div></td><div class="btnM">
                  <a href="modulo_horario_tsg.html#palanca" >
                    <p>TSG...
                  </a></div></td></tr></table></div></div>
```

Procedimiento

El diseño establecido en el boceto de la sub-sección “EVENTOS” propuso que se emplearan etiquetas que manejen texto e imágenes y en ciertos casos *links*, por lo que el punto fuerte de esta sub-sección recae en el diseño para atraer y mantener a los usuarios.

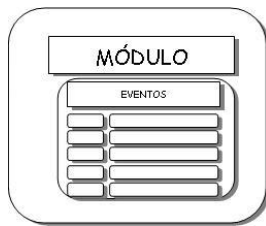


Ilustración 42 Eventos organizados por el Módulo

```
<div id="content">
<div id="modulo" style="float: left"><br /></div>
<a name="titulo"></a><div class="titulo"></div>
<div class="clearfix"></div><h2 class="title_section_modulo">Eventos:
<ul class="modulo_event"><div class="item" >
<h2>Diseño de Marca
<div class="box">

<h3>titulo primer evento</h3>
<p>descripcion del primer evento con el que iniciamos esta página
Cada evento tendrá su propia descripción y de ser necesario se
Manejaran links en la misma descripción . </p></br>
<a href="evento_001.html" class="event" title="evento_001"
target="_blank">Leer M&aacute;rcas</a>
</div></div></li></ul>
</div>
```

El boceto de la página PLAN DE ESTUDIOS de la sección MÓDULO indica que se tiene que agregar una imagen, el plan de estudios de la carrera de Ingeniería en Computación por parte de la Facultad de Ingeniería y mediante la técnica de “mapa de imágenes” se agregaron los *links* que conducen a los usuarios a las diferentes páginas de las materias del módulo.



Ilustración 43 Plan de estudios agregando las materias del Módulo

```
<div id="content">
<div id="modulo" style="float: left"><br /></div>
<a name="titulo"></a><div class="titulo">
</div>
<h2 class="title_section_modulo">Plan de Estudios: </h2>
<div class="planE">

<area shape="rect" coords="594,598,678,663"
href="materias_compGraf_txt.html#palanca"
title="COMPUTACIÓN GRÁFICA">
<area shape="rect" coords="166,756,252,818"
href="materias_pdi_txt.html#palanca"
title="PROCESAMIENTO DIGITAL DE IMÁGENES">
<area shape="rect" coords="267,757,352,820"
href="materias_tsg_txt.html#palanca" title="TEMAS
SELECTOS DE GRAFICACIÓN">
</map>
</div>
</div>
```

Procedimiento

En la página proyectos (según indica el boceto), se agregaron imágenes, textos y *links*, estos proyectos pueden ser temas de tesis, servicio social, conferencias etc., todo con relación a la Computación Gráfica, por lo que esta sección estará en constante crecimiento y cambio.



Ilustración 44 Proyectos relacionados con el Módulo

```
<div id="content">
<div id="proyectos" style="float: left"><br /></div>
  <a name="titulo"></a>
    <div class="titulo"></div>
    <ul class="proyectos"><li>
      <div class="item" id="proyecto_1">
        <h2>titulo proyecto 001</h2>
        <div class="box">
          <a href="proyecto_001.html"
            title="proyecto_001" target="_blank">
            
            <p>Descripci&oacuten del proyecto</p>
          </a></div></div>
        </li></ul>
  </div></div>
```

La página “MATERIAS” se divide en dos secciones, en la primera (que se muestra a continuación) se colocan botones/*links* que envían a los usuarios a la segunda sección dependiendo que materia fue elegida por ellos.

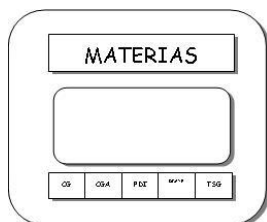


Ilustración 45 Materias que integran el Módulo (primera sección)

```
<div id="content">
<div id="proyectos" style="float: left"><br /></div>
  <a name="titulo"></a>
    <div class="titulo"></div>
    <div class="clearfix"></div>
    <div class="botoneraMaterias">
      <div class="btnM_left">
        
      </div><div class="btnM_right">
        <table><tr><td><div class="btnM">
          <a href="materias_compGraf_txt.html#palanca">
            <p>CG...
          </a></div></td><td><div class="btnM">
          <a href="materias_compGrafA_txt.html#palanca">
            <p>CGA...
          </a></div></td><td><div class="btnM">
          <a href="materias_pdi_txt.html#palanca">
            <p>PDI...
          </a></div></td><td><div class="btnM">
          <a href="materias_dac_txt.html#palanca">
            <p>DAC...
          </a></div></td><td><div class="btnM">
          <a href="materias_tsg_txt.html#palanca">
            <p>TSG...
          </a></div></td></tr></table>
      </div>
    </div></div>
```

La segunda sección de “MATERIAS” está compuesta por páginas con texto, imágenes y videos, dependiendo de la materia que se haya elegido, mostrará la información relacionada. Además de los botones necesarios para intercambiar entre texto, imágenes y videos, se proporciona el poder cambiar de materia.

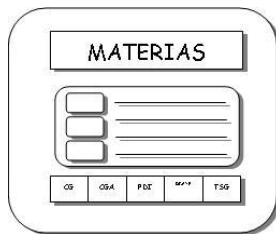
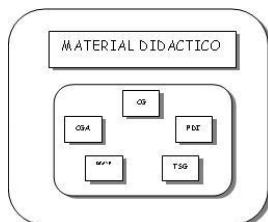


Ilustración 46 Materias que integran el Módulo (segunda sección)

```
<div id="content">
<div id="proyectos" style="float: left">    <br /></div>
    <div class="mat_right">
        <ul class="imag_box">
            </li>
            </li>
            </li>
            </li>
            </li>
            </li>
            </li>
            </li>
            </li>
            </li>
        </ul></div>
</div></div>
```

La página “MATERIAL DIDACTICO” igualmente está compuesta por dos partes, la primera solo contiene botones/*links* que permiten navegar entre las diferentes materias del Módulo; la segunda parte (dependiendo de la materia elegida) muestra el temario de la respectiva materia donde cada título es un *link* que conduce a la tercer parte de esta sección.



**Ilustración 47 Material
didáctico referente a
computación gráfica
(primera parte)**

[illegible]

Procedimiento

La tercer parte de "MATERIAL DIDACTICO" maneja seis botones dos de ellos para navegar entre las tres partes de esta sección (temario y materias) y los otros cuatro proporcionan imágenes, videos, texto y código del tema elegido.



Ilustración 48 Material didáctico referente a computación gráfica (tercer parte)

```
<div id="content">
<div id="modulo" style="float: left"><br /></div>
  <a name="titulo"></a>
  <div class="titulo"></div>
  <div class="clearfix"></div>
  <div class="titulo_03">
    <h4>TEMAS SELECTOS DE GRAFICACI&Oacute;N</h4>
  </div><div class="matDid"><ul class="temario">
    <li><a href="construccion.html" target="_blank">
    1 Introducci&Oacute;n.</a></li><li>
    <a href="construccion.html" target="_blank">
    1.1 Objetivo del curso.</a></li><li>
    <a href="construccion.html" target="_blank">
    1.2 Antecedentes acad&Eacute;micos necesarios.</a></li>
    <a href="construccion.html" target="_blank">
    2.1 Temas especiales de Computaci&Oacute;n Gr&Aacute;fica
    Avanzada.</a></li>
    <a href="construccion.html" target="_blank">
    2.2 Geometr&Iacute;a Computacional.</a></li>
    <li><a href="construccion.html" target="_blank">
    </li><a href="construccion.html" target="_blank">
  </ul>
</div></div>
```

En la sección noticias fue necesario agregar imágenes y texto, por lo que solo fueron necesarias algunas etiquetas que nos permitieran hacer esto y el trabajo pesado esta en el manejo de estilos mediante las CSS.



Ilustración 49 Noticias referentes a computación gráfica

