

**FUNDAMENTOS BÁSICOS PARA CREAR UNA INTERFAZ GRÁFICA PARA
UN APLICATIVO WEB**

**ANGHIE VELASCO MAMIÁN
ALEJANDRO VARÓN TORRES**

**FUNDACIÓN UNIVERSITARIA CATÓLICA - LUMEN GENTIUM
UNIDAD DE POSGRADOS Y EDUCACIÓN CONTINUA
CALI, VALLE DEL CAUCA**

2016

**FUNDAMENTOS BÁSICOS PARA CREAR UNA INTERFAZ GRÁFICA PARA
UN APLICATIVO WEB**

**ANGHIE VELASCO MAMIÁN
ALEJANDRO VARÓN TORRES**

**Trabajo presentado como requisito parcial de grado para optar al título de
INGENIERIO DE SISTEMAS**

Director/Asesor

**FUNDACIÓN UNIVERSITARIA CATÓLICA- LUMEN GENTIUM
UNIDAD DE POSGRADOS Y EDUCACIÓN CONTINUA
CALI, VALLE DEL CAUCA**

2016

Dedicatoria

*Gracias Dios,
Gracias familia,
Gracias amigos,
Gracias maestros,
Gracias compañeros,
Gracias Jacke&Hector,
Gracias por hacer esto posible.*

AGRADECIMIENTOS

Damos gracias a nuestros familiares, profesores, amigos y compañeros que estuvieron presentes en este proceso de convertirnos en ingenieros de sistemas con ética profesional, acompañándonos en cada triunfo y fracaso de nuestras vidas.

A todos ustedes, infinitas gracias por cada palabra de aliento, que nos animó a ser mejores.

CONTENIDO

AGRADECIMIENTOS iv

INTRODUCCIÓN3

1. Planteamiento del problema.....3

2. Justificación.....4

3. Objetivos4

3.1. Objetivo general.....4

3.2. Objetivos específicos4

4. Marco teórico.....4

4.1. Antecedentes4

4.1.2. La Web6

5. Estado del arte7

5.1. La Web 2.0.....7

5.2. Aplicación Web7

5.2.1. Tipos de desarrollo de aplicaciones web7

5.3. Interfaz gráfica de usuario.....8

6. Conceptos básicos de una interfaz gráfica para un aplicativo web.9

6.2. Tipo de Usuario.....9

6.3. Usabilidad9

6.4. Fiabilidad.....9

6.5. Seguridad.....10

7. Principios esenciales para desarrollar una interfaz gráfica de un aplicativo web.
10

7.2. Familiaridad con el usuario10

7.3. Uniformidad de la interfaz10

7.4. Mínima sorpresa11

7.5. Recuperabilidad11

CONCLUSIÓN11

REFERENCIAS BIBLIOGRÁFICAS.....12

FUNDAMENTOS BÁSICOS DE UNA INTERFAZ GRÁFICA PARA UN
APLICATIVO WEB

Anghie Velasco Mamián

anghievelasco@gmail.com

Alejandro Varón

alejandrovartorres@gmail.com

Resumen

Este artículo explora los fundamentos básicos que debe tener en cuenta un desarrollador a la hora de implementar el diseño de interfaz gráfica de usuario para un aplicativo web, sin caer en largos y desgastantes procesos ineficientes, producto de la falta de experiencia del desarrollador o la falta de comunicación entre las partes interesadas. También teniendo en cuenta que el desarrollador debe estar a la vanguardia de la época, haciendo uso eficiente de los recursos que le proporciona la tecnología.

Palabras claves: interfaz gráfica de usuario, aplicativo web, web, usabilidad, fiabilidad, seguridad.

Abstract

This article explores the basics that should consider a developer when implementing the design of graphical user interface for a web application, without falling into long and exhausting inefficient processes, due to lack of developer experience or lack of communication among stakeholders. Also given that, the developer should be at the forefront of the time, making efficient use of the resources that technology provides.

Keywords: Graphical User Interface, web application, web, usability, reliability, security.

INTRODUCCIÓN

Actualmente empresas desarrolladoras de software y los desarrolladores no poseen claridad en el instante de realizar avances o progresos para las interfaces graficas diferentes proyectos web y enfoques.

