

Posgrado en Edición Digital (on line y off line) Desarrollo, integración y control de calidad

Conceptos previos

Una **plataforma on-line** es un sistema que permite la distribución de información entre un productor (emisor de información) y un consumidor (receptor de información) utilizando las Tecnologías de la Información y la Comunicación (TIC).

El rasgo característico de las plataformas *on-line* es que se establece una conexión **simultánea** entre los elementos técnicos participantes en la transmisión de información. La información puede estar almacenada en un sistema informático o generarse en el instante de acceso a partir de la información residente en otro sistema combinada con otros datos enviados por el consumidor.

Las **plataformas off-line**, por ejemplo un DVD o un CD-ROM, distribuyen información de manera que no existe esta conexión simultánea entre productor y consumidor.

Las TIC tienen un considerable nivel de complejidad y están plagadas de siglas que pueden confundir al profano en la materia. El objetivo de este apartado es familiarizarnos con esta tecnología de manera que nos permita profundizar más adelante en cualquier aspecto por medio de bibliografía o recursos web. Nos daremos por satisfechos si tras estudiar este apartado obtenéis un mapa mental de lo que sucede cuando se establece la comunicación *on-line*.

Adicionalmente, se expondrán algunos conceptos básicos sobre el funcionamiento de las plataformas *off-line*. De este modo, como editores multimedia, dispondréis de una serie de criterios a la hora de escoger la mejor manera de distribuir los materiales multimedia.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Conceptos previos – Funcionamiento de las plataformas *off-line*

Cualquier plataforma *on-line* se basa en una serie de elementos (físicos y lógicos) que definen las características de los servicios que ofrecerá al cliente, que son el objetivo final de cualquier plataforma. A partir de este punto vamos a estudiar los elementos que definen una plataforma *on-line*:

- Redes de comunicación
 - Direccionamiento e interconexión de redes
 - Protocolos de comunicación
 - Servicios *on-line*
-

▪ Redes de comunicación

Cualquier plataforma *on-line* se basa en alguna de las múltiples maneras de interconectar ordenadores. Describiremos brevemente las principales características de los tipos de conexiones más habituales.

Comunicación del usuario final

Desde el punto de vista del usuario final de un servicio de telecomunicaciones, actualmente se dispone de diversos tipos de conexión que ofrecen una velocidad de transferencia de información diferente (ancho de banda):

RTB (Red Telefónica Básica): usa una línea telefónica analógica y requiere un aparato denominado **módem** para poder traducir en sonidos los bits que se quieren transmitir o recibir. El ancho de banda está limitado, por motivos de diseño del sistema telefónico analógico, a velocidades máximas de 54.000 bits por segundo. Mientras se usa el módem no es posible usar la línea para hablar.

RDSI (Red Digital de Servicios Integrados): es una línea digital que tiene, en el denominado modo de acceso básico, dos canales de 64.000 bits por segundo (64 Kbps) y uno de 16.000 bits por segundo (16 Kbps). Se puede establecer una comunicación por ordenador usando uno de los canales de 64 Kbps y dejar el otro libre para llamadas telefónicas de voz.

ADSL (*Asymmetric Digital Subscriber Line*): permite usar una línea telefónica analógica mediante un módem diferente del de RTB, y ofrece anchos de banda de 256 Kbps, 512 Kbps y 2 Mbps en la descarga de información y la mitad en el sentido de subida. Está desbancando a la RDSI y a los módems convencionales debido a su mayor ancho de banda y facilidad de instalación. Asimismo, la línea siempre está abierta y es posible simultanear la comunicación de datos con llamadas de voz.

Cable: es una alternativa ofrecida por compañías de distribución de televisión por cable (a diferencia de las anteriores, que provenían de compañías telefónicas). Dichas compañías disponen de un tendido de red coaxial (similar a la de la distribución de la señal de televisión) o fibra óptica que llega a las casas de los abonados. Al digitalizar la red de distribución de televisión por cable es posible añadir servicios como llamadas telefónicas o comunicación por red. Actualmente, los anchos de banda son comparables a los de la ADSL y permiten simultanear servicios de voz, red y televisión.

El tipo de comunicación que utilicemos para conectarnos a un servicio determinado nos condicionará la calidad de acceso al mismo.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Actualmente se define un canal de banda ancha como aquel medio de comunicación capaz de transmitir voz, datos, imágenes y vídeo de alta calidad de forma universal.

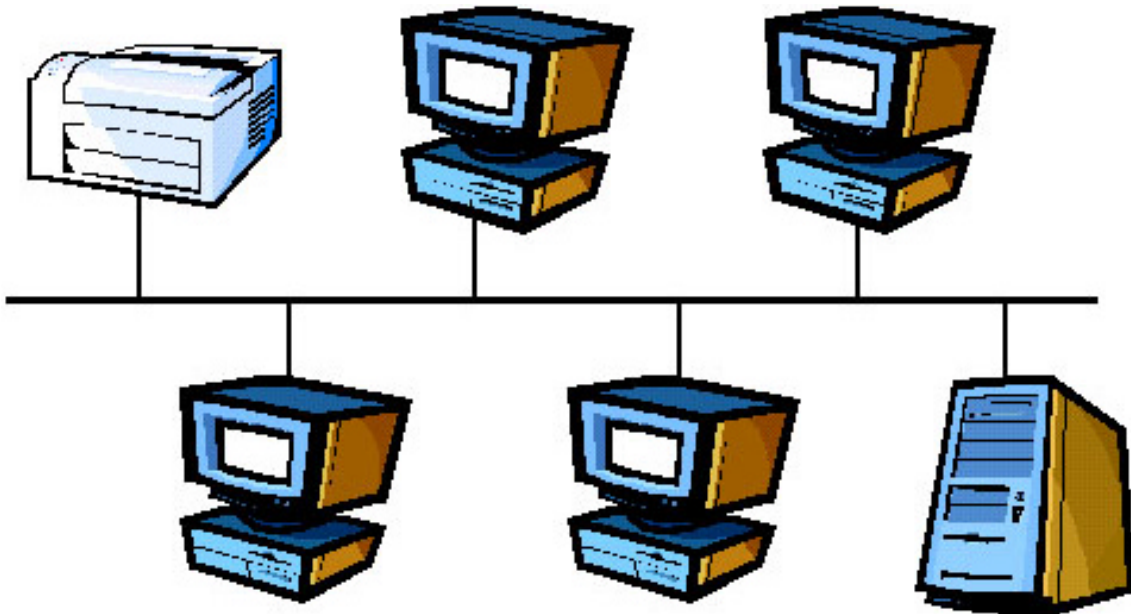
Comunicaciones dentro de las empresas

Las redes de área local (en inglés LAN, siglas de *Local Area Network*) son redes de ordenadores y otros dispositivos interconectados entre sí, cada uno identificado con una dirección única. Este tipo de redes es habitual en oficinas, colegios, empresas pequeñas e incluso en domicilios particulares. Suelen ser de los tipos siguientes:

Ethernet: mediante una placa de red ethernet es posible establecer una comunicación de datos de un ancho de banda máximo de 10 Mbps; suele ser la más habitual entre este tipo de redes.

FastEthernet o FDDI (*Fiber Distributed Data Interface*) con velocidades de 100 Mbps.

Inalámbricas: basadas en tarjetas Wireless y la tecnología Bluetooth; este tipo de redes permiten la comunicación entre equipos (habitualmente ordenadores portátiles, teléfonos móviles, cámaras...) eliminando los cables, ya que la información se transmite vía radiofrecuencia.



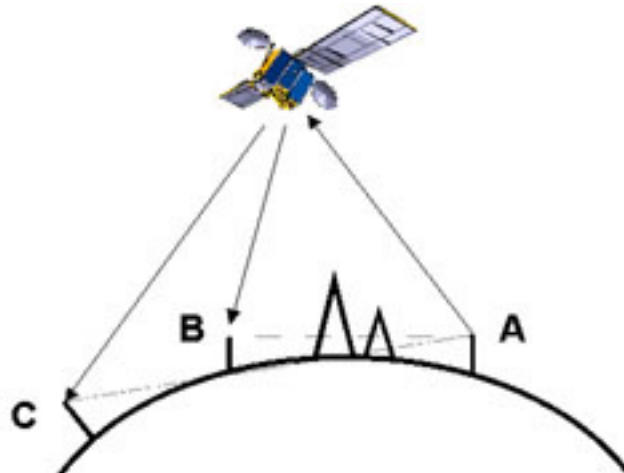
Comunicaciones vía satélite

La comunicación vía satélite permite que muchos usuarios puedan recibir una señal electromagnética en ubicaciones muy alejadas entre sí.

Puesto que la Tierra es redonda y hay mares y montañas, a veces resulta muy caro o imposible establecer un enlace para intercambio de señales electromagnéticas entre dos puntos de la superficie terrestre: realizar un tendido de cable a través del mar puede resultar muy costoso y, evidentemente, no llega a cualquier punto del mar en el que un barco se encuentre. Las montañas y la curvatura de la Tierra pueden actuar como obstáculos en la emisión de señales de radio o de microondas. En consecuencia, se utilizan los denominados satélites de comunicaciones.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad



Los satélites aprovechan una ley del físico y matemático Kepler. Siguiendo esta ley, si se lanza un satélite al espacio describiendo una órbita alrededor de la tierra a unos 36.000 km por encima del ecuador, su periodo de rotación será igual al de la Tierra. Es decir, un observador en la superficie de la Tierra ve el satélite ubicado siempre en el mismo punto del cielo. Es lo que se denomina «posición geoestacionaria».

De este modo, es posible utilizar un satélite para rebotar una señal electromagnética entre dos puntos alejados entre sí. Cada estación terrestre debe tener una antena enfocada al satélite, sin obstáculos entre su antena y la del satélite (edificios altos, montañas, etc.).

Para prevenir el caos en el cielo por la puesta en órbita de demasiados satélites de comunicaciones, se han establecido diversos acuerdos internacionales mediante los que se asignan órbitas y bandas en el espectro de intercambio de señales electromagnéticas.

Un satélite habitualmente utiliza un ancho de banda de 500 MHz que se divide en subcanales de 36 MHz de ancho de banda. Esta manera de funcionar es similar a la que se usa en las emisiones radiofónicas terrestres, en las que cada emisora de radio utiliza un subcanal dentro de la Frecuencia Modulada o la Onda Media, las cuales tienen un espectro de frecuencias asignado. En cada uno de estos subcanales del satélite, por ejemplo, pueden intercambiarse datos a una velocidad de 50 Mbps, o bien establecerse 800 conversaciones telefónicas simultáneas, digitalizadas a 64 Kbps.

Los primeros satélites ofrecían señales analógicas que se usaban para diversos propósitos tales como la emisión de programas televisivos, comunicación telefónica en alta mar o en el aire, etc. Actualmente la comunicación vía satélite está pasando por un proceso de digitalización que permite, mediante técnicas de compresión, usar menor ancho de banda para intercambiar la misma cantidad de información.



Actividad

Discutid hasta qué punto la comunicación vía satélite es una alternativa para proporcionar servicio de Internet a lugares aislados como casas rurales. Temas a tener en cuenta: precio de líneas mediante cable o enlace terrestre, saturación del espacio, coste de las antenas de comunicación vía satélite.

Comunicaciones móviles: GSM, WAP, UMTS, GPRS

Las comunicaciones mediante teléfonos móviles se basan en la división de la superficie terrestre en las denominadas «celdas» de forma hexagonal (en inglés, los teléfonos móviles se llaman teléfonos celulares por este motivo).

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Para dar cobertura a los teléfonos móviles ubicados en dichas celdas se distribuyen una serie de antenas terrestres que se intercomunican entre sí. A medida que un teléfono se va moviendo pasa de una celda a otra y deja de comunicar con una antena para pasar a la siguiente. La continuidad de la comunicación se consigue mediante una serie de ordenadores que controlan todas las antenas que ofrecen la cobertura a los teléfonos.

Originalmente este tipo de comunicación móvil estaba orientada únicamente a las conversaciones telefónicas, en un principio analógicas y actualmente digitales: los denominados teléfonos GSM. El sistema GSM permite disponer de servicios avanzados como desvío de llamadas, llamada en espera, mensajes y *roaming* (cambio de red entre distintos países y operadores) y transmisión de datos (aunque a una velocidad muy baja: 9.600 bps). Otro problema planteado por este sistema es que, con el aumento de usuarios de estos teléfonos, las bandas disponibles se han quedado cortas.

La tecnología **WAP** (*Wireless Application Protocol*) permite navegar por Internet mediante un teléfono GSM. Esta tecnología nació como un acuerdo entre los fabricantes de teléfonos móviles Motorola, Ericsson y Nokia, con Microsoft y Phone.com. Esta tecnología incorpora un mini navegador en el teléfono móvil cuyo lenguaje de intercambio de contenidos no es HTML sino WML (*Wireless Markup Language*).

Sin embargo, esta idea no ha tenido éxito debido a diferentes aspectos como la imposibilidad de acceder a ciertos contenidos existentes en Internet con WML, la baja velocidad a la que se envían los datos (9.600 bps) y el reducido tamaño de las pantallas de baja resolución y en blanco y negro.

Otras tecnologías:

GPRS (*General Packet Radio System*) permite una velocidad teórica de 172.100 bps y ofrece ventajas económicas frente al WAP, ya que no es preciso pagar por el tiempo que se navega sino por el volumen de información visualizada.

UMTS (*Universal Mobile Telecommunication System*) es una evolución del actual sistema de telefonía digital europeo (GSM) y es conocida como la tercera generación de móviles. Estas son algunas de las ventajas de UMTS:

- Mantendrá la compatibilidad con las redes GSM.
- La frecuencia para UMTS es de 2 GHz y será posible transmitir datos a 2 Mbps. Con estas velocidades, la videoconferencia móvil es una realidad.
- UMTS dispondrá de conexión permanente a la red (no sólo al efectuar una comunicación) y se facturará por volumen de datos en lugar de por tiempo.
- Velocidad adaptable: el ancho de banda de cada llamada se asigna de forma dinámica (no es lo mismo una llamada de voz que una transmisión de imágenes), con lo que se optimiza su uso.
- Global: el sistema está diseñado para funcionar en todo el mundo, empleando tanto redes terrestres como enlaces por satélite; además, es compatible con GSM.
- Sencillo de utilizar: como es un sistema único, el *roaming* (cambio de red) es prácticamente instantáneo, sin cortes en la comunicación.

Aunque las principales operadoras de telefonía móvil compraron las licencias a los gobiernos para usar este sistema hace años, la difusión del mismo se ha retrasado por motivos técnicos y todavía no está implantado. Las últimas noticias dicen que para finales del año 2003 o principios de 2004 las redes UMTS comenzarán a ser una realidad.

Posgrado en Edición Digital (on line y off line) Desarrollo, integración y control de calidad

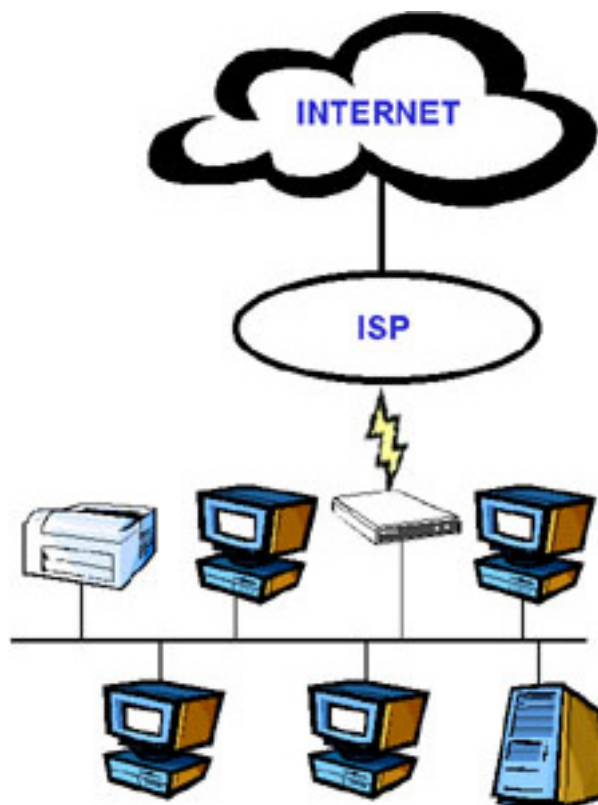
▪ Direccionamiento e interconexión de redes

Un conjunto de ordenadores interconectados forma lo que se denomina una **red de comunicación de ordenadores**. Una red local (LAN) es de este tipo, pero habitualmente cuando hablamos de redes de comunicación nos referimos a una red de redes y más concretamente a Internet.

Internet es la unión de miles de redes informáticas conectadas entre sí mediante una serie de protocolos (TCP/IP) que hacen posible, para cualquier usuario de una de estas redes, comunicarse o utilizar los servicios de cualquiera de las otras.

El origen de Internet se remonta a 1969, cuando se construyó una red experimental llamada ARPAnet de la Agencia de Proyectos de Investigación Avanzada de Defensa del Departamento de Defensa de Estados Unidos. Dicha red nació para interconectarse con otras mediante diferentes tipos de enlace (satélite, radio y cableado) y con la propiedad de ser resistente a fallos parciales, esto es, si una parte de la red fallara, el resto continuaría operativa.

Normalmente la conexión de nuestra red local a Internet la conseguimos a través de los servicios que nos ofrecen los **ISP** (*Internet Service Provider*) o proveedores de acceso a Internet. La conexión a dicho proveedor la realizamos a través de diversos tipos de conexión como los vistos en el apartado Comunicación del usuario final.



En Internet, cada una de las máquinas debe tener un identificador único: las **direcciones IP**, que son grupos de cuatro cifras con valores entre 0 y 255 (por ejemplo, 195.189.12.45).

Posgrado en Edición Digital (on line y off line)

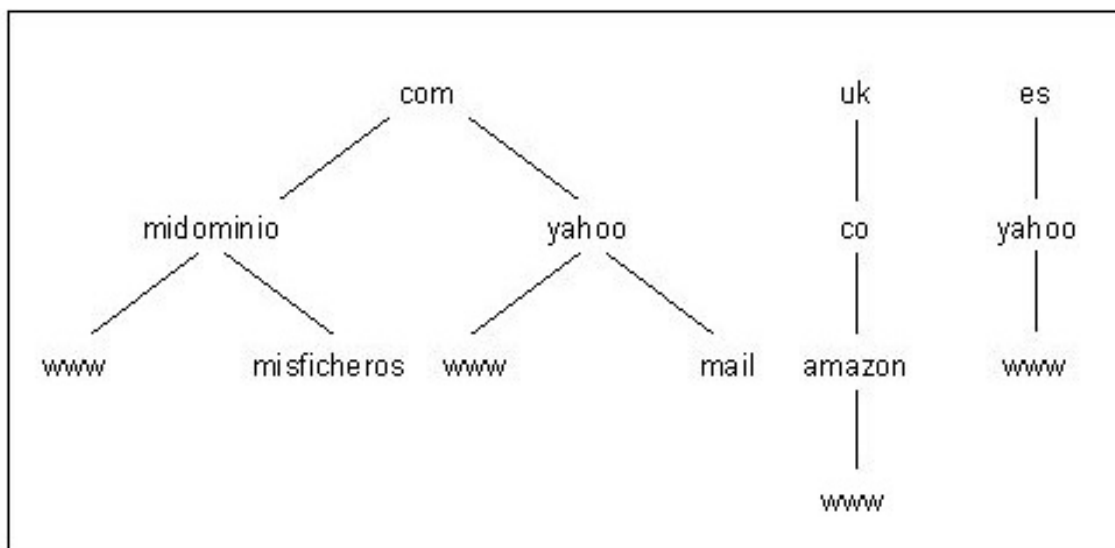
Desarrollo, integración y control de calidad

DNS

Recordar una dirección IP resulta poco práctico, por lo que estas direcciones no suelen ser las utilizadas para localizar un ordenador en Internet, sino que se les asocia un nombre mediante el sistema llamado DNS.

El sistema **DNS** (*Domain Name Service*) permite realizar la traducción entre los nombres que suelen divulgarse para usarlos por las personas (palabras o secuencias alfanuméricas) y las direcciones IP reales (números) usadas por los ordenadores para comunicarse.

En DNS se establece una estructura jerárquica entre los diferentes nombres de ordenadores que permite ubicar a una máquina dentro de una familia (dominio en DNS).



El nombre `www.yahoo.es` quiere decir que dentro del dominio de nivel superior «es», que corresponde a España, existe un subdominio llamado «yahoo» el cual tiene un ordenador que se llama «www».

Cada país tiene su dominio asignado aunque algunos de los dominios más conocidos son los de nivel superior de los Estados Unidos:

- `com`: entidades comerciales.
- `net`: entidades que no están asociadas a un país concreto.
- `org`: organizaciones no comerciales.
- `gov`: gobierno (en inglés *government*) de Estados Unidos.

Recientemente se han añadido otros como:

- `tv`: de uso restringido para compañías de televisión, aunque inicialmente era el dominio asignado a una pequeña isla del Pacífico: Tuvalu.
- `info`: para sitios web que proporcionan información.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

- biz: orientado a negocios (en inglés *business*; *biz* sería un derivado de argot).

Acceso a recursos de internet: URL

Los diferentes elementos accesibles en Internet se localizan mediante la denominada **URL** (*Uniform Resource Locator* o *Universal Resource Locator*). Una URL se divide en cuatro partes que sintácticamente se expresan:

Protocolo://servidor[:puerto]/camino

- **Protocolo:** por ejemplo, http para acceso a páginas web.
- **Servidor:** nombre DNS o dirección IP del ordenador que proveerá el servicio requerido.
- **Puerto:** identificador del canal dentro del servidor por el que se ofrece el servicio. Cada protocolo tiene asignado un puerto por defecto por lo que si no se especifica se utilizará implícitamente aquél.
- **Camino:** cada servidor interpreta esta parte de la URL de acuerdo con su configuración de servicio.



Ejemplos de URL:

http://www.uoc.edu/ejemplo/pagina/web.html: el cliente contacta con el servidor que responde a la dirección *www.uoc.edu* utilizando el protocolo *http* y le pide que le muestre la página web contenida en el archivo «*/ejemplo/pagina/web.html*». Como no ha especificado ningún puerto se utilizará el 80, que es el puerto por defecto para el protocolo *http*.

ftp://ftp.uoc.edu:1050/ejemplo/fichero/documento.doc: el cliente contacta con el servidor que responde a la dirección *ftp.uoc.edu* para que le sirva el fichero cuyo camino es «*/ejemplo/fichero/documento.doc*» a través del puerto 1050.

▪ Protocolo de comunicación

Cuando dos personas se encuentran en la calle el protocolo de comunicación que se establece entre ellas es algo similar a este diálogo:

A: Hola
 B: Buenas tardes
 A: Quería hablar contigo de...
 ...
 A: Te tengo que dejar porque me esperan.
 B: De acuerdo, nos vemos.
 A: Adiós

Nadie que quiera tener buena relación con el resto de la gente interrumpirá los pensamientos de otra persona de sopetón ni se irá dejando a la otra persona con la palabra en la boca. Si no se establece este protocolo de conexión y desconexión, la gente puede no entender de qué le están hablando o quedarse con cosas por decir. El protocolo de conexión hace que las

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

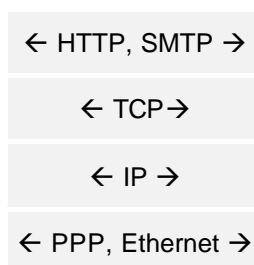
dos personas dejen de pensar en lo que pensaban y se ubiquen en una conversación provechosa. Lo mismo pasa con el protocolo de desconexión: se indica a ambos participantes que el tema se da por concluido, al menos temporalmente.

A parte de la conexión y desconexión, el intercambio de información se rige por un conjunto de reglas que se consideran de buena educación.

Estos protocolos de comunicación humana se trasladan al mundo de los ordenadores por el mismo tipo de motivos: establecer conexión, intercambiar información o servicios y desconectar.

El protocolo de comunicación es el conjunto de reglas que ordenan y facilitan la comunicación entre los diferentes ordenadores que se encuentran en la red.

