
Apuntes de Multimedia

David Escudero-Mancebo



Apuntes de Multimedia

Apuntes de Multimedia

David Escudero-Mancebo
Departamento de Informática
Universidad de Valladolid

ÍNDICE

Aplicaciones y proyectos multimedia	1
Características básicas	1
Requisitos y preproducción	9
Diseño y producción	21
Posproducción, validación y despliegue	33
Recursos multimedia	45
Imagen digital	45
Audio digital	57
Vídeo digital	65
Texto digital	75
Animaciones	85

Portada generada automáticamente con
<https://github.com/mgiraldo/tenprintcover-py>
Contenidos propiedad del autor
distribuidos con licencia
Creative Commons CC BY-SA.
Tablas e imágenes propiedad de los autores.

Edición 2023

Prólogo

El presente volumen contiene las notas del curso titulado “Técnicas de Presentación Multimedia”, ofrecido a los estudiantes del Grado en Ingeniería en Diseño Industrial y Desarrollo de Producto de la Universidad de Valladolid. Los contenidos se corresponden con la parte teórica de la asignatura. La formación se complementa con seminarios prácticos de manejo de un software de edición de audio, vídeo, animaciones y edición web.

Estos apuntes se ponen a disposición de los alumnos, que acceden a ellos empleando la plataforma virtual de la Universidad. Los contenidos se distribuyen por capítulos. Este volumen ha sido editado para ser puesto a disposición del tribunal de la oposición a la que se presentó el autor 2023.

Introducción a las aplicaciones y proyectos multimedia

OBJETIVOS

El objetivo de este tema es dejar claro a qué nos referimos por aplicaciones y proyectos multimedia para destacar su importancia en las presentaciones de productos. Después de este tema, el estudiante deberá:

- Comprender las especificidades de las aplicaciones y proyectos multimedia.
- Comprender las particularidades del desarrollo de las aplicaciones y proyectos multimedia.
- Comprender la importancia de utilizar aproximaciones sistemáticas a la hora de realizar presentaciones y proyectos multimedia efectivos y eficientes.

ÍNDICE

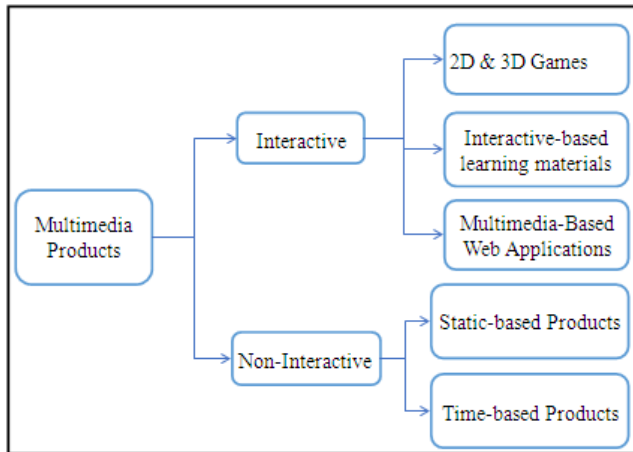
I.	Tipos de productos multimedia	2
II.	Producción de productos multimedia	3
III.	El ciclo de vida del software	3
IV.	El ciclo en el desarrollo de aplicaciones multimedia	3
	IV-A. Preproducción y requisitos	4
	IV-B. Producción y diseño	5
	IV-C. Posproducción, validación	5
	IV-D. Distribución	5
V.	Presentaciones multimedia	6
	V-A. El objetivo, la audiencia y el contexto .	6
	V-B. Los contenidos	6
	V-C. La organización	7
VI.	Conclusiones	8
	Referencias	8

La potencia comunicativa de la multimedia hace que multitud de aplicaciones utilicen estos recursos para ser más eficaces en la interacción con los usuarios. Este tema tiene por objetivo presentar una serie de consideraciones que tienen que ver con el desarrollo eficiente de aplicaciones multimedia. Estas consideraciones tienen que ver tanto con las características básicas de las aplicaciones multimedia, como el desarrollo de material multimedia y de aplicaciones informáticas en general y de aplicaciones multimedia en particular. Se trata pues de un tema en el que se da a conocer abundante terminología de uso común en el ámbito del desarrollo de aplicaciones multimedia.

Por multimedia entendemos cualquier combinación de texto, gráficos, vídeo, audio y animación, en un formato distribuible, con el que los consumidores pueden interactuar utilizando un dispositivo digital. Para que una aplicación o producto digital sea considerado multimedia debe reunir una serie de características que se presentan en la sección I.

La creación de material multimedia debe ser un proceso sistemático organizado en una serie de etapas que responden a una planificación. Lo mismo sucede con el desarrollo de programas informáticos. En la sección II se presenta el ciclo de producción de material multimedia, en la sección III el ciclo de producción de aplicaciones informáticas en general y en la sección IV la adaptación del ciclo de producción de software a la producción de aplicaciones multimedia.

Figura 1: Tipos de productos multimedia según la clasificación descrita en [?]



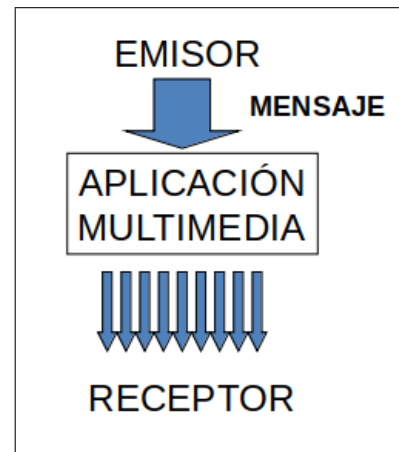
I. TIPOS DE PRODUCTOS MULTIMEDIA

Según [?], los productos multimedia pueden ser clasificados en dos categorías: productos interactivos y productos no interactivos (ver figura 1). Los productos no interactivos, a su vez, pueden ser clasificados en productos estáticos como carteles, logotipos, modelos 3D etc... frente a productos no estáticos que cambian a lo largo del tiempo como animaciones, vídeos, anuncios etc... Los productos multimedia interactivos, o aplicaciones multimedia, son programas informáticos que incluyen productos multimedia como pueden ser los videojuegos, las aplicaciones interactivas o las aplicaciones para la web.

Bauza afirma que para poder considerar que una aplicación es una aplicación multimedia debe cumplir la siguientes características: Múltiple salida, interactividad, libertad, necesidad y vitalidad y atención [1]. Para entender esta posición hay que entender que el autor se basa en el modelo clásico de comunicación entre emisor y receptor (ver figura 2). **Múltiple salida** hace referencia a que el receptor recibe información codificada en canales de diferente tipo: gráficos, vídeo, audio etc... La restricción de **interactividad** implica que el usuario que recibe la información debe poder interactuar con ella. La forma de interactuar puede ser tan simple como hacer click en diversos botones o tan compleja como dialogar con el sistema empleando la voz. El principio de **libertad** hace referencia a que el usuario, además de poder interactuar debe tener la posibilidad de elegir entre más de una opción. En función de la elección del usuario, la aplicación puede evolucionar de formas diferentes. El principio de **necesidad** implica que todos los elementos que aparecen en la aplicación deben responder a un determinado fin, que estará subordinado al mensaje que se quiera transmitir al usuario. Una aplicación multimedia no debe incluir elementos accesorios que despisten sobre el mensaje principal que se quiere transmitir. El último principio de **vitalidad y atención** indica que es importante tener en cuenta la potencia visual de los distintos elementos que componen la aplicación multimedia para evitar que el usuario pierda el interés.

Los cinco elementos de Bauza hay que entenderlos como

Figura 2: Modelo de aplicación multimedia bajo la óptica del modelo clásico de comunicación entre emisor y receptor.



una revisión incremental que va desde el mínimo exigido de múltiple salida al máximo de cobertura de los cinco principios. Se trata de principios que nos van a servir para analizar diversas aplicaciones y sistemas para entender hasta qué punto pueden ser consideradas aplicaciones multimedia.

Por ejemplo nos serviría para analizar si diversos tipos de sistemas son o no aplicaciones multimedia de acuerdo a las definiciones y principios destacados anteriormente. Una página web puede ser un buen ejemplo de aplicación multimedia porque suele incluir varios medios (al menos texto e imágenes); es interactiva cuando incluye enlaces a los que se puede acceder con libertad; si está bien diseñada, los elementos que incluyen deben estar por algo (principio de necesidad) y si está bien diseñada, puede también responder al principio de vitalidad y atención para dirigir al usuario en su uso. Una presentación PowerPoint, podría ser otro ejemplo a discutir: satisface todos los principios excepto el de libertad, ya que no suele haber distintas opciones que permitan navegar por la misma. En el caso de un anuncio publicitario o de una película de Hollywood, los principios que fallan son los de interactividad y de libertad.

Cuestión: ¿Una actuación de teatro de calle podría ser considerada un producto multimedia? Una actuación de teatro podría satisfacer todos los principios exigidos a una aplicación multimedia, sin embargo, aquí estamos hablando de aplicaciones que se ejecutan en sistemas electrónicos.

Aunque estrictamente hablando, una aplicación multimedia, para ser considerada como tal, debe cumplir los cinco principios, muchos de los aspectos que veremos en este curso son aplicables también a otros productos multimedia aunque no satisfagan todos los principios. Este es el caso del ciclo de las tres Ps que tratamos en la siguiente sección que de hecho es heredado del ámbito del cine y la publicidad.

II. PRODUCCIÓN DE PRODUCTOS MULTIMEDIA

El **ciclo de las tres Ps** se aplica de forma masiva en el desarrollo de contenido audiovisual, que incluye tanto aplicaciones multimedia como otro tipo de desarrollos cercanos al cine y la televisión. Detrás de las tres Ps se incluye la preproducción, la producción y la postproducción. Aquí empleamos la descripción de los procesos descrita en [2].

La **preproducción** es la fase de planificación del proceso de diseño. Las actividades están encaminadas a establecer el alcance, la dirección, la planificación y el coste del proyecto antes de comenzar con la producción real. Si se hace bien, se ahorra mucho tiempo y dinero en el resto de fases. El primer paso consiste en la **identificación del problema** y las necesidades del cliente. Es importante realizar un trabajo de investigación y validación de las mismas para estar seguros de que lo que se va a hacer responde a la necesidad del cliente. Se pasa después a la generación de ideas y el **desarrollo de borradores del concepto**. A continuación se procede al desarrollo de **presupuestos** y a la **planificación del proyecto** que conduzca a la creación del producto final.

La **producción** es la fase de diseño del proyecto donde se adquieren o producen las distintas partes del proyecto multimedia. El proceso de adquisición o producción de las partes puede consistir en la grabación de los cortes de vídeo, en la escritura de los textos, en la grabación de audios, en la edición de rótulos, la preparación de los botones de interacción y un largo etcétera.

En la **posproducción** se juntan los componentes individuales del proyecto para crear un proyecto unificado. El material generado en la etapa de producción se llama material en bruto (*raw material*). Este material se revisa en la etapa de posproducción para crear el producto final. El conjunto final de actividades depende del tipo de proyecto. En cine, la posproducción consiste en la preparación de los cortes de vídeo y la posterior creación de la secuencia final.

III. EL CICLO DE VIDA DEL SOFTWARE

Como estamos hablando de aplicaciones multimedia, nos toca referirnos ahora al término “aplicaciones” dejando por un momento el término “multimedia”. Cuando hablamos de aplicaciones no referimos a programas informáticos y cuando nos referimos al ciclo de vida del software nos referimos al desarrollo de dichas aplicaciones.

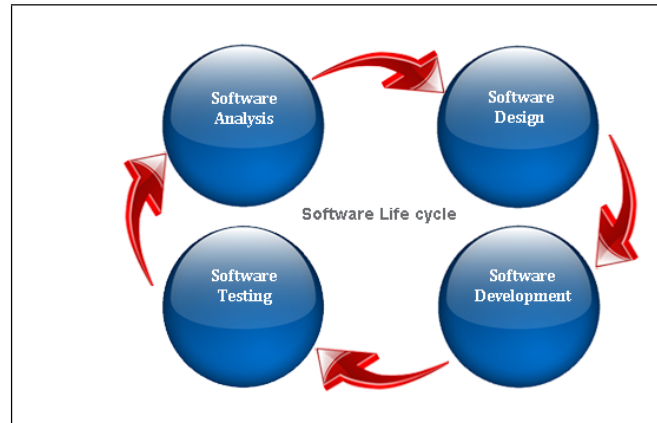
La figura 3 muestra las etapas necesarias en el desarrollo de las aplicaciones informáticas: planificación, análisis, diseño, implementación y mantenimiento.

En la **fase de planificación** se establecen los requisitos del software y se establecen las actividades y tareas necesarias para desarrollar una solución teniendo en cuenta los recursos de los que se dispone.

En la **fase de análisis** se establecen los requisitos del sistema que da solución al producto demandado. Es el momento de determinar qué debe hacer la aplicación. El resultado de etapa es una lista completa y detallada de los requisitos que recogen la funcionalidad del programa final.

En la **fase de diseño** se resuelve cómo el programa satisface los requisitos. En la etapa de diseño se establecen los componentes del programa, como su interfaz o el acceso a las bases

Figura 3: Esquema del ciclo básico del software (autor Dzonatas CC BY-SA).



de datos. El resultado de esta etapa es documentación técnica consistente en diagramas que describen los componentes de la aplicación y el intercambio de información entre ellos.

En la **fase de implementación** se hace el código. Teniendo en cuenta los requisitos del programa y cómo se acomete la solución, los programadores estarán en condiciones de escribir el programa que resuelva el problema formulado. El resultado de esta fase es un programa ejecutable.

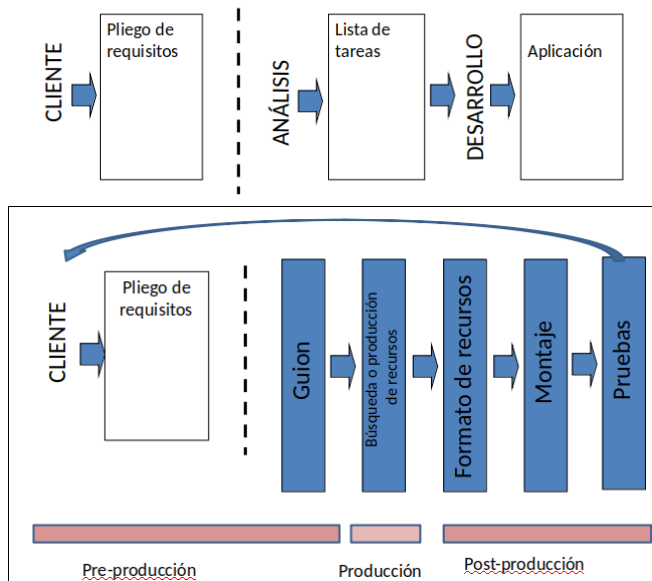
En la **fase de mantenimiento** se prueba el código con el objetivo de encontrar fallos en el programa. Se establecen una serie de tests que el programa debería pasar para considerar que funciona correctamente. El resultado de esta etapa son los resultados de dichos tests.

Se considera que la fase de pruebas tiene éxito cuando se encuentran fallos. La razón por la que el diagrama de la figura 3 presenta un **ciclo** es porque, los fallos encontrados deben ser analizados motivando posteriores cambios que afectarán tanto al diseño como a la programación de la aplicación informática. Este ciclo, tal y como está expuesto en la figura es un ideal que rara vez se cumple. Lo habitual es que, como resultado del proceso de pruebas realizadas en diversas etapas del desarrollo, sea necesario volver atrás a distintas etapas del proceso. Veremos en las siguientes secciones la importancia de la realización de prototipos en este proceso.

IV. EL CICLO EN EL DESARROLLO DE APLICACIONES MULTIMEDIA

La figura 4 presenta el esquema básico seguido en el desarrollo de aplicaciones multimedia. Vale la pena comparar este esquema con el relativo al desarrollo de cualquier aplicación. Se parte de un pliego de condiciones que generalmente suele ser vago e incompleto. Se realiza un proceso de análisis como el descrito en la sección anterior después del cual comienza el desarrollo. En el caso de las aplicaciones multimedia, hemos desglosado las etapas de análisis y desarrollo en las fases de guionado, búsqueda de producción y recursos, formato de los recursos, montaje y pruebas. Además se asume que el desarrollo es cíclico de manera que el producto resultante de las pruebas vuelve a pasar por el cliente hasta que éste de el visto final. En la parte inferior de la figura se alinean

Figura 4: Esquema del ciclo básico de desarrollo de aplicaciones (arriba) comparado con el ciclo de desarrollo de aplicaciones multimedia.



las fases propuestas con las de preproducción, producción y posproducción que se presentan a continuación.

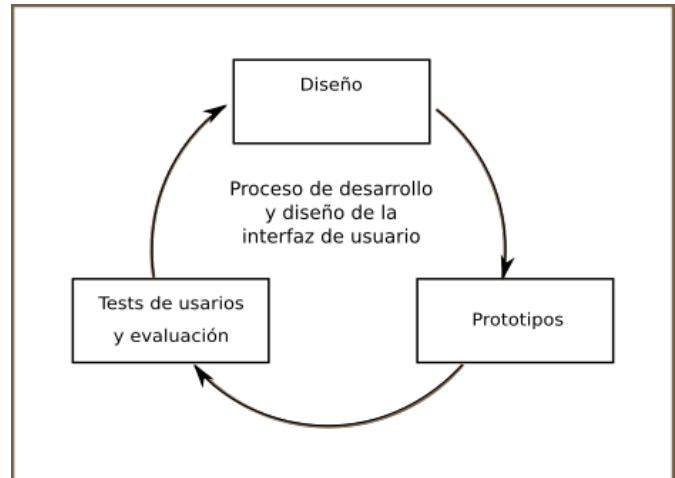
IV-A. Preproducción y requisitos

Se corresponden con las etapas de preproducción la especificación del pliego de condiciones y el guionado. Los desarrolladores sueñan con **pliegos de condiciones** completos y sin ambigüedades en los que se refleje cada detalle del producto final. Sin embargo la realidad es que el cliente suele tener una idea más bien vaga de cómo quiere que sea su producto final. El cliente sabe que tiene una necesidad, pero la mayor parte de detalles que respondan a la necesidad del cliente deben ser establecidos por el equipo de desarrollo. Obtener información del cliente que pueda servir para definir un producto final, es un proceso clave para el éxito del desarrollo. La herramienta principal para obtener información precisa por parte del cliente es la entrevista pero ya hemos visto en el apartado anterior que hay otras técnicas.

Las **entrevistas con clientes** son muy importantes para obtener información sobre las necesidades y opiniones de los clientes en relación a sus necesidades. Las entrevistas deben servir para entender lo que quiere el cliente. Las dos preguntas más importantes que deben quedar resueltas después de las entrevistas serían entender primero entender para qué sirve la aplicación multimedia y segundo determinar qué contenidos debe tener. Saber para qué sirve la aplicación multimedia tiene que ver con conocer el objetivo, el mensaje clave y la audiencia a la que va dirigida; determinar los contenidos está relacionado con los argumentos que se van a desplegar para dar cobertura al objetivo propuesto.

En este punto no debemos esperar que el cliente tenga claros todos y cada uno de los detalles de la aplicación final. No podemos esperar eso de un cliente. De hecho, gran parte de

Figura 5: Proceso de desarrollo de interfaz de usuario (inspirado en [3])

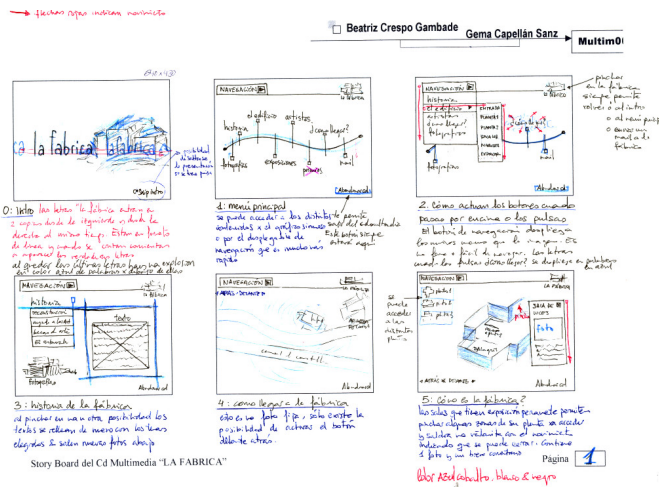


los detalles del producto final serán responsabilidad del equipo de diseño y desarrollo. Para evitar discordancias entre las propuestas del equipo de diseño y las expectativas de los clientes, es importante plantear un esquema cíclico desde el principio (ver figura 5). Se trata de desarrollar **prototipos** incompletos que se presentan regularmente a los clientes para volver a sondear sus opiniones. Los prototipos incompletos pueden ser tan sencillos como pequeños borradores realizados en lápiz y papel o tan completos como aplicaciones demostradoras en las que falte por definir alguna opción de los menús. Fijar fechas para las siguientes reuniones donde nosotros explicaremos al cliente lo que vamos a hacer es muy importante para evitar errores de concepto que después ocasionen retardos importantes en el proyecto. Estas reuniones cíclicas son necesarias porque muchas veces los clientes no saben con precisión el producto final que quieren o porque la información que aportan puede ser excesiva y poco útil para el objetivo final.

El **guion multimedia** es una herramienta de ayuda fundamental en el análisis de las necesidades del cliente y en la definición inicial de la solución. Un guion multimedia es un conjunto de bocetos anotados en los que se describe la funcionalidad del producto y la interacción entre el programa y el usuario (ver ejemplo en figura 6). Los guiones multimedia, cuando son completos, deben servir para que el cliente tenga una primera visión del cómo va a ser el producto final. También son un buen punto de partida para identificar las necesidades de producción y posterior posproducción del producto final. Los guiones son empleados en las entrevistas con los clientes para sondear opiniones.

Debemos prestar especial atención a las etapas de preproducción porque un mal análisis o un mal guion se considera una de las principales causas del fracaso de los proyectos debido al alto coste que tiene modificar un producto cuando los prototipos tienen un nivel de acabado considerable. Cambiar el borrador de un esquema de interfaz realizado en papel puede resolverse con unas tachaduras, cambiar un módulo software que realice una tarea concreta pueden suponer replantear el desarrollo completo por el hecho de que los componentes

Figura 6: Ejemplo de guion multimedia. Las viñetas describen visualmente los distintos estados por los que pasa la aplicación. Las anotaciones explican los detalles de interacción y las transiciones.



software pueden tener dependencias entre sí.

Son actividades que se pueden resolver en preproducción no sólo la especificación del objetivo y los requisitos de la aplicación multimedia (determinar qué quiero transmitir y cómo) sino también la **identificación de recursos** que serán necesarios: Texto, Grafismos, secuencias, Fotografía, Audio, Vídeo; y la determinación de **requisitos de usabilidad** con la realización de scripting de escenas en los guiones: determinar qué va a pasar y cuándo; determinar cómo va a ser la interacción entre la aplicación y el usuario; y también con la imposición de las restricciones del diseño gráfico para que sea coherente y homogéneo. Todo ello puede ser resuelto con un buen guion multimedia.

IV-B. Producción y diseño

Las actividades de producción son por un lado la búsqueda y generación de contenidos multimedia y por el otro el formateo de los mismos. Con respecto a la **generación de contenidos multimedia**, se trata de compilar los textos, grafismo, fotografías, audios, vídeos y animaciones que formarán parte de la aplicación multimedia. Algunos de ellos ya estarán disponibles, por ejemplo las fotografías de los catálogos de los clientes, pero otros hay que generarlas desde cero. El tiempo de generación de estos recursos puede ser nada despreciable. Tanto si los recursos ya estaban disponibles como si hay que generarlos, suele ser necesario el **formateo de material multimedia** para adaptarlo para que sea válido en la aplicación final. Por ejemplo, se puede disponer de un vídeo grabado previamente por el cliente, pero para nuestras aplicaciones solo se va a utilizar un corte y además el formato de vídeo que necesitamos será otro diferente al disponible inicialmente. Será necesario un trabajo de formateo y adaptación que no se puede despreciar. Otro ejemplo típico son las fotografías de catálogo. Será necesario escanearlas y darles una resolución adecuada al destino final dentro de la aplicación multimedia para optimizar el rendimiento.

IV-C. Posproducción, validación

Con respecto a la posproducción, las actividades de esta etapa son la integración y las pruebas. La **integración** consiste en la compilación del material multimedia dentro de una aplicación final de manera que todo el material responda de forma consistente a los objetivos marcados. Con el objeto de reducir el tiempo de integración al mínimo, se emplean herramientas de autor. Una **herramienta de autor** es un programa informático que sirve para generar aplicaciones multimedia facilitando al máximo la integración de material multimedia. Las principales compañías de software como Adobe, Microsoft, Apple y Android ofrecen herramientas de autor para facilitar la creación de aplicaciones multimedia en sus sistemas operativos. También existen sistemas para facilitar el desarrollo de aplicaciones multimedia que se puedan ver empleando navegadores en la web. A pesar de que estas herramientas facilitan mucho el desarrollo de aplicaciones web, al final suele ser necesario saber programar para realizar la integración por lo que disponer de profesionales de la informática en esta etapa suele ser necesario.

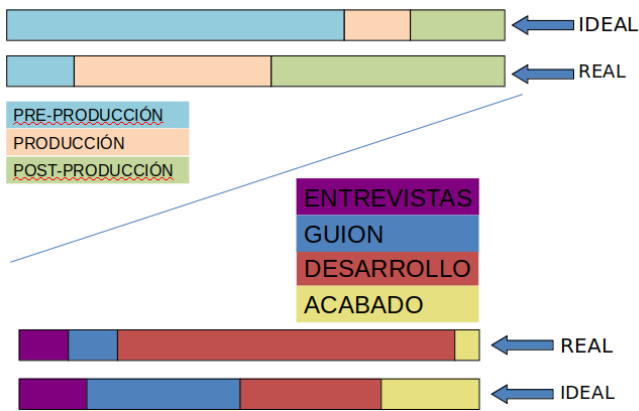
La etapa de **pruebas** es importante para validar si el producto cumple con las especificadas en la etapa de análisis. Se distingue entre test del programa y test de usabilidad. Los **test de programa** tiene como finalidad comprobar que la aplicación ejecuta correctamente una serie de casos predefinidos. Los **test de usabilidad** se realizan con usuarios reales a los que se les propone realizar una actividad con el programa. Como resultado de las pruebas es posible que haya que modificar el programa retomando el ciclo.

IV-D. Distribución

Una etapa importante en el ciclo de vida de un producto multimedia es la **distribución del producto**. Hace unos años, la distribución de aplicaciones multimedia se hacía mediante CD-ROMs o DVDs. Se empleaba estos soportes porque la multimedia exige grandes tamaños de almacenamiento. Hoy el formato de distribución más extendido es la red. Las aplicaciones multimedia, o bien son accesibles mediante un navegador web, o bien pueden descargarse de un repositorio de software como el Apple Store o Google Play. En cualquier caso, la preparación de la aplicación para poder ser distribuida es una actividad costosa en tiempo y recursos y no debe despreciarse.

Volvemos a insistir en este punto sobre la importancia de dedicar el tiempo suficiente a las etapas de análisis y preproducción. En la figura 7 compara los tiempos que suelen llevar cada una de las etapas en una situación ideal frente a lo que desafortunadamente suele ser lo habitual. Para reducir el tiempo de desarrollo, es importante dedicar tiempo a las etapas de preproducción. Reducir tiempo de desarrollo es muy importante porque es el más costoso. Una buena planificación y un buen análisis de requisitos posibilita reducir el tiempo de desarrollo porque evita dedicar tiempo a imprevistos. Reducir el tiempo de desarrollo posibilita dedicar tiempo a las etapas de acabado, incluyendo pruebas y distribución donde realmente el producto se juega buena parte de su éxito.

Figura 7: Línea de tiempos del desarrollo de aplicaciones software en general frente a aplicaciones multimedia en particular.



V. PRESENTACIONES MULTIMEDIA

Una presentación en general y una presentación multimedia en particular, es una herramienta eficiente para comunicar un mensaje. Entre los usos de una presentación están: persuadir a los oyentes para que hagan determinada acción, dar información, proporcionar un foro de discusión, averiguar cómo va a reaccionar la gente ante una determinada situación o idea, lograr compromisos, involucrar a las personas en la resolución de un problema etc. . . .

Existen diferentes tipos de escenarios para las presentaciones: ventas, demostraciones, persuasión, informes, plan de negocio o estrategia. Independientemente del tipo de escenario en el que se desarrolle la presentación y del uso de la misma hay una serie de consideraciones que son genéricas para todas ellas y que siempre se deben tener en cuenta. Este apartado es una síntesis de los consejos aportados en [4], que es a su vez un resumen de las ideas desarrolladas en [5]. Aunque muchos de estos consejos están pensados para presentaciones presenciales orales, las actividades preliminares y las de organización de contenidos son genéricas y sirven también para aplicaciones multimedia e incluso para anuncios publicitarios.

V-A. El objetivo, la audiencia y el contexto

Antes de ponerse a plantear detalles de la presentación, es muy importante **definir el objetivo** de la misma: Vender, recibir ideas, convencer. . . El objetivo condiciona el desarrollo de la presentación. Los elementos que van a componer después de la presentación deben estar justificados por el cumplimiento del objetivo que se establece a priori. Si algún elemento de la presentación no responde al objetivo, tal vez se pueda quitar o aportar como documentación extra. Algunas preguntas que podemos hacernos para establecer el objetivo son: ¿Para qué hago esta presentación? ¿Qué quiero que la audiencia haga como resultado?

Establecer el objetivo nos ayudará a determinar el sentido de hacer la presentación. También a hacernos preguntas como si realmente la presentación es el modo más efectivo de conseguir el objetivo buscado. ¿Debe hacerse esta presentación

a esta audiencia bajo estas circunstancias? ¿Qué debo decir? ¿Qué objeciones puede haber?

Una presentación es un diálogo interactivo entre el ponente y la audiencia. **Entender a la audiencia** ayuda a definir la presentación. Es importante poder responder a estas preguntas antes de desarrollar la presentación: ¿Quiénes son los miembros de la audiencia y cuál es su relación con el tema? ¿Cuánta información tienen sobre el tema? ¿Cuáles son sus expectativas? ¿Están acostumbrados a recibir algún tipo de información? ¿Qué saben de ti? ¿Están receptivos? ¿Es un tema controvertido? La presentación puede ser muy diferente en función de la respuesta a estas preguntas.

Cuestión: Imagina que has estado trabajando en el desarrollo de un diseño de un producto de consumo, por ejemplo, de una barbacoa. Si tienes que presentar el diseño a un fabricante, ¿harías la misma presentación que la que harías a un posible usuario? seguramente no.

Cuestión: Imagina un científico que lleva meses trabajando en una vacuna contra el COVID. ¿Crees que debería hacer la misma presentación de resultados para una revista científica que para una web divulgativa? seguramente no.

Volviendo atrás, la audiencia condiciona el objetivo, y el objetivo determina la presentación a realizar. Pero no solo hay una cuestión de objetivo. Hay cuestiones que determinada audiencia puede captar y otra audiencia diferente no entender en absoluto. Volviendo al caso de la vacuna, seguramente no sea buena idea emplear tecnicismos para presentar el trabajo en un artículo de divulgación porque la mayoría de lectores no lo entenderán.

Es muy importante **entender el contexto** en el que se realiza la presentación. Las presentaciones no se dan en el vacío, el contexto juega un papel importante en cómo va a ser recibida. Para valorar el contexto, puede ayudar contestar a estas preguntas: ¿Es una situación formal o informal? Cuando la presentación termine, ¿qué hará la audiencia? ¿Quién va a presentar antes o después? ¿Harán preguntas? ¿El escenario físico condicionará?

Hay presentaciones que se hacen en escenarios con un protocolo totalmente cerrado como puede ser la presentación de un caso frente a un tribunal. En estos casos hay poco margen para la originalidad, sin embargo, entender el contexto es muy importante. Una presentación que se realice en un juicio y que no entienda el contexto es una presentación llamada a fracasar porque el juez le llamará al orden pudiendo incluso retirarle la palabra.

En otros casos, la situación puede ser más abierta, sin embargo conviene siempre valorar el contexto. Si vas a presentar un nuevo amigo en tu pandilla más te vale entender que el contexto es importante para no meter la pata. No es lo mismo presentarlo en la cafetería de un bar que en el medio de un espectáculo.

V-B. Los contenidos

Una vez decidido el objetivo y conocida la audiencia, el siguiente paso es establecer qué debemos presentar. Es

importante entender que primero es el objetivo y la audiencia y después van los contenidos. Los contenidos deben estar determinados por el objetivo de la presentación, deben tener en cuenta a la audiencia y entender los posibles del contexto. Sólo entonces hay que plantear los contenidos.

A la hora de plantear los contenidos es recomendable hacerlo en tres etapas: primero definir un mensaje clave; segundo identificar los argumentos en los que se apoya el mensaje clave; y tercero identificar dónde es importante captar la participación de la audiencia.

Con respecto a la **definición del mensaje clave**, no debemos confundirlo primero con el objetivo de la presentación. El objetivo es aquello que queremos conseguir con la presentación, el mensaje clave tiene que ver con aquello que vamos a presentar para conseguir el objetivo.

El mensaje clave tiene que ver con qué quieres que la audiencia recuerde de tu presentación o con qué acción quieres que tomen después de verla. Se deriva del objetivo, pero no tiene por qué coincidir con el objetivo. Por ejemplo, en una presentación de TFG el objetivo es conseguir una buena calificación, el mensaje clave de la presentación debería ser: mi trabajo demuestra la suficiente madurez y solvencia técnica como para merecer una muy buena nota.

El mensaje clave es uno solo. Puede haber muchos argumentos, ideas, hechos, pero mensaje clave sólo hay uno.

Con respecto a la **identificación de los argumentos en los que se apoya el mensaje clave**, se trata de dar razones que refuercen el mensaje. En el caso de la defensa del TFG se trata de aportar evidencias que indiquen que el trabajo presentado es de una calidad elevada.

A la hora de aportar argumentos, conviene tener en cuenta que no es necesario aportar detalles excesivos. No es conveniente ser exhaustivo; suele ser más eficaz elegir los dos o tres argumentos más potentes y centrarse en ellos. Si la audiencia es crítica, se fijará en los aspectos de la defensa que queden más débiles restando puntos al conjunto. Si algún argumento puede sugerir una debilidad del trabajo, mejor no presentarlo.

Por último, con respecto a la **identificación de dónde es importante captar la participación de la audiencia**, hay que asegurarse de hablar sobre algo más que sólo los hechos. Es importante identificar y abordar las claves emocionales de tu mensaje. En este punto es importante pensar en la audiencia para identificar aquellos argumentos que pueden ser más efectivos.

Hay que tener en cuenta que lo que se aporte y que no contribuya a apoyar argumentos o evidencias son opiniones y que las opiniones tienen poco impacto en la audiencia. Se trata aquí de aportar razones para compartir tu punto de vista con la audiencia: ¿Por qué es importante mi presentación para la audiencia?

El proceso de definición de mensaje clave, identificación de argumentos y captación de audiencia se debe revisar y refinar hasta llegar a un grado lo suficientemente sólido. Hay que detenerse en este proceso para llegar a realizar una presentación potente. Sobre un conjunto inicial de ideas debes preguntarte: ¿El mensaje clave que hemos elegido es el más crítico? ¿Apoya los objetivos de la presentación? ¿Los argumentos aportados serán entendibles por todos los miembros

de la audiencia? ¿Serán convincentes? ¿Qué argumentos debo desarrollar? ¿Qué argumentos contrarios hay que neutralizar?

Si tu objetivo es convencer, hay que preguntarse qué más se puede hacer. Para convencer, hay que combinar lógica y emoción. No hay que olvidar que sólo se deben incluir los detalles que sirvan para convencer.

V-C. La organización

Sin perder de vista el objetivo y los argumentos en los que nos vamos a apoyar hay que ir dando forma a la presentación. La mayoría de presentaciones se organizan con una apertura, el descubrimiento de una necesidad o el establecimiento de un problema y una solución.

