

4 — Implantación de contenido multimedia

4.1 Propiedad intelectual

El Ministerio de Cultura define la propiedad intelectual como el conjunto de derechos que corresponden a los autores y a otros titulares (artistas, productores, organismos de radiodifusión, etcétera) respecto de las obras y prestaciones fruto de su creación y es este Ministerio el encargado de proponer las medidas, normativas o no, para lograr la adecuada protección de la propiedad intelectual.

La Ley de propiedad intelectual ofrece mecanismos que permiten, entre otras acciones, proteger la obra frente a vulneraciones de carácter moral y percibir una remuneración económica cuando la obra es utilizada por terceros, sean cuales sean los propósitos de esta utilización.

El 12 de abril de 1996 fue aprobado el texto refundido de la Ley de propiedad individual mediante el Real Decreto Legislativo 1/1996. En este texto se derogaron las siguientes leyes:

- Ley 22/1987, de 11 de noviembre, de propiedad intelectual.
- Ley 20/1992, de 7 de julio, de modificación de la Ley 22/1987, de 11 de noviembre, de Propiedad Intelectual.
- Ley 16/1993, de 23 de diciembre, de incorporación al Derecho Español de la Directiva 91/250/CEE, de 14 de mayo, sobre la protección jurídica de programas de ordenador.
- Ley 43/1994, de 30 de diciembre, de incorporación al Derecho Español de la Directiva 92/100/CEE, de 19 de noviembre, sobre derechos de alquiler y préstamo y otros derechos afines a los derechos de autor en el ámbito de la propiedad intelectual.
- Ley 27/1995, de 11 de octubre, de incorporación al Derecho Español de la Directiva 93/98/CEE, del Consejo, de 29 de octubre, relativa a la armonización del plazo de protección del derecho de autor y de determinados derechos afines.
- Ley 28/1995, de 11 de octubre, de incorporación al Derecho Español de la Directiva 93/83/CEE, del Consejo, de 27 de septiembre, sobre coordinación de determinadas disposiciones relativas a los derechos de autor y derechos afines a los derechos de autor en el ámbito de la radiodifusión vía satélite y de la distribución por cable.

Este texto refundido fue objeto de modificaciones en las siguientes leyes:

- LEY 5/1998, de 6 de marzo, de incorporación al Derecho Español de la Directiva 96/9/CE, del Parlamento Europeo y del Consejo, de 11 de marzo de 1996, sobre la protección jurídica de las bases de datos.
- LEY 19/2006, de 5 de junio, por la que se amplían los medios de tutela de los derechos de propiedad intelectual e industrial y se establecen normas procesales para facilitar la aplicación de diversos reglamentos comunitarios.

Finalmente se aprobó la LEY 23/2006, de 7 de julio, por la que se modifica el texto refundido de la Ley de Propiedad Intelectual, aprobado por el Real Decreto Legislativo 1/1996, de 12 de abril, respondiendo a la necesidad de incorporar al derecho español una de las últimas directivas aprobadas en materia de propiedad intelectual, la Directiva 2001/29/CE del Parlamento Europeo y del Consejo, de 22 de mayo de 2001, relativa a la armonización de determinados aspectos de los derechos de autor y derechos afines a los derechos de autor en la sociedad de la información, con la que la Unión Europea, a su vez, ha querido cumplir los Tratados de la Organización Mundial de la Propiedad Intelectual (OMPI) de 1996 sobre Derecho de Autor y sobre Interpretación o Ejecución y Fonogramas.

4.1.1 Sujetos de propiedad intelectual

Es preciso distinguir entre los sujetos de los derechos de autor, y los sujetos de los otros derechos de propiedad intelectual (conocidos también como derechos afines, conexos o vecinos):

- **Sujetos de los derechos de autor:**

Se considera autor a la persona natural que crea alguna obra literaria, artística o científica. Son objeto de propiedad intelectual todas las creaciones originales literarias, artísticas o científicas expresadas por cualquier medio o soporte, tangible o intangible, actualmente conocido o que se invente en el futuro. La propiedad intelectual de una obra literaria, artística o científica corresponde al autor por el solo hecho de su creación.

La condición de autor tiene un carácter irrenunciable; no puede transmitirse "inter vivos" ni "mortis causa", no se extingue con el transcurso del tiempo así como tampoco entra en el dominio público ni es susceptible de prescripción.

- **Sujetos de los otros derechos de propiedad intelectual:**

- **Artistas intérpretes o ejecutantes.** Se entiende por tal a la persona que represente, cante, lea, recite o interprete en cualquier forma una obra. A esta figura se asimila la de director de escena y de orquesta.
- **Productores de fonogramas.** Persona natural o jurídica bajo cuya iniciativa y responsabilidad se realiza por primera vez la fijación exclusivamente sonora de la ejecución de una obra o de otros sonidos.
- **Productores de grabaciones audiovisuales.** Persona natural o jurídica que tiene la iniciativa y asume la responsabilidad de la grabación audiovisual.
- **Entidades de radiodifusión.** Personas jurídicas bajo cuya responsabilidad organizativa y económica se difunden emisiones o transmisiones.
- **Creadores de meras fotografías.** Persona que realice una fotografía u otra reproducción obtenida por procedimiento análogo a aquélla, cuando ni una ni otra tengan el carácter de obras protegidas en el Libro I de la Ley de Propiedad Intelectual.
- **Protección de determinadas producciones editoriales.** Hace referencia a las obras inéditas en dominio público y a determinadas obras no protegidas por las disposiciones del Libro I del TRLPI.

4.1.2 Derechos de la Propiedad Intelectual

Por lo que respecta a los derechos que conforman la propiedad intelectual se distinguen los derechos morales y los derechos patrimonial:

- **Derechos morales:**

Frente a los sistemas de corte anglosajón, la legislación española es claramente defensora de los derechos morales, reconocidos para los autores y para los artistas intérpretes o ejecutantes. Estos derechos son irrenunciables e inalienables, acompañan al autor o al artista intérprete o ejecutante durante toda su vida y a sus herederos o causahabientes al fallecimiento de aquellos. Entre ellos destaca el derecho al reconocimiento de la condición de autor de la obra o del reconocimiento del nombre del artista sobre sus interpretaciones o ejecuciones, y el de exigir el respeto a la integridad de la obra o actuación y la no alteración de las mismas.

- **Derechos de carácter patrimonial:** Hay que distinguir entre:

- Derechos relacionados con la explotación de la obra o prestación protegida, que a su vez se subdividen en derechos exclusivos y en derechos de remuneración:
Los derechos exclusivos son aquellos que permiten a su titular autorizar o prohibir los actos de explotación de su obra o prestación protegida por el usuario, y a exigir de este una retribución a cambio de la autorización que le conceda. Los derechos de remuneración, a diferencia de los derechos exclusivos, no facultan a su titular a autorizar o prohibir los actos de explotación de su obra o prestación protegida por el usuario, aunque si obligan a este al pago de una cantidad dineraria por los actos de explotación que realice, cantidad esta que es determinada, bien por la ley o en su defecto por las tarifas generales de las entidades de gestión.
- Derechos compensatorios, como el derecho por copia privada que compensa los derechos de propiedad intelectual dejados de percibir por razón de las reproducciones de las obras o prestaciones protegidas para uso exclusivamente privado del copista.

4.1.3 Preguntas más frecuentes sobre protección intelectual

- **¿Qué es la propiedad intelectual?**

La propiedad intelectual está integrada por una serie de derechos de carácter personal y/o patrimonial que atribuyen al autor y a otros titulares la disposición y explotación de sus obras y prestaciones.

- **¿Qué ley regula la propiedad intelectual?**

El texto Refundido de la Ley de Propiedad Intelectual, aprobado por Real Decreto Legislativo 1/1996, de 12 de abril.

- **¿Qué protege la propiedad intelectual?**

La propiedad intelectual protege las creaciones originales literarias, artísticas o científicas expresadas en cualquier medio, tales como libros, escritos, composiciones musicales, obras dramáticas, coreografías, obras audiovisuales, esculturas, obras pictóricas, planos, maquetas, mapas, fotografías, programas de ordenador y bases de datos. También protege las interpretaciones artísticas, los fonogramas, las grabaciones audiovisuales y las emisiones de radiodifusión.

- **¿Qué se excluye de la protección de la propiedad intelectual?**

Se excluyen las ideas, los procedimientos, métodos de operación o conceptos matemáticos en sí, aunque no la expresión de los mismos. También se excluyen las disposiciones

legales o reglamentarias, sus correspondientes proyectos, las resoluciones de los órganos jurisdiccionales y los actos de los organismos públicos, así como las traducciones de dichos textos.

■ **¿Por qué es necesario proteger la propiedad intelectual?**

Los derechos de propiedad intelectual otorgan además del reconocimiento a los creadores, la retribución económica que les corresponde por la realización de sus obras y prestaciones. Es también un incentivo a la creación y a la inversión en obras y prestaciones de la que se beneficia la sociedad en su conjunto.

■ **¿Cuándo una obra o prestación está protegida?**

Se protegen desde el momento de su creación, recibiendo los titulares la plena protección de la ley desde ese momento y sin que se exija el cumplimiento de ningún requisito formal.

■ **¿Qué son las medidas tecnológicas de protección?**

Las medidas tecnológicas se pueden definir como los instrumentos que están destinados a impedir o restringir al usuario de obras o prestaciones protegidas que no cuente con la pertinente autorización, la reproducción, comunicación pública, etc. de ellas. Lo que no pueden impedir las medidas tecnológicas a los usuarios de los derechos de explotación es disfrutar de algunos de los límites impuestos a los derechos de explotación. Estos límites impuestos por razones de tipo social, cultural, de seguridad nacional, etc. permiten a los usuarios-beneficiarios que puedan utilizar en determinados casos los derechos de reproducir, comunicar, distribuir, etc sin requerir la pertinente autorización a sus respectivos titulares.

Toda vez que la aplicación de las medidas tecnológicas puede en la práctica desvirtuar alguno de los límites impuestos a los derechos de explotación de los titulares de derechos de propiedad intelectual, ha sido necesario establecer que estos últimos faciliten los medios para que los beneficiarios de esos límites disfruten de los mismos. Cuando los titulares no faciliten estos medios voluntariamente, los beneficiarios de esos límites podrán acudir a órganos jurisdiccionales a los fines de hacer efectivos el cumplimiento de dicha obligación.

■ **¿Necesito autorización para utilizar obras y prestaciones divulgadas?**

Sí. La utilización de las obras o prestaciones requiere la autorización de los titulares de los derechos. Sin embargo, existen unos supuestos limitados, previstos en los artículos 31 a 40 bis de la ley, para los cuales no es necesario solicitar dicha autorización.

■ **¿Necesito autorización para utilizar obras y prestaciones difundidas en internet?**

Sí. La difusión pública de las obras y prestaciones a través de internet no implica que el titular de los derechos haya declinado su derecho a autorizar la explotación de su obra o prestación ni renunciado a obtener una remuneración.

■ **¿Y cuando intercambio obras y prestaciones protegidas a través de programas peer to peer (P2P), también necesito autorización?**

Los programas peer to peer han suscitado la duda sobre su posible legalidad o ilegalidad. Sin entrar en la polémica que pudiera suscitar su naturaleza, sí es necesario reconocer que su funcionamiento supone la ejecución de una serie de actos que implican el uso de derechos de explotación, que de no estar autorizados por sus respectivos titulares, suponen la infracción de los derechos reconocidos en el Texto Refundido de la Ley de Propiedad Intelectual.

- **¿Cómo obtengo autorización para utilizar obras y prestaciones?**

Una forma es ponerse en contacto con los titulares de los derechos y solicitarla. Para ciertas obras o prestaciones puede dirigirse a las Entidades de gestión colectiva de derechos de propiedad intelectual.

- **¿La copia privada es un derecho de los usuarios de obras y prestaciones protegidas?**

No es un derecho, es un límite al derecho de reproducción que ostentan los titulares de los derechos de propiedad intelectual de las obras y prestaciones protegidas.

Este límite permite que determinadas obras divulgadas a la cual haya tenido acceso legal una persona física pueda ser reproducida por esta, siempre que la copia que obtenga no sea utilizada de forma colectiva, ni lucrativa.

- **¿Por qué se ha establecido este límite al derecho de reproducción?** Para permitir que las personas físicas puedan mediante equipos, aparatos y soportes idóneos realizar copias de obras y prestaciones protegidas sin necesidad de tener que contar con la autorización de su titular.

- **¿Por qué la copia privada origina una compensación a favor de los titulares de las obras?**

Para compensar el daño económico que supone la realización de copias para uso privado que la ley permite que se hagan las personas físicas.

- **¿Quién está obligado a pagar la compensación por la copia privada?**

En España la obligación del pago de la compensación corresponde a los fabricantes e importadores de equipos y soportes idóneos para la reproducción de obras, que vayan a destinarlos a la distribución comercial o utilización dentro del territorio español. Además de los anteriores sujetos obligados al pago, los sucesivos adquirentes en la cadena de distribución que no sean considerados consumidores finales o usuarios de los equipos y soportes, son considerados responsables solidarios del pago de la compensación.

- **¿Quién determina qué equipos y soportes idóneos para la reproducción están sujetos al pago de la compensación por copia privada y el importe de la misma?**

En el artículo 25 del Texto Refundido de la Ley de Propiedad Intelectual se especifican los equipos, materiales y soportes de reproducción analógica que están sujetos al pago de una compensación, así como el importe de la misma.

La determinación de los equipos, materiales y soportes digitales sujetos al pago de la compensación y su importe lo determinan conjuntamente el Ministerio de Cultura y el Ministerio de Industria, Turismo y Comercio, previa negociación de los acreedores y deudores de la compensación y consulta al Consejo de Consumidores y Usuarios.

- **¿Puedo hacer una copia privada de cualquier tipo de obra o prestación protegida que se haya divulgado?**

Sí, excepto de los programas de ordenador y las bases de datos electrónicas.

- **¿Es necesario registrar una obra para protegerla?**

No. Al protegerse la obra por el sólo hecho de su creación, la inscripción en el Registro de la Propiedad Intelectual es voluntaria. Es conveniente, sin embargo, indicar la reserva de derechos y el símbolo ©, en el caso de una obra o prestación o, si se trata de fonogramas

el símbolo ©.

Las ventajas que ofrece la inscripción en Registro son, además de proporcionar una prueba cualificada de que los derechos inscritos existen y pertenecen a su titular, salvo que se demuestre lo contrario, la de dar publicidad a los derechos inscritos.

■ **¿Cómo puedo registrar una obra o prestación?**

Se debe presentar una solicitud ante el Registro de la Propiedad Intelectual y pagar una tasa. El modelo de solicitud así como la documentación que hay que acompañar en cada caso, se facilitan en El Registro General de la Propiedad Intelectual.

■ **¿Dónde se registra?**

En cualquier Registro Territorial o en sus Oficinas Delegadas, si las tuviere, o en las Oficinas Provinciales del Registro Central, a voluntad e interés del autor. Las direcciones de las oficinas del Registro General de la Propiedad Intelectual pueden consultarse en Direcciones de interés

■ **¿Cuál es la duración de los derechos de propiedad intelectual?**

El plazo general de los derechos de explotación de la obra es la vida del autor y setenta años después de su muerte. Existen otros plazos para los derechos morales y para otras prestaciones, así como para las obras de autores fallecidos antes de 1987.

■ **¿Cuándo una obra o prestación está en dominio público?**

Cuando el plazo de protección de los derechos ha expirado la obra o prestación pasa al dominio público, pudiendo ser utilizada por cualquiera, de forma libre y gratuita.

■ **¿Son transmisibles los derechos de propiedad intelectual?**

En general, los derechos patrimoniales son transmisibles a través de documento escrito.

■ **¿Qué consecuencias tiene la utilización de obras y prestaciones sin autorización?**

La utilización de obras y prestaciones sin autorización de los titulares puede suponer la infracción de los derechos de propiedad intelectual. Ante una infracción, el titular puede ejercitar las acciones penales y/o civiles previstas en la ley.

■ **¿Cómo se castiga la piratería?**

La piratería, en los supuestos más graves, está tipificada en los artículos 270, 271 y 272 del Código Penal, castigándose con penas de prisión y multa.

Si alguien infringe mis derechos de propiedad intelectual ¿me pueden asesorar? La Subdirección General de Propiedad Intelectual no tiene entre sus funciones la asesoría a personas físicas o jurídicas. Para ello debe consultar a un abogado especialista en la materia.