Los protocolos en Internet están estructurados en capas. En el nivel más bajo se encuentra el protocolo que maneja y agrupa los bits que se envían por la red física para conseguir la transmisión correcta entre dos extremos. A partir de este nivel se definen otros protocolos que utilizan los que se encuentran en la capa inferior hasta llegar al protocolo de nivel más alto, que es el utilizado por las aplicaciones para comunicarse. Algunos de los protocolos en Internet más conocidos son:



El protocolo PPP (*Point to Point Protocol*) establece la conexión «física» entre dos módems, mientras que Ethernet es el protocolo utilizado por los dispositivos conectados a redes <link nodo 7.1.1.1: "Redes de comunicación">Ethernet.

Un protocolo situado en un nivel superior es el IP (*Internet Protocol*) usado en Internet. En caso de establecerse la conexión física a Internet a través de un módem, IP utilizará el protocolo PPP por debajo, mientras que utilizará el protocolo Ethernet en caso de conectarse a través de una red Ethernet (el protocolo IP no cambiará, independientemente del medio físico utilizado por debajo).

Más arriba nos encontramos uno de los protocolos base de cualquier comunicación en Internet: TCP (*Transmission Control Protocol*). Habitualmente se habla del protocolo base de Internet como TCP/IP puesto que la combinación de protocolos TCP e IP es la más utilizada en la red.

Y en el nivel más alto nos podemos encontrar, por ejemplo, el protocolo HTTP (*Hypertext Transport Protocol*) utilizado para la distribución de contenidos en la WWW (*World Wide Web*) o el SMTP (*Simple Mail Transfer Protocol*) para el envío de correos.



Diálogo que se establece en el protocolo SMTP (correo electrónico).

El emisor establece una conexión con el receptor y comienza el diálogo.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

```
R:    Connected to servidor.tudominio.com.
R:    Escape character is '^]'.
R:    220 servidor.tudominio.com ESMTP Exim 3.33 #1 Tue, 11 Mar 2003 18:33:51
+0100
E:    mail from: yo@midominio.es
R:    250 <yo@midominio.es> is syntactically correct
E:    rcpt to: tu@fbg.ub.es
R:    250 <tu@tudominio.com> verified
E:    data
R:    354 Enter message, ending with "." on a line by itself
E:    bla bla bla
E:    .
R:    250 OK id=18sne1-0004SR-00
E:    quit
R:    221 servidor.tudominio.com closing connection
```

El emisor usa los comandos que indican de quién es el *e-mail* (*mail from*) a quién va dirigido el *e-mail* (*rcpt to*) y el cuerpo del mensaje acabado por un «.». El emisor finaliza la conexión mediante el comando «quit».

El intercambio de información mostrado en el ejemplo es el que se da en la aplicación (emisor y receptor del correo electrónico) pero también se están utilizando los protocolos TCP/IP entre el origen y el destino para garantizar el intercambio ordenado de información. También se utiliza el protocolo Ethernet (suponemos que nos encontramos en una red Ethernet) para garantizar el paso de información entre cada par de puntos de red por los que viaja la información.

■ Servicios on-line

Podríamos definir un servicio *on-line* como la agrupación de un ordenador y un programa que implementa un protocolo de comunicaciones con otro ordenador de la red al recibir la petición de éste.

Algunos de los servicios Internet más conocidos son el de acceso a páginas web, el correo electrónico o la transferencia de ficheros, que veremos más adelante. Un mismo ordenador puede actuar como proveedor de diferentes servicios.



<http://ordenador.dominio.com> → URL que identifica el servicio de páginas web que ofrece el servidor ordenador.dominio.com utilizando el protocolo http.
<ftp://ordenador.dominio.com> → el mismo ordenador sirve ficheros a través del protocolo FTP.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Arquitectura cliente-servidor

La arquitectura cliente-servidor es la base teórica sobre la que se desarrollan la mayor parte de los servicios *on-line*.

Bajo el punto de vista de esta arquitectura existen dos entidades: el cliente y el servidor. La entidad que actúa como servidor ofrece una serie de servicios que los clientes conocen y solicitan.



Podríamos establecer un símil de esta arquitectura con el servicio que proporciona el teléfono de emergencias 112.

Toda persona hace su vida normal hasta que tiene una emergencia, en tal caso se pone en contacto con el 112. Este servicio tiene un conjunto de personas que responden, evalúan la naturaleza de la petición y gestionan lo necesario para hacer llegar una grúa, los bomberos, etc.

Mientras se gestiona la petición de servicio el cliente permanece a la espera. Un comportamiento similar sigue la mayoría de servicios que se ofrecen en la red.

En esta arquitectura se cataloga a los agentes según uno de los dos comportamientos siguientes:

Servidor: (en el ejemplo operadores del 112)

- permanece a la escucha de una petición;
- cuando recibe una petición la analiza;
- ejecuta el servicio asociado a ella;
- genera una respuesta;
- vuelve al estado inicial de escucha.

Cliente: (en el ejemplo persona en situación de emergencia)

- desarrolla un trabajo independiente hasta que requiere de un servicio externo;
- elabora una petición y se pone en contacto con un proveedor del servicio requerido;
- se bloquea en espera de una respuesta;
- al recibir la respuesta al servicio utiliza los datos enviados por el servidor y continúa con su trabajo.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad



Un ejemplo de esta arquitectura en el ámbito tecnológico sería la lectura del correo electrónico:

El programa lector de correo se inicializa y pide al usuario una contraseña para validar su identidad contra el servidor de correo electrónico. Al obtener dicha contraseña emite una petición de inicio de sesión en el servidor de correo.

El servidor le responde afirmativamente con el número de mensajes por leer o bien le señala que las credenciales no son inválidas.

Si el lector de correo electrónico consigue establecer la sesión, irá haciendo peticiones de lectura/borrado de los mensajes del buzón en el servidor hasta que se haga la petición de cierre de sesión.

Cabe destacar que ambos programas, cliente y servidor, pueden residir en el mismo ordenador o no.

Asimismo, un programa que actúa como servidor de un determinado tipo de servicio (por ejemplo, servidor páginas web) puede a su vez actuar como cliente de otros servicios (por ejemplo, cliente de acceso a bases de datos).

Caracterización y breve descripción de los servicios:

Distribución de páginas web: basado en los protocolos **HTTP** (*HyperText Markup Protocol*) o **HTTPS** (*HyperText Markup Protocol Secure*) son servicios que permiten difundir páginas web. La diferencia entre ambos es que en el segundo caso la comunicación entre cliente y servidor está encriptada de manera que un tercero es incapaz de «escuchar» su diálogo.

Correo electrónico: el envío y recepción de mensajes electrónicos es uno de los servicios más utilizados en Internet. Se basa en los protocolos **SMTP** (*Simple Mail Transfer Protocol*) para el envío y **POP3** (*Post Office Protocol version 3*) o **IMAP** (*Internet Message Access Protocol*) para la recepción de correos.

Transferencia de archivos: permite la transferencia de ficheros desde o hacia un servidor a través del protocolo **FTP** (*File Transfer Protocol*). Este servicio puede ser anónimo (cualquiera puede acceder al servidor) o a través de credenciales de acceso (usuario y contraseña).

Emulación de terminal: permite establecer una sesión de emulación de terminal en un servidor. Estas sesiones tienen diversas utilidades, aunque la mayor parte de las veces se usan para administrar las funcionalidades del servidor. Se basa en el protocolo **TELNET** (*Telecommunications Network*).

Navegadores

Un navegador es un programa que actúa como cliente para el acceso a las diferentes URL disponibles en Internet que visualiza los documentos escritos en lenguaje HTML, los cuales pueden incluir texto, gráficos, sonidos, objetos multimedia o enlaces (*links*) a otros documentos o servidores Web.

Marc Andreessen fue el desarrollador del primer navegador para superficies gráficas: Mosaic. Los programadores de Mosaic inventaron nuevas características que implementaron en el navegador sin que estuvieran estandarizadas. Entonces fue cuando surgió la avalancha que

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

hasta este momento sigue creciendo. Marc Andreessen, por su parte, abandonó el proyecto Mosaic y fundó una nueva compañía de software: Netscape.

Inicialmente los navegadores tan sólo se utilizaban para visualizar páginas web. Rápidamente se han ido dotando de mayores capacidades y pueden llegar a actuar como clientes de correo electrónico, clientes de transferencia de ficheros, clientes del servicio de noticias (o *news*), clientes de mensajería instantánea, editor de páginas web, gestor de direcciones, agenda, etc.

Centrémonos en la utilidad básica para la que surgieron los navegadores: la visualización de páginas web.

El editor multimedia debe estar familiarizado con las siguientes características, puesto que la web es uno de los soportes más usados para la difusión de contenidos multimedia:

- Los navegadores funcionan sobre ordenadores personales convencionales y set-top-boxes (WebTV).
- Los navegadores deben ser capaces de interpretar las instrucciones JavaScript, Java y Active-X que les envían los servidores ya que si no lo hacen, los contenidos multimedia no se visualizarán correctamente. En el apartado 7.2 se describirá la utilidad de las tecnologías que se acaban de citar; aquí nos limitaremos a destacar que no todos los navegadores interpretan las mismas instrucciones de igual forma, por lo que no debemos olvidar que el aspecto de los diseños multimedia no será idéntico en todos los navegadores. Incluso la misma versión del mismo navegador sobre diferentes plataformas (Mac, PC, Unix, etc.) puede presentar comportamientos diferentes en cuanto a la interpretación del contenido.
- Los navegadores pueden no ser autosuficientes y requerir *plug-in* (su traducción literal es «enchufes») que son pequeños programas que se incorporan a nuestro navegador para visualizar determinados contenidos multimedia. Existen *plug-in* para visualizar archivos de contenidos Flash, vídeo, PDF, etc.
- A pesar de que el navegador más utilizado inicialmente en Internet fue Netscape, a partir del 1995 y con la incorporación en la distribución básica del sistema operativo Windows 95 del navegador de Microsoft Internet Explorer, éste se ha convertido en el navegador más popular.

Otros navegadores disponibles en la actualidad son:

- Mozilla, basado en una versión de Netscape y que se distribuye en código abierto.
- Opera, cuyas características principales son su velocidad y los bajos recursos de ordenador que se requieren para ejecutarlo gracias a la optimización realizada en el código del mismo.

Los cuatro *browsers* (navegadores) comentados están disponibles para plataformas Windows, Mac y Unix, excepto Internet Explorer, que no se distribuye para Unix.

Servidores de páginas web

Usando la arquitectura cliente/servidor aparecen servidores especializados en distribuir páginas web estáticas y dinámicas.

Las páginas estáticas están compuestas por un conjunto de ficheros en formato HTML almacenados en una zona de disco accesible por el servidor de páginas web. Dichos ficheros a su vez se vinculan con otros ficheros HTML o con elementos multimedia como imágenes fijas, películas, sonidos, etc.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Las páginas dinámicas generan su contenido en el momento en el que el servidor recibe la petición.

Las técnicas para generar páginas dinámicas son variadas: CGI (*Common Gateway Interface*), Servlets, etc. Todos estos métodos establecen un mecanismo mediante el cual los servidores web invocan a programas que proveen los contenidos dinámicos. Estos programas habitualmente realizan tareas como:

- Tratamiento de los datos enviados mediante formularios.
- Consulta y/o escritura en bases de datos.
- Creación de pasarelas hacia otros servicios (por ejemplo, envío de mensajes vía correo electrónico).

Los programas mencionados pueden estar escritos en cualquier lenguaje de programación. Independientemente del tipo de contenido ofrecido por los servidores web (estático o dinámico), éstos tienen una serie de características que pueden ser de especial importancia para un editor multimedia:

- Mediante la gestión del control de acceso a estos servidores se crean las denominadas **intranets**. Éstas no son más que sitios web de acceso restringido que pueden ofrecer diferente contenido según las credenciales usadas en la validación del control de acceso (típicamente un nombre de usuario y una contraseña).
- Usando la gestión de cookies es posible el mantenimiento de sesiones que permiten funcionalidades como las populares «listas de compra».
- Usando la persistencia de cookies es posible almacenar datos en el ordenador donde reside el navegador. Estos datos pueden ser útiles para mantener información introducida en una sesión y enviarlas al servidor en posteriores visitas, evitando tener que realizar de nuevo las consultas realizadas por el servidor.
- Si la cantidad de visitas que se recibe es superior a las que puede soportar un solo servidor es posible añadir varios ordenadores al servicio de páginas web. Existen **sistemas de balanceo de carga** entre los servidores que ofrecen al usuario final la ilusión de que el sitio web está atendido por un solo ordenador con una capacidad prácticamente ilimitada.
- Mediante el análisis de **<glosario “fichero de log”>**logs de los servidores se obtiene información muy importante para adecuar y mejorar los contenidos de nuestro sitio web a partir del estudio del comportamiento de los visitantes dentro del mismo. También se puede consultar el volumen de visitas, distribución horaria de las mismas, popularidad de las diferentes páginas, etc.
- También existen programas que realizan una clasificación de los documentos de los sitios web que permiten ofrecer buscadores de contenidos dentro del mismo.

Servidores de audio y video

El audio y el vídeo pueden almacenarse digitalizados en ficheros. En ambos casos, el tamaño de estos ficheros puede ser considerablemente grande. Existen dos técnicas para servir audio y vídeo en una plataforma *on-line*:

- Si el usuario descarga estos ficheros se habla técnicamente de que está haciendo un **download** de **audio o vídeo**. El principal inconveniente es que no es posible oír el audio o ver el vídeo hasta que la descarga del fichero se ha completado.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

- Para solventar este inconveniente aparecen los denominados servidores de **audio y vídeo streaming**. Mediante esta técnica y el correspondiente *plug-in* en el navegador del cliente se puede oír el audio o visualizar el vídeo mientras se descarga. El cliente y el servidor mantienen una sincronía que permite apreciar los contenidos multimedia de forma más o menos continua e independiente de la velocidad de la red. Es decir, si la conexión es lenta se visualizan menos fotogramas por segundo que si la conexión es rápida. En cualquier caso, el vídeo dura el mismo tiempo.

Los principales programas de reproducción de este tipo de ficheros son el Real One de la empresa Real Networks, el Windows Media de Microsoft y el QuickTime de Apple.

▪ Seguridad en los servicios en red : proxies y firewalls

Cuando un ordenador se conecta a una red de comunicaciones es susceptible de recibir ataques malintencionados por parte de usuarios. Dichos ataques pueden consistir en intentar obtener información confidencial de forma fraudulenta almacenada en el servidor, provocar un mal funcionamiento de cualquiera de los servicios que ofrece un servidor, saturarlo con peticiones de manera que no sea capaz de realizar su trabajo de manera correcta, etc.

El estudio de cómo ofrecer seguridad en los servicios de una red de ordenadores se puede dividir en dos partes: la red local y el exterior de la red local. Habitualmente los ataques provienen del exterior de la red local que se quiere proteger. Sin embargo, dependiendo del tamaño de ésta o de la confianza que nos ofrezcan los usuarios, queremos protegernos también de posibles ataques provenientes de su interior.

Con estas premisas en mente, separaremos nuestra red local (LAN) en dos subredes: la red de ordenadores que no precisan ser accedidos desde el exterior (LAN privada) y la red en la que se encontrarán los ordenadores que actúan como servidores hacia el exterior (esta última subred se denomina DMZ, *DesMilitarized Zone*).

Un cortafuegos o *firewall* es una aplicación (o un dispositivo *hardware*) que se utiliza para proteger una red privada.

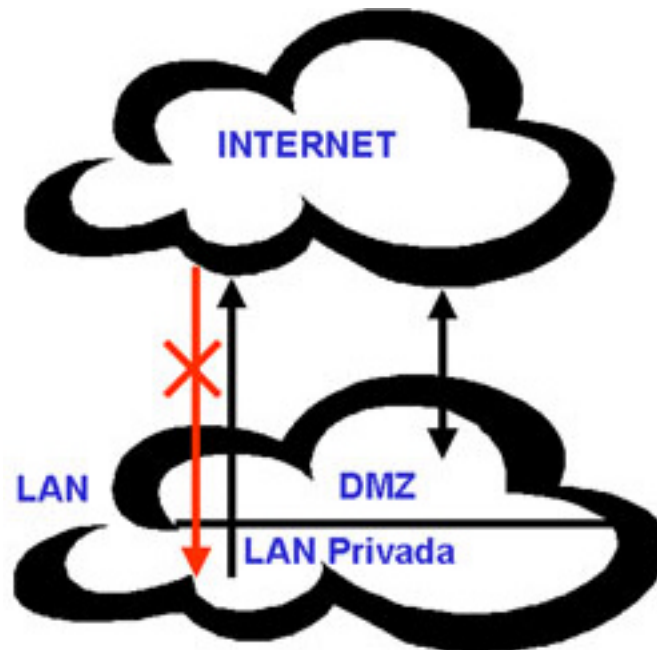
El cortafuegos es el encargado de denegar o permitir los intentos de comunicación desde Internet hacia nuestra red y el tráfico generado desde nuestra red hacia el exterior mediante unas reglas definidas por el administrador de seguridad. De esta manera podemos evitar el contacto de los ordenadores situados en nuestra red privada con cualquier ordenador de la red externa minimizando el riesgo de ataques a éstos.

Pero si no permitimos el acceso directo de nuestros ordenadores hacia Internet, ¿cómo conseguimos acceder a los servicios *on-line*?

Un *proxy* es una aplicación o un dispositivo *hardware* que hace de intermediario entre los ordenadores situados en la red privada e Internet.

Un *proxy* recibe peticiones de usuarios de acceso a Internet y se pone en contacto con el exterior. Cuando obtiene el resultado se lo devuelve al usuario que lo requirió. Como la función de este *proxy* es conocida por el administrador de seguridad, se habrá configurado la política de seguridad del cortafuegos para permitir el acceso hacia el exterior a dicho servidor. Normalmente, un *proxy* es a su vez un servidor de caché. La función de la caché es almacenar las páginas web a las que se accede más asiduamente en una memoria. Así, cuando un usuario quiere acceder a Internet lo hace a través del *proxy*, que mirará en la memoria caché si

tiene la página a la cual quiere acceder. Si es así, le devolverá la página de la caché y si no accederá a Internet, obtendrá la página y se la enviará al usuario. Con la caché se aceleran los accesos a Internet, sobre todo si los usuarios suelen acceder a las mismas páginas.



- Comercio electrónico

El comercio electrónico (en inglés *e-commerce*) es un paso más allá a la distribución de contenidos usando una red de comunicaciones. Mediante el comercio electrónico se pretende realizar la oferta de productos o servicios por una contraprestación económica.

Los modelos existentes en este comercio van desde empresas proveedoras de productos y/o servicios que los ofrecen a clientes finales, como venta de billetes o libros (BTC, siglas inglesas de *Business to Consumer*), intercambios de empresas con sus clientes y proveedores de manera electrónica para gestión de pedidos o facturación (BTB, *Business to Business*) o incluso lo que sería la traslación electrónica de los mercadillos o revistas de compra-venta de segunda mano (CTC, *Consumer to Consumer*) por poner algunos ejemplos.

No hay que pensar demasiado para darse cuenta de que este tipo de actividades es de un especial atractivo para los amantes de las actividades fraudulentas, por lo que las medidas de seguridad son un punto clave para ofrecer estos servicios.

Una de las medidas básicas que se deben tomar es la encriptación de la comunicación entre los participantes en la transacción. Para ello se utiliza el protocolo HTTPS mencionado anteriormente.

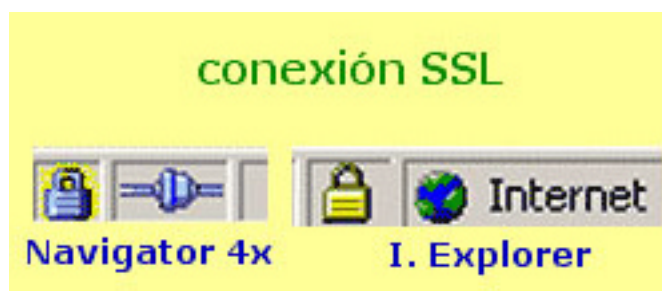


Actividad

Acceder a una página web segura (por ejemplo, Agencia Tributaria) y comprobar que en algunas de las secciones de alta seguridad (por ejemplo *Oficina virtual* de la Agencia Tributaria) el candado existente en la parte inferior de nuestro navegador se encuentra cerrado.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad



Un tema a resolver de manera elegante es la fiabilidad de los sistemas de pago. Existe la creencia de que enviar los datos de una tarjeta de crédito por Internet es una manera peligrosa de comprar. Sin embargo dejar esta misma tarjeta a un camarero de un restaurante para que se cobre puede suponer un peligro potencial similar.

Las entidades bancarias, principales interesadas en recibir una comisión por la transacción realizada, han sido las responsables de idear los diversos métodos existentes para garantizar la fiabilidad de los pagos en Internet.

El mecanismo más extendido a la hora de realizar pagos en Internet es la llamada *pasarela de pago*, que es la versión electrónica del TPV (Terminal de Punto de Venta) existente en todos los comercios que permiten el pago con tarjeta de crédito. La pasarela de pago permite la comunicación directa a través de Internet entre el comerciante y las redes bancarias, con lo que el papel del vendedor queda limitado a un mero intermediario entre el cliente y su banco. Puede ser una entidad independiente o el mismo banco del comerciante y cada entidad bancaria tiene su propia solución implementada.

Para evitar estos problemas e intentar estandarizar todas las soluciones existentes, en febrero de 1996 un grupo de empresas del sector financiero, informático y de seguridad (Visa International, MasterCard, Microsoft, Netscape, IBM, RSA, etc.) anunciaron el desarrollo de una nueva tecnología común destinada a proteger la compras a través de redes abiertas como Internet basadas en el uso de tarjetas de crédito.

Esta nueva tecnología se conoce con el nombre de *Secure Electronic Transaction* (Transacciones Electrónicas Seguras) o SET.

La realidad es que el uso de esta tecnología no ha alcanzado los niveles de utilización previstos en sus inicios y se están ideando otros métodos de pago: tarjetas inteligentes (con microchip incorporado), pagos asociados a facturas telefónicas de teléfonos fijos o móviles, etc.

Según los organismos y asociaciones de comercio electrónico, a pesar de la creencia popular, el índice de fraude en este método de intercambio comercial es relativamente bajo (Asociación Española de Comercio Electrónico).

■ Televisión interactiva

La televisión interactiva es un servicio mediante el cual el usuario recibe señales que contengan, entre otros, programas de televisión o información convertible a imágenes o vídeo, y que le permitan por algún canal de retorno enviar información con el fin de determinar los programas recibidos o realizar transacciones de cualquier tipo mediante el acceso a bancos o servidores de datos.

Se incluyen en esta definición servicios tales como eventos por pago, vídeo bajo demanda, telecompras, educación a distancia, información, transacciones bancarias o cualquier otro servicio de naturaleza similar que se desarrolle en el futuro.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Un decodificador (también llamado Set-top box) es el dispositivo que posibilita la recepción en el hogar de la televisión digital y todas sus ventajas: los servicios interactivos, el acceso condicional o la televisión de alta definición. Básicamente se encarga de recibir una señal digital en alguno de los estándares de televisión digital existentes, comprueba que tenga permiso para mostrarla y envía la señal de forma analógica al televisor.

Para poder ejecutar los datos o programas que se descargan de la señal se necesitan una serie de elementos que se pueden describir con el siguiente esquema de capas:

- **Hardware:** componentes que forman el Set-top box (CPU, memoria, acceso condicional, decodificador de MPEG, etc.).
- **Sistema operativo:** requieren un sistema operativo para la ejecución de aplicaciones interactivas. En este caso se utilizan sistemas operativos en tiempo real ya que hay una serie de operaciones como la decodificación de MPEG que necesitan ser realizadas al instante. Algunos ejemplos son Linux o Windows CE.
- **Plataforma (middleware):** conjunto de módulos que permiten el desarrollo más eficiente de las aplicaciones. La plataforma facilita un API (*Application Programming Interface*) para cada lenguaje de programación que soporte, como OpenTV, MediaHighway, Microsoft TVPAK o Liberate, con API para C, PanTalk, Visual Basic o HTML/Javascript, respectivamente. Un Set-top box puede soportar diferentes lenguajes de programación mediante la instalación de varias API.
- **Aplicaciones:** funcionalidades interactivas que, una vez descargadas, se pueden ejecutar a través del *middleware* (por ejemplo, las EPG (*Electronic Program Guides*), anuncios interactivos, chats, etc.).