```
<div id="content">
  <a name="titulo"></a>
  <div class="titulo"></div>
  <div id="noticias" style="float: left"><br /></div>
  <div class="noticias">
    <div class="box">
      <h1>UNAM 360, aplicaci&Oacute;n
      desarrollada por Facultad de
      Ingenier&Iacute;a para iPhone </h1>
      <p>La Universidad Nacional
      Aut&Oacute;noma de M&Eacute;xico dio a
      conocer del Campus.</p>
      <a href="http://www.mediatelecom.com.mx
      Ingenier&Iacute;a para iPhone"
      target="_blank"><span>Ver
      M&Aacute;ximas</span></a>
    </div></div></div></div>
```


La sección enlaces es de las más amplias, fue necesario realizar una búsqueda exhaustiva con la cual se obtuvo un gran número de páginas web que contienen información relacionada con la computación gráfica. Se clasificaron estas páginas en las diferentes divisiones con las que se cuenta (animación, videojuegos, escuelas, estudios de animación, granjas de *render*, *software*, Etc.), se muestran tomas de pantalla de cada portal, además de una breve descripción del contenido, esto para proporcionar a los usuarios la información que necesitan en el momento en el que lo requieren.

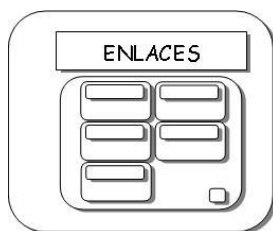


Ilustración 50 Enlaces
referentes a computación
gráfica

```
<div id="content">
<div id="proyectos" style="float: left"><br /></div>
  <a name="titulo"></a>
  <div class="titulo"></div>
  <div class="clearfix"></div><div class="nube">
    <a style="color:#0030fb; font-size: 15px;"
href="enlaces_aplicaciones.html#titulo"> APLICACIONES DE
COMPUTACI&Oacute;N GR&Aacute;FICA</a>
    <a style="color:#0908ab; font-size: 18px;"
href="enlaces_geometriaComputacional.html#titulo">
GEOMETR&Iacute;A COMPUTACIONAL </a>
    <a style="color:#4400ff; font-size: 14px;"
href="enlaces_videojuegos.html#titulo">
VIDEOJUEGOS </a><a style="color:#0f0099; font-size: 20px;"
href="enlaces_empresasVideojuegos.html#titulo">
EMPRESAS DE VIDEOJUEGOS </a><a style="color:#300ffa; font-size: 17px;"
href="enlaces_DAC.html#titulo">
DISE&Ntilde;O ASISTIDO POR COMPUTADORA </a>
    <a href="http://www.coyoteanimacion.com/" target="_blank">
</a>
    <h3>http://www.coyoteanimacion.com/</h3>
    <label>Coyote sociedad animada es un grupo de servicios de creaci&oacute;n y
producci&oacute;n audiovisual de dibujos animados, ideogr&aacute;ficos y efectos
especiales.
    <a href="http://www.wanako.tv/" target="_blank">
</a>
    <h3>http://www.wanako.tv/</h3><label>Wanako nace de la necesidad de
brindar una soluci&oacute;n publicitaria que no tenga l&iacute;mites creativos.
Es por eso que nuestro equipo est&a integrado por directores de cine,
</ul></div></div></div>
```

La página para los “*DEMO-REEL*” es de las que estarán en constante crecimiento, esto debido a que se formará gracias a las aportaciones de los usuarios, en un principio se incluyeron tres videos con sus respectivas descripciones; estará formada por la función necesaria para agregar y reproducir videos además de texto (ingresado mediante la etiqueta párrafo).

Procedimiento

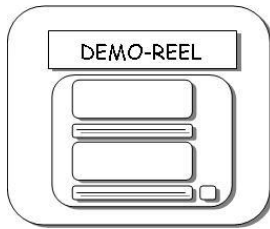


Ilustración 51 Demo-reel
de los alumnos del
Módulo

```
<div id="content">
<a name="titulo"></a>
<div class="titulo"></div>
<div id="demoreel" style="float: left"><br /></div>
<div class="demo"><div class="videos_01">
<object width="600" height="350" type="application/x-shockwave-flash"
data="http://www.youtube.com/v/xfxRkXUu8_c">
<param name="movie" value="http://www.youtube.com/v/zdOHxNL7mK0" />
<param name="wmode" value="transparent" />
</object></div><div class="box_demo"><p>Nombre:</br>
Especialidad:</br>Correo Electronico:</p></div>
<div class="demo"><div class="videos_01"><object width="600" height="350"
type="application/x-shockwave-flash"
data="http://www.youtube.com/v/xfxRkXUu8_c"><param
name="movie" value="http://www.youtube.com/v/zdOHxNL7mK0" />
<param name="wmode" value="transparent" /></object></div>
<div class="box_demo"><p>Nombre:</br>Especialidad:</br>
Correo Electronico:</p>
</div></div></div></div></div></div>
```

Para el buscador interno se agregó un motor de búsqueda personalizado, aprovechando las facilidades ofrecidas por Google mediante el *Google Site Search (beta)*.



Ilustración 52 Buscador
interno del Portal

```
<div id="content">
<a name="titulo"></a>
<div class="titulo"></div>
<div id="buscador" style="float: left"><br /></div>
<div class="buscador"><div id="cse" style="width: 100%;">
<p>Buscador para Computaci&oacuten Gr&aacutefica FI, UNAM.</p>
</div><script src="https://www.google.com/jsapi" type="text/javascript">

</script><script type="text/javascript">google.load('search', '1', {language: style :
google.loader.themes.SHINY});google.setOnLoadCallback(function() {
customSearchControl = new google.search.CustomSearchControl(
'008081750982527658461:r-9xjfr398q');
customSearchControl.setResultSetSize(google.search.
Search.FILTERED_CSE_RESULTSET);
customSearchControl.draw('cse');}, true);</script>
<div class="clearfix"></div>
</div></div>
```

El formulario aportaciones es el encargado de comunicar a los usuarios con el *webmaster* del portal, con el fin de que este reciba información proporcionada por los usuarios para hacer crecer la *webapp*, esta información puede ser desde imágenes, videos material didáctico, enlaces hasta un simple comentario o aportación al diseño del sitio.

Procedimiento



Ilustración 53 Formulario
para las aportaciones de
los usuarios

```
<div id="content">
<a name="titulo"></a>
<h1 class="title_section">Aportaciones: </h1>
<div class="clearfix"></div>
<div class="main_contact">
<form method="post" action="enviar_mail.php" enctype="multipart/form-
data">
<div class="box_left"><label>Nombre</label>
<input type="text" size="30" value="" name="nombre_A" id="nombre_A"
maxlength=50/><label>Email</label><input type="text" size="37" value="" name="email1_A"
id="email_A" maxlength=50/><label>Tel&eacute;fono</label><input type="text" size="37"
value="" name="your-name_A"/><label>Adjunta tu archivo comprimido:</label>
<input type="file" name="archivo" size="30" maxlength="100">
</div><div class="box_right"><label>Tema</label>
<input type="text" size="37" value="" name="tema_A" id="tema_A"
maxlength=50/><label>Descripci&oacute;n</label><textarea rows="4" cols="28"
name="mensaje_A"></textarea><input type="submit" name="enviar_A" value="Enviar"/>
<input type="reset" name="borrado_A" value="Borrar"/>
</div></form></div>
```

El formulario contáctanos es indispensable ya que es el utilizado para comunicar a los usuarios con el *webmaster* del portal, es mediante este formulario con el cual se pueden hacer comentarios y preguntas sobre el portal o bien el módulo, además de ser la manera más directa de comunicación entre los usuarios, el administrador del sitio y los profesores del Módulo.



Ilustración 54 Formulario
para establecer
comunicación entre los
usuarios y el webmaster