En la **apertura** podemos utilizar un gancho: un comentario, una pregunta, una historia relevante, una sentencia, un ejemplo. Buscamos con ello captar la atención de la audiencia. Una buena apertura se caracteriza porque: define el propósito de la presentación; deja clara tu credibilidad; describe la importancia del tema para la audiencia; y apunta los puntos a tratar.

Con respecto a la **necesidad o el establecimiento del problema**, debemos dejar claro a la audiencia que les interesa el tema. Se pueden incorporar ejemplos o hacer preguntas.

En la tercera parte se aporta la **solución**. Se debe dejar claro aquí a la audiencia cómo solucionar el problema. La presentación debe estar encaminada a ayudar a la audiencia a visualizar la solución. La solución aportada debe estar al alcance de la audiencia. Asegura que la solución aportada satisfice la necesidad.

Para terminar puedes concluir **llamando a la acción**. Para ello, puedes reiterar el mensaje clave o integrar los puntos de la apertura en los comentarios finales. Recomienda acciones, sugiere acuerdos para que la audiencia se sienta partícipe del mensaje clave de la presentación.

Una pregunta que suele surgir es cómo de larga debe ser una presentación. A veces la duración de la presentación está impuesta, pero otras veces se puede decidir.

Siempre hay que tener en cuenta que la presentación debe ser tan larga como sea necesario para transmitir el mensaje clave de forma clara y completa. A la vez debe ser sólo lo suficientemente larga como para ser clara y completa dentro del tiempo disponible. Quiere decir que una vez transmitido el mensaje, todo lo que sea alargar la presentación puede ser contraproducente.

Para evitar alargar innecesariamente la presentación es mejor desarrollar menos argumentos que hacerlos todos. Si no tienes tiempo para plantear puntos de forma clara y aceptable, pueden guardarse para otra presentación.

Acabar pronto es mejor que acabar de forma acelerada. Planifica qué no decir si el tiempo se alarga. Para ello pueden priorizarse los argumentos y elegir los más sólidos para ser desarrollados.

Las personas responden mejor a las imágenes (y a otros recursos multimedia) que a las palabras. La gente adquiere visualmente el 75 % de lo que sabe, el 13 % de lo que oye y el 12 % de lo que huele. Una imagen es 3 veces más efectiva en transmitir información que las palabras sólo. Las palabras y

las imágenes juntas son 6 veces más efectivas que las palabras solas. Utiliza las imágenes para apoyar ideas y argumentos es una postura inteligente de cara a potenciar las presentaciones.

Sin embargo el abuso de grafismos no siempre es efectivo. Se ha acuñado la expresión “death by PowerPoint” para hablar de presentaciones en las que el uso excesivo de grafismos despista y no aporta nada a la hora de transmitir el mensaje clave. El abuso de grafismos combinado con animaciones puede llegar a ser perjudicial.

Cuestión: Se considera que para ser efectivo en las presentaciones PowerPoint aportamos esta serie de consejos:

- Ser simple.
- Los grafismos sirven para reforzar o comunicar un concepto.
- Mejor palabras clave que frases completas.
- Utilizar un solo concepto por transparencia y no más de seis líneas por página.
- Usa color pero no excesivo. Usa imágenes.
- Como máximo dos tipos de letra y tamaños por página.

¿Con cuales de los aspectos mencionados en esta sección relacionaría cada uno de estos puntos?

VI. CONCLUSIONES

Una presentación es un proceso, y como tal se puede sistematizar y optimizar. Seguir las claves presentadas en este tema puede ayudar a organizar futuras presentación de manera que el trabajo pueda ser empleado de forma eficiente.

Lo más importante es centrar el objetivo y tener en cuenta que las acciones que tomemos en la presentación deben estar justificadas por dicho objetivo. Los elementos que formen parte de nuestra presentación deben estar justificados para satisfacer el objetivo puesto inicialmente.

El objetivo de la presentación se debe tomar teniendo en cuenta a la audiencia, que condicionará los contenidos de la presentación.

Aunque la última parte del tema puede parecer centrada en presentaciones orales, los puntos relativos a la organización de contenidos y los preliminares relativos al establecimiento del tema y los argumentos sirven para cualquier tipo de presentación.

MÁS INFORMACIÓN

La estructura y contenidos de este tema, en lo referente al análisis de requisitos, puede ampliarse información con manuales sobre ingeniería del software como [6]. Manuales específicos de desarrollo de aplicaciones multimedia pueden ser [7] (centrado en los guiones) y, sobre todo, [8].

En relación a cómo hacer presentaciones multimedia eficientes, se recomienda ver TED Talks como ejemplo de buenas presentaciones. TED es el acrónimo de (Tecnología, Entretenimiento, Diseño) y es una organización sin ánimo de lucro dedicada a difundir ideas que vale la pena que sean difundidas. Youtube está llena de videos de estas presentaciones de los que se puede aprender mucho. Es fácil encontrar vídeos TED de personajes influyentes en tecnología y diseño. En particular, se

recomiendo ver la presentación TED titulada “How to avoid death by PowerPoint” impartida por David JP Phillips en Estocolmo 2014. La presentación está disponible en Youtube.

CUESTIONES

1. Enumere las cinco características que debe cumplir una aplicación para se considerada una aplicación multimedia.
2. A qué se refiere el principio de salida múltiple en aplicaciones multimedia.
3. A qué se refiere el principio de interactividad en aplicaciones multimedia.
4. A qué se refiere el principio de libertad en aplicaciones multimedia.
5. A qué se refiere el principio de necesidad en aplicaciones multimedia.
6. A qué se refiere el principio de atención en aplicaciones multimedia.
7. Qué es el ciclo de las tres Ps.
8. Enumere actividades de preproducción, posproducción y producción en la industria del cine.
9. Explique el ciclo de desarrollo del software teniendo en cuenta las cuatro etapas fundamentales.
10. Qué es el análisis de requisitos software.
11. Qué es elicitación de requisitos.
12. Diferencia entre necesidades observadas, expresadas y normativas.
13. Indique tres técnicas de elicitación de requisitos.
14. Características de una lista de requisitos software.
15. Diferencia entre requisitos funcionales y requisitos de usabilidad.
16. Diferencia entre requisitos funcionales y requisitos no funcionales.
17. Qué actividades podemos considerar de pre-producción en desarrollo de aplicaciones multimedia.
18. Lo mismo sobre producción.
19. Lo mismo sobre posproducción.

REFERENCIAS

- [1] G. B. Bauzá, *El guión multimedia*. Univ. Autònoma de Barcelona, 1997.
- [2] V. Costello, *Multimedia foundations: Core concepts for digital design*. CRC Press, 2016.
- [3] S. Greenberg, “Teaching human computer interaction to programmers,” *interactions*, vol. 3, no. 4, pp. 62–76, 1996.
- [4] P. Mentor, *Giving Presentations: Expert Solutions to Everyday Challenges*. Harvard Business School Press, 2007.
- [5] G. L. Morrissey, T. L. Sechrest, and W. B. Warman, *Loud and clear: How to prepare and deliver effective business and technical presentations*. Basic Books, 1997.
- [6] R. S. Pressman, *Software engineering: a practitioner’s approach*. Palgrave macmillan, 2005.
- [7] G. B. Bauza, *El guión multimedia*. Madrid. Edit. Anaya Multimedia, 2003.
- [8] M. Elsom-Cook, *Principles of interactive multimedia*. McGraw-Hill, 2001.

Requisitos y preproducción de proyectos multimedia

OBJETIVOS

El objetivo de este tema es el de introducir al lector en la problemática de la especificación de requisitos, de su validación y la planificación del proyecto. Como resultado, después del tema el estudiante será capaz de:

- Crear un documento de requisitos de un proyecto multimedia empleando técnicas de elicitación de requisitos.
- Comprender la importancia del uso mockups y storyboards para la validación de los requisitos.
- Comprender terminología básica relacionada con la gestión de proyectos multimedia.

ÍNDICE

I.	Los requisitos	2
I-A.	Elicitación de requisitos	2
I-B.	La lista de requisitos	2
II.	Validación de requisitos	3
II-A.	Storyboard secuencial	4
II-B.	Storyboard narrativo	5
III.	Planificación del proyecto	6
III-A.	Descripción de tareas	6
III-B.	Planificación de recursos	7
III-C.	Actividades típicas	9
IV.	Conclusiones	10
	Referencias	11

En todo proyecto multimedia, la etapa de preproducción es una fase creativa en la que se identifica el problema que el proyecto debe resolver, respondiendo a la pregunta de “qué” debe hacer la aplicación que se va a desarrollar. Para ello, es importante establecer la funcionalidad del producto, validar dicha idea con el cliente y planificar las etapas de producción que sirvan para concebir una solución.

En la etapa de definición de requisitos, el equipo de diseño debe contactar con los clientes para entender el dominio de aplicación, la audiencia, los servicios que debe ofrecer el sistema a desarrollar e intentar determinar todos los condicionantes que pueda tener el producto.

El resultado de la elicitación de requisitos es una lista de requisitos del producto multimedia que debe ser validado con el cliente. Para ello, existen técnicas entre las que destacan las de realización de prototipos de bajo coste con bocetos que presentan el producto antes de comenzar su desarrollo.

Una vez que queda clara la idea de qué producto va a ser desarrollado, la fase de preproducción incluye una etapa de planificación, en la que se asignan tiempos y recursos a las diferentes tareas que debe llevar a la producción de la solución final.

En este tema, primero se detallan una serie de conceptos clave relacionados con la elicitación de requisitos software. Después se presenta el desarrollo de mockups y storyboards para la validación de los requisitos. El tema finaliza con la presentación de la etapa de planificación del proyecto multimedia.

I. LOS REQUISITOS

La etapa de análisis es crítica porque debe suponer el punto de partida del trabajo de un equipo de desarrolladores. Solucionar los errores cometidos en la etapa de análisis tiene un alto coste: un error de programación se resuelve cambiando unas líneas de código; un error de diseño, cambiando o reprogramando algún módulo; pero un error en análisis supone replantear partes enteras de la aplicación afectando a la fase de diseño.

I-A. Elicitación de requisitos

Se conoce como elicitación de requisitos el proceso mediante el cual se obtiene una lista de requisitos del sistema.

El primer paso es el de **análisis de las características de usuario** que conduce a extraer requisitos específicos que respondan a tales características. La tabla I muestra algunos de los requisitos de la aplicación que pueden surgir del análisis de los usuarios presentado en las tablas II y III.

Una vez identificados y caracterizados a los usuarios llega el momento de analizar lo que quieren o lo que esperan los usuarios del sistema. Hay que tener en cuenta aquí que es fácil encontrar situaciones en las que los usuarios no sepan expresar lo que quieren, o incluso que no sepan lo que quieren. Distinguimos entre las necesidades observadas y las necesidades expresadas. Las **necesidades observadas** están ocultas en las primeras etapas y deben surgir del análisis del equipo de diseño y del conocimiento del dominio. Las **necesidades expresadas**, como su nombre indica, son propuestas por lo usuarios. A pesar de ser más evidentes, necesitan también ser analizadas porque los usuarios no tienen por qué saber hasta dónde puede llegar a ofrecer soluciones la tecnología en su estado actual. Por otro lado, están las **necesidades normativas**, que serán aquellas propuestas a los clientes o usuarios por el equipo de desarrollo. Surgen del conocimiento de los diseñadores del dominio. Aunque habrá que implementarlas de todos modos, conviene que el cliente no las vea como una imposición. Un ejemplo de este tipo de necesidades es el hecho de tener que avisar de que una aplicación va a acceder a la cámara del móvil o a la carpeta de imágenes cuando se instala.

La **observación de los usuarios** en su entorno, realizando las tareas en las que se espera que ayude el sistema informático es una parte esencial del diseño basado en usuario. Si hacemos una buena observación podemos anticiparnos y entender qué aspectos le van a gustar al usuario y qué aspectos no; qué aspectos le van a resultar útiles y cuáles no.

Distinguimos entre observación directa y la observación indirecta. En la **observación directa** distinguimos entre la que es realizada en estudios de campo y la que es realizada en estudios controlados. En los **estudios de campo** se observa directamente a los usuarios en su entorno natural, realizando sus tareas habituales a la vez que se toma nota sobre aspectos interesantes de su comportamiento. En los **estudios controlados** la observación no se hace en el entorno natural sino en espacios preparados que intentan representar los aspectos que son relevantes para el producto que se está desarrollando.

La **observación indirecta** se realiza grabando en vídeo a los usuarios en su entorno natural o en su entorno controlado.

Las **entrevistas** son otro recurso imprescindible. En una entrevista se establece un diálogo con los usuarios para obtener información. Es una forma de obtener información más rápida que mediante la observación, pero hay que tener en cuenta que las entrevistas deben ser preparadas cuidadosamente para que la información obtenida sea realmente útil. Para ello se deben establecer objetivos y un plan para ser lo más eficiente posible.

Se distingue entre entrevistas estructuradas y entrevistas flexibles. En las **entrevistas estructuradas**, se dispone de una serie de preguntas predeterminadas que se realizan en orden, con poco espacio para la improvisación. En la **entrevistas flexibles** no hay lista de preguntas prefijadas pero sí que hay una serie de temas de discusión y exploración sobre los que se debe hablar. Es un formato más informal, pero si no se realiza de acuerdo a un plan se corre el riesgo de que sea muy poco productiva.

Crear un clima de confianza en la entrevista es importante para que el interlocutor no se sienta intimidado o incómodo. En el caso de la entrevista estructurada, esto es menos importante, llegándose a hacer entrevistas enviando previamente las preguntas al interesado.

El uso de **cuestionarios y encuestas** es una aproximación alternativa a las entrevistas en la que se preparan formularios rígidos que los interlocutores deben rellenar.

I-B. La lista de requisitos

El objetivo de la etapa de análisis es obtener una lista de requisitos de la aplicación que debería tener las siguientes características:

Completa : todo lo que debe hacer la aplicación debe estar reflejado en la especificación de requisitos.

Consistente : debe ser coherente sin incluir contradicciones, no puede haber un requisito que contradiga lo que dice algún otro.

Inequívoca : con una redacción clara de manera que no haya pie a equivocaciones por ambigüedades o dobles interpretaciones.

Correcta : Todos los requisitos se deben poder cumplir.

Trazable : Debemos poder comprobar si el requisito se ha cumplido realmente en la etapa de pruebas.

Esta lista puede ser empleada para controlar la calidad de las especificaciones de requisitos que se definan para los productos software, de manera que hasta no cumplir con todos los criterios, es mejor no avanzar en las etapas de diseño e implementación del producto final.

Se distingue entre los **requisitos funcionales** que describen las tareas que debe realizar el sistema informático; los **requisitos de usabilidad** que tienen que ver con el punto de vista del usuario teniendo en cuenta que el sistema debe ofrecer niveles altos para las siguientes características: [2]:

Facilidad de aprendizaje (*learnability*) que es el tiempo y esfuerzo requerido para aprender a manejar el sistema.

Rendimiento (*throughput*) la velocidad de la ejecución de tareas por usuarios experimentados.

Característica	Requisito
Edad entre 12 y 80+	Las pantallas de ATM deben estar a una altura que se acomode a la altura de diferentes usuarios
Acceso de personas con limitaciones físicas	La altura de las pantallas se debe acomodar a los usuarios sin discapacidad, a los que caminen con bastones y a los que usen silla de ruedas. La artritis en las manos puede ser problemática por lo que se debe tener en cuenta a la hora de diseñar los controles
Puede haber discapacidad auditiva	Debe haber feedback visual y sonoro
Puede haber discapacidad visual	El texto en los displays debe ser grande
Poca experiencia TICs	La aplicación debe ser fácil de usar. Debe ser fácil de aprender.

Cuadro I: Requisitos extraídos del análisis de las características de usuarios (de [1]).

Característica	Descripción
Edad	Entre 12 y 80+
Género	Ambos
Limitaciones físicas	La mayoría serán personas con completas capacidades, pero puede haber alguno con limitaciones físicas relacionadas con la audición, vista, movilidad, uso de las manos o uso de sillas de ruedas.
Nivel educativo	Pueden tener un nivel educativo mínimo. Pueden tener mínimas capacidades para leer y entender números.
Uso TICs	Puede haber usuarios con ninguna experiencia en TICs.
Motivación	Puede haber usuarios muy motivados, especialmente porque podrán resolver tareas más fácilmente sin hacer colas
Actitud	Muy diversas dependiendo del servicio y de la fiabilidad de la tecnología

Cuadro II: Características de usuarios clientes de ATM (de [1]).

Característica	Jóvenes	Adultos	Mayores
Edad	Entre 12 y 25	Entre 25 y 50 y	Entre 50 y 80+
Limitaciones físicas	La mayoría serán personas con completas capacidades, pero puede haber alguno con limitaciones físicas relacionadas con la audición, vista, movilidad, uso de las manos o uso de sillas de ruedas.	La mayoría serán personas con completas capacidades, pero puede haber alguno con limitaciones físicas relacionadas con la audición, vista, movilidad, uso de las manos o uso de sillas de ruedas.	Limitaciones en movilidad, uso de las manos o uso de sillas de ruedas.
Uso TICs	Nativos digitales	Todos tienen experiencia en TICs	Puede haber usuarios con ninguna experiencia en TICs.
Motivación	Probablemente muy motivados	Puede haber usuarios muy motivados	La mayoría prefiere hacerlo como hasta la fecha

Cuadro III: Características de usuarios clientes de ATM dividiendo por tramos de edad (de [1]).

Flexibilidad la capacidad del sistema para cambiar con respecto al entorno para el que fueron definidos.

Actitud positiva que se genera en los usuarios después de utilizar el sistema.

Un último tipo de requisitos son los **requisitos no funcionales**, que tienen en cuenta los condicionantes de tipo técnico y económico que deben tenerse en cuenta. Por ejemplo son requisitos de este estilo el hecho de que la aplicación deba ejecutarse en entorno Windows o Mac, o que deba ejecutarse con un mínimo de memoria de sistema. También puede ser un requisito no funcional que una página web pueda verse en diferentes tamaños de pantalla.

II. VALIDACIÓN DE REQUISITOS

Una lista de requisitos no debe darse por cerrada hasta no recibir el OK del cliente. En ocasiones la comunicación con el cliente no es fácil porque éste puede no entender el léxico, no siempre fácil, del diseñador de aplicaciones multimedia. Para facilitar la comunicación entre desarrolladores y clientes, el uso de **prototipos de bajo coste** suele ser de gran utilidad. El tipo de prototipo de bajo coste más utilizado es el de los bocetos o *sketches* que no son otra cosa que dibujos que presentan la interfaz del sistema a desarrollar.

El boceto sirve para presentar un diseño en base a dibujos que muestran la estructura y aspecto final de la aplicación.

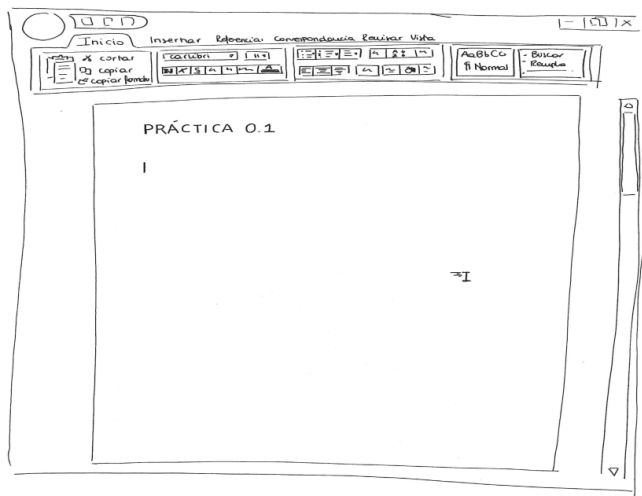
Realizar este tipo de dibujos suele requerir de cierta destreza por lo que se suele considerar una actividad muy especializada con la que, en ocasiones, se cuenta con dibujantes. Sin embargo, hay que tener en cuenta que lo importante en el bocetaje, no es hacer una obra de arte, sino crear un recurso funcional que permita comunicar las ideas de la solución al cliente. Es importante, además, que el prototipo sea de bajo coste, por lo que no se espera un boceto excesivamente detallado sino más bien, una solución rápida. Aunque ser buen dibujante ayuda, no ser especialmente hábil con el dibujo no debería ser impedimento para la realización de bocetos recomendándose practicar dibujando interfaces de aplicaciones conocidas.

Ejercicio: Realice un boceto de una aplicación bien conocida como puede ser PowerPoint o Word.

La figura 1 muestra un ejemplo de solución. Se recomienda hacer este ejercicio primero visualizando la pantalla del ordenador con el programa activo y después sin ver el programa. Hasta que no adquiera soltura dibujando es mejor realizar los bocetos con aplicaciones a la vista.

Un boceto de interfaz para la comunicación efectiva de un producto debe tener una serie de **elementos básicos** cuales son el dibujo, las anotaciones, las flechas y las notas. El **dibujo** es lo que entendemos por boceto pero por sí solo no suele

Figura 1: Ejemplo de boceto de interfaz de programa (cortesía de Inés del Hoyo Alonso).



servir. Las **anotaciones** son etiquetas cuya posición espacial identifica la parte del sketch al que se refieren. Son marcas gráficas que se incorporan en el dibujo. Las **flechas** son marcas gráficas que pueden posicionar anotaciones. Cuidado con las flechas, porque en ocasiones pueden ser ambiguas; éstas no sólo posicionan anotaciones sino que también pueden realizar otras funciones como indicar dirección, mostrar movimiento, indicar una secuencia de eventos o indicar el flujo de la interacción. Se recomienda que utilizar flechas de diferente formato para indicar diferentes funciones (por ejemplo cambiando el color, el grosor o la forma). Por último, **las notas** son anotaciones cuya posición espacial no es importante y pueden ser ideas sobre el diseño, opciones alternativas, explicaciones, detalles no incorporados como efectos sonoros. Se emplea lenguaje natural y pueden ser tan extensas como se considere necesario para explicar la interfaz de la aplicación.

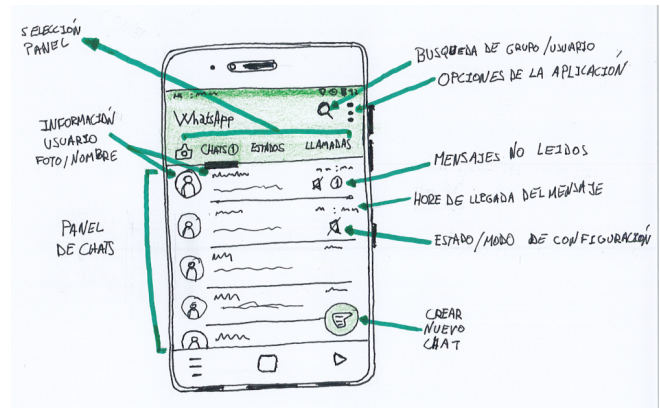
Ejercicio: Realice un boceto que presente la interfaz principal de WhatsApp

La figura 2 muestra un ejemplo de solución. Puede mejorarse con flechas que indiquen desplazamiento y notas al pie que expliquen aspectos básicos como qué es un mensaje o qué es un contacto.

Utilizar dibujos simples empleando lápiz y papel es rápido y económico. Usar un interfaz como PowerPoint o un editor gráfico puede ser una alternativa, aunque dar un acabado razonable en estas herramientas no especializadas puede ser costoso en tiempo. Como alternativa, existen aplicaciones para hacer *mockups* (un *mockup* se considera más completo que el boceto) que utilizan modelos de alambres o *wireframe* y que, en ocasiones pueden incluir funcionalidades básicas para probar la herramienta final (un ejemplo de este tipo de herramientas es Balsamiq, pero hay decenas).

Algunos recursos para agilizar el proceso de bocetaje de prototipos de bajo coste es el usar pequeños **post-it** que puedan

Figura 2: Ejemplo de boceto de interfaz de app (cortesía de Álvaro Juarros Izquierdo).



ser reubicados sin tener que redibujar. Otro el de utilizar **plantillas** que tengan predibujados algunos elementos como la ventana del navegador o la forma del teléfono móvil. El uso de **rotoscopia**, con el uso de fotografías sobre las que se copian dibujos puede acelerar extremadamente el proceso. Dibujar elementos animados supone un desafío que puede resolverse empleando curvas P (ver el tema de animación) o pequeños iconos alusivos al cambio.

II-A. Storyboard secuencial

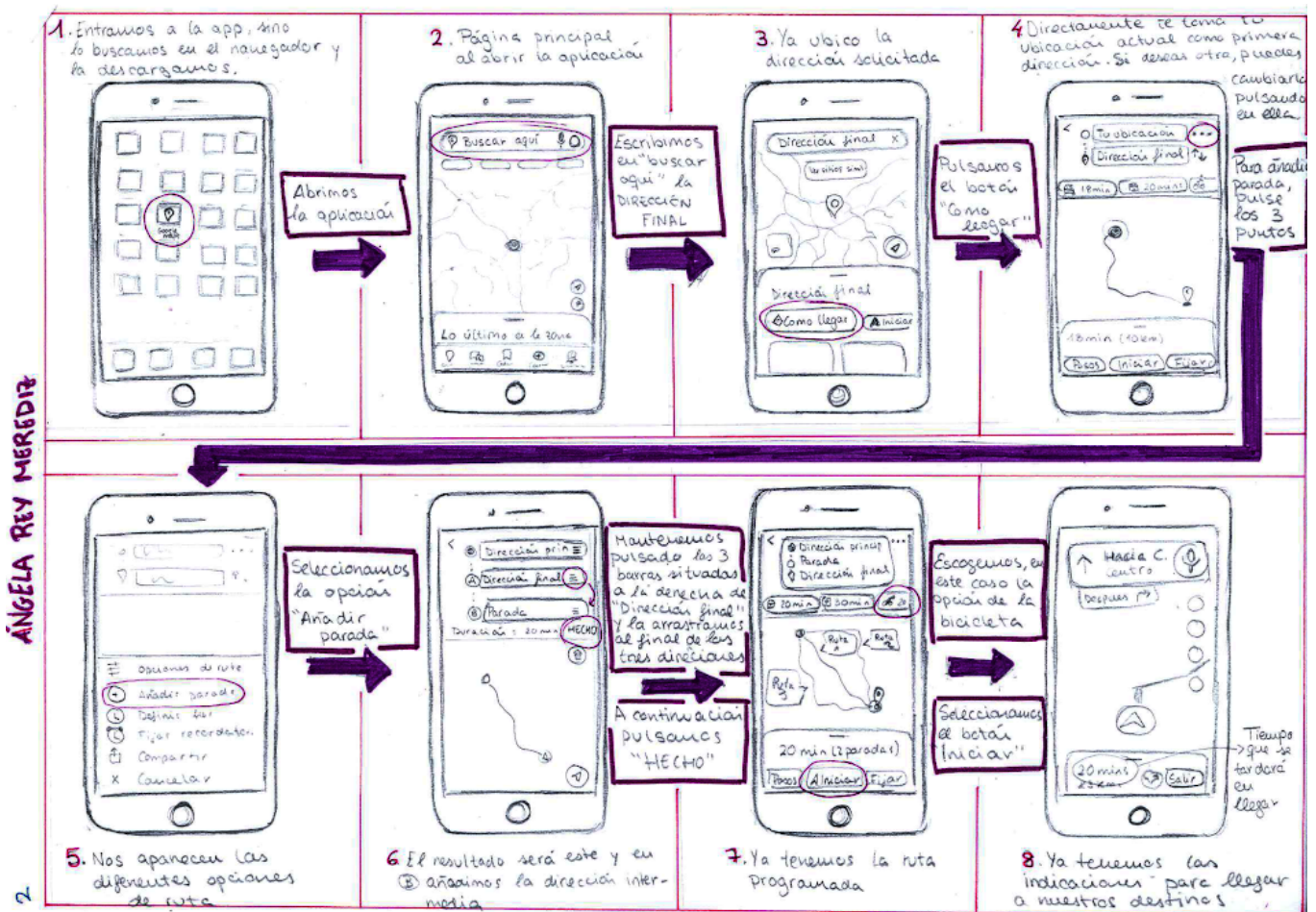
Un sketch captura la idea de cómo va a ser la interfaz de una aplicación en un instante de tiempo determinado. Para la mayoría de sistemas, esto es insuficiente para expresar cómo va a ser la experiencia de usuario porque el interfaz evoluciona a lo largo del tiempo. Un boceto efectivo necesita incorporar las acciones, la interacción y los cambios en la experiencia a lo largo del tiempo. El storyboard se utiliza para expresar esta evolución del sistema a lo largo del tiempo.

Un storyboard básico es sólo una secuencia de imágenes donde cada imagen muestra un instante de tiempo en una historia visual. Los comics hacen esto mismo, son una secuencia de dibujos donde cada dibujo es un *key frame* o fotograma clave o un **estado**, que ilustra un momento importante en la secuencia. El espacio entre claves son las **transiciones** entre estados, que suceden como resultado de una actividad. Este espacio en blanco entre claves lo completa la imaginación del lector. De igual manera, un storyboard secuencial aplicado a la descripción de la interfaz de las aplicaciones multimedia, traza una secuencia de estados y las transiciones entre estados.

La primera decisión a la hora de hacer un storyboard es decidir cuáles son los cuadros clave a dibujar. Para evitar hacer storyboards demasiado largos, hay que elegir lo esencial. Sólo deben dibujarse los estados relevantes de manera que cada transición pueda incluir varias subacciones que pueden omitirse.

La segunda decisión tiene que ver con las **anotaciones**. Si sólo hay imágenes dejamos a la imaginación del lector la reconstrucción de las acciones. Incluyendo anotaciones sobre las transiciones y sobre los estados podemos aclarar cómo se producen los cambios.

Figura 3: Ejemplo de storyboard secuencial que explica la búsqueda de una ruta en Google Maps (cortesía de Ángela Rey Merediz).



Ejercicio: Realice un storyboard que describa cómo se busca una ruta en Google Maps.

La figura 3 muestra un ejemplo de solución. Es importante marcar en los bocetos cómo se desencadena un evento e incluir anotaciones sobre las flechas de las transiciones cuando sea necesario.

II-B. Storyboard narrativo

Un storyboard narrativo es una sofisticación del storyboard convencional que proporciona el entorno narrativo en el que se desarrolla la interacción. Informa de cómo el usuario utiliza el sistema en el contexto. Por el hecho de incluir personajes interactuando con el entorno, son una variación del storyboard cinematográfico.

En cine, las escenas se suelen bocetar para facilitar las decisiones de dirección y de fotografía. Esto es así porque realizar los distintos cortes que componen la película es un proceso extremadamente costoso que requiere alquilar equipos, espacios y pagar a actores. Para evitar alargar tiempos, todo debe planificarse escrupulosamente.

En un storyboard narrativo, al igual que en el cine se emplean seis tipos de vistas principales:

1. El plano extremadamente largo (*wide shot*), que es una vista que muestra detalles del entorno, localizaciones, etc....
2. El plano largo (*long shot*), que muestra a los personajes en toda su altura.
3. El plano medio (*medium shot*), que muestra la cabeza de un personaje y sus hombros.
4. El plano sobre el hombro (*over-the-shoulder shot*), que mira por encima del hombro del personaje.
5. El plano de punto de vista (*point of view shot*), que muestra lo que ve el personaje.
6. El plano de detalle (*close-up*), que muestra los detalles como podría ser un punto concreto de la interfaz del sistema que manipula el personaje.

Por supuesto que hay otras muchas opciones de planos, sin embargo, en la mayoría de los casos, sólo se emplean los citados anteriormente.

A la hora de planificar un storyboard narrativo debemos tener en cuenta que debe servir para responder, al menos, a las siguientes preguntas:

1. ¿Dónde?
2. ¿Cuál es el problema?
3. ¿Cuál es la tarea que se intenta hacer?
4. ¿Quién participa y qué hace?
5. ¿Qué dispositivos utiliza?
6. ¿Cuál es el input y output de los dispositivos?
7. ¿Cómo hacen las acciones los usuarios y las personas?

Es importante utilizar el menor número de fotogramas posibles e intentar mantener la atención del usuario, de manera que el storyboard debe contar una historia que, desde un principio vaya dando detalles de una historia hasta llegar a un climax, y concluir después rápidamente en el fin de la historia.

En un storyboard narrativo se dibuja primero cada escena con un plano de cámara adecuado. A continuación se destacan detalles que muestren las acciones y los movimientos del personaje. Al igual que ocurría en el caso de los bocetos aislados, puede emplearse rotoscopía para acelerar el proceso empleando fotografías.

Ejercicio: Realice un storyboard narrativo que describa una app para traducir mensajes en chino apuntando con la cámara del móvil.

La figura 4 muestra un ejemplo de solución. Podría mejorarse acentuando con un color diferente los gestos del usuario del móvil.

III. PLANIFICACIÓN DEL PROYECTO

Una vez realizada la lista de requisitos del proyecto se concretan los objetivos del proyecto que vamos a desarrollar, y una vez definido un diseño de una posible solución se pasa a la etapa de desarrollo. El desarrollo del proyecto es un proceso cuyo éxito va a depender de la realización de una buena planificación y gestión (*management*).

La principal razón para realizar la planificación y gestión del proyecto es garantizar que el proyecto sea factible, sea realizable (*feasible*). Para ello hay dos factores clave: el coste y el tiempo. La planificación y gestión del proyecto pueden entenderse entonces como aquellas actividades cuyo fin es la realización del proyecto optimizando el coste de desarrollo y el tiempo de ejecución.

La planificación plantea un ideal que rara vez se cumple. Los planes ideales rara vez se cumplen porque se cometen errores y porque ocurren contingencias. Eso no significa que la planificación fracase porque la gestión de contingencias forma parte de la gestión del proyecto. La planificación y gestión se realiza siguiendo los siguientes pasos.

1. Monitorizar el progreso, evaluando a lo largo del proyecto el estado de ejecución del mismo.
2. Identificar problemas y desviaciones con respecto al plan.
3. Corregir los problemas antes de que sus efectos afecten al conjunto del proyecto.

De esta manera, se espera que la gestión sirva para corregir los problemas que puedan surgir reorganizando la planificación.

Un proyecto multimedia no suele implicar recursos físicos de gran volumen como otros proyectos de ingeniería como

los desarrollados en ingeniería civil. El desarrollo software es, por lo general, un trabajo intelectual que implica a personas. Salvo excepciones como puede el desarrollo de sistemas como Google Maps que han necesitado una gran plataforma de vehículos, en la mayoría de proyectos multimedia el equipamiento físico no va más allá del equipamiento tecnológico con el que trabajan las personas del equipo. Este hecho simplifica la planificación al máximo: por un lado está la descripción de tareas y por el otro la planificación de recursos (principalmente humano).

En este tema presentamos primero la descripción de tareas como elemento básico en la gestión de proyectos. Después detallamos la descripción de recursos como segundo elemento clave. La tercera sección tiene que ver con la descripción de detalles relativos a actividades típicas de desarrollo de proyectos multimedia. Finaliza el tema con algunas consideraciones de derechos de autor relativas a los productos generados y a los assets empleados en los mismos.

III-A. Descripción de tareas

El trabajo global se divide en unidades llamadas tareas (*tasks*). Una tarea se puede subdividir creando sub tareas. Un número de tareas entre 6 y 12 es razonable, aunque depende del tipo de proyecto. En ocasiones las tareas se suelen agrupar en actividades.

El plan de proyecto tiene un punto de entrada (el comienzo del proyecto) y un punto de salida (indica cuando el proyecto se termina). Para llegar de un punto a otro, hay que realizar un número de tareas. Algunas tareas son independientes pero la mayoría son dependientes unas de otras. Por ejemplo, no puedes integrar los grafismos en la interfaz con los grafismos hasta que éstos no estén preparados.