Los decodificadores serán un paso intermedio hasta que se generalice el uso de los televisores digitales (de hecho ya existen algunos televisores de estas características).

En la actualidad existen tres medios para la transmisión de televisión digital que estén siendo utilizados de manera comercial: satélite, cable y TDT (Televisión Digital Terrestre). También hay un cuarto medio que se está probando en diferentes países como Inglaterra: el ADSL.

Hoy en día existen dos grandes grupos de estándares para la transmisión de televisión digital. Uno es europeo y se llama DVB (*Digital Video Broadcasting*), y el otro es estadounidense y se llama ATSC (*Advanced Television Systems Committee*).

En España todas las plataformas de televisión digital utilizan los estándares de transmisión de DVB: DVB-C para cable (Auna), DVB-S para satélite (Canal Satélite y Via Digital) y DVB-T para terrestre (la desaparecida Quiero TV y, desde el año 2002, obligadas por ley, todas las empresas españolas de televisión, tanto públicas como privadas).

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Conceptos previos – Funcionamiento y formatos de los soportes *off-line*

■ CD-ROM

Los CD (*Compact Disc*) de audio aparecieron en 1982. En 1984 se extendió la tecnología al CD-ROM (*Compact Disc Read Only Memory*). Desde entonces se ha convertido en una manera ampliamente utilizada de difundir música y datos.

Un CD-ROM se comportaría de manera similar a un disco duro de un ordenador. La principal diferencia es que se trata de un dispositivo de sólo lectura (*Read Only Memory*).

La tecnología de los CD-ROM no está basada en la orientación de partículas según campos magnéticos, como los discos de lectura-escritura (discos duros, disquetes, *zip*), sino en tecnología óptica. Un rayo láser recorre la superficie del CD-ROM de manera que interpreta y transmite al ordenador los bits allí grabados.

Cualquier programa o fichero susceptible de ser almacenado en un disco magnético lo es también de serlo en un disco óptico. Si es necesario modificar datos a lo largo de la ejecución de los programas, será necesario disponer de una zona en un disco regrabable, como puede ser el disco duro que almacena el sistema operativo del ordenador.

La capacidad estándar original de un CD-ROM es de 650 Mb como máximo, ya que originalmente se diseñó para almacenar 74 minutos de música digitalizada. Actualmente se ha extendido el uso de los CD-ROM de 700 Mb, que permiten guardar hasta 80 minutos de música. Dependiendo del tipo de contenidos multimedia o la cantidad de ficheros o programas, este tamaño máximo puede ser un inconveniente de peso.

Hay diversos tipos físicos de CD que se diferencian por la velocidad de acceso a los mismos y la cantidad de veces que se puede grabar sobre ellos. Al mirar la carátula de un CD grabable podemos observar los datos que nos dan esta información: por un lado aparecen tres cifras con un «x» al final nos indica la velocidad de grabación, regrabación y lectura del CD respectivamente. Esta cifra situada antes de «x» es el múltiplo de 150 Kps que puede utilizarse a la hora de acceder a dicho soporte. También aparece CD-R si se trata de un CD-ROM que sólo puede ser grabado una vez, o CD-RW si es regrabable.



Un CD-ROM que muestra en su carátula la etiqueta «CD-RW 4x-8x-32x» significa que soporta diversas grabaciones (RW) con velocidades de 600 Kps (4x150) a la hora de grabar, 1.200 Kps para regrabar (8x150) y 4.800 Kps (32x150) al leer sobre dicho soporte.

Han aparecido, y se usan actualmente, diferentes formatos para almacenar los datos en un CD-ROM:

- Windows CD-ROM: sigue el estándar ISO 9660 y se usa para ordenadores personales con sistema operativo Windows.
- Mac CD-ROM: estructura su contenido según el formato de Macintosh HFS y se usa para ordenadores personales con sistema operativo Macintosh.
- CD-ROM híbridos: contienen información en los dos formatos anteriores. Normalmente contienen la información por duplicado y programas ejecutables específicamente para cada una de las plataformas de sistema operativo.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

- CD-ROM XA (de *eXtended Architecture*): da soporte a diferentes tipos de usos de los CD-ROM, no tan orientados al almacenamiento de datos y programas para ordenadores.
- CD-i o CD-Interactivo: se desarrolló a principios de la década de 1980 por Phillips y Sony para proporcionar contenidos multimedia en los hogares. Actualmente ha quedado obsoleto porque el DVD ofrece una mejor alternativa.
- Vídeo CD: es capaz de almacenar 74 minutos de vídeo comprimido con la técnica MPEG-1. Permite acceder a listas de reproducción, menús y avance rápido hacia delante y atrás. Ha dado lugar a ciertas variantes como el SVCD (*Super Video CD*), el CVCD (*Compressed Video CD*) y el KVCD (*Kompressed Video CD*), formatos que aplican técnicas de compresión más elevadas y que, por tanto, permiten almacenar mayor cantidad de tiempo de vídeo en un solo CD.
- Photo CD: creado por Kodak a principio de la década de 1990. Almacena fotos en diferentes resoluciones, ya sea para visualizar como para imprimir.
- Enhanced CD: permite añadir datos en pistas de un CD de audio. En estas pistas pueden almacenarse desde vídeos y letras de las canciones hasta pequeñas presentaciones interactivas.

En la publicación de un CD-ROM se debe realizar el original (o master) y, una vez optimizada la distribución de los archivos, se utilizará para generar las imágenes (copias exactas) del mismo que serán las que se distribuirán.

▪ DVD-ROM y DVD-Video

Entre 1996 y 1998 apareció la tecnología DVD (*Digital Versatile Disc*, aunque originalmente se denominó *Digital Video Disc*) que se extendió desde Japón a Europa y Estados Unidos. Al contrario que los CD, pensados originalmente para la distribución de audio, el diseño original de los DVD tenía en mente la distribución de vídeo y, posteriormente, de todo tipo de contenidos multimedia.

Los DVD son de aspecto similar a los CD. Cada una de las dos caras del DVD es capaz de almacenar dos capas. La capacidad de una capa es de 4,4 Gb, por lo que les permite almacenar volúmenes de datos de hasta 17,1 Gb.

Las estadísticas muestran que el éxito de este medio de distribución de información *off-line* está superando a cualquier otro medio. Las aplicaciones del DVD son diversas:

- DVD-Video para películas en alta calidad junto con bandas sonoras multilingües y con efectos *surround*. La calidad de la imagen supera con creces a la del vídeo doméstico y el sonido puede compararse al de una sala de cine comercial.
- DVD-ROM para juegos por ordenador y almacenamiento de datos. La mayor capacidad de almacenamiento y la rápida bajada de precio de los reproductores está haciendo que los DVD-ROM superen a los CD-ROM.
- DVD-Audio para música en alta calidad. Se lanzó a principios del año 2000 y permite ofrecer música con efectos *surround*.

Los contenidos multimedia y las aplicaciones distribuidas mediante DVD gozan de diversas ventajas:

- Mucha mayor capacidad de almacenamiento.
- Se usa el estándar de distribución de vídeo MPEG-2, de mayor calidad que su antecesor.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

- Es posible usar Internet para desbloquear el contenido de un DVD y visualizarlo bajo el concepto de *pay-per-view*.

Los fabricantes de lectores y discos DVD-Video han dividido el mundo en seis zonas, de manera que un disco producido para una zona no pueda funcionar en otra zona. Además, hay que tener en cuenta temas como la frecuencia o el número de líneas de los televisores: 50 Hz y 625 líneas en Europa y 60 Hz y 565 líneas en Japón (ambas áreas de zona 2).

Al igual que en el caso de los CD-ROM, existen DVD de grabación única (DVD-R) y regrabables (DVD-RW).

Posgrado en Edición Digital (on line y off line) Desarrollo, integración y control de calidad

Proceso de programación y desarrollo tecnológico

Cualquier desarrollo tecnológico tiene un objetivo determinado, por lo que antes de explicar las herramientas disponibles para alcanzar dicho objetivo, vamos a introducir algunos conceptos básicos sobre los proyectos de desarrollo de *software*.

El proceso de desarrollo de *software* puede definirse como un conjunto de herramientas, métodos y prácticas que se emplean para producir programas. Como cualquier otra organización, las dedicadas al desarrollo de *software* mantienen entre sus principales fines la producción de *software* de acuerdo con la planificación inicial realizada, además de una constante mejora con el fin de lograr los objetivos últimos de cualquier proceso de producción: alta calidad, bajo coste y mínimo tiempo de elaboración. La gestión de este proceso engloba, por tanto, todas las funciones que mantengan un proyecto dentro de unos objetivos de coste, calidad y calendario previamente estimados.

Un proyecto consiste en una serie de acciones, actividades, cambios, etc., conectados entre sí y realizados por agentes con el fin de satisfacer un propósito o conseguir una meta.

Un proyecto de desarrollo se basa en un plan de proyecto en el que se definen las principales características e hitos del mismo, así como otra información necesaria para la consecución del objetivo marcado.



Actividad

Para comprobar que la teoría de los proyectos no es más que la aplicación de la lógica deberéis plasmar por escrito las tareas que se llevan a cabo antes, durante y después de realizar unas vacaciones. Posteriormente se verá la relación directa entre la experiencia real y la teoría presentada.

Las fases principales de un proyecto son:

- **Propuesta del proyecto:** para plantearnos el inicio de cualquier proyecto es necesario que exista una propuesta o iniciativa que fije un objetivo. En esta fase se debe realizar un análisis del coste-beneficio que nos va a aportar el proyecto; además, se deben estimar los esfuerzos implicados en el proyecto.
- **Aprobación del proyecto:** una vez propuesto, se debe seguir el proceso establecido por cada organización para la aprobación del mismo por parte de sus responsables.
- **Inicio del proyecto:** en este punto se deben definir las actividades del proceso, se deben planificar las mismas en un cronograma y se debe realizar la asignación presupuestaria.
- **Ejecución del proyecto:** durante esta etapa se deben llevar a cabo las actividades del proceso según lo previsto en el plan de ejecución. Es importante realizar un seguimiento del avance de las mismas, así como valorar y solucionar los problemas que aparezcan hasta culminar en la entrega y aceptación de los resultados. De todas maneras, aunque a veces no se le dé la importancia que realmente tiene, uno de los puntos más importantes que tiene lugar en esta fase es la consecución exhaustiva de los requerimientos del proyecto.
- **Cierre del proyecto:** en este punto se evaluará la actuación de los participantes y los resultados, y se elaborará un informe final.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Dentro de la fase de ejecución del proyecto se realiza la secuencia de tareas que darán como resultado la aplicación informática definida en el proyecto. Las fases principales durante la ejecución del proyecto son:

- **Recogida de requerimientos:** en esta fase los analistas se reúnen con el responsable funcional de la aplicación (RFA) para recoger las necesidades que debe satisfacer la aplicación. Como resultado se obtiene una matriz con todos los requerimientos ordenados y numerados donde se recoge el grado de cumplimiento de la aplicación para cada uno de ellos (total / parcial / nulo). Este documento tiene que ser validado por el RFA antes de iniciar la siguiente fase.
- **Análisis funcional:** en esta fase se elabora un documento llamado *análisis funcional* donde se explica la funcionalidad que ofrecerá el *software* a partir de la matriz de requerimientos, sin entrar en detalles técnicos. En este documento también aparecen los casos de uso del desarrollo, donde se puede validar la forma de integrar a los diferentes actores que van a intervenir en la aplicación. Este documento debe ser aprobado por el responsable de la aplicación antes de pasar a la siguiente fase.
- **Diseño técnico:** una vez validado el análisis funcional se inicia esta fase, que tiene como objetivo el diseño técnico de la aplicación, que conseguirá resolver las necesidades funcionales planteadas en la fase anterior. En esta fase tiene lugar el diseño de las estructuras de datos necesarias (bases de datos), módulos o partes en las que se dividirá la aplicación o, incluso, la solución tecnológica necesaria para ponerla en marcha (lenguaje a utilizar, tipo de gestor de base de datos, servidores web, etc.). Este documento no debe ser revisado por el usuario, puesto que la parte técnica corresponde a los equipos de sistemas.
- **Construcción:** una vez dividida en módulos la aplicación, en esta fase se asigna cada uno de estos módulos (con las especificaciones de lo que debe hacer y cómo lo debe hacer) a un equipo de programadores para su construcción. En esta fase se necesitará el compilador o intérprete del lenguaje escogido para llevar a cabo la aplicación (ver Tipos de lenguajes) para obtener el *ejecutable*, así como la ayuda de un depurador (*debug*) para detectar el origen de los errores que puedan surgir en las pruebas de cada uno de los módulos.
- **Pruebas integradas:** una vez desarrollados todos los módulos por separado se unen en una única aplicación y se comprueba el correcto funcionamiento de todos ellos según las especificaciones solicitadas. En caso de detectarse cualquier error, se identifica el módulo donde se encuentra y se notifica al equipo de programadores para su corrección.
- **Validación:** antes de poner en marcha la aplicación, se convoca una reunión con el RFA para llevar a cabo la prueba del producto y validarlo. De este modo tenemos una garantía de que lo desarrollado por los equipos de sistemas es realmente lo requerido por el usuario.
- **Producción:** una vez obtenida la validación del usuario, se debe poner en marcha la producción de la aplicación. Previamente, suelen realizarse sesiones de formación de los usuarios para garantizar su correcta utilización por parte de los destinatarios del desarrollo.
- **Seguimiento y garantía:** durante un determinado intervalo de tiempo posterior a la puesta en marcha de la aplicación, debe realizarse un seguimiento de su funcionamiento para detectar posibles errores no detectados en las fases anteriores.

A pesar de que en cada fase hay una o varias personas responsables de la misma, la responsabilidad última del proyecto durante todo el ciclo de vida del mismo es del jefe de proyecto.

Para llevar a cabo todo el ciclo de vida del desarrollo existen herramientas que permiten integrar todas estas fases y que se basan en estándares definidos. Un ejemplo puede ser Rational Rose, que ofrece soluciones para implementar la metodología UML.

Posgrado en Edición Digital (on line y off line) Desarrollo, integración y control de calidad

Proceso de programación y desarrollo tecnológico – Tipos de lenguajes

En este apartado nos centraremos en una de las herramientas necesarias para desarrollar proyectos multimedia: los tipos de lenguajes.

En 1981 apareció la denominada *quinta generación* de ordenadores, en la que se abandonó definitivamente el concepto de ordenador como una simple máquina de cálculo. A partir de ese momento la ordenación y comunicación de información prevalecen.

Los ordenadores personales pasan a ser un electrodoméstico más y la digitalización transforma la distribución de la música, la imagen, la telefonía, etc. Todo el mundo parece aventurar un mundo completamente digital.

Un ordenador puede dividirse en tres grandes partes:

- Unidad Central de Proceso (CPU en inglés).
- Memoria donde se almacenan datos e instrucciones para que sean procesados por la CPU.
- Subsistema de entrada/salida que les permite interactuar con su entorno.

Se han creado diferentes lenguajes para comunicar a los ordenadores lo que deseamos que hagan.

Los ordenadores funcionan procesando un conjunto de datos que se obtienen o comunican mediante el subsistema de entrada/salida y que residen en la memoria. La manera como esos datos deben transformarse es lo que se denomina **programa informático**.

▪ Lenguajes de programación

Un programa informático se escribe siguiendo una serie de reglas y una sintaxis definida por el lenguaje de programación utilizado. Existen distintos lenguajes de programación, cada uno con sus características concretas, que hacen que sean más o menos apropiados para conseguir nuestro programa final.

Veamos algunas de las diferentes tipologías de lenguajes.

Lenguajes de alto y bajo nivel

La CPU de un ordenador sólo sabe ejecutar un conjunto de instrucciones que están codificadas mediante ceros y unos, que es lo que se denomina **lenguaje máquina**.

Crear programas en este lenguaje resulta complejo para las personas, ya que la diferencia semántica entre el lenguaje máquina y el lenguaje natural es enorme.

Si le preguntamos a alguien que nos describa cómo transformar datos, su relato tendrá poco o nada que ver con un conjunto de instrucciones de suma, resta, desplazamiento y comparación de bits que son las únicas operaciones que saben hacer las CPU.

Cuando los primeros ordenadores aparecieron no había otro remedio que programar directamente en estos complicados lenguajes. Sin embargo, con la aparición de nuevos lenguajes de programación, se constató que un programador era capaz de producir una mayor cantidad de líneas de programa por unidad de tiempo. Por tanto, cuanto mayor sea la potencia semántica de las instrucciones, más productivo será su trabajo.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Poco a poco fueron creándose lenguajes de programación más lejanos a la semántica de la CPU y más cercanos a la semántica de las personas: los programadores.

El lenguaje máquina (ceros y unos) es el que se encuentra en el nivel más bajo de la categorización. Nos referimos a lenguajes de alto nivel como aquellos que se encuentran más cercanos a una «sintaxis humana». Un ejemplo de lista de lenguajes ordenada de bajo a alto nivel es: lenguaje máquina, ensamblador, C y Java.

Lenguajes compilados e interpretados

Otro concepto interesante es la manera como los lenguajes de programación se transforman en acciones que deben ser ejecutadas por el procesador:

Un lenguaje puede ser compilado si mediante un proceso informático existe una traducción de las instrucciones del lenguaje original a lenguaje máquina. Los programas que realizan esta traducción se denominan compiladores.

Un lenguaje será **interpretado** si existe otro programa, denominado **intérprete**, que lea las instrucciones del lenguaje original y ponga en marcha rutinas que ejecuten lo que corresponde a esas instrucciones del programador.

Es importante remarcar que cuando un lenguaje es compilado, el resultado de este proceso es una tira de bits que sólo tiene sentido sobre la plataforma en la que se ha llevado a cabo la compilación. Esto se debe a que las instrucciones resultantes son de bajo nivel y sólo pueden ser ejecutadas por el mismo tipo de CPU con la que se obtuvo el ejecutable. Si quisiéramos usar un programa compilado sobre un PC en un MAC, tendríamos que emplear el código fuente del programa, utilizar un compilador del lenguaje en la nueva arquitectura (MAC en este caso) y crear una nueva versión del ejecutable. Fijaos que no es necesario rescribir ni una línea de código, sólo hay que repetir el proceso de compilación.

Este problema de portabilidad del programa compilado entre plataformas desaparece en el caso de un programa interpretado, ya que la entrada del intérprete es el propio programa escrito por nosotros. Así, un programa escrito en Perl, podrá ser ejecutado tanto en MAC, Unix o PC compatible siempre que dispongamos en dicho ordenador del intérprete Perl.

Lenguajes orientados a sentencias y orientados a objetos

La evolución de los lenguajes de programación de bajo a alto nivel ha traído consigo una nueva tipificación de los mismos al evolucionar e integrar interfaces gráficas que les han permitido ser más cómodas para los programadores (*user friendly*).

Dentro de los lenguajes de programación podríamos mantener estos dos tipos:

1. *Lenguajes orientados a sentencia*. Siguen utilizándose, pero cada vez menos, ya que los lenguajes orientados a objetos gozan cada día de mayor popularidad. Se consideran una evolución de los lenguajes máquina pero con un mayor nivel semántico. Ejemplos: Fortran, Basic, C, Pascal.
2. *Lenguajes basados en la orientación a objetos*. Son considerados lenguajes de programación de alto nivel semántico. Se basan en el concepto de que un objeto ofrece una serie de funciones para interactuar con el resto de elementos. Una de las ventajas de estos lenguajes es la facilidad de reutilizar código ya escrito y simplificar los desarrollos.

Ejemplos: Java y C++.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

▪ Lenguajes de autor

La línea que separa los lenguajes de programación de los lenguajes de autor es de difícil definición. Los lenguajes de autor integran habitualmente un lenguaje de programación que permite la potenciación de sus posibilidades.

Los lenguajes de autor son programas que tienen elementos pre-programados para el desarrollo de elementos multimedia interactivos. Algunos de estos sistemas han sido desarrollados de tal manera que el usuario no necesita las habilidades de un programador; tan sólo necesita comprender cómo trabaja el programa y utilizar, de algún modo, una lógica de programación. Estos son los sistemas basados en un paradigma de lenguaje de programación y el usuario tendrá que aprender a manejar códigos para operarlo. Entre estos está el Toolbook.

Con otros, los basados en un paradigma de iconos, el desarrollo del producto será aún más fácil y rápido; algunos de ellos poseen ilimitadas capacidades de interacción, como el Quest. Otros sistemas de esta naturaleza son Authorware e IconAuthor.

Los sistemas de autor permiten que los cursos basados en computador puedan adaptarse fácilmente para la web.

Podríamos agruparlos en:

1. Lenguajes gráficos orientados a la estructura: se basan en una interfaz gráfica que se desarrolla a partir de una secuencia de acciones relacionadas entre sí (diagrama de flujo). Resultan especialmente útiles para la creación de programas de autoaprendizaje y tutoriales.

Ejemplos: AuthorWare, IconAuthor, Apple-MediaTool, Orgue, Course Builder.

2. Lenguajes basados en la orientación a objetos pero referidos a modelos hipertexto: permiten la construcción de libros y tarjetas electrónicas mediante la combinación de diferentes objetos predefinidos como pueden ser botones, sonidos, áreas activas, campos de texto, etc.

Ejemplos: Hypercard, Plus, Supercard, Toolbook, Linkway, Hyperstudio.

3. Lenguajes orientados a la presentación multimedia: son los que se usan en diferentes formatos de interfaz; fueron creados pensando únicamente en el desarrollo de animaciones o presentaciones. Su evolución, al incorporar lenguajes de programación, ha permitido ofrecer gran versatilidad. Asimismo, se están orientando cada vez más a un uso en la web y a sistemas *on-line*.

Ejemplos: Macromedia Director, Macromedia Flash, StoryBoard-Plus.

Veamos las características de algunos de los lenguajes de autor más utilizados en la actualidad.

AuthorWare

AuthorWare permite crear aplicaciones de aprendizaje interactivas en la red (*e-learning*) a través de un lenguaje de autor visual. Permite la integración de gráficos, sonidos, texto, animación y vídeos para su distribución en distintos formatos: redes corporativas, CD-ROM o Internet. También tiene facilidades integradas para monitorizar el progreso de los alumnos.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

ToolBook

ToolBook es una alternativa a AuthorWare, y es una de las herramientas más potentes para la creación de simulaciones avanzadas y contenido interactivo, que es la base de cualquier solución *e-learning*.

Macromedia director

Macromedia Director es un sistema de autor, una herramienta de creación multimedia de posibilidades casi ilimitadas. Sin apenas necesidad de programar podemos desarrollar nuestras propias aplicaciones (presentaciones sencillas, juegos más complicados, enciclopedias interactivas, etc.). Si utilizamos el lenguaje propio de Director, Lingo, se aumentan las posibilidades.

La aplicación nos permite combinar gráficos, sonido, vídeo y prácticamente cualquier tipo de elemento multimedia en el orden que queramos; de hecho, el propio nombre del programa nos da una idea de cómo organiza el trabajo: la metáfora de uso del producto se basa en concebir la producción como si de una película se tratara, en la que nosotros actuamos como director, decidiendo qué actores entran a escena, cómo se sitúan y cuándo.

▪ Lenguajes de marcas

En los lenguajes de marcas, el fichero que contiene el programa distingue dos tipos de objetos: las marcas y los datos sobre los que deben actuar las marcas.

Normalmente estos lenguajes son interpretados y las marcas implican opciones de formato de los datos que las acompañan.

HTML

Antes de conocer más a fondo las características de este lenguaje de marcas vamos a repasar el origen del mismo y de la *World Wide Web*.