■ **¿Qué son las entidades de gestión de derechos de propiedad intelectual?**

Las entidades de gestión son entidades sin ánimo de lucro que tienen por objeto "la gestión de derechos de explotación u otros de carácter patrimonial, por cuenta y en interés de varios autores u otros titulares de derechos de propiedad intelectual". Estas entidades, que en España se han constituido como asociaciones de titulares de derechos de propiedad intelectual (autores, artistas, productores, etc), deben contar con la autorización del Ministerio de Educación, Cultura y Deporte para funcionar como tales.

■ **¿Cuántas son las entidades de gestión de derechos de propiedad intelectual?**

Hasta la fecha, el Ministerio de Educación, Cultura y Deporte ha autorizado ocho entidades de gestión, que representan a los siguientes titulares de derechos:

- **De autores:** SGAE (Sociedad General de Autores y Editores), CEDRO (Centro español de derechos reprográficos), VEGAP (Visual entidad de gestión de artistas plásticos), DAMA (Derechos de autor de medios audiovisuales).
- **De Artistas intérpretes o ejecutantes:** AIE (Artistas intérpretes o ejecutantes, sociedad de gestión de España, AISGE (Artistas intérpretes, sociedad de gestión).
- **De Productores:** AGEDI (Asociación de gestión de derechos intelectuales), EGEDA (Entidad de Gestión de Derechos de los productores audiovisuales).

■ **¿Para qué sirven las entidades de gestión de derechos de propiedad intelectual?**

Con el establecimiento y funcionamiento de las entidades de gestión se garantiza una explotación eficaz de las obras y prestaciones protegidas en beneficio de sus titulares y de los usuarios. De los titulares, porque estos tienen asegurado el control del uso de sus obras tanto en España como en el extranjero (esto último en virtud de los acuerdos de reciprocidad que suscriben con las entidades de gestión extranjeras); y de los usuarios, porque estos acudiendo a las entidades de gestión tienen asegurado en los supuestos de usos masivos de obras y prestaciones, el uso pacífico del repertorio mundial representado por estas entidades.

■ **¿Las entidades de gestión de derechos de propiedad intelectual son privadas o públicas?**

En su organización y funcionamiento las entidades de gestión, como entes privados que son, funcionan con total autonomía y están solo sujetas a la observancia de las normas del ordenamiento jurídico y en particular a lo dispuesto en la Ley de Propiedad intelectual.

■ **¿Como desempeñan sus funciones las entidades de gestión de derechos de propiedad intelectual?**

Para gestionar los derechos que sus estatutos le tienen encomendados, las entidades de gestión conceden a los usuarios autorizaciones no exclusivas para utilizar los derechos de los colectivos de titulares que representan a cambio de una contraprestación económica. La determinación de las contraprestaciones económicas que los usuarios deben abonar a las entidades de gestión por las autorizaciones que reciben, las fijan aquellas mediante el establecimiento de las tarifas generales que no están sujetas a la previa o posterior aprobación por el Ministerio de Educación, Cultura y Deporte, sin perjuicio de la obligación de negociar las tarifas con asociaciones de usuarios que quieran utilizar los derechos que tienen encomendados para su gestión.

Las cantidades recaudadas son abonadas a sus legítimos titulares previo el descuento de unos porcentajes variables destinados a atender los gastos en que incurren para prestar estos servicios. Además de atender a su finalidad de gestionar derechos, las entidades por imposición legal deben prestar a los colectivos de titulares que representan servicios asistenciales, formativos y promocionales.

■ **¿Por qué debo de pagar en algunos casos a más de una entidad de gestión por el uso de obras o prestaciones protegidas (grabaciones musicales, audiovisuales, etc.)?**

Las entidades de gestión son entidades sin ánimo de lucro que tienen por objeto, *la gestión de derechos de explotación u otros de carácter patrimonial, por cuenta y en interés de varios autores u otros titulares de derechos de propiedad intelectual*. Con este fin, han sido autorizadas hasta la fecha por el Ministerio de Educación, Cultura y Deporte ocho entidades de gestión:

- **De autores:** SGAE (Sociedad General de Autores y Editores), CEDRO (Centro español de derechos reprográficos), VEGAP (Visual entidad de gestión de artistas plásticos), DAMA (Derechos de autor de medios audiovisuales).
- **De Artistas intérpretes o ejecutantes:** AIE (Artistas intérpretes o ejecutantes, sociedad de gestión de España, AISGE (Artistas intérpretes, sociedad de gestión).
- **De Productores:** AGEDI (Asociación de gestión de derechos intelectuales), EGEDA (Entidad de Gestión de Derechos de los productores audiovisuales).

Los derechos de propiedad intelectual, que corresponden a los autores, los artistas intérpretes o ejecutantes, los productores fonográficos o audiovisuales y otros titulares, se caracterizan por su independencia y compatibilidad. La compatibilidad permite que sobre un soporte (cd o cassette) que contiene por ejemplo, una grabación exclusivamente sonora (fonograma), coexistan o converjan los derechos de explotación reconocidos por nuestra ley a distintos titulares, así encontramos los derechos del autor de la letra y la música, del intérprete vocalista o ejecutante de los instrumentos musicales y los del productor de la grabación que es aquel bajo cuya iniciativa y responsabilidad se lleva a cabo la grabación. Esta variedad de titulares de derechos presentes en determinadas explotaciones de obras y prestaciones protegidas (canciones, film, etc) contenidas en un soporte determinado (cd,dvd), se traduce en la intervención de las respectivas entidades de gestión que tienen encomendadas la gestión de los derechos de explotación.

Así las cosas si hacemos uso de la grabación musical contenida en un soporte (cd o cassette) por ejemplo para ambientar o amenizar un establecimiento esto es comunicar al público el contenido de la grabación, estaremos utilizando los derechos que la ley ha reconocido a los distintos titulares. Y como quiera que estos titulares han encomendado su gestión, bien voluntariamente o por disposición legal a las entidades de gestión, necesariamente deberemos acudir a cada una de ellas para solicitar la autorización para usar los derechos que ellas gestionan y pagar el precio que ellas fijen mediante sus tarifas.

4.1.4 Licencias

En el derecho anglosajón se utiliza el término Copyright o derecho de copia para hacer referencia a los derechos patrimoniales de los autores sobre su obra.

En el **Artículo 146** de la ley de propiedad intelectual se especifica lo siguiente: El titular o cesionario en exclusiva de un derecho de explotación sobre una obra o producción protegidas por esta Ley podrá anteponer a su nombre el símbolo © con precisión del lugar y año de la divulgación de aquéllas, y que dicho símbolo y referencia deben hacerse constar en modo y colocación tales que muestren claramente que los derechos de explotación están reservados.

¿Significa eso que si creamos una obra todo el mundo tiene la obligación de pagarnos por usarla? No, nosotros podemos ceder u otorgar derechos de explotación a terceras personas.

La Wikipedia define Licencia como un contrato mediante el cual una persona recibe de otra el derecho de uso de varios de sus bienes, normalmente de carácter no tangible o intelectual y describe algunos tipos de licencias:

- **Licencia copyleft:** El autor permite la libre distribución de copias y versiones modificadas de una obra u otro trabajo, exigiendo que los mismos derechos sean preservados en las versiones modificadas. Un ejemplo de esta licencia es la GFDL (GNU Free Documentation License o Licencia de documentación libre de GNU).
- **Dominio público:** Cualquiera puede manipular, distribuir y utilizar una obra de dominio público sin consecuencias legales. Un trabajo liberado al dominio público por su autor es

contenido libre.

- **Creative Commons:** Ofrecen determinados derechos a terceras personas bajo ciertas condiciones. Las licencias de Creative Commons están adaptadas a la legislación sobre propiedad intelectual del Estado Español. En este enlace podrás consultar la página relativa a las licencias de CC es (Creative Commons) España: <http://es.creativecommons.org/licencia/>
- **Contenido libre o información libre:** cualquier obra funcional, de arte u otro contenido creativo que no posee restricciones legales significativas en relación a derecho de uso, redistribución y creación de versiones modificadas o derivadas por parte de terceros. Incluye las obras de dominio público y algunas con derecho de autor (copyright) cuyas licencias cumplen y conservan las libertades mencionadas anteriormente.

Ejercicio 4.1 ¿Qué es la propiedad intelectual?

Ejercicio 4.2 ¿Cuál es la ley que incorpora los programas informáticos? ¿Esta ley deroga las anteriores?

Ejercicio 4.3 ¿La propiedad intelectual define como sujeto solamente a los autores de las obras? En caso contrario describe los otros sujetos que define la propiedad intelectual.

Ejercicio 4.4 ¿El derecho al honor y reconocimiento de las obras esta defendido en la ley de propiedad intelectual?

Ejercicio 4.5 ¿Qué dos derechos patrimoniales recoge la ley de propiedad intelectual?

Ejercicio 4.6 ¿Cuáles son las entidades de gestión de los derechos de la propiedad intelectual?

Ejercicio 4.7 Busca información sobre la propiedad intelectual aplicada a los entornos actuales: Música, películas (online). ¿Cuál crees que debe ser el modelo de negocio adecuado para respetar la ley de propiedad intelectual?

Ejercicio 4.8 ¿Cuándo y quién puede utilizar el símbolo ©?

Ejercicio 4.9 ¿Es obligatorio que toda persona o entidad que utilice nuestras creaciones debe pagarnos?

Ejercicio 4.10 Busca las diferentes variantes de las leyes Creative-Commons. ¿Qué proyectos crees que se puede adaptar a cada una de estas variantes?

4.2 Imágenes

En esta sección se explicarán los tipos de imágenes, formatos y herramientas necesarias para adaptar las imágenes a la Web. La información de esta sección ha sido extraída de las siguientes URL:

<https://sites.google.com/site/ticvalcarcel/>

<http://www.emezeta.com/articulos/formatos-de-imagen-guia-de-optimizacion>.

4.2.1 Tipos de Imágenes en la Web

A grandes rasgos, hay dos tipos de imágenes digitales:

- Imágenes **vectoriales**.
- Imágenes de **mapa de bits**.

Las imágenes vectoriales son imágenes constituidas por objetos geométricos autónomos (líneas, curvas, polígonos,...), definidos por ciertas funciones matemáticas (vectores) que determinan sus características (forma, color, posición,...). Por otro lado, las imágenes de mapa de bits están formadas por una serie de puntos (píxeles), cada uno de los cuales contiene información de



Figura 4.1: Diferencia entre imágenes: Vectorial y mapas de bits.

color y luminosidad. Salvando la diferencia, podemos compararla con un mosaico y sus teselas. En la Figura 4.1 se muestran dos imágenes, una vectorial y otra mapa de bits.

Las imágenes vectoriales se crean con programas de diseño o dibujo vectorial (*Adobe Illustrator*, *Corel Draw*, *Inkscape*...) y suelen usarse en dibujos, rótulos, logotipos... Su principal ventaja es que una imagen puede ampliarse sin sufrir el efecto de pixelado que tienen las imágenes de mapa de bits al aumentarse.

En la imagen (vectorial) del ratón de la Figura 4.2 puede apreciarse que al ampliar una zona no hay pérdida de detalle, mientras que en la fotografía del busto Nefertiti (mapa de bits) al ampliar mucho una zona, se observan los píxeles y la imagen se degrada.

Las imágenes de mapa de bits presentan una mayor gama de colores y de tonos que las vectoriales, por lo que son el tipo de imágenes usado en fotografía y, se crean con las cámaras de fotos, los escáneres y con programas de edición de imagen y dibujo (*Adobe Photoshop*, *Gimp*, *etc.*). Las imágenes mapa de bits generan archivos que ocupen mucha más memoria (bytes) que las imágenes vectoriales. Para poder reproducirse o utilizarse en un ordenador u otros dispositivos las imágenes vectoriales y de mapa de bits se guardan en archivos o ficheros (conjunto de datos que se almacenan en algún medio: disco duro, DVD, lápiz de memoria...). Cada archivo gráfico, se identifica además de por su nombre, por su extensión, que indica el tipo o formato de que se trata.

Algunos formatos de imagen vectorial son: AI (*Adobe Illustrator*), CDR (*Corel Draw*), DXF (*Autodesk*), EMF, EPS, ODG (*Open Office Draw*), SVG (*Inkscape*), SWF (*Adobe flash*), WMF (*Microsoft*)

Formatos de imagen de mapa de bits

Antes de empezar, debemos tener ciertos conceptos muy claros, ya que influyen en cada formato:

- **Formato con patentes/libre:** Un formato con patentes está ligado a una empresa que posee los derechos para su utilización, pudiendo fijar unas condiciones para utilizarlo (o restringirlo totalmente). Al contrario que la anterior, el formato libre no tiene restricciones y cualquiera puede utilizarlo o implementarlo en sus productos.
- **Tamaño/Calidad:** Los formatos pueden ser más o menos eficientes, pero el peso o tamaño de la imagen siempre estará reñido con la calidad de la imagen.



Figura 4.2: Diferencia entre imágenes: Vectorial y mapas de bits (II).

- **Uso del formato:** Por último, otro detalle que debemos tener en cuenta es lo extendido que esté el formato. Nos puede gustar un formato concreto, pero si no existe software que lo soporte (leer y guardar), tendremos problemas para su uso.

Imágenes con pérdidas

Se les llama formato de imagen con pérdidas (*lossy*) porque estos formatos no guardan toda la información que tiene la imagen original, sino que descartan datos que se consideran prescindibles (información no interpretable, simplificación, etc...), lo que lo hace ideal para fotografías o imágenes con muchas texturas y alto nivel de detalles. Aunque **JPEG** es el formato más conocido, existen varias alternativas que han intentado mejorar este formato y renovarlo, a la vez de dotarlo de mayor calidad en menos tamaño. Es importante remarcar, que algunos de estos formatos sucesores también permiten guardar imágenes sin pérdidas.

En la imagen de la Figura 4.3 se muestra una comparativa de los formatos anteriores, mostrando la pérdida de calidad y el peso de cada una de las imágenes.

- **WEBP:** El formato WebP, desarrollado por Google, está basado en el códec de video VP8 (WebM). Pretende ser un formato de imagen que reemplace al conocido JPEG, mejorando su calidad y reduciendo su tamaño, y por consiguiente, aumentando la velocidad con la que se descargan por Internet. Tiene una licencia BSD, por lo que casi se puede decir que es un formato libre.
- **JPEG 2000:** El formato JPEG 2000 fue presentado en el mismo año por los mismos autores del formato JPEG (1992), con la intención de mejorarlo y sustituirlo. Como desventaja, las imágenes con este formato resultan más difusas que con JPEG. Por otro lado, tiene varias mejoras como la posibilidad de añadir canales alfa (transparencias) o incluso, uno muy curioso, en el que se selecciona una zona para concentrar un nivel de detalles superior al exterior de esa zona. De los *sucesores* de JPEG es el más extendido (aún así, ha tenido poco éxito), pero también ha sido muy criticado y poco aceptado por las patentes adoptadas. En OpenJPEG se ha desarrollado una librería open source (licencia BSD) para adoptar este formato, incluso en productos comerciales.
- **JPEG XR:** Este formato, conocido anteriormente como HD Photo, está desarrollado por Microsoft. A pesar de haber sido criticado, al igual que JPEG 2000 por sus patentes, Microsoft lo ha estandarizado en 2010, convirtiéndolo en un candidato a sucesor de JPEG.



Figura 4.3: Diferencia entre imágenes: Imágenes con pérdida de calidad.