En el caso de definición de una web de presentación de un producto de consumo, podríamos considerar las actividades de preproducción, producción y posproducción y las tareas podrían ser en un caso típico y a modo de ejemplo:

Actividad 1: Preproducción

Tarea 1.1: Investigación/objetivo

Tarea 1.2: Concepción

Tarea 1.3: Bocetos preliminares

Tarea 1.3: Storyboard definitivo

Actividad 2: Producción

Tarea 2.1 Grafismos

Tarea 2.2 Tipografía

Tarea 2.3 Animaciones

Tarea 2.4 Audio

Tarea 2.5 Vídeo

Tarea 2.6 Programación

Tarea 2.7 Inicio

Tarea 2.8 Menú

Tarea 2.9 Contenidos de sección 1

Tarea 2.10 Contenidos de sección 2

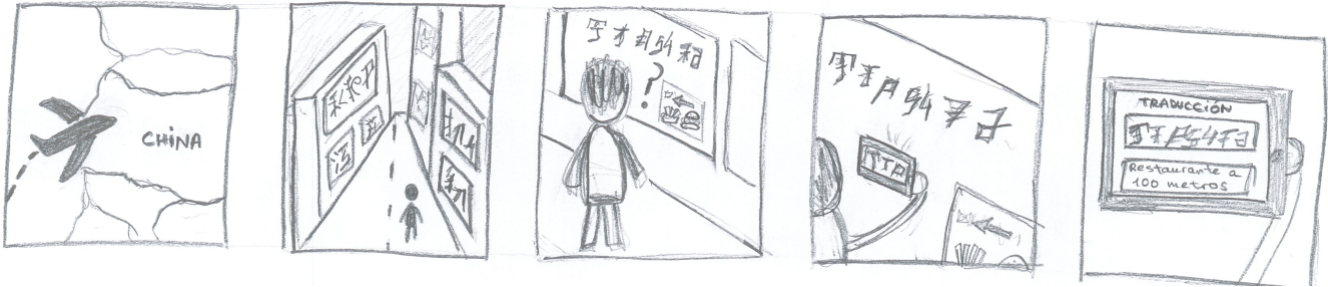
...

Tarea 2.N Contenidos de sección N

Actividad 3: Posproducción

Tarea 3.1 Montaje

Figura 4: Ejemplo de storyboard narrativo que explica una aplicación de traducción con la cámara (cortesía de Diana Torres).



Tarea 3.2 Instalación

Tarea 3.3 Despliegue

Actividad 4: Pruebas

Tarea 4.1 Test de concepto

Tarea 4.2 Test de software

Tarea 4.3 Test de usabilidad

Actividad 5: Documentación

Cada una de estas tareas debería venir acompañada de una breve descripción, de la identificación de los responsables de las mismas, de su duración y de los resultados esperados.

Las dependencias entre tareas determinan un grafo dirigido en el que se especifica el plan de proyecto. Se crea un diagrama llamado **diagrama de red PERT** (en inglés *PERT chart*, que son las siglas de *Project Evaluation and Review Technique*). Estos diagramas reflejan el hecho de que una tarea no puede empezar hasta que no han terminado las que están unidas a ellas mediante líneas de entrada. La figura 5 muestra la red PERT del proyecto descrito con la lista de actividades presentada en el párrafo anterior.

Es importante dejar claro qué se hace en cada tarea. Se debe establecer cuál es el punto de partida y el resultado esperado. Los **hitos** o *milestones* en inglés son puntos identificables que se alcanzan a lo largo del proyecto. Algunos ejemplos de hitos podrían ser terminar una parte de código, una reunión de revisión ... Determinarán el cierre (*closure*) de una actividad o tarea. Si el resultado es algo concreto (típicamente un documento o un programa o un storyboard) el milestone se llama **entregable** o *deliverable* en inglés.

Para nuestro proyecto de ejemplo, hemos identificado dos entregables y cuatro hitos:

Entregables :

- 1: Un storyboard para realizar pruebas de concepto.
- 2: El sitio web final.

Hitos :

- 1: La evaluación de resultados de la prueba de concepto.
- 2: La evaluación del prototipo software.
- 3: La evaluación son usuarios.
- 4: La primera versión del prototipo.
- 5: La versión del sitio web instalada en un servidor.

La figura 5 marca los hitos y los deliverables dentro del diagrama de PERT.

Una información muy importante que no se aporta en los diagramas de PERT es la que tiene que ver con la duración

de cada tarea. Los **diagramas de Gantt** vienen a cubrir esta carencia aportando información de cuando se realiza cada tarea y de cuál es su duración. Un diagrama de Gantt es una matriz cuadrículada en la que cada fila se corresponde con una tarea y donde las columnas indican duraciones y periodos de tiempo.

La figura 6 muestra un diagrama de Gantt básico para el proyecto que estamos describiendo como ejemplo. Los diagramas pueden completarse con información sobre las dependencias entre tareas y también se puede incluir incertidumbre en la duración de cada tarea señalando las celdas temporales con algún tono intermedio.

Un diagrama de Gantt permite ver la duración total del proyecto. En un diagrama PERT es algo más complejo. Si sabemos la duración de cada tarea, podemos calcular la duración total empleando el grafo. Cuando hay tareas en paralelo, el camino más largo es el que determina la duración total del proyecto. Al camino más largo se le llama **camino crítico**. Cualquier tarea dentro del camino crítico es fundamental para evitar retrasos. Las actividades que no están en el camino crítico pueden retrasarse hasta cierto grado sin afectar a la duración total del proyecto.

III-B. Planificación de recursos

La planificación de recursos es muy importante porque los recursos que se emplean en cada tarea van a condicionar su duración así como las actividades que se llevan a cabo dentro de cada una de ellas. Una buena gestión de tareas necesita tener en cuenta la disponibilidad de recursos que serán necesarios para cada una de ellas.

Distinguimos dos procesos importantes en la etapa de planificación de recursos, de un lado la identificación de los recursos necesarios para cada tarea, de otro el cálculo del coste.

III-B1. Identificación de recursos: Debería hacerse una lista exhaustiva de los recursos necesarios para cada tarea. Los recursos humanos en un proyecto multimedia serán diversos porque hacer un proyecto multimedia suele ser un trabajo interdisciplinar. Entre los integrantes de un equipo de desarrollo multimedia pueden intervenir: el project manager, diseñadores de software, programadores, especialistas en hardware, diseñadores de interfaz, editores de texto, diseñadores de gráficos, fotógrafos, guionistas, realizadores de vídeo, etc. . .

Figura 5: Ejemplo de diagrama de PERT. Los entregables *deliberables* aparecen identificados en con círculos rojos. Los hitos *milestones* aparecen marcados con rectángulos verdes.

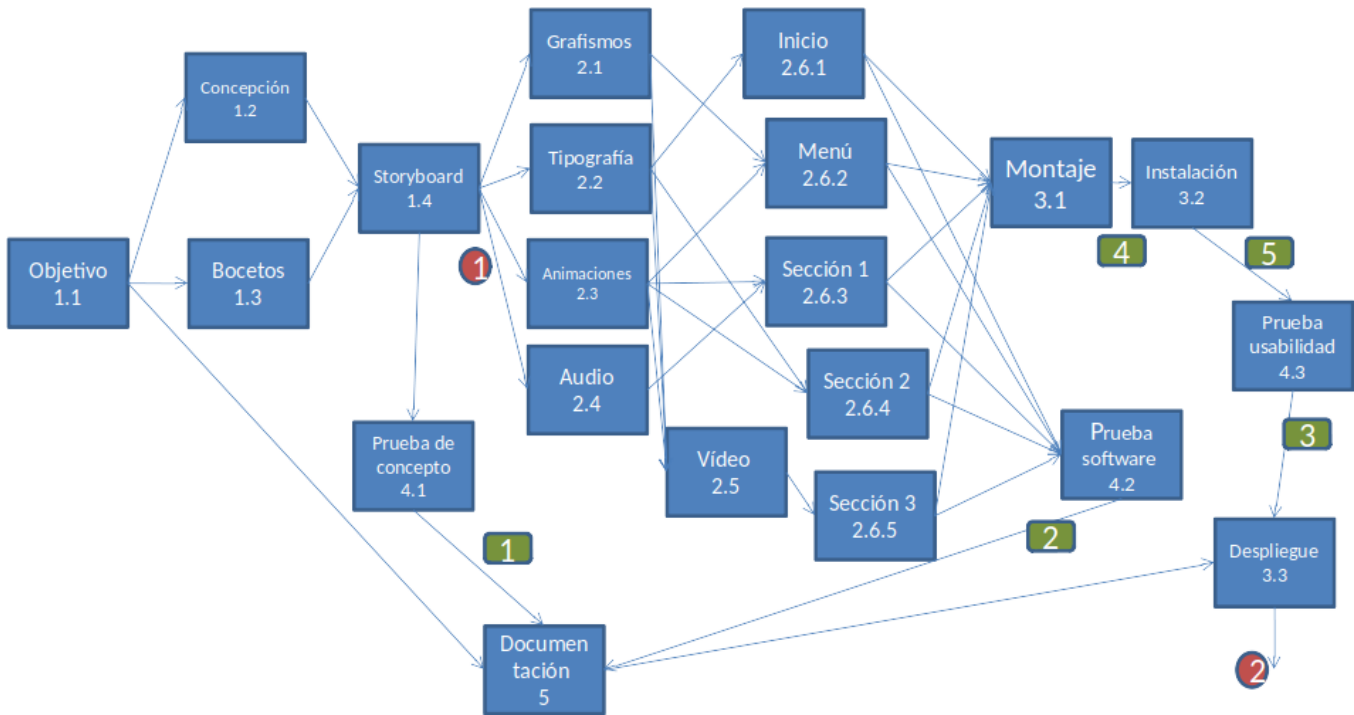


Figura 6: Diagrama de Gantt. Cada columna indica la semana del proyecto.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
T1.1 Investigación/objetivo	█	█	█	█											
T1.2 Concepción		█	█	█	█										
T1.3 Bocetos preliminares			█	█	█	█									
T1.4 Storyboard definitivo				█	█	█	█	█							
T2.1 Grafismos					█	█	█	█							
T2.2 Tipografía						█	█	█							
T2.3 Animaciones							█	█							
T2.4 Audio								█	█						
T2.5 Vídeo									█	█					
T2.6.1 Inicio										█	█	█	█		
T2.6.2 Menú											█	█	█		
T2.6.3 Sección 1												█	█	█	
T2.6.4 Sección 2													█	█	█
T2.6.5 Sección 3														█	█
T3.1 Montaje														█	█
T3.2 Instalación															█
T3.3 Despliegue															█
T4.1 Test de concepto							█								
T4.2 Test de software												█	█		
T4.3 Test de usabilidad														█	█
T5 Documentación															█

La tabla siguiente muestra una serie de tareas junto con el perfil de los recursos humanos que participarían en cada una de ellas.

Tarea	Recursos humanos
1	Project manager, guionista, editor de vídeo
2	Programadores de alto nivel, programadores de bajo nivel
3	Director de vídeo, fotógrafo (estudio)
...	...

Aunque no es el ejemplo que hemos desarrollado en el apar-

tado anterior, ilustra cómo diferentes tipos de tareas implican la participación de diferentes perfiles profesionales.

Con respecto al equipamiento multimedia, entre los equipos necesarios destacan los ordenadores y el software de desarrollo, las cámaras y equipo de edición fotográfica y de vídeo, el equipo para hacer la distribución (un servidor de internet o una cuenta en los repositorios de apps de Apple o de Google), y los assets (recursos multimedia) que puedan ser reutilizados: fotos, vídeos, módulos...

Al igual que en el caso de los recursos humanos, es importante que cada tarea venga descrita con el equipamiento necesario para llevarla a cabo.

III-B2. Costing and timing de los recursos: Los recursos por tanto deben ser asociados con las tareas. Para ello se hace una lista de tareas y se identifican los recursos asociados a cada tarea. Para los recursos físicos sólo es importante asegurarse de los recursos estarán disponibles antes de empezar cada tarea. Para la planificación de recursos humanos, debe asignarse cada tipo de recurso humano a cada tarea.

Como ya hemos dicho, la planificación de recursos en proyectos multimedia afecta principalmente a los recursos humanos. El resultado final de la planificación de recursos será conocer el número total de personas-día para cada tarea y el número total de días necesarios para cada tarea. **Persona-día** es una medida que refleja la cantidad de trabajo que una persona hace en un día laboral en condiciones ideales.

A cada tarea se le asigna una cantidad de recursos humanos que se mide en personas-día. Una persona-día es la cantidad de trabajo que una persona puede hacer en un día. A modo de ejemplo, la siguiente tabla establece una lista de tareas con los

correspondientes recursos humanos necesarios para resolverla:

Tarea	1	2	3	4	5	6
Project manager	10	0	2	3	2	4
Script writer	1	0	0	5	10	5
Director	4	4	2	12	3	7
Programadores	3	1	2	0	0	0

El objetivo de la planificación será calcular el esfuerzo total en personas día que cada persona implicada en cada tarea va a realizar. Con ese dato, de forma indirecta, podemos calcular la cantidad de tiempo total de la tarea. En la tabla superior vemos que la tarea 4 necesita 3 personas-día de trabajo del manager, 5 de guionista y 12 de director. En total necesitaría un esfuerzo en recursos humanos de 20 personas día, lo que equivale a 12 días de trabajo o 16 días de calendario.

A partir de la tabla de personas-día, se puede hacer una buena estimación del coste de los recursos humanos. En la siguiente tabla se computa el número de personas días asociados a cada tipo de recurso humano y de ahí se deriva el coste total del proyecto en recursos humanos teniendo en cuenta el coste por día de cada tipo de trabajador.

Recurso	Personas-día	Coste por día	Coste total
Project manager	21	500€	10500€
Script writer	21	300€	6300€
Director	32	700€	22400€
Programadores	6	500€	3000€
Total			42200€

Conviene hacer un diagrama de Gantt para cada persona. Cuidado porque el esfuerzo puede repartirse entre varios días: indicar 1 persona-día no quiere decir que todo el trabajo se haga el mismo día.

III-C. Actividades típicas

La figura 7 muestra un diagrama PERT genérico para el desarrollo de proyectos multimedia. En un proyecto multimedia hay diferentes actividades típicas, cada una de las cuales tiene una serie de peculiaridades a tener en cuenta:

III-C1. Programación: Cuando se emplea una herramienta de autor como puede ser WordPress el trabajo de programación puede ser despreciable. Sin embargo, cuando queremos un producto con alguna peculiaridad técnica, tener que realizar programación es prácticamente inevitable.

La **división en módulos** es una ayuda importante en los trabajos de programación. Los programas se dividen en módulos independientes que se comunican empleando una interfaz. Cada módulo puede ser realizado de forma independiente. Debe definirse una estructura clara del proyecto global dividido en módulos, clases o componentes. Diferentes programadores desarrollarán diferentes componentes de manera que el trabajo puede realizarse en paralelo.

Algunos módulos o componentes diferenciados son: las consideraciones de interfaz, las estructuras de datos y bases de datos, los algoritmos internos relativos a la multimedia, la programación de dispositivos...

El perfil de los **programadores** suele ser muy heterogéneos. El grado de experiencia en un sistema, plataforma o lenguaje de programación concreto determina el rendimiento. Es muy difícil estimar el tiempo, sobre todo si la tarea no se ha hecho

antes. Hay que dejar margen. Como posible regla práctica [3] sugiere preguntar y multiplicar por 2 y añadir un 25%. El mismo autor apunta que hay dos tipos de programadores: los organizados y los freakies. No desprecies a los freakies pero no lo dejes todo en sus manos.

III-C2. Los assets: Los assets son los recursos multimedia: gráficos, vídeos... Si no vamos a generar nosotros los assets, hay que tener en cuenta que las fuentes deben garantizar que estén a tiempo. Cuando se utilizan recursos preexistentes hay que tener cuidado con el copyright. Aunque los assets se proporcionen con un nivel de acabado profesional siempre habrá que hacer posproducción.

Si los generamos nosotros debemos tener en cuenta, para los distintos tipos de recursos que el texto, aunque es lo más fácil debemos prestarle mucha atención porque es muy importante. Con respecto al sonido, éste requiere una cuidada selección y generalmente una edición para adecuar formatos.

El trabajo de fotografía es sencillo de presupuestar porque existen bancos de imágenes donde se ofrecen precios dentro de un intervalo conocido a priori. Siempre es mejor contar con profesionales que hacer las imágenes por uno mismo.

Para los gráficos hay que tener cuidado con el grado de perfeccionismo de los diseñadores. Retocar los diseños puede ser un proceso que conviene recortar poniendo plazos estrictos.

El vídeo no es sólo que un amigo coja una cámara y grabe. Contar con profesionales es importante cuando se requiere un acabado profesional.

III-C3. Integración: La integración comienza cuando tenemos una estructura del sistema vacía de contenido, y pasamos a colocar los assets en su lugar correspondiente. Si pensamos en el desarrollo de un sitio web empleando WordPress, podemos haber elegido un tema y tenemos que integrar nuestras imágenes, vídeos, textos y menús propios.

Conviene tener una versión previa vacía de contenido porque en la integración surgen aspectos como la reprogramación de aspectos que no hemos tenido en cuenta, el reprocesado de assets porque su aspecto no es el esperado o la creación nueva de assets porque algunos se descartan.

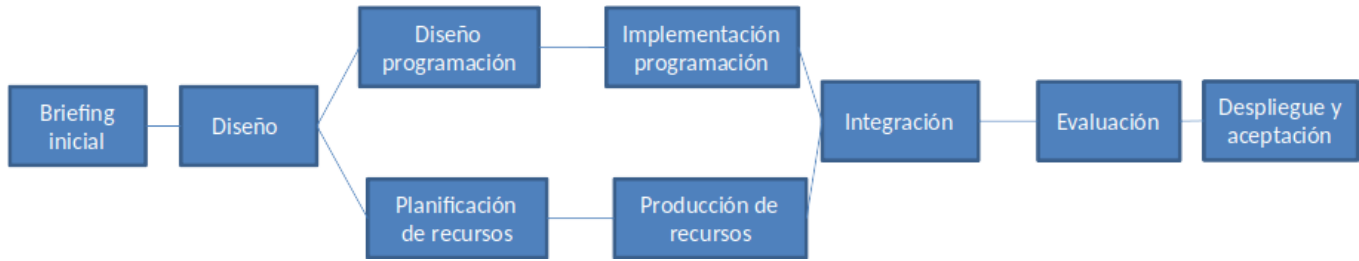
III-C4. Evaluación: La evaluación es un proceso de gran importancia porque sirve para valorar si hemos cumplido los objetivos (**evaluación sumativa**). También como fuente de información para futuras mejoras (**evaluación formativa**).

El objetivo de una prueba de evaluación es encontrar problemas en el programa. Hay que someter el sistema a una prueba que simule el uso del mismo y comprobar que funciona en cada caso.

La interacción se evalúa con un test de usabilidad. El test de usabilidad puede encargarse a un profesional externo, en cuyo caso sólo es importante tenerlo en cuenta a la hora de hacer el presupuesto. Cuando se realizan tests de usabilidad con usuarios reales, se requiere una planificación y la dedicación de recursos humanos para coordinar la ejecución de los tests.

III-C5. Despliegue y aceptación: El sistema se lanza para su uso por parte de clientes finales. En un sitio web, el despliegue se realiza cuando el servicio está alojado en un servidor web y se dispone de un dominio. Es importante que en la etapa de pruebas participen usuarios cuyo perfil sea no más parecido al de los usuarios finales.

Figura 7: Diagrama PERT genérico para un proyecto multimedia.



La aceptación es el proceso mediante el cual, tras la entrega del producto al cliente, éste considera que el trabajo ha finalizado. El objetivo es que el cliente pague por el trabajo, para lo cual hay que demostrar que se ha cumplido con el encargo.

La argumentación que debe seguirse para culminar el proceso de aceptación con éxito puede resumirse en la secuencia: 1.- este fue el problema que me ha planteado, 2.- esta es la solución, 3.- estos son los criterios por los que has de juzgar la solución, 4.- aquí está el sistema, 5.- cumple los criterios y a los usuarios les gusta, 6.- lo hemos entregado en tiempo 7.- Págame de la manera acordada.

III-C6. Documentación: Es importante porque se deja constancia del trabajo realizado. Distinguimos entre documentación para cliente, para usuarios, para el equipo de desarrollo y para el mantenimiento.

En la **documentación para el cliente** es importante dejar claro el establecimiento del problema y solución. Se puede hacer una revisión de la solución del problema y añadir un informe de evaluación final. Incluir planes futuros puede ayudar a mantener la fidelidad del cliente.

En la **documentación para los usuarios** hay que incluir los manuales, con información completa de cómo emplear el producto. También una introducción fácil mínima para que el usuario pueda empezar a manejar el producto. La documentación de usuario está denostada porque un principio básico de usabilidad requiere que no sea necesaria: el programa debe ser utilizable sin necesidad de leer ninguna documentación o acudir a ninguna ayuda.

Con respecto a la **documentación para el equipo de desarrollo**, se debe dar una visión general del producto que ayude a entender la arquitectura general del sistema y también una documentación para el equipo de mantenimiento, que incluya la documentación de código, y la documentación del estilo de programación a seguir en futuras revisiones.

III-C7. Investigación y diseño: Dejamos para lo último estos dos aspectos porque normalmente se considera que el proyecto empieza cuando hemos decidido ya el diseño. Son procesos delicados que implican el contacto con el cliente que, en ocasiones no tiene claro cómo va a ser el producto final que necesita. Conviene hacer una buena planificación de estas entrevistas y de la presentación de storyboards al cliente para que éste dé el visto bueno del trabajo a desarrollar.

III-C8. Management: Al igual que la documentación, el management queda fuera del esquema presentado en la figura

7 porque es una actividad transversal, que se realiza a lo largo de todo el ciclo de desarrollo del producto.

Existe una amplia bibliografía sobre management que se puede utilizar [4], [5], [6] porque los métodos suelen ser lo suficientemente generales como para poder ser utilizados en gestión de proyecto multimedia. Quizá la principal peculiaridad de los proyectos multimedia es la naturaleza interdisciplinar de los productos, para lo cual hay que tender puentes entre perfiles diferentes que hablan lenguajes diferentes. Pero esto es algo que ocurre también en otro tipo de proyectos por lo que un buen gestor debe saber lidiar con ello.

Algunas claves para realizar una gestión eficaz son:

Monitorización : estar pendiente de lo que van haciendo los miembros del equipo. Hay que preguntar mucho y evitar pedir informes escritos que quitan tiempo y luego no te les.

Información : Todos los miembros del equipo deben estar informados si hay cambios.

Reconocimiento de problemas y resolución : Hay que saber reconocer los problemas que surjan y también intentar anticiparse a las consecuencias.

Re-planificación : Es muy raro que todo salga como se esperaba. Hay que saber reaccionar y comunicar cuando haya cambios.

IV. CONCLUSIONES

El proceso de elicitación de requisitos es una actividad clave para entender las necesidades de los clientes. Se trata de un proceso creativo cuyo resultado es una lista escrita de requisitos que podrá ser empleada para validar la solución final.

Los prototipos de baja calidad como los storyboards son utilizados para facilitar la comunicación con el usuario y asegurarse de que la lista de requisitos es correcta.

La planificación de la producción es una etapa fundamental de la preproducción que consiste en asignar recursos y establecer los tiempos de realización de cada una de las etapas que conducen a la solución final.

MÁS INFORMACIÓN

Para aclarar conceptos relacionados con la elicitación de requisitos se recomienda acudir a [7]. Los contenidos de la sección son una adaptación de [8].

Con respecto a la realización de bocetos y storyboards, la principal referencia es [9], y como referencia complementaria [10], [11].

Los contenidos de la sección sobre planificación han sido extraídos en su mayoría de [3]. En otros textos de management puede obtenerse abundante información [4], [5], [6].

CUESTIONES Y PROBLEMAS

1. Qué es el análisis de requisitos software. Qué es elicitación de requisitos.
2. Diferencia entre necesidades observadas, expresadas y normativas
3. Indique tres técnicas de elicitación de requisitos.
4. Características de una lista de requisitos software.
5. Diferencia entre requisitos funcionales y requisitos de usabilidad
6. Haga un sketch para indicar cómo cambiar de vídeo en TikTok. Use dibujo, flechas y anotaciones.
7. Dibuje el interfaz del navegador Chrome.
8. En qué consiste la técnica de rotoscopía aplicada al sketching
9. Ventajas e inconvenientes de hacer sketch a mano o hacerlos con programa.
10. Qué se entiende por wireframing en el mundo del diseño de interfaces de usuario
11. Haga un sketch de la página web que está desarrollando en las prácticas
12. Cómo indicaría en un sketch que un eslogan cambia de aspecto, por ejemplo haciendo ciclos opaco transparente
13. Haga un storyboard que explique cómo publicar una fotografía en Instagram
14. Haga un storyboard que explique cómo instalar una aplicación en Android o en iOS
15. Realice un storyboard narrativo para presentar una aplicación para móvil de realidad aumentada que permite a los excursionistas conocer el nombre de los picos de una cordillera que se ve desde un mirador.
16. Haga un storyboard narrativo para presentar el nuevo invento del profesor Bacterio: una gafas que permiten ver desnuda a la gente.
17. Indique qué es un diagrama PERT y para qué se utiliza.
18. ¿Qué es un entregable o deliberable en gestión de proyecto?
19. ¿Qué es un hito o milestone en gestión de proyecto? Defina el término persona-día.
20. ¿Por qué decimos que en un proyecto multimedia el principal recurso a gestión es el recurso humano?
21. En qué consiste la integración en un proyecto multimedia.
22. Qué es un asset en un proyecto multimedia.
23. Enumere tres tipos diferentes de documentación a tener en cuenta en el desarrollo de proyectos.
24. Por qué se dice “la mejor documentación de usuario es la que no existe”.
25. Qué es la monitorización en las actividades de management.
26. Nombre cuatro actividades del management en gestión de proyecto.

- [7] I. Sommerville, “Software engineering 9th edition,” *ISBN-10*, vol. 137035152, p. 18, 2011.
- [8] D. Stone, C. Jarrett, M. Woodroffe, and S. Minocha, *User interface design and evaluation*. Elsevier, 2005.
- [9] S. Greenberg, S. Carpendale, N. Marquardt, and B. Buxton, *Sketching user experiences: The workbook*. Elsevier, 2011.
- [10] B. Buxton, *Sketching user experiences: getting the design right and the right design: getting the design right and the right design*. Morgan Kaufmann, 2010.
- [11] P. Laseau, *Graphic thinking for architects and designers*. John Wiley & Sons, 2001.

REFERENCIAS

- [1] A. Chapanis, “Human factors in,” *Systems Engineering*, 1996.
- [2] J. L. Bennett, “Managing to meet usability requirements: establishing and meeting software development goals,” *Visual Display Terminals*, Prentice-Hall, pp. 161–184, 1984.
- [3] M. Elsom-Cook, *Principles of interactive multimedia*. McGraw-Hill, 2001.
- [4] C. S. Dionisio, *A Project Manager's Book of Forms: A Companion to the PMBOK Guide*. John Wiley & Sons, 2017.
- [5] M. D. Lock, *Project management in construction*. Gower Publishing, Ltd., 2012.
- [6] T. L. Young, *Successful project management*. Kogan Page Publishers, 2016.

Producción y diseño de aplicaciones multimedia

OBJETIVOS

El objetivo de este tema es dar a conocer una serie de aspectos teórico prácticos relacionados con el diseño e implementación de aplicaciones multimedia, poniendo el acento en el diseño de la interfaz. Después de este tema, el alumno deberá:

- Entender el diseño e implementación de aplicaciones y productos multimedia como una búsqueda incremental de soluciones satisfactorias.
- Comprender la importancia del diseño sistemático de aplicaciones, respetando patrones, guías y principios de diseño.
- Entender cuáles son los principales controles que permiten definir la interfaz interactiva de las aplicaciones multimedia.

ÍNDICE

I.	El diseño como búsqueda de soluciones	2
II.	Diseño e implementación de aplicaciones multimedia	2
III.	Diseño de la interfaz centrado en el usuario	4
	III-A. Definición de usabilidad	4
	III-B. Claves sobre usabilidad	4
IV.	Principios y guías de diseño de la interfaz	5
	IV-A. Áreas de diseño	5
	IV-B. Cualidades del diseño	5
V.	Controles para la interacción	8
	V-A. Controles de introducción de datos . . .	8
	V-B. Controles de presentación de datos . . .	9
	V-C. Estructuración de contenido	10
	V-D. Controles para la navegación	11
VI.	Conclusiones	11
	Referencias	12

La fase de producción, dentro del desarrollo de aplicaciones y proyectos multimedia, permite la definición de una solución práctica a los requisitos planteados en la etapa de preproducción. En este tema, se abordan una serie de fundamentos relacionados con el diseño de aplicaciones y proyectos multimedia. Para ello, comenzamos estableciendo a qué nos referimos cuando empleamos el término “diseño” en este ámbito (sección I) para hacer referencia a la concepción del diseño como la búsqueda de soluciones satisfactorias de un espectro amplio de alternativas.

En lo que hace referencia al diseño de aplicaciones, centramos el término (sección II) para presentar los patrones de diseño y las herramientas de autor como recursos específicos dentro del ámbito del desarrollo multimedia que permiten facilitar el trabajo de los programadores.

Nos centramos después en el componente de la interfaz de usuario por ser éste un componente que está íntimamente relacionado con la información multimedia. Se introduce el término de usabilidad (sección III), como concepto clave en la definición de sistemas informáticos funcionales y prácticos, para después describir los controles que integran, típicamente, la interfaz de las aplicaciones. (sección V).

La apariencia de la interfaz es una de las claves más importantes entre los condicionantes que guían el diseño de soluciones satisfactorias. Presentamos en la sección IV los principios de diseño y las guías de estilo que son útiles en la definición de interfaces atractivos.

I. EL DISEÑO COMO BÚSQUEDA DE SOLUCIONES

Diseñar puede considerarse como el proceso de búsqueda de una solución satisfactoria que dé respuesta a unos requisitos preestablecidos. Podemos entender que partimos de una idea sencilla que evoluciona y mejora para acercarse al diseño óptimo. La evolución finaliza cuando llegamos a una solución satisfactoria. El objetivo es conseguir un camino hacia la solución satisfactoria que se basa en la modificación iterativa de diferentes propuestas.

Esta concepción del diseño fue propuesta por Herber Simon (premio Nobel de Economía y padre de la Inteligencia Artificial) en los años 50 del siglo XX en su obra “Las ciencias de lo artificial” [?] y ha influido en la concepción del diseño como disciplina y área de investigación en diferentes ámbitos de la ingeniería y de las ciencias sociales. El proceso de búsqueda de la solución satisfactoria supone realizar cambios sobre una propuesta inicial. Estos cambios pueden tener naturaleza técnica y también naturaleza social y deben tener en cuenta las restricciones de tiempo y de presupuesto. Los **condicionantes técnicos** implican que no toda idea puede desarrollarse porque la tecnología disponible en un momento determinado dependerá fuertemente del estado de la materia; por ejemplo podemos pensar en realizar un interfaz empleando pantallas curvas, pero sabemos que todavía no es una tecnología madura. Los **condicionantes sociales** tienen que ver con la audiencia para la que vaya dirigido el producto, las tendencias vigentes en un momento dado y las normativas aplicables; por ejemplo en un interfaz hay legislación específica sobre accesibilidad.

La solución a la que nos lleva el proceso de diseño depende mucho de la primera idea o del **punto de partida** desde el cual vayamos detallando el producto final. Si la idea inicial no era la mejor, el resultado final no puede ser el mejor. Por muy bien que se realice el proceso de diseño, si el punto de partida no permitía llegar a un producto óptimo el resultado al que llegues tendrá otros con los que deba competir que serán mejores. La historia del diseño está llena de productos que, a pesar de ser soluciones técnicas muy buenas, fracasaron porque surgió una idea inicial que llevó a productos mucho más competitivos. Un buen ejemplo es el de los teléfonos móviles, donde empresas como Nokia o Motorola se hundieron por no apostar por los terminales con pantalla táctil. Por esto, generar nuevas ideas sobre las que construir un producto final tiene un valor tan importante en el mundo del diseño.

Fijada una idea de producto, el proceso de diseño consiste en un proceso iterativo de elaboración y reducción de alternativas. La elaboración consiste en la generación de soluciones buscando oportunidades para llegar a soluciones óptimas. La **reducción** consiste en decidir cuáles son las alternativas que merecen la pena y trabajar sobre ellas. La creatividad permite generar múltiples opciones posibles y también la creatividad nos permite desechar las menos viables empleando criterios heurísticos.

Este proceso nos lleva a hablar del **embudo del diseño**, entendido como el proceso de generación de ideas y reducción que se alternan en un proceso que converge hacia el concepto final. En cada etapa se generan nuevas ideas y se descartan otras. La granularidad de las opciones se va reduciendo hasta

llegar a un diseño final. El número de opciones al principio del proceso es elevado y éstas se van reduciendo al ir tomando decisiones de diseño o de implementación.

Entre las actividades que sirven para ser creativos en la búsqueda de las primeras ideas en el diseño de soluciones están las siguientes [?]:

Actividades de observación: Etnografía, diarios de fotos y vídeos, shadowing, predicción de tendencias, autopsia de productos, sketching.

Actividades de aprendizaje: Cultural probes, análisis de productos de la competencia, revisión de bibliografía, búsquedas web, comparaciones culturales, mapas mentales.

Actividades de consulta: Cuestionarios y surveys, focus groups, entrevistas, investigación de mercado.

Actividades de evaluación: Checklists, external decision making, intuición, campañas de producto, matriz de evaluación, crowdsourcing.

Actividades de comunicación: Creación de presentaciones, creación de informes, presentaciones visuales y modelos, enganchar al público.

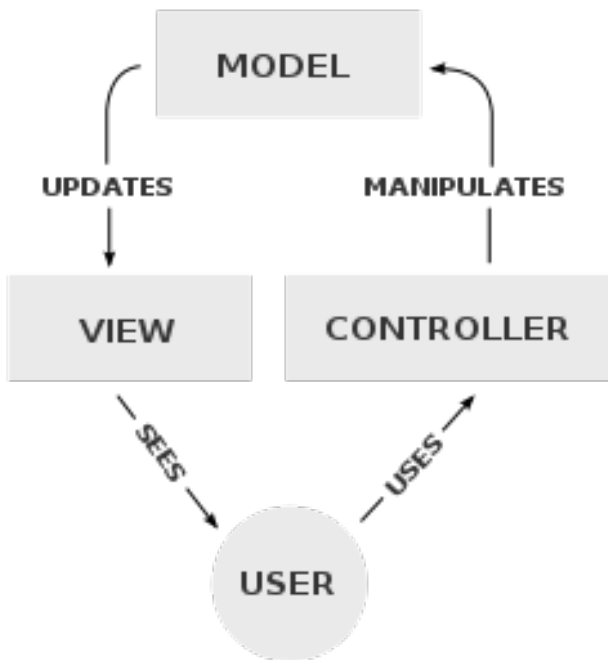
La evolución en el embudo del diseño pasa por fases iterativas que, según diferentes autores, pueden tomar diferentes nombres como: Entender → Observar → Visualizar → Revisar → Implementar → Entender; o bien Mirar → Aprender → Preguntar → Prototipar → Probar → Evaluar → Comunicar → Entender; o bien Research → Brief → Concept design → Design development → Detail design → Production (aunque los términos son intuitivos, puede consultarse [?] para una explicación de los mismos).

II. DISEÑO E IMPLEMENTACIÓN DE APLICACIONES MULTIMEDIA

El diseño de aplicaciones informáticas es una actividad creativa que consiste en identificar y programar los componentes software que componen la solución que satisface los requisitos. Para satisfacer los requisitos de la aplicación, debe establecerse la arquitectura de la solución, debe decidirse cómo introducir y obtener la información, y finalmente cómo implementar la solución diseñada como un programa informático ejecutable. Aunque se trata de un proceso creativo, los equipos de desarrollo suelen ser sistemáticos y seguir las etapas de selección de la estrategia de diseño, el establecimiento de arquitectura del programa, la definición de los diagramas de diseño y la programación de la solución final.