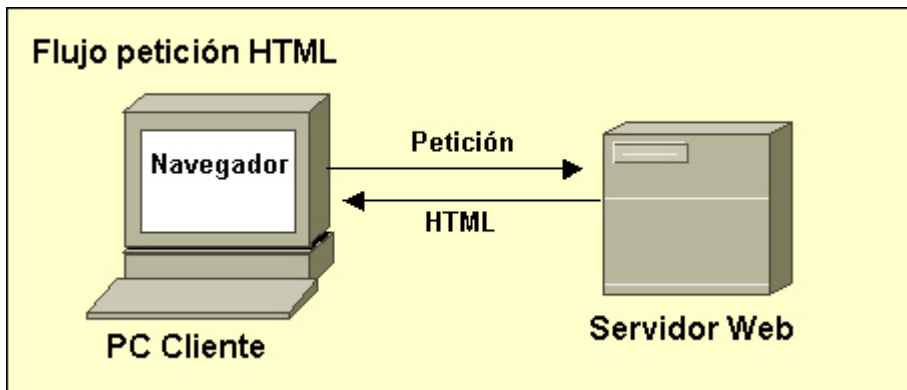
La historia de la *World Wide Web* es bastante joven puesto que comenzó en 1990 en Ginebra, Suiza. Tim Berners-Lee, informático británico en la Organización Europea de Investigaciones Nucleares CERN, inició con ayuda de algunos colegas un proyecto para desarrollar un nuevo sistema de intercambio de informaciones entre científicos a través de Internet. Se trataba de poder compartir documentos científicos en línea, manteniendo el formato de los textos y permitiendo el enlace de imágenes. Pero lo más importante era la idea de emplear una concepción hipertextual, de tal manera que los documentos pudieran contener enlaces a otros documentos, aun cuando éstos se encontrasen en diferentes servidores de Internet.

Para lograr estos objetivos, se concibió un nuevo formato de archivo, denominado HTML (*Hypertext Markup Language*) y se pensó un protocolo de transferencia, llamado HTTP (*Hypertext Transfer Protocol*). Para representar los archivos en línea se recurrió a un nuevo *software*: los navegadores. El proyecto fue bautizado *World Wide Web* (Red mundial) por el carácter de hipertexto.

Al mismo tiempo se comenzaron a configurar servidores WWW que deberían apoyar el nuevo protocolo http.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad



Una vez ubicados en la historia, ya podemos afirmar que HTML es el estándar de facto para la publicación de hipertextos en la *World Wide Web* y es un claro ejemplo de lenguaje de marcas. Su definición no pertenece a ningún fabricante de *software* en concreto, es un *software no propietario*. Sin embargo, los fabricantes de navegadores han introducido pequeñas variaciones en el lenguaje que hacen que los documentos funcionen de forma diferente en los distintos navegadores. Para dejar de lado la universalidad original de este lenguaje, existen organismos que velan por su estandarización y comprueban que los contenidos programados cumplan estrictamente los estándares.

La manera de crear hipertextos con HTML varía desde un simple editor de textos a complejas herramientas WYSIWYG (*What you see is what you get*) en las que se diseñan páginas web según su aspecto final. Un ejemplo de dicha herramienta es el DreamWeaver de Macromedia.

La estructura básica de HTML es un conjunto de marcas diferenciadas del resto de texto por su inclusión entre paréntesis angulares (< >). Las marcas se abren con el nombre de la marca entre paréntesis angulares y se cierran mediante el nombre de la marca precedido por una barra (/) entre paréntesis angulares. Dichas marcas pueden anidar otras marcas y/o textos. En HTML varios espacios en blanco seguidos, tabuladores y los saltos de línea de los ficheros de texto no tienen significado. Deben especificarse con las marcas apropiadas.



Podemos estructurar un fichero HTML de manera que se aprecie visualmente la apertura de una marca y su cierre estableciendo diferentes niveles de tabulación.

```

<html>
  <body>
    <h1> Este es el <i>Título</i> del documento </h1>
    <p> Y el resto es el <b>Contenido</b> del documento</p>
  </body>
</html>
  
```

La marca <html> comienza la definición de un documento, <body> indica el cuerpo del documento, <h1> es un tipo de párrafo de título, y <p> un párrafo estándar. Las marcas <i> y implican que el texto que aparece afectado por ellas será visualizado en itálica (cursiva) y en negrita respectivamente. Su visualización sería algo similar a:

Este es el *Título* del documento

Y el resto del **Contenido** del documento.

De cara a facilitar la portabilidad de documentos HTML en diferentes plataformas, se pueden usar caracteres ASCII cuyo código sea menor de 128. Por lo tanto los caracteres internacionales (acentos, tildes, etc.) se marcan mediante una notación especial que comienza

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

por el signo ampersand (&) y acaba por punto y coma (;). Por ejemplo la *a* minúscula acentuada en castellano se representa por «á»; y la letra ñ por «ñ».

Un documento HTML está asociado a una URL y puede referenciar tantos documentos HTML como se requiera usando hipertextos que los ligan con las URL respectivas.



```
<p>La p&aacute;gina principal de la <a href="http://www.uoc.es">UOC</a> es un lugar a visitar.</p>
```

La página principal de la [UOC](http://www.uoc.es) es un lugar a visitar.

Asimismo, un documento HTML puede incluir referencias a cualquier tipo de elementos multimedia, imágenes, películas, etc., mediante la utilización de las marcas adecuadas.



```
<p>El esquema de la circulaci&oacute;n sangu&iacute;nea en el que se puede observar  el coraz&oacute;n como parte central del sistema.</p>
```

En vista de lo anterior, resumiremos diciendo que HTML ofrece hipertextos formateados acompañados de elementos multimedia pero bastante estáticos.

Para permitir la creación de páginas web más dinámicas, se enriqueció este lenguaje y se desarrolló el DHTML (*Dynamic HTML*) que veremos a continuación.

Un paso más avanzado en la obtención de interactividad con el usuario se consigue con la inclusión dentro de las páginas HTML de lenguajes como JavaScript, Java o Flash, de los que se hablará más adelante.

DHTML

El HTML dinámico es un término colectivo para designar una combinación de nuevas etiquetas y opciones, hojas de estilo y programación en Lenguaje de Marca de Hipertexto (HTML, siglas inglesas de *Hypertext Markup Language*) que permite crear páginas web más animadas y que respondan en mayor grado a la interacción del usuario que las versiones previas del HTML.

Ejemplos de las posibilidades de páginas creadas con DHTML podrían ser (1) que el color de un encabezado de texto cambie cuando el usuario pase el ratón sobre él o (2) permitir que el usuario «arrastre y suelte» una imagen hacia otro lugar de la pantalla. El HTML Dinámico permite que los documentos de la página web se vean y se comporten como aplicaciones del escritorio o producciones multimedia.

Aunque el HTML 4.0 está soportado tanto por los navegadores Netscape como Microsoft, algunas capacidades adicionales sólo sirven para uno u otro. El mayor obstáculo para el uso del HTML Dinámico es que, como no existe un estándar rígido, los sitios web deben crear varias versiones de cada página y ofrecer la apropiada para la versión de navegador del usuario.

Los principales conceptos y características del HTML Dinámico, desde el punto de vista de usuario son:

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Hojas de estilos y capas

Una hoja de estilo en cascada o CSS (*Cascading Style Sheets*) describe las características de estilo por omisión (incluyendo el diseño de la página, el tipo y estilo de la fuente, y el tamaño de los elementos de texto, tales como encabezados y texto de cuerpo) de un documento o parte de él. Para las páginas web, una hoja de estilo también describe el color o imagen de fondo por omisión, los colores de los enlaces de hipertexto y, posiblemente, el contenido de una página. Las hojas de estilo ayudan a asegurar la coherencia en todas las páginas (o un grupo de ellas) en un documento o un sitio web. También facilitan la modificación de presentación de toda la web ya que, cambiando únicamente los archivos CSS de la misma, tendremos un nuevo estilo de presentación en todas aquellas páginas que utilicen las hojas de estilo modificadas, evitando tener que realizar los cambios de forma manual, página a página.

La distribución en capas (*layering*) es el uso de hojas de estilo alternas u otras aproximaciones para variar el contenido de una página proporcionando capas (*layers*) o niveles de contenido que puedan situarse por encima de secciones de contenido existentes, reemplazándolas o superponiéndose a ellas. Las capas pueden programarse para aparecer como parte de una presentación, vinculadas a una línea de tiempo predeterminada o como resultado de la interacción del usuario. Internet Explorer 4.0 de Microsoft implementa capas en todas las hojas de estilo. Netscape apoya la aproximación de hojas de estilo pero también ofrece un juego de etiquetas HTML (que Microsoft no soporta). Ambas aproximaciones están siendo analizadas por el Comité de Trabajo W3C, y las dos empresas dicen que apoyarán cualquiera que sea la aproximación recomendada que decida W3C.

Fuentes dinámicas

Netscape incluye fuentes dinámicas como parte del HTML dinámico. Esta característica del navegador Netscape permite a los diseñadores de páginas web incluir archivos de fuentes que contengan estilos, tamaños y colores específicos como parte de la página web para facilitar que las fuentes se descarguen junto con la página, consiguiendo que la elección de fuente ya no dependa de la tipografía disponible en el sistema del usuario.

No hay una especificación única para el HTML dinámico dado que es, en realidad, una combinación de varios elementos. Netscape y Microsoft tienen, cada uno, sus propias definiciones en sus sitios web respectivos.

XML

XML es la especificación de un lenguaje de marcas para documentos que contienen información estructurada.

La versión 1.0 del lenguaje XML es una recomendación del W3C de febrero de 1998. Está basado en el anterior estándar SGML (*Standard Generalized Markup Language*, ISO 8879), cuya traducción significa «Lenguaje de Etiqueta Generalizado Estándar», que data de 1986 y está basado en el GML creado por IBM en 1969.

Esto significa que aunque XML pueda parecer moderno, sus conceptos están asentados y aceptados de forma amplia; XML está asociado a una recomendación internacional de modelización de la estructura interna de documentos (DOM, siglas inglesas de *Document Object Model*) aprobada en 1998.

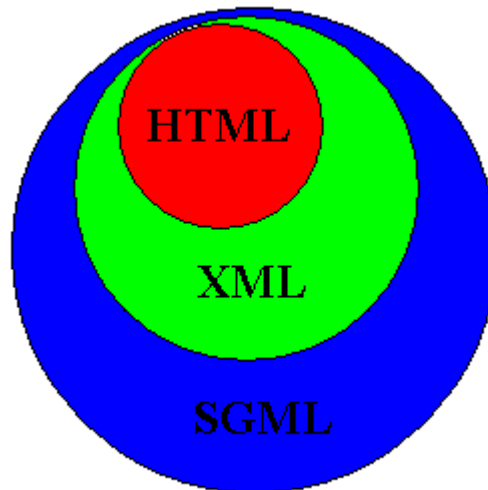
SGML proporciona un modo coherente y preciso de aplicar etiquetas para describir las partes que componen un documento, permitiendo el intercambio de documentos entre diferentes plataformas. El problema que se atribuye a SGML es su excesiva dificultad. Manteniendo su

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

misma filosofía, de él se derivó XML como subconjunto simplificado, eliminando las partes más engorrosas y menos útiles. Como su predecesor, XML es un metalenguaje, es decir, un lenguaje para definir lenguajes. Los elementos que lo componen pueden dar información sobre lo que contienen, no necesariamente sobre su estructura física o presentación como ocurre en HTML.

En una primera aproximación se puede decir que mediante XML también podríamos definir el HTML, con lo que podríamos considerar los siguientes conjuntos:



Una pregunta que nos podemos hacer es si XML será el sustituto de HTML. La respuesta es NO. Básicamente XML no ha nacido sólo para su aplicación en Internet, sino que se propone como lenguaje de bajo nivel (en el ámbito de la aplicación, no de la programación) para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo, y casi cualquier cosa que podamos pensar.

A pesar de que XML no sustituirá al HTML, éste está mejorando algo de lo que HTML adolece desde hace tiempo: establece un estándar fijo y separa el contenido de su presentación. Esto significa que para ver un documento web no estaremos sujetos a la parte del estándar de hojas de estilo (CSS) que soporte el Navigator de Netscape o el IExplorer de Microsoft, ni al lenguaje de *script* del servidor y al modelo de objetos definido por el navegador en cuestión. Además, tampoco estaremos atados a la plataforma, pudiendo ver la misma información desde nuestro PC, desde un navegador textual, una lavadora, un microondas o un reloj con acceso a Internet, con presentaciones adecuadas a cada entorno.

La información estructurada por los ficheros XML está compuesta por palabras, imágenes y otros elementos multimedia junto con una indicación del papel que juega cada elemento dentro de la estructura del documento.

Mediante esta estructuración podemos contextualizar los elementos y lograr diferenciar, por ejemplo, un texto que aparece en el título del documento de otro que aparece al pie de una figura. Ésta es, por ejemplo, una gran ventaja sobre la información difundida en HTML, donde se pierde la noción del contexto a fin de incluirla en buscadores.

El concepto de documento en XML va más allá de lo que podamos pensar en un primer momento. Lejos de ser otra manera de escribir información con un formato más o menos agradable visualmente, un documento XML puede representar ecuaciones matemáticas, transacciones económicas de comercio electrónico, definiciones de piezas mecánicas, especificaciones de maquinaria de cualquier tipo, etc.

Posgrado en Edici3n Digital (on line y off line)

Desarrollo, integraci3n y control de calidad

Una gran ventaja que ofrecen los documentos XML es que permiten estructurar la informaci3n de manera que pueden obtenerse diversas y variopintas visiones de ella: bases de datos, documentos impresos, documentos sonoros, etc. Todo ello es posible gracias a los programas que interpretan los documentos XML.

Pero la mayor ventaja de los documentos XML es, sin duda, la separaci3n existente entre el contenido y la presentaci3n gr1fica del mismo.

Reflexionemos sobre la gran ventaja que supone para un editor multimedia poder actualizar una obra en XML y ser capaz de obtener autom1ticamente la versi3n correspondiente en HTML, PDF, WAP, texto braille y audible (mediante programas de s3ntesis de voz). Este enorme potencial hace que XML gane cada d3a m1s terreno.

Para entender el porqu3 de un lenguaje de marcas tan innovador se debe tener en cuenta que se cre3 para poder difundir documentos altamente estructurados en Internet. Los requerimientos que se tuvieron en cuenta en su creaci3n fueron diversos:

- Deb3a ser f1cil usar XML en Internet. Los usuarios deb3an poder visualizar esos documentos tan f1cil y r1pidamente como los documentos HTML. Actualmente no es una realidad, pero tan s3lo debemos esperar a que los navegadores de XML hayan evolucionado tanto como los navegadores de HTML.
- XML deb3a dar soporte a una amplia variedad de aplicaciones.
- Deb3a ser sencillo escribir programas que procesen documentos XML. En t3rminos pr1cticos, es deseable que un ingeniero en inform1tica con escasa experiencia profesional no tarde m1s de dos semanas en crear un programa que procese documentos XML.
- Se deber3a mantener en un m3nimo, idealmente cero, la cantidad de caracter3sticas opcionales de XML, cuya aparici3n no hace sino crear problemas de compatibilidad a la hora de compartir documentos.
- Los documentos XML deber3an ser legibles sin necesidad de un navegador para que cualquier persona con conocimientos de XML pudiera hacerse una idea general del contenido del documento abri3ndolo con un simple editor de textos.

A primera vista un documento XML puede parecer id3ntico en estructura a un documento HTML ya que ambos son un conjunto de marcas entre par3ntesis angulares y textos a los que se aplican. Pero la gran diferencia es que las marcas de XML no vienen predefinidas como en HTML.

XML es un metalenguaje que carece de sem1ntica y de un conjunto prefijado de marcas. Por lo tanto la sem1ntica de un documento XML se define por las aplicaciones que lo usan o por las hojas de estilo que lo acompa1an.



```
<?xml version="1.0"?>
<cita>
<texto>Hay m1s llantos en la Tierra por favores concedidos que por favores
negados</texto>
<autor> Santa Teresa de Jes1s </autor>
<pausa/>
<moraleja> Cuidado con lo que pides porque a lo mejor lo consigues </moraleja>
```

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

</cita>

De este pequeño documento en XML, resulta interesante poner de relieve:

El documento comienza con la instrucción de proceso <?xml ...?> la cual no es imprescindible según el estándar, pero informa de que se trata de un documento XML y de la versión del mismo.

El significado de cada una de las marcas es arbitrario y debe especificarse aparte.

Los elementos vacíos tienen una sintaxis modificada y empiezan por (<) y acaban por (/>). Mientras que la mayoría de las marcas delimitan algún contenido, los elementos vacíos son simplemente marcas que implican alguna acción. Sería similar a las marcas
 o <hr> de HTML que implican un salto de línea o el dibujo de una raya horizontal.

Los documentos XML tienen seis tipos de marcas: elementos, referencias a entidades, comentarios, instrucciones de proceso, secciones marcadas y declaraciones de tipo de documento (DTD).

En vista de la libertad que permite la definición de un documento XML, es imprescindible que quien deba interpretar el contenido sepa cómo hacerlo. Para ello los documentos XML deben referir a un DTD.

DTD son las siglas inglesas de *Document Type Definition*, y es la definición de los elementos del documento XML y su relación entre ellos, sus atributos, posibles valores, etc. Es una especie de definición de la gramática del documento. Cuando se procesa cualquier información formateada mediante XML lo primero es comprobar si está bien formada, y luego, si incluye o referencia a un DTD, comprobar que sigue sus reglas gramaticales.

Es una buena práctica escribir el DTD como un documento separado para:

- No tener que incluir el DTD en cada documento XML que usa esa definición.
- Estandarizar la descripción de los elementos usando DTD que se entiendan como válidos para un ámbito de conocimiento, para un conjunto de empresas, etc. Ya existen tales estándares para ámbitos como la creación de libros (DocBook) o el intercambio de fórmulas matemáticas (MML).

Hay dos tipos de documentos XML:

Bien formados: son todos los que cumplen las especificaciones del lenguaje sin estar sujetos a unos elementos fijados en un DTD. De hecho, los documentos XML deben tener una estructura jerárquica muy estricta y los documentos bien formados deben cumplirla.

Válidos: además de estar bien formados, siguen una estructura y una semántica determinada por un DTD; sus elementos y sobre todo la estructura jerárquica que define el DTD, además de los atributos, deben ajustarse a lo que el DTD dicte.

Presentación y transformación de XML: CSS y XSLT

Ya hemos visto anteriormente que un documento XML no lleva información sobre su presentación gráfica; sin embargo, está claro que necesitamos un mecanismo para poder realizar la presentación de la información del documento. Esto se puede conseguir a través de dos mecanismos:

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

1) Hojas de estilo CSS

Como hemos visto, este lenguaje de hojas de estilo se utiliza normalmente para controlar la presentación de documentos HTML, pero también se puede utilizar con documentos XML. Éstas son las principales ventajas y desventajas respecto al lenguaje que veremos a continuación:

Ventajas:

- Fácil de aprender y utilizar y de amplia difusión entre los desarrolladores.
- Consume poca memoria y tiempo de proceso, pues no construye una representación en árbol del documento.
- Muestra el documento según se va procesando.

Desventajas:

- Sólo sirve para visualizar documentos en un navegador.
- No permite realizar manipulaciones sobre el documento, tales como añadir y borrar elementos, realizar ordenaciones, etc.
- Sólo permite acceder al contenido de los elementos y no a los atributos, instrucciones de proceso, etc.
- Utiliza una sintaxis diferente a la del XML.

2) Lenguaje XSLT

Para subsanar las limitaciones presentadas por las hojas de estilo CSS, el consorcio W3C se puso a trabajar en la nueva recomendación XSL (*eXtensible Stylesheet Language*). Según el W3C, XSL es «un lenguaje para transformar documentos XML», así como un vocabulario XML para especificar la semántica de formateo de documentos.

En definitiva, además del aspecto que ya incluía CSS referente a la presentación y estilo de los elementos del documento, añade una pequeña sintaxis de lenguaje de programación para poder procesar los documentos XML de forma más cómoda.

XSL se compone de dos partes diferenciadas:

XSLT: lenguaje para transformar documentos XML en otro formato (otro XML, HTML, DHTML, texto plano, PDF, RTF, Word, etc.).

XSL-FO: especificación que indica cómo deben ser los objetos de formato para convertir XML a formatos binarios (PDF, Word, imágenes, etc.).

La principal característica del lenguaje XSLT es su potencia. No es sólo un lenguaje para visualizar documentos, sino para transformarlos y manipularlos.

Esta manipulación la gestiona un programa especial, el procesador XSLT. Existen distintos procesadores disponibles, como Xalan, del proyecto en colaboración de código abierto Apache.

Las ventajas y desventajas de este lenguaje son:

Ventajas:

- La salida no tiene por qué ser HTML para visualización en un navegador, sino que puede estar en muchos formatos.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

- Permite manipular de muy diversas maneras un documento XML: reordenar elementos, filtrar, añadir, borrar, etc.
- Permite acceder a todo el documento XML, no sólo al contenido de los elementos.
- XSLT es un lenguaje XML, por lo que no hay que aprender nada especial acerca de su sintaxis.

Desventajas:

- Su utilización es más compleja.
- Consume cierta memoria y capacidad de proceso, pues se construye un árbol con el contenido del documento.

Las formas de uso de este lenguaje son múltiples, aunque éstas son las más habituales:

- Visualizar directamente en un navegador el documento XML que tiene asociada una hoja XSLT. El navegador debe tener incorporado un procesador XSLT.
- Ejecutar el procesador XSLT independientemente del navegador. Se le pasan las entradas necesarias (fichero origen y hoja XSLT a utilizar) y genera la salida en un fichero, con el que podemos hacer lo que queramos.
- Realizar las transformaciones dentro de un programa en el servidor y enviar a los clientes sólo el resultado de la transformación.