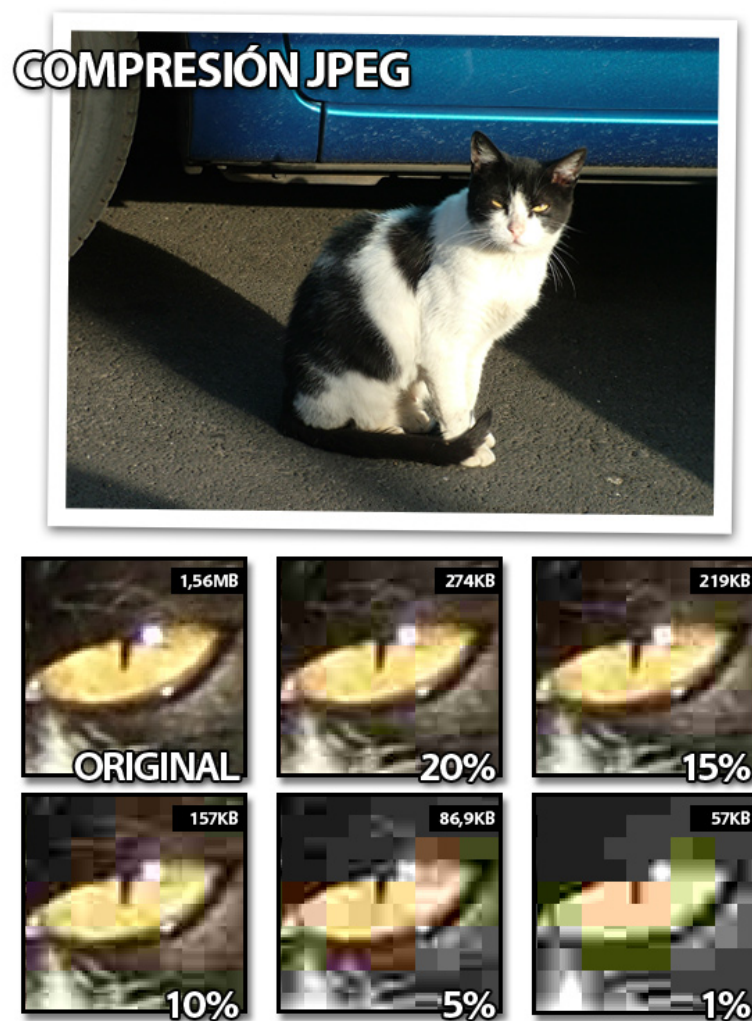


Figura 4.4: Pérdida de calidad en imágenes JPEG.

- **JPEG:** Todos los formatos anteriores conviene ir teniéndolos en cuenta, aún no son soportados por todo tipo de dispositivos y/o aplicaciones, por lo que (de momento) tendremos que optar por el formato JPEG. En estos formatos de imágenes con pérdidas, es posible establecer un nivel de calidad, que en el ejemplo es representado por un porcentaje (a mayor porcentaje, mayor calidad y tamaño, y menor compresión).

En la imagen de la Figura 4.4 se puede ver claramente, a niveles muy altos de compresión, los llamados artifacts JPEG, además de la diferencia de tamaño en los distintos niveles de calidad. El nivel de calidad deseado dependerá del uso que le vayamos a dar a nuestra imagen. Por ejemplo:

- **Impresión:** Si queremos imprimir nuestra imagen, lo ideal sería una calidad alta (compresión baja, tamaño alto), ya que queremos que los detalles sean perfectamente nítidos.
- **Página web:** Si queremos colocar una imagen en una página Web, nos interesará un tamaño bajo (compresión alta, calidad menor), y por lo tanto, a menos tamaño, más rápido se descargará de Internet. El establecer un nivel de calidad determinado (un 80 %, un 90 % o un 60 %), dependerá de la opinión subjetiva de la persona que guarde el formato. Por lo general, las aplicaciones vienen con el nivel de calidad predefinido a medio-alto.

Imágenes sin pérdidas

Los formatos de imágenes sin pérdidas (*lossless*), al contrario que los anteriores, no descartan información, sino que la almacenan y/o comprimen, de forma que siempre será posible reconstruir (exactamente) la imagen original.

Este tipo de formatos no debe utilizarse para fotografías o imágenes con muchas texturas y colores, sino para imágenes con colores sólidos o lisos: dibujos, gráficas, capturas de ventanas del sistema operativo, imágenes lineales, etc...

En las imágenes de la Figura 4.5 se muestra una comparativa de diferentes formatos para almacenar imágenes sin pérdida de calidad.

- **BMP:** El formato BMP, desarrollado por Microsoft, fue el formato propio del programa Paint, incluido en Windows. Su predecesor, el formato PCX también fue bastante conocido en entornos DOS, por incluirse en programas como PC PaintBrush o Deluxe Paint. Hoy en día, estos formatos son prácticamente históricos y poco prácticos, debido a su gran tamaño.

El formato BMP utilizaba un método de compresión llamado RLE, el cuál comprimía información idéntica adyacente, cosa que funcionaba bien en imágenes con pocos colores (o en blanco y negro).

- **TIFF:** El formato TIFF, muy similar al BMP, mejoraba sustancialmente por la introducción de la posibilidad de guardar imágenes utilizando compresión LZW. Más adelante, se añadieron posibilidades de comprimir mediante ZIP o JPEG, pero los archivos resultantes daban un tamaño muy elevado, razón por la cual dejó de utilizarse tan activamente, y se utilizó como formato para guardar fotografías a alta calidad en discos donde no importa demasiado la capacidad de almacenamiento.

Finalmente, se acabó delegando a tareas de programas de impresión, debido a su eficiencia con imágenes a grandes resoluciones y en blanco y negro. Fue inicialmente desarrollado por Microsoft, aunque actualmente es propiedad de Adobe.

- **GIF:** El formato GIF es uno de los más populares en la historia de Internet, desarrollado por CompuServe. Quizás, debido a la posibilidad de soportar animaciones, pero de eso hablaremos más adelante. Su principal característica es que GIF sólo puede tener, como

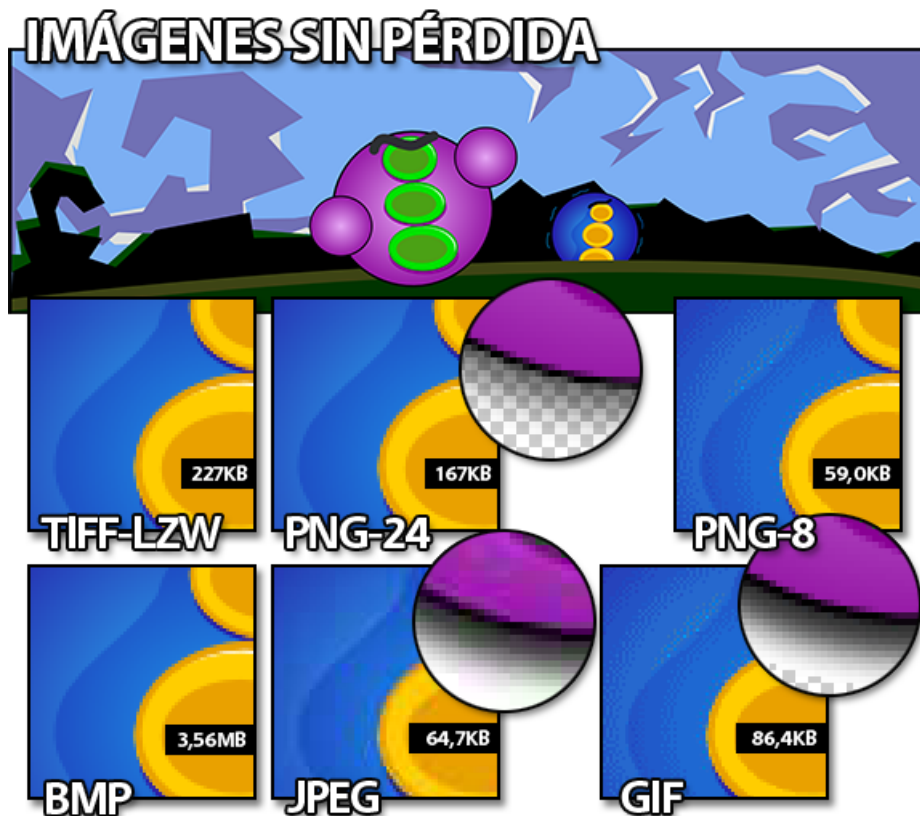


Figura 4.5: Diferencia entre imágenes: Imágenes sin pérdida de calidad.

máximo, 256 colores. Esto hace que su tamaño sea muy pequeño, aunque su calidad en imágenes como fotografías sea muy pobre.

Es ideal para pequeños logotipos, gráficos sólidos o animaciones. Fue el primer formato en soportar transparencia (especificando el color a convertir en *color transparente*). Tuvo problemas por temas de patentes (al utilizar LZW para comprimir) lo que aceleró el reemplazarlo por el formato PNG, que además, lo mejoraba considerablemente.

- **PNG:** Es un formato libre, que se creó para reemplazar a GIF. Cabe destacar, que dentro de PNG existen dos vertientes: PNG-8 y PNG-24. La primera de ellas, es la más similar a GIF, ya que se limita a 8 bits (256 colores como máximo) y está diseñado para, al igual que a GIF, guardar imágenes como logotipos, dibujos o gráficos lineales y simples, capturas con colores sólidos... de forma que su tamaño sea muy ligero (útil para Internet). Por otra parte, PNG-24 se limita a 24 bits, evitando la disminución de calidad en fotografías que sufría GIF y permitiendo canales alfa para transparencias (de mucha mejor calidad en sombras, como se puede ver en los círculos de la imagen). Eso sí, el tamaño del archivo aumenta considerablemente en este segundo caso. Existe una versión derivada de PNG para imágenes con pérdidas, llamada JNG (JPEG-NG) que aunque cabe mencionarla, no ha sido incluida en el apartado anterior, por no tener muy buen rendimiento en cuanto a tamaño/calidad.

Imágenes vectoriales

Uno de los problemas que se presentaba en la representación de imágenes (gráficos, logotipos, imágenes basadas en formas geométricas...) es que, en algunos casos, se necesitaba tener una misma imagen con diferentes resoluciones (mayores o menores).



Figura 4.6: Diferencia entre imágenes: Imágenes vectoriales.

En esos casos, el método al que se recurría era redimensionar la imagen, lo que producía -sobre todo en ampliaciones, que requiere interpolación- un desagradable efecto de *dientes de sierra* (*aliasing*) o mala calidad en general. Para evitar esto, se utilizan formatos vectoriales, en los que, se guarda la información de los objetos geométricos (con sus formas, colores, tamaños, etc...), de forma que si queremos cambiar el tamaño, es muy sencillo y no pierde calidad. En la imagen de la Figura 4.6 se muestra en una imagen el efecto de dientes provocado en imágenes no vectoriales.

- **SVG:** El formato vectorial más aceptado es el formato libre SVG, basado en el lenguaje de etiquetas XML, lo que lo hace muy fácil de generar y modificar. Al margen de SVG, existen otros formatos vectoriales, como EPS (Adobe), WMF (Microsoft), etc. Sin embargo, con la buena crítica y posibilidades del formato SVG, está rápidamente reemplazando a los demás. El formato SWF de Adobe, aunque no sea exactamente una imagen, utiliza también objetos vectoriales.

Imágenes propietarias

Los programas de diseño gráfico suelen contar con un formato propio, que incluyen técnicas como el de vectorizado, junto a otros datos como capas, selecciones, metadatos, e incluso un historial de acciones. Ficheros como **.AI** (Adobe Illustrator), **.CDR** (Corel Draw), **.PSD** (Adobe Photoshop), **.PSP** (Paint Shop Pro) deberían conservarse siempre como archivos originales para realizar posibles modificaciones en el futuro.

Recuerda no asumir nunca que los demás utilizan esa aplicación (y por lo tanto, son capaces de importar o abrir ese formato desde su programa habitual). Si nuestro objetivo es publicar la imagen, lo mejor es guardarla en un formato como JPEG o PNG, que son los más extendidos y soportados.

Imágenes RAW

Existe un último tipo que quiero mencionar, que es el de las imágenes RAW (en inglés, crudo). Este formato lo utilizan mucho los fotógrafos porque almacena la información exactamente como la captura el sensor de la cámara, sin ninguna alteración. Esto es interesante para personas que buscan un alto nivel de nitidez de imagen.

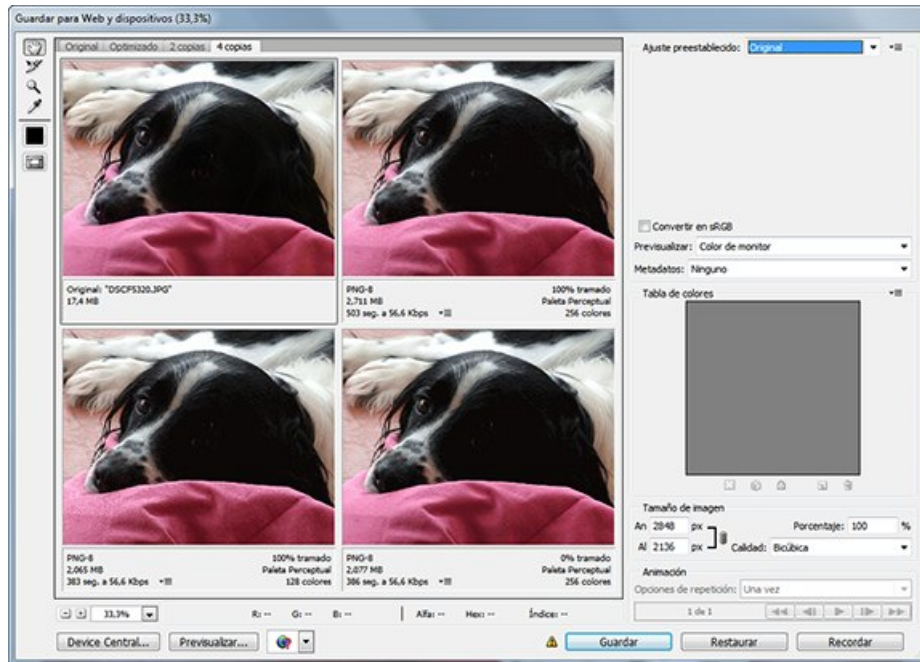


Figura 4.7: Photoshop: Save for Web.

Practicamente, cada compañía tiene su propio formato, como por ejemplo **.NEF** (Nikon), **.ORF** (Olympus), **.X3F** (Sigma), **.RAW** (Panasonic) o **.PXN** (Logitech), entre otras. Evidentemente, estos archivos ocupan cantidades muy elevadas en disco, lo que hace totalmente imposible (o al menos, desaconsejable) de utilizar para compartir o descargar desde Internet.

Ejercicio 4.11 ¿Cuáles son las diferencias principales entre imágenes escalares y vectoriales?

Ejercicio 4.12 ¿Existen imágenes escalares sin pérdida de calidad? ¿Cuáles son?

Ejercicio 4.13 En un diseño Web identifica claramente en qué imágenes utilizarás imágenes en formato RAW, PNG, JPEG y SVG.

4.2.2 Software para optimizar imágenes

Ahora que ya sabemos elegir el formato adecuado para nuestro caso, lo ideal es optimizarla al máximo, para que tenga el menor tamaño y la mayor calidad posible.

Existen varias utilidades para este tipo de optimizaciones:

- **Save for web (Photoshop):** *Save for Web* es una opción integrada en *Adobe Photoshop* que permite optimizar el tamaño de nuestra imagen con la mejor calidad posible. *Save for web* permite guardar formatos como JPG, GIF, PNG-8 y PNG-24 con numerosos parámetros de ajustes. En la Figura 4.7 se muestra la imagen de utilización de la opción *save for Web* de Photoshop.
- **RIOT: Radical Image Optimization Tool:** Probablemente, la joya más valiosa de este apartado. RIOT es una herramienta para realizar optimizaciones de nuestras imágenes, soportando JPEG, GIF o PNG. También permite varios parámetros de ajustes. Por si fuera poco, RIOT permite el uso de optimizadores externos complementarios como AdvPNG, OptiPNG o PNGOut. También puede utilizarse como plugin en varios programas como *GIMP*, *IrfanView* u otros. En la Figura 4.8 se muestra la aplicación ROIT con dos imágenes

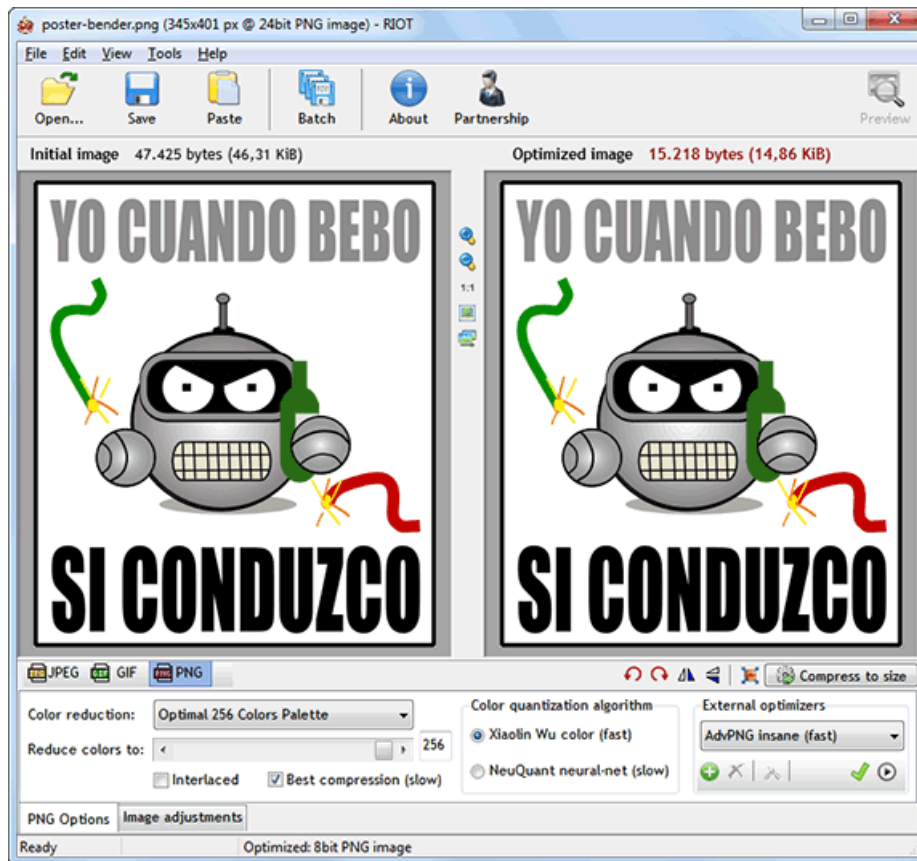


Figura 4.8: ROIT: Save for Web.

a optimizar.