Por lo anterior se pretende definir fundamentos básicos para la creación de una interfaz gráfica de un aplicativo web, se deberá conocer la historia desde los inicios, como se creó y como ha cambiado a través del tiempo, además se basara en conceptos teóricos que serán el ¿cómo está compuesta una interfaz gráfica? ¿Qué es lo que se debe de conocer para implementarla?,

Tener claridad en los conceptos básicos es sumamente importante dado que de esta manera se podrá realizar el desarrollo de manera eficiente y eficaz, ya que al final el usuario es el encargado de juzgar el nivel de usabilidad que se genere en la interacción ya sea en el momento de la comunicación con el sistema o manipulación de la información de la mejor manera.

1. Planteamiento del problema

Cuando los usuarios tienen un problema o una necesidad de un aplicativo web suelen acudir a personas que tengan experiencias en ese entorno, exponiendo las necesidades de forma verbal y escrita, al momento de recibir esa información el sujeto encargado de recibirla la interpreta y expone la solución

después de un análisis basado a su experiencia. El sujeto no tiene unas directrices establecidas para realizar una mejor aplicación y dar un valor agregado, causando una satisfacción buena más no excelente por parte del usuario, este procedimiento se vuelve cíclico hasta llegar un acuerdo con el usuario que tiene la necesidad, cuando se comienza a realizar el desarrollo comienzan a llegar muchas interrogantes del cómo sería la manera más óptima de realizar las pantallas.

De tal manera que si existieran unas directrices ya establecidas, sería más fácil realizar el análisis y la propuesta para el usuario.

2. Justificación

Este artículo pretende encontrar una buena manera de diseñar interfaces gráficas de usuario. Puesto que el modo de diseñar debe ir de la mano con la evolución de pasos agigantados sobre tecnología a la hora de usar herramientas que permitan un diseño usable, fiable y seguro para perfilar aplicativos web.

3. Objetivos

3.1. Objetivo general

Proponer los fundamentos básicos necesarios para diseñar una interfaz gráfica de un aplicativo web.

3.2. Objetivos específicos

- Identificar los conceptos básicos de una interfaz gráfica de un aplicativo web.
- Sintetizar los principios esenciales para desarrollar una interfaz gráfica de un aplicativo web.

4. Marco teórico

4.1. Antecedentes

En un principio la Web era sencillamente, una colección de páginas estáticas, documentos, etc., para su consulta o descarga. El paso inmediatamente posterior en su evolución fue la inclusión de un método para elaborar páginas dinámicas que permitieran que lo mostrado tuviese carácter dinámico

(es decir, generado a partir de los datos de la petición). Este método fue conocido como CGI ("Common Gateway Interface") y definía un mecanismo mediante el que se podía pasar información entre el servidor y ciertos programas externos.

Los CGIs siguen utilizándose ampliamente; la mayoría de los servidores web permiten su uso debido a su sencillez. Además, dan total libertad para elegir el lenguaje de programación que se desea emplear.

El funcionamiento de los CGIs tenía un punto débil: cada vez que se recibía una petición, el servidor debía lanzar un proceso para ejecutar el programa CGI. Como la mayoría de CGIs estaban escritos en lenguajes interpretados, como Perl o Python, o en lenguajes que requerían "run-time environment", como Java o Visual Basic, el servidor se veía sometido a una gran carga. La concurrencia de múltiples accesos al CGI podía comportar problemas graves.

Por eso se empiezan a desarrollar alternativas a los CGIs que solucionaran el problema del rendimiento. Las soluciones llegan básicamente por 2 vías:

1. se diseñan sistemas de ejecución de módulos mejor integrados con el servidor, que evitan la instanciación y ejecución de varios programas.
2. se dota a los servidores un intérprete de algún lenguaje de programación que permita incluir el código en las páginas

de forma que lo ejecute el servidor, reduciendo el intervalo de respuesta.

Entonces se experimenta un aumento del número de arquitecturas y lenguajes que permiten desarrollar aplicaciones web. Todas siguen alguna de estas vías. Las más útiles y las más utilizadas son las que permiten mezclar los 2 sistemas: un lenguaje integrado que permita al servidor interpretar comandos "incrustados" en las páginas HTML y, además, un sistema de ejecución de programas mejor enlazado con el servidor, que no implique los problemas de rendimiento propios de los CGIs.

Una de las más potentes es la seguida por Sun Microsystems con su Java, integrado por 2 componentes; un lenguaje que permite la incrustación de código en las páginas HTML que el servidor convierte en programas ejecutables, JSP ("Java Server Pages" o "Páginas de Servidor de Java"), y un método de programación muy ligado al servidor, con un rendimiento superior a los CGIs, denominado "Java Servlet".