A partir del documento XML inferior, vamos a realizar una hoja de estilo XSLT que mostrará los datos en una tabla con las siguientes características: las empresas estarán ordenadas por el precio de cotización, los precios superiores a 75 aparecerán en azul y los inferiores a 25 en rojo, las empresas del índice general se marcarán con un asterisco (*), con una explicación debajo de la tabla. El título del documento contendrá el día y hora de la información:

```
<?xml version="1.0" ?>
<Bolsa xmlns="http://www.labolsa.com"
  dia="5-7-2001"
  hora="11:34">
  <Empresa indice="general">
    <Nombre>General Motors</Nombre>
    <Simbolo>GMO</Simbolo>
    <Precio>28.875</Precio>
  </Empresa>
  <Empresa indice="tecno">
    <Nombre>Adobe</Nombre>
    <Simbolo>ADB</Simbolo>
    <Precio>92.250</Precio>
  </Empresa>
  <Empresa indice="tecno">
    <Nombre>Microsoft</Nombre>
    <Simbolo>MSF</Simbolo>
    <Precio>20.313</Precio>
  </Empresa>
  <Empresa indice="general">
```

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

```

    <Nombre>Coca-Cola</Nombre>
    <Simbolo>COC</Simbolo>
    <Precio>38.895</Precio>
  </Empresa>
  <Empresa indice="tecno">
    <Nombre>Sun Microsystems</Nombre>
    <Simbolo>SUN</Simbolo>
    <Precio>45.119</Precio>
  </Empresa>
</Bolsa>

```

Ésta es la hoja de estilo XSLT que nos presentará el resultado con las características comentadas:

```

<?xml version="1.0" encoding="ISO-8859-1" ?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:Bolsa="http://www.labolsa.com"
  version="1.0">

  <xsl:template match="Bolsa:Bolsa">
    <html>
      <head>
        <title>La bolsa el <xsl:value-of select="@dia" /> a las
          <xsl:value-of select="@hora" /></title>
      </head>
      <body>
        <table border="2" align="center">
          <tr>
            <th>Simbolo</th>
            <th>Nombre</th>
            <th>Precio</th>
          </tr>
          <xsl:for-each select="Bolsa:Empresa">
            <xsl:sort select="Bolsa:Precio" order="descending"/>
            <tr>
              <td>
                <xsl:value-of select="Bolsa:Simbolo"/>
              </td>
              <td>
                <xsl:value-of select="Bolsa:Nombre"/>
                <xsl:if test="@indice='general'"> (*)</xsl:if>
              </td>
              <td>
                <xsl:choose>
                  <xsl:when test="Bolsa:Precio > 75">
                    <font color="blue"><xsl:value-of select="Bolsa:Precio"/></font>
                  </xsl:when>
                  <xsl:when test="Bolsa:Precio < 25">
                    <font color="red"><xsl:value-of select="Bolsa:Precio"/></font>
                  </xsl:when>
                  <xsl:otherwise>
                    <xsl:value-of select="Bolsa:Precio"/>
                  </xsl:otherwise>
                </xsl:choose>
              </td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </template>

```


Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

```
</table>
<p align="center">(*) Estas empresas son del índice general</p>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Las expresiones que pueden aparecer en los atributos *select*, *match* o *test* de los elementos incluidos en XSLT siguen el lenguaje XPath. La especificación XPath contiene muchas características; aunque aquí no las explicaremos, sí es interesante comentar que con este lenguaje se pueden crear funciones tan complejas como se requiera.

XHTML

XHTML es una reformulación del tipo de documento HTML como aplicaciones de XML. Su finalidad es su uso como lenguaje de contenidos que, a su vez, sea conforme a XML por lo que, siguiendo algunas sencillas directrices, funcione tanto en agentes de usuario conformes a HTML (por ejemplo los navegadores) como en herramientas estándares XML.

La familia XHTML es el siguiente paso en la evolución de Internet. Al migrar hacia XHTML, los desarrolladores de contenidos web entran en el mundo de XML con todos los beneficios de éste, a la vez que se aseguran la compatibilidad con agentes de los usuarios pasados y futuros.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Proceso de programación y desarrollo tecnológico – Aplicaciones para la interactividad

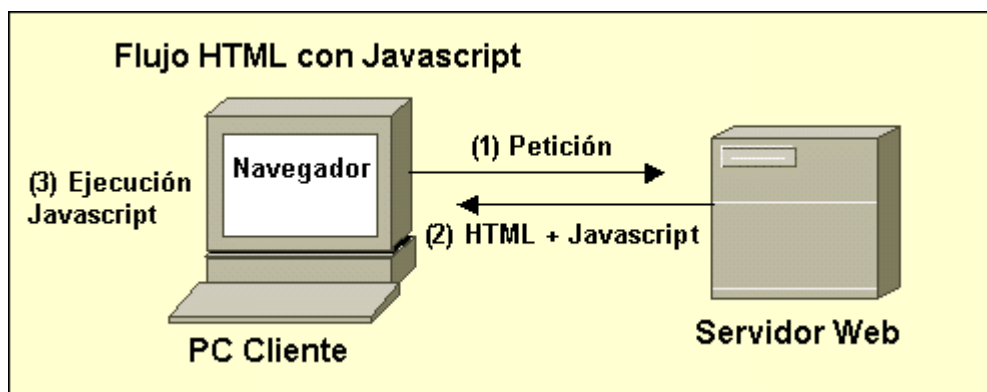
En este apartado veremos cómo dotar de interactividad a nuestros sitios web.

■ JavaScript

JavaScript no puede considerarse estrictamente un lenguaje de programación. Su misión es acompañar al lenguaje de marcas HTML para asociar pequeñas rutinas que ejecuta el navegador. Estas rutinas pueden servir para:

Asociación de animación a diferentes eventos generados por el usuario del navegador: apertura de una nueva ventana de ayuda al pasar el cursor por un campo, creación por parte del navegador de mensajes a partir de datos introducidos por el usuario e información proporcionada por el navegador.

Validación de los datos introducidos en formularios: validación previa antes de que esos datos pasen al servidor para ser tratados. Ejemplos de esta validación podrían ser: detección de si un campo obligatoriamente sigue vacío o no, validación de un NIF, etc.



Sin JavaScript la interactividad se debe obtener empleando rutinas que se ejecutan en el servidor de páginas web; esto da lugar a una nueva transacción, sobrecarga el trabajo del servidor y aumenta los tiempos de espera del cliente.

JavaScript, por muy parecido a Java que suene, no tienen mucho que ver. Java sí es un lenguaje de programación con el que pueden construirse los más complejos algoritmos. Es importante tener clara la diferencia entre ambos

■ Flash y SVG

Flash es la tecnología propietaria de Macromedia más comúnmente utilizada en Internet y permite la creación de animaciones vectoriales. El interés en el uso de gráficos vectoriales es que éstos permiten llevar a cabo animaciones de poco peso, por lo que tardan poco tiempo en ser cargadas por el navegador.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad



Flash se sirve, aparte de las posibilidades que ofrece el trabajar con gráficos vectoriales, de un almacenamiento inteligente de las imágenes y sonidos de sus animaciones para optimizar el tamaño de los archivos. Esta optimización del espacio, combinada con la posibilidad de cargar la animación al mismo tiempo que ésta se muestra en el navegador (*streaming*), permite aportar elementos visuales que dan vida a una web sin que para ello el tiempo de carga de la página se prolongue excesivamente.

Además, Flash introduce en su entorno la posibilidad de interactuar con el usuario invocando un lenguaje de programación llamado Action Script, orientado a objetos y con influencias del Javascript que permite, entre otras cosas, gestionar el relleno de formularios, ejecutar distintas partes de una animación en función de eventos producidos por el usuario, saltar a otras páginas, etc. De este modo conseguimos una tecnología que aporta vistosidad a nuestra web al tiempo que nos permite interactuar con nuestro visitante.

Las animaciones Flash son producidas mediante una aplicación de diseño. Se puede descargar una aplicación en versión de prueba desde el sitio de Macromedia.

La aplicación se divide en tres partes principales:

- **Escenario:** se trata del espacio en el cual llevaremos a cabo todas las tareas de edición de gráficos. En cierto modo es como el papel donde dibujaríamos nuestro proyecto de diseño.
- **Línea de tiempo:** esta sección es donde podremos organizar en el tiempo cada una de las imágenes diseñadas en el escenario.
- **Caja de herramientas:** aquí encontraremos las herramientas de edición gráfica que Flash pone a nuestra disposición.

El modo de trabajo de Flash es similar al de una película. Una animación es una sucesión de imágenes fijas que, al pasar rápidamente unas tras otras, dan la impresión de movimiento. Cada una de estas imágenes fijas es llamada también *fotograma*. Los fotogramas, en Flash, pueden ser de tres tipos:

- **Imágenes clave:** son las imágenes que nosotros mismos dibujamos.
- **Imágenes fijas:** son las imágenes clave copiadas en los fotogramas siguientes al de la primera imagen clave para producir un efecto de objeto estático.
- **Imágenes de interpolación:** se trata de imágenes calculadas por Flash que permiten la transición gradual entre dos imágenes clave. Estas imágenes son muy útiles, ya que se generan automáticamente y proporcionan un efecto suave de movimiento o transformación.

Por otra parte, una animación está constituida por una variedad de objetos diferentes, cada uno de los cuales se introduce en un momento diferente y presenta un comportamiento

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

independiente del resto de los objetos. Para organizar y editar todos estos elementos, Flash permite el uso de capas o calcos.

Así, una animación Flash está compuesta por una superposición de capas sobre las que podemos introducir un objeto que tendrá su propia línea de fotogramas. Estas capas nos permiten trabajar la animación en distintos planos independientes. La base para crear animaciones de calidad es el uso inteligente de las capas.

Por último, cabe añadir que la utilización de bibliotecas es una de las claves para la obtención de un almacenamiento óptimo de las animaciones Flash. Las bibliotecas no son más que almacenes de objetos (gráficos o sonidos) que podrán ser utilizados en una misma animación en una o más ocasiones. Ésta puede ser propia a la animación, compartida por varias animaciones o bien permanente (empleada por todas las animaciones).

SVG es un formato alternativo a Flash para la creación de gráficos vectoriales, aunque todavía no ha alcanzado ni la potencia ni la popularidad de aquél. Su nombre proviene de las siglas *Scalable Vector Graphics* (Gráficos Vectoriales Escalables) y es un formato basado en XML; su desarrollo está a cargo del consorcio W3C (*World Wide Web Consortium*).

■ CGI

La denominada *Common Gateway Interface* (CGI) es una definición de la manera como se debe realizar la comunicación entre un navegador de Internet y un servidor de páginas web para obtener contenidos dinámicos.

Por extensión se denomina CGI a cualquier programa que se ejecute en el servidor de páginas web a petición de un navegador.

Programar un CGI es relativamente sencillo puesto que las reglas que se imponen son pocas y simples: existen dos métodos para enviar parámetros hacia el servidor: GET y POST. En el primer caso los parámetros forman parte de la URL; en el segundo se comunican de manera que no resulta visible al usuario.

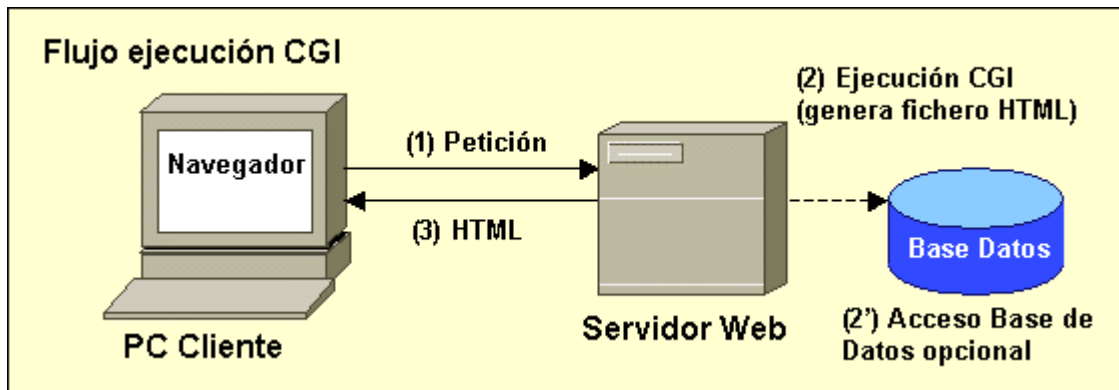
Al acceder a la URL de un CGI, el servidor de páginas web ejecuta el programa al que se refiere ésta y le pasa:

- Los parámetros mencionados anteriormente.
- Un conjunto de variables internas que el CGI usa para obtener datos como: el navegador que usa el cliente, el nombre del servidor de páginas web, la dirección del ordenador sobre el que se ejecuta el navegador, credenciales de identificación de usuario (si es que existen), lista de cookies existentes en el navegador, etc.

El servidor de páginas web espera a que concluya la ejecución del programa y devuelve al navegador toda la información que el CGI le ha proporcionado mediante un canal de comunicación interno que se establece entre ambos. Típicamente la información proporcionada es código HTML.

Posgrado en Edici3n Digital (on line y off line)

Desarrollo, integraci3n y control de calidad



Lenguajes de programaci3n que se usan habitualmente para este prop3sito son Perl o Python. Tambi3n es posible programar CGI en C, C++, o cualquier otro lenguaje de programaci3n.

El uso que se puede dar a los CGI es muy diverso, tan s3lo depende de la imaginaci3n del programador. Usos t3picos de esta tecnologa son:

- **Interacci3n con bases de datos:** introducci3n de datos o generaci3n de contenidos dinamicos en funci3n de datos almacenados en ellas.
- **Pasarela hacia otros servicios como:** correo electr3nico, servicios de b3squeda de contenidos, banca electr3nica, informaci3n meteorol3gica, etc.

Por 3ltimo, es importante comentar que PHP es un lenguaje de programaci3n que se ejecuta en el servidor web, permite crear contenido dinámico en las p3ginas HTML y dispone de m3ltiples herramientas que permiten acceder a bases de datos de forma sencilla, por lo que es ideal para crear aplicaciones en Internet.



C3digo de HTML con PHP:

```

<html>
<head>
<title> Ejemplo PHP </title>
</head>
<body>
<?php
$navigator = getenv("HTTP_USER_AGENT");
?>
<P>Estas usando el navegador
<b>
<?php
echo($navigator);
?>
</b>
</P>
</body>
</html>

```

Visualizaci3n en el navegador:

Est3s usando el navegador **Mozilla/4.71 [en] (Win95; I)**.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Como se puede observar, el código PHP empieza con (<?php) y termina con (?>), las instrucciones finalizan con punto y coma (;) y se puede insertar código donde se quiera.

■ Java

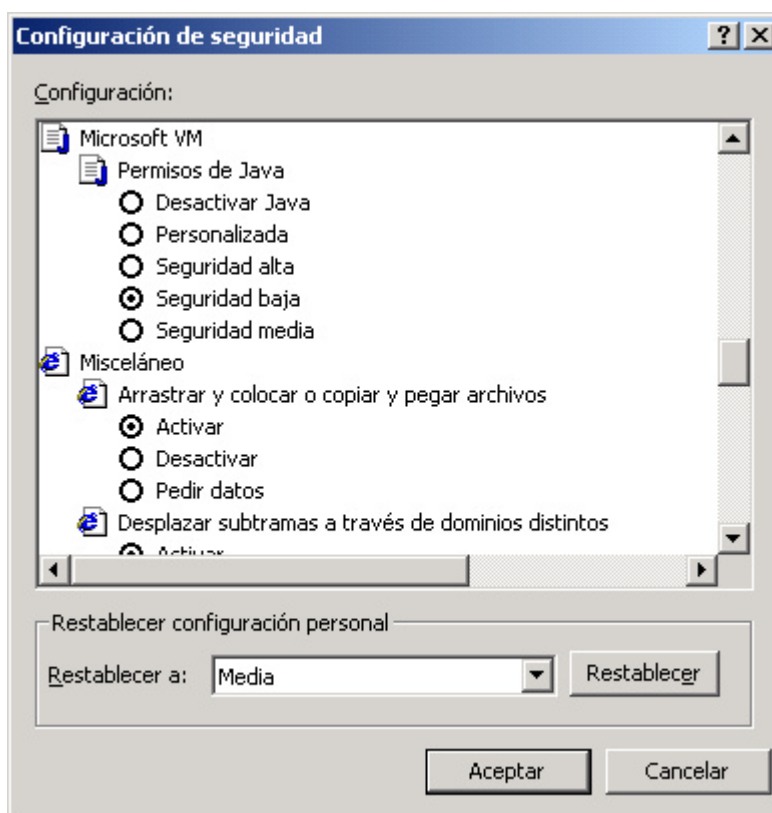
El lenguaje de programación Java fue creado por Bill Joy; está orientado a objetos y puede ser equivalente en funcionalidades a, por ejemplo, C++.

Este lenguaje siempre se compila a un lenguaje intermedio (*bytecode*) que es interpretado por la que se denomina «máquina virtual Java» (*Java Virtual Machine*). Todo navegador o servidor de páginas web que soporte Java debe tener una JavaVM. Para ejecutar programas de propósito general sólo es preciso disponer de una JavaVM que se ejecute sobre el sistema operativo de nuestra preferencia.

Las ventajas de Java son numerosas y por ello se está popularizando para el desarrollo de aplicaciones de todo tipo.

Navegadores

En nuestros navegadores, si no tenemos activa la opción de «Desactivar Java» (ver imagen inferior de ejemplo en Internet Explorer) disponemos de una máquina virtual Java que nos permite la ejecución de programas escritos en este lenguaje integrados en las páginas web que nos descargamos: los denominados *applets*.

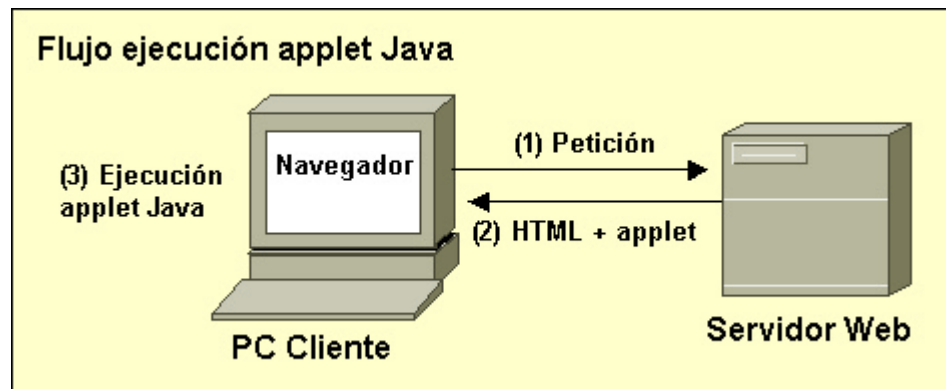


Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

La definición más extendida de *applet* indica que es «una pequeña aplicación accesible en un servidor Internet, que se transporta por la red, se instala automáticamente y se ejecuta in situ como parte de un documento web». En realidad, un *applet* es una aplicación corta (aunque en teoría nada impide que ocupe más de un gigabyte) basada en un formato gráfico sin representación independiente, por lo que es un elemento que se añadirá en otras aplicaciones.

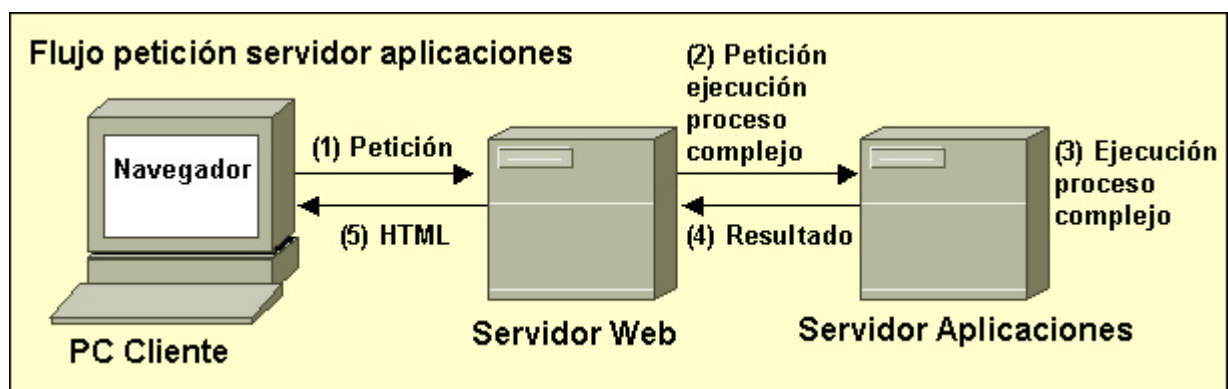
Dado que no existe una base adecuada para soportar aplicaciones industriales Java en las que insertar nuestras miniaplicaciones, los *applets* se han desarrollado mayoritariamente como pequeñas aplicaciones interactivas, con movimiento, luces y sonido, etc., en Internet.



Servidores de aplicaciones

Una vez extendidos por toda la red, los servidores web con sus contenidos estáticos y sus programas accesibles a través de un CGI, surgió la necesidad de desarrollar aplicaciones más complejas, con mayor seguridad y con una potencia de cálculo cada vez mayor.

Para solucionar esta necesidad aparecen las llamadas aplicaciones J2EE que resuelven el problema del coste y la complejidad del desarrollo de servicios escalables, de alta disponibilidad, seguros y eficientes. Esto se consigue proporcionando una arquitectura de estándar abierto a través de la Plataforma J2EE y del modelo de Aplicación J2EE. J2EE es un entorno abierto para desarrollar y desplegar servicios donde pequeñas aplicaciones cliente (habitualmente navegadores) invocan aplicaciones que se ejecutan en un servidor de aplicaciones como BEA Weblogic, IBM Websphere o iPlanet Application Server. El lenguaje Java, la máquina virtual Java y los componentes JavaBeans son la base de J2EE.



Veamos cuáles son los elementos base de esta plataforma:

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Servlets

Los *servlets* son la respuesta de la tecnología Java a la programación CGI, y son como los *applets*, excepto que se ejecutan en el servidor en vez de en el ordenador del usuario.

Los *servlets* se usan para manejar solicitudes y respuestas desde navegadores clientes y generar páginas web; son útiles en casos como:

- **La página web está basada en datos enviados por el usuario:** por ejemplo, las páginas de resultados de los motores de búsqueda se generan de esta forma.
- **Los datos cambian frecuentemente:** un informe sobre el tiempo o páginas de cabeceras de noticias podrían construir la página dinámicamente, quizás devolviendo una página previamente construida y actualizada.
- Las páginas web que usan información desde bases de datos corporativas u otras fuentes. Por ejemplo, para hacer una página web en una tienda *on-line* que liste los precios actuales y el número de artículos en disponibles.

Las principales ventajas de los *servlets* Java respecto a los CGI son su eficiencia, potencia y la posibilidad de ejecutar el mismo código en cualquier servidor de aplicaciones.

JSP (JavaServer Pages)

Las *JavaServer Pages* son otro tipo de programas que se ejecutan en un servidor de aplicaciones o un servidor web para ofrecer una funcionalidad compleja que difícilmente se podría conseguir con un CGI.

Son páginas de navegador en HTML o XML en las que se puede incluir código Java; esto permite realizar procesos complejos, salidas condicionales y comunicarse con otros objetos de nuestra aplicación.



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Bienvenido a tu tienda on-line</TITLE></HEAD>
<BODY>
<H1>Bienvenido a tu tienda on-line !</H1>
<SMALL>Hola,
<!-- Username es "New User" para los visitantes por primera vez -->
<% out.println(Utils.getUserNameFromCookie(request)); %>
Para registrarte, accede <A HREF="registro.html">aquí.</A></SMALL>
<P>
HTML normal para el resto de la página
</BODY></HTML>
```

Microsoft tiene una tecnología similar a las JSP en la que ha substituido el lenguaje Java por Visual Basic: es la conocida tecnología ASP (*Active Server Pages*).

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Estas son las ventajas más revelantes de las JSP:

- **ASP:** las ventajas de JSP vs ASP son dos: la parte dinámica está escrita en Java, no en Visual Basic, que es un lenguaje específico de Microsoft (un lenguaje más poderoso y fácil de usar); la segunda ventaja radica en la portabilidad a otros sistemas operativos y servidores web.
- **Servlets:** JSP no nos da nada que no pudiéramos hacer con un *servlet* aunque es mucho más conveniente escribir o modificar HTML normal que tener que escribir muchas sentencias en Java que generen HTML. Además, separando el formato del contenido podemos poner diferentes personas en diferentes tareas: nuestros expertos en diseño de páginas web pueden construir el HTML dejando espacio para que nuestros programadores de *servlets* inserten el contenido dinámico.
- **JavaScript:** JavaScript puede generar HTML dinámicamente en el cliente; aunque se trata de una capacidad útil, sólo maneja situaciones donde la información dinámica está basada en el entorno del cliente, por lo que no puede acceder a los recursos en el lado del servidor, como bases de datos, catálogos, información de precios, etc.

Web services y .NET

Las aplicaciones web actuales ya no son suficientes. El modelo actual de negocio electrónico no facilita la integración de las aplicaciones de Internet con el resto de *software* de las empresas. Si las compañías quieren extraer el máximo beneficio de Internet, los sitios web deben evolucionar. Este es el contexto en el que surgen los *web services*.

Los *web services* son componentes *software* que permiten a los usuarios usar aplicaciones de negocio que comparten datos con otros programas modulares vía Internet. Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos web estándar. El objetivo final es la creación de un directorio *on-line* de *web services*, similar al directorio de nombres de servidores (DNS), donde se publican los *web services* que las diferentes empresas ponen a disposición en Internet.

Aunque la idea de la programación modular no es nueva, el éxito de esta tecnología reside en que se basa en estándares conocidos en los que ya se tiene una gran confianza, como el XML. Además, el uso de los *web services* aporta ventajas significativas a las empresas. El principal objetivo que se logra es la interoperabilidad y la integración. Mediante los *web services* las empresas pueden compartir servicios *software* con sus clientes y sus socios de negocio. Esto ayudará a las compañías a escalar sus negocios, reduciendo el coste en desarrollo y mantenimiento de *software*, y sacando los productos al mercado con mayor rapidez. La integración de aplicaciones hará posible obtener la información demandada en tiempo real, acelerando el proceso de toma de decisiones.