- **Save for web (GIMP):** El editor gráfico gratuito GIMP no posee una opción para guardar para Web por defecto, pero existe un plugin para tal fin. Permite guardar en JPEG, GIF, PNG-8 y PNG-24 y varios parámetros como calidad, compresión o número de colores. En la Figura 4.9 se muestra el plugin de GIMP para salvar imágenes optimizadas a la Web.
- **Export (XnView):** XnView es un visualizador de imágenes que permite realizar todo tipo de tareas de edición gráfica, organización, visualización de metadatos, etc... Esta aplicación incorpora una función Exportar muy potente, donde puedes optimizar y guardar en formatos como JPEG, GIF, PNG entre otros.
- **FastStone Image Viewer:** Otra de estas aplicaciones que permiten optimizar imágenes es FastStone Image Viewer, que en su opción de Guardar como... contiene un panel muy simple, pero más que suficiente para optimizar imágenes.

Ejercicio 4.14 Instala todas las aplicaciones software presentadas en la sección anterior salvo Photoshop (no se dispone de licencia). Busca en Google Images dos imágenes de diferentes formatos: PNG, JPEG, SVG, GIF y realiza las siguientes tareas:

- Crea una ficha en la que se especifica de cada imagen: Tamaño, peso, formato, número de colores, otros datos de interés.
- Aplica las técnicas de optimización de las imágenes para la Web y completa nuevamente las fichas anteriores.

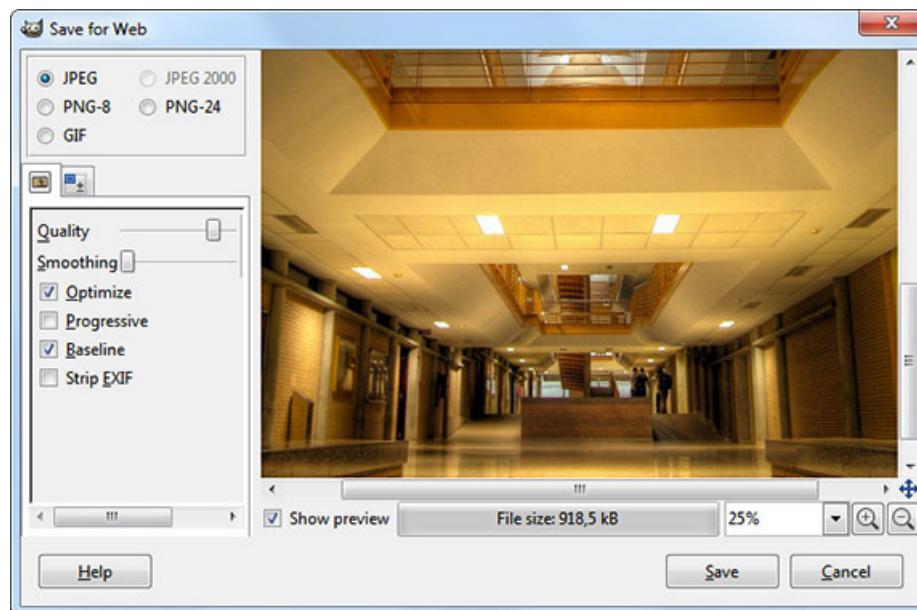


Figura 4.9: GIMP: Save for Web.

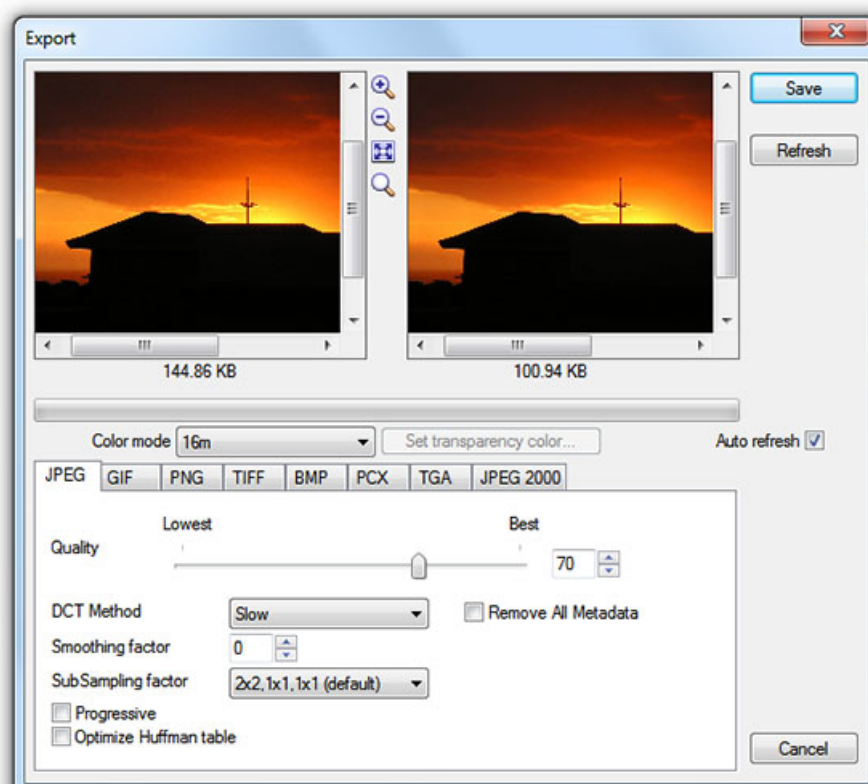


Figura 4.10: Export: Save for Web.

- Razona si se debe aplicar esta optimización a todas las imágenes del sitio Web.

4.3 Integración de audio y vídeo en la Web

4.3.1 Audio: Formatos y software

En el primer curso se estudió como se representa el sonido en el sistema informático. No obstante, algunos conceptos básicos de sonido digital son los siguientes:

- **Frecuencia:** Es el número de vibraciones por segundo que da origen al sonido analógico. El espectro de un sonido se caracteriza por su rango de frecuencias. Ésta se mide en Hertzios (Hz). El oído humano capta sólo aquellos sonidos comprendidos en el rango de frecuencias 20 Hz y 20.000 Hz.
- **Tasa de muestreo (sample rate):** Un audio digital es una secuencia de ceros y unos que se obtiene del muestreo de la señal analógica. La tasa de muestreo o *sample rate* define cada cuánto tiempo se tomará el valor de la señal analógica para generar el audio digital. Esta tasa se mide en Hertzios (Hz). Por ejemplo: 44100 Hz. nos indica que en un segundo se tomaron 44100 muestras de la señal analógica de audio para crear el audio digital correspondiente. Un audio tendrá más calidad cuanto mayor sea su tasa de muestreo. Algunas frecuencias estándares son 44100 Hz., 22050 Hz., y 11025 Hz.
- **Resolución (bit resolution):** Es el número de bits utilizados para almacenar cada muestra de la señal analógica. Una resolución de 8-bits proporciona 256 (2^8) niveles de amplitud, mientras que una resolución de 16-bits alcanza 65536 (2^{16}). Un audio digital tendrá más calidad cuanto mayor sea su resolución. Ejemplo: El audio de calidad CD suele ser un sonido de 44.100 Hz – 16 bits – estéreo.
- **Velocidad de transmisión (bitrate):** El bitrate define la cantidad de espacio físico (en bits) que ocupa un segundo de duración de ese audio. Por ejemplo, 3 minutos de audio MP3 a 128kBit/sg, ocupa 2,81 Mb de espacio físico ($3\text{min} \times 60\text{seg/min} \times 128\text{kBit/seg} = 23040\text{kBits} \rightarrow 23040\text{kBits} \times 1024\text{bits/Kbit} : 8\text{bits/bytes} : 1024\text{bytes/Kbytes} : 1024\text{Kbytes/Mbytes} = 2,81\text{Mbytes}$). Por ejemplo en los audios en formato MP3 se suele trabajar con bitrates de 128 kbps (kilobits por segundo). El audio tendrá más calidad cuanto mayor sea su bitrate y el archivo que lo contiene tendrá mayor peso. Esta magnitud se utiliza sobre todo en el formato MP3 de audio más destinado a la descarga por Internet.
- **CBR/VBR Constant/Variable Bitrate.** CBR indica que el audio ha sido codificado manteniendo el bitrate constante a lo largo del clip de audio mientras que VBR varía entre un rango máximo y mínimo en función de la tasa de transferencia.
- **Códec** Acrónimo de *codificación/decodificación*. Un códec es un algoritmo especial que reduce el número de bytes que ocupa un archivo de audio. Los archivos codificados con un codec específico requieren el mismo códec para ser decodificados y reproducidos. El códec más utilizado en audio es el MP3.
- **Decibelio** Unidad de medida del volumen o intensidad de un sonido. El silencio o ausencia de sonido se cuantifica como 0 dB y el umbral del dolor para el oído humano se sitúa en torno a los 130-140 dB.

Formatos

Existen 3 tipos de formatos: sin comprimir (WAV, AIFF o AU), comprimidos sin pérdida (o también conocidos como “lossless”, WMA, FLAC, TTA, ATRAC) y comprimidos con pérdida (o “lossy”, como el famoso MP3, AAC... etc.).

En el caso del formato sin comprimir, en el que no existe ningún tipo de procesamiento de la señal, el audio que escuchamos es el audio real que se grabó; por otro lado, el formato

comprimido sin pérdida, simplemente es un aprovechamiento de los silencios y otros factores para que el archivo pese algo menos, pero sin pérdida de señal o recorte en frecuencias. Y por último, los formatos con pérdida directamente recortan “frecuencias no audibles” para hacer que el archivo de audio sea muy liviano.

Personalmente os propongo que escuchéis un audio en MP3 y otro en WAV para que veáis la enorme diferencia que existe entre ambos formatos, la pérdida es considerable y para mi gusto un “temazo” en MP3 no tiene nada que ver con la misma canción en WAV.

Ahora veamos algunas características de algunos formatos de los formatos más populares:

- **WAV:** Si queremos que no se pierda calidad, se recomienda el formato WAV (o su equivalente en Mac, el formato AIFF), pero ocupan mucho espacio. Un minuto de música en calidad CD ocupa aproximadamente unos 15 Megs; audio a 44.100 Hz a 16 bits en estéreo ocupa 172 Kbps, dato nada despreciable, y más si lo que queremos es mover dicho archivo por Internet. Una canción en formato WAV suele rondar los 45 megas. La solución será comprimir el archivo o bajarlo de calidad. La compresión ofrece, obviamente, muchas más garantías.

Éste y otros formatos sin compresión similares, son *fieles al oído humano* y reproducen lo grabado en todo el espectro audible (20Hz-20KHz), proporcionando una escucha real y fiel del producto o el archivo de audio en cuestión, pero el formato WAV no es nada popular en la red debido a su enorme tamaño.

- **MP3:** Aún así, todo parece indicar que el MP3 es *el mejor formato*, pero si lo que queréis son unos graves profundos, redondos, y unos agudos brillantes, o un estéreo con abertura y espacio sonoro, con el MP3 no va a ser posible conseguirlo, pues se sabe que este formato hace una criba conservando solo frecuencias comprendidas aproximadamente entre los 200Hz-15KHz... se come por completo las frecuencias fundamentales de instrumentos como el bajo, el bombo, el charles o el crash de la batería.

Con el uso de formatos en MP3 o MP4, archivos que ocupaban 45 megas pasan a ocupar 4 o 5 megas o menos, lo cual hace que el sistema de distribución por paquetes de Internet y las relaciones y *feedbacks* entre profesionales del audio sea muchísimo más llevadera y fácil, sacrificando un poco de calidad por tener unos archivos muchísimo más manejables. El MP3 a 192K es pues del formato más cómodo y eficiente a la hora de moverlo de un sitio o de un programa a otro.

- **MP4:** Otro formato bastante importante y que ha tenido un boom en los últimos años es el MP4. En realidad la calidad de este formato es muy parecida a la del MP3, pero con un tamaño mucho más reducido. También se puede encontrar este tipo de compresión en los archivos con formato FLV de Youtube. El MP4 es el rey de los formatos de audio en Internet, y generalmente es el que encontraremos a la hora de reproducir audio en la Web y sobre todo en smartphones y cualquier otro pequeño dispositivo. Dada su versatilidad y su extraordinaria compresión es el formato ideal para mover archivos por la Red.
- **RAX:** En el caso de los famosos *streamings* se utiliza casi exclusivamente un formato conocido como RealAudio (RAX), basado en el formato ACC propiedad de Apple que mejora el rendimiento del MP3 y está destinado principalmente a reproductores portátiles.
- **FLAC:** Sin embargo si lo único que queremos es preservar nuestros discos de música favoritos, una de las mejores opciones a elegir es el formato libre de compresión sin pérdida FLAC (o .FLA). Una canción comprimida mediante este formato pesa entre el 70% y el 50% que la original (menos que en ZIP), con una gran calidad y reproducible en la mayoría de los reproductores de los distintos sistemas operativos (Linux, Microsoft, Apple... etc.).

No es una gran reducción, pero eso es debido a que no elimina información del contenido original. Con la creciente irrupción de conexiones con mayor ancho de banda, este formato se ha convertido en uno de los favoritos a la hora de comprar música por Internet y como una de las alternativas al MP3 para no perder tanta calidad con la reducción de tamaño.

A la hora de elegir un formato de audio en nuestros proyectos (vídeos, canciones o películas Flash) lo mejor es usar la mejor calidad posible y luego ya convertiremos la mezcla final o el archivo final al formato que deseemos para moverlo por la Red. No sería buena idea utilizar archivos de audio en MP3 y luego pretender que la mezcla final tenga la calidad de un WAV o FLAC.

Software para manipular audio

Para esta sección se utilizará el documento del curso de **Diseño de Materiales Multimedia 2.0** del ITE (Instituto de Tecnologías Educativas) facilitado por el profesor.

4.3.2 Vídeo: Formatos y Software

Antes de avanzar hacia la conversión de los formatos de vídeo se deben conocer algunos conceptos básicos relacionados con el vídeo:

- **Dimensiones:** Es el tamaño del video (ancho x alto) expresado en píxeles cuando se visualiza al 100 %, sin agrandar ni reducir. Los reproductores pueden mostrar un video a pantalla completa o con una ampliación del 200 %, 300 %, etc. En estos casos el video pierde calidad de imagen y esta pérdida depende del formato de archivo. Un video AVI puede tener cualquier ancho y alto mientras que los estándares de VideoCD son 352 x 288 y de DVD 720 x 576.
- **Codec:** Acrónimo de *codificación/decodificación*. Un códec es un algoritmo especial que reduce el número de bytes que ocupa un archivo de video. Los archivos codificados con un códec específico requieren el mismo códec para ser decodificados y reproducidos. Algunos de los códec más utilizados para el formato AVI son: DivX, XviD, CinePak, Intel Indeo 5, DV, etc.
- **Velocidad de transmisión (bitrate):** El bitrate define la cantidad de espacio físico (en bits) que ocupa un segundo de duración de ese video. El video tendrá más calidad cuanto mayor sea su bitrate y el archivo que lo contiene tendrá mayor peso. El bitrate puede ser fijo o variable. El bitrate variable consigue mayor calidad de imagen porque recoge más calidad en escenas muy cargadas o con mucho movimiento y ahorra en aquellas más estáticas.
- **Fotogramas por segundo:** Un video resulta de la exposición imágenes o fotogramas uno detrás de otro. Un parámetro de la calidad del video es el número de fotogramas por segundo que muestra durante su reproducción. Este valor oscila entre 15 y 30. Por ejemplo los vídeos en DVD en Europa exhiben 25 fotogramas por segundo (25 fps).
- **Fotogramas Clave:** Cuando se aplica un códec de compresión a un video, se suele producir cierta pérdida de la información de sus fotogramas. Algunos fotogramas (los fotogramas clave) se almacenan completamente en el archivo comprimido, mientras que el resto sólo se guardan parcialmente. En la descompresión, estos fotogramas intermedios se reconstruyen a partir de los fotogramas clave.
- **Sistemas de televisión:**
 - **NTSC (National Television Standards Comité = Comité Nacional de Estándares de Televisión).** Cada fotograma está formado por 525 líneas y reproduce 30 fotogramas por segundo. Se utiliza en América del Norte, Centroamérica, Japón, etc.