```
<div id="content">
<a name="titulo"></a>
<h1 class="title_section">Contactanos: </h1>
<div class="clearfix"></div><div class="main_contact">
<form method="post" action="enviar_mail.php" enctype="multipart/form-
data"><div
class="box_left"><label>Nombre</label><label>Email</label>
<label>Tel&eacute;fono</label><input type="text" size="37" value="" name="your-name"/>
</div><div
class="box_right"><label>Asunto</label>
<label>Mensaje</label>
<textarea rows="4" cols="28"
name="mensaje"></textarea>
<input type="submit"
name="enviar" value="Enviar"/>
<input type="reset"
name="borrado" value="Borrar"/></div>
</form>
<div class="address_box">
<h3>Direcci&oacute;n:</h3>
<p>Escolar Conjunto Norte
facultad de Ingenier&iacute;a, 04510 Ciudad de M&eacute;xico, Distrito Federal</p>
<div class="googlemap"><iframe width="800" height="300"
frameborder="0" scrolling="no" marginheight="0" marginwidth="0"
src="http://maps.google.com.mx/ m+cu& d"></iframe><br /><small>
<a href="http://maps.google.com. lign:left">Ver mapa en grande</a></small>
</div></div></div></div></div>
```

Imágenes para la webapp

En este sitio se hace uso de una gran cantidad de imágenes tanto para mostrar la información como para aportar al diseño visual de la misma, debido a esto el portal puede hacerse muy pesado, con lo cual se provocaría que tarde demasiado tiempo en

cargarse en los navegadores. Así que a estas imágenes se les proporciona un cierto proceso para aligerarlas sin perder calidad.

Las imágenes necesarias para la *webapp* se optimizaron mediante *Image Optimizer*. Fue un proceso de compresión de imágenes con el cual se buscó reducir el peso de estas para poder subirlas a la *web* y así lograr que el sitio sea cargado mucho más rápido; se procuró tener un equilibrio entre el peso y la calidad de las imágenes para evitar que lucieran borrosas o con mala resolución.

Otras imágenes que se muestran en el portal son los botones, como los del menú principal, este tipo de imágenes se crearon con *Inkscape*, debido a que es un editor de gráficos vectoriales de código abierto y hace uso del estándar de la W3C: el formato de archivo *Scalable Vector Graphics (SVG)*.

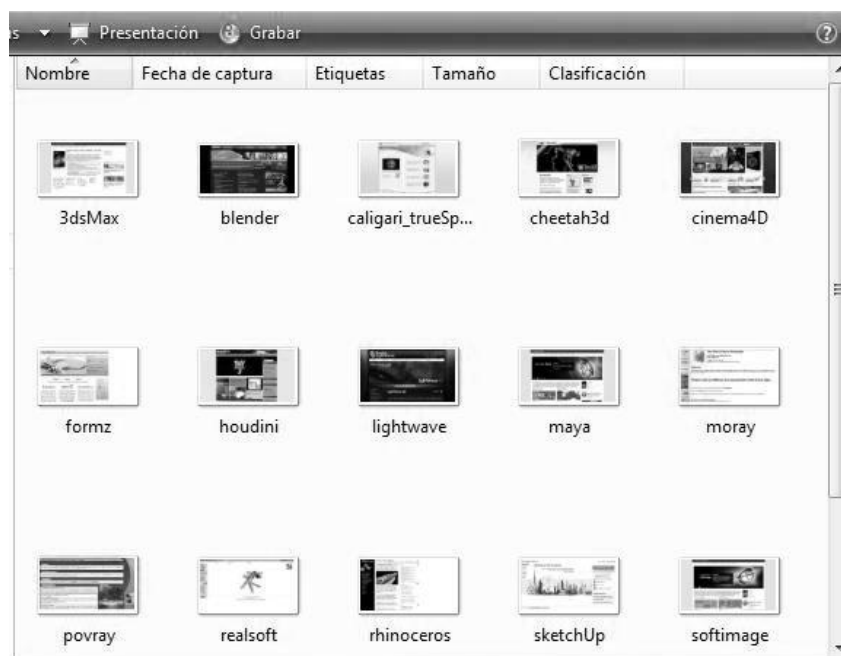


Ilustración 55 Imágenes listas para web (sección Enlaces)

Apis

Mediante el uso de una Interfaz de Programación de Aplicaciones (*API*) se añadió funcionalidades extras a la *webapp*. De la gran cantidad de añadidos que nos permiten ampliar la funcionalidad del sitio (ofrecidos por *Internet*) se eligió usar *Google Maps* ya que permite usar mapas de cualquier rincón del mundo, los cuales se aprovecharon para mostrar la ubicación de la Facultad de Ingeniería.

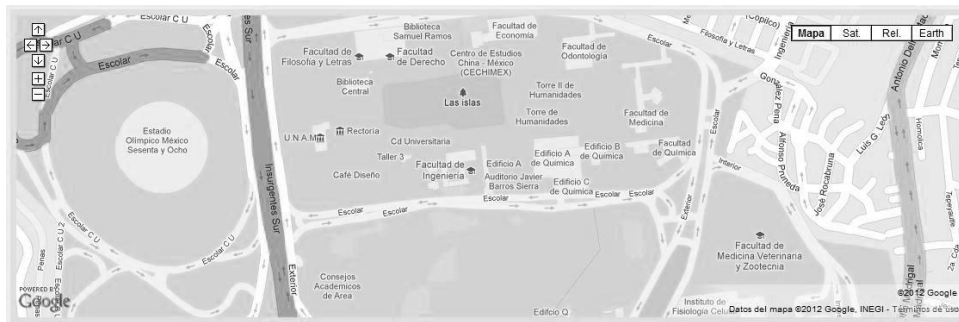


Ilustración 56 API GoogleMaps

Formularios

Al desarrollar el código para las diferentes secciones del portal se crearon dos formularios, uno para la sección *CONTACTANOS* y otro para *APORTACIONES*, con los cuales se entablara la comunicación entre los usuarios y el administrador del sitio, esto para enviar comentarios, dudas, observaciones e incluso material (texto, imágenes, videos, etc.), por lo que se empleó el lenguaje *PHP* para el envío de la información de manera segura.

Parte del código empleado para el archivo *envio.php*:

```
<html>
  <body>
    Nombre: <?php echo $_GET['nombre'];?> <br/>
    Asunto: <?php echo $_GET['asunto'];?> <br/>
    Comentario: <?php echo $_GET['comentario'];?> <br/>
  </body>
</html>
```

5.5.- Pruebas

Después de concluir el desarrollo de la *webapp* fue necesario someterla a diferentes estrategias de pruebas (las cuales abarcaron tanto revisiones como pruebas ejecutables) para descubrir errores en el contenido, la función, la facilidad de uso, la navegabilidad, el desempeño, la capacidad y la seguridad de la *webapp*.

Se procuró obtener una buena consideración de las diversas dimensiones de la calidad de la *webapp*, para lo cual se tomaron en cuenta diferentes dimensiones para evaluar la *webapp*, que son particularmente relevantes. La forma en la que se evaluó esta fue mediante la aplicación de una serie de revisiones técnicas que valoraron varios elementos del modelo de diseño.

Para llevar a cabo las pruebas se eligió una población controlada y monitoreada de usuarios finales; además de probar la compatibilidad al ejecutarla en varias configuraciones huésped, en los lados tanto del cliente como del servidor.

Se pidió a esta población que empleara la Ilustración 57 para llevar a cabo su evaluación final:

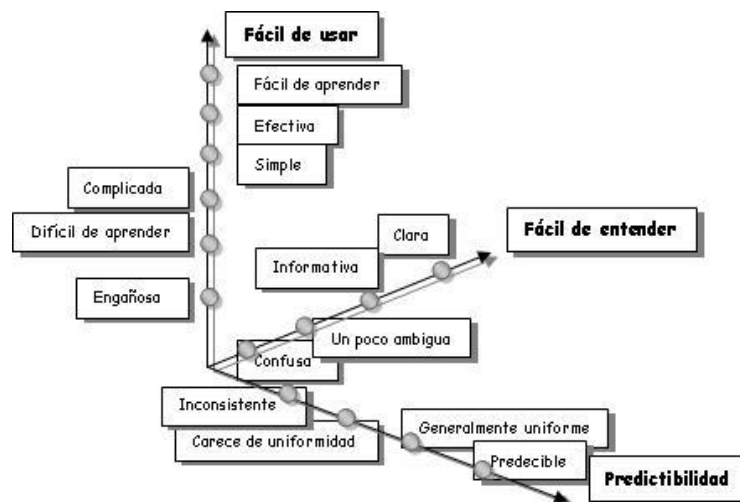


Ilustración 57 Valoración cualitativa de la facilidad de uso

El proceso de prueba

La población elegida para las pruebas de calidad fue integrada por diez personas de diferentes edades a fin de cubrir una buena parte del mercado (usuarios finales), de entre los participantes se incluyeron: un estudiante de preparatoria (16 años), un ama de casa (44 años), un estudiante de posgrado (28 años), un licenciado en comunicación (45 años), un estudiante de carrera técnica (23 años), cinco estudiantes de la Facultad de Ingeniería (de los cuales uno es de Ingeniería Industrial, uno más de Ingeniería Civil y tres de Ingeniería en Computación).

A esta población se le pidió que usara la aplicación en su totalidad, a fin de encontrar fallos en su estructura, algún *link* roto, incomodidades en su manejo, dificultad de uso, mal diseño, entre otras cosas. Esto para poder corregir y mejorar la *webapp* y cumplir con el ciclo de desarrollo *web*.

Para poder llevar a cabo esta prueba se desarrolló el siguiente cuestionario y se dejó a criterio de cada participante algún comentario extra:

DESARROLLO DEL PORTAL WEB DEL MÓDULO DE COMPUTACIÓN GRÁFICA
DE LA FACULTAD DE INGENIERÍA, UNAM

Procedimiento

**Cuestionario para evaluar la calidad del Portal web del Módulo de
Computación Gráfica de la Facultad de Ingeniería**

(Marca con una X tu respuesta)

Prueba de la interfaz				
¿Se trata de una interfaz estéticamente agradable?				
Nada agradable	Poco agradable	Indiferente	Agradable	Muy agradable
¿Existen videos o imágenes que no se puedan visualizar?				
Si		No		
¿Los videos e imágenes que se presentan son acorde a la webapp?				
En desacuerdo	Obsoletos	Acorde	Buenos	Si, aunque los cambiaria
¿Fue fácil encontrar el tema de interés?				
Muy difícil	Difícil	No lo busque	Fácil	Muy Fácil
Prueba de la facilidad de uso				
¿La <i>webapp</i> interactúa de manera efectiva con los usuarios?				
Nada	Poco	Interactúa	Buena interacción	Muy buena interacción
¿En qué grado la <i>webapp</i> influye para guiar las acciones de los usuarios?				
Nada	Poco	Indiferente	Influye	Influye mucho
¿La interfaz de la <i>webapp</i> facilita las tareas de los usuarios?				
Nada	Poco	Indiferente	Facilita	Facilita mucho
Pruebas de compatibilidad				
¿En qué dispositivos de despliegue empleó la webapp?				
Computadora personal	Laptop	Tableta electrónica	Teléfono móvil	Otro
¿Qué sistema operativo emplea?				
Windows	Linux	Mac OS	Android	Otro
¿Qué navegador prefiere usar?				
Internet Explorer	Firefox	Opera	Safari	Google Chrome
Otro (¿Cuál?)				
Pruebas de navegación				
¿Qué tan funcionales son los mecanismos que permiten al usuario de la <i>webapp</i> navegar a través de ella?				
Nada funcionales	Poco funcionales	Indiferente	Funcionales	Muy funcionales
¿Los mecanismos de navegación ayudaron a ubicar donde se encontraba dentro de la webapp y llegar a la información deseada?				
Nada	Poco	No los ubique	Si ayudan	Ayudan mucho

Puedes escribir algún comentario al reverso de esta hoja. Gracias.

Procedimiento

Valoración cualitativa de la facilidad de uso (señala con una X tu respuesta)

Engañosa	Difícil de entender	Complicada	Simple	Efectiva	Fácil de aprender	Fácil de usar
		Confusa	Poco ambigua	Informativa	Clara	Fácil de entender
		Inconsistente	Carece de uniformidad	Generalmente uniforme	Predecible	predictibilidad

5.6.- Entrega

Para tener el portal en línea es necesario subir el conjunto de los archivos que lo conforman a un servidor web; al momento de subir estos archivos existen varias opciones como es el utilizar las facilidades proporcionadas por alguna empresa de *hosting*, utilizar un editor de páginas web que incorpore esta funcionalidad (por ejemplo, el *Dreamweaver*) o bien utilizar un programa específico para ello como lo es *Filezilla*.

No necesariamente se tiene que hacer uso de algún programa especializado para subir los archivos al servidor, puesto que toda empresa de *hosting* proporciona alguna interfaz a través de la cual se puedan subir los archivos.

En su mayoría, los servicios gratuitos manejan un interfaz simple con la cual solo se puede subir un archivo a la vez, a diferencia de los servicios de pago en los cuales la interfaz maneja varios archivos al mismo tiempo.

Otra forma de subir el sitio es mediante el uso de un editor de páginas web que incorpore esta funcionalidad. En estos editores (profesionales) se incorpora el acceso *FTP*; con esto se permite subir y descargar los archivos de una manera mucho más eficiente manejando una interfaz más compleja. Una característica particular de este tipo de editores es que permiten tener los archivos sincronizados.

Procedimiento

Para la *webapp* se utilizó un programa específico para ello "*Filezilla*". Se trata de un cliente *FTP*. Para transferir ficheros por *FTP* se necesita tener instalado un programa cliente de *FTP* en la computadora local y conocer la dirección del servidor *FTP* con la que se quiere conectar. Esta dirección la proporciona la empresa con la que se contrata el alojamiento (*hosting*).

Los programas *FTP* tienen un interfaz bastante simple. Esta interfaz muestra los archivos (contenidos en la computadora local) en un lado de la pantalla y los que ya están en el servidor al otro lado. Con esto lo único que se tiene que hacer es arrastrar y soltar archivos de un lado a otro.

La principal ventaja del *FTP*, es que permite subir varios archivos y carpetas completas al mismo tiempo, sin tener que ir uno por uno, a parte de su comodidad.

6.- Resultados

Se desarrolló el portal web del Módulo de Computación Gráfica de la Facultad de Ingeniería, el cual servirá como plataforma para el crecimiento de los alumnos del módulo y punto de partida para tener acceso a una serie de recursos y servicios de computación gráfica, además de cubrir otras necesidades de la comunidad del dicho Módulo.

Dicho portal ésta disponible en línea en la dirección:

<http://www.ingenieria.unam.mx/computaciongraficafi>

Se emplearon los pasos necesarios (basados en la opinión de un gran número de ingenieros *web*) para desarrollar y mantener una *webapp*: definición del proyecto, estructura del sitio, diseño (gráfico), desarrollo, pruebas y entrega. Se diseño y desarrolló esta *webapp* desde la recopilación de información inicial, pasando por la creación y desarrollo y llegando hasta el mantenimiento y actualización de la misma.

Se logró obtener una *webapp* que cumple con los objetivos inicialmente planteados, la cual es funcional y estéticamente agradable. Está dividida en nueve secciones, las cuales son: Módulo, Material didáctico, Enlaces, Proyectos, *Demo-reel*, Materias, Buscador, Contáctanos y Noticias; estas a su vez (en su mayoría) están conformadas por sus respectivas subsecciones de manera que sean cubiertas las necesidades planteadas.