La evolución de Internet hacia los *web services* mejorará los resultados globales de las empresas, reduciendo sus gastos y guiándolas hacia una mejora progresiva de la calidad. La adopción de esta tecnología por la industria es el primer paso hacia una economía global.



Buscador Google

Podemos acceder a este *web service* del buscador Google desde nuestras aplicaciones situadas en cualquier lugar de Internet utilizando los protocolos utilizados por los *web services*. De esta manera podemos integrar su potente buscador en nuestro sitio Internet.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Microsoft, por su parte, ha emprendido la iniciativa .NET. Se trata de una «plataforma de *software*» que se define como un ambiente donde pueden interactuar diversos componentes independientemente del lenguaje, esto es, en lugar de escribir componentes para una combinación *hardware*/sistema operativo será escrito para .NET (DotNet). Ahora bien, .NET es el nombre asignado por Microsoft a diversos *productos y servicios*, en este caso productos como VisualStudio.NET y Windows.NETServer, mientras los servicios incluyen Passport y Hailstorm que pretenden ofrecer una manera universal de acceder recursos en Internet.

Esta plataforma es una alternativa a la plataforma J2EE y tiene como principal inconveniente que es propiedad de una empresa (Microsoft), mientras que la primera es un estándar que permite la interacción entre aplicaciones independientemente de los vendedores de éstas.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Proceso de programación y desarrollo tecnológico – Bases de datos

Una base de datos es una colección de archivos interrelacionados creados con un sistema gestor de bases de datos (DBMS). El contenido de una base de datos engloba la información necesaria para representar una organización de tal manera que los datos estén disponibles para los usuarios. Los tres componentes principales de un sistema de base de datos son el *hardware*, el *software* DBMS y los datos a manejar.

▪ Acceso a bases de datos: SQL

El lenguaje que se utiliza para trabajar con los datos de las bases de datos es el lenguaje SQL.

La historia de SQL empieza en 1974 con la definición, por parte de Donald Chamberlin y de otras personas que trabajaban en los laboratorios de investigación de IBM, de un lenguaje para la especificación de las características de las bases de datos que adoptaban el modelo relacional. En el curso de la década de 1980, numerosas compañías como Oracle y Sybase comercializaron productos basados en SQL, que se convirtió así en el estándar industrial por lo que respecta a las bases de datos relacionales. En 1986 el ANSI adoptó SQL como estándar para los lenguajes relacionales, y en 1987 se transformó en estándar ISO.

En este apartado se incluye un breve extracto de las operaciones que pueden realizarse en las bases de datos relacionales. En este tipo de base de datos, los lenguajes de acceso a los datos se basan en el álgebra relacional o en el cálculo relacional (teoría próxima a la teoría matemática de conjuntos). Se obtienen relaciones a partir de relaciones y sirven tanto para actualizar datos como para extraer subconjuntos de datos. Es posible identificar las tablas o relaciones con conjuntos y aplicarles las conocidas operaciones de unión, intersección, producto cartesiano, diferencia, etc., que se aplican a los conjuntos.

Para crear las relaciones mencionadas anteriormente en SQL se debería escribir:

```
CREATE TABLE maquetador(
    Nombre VARCHAR(15) NOT NULL,
    Apellidos VARCHAR(15) NOT NULL,
    Telefono VARCHAR(15) NOT NULL PRIMARY KEY,
    Email VARCHAR(15) NULL,
    Calle VARCHAR(15) NULL,
    Población VARCHAR(15) NULL,
    Provincia VARCHAR(15) NULL,
    Pais CHAR(2) NULL
)
CREATE TABLE Pais(
    Siglas CHAR(2) NOT NULL PRIMARY KEY,
    Nombre VARCHAR(15) NOT NULL
)
```

En estas líneas de código se define el tipo de datos que se usará en cada atributo, cuáles son los atributos que forman la clave primaria en cada una de las tablas (PRIMARY KEY) y si se permite que el atributo pueda estar vacío (NULL) o no (NOT NULL).

Para introducir nuevos registros se usaría:

```
INSERT INTO Pais (Siglas, Nombre)
VALUES ("ES","España")
INSERT INTO Pais (Siglas, Nombre)
VALUES ("UK","Reino Unido")
INSERT INTO Maquetador (Nombre,Apellidos,Telefono)
VALUES ("Margarita","Suarez García","34892348128")
```

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Y una posible consulta sería:

```
SELECT Maquetador.Nombre, Maquetador.Apellidos, Pais.Nombre  
FROM Maquetador, Pais  
WHERE Maquetador.Pais = Pais.Siglas  
ORDER BY Maquetador.Pais, Maquetador.Apellidos
```

La cual nos devolvería una relación o tabla con tres atributos: nombre, apellidos y país de los maquetadores de nuestra base de datos. Asimismo, le indicamos que queremos el nombre completo del país haciendo coincidir las siglas de nuestras tabla «Maquetador» y «Pais» (WHERE). Además nos interesa que esta nueva tabla aparezca ordenada por países y, dentro de cada país, por apellidos (ORDER BY).

La consulta daría lugar a una tabla temporal que no se almacenaría en la base de datos, simplemente nos serviría para construir un listado. Si se prevé usar esta consulta habitualmente es posible crear una vista; esto consistiría en una pseudo tabla que se construiría, utilizando la consulta, cada vez que dicha vista se abriera. De este modo se separaría la estructura interna de tablas de la base de datos de las vistas, que pueden ser más apropiadas para el tratamiento por parte de usuarios y programas.

El lenguaje SQL permite un gran número de operaciones sobre los datos de la base de datos que no son objeto de mención en esta obra, pero que pueden ser de vuestro interés. En el mercado se encuentran diversas obras de referencia sobre este lenguaje. Hay que tener en cuenta las pequeñas diferencias entre los SQL de Oracle, MS-SQL, MySQL, PostgreSQL, etc., por lo que es importante consultar los manuales de dichos productos.

Muchas plataformas ofrecen interfaces visuales de SQL para facilitar la gestión de los datos, de forma básica, sin tener que recurrir al aprendizaje del lenguaje.

Una transacción es la ejecución de un programa (o parte de un programa) que realiza accesos de lectura y/o actualización de una base de datos. En caso de actualización, una transacción debe siempre completar sus actualizaciones o bien dejar la base de datos tal y como estaba antes de comenzar. Para ello, SQL provee las instrucciones:

TRANSACTION: señala el comienzo de la transacción.

COMMIT: se dan por buenos los cambios efectuados desde el inicio de la transacción.

ROLLBACK: se ha detectado un error de cualquier tipo y se desean anular todos los cambios hechos desde el principio de la transacción.



Actividad

Señala en cuáles de los siguientes escenarios sería bueno introducir el concepto de transacción:

1. Visualizar las horas de llegada de los aviones al aeropuerto más cercano a vuestra ciudad en la próxima hora.
2. Reservar por Internet dos butacas en el teatro para la función del sábado.
3. Obtener una lista de las últimas transacciones de vuestra cuenta corriente.
4. Sacar dinero en un cajero automático.

Respuesta:

1. No hace falta para leer información.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

2. Sí, pues es preciso bloquear las butacas hasta que la venta no se realiza completamente, en caso contrario se podrían vender las mismas butacas a diferentes personas.
3. No es necesario pues es un volcado de información.
4. Sí, pues el cajero puede estropearse a mitad del proceso.

■ Motores de bases de datos

Hay muchos motores de bases de datos existentes en el mercado, cada uno con sus características específicas, aunque todos ellos ofrecen una funcionalidad similar mediante un lenguaje de acceso común: SQL. Así, dentro del mundo Linux y su filosofía de *software* gratuito, tenemos como principal representante a MySQL. De entre los gestores de BBDD comerciales multiplataforma encontramos a Oracle, DB2 o Sybase. Sobre plataforma Microsoft tenemos el Microsoft SQL Server. El programa Access es un gestor de base de datos, pero no está indicado para gestionar grandes volúmenes de información.



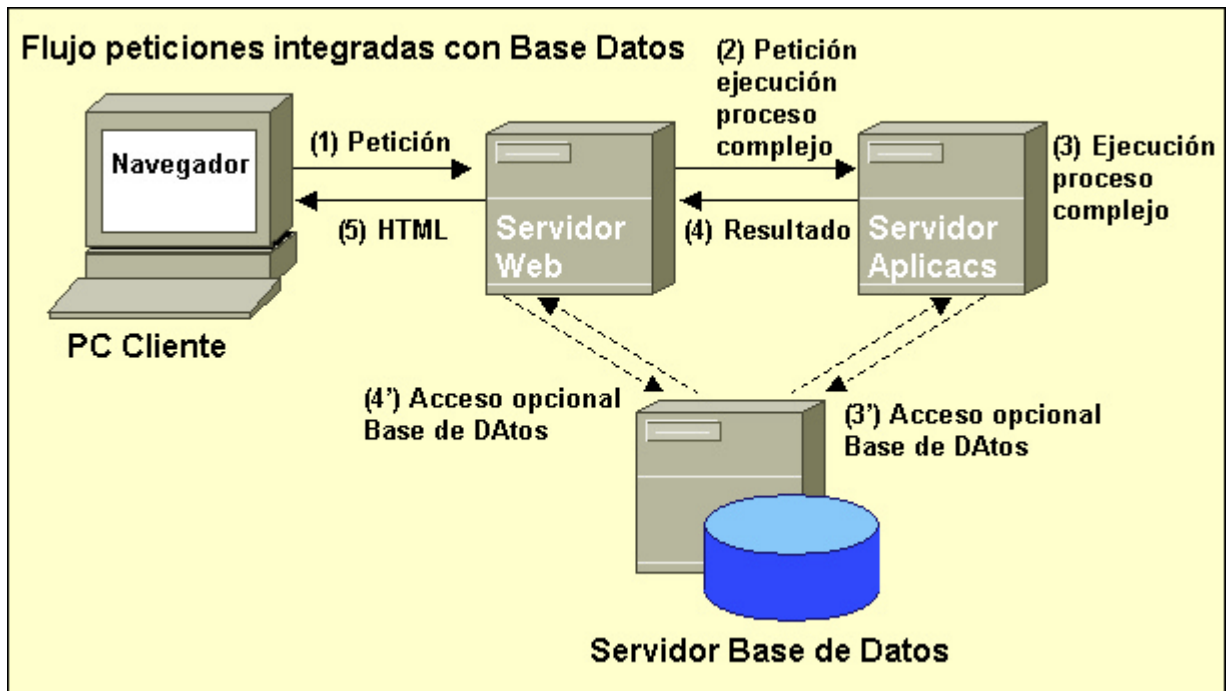
■ Integración web con bases de datos

En el mundo Internet las bases de datos juegan un papel muy importante puesto que mucha información publicada se encuentra almacenada en ellas, y muchas páginas web se construyen cuando el cliente accede a la misma. De este modo las modificaciones en los contenidos se realizan en dichas bases de datos sin tener que mantener contenidos estáticos y teniendo la seguridad de que la información mostrada será actualizada, ya que normalmente dichas bases de datos son el **repositorio** único de información.



Cuando estamos realizando una reserva de un vuelo desde la web, la estamos introduciendo en la base de datos Amadeus, que es el repositorio de información y reservas utilizado por cualquier agencia de viajes de todo el mundo. De esta forma nuestra plaza no podrá ser ocupada por otra agencia de viajes situada en cualquier punto del mundo, independientemente del mecanismo de acceso a la misma que utilice.

Posgrado en Edición Digital (on line y off line)
Desarrollo, integración y control de calidad



Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Integración multimedia

Esta es la fase del proyecto en que todas las tareas de producción convergen con el objetivo de integrar contenidos y funcionalidad en un solo producto. Iniciaremos la integración multimedia en el momento en que:

- Disponemos de una parte significativa de la funcionalidad de la aplicación.
- Damos por terminada la creación de contenidos textuales y multimedia.
- Nos disponemos a generar la primera versión del producto.

Dependiendo del tipo de producto que estemos desarrollando, las tareas y la planificación de esta fase variarán sustancialmente:

- **Proyectos de montaje:** son aquellos proyectos generalmente basados en herramientas de autor, en los que los contenidos se incorporan gradualmente en la aplicación, de forma que la fase de integración discurre de forma paralela a la del montaje de las pantallas o páginas de la aplicación. Los proyectos realizados con herramientas de autor, como Authorware o Director, entran dentro de este tipo.
- **Proyectos de desarrollo:** son proyectos que requieren un desarrollo informático específico que avanza de forma independiente de la creación de los contenidos. Al programa informático que da funcionalidad al producto lo denominaremos *motor*.

También el entorno de publicación condiciona las tareas de integración: los requisitos son sustancialmente diferentes si nos planteamos la publicación en un entorno *on-line* o si nuestro proyecto se publica en un soporte físico (CD-ROM, DVD).

Precisamente debido a la gran variabilidad de esta fase del proyecto, nos centraremos en las diversas tareas de integración, que incluyen:

- Transformación de los contenidos al formato requerido por el motor.
- Distribución de los archivos de contenido en la forma óptima para su acceso.
- Preparación de *software* auxiliar.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Integración multimedia – Transformación de contenidos

A lo largo de la fase de producción se han ido desarrollando los diferentes contenidos que formarán parte de nuestro producto. En este momento nos disponemos a hacer que estos contenidos (texto, imágenes, audiovisuales) estén disponibles y accesibles para el motor de nuestra aplicación multimedia de acuerdo con la estructura, funcionalidad y presentación previstas en la fase de diseño.

Al plantearnos los procesos de transformación debemos tener muy claro cuál va a ser la versión «maestra» de los contenidos que manejamos.



Supongamos que nuestro producto requiere que todo el texto esté disponible en una gran base de datos y que al finalizar la producción disponemos de una gran colección de archivos de procesador de texto, almacenados en una carpeta de nuestro servidor.

El equipo de desarrollo ya ha previsto un proceso para volcar todos los textos en la base de datos del producto. Sin embargo, antes de hacerlo se plantea la siguiente cuestión: en caso de que más adelante sea necesario corregir o modificar una parte específica del texto de la obra, ¿dónde debe efectuarse la modificación, en los documentos de texto o en la base de datos?

En este momento tenemos que decidir cuál de los dos repositorios de texto va a ser el «maestro», para lo cual debemos considerar varios aspectos:

La dificultad para acceder, localizar y modificar el contenido. Puede darse el caso de que la base de datos que requiere nuestro producto tenga una estructura muy compleja, que dificulte la edición de su contenido. En este caso la introducción de modificaciones sería muy costosa.

La dificultad para volcar el contenido en la base de datos. El formato de los archivos de texto condiciona el proceso para volcar su contenido sobre la base de datos del producto. El proceso de volcado puede ser lento y caro, de manera que si un editor modifica uno o más archivos de texto deberá realizar nuevamente el proceso, con el consiguiente impacto en el presupuesto y calendario del proyecto.

Otro aspecto a valorar a la hora de decidir sobre el repositorio «maestro» es la posibilidad de reutilización futura de los contenidos.

Aunque en el ejemplo nos hemos referido a documentos de texto, el mismo criterio debe aplicarse a las imágenes y a los elementos audiovisuales, que suelen precisar un proceso de compresión previo a su integración y al que nos referiremos más adelante.

La solución ideal para el mantenimiento de los contenidos consiste en disponer de una herramienta de gestión de contenidos que responda a las necesidades de producción del equipo editorial, y que a la vez pueda comunicarse directamente con la base de datos del producto.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

▪ Conversión de textos

Diversas acciones se aplican a la conversión de textos para su integración en nuestra aplicación:

- **Transformación de bases de datos:** como se ilustraba en el ejemplo del apartado anterior, es bastante frecuente la necesidad de disponer del texto estructurado en una base de datos accesible por el motor.
- **Creación de contenidos auxiliares:** otra de las tareas que suele ser necesaria es la generación de nuevos contenidos a partir del propio texto. Un ejemplo podría ser la creación de un índice de palabras para que la búsqueda de texto sea rápida y eficiente.
- **Codificación:** salvar el vacío existente entre el sistema de edición y las posibilidades de presentación que previmos para nuestro producto puede requerir cambios en la codificación del texto. Por ejemplo, cuando queremos convertir una expresión en nuestro procesador de textos, como «H₂O» en su equivalente en HTML «H₂O».
- **Transformación XML/XSLT:** como ya dijimos en introducción del lenguaje XML, una de las grandes ventajas de su uso es el hecho de que el contenido está separado de su presentación. El uso de XSLT permite una rápida transformación del contenido textual a otros formatos como los que acabamos de mencionar. No obstante, es importante recordar que la programación de una XSLT requiere conocimientos técnicos específicos.

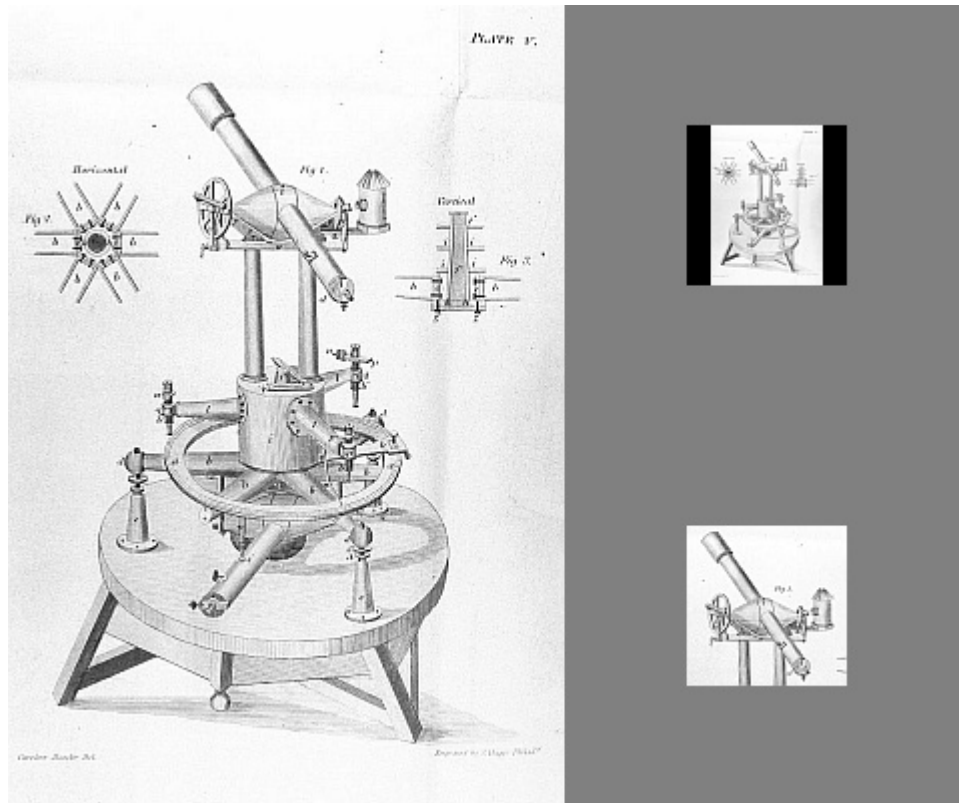
▪ Edición de imágenes

Después del proceso de producción, las imágenes que formarán parte del producto estarán correctamente seleccionadas y ajustadas para su publicación. No obstante, con frecuencia es necesario un último proceso para adecuar el formato y configuración de estas imágenes de forma que el motor de la aplicación pueda presentarlas correctamente y de la forma más efectiva. Este proceso suele realizarse de forma automatizada, lo cual permite un importante ahorro de tiempo y recursos. Algunos de los procesos más frecuentes son:

- **Cambio de nombre de los archivos:** es posible que durante la producción hayamos trabajado con nombres de archivo sin normalizar. Así, por ejemplo, antes de integrar los ficheros a nuestro proyecto, debemos renombrar estos archivos de forma que el motor de nuestra aplicación pueda localizarlos y acceder a ellos.
- **Creación de nuevas imágenes de tamaño reducido:** si hemos trabajado con imágenes de alta resolución, necesitaremos adecuarlas al tamaño de la pantalla para la que hemos diseñado nuestro producto. También es frecuente utilizar imágenes reducidas que den una representación de la imagen real. Estas miniaturas, que en la jerga técnica se denominan *thumbnail* (literalmente, uña de pulgar), recuerdan bastante a los contactos de negativos que se realizan en fotografía.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad



Al generar la versión miniatura de una imagen (*thumbnail*), puede ser interesante modificar el encuadre para destacar una parte concreta de la imagen. En este ejemplo podéis observar la diferencia entre una simple reducción realizada mediante un proceso automático, y un *thumbnail* que se centra en la parte más significativa de la ilustración. Aunque la producción del segundo puede automatizarse en parte, es imprescindible que una persona seleccione el fragmento adecuado sobre la imagen original.

National Oceanic & Atmospheric Administration (NOAA)

- **Creación de una versión con menor profundidad de color:** tal como vimos producción, la profundidad de color en un archivo de imagen determina en gran medida su tamaño y, en consecuencia, la velocidad con que el ordenador puede procesarla y mostrarla.
- **Conversión de formato de archivo:** podemos convertir archivos en formato vectorial a formato de mapa de bits. Este tipo de conversión suele realizarse cuando el motor de nuestra aplicación no nos permite mostrar imágenes en formato vectorial. Por ejemplo: convertir mapas cartográficos (que suelen estar en archivos vectoriales) a formato de mapa de bits.
- **Creación de una versión comprimida del archivo de imagen:** uno de los formatos más utilizados para las imágenes fotográficas en producción multimedia es JPEG. La compresión que se aplica para la obtención de un archivo JPEG produce una pérdida de calidad de la imagen. Aunque se trate de un sacrificio mínimo de la calidad de la fotografía, es conveniente conservar siempre una versión «maestra» de la imagen sin pérdidas, de forma que si en un futuro fuera necesario modificarla se pueda hacer sobre el archivo original, que no ha sufrido pérdidas por compresión.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Todas estas transformaciones pueden realizarse mediante uno o varios procesos automáticos usando herramientas especializadas como deBabelizer, Graphic Converter (para Apple Macintosh) e incluso herramientas simples y gratuitas como IrfanView.

■ Compresión de audiovisuales

Los audiovisuales (fragmentos sonoros, vídeos y animaciones) que componen nuestro producto suelen precisar un tratamiento específico para su publicación: se trata del proceso de compresión.

Como ya se comentó en el apartado compresión de vídeo, la compresión permite que los audiovisuales ocupen menos espacio en disco y facilitan la reproducción en el ordenador del usuario.

El proceso de compresión de audiovisuales se suele realizar durante la fase de integración, después de un análisis a fondo de las especificaciones técnicas del producto y de algunas pruebas que nos conducirán a los resultados que pretendemos obtener. Este análisis previo nos permitirá establecer los parámetros óptimos de compresión, aquellos que nos dan una mejor calidad de imagen en la reproducción con un mínimo de pérdida de calidad y un tamaño de archivo ajustado. Precisamente el hecho de establecer estos parámetros a la vista del material existente, permitirá que los audiovisuales de nuestra obra tengan un nivel de calidad homogéneo.

Una vez decididos los parámetros de compresión, iniciamos el procesamiento de los archivos de audiovisuales. Hemos de tener en cuenta que esta tarea requiere un ordenador bien equipado (en memoria RAM, disco duro y procesador) y debemos contar con un periodo de tiempo razonable.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Integración multimedia – Distribución de archivos multimedia

Para poder mostrar el contenido multimedia, el motor de nuestro producto requiere que los archivos multimedia estén ubicados en determinadas carpetas dentro del sistema de archivos del disco que alojará la aplicación.

Esta estructura de carpetas se habrá diseñado en la fase de desarrollo del motor de la aplicación y, en el momento de la integración, debemos ocuparnos de que cada uno de los archivos ocupe su lugar.

Una vez los archivos estén situados en la ubicación prevista, es conveniente que el equipo de desarrollo efectúe las comprobaciones necesarias para asegurarse de que todas las llamadas a archivos externos pueden resolverse (es decir, que no hay llamadas a archivos inexistentes) y que todos los archivos incluidos sean llamados desde alguna parte de la aplicación (que no hayan archivos sobrantes o falten llamadas). Esta comprobación nos permitirá también corregir errores en los enlaces entre los diferentes componentes.