Con respecto a la **elección de la estrategia de diseño**, se hace referencia a la decisión de si el programa sigue una estrategia **top-down**, realizando primero la estructura global del programa y después cada módulo individual o una estrategia **down-top**, en la que primero se hacen los componentes individuales finalmente se unen. También es posible realizar estrategias híbridas, pero en este punto es importante tomar decisiones al respecto que permitan planificar y valorar pros y contras. Por ejemplo, si estamos haciendo un sitio web, una estrategia top-down priorizaría el esquema general del sitio web, mientras que una estrategia down-top haría primero cada una de las secciones (o incluso primero los recursos

Figura 1: Patrón Modelo-Vista-Controlador (MVP) (imagen de dominio público vía Wikipedia)



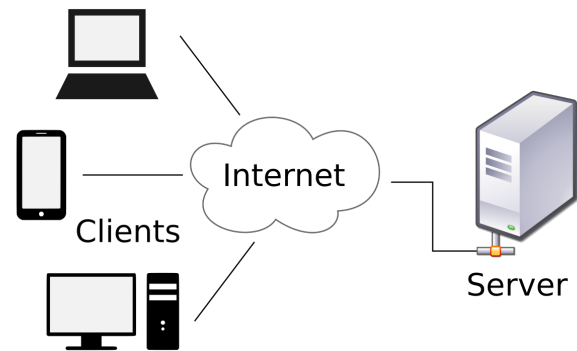
multimedia) y sólo al final uniría cada sección. Las estrategias **híbridas** son muy frecuentes e incluso deseables porque permiten ir realizando los distintos apartados a la vez que puede irse comprobando el resultado de forma incremental.

Con el **establecimiento de la arquitectura del programa** se establecen cuáles son las componentes principales que se integran en el programa y cómo se relacionan entre ellos. Los **diagramas de diseño** muestran, de forma semejante, los componentes de nivel inferior y también su relación entre ellos. Para facilitar estas definiciones se suelen emplear **patrones de diseño**, que establecen configuraciones típicas que se ajustan razonablemente a un amplio número de problemas.

Dos de los patrones de diseño más empleados en desarrollo de aplicaciones multimedia son el patrón **Modelo-Vista-Controlador (MVC)** y el modelo cliente servidor [1]. En el patrón MVC, los componentes de presentación e interacción se separan de los componentes responsables de acceder a los datos. El sistema se estructura en tres componentes lógicos que interactúan entre sí. El componente Modelo gestiona los datos del sistema y las operaciones asociadas con los datos. El componente Vista define y gestiona la manera en la que los datos son presentados al usuario. El componente Controlador gestiona la interacción con el usuario y envía estas interacciones a los componentes de la Vista y del Modelo.

En el patrón **cliente-servidor** la funcionalidad del sistema se organiza en servicios, de manera que cada uno de los servicios se ofrece desde distintos servidores. Los clientes son usuarios de dichos servicios y acceden a los servicios cuando los necesitan. Por ejemplo, un servidor puede ofrecer el acceso a un determinado vídeo. El servicio de vídeo puede

Figura 2: Esquema de aplicación distribuida empleando el modelo cliente servidor (imagen de D. Vignoli CC BY-SA vía Wikipedia)



ser ofrecido a más de un cliente simultáneamente.

En el desarrollo de aplicaciones multimedia, el uso de un modelo arquitectónico o de un patrón de diseño determinado puede venir impuesto por la **herramienta de autor** empleada. Las herramientas de autor facilitan el diseño de la solución haciendo que el desarrollador se olvide de cuestiones relativas a la programación de bajo nivel y simplificando el diseño de la interfaz y de la interacción. En diseño web por ejemplo, el uso de sistemas como WordPress permite desarrollar páginas web sin conocer el lenguaje HTML.

La **programación de la solución final** consiste en llevar a código ejecutable el diseño establecido. Incluso cuando se emplea una herramienta de autor, conocer el lenguaje de programación sobre el que se asienta la herramienta de autor suele ser importante para ganar en productividad. Un buen ejemplo de este hecho es el caso del diseño web: aunque se utilice WordPress para diseñar sitios web, conocer PHP y JavaScript supone una ventaja importante (a veces imprescindible) a la hora de incorporar nuevas funcionales.

Uno de los componentes software más importantes de todo programa informático es la interfaz de usuario. La **Interacción Persona-Computadora (IPC)** o *Human-Computer Interaction (HCI)* en inglés es la rama de la informática que se encarga de estudiar cómo interactúan los usuarios con los sistemas informáticos. Cuando los usuarios interactúan con un sistema informático lo hacen empleando la **interfaz de usuario** (en inglés *user interface, UI*). El diseño de la interfaz de usuario trae asociado el estudio de un importante número de aspectos que serán tratados en las siguientes secciones. Otros componentes relacionados con los datos o con las comunicaciones quedan fuera del alcance de este curso.

El desarrollo de los componentes multimedia de la interfaz, como pueden ser las imágenes, los vídeos y las animaciones, son parte fundamental de todo proyecto multimedia. Dedicamos una serie de temas específicos para tratar los aspectos relativos a estos recursos multimedia en otros temas específicos del curso.

III. DISEÑO DE LA INTERFAZ CENTRADO EN EL USUARIO

Desarrollar buenas interfaces de usuario es una necesidad hoy en día porque el uso del software ya no es exclusividad de expertos. Las aplicaciones informáticas son utilizadas por el gran público y no podemos asumir que tengan conocimiento experto para emplear los programas. Una mala interfaz de usuario puede hacer que un cliente se pase a la competencia que le ofrece un programa más ajustado a sus necesidades concretas.

III-A. Definición de usabilidad

Una interfaz de usuario se valora como mejor o peor en términos de lo que llamamos usabilidad. **Usabilidad** se refiere a la facilidad con la que las personas pueden utilizar una herramienta particular o cualquier otro objeto fabricado por humanos con el fin de alcanzar un objetivo concreto. El estándar ISO 9241 [2] define usabilidad como “hasta qué punto un producto puede ser utilizado por determinados usuarios para alcanzar los objetivos especificados con eficacia, eficiencia y satisfacción en un determinado contexto de uso”. La **efectividad** (*effectiveness*) es la precisión y completitud con la que la cual unos usuarios específicos pueden alcanzar los objetivos especificados en un contexto determinado. La **eficiencia** (*efficiency*) se define como los recursos consumidos en relación con la precisión y completitud alcanzada. La **satisfacción** es el grado de confort y de aceptación del trabajo realizado por el sistema por parte de los usuarios y otras personas afectadas.

Es importante destacar que en la definición de usabilidad se indica que hay que tener en cuenta el tipo específico de usuario con el que se trabaja y el contexto en el que se hace el trabajo. Ambas cosas son indisolubles del concepto de usabilidad. Llevado al modelo de comunicación básica, siempre hay que tener en cuenta al receptor del mensaje y el contexto en el que se realiza la comunicación. En el caso de aplicaciones informáticas, el producto es el propio software a desarrollar y el mensaje tiene que ver con las tareas concretas que esperamos que realicen los usuarios con el programa.

Desarrollar interfaces de usuarios pobres trae consecuencias negativas que tienen que ver con la frustración de los usuarios, que no estarán satisfechos con el uso de las aplicaciones. Realizar una tarea con un sistema informático mal diseñado puede suponer tener que emplear más tiempo del necesario para realizar actividades con consecuencias negativas desde el punto de vista de pérdida de productividad y aumentos de costes económicos.

III-B. Claves sobre usabilidad

Los usuarios buscan calidad y credibilidad. Si un sitio tiene contenido de calidad, los usuarios están dispuestos a “lidiar” con anuncios y un mal diseño del sitio. El contenido puede ser más importante que otros aspectos. Podemos pensar que si ofrecemos contenido de calidad pues, el usuario será fiel a nuestro sitio web o aplicación. Sin embargo, los usuarios de sitios web son impacientes e insisten en la gratificación inmediata Si un sitio web no le satisface, el usuario se irá a otra

página Si se impone una carga cognitiva alta o la navegación no es intuitiva, el usuario se irá a otra página.

Entender al usuario y el objetivo y modo con el que el usuario accede al sistema es clave para crear interfaces usables. Para ello, hay que tener en cuenta tres principios básicos que son: no hacer pensar al usuario, entender cómo el usuario accede a la información electrónica y facilitar la navegación.

III-B1. No hacer pensar al usuario: La primera de las claves sobre usabilidad es no hacer pensar al usuario. Cualquier esfuerzo cognitivo al que se obligue a los usuarios será penalizado. La interfaz debe estar pensada para que el usuario necesite el menor tiempo de adaptación posible (a ser posible tiempo 0) para poder emplear la aplicación. Para ello es importante **seguir las convenciones de diseño**, haciendo que el diseño se corresponda con las expectativas del usuario. Innova sólo si sabes que lo nuevo es mejor.

Para evitar hacer pensar al usuario, los títulos y los formatos de los elementos son importantes. Es importante utilizar rótulos obvios, por ejemplo, si se ofrece diseñar una web en la que se ofrece trabajo, un botón con el título “Jobs” será más evidente que un enlace en el que ponga “Employment Opportunities” o “Job-oRama”.

III-B2. Entender cómo el usuario accede a la información: Los usuarios no leen ni analizan sistemáticamente la interfaz, más bien **los usuarios son escaneadores de información**. Los usuarios buscan puntos fijos, no leen el contenido de izquierda a derecha, arriba abajo. La figura muestra el patrón “F” obtenido a partir de una herramienta de eyetracking. En el caso del diseño web este hecho es muy claro: los usuarios más que lectores son escaneadores que hacen recorridos rápidos de los elementos de la página buscando enlaces que sean de su interés para poder hacer click sobre ellos.

Para diseñar para escaneadores y no para lectores se debe tener en cuenta:

1. Lo más importante debe destacar más. Para hacer destacar los elementos, puede emplearse el tamaño o el contraste.
2. Las cosas relacionadas lógicamente deben estar relacionadas visualmente. Por ejemplo los menús o las barras laterales deben estar claramente marcados.
3. Las cosas están anidadas visualmente para mostrar las partes que están relacionadas entre sí. Por ejemplo, las distintas entradas de una web deben estar en un marco común.

Hay que evitar las barreras de entrada. Los usuarios de un sitio web quieren probar el servicio que éste ofrece. Cuantas menos acciones le pidamos al usuario, más fácil será que permanezca. No pidas suscripción o registro antes de que el usuario haya podido saber si el sitio le interesa.

Es importante gestionar la atención del usuario para lo cual hay que utilizar elementos que llaman su atención para guiar al usuario en la navegación. El ojo humano no busca de forma lineal. Se fija y busca patrones concretos, movimiento, etc. Las imágenes y vídeos llaman más la atención, el texto en negrita, el texto en grande, los botones que llamen a la acción son excelentes pistas para que el usuario fije su atención en los puntos que nosotros queramos.

Otra consecuencia que se deriva de la forma de acceder a la información es que hay que **evitar texto innecesario**.

Demasiado texto puede crear una impresión nefasta a un usuario.

Por último y no menos importante, para hacer interfaces usables hay que conocer a los usuarios, **intentar pensar como ellos**. Un usuario cuando arranca una aplicación o cuando accede a una página web se preguntará ¿Qué es esto? ¿Qué puedo hacer? ¿Qué podemos hacer? ¿Qué se supone que puedo hacer? Muchas veces, cuando el usuario empieza tiene muy claro por qué ha abierto la aplicación. En ese caso es importante anticiparse y dejarle claro cómo debe utilizar la aplicación para resolver las tareas que ha venido a hacer. La pregunta ¿Por dónde empezar? Debe estar resuelta nada más arrancar la aplicación sin necesidad de leerse ninguna documentación adicional. Acabar con los manuales de usuario es una ambición de las aplicaciones y sitios web usables.

III-B3. Facilitar la navegación: Es extremadamente importante facilitar la navegación.

Existen tres tipos principales de estilos de navegación:

- Navegación jerárquica, donde hay una relación padre hijo entre las secciones o páginas. La forma de navegar suele ser mostrando en cada nodo padre los hijos a los que se puede navegar con la opción adicional para “volver a superior”.
- Navegación plana, donde las páginas principales se relacionan al mismo nivel. La forma de navegar suele ser “anterior”, “siguiente”
- Navegación content-driven o experience-driven donde, en función de la aplicación las conexiones son mucho más complejas.

La navegación debe ser intuitiva, natural y sin quitar atención a los contenidos. Para facilitar la navegación intuitiva debemos utilizar elementos de navegación estándar, conocidos por los usuarios. No debemos hacer perder tiempo al usuario pensando para qué servirá determinado control.

Siempre debe estar claro el camino que nos ha llevado a estar donde estamos. Una forma de conseguirlo es emplear encabezados descriptivos con un icono común o una línea gráfica uniforme para toda la aplicación.

Siempre se debe saber qué es lo siguiente o cuáles son las alternativas. El camino ha de ser lógico, practicable y fácil de seguir. Una forma de alcanzar este objetivo es empleando secciones y subsecciones bien señalizadas. Si hay muchos niveles, deben emplearse menús.

El usuario debe saber en qué punto se encuentra de la estructura en cada momento. Para ello, debemos resaltar de alguna manera la sección en la que se encuentra el usuario dentro de toda la estructura, por ejemplo marcando la opción dentro del menú o marcando la pestaña.

Debemos organizar la información con una estructura que ayude a organizar el interfaz. Un buen ejemplo de organización eficaz de la información está en Wordpress. La organización en páginas, entradas, categorías es muy práctica para después organizar las zonas de visión, los menús y los gadgets que permitan acceder rápidamente a los contenidos.

IV. PRINCIPIOS Y GUÍAS DE DISEÑO DE LA INTERFAZ

IV-A. Áreas de diseño

Distinguimos entre diferentes áreas de diseño en función de la tecnología para la que se haga la interfaz. Cada una de estas áreas tiene sus propias guías de diseño que conviene tener en cuenta. Las áreas de diseño principales son los Graphical User interfaces (GUI), las páginas web, los dispositivos móviles y los sistemas empotrados.

Las GUI son necesarias para aplicaciones de escritorio y desarrollo de aplicaciones en los distintos sistemas operativos. Apple, Microsoft y Linux ofrecen sus propias guías de estilo y también se ofrecen guías de estilo para diferentes lenguajes de programación como son la de Java Sun Microsystems.

Para las páginas web existe abundante bibliografía y recomendaciones del consorcio WWW. Desarrollar interfaces para la web es un desafío porque es importante captar la atención de usuarios que fundamentalmente son rastreadores de información y no usuarios de sistemas.

Para las aplicaciones móviles, tanto Android como Apple ofrecen sus propias guías de estilo. Estas guías se basan en el concepto de “actividad” asociada a su propia pantalla y a la navegación entre actividades. Apple revisa las interfaces de cada aplicación que se quiere publicar para el Apple Store, por lo que vigilar las normas de edición es muy importante.

Para los sistemas empotrados existen menos bibliografía por lo que se deja más libertad a los diseñadores. Los propios fabricantes terminan identificándose por cómo configuran la interfaz de sus dispositivos. Pensemos por ejemplo en los mandos de control de los automóviles y cómo cada marca establece sus criterios.

IV-B. Cualidades del diseño

Todo diseño debe responder a dos vectores fundamentales que son la utilidad y la estética. El principal objetivo de todo diseño es la utilidad, en sentido de que los productos diseñados deben servir para algo. Es quizá la principal diferencia con el respecto al arte: mientras que el arte no responde a ningún propósito funcional, el diseño sí. No obstante, como creativo, el diseñador debe anhelar que el resultado tenga un aspecto agradable y con ello llegamos al segundo objetivo de todo diseño: ser estéticamente atractivo.

Una presentación multimedia, una página web, un vídeo corto, como diseño en sí mismo, debe cumplir con esta circunstancia. Por un lado debe responder a una utilidad y por otro debe ser atractivo desde un punto de vista estético.

Cualidades estéticas de un diseño tienen que ver la búsqueda de la armonía en los elementos que lo componen, lo cual se consigue jugando con los cinco recursos básicos del diseño: el ritmo o repetición sistemática de formas o de tonos; con la proporción o búsqueda de proporciones atractivas; con el equilibrio o relación balanceada de las masas del diseño; con el movimiento para dirigir intencionadamente la mirada del espectador; y con el ornamento para completar la estética.

Cualquier trabajo que puede analizarse como una serie de partes constituyentes y por la relación entre ellas se llama, en abstracto, un diseño. Puede ser tanto un edificio como un póster, una presentación multimedia o un objeto de consumo.

Cualquier diseño de éxito tendrá las cualidades de aptitud y belleza. La aptitud del diseño depende de su propósito: una casa puede ser horrible y proteger. La belleza depende de cualidades estéticas que tendrán en cuenta: La armonía (de la forma, el tono, el color, la concepción) El equilibrio y la proporción (de masas, formas y color) El ritmo (de las formas, líneas, tonos y color).

Armonía, equilibrio, proporción y ritmo son los principios de diseño que se repiten a lo largo de diferentes civilizaciones para conseguir un resultado estético. En este tema analizamos cada uno de estos aspectos centrándonos principalmente en su aplicación al diseño web y de presentaciones multimedia.

Comenzamos con unas definiciones básicas tomadas de [3]. Se trata de uno de los primeros tutoriales conocidos para la formación en la industria gráfica. A pesar de su antigüedad, tiene total vigencia en lo que se refiere a la presentación de los aspectos fundamentales.

Armonía: “estado integral de normalidad en la relación unos elementos con respecto a otros”. Este estado es tal que no queremos cambiar o modificar ningún detalle o característica.

Balance: “estado de equilibrio”. En diseño, hace referencia al equilibrio o balance de la atracción visual entre las diferentes masas.

Proporción: “relación comparativa de una cosa con otra” con respecto al tamaño.

Ritmo: “movimiento caracterizado por la aparición de elementos recurrentes”.

Ornamento: “Adornos que completan la estética sin aportar contenido aparente”.

A continuación desarrollamos cada uno de estos apartados.

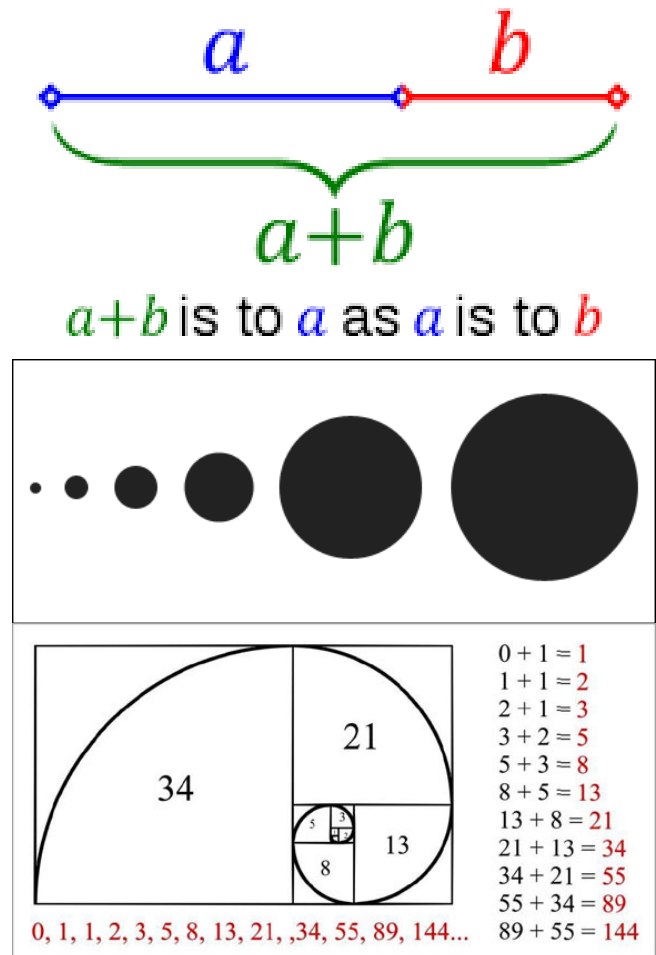
IV-B1. El ritmo: El ritmo, como decíamos, se consigue con la repetición recurrente de elementos. Distinguimos entre ritmo en las formas y ritmo en el tono.

El **ritmo en las formas** se consigue con la recurrencia regular de formas similares en un diseño. Puede ser mediante la repetición de la misma forma o mediante el incremento o decremento regular del tamaño de las formas en un diseño. Cuando es la misma forma la que se repite hablamos de **ritmo regular**; cuando la forma incrementa o decrecienta su tamaño hablamos de **ritmo progresivo**.

Ejercicio: Existen diferentes blogs de diseño web en los que se discuten las cualidades de diseño en base a ejemplos. Visite el artículo de codrops¹ y busque ejemplos de páginas webs que consiguen ritmo visual al repetir diferentes geometrías (también ritmo progresivo). Haga una búsqueda de nuevos sitios similares y haga una lectura detallada de los comentarios de los ejemplos. Intente realizar comentarios similares con otras webs que le resulten atractivas.

El **ritmo en el tono** se consigue con la recurrencia de tonos similares en un diseño. Generalmente se emplea la progresión regular de tonos de un mismo color básico que se consigue oscureciendo o iluminando de forma regular.

Figura 3: Proporciones basadas en la relación áurea (Imágenes de dominio público vía Wikipedia).



IV-B2. Proporción: La proporción es la relación placentera de los tamaños y dimensiones. El uso de masas de igual tamaño en el diseño resulta monótono. El ojo encuentra más interesante que haya variedad. Encontrar la variedad ideal es difícil, pero se encuentran invariantes intemporales que conviene conocer y tener en cuenta.

La **relación ancho y alto** de los elementos del diseño o del marco en el que se incluye el diseño se elige con valores enteros. 3:2; 4:3; 16:9 son los más utilizados. Otras combinaciones suelen resultar también armónicas. El tamaño del folio de papel es Estados Unidos, conocido como *letter* tiene una relación de aspecto 17:22. La norma europea para el folio de papel (norma DIN) no cumple la relación de aspecto basada en números enteros. En este caso la relación es $1 : \sqrt{2}$. En este caso se dio prioridad a la ventaja técnica de que al dividir la hoja en dos partes repetidamente, la relación de aspecto era la misma.

Un problema que suele surgir con frecuencia es el de posicionar un elemento en relación a un marco. Emplear divisiones basadas en números enteros suele funcionar en la mayoría de los casos. Una relación 3:2 (tres partes por debajo, dos por encima) es empleada en la mayoría de los títulos por

¹<https://tympanus.net/codrops/2011/08/19/developing-visual-rhythm-in-web-design/>

considerarse un punto de gran fuerza visual.

Para organizar las divisiones basadas en números enteros, la relación aurea, basada en las series de Fibonacci suele ser también muy efectiva. Una serie de Fibonacci se obtiene sumando repetidamente los dos números anteriores de la serie para obtener el número nuevo. Si empezamos por el 0 y el 1, la serie sería 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 etc. . . . Esta secuencia permite levantar rectángulos, realizando divisiones armónicas sobre el mismo. Los rectángulos que se van formando, y los puntos de división suelen formar relaciones armónicas. Las relaciones formadas con estas secuencias se denominan relaciones áureas, y aparecen frecuentemente en la naturaleza, rostros humanos, cuerpos etc. . . y han sido empleadas de forma masiva en la arquitectura y el arte clásico. Se emplean también en fotografía y gobiernan la estructura de múltiples presentaciones multimedia y diseños web. La figura 3 muestra las divisiones típicas basadas en la relación de Fibonacci.

Ejercicio: Existen diferentes blogs de diseño web en los que se discute sobre el uso de la relación áurea y la espiral áurea en el diseño de interfaz y en el diseño web. Lea el artículo de Apiumhub ² al respecto. Haga una búsqueda de nuevos sitios similares y haga una lectura detallada de los comentarios de los ejemplos. Intente realizar comentarios similares con otras webs que le resulten atractivas.

Aunque la relación áurea se considera una relación armónica que suele funcionar en la mayoría de aplicaciones, conviene tener en cuenta que no es la única relación que funciona, y otras claves como puede ser la de realizar divisiones basadas en números enteros también suelen funcionar.

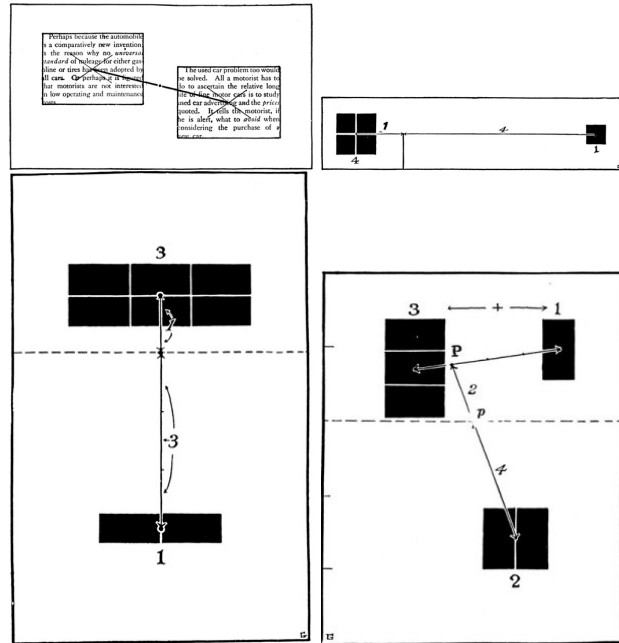
La división del marco de trabajo, y la colocación de elementos en puntos de referencia, no es una cuestión que deba dejarse a la improvisación. La sensación de orden suele ser agradable para el ser humano, y despreciar la proporción suele significar contribuir a dar sensación de desorden o descuido, haciendo que el diseño no sea armónico.

IV-B3. Equilibrio: El equilibrio óptico es el resultado del ajuste que ocurre entre las masas de los elementos dentro de los márgenes de un diseño. Es similar al equilibrio de los balancines; en el sentido de que ambos son el resultado de equilibrar fuerzas. En el balancín, la fuerza es la gravedad, en el diseño la fuerza es la atracción de la vista, que depende del tamaño y del tono de los elementos del diseño.

La atracción que posee un elemento de diseño depende directamente de su tamaño y tono. Un elemento de cuatro centímetros cuadrados de color negro sólido sobre fondo blanco, será el doble de fuerte en atracción visual de lo que lo es un elemento de dos centímetros cuadrados negro sólido. También será el doble de atractiva que una masa de cuatro centímetros cuadrados gris neutro (el gris de la mitad del valor que el negro).

El valor de atracción de las masas depende mucho de su contenido: fotografías, texto con diferente espacio, grosor etc. . . Una fotografía con un rostro en primer plano suele ser más atractiva que un paisaje; un texto con un eslogan suele

Figura 4: Cálculo de composiciones equilibradas en función de su fuerza visual.



ser más atractivo que un texto largo con una descripción; un texto en negrita atrae más que un texto sin resaltar. Todos estos aspectos han de tenerse en cuenta a la hora de buscar el equilibrio.

Igual que un balancín necesita un punto de apoyo, en diseño el punto de apoyo viene determinado por la posición relativa de los elementos. El **punto de equilibrio** debe tener relación con las proporciones del diseño. Cuando posicionamos dos elementos con la misma fuerza visual, el punto de equilibrio está en el medio. Cuando la fuerza visual de los elementos no es la misma, el punto de equilibrio no es equidistante con respecto a los centros de los elementos. La relación entre la fuerza visual de los elementos marca la distancia entre los centros de los elementos y el punto de equilibrio: cuando más fuerza visual, más proximidad al punto de equilibrio.

Una regla muy recomendable a la hora de colocar los puntos de equilibrio es posicionar los elementos para que los puntos de equilibrio queden posicionados en una división que tenga en cuenta la proporción: por ejemplo por estar en relación aurea o por ser una división basada en números enteros. Cuando hay más de dos masas, se deben equilibrar de dos en dos teniendo en cuenta las mismas consideraciones.

La figura 4 muestra cómo calcular puntos de equilibrio para distintas combinaciones de elementos.

Ejercicio: Existen diferentes blogs de diseño en los que se discute sobre cómo conseguir equilibrio en páginas web. En el blog codrops³ se muestran bastantes ejemplos. Lea los argumentos del articulista e intente desarrollar argumentos similares con otras páginas web.

²<https://apiumhub.com/tech-blog-barcelona/golden-ratio-in-web-design/>

Figura 5: Ejemplo de composición que consigue el equilibrio utilizando simetría a la izquierda. A su derecha, composición más compleja en la que hay equilibrio.

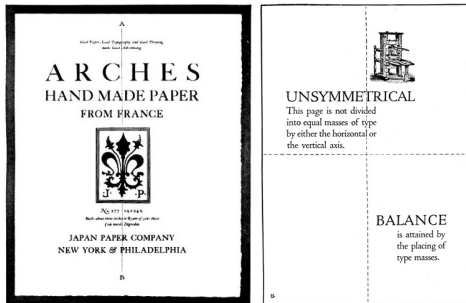
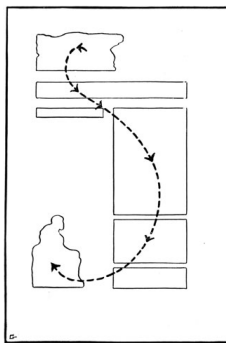


Figura 6: Composición en la que hay movimiento.



Una forma sencilla a la vez que efectiva de garantizar el equilibrio es utilizar la simetría. La **simetría** se consigue cuando dos partes del diseño son iguales en cada detalle de manera que si se pliega una mitad sobre la otra se superponen de forma perfecta. Entonces tenemos un estado de simetría y el diseño se dice que es simétrico. La línea sobre la cual el diseño se pliega se llama **eje de simetría**.

La ausencia de simetría en un diseño impone el carácter de **variedad**, que se define como un estado de inequidad en el orden de las partes del diseño. La variedad precisa de equilibrio para componer un balance armónico. Aunque la simetría suele funcionar en cualquier diseño, la variedad aporta un plus de belleza que la convierte en deseable en cualquier diseño. El riesgo de la variedad es perder el equilibrio y la sensación de proporción en la composición. La figura 5 ilustra cómo puede explotarse la variedad.

IV-B4. Movimiento: Muy relacionado con la variedad está la cualidad de movimiento. En cualquier composición, el ojo del observador está conectado con varias partes en sucesión, dependiendo de su carácter y posición relativas. Esta cualidad llamada movimiento, será más pronunciada cuando unas partes se aproximen a otras de forma más definida.

En un cartel o en un póster, el ojo del espectador puede ser dirigido para respetar un orden de lectura pre-establecido. En aplicaciones multimedia y diseño web el movimiento se explota para dirigir la mirada del usuario en que está empleando la aplicación y para mejorar la usabilidad.

La figura 6 muestra composiciones en la que hay movimien-

to.

Ejercicio: De nuevo se anima a visitar la web de codrops⁴ para ver ejemplos en los que se explota el movimiento como recurso visual. Intente analizarlos y desarrollar un comentario sobre las cualidades del diseño de los mismos.

IV-B5. Ornamento: A la vez que los elementos del diseño atienden a un esquema funcional, el ornamento sólo atiende a una necesidad estética. El ornamento se añade para decorar el diseño una vez concebido, con la intención de hacerlo más atractivo.

El ornamento puede ser tanto simbólico como puramente estético. Simbólicos si sus elementos tienen alguna relación con el diseño. Estéticos si son formas elegidas sólo por su belleza.

Destellos, sombras, degradados son recursos muy empleados en presentaciones multimedia. El ornamento puede responder a tendencias o modas y son un recurso muy útil para cambiar el aspecto general de producto final sin modificar esencialmente el diseño.

Los cambios en la tipografía pueden relacionarse directamente con la mejora del ornamento. También el uso de fondos de pantalla sobre los que posicionar los elementos.

Ejercicio: Escriba en el buscador de Google el término “fondos para páginas web”. Intente argumentar sobre la connotación que aportan estos fondos.

V. CONTROLES PARA LA INTERACCIÓN

Distinguimos entre controles que sirven para que los usuarios introduzcan datos que utilizan los sistemas informática, controles que sirven para que el sistema presente información, sistemas que permiten organizar los elementos del interfaz y elementos que sirven para gestionar la navegación entre diferentes pantallas.

V-A. Controles de introducción de datos

La introducción de datos en las aplicaciones suele ser un proceso tedioso y provoca abandono. Independientemente de los controles que se empleen, algunos consejos para facilitar la introducción de datos son:

- Siempre que sea posible presenta opciones.
- Toma la información del sistema siempre que sea posible. Si lo puedes evitar, no se la pidas al usuario.
- Proporciona valores por defecto razonables.
- Antes de habilitar datos accesorios asegúrate de que los principales ya están introducidos.
- Valida los campos de forma dinámica.
- Los campos serán obligatorios sólo cuando sean necesarios realmente.
- Facilita la navegación sobre las listas de valores.

⁴<https://tympanus.net/codrops/2011/01/14/visual-movement-flow/>

Figura 7: Ejemplos de fondos con degradados y motivos orgánicos muy empleados en tecnologías de la información.



- Muestra hints (mensajes con consejos) cuando sea necesario.

A continuación, presentamos los controles más frecuentes en la introducción de datos en la interfaz de usuario.

V-A1. Checkbox: El checkbox es una casilla de verificación que aparece generalmente organizada en grupos y que permite seleccionar una o varias de las opciones presentadas.

También se conocen como *toggle buttons* o interruptores. Las opciones están activadas o desactivadas. En este caso la apariencia visual se asemeja a la de un interruptor.

V-A2. Opción: El control llamado opción *choice* aparece generalmente organizado en grupos también pero, a diferencia del checkbox, en esta ocasión sólo puede seleccionarse uno de los elementos del grupo. Se utiliza el tecnicismo de *Radio button* para referirnos a este recurso.

V-A3. Pickers o combos: Pickers o combo o comobox es un marco combinado en el que aparece un texto, una lista con diferentes valores seleccionables y un scroll para desplazarse por la lista. Los valores de despliegan in-situ o debajo.

Conviene utilizar para la lista un orden de valores lógicos y predecibles. Es importante que al desplegarse la lista no obligue a cambiar de pantalla para seleccionar valores. Cuando hay muchos datos, es mejor una tabla que una lista. Este es el caso del colorpicker, en el que se presentan colores para elegir uno de ellos.

Cuando se permite seleccionar más de un dato de una lista, hablamos de **listas de selección múltiple**. En este caso el texto

superior no tiene sentido.

V-A4. Sliders: Un slider es una barra horizontal (puede ser vertical) con un mando que se desplaza sobre dicha barra. Al desplazarse, permite seleccionar valores numéricos entre un máximo y un mínimo. Generalmente se establece la posibilidad de establecer un paso para no ofrecer un número de valores discretos.

Cuando se usan, conviene personalizar para que el valor concreto pueda visualizarse a la vez que manipula. No debe usarse para controlar el volumen: mejor lanzar una ventana especial.

V-A5. Steppers: Un stepper incrementa un valor numérico con un control, generalmente unas flechas. Cada vez que se pulsa, el valor numérico se incrementa o decrementa.

Conviene usarlo para valores pequeños porque si no, el número de toques que hay que aplicar para hacer cambios de una magnitud elevada sería excesivo. El hecho de que al tocar en stepper cambia el valor debe ser obvio.

V-A6. TextFields: Son línea de texto de ancho fijo que sirve para introducir información. Generalmente el formato del texto introducido suele ser libre, pero también se puede restringir para controlar que se introduzcan sólo números o emails o fechas etc... En estos casos algunos sistemas ofrecen teclados especiales para facilitar la introducción de datos y controlan que la información introducida esté en el formato correcto. También se puede hacer que el texto introducido no sea visible para permitir introducir passwords.

Cuando se admite más de una línea se denomina TextArea.

Se recomienda mostrar un mensaje para indicar el propósito. Estos mensajes se llaman “hints” y aparecen en el textField cuando el usuario no ha escrito aún nada.

Ejercicio Vaya a la guía de desarrollo de aplicaciones iOS⁵ y busque información sobre los componentes tipo “Selection and Input”. ¿Cuáles son las principales diferencias entre estos controles y los controles de que encontramos en las páginas web?

V-B. Controles de presentación de datos

V-B1. Etiquetas: Las etiquetas son campos de texto que describen algún elemento de interface o aportan un mensaje

Como todo texto que aparezca en el interfaz, deben ser legibles. Si se utiliza un tipo de letra especial hay que cuidar que se lea bien.