La primer sección con la que se encuentran los usuarios es el *HOME (Index)*, es a partir de esta de donde se puede iniciar la navegación del sitio puesto aquí se encuentran los enlaces a las diferentes secciones del portal (módulo, materias, material didáctico, enlaces, buscador, proyectos, noticias y contáctanos), se presenta una navegación sencilla, clara e intuitiva para facilitar en lo posible el entendimiento de la *webapp* por parte de los usuarios.

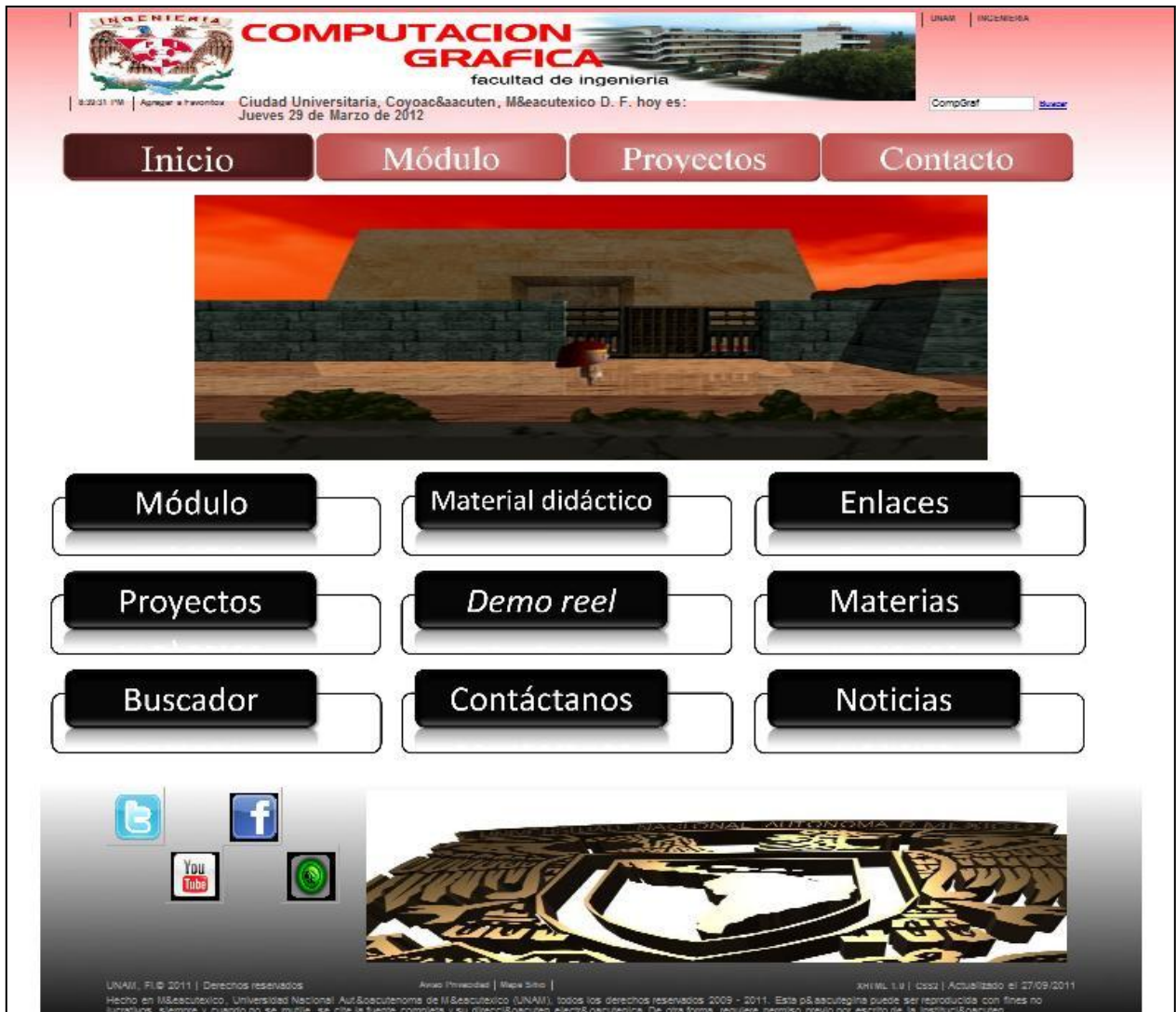


Ilustración 58 HOME

Las partes del *header* y *footer* se presentan exactamente igual en todas las secciones de la *webapp* esto por estandarización, por lo que en las siguientes imágenes se pretende solo mostrar el *content*, el cual si estará cambiando dependiendo la sección en la que se encuentre el usuario.



Ilustración 59 Módulo

La sección Módulo da acceso a toda la información relevante del Módulo de Computación Gráfica de la Facultad de Ingeniería; proporciona el directorio de los profesores (nombre, correo electrónico, ubicación dentro de Ciudad Universitaria, etc.), los horarios y ubicación de las clases del módulo y, dado el caso, se colocará información sobre eventos tales como conferencias, cursos, pláticas, talleres, etc., así también se presenta la información importante para los alumnos aspirantes a este módulo como: campo de desarrollo, campo de trabajo, perfil de los estudiantes y egresados entre otras cosas.

La sección “Módulo” es la parte más importante del portal ya que es la sección que resuelve el problema planteado que dio origen a esta tesis (la falta de un lugar en el que se encuentre concentrada toda la información como son directorio, eventos, horarios, etc.).



Ilustración 60 Material didáctico

La sección “Material didáctico” es de las más amplias que componen el portal está diseñada para entregar a los usuarios material que complementen sus estudios en materias del Módulo, este material puede ser texto, imágenes, etc.; el objetivo principal es proporcionar de una manera sencilla todo este material a los usuarios finales.

Se puede navegar entre las seis materias para elegir cuál es la que se quiere o se necesita explorar, una vez que se ha elegido se ingresa a su página para que se muestre el temario en pantalla; cada título o subtítulo de los temarios son un hipervínculo a las siguientes secciones en las cuales se muestra la información solicitada, código, texto, imágenes, etc.

El objetivo de crear la sección “Enlaces” es que sirva como puerto de salida para los usuarios ya que se pone al alcance de estos un gran número de hipervínculos con sitios especializados llenos de información de gráficos por computadora, respecto a sus diferentes tópicos; estos enlaces se clasificaron en diferentes secciones para su

fácil localización (animación, videojuegos, escuelas, estudios de animación, granjas de render, software, aplicaciones, realidad virtual, ambientes gráficos interactivos Etc.).



Ilustración 61 Enlaces

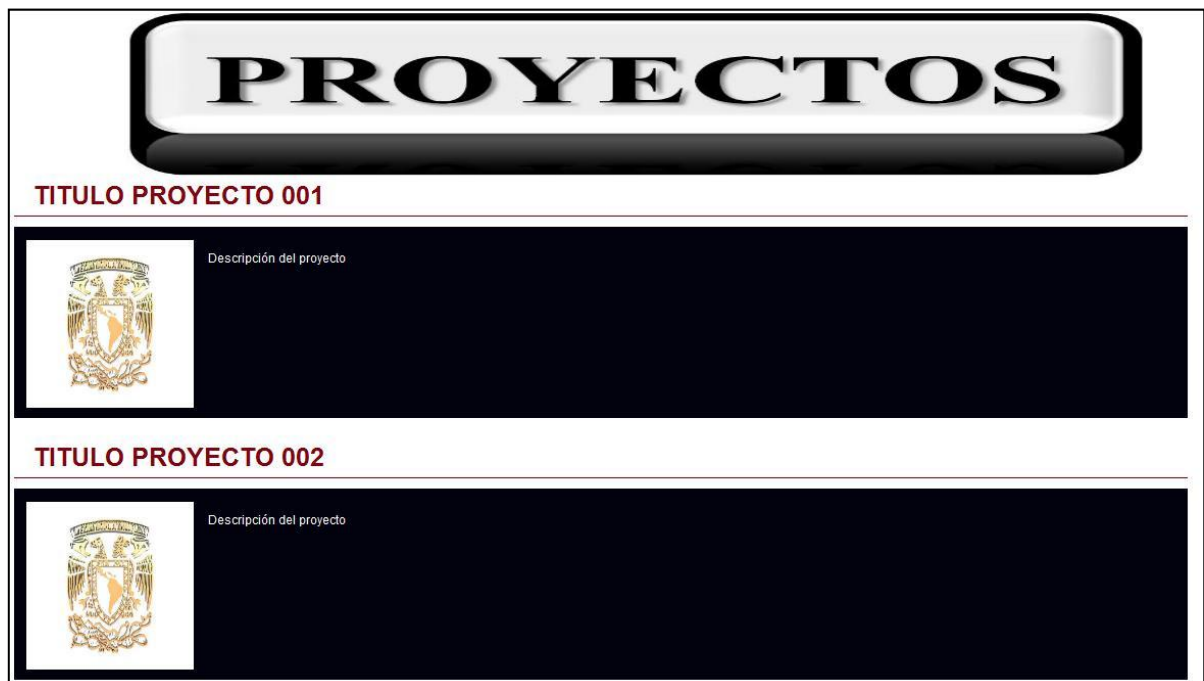


Ilustración 62 Proyectos

Resultados

El objetivo de desarrollar la sección proyectos es proporcionar información acerca de temas de tesis, servicio social, conferencias etc., todo aquello en lo cual los usuarios puedan tener interés pues son proyectos relacionados con la Computación Gráfica.



Ilustración 63 *Demo reel*

Con el desarrollo de la sección *Demo-Reel*, se pretende establecer un vínculo con la industria por medio de la exposición permanente del trabajo de los usuarios del portal, una de las consecuencias de esta exposición probablemente sea la generación de empleos relacionados con los gráficos por computadora como son: animadores, desarrolladores de videojuegos, desarrollo de aplicaciones CAD o GIS, tratamiento de imágenes, etc.

Resultados

El resultado de la sección “MATERIAS” es un conjunto de páginas *web* empleadas para exponer contenido audiovisual referente a las materias que componen al Módulo, imágenes, videos incluso texto; esperando con esto aclarar los conceptos o ideas que se tienen sobre estas, e informar de manera correcta a los usuarios que se inician en este increíble mundo.



Ilustración 64 Materias

La pestaña buscador muestra un motor de búsqueda interno y personalizado, el cual es aprovechado para ubicar los temas de interés sobre computación gráfica, primeramente dentro de la *webapp* y ya después a nivel de *Internet*; sirve como plataforma de navegación dentro de un gran número de sitios por explorar.

DESARROLLO DEL PORTAL WEB DEL MÓDULO DE COMPUTACIÓN GRÁFICA
DE LA FACULTAD DE INGENIERÍA, UNAM

Resultados



Ilustración 65 Buscador

The image shows a contact form titled 'CONTÁCTANOS' in a large, stylized font. The form has two columns. The left column contains input fields for 'Nombre', 'Email', and 'Teléfono'. The right column contains input fields for 'Asunto' and 'Mensaje'. Below the 'Mensaje' field are 'Enviar' and 'Borrar' buttons. Below the form is a section titled 'Dirección:' with the text 'Facultad de Ingeniería, Escolar Conjunto Norte facultad de Ingeniería, 04510 Ciudad de México, Distrito Federal.' Below the text is a map showing the location of the faculty, with labels for 'Estadio Olímpico México Sesenta y Ocho', 'UNAM', 'Biblioteca Central', 'Rectoría', 'Cd. Universitaria', 'Taller 3', 'Facultad de Ingeniería', 'Café Diseño', 'Torre II de Humanidades', 'Torre de Humanidades', 'Edificio A de Química', 'Edificio B de Química', 'Edificio C de Química', 'Facultad de Medicina', 'Facultad de Química', 'Escuela', 'González Peña', 'Alfonso Pineda', 'Rocabrava', 'Luis G. León', 'Escuela', 'Occidental', and 'Escuela'. The map includes a legend with 'Mapa', 'Sat.', 'Rel.', and 'Earth' buttons.

Ilustración 66 Contáctanos

El formulario que se desarrolló para entablar la comunicación entre los usuarios y los titulares del Módulo de Computación Gráfica (sección Contáctanos), da la oportunidad de hacer preguntas, comentarios y sugerencias sobre el portal web o bien sobre el mismo Módulo, es la manera más directa de comunicación entre los usuarios, el administrador del sitio y los profesores del Módulo.

Resultados

Ahora ya se tiene una página dentro del portal en la cual se exponen las noticias más relevantes en cuanto a gráficos por computadora se refiere; aquí se muestra que no solo se habla sobre animaciones o videojuegos, sino que este campo de desarrollo abarca ya incluso el mundo científico y que existe un mundo entero en el cual se puede desarrollar el trabajo del Módulo.



Ilustración 67 Noticias



Ilustración 68 Aportaciones

Resultados

Además de estas pestañas se desarrolló “Aportaciones”, esta sección será muy útil ya que es por medio de la cual los usuarios se pueden comunicar con el *webmaster* del portal, para poder enviar el material que hará crecer la *webapp* este material contempla desde definiciones, imágenes, material audiovisual, hasta información sobre eventos, proyectos, o sitios *web* de interés.

7.- Conclusiones

Se resolvieron las necesidades de información específica sobre el Módulo de Computación Gráfica ya que se puso al alcance del público en general toda la información concerniente a éste, como son horarios, directorio, plan de estudios, eventos, etc. Además de proporcionar material de apoyo académico (texto/definiciones, imágenes, videos, material multimedia etc.) con lo cual se espera obtener el máximo beneficio en el proceso de aprendizaje.

Se proporcionó una sección de la *webapp* en la cual se centralizaron hipervínculos, los cuales facilitarán la navegación web en temas relacionados con Computación Gráfica. Y se desarrolló otra sección a modo que se procure la vinculación entre los usuarios y la industria, esto mediante exposición permanente de sus trabajos gracias a la creación de *demo-reel's*.