Un producto en CD-ROM o DVD puede contener en sus carpetas una gran cantidad de texto y elementos multimedia, accesibles al usuario desde el sistema operativo. Con el fin de prevenir la manipulación de estos archivos en formato «abierto», es bastante común empaquetar textos, imágenes e incluso elementos audiovisuales en grandes archivos. Esta solución suele facilitar el acceso del motor informático a los archivos y, en consecuencia, se mejora la velocidad de acceso al disco.



Con frecuencia el desarrollo del motor de *software* requiere disponer de contenidos multimedia antes de que estén listos para su incorporación. En estos casos, es muy útil generar archivos con contenido ficticio, pero que permitan probar la funcionalidad y la navegación de nuestro producto.

Estos archivos de contenido ficticio se denominan archivos *dummy*, palabra inglesa que significa «maqueta».

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Integración multimedia – Distribución de archivos multimedia

Como señalábamos al principio, la fase de integración tiene por objetivo la obtención de la primera versión funcional de nuestro producto. Por lo tanto, es en esta fase cuando debemos ocuparnos de todo lo necesario para que el contenido del disco (o del sitio web) sea accesible desde cualquier ordenador.



Actividad

Piensa en el programa de instalación que suele incluir la mayoría de productos de *software* y prepara una lista de todos los contenidos (texto, imágenes, etc.) necesarios para una aplicación de este tipo.

■ Instalador

Este requisito requiere que el equipo de desarrollo prepare el instalador de la aplicación, es decir, el programa que prepara el ordenador del usuario para ejecutarla. El programa de instalación copia los archivos precisos en el disco duro y modifica los archivos de configuración para que el motor de *software* funcione de manera óptima en el ordenador del usuario. Algunos de los programas que permiten crear instaladores son:

- Installer VISE: <http://www.mindvision.com/>
- InstallShield: <http://www.installshield.com/>
- Wise for Windows Installer: <http://www.wise.com/wfwi.asp>

■ Componentes de *software* de terceras partes

En este momento también debemos incluir en el disco del producto el *software* de soporte necesario para la correcta visualización del contenido de nuestra aplicación. Este *software* puede ser:

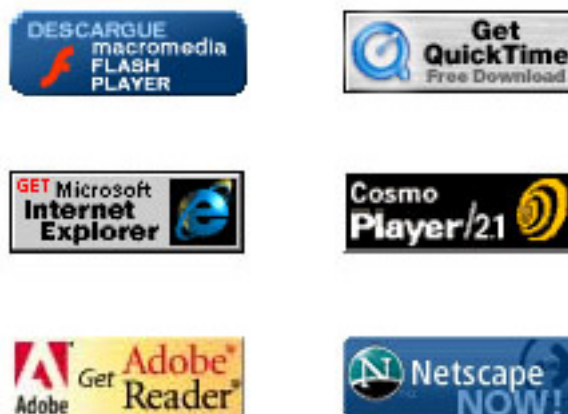
- Aplicaciones, por ejemplo los navegadores, Netscape Navigator o Microsoft Internet Explorer; Acrobat Reader, una herramienta para visualizar e imprimir documentos, etc.
- Reproductor de elementos multimedia, como por ejemplo QuickTime, Windows Media Player, RealPlayer, etc.
- *Plug-in*, componente de *software* que aumenta la funcionalidad del navegador como, por ejemplo, el Macromedia Shockwave Flash.

Casi todos estos componentes son de distribución gratuita, lo cual no significa que se puedan distribuir sin límites. Para cada uno de ellos debemos consultar las condiciones del fabricante, que normalmente requiere la identificación del editor, la inclusión de un logotipo en el envase y el contenido del producto y, en algunos casos, una copia del producto en su versión final.

En el caso de que nuestro producto se vaya a publicar *on-line* es mucho más sencillo proporcionar al usuario los componentes necesarios para visualizar el contenido: simplemente dispondremos un enlace hacia la página del sitio web del fabricante en la que se puede descargar el componente en cuestión.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad



Estas son algunas de las marcas de software auxiliar que con frecuencia necesitaremos para nuestros productos.

■ Software anticopia

Existen algunos componentes de *software* que tienen por objetivo impedir la copia o el acceso a personas no autorizadas. Normalmente este tipo de protección debe implementarse en la fase de integración multimedia dentro de procesos de transformación de contenidos como los descritos en el punto 7.3.1.



SealedMedia es un *software* que permite el control de acceso y manipulación para contenidos *on-line*. En su página web (<http://www.sealedmedia.com/products/default.asp>), se describe cómo funcionan este tipo de utilidades y su aplicación en la edición digital.

Existe otro tipo de protección que tiene por objetivo impedir la copia ilegal del disco. Para aplicar un sistema de este tipo a nuestro producto deberemos considerarlo en la fase de publicación. Hay múltiples opciones, como por ejemplo:

- **Safedisc:** sistema basado en la aplicación de una firma digital sobre el CD-ROM.
- **Securerom:** sistema basado en la aplicación de un código electrónico en el CD/DVD-ROM.
- **CD-COPS II / DVD-COPS:** se basa en la medición de ciertas características del CD/DVD-ROM para establecer una marca y asegurar que no se aceptan copias. La protección se añade al ejecutable principal.
- **STARFORCE:** como el anterior, también se basa en la medición de ciertas características del CD/DVD-ROM para establecer una marca y asegurar que no se aceptan copias. La protección se añade al ejecutable principal.
- **CRYPTKEY:** basa en la utilización de un código de usuario (*site code*) y una clave de usuario (*site key*).

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

La integración multimedia consiste en preparar los contenidos y programas que forman nuestro producto para su publicación. Debemos transformar los archivos para darles el formato, denominación y ubicación definitivos dentro de nuestra aplicación. También prepararemos los módulos de *software* necesarios para que el producto pueda ser instalado y usado por nuestros usuarios.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Control de calidad

Como ya describimos en el módulo 4, la realización de un proyecto requiere gestionar cuatro aspectos: el coste, el tiempo, el riesgo y la calidad. En este apartado nos referiremos a este último aspecto.

Generalmente, los proyectos multimedia se organizan de manera que los diversos componentes se desarrollan de manera independiente y simultánea en el tiempo. Por ejemplo, mientras el equipo editorial trabaja en la producción del texto, el equipo audiovisual se ocupa de las animaciones que lo ilustrarán y el equipo de desarrollo trabaja en la programación del motor de software.

Precisamente debido a este modelo de desarrollo en paralelo, debemos ser muy conscientes de la importancia del control de la calidad en todos los procesos que se llevan a cabo a lo largo del proyecto.

Cada uno de los procesos que se realizan durante las fases de producción de los contenidos y desarrollo del motor de la aplicación contempla sus propios mecanismos para el control de calidad: la validación y corrección de textos, la depuración de las rutinas de programa, la revisión de audiovisuales, etc.

Estos controles de calidad parciales son de gran importancia y deben ser observados rigurosamente ya que, de esta forma, limitaremos al máximo el esfuerzo para el control de calidad de las versiones finales.

En el momento en que todos los componentes convergen en la fase de integración multimedia y obtenemos las primeras versiones de nuestra obra, deberemos asegurarnos que el producto cumple con todos los requisitos: que funciona de acuerdo con las especificaciones técnicas y que los contenidos y la funcionalidad se ajustan a lo previsto en el documento de diseño. Con independencia de las revisiones intermedias, el control de calidad final del producto es imprescindible.

En este apartado trataremos con mayor profundidad la fase de control de calidad que se efectúa al final de la fase de desarrollo y que se basa esencialmente en pruebas (test) sobre versiones más o menos funcionales de nuestra aplicación.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Control de calidad – Tipos de test

Distinguimos tres tipos de test de calidad de un producto multimedia:

▪ Test de usuario

El objetivo del test de usuario es identificar los aspectos relacionados con la usabilidad (navegación e interfaz de usuario) y la presentación del producto que pueden ser mejorados. También pueden ser de utilidad para poner a prueba la estructura de contenidos de la obra.

Se realiza durante la fase de diseño y consiste en poner a prueba de usuarios-tipo (personas que cumplen las características del público al que nos dirigimos) un prototipo de la aplicación que estamos desarrollando.

El test de usuario es muy importante para conseguir un buen diseño de producto. Disponer del tiempo y los recursos necesarios para poner a prueba el prototipo y adecuarlo a los requisitos de nuestros usuarios es básico para obtener un producto de calidad. No obstante, esta tarea suele ser difícil de “vender” a nuestros clientes, que no siempre son conscientes de la complejidad del diseño multimedia. Este artículo (en inglés) (<http://www.useit.com/alertbox/980503.html>) trata de la importancia del test de usabilidad y da algunos consejos para llevarlo a cabo con un mínimo impacto en el coste.



Actividad

Plantea la tarea de test de usuario para tu proyecto: prepara una lista de las acciones a realizar, las personas que van a intervenir y un cálculo preliminar del coste y la duración de la tarea. Después prepara una lista con los argumentos que expondrás a tu cliente para convencerle de la necesidad de realizar el test de usuario.

▪ Test funcional

El test funcional (o de funcionalidad) tiene como objetivo garantizar que el producto ofrece la funcionalidad prevista en la fase de diseño y que cumple con los requisitos técnicos previstos en la especificación.

Puede realizarse sobre versiones preliminares (alfa) antes de la fase de integración o bien después de las primeras versiones del producto (beta), cuando se dispone de todos los contenidos y la funcionalidad de la aplicación.

▪ Test de contenidos

Consiste en revisar la adecuación y la corrección estilística y formal de los contenidos de la obra.

Este tipo de revisión debe efectuarse a lo largo del proceso de producción: corrección de textos, edición de imágenes, etc., y si bien es cierto que al finalizar la fase de producción deberíamos contar con un contenido correcto y listo para su publicación, es frecuente que algunos aspectos –como la presentación o la vinculación entre diferentes unidades de contenido– sólo puedan ser identificadas en la versión final del producto.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Control de calidad – Planificación de la fase de control de calidad

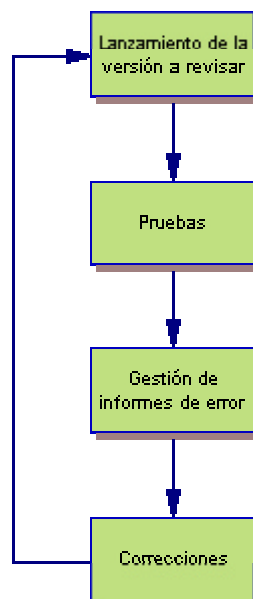
En un proyecto multimedia, la fase de control de calidad debe contar con el plazo y los recursos suficientes para garantizar que las características del producto se ajustan a los requisitos de nuestro cliente. El esfuerzo a reservar para esta tarea será proporcional a la complejidad editorial y técnica del producto.

■ Tareas

Antes de empezar:

- Planificar las fases en que se hará la revisión, de acuerdo con las tareas de desarrollo informático. Esto supone establecer las fechas en que se lanzarán las diferentes versiones de test. Dependiendo de la complejidad del producto y del estado de la versión a revisar, puede ser aconsejable organizar un test por partes, así nos aseguramos que el esfuerzo dedicado a la revisión se aprovecha al máximo.
- Diseñar un formulario para informes de error que permita anotar con facilidad las incidencias en el producto.
- Asignar a una persona la responsabilidad de gestionar las incidencias: interpretar los informes de error, establecer una prioridad en su solución y hacer el seguimiento hasta su corrección.

Durante la fase de revisión se repetirá el siguiente ciclo:



El ciclo de test:

1. Cada una de las versiones del producto a revisar debe ir acompañada de la lista de errores conocidos, es decir, de todas aquellas funciones que los desarrolladores saben que no funcionan con normalidad. Después de la primera versión a revisar, se adjuntará también una lista de los errores existentes en la versión anterior y corregidos en la actual.
2. Los testadores a cargo de la revisión probarán el producto con dos objetivos:

Comprobar el funcionamiento de las últimas correcciones aplicadas (sólo después de la primera versión).

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Describir cada nueva incidencia en el informe de errores.

3. El responsable de gestión de las incidencias consolidará los informes de los testadores, asignará a cada incidencia una prioridad y encargará su resolución a la persona correspondiente en el equipo de producción. Después del primer ciclo de revisión, también descartará aquellas incidencias que fueron resueltas, eliminándolas del ciclo de test.

Una vez concluida la fase de revisión por parte de los testadores y corregidas las incidencias detectadas, se producirá una versión a revisar, iniciando de nuevo el ciclo en el punto 1.

■ Calendario

A la hora de planificar la fase de control de calidad debemos saber en qué momento dispondremos de versiones para probar y en qué estado se encontrará cada una. A partir de esta previsión, debemos establecer un calendario detallado que contemple el tiempo que vamos a dedicar al test de cada lanzamiento y a la resolución de las incidencias que se produzcan antes de reiniciar el ciclo de test.

Las diversas versiones de la aplicación a testar reciben nombres específicos en función de los siguientes criterios:

Versiones alfa: son las primeras versiones del producto, que pueden carecer de ciertos aspectos de la funcionalidad y de partes específicas del contenido. El test de las versiones alfa debe hacerse con sumo cuidado ya que, por tratarse de productos «a medio hacer», suele darse un gran número de incidencias a tratar.

Versiones beta: suelen denominarse así aquellas versiones que incluyen toda la funcionalidad y contenidos de la aplicación, aunque en un estado preliminar (en función del grado de avance en la tarea de control de calidad).

En el ámbito del desarrollo de *software* la denominación de las versiones alfa y beta viene acompañada de una cifra que indica el grado de fiabilidad de la copia. Esta numeración se expresa en dos cifras separadas con un punto, iniciándose en el 1 (1.0, 1.1, 2.0...) de manera que una cifra mayor indica un mayor avance. Suele denominarse alfa 1.0 a la primera versión que va a ser sometida a test. A partir de ésta, cada nueva versión se diferenciará por una cifra mayor, por ejemplo 2.0 para una versión que incorpora grandes cambios con respecto a su predecesora o 2.1 (o 2.2, 2.3...) para las versiones que incorporan sólo pequeñas mejoras.

Release candidate: la fase de test concluirá con la obtención de una versión presuntamente libre de errores sobre la que efectuaremos la última iteración prevista del ciclo de test. En caso de que no aparezca ninguna incidencia, ésta será la versión final del producto. Precisamente, debido a la presunción de que ésta será la última versión a revisar, se le considera candidata, y la denominamos con el término *release candidate*.

Versión 1.0: la primera versión definitiva de la aplicación, la que entregaremos a nuestro cliente, recibirá la denominación 1.0. No obstante, también los productos terminados pueden recibir una denominación numérica que represente su grado de evolución.



Muchas de las aplicaciones que usamos en nuestros PC tienen la información relativa a su versión en uno de los menús de la parte superior de la pantalla. Por ejemplo, puedes ver el número de versión de tu navegador de Internet consultando la sección «Acerca de...» dentro del menú ayuda.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

▪ Recursos

Para llevar a cabo nuestra tarea, precisaremos dos tipos de recursos: los recursos humanos y los recursos materiales.

Recursos humanos

Responsable del control de calidad: será la persona responsable de recoger las incidencias del test, categorizarlas, priorizarlas y organizar su resolución. Debe prestar especial atención a que todos los informes de error se interpreten adecuadamente, a qué personas corresponde resolver las incidencias y cuál es su disponibilidad, y a la coordinación del trabajo de los testadores. En proyectos pequeños el responsable del proyecto puede asumir el rol de controlar la fase de test.

Testadores: son las personas que van a trabajar con la aplicación con el objetivo de localizar defectos. Es interesante que provengan de fuera del proyecto para evitar que tengan una visión sesgada del producto, que podría evitar las zonas «conflictivas». Además, conviene que tengan un perfil lo más parecido posible al de los usuarios a los que nos dirigimos para, así, aseguramos que el test se haga con interés y que se prueben las partes más significativas de la aplicación. En proyectos complejos puede ser necesario que —al menos para las primeras versiones— dispongan de cierto nivel técnico, de manera que puedan resolver las dificultades de funcionamiento que puedan surgir.

Equipo de desarrollo: el equipo de programadores que ha participado en el desarrollo técnico participará en la corrección de los defectos que surjan. Es interesante que estén al corriente de la evolución del test ya que, como «padres de la criatura», pueden identificar con facilidad muchos de los problemas que se detecten.

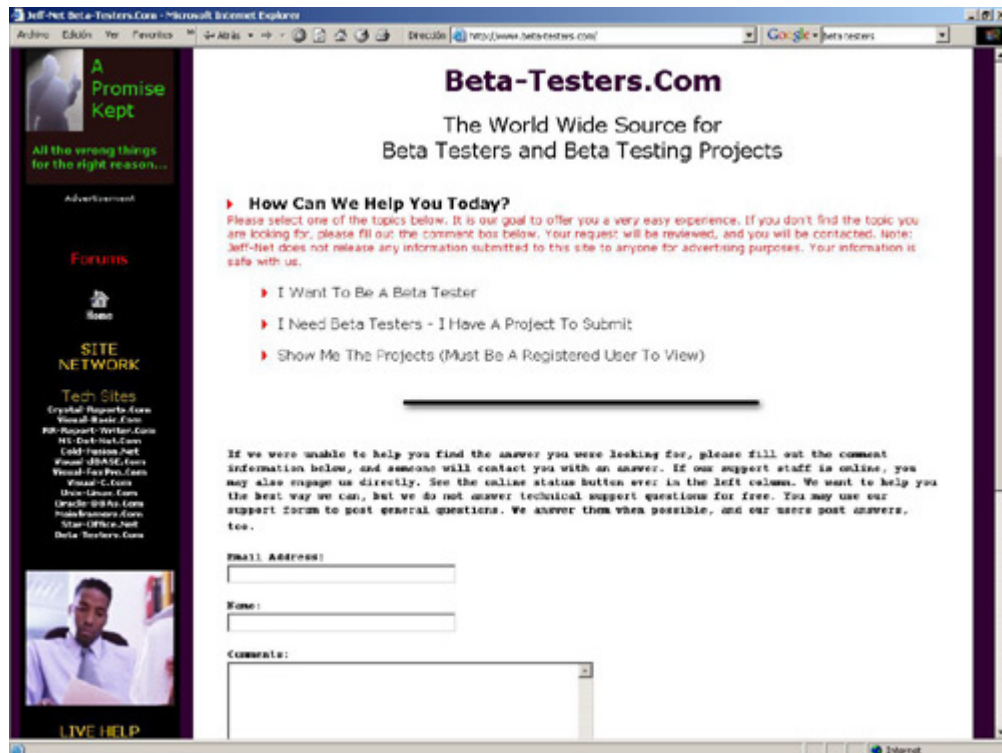
Jefe de proyecto: como decíamos al principio, la calidad es uno de los factores clave en la gestión de un proyecto. Por esta razón es fundamental la participación del jefe de proyecto en esta fase de control de calidad. Además del seguimiento de las tareas, su responsabilidad es la de asegurarse de que el producto cumple con los requisitos definidos durante la fase de diseño y gestionar cualquier incidencia que pueda tener impacto sobre los mismos.

Recursos materiales

Formarán el banco de pruebas de nuestro producto y, fundamentalmente, consisten en una selección de ordenadores con diversas configuraciones de *hardware* y *software*. Cuanto más amplio y diverso sea el banco de pruebas, mayor será la fiabilidad de nuestro test. Para cumplir con este requisito, algunas empresas de *software* reclutan voluntarios ajenos a la empresa a fin de probar versiones preliminares de sus productos. A las personas que realizan este tipo de test se les denomina *beta testers*.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

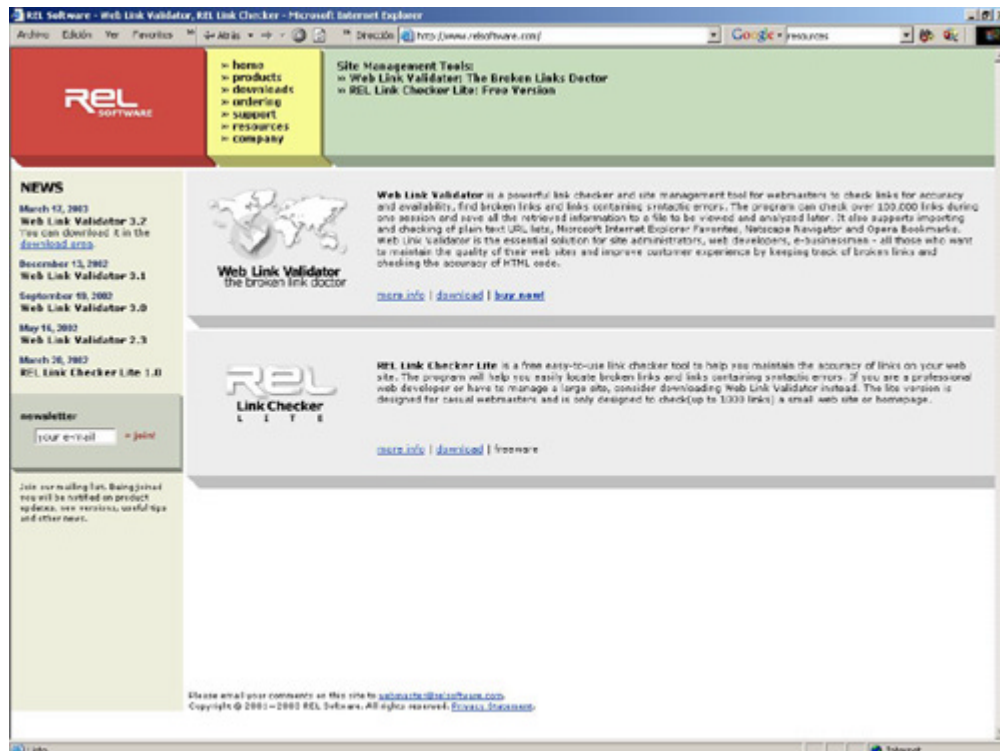


Beta-testers.com es un sitio web dedicado a poner en contacto testadores y productores de software.

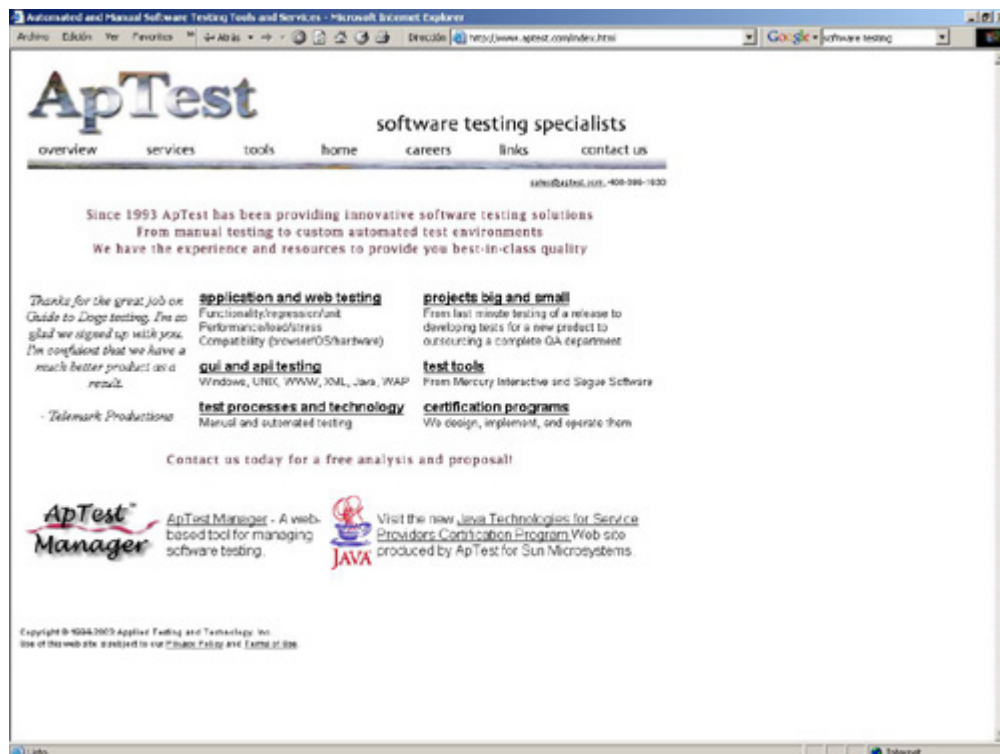
Existen empresas especializadas en el control de calidad de *software* que conocen a fondo la diversidad de configuraciones que puebla el mercado de ordenadores personales. Estas empresas pueden sernos útiles en el caso de que nuestro producto se base en un desarrollo técnico muy complejo en el que el riesgo de errores de tipo técnico es muy alto. También existen herramientas informáticas como Web Link Validator de Rel Software para el control de calidad que ponen a prueba la funcionalidad de nuestro producto sin la participación de un testador

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad



Página web de Rel Software



ApTest es una empresa especializada en soluciones para el test de software.