V-B2. Feedback o status: Ayuda al usuario a saber lo que el programa está haciendo. Los sistemas suelen tener una posición concreta en la interfaz para mostrar el status, por ejemplo en una barra inferior.

El estatus no debe ser obstrusivo, en el sentido de que debe permitir seguir trabajando sin molestar. Es necesario evitar alertas innecesarias para no despistar al usuario.

El status puede ser háptico, por ejemplo una vibración para avisarnos de que has completado una acción.

⁵<https://developer.apple.com/design/human-interface-guidelines/components/selection-and-input/text>

V-B3. Indicadores de progreso: Indican el estado de avance de alguna acción que esté realizando el ordenador y que lleve un tiempo poder acabarla. Por ejemplo cuando se carga el programa o cuando se lee algún fichero grande.

Son importantes porque hay que evitar que el usuario esté sentado frente a la interfaz sin que pase nada. Es importante que aparezca este elemento para indicar que la aplicación no se ha colgado.

No debe confundirse con las marcas de estatus. Las barras de progreso indican que algo está procesándose y, si es posible, cuánto falta para que termine, y las marcas de estatus indican cuando ha terminado y/o en qué fase se encuentra.

Es mejor un indicador de progreso frente a una barra de actividad o estatus por ser más informativo. Debe mantenerse en movimiento mientras el proceso no se haya acabado. Se pueden añadir una etiqueta para indicar a qué hace referencia. Por ejemplo un mensaje tipo “cargando” o “identificando”.

V-B4. Imágenes, vídeos y canvas: Es una zona en la que se puede colocar una imagen, un vídeo o un canvas. El canvas es un área de dibujo 2D o 3D en el que se pueden representar elementos gráficos vectoriales. Los canvas suelen ser interactivos y dinámicos, de manera que es la mejor opción para representar animaciones.

Estos elementos tienen una alta fuerza comunicativa que motiva al usuario y aporta información. Debe tenerse cuidado con la integración de estos elementos con el resto del interfaz, con el tamaño, y tener en cuenta que siempre deben responder a una finalidad concreta.

Los vídeos y animaciones ilustran movimiento y pueden aportar una importante realimentación, pero hay que tener en cuenta que atraen fuertemente la atención del usuario. Las animaciones son importantes para mostrar que el sistema está trabajando.

El sonido es imprescindible para determinados usuarios (discapacitados visuales) y en determinados entornos de trabajos en los que la vista deba estar centrada en otras tareas (conducción). Las funciones del sonido en interacción persona computadora tienen que ver con reforzar el componente visual, con confirmar que la tarea ha terminado y con atraer la atención del usuario. El sonido tiene el riesgo de ser intrusivo, en el sentido de que puede molestar a otras personas que estén cerca. Como norma general es mejor presentar la información en texto que en audio porque es más rápido y eficaz.

Ejercicio: Consulte la guía de usabilidad del Gobierno Americano⁶ e indique qué controles de presentación de información aparecen citados en la guía que no aparecen en este texto.

V-C. Estructuración de contenido

V-C1. Las ventanas : La ventana principal es la zona en la que la aplicación presenta datos y donde se despliega la información. Las **ventanas primarias** deben reflejar las principales tareas para las que está pensada la aplicación. Por ejemplo en Word, la ventana principal permite escribir texto, en CATIA permite diseñar etc... Suele haber pocas por

aplicación y se suelen utilizar como base a la que se vuelve constantemente. En la ventana principal se suele incluir el marco, la barra de menús, la barra de herramientas, controles de desplazamiento etc... .

Las **ventanas secundarias** complementan a las ventanas primarias, proporcionando funcionalidad y apoyo al usuario. Los **cuadros de mensaje** son un tipo de ventana secundaria que permite comunicar mensajes al usuario que deben ser tenidos en cuenta. Debe evitarse el uso de jerga, los términos informáticos, códigos de error... y también deben evitarse mensajes innecesarios del tipo “No hay ningún error”. Los **cuadros de diálogo** se utilizan cuando es necesaria información adicional para llevar a cabo una tarea. Un ejemplo típico es el cuadro de diálogo para imprimir un documento. Las cajas de diálogo pueden ser complejas, ya que combinan muchos elementos (campos de texto, etiquetas, botones, etc.). Hay que diseñarlas con cuidado, disponiendo los elementos de forma cuidadosa en la pantalla.

Una distinción básica es la que diferencia entre diálogos modales y no modales. Un **diálogo modal** es una ventana secundaria que inhabilita la ventana principal. Hay que usarlas si la situación requiere atención inmediata y los datos requeridos son imprescindibles para continuar. Tienen la ventaja de que evitan errores, proporcionan sensación de cierre del diálogo. Un **diálogo no modal** no bloquea la interacción con otras ventanas. Un ejemplo sería la opción de buscar que aparece en el Acrobat Reader. Se recomienda colocarlas en “barras laterales” para no entorpecer el trabajo del usuario.

En aplicaciones móviles, las diferentes ventanas se llaman **actividades**. Las actividades son cada una de las ventanas que la aplicación va presentando a lo largo de la ejecución del programa.

V-C2. Los agrupamientos: Un agrupamiento permite crear áreas que contengan diferentes elementos de interacción. Un agrupamiento contribuye a dar sensación de orden y a combinar en una única masa diversos controles.

Los agrupamientos se marcan con una línea de contorno y con una etiqueta descriptiva. Pueden establecerse agrupamientos dentro de otros agrupamientos para mejorar la presentación.

V-C3. Las pestañas: Las pestañas son útiles para clasificar las propiedades de un objeto de la tarea representado en una ventana. Un ejemplo típico es de configuración de la pantalla: hay tantas opciones que aparecen separadas. Se recomienda que la información en las diferentes pestañas debe ser independiente. Se debe evitar el uso de demasiadas pestañas, especialmente de varias filas de pestañas. No se debe utilizar cuando sea necesario realizar pasos en una secuencia, ya que no se ajusta a la metáfora.

V-C4. Scrolling: El scrolling se produce cuando el contenido que se presenta no cabe en el puerto de visión. Puede ser tanto vertical como horizontal, aunque este último no está recomendado. El scrolling vertical se ha convertido en un estándar en acceso a web y también al acceso en aplicaciones móviles.

El scrolling debe ser evidente, bien porque aparece una barra de scroll bien porque en el tipo de dispositivo el scrolling es una forma habitual de interactuar. La barra de scroll estándar puede ser mejorada visualmente para integrarla en

⁶<https://www.usability.gov/how-to-and-tools/methods/user-interface-elements.html>

la interfaz de la aplicación (por ejemplo, usando colores e iconos adaptados).

Ejercicio: En la web oficial de desarrollo de aplicaciones para Windows de Microsoft⁷ encontramos las recomendaciones de Windows para definir los layouts. Compare los contenidos de este documento con el contenido de esta sección.

V-D. Controles para la navegación

V-D1. Barras de menús: Se distingue entre los menús desplegables en los que al activar una opción aparecen nuevas opciones, menú en cascada en los que en cada nueva opción puede haber nuevas opciones, los menús enrollables que se despliegan y se repliegan bajo demanda y los menús emergentes que son menús flotantes que se activan en el sitio donde se encuentra el cursor.

A la hora de diseñar un menú debemos preguntarnos cómo nombrar a los menús y a las opciones de menú y si permiten los menús al usuario hacer lo que tienen que hacer, pero no más. Determinar cómo ordenar los menús y las opciones es crucial para proporcionar una interfaz usable.

V-D2. Barra de herramientas: Las barras de herramientas contienen los comandos utilizados frecuentemente complementados por iconos. Ofrecen una forma rápida y sencilla de acceder a estos comandos. Se pueden estructurar en diferentes barras con comandos relacionados (y hacerse visibles e invisibles), o como paletas flotantes.

Los iconos que forman parte de estas barras de herramientas deben ser sencillos y deben poder reconocerse fácilmente. Los iconos deben representar objetos concretos, fáciles de distinguir entre sí. Las “pistas” (tooltips), mensajes emergentes con texto explicativo que aparecen al pasar por los iconos, ayuda a entender su significado.

V-D3. Botones de comando: Los botones inician acciones. Se diferencian de los enlaces en que tienen un fondo personalizado y permiten incluir un título o un icono. Normalmente aparecen en una barra de navegación.

Deben nombrarse de forma que indiquen claramente la acción que ejecutan, utilizando verbos en los títulos. Los títulos deben ser cortos, y se suele poner en mayúsculas la primera letra de la palabra.

Se deben poner bordes y fondos cuando sea necesario. Deben ser de la misma forma y tamaño. Se pueden aceptar de diferente tamaño si las etiquetas también lo son y si están en la misma fila.

V-D4. Paginación: La paginación sirve para indicar la posición de la página dentro de una lista de ellas. Suelen ser puntos que hacen referencia cada uno a una página. El punto relleno se refiere a la página actual. Normalmente al pulsar se muestra le siguiente o el anterior. No se pueden utilizar cuando hay relaciones jerárquicas entre páginas. No deben utilizarse con demasiadas páginas: por ejemplo más de 10. Deben estar centrados en la parte inferior

Cuando hay una relación jerárquica entre páginas, deben usarse los breadcrumbs. Por ejemplo: Libros>Historia>Siglo XX

Ejercicio: La documentación oficial para desarrolladores de Android⁸ muestra un artículo sobre principios de navegación. Intente poner en relación lo presentado en dicho artículo con lo expuesto en esta sección.

VI. CONCLUSIONES

El diseño es una actividad creativa que conduce a la producción de una solución satisfactoria a los requisitos del sistema, en un tiempo y con unos recursos determinados.

El diseño de aplicaciones consiste en el establecimiento de una serie de módulos o componentes que resuelven tareas concretas y que interoperan siguiendo una arquitectura o patrón determinado. Las herramientas de autor ayudan a la implementación de aplicaciones multimedia descargando a los desarrolladores de muchas tareas de programación.

La interfaz de usuario es uno de los integrantes fundamentales de todo sistema informático que de ella depende la usabilidad de la aplicación. La interfaz está integrada por una serie de componentes básicos que resuelven las cuestiones relativas a la presentación de información, a la recogida de información, a la navegación y a la composición de los elementos.

Las guías de estilo y los principios de diseño son útiles para hacer que la interfaz de las aplicaciones multimedia satisfaga las expectativas de los usuarios desde un punto de vista estético y de usabilidad.

MÁS INFORMACIÓN

Libros genéricos de multimedia incluyen capítulos sobre interacción como el capítulo 6 de [4]. Los libros [5] o [6] que incluyen apartados en los que explican los elementos de interacción.

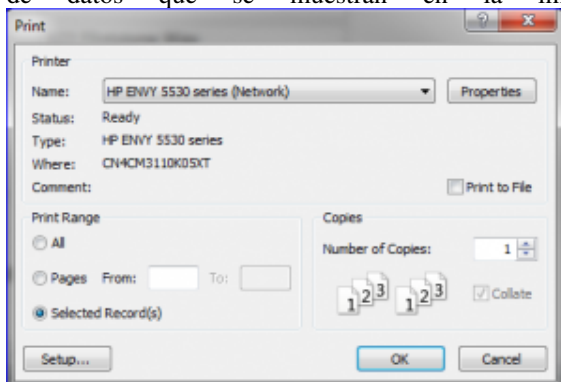
La principal referencia sobre diseño de interfaz centro en el usuario ha sido [7]. Para los controles de interacción se han empleado guías de estilo de diseño web y guías de diseño iOS, Android y Windows.

⁷ <https://learn.microsoft.com/en-us/windows/apps/design/layout/>

⁸ <https://developer.android.com/guide/navigation/navigation-principles>

CUESTIONES Y EJERCICIOS

1. Haga una definición propia de interfaz de usuario.
2. Haga una definición propia de interacción persona-computadora.
3. Haga una definición propia de diseño centrado en usuario.
4. Defina usabilidad.
5. Indique cual es el papel de la eficacia en la definición de usabilidad.
6. Indique cual es el papel de la eficiencia en la definición de usabilidad.
7. Indique cual es el papel de la satisfacción de usuario en la definición de usabilidad.
8. ¿Por qué es importante tener en cuenta el tipo de usuario a la hora de juzgar si un producto es o no usable?
9. ¿Por qué es importante tener en cuenta el contexto a la hora de juzgar si un producto es o no usable?
10. Indique por qué se considera que “no hacer pensar al usuario” es una clave de usabilidad.
11. Indique por qué se considera que “entender cómo el usuario accede a la información” es una clave de usabilidad.
12. Describa los tipos de navegación. ¿Cuál es el tipo de navegación que se emplea en WordPress?
13. Indique tres buenas prácticas para facilitar la navegación.
14. ¿Qué es el ciclo de acción del ser humano?
15. ¿Por qué es importante tener en cuenta el ciclo de acción del ser humano para hacer interfaces usables?
16. Indique qué es el modelo del diseñador del sistema, qué es el modelo de usuario del sistema y las implicaciones que tiene a la hora de diseñar interfaces usables.
17. ¿Qué es una metáfora en interfaces persona computadora?
18. Enumere los cinco estilos de interacción persona computadora.
19. ¿Qué tipos de estilo de interacción se emplea en 3D Studio?
20. Nombre cuatro áreas de diseño que se utilizan para clasificar las interfaces persona computadora.
21. Anote los distintos controles de introducción de datos que se muestran en la imagen.



conference on human-computer interaction. Springer, 2016, pp. 268–278.

- [3] H. L. Gage, *Applied Design for Printers. A handbook of the principles of arrangement, with brief comment on the periods of design which have most strongly influenced printing.* Committee on Education United Typothetae of America, 1920.
- [4] V. Costello, *Multimedia foundations: Core concepts for digital design.* CRC Press, 2016.
- [5] M. Elsom-Cook, *Principles of interactive multimedia.* McGraw-Hill, 2001.
- [6] N. Chapman and J. Chapman, *Digital multimedia.* Wiley Publishing, 2009.
- [7] S. Krug and R. Don't Make Me Think, “A common sense approach to web usability,” *New Riders*, 2014.

REFERENCIAS

- [1] I. Sommerville, “Software engineering 9th edition,” *ISBN-10*, vol. 137035152, p. 18, 2011.
- [2] N. Bevan, J. Carter, J. Earthy, T. Geis, and S. Harker, “New iso standards for usability, usability reports and usability measures,” in *International*

Posproducción, validación y despliegue

OBJETIVOS

El objetivo de este tema es presentar conceptos básicos relacionados con las pruebas y el despliegue de las aplicaciones multimedia. Después del tema, el lector habrá adquirido, como resultados de aprendizaje deberá:

- Realizar pruebas de usabilidad de las aplicaciones multimedia.
- Comprender detalles técnicos y legales relacionados con cuestiones de despliegue

ÍNDICE

I.	Evaluación de la usabilidad	2
I-A.	Pruebas con usuarios	2
I-B.	Tests alternativos	3
II.	Plataformas multimedia	3
II-A.	Dispositivos de visualización e interacción	3
II-B.	Almacenamiento y gestión de los recursos multimedia	4
II-C.	Acceso a los recursos multimedia en la plataforma	4
III.	Distribución multimedia y compatibilidad	5
III-A.	Compatibilidad de código de la aplicación	5
III-B.	Compatibilidad de medios	6
III-C.	Sistemas de gestión de color	6
IV.	Multimedia y derechos de autor	9
V.	Conclusiones	10
	Referencias	11

Antes de lanzar un producto multimedia, es importante hacer pruebas que garanticen una operatividad mínima, y también, que permitan validar si la aplicación cumple con los requisitos establecidos de antemano. En este tema se presentan algunas consideraciones importantes para realizar las pruebas de forma sistemática, teniendo en cuenta también consideraciones de usabilidad de la aplicación (sección I).

Una vez que el proyecto está listo para ser presentado, tenemos que decidir en qué plataforma queremos mostrarlo. La decisión depende de muchos factores y para tener un buen criterio tenemos que conocer algunos detalles técnicos que pueden ser determinantes.

No cualquier plataforma sirve para distribuir nuestro proyecto. Dependerá del tipo de proyecto que podamos pensar en unos canales u otros. Por ejemplo, si el proyecto tiene una gran interactividad, las plataformas de distribución de vídeo se quedan escasas. En este tema presentaremos primero los detalles técnicos comunes a cualquier plataforma que nos van a ayudar a conocer las limitaciones de estos sistemas de cara a la distribución de nuestros proyectos.

El criterio de decisión determinante a la hora de elegir la plataforma de distribución es el de intentar maximizar el impacto en la audiencia objetivo para la que el proyecto ha sido diseñado. No obstante, maximizar la audiencia no sólo tiene que ver con la plataforma empleada sino con la difusión del producto en otros canales. Hoy en día, las redes sociales son un medio ideal para intentar llegar a potenciales audiencias de manera selectiva. Un mensaje oportuno en una red social puede multiplicar el número de accesos a una página web o a un vídeo de Youtube, que garantizan máxima difusión.

Otro aspecto que debemos tener en cuenta en la distribución de nuestros proyectos es el de la compatibilidad. Puede ocurrir que un determinado proyecto desarrollado en un sistema dado sólo pueda ser visualizado dentro de dicho sistema. En la sección III veremos las estrategias a emplear para que esto no suceda.

Una cuestión que también se plantea a la hora de distribuir nuestro proyecto es saber hasta qué punto está protegido frente a plagios o copias por parte de terceros. De nuevo, también conviene saber hasta qué punto puedo emplear recursos multimedia realizados por terceros. En la sección IV, resolveremos estas cuestiones.

I. EVALUACIÓN DE LA USABILIDAD

La evaluación tiene por objeto establecer el grado de usabilidad del sistema construido. En la lista de requisitos de evaluación deberían aparecer los requisitos de usabilidad conocidos como las cinco Es [1], que son:

Eficacia. El grado de completitud y precisión con la que se alcanzan los objetivos.

Eficiencia. La velocidad (con precisión) con la que los usuarios realizan las tareas.

Engaging. El grado con el que el estilo del interfaz hace que el producto sea agradable y satisfactorio.

Error tolerant. Cómo el diseño previene errores o ayuda a recuperarse cuando se producen errores.

Easy to learn. Cómo el sistema ayuda a los usuarios a comenzar el juego y a profundizar en el entendimiento del mismo.

Las actividades de prueba deben estar orientadas a evaluar en grado de cumplimiento de los requisitos de usabilidad. En este apartado se describen algunos de los tipos de pruebas más frecuentes distinguiendo entre las pruebas en las que participan usuarios y las pruebas alternativas.

Parafraseando a Andy Pressman “Hay que buscar la crítica. Recibir críticas es una parte fundamental del aprendizaje. Asegúrate de que tu trabajo se revisa y se discute todo lo que se pueda. La crítica es un hecho en la vida profesional. Aunque la utilidad de las críticas es incuestionable, no siempre es fácil recibirlas. Aunque no sea lo más elegante, incorporar las críticas a la solución supone siempre una oportunidad para mejorar” [2]. Una evaluación tiene éxito cuando se detectan problemas. Las soluciones perfectas no existen. Siempre hay espacio para la mejora. Una evaluación que no ofrece propuestas de mejora es un fracaso.

I-A. Pruebas con usuarios

En un test de usabilidad se eligen los usuarios de manera que sean lo más parecido posible a los usuarios finales. Se considera que cinco usuarios de prueba suelen ser suficientes para obtener abundante información.

Las sesiones de prueba deben ser de corta duración para no sobrecargar al usuario y deben estar orientadas a que el usuario resuelva tareas concretas. El espacio en el que se realiza la prueba debe estar también pensado para poder obtener información del usuario sin ser intrusivo (sin molestarlo o incomodarle).

Según el prototipo que se utiliza para realizar la prueba tenemos las pruebas de concepto y las pruebas de lanzamiento. Las **pruebas de concepto** se realizan con prototipos de muy baja fidelidad (incluso bocetos). Una prueba de concepto se realiza para conocer cómo actuaría el usuario ante el sistema que se vaya a desarrollar. No se utiliza el sistema final sino prototipos iniciales aún por completar o incluso storyboards. Estas pruebas son muy útiles para conocer el modelo mental del usuario antes de implementar el código del programa. Las **pruebas de lanzamiento** deben realizarse con prototipos de alta fidelidad, versiones casi terminadas del sistema. Las pruebas de lanzamiento son pruebas realizadas con el producto terminado o prácticamente terminado.

Cuando se dispone de un prototipo de baja fidelidad, puede emplearse la **técnica de mago de Oz**. El término evoca al personaje literario: un gran mago envuelto en misterio que resultó ser un pufo. Se trata de simular el comportamiento de la aplicación antes de programarla.

A continuación describimos técnicas de recogida de información empleadas en estos tipos de prueba.

I-A1. Think aloud: Esta técnica consiste en que el usuario verbalice lo que se le pasa por la cabeza cuando está interactuando con el sistema. Esta prueba es muy informativa porque se consigue feedback inmediato y porque ayudan a que el usuario se concentre en la prueba. El inconveniente de estas pruebas es que no todo el mundo vale para realizarla porque hay usuarios que se sienten muy incómodos, el proceso puede enlentecerse y además, es cansado para el usuario. Generalmente lo mejor es no dar ninguna instrucción al usuario sobre cómo manejar el sistema.

I-A2. Observación de usuarios mientras prueban: También conocida como observación etnográfica, consiste en registrar, tomando notas o haciendo grabaciones, cómo se desarrolla el test de usabilidad.

Como clave para realizar estas pruebas con eficacia está la de no interferir o interferir lo más mínimo. También es muy importante dejar registro, ya sea con notas insitu o con notas analizando las grabaciones.

I-A3. Cuestionarios: Se trata ahora de preguntar la opinión del usuario con respecto a la usabilidad del mismo. Pueden usarse escalas numéricas (del 1 al 5 por ejemplo), pero mejor si se da un significado a cada cifra. En el cuestionario conviene intercalar preguntas que demuestren que el usuario ha utilizado correctamente la aplicación.

En los cuestionarios hay que dejar opciones para que el usuario incluya texto libre. Algunas preguntas son especialmente informativas, por ejemplo ¿Volvería a usarla? ¿La recomendaría?

I-A4. Focus group: Un focus group es una reunión con usuarios del programa en la que estos dan su opinión. Un grupo de usuarios (entre seis y diez) comentan su experiencia de interacción con el sistema. Deben prepararse temas de debate para animar la interacción. Las preguntas no deben ser tendenciosas o excesivamente dirigidas. La utilidad de estas reuniones depende mucho de la capacidad del moderador para extraer información de los usuarios y saber centrar la reunión en los aspectos de interés. Conviene grabar la reunión para dejar evidencias.

I-A5. Registros log: Los registros log son ficheros en los que quedan grabados todos los eventos de interacción que realiza el usuario junto con el instante de tiempo en el que lo realiza. Permiten reconstruir fielmente cómo ha sido el flujo de interacción del usuario con el programa. Son especialmente útiles para hacer estimaciones de la eficiencia con la que el usuario ha realizado las tareas asignadas.

Ejercicio: Imagine que le encargan desarrollar una página web para promocionar determinada marca de vino. La página incluye un pequeño videojuego y también una tienda virtual. ¿Qué técnicas emplearía

para realizar un test de usabilidad con usuarios reales?

I-B. Tests alternativos

Como alternativa a la realización de tests de usabilidad con usuarios está la realización de pruebas con sistemas de evaluación automática y las evaluaciones heurísticas.

Los **sistemas de evaluación automática** son programas informáticos cuyo input es la aplicación (podría ser un sitio web) y cuyo output es diagnóstico o una medida de la calidad del sistema atendiendo a diferentes criterios. Sobre todo se utiliza para resolver cuestiones de accesibilidad (facilidad de acceso de personas con discapacidad) y de seguridad (vulnerabilidad ante ataques). Pueden ser un futuro pero aún no existe una evaluación automática de la calidad del diseño.

Las **evaluaciones heurísticas** son una alternativa a las evaluaciones con usuarios, que son costosas en tiempo y recursos. Las evaluaciones heurísticas consisten en inspecciones o revisiones del interfaz cuya finalidad es examinar el interfaz del sistema comprobando si cumplen con las guías de diseño para encontrar defectos y realizar recomendaciones. Los inspectores son idealmente expertos en interacción persona computadora con un amplio conocimiento tanto del dominio como de las guías de diseño.

Los principales inconvenientes tienen que ver con el hecho de que las opiniones de los expertos no dejan de ser opiniones de alguien ajeno al sistema, que pueden estar sesgadas por gustos personales.

II. PLATAFORMAS MULTIMEDIA

Se llama plataforma de reproducción y/o distribución de contenido multimedia al conjunto de recursos hardware y software necesarios para facilitar que el usuario final pueda acceder a los contenidos de los proyectos multimedia implementados. En esta sección distinguimos entre los dispositivos en los que se visualizarán los proyectos multimedia, el software que permite el acceso a los recursos (incluidas las aplicaciones) y las consideraciones de almacenamiento y gestión de los recursos multimedia.

II-A. Dispositivos de visualización e interacción

El dispositivo final de visualización e interacción con un proyecto multimedia puede ser un ordenador de sobremesa o portátil, un dispositivo móvil como un teléfono o una tablet o un televisor o proyector de imagen. Cualquiera de estos dispositivos comparte una estructura común sin la cual no podrían formar parte de la plataforma.

La figura 1 muestra los elementos constitutivos del dispositivo de visualización e interacción de una plataforma multimedia. **Output** es típicamente el *display* o monitor o pantalla y los altavoces. **Input** es típicamente el ratón o la pantalla táctil o el mando a distancia. El usuario recibe (visualiza y escucha) los contenidos de la aplicación a través de los dispositivos de salida (output) e introduce comandos y otra información en la aplicación con los dispositivos de entrada (input). El bloque que llamamos *System unit* es un ordenador,

elemento fundamental para realizar la decodificación de los distintos recursos multimedia y gestionar la interacción con el usuario. **Almacenamiento interno** se refiere a la memoria interna del dispositivo, accesible desde el sistema, donde se almacena la información necesaria para poder ejecutar la aplicación. **Almacenamiento externo** representa los recursos multimedia que son empleados por la aplicación, pero que, por su tamaño o por cuestiones estratégicas no se descargan en la memoria interna del sistema.

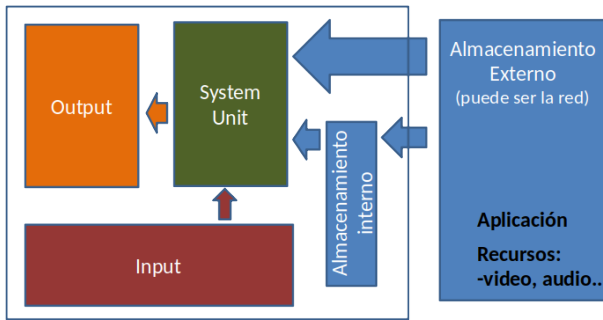
Los sistemas más empleados son los **ordenadores personales** (portátiles o de sobremesa), que en función del sistema operativo que utilizan distinguimos los ordenadores que emplean MacOS (fabricados por Apple), los ordenadores Windows (de múltiples fabricantes) y los ordenadores Linux (también de múltiples fabricantes). Los **dispositivos móviles** (smart phones o tablets), también se distinguen por el sistema operativo que emplean entre dispositivos iOS, dispositivos Android y dispositivos Windows Mobile. Por otro lado, los **smart TV** utilizan también sistemas operativos diversos, los más extendidos son *AndroidTV* y *WebOS* pero, en este caso existen multitud de sistemas operativos muy dependientes de dispositivo.

Como hemos dicho, los **dispositivos de salida**, para la componente gráfica, pueden ser monitores, pantallas y también proyectores; para la componente sonora serían los altavoces; puede haber otros dispositivos adicionales de típico mecánico como sistemas de vibración o brazos articulados. Con respecto a los **dispositivos de entrada**, el ratón y la pantalla táctil son los más extendidos para la interacción *háptica*, también el mando a distancia o el puntero portátil estarían en este ámbito. Más sofisticados son los guantes de realidad de virtual o las cámaras y sensores de posición. La cámara también puede ser empleada para introducir información.

Dentro de la **unidad del sistema** se distinguen principalmente la CPU, la RAM, la memoria caché, la placa base y los interfaces. La Unidad Central de Proceso (**CPU**) es el microprocesador en el que se ejecutan las instrucciones de las aplicaciones. Entre los parámetros importantes que caracterizan la están la velocidad de reloj, el tamaño de palabra y el multiprocesamiento. La velocidad de reloj es la tasa a la que puede ejecutar las instrucciones. El tamaño de palabra es la cantidad de información que puede procesar en cada ciclo de reloj. El multiprocesamiento, multicore, pipeline hace referencia a la posibilidad de ejecutar operaciones en paralelo. La **memoria principal** (RAM), es donde deben almacenarse los programas (tanto aplicaciones como sistema operativo) y los datos que se ejecutan en un momento dado en la unidad del sistema. La **memoria caché** es una memoria ultrarápida, de menor tamaño, intermedia entre la CPU y la RAM, que se utiliza para guardar los datos de uso más frecuente o los que son más probables de ser necesitados por los programas que se estén ejecutando. La **placa base** (motherboard) conecta el microprocesador, memoria y los buses de datos a los que acceden los periféricos. Los **interfaces** hardware son las conexiones para periféricos externos tales como el puerto serie o el USB o los HDMI.

El módulo de **almacenamiento secundario** es necesario porque la RAM es volátil, lo cual significa que si se corta el

Figura 1: Esquema de dispositivo hardware para la visualización e interacción con proyectos multimedia



suministro eléctrico, la información almacenada se pierde. El almacenamiento secundario permite guardar información que no queremos que se pierda, guardar copias de seguridad y distribuir aplicaciones y datos. En el caso de la multimedia, el almacenamiento secundario es imprescindible debido al gran volumen de información que debe almacenarse.

Las características principales de los medios de almacenamiento son la capacidad (medida en bytes), el tiempo de acceso (tiempo que se tarda en localizar la información) y la tasa de transferencia medida en bits por segundo (bps) o mega bytes por segundo (MB/s).

II-B. Almacenamiento y gestión de los recursos multimedia

Los soportes de almacenamiento secundario pueden ser de distinto tipo, entre ellos, CDs, DVDs, memorias USB, discos duros y discos de estado sólido. El almacenamiento en la nube entraría dentro de esta categoría. La nube es un servidor que gestiona el acceso a diferentes discos duros. Un **disco duro** (HDD) es una unidad en la que giran distintos discos de material magnético a una velocidad elevada (en torno a 7200rpm) para facilitar el acceso a los datos. Unos brazos móviles permiten acceder a las distintas superficies. Son económicos pero poco duraderos.

Los *compact discs* o **CD** fueron ideado inicialmente para almacenar audio digital, pero el CD-ROM permite, de hecho, almacenar cualquier dato y son supuestamente duraderos. Un CD-R es permite leer información y un CD-RW permite leer y escribir. La unidad lectora/grabadora de CD utiliza un láser que rastrea la superficie. La velocidad estándar de lectura o 1X es de 0.15MB/s. Esta velocidad se multiplica en las unidades 48X o 16X. Es importante tener en cuenta que si un disco se graba en una unidad que tiene una velocidad de grabación dada, puede no poder ser leído en otra unidad que trabaje a una velocidad diferente.

Los **DVD**, fueron ideados inicialmente para guardar películas de vídeo, sin embargo la "V" del acrónimo significa versátil, lo cual da una idea de que han sido ideados con un propósito general. La tecnología es la misma que la tecnología CD, pero el láser tiene una precisión mayor, permitiendo guardar más bits por pulgada. La tasa de transferencia estándar es de 1.321 MB/s. El Blue-Ray es la siguiente generación alcanzando tamaños muy superiores.

Cuadro I: Comparativa de tamaños y velocidades de distintos dispositivos de almacenamiento externo

Dispositivo	Tasa de transferencia	Capacidad
Memoria cache	30000MB/s	MB & GB
Disco duro HDD	15MB/s-160MB/s	GB & TB
Disco duro externo HDD	20MB/s-100MB/s	GB & TB
Memoria de estado sólido SSD	200MB/s-500MB/s	GB & TB
CD 1X	150KB/s	650 MB
CD 56X	8.6MB/s	650 MB
DVD-ROM 1X	1.350 KB/s	4.7 GB
DVD-ROM 6X	7.93 MB/s	4.7 GB
Blue-Ray 1X	4.5 MB/s	50 GB
Flash drives	60 MBs – 200 MBs	GB & TB
USB 1.0	1.5 MB/s	
USB 2.0	60 MB/s	
USB 3.0	625 MB/s	
Memory cards Class 2	15 MB/s	MB & GB
Memory cards Class 10	40 MB/s	MB & GB
Tarjeta de red	100Mbps	
ADSL	35Mbps	
Wifi	54Mbps	

Las **memorias de estado sólido** (SSD) son discos electrónicos, sin partes móviles y no volátiles. En esta categoría entran las memorias flash y las tarjetas de memoria. La desventaja frente a los discos duros es que tienen menor tiempo de vida y menor capacidad.

El **almacenamiento en la nube** se apoya en la red para transferir información y en servicios como Google Docs, iCloud o Microsoft OneDrive. Cuando se utiliza el almacenamiento en la nube hay que tener precauciones de seguridad, privacidad y riesgos de pérdidas de información.

La tabla I compara diferentes dispositivos de almacenamiento de información. El orden de velocidad es caché > SSD > HDD > Nube > memorias externas > DVD > CD. Es importante tener en cuenta las diferencias de velocidad que explican el éxito de la nube y el declive de los CDs y DVDs.

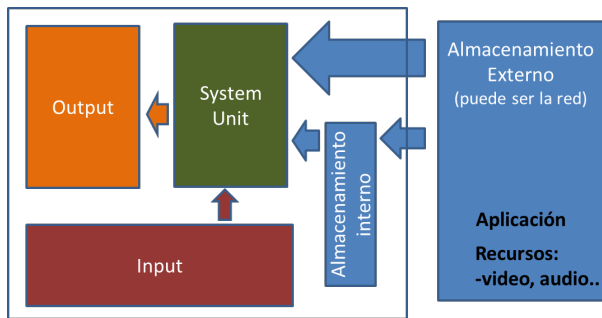
Ejercicio: Un USB 2.0 alcanza tasas de transferencia de 280Mbps mientras que HDMI 2.1 alcanza tasas de 48Gbps. Si tiene un fichero que tarda tres segundos en ser transferido por el puerto USB 2.0, ¿Cuánto menos podría tardar si pudiera usar el puerto HDMI?

Tardaría $48 \cdot 10^9 / 280 \cdot 10^6$ veces menos, unas 200 veces menos, que es una diferencia enorme.

II-C. Acceso a los recursos multimedia en la plataforma

La figura 2 describe el término plataforma multimedia de manera alternativa. Distinguimos ahora los módulos llamados recursos, software de sistema, API, aplicaciones y almacenamiento externo. Ahora el término **recursos físicos** se emplea para referirnos al dispositivo físico en el que se ejecuta las aplicaciones multimedia e incluye la CPU, memoria, gráficos e I/O; el **software de sistema** se incluye porque las aplicaciones no acceden directamente a los recursos del sistema, sino que lo hacen a través de una capa intermedia. Esta capa intermedia la constituyen las **API** acrónimo de *Application Program Interface*, que son las funciones que invocan las

Figura 2: Esquema de acceso a recursos de sistema y recursos multimedia en plataformas



aplicaciones multimedia para que los recursos multimedia (vídeo, imagen, audio, etc. . .) se presenten en el recurso físico. Las **aplicaciones** utilizan las API y acceden a los recursos empleando la red.

Ejercicio: Se quiere hacer una web que incluya vídeo y audio en ficheros externos. Además se quiere que dicha web incluya una imagen de fondo. Enumere los recursos multimedia que toman juego en este esquema de proyecto multimedia. Enumere los sistemas informáticos que participan cuando acceda a la web desde un ordenador portátil.