Se creó una lista de proyectos, en los cuales los alumnos se pueden incorporar para cubrir servicio social, titulación, prácticas profesionales, además de fomentar el desarrollo de éstos en áreas como: visualización científica, animación y producción digital, desarrollo de videojuegos, desarrollo de interfaces y realidad virtual, CAD y GIS; entre otras.

Se diseñó y se desarrolló el portal web correspondiente al Módulo de Computación Gráfica de la Facultad de Ingeniería, UNAM. Este portal será utilizado como punto de partida para los usuarios que deseen tener acceso a una serie de recursos y servicios relacionados con los temas de Computación Gráfica desde definiciones hasta material propio del Módulo de la Facultad de Ingeniería; y servirá como plataforma para el desarrollo de los alumnos de esta facultad.

8.- Recomendaciones

En este capítulo se presentan algunas recomendaciones para que en un futuro próximo se pueda hacer crecer y mejorar el Portal web del Módulo de Computación Gráfica, además de que estas recomendaciones sirvan para que los compañeros (futuros tesis) puedan inspirarse y/o aprovecharlas para realizar sus trabajos de investigación (tesis).

Recomendación 1. Conectando con una base de datos. La primer recomendación es crear y conectar una base de datos con el portal; esto es para dar la posibilidad a los usuarios de registrarse en el sistema y así obtener algunos beneficios, entre las ventajas que esto representa esta un trato más personal puesto al saber que usuario es el que se encuentra conectado se pueden manejar mensajes personalizados, con esto también se pretende llevar un registro de las visitas y aportaciones de cada usuario para hacer menciones o bien de ser posible premiarlos.

Recomendación 2. Diseño del logo del Módulo. Como se constató mediante el *Brief* conceptual, el Módulo de Computación Gráfica de la Facultad de Ingeniería no cuenta con un logo representativo, el portal web muestra una imagen compuesta por elementos gráficos la cual fue proporcionada por el Ing. Genaro Andrés Garrido Lazcano; mediante un concurso para diseñar el logo se pueden obtener buenos resultados para tener algo representativo del Módulo el cual se mostrara no solo en la parte digital (*web*) sino también en carteles o incluso papeles oficiales pertenecientes al Módulo.

Recomendación 3. Mejorando el diseño estético. El portal web es totalmente funcional, pero al no contar con el apoyo de alguna persona dedicada al diseño gráfico se puede apreciar que el diseño estético si bien es agradable a la vista puede mejorar en gran medida al contar con esta asesoría; esto proporcionara un *plus* para atraer a un mayor número de usuarios y conservar a los que ya se tienen, al ser un portal especializado en gráficos por computadora el aspecto visual debe ser prioridad.

Recomendaciones

Recomendación 4. Posicionamiento *Web*. Si bien el portal ya esta optimizado para el posicionamiento *web* no se tiene que dejar este tema de lado, hay que estar en constante evolución para no tener un portal obsoleto y estar bien posicionado para tener un mayor tráfico, muchas visitas y generar mejores y más preparados usuarios.

El *SEO* también conocido como posicionamiento natural se genera a través de varias estrategias para optimizar el sitio *web*. Entre las estrategias útiles para el posicionamiento se encuentran:

- Estrategia de *Keywords*.
- Estructura del sitio.
- Links internos.
- Generación de Contenido.
- Links externos.

Recomendación 5. Publicidad (dar a conocer el portal Módulo). La existencia de este portal resuelve el problema planteado en un principio, pero si nadie lo conoce de poco servirá su existencia; por lo que se recomienda crear una campaña mediante la cual se dé a conocer su existencia y uso, bien sea con carteles en las instalaciones de la Facultad, mediante ligas en las diferentes páginas tanto de la UNAM como de la Facultad, e incluso de voz directa por parte de comunicados dirigidos por los profesores en sus clases ya sea teóricas o de laboratorio.

Recomendación 6. Incrementar el *Pagerank*. El *page-Rank* es un *Rank* que realiza *Google* para valorar y calificar la importancia de las páginas *web* se valora del 0 al 10 en grado de importancia, donde el 10 es el máximo valor y el 0 el mínimo (sirve para dar distinción). Aproximadamente cada tres meses *Google* hace una valoración donde califica a las páginas *web*, con esto algunas aumentan su *pagerank* y otras pueden llegar a disminuirlo, se sugiere que se consigan enlaces de páginas que tengan contenidos similares al del portal para así facilitar el aumento de este *pagerank*.

9.- Glosario

Apache: *Apache Web Server*, es un servidor de páginas *web* desarrollado por la *Apache Software Foundation*, organización formada por miles de voluntarios que colaboran para la creación de software de libre distribución. <http://www.apache.org/>

Buscador: Sitios en la red que actúan como herramientas que recopilan y estructuran sistemáticamente la información contenida en ella. Ayudan a los usuarios a buscar datos específicos, para lo cual se organizan en buscadores por palabras o índices y buscador temático o directorios.

CMS: Sistema de gestión de contenidos (en inglés *content management system*, abreviado *CMS*) es un programa que permite crear una estructura de soporte (*framework*) para la creación y administración de contenidos, principalmente en páginas *web*, por parte de los participantes.

CSS: (*Cascading Style Sheet*) Hojas de estilo en cascada. Un término usado para describir un método para separar el diseño *web*. También como un tipo de archivo utilizado para el diseño del estilo de la página *web*.

Las CSS se utilizan por los diseñadores *web* para controlar cómo los documentos se muestran en los navegadores *web*. CSS es realmente un lenguaje básico de programación que permite a desarrolladores definir como se exhiben a los usuarios los aspectos concretos de una página *web*. Elementos tales como fuentes, colores, disposición y acoplamientos se pueden todos controlar por CSS.

DNS: *Domain Name System*. Sistema de Nombres de Dominio. Servicio búsqueda de direcciones *IP* de servidores (*hosts*) basándose en sus nombres. El estilo de esos nombres es llamado nombre de dominio, consistente en una marca que puede ser o no comercial y un sufijo que representa la actividad. Por ello, los primeros sufijos para identificar dominios fueron: *.COM* (comercial- empresas), *.EDU* (educación, centros docentes), *.ORG* (organización sin fines de lucro), *.NET* (operación de la red), *.GOV* (Gobierno en Estados Unidos) y *.MIL* (Ejército en Estados Unidos). La mayoría de los países tienen un dominio propio, como *.AR* (Argentina), *.CL* (Chile) y *.MX* (México).

Doctype: Se utiliza para declarar el tipo de documento y define que tipo de documento compila un sitio *web*. En términos simples, *Doctype* especifica qué versión de *HTML* se ha utilizado para construir un sitio *web* particular. El código en este sitio *Web* debe entonces utilizar los estándares definidos por este *Doctype* y no contener ningunas cualidades o elementos de programación de otras versiones de *HTML*.

Domain: Dominio, sistema de denominación de *hosts* en Internet. Los dominios van separados por un punto y jerárquicamente están organizados de derecha a izquierda. En el caso de una dirección de correo electrónico, está ubicado a la derecha del signo @ e informa a un servidor hacia dónde dirigir la comunicación. Las direcciones en la red suelen ser una combinación de varios dominios de diferentes niveles. Los de primer nivel son los que identifican a países, como, por ejemplo, *.mx* para el caso de México.