Este tipo de recursos (empresas o herramientas) sólo puede sernos útil en el test de funcionalidad, ya que nunca podrá detectar deficiencias en el diseño o en los contenidos de nuestro producto.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

La correcta planificación de la tarea de control de calidad de nuestro producto es esencial. Sólo si prevemos la disponibilidad de recursos humanos y materiales, así como un plazo de ejecución suficiente, podremos asegurarnos que el producto que vamos a lanzar responda a los objetivos iniciales. Al igual que ocurre con el test de usuario durante el diseño, a veces es difícil transmitir la importancia del control de calidad a nuestro cliente.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Control de calidad – Realizar y gestionar el control de calidad

■ Estrategias

La manera de llevar a cabo la fase de test varía enormemente en función del tipo de proyecto de que se trate. Existen diversos condicionantes que influyen en la forma de realizar el test del producto:

Tipo de producto: en general, la función del producto determina la forma de llevar a cabo el control de calidad. Por ejemplo, siendo ambos productos multimedia, el test de un videojuego basado en una aventura gráfica será mucho más complejo de testar que el de una presentación de negocios. Mientras en el primero deberemos esforzarnos por comprobar que la funcionalidad es correcta cualquiera que sea el recorrido que realice el usuario, en el caso de una presentación corporativa, con una estructura de navegación mucho más simple, deberemos dedicar más esfuerzo en la corrección de los textos y la adecuación de la presentación.

Plataforma: también la plataforma para la que estamos produciendo condiciona la forma de realizar el test. En el caso de que desarrollemos un producto *on-line*, además del control de calidad previo a la publicación que asegure la corrección de los contenidos y la funcionalidad del sitio web, deberemos establecer una evaluación periódica de la capacidad del servidor y del ancho de banda disponible para atender a todos los usuarios.

En los productos *off-line*, la plataforma para la que desarrollamos influye en el control de calidad a realizar. Si los requisitos del producto establecen que éste funcione para una versión específica de un sistema operativo, tendremos muchas menos dificultades que si desarrollamos un producto multiplataforma. Por ejemplo, si nuestro producto es un quiosco multimedia que va a ser implantado bajo un único modelo de ordenador, tendremos menos dificultades que si desarrollamos un CD-ROM multiplataforma.

Público objetivo: el público al que nos dirigimos condicionará también la forma de llevar a cabo el control de calidad. Por ejemplo, si desarrollamos un producto para niños de corta edad deberemos pensar en que los controles en pantalla sean fácilmente identificables y respondan a los intereses de este perfil de usuario. En cambio, un producto para ancianos tendrá unos requisitos de navegación y presentación claramente diferentes. Como ya mencionamos en el apartado de recursos, es importante contar con personas del público objetivo para efectuar un buen control de calidad.

Complejidad técnica: la componente tecnológica condiciona enormemente los requisitos del control de calidad. Los productos desarrollados con herramientas de autor serán en general más simples que aquellos que utilizan desarrollos propios combinando tecnologías diversas. El equipo de desarrollo conoce en cada caso los riesgos de los recursos que utiliza nuestro producto.



Actividad

Piensa en las principales acciones que requeriría el control de calidad de los siguientes productos:

un buscador de Internet,
una enciclopedia multimedia en CD-ROM,
un DVD-video sobre recetas de cocina japonesa.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

▪ Referentes

Durante el test de la aplicación debemos disponer en todo momento de los documentos en los que se establecen los objetivos del producto. Estos documentos serán nuestra referencia a la hora de detectar las deficiencias a corregir:

El prototipo de producto o maqueta: en la fase de diseño de producto se trata de un elemento clave durante el test de usuario. Nos será de gran utilidad para evaluar las percepciones del usuario respecto a la navegación y la presentación del producto. Ha de ser lo suficientemente flexible para que podamos modificarlo con facilidad y someterlo de nuevo a test. Lo mantendremos «vivo» hasta el cierre de la fase de diseño, momento en que el prototipo pasará a ser un referente más para el control de calidad.

Documentos de diseño: la especificación de los diseños de la información, interacción y presentación serán especialmente útiles a la hora de valorar el funcionamiento de la navegación, la usabilidad del producto o su presentación gráfica.

Lista de especificaciones técnicas: es esencial para conocer los requisitos de los ordenadores que formarán el banco de pruebas e imprescindible durante el test. Las especificaciones técnicas del producto establecen el sistema operativo para el que se ha diseñado la aplicación, la configuración mínima del ordenador del usuario (CPU, memoria RAM, tarjeta de gráficos, tarjeta de sonido), el *software* de soporte necesario (que siempre deberíamos incluir en el propio soporte en caso de los productos *off-line* o con un enlace a los sitios web).

Libro de estilo: el documento que recoge las directrices que deberán cumplir los contenidos del sitio. Será de utilidad ante las dudas que puedan surgir con respecto al texto, imágenes, audiovisuales que componen la obra.

Lista de errores conocidos: una lista con la descripción de las características no disponibles —o con errores ya identificados— de la versión en fase de pruebas.

De forma inevitable, durante la fase de control de calidad surgirán aspectos que no estaban previstos en los documentos de referencia y que nos obligarán a adoptar nuevos criterios o incluso modificar los existentes. El manejo de esta variabilidad es un elemento clave durante la fase de control de calidad: el responsable del proceso debe estar atento a cualquier propuesta de modificación o incorporación de nuevos criterios y valorar con detenimiento si su adopción es adecuada.

Cuando estamos en la fase de control de calidad final, un cambio en los criterios generales en que se basa el producto puede significar una nueva revisión para muchos elementos o funciones previamente aprobados.

Sólo cuando sea realmente imprescindible debemos modificar los criterios establecidos durante la fase de diseño de nuestro producto. Es precisamente durante el control de calidad cuando podremos apreciar la enorme importancia de disponer de un diseño sólido, a prueba de cualquier incidencia.

▪ Informes de error

Como ya mencionamos en el apartado de planificación, el ciclo de test requiere que las incidencias que se encuentren se documenten de la mejor forma posible. A tal efecto, se utilizan los llamados *informes de error* (o *bug reports* en la jerga del sector): para esta función

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

normalmente se diseñan unos formularios que facilitan al máximo la recopilación de información por parte de los testadores.

Normalmente un formulario de informe de error consta —como mínimo— de los siguientes apartados:

- Nombre de la persona que realiza el test.
- Configuración del ordenador: que puede ser una referencia a otro documento donde se describa con detalle.
- Versión de la aplicación: alfa, beta, *release candidate*.
- Clase: error, deficiencia, mejora.
- Tipo: contenido, funcionalidad, diseño.
- Pantalla en la que se produce: el lugar dentro de la aplicación en que se da el error, por ejemplo: página de resultados de búsqueda.
- Antecedentes: descripción de la situación desde la que se ha llegado a producir el error.
- ¿Se puede reproducir? Sí/no: se trata de determinar si se puede provocar de nuevo el error o si por el contrario no hay forma aparente de que suceda de nuevo.
- Descripción: texto descriptivo.

Puede ser chocante la inclusión de un campo «clase» que permite distinguir entre errores, deficiencias y mejoras. La intención es aprovechar el proceso de control de calidad al máximo, de manera que cualquier incidencia pueda detectarse en este momento, incluso si no se trata propiamente de un error, sino de una deficiencia o una propuesta de mejora. Posteriormente, el responsable de control de calidad determinará las acciones a emprender considerando también el contenido de este apartado.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Enciclopedia de la cocina		Informe de error	
Versión	Fecha	Testador	Ordenador
<input type="checkbox"/> Error <input type="checkbox"/> Deficiencia <input type="checkbox"/> Mejora		<input type="checkbox"/> Contenido <input type="checkbox"/> Funcionalidad <input type="checkbox"/> Diseño	
Pantalla	<input type="checkbox"/> Presentación <input type="checkbox"/> Menú <input type="checkbox"/> Receta <input type="checkbox"/> Búsqueda <input type="checkbox"/> Resultados <input type="checkbox"/> Otra	¿Se puede reproducir? <input type="checkbox"/> Sí <input type="checkbox"/> No ¿Cómo?	
Antecedentes			
Descripción			

Formulario de informe de error

Algunos de los apartados podrían adecuarse a la estructura del producto que estamos desarrollando, por ejemplo incluyendo en la sección «pantalla» una lista de las pantallas de la aplicación, de manera que evitemos denominaciones ambiguas; así facilitaremos todavía más la tarea de los testadores. Éstos deben tener ejemplares del formulario de informe de error siempre a mano y deben estar instruidos acerca de los criterios que se aplican a la hora de rellenarlos.

Gestión de los informes de error

El responsable de control de calidad se ocupará de gestionar los informes de error basándose en el siguiente proceso:

1. Consolidar los diversos informes de error generados por los testadores, identificando los duplicados y asegurando la correcta descripción de todos ellos.
2. Clasificar y priorizar la lista de incidencias de acuerdo con los responsables editoriales, de desarrollo y de diseño. Es necesario determinar qué incidencias se corregirán y en qué momento estará disponible la nueva versión para comprobar su implementación.
3. Asignar un responsable para la resolución de cada incidencia en función de su naturaleza: los programadores para la funcionalidad, los editores para los contenidos y la navegación, y los diseñadores gráficos para la presentación.
4. Hacer el seguimiento de la resolución de la incidencia.
5. Preparar el siguiente ciclo de test sobre la nueva versión de la aplicación en que los testadores deberán comprobar la implementación de las incidencias que se determinaron en su momento.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

En general, además, el responsable de test deberá mantener actualizada la lista de errores conocidos, uno de los documentos de referencia descrito más arriba.

A continuación detallamos algunas buenas prácticas a efectuar durante la fase de control de calidad en función del tipo de test al que se refieren:

Test de usuario

- No intervenir durante la sesión de test: cuando nuestro producto salga al mercado, evidentemente no estaremos detrás de cada usuario para explicarle lo que significa cada icono o la función de determinado botón. Es importante que el usuario reaccione de forma independiente. Sólo de esta manera podemos valorar su aportación.
- Registrar los comentarios y reacciones del usuario: tomar nota de todos sus comentarios por insignificantes que parezcan.
- En algunos casos, por ejemplo cuando trabajamos en un producto orientado a niños de corta edad, puede ser útil disponer de una cámara de vídeo que recoja las expresiones del usuario de forma simultánea a lo que aparece en la pantalla.

Test de funcionalidad

- Asegurarse de que se prueban todas las opciones y las combinaciones posibles: es necesario equivocarse, hacer lo imprevisto y llevar la aplicación a situaciones extremas.
- Probar bajo el máximo número posible de ordenadores: con las más diversas configuraciones de *hardware* y *software*.
- Realizar revisiones sobre versiones previas: de esta forma optimizamos el tiempo dedicado al control de calidad y podemos detectar posibles incidencias con la antelación necesaria.
- Realizar controles de calidad incluso *a posteriori*: especialmente en los productos *on-line*, cada vez que aparece una nueva versión del sistema operativo o del navegador, debemos asegurarnos que nuestra aplicación funciona correctamente.

Test de contenidos

- Revisar los contenidos durante el proceso de producción: de esta manera la única revisión que deberán sufrir será la validación de la presentación de los mismos.
 - Controlar y acotar al máximo las modificaciones a los criterios de diseño y al libro de estilo de la obra; de esta forma evitamos duplicar los procesos de revisión.
 - Evitar la tentación de modificar los contenidos que han pasado control de calidad con el fin de evitar errores y controlar la implementación de las nuevas versiones de contenido. Cualquier modificación debe preverse de acuerdo con el equipo de producción.
-

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Si bien el control de calidad debe ser una constante durante todo el proyecto, la complejidad tecnológica y editorial de un producto multimedia hace imprescindible una revisión a fondo de la funcionalidad y los contenidos de nuestra aplicación.

A la hora de realizar el plan de proyecto debemos considerar la importancia del control de calidad, reservando los recursos humanos y materiales necesarios para el mismo. La estrategia a seguir para llevar a cabo esta tarea dependerá del diseño, la tecnología y el público objetivo al que nos dirigimos.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Publicación

En los apartados siguientes haremos un repaso a las principales tareas a realizar en esta fase del proyecto. Algunas de ellas son independientes del medio en que vamos a publicar nuestro contenido. Otras, en cambio, dependen fundamentalmente del entorno al que nos dirigimos: *off-line* u *on-line*.

La publicación es la fase final de nuestro proyecto (a veces, incluso, puede quedar fuera del mismo). Se trata de la materialización de nuestro trabajo: si publicamos un producto *off-line*, obtendremos un soporte físico que requerirá un envase y una presentación adecuada; si se trata de un sitio web, publicar supondrá poner el resultado de nuestro trabajo en manos de los usuarios. Tanto en un caso como en el otro, publicar supone el fin último de todo proyecto editorial.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Publicación – Publicación *off-line*

La publicación en soportes físicos requiere la preparación de los materiales que forman nuestro producto para la fabricación de las copias que se distribuirán en el mercado, así como la preparación de los contenidos y materiales necesarios para el *packaging* del producto.

En primer lugar debemos seleccionar una empresa especializada en la producción de copias en soportes físicos. Algunos de los proveedores más conocidos son: MPO, GEMA OD y Grupo Condor. En sus páginas web también encontraréis información sobre los tipos de soporte que admiten, las especificaciones para el estuchado, etc.

▪ **Recopilación y entrega de los materiales**

En primer lugar debemos preparar el contenido de nuestro disco para la entrega a la empresa de replicado.

Normalmente este proceso se puede realizar a partir de:

- Un disco CD-R que nosotros mismos podemos crear si disponemos de un aparato de grabación.
- Un soporte de almacenamiento masivo, como un disco duro extraíble o una cinta de *back up*.

Cuando el volumen de material es muy grande —como por ejemplo los que se publican en DVD— nos veremos obligados a utilizar un soporte de almacenamiento masivo. En cualquier caso, debemos establecer el soporte y el formato de nuestro producto de acuerdo con el proveedor.

Además del material digital que conforma el producto propiamente dicho, también debemos tener preparados los materiales que lo acompañan, es decir, los archivos de preimpresión con el diseño del CD y de su estuche, que pueden contener:

- Serigrafía del disco: los discos compactos tienen la posibilidad de exhibir un diseño en una de sus dos caras, la que no incluye datos.
- Logotipos de terceras partes: como por ejemplo los proveedores de tecnología que, a cambio de autorizar la distribución gratuita de algunos de sus productos, suelen exigir que su logotipo figure en el estuche y el disco.
- Requisitos mínimos de instalación.
- Textos de promoción.
- Información del editor, teléfono de soporte al cliente.
- Copyright.
- Números de depósito legal y de ISBN.
- Manual de instalación.
- Catálogo de otros productos.
- Tarjetas de respuesta comercial.

El tipo de estuche más usual se denomina *jewel box*, el más habitual también en los CD de música. Este tipo de estuche incluye dos elementos impresos: el *inlay* (que es una hoja que

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

forma la contracubierta y los dos lomos de la caja), y el *booklet*, que es el librito que hace a la vez de portada y que normalmente se aprovecha para incluir el manual de instalación del producto.

Otros tipos de estuche van desde la simple funda de plástico transparente o de cartón, hasta los más sofisticados, como los que se utilizan en los DVD-video, que incluyen un *booklet* de mayor tamaño además de una gran portada.

El coste de la producción de un CD presentado en *jewel box* y con un *booklet* simple puede oscilar entre 0,5€ y 1€ por unidad, dependiendo fundamentalmente de la tirada, aunque también de las características del material impreso y la serigrafía del disco.

▪ Replicado y estuchado

La empresa de replicado preparará un primer ejemplar del disco definitivo, que debemos someter a algunas pruebas con el objetivo de asegurarnos que no contiene ningún error que pudiera invalidar todas las copias obtenidas.

Hecho esto, la empresa de replicado iniciará el proceso de producción de copias. Puesto que se trata de un proceso totalmente automatizado, no suele requerir mucho tiempo.

Con frecuencia los productos en CD o DVD se suelen presentar dentro de cajas de cartón más grandes que les dan una mayor presencia en las estanterías de los comercios. Debemos prever el diseño de estos contenedores junto con el del resto de materiales gráficos que acompañan al producto. Normalmente los envases son producidos por alguna empresa de artes gráficas especializada que asume también el proceso de manipulación del estuche. El proceso suele consistir en encajar el disco (o los discos) en el envase de cartón, cerrarlo y envolverlo con un material de plástico retráctil que hará las veces de precinto.



Producto multimedia con su estuche

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Publicación – Publicación en plataformas *on-line*

A diferencia de los productos *off-line*, la publicación en la red funciona de una forma completamente diferente. En primer lugar, no existe un soporte físico que fabricar, estuchar ni distribuir. Además tenemos control absoluto sobre el servidor (el ordenador que albergará los contenidos) y desarrollará la mayor parte de la funcionalidad de nuestra aplicación.

El desarrollo de los proyectos *on-line* suele llevarse a cabo en un servidor interno al que denominamos *servidor de producción*. Llegados a la fase de publicación, necesitaremos disponer de un servidor conectado a Internet que pueda albergar los contenidos y programas que configuran nuestro producto.

En los proyectos que tienen un fuerte componente tecnológico, debemos procurar que las diferencias de configuración entre el servidor de producción y el de publicación sean mínimas. Sólo de esta forma podemos estar seguros de que no tendremos problemas a la hora de publicar.

Llegado el momento del lanzamiento de la obra, transferiremos los archivos y programas residentes en el servidor de producción al servidor de Internet y, a continuación, deberemos efectuar algunas pruebas a fin de asegurarnos que el contenido de nuestro sitio web es accesible desde cualquier punto de la Red.

▪ **Determinar el servidor y el ancho de banda**

Un aspecto crítico en los proyectos de publicación *on-line* es determinar las características que debe cumplir el servidor de Internet y el ancho de banda de su conexión para que éste sea capaz de manejar todas las peticiones de nuestros usuarios. Los factores principales a tener en cuenta son los siguientes:

- En el servidor: la capacidad de la CPU, el tamaño de la memoria RAM y el tamaño disponible en el disco.
- En cuanto al ancho de banda: la dimensión correcta dependerá no sólo del número de consultas que precisemos atender, sino también del tipo de contenido que estemos publicando. En caso de que dispongamos de una gran cantidad de audiovisuales, necesitaremos un ancho de banda mucho mayor que si nuestro sitio se compone básicamente de texto.

En cualquier caso, debemos dejar en manos de los ingenieros de red la tarea de establecer las características del servicio que precisamos.

▪ **Tipos de servicio**



buscarhosting.com es un buscador de servicios de *hosting* en lengua española

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad



En la mayoría de los proyectos de pequeño y mediano tamaño, los contenidos se publican mediante una empresa especializada. Estas empresas ofrecen diferentes servicios relacionados con la publicación *on-line*:

Hosting: consiste en el alquiler de un espacio en un servidor conectado a la red. De esta manera nos convertimos en administradores de una parcela de servidor web sobre la que dispondremos nuestros contenidos. A la hora de contratar un servicio de *hosting* debemos considerar el sistema operativo del servidor, si existe algún sistema de gestión de bases de datos, de qué tipo de soporte CGI dispone, si incluye servicio de correo electrónico, si permite el *streaming* de audio y vídeo, etc. Además, los proveedores de *hosting* suelen establecer un límite en el tráfico, es decir, el volumen de datos que se transfieren en un periodo determinado de tiempo.

La oferta de servicios de *hosting* es enorme y muy versátil: podemos encontrar servicios sencillos, por debajo de los 20€ al mes, que nos permitirán disponer de un pequeño website estático sin grandes dificultades.

Housing: los proyectos más complejos o de mayor entidad pueden requerir un ordenador con una configuración específica conectado a Internet. A diferencia del *hosting*, con el *housing* dispondremos de un servidor dedicado en exclusiva a nuestro proyecto, con la ventaja que supone el hecho de que personal especializado realice las tareas de mantenimiento que requiere este tipo de ordenador.

Gestión de dominios: el servicio de gestión de dominios comprende el mantenimiento de una dirección IP específica asociada a nuestro dominio de Internet. Suele estar incluido en los servicios de *hosting* o *housing*.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad



Actividad

Prepara los textos promocionales de tu producto: piensa en la forma y el espacio en que se van a presentar. Si se tratara de un producto *off-line*, se incluirían en la caja del disco; pero si lo que vamos a vender no necesita un soporte físico, podemos pensar en el contenido de una página web promocional o incluso en un anuncio en prensa.

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

Cierre del proyecto

Nos encontramos en la fase final de nuestro proyecto. Ya tenemos nuestro producto en el mercado y ahora debemos documentar cómo ha transcurrido el proceso, así como procesar y archivar todo el material producido. A continuación se describen las principales tareas a efectuar:

■ Preparar la memoria del proyecto

Al llegar a la fase final de nuestro proyecto, debemos recopilar toda la información relativa al mismo para ordenarla, clasificarla y archivarla con el objetivo de servirnos de referencia futura. La memoria de nuestro proyecto debe incluir:

- Documentación actualizada referente al diseño: prototipo, especificaciones de los diseños de la información, interacción y presentación, libro de estilo.
- Especificaciones técnicas del producto.
- Memoria técnica: tecnologías utilizadas, descripción de los recursos técnicos utilizados en la producción.
- Documentación relativa a la propiedad intelectual de los contenidos.
- Contratos realizados con empresas e individuales externos.
- Estado de cuentas: resumen de los costes internos y externos, así como relación de los recursos humanos y materiales utilizados.
- Informe de incidencias en la producción: este informe puede ser de gran utilidad para identificar los riesgos e incidencias que pueden producirse en futuros proyectos.

■ Recopilar los materiales producidos

A lo largo del proyecto habremos producido una gran cantidad de material que deberá ser correctamente identificado y archivado de forma que si, en el futuro, fuera preciso actualizar o ampliar nuestro producto o reaprovechar el contenido para otro proyecto, la localización de cada uno de los componentes sea una tarea fácil y eficiente.

La mayoría de los contenidos y aplicaciones que habremos generado existirán en dos versiones: la abierta, que admite modificaciones y puede utilizarse para generar nuevos contenidos, y la implementada, que es una copia de lo que se incluyó en nuestro producto. El cuadro siguiente muestra algunos ejemplos de ambas versiones:

	Versión abierta	Versión implementada
Textos	Estructurados en bases de datos, marcados e identificados	Formateados para su presentación en pantalla
Imágenes	En alta resolución, sin comprimir ni encuadrar	Comprimidas y en baja resolución
Audiovisuales	Material de rodaje, en cinta de vídeo sin editar	Archivos de vídeo comprimidos

Posgrado en Edición Digital (on line y off line)

Desarrollo, integración y control de calidad

	Versión final sin comprimir	
Aplicaciones	Archivos maestros Director, Flash, código fuente.	Aplicaciones ejecutables

Sólo si disponemos de las versiones abiertas de cada uno de los componentes del programa podemos asegurarnos la posibilidad de modificarlo o reaprovecharlo en el futuro.

Es imprescindible que todo el material esté debidamente documentado, tanto en lo referente a propiedad intelectual (autor, personas que han intervenido en la edición o transformación, etc.) como en lo técnico (comentarios al código fuente, descripción de formatos, etc.).

▪ Realizar copias de seguridad

Debemos prestar una atención especial a la conservación de copias de seguridad de todo el material digital de nuestro proyecto. Es prudente disponer de copias en más de una ubicación y, con el paso del tiempo, debemos plantearnos la realización de nuevas copias en soportes y formatos actualizados.