- **PAL (Phase Alternation Line = Línea Alternada en Fase).** El vídeo PAL tiene 625 líneas por fotograma y 25 fotogramas por segundo. Es el sistema más extendido actualmente en Europa.
- **SECAM (Séquentiel Couleur à Mémoire = Color secuencial con memoria).** Muestra 625 líneas y 25 fotogramas por segundo. De origen francés, ha perdido mercado en Europa a favor del sistema PAL.
- **Proporción o ratio de aspecto:** Es la proporción entre la anchura y altura de un video. Cuando se reproduce un video se suele mantener por defecto esta proporción para evitar deformación de las imágenes. Por este motivo cuando se elige la visualización a pantalla completa, aparecen franjas negras arriba y abajo. Es habitual una relación 4:3 para los videos domésticos (352x288 píxeles, por ejemplo) mientras que en DVD se suele trabajar con ratios de 16:9.

Formatos

Los videos digitales se pueden guardar en archivos de distintos formatos. Cada uno se corresponde con una extensión específica del archivo que lo contiene. Existen muchos tipos de formatos de video. Aquí se citan algunos de los más utilizados. Asimismo cada tipo de archivo admite en cada momento un códec de compresión distinto.

- **AVI (Audio Video Interleaved = Audio y Video Intercalado):**
 - Es el formato estándar para almacenar video digital.
 - Cuando se captura video desde una cámara digital al ordenador, se suele almacenar en este formato con el códec DV (Digital Video).
 - El archivo AVI puede contener video con una calidad excelente. Sin embargo el peso del archivo resulta siempre muy elevado.
 - Admite distintos códecs de compresión como CinePak, Intel Indeo 5, DV, etc. Los códecs con más capacidad de compresión y una calidad aceptable son DivX y XviD.
 - El formato AVI puede ser visualizado con la mayoría de reproductores: Windows Media, QuickTime, etc. siempre y cuando se encuentren instalados en el equipo los adecuados códecs para cada tipo de reproductor.
 - Es ideal para guardar videos originales que han sido capturados de la cámara digital (codificados con DV).
 - No es recomendable publicarlos en Internet en este formato por su enorme peso.
 - Los códecs CinePak, Intel Indeo, DV, etc. no ofrecen una gran compresión. Los códecs DivX y XviD por el contrario consiguen una óptima compresión aunque se suelen destinar sobre todo a la codificación de películas de larga duración.
- **MPEG (Moving Pictures Expert Group = Grupo de Expertos de Películas)**
 - Es un formato estándar para la compresión de video digital.
 - Son archivos de extensión *.MPG ó *.MPEG.
 - Admite distintos tipos de códecs de compresión: MPEG-1 (calidad CD), MPEG-2 (calidad DVD), MPEG-3 (orientado al audio MP3) y MPEG-4 (más orientado a la web).
 - Se reproducen con Windows Media Player y QuickTime.
- **MOV (<http://www.apple.com/es/quicktime/>)**
 - Es el formato de video y audio desarrollado por Apple.
 - Utiliza un códec propio que evoluciona en versiones con bastante rapidez.
 - Este tipo de archivos también pueden tener extensión *.QT
 - Se recomienda utilizar el reproductor de QuickTime. Existe una versión gratuita del mismo que se puede descargar de Internet.
 - Es ideal para publicar videos en Internet por su razonable calidad/peso.

- Admite streaming.
- **WMV (<http://www.microsoft.com/windows/windowsmedia/es/>)**
 - Ha sido desarrollado recientemente por Microsoft.
 - Utiliza el códec MPEG-4 para la compresión de vídeo.
 - También puede tener extensión *.ASF
 - Sólo se puede visualizar con una versión actualizada de Windows Media 7 o superior. Esta aplicación viene integrada dentro de Windows.
 - Es ideal para publicar vídeos en Internet por razonable calidad/peso.
 - Admite streaming.
- **RM (<http://spain.real.com/>)**
 - Es la propuesta de Real Networks para archivos de vídeo.
 - Utiliza un códec propio para comprimir el audio.
 - Este tipo de archivos tiene extensión *.RM y *.RAM.
 - Se visualiza con un reproductor específico: Real Player. Existe una versión gratuita del mismo que se puede descargar de Internet.
 - Se puede utilizar para publicar vídeos en Internet por su aceptable calidad/peso.
 - Admite streaming.
- **FLV (<http://www.adobe.com>)**
 - Es un formato que utiliza el reproductor Adobe Flash para visualizar vídeo en Internet.
 - Utiliza el códec Sorenson Spark y el códec On2 VP6. Ambos permiten una alta calidad visual con bitrates reducidos.
 - Son archivos de extensión *.FLV.
 - Se pueden reproducir desde distintos reproductores locales: MPlayer, VLC media player, Riva, Xine, etc.
 - Opción recomendada para la web por su accesibilidad. Al visualizarse a través del reproductor de Flash es accesible desde la mayoría de los sistemas operativos y navegadores web.
 - Los repositorios de vídeo más conocidos en Internet utilizan este formato para la difusión de vídeos: YouTube, Google Video, iFilm, etc.
 - Permite configurar distintos parámetros del vídeo para conseguir una aceptable calidad/peso.
 - Admite streaming.

Software para manipular vídeo

Para esta sección se utilizará el documento del curso de **Diseño de Materiales Multimedia 2.0** del ITE (Instituto de Tecnologías Educativas) facilitado por el profesor.

4.4 HTML5: Audio y vídeo

4.4.1 Reproduciendo vídeo en HTML5

El elemento de vídeo es una de las funciones de HTML5 que suscitan más interés. Esta etiqueta se suele presentar como alternativa a Flash para el contenido multimedia, pero tiene más aplicaciones. Aunque recientemente se ha incorporado al resto de etiquetas HTML omnipresentes, sus posibilidades y su compatibilidad con diferentes navegadores han aumentado a una increíble velocidad.

En el Código 4.1 se muestra el uso básico de vídeos en HTML5. En dicho código se hace uso del elemento <source>, el cual permite incluir varios formatos como tipos alternativos en caso de que el navegador del usuario no admita algunos de ellos.

Código 4.1: HTML5: Video.

```

1 <video>
2   <source src="movie.mp4" type='video/mp4; codecs="avc1.42E01E, _mp4a.40.2
   " ' />
3   <source src="movie.webm" type='video/webm; codecs="vp8, _vorbis" ' />
4 </video>

```

No obstante, el uso ideal del elemento `<video>` puede ser el mostrado en el Código 4.2 en el cual se utiliza el atributo `src` de un modo similar al elemento ``. Esto hoy en día no es posible debido a que no todos los navegadores soportan todos los formatos de vídeo y hay que especificar diversos formatos para conseguir compatibilidad con todos los navegadores.

Código 4.2: Uso ideal del elemento `<video>`. Aún no estandarizado.

```

1 <video src="movie.webm"></video>

```

Los atributos del elemento `<video>` son los siguientes:

- **autoplay:** Especifica si el vídeo comenzará a ejecutarse tan pronto como esté listo.
- **controls:** Especifica si los controles (inicio/paro) se deben mostrar.
- **height:** Especifica la altura del reproductor de vídeo.
- **loop:** Especifica si el vídeo debe comenzar de nuevo una vez haya terminado.
- **muted:** Especifica si el audio del vídeo debe estar apagado.
- **poster:** Especifica una imagen para ser mostrada mientras el vídeo se carga y hasta que el usuario pulse el botón de inicio.
- **preload:** Especifica cómo se debe cargar el vídeo cuando se cargue la página.
- **src:** Especifica la URL del origen del vídeo.
- **width:** Especifica la anchura del reproductor de vídeo.

Código 4.3: Ejemplo de uso del elemento `<video>` con diferentes atributos.

```

1 <video poster="star.png" autoplay loop controls tabindex="0">
2   <source src="movie.webm" type='video/webm; codecs="vp8, _vorbis" ' />
3   <source src="movie.ogv" type='video/ogg; codecs="theora, _vorbis" ' />
4 </video>

```

Formatos de vídeo

El concepto de formato de vídeo se puede entender como un archivo zip que contiene secuencias de vídeo y secuencias de audio. Los tres formatos que se deben tener en cuenta para la Web son WebM, MP4 y OGV:

- **.mp4** = H.264 + AAC.
- **.ogg/.ogv** = Theora + Vorbis.
- **.webm** = VP8 + Vorbis.

```

1 <video>
2   <source src="movie.webm" type='video/webm; codecs="vp8, _vorbis" ' />
3   <source src="movie.mp4" type='video/mp4; codecs="avc1.42E01E, _mp4a.40.2
   " ' />
4   <source src="movie.ogv" type='video/ogg; codecs="theora, _vorbis" ' />
5   Video tag not supported. Download the video <a href="movie.webm">here</
   a>.
6 </video>

```

Uno de los principales problemas relacionados con el uso del formato .mp4 es que su códec de vídeo (h.264) no es un códec abierto y que las licencias que tendría que pagar una empresa para utilizarlo son bastante complejas.

Otro problema que afecta al formato .mp4 es que el atributo type tiene que ser más específico que otros formatos dependiendo del perfil que se utilice para codificar el vídeo. Aunque el más común es *avc1.42E01E*, *mp4a.40.2*, es importante comprobar esta lista de perfiles para asegurarse.

Es importante que el servidor muestre los archivos de vídeo con el tipo MIME correcto en el encabezado Content-Type. De lo contrario, los vídeos podrían no funcionar correctamente (aunque sí funcionen en una copia local del sitio). En un archivo de configuración *httpd.conf* de Apache, solo habría que añadir estas líneas:

```
1 AddType video/ogg .ogv
2 AddType video/mp4 .mp4
3 AddType video/webm .webm
```

Además, para mejorar el rendimiento del cliente, es importante no olvidarse de especificar el atributo type en las etiquetas source cuando se tenga que hacer referencia a varios formatos. De ese modo, el navegador podrá decidir qué formatos puede descargar y reproducir y no descargará los formatos que no pueda reproducir para aumentar el rendimiento del sitio.

Ejercicio 4.15 Descarga un vídeo en cualquier formato. Observa inicialmente las características del vídeo (tamaño, peso, formato,...). Posteriormente busca un método para conseguir convertir dicho formato en los formatos soportados por la Web (ogg, mp4 y webm).

Ejercicio 4.16 Integra en un documento HTML5 un vídeo que esté soportado por todos los formatos permitidos. Comprueba que el vídeo es reproducido de modo satisfactoriamente en diferentes navegadores.

Ejercicio 4.17 Haz las modificaciones necesarias en el código anterior para que los botones de control sean visibles permanentemente, el vídeo se cargue automáticamente y tenga una imagen de poster inicialmente.

4.4.2 Construir un reproductor de vídeo en HTML5

Los videos tienen un aspecto diferente según cada navegador. Pero lo normal es que se quiera adaptar el reproductor de vídeo al diseño de nuestro sitio Web. En esta sección se va a presentar el modo de conseguir construir un reproductor de vídeo en HTML5.

HTML5 proporciona nuevos eventos, propiedades y métodos para manipular video e integrarlo al documento. De ahora en más, podremos crear nuestro propio reproductor de video y ofrecer las funciones que queremos usando HTML, CSS y JavaScript.

El diseño

Todo reproductor de video necesita un panel de control con al menos algunas funciones básicas. Es por ello que en primer lugar se define el documento HTML que estructurará el documento tal y como se muestra en el Código 4.4.

Código 4.4: Estructura HTML del reproductor de vídeo.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <title>Reproductor de Video</title>
5     <link rel="stylesheet" href="reproductor.css">
6     <script src="reproductor.js"></script>
```

```

7   </head>
8   <body>
9     <section id="reproductor">
10      <video id="medio" width="720" height="400">
11        <source src="./content/trailer.mp4">
12        <source src="./content/trailer.ogg">
13      </video>
14      <nav>
15        <div id="botones">
16          <button type="button" id="reproducir">Reproducir</button>
17        </div>
18        <div id="barra">
19          <div id="progreso"></div>
20        </div>
21        <div style="clear:_both"></div>
22      </nav>
23    </section>
24  </body>
25 </html>

```

Observe que en el código se ha creado una sección específica denominada *reproductor*. Posteriormente se ha incorporado el elemento `<video>`. Para redefinir la estructura de nuestro reproductor se ha creado un menú de navegación (`<nav>`) en el cual se han incorporado elementos botones (con un botón reproducir) y una barra de progreso.

El aspecto es presentado en la hoja de estilos *reproductor.css* (Código 4.5).

Código 4.5: Hoja de estilos de nuestro reproductor de video.

```

1  body{text-align:_center;}
2  header, section, footer, aside, nav, article, figure, figcaption{display
   :_block;}
3  #reproductor{
4    width:_720px;
5    margin:_20px_auto;
6    padding:_5px;background:_#999999;
7    border:_1px_solid_#666666;
8    -moz-border-radius:_5px;
9    -webkit-border-radius:_5px;
10   border-radius:_5px;}
11  nav{margin:_5px_0px;}
12  #botones{
13    float:_left;
14    width:_100px;
15    height:_20px;}
16  #barra{
17    position:_relative;
18    float:_left;
19    width:_600px;
20    height:_16px;
21    padding:_2px;
22    border:_1px_solid_#CCCCC;
23    background:_#EEEEEE;}
24  #progreso{
25    position:_absolute;
26    width:_0px;
27    height:_16px;
28    background:_rgba(0,0,150,.2);}

```



Figura 4.11: HTML5 Video: Apariencia de un reproductor propio.

Revise el código de la hoja de estilos. No se presenta ninguna novedad frente a las presentadas en el capítulo anterior. El resultado de la apariencia debe ser similar a la mostrada en la Figura 4.11.

Código del reproductor HTML5

Es momento de escribir el código Javascript para nuestro reproductor. En el reproductor que se presenta se utilizan pocas funcionalidades que permiten reproducir y pausar el vídeo. Además, se dispondrá de una barra de progreso que mostrará el avance del vídeo permitiendo hacer clic sobre esta barra para adelantar o retroceder el vídeo. e

Los eventos

HTML5 incorpora nuevos eventos que son específicos de cada API. Para el procesamiento de video y audio, por ejemplo, los eventos fueron incorporados con el objetivo de informar sobre la situación del medio (el progreso de la descarga, si la reproducción del medio finalizó, o si la reproducción del medio es comenzada o pausada, entre otras). Algunos de los eventos más relevantes son:

- **progress:** Este evento es disparado periódicamente para informar acerca del progreso de la descarga del medio. La información estará disponible a través del atributo *buffered*.
- **canplaythrough:** Este evento es disparado cuando el medio completo puede ser reproducido sin interrupción. El estado es establecido considerando la actual tasa de descarga y asumiendo que seguirá siendo la misma durante el resto del proceso. Existe otro evento más para este propósito, *canplay*, pero no considera toda la situación y es disparado tan pronto como algunas partes del medio se encuentran disponibles (luego de descargar los primeros cuadros de un video, por ejemplo).
- **ended:** Es disparado cuando el reproductor llega al final del medio.
- **pause:** Es disparado cuando el reproductor es pausado.

- **play:** Es disparado cuando el medio comienza a ser reproducido.
- **error:** Este evento es disparado cuando ocurre un error. Es relacionado con el elemento `<source>` correspondiente a la fuente del medio que produjo el error.

Para el ejemplo que se presenta se utilizarán los eventos habituales de *click* y *load*.