Intervienen los archivos de vídeo, audio e imágenes así como el archivo HTML. Los archivos deben estar almacenados en servidores. Normalmente el vídeo y el audio están en servidores tipo *streaming*. Además de los servidores, es necesario disponer de un ordenador o terminal móvil que ejecute un navegador que descargue y renderiza la información de todos estos archivos.

Cuando las aplicaciones acceden a los recursos de la red, no descargan el recurso completo sino que suelen emplear el streaming. **Streaming** es una técnica que permite procesar datos mientras se están recibiendo (o enviar mientras se van generando). Muy utilizados para ver vídeo o escuchar música, permiten por ejemplo ver una película sin necesidad de descargarla. La clave es que la descarga depende del punto del vídeo o del audio que se esté reproduciendo, descargando en un buffer apenas unos segundos adicionales. Conforme el cabezal de reproducción avanza se descargan datos adicionales, pero los datos ya visualizados no quedan almacenados.

Esta técnica que contrapone a la necesidad de recibir todos los datos para empezar a procesar (o de terminar de grabar para empezar a enviar). Streaming se emplea en vídeo digital o en transmisión de radio en tiempo real. Permiten ahorrar espacio en disco y memoria local porque sólo se carga una parte del recurso multimedia.

Para permitir que el flujo de información sea continuo durante la visualización del material multimedia, las plataformas utilizan protocolos de red alternativos a TCP y HTTP. Los protocolos favoritos del streaming son UDP y RTSP. La diferencia fundamental es que TCP y HTTP garantizan que llega toda la información a destino mientras que UDP y RTSP no lo hacen asumiendo que parte de la información

puede perderse. En transmisión multimedia se asume que parte de la información puede perderse. Perder alguna trama de vídeo no suele ser un problema cuando el espectador puede seguir la trama narrativa por el contexto. En los protocolos de streaming, el objetivo es garantizar un flujo continuo a una velocidad dada. Para garantizar la **calidad de servicio** las plataformas multimedia pueden servir diferentes versiones de los vídeos o audios en función de la velocidad de la red o del tamaño de los displays. Los buffers son necesarios para almacenar la parte del stream multimedia ya descargado y aún no visualizado.

A modo de ejemplo, la diferencia entre visualizar un vídeo en stream o con HTTP se puede comprobar al comparar la visualización de un vídeo en Youtube o de un vídeo incrustado en una página web cuando dejamos el vídeo en el propio disco duro en el que está la página. A no ser que el servidor web esté preparado para visualizar vídeo en stream, la segunda opción es mucho peor porque el navegador deberá descargar el vídeo entero y cargarlo en memoria antes de reproducirlo, lo cual entorpece el flujo de interacción peligrosamente.

III. DISTRIBUCIÓN MULTIMEDIA Y COMPATIBILIDAD

Cuando un proyecto multimedia se realiza en un tipo concreto de sistema lo deseable es que pueda visualizarse en otros tipos de sistema diferentes. Por ejemplo, si desarrollamos una aplicación en un entorno Apple, nos gustaría que dicha aplicación pudiera ser utilizada por usuarios de sistemas Windows. Del mismo modo, también nos gustaría que una página web pueda verse igual en el navegador Chrome o en el navegador Safari. Si contratamos Netflix, nos gustaría ver las películas en el móvil o un televisor . . .

Incluso dentro de un mismo tipo de sistemas puede haber problemas derivados de los **mínimos recursos** necesarios para ejecutar la aplicación. Las aplicaciones multimedia suelen ser bastante demandantes de CPU y de memoria, por lo que se suele sugerir un mínimo para poder ejecutar algunas aplicaciones. Aquí nos centramos en presentar los problemas de compatibilidad de código, los problemas de compatibilidad entre los medios y finalizamos hablando de un problema particular de compatibilidad que tiene que ver con el la compatibilidad de color.

III-A. Compatibilidad de código de la aplicación

Los problemas de compatibilidad entre sistemas se resuelven con máquinas virtuales y los problemas de compatibilidad de los medios se resuelven con los estándares. Una máquina virtual es un software que permite ejecutar programas. Una **máquina virtual** es una aplicación que puede ejecutar programas. Las máquinas virtuales se pueden instalar en diferentes plataformas y sistemas operativos. Las aplicaciones se ejecutan en la máquina virtual, facilitando la compatibilidad entre sistemas. La máquina virtual más extendida es JavaVM. Toda aplicación programada en JAVA podría ser ejecutada en JavaVM. Existen versiones de JavaVM para Windows, Mac y Unix. También se pueden añadir pluggings Java en los navegadores para poder ejecutar aplicaciones Java en

los Navegadores. Además, el desarrollo de aplicaciones para Android se basa en Java.

Los **navegadores** hacen el papel de una máquina virtual. Hay lenguajes específicos como JavaScript y PHP para ejecutar aplicaciones en el navegador. La diferencia entre javascript y PHP es que el código JavaScript se ejecuta en el navegador mientras que el código PHP se ejecuta en el servidor donde está la web. Los navegadores pueden añadir pluggings que permiten ejecutar código. Un plugging es una extensión que permite al navegador ejecutar código. Como hay versiones de navegadores para diferentes sistemas operativos, se garantiza la compatibilidad.

La tercera solución para garantizar la compatibilidad es emplear una **herramienta de autor** que permita exportar el proyecto para diferentes sistemas. Por ejemplo podríamos emplear una herramienta para desarrollar aplicaciones de manera independiente de plataforma. A la hora de distribuir el proyecto, éste se exportaría para diferentes plataformas.

Los estándares son normas que, si se siguen, garantizan la compatibilidad. Un estándar de iure es una norma fijada por un comité internacional: ISO, ITU, W3C. Un estándar de facto es un estándar que, sin ser diseñado por un organismo internacional se termina imponiendo (por ejemplo web). Hay estándares de iure en imagen (e.g. JPEG, SVG), video (e.g. MPEG), texto (e.g. UNICODE) No hay estándares en animación 3D ni en interacción.

III-B. Compatibilidad de medios

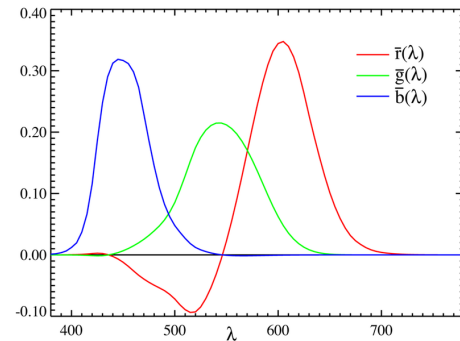
Por compatibilidad de medios nos referimos a la posibilidad de que un vídeo grabado por un dispositivo determinado de un fabricante dado pueda ser reproducido en un dispositivo de otro fabricante diferente. La solución a este tipo de problemas de compatibilidad está en los estándares.

Distinguimos entre los estándares de iure y los estándares de facto. Un **estándar de iure** es una norma fijada por una institución internacional como la *International Standard Organization* (ISO)¹, la *International Telecommunication Unit* (ITU)² o el *World Wide Web Consortium* (W3C)³. Estas instituciones nombran comités de expertos para definir normas que establecen los aspectos que deben cumplir los contenidos de los medios para ajustarse a un estándar. El que un estándar pase a ser de obligado cumplimiento debe pasar por una institución legislativa de ámbito nacional o supranacional, lo cual hace complicado impedir a un fabricante utilizar sus propios formatos.

Un **estándar de facto** es un estándar que, sin ser diseñado por un organismo internacional se termina imponiendo (por ejemplo la web). Cuando un fabricante tiene una posición dominante o desarrolla una solución técnica muy buena, el resto de fabricantes tenderán a imitarlo. Si el diseñador de la primera solución lo permite (podría protegerla), el resto terminan siguiendo su norma convirtiéndolo en un estándar.

En imagen vectorial, el estándar es SVG definido por la W3C (W3C). En imagen digital el estándar para intercambio

Figura 3: Curvas resultantes del experimento perceptual de suma de colores primarios (imágenes de dominio público vía Wikipedia).



de fotografía es JPG (ISO e ITU) y para para web es PNG (W3C). En audio tenemos el WAV (Microsoft e IBM), el MP3, el AAC (ISO) y el A3C (Dolby). En vídeo, el estándar es MPEG (ISO) y AVI (Microsoft). En texto, tenemos ASCII (ISO) y Unicode (consorcio de empresas). Para la definición de escenas 3D, los estándares de iure son VRML y X3D (ISO) pero otros ficheros como OBJ o STL se están convirtiendo en estándares de facto. No existen un estándar de iure para animaciones 3D, donde FBX (Autodesk) se está convirtiendo en un estándar de facto.

III-C. Sistemas de gestión de color

Los sistemas de gestión de control se introducen para hacer compatibles los colores entre los distintos dispositivos. ¿Por qué el color que veo en la pantalla no es el mismo que el color que imprimo? RGB y CMY son dependientes de dispositivo porque sus componentes indican lo que contribuye cada componente en la creación del color final. Pero la calidad de los sensores led de los monitores o de los cabezales de las impresoras puede ser diferente entre fabricantes. La consecuencia es que un mismo color RGB puede verse diferente en distintos monitores o al imprimir la imagen.

En la figura se muestran tres diferentes gamas de color diferentes sobre un mismo diagrama de cromaticidad CIE. En situaciones prácticas, cada dispositivo puede tener su propia gama. La consecuencia es que muchos colores del monitor pueden no ser reproducibles en la impresora. Además, colores captados por la cámara pueden no verse en el monitor.

Se hace necesario disponer de un modelo común que los distintos dispositivos pueden emplear como referencia. Una buena metáfora que representa bien el problema es la de una habitación con cuatro personas donde cada una de ellas habla en un idioma diferente. Necesitan un intérprete que hable los cuatro idiomas. También necesitan un lenguaje neutral que entiendan todos. Cada uno se dirige al intérprete que traduce lo que le dicen al lenguaje común. De esta manera cada uno puede seguir hablando en su idioma y entender al resto. El idioma común es CIELab y el intérprete es el **Color Management System** (sistema de gestión del color).

Cada dispositivo tiene un módulo (el **device profile**) para convertir los colores en un modelo común. Estos módulos

¹<https://www.iso.org>

²<http://www.itu.int>

³<http://www.w3.org>

pueden venir en el *driver* o controlador del dispositivo o venir incrustados dentro del fichero que contenga los gráficos en color.

Como ejemplo, veamos la operativa del sistema de gestión a la hora de imprimir. El programa que estemos utilizando (por ejemplo *Word* o *Illustrator*) emplea **Color Management Module** para convertir los valores RGB en valores CIELab. Después, estos valores CIELab son convertidos en CMYK utilizando el device profile declarado por la impresora en el driver. Compara la gama de los colores de entrada con los de salida aplicando y realiza una proyección aplicando las transformaciones oportunas. Estas transformaciones las realiza el color management module. Este color management module proyecta los colores de un device profile en los del otro. Debe tomar decisiones cuando las gamas de color no son iguales. Todo este sistema es dependiente del sistema operativo. Es por esto que es común que un mismo fichero impreso desde diferentes ordenadores en una misma impresora resulte tener colores diferentes dependiendo del sistema operativo.

Para entender el modelo de color CIELab, primero tenemos que explicar lo que son los colores primarios y los diagramas de cromaticidad. Mediante la combinación de fuentes de luz de colores diferentes se pueden formar nuevos colores. Si se combinan dos colores para producir el color blanco dichos colores se denominan **colores complementarios**. Seleccionando adecuadamente un conjunto de colores (**colores primarios**) podemos formar cualquier otro color. El conjunto de colores que se obtienen combinando dichos colores se denomina **gama de colores**.

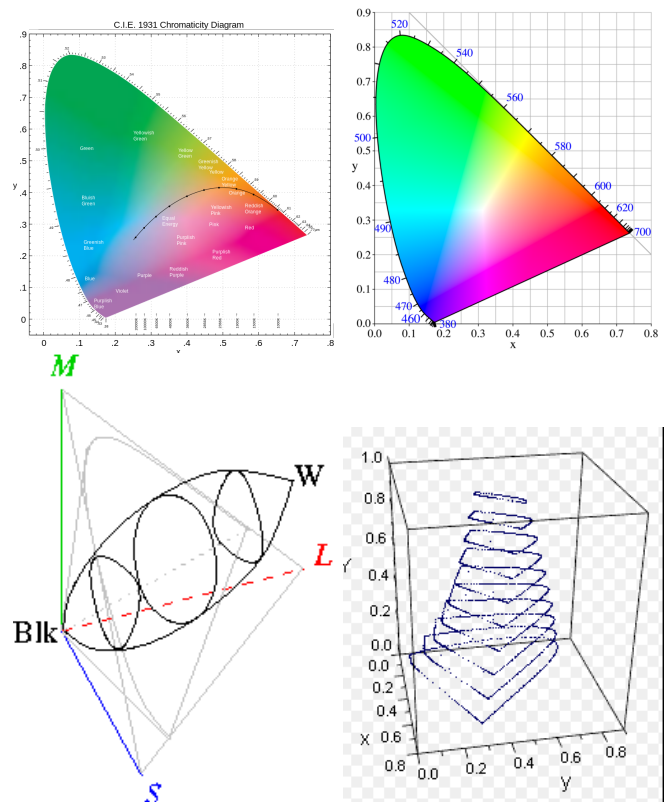
¿Existen tres colores primarios a partir de los cuales se puede generar cualquier color? En realidad, ningún conjunto finito de colores permite reproducir todos los colores posibles. En 1931, la Comisión Internacional sobre Iluminación conocida como CIE se plantea el objetivo de conseguir tres colores primarios estándar. Definen el experimento que se esquematiza en la figura 3. De un lado se muestra un color de longitud de onda dominante λ . De otro se muestra un color que se obtiene combinando los tres colores básicos. Cada uno de los tres colores puede modularse de manera que puede modificarse su intensidad. Cuando ambos colores (el color objetivo y el color generado) se perciben como un mismo color, se anota la cantidad de cada color básico empleado. El experimento se repite para múltiples evaluadores y para todos los valores de λ posibles. El resultado se muestra en figura 3.

En la figura 3 puede verse la cantidad de rojo, de verde y de azul que se necesita para generar un color de un λ dado. El hecho de que aparezcan valores negativos, pone en evidencia que no es posible generar cualquier color con estos tres colores básicos. Para salvar este inconveniente, CIE definió tres colores primarios que no son colores reales sino imaginarios. En la siguiente sección veremos cómo se define el modelo, de manera que todo color puede definirse a partir de los primarios.

Pero ¿cómo pueden resultar valores negativos? La respuesta está en la aplicación de las **leyes de Grassman** en los experimentos:

- Ley simétrica: Si el color A confronta con B, B confronta con A.

Figura 4: Diagrama CIE de cromaticidad. Sobre el contorno se sitúan los colores espectrales. El número se refiere al valor de la frecuencia dominante (la primera imagen es cortesía de P. Shou, la cuarta de A. Cik, el resto son de dominio público).



- Ley transitiva: Si el color A confronta con B y B confronta con C, entonces A confronta con C.
- Ley proporcional: Si el color A confronta con B el color kA confronta con kB .
- Ley aditiva: Si A confronta con B y C con D, si $A+C$ confronta con $B+D$ entonces $A+D$ confronta con $B+C$

Los valores de los colores primarios pueden ser negativos porque en realidad lo que se hace es añadir cantidades a los otros colores primarios para representar los efectos que producirían valores negativos.

El conjunto de colores primarios de este modelo se denomina X , Y y Z . Cada color C_λ se expresa como

$$C_\lambda = XX + YY + ZZ \quad (1)$$

donde X , Y y Z designan las cantidades de cada color que necesitamos para generar C_λ .

No debe sorprendernos ver el color definido como un vector. Tengamos en cuenta que un color en el modelo RGB es un punto en el interior del espacio tridimensional que define el cubo RGB.

Podemos normalizar los valores X , Y y Z de manera que

$$x = \frac{X}{X + Y + Z}, \quad y = \frac{Y}{X + Y + Z}, \quad z = \frac{Z}{X + Y + Z} \quad (2)$$

Figura 5: Usos del diagrama de cromaticidad a) colores complementarios b) gama de colores (cortesía de R. Ben) c) temperaturas del color (cortesía de S. Pig) d) zonas de color (de [?])

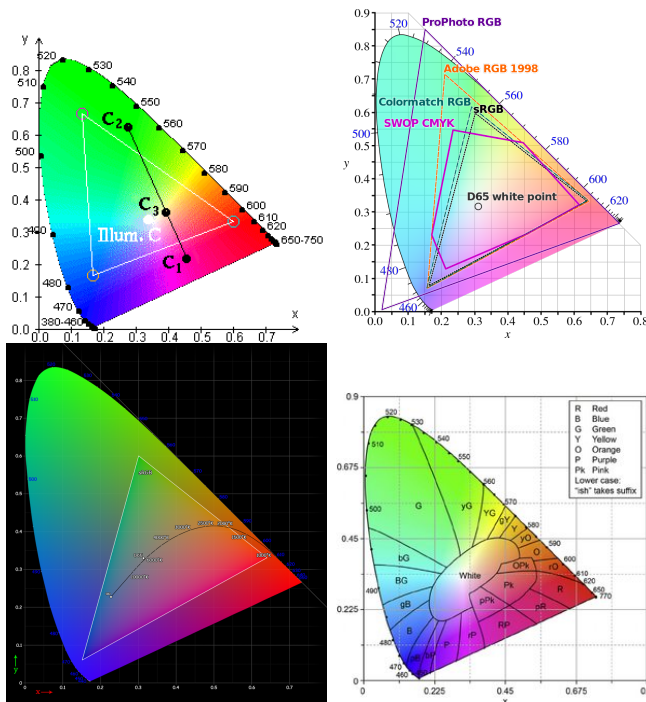
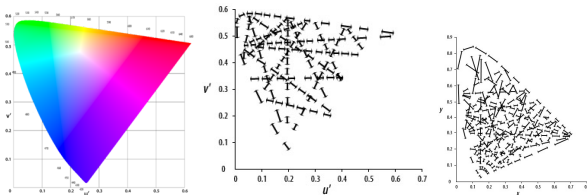


Figura 6: Diagrama de cromaticidad CIELUV. Los colores sobre los segmentos se perciben como si fueran el mismo color (de [3]).

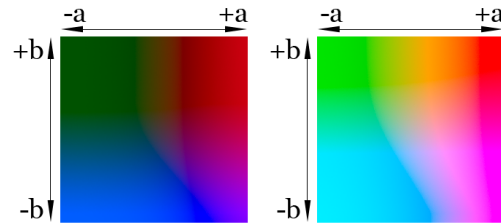


siendo $x+y+z = 1$. Por tanto, cualquier valor se puede representar con sólo los valores de x e y . Por estar normalizados, estos valores se denominan valores de **cromaticidad** ya que sólo afectan al matiz y de la pureza. Para definir un color de manera completa necesitamos un parámetro más. Se escoge el parámetro Y que se refiere a la **luminosidad o brillo** del color.

III-C1. Diagrama de cromaticidad CIE: El diagrama de cromaticidad representa los colores del espectro visible para una luminancia dada (figura 4). A lo largo de la curva aparecen los colores puros. La recta que va del violeta al rojo se denomina *línea púrpura* y es el límite del espectro visible. El punto C en el diagrama corresponde a la luz blanca. Los colores con la misma cromaticidad pero con distinta intensidad se representan en el mismo punto.

III-C2. Diagrama CIELab: En la figura 6.a se muestra una debilidad importante del diagrama de cromaticidad CIE. Los

Figura 7: Representación gráfica de los parámetros del modelo de cromaticidad CIELab con luminosidad al 75% y al 25% respectivamente (cortesía de J. Voss).



colores que caen en el mismo segmento se perciben como colores si fueran el mismo color. En el mismo segmento aparecen colores perceptualmente equivalentes.

El espacio LUV corrige este problema. Las figuras 6.b y .c muestran un diagrama modificado donde ahora las distancias de equivalencia perceptual son las mismas, independientemente de la zona del diagrama en la que nos encontremos. Para conseguirlo, los valores x e y se cambian por u' y v' utilizando las siguientes fórmulas:

$$u' = \frac{4X}{X + 15Y + 3Z} \quad (3)$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \quad (4)$$

También se cambia Y por L^* . Y emplea una escala que no representa los cambios perceptuales. En la escala de L^* , los cambios en L^* de igual valor se perciben igual, son perceptualmente equivalentes.

Esta nueva modificación del diagrama se apoya en la tesis de que el cerebro distingue principalmente entre opuestos, en este caso los opuestos serían:

- Luminoso vs. oscuro
- Rojo vs. verde
- Azul vs. amarillo

La elección de estos colores se apoya en el hecho de que no existe un azul amarillento o un rojo verdoso (ver figura 5).

La figura 7 muestra el interfaz del modelo CIELab. L^* es la luminosidad. En el eje $a - a'$ los valores positivos indican cantidades de rojo y los negativos cantidades de verde. En el eje $b - b'$, los valores positivos son para el amarillo y los negativos son para el azul.

El sistema CIELab es importante, no solo porque transforma las coordenadas del diagrama CIE en valores fácilmente interpretables, sino también porque se trata de un sistema independiente de dispositivo (a diferencia de RGB y CMYK). Es el sistema que utiliza Adobe Postscript y en el que se apoyan los dispositivos hardware para dar soporte a la independencia de dispositivos. De hecho, es el sistema que es estándar adoptado por el ICC (International Color Consortium) para hacer modelos de color independientes de dispositivo.

III-C3. Referencias de colores: Pantone Inc. es la empresa creadora del *Pantone Matching System*, un sistema de identificación, comparación y comunicación del color para las artes gráficas.

Figura 8: Paleta de colores de Pantone (foto cortesía de P. Nuncier).



El sistema se basa en una paleta o gama de colores conocida como las guías Pantone. Estas guías son unas plantillas con colores impresos. Las guías son un número de pequeñas tarjetas sobre las que se muestra cada color y un código asociado. El código permite acceder a un diccionario en el que indica cómo construir dicho color. Por ejemplo en pintura, es posible obtener colores por mezclas de tintas predeterminadas que proporciona el fabricante.

Conviene utilizarlas en diseño gráfico para tener una referencia única del color que se persigue. Si la imprenta tiene las guías Pantone de sus sistemas de impresión, pueden utilizar el color concreto en los selectores de color que te ofrezca tu programa de diseño gráfico.

Pantone no es la única compañía que ofrece este servicio pero sí que es la más extendida. Otra empresa que ofrece productos similares es **Munsell**.

Tanto Pantone como Munsell reconocen el estándar **sRGB** (W3C). Este estándar establece una correspondencia con el modelo CIEXYZ. sRGB es reconocido por los navegadores web y por la mayoría de fabricantes de dispositivos de captura de imagen.

IV. MULTIMEDIA Y DERECHOS DE AUTOR

Tanto para la aplicación multimedia desarrollada como para los assets empleados en la misma, es necesario tener en cuenta consideraciones relacionadas con los derechos de autor.

El **copyright** hace referencia a los derechos de un autor a la explotación de su obra durante un periodo de tiempo determinado. Las condiciones que pone la ley para poder registrar una obra son la originalidad, la persistencia (*fixity*) y un mínimo de creatividad. Las aplicaciones informáticas, cuando cumplan estas tres condiciones, y por tanto, pueden ser registradas. Para poder registrar una obra, la ley establece un protocolo que pasa por el acceso al **registro de la propiedad intelectual**. En la Unión Europea no existe la posibilidad de patentar software. En otros países como EEUU sí es posible patentar programas informáticos, aumentando la protección del productor sobre las aplicaciones.

Aunque no exista la posibilidad de patentar las aplicaciones multimedia, el registro de la propiedad de las mismas tiene sentido para que se reconozca la autoría y poder emprender acciones legales contra un uso no autorizado. Si empleamos material protegido por algún autor, la ley tiene en cuenta los siguientes supuestos a la hora de valorar la gravedad del uso sin consentimiento de un producto protegido. El primer aspecto a tener en cuenta es el **propósito del uso**; si hay lucro con un producto realizado por otro la ley obliga a compensar al autor. El segundo aspecto es la **naturaleza del material protegido**: si el material no ha sido publicado aún por el autor, el agravio es mucho mayor. También se tiene en cuenta la **cantidad de material** utilizado teniendo en cuenta que cuanto más material se pretenda emplear más necesario es el acuerdo. Por último, la ley también tiene en cuenta el **impacto en el mercado** que haya podido tener un uso ilegítimo del material.

El hecho de que un producto tenga copyright no significa que no se pueda utilizar. Por un lado, está la opción de negociar con el autor la posibilidad de emplear el producto, sea una imagen, un audio etc. . . . De hecho, cuando desarrollamos una aplicación multimedia para un cliente, el derecho de autor no se pierde por más que se venda el producto.

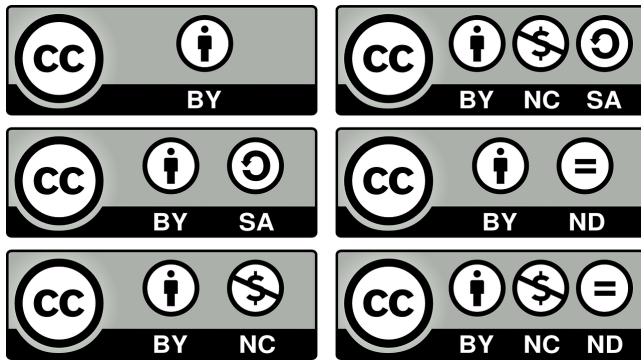
Al margen del copyright está la posibilidad de hacer un **uso limpio** del material multimedia. El uso limpio de un producto supone cumplir una serie de condiciones para poder utilizar algo que tenga copyright. En el caso no recomendable de utilizar un recurso multimedia realizado por un tercer autor sin permiso expreso, se recomienda en primer lugar solicitar dicho permiso, y cuando no sea posible o no se reciba respuesta, al menos se debe **citar la autoría** del material utilizado.

Como alternativa al copyright surge el **copyleft**, cuyo objetivo es garantizar que la propiedad intelectual no impida el uso de las obras. En el caso del software existe una comunidad muy grande y activa que aboga por el **software libre**, que implica no sólo la libre distribución de programas informáticos a coste cero, sino también el acceso al código fuente para poder ampliar o adaptar la funcionalidad de los programas. En el caso del software, las licencias más extendidas son la licencia GNU **GPL** (*General Public Licence*) en la que el autor permite utilizar sus obras a terceros siempre que éstos se comprometan a aplicar las mismas condiciones a los productos resultantes. Más permisiva es la licencia **MIT** o la licencia **Apache** que permite la copia, modificación y venta del producto original o derivados siempre que se reconozca la autoría.

Para dar un marco común a todo este tipo de licencia y extenderlo a otros dominios diferentes del software surge la iniciativa *Creative Commons*. **Creative Commons (CC)** es una organización sin ánimo de lucro que proporciona instrumentos jurídicos para permitir el uso compartido de obras. La figura 9 muestra los términos en los que articulan las licencias CC y las seis licencias posibles. Los términos son:

- BY Attribution:** permite que otros puedan copiar, mostrar, modificar o fusionar la obra si hacen referencia al autor.
- SA Share Alike:** permite que terceros puedan distribuir la obra en las mismas condiciones que ha puesto el autor.
- NC Non-Commercial:** permite que otros puedan distribuir, copiar, mostrar, modificar o fusionar la obra siempre que

Figura 9: Opciones Creative Commons (logotipos oficiales de CC).



no haya fines comerciales.

ND *No Derivative Works*: permite que terceros puedan copiar, distribuir, mostrar y ejecutar la obra, pero no pueden modificarla.

La figura 9 muestra el abanico de opciones que permite esta licencia, que van del **dominio público**, llamado CC0, que libera totalmente de los derechos de autor, al extremo opuesto del *copyright*. Entre medias tenemos las licencias tipo CC BY y CC BY-SA que serían licencias permisivas, que sólo obligan a reconocer la autoría y, en el caso de la licencia CC BY-SA, a compartir en las mismas condiciones que estableció el autor los resultados del trabajo. La licencia CC BY-ND no permite que la obra sea manipulada; la licencia CC BY-NC no permite que la obra o derivados sea vendida; la licencia CC BY-NC-SA no permite que sea vendida y obliga a poner las mismas condiciones que las fijadas por el autor en las obras derivadas; la licencia CC BY-NC-ND no permite ni modificar la obra ni comercializarla.

En el mundo de las tecnologías de la información el uso de licencias creative commons parece una tendencia frente al uso de patentes. Existen multitud de casos de éxito asociados al creative commons. Un producto creative commons se extiende con facilidad y con dicha extensión también crece el reconocimiento al autor. Los autores suelen ser compensados con futuros trabajos de consultoría o con la implicación en proyectos derivados.

V. CONCLUSIONES

La elección de la plataforma, junto con la difusión del proyecto, son claves para el éxito del mismo. Una plataforma de distribución multimedia necesita un sistema informático para ejecutar las aplicaciones, gestionar la interacción y acceder a los recursos multimedia.

El tamaño de la memoria y la velocidad de la CPU y de la red son parámetros muy importantes. Las técnicas de streaming evitan tener que descargar los recursos.

Los problemas de compatibilidad limitan la distribución. El uso de estándares soluciona buena parte de los problemas de compatibilidad. Hay que tener en cuenta el tipo de licencia a la hora de distribuir nuestros proyectos y utilizar los de otros.

INFORMACIÓN ADICIONAL

Las pruebas permiten validar si las aplicaciones multimedia satisfacen los requisitos establecidos en las fases de preproducción.

Para la evaluación de interfaz de usuario, la principal referencia seguida en estos apuntes es [4]. Una guía ofrecida por Apple para observación de usuario⁴

Los contenidos de este tema han sido extraídos en su mayoría de [5]. En otros textos de management puede obtenerse abundante información [6], [7], [8].

Para cuestiones de copyright el código civil recoge la mayoría de información necesaria. Como alternativa creative commons tiene su propio portal web con el que distribuye información creativecommons.org

Las *Technical Guides* de ADOBE en tienen un apartado muy completo sobre modelos de color con ilustraciones útiles. La compañía no ofrece estas páginas ya entre sus contenidos. Sin embargo, es fácil acceder a ellas desde enlaces diversos.

La página oficial de la CIE <http://www.cie.co.at/cie> ofrece información detallada de sus diversos comités.

Las páginas web de Pantone y de Munsell ofrecen abundantemente información técnica sobre el color.

En todos los libros de informática gráfica aparece un capítulo dedicado al color: capítulo 28 en [9], capítulo 17 en [10], capítulo 18 en [11] ...

En libros menos técnicos como [12], [13] o [14] también aparecen descritos los modelos de color (capítulo 11, 8 y 10 respectivamente).

⁴http://courses.cs.washington.edu/courses/cse440/08au/readings_files/gomoll.html

CUESTIONES Y PROBLEMAS

1. Si tiene que cargar una imagen de 20Mbytes en memoria de su ordenador, indique el tiempo que tarda en cargar si la tiene en un CD, si la tiene en un USB o si la tiene en el disco duro (emplee la tabla I).
2. Enumere tres tipos de sistemas operativos de escritorio y tres tipos de sistemas operativos de plataformas móviles.
3. Si tiene un vídeo 500MBytes, indique cuanto tardaría en cargarse en un navegador para poder verla si la velocidad de red es de 2Mbps sin utilizar streaming.
4. Indique las diferencias y parecidos de la tecnología DVD y CD.
5. ¿Cuándo se dice que una aplicación es multiplataforma?
6. ¿Qué es streaming?
7. Enumere cuatro aspectos a tener en cuenta cuando alguien utiliza la obra de un tercero sin su consentimiento.
8. ¿Qué es el uso limpio de una obra de un tercero?
9. Diferencia entre la licencia GNU GPL y las licencias MIT y Apache.
10. Indique a qué hacen referencia los términos BY, SA, NC y ND en la licencia Creative Commons.
11. ¿Cuándo se dice que una obra es de dominio público?
12. Una obra tipo CC BY-NC-ND ¿qué permite hacer con ella a terceros? ¿Qué otras alternativas hay dentro de creative commons?
13. Defina “Sistema de Gestión de Color” e indique por qué es importante.
14. En qué se diferencia el modelo CIELab y el CIEXYZ.
15. En qué se diferencia el modelo CIELUV y el CIEXYZ.
16. Cuáles son los parámetros del modelo CIELab? Y del CIELUV? Qué representa cada uno.
17. Qué es “gama de color”

REFERENCIAS

- [1] W. Quesenbery, “The five dimensions of usability,” *Content and complexity: Information design in technical communication*, pp. 81–102, 2003.
- [2] A. Pressman, *Architecture 101: a guide to the design studio*. John Wiley & Sons Incorporated, 1993.
- [3] S. Marschner and P. Shirley, *Fundamentals of computer graphics*. CRC Press, 2015.
- [4] D. Stone, C. Jarrett, M. Woodroffe, and S. Minocha, *User interface design and evaluation*. Elsevier, 2005.
- [5] M. Elsom-Cook, *Principles of interactive multimedia*. McGraw-Hill, 2001.
- [6] C. S. Dionisio, *A Project Manager’s Book of Forms: A Companion to the PMBOK Guide*. John Wiley & Sons, 2017.
- [7] M. D. Lock, *Project management in construction*. Gower Publishing, Ltd., 2012.
- [8] T. L. Young, *Successful project management*. Kogan Page Publishers, 2016.
- [9] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, E. Angel, and J. Hughes, *Computer graphics: principles and practice*. Addison-Wesley Professional, 2014.
- [10] D. Hearn, M. P. Baker *et al.*, *Computer graphics with OpenGL*. Upper Saddle River, NJ: Pearson Prentice Hall, 2004.
- [11] S. Marschner and P. Shirley, *Fundamentals of Computer Graphics*, 5th ed. CRC Press, Taylor and Francis Group, 2022.
- [12] V. Wisslar, *Illuminated pixels: the why, what, and how of digital lighting*. Cengage Learning, 2013.
- [13] J. Birn and J. Birn, “Digital lighting and rendering new riders,” 2006.
- [14] R. Yot, *Light for visual artists: understanding & using light in art & design*. Laurence King Publishing Limited, 2011.

La imagen digital como recurso multimedia

OBJETIVOS

El objetivo de este tema es presentar una serie de conceptos técnicos y definiciones básicas relacionadas con las imágenes digitales como recurso dentro de la multimedia. Después del tema, el lector, como resultados de aprendizaje deberá:

- Conocer las propiedades básicas de las imágenes digitales y vectoriales.
- Reconocer la importancia de la compresión de imagen y conocer los diferentes métodos de compresión.

ÍNDICE

I.	Características básicas de las imágenes digitales	2
I-A.	El canal alfa	2
I-B.	La relación de aspecto	3
I-C.	Imágenes de medio tono	3
II.	Imágenes vectoriales frente a imágenes digitales	3
III.	Variedad de formatos de fichero para almacenar imágenes	5
III-A.	Formatos de archivo de imágenes digitales	5
III-B.	Formatos de archivo de imágenes vectoriales	5
III-C.	Formatos de archivo metaficheros de imágenes	5
IV.	Compresión de imágenes digitales	6
IV-A.	Run Length Encoding	6
IV-B.	Look-up Tables	7
IV-C.	Compresión JPEG	7
IV-D.	Compresión con wavelets JPEG 2000 .	8
IV-E.	Compresión fractal	9
V.	Transmisión progresiva de imágenes	9
VI.	Conclusiones	10
	Referencias	11

El término imagen tiene diversas connotaciones que pasan desde una representación mental de algo o de alguien, a una proyección en un medio físico de ese algo o alguien. En tecnologías de la información, el término imagen también tiene varias acepciones que iremos matizando a lo largo del tema. Por ir concretando, podemos decir que el término imagen se refiere a imágenes fijas, sin animación o dinamismo explícito (en inglés *still images*). Esta primera matización permite distinguir entre imágenes fijas frente a animaciones y/o vídeos, donde se representa el movimiento. En una imagen fija, la realidad representada es estática, sin evolución a lo largo del tiempo.