Otra tecnología de éxito y una de las más utilizadas es el lenguaje PHP. Se trata de un lenguaje interpretado que permite la incrustación de HTML en los programas, con una sintaxis derivada de C y Perl. El hecho de ser sencillo y potente ha contribuido a hacer de PHP una herramienta muy apropiada para determinados desarrollos.

A través de los años se fueron ofreciendo introduciendo nuevos cambios:

En 1995, el navegador Netscape introdujo el lenguaje de scripts de nombre JavaScript para funcionar del lado del cliente. Esto permitió a los programadores agregar algunos elementos dinámicos a la interfaz de usuario que se veía en el lado del cliente. Entonces, en lugar de enviar todos los datos al servidor para generar una página web completa, los scripts embebidos (incrustados) en la página web podían realizar ciertas tareas sin realizar todo ese procedimiento: validación de entradas, mostrar u ocultar partes de una página web, etc.

En 1996, Macromedia lanzó Flash, un reproductor de animación por vectores que podía agregarse a los navegadores como plugin para así insertar animaciones e interactividad en las páginas web. También permitía usar lenguajes de scripts en el lado del cliente para que no haya necesidad de comunicarse con el servidor en las interacciones.

En 1999, el concepto de "aplicación web" fue introducido al lenguaje Java en el Servlet Specification versión 2.2. Al mismo tiempo se desarrolló JavaScript y XML, pero aún Ajax no había sido acuñado. El objeto XMLHttpRequest recién se introdujo en Internet Explorer 5 (marzo de 1999) como un objeto ActiveX.

En 2005, fue acuñado el término Ajax, y nacieron aplicaciones web sumamente interactivas (como Gmail de Google). Ajax permite que un script

se contacte con el servidor para almacenar o recibir datos sin necesidad de descargar o actualizar la página web completa.

En 2011, fue finalizado HTML5, que ofrece características multimedia y de gráficos a una página web sin la necesidad de plugins del lado del servidor. Las aplicaciones Flash comienzan a morir poco a poco.

HTML5 también enriquecen la semántica del contenido de los documentos. Se mantiene el concepto de DOM (document object model), diseñado muchos años antes.

HTML5 introdujo soporte de lenguaje explícito para hacer aplicaciones que son abiertas como página web, pero pueden almacenar datos localmente y continuar funcionando mientras se está fuera de línea.

La API WebGL permite gráficos 3D avanzados basados en canvas HTML5 y lenguaje JavaScript. Estas tienen una importancia significativa en la creación de verdaderas aplicaciones web independientes del navegador y la plataforma. (Alegsa, 2016).

4.1.2. La Web

Después del surgimiento del internet viene este concepto el cual Tim Berners-Lee fue el padre de la web el cual tiene tres elementos claves: HTML como lenguaje para crear los contenido e la web, HTTP es el protocolo usado para la conexión del

usuario la web, y la URL es la dirección del recurso del web. En 1990 Tim Berners-Lee publica la primera Web el cual se comenzó a realizar investigación e implementaciones en el lenguaje de la web.

El primer navegador se programó en el lenguaje Objective C en un ordenador NeXT con la ayuda de la tecnología What You See Is What You Get (WYSIWYG).

La primera generación surgen cuando se crearon las páginas web los recursos eran muy limitados desde la navegación del internet hasta las herramientas, por lo tanto se emplea la web como usos tradicionales como libros, revistas el cual nomina como la primera generación.

La segunda generación se implementan imágenes, iconos, fondos de imágenes aunque siguen con algunas desventajas como la carga lenta de las paginas, muchos enlaces para llegar a un punto específico, surgen plug-in para la multimedia, Después surgen la tercera generación donde nace el concepto Cascading Style Sheets (CSS) el cual optimiza el código HTML donde nacen páginas con diferentes enfoques como informar, vender, ofrecer servicios ETC.

En la tercera generación se enfoca en el diseño de las páginas web para poder que sean más llamativas para ser visitadas y toman principio básico de usabilidad y de accesibilidad.

La cuarta generación donde se implementan más recursos gráficos, también se extiende el uso de la tecnología CSS y se comienzan a usar

herramientas multimedia, y el uso excesivo de gráficos. La definición de seguridad se centra en la prevención de amenazas, situación que en la actualidad es muy difícil de sobrellevar, la sociedad está ligada a riesgos los mismos que aparecen por cada una de las actividades que se realizan hoy en día. El riesgo es permanente y de carácter propio de la sociedad, por ende la seguridad se centra en la prevención de dichos riesgos (Luján, M.S, 2002).