```
1 function iniciar() {
2   maximo=600;
3   medio=document.getElementById('medio');
4   reproducir=document.getElementById('reproducir');
5   barra=document.getElementById('barra');
6   progreso=document.getElementById('progreso');
7   reproducir.addEventListener('click', presionar, false);
8   barra.addEventListener('click', mover, false);
9 }
```

El primer código que se muestra es lanzado cuando la página es cargada y permite definir las variables globales para configurar el reproductor. Se hace uso del selector *getElementById* para referenciar a cada uno de los elementos del reproductor para poder acceder a dichos elementos en el resto del código. Además, se declara la variable *maximo* para conocer siempre el máximo tamaño posible para la barra de progreso (600 píxeles).

Se deben construir funciones para dos acciones: cuando el usuario hace clic sobre el botón reproducir y cuando hace clic sobre la barra de progreso para avanzar o retroceder el vídeo. Dos escuchas para el evento clic fueron agregadas con el propósito de controlar estas situaciones.

Primero agregamos la escucha al elemento reproducir que ejecutará la función *presionar()* cada vez que el usuario haga clic sobre el botón *Reproducir*. La otra escucha es para el elemento barra. En este caso, la función *mover()* será ejecutada cada vez que el usuario haga clic sobre la barra de progreso.

```
1 function presionar() {
2   if(!medio.paused && !medio.ended) {
3     medio.pause();
4     reproducir.innerHTML='Reproducir';
5     window.clearInterval(bucle);
6   }else{
7     medio.play();
8     reproducir.innerHTML='Pausa';
9     bucle=setInterval(estado, 1000);
10  }
11 }
```

Los métodos *play()* y *pause()* son parte de una lista de métodos incorporados por HTML5 para procesamiento de medios. Los siguientes son los más relevantes:

- **play():** Este método comienza a reproducir el medio desde el inicio, a menos que el medio haya sido pausado previamente.
- **pause():** Este método pausa la reproducción.
- **load():** Este método carga el archivo del medio. Es útil en aplicaciones dinámicas para cargar el medio anticipadamente.
- **canPlayType(formato):** Con este método podemos saber si el formato del archivo es soportado por el navegador o no.

La función *presionar()* también usa unas pocas propiedades para recabar información sobre el medio:

- **paused:** Esta propiedad retorna true (verdadero) si la reproducción del medio está actualmente pausada o no a comenzado.
- **ended:** Esta propiedad retorna true (verdadero) si la reproducción del medio ha finalizado porque se llegó al final.
- **duration:** Esta propiedad retorna la duración del medio en segundos.
- **currentTime:** Esta es una propiedad que puede retornar o recibir un valor para informar sobre la posición en la cual el medio está siendo reproducido o especifica una nueva posición donde continuar reproduciendo.
- **error:** Esta propiedad retorna el valor del error ocurrido.
- **buffered:** Esta propiedad ofrece información sobre la parte del archivo que ya fue cargada en el buffer. Nos permite crear un indicador para mostrar el progreso de la descarga. La propiedad es usualmente leída cuando el evento **progress** es disparado. Debido a que los usuarios pueden forzar al navegador a cargar el medio desde diferentes posiciones en la línea de tiempo, la información retornada por **buffered** es un array conteniendo cada parte del medio que ya fue descargada, no solo la que comienza desde el principio. Los elementos del array son accesibles por medio de los atributos **end()** y **start()**. Por ejemplo, el código `buffered.end(0)` retornará la duración en segundos de la primera porción del medio encontrada en el buffer. Esta propiedad y sus atributos están bajo desarrollo en este momento.

La función **reproducir()** es ejecutada cuando el usuario presiona el botón *Reproducir* en nuestro reproductor. Este botón tendrá dos propósitos: mostrará el mensaje *Reproducir* para reproducir el video o *Pausa* para detenerlo, de acuerdo a las circunstancias. Por lo tanto, cuando el video fue pausado o no comenzó, presionar este botón comenzará o continuará la reproducción. Lo opuesto ocurrirá si el video está siendo reproducido, entonces presionar el botón pausará el video.

Para lograr esto el código detecta la situación del medio comprobando el valor de las propiedades **paused** y **ended**. En la primera línea de la función tenemos un condicional *if* para este propósito. Si el valor de *medio.paused* y *medio.ended* es falso, significará que el video está siendo reproducido, entonces el método **pause()** es ejecutado para pausar el video y el texto del botón es cambiado a *Reproducir* usando `innerHTML`.

Si lo opuesto ocurre, el video fue pausado previamente o terminó de ser reproducido, entonces la condición será falsa (*medio.paused* o *medio.ended* es verdadero) y el método **play()** es ejecutado para comenzar o restaurar la reproducción del video. En este caso también realizamos una importante acción que es configurar un intervalo usando **setInterval()** para ejecutar la función **estado()** una vez por segundo.

```
1 function estado(){
2   if(!medio.ended){
3     var total=parseInt(medio.currentTime*maximo/medio.duration);
4     progreso.style.width=total+'px';
5   }else{
6     progreso.style.width='0px';
7     reproducir.innerHTML='Reproducir';
8     window.clearInterval(bucle);
9   }
10 }
```

La función **estado()** es ejecutada cada segundo mientras el video es reproducido. También se utiliza un condicional *if* en esta función para controlar el estado del video. Si la propiedad **ended** retorna falso, se calcula qué tan larga la barra de progreso debe ser en pixeles y se asigna el valor al elemento `<div>` que la representa.

En caso de que la propiedad sea verdadera (lo cual significa que la reproducción del video ha terminado), se retorna el valor de la barra de progreso a 0 píxeles, y el valor del botón cambia a *Reproducir*, y se cancela el intervalo usando *clearInterval*. En este caso la función *estado()* no será ejecutada nunca más.

A continuación se prepara la función que responde a los clics sobre la barra de progreso:

```
1 function mover(e){
2   if(!medio.paused && !medio.ended){
3     var ratonX=e.pageX-barra.offsetLeft;
4     var nuevoTiempo=ratonX*medio.duration/maximo;
5     medio.currentTime=nuevoTiempo;
6     progreso.style.width=ratonX+'px';
7   }
8 }
```

Al elemento de la barra se le agrega un escuchador para el evento clic. Este escuchador utiliza la función *mover()* para responder al evento cuando es disparado. Esta función comienza con una comprobación de que el vídeo se está reproduciendo. A continuación se calcula el tiempo en el cual el vídeo debería comenzar a ser reproducido. Para ello se determina cuál era la posición del ratón cuando el clic sobre la barra fue realizado, cuál es la distancia en píxeles desde esa posición hasta el comienzo de la barra de progreso y cuantos segundos esa distancia representa en la línea de tiempo.

Los procesos para agregar una escucha (o registrar un evento), tales como *addEventListener()*, siempre envían un valor que hacer referencia al evento. Esta referencia es enviada como un atributo a la función que responde al evento.

Tradicionalmente la variable *e* es usada para almacenar este valor. La propiedad *pageX* sirve para capturar la posición exacta del puntero del ratón al momento en el que el clic fue realizado. El valor retornado por *pageX* es relativo a la página, no a la barra de progreso o la ventana. Para saber cuántos píxeles hay desde el comienzo de la barra de progreso y la posición del puntero, se tiene que substrair el espacio entre el lado izquierdo de la página y el comienzo de la barra (Recuerde que la barra está localizada en una caja que se encuentra centrada en la ventana). Los valores dependerán de cada situación en particular. Una vez obtenido este valor, se debe convertir a segundos. Usando la propiedad *duration*, la posición exacta del puntero del ratón en la barra y el tamaño máximo de la barra se construye la fórmula *ratonX * video.duration / maximo* y se almacena el resultado dentro de la variable *nuevoTiempo*. Este resultado es el tiempo en segundos que la posición del puntero del ratón representa en la línea de tiempo.

El siguiente paso es comenzar a reproducir el video desde la nueva posición. La propiedad *currentTime*, como ya se mencionó, retorna la posición actual del video en segundos pero también avanza o retrocede el video a un tiempo específico si un nuevo valor le es asignado. Con el código *medio.currentTime=nuevoTiempo* se mueve el video a la posición deseada.

Lo único que resta por hacer es cambiar el tamaño del elemento progreso para reflejar en pantalla la nueva situación. Utilizando el valor de la variable *ratonX* se cambia el tamaño del elemento para alcanzar exactamente la posición donde el clic fue hecho.

El código para nuestro reproductor de video ya está casi listo. Solo hay una cosa más que se debe hacer, es registrar el evento *load* con el método *iniciar*.

```
1 window.addEventListener('load', iniciar, false);
```

Ejercicio 4.18 Construye un reproductor de vídeo en HTML5 que se adapte perfectamente al diseño de tu Web.



Figura 4.12: Ejercicio: Construye un reproductor HTML5.

Ejercicio 4.19 Realice las siguientes tareas:

1. Descargue las siguientes bibliotecas/kits pensados para facilitar la tarea del desarrollo y personalización de vídeo en HTML5:
 - **Mediafront:** <http://mediafront.org/>
 - **Video JS:** <http://videojs.com/>
2. Hágalos funcionar en su sitio Web con la configuración base.
3. Hágalos funcionar en su sitio Web realizando cambios en la configuración.
4. Razóne cuándo y por qué deben ser utilizadas estas bibliotecas externas.

Ejercicio 4.20 Construye un reproductor personalizado (sin usar bibliotecas externas) que tenga un aspecto y funcionalidad similar a la mostrada en la Figura 4.12.

Ejercicio 4.21 Construye el reproductor con las funcionalidades mostradas en la Web: <http://www.w3.org/2010/05/video/mediaevents.html>.

4.4.3 Reproduciendo audio en HTML5

El audio no es un medio tan popular como video en Internet. No obstante el audio está ganando su propio mercado en shows de radio y podcasts en toda la red. HTML5 provee un nuevo elemento para reproducir **<audio>**.

```

1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <title>Reproductor de Audio</title>
5   </head>
6   <body>
7     <section id="reproductor">
8       <audio src="http://minkbooks.com/content/beach.mp3" controls></
          audio>
9     </section>
10  </body>
11 </html>

```

El elemento `<audio>` trabaja del mismo modo y comparte varios atributos con el elemento `<video>`:

- **src:** Este atributo especifica la URL del archivo a ser reproducido. Al igual que en el elemento `<video>` normalmente será reemplazado por el elemento `<source>` para ofrecer diferentes formatos de audio entre los que el navegador pueda elegir.
- **controls:** Este atributo activa la interfaz que cada navegador provee por defecto para controlar la reproducción del audio.
- **autoplay:** Cuando este atributo está presente, el audio comenzará a reproducirse automáticamente tan pronto como sea posible.
- **loop:** Si este atributo es especificado, el navegador reproducirá el audio una y otra vez de forma automática.
- **preload:** Este atributo puede tomar tres valores diferentes: *none*, *metadata* o *auto*. El primero indica que el audio no debería ser cacheado, normalmente con el propósito de minimizar tráfico innecesario. El segundo valor, *metadata*, recomendará al navegador obtener información sobre el medio (por ejemplo, la duración). El tercer valor, *auto*, es el valor configurado por defecto y le aconseja al navegador descargar el archivo tan pronto como sea posible.

Del mismo modo que para el video se debe tener en cuenta que todos los navegadores Web no reproducen todos los formatos y es por ello que hay que utilizar el elemento `<source>` tal y como se muestra en el siguiente código:

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <title>Reproductor de Audio</title>
5   </head>
6   <body>
7     <section id="reproductor">
8       <audio id="medio" controls>
9         <source src="http://minkbooks.com/content/beach.mp3">
10        <source src="http://minkbooks.com/content/beach.ogg">
11      </audio></section>
12    </body>
13  </html>
```

4.4.4 Construir un reproductor de audio en HTML 5

La API para medios fue desarrollada tanto para video como para audio. Cada evento, método y propiedad incorporada para video funcionará también con audio. Debido a esto, solo necesitamos reemplazar el elemento `<video>` por el elemento `<audio>` en nuestra plantilla e instantáneamente obtenemos un reproductor de audio tal y como se muestra en el siguiente código:

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <title>Reproductor de Audio</title>
5     <link rel="stylesheet" href="reproductor.css">
6     <script src="reproductor.js"></script>
7   </head>
8   <body>
```

```

9   <section id="reproductor">
10   <audio id="medio">
11       <source src="http://minkbooks.com/content/beach.mp3">
12       <source src="http://minkbooks.com/content/beach.ogg">
13   </audio>
14   <nav>
15       <div id="botones">
16           <button type="button" id="reproducir">Reproducir</button>
17       </div>
18       <div id="barra">
19           <div id="progreso"></div>
20       </div>
21       <div style="clear:_both"></div>
22   </nav>
23   </section>
24 </body>
25 </html>

```

No es necesario modificar nada más puesto que todo lo desarrollado en la sección anterior es más que suficiente.

1 Basado en los reproductores de video desarrollados en los ejercicios anteriores debe realizar un reproductor de audio personalizado.

4.5 Animaciones

En la actualidad existen diferentes maneras para interactuar con animaciones en la Web. Entre ella existen soluciones como son FLASH, SilverLight o JAVA FX. No obstante, en este curso se ha preferido utilizar el estándar CSS3, el cual permite crear verdaderos efectos y animaciones que están a la altura de cualquier otra tecnología teniendo como principal ventaja su estandarización.

Esta sección es dividida en tres secciones:

- **Transformaciones 2D y 3D.**
- **Transiciones.**
- **Animaciones.**

4.5.1 Transformaciones en 2D

Las transformaciones de un elemento utilizando sólo CSS son uno de los proyectos de la versión 3 que aún se discute y que algunos navegadores ya implementan con ciertas variantes. En teoría, será tan simple como escribir cualquier otra propiedad:

```
1 transform: funcion(parámetros);
```

En los navegadores se usan de este modo:

```

1 -moz-transform: funcion(parámetros); // en Mozilla
2 -webkit-transform: funcion(parámetros); // en Safari y Chrome
3 -o-transform: funcion(parámetros); // en Opera

```

En principio, hay varias funciones:

- **scale(x,y):** Aumenta o disminuye el tamaño del elemento; su valor normal es 1. Valores superiores aumentan su tamaño e inferiores lo disminuyen así, 1.5 hará que se vea un 50 % más grande y 0.5 un 50 % más chico; si sólo se coloca un valor, este se aplica a ambas direcciones, caso contrario, el primero indica el ancho y el segundo el alto. Además, se

dispone de la posibilidad de utilizar **scaleX(x)** y **scaleY(y)** que solamente afectan al ancho (X) o al alto (Y).

En IE es posible utilizarla con bastante facilidad cuando se trata de cosas simples porque el resultado no va a ser el mismo, para eso, es necesario que el elemento sea un layout y lo conseguimos adicionándole la propiedad zoom: 1; que es la misma que usaremos para luego cambiarle la escala:

```
1 zoom: 1; el elemento no cambia
2 zoom: 1.5; aumentar su tamaño
3 zoom: 0.5; disminuye su tamaño
```

En el siguiente código se muestra el fragmento de CSS3 que corresponde con el escalado de diversos elementos (en los que se encuentran imágenes) cuando se sitúa el cursor sobre dicho elemento. Es realmente interesante observar el resultado en un documento HTML y comenzar a ver como la Web interactúa fácilmente con el usuario.

```
1 #DTI-scale1:hover {
2   -moz-transform:scale(1.2);
3   -webkit-transform:scale(1.2);
4   -o-transform:scale(1.2);}
5 #DTI-scale2:hover {
6   -moz-transform:scale(0.8);
7   -webkit-transform:scale(0.8);
8   -o-transform:scale(0.8);}
9 #DTI-scale3:hover {
10   -moz-transform:scale(1.5,2);
11   -webkit-transform:scale(1.5,2);
12   -o-transform:scale(1.5,2);}
13 #DTI-scale4:hover {
14   -moz-transform:scale(2,1.5);
15   -webkit-transform:scale(2,1.5);
16   -o-transform:scale(2,1.5);}
17 #DTI-scale5:hover {
18   -moz-transform:scaleX(2);
19   -webkit-transform:scaleX(2);
20   -o-transform:scaleX(2);}
21 #DTI-scale6:hover {
22   -moz-transform:scaleY(2);
23   -webkit-transform:scaleY(2);
24   -o-transform:scaleY(2);}
25 #DTI-scale7:hover {
26   -moz-transform:scale(1.2);
27   -webkit-transform:scale(1.2);
28   -o-transform:scale(1.2);
29   zoom:1.5;}
30 #DTI-scale8:hover {
31   -moz-transform:scale(0.8);
32   -webkit-transform:scale(0.8);
33   -o-transform:scale(0.8);
34   zoom:0.5;}
```

- **rotate(a):** Gira un elemento. El valor se expresa en grados, si es positivo gira en el sentido de las agujas del reloj, si es negativo, en el sentido contrario.