Se trata de un término amplio porque las imágenes aparecen en múltiples formas y aplicaciones, por ejemplo:

- Imagen fotográfica, tanto las originales obtenidas con una cámara digital, documento escaneado, o como las manipuladas con un editor de imagen como puede ser Gimp o Photoshop. Por ejemplo las fotografías que tomamos con móvil y que después subimos a Instagram, retocadas o no, son imágenes digitales.
- Generadas con un editor gráfico como puede ser InkScape o Illustrator. Un cartel de las fiestas del barrio, que hayamos creado en un editor es también una imagen, que en este caso, puede ser una imagen vectorial o digital. Las imágenes que obtenemos al renderizar los modelos 3D que hacemos en CATIA o en 3D Studio también son imágenes digitales.
- Como resultado de una simulación de un algoritmo que manipule datos reales. Por ejemplo una gráfica de una función matemática obtenida con Matlab sería una imagen; los mapas con información de las predicciones meteorológicas también son imágenes que en este caso se construyen a partir de datos recogidos con múltiples sensores.

Las imágenes que se generan a partir de datos aplicando algún tipo de algoritmo se llaman **imágenes sintéticas**. Las imágenes sintéticas se contraponen con las fotografías obtenidas directamente de la realidad. También son imágenes sintéticas las creadas por un diseñador empleando un programa de edición o diseño gráfico, por ejemplo, los carteles publicitarios.

En este tema distinguimos una primera parte en la que se presentan las características básicas de las imágenes digitales y una segunda parte que se centra en compresión de imágenes digitales. La primera parte se aprovecha para distinguir entre imágenes digitales y vectoriales y la segunda parte presenta los algoritmos más empleados en compresión y codificación de imagen digital.

Figura 1: Características básicas de una imagen digital. En la parte superior se muestra el pixel como elemento constituyente de las imágenes digitales. En la parte inferior los canales R (rojo) G (verde) y B (azul) de la imagen (imagen base cortesía de A. Pazani CC BY vía Pexels.com)



I. CARACTERÍSTICAS BÁSICAS DE LAS IMÁGENES DIGITALES

Una imagen digital, independientemente de la manera en la que haya sido obtenida, se representa mediante pixels ordenados en una cuadrícula de dos dimensiones. Una imagen digital tiene un ancho, un alto y una profundidad (*pixel depth*). El **ancho** es el número de pixels en la dimensión horizontal y el **alto** el número de pixels en la dimensión vertical. El **pixel** es la unidad mínima de información que compone la imagen y representa un color. Para representar el color de cada pixel se emplea una secuencia de bits que es lo que llamamos **profundidad de pixel**. La profundidad es la misma para todos los pixels de la imagen, todos los pixels de una misma imagen se representan con el mismo número de bits.

El número de bits por pixel (la profundidad) depende del espacio de representación del color. Por ejemplo, una imagen en tonos de gris no requiere el mismo número de bits que una imagen en color. Los bits correspondientes con cada pixel se organizan en **canales**, de manera que el número total de bits por pixel es la suma de los pixels usados en cada canal. En imágenes en escala de grises se emplea un solo canal con 8 bits por pixel. En imágenes en color cada componente RGB (o canal) puede emplear 8 bits resultando 24 bits por pixel. A veces se utiliza un cuarto canal llamado canal alfa para representar la transparencia. Si se usa, tenemos 8 bits más, llegando a 32 bits por pixel.

El ancho y alto de una imagen y la profundidad determinan el **tamaño de la imagen** sin comprimir que es el número

Figura 2: Ejemplo de imagen con canal alpha (la imagen de fondo es cortesía de M. Ehlers CC BY vía pexels.com).



de bytes necesarios para representarla. Así por ejemplo, para representar una imagen de 640×480 pixels, si se usa color RGB con 8 bits por canal la imagen tendría un tamaño de 7.37 Mbits (921.6 KBytes). En escala de grises 2.45 Mbits (307.2 KBytes) — Recordamos aquí que 1 Byte = 8 bits.

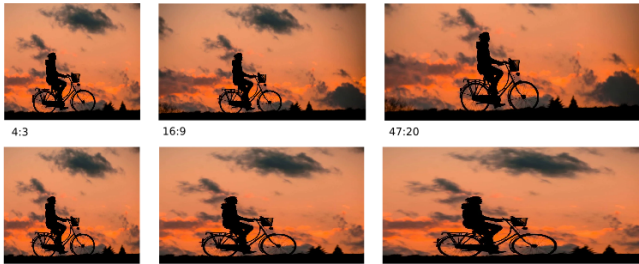
Ejercicio: Si tenemos que transmitir 20 imágenes de 2024×2024 que utiliza 32 bits de profundidad por una línea telefónica de 788 Kbps (Kilo bits por segundo). ¿Cuánto tardaría en hacerse la transmisión si las imágenes no están comprimidas?

Cada imagen son $T_i = 2024 \times 2024$ pixels, y en bits $T_i = 2024 \times 2024 \times 32$ bits. Como son 20 imágenes, el tamaño total en bits a transmitir sería $T = 20 \times T_i = 2.621.808.640$ bits. Como la velocidad de transmisión es de $V = 788000$ bits por segundo, tardaría $R = T/V = 13.945$ seg = 232 min, esto es, aproximadamente 4 minutos.

I-A. El canal alfa

El canal alfa se emplea para definir zonas de la imagen que pueden tener cierto grado de transparencia. Se utiliza para poder superponer imágenes dejando ver las imágenes que están por debajo a través de las transparencias de las imágenes situadas en capas superiores. En la imagen de la figura 2, podemos ver como el rostro de la modelo se superpone a la imagen del puente porque la imagen que contiene el rostro de la modelo tiene un canal alfa que indica que hay zonas de transparencia.

Figura 3: Diferentes imágenes que emplean diferente relación de aspecto. Arriba, la imagen original se deforma al cambiar la relación de aspecto. Abajo, la imagen no se deforma sino que se recorta empleando *letterboxing* y *pillarboxing* (imagen cortesía de Pixabay CC0 vía pexels.com).



Esta técnica se utiliza muchísimo en la industria de los videojuegos. Cuando Super Mario Bros se mueve sobre los escenarios, el avatar de Mario no es rectangular porque tiene asociado un canal alpha que indica que los pixels exteriores son transparentes. También se utiliza en interfaces hombre máquina cuando se presentan logos en botones; no son rectangulares porque contienen un canal alpha.

El canal alpha suele tener la misma profundidad de bits que el resto de canales. Así por ejemplo una imagen de 32 bits por pixel tendría 8 bits para cada componente RGB y 8 bits para el canal alpha. El canal alfa de 8 bits representa el grado de transparencia del pixel correspondiente a la hora de componer imágenes: Un valor cero, o lo que es lo mismo un color negro, indica que el pixel no va a ser empleado y que la zona correspondiente será transparente. Un valor elevado de 255 (para valores alpha de 8 bits) indica que el color del pixel va a ser presentado al completo. Valores intermedios indican que se represente con un cierto grado de transparencia.

Se suelen emplear 8 bits por canal pero hay otras posibilidades. En lo que llama *high color resolution* se emplean 16 bits por canal. Este modo se puede emplear en postproducción de video digital pero es poco frecuente. En algunas consolas de videojuegos se usan 16 bits en total: 5 bits por cada canal de color y un bit para el canal alpha. Cuando se utiliza 1 bit por canal, el pixel será opaco si su valor alpha vale 1 y transparente si el valor del pixel vale 0.

I-B. La relación de aspecto

La relación de aspecto es la relación entre el ancho y el alto de la imagen. Existen combinaciones que son estándar y están asociadas a determinados usos. Por ejemplo la relación de aspecto 3:2 está asociada a fotografía convencional; la relación 4:3 a los televisores antiguos; la relación 16:9 se emplea en los televisores modernos y las imágenes se denominan *high-definition*; la resolución 47:20 se emplean en cines y los formatos se llaman anamórficos. Los teléfonos móviles han traído muchas nuevas resoluciones de aspecto hasta antes no empleadas en dispositivos digitales.

Al cambiar la relación de aspecto, la imagen se puede deformar. Para reducir la deformación se emplea el *pixel*

aspect ratio (PAR) o *sample aspect ratio* (SAR) en español la relación de aspecto. Los dispositivos de captura suelen tener la misma densidad de pixels en una y otra dirección: el pixel es cuadrado PAR=1:1. Si cambias el aspect ratio, la imagen también cambia. Si la imagen no se deforma es porque hay procesamiento que suele consistir en un corte o eliminación de zonas. Como alternativa a eliminar parte de la imagen, se pueden incluir zonas negras en las partes laterales o en las partes inferior y superior de la pantalla. Cuando la zona que se completa con barras negras es la zona lateral hablamos de *pillarboxing* y cuando la zona que se completa es la zona superior e inferior hablamos de *letterboxing*. La figura 3 ilustra estos efectos.

I-C. Imágenes de medio tono

El concepto de pixel está vinculado estrechamente a dispositivos tipo pantalla (*display*) que están compuestos por un número fijo de puntos de color. Son dispositivos de este tipo los monitores, las pantallas de móvil, los sensores de las cámaras de vídeo y fotografía digital etc... En tecnología impresa las cosas son diferentes.

La versatilidad de un dispositivo de copia impresa, como puede ser una impresora o una imprenta condiciona el tipo de imagen que producen. En el caso de las impresoras por ejemplo, el tamaño del punto es mucho menor y la posibilidad de posicionarlo en un punto concreto de la hoja de papel puede ser muy precisa. En imprenta, por otro lado, el número de colores suele estar limitado por el número de tintas disponibles. Hacer colores diferentes a los disponibles en los tintes básicos no puede hacerse por simple mezcla como ocurre con la luz de los pixels RGB. De hecho al superponer tintas suelen perderse matices. Una de las soluciones adoptadas para conseguir imágenes de calidad son las imágenes de medio tono o imágenes *halftone*.

Las tecnologías de impresión suelen imprimir imágenes de medio tono (*halftone*). El número de colores se minimiza para reducir los costes de impresión. En las imágenes de medio tono, se crean los rangos de color o de gris con puntos de tamaño variable. La resolución de estas imágenes y dispositivos se mide en términos de frecuencia de puntos de medio tono (*dots per inch*) en lugar de en pixels. Los puntos controlan cuanta tinta se deposita en una posición específica del papel mientras se imprime. Variando el tamaño y densidad de los puntos se crea la ilusión de gris y de color continuo. Se suelen emplear cuatro canales de halftone: cian, magenta, amarillo y negro. Uno por cada tinta utilizada.

En la figura 4 se muestra un ejemplo de una imagen de medio tono. Como puede verse, al acercarnos a la imagen aparecen los puntos de los colores básicos que se superponen en mayor o menor grado para componer la imagen final. Las imágenes de medio tono son empleadas de forma masiva en impresión de carteles y embalajes.

II. IMÁGENES VECTORIALES FRENTE A IMÁGENES DIGITALES

Las imágenes digitales (también llamadas **imágenes de rastreo** o *raster*), tal y como las hemos definido en la sección

Figura 4: Ejemplo de imagen de medio tono.

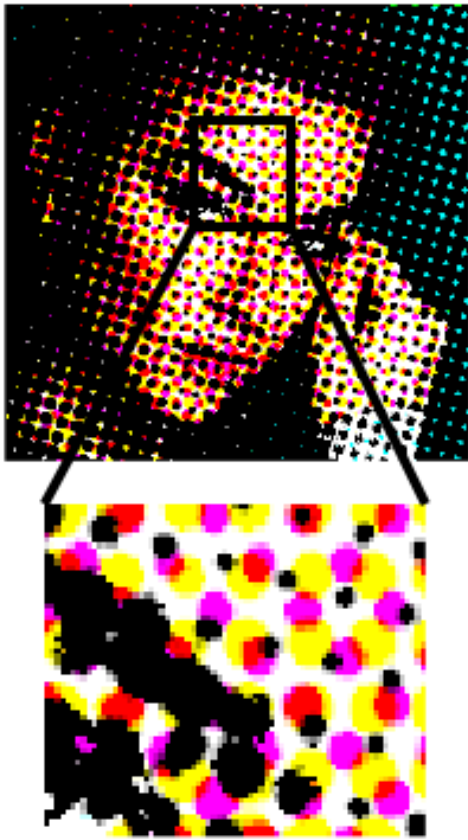
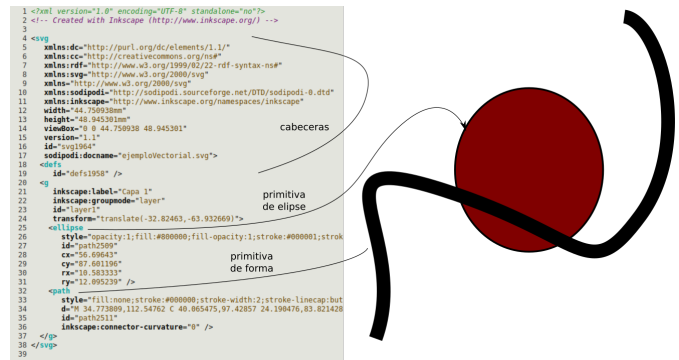


Figura 5: Ejemplo de imágenes vectoriales frente a imágenes digitales de tipo raster (imagen cortesía de Soco St CC BY vía svgrepo.com).



Figura 6: Ejemplo de imagen vectorial codificada en SVG



anterior, almacenan la información en pixels ordenados en una matriz de dos dimensiones. Las **imágenes vectoriales**, por contra, almacenan la información como objetos gráficos o primitivas geométricas en las que se especifica el tipo de objeto gráfico a representar, su posición y los parámetros del mismo como tamaño, color etc. . . .

Las diferencias entre una imagen raster y una imagen vectorial se observan claramente cuando se amplía la vista. Al hacer zoom sobre una imagen raster veremos los pixels. En una imagen vectorial no aparecen los pixels porque la imagen se redibuja cada vez que la ampliemos haciendo zoom. Una imagen vectorial como las de la figura 5 incluye la información necesaria para redibujar la escena con precisión, algo que no está disponible en las imágenes de tipo raster en las que sólo está disponible la información del color de cada pixel.

Se llama **digitalización** o renderizado o rasterizado al proceso que convierte una imagen vectorial en una imagen raster. Se llama **vectorizado** o trazado al proceso de conversión de una imagen digital en una imagen vectorial. Por ejemplo cuando guardamos un archivo realizado en Illustrator en un fichero JPEG estamos rasterizando la escena. Algunas herramientas como Animate CC de Adobe o Corel Draw incluyen herramientas de vectorización o de trazado de mapa de bits que localizan contornos en imágenes digitales.

Los editores gráficos tipo Illustrator o Inkscape trabajan con gráficos vectoriales mientras que los programas de tratamiento

de imagen como Photoshop o Gimp trabajan con imágenes digitales. Las ventajas de trabajar en un formato vectorial se ven claramente a la hora de manipular la imagen. Al tener acceso a los objetos gráficos y a sus propiedades, las escenas pueden ser manipuladas sin pérdida de información. Esto incluye realizar operaciones de transformación geométrica sobre los objetos gráficos (rotación, traslación, escala etc. . .) o cambiar sus atributos (color, grosor del trazo etc. . .). Una vez rasterizada una imagen digital, la posibilidad de retocar la escena sin pérdida de información se pierde.

SVG es un estándar para la especificación de gráficos vectoriales 2D. La figura 6 muestra un ejemplo en el que

se emplean la primitiva *path* para trazar cuatro líneas que unen tres puntos trazados con la primitiva *circle*; los puntos están etiquetados con tres letras dibujadas con la primitiva *text*; y una estrella que se dibuja con la primitiva *polygon*. Cada primitiva tiene unos atributos asociados relativos a la posición: *x*, *y*, *cx*, *cy*, *dx*, *dy*; al tamaño *r*, *stroke-width*, *font-size*; estilo *fill*, *font-family*, *stroke*; los polígonos se definen especificando la secuencia de puntos (pares *x* y) que los componen y los paths con el punto inicial y desplazamientos relativos. Los ficheros SVG se declaran en texto legible, otros ficheros de tipo vectorial pueden venir codificados en binario, de manera que no podemos acceder a su contenido (por ejemplo los ficheros pdf).

El equivalente en tres dimensiones a SVG se llama VRML o X3D y permite declarar escenas 3D escribiendo directamente en ficheros de texto. Para poder visualizar la escena representada hay que renderizar el fichero.

Los **metaficheros gráficos** o *metafiles* combinan información de rastreo y vectorial. Significa que dentro de una composición de objetos gráficos puede incluirse una imagen digital. La mayoría de formatos de fichero vectorial incluyen esta opción.

III. VARIEDAD DE FORMATOS DE FICHERO PARA ALMACENAR IMÁGENES

Una característica del mundo del almacenamiento de ficheros relacionados con las imágenes y con los datos gráficos en general es la gran diversidad de formatos de archivo diferentes. No es sólo que cada programa tenga su propio tipo de archivo, sino que también existen diferentes modalidades de archivos estándar como por ejemplo png y JPEG, ambos estándar.

¿Por qué hay tantos formatos diferentes de archivos asociados al almacenamiento de las imágenes digitales? La primera razón está en el modo de producción de dichas imágenes: una imagen capturada por una cámara de fotos digital tiene unas características derivadas de la forma de capturarla; a la vez una imagen generada para servir de cartel de fiestas de una ciudad, por ejemplo, tendrá otras diferentes, derivadas del uso del programa de edición con el que haya sido creada.

La segunda razón tiene que ver con los intereses comerciales de los programas de creación y edición de imágenes. Un programa informático quiere crear dependencias con sus clientes para que renueven licencias. Para ello, utiliza un formato de fichero propietario para que, cuando sea necesario modificar la imagen, se recurra a dicho programa. Esta filosofía no sólo hace que proliferen los tipos de fichero, sino también los formatos estándar para facilitar el intercambio de las imágenes.

La tercera razón son los algoritmos de compresión, que se especializan en determinados tipos de imagen. No es lo mismo comprimir una imagen que contenga pocos colores que una fotografía de naturaleza. Al principio el tamaño de las imágenes no era grande y era posible utilizar formatos binarios no comprimidos. En Windows por ejemplo se utilizaba el formato bitmap o .bmp sin comprimir para guardar pequeños iconos. Actualmente los archivos gráficos se comprimen siempre, salvo con escasas excepciones como pueden ser el formato *raw image* empleado en fotografía digital profesional para mantener la máxima calidad en las imágenes.

A continuación listamos algunos ejemplos de formatos de ficheros gráficos comentando sus características. Distinguimos entre formatos de fichero asociados a imágenes digitales, a imágenes vectoriales y a metaficheros gráficos. En la siguiente sección presentamos algunos de los algoritmos de compresión más utilizados.

III-A. Formatos de archivo de imágenes digitales

.bmp (Windows bitmap) Es el formato propiedad de Windows muy empleado en las primeras versiones del sistema operativo. Puede utilizar desde 1 a 24 bits por pixel. Normalmente la información no está comprimida, pero puede usar el método de compresión sin pérdidas *run length encoding* RLE.

.gif (Graphics Interchange Format) Su uso en la web era masivo hasta la aparición del formato png. Permite hasta 256 colores indexados y también animaciones simples. Utiliza una paleta de color de colores de hasta 256 entradas. Cada color en la paleta se representa con 24 bits RGB. Utiliza el algoritmo de compresión LZW que es propiedad de la multinacional CompuServe. No soporta el canal alpha.

.jpeg (Joint Photographic Experts Group) Diseñado expresamente para guardar fotografía, su uso en la Web es masivo. Combina métodos de compresión con pérdidas y sin pérdidas. El nivel de compresión puede ser definido por el usuario de manera que a mayor compresión peor calidad. No soporta el canal alpha.

.psd (Adobe Photoshop) Las imágenes pueden guardarse sin comprimir y con diferentes capas (layers). Permite combinar y utilizar diferentes modelos de color y diferentes profundidades de pixel.

.tiff (Tagged Image File Format) Se trata de un formato muy popular en el mundo de la ilustración y el diseño gráfico. La ventaja frente a otros formatos es que puede guardar la información sin comprimir o utilizando compresión sin pérdidas.

III-B. Formatos de archivo de imágenes vectoriales

.swf (Sockwave Flash format) Este formato es propiedad de Adobe (antes Macromedia). Contiene información vectorial de gráficos animados (también puede insertar imágenes). Su diseño está optimizado para presentar imágenes y animaciones en la web. Casi todos los navegadores anunciaron que dejarían de soportarlo a partir de 2020.

.dxf (AutoCAD ASCII Drawing Interchange Format) El formato almacena gráficos 2D y 3D en texto ASCII.

.svg El formato estándar para definir gráficos 2D vectoriales escalables.

III-C. Formatos de archivo metaficheros de imágenes

.png (Portable Network Graphics) Formato estándar (aprobado por el W3C) definido para superar las restricciones de patentes de GIF y CompuServe. Permite color de entre 1 a 48 bits por pixel. Soporta el canal alpha para las transparencias. Permite incluir datos vectoriales que se superponen a la imagen digital.

- .ps o .eps (Postscript o Postscript encapsulado)** En un meta-fichero que no comprime la información. De hecho, es un fichero de texto. Permite incluir texto, gráficos vectoriales e imágenes en un mismo documentos.
- .pdf (Portable Document Format)** Permite guardar texto, fuentes e imágenes.

IV. COMPRESIÓN DE IMÁGENES DIGITALES

Las imágenes digitales necesitan ser comprimidas porque el volumen de información que contienen es muy grande. La tabla I muestra cómo para resoluciones de imagen pequeñas, el tamaño no es un problema, pero cuando la calidad de las imágenes sube, el tamaño de las mismas crece tanto, que transmitir las por la red o guardarlas en un disco llega a ser un problema. Así por ejemplo, enviar una fotografía por Whatsapp podría tardar varios minutos si no se comprime; guardar más de un centenar de fotografías en el móvil sería prohibitivo.

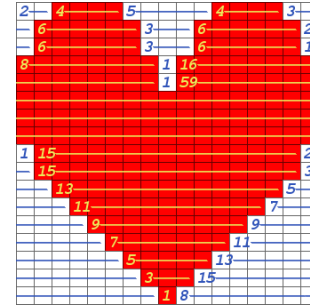
Un algoritmo de compresión utiliza la información contenida en la imagen digital para hacer una nueva versión de la misma cuyo tamaño es sensiblemente menor. La eficacia de la compresión de imágenes se basa en dos aspectos fundamentales: la reducción de la irrelevancia y la reducción de la redundancia.

La **reducción de la irrelevancia** hace referencia a su vez a dos aspectos fundamentales la irrelevancia visual y la irrelevancia específica de la aplicación en la que se va a utilizar la imagen. Con respecto a la **irrelevancia visual** explota el hecho de que la precisión de las imágenes capturada con determinados dispositivos como puede ser una cámara digital puede ser excesiva ya que el ojo no siempre es sensible a toda la gama de colores captada por estos dispositivos. Por otro lado, la **irrelevancia específica de aplicación** tiene en cuenta que determinadas aplicaciones no requieren toda la precisión que puede llegarse a alcanzar. Por ejemplo, en imágenes médicas, algunas zonas completas pueden no interesar por lo que pueden eliminarse o reducir su calidad.

Con respecto a la **reducción de la redundancia**, se aplica en dos aspectos fundamentales, en la redundancia espacial y en la espectral. Con respecto a la **redundancia espacial** ésta hace referencia a la existencia de zonas contiguas con valores iguales de píxeles; en lugar de codificar los colores de cada píxel de las zonas redundantes, sólo se codifica el color de una zona menor y se referencia en el resto de píxeles redundantes. Por otro lado, la **redundancia espectral** cambia la representación de la imagen para encontrar patrones repetitivos de la imagen, por ejemplo una imagen que represente unas ondas en el agua, puede ser fácil de comprimir si se conoce la frecuencia de oscilación de dichas ondas.

En compresión de imágenes se combinan métodos con pérdidas (*lossy*) y métodos sin pérdidas (*lossless*). En los **métodos lossy** hay pérdidas de calidad que pueden ser imperceptibles. Se aprovechan de la irrelevancia de algunos aspectos de la imagen que se intentan eliminar. Por contra, en los **métodos lossless** no hay pérdidas de calidad. Los métodos sin pérdidas se aprovechan principalmente de la redundancia de las zonas de la imagen.

Figura 7: Ejemplo para ilustrar la codificación *run length*.



Al proceso que convierte una imagen en su versión comprimida se le llama **compresión** y al proceso que recupera la imagen a partir de la imagen comprimida se le llama **descompresión**. En compresión con pérdidas la imagen descomprimida es diferente a la original, generalmente se pierde calidad en el proceso. Los algoritmos de compresión y descompresión de imágenes deben estar incorporados en los programas de edición gráfica, en los visores de imágenes y, a veces, en el propio hardware de los dispositivos como es el caso del algoritmo de compresión y descompresión JPEG en las cámaras digitales.

Se llama **tasa de compresión** a la relación entre el número de bits del fichero comprimido y del fichero original. Así por ejemplo decimos que un archivo está comprimido con una tasa de 1:100 si por cada bit del fichero comprimido había 100 bits en el fichero original.

A continuación presentamos cinco métodos de compresión de imágenes, uno sin pérdidas como es el método *run length encoding* y el resto con pérdidas, como son el método de *lookup table* empleado por GIF y PNG; el método empleado por JPEG que se basa en cuantización, la transformada discreta del coseno y entropías; el método basado en Wavelets de JPEG2000 y los métodos de codificación fractal.

IV-A. Run Length Encoding

El algoritmo run length encoding es un buen ejemplo que sirve para explicar la compresión sin pérdidas. Explica cómo es posible reducir el tamaño de un fichero con datos sin perder información. Este algoritmo aprovecha la existencia de secuencias de símbolos idénticos en el fichero, en este caso la existencia de secuencias de colores idénticos que ocupan posiciones consecutivas en la imagen. Cada secuencia se codifica con un par de elementos: un número, que indica cuantas veces se repite el símbolo y el propio símbolo.

El siguiente ejemplo ilustra cómo funcionaría el método para comprimir secuencias de caracteres. El fragmento de archivo BBBBEEEEEEEECCCCDAAAAA pasaría a ser 4B8E4C1D5A. Como se ve, las cuatro primeras B se codifican como 4B, pasando de 4 símbolos a 2 símbolos. En casos en los que hay redundancia elevada, como puede ser en determinadas imágenes, las tasas de compresión pueden ser enormes.

Considérese la imagen de la figura 7. Si el color blanco se codifica en hexadecimal #FFFFFF y el rojo como

Cuadro I: Características de diferentes tipos de imágenes. El tamaño sin comprimir se obtiene multiplicando el ancho por el alto por el número de bits por pixels y dividiendo por 1024 y por 8 para convertirlo en kilo bytes (KBs). La velocidad de transmisión se calcula dividiendo el tamaño en bits entre la velocidad en bits por segundo (bps) y entre 10^6 para convertirlo en megas. El espacio de cada imagen sin comprimir en GB se calcula dividiendo en tamaño en KB entre 1024 para convertirlo en MB (megabytes) y entre 101024 para convertirlo en GB (gigabytes).

Tipo de imagen	Imagen en escala de grises	Imagen en color	Fotograma HDTV	Imagen Médica	Fotografía alta resolución
Ancho×Alto	512×512	512×512	1280×720	2048×1680	5184×3456
Bits por pixel	8 bpp	24 bpp	12 bpp	12 bpp	24 bpp
Tamaño sin comprimir	256 KB	768 KB	1024 KB	4966 KB	52488 KB
Tiempo de transmisión de 1 imagen (2Mbps)	1 s	3 s	4 s	20 s	215 s
Tiempo de transmisión de 1 imagen (10Mbps)	0,2 s	0,6 s	4,4 s	20,3 s	43,0 s
Espacio de 500 imágenes	0,1 GB	0,4 GB	0,5 GB	2,3 GB	25,0 GB

#FF0000, la imagen de 17×17 necesitaría 1734 símbolos. Empleando un esquema run length, las primeras filas serían 2#FFFFFF4#FF00005#FFFFFF4#FF00003#FFFFFF... pasando a necesitar mucho menos símbolos.

Es uno de los primeros esquemas de codificación de imágenes que se emplearon. Lo utilizaba BMP en Windows y PICT en Macintosh. En el formato TIFF aparece para permitir introducir esquemas de codificación sin pérdidas.

Ejercicio: Dada la imagen de la de la figura 7, indique la relación de compresión si se utiliza compresión run length.

Empleando run length, tal y como es la imagen, necesitamos representar la secuencia 2B 4R 5B 4R 3B 6R 3B 6R 2B 6R 3B 6R 1B 8R 1B 16R 1B 19R 1B 15R 2B 15R 2B 15R 3B 13R 5B 11R 7B 9R 9B 7R 11B 5R 13B 3R 15B 1R 8B, donde B es #FFFFFF y R es #FF0000. El número que precede al color también debo representarlo. Como la secuencia máxima podría ser de 17 veces 17 símbolos (una imagen de un único color), son necesarios 9 bits por número porque $2^9 \geq 289 = 17 \times 17$. Como en la imagen tenemos 36 bloques de colores uniformes (contar los números en la secuencia entera), y cada bloque contiene un número de 9 bits y un color de 24 bits. Por tanto necesito $36 \times (9 + 24) = 1188$ bits.

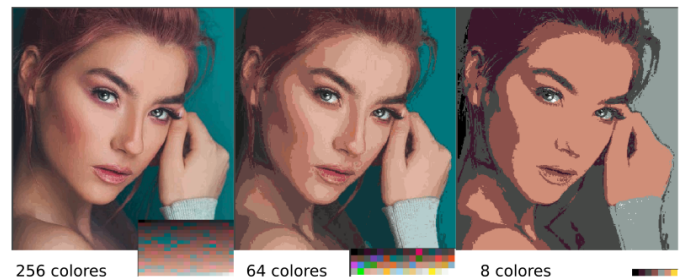
La razón de compresión sería $1188 : 6936 \rightarrow 1 : 6936/1188$ aproximadamente de 1:6

IV-B. Look-up Tables

El método del *look-up table* o de la **paleta de colores** consiste en generar una tabla o diccionario en el que están todos los colores de la imagen (la paleta propiamente dicha). Por otro lado se representan los pixels, pero estos pixels, en lugar de codificar colores, representan índices al diccionario.

La figura 8 muestra una imagen digital y su correspondiente paleta de colores. La paleta en este caso contiene 256 colores que suele ser el máximo número de colores empleados con este método. Para guardar la paleta de colores se necesita en este caso 256×3 Bytes ya que suponemos que cada color se representa con en modelo RGB 24 bits=3 Bytes. Además, para guardar la imagen, suponiendo que su tamaño es de 400×200 , necesitamos $400 \times 200 \times 1$ Bytes porque los índices de la paleta necesitan 8 bits para codificar las 256 entradas de la paleta: $256 \leq 2^8$.

Figura 8: Ejemplo para ilustrar el efecto del tamaño de la paleta de colores en la calidad de imagen.



Este método es recomendable sólo para imágenes con un número reducido de colores, por ejemplo para logotipos, pequeños diseños gráficos sin degradados. Cuando el número de colores es mayor, por ejemplo para fotografías, o bien se pierde demasiada información o bien la tasa de compresión no compensa.

Los archivos GIF utilizan este método. Sobre los índices se aplica un algoritmo llamado LZW que está protegido por patente. Este hecho, y la necesidad de añadir imágenes vectoriales para crear metaarchivos de imagen y la consideración de un canal alpha, es lo que propició la aparición de los ficheros PNG, que también utilizan paletas de color.

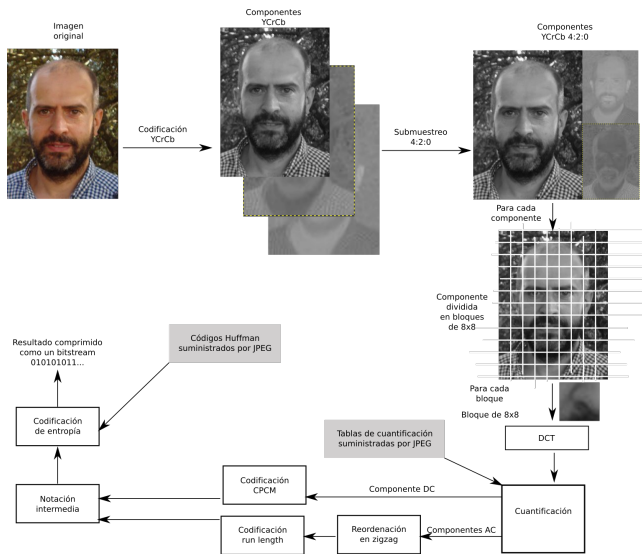
Ejercicio: Dada la imagen de la de la figura 7, indique la relación de compresión si se utiliza una paleta de colores optimizada.

Como sólo hay dos colores solo hace falta utilizar 1 bit por color en la paleta porque $2 \leq 2^1$, de manera que la paleta ocuparía 2×24 bits y la imagen $17 \times 17 \times 1$ bits. Pasamos pues de los $17 \times 17 \times 24$ bits originales a $17 \times 17 \times 1 + 2 \times 24$ bits comprimidos, resultado una tasa de compresión de 1:21. Nótese que empleando una paleta de colores, a la hora de calcular el tamaño de la imagen tenemos en cuenta tanto el tamaño de la imagen ($17 \times 17 \times 1$), como el de la paleta (2×24).

IV-C. Compresión JPEG

La compresión JPEG es un estándar internacional pensado para comprimir imágenes fotográficas con calidad variable:

Figura 9: Secuencia de procesos para la codificación JPEG.



de hecho JPEG es el acrónimo de *Joint Photographic Experts Group*. Calidad variable significa que la misma imagen puede ser comprimida con distintas razones de compresión resultando imágenes de diferente calidad, a mayor tasa de compresión, peor calidad. La compresión emplea diversas técnicas conjuntas que se aplican de forma secuencial como se muestra en la figura 9.

Lo primero que se hace es cambiar el formato de codificación de los píxeles de color, de manera que las imágenes de 24 bits por color (8 por cada canal RGB) se convierten en imágenes con **codificación YCrCb** con subsampling 4:2:0, lo que quiere decir, que se guarda por un lado la información de iluminación de la imagen en versión en escala de grises (canal Y) y por otro lado información de cromaticidad CrCb con una cuarta parte de los bits originales al aplicar el submuestreo: Se toma una muestra de color cada dos píxeles en horizontal y vertical. De los 24 bits por píxeles originales tenemos ahora 12 bits por píxel.

Lo siguiente que se hace es la **división en bloques** de la imagen original. Esta división se hace porque el algoritmo que se utiliza en la etapa posterior opera sobre bloques de 8×8 píxeles. El algoritmo de compresión aplica la transformada discreta del coseno DCT que lo que hace es buscar regularidades periódicas dentro de cada bloque. Es una variante de la transformada de Fourier que explota la redundancia espectral de la imagen. Por cada bloque de 64 valores (8×8), la transformada del coseno genera una matriz de 8×8 donde cada elemento representa la variación en x y en y a una determinada periodicidad: mayor frecuencia de cambio cuanto mayor es el índice en la matriz.

El siguiente paso es una **cuantificación** de los valores resultantes de la codificación DCT. Esto se hace porque el ojo humano es menos sensible a determinados cambios en la imagen. Los valores resultantes de la aplicación del algoritmo DCT se reducen dividiendo por un valor (cuantificación) que es mayor para las frecuencias en las que la visión humana es

menos sensible. Al dividir, tenemos números más pequeños y por tanto más fáciles de comprimir. De hecho, muchos de los elementos de la matriz suelen valer 0. El primer elemento de la matriz se llama coeficiente DC y equivale al color medio de la imagen. El resto de elementos se llaman AC1, AC2, etc. . .

Finaliza aplicando una compresión **basada en entropía** (códigos Huffman) que asigna menos bits a las secuencias de números que más se repiten. Para crear una secuencia los elementos del bloque se ordenan en zigzag.