5. Estado del arte

5.1. La Web 2.0

(O'Reilly & Battelle, 2009) Este concepto es relativamente nuevo, **O'Reilly Media** una compañía que principalmente está enfocada a libros de tecnología e informática expuso que la web 2.0 está apuntando a la inteligencia colectiva para lograr que las interfaces graficas de las páginas web sea más simples y puntales, esto se debe a que hubieron avances tecnológicos algunos sitios web fueron denominados la web 2.0 como Amazon, (Amazon,1996-2016) “En Amazon, siempre y cuando hagamos bien nuestro trabajo, nuestra mayor contribución al bienestar social se manifiesta a través de las actividades propias de nuestro negocio: precios bajos, creciente selección, facilidad de compra, sistema de empaquetado abrefácil, Kindle, innovación en los servicios web y otras iniciativas en las que trabajaremos duro en un futuro”.

“los principios de compartir, reutilizar, mejora continua, consideración del usuario como fuente de información, confianza, aprovechamiento de la inteligencia colectiva, etc., los que han impulsado el establecimiento de la actitud 2.0, haciendo que la tecnología pase a un segundo plano. En la actualidad cuando se habla de web 2.0 se está haciendo referencia al uso de determinadas tecnologías (ajax, mashups, software social, rss) y a la aplicación de una determinada actitud en el diseño de servicios web, basada en los principios citados”. (Margaix, 2007).

5.2. Aplicación Web

La aplicación web es la respuesta a una necesidad o problemática, está compuesta de páginas web en el que el usuario final pueda navegar con las directicas propuestas en el sistema, estas se solicitan a un servidor web por medio del internet con acción realizada por el visitante por medio de una herramienta física, en la actualidad son muy usada a causa del fácil acceso y la utilidad que ofrece para diferentes enfoques de trabajo.

5.2.1. Tipos de desarrollo de aplicaciones web

El desarrollo de las aplicaciones web es dependiente del contenido que debe mostrar y así poder aplicar, para

ello se debe de conocer los tipos de aplicaciones web, y son las siguientes:

- Aplicaciones web estáticas: son páginas web que no utilizan bases de datos el cual la información que se visualiza solo se puede modificar a nivel de servidor, el cliente (usuario final) solo puede consultarla.

El contenido básicamente solo necesita código HTML o XHTML un ejemplo de una página web estático puede ser una página que tenga el fin de mostrar la información de una empresa.

- Aplicación web dinámica: son páginas web que utilizan bases de datos donde el usuario puede consultar y modificar la información a petición del cliente (usuario final). Se utilizan lenguajes como HTML, ASP, JSP. Un ejemplo muy claro de una aplicación web dinámica puede ser un foro donde las personas puedan ingresar información en cualquier momento.

5.3. Interfaz gráfica de usuario

Para definir que es la interfaz de usuario vamos a conocer cada uno de los conceptos y poder tener una definición clara, comencemos con la interfaz, ¿qué es? “Inter proviene del latín inter, y significa, “entre” o “en medio”, y Faz proviene del latín facies, y significa “superficie, vista o lado de una cosa”. Por lo tanto una traducción

literal del concepto de interfaz atendiendo a su etimología, podría ser “superficie, vista, o lado mediador”.”

Otra definición en el enfoque de la informática “Conexión, física o lógica, entre una computadora y el usuario, un dispositivo periférico o un enlace de comunicaciones.”

Ahora si definamos que es la interfaz de usuario “En los sistemas informáticos, la relación humano-computadora se realiza por medio de la interfaz, que se podría definir como mediador. Cuando existen dos sistemas cualesquiera que se deben comunicar entre ellos la interfaz será el mecanismo, el entorno o la herramienta que hará posible dicha comunicación.

Podríamos definir básicamente dos tipos de interfaces:

- La interfaz física: un ratón y un teclado que sirven para introducir y manipular datos en nuestro ordenador.
- La interfaz virtual o interfaz gráfica (GUI) que permite, mediante iconos (cursor + objetos gráficos metafóricos), interactuar con los elementos gráficos convirtiendo al ser humano en usuario de la aplicación.

Estas dos mediaciones son relaciones del tipo entrada de datos (*input*). Al igual que tenemos una entrada, necesitamos algo que facilite la salida de datos (*output*), para esto tenemos, por ejemplo, la pantalla de la computadora, donde se visualizan estas interfaces gráficas, o la

impresora, donde se imprimen los datos” (Marrero, 2004-06).