Enlace: Conexiones basadas en tecnología de hipertexto, que pueden ser textos o gráficas, contenidas en una página *web*, que al ser activadas con un *clíc* permiten acceder a otras secciones del mismo documento, a otros documentos dentro de un sitio o a otros sitios, según la conveniencia del usuario. Cada uno de estos saltos también se denomina *link*.

Grupos de Noticias: Grupos de personas que tienen interés por un tema determinado y que se intercambian mensajes. A diferencia de las listas de discusión, no es necesario estar inscrito, sino que basta con acceder a ellos. Hay listas de noticias prácticamente para cualquier tema. Se puede participar anónimamente, sólo leyendo los mensajes o interviniendo y opinando. En principio, los únicos límites son ser respetuosos y mantenerse en la temática del grupo. Estos mensajes se almacenan en unos servidores especiales, que son los de noticias propiamente. Para encontrar grupos de discusión se puede utilizar.

GUI: (*Graphical User Interfaz*) Interfaz gráfica de usuario. Esto es lo que se muestra en una pantalla de ordenador. Un formulario de botones, por ejemplo.

Hipertexto: Se refiere a la posibilidad de pasar de un documento a otro relacionado, mediante una relación establecida usando el lenguaje *HTML* dentro de la plataforma de las páginas *web*.

Hipervínculo: *Link*. Es una palabra, una frase o un gráfico que cuenta con un hipertexto que conecta la página *web* que lo contiene, con otra información.

Homepage: Primera página o página de acceso inicial a un Sitio *web*. Suele presentar información general de las diferentes secciones que contiene. Es también el punto de inicio cuando un navegador se conecta por primera vez a la red.

HTML: (*HyperText Markup Language*) Lenguaje de marcado de hipertexto. Un lenguaje de codificación utilizado para hacer páginas *web*.

HTTP: (*HyperText Transfer Protocol*) Protocolo de transferencia de hipertexto. Un protocolo de comunicaciones utilizado para la entrega de páginas *web* a un navegador.

IE: *Internet Explorer*. Un navegador ampliamente utilizado, pero de muy mal desempeño para procesar páginas *web*.

iFrames: Con los *iFrames*, podemos insertar una página *web* dentro de otra, de la misma forma que insertamos un gráfico o una tabla. El *iFrame* es un objeto que se manipula con la misma libertad que cualquier otro elemento dentro del espacio *HTML*. Teniendo en cuenta el tiempo de carga, al usuario se le presenta una portada con el contenido del sitio de forma directa, cargando el resto de elementos (incluidos el *iframe*) de forma natural.

Interactivo: Capacidad de un sistema de permitir un intercambio continuo de información entre el usuario y la aplicación, respondiendo ésta a las órdenes de aquel.

Interfaz: Conexión e interacción entre hardware, software y el usuario.

Interfaz de usuario: Engloba la forma en la que el operador interactúa con la computadora, los mensajes que éste recibe en pantalla, las respuestas de la computadora a la utilización de periféricos de entrada de datos, etc.

Interfaz gráfico de usuario: Evolución de los interfaces de usuario que permiten una mejor y más fácil interacción con la computadora. Los interfaces gráficos -*Windows* es el ejemplo típico- permiten el aprendizaje intuitivo de los programas, facilitando y reduciendo el tiempo de formación y aumentando la productividad.

Internet: Es la red de redes. Conjunto de redes de computadores que conecta y comunica a millones de personas en todo el mundo. Es una red no comercial que nació en Estados Unidos en 1969 y está integrada por millones de computadores, llamados servidores, que comparten un lenguaje común. Los computadores personales que se conectan y consultan datos de los servidores se denominan clientes.

Internet Society: Organización privada que coordina el desarrollo de Internet, unificando criterios y protocolos para permitir el máximo acceso. La afiliación es abierta a toda la organización o usuario interesado.

IP: (*Internet Protocol*) Protocolo de Internet. Un protocolo de comunicaciones utilizado para acceder a la *web*. Esto se utiliza principalmente para definir un número de conexión o "dirección *IP*".

Javascript: Es un lenguaje de programación que se utiliza con frecuencia en diseño *web* para generar efectos más complejos que no se puedan alcanzar usando *HTML*. *Javascript* se utiliza para una gran variedad de usos, pero una de las aplicaciones más comunes está en la creación de los botones de la navegación que cambian de color o aspecto cuando el usuario mueve el cursor sobre el botón.

Link: Conexiones basadas en tecnología de hipertexto, que pueden ser textos o gráficas, contenidas en una página *web*, que al ser activadas con un *clíc* permiten acceder a otras secciones del mismo documento, a otros documentos dentro de un sitio o a otros sitios, según la conveniencia del usuario. Cada uno de estos saltos también se denomina *link*.

Mapa de Imágenes: Es un tipo especial de archivos de gráficos que permite al *browser web* proporcionar acceso hipermedia a otras áreas de *web*. Con un mapa de imágenes se pueden crear múltiples áreas de una imagen que proporcionan enlaces a otras zonas.

Meta tag: Es una etiqueta de *HTML* que describe algunos de los aspectos de una página *web*. La información proporcionada a través de estas etiquetas, que se instalan en su parte inicial, es usada por los sistemas de búsqueda para ingresar la información y luego ofrecerla a usuarios que buscan información relacionada.

Motores de búsqueda: Son sistemas de búsqueda por palabras clave. Son bases de datos que incorporan automáticamente páginas *web* mediante “robots” de búsqueda en la red.

MPEG: *Motion Pictures Expert Group*. Grupo de Expertos en Imagen en Movimiento. Estándar de compresión de vídeo digital que utiliza altos niveles de compresión, con bajas pérdidas alcanzando relaciones de tamaño / resolución muy altos.

Navegador o Browser: Un *browser* o navegador es un programa de software usado para ver un sitio *web* y para navegar en Internet. Existen varios *browsers* disponibles gratuitamente para los usuarios, tal como *Internet Explorer*, *Mozilla Firefox* y *Ópera*. Algunos usuarios pueden también hacer uso de los navegadores de texto o de lectores de la pantalla para ver los sitios *web* que se han sido diseñados específicamente para los usuarios con discapacidad visual.

Opera: *Browser* de Internet desarrollado por una empresa en Noruega. Aparece como el tercer competidor en este segmento dominado por *Microsoft Explorer* y *Netscape*.

Página web: Archivo o unidad básica de información en la red que se construye con lenguaje *HTML*. No tiene un límite en cuanto a su dimensión y se accede a ella a través de su dirección específica o de un enlace. Es visible en un *browser* o programa cliente.

PDF: (*Portable Document Format*) Formato de Documento Portátil. Un documento electrónico que debe leerse en su computadora con el programa *Adobe Acrobat*.

PHP: (*Hypertext Preprocessor*) Preprocesador de Hipertexto. Lenguaje de programación que se procesa en el servidor. Lenguaje interpretado parecido a *JavaScript* o *VBScript* y muy utilizado en los servidores *web* bajo *Linux* para la construcción de páginas *web*. Es una alternativa a la tecnología *ASP* de *Microsoft*. La sigla proviene de *Personal Home Page Tools*, que fue su nombre original.

PNG: *Portable Network Graphics*. Gráficos Portables de Red. Formato gráfico muy completo especialmente pensado para redes.

Portal: Sitio *web* que ofrece múltiples prestaciones y servicios con diferentes canales temáticos y secciones de noticias, enlaces de interés, foros, etc. Su objetivo es convertirse en la página referencia del mayor número de navegantes posible para captar un gran tráfico. Algunos de ellos son *Yahoo*, *Terra*, etc. También se les denomina portales horizontales (en contraposición a portal vertical) por ofrecer contenidos de varias áreas.