```
1 -moz-transform:rotate(10deg); -webkit-transform:rotate(10deg); -o-
  transform:rotate(10deg);
```

En IE puede usarse algún filtro que calculamos usando esta herramienta; por ejemplo:

```
1 -moz-transform:rotate(5deg);
```

```

2 -webkit-transform:rotate(5deg);
3 -o-transform:rotate(5deg);
4 filter: progid:DXImageTransform.Microsoft.Matrix(sizingMethod='auto
   expand', M11=0.9961946980917455, M12=-0.08715574274765817, M21=0
   .08715574274765817, M22=0.9961946980917455);
5 zoom: 1;

```

Al igual que el ejemplo mostrado para la anterior operación, en esta ocasión se realizan diferentes rotaciones en varios elementos:

```

1 #DTI-rotate1: hover {
2   -moz-transform:rotate(5deg);
3   -webkit-transform:rotate(5deg);
4   -o-transform:rotate(5deg);
5   filter: progid:DXImageTransform.Microsoft.Matrix(sizingMethod='
   auto expand', M11=0.9961946980917455, M12=-0.08715574274765817,
   M21=0.08715574274765817, M22=0.9961946980917455);}
6 #DTI-rotate2: hover {
7   -moz-transform:rotate(-5deg);
8   -webkit-transform:rotate(-5deg);
9   -o-transform:rotate(-5deg);
10  filter: progid:DXImageTransform.Microsoft.Matrix(sizingMethod='
   auto expand', M11=0.9961946980917455, M12=0.08715574274765817,
   M21=-0.08715574274765817, M22=0.9961946980917455);}

```

- **skew(a, b):** Skew significa sesgar o sea, inclinar y eso hace esta propiedad que también utiliza ángulos como valor, el primero afecta al eje X (horizontal) y el segundo al eje Y (vertical). Además, se dispone de las variantes **skewX(a)** y **skewY(a)** que hacen lo mismo pero sólo sobre uno de los ejes.

```

1 -moz-transform:skew(5deg); -webkit-transform:skew(5deg); -o-
   transform:skew(5deg);

```

```

1 #DTI-skew1: hover {
2   -moz-transform:skew(10deg);
3   -webkit-transform:skew(10deg);
4   -o-transform:skew(10deg);}
5 #DTI-skew2: hover {
6   -moz-transform:skew(-10deg);
7   -webkit-transform:skew(-10deg);
8   -o-transform:skew(-10deg);}
9 #DTI-skew3: hover{
10  -moz-transform:skew(20deg,-10deg);
11  -webkit-transform:skew(20deg,-10deg);
12  -o-transform:skew(20deg,-10deg);}
13 #DTI-skew4: hover{
14  -moz-transform:skew(-10deg,20deg);
15  -webkit-transform:skew(-10deg,20deg); -o-transform:skew(-10deg,20
   deg);}
16 #DTI-skew5: hover{
17  -moz-transform:skewX(10deg);
18  -webkit-transform:skewX(10deg);
19  -o-transform:skewX(10deg);}
20 #DTI-skew6: hover{
21  -moz-transform:skewY(10deg);
22  -webkit-transform:skewY(10deg);
23  -o-transform:skewY(10deg);}

```

- **translate(x, y):** Desplaza el elemento, el primer valor lo hace en el eje horizontal y el segundo en el eje vertical. **translateX(x)** y **translateY(y)** hacen lo mismo pero sólo sobre uno de los ejes.

```
1 -moz-transform:translate(20px); -webkit-transform:translate(20px); -
  o-transform:translate(20px);
```

```
1 #DTI-translate1:hover{
2   -moz-transform:translate(20px,-10px);
3   -webkit-transform:translate(20px,-10px);
4   -o-transform:translate(20px,-10px);}
5 #DTI-translate2:hover{
6   -moz-transform:translateX(100px);
7   -webkit-transform:translateX(100px);
8   -o-transform:translateX(100px);}
9 #DTI-translate3:hover{
10  -moz-transform:translateY(20px);
11  -webkit-transform:translateY(20px);
12  -o-transform:translateY(20px);}
```

- **matrix(a, c, b, d, x, y):** Es una combinación de todos ellos, por ejemplo:

```
1 matrix (1, 0, 0, 1, x, y) equivale a translate x e y
2 matrix (x, 0, 0, y, 0, 0) equivale a scale x e y
3 matrix (1, y, x, 1, 0, 0) equivale a skew x e y
```

Por defecto, un elemento sin deformaciones tendría esta propiedad: `matrix(1, 0, 0, 1, 0px, 0px)` así que en este ejemplo:

```
1 matrix(1.5, 0.1, 0.8, 1.2, -10px, 0px):
```

- 1.5 es la escala X.
- 0.1 es la inclinación sobre el eje X.
- 0.8 es la inclinación sobre el eje Y.
- 1.2 es la escala Y.
- -10px es el desplazamiento horizontal.
- 0px es el desplazamiento vertical.

```
1 #DTI-matrix:hover {
2   -moz-transform:matrix(1.5, 0.1, 0.8, 1.2, -10px, 0px);
3   -webkit-transform:matrix(1.5, 0.1, 0.8, 1.2, -10px, 0px);
4   -o-transform:matrix(1.5, 0.1, 0.8, 1.2, -10px, 0px);
5 }
```

Ejercicio 4.22 Desarrolla el efecto mostrado en la Figura 4.13 en el cual al situar el ratón sobre cada uno de los diferentes elementos estos se transformarán y cambiarán el color. Al situar el ratón sobre el elemento activar se realizarán las transformaciones de todos los elementos de un modo similar al mostrado en la Figura.

Ejercicio 4.23 Investiga a fondo cuál es el funcionamiento de `transform-origin`. ¿Para qué sirve? Construye un ejemplo.

Ejercicio 4.24 Utilizando la herramienta online http://ie.microsoft.com/testdrive/graphics/hands-on-css3/hands-on_2d-transforms.htm para generar un estilo con las propiedades de `transform`.

Ejercicio 4.25 Investiga a fondo cuál es el funcionamiento de la operación `Matrix`. Entiéndela, explícala y construye algún ejemplo.

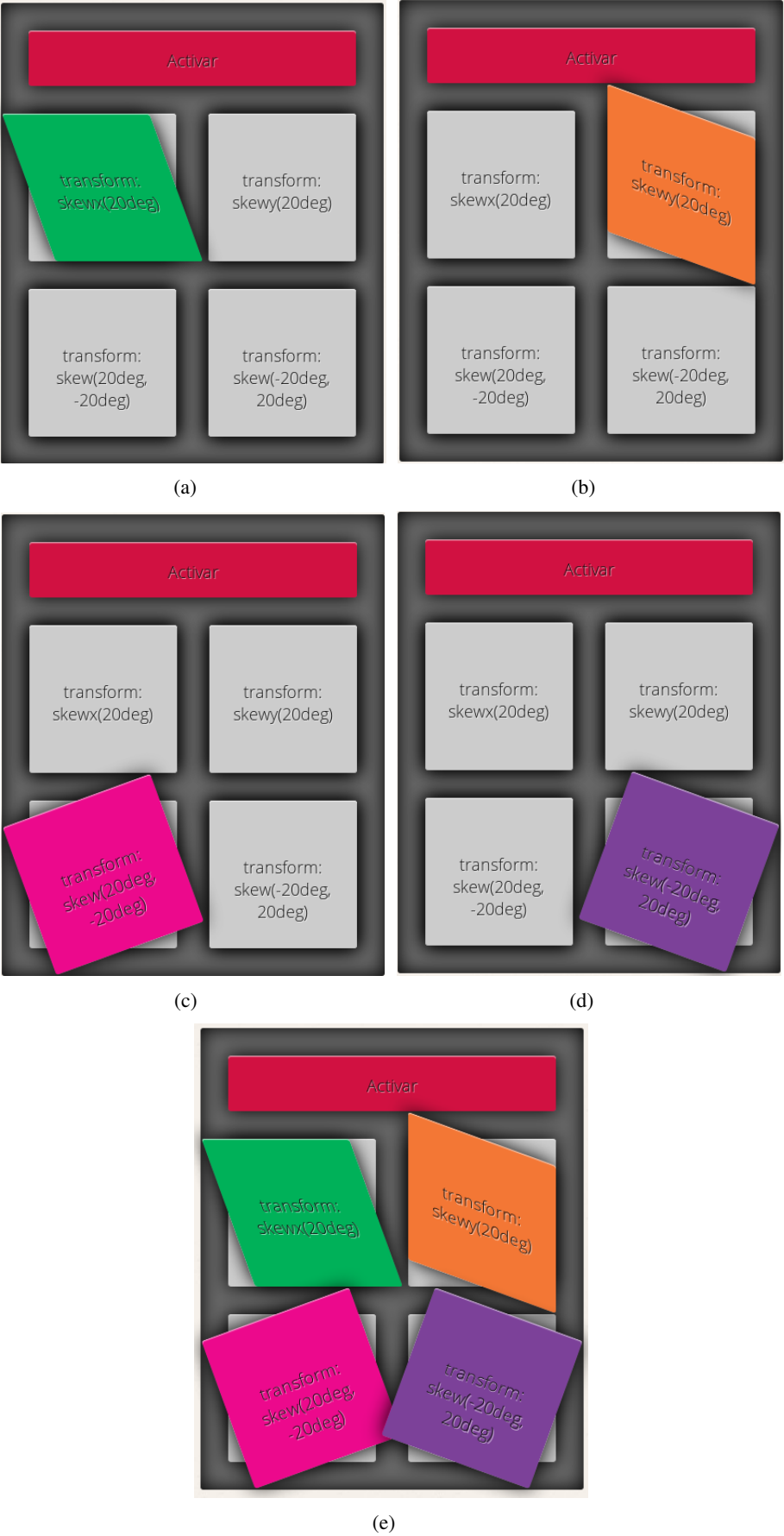


Figura 4.13: Ejercicio CSS3: Efecto de transform.

4.5.2 Transformaciones en 3D

CSS3 permite interactuar además con efectos de transformación 3D. En la descripción de esta sección se va a describir la construcción de las seis caras de un cubo en 3D. De este modo el documento HTML que servirá de base será el siguiente:

```
1 <div class="cubo">
2   <div class="cara" id="superior"></div>
3   <div class="cara" id="frente"></div>
4   <div class="cara" id="derecha"></div>
5   <div class="cara" id="izquierda"></div>
6   <div class="cara" id="atras"></div>
7   <div class="cara" id="inferior"></div>
8 </div>
```

Observe que cada cara ha sido identificada con un *id* diferente para poder controlar cada una de manera individual y una clase en común para generar el CSS básico (*cubo*).

Tanto el cubo, como cada cara, tendrán un ancho y alto igual, así como cada cara tendrá una posición absoluta (para poder transformarlo con libertad). Y alinearemos también el cubo al centro para este ejemplo. Así que, el estilo básico para el cubo es el siguiente:

```
1 .cubo, .cubo .cara {
2   width: 200px;
3   height: 200px;
4 }
5 .cubo .cara { position: absolute;}
6 .cubo { margin: 100px auto;}
7 .cubo .cara#superior {background-color: orange;}
8 .cubo .cara#frente {background-color: red;}
9 .cubo .cara#atras {background-color: green;}
10 .cubo .cara#derecha {background-color: blue;}
11 .cubo .cara#izquierda {background-color: yellow;}
12 .cubo .cara#inferior {background-color: fuchsia;}
```

Para definir que un elemento va a someterse a transformaciones en 3D hay que utilizar el atributo *transform-style* especificando el valor *preserve-3d*. Para conseguir una perspectiva interesante se añade la propiedad *transform* con rotaciones.

```
1 .cubo {
2   transform-style: preserve-3d;
3   transform: perspective(700px) rotateX(-30deg) rotateY(-30deg);
4 }
```

Una vez que se ha definido el conjunto del cubo en una perspectiva es necesario ir colocando cada una de las caras en 3D para que formen el cubo.

```
1 .cubo .cara#superior {
2   background-color: orange;
3   transform: rotateX(90deg) translateZ(100px);
4 }
5 .cubo .cara#frente {
6   background-color: red;
7   transform: translateZ(100px);
8 }
9 .cubo .cara#atras {
10  background-color: green;
11  transform: translateZ(-100px);
12 }
```

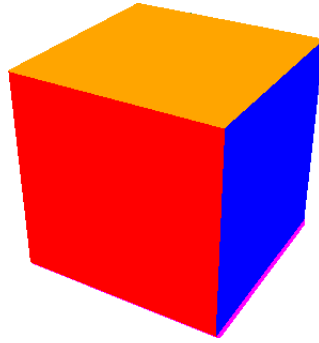


Figura 4.14: Ejercicio CSS3: Efecto de transform 3D.

```
13 .cubo .cara#derecha {  
14     background-color: blue;  
15     transform: rotateY(90deg) translateZ(100px);  
16 }  
17 .cubo .cara#izquierda {  
18     background-color: yellow;  
19     transform: rotateY(-90deg) translateZ(100px);  
20 }  
21 .cubo .cara#inferior {  
22     background-color: fuchsia;  
23     transform: rotateX(-90deg) translateZ(100px);  
24 }
```

El resultado final es el mostrado en la Figura 4.14.

4.5.3 Transiciones

Las transiciones permiten modificar una propiedad CSS de un estado a otro de un modo gradual y no en un salto brusco.

Hay 4 propiedades para modificar las transiciones:

- **transition-property:** Sirve para indicar la propiedad en la que se quiere aplicar la transición. Ej. `transition-property: width;`
- **transition-duration:** Indica el tiempo que dura la transición (Por defecto 0s). Ej. `transition-duration: 5s;`
- **transition-timing-function:** Para indicar como varia la velocidad de la transición. Puede tomar los siguientes valores: `linear`, `ease`, `ease-in`, `ease-out`, `ease-in-out`, `cubic-bezier(x,x,x,x)` (Por defecto `ease`). Ej. `transition-timing-function: linear;`
- **transition-delay:** Indica el tiempo que pasa hasta que comienza la transición (Por defecto 0s). Ej. `transition-delay: 2s;`
- **transition:** Para unificar las cuatro propiedades en una sola. Ej. `transition: width 5s linear 2s;`

Las diferentes funciones que se pueden especificar en el parámetro *transition-timing-function* son las siguientes:

- **linear:** La transición se produce en forma de progresión lineal. Tarda lo mismo desde que empieza hasta que acaba.
- **ease:** La transición se produce de forma lenta al principio, luego más rápido, y termina otra vez lenta (valor por defecto).
- **ease-in:** La transición comienza lenta y luego va más rápido.

- **ease-out:** La transición comienza rápida y termina lenta.
- **ease-in-out:** la transición comienza y acaba lenta. Igual que 'ease' pero de forma más notable.
- **cubic-bezier(n,n,n,n):** Permite definir nuestro propio intervalo de progresión. n: valor entre 0 y 1.

El uso de las transiciones se debe definir en el elemento origen puesto que en el destino (final de la transición) solamente se debe definir el estado final que deberá tener la transición. En el siguiente código se muestra como se hace crecer un elemento <div> en su propiedad *width* una vez que se sitúa el cursor sobre el div.

```

1 #div1 {
2   width: 50px;
3   height: 50px;
4   background: orange;
5   transition: width 2s;
6 }
7
8 #div1:hover {
9   width: 200px;
10 }
```

Los efectos de transición dan un juego increíble en la creación de efectos en la Web y ha sido una de las propiedades que ha explotado en mayor medida el uso de CSS3. Basados en el ejemplo anterior, si se desean realizar diversas transiciones como pueden ser el color, una transformación, tamaño, etc.. Se puede definir como muestra el siguiente código:

```

1 .animated_div{
2   background:#E11E52;
3   border-radius: 5px 5px 5px 5px;
4   color: #FFFFFF;
5   font-size: 15px;
6   font-weight: bold;
7   height: 120px;
8   margin: 5px;
9   opacity: 0.4;
10  padding: 10px;
11  transition-duration: 1s, 1s, 1s, 1s, 1s, 1s;
12  transition-property: width, height, transform, background, opacity;
13  width: 120px;
14 }
15 .animated_div:hover {
16   background:red;
17   font-size: 30px;
18   height: 110px;
19   opacity: 1;
20   transform: rotate(360deg);
21   width: 170px;
22 }
```

Otro interesante ejemplo sería aquel en el cual se dispone de un conjunto de pestañas y lo que se construye es un efecto en el cual cuando el cursor del usuario se sitúa en cada pestaña está cambiando de color y el sombreado de las mismas de un modo suave.