6. Conceptos básicos de una interfaz gráfica para un aplicativo web.

6.2. Tipo de Usuario

Cuando se toman decisiones de diseño se debe tener cuenta lo siguientes factores humanos:

- Las personas tiene limitada memoria a corto plazo por lo tanto si a un usuario se le presenta demasiada información no podrá asimilarla.
- Todos cometemos errores especialmente cuando hay exceso de información, también cuando los sistemas fallan
- Los seres humanos tiene capacidades diferentes
- Tenemos diferentes preferencias de interacción, algunas personas les gusta trabajar con imágenes, otras con texto, etc.

En el momento de crear un aplicativo web primero se debe tener claro quién es el cliente, con quien se va tratar en toda la duración del proyecto, cuál es su necesidad.

También debe conocer el usuario final(Usuario que va utilizar el aplicativo), cuales son las tareas que va realizar, el entorno de trabajo, los sistemas que utilizan, para ser más efectivo el diseño esto se debe hacer

en un grupo de discusión con varios usuarios y así llegar a un diseño final.

Por experiencia el usuario cliente va dar los requisitos (necesidades) que debe tener como funcionalmente y lo requisitos no funcionales que comúnmente lo ofrece quien está ofreciendo el producto.

Muchas veces el desarrollador o la empresa quien ofrece el servicio solo se enfoca en los requerimientos funcionales dejando como ultima prioridad los requerimientos no funcionales como la usabilidad, fiabilidad y seguridad, a continuación se definirán los conceptos principales para los requerimientos no funciones

6.3. Usabilidad

Este concepto se refiere básicamente a facilidad de usar un producto en este caso un aplicativo web, Para ir más allá se define el concepto “Accesibilidad” que su función es unir información web con los usuarios el cual puedan obtener información sin ninguna dificultad.

Por otro lado, está la “Navegabilidad” el cual permite al usuario navegar fácilmente por un sitio web. (Ref. Diseño de materiales multimedia).

6.4. Fiabilidad

La Real academia española lo define “Probabilidad de buen funcionamiento de algo”, este concepto es muy importante ya que esto brinda confianza al momento de ingresar, modificar, consultar y eliminar información del aplicativo web que

está usando, para lograr este objetivo se debe tener en cuenta lo siguiente:

- Tratar la información en el tamaño de toda la página para que el usuario no tenga que tener que desplazarse hacia abajo para seguir viendo toda la información completa
- No saturar la página con mucha información
- Cuando el usuario ingrese información la página debe tener claro como ingresarla intuitivamente, en el caso que la información que deba ingresar tiene un grado de complejidad debe indicar como hacerlo
- En el momento modificar la información en el aplicativo mostrar mensaje dinámico para que el usuario tenga la certeza de que el procedimiento se realizó correctamente
- Cuando el usuario quiera realizar modificaciones de la información debe mostrar mensaje dinámicos de confirmación

6.5. Seguridad

La seguridad viene de la mano con la usabilidad ya que en el momento de generar métodos de seguridad puede afectar la usabilidad, pero existe métodos fáciles de usar para el usuario legítimo y difícil para atacar para el usuario ilegítimo pero es recomendable usar credenciales de acceso el cual va de la mano con la información del sistema el cual pueda acceder a toda la información, también utilizar métodos para mostrar URLS

que ilegítimos no puedan acceder. (Romero, M. & Vásquez, M. 2016)

7. Principios esenciales para desarrollar una interfaz gráfica de un aplicativo web.

7.2. Familiaridad con el usuario

El usuario final no debe adaptarse a la interfaz gráfica por más conveniente que sea el diseño, por lo contrario debe ser familiar para el usuario, deben estar relacionados con el entorno de trabajo, por ejemplo en un aplicativo web de carros los componentes que utiliza debe estar de relacionado con carros. (Sommerville, 2005).