SEO (*Search Engine Optimisation*) Optimización en buscadores: Consiste en aplicar diversas técnicas tendientes a lograr que los buscadores de Internet sitúen determinada página *web* dentro de su página de resultados para determinadas palabras, términos y frases clave de búsqueda.

Servlet: Aplicación sin interfaz gráfica que se ejecuta en un servidor de Internet, procesando información *HTML* previamente recogida por un navegador.

Sitemap: Es un archivo *XML* que lista las páginas en un sitio *web*, organizadas comúnmente de forma jerárquica. Esto ayuda a los visitantes y a los motores de búsqueda a hallar las páginas en un sitio *web*.

Sitio Web: Conjunto de páginas *HTML* visibles permanentemente en Internet. Cada Sitio Web se identifica mediante una dirección (*URL*), que es la de la página principal o portada. Siguiendo los enlaces que se encuentran en la portada, se pueden visitar todas las páginas.

Posicionamiento Web: El posicionamiento en buscadores o posicionamiento *web* es el resultado de la recuperación de información en la base de datos de los grandes Motores de Búsqueda de Internet por el uso de robots que utilizan algoritmos de búsqueda.

Upload: Cargar. Proceso de transferir información desde un computador personal a un servidor de información. Es el antónimo de *download*.

URL: (*Uniform Resource Locator*) Localizador de recursos uniforme. También una dirección *web*. Sistema unificado de identificación y localización de recursos de cualquier tipo en Internet, que permite acceder en forma sencilla y homogénea a los documentos que contiene. Esta denominación representa una dirección de Internet y un recurso concreto dentro de esa dirección.

Vínculos rotos / Callejón sin salida (*Dead Link/Dead End*): Los vínculos rotos de un sitio *web* se refieren a cuando un usuario hace *clic* en un vínculo y te lleva a una página que no existe, entonces da lugar a que se presente una página de error en el navegador, o bien no realice ninguna operación. Un callejón sin salida en un sitio *web* ocurre cuando un usuario hace *clic* en un vínculo a una página que cuando está cargada no proporciona ninguna manera de volver de nuevo a la página anterior o a ningún otro sitio *web*, por lo tanto el usuario ha alcanzado un “callejón sin salida”.

Visitor: Visitante. Usuario que ingresa a un Sitio *web*, quien puede ser identificado por medio de diversos mecanismos.

W3C: *World Wide Web Consortium*. La organización que rige todas las cosas de la *web*. Consorcio compuesto por un conjunto de actores de la industria de las tecnologías de información (compañías, universidades, centros de investigación, personajes destacados) cuyo objetivo es promover la evolución e interoperabilidad del *web*. <http://www.w3c.org/>

WAI: (*Web Accessibility Initiative*) Iniciativa de Accesibilidad en la Web. Un segmento de la W3C enfocada en la accesibilidad a la *web*.

WCAG: (*Web Content Accessibility Guidelines*) Lineamientos de accesibilidad al contenido web. Estas son normas escritas que rigen la accesibilidad a la *web* publicada por la WAI del W3C.

Web 2.0: Término acuñado por Tim O'Reilly en 2004 para referirse a una segunda generación de *web* basada en comunidades de usuarios y una gama especial de

servicios, como las redes sociales, los *blogs*, que fomentan la colaboración y el intercambio dinámico de información entre los usuarios.

Web Hosting: Es el hospedaje de un sitio *web*. Es decir, el arriendo de espacio, de disco y de computadores que una empresa le ofrece a otra para tener presencia en Internet.

Webmaster: Persona encargada de la administración y mantenimiento de un Sitio *web*.

WWW, WEB o World Wide Web: Es una forma de representar la información en Internet basada en páginas. Una página puede incluir tres tipos de información: texto, gráficos e hipertexto (texto resaltado que, al activarlo, se carga en otra página *web*). Fue creada a principios de la década del 90 y está compuesta por servidores que proveen información organizada en sitios -que contienen páginas relacionadas- a los que se accede fácilmente gracias a los programas navegadores.

XHTML: (*Extensible Hyper Text Markup Language*) Lenguaje de marcado de hipertexto. Una forma más extensible o amplia del *HTML*. *XHTML 1.0* es el mismo conjunto de elementos y atributos, escrita para ajustarse a las normas de *XML*. Se puede saber si un documento independiente está bien formado en *XML*, porque requiere que todas las etiquetas de apertura contengan las etiquetas de cierre.

XML: *eXtensible Markup Language*. Un lenguaje que se utiliza para ampliar al *XHTML* y es la base de la *RSS*. Permite definir el detalle de contenidos a nivel global del sitio, identificar secciones, tipos de páginas, etc. Luego de tener el panorama completo del sitio debemos pasar a identificar y modelar el detalle de contenidos a nivel de página.

10.- Consultas realizadas

ROLDÁN MARTÍNEZ, David; VALDERAS ARANDA, Pedro J.; PASTOR LÓPEZ, Óscar. Aplicaciones web: un enfoque práctico. Editorial Alfaomega, 1a edición, España 2010.

HARRIS, Andy. Manual imprescindible de Programación con PHP 6 y MySQL. Editorial ANAYA Multimedios, traducción de Pedro Dans Álvarez, Madrid España 2009.

SZNAJDLEDER, Pablo. Java a fondo: estudio del lenguaje y desarrollo de aplicaciones. Editorial Alfaomega, 1ª edición Buenos Aires 2010.

MINERA, Fransico; Desarrollo Web profesional, 1ª edición Banfield 2008.

Aplicaciones profesionales en internet, centro de computación profesional de México. Editorial PEARSON Prentice Hall, México 2007.

LÓPEZ, José; Domine HTML Y DHTML. Editorial alfaomega, 2da edición México 2008.

WEINSCHENK, Susan; Haz clic aquí: Neuro Web Design. Editorial PEARSON Prentice Hall, Madrid España 2009.

DIAZ, Ma. paloma; MONTERO, Susana; AEDO, ignacio; Ingeniería de la Web y patrones de diseño. Editorial PEARSON Prentice Hall, Madrid 2005.

DE LUCA, Damián; Webmaster Profesional. 1ª edición Banfield 2009.

Diccionario de informática, Editorial Cultural, S.A. Madrid España 2001.

PRESSMAN, Roger S.; Ingeniería del software: un enfoque práctico. Editorial McGraw-Hill Interamericana Sexta edición 2005.

Consultas realizadas

Tutorial para desarrollo web. Recuperado el 17 de Mayo del 2011, de <http://www.maestrosdelweb.com/>

Tutorial de desarrollo Web. Recuperado el 6 de Junio del 2011, de <http://www.htmlcodetutorial.com/>

Manuales para desarrollo web, creación y consulta. Recuperado el 9 de Junio del 2011, de <http://www.desarrolloweb.com/>

Estándares para desarrollo web, por parte de la UNAM. Recuperado el 20 de Junio del 2011, de <http://www.unam.mx/>

Material recopilado para el portal (horarios, temarios, plan de estudios, etc.). Recuperado el 20 de Junio del 2011, de <http://ingenieria.unam.mx/>

Estándares para el desarrollo web. Recuperado el 23 de Junio del 2011, de <http://www.w3c.es/>

Herramientas para desarrollo y navegadores (Web Design and Applications). Recuperado el 23 de Junio del 2011, de <http://www.w3.org/standards/agents/>

Diseño y aplicaciones Web (Browsers and Authoring Tools). Recuperado el 29 de Junio del 2011, de <http://www.w3.org/standards/webdesign/>

Arquitectura Web (Web Architecture). Recuperado el 29 de Junio del 2011, de <http://www.w3.org/standards/webarch/>

La web semántica (Semantic Web) Recuperado el 5 de Julio del 2011, de <http://www.w3.org/standards/semanticweb/>

Consultas realizadas

Tecnología XML (XML Technology). Recuperado el 12 de Julio del 2011, de <http://www.w3.org/standards/xml/>

Servicios en la web (Web of Services) Recuperado el 18 de Julio del 2011, de <http://www.w3.org/standards/webofservices/>

Buscador personalizado para las páginas web. Recuperado el 20 de Septiembre del 2011, de <http://www.google.com/cse/>

Sitio oficial de la sociedad de computación grafica. Recuperado el 10 de Marzo del 2012; <http://www.cgsociety.org>