```

1 .MENU_pestañas li{
2   position:relative;
3   float:left;
4   list-style:none;
5   width:140px;
6   height:26px;
```

```

7 padding-top:15px;
8 text-align:center;
9 background:#999;
10 color:#FFF;
11 border:1px solid #555;
12 font-weight:bold;
13 text-shadow:1px 1px 0px #000;
14 border-radius:10px 10px 0px 0px;
15 -webkit-border-radius:10px 10px 0px 0px;
16 -moz-border-radius:10px 10px 0px 0px;
17 -webkit-transition:all 1s ease 1s;
18 -moz-transition:all 1s ease 1s;
19 -o-transition:all 1s ease 1s;
20 transition:all 1s ease 1s;
21 }
22 .MENU_pestañas li:hover{
23 background:#9A44A6;
24 border-bottom-color:transparent;
25 z-index:2;
26 box-shadow:2px -2px 4px rgba(0,0,0,0.4);
27 -webkit-transition:all 1s ease;
28 -moz-transition:all 1s ease;
29 -o-transition:all 1s ease;
30 transition:all 1s ease;
31 }

```

Es interesante los efectos que se pueden conseguir cuando se coloca el cursor sobre el elemento. En la dirección: <http://designshack.net/articles/css/5-cool-css-hover-effects-you> se describen cinco efectos básicos para *hover*:

- **Bump up:** Excelente para usar una lista de imágenes horizontal, lo que hace es elevar el elemento modificando el margin-top.
- **Stack & Grow:** Al pasar el cursor por encima de la imagen, esta aumenta su anchura y altura mediante una transición CSS.
- **Fade Text in:** Este efecto es bastante interesante, es un div que contiene una imagen y un texto, primero se alinea la imagen a la izquierda y al div contenedor se le definen las propiedades line-height: 0px y color: transparent, para que cuando el cursor pase por la imagen, esto cambie de manera animada, dando el efecto que el texto cae.
- **Crooked Photo:** Ideal para una galería de fotos, este efecto usa la propiedad CSS3 - transform: rotate, hace que cuando el cursor pase por la imagen esta gire 7 grados a la izquierda.
- **Fade In and Reflect:** Este efecto solo es compatible con navegadores -webkit, ya que usa la propiedad -webkit-box-reflect, que solo es soportada por Chrome y Safari, de igual manera se ve bastante bien.

Rotación 3D

Otro interesante efecto es la rotación en 3D en un efecto flip en la cual se realiza un giro en 3D del elemento.

Esto se hace con la función *translate* de la propiedad *transform*, con la que podemos cambiar la posición del elemento. Lo que se va a construir son tres efectos diferentes:

- Rota desde el centro.
- Rota desde el lateral y se desplaza.

- Rota desde el lateral.

La diferencia está en que el elemento del centro realiza la rotación sin salirse de su posición inicial.

```
1 /* Primer bloque */
2   transform:_perspective(300px)_rotateY(-180deg);
3   transform-origin:_50%_50%;
4   /* Segundo bloque */
5   transform:_perspective(300px)_rotateY(-180deg)_translationX(100%);
6   transform-origin:_100%_50%;
7   /* Tercer bloque */
8   transform:_perspective(300px)_rotateY(-180deg);
9   transform-origin:_100%_50%;
```

4.5.4 Animaciones

Las animaciones de CSS3 son la herramienta perfecta para animar cualquier elemento, para crear un efecto puntual o para crear un movimiento continuo de algún elemento (es decir, se pueden mover varios elementos de forma continuada pudiendo crear animaciones que antes solo se veían en flash o con JavaScript). Las animaciones usan el parámetro *animation* en CSS, para su uso estándar, por lo que es más correcto (por cuestiones de compatibilidad) también incluir el parámetro de cada navegador. La mayoría de los navegadores son compatible, pero *Internet Explorer* no (en su versión 9).

```
1 /* Parámetro estándar */
2   animation:
3   /* Parámetros según navegador */
4   -webkit-animation:/* Chrome y Safari */
5   -moz-animation:/* Mozilla */
```

Como funciona

Las animaciones de CSS3 funcionan distinto a otros parámetros de CSS, debemos de establecer en el parámetro *animation*, además de la duración de la animación, y el número de repeticiones que hace, un *keyframes*, que es el fragmento de código donde se escribe la secuencia que hará el elemento con esta animación (es decir, se especifica que propiedades tendrá en cada momento, pudiendo modificar por ejemplo: el color, la altura, la posición...).

Como usarlo

El *keyframes* es una propiedad de CSS3 que va independiente de los selectores (que son el conjunto de palabras que usamos para seleccionar elemento de una página, como: *#CONTENEDOR div.cuerpo*, esta propiedad también usa los prefijos de ciertos navegadores por motivos de compatibilidad. Esta propiedad precede a un contenedor que se abre y cierra con {}, es decir pondremos la propiedad *keyframes* y dentro de las llaves su contenido.

```
1   /* Parámetro estándar */
2   @keyframes [Nombre del keyframes] {}
3
4   /* Parámetros según navegador */
5   @-webkit-keyframes [Nombre del keyframes] {}/* Chrome y Safari */
6   @-moz-keyframes [Nombre del keyframes] {}/* Mozilla */
7
8   /* Ejemplo de un keyframes */
9   @keyframes Ejemplo {
10      0%{width:100px}
```

```

11     100%{width:200px}
12 }

```

Dentro del keyframes se debe especificar los keyframe de forma individual, donde se especifica el porcentaje del tiempo de la animación en el que cierta propiedad cambia y posteriormente se especifican las propiedades que cambian escribiéndolas dentro de llaves. Si queremos que empiece la animación de determinada forma se debe añadir un keyframe con el porcentaje del tiempo de la animación en el que empieza (en el caso del inicio es el 0%), de esta forma podemos especificar que en cualquier momento de la animación cambien los parámetros (en el 30 % de la animación o en el 50 % por ejemplo), hasta máximo del 100%. **Los puntos 0 % y el 100 % son necesarios.**

A continuación se describe una animación en la cual el elemento va cambiando su tamaño progresivamente.

```

1 @keyframes Ejemplo {
2     0%{width:10%}
3     33%{width:20%}
4     66%{width:60%}
5     100%{width:10%}
6 }

```

La propiedad *animation* puede contener distintos parámetros, como el nombre del keyframes que especifica su animación, su duración, el número de repeticiones y la función de tiempo.

```

1 animation: [Nombre del keyframes] [Tiempo] [Número de repeticiones] [
    Función de Tiempo];

```

La propiedad *animation* tiene los siguientes parámetros:

- **Nombre del keyframes:** En este parámetro es donde establecemos el nombre del keyframes, teniendo que ser igual al keyframes que usemos para especificar la animación.
- **Tiempo:** Este parámetro establece el tiempo que tarda en hacerse la animación en segundos (su unidad es *s*), es decir si queremos que dure un segundo tenemos que poner *1s*.
- **Repeticiones:** Con este parámetro se puede especificar con un número entero las veces que se repite la animación o hacer que se repita infinitamente estableciéndolo como *infinite*.
- **Función de tiempo:** Este parámetro establece la curva de aceleración de la transición, es decir, si empieza más rápido y o más despacio o es un avance lineal, sin variar la aceleración.

Este parámetro tiene varias funciones de tiempo preestablecida: lineal, ease, ease-in... Por defecto usa el ease, y es el más recomendado, consiste en un comience lento y una terminación lenta, acelerándose en la parte media.

En el primer ejemplo se va a definir un elemento `<div>` que se cambia de posición, aumenta y disminuye de tamaño y cambia de color.

```

1 <div class="ANI_cont">
2   <div> <!-- Este DIV es el que animamos -->
3   </div>
4 </div>

```

El código CSS será el siguiente:

```

1 /* Este es el estilo del DIV que esta en el interior de div.ANI_cont
2   */
3 .ANI_cont div{

```



```

4      width:10%;
5      height:60px;
6      margin-left:0%;
7      background:#0F0;
8      animation: Ejemplo 10s infinite ease; /* En esta animación se
        especifica que el keyframes se llama "Ejemplo" y durara 10
        segundos infinitamente */
9  }
10  @keyframes Ejemplo {
11      0%{width:10%}
12      33%{width:20%}
13      66%{width:60%}
14      100%{width:10%}
15  }

```

Para crear animaciones algo más complejas, que hagan una función útil suele ser necesario más de un elemento animado, que actúan conjuntamente.

```

1 <div class="ANI_ex1">
2     <div class='progres'></div> <!-- Esta es la barra verde -->
3     <div class='gloss'></div> <!-- Este DIV contiene las barras en
        movimiento -->
4 </div>

```

```

1  /* Este es el estilo del DIV que esta en el interior de div.ANI_cont
   */
2  .ANI_ex1{
3      width:60%;
4      height:20px;
5      margin:auto;
6      background:#999;
7  }
8  .ANI_ex1 .progres{
9      position:absolute;
10     height:100%;
11     margin:auto;
12     background:url(imgs/ANI_ex1_progres.jpg); /* Esta imagen es el
        degradado de la barra */
13     animation:_ANI_pro_10s_infinite_ease;
14 }
15 @keyframes ANI_pro {
16     0%{width:0%}
17     100%{width:100%}
18 }
19 .ANI_ex1 .gloss{
20     position:absolute;
21     width:120%;
22     height:100%;
23     margin-left:-20px;
24     background:url(imgs/ANI_ex1_bars.png); /* Esta imagen muestra las
        barras diagonales que avanzan */
25     animation:_ANI_bars_10s_infinite_ease;
26 }
27 @keyframes ANI_bars {
28     0%{margin-left:-20px}
29     100%{margin-left:0px}
30 }

```



Figura 4.15: Ejercicio CSS3: Efecto de transición.

Ejercicio 4.26 Crear el efecto de la caja de búsqueda de la compañía Apple. En la cual se dispone de una caja de búsqueda en la parte superior con un icono de lupa. Al situar el foco en la caja de búsqueda, ésta crecerá y cambiará de color. El efecto previo y posterior de situar el foco es el mostrado en la Figura 4.15.

Ejercicio 4.27 Realiza la rotación en 3D de un elemento caja en el cual en la cara lateral aparezca una imagen.

Ejercicio 4.28 Realizar animaciones de transición en el texto de diferentes pestañas. Los ejemplos se mostrarán en la clase por el profesor.

Ejercicio 4.29 Realiza un efecto de rotación y transición de manera simultánea para conseguir el mismo efecto de un contador de kilómetros de un coche que aumenta a medida que se mantiene el cursor sobre el elemento.

Ejercicio 4.30 Descarga la biblioteca CSS Animate.css: <http://daneden.me/animate>. Observa todos los efectos que se pueden utilizar con CSS.

1. Utilizando dicha biblioteca haz uso de algunos efectos sobre tu sitio Web.
2. Construye una hoja de estilos CSS que utiliza exclusivamente los efectos que te interesan.
3. Selecciona dos efectos cualesquiera y observando el código fuente, examina y explica cómo están implementado.

Ejercicio 4.31 Documente las siguientes tareas:

1. Instale Google Web Designer.
2. Documente las principales operaciones que se pueden hacer con esta herramienta.
3. Construya animaciones utilizando la barra de tiempo.
4. Exportelas y observe el código CSS3 generado. ¿Es código CSS3 válido?
5. Reflexione sobre el uso de esta herramienta para construir una Web desde cero.

4.6 Prácticas

En esta sección se describen un conjunto de prácticas que debe conseguir realizar el alumno.

4.6.1 Efecto de imágenes de Polaroid

En esta práctica se tiene como objetivo conseguir un mural con fotografías al estilo de las antiguas y desaparecidas cámaras Polaroid. El resultado final, muestra en la Figura de la imagen 4.16 y debe cumplir las siguientes características:

- Debe existir un espacio con una textura de pared (no tiene que ser de madera).
- Deben aparecer chinchetas sujetando las fotografías.
- Las fotografías deben aparecer giradas de diferente modo.
- Al situar el cursor sobre una fotografía esta debe maximizarse y girarse de manera adecuada.

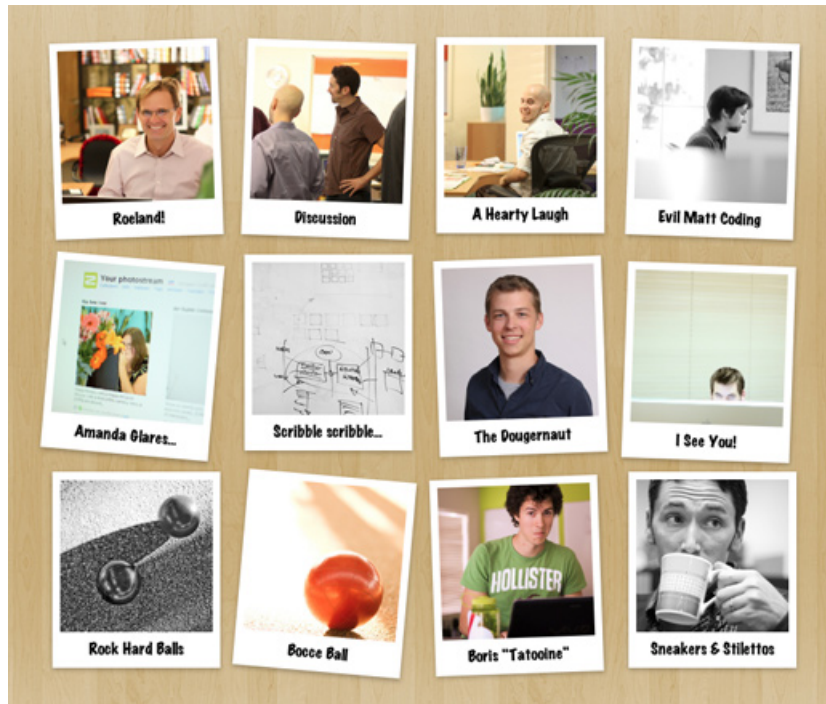


Figura 4.16: Práctica 1: Efecto Polaroid.

4.6.2 Galería con un efecto de *slider*

En esta práctica el alumno debe ser capaz de conseguir desarrollar una galería de imágenes con efecto *slider*. El aspecto que se mostrará en la galería es similar al mostrado en la Figura 4.17. Las características de esta práctica son los siguientes:

- Existirán un elemento que contendrá una fotografía y un menú de botones para cambiar entre las imágenes.
- Al hacer clic en cada uno de los diferentes botones, aparecerá de izquierda a derecha y de menor a mayor opacidad la nueva fotografía hasta situarse en el espacio ocupado por la anterior fotografía. Es decir, la capa principal o primaria del elemento será de la nueva fotografía seleccionada.

4.6.3 Cajas informativas animadas

En esta práctica el alumno deberá incluir animaciones (al margen que transiciones) para conseguir un efecto mucho más dinámico. Se dispone de varias cajas informativas inicialmente que responden del siguiente modo cuando el cursor se sitúa sobre ellas:

- El color de fondo se cambiará a negro.
- Los textos que indica el número de la caja y el proyecto, desaparecerán y aparecerán en una animación cayendo de la parte superior hacia su situación en el estado inicial.
- El texto del contenido desaparecerá y aparecerá de abajo hacia arriba.
- El botón permanecerá inmóvil.

En la imágenes de la Figura 4.18 se muestra el resultado previo y finalizado de la animación. Para visualizar adecuadamente la animación se mostrará en clase por parte del profesor.

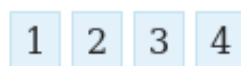
Figura 4.17: Práctica 2: Galería con un efecto de *slider*.

Figura 4.18: Práctica 3: Cajas informativas animadas.



Figura 4.19: Práctica 5: Elementos girando en 3D.

4.6.4 Efectos de círculos

En esta práctica se desarrollarán diferentes efectos en *hover* utilizando transiciones que se mostrarán en clase para un conjunto de círculos.

4.6.5 Efecto 3D girando elementos

En esta práctica se disponen de varios elementos con grandes números en los cuales al pasar el cursor por encima de ellos harán un giro en 3D de abajo hacia arriba mostrando el contenido de un mensaje. En la Figura 4.19 se muestra una de las cajas girando hacia el mensaje oculto. Se debe desarrollar utilizando las transformaciones en 3D.

4.6.6 Presentación de contenido

En esta práctica se desarrollará una presentación de texto y líneas que aparecen y desaparecen en la pantalla con una lógica. Para poder desarrollarla el profesor mostrará la animación varias veces en clase para que se pueda observar lo que se desea realizar.

4.6.7 Cuento digital

Entre parejas construir un cuento (el que os guste) digital con efectos (el aspecto puede ser personalizado) similares al mostrado en la Web: <http://www.20thingsilearned.com/en-US/what-is-the-internet/>