7.3. Uniformidad de la interfaz

La interfaz debe ser uniforme, así el usuario tendrá un aprendizaje más eficaz, para cumplir con ese principio el aplicativo web debe tener lo siguiente:

- Definir Colores principales, manejar un color estándar para el fondo, el menú y diferente componentes como tablas, campos, botones(En este punto se aplicaría las CCS para toda la aplicación)
- Definir los encabezados para página de la aplicación con un color, tipográfica e información estándar
- Definir el pie de página para todas las páginas web con un

color, tipográfica e información estándar

- No saturar las páginas con excesiva información, buscar la forma de mostrar o conllevar al usuario a la información que desea.
- Definir el tamaño y la tipográfica del texto que se muestran en las páginas dependiendo de la prioridad de la información como títulos,
- No abusar con las tipográficas en la aplicativo (dos o tres como máximo).
- Definir todos los modelos de contenido (La información dinámica donde se puede presentar diferentes maneras.
- La información debe ser clara y asertiva.
- Utilizar el mismo tipo de componentes en todas las páginas. Ejemplo: un estilo estándar para todas las tablas. (Ref. Diseño de materiales multimedia).

7.4. Mínima sorpresa

(Sommerville, 2005). El aplicativo debe de tener los mismos métodos de trabajo y así no sorprender al usuario con cambio muy representativos que puede llegar a confundirlo, se puede generar diferentes métodos de trabajo, por ejemplo método de ingreso de información y el otro método sería edición de información, donde el funcionamiento puede ser distinto.

- Evitar que el usuario al momento de navegar quede sin

salida creando hipervínculos u otro tipo de componente para navegar en la pantalla.

7.5. Recuperabilidad

(Sommerville, 2005). El aplicativo debe permitir al usuario recuperarse de los errores, para cumplir este principio se debe tener en cuenta lo siguiente:

- En el momento de realizar modificaciones en la información se debe mostrar mensajes de confirmación para asegurar que el usuario si desea realizar la transacción.
- Reversar transacciones realizadas
- Generar puntos de control en el aplicativo en caso de que el usuario quiera reversar procesos ya realizados.

CONCLUSIÓN

Como resultado de la investigación es posible concluir que en el momento de crear una interfaz gráfica de usuario para un aplicativo web más óptima, se deben tener en cuenta los conceptos básicos que conforman la interfaz gráfica, como los tipos de usuario que utilizaran la aplicación, ¿cómo debe ser la usabilidad de la aplicación a desarrollar?, ¿cómo se debe comportar la aplicación al momento de gestionar la información?, tener en cuenta cuales son los métodos que se puede usar para que la aplicativo web sea seguro, también se debe tener muy claro los principios del diseño del

aplicativo web, además de cómo se debe implementar identificando el tipo de aplicación web.

Por otro lado, se debe tener en cuenta la necesidad del cliente, creando propuestas e implementando los fundamentos básicos del diseño de una interfaz gráfica de usuario considerando las herramientas de diseño y las tendencias de la web en el pasar del tiempo.

REFERENCIAS BIBLIOGRÁFICAS

- Luján, M.S, (2002). Programación de aplicación web: Historia, Principios básicos y cliente web (cap2-pag25).
- Sommerville, I. (2005). Ingeniería del software, 7ma. ed.
- Marrero E. (2004-06). Interfaz gráfica de usuario: Aproximación semiótica y cognitiva. Recuperado de www.chr5.com/investigacion/investiga_igu/igu_aproximacion_semio-cognitiva_by_chr5.pdf.
- Margaix, D.A. (2007). Conceptos de la web 2.0 y biblioteca 2.0: origen, definiciones y retos para las bibliotecas actuales. pag96. Recuperado de eprints.rclis.org/9521/1/kx5j65q110j51203.pdf
- O'Reilly & Battelle (2009). What's next for Web 2.0? in Web Squared: Web 2.0 Five Years On. Recuperado de www.oreilly.com/pub/a/web2/archive/what-is-web-20.html
- Romero, M. & Vásquez, M. (2016). Coordinación de Seguridad de la Información. Recuperado de www.seguridad.unam.mx/documento/?id=17
- Amazon. (1996-2016). Recuperado de www.amazon.es/b/ref=amb_link_177305467_1?ie=UTF8&node=2901930031&pf_rd_m=A1AT7YVPFBWXBL&pf_rd_s=merchandised-search-left-2&pf_rd_r=RAWHEP54SRDBN25TQ51F&pf_rd_t=101&pf_rd_p=628278967&pf_rd_i=923373031
- Alegsa, L (2016). Diccionario de informática y tecnología. Recuperado de www.alegsa.com.ar/Dic/aplicacion%20web.php.
- Diseño de materiales multimedia. Web 2.0, Usabilidad de páginas web. Recuperado de www.ite.educacion.es/formacion/materiales/107/cd/html/pdf/html17.pdf