Entre los problemas del método JPEG, se considera que su **tasa de compresión es baja**: aunque JPEG tiene una excelente relación tasa-distorsión para tasas de compresión medias y altas, a tasas bajas de bit rate la distorsión se hace inaceptable. No permite **acceso aleatorio** a las zonas de la imagen: Los bloques de 8×8 son interdependientes, de manera que para acceder a la información de un determinado bloque hay que recorrer la secuencia de bits codificados mediante Huffman que llevan hasta él. Como el tamaño de estas secuencias puede ser variable debido a la codificación Huffman, no puede ir directamente a un bloque concreto hasta que no hayas leído todos los bloques anteriores al buscado. El algoritmo no está preparado para **imágenes grandes**: no permite la compresión de imágenes de más de 64K por 64K sin hacer *tiling* (división en mosaico). **La arquitectura no es sencilla** ya que tiene 44 modos de funcionamiento y la mayoría de aplicaciones no emplean tantos modos. También da problemas en **transmisión en entornos ruidosos**: si se pierde parte del fichero no es posible recuperar el original. JPEG se creó cuando no había wifi donde los cortes no son improbables.

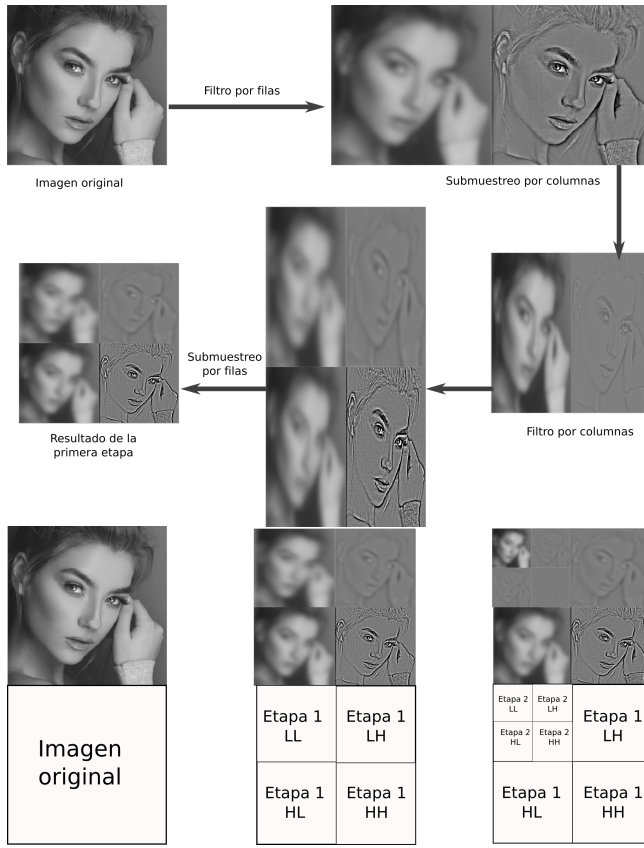
JPEG es especialmente malo para comprimir imágenes generadas por ordenador y para comprimir documentos. El método se optimizó para imágenes naturales, no sintéticas. No va bien con imágenes compuestas por tonos uniformes. Cuando guardamos una imagen sintética con pocos colores en formato JPEG, al ampliar las zonas de cambio de color aparece un ruido relacionado con la separación entre bloques y la cuantificación.

IV-D. Compresión con wavelets JPEG 2000

El formato JPEG 2000 surge para paliar los problemas identificados en el formato JPEG. Para evitar la aparición del enladrillado de los bloques 8×8 ahora se emplea un formato de división en bloques que pueden ser de un tamaño mayor. De hecho, podría emplearse sólo un cuadrado. Los cuadros no pueden ser demasiado grandes para no aumentar las necesidades de memoria de los procesos siguientes.

La siguiente modificación, consiste en emplear una transformación *wavelet* (DWT) en lugar de una transformación discreta del coseno. Antes de aplicar la transformada wavelet y seguir con la cuantificación y la codificación, se procede a realizar un proceso iterativo de división en bandas y subbandas en las que se toman sistemáticamente (mediante la aplicación de filtros) las variaciones de baja frecuencia (bandas L) y las de alta frecuencia (bandas H). El proceso se repite sobre las bandas L. En cada división se hace un submuestro para no multiplicar el número de valores en cada división. El estándar soporta de 0 a 32 etapas. Usualmente se utilizan de 4 a 8 etapas. Las imágenes de la figura 10 ilustran el proceso.

Figura 10: División por bandas en la codificación JPEG 2000

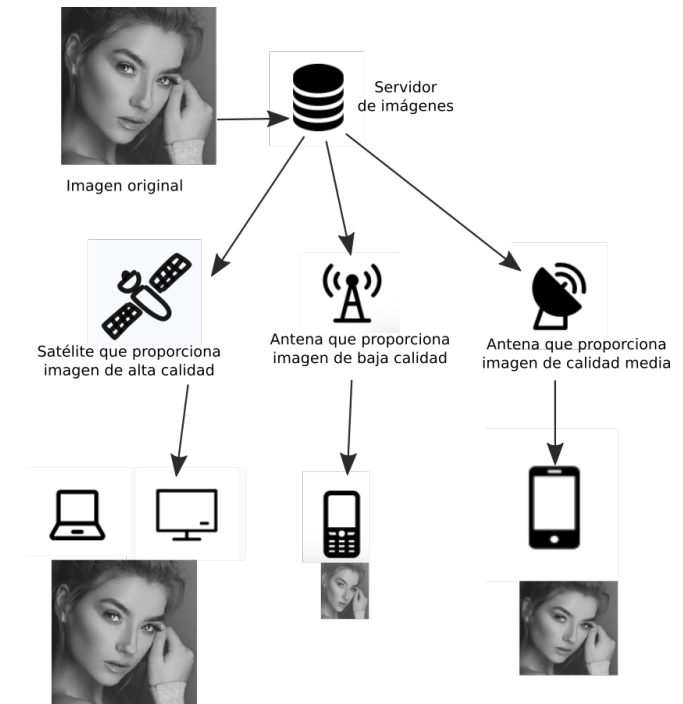


Cada componente (o los tiles de cada componente YCrCb) se codifica utilizando DWT en una serie de subbandas. Las subbandas contienen coeficientes frecuenciales que describen las características frecuenciales horizontales y verticales de los datos originales. Estos coeficientes se cuantifican para reducir la precisión, lo cual produce una pérdida. Los coeficientes se reorganizan en paquetes donde cada paquete contiene los coeficientes en una determinada resolución. Como resultado, los primeros paquetes contienen las aproximaciones de baja calidad de la imagen. Con cada paquete nuevo que llega de la imagen, la calidad de la imagen se puede mejorar de forma incremental. Los paquetes se organizan en bloques y la codificación de entropía se aplica a cada bloque independientemente.

El formato JPEG 2000 es más actual y tiene claras ventajas con respecto al formato JPEG original. La figura 11 muestra cómo este formato es utilizado para describir imágenes de manera que, para una misma codificación, el decodificador decide hasta donde decodifica dependiendo de lo que necesite el receptor final. Además con este formato se puede trabajar con la imagen comprimida para operaciones de acceso aleatorio y transformaciones básicas.

Un concepto asociado a este formato es el de **regiones de interés** en inglés *regions of interest* (ROI). Se habilitan máscaras que determinan el número de etapas de codificación o de decodificación dentro de una zona determinada. De esta manera, habrá zonas de la imágenes en las que se comprime más porque no se consideran importantes reduciendo así el

Figura 11: Distribución de imágenes que explota las ventajas del formato JPEG 2000.



tamaño de la imagen.

IV-E. Compresión fractal

El último esquema de compresión de imágenes que vamos a tratar es el de compresión fractal. Un fractal es una forma geométrica cuya estructura básica se repite sistemáticamente. Este fenómeno es común en formas de la naturaleza como en las ramas de los árboles o en la superficie de las montañas. También existen fractales puramente sintéticos, construidos empleando fórmulas matemáticas y procedimientos recurrentes.

El fractal se define por una **semilla** y por una **función de transformación** que se aplica sistemáticamente a la nueva imagen o subimagen obtenida.

En la **codificación fractal de bloques** en inglés *fractal block coding*, la imagen final es el atractor. Se buscan las transformaciones para que, a partir de una semilla o de una serie de semillas podamos llegar al atractor. La imagen se divide en bloques que son el rango de las transformaciones porque están en el atractor. Cada bloque rango debe buscar el bloque dominio (que también está en la imagen). Hay compresión porque la imagen final se obtiene a partir de unos pocos bloques y de un número de transformaciones.

El método es muy costoso computacionalmente, pero puede ser extraordinariamente eficaz. Existe un formato llamado FIF que implementa codificación fractal, pero no es un estándar de especificaciones abiertas.

V. TRANSMISIÓN PROGRESIVA DE IMÁGENES

El objetivo es poder visualizar o tratar la imagen antes de que haya llegado por completo. Los ficheros GIF utilizan un

formato **entrelazado** que consiste en recibir primero las líneas impares y luego las pares. En JPEG hay tres modos:

- Baseline donde todos los bloques se envían en secuencia y se procesan independientemente.
- Selección espectral donde se transmite primero el DC de cada bloque, luego en AC1 ...
- Successive bit approximation donde se envían primero los bits más significativos de todos los bloques y luego los menos significativos.

En JPEG 2000 este problema se soluciona de forma natural ya que el uso de bandas implica una definición aproximada de la imagen final.

VI. CONCLUSIONES

Las imágenes digitales son un conjunto de pixels ordenados en dos dimensiones. Cada pixel tiene una profundidad o número de pixels que codifican diferentes canales para determinar el color y la transparencia.

Las imágenes digitales se contraponen a las imágenes vectoriales, que definen las escenas con representaciones geométricas.

La compresión de imágenes explota las propiedades de irrelevancia y de redundancia. Los diferentes métodos de compresión tienen su origen en las distintas aplicaciones de las imágenes digitales.

MÁS INFORMACIÓN

Este tema es una adaptación del capítulo correspondiente del libro [1]. En otros libros sobre multimedia también puede encontrarse abundante información, por ejemplo en [2].

El capítulo 8 de [3] explica muy bien la diferencia entre imagen digital y vectorial. El capítulo 10 está dedicado a la fotografía. El capítulo 3 de [4] trata en mucha extensión y en muy poca profundidad el tema de las imágenes digitales.

CUESTIONES Y PROBLEMAS

1. Elabore una definición para el concepto de imagen digital entendida como “still image”.
2. Enumere las tres características básicas que definen una imagen digital.
3. ¿Qué es la profundidad de una imagen digital?
4. Una imagen de 1.024×640 pixels con una profundidad de 3 bytes. ¿Cuánto espacio ocupa sin comprimir? Y si aplicamos un zoom del 200 %.
5. ¿Qué significa que una imagen tenga 8 bits por canal?
6. ¿Cuál es la profundidad de una imagen en BN? ¿Y de una imagen en escala de grises?
7. ¿Qué es una imagen de medio tono?
8. ¿Cómo se mide la resolución en imágenes de medio tono?
9. ¿Qué es un canal alpha?
10. Si una imagen tiene un canal alpha. ¿Cuánto más espacio necesita con respecto a otra que no lo tiene?
11. High color resolution ¿a qué hace referencia?
12. Si una imagen tiene 16 bits por color, ¿Cómo se suelen distribuir los bits para cada canal?
13. Las cuatro relaciones de aspecto más empleadas son 3:2, 4:3, 16:9 y 47:20. Indique algún uso típico de estas resoluciones y cómo se las denomina.
14. ¿Qué es el pixel aspect ratio?
15. ¿Qué es una raw image?
16. Indique dos razones que justifiquen que haya tantos formatos de ficheros gráficos.
17. ¿Cuál es la diferencia entre imagen raster e imagen vectorial?
18. ¿Qué es un metafile en términos de tipo de información gráfica contenida?
19. Para representar la letra “e”, un formato vectorial precisa guardar 12 puntos de control de una curva de Bézier (2 dimensiones, un punto interpolante y dos asideros por cada punto). La misma “e” en formato digital necesita una matriz de 24×36 puntos blanco y negro.
20. Al guardar un fichero en formato digital es preferible utilizar el formato tif al formato JPEG ¿Por qué?
21. Indique un tipo de ficheros gráficos vectoriales, otro de tipo raster y otro de tipo metafile.
22. Describa dos aspectos básicos en los que se basan las técnicas de compresión de imágenes.
23. En qué se diferencia la redundancia espacial de la redundancia espectral.
24. ¿Qué es una técnica de compresión lossless?
25. Para la misma figura, ¿Cuál es la razón de compresión si emplea un mapa de colores?
26. Explique el concepto de paleta de colores (look-up table) empleado por el formato GIF.
27. Una imagen de 1024×640 emplea una paleta de colores de 256 entradas. Cada palabra de la paleta codifica un color de 24 bits. ¿Cuál es el tamaño final del fichero? (paleta más imagen)
28. Explique el concepto de paleta de colores (look-up table) empleado por el formato GIF.
29. Una imagen de 1024×640 emplea una paleta de colores de 256 entradas. Cada palabra de la paleta codifica un color de 24 bits. ¿Cuál es el tamaño final del fichero? (paleta más imagen)
30. A efectos prácticos, qué diferencia hay entre disponer de un canal alpha de 1 bit o de 1 byte a la hora de almacenar una imagen y a la hora de utilizarla.
31. Explique el papel de la redundancia y la relevancia a la hora de comprimir imágenes.
32. ¿Por qué no es una buena idea guardar la imagen de un cartel que no incluye fotografías ni degradados en formato JPEG? ¿Qué formato alternativo emplearía para este tipo de imágenes?
33. Una imagen de 64×64 de cuatro colores RGB 24 bits se puede guardar empleando una paleta o sin emplearla. Haga una estimación del tamaño de esta imagen en ambos casos.

REFERENCIAS

- [1] P. Havaldar and G. Medioni, *Multimedia systems: algorithms, standards, and industry practices*. Course Technology Press, 2009.
- [2] M. Elsom-Cook, *Principles of interactive multimedia*. McGraw-Hill, 2001.
- [3] V. Costello, *Multimedia foundations: Core concepts for digital design*. CRC Press, 2016.
- [4] T. Vaughan, "Multimedia: Making it work," *Yogyakarta: Penerbit Andi*, 2006.

El audio digital como recurso multimedia

OBJETIVOS

El objetivo de este tema es presentar el audio digital como un recurso fundamental de los proyectos y aplicaciones multimedia. Después de este tema, el estudiante deberá:

- Conocer cuáles son las características básicas del audio digital.
- Comprender las necesidades y detalles de las técnicas de compresión de audio.

ÍNDICE

I.	Principios físicos	2
II.	Características básicas del audio digital	2
III.	Compresión de audio	3
III-A.	Necesidad de comprimir audio	3
III-B.	Marco teórico	3
III-C.	Compresión basada en la forma de onda	4
III-D.	Compresión basada en modelos psicoacústicos	4
III-E.	Compresión basada en el modelado de la fuente	6
IV.	Estándares de codificación de audio	6
V.	Conclusiones	6
	Referencias	8

El audio, a pesar de tener un papel secundario en la mayoría de los interfaces persona-computador, tiene gran relevancia porque es el único que explota exclusivamente un modo no visual. Esto lo hace determinante en condiciones de trabajo como pueden ser en aquellas en las que la atención visual deba centrarse en otras actividades (por ejemplo en la conducción) y también lo hace imprescindible para favorecer la interacción de usuarios con discapacidad visual. También es el medio principal para la distribución de contenidos sonoros como la música, la radio o los podcasts.

En este tema, primero se presentan los principios básicos del sonido y la forma de captar y reproducir audio en medios electrónicos. Después, centrándonos ya en audio digital, se detallan las características básicas de este medio para pasar después a describir asuntos relacionados con la compresión de audio. Finalizamos el tema distinguiendo diferentes estándares de compresión así como diferentes formatos de fichero.

I. PRINCIPIOS FÍSICOS

La física describe el sonido como una forma de energía que se transmite a través del medio como una onda de presión [1]. Los cambios de presión en el aire llegan al sistema auditivo del ser humano y se perciben como tal sonido. Cuando el sonido se procesa para ser grabado, almacenado o transmitido hablamos de audio.

Los cambios de presión se pueden capturar por un dispositivo electromecánico, como un micrófono y convertirlo en una señal analógica. Una señal analógica es una señal continua a lo largo del tiempo, en multimedia trabajamos con señales discretas, pero de momento vamos a conocer algo más sobre la señal continua sin discretizar aún.

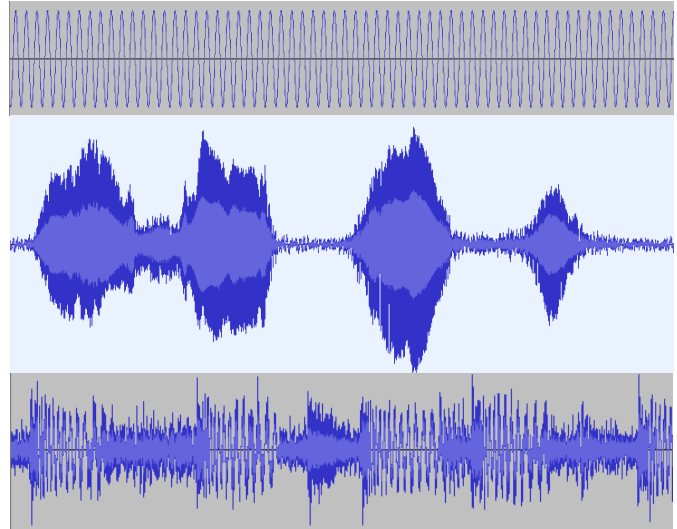
Un **micrófono** convencional consiste en una bobina magnética que se conecta a un diafragma. La onda hace vibrar el diafragma y la bobina se desplaza en el interior de un campo magnético generado por un imán. El movimiento relativo entre bobina e imán induce una corriente eléctrica que es proporcional a la amplitud de vibración. La corriente eléctrica se captura en forma de señal analógica que puede ser grabada por un dispositivo de grabación analógico como, por ejemplo, una cinta de audio. Las variaciones de dicha corriente eléctrica una vez discretizada es la representación del sonido con la que trabajan los dispositivos electrónicos y es lo que llamamos audio digital.

Un **altavoz** convencional tiene un imán fijo y una bobina pegada a un diafragma que puede moverse en el interior del imán. La bobina altera el campo magnético cuando se hace circular una corriente eléctrica por ella. Al cambiar la intensidad de la corriente cambia el campo magnético provocando un movimiento en la bobina. El movimiento de la bobina hace mover un diafragma que genera el sonido al cambiar la presión del aire.

Las señales de audio analógica se representan por su **forma de onda**. Una señal sinusoidal se corresponde con una nota simple y dependiendo de la frecuencia de oscilación se produce una sensación sonora u otra. La amplitud de la onda da idea de la fuerza o intensidad del sonido a lo largo del tiempo. La figura 1 muestra ejemplos de distintas formas de onda asociadas a ejemplos de audio típicos.

La amplitud del sonido se mide en **decibelios** (dB). Es una medida relativa (no absoluta), lo que significa que se obtiene comparando la intensidad sonora I con respecto a un nivel de referencia I_0 . La escala de decibelios es una manera sencilla de comparar dos niveles de intensidad sonora (vatios por centímetro cuadrado). Se define como $10\log(I/I_0)$, donde I es la intensidad del sonido e I_0 es una intensidad de referencia arbitraria. En sonido, I_0 es la intensidad del sonido que los humanos apenas oyen. Experimentalmente este sonido es $10^{-16} \text{watts/sqcm}$. Cualquier sonido por debajo tendrá valores de dB negativos y por encima positivos. En un entorno silencioso como un bosque, se recogen sonidos que están por debajo de los 20 dB; en un hogar hay sonidos por debajo de 40 dB; en una oficina por debajo de 60 dB; en un entorno industrial por debajo de 90 dB; por encima de 90 dB estamos en entornos ruidosos donde el volumen o intensidad sonora es perjudicial para la salud.

Figura 1: Diferentes formas de onda de señales de audio. Arriba la forma de onda de una nota simple y de la misma nota con armónicos. En el medio una forma de onda de voz. Debajo la forma de onda de un tema musical. (Imágenes creadas con Audacity)

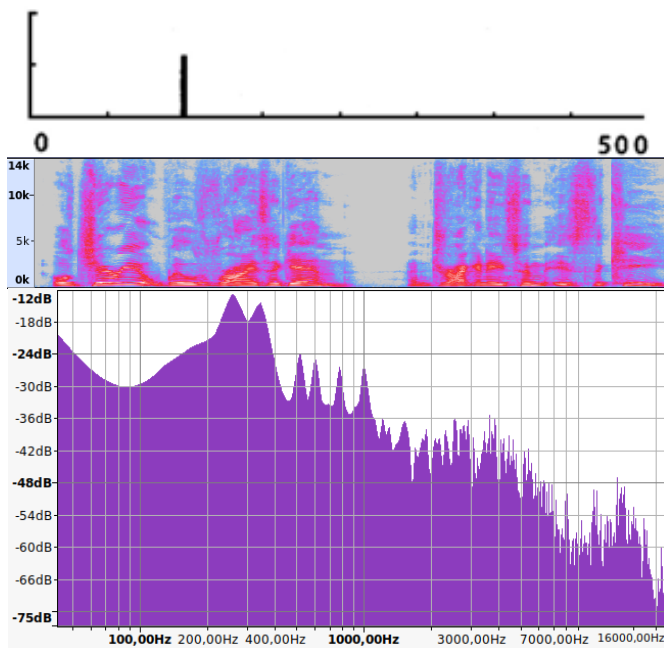


La mayoría de señales de audio se componen combinando diversas frecuencias elementales. Una señal de audio consiste en la suma ponderada de diversas sinusoidales de diversa frecuencia. La amplitud de la señal compleja se forma con la combinación de amplitudes de frecuencias individuales. A la cantidad de cada componente frecuencial necesaria para reconstruir una señal de audio se le llama **espectro de la señal**. La forma de onda es una representación de la señal en el dominio del tiempo, el espectro es una representación de la señal en el **dominio de la frecuencia**. La figura 2 muestra diferentes representaciones de señales de audio en el dominio de la frecuencia. Los sonidos más graves ocupan posiciones más bajas en el espectro frente a los sonidos más agudos. Los ruidos tienen componentes de frecuencia inusualmente altos. La voz humana se compone de frecuencia entre 250 Hz y 8000 Hz. El oído humano capta frecuencias entre 20 Hz y 20 kHz.

II. CARACTERÍSTICAS BÁSICAS DEL AUDIO DIGITAL

El audio digital con el que trabajamos en tecnologías de la información es una representación discreta del audio analógico. La señal analógica se digitaliza mediante un proceso de **muestreo y codificación** que se conoce como **pulse code modulation** (PCM). El muestreo consiste en la toma de valores de la señal a intervalos de tiempo uniformes. El número de muestras por segundo determina el intervalo de muestreo. Dada una muestra de la amplitud de la señal en un instante de tiempo, ésta se codifica con un número fijo de bits. Al proceso mediante el cual se asigna un número a cada valor de la amplitud de la señal se le llama **cuantificación** (en inglés *quantization*). Todas las muestras se codifican con el mismo número de bits. La **tasa de muestreo** (*sampling rate*) y el **número de bits por muestra** (*bit depth* o *sample size*) son los parámetros principales de un fichero de audio PCM.

Figura 2: Diferentes representaciones de la señal de audio en el dominio de la frecuencia. La imagen de arriba se corresponde con la forma de onda de la parte superior de la figura 1, un tono básico. La imagen del centro contiene el espectro de un ejemplo de voz humana. Ahora, para cada instante de tiempo se representa un espectro frecuencial, de manera que cada punto en la figura es un color que representa el valor de amplitud de la componente frecuencial en el instante de tiempo indicado en la ordenada. Los fonemas sonoros como las vocales están por debajo de 2000 Hz y los fonemas fricativos están por encima. En la parte inferior se muestra una representación del espectro de un audio musical en un instante de tiempo dado. Dependiendo del tipo de instrumento se activan unas frecuencias u otras.



Otra característica importante es el **número de canales** de audio que se utilizan: si se utilizara un canal (mono) dos canales (estéreo) o más de dos **canales**. Un canal es una fuente de sonido, generalmente un altavoz. Cuando nos ponemos unos cascos auriculares y escuchamos **sonido estéreo** tenemos una fuente de sonido para el oído derecho y otra diferente para el oído izquierdo. El sistema surround pretende crear sensación de inmersión en el espectador empleando un sistema de altavoces que se disponen alrededor del espectador. Cada altavoz tiene un canal o pista de audio diferente asignada. El sistema **surround 5.1** tiene seis altavoces: el central lleva los diálogos; los fronto-laterales llevan la banda sonora; los otros dos efectos sonoros y ambientales; un subwoofer se utiliza para las bajas frecuencias. Los teatros y salas de cine usan el sistema **THX** o el **Dolby Digital Sound** system. El THX system es 10.2 surround sound system (5 altavoces frontales, 5 laterales y 2 subwoofer)

El **audio espacial** es una alternativa que utiliza menos altavoces para crear sensación de inmersión. Cambia la fase de algunos sonidos del mismo canal consiguiendo que el sonido

puede percibirse como proveniente de una dirección dada.

III. COMPRESIÓN DE AUDIO

III-A. Necesidad de comprimir audio

Cuando se graba audio digital, la primera decisión importante es saber qué frecuencia de muestreo y qué resolución o bits por muestra se van a utilizar. Emplear tasas de muestreo altas y una alta resolución implica más calidad en la grabación, pero también más cantidad de información recogida, y por tanto mayor tamaño en el fichero final si se graba.

Las grabadoras digitales ofrecen calidades de grabación que pueden estar muy por encima de lo necesario si tenemos en cuenta las capacidades del sistema auditivo del ser humano. Por ejemplo, no es raro encontrar programas de grabación y tarjetas digitalizadoras en Windows que ofrecen 64 bits por muestra. La calidad de audio CD a la que estamos habituados, muestrea 44.1kHz con 16 bits estéreo. Cuando empleamos más bits por muestra estamos muestreando la señal con un nivel de calidad innecesario porque, sencillamente, no vamos a poder apreciar la diferencia. Además, el tamaño del fichero final crecerá considerablemente: Si el tamaño en disco o el ancho de banda es un problema, 22.05 kHz a 8 bits se considera calidad suficiente en la mayoría de los casos, y para voz 11 kHz a 8 bits por muestra.

La tabla I muestra cómo el uso de diferentes tasas de muestreo y número de bits por muestra tiene implicaciones importantes en el espacio necesario para almacenar el audio y en el ancho de banda necesario para transmitir.

Ejercicio: el audio en alta fidelidad utiliza 44.1KHz a 16 bits por muestra, ¿cuántos MB de audio pueden grabarse en un CD de 74 minutos de música sin comprimir? ¿Qué velocidad de transmisión debería soportar un canal que permitiera reproducir el audio en tiempo real?

La información ocuparía al menos $74 \text{ min} \times 60 \text{ s/min} \times 44100 \text{ samples per second} \times 16 \text{ bps} \times 2 \text{ canales} \approx 800 \text{ MB}$.

Transmitir en tiempo real necesita una red de un ancho de banda muy alto. En 1 segundo se generan $44100 \text{ samples per second} \times 16 \text{ bits per sample} \times 2 \text{ canales}$, por lo que se necesita un ancho de banda de 1.4 Mbps mínimo.

La mayoría de sistemas de comunicación superan hoy en día este ancho de banda sin problema, no obstante, comprimir es importante porque mucha de esta información es redundante y se puede prescindir de ella sin perder calidad.

III-B. Marco teórico

La señal de audio es unidimensional (1D) porque se trata de muestras de valores de amplitud sonora a lo largo del tiempo. Esto contrasta con la señal de imagen que es bidimensional (2D) al repartir los pixels a lo largo y ancho de una matriz espacial de puntos y contrasta también con la señal de vídeo que es tridimensional (3D), haciendo variar imágenes 2D a lo largo del tiempo. Podría pensarse que la señal de audio es por tanto más sencilla de comprimir que la señal de vídeo y que es también más sencilla que las imágenes pero sin embargo no es así. No es así porque el sistema auditivo humano es más

Cuadro I: Ejemplos de formatos e implicaciones en tamaños y tiempos de transmisión

	Voz (mono)	CD (estéreo)	FM radio (estéreo)	5.1 surround sound
Tamaño de muestra	8 bits	16 bits	16 bits	16 bits
Tasa de muestreo	8 KHz	44.1 KHz	22 KHz	44.1 KHz
Canales	1	2	2	6
Un segundo de habla no comprimida (B bytes)	8 KB	88.2 KB	44 KB	553 KB
Un segundo de transmisión (b bits)	64 Kpbs	1.4 Mbps	704 Kpbs	4.23 Mbps
Tiempo de transmisión de 1 segundo (56Kb modem)	1.14 s	25 s	12.5 s	76 s
Tiempo de transmisión de 1 segundo (780Kb DSL)	0.08 s	1.8 s	0.9 s	5.4 s

complejo que el sistema visual y eso tiene implicaciones en la forma de localizar información redundante.

El **objetivo de la compresión de audio** es minimizar la cantidad de información necesaria para representar la señal a la vez que se maximiza la calidad percibida después de la descompresión. El **codec** es el algoritmo o programa responsable de reducir el tamaño en bits de los ficheros de audio.

En el proceso de compresión de audio, se han de tener en cuenta estas propiedades:

Fidelidad : que describe cómo de diferente, perceptualmente, es el audio que obtenemos a la salida del codec cuando se compara con el audio original. Varía en función de la aplicación: no es lo mismo la calidad telefónica que la utilizada para grabar a una orquesta sinfónica.

Data rate o bit rate : que está relacionado con el rendimiento en el envío y el almacenamiento de audio. Se obtiene teniendo en cuenta la frecuencia de muestro, el número de muestras por segundo y el número de canales. Como hemos visto, tasas más altas implican más calidad en el audio. Se mide en bits por segundo.

Complejidad : que tiene que ver con el coste en software y hardware necesarios para codificar y decodificar la señal de audio. Un codec es más complejo si el proceso de codificación o decodificación tarda más en ejecutarse o si necesita más memoria. En aplicaciones que no son en tiempo real podemos destinar más tiempo para la codificación y decodificación que en aplicaciones de tiempo real como la videoconferencia.

En general es deseable tener una fidelidad alta, con bajas data rate y con la menor complejidad posible. Sin embargo, existe un compromiso entre estos parámetros ya que a más fidelidad más bits por segundo y a menos bit por segundo más complejidad del codec.

Ejercicio: Queremos transmitir un fichero de audio de 3 minutos que tiene calidad CD por un canal de 128Kbps. ¿Qué tasa de compresión mínima debemos aplicar al fichero?

De acuerdo al resultado del ejercicio anterior, el audio CD genera 1.4Mbps por lo que la relación de compresión debería ser de más de 10:1 porque $1400\text{Kbps} > 128\text{Kbps} \cdot 10$

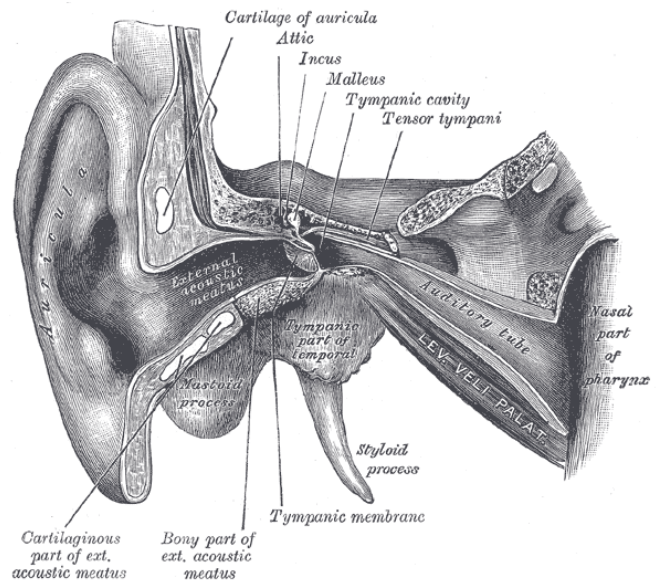


Figura 3: Anatomía del sistema auditivo humano (imagen cortesía de [2]).

III-C. Compresión basada en la forma de onda

En estos algoritmos no se hace ninguna suposición sobre la naturaleza del sonido y se trata simplemente como una señal 1D, esto es, como una serie temporal. Son los primeros métodos en utilizarse. Son sencillos de utilizar aunque poco eficientes. Comentamos a continuación alguno de ellos.

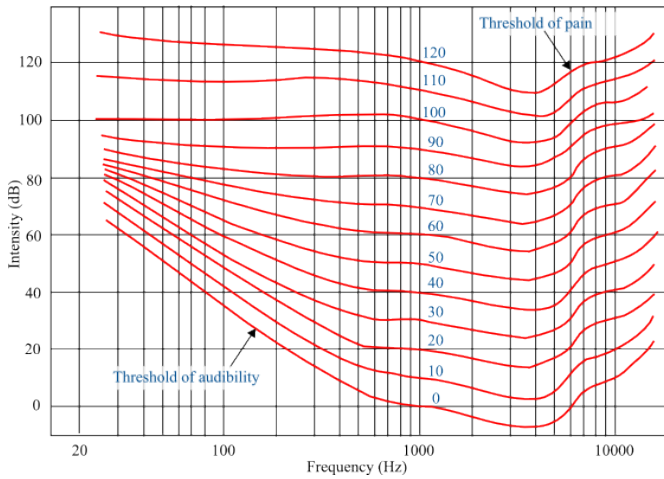
DPCM es el acrónimo de *Differential pulse code modulation*. En lugar de guardar el valor de cada muestra, se guardan las diferencias entre muestras sucesivas. Como tienen un valor similar, los valores a representar tienen un rango de variación menor y podrán emplearse menos bits para codificarlos. Su codificación reduce la entropía, lo que quiere decir que alguno de los valores nuevos (las diferencias) se repiten con mayor frecuencia. Los códigos basados en entropía asignan menos bits a los valores más frecuentes, de ahí que se consiga comprimir.

La variante conocida como *Delta Modulation* nos sirve para ver cómo estos métodos pueden introducir un error importante a la hora de reconstruir la señal. Para representar la diferencia entre la muestra actual y la anterior se emplea con un único bit: 0 significa que el valor se ha incrementado, 1 significa que el valor se ha decrementado. Para incrementar el valor se suma un delta. Para decrementar se resta. El valor del delta para incrementar o decrementar es fijo y depende del rango de variación de la señal. Cada paso introduce un error que depende del rango de variación de la señal.

III-D. Compresión basada en modelos psicoacústicos

La **psicoacústica** es la rama de la psicología que estudia la forma en que el ser humano percibe el sonido. La percepción está condicionada primeramente por la fisonomía del oído. Los cambios en la presión del aire (sonido) entran por el canal auditivo y mueven la membrana auditiva. El movimiento se transmite a la cloquea que excita el nervio auditivo (ver figura

Figura 4: Curva de percepción de sonido por parte del oído humano (imagen cortesía de R. Dadson).



3). La cloquea contiene fibras y fluidos. Su comportamiento no es lineal con respecto a la cantidad de presión y a su frecuencia.

El sistema auditivo es capaz de percibir frecuencias entre 20 Hz y 20 KHz. Antes de comprimir un sonido, puede filtrarse para reproducir sólo frecuencias en el rango audible. Esto explica por qué los CDs emplean audio muestreado a 41KHz. Esta frecuencia es mayor al doble de 20KHz, por lo que el muestreo no capta variaciones por encima de la mitad de la frecuencia de muestreo (Teorema de Nyquist-Shannon [3]). Además, el rango dinámico que puede captar el oído humano es de 120 dB, pero es función de la frecuencia del sonido. La curva de la figura 4 es la curva de percepción auditiva. Sonidos de determinada frecuencia cuya amplitud quede por debajo de dicha curva no se perciben por el oído del ser humano.

El fenómeno del enmascaramiento (*masking o hiding*) limita las frecuencias audibles en función del contexto. Un sonido, analizado en el tiempo, puede descomponerse en diferentes eventos que suceden en instantes de tiempo diferentes. Para poder discriminar entre dos eventos, el oído necesita que estén separados al menos 30ms. Cuando dos sonidos llegan al oído, la presencia de uno de ellos puede ocultar la presencia del otro de manera que sólo uno de ellos es percibido.

En presencia de algún tono, la curva de percepción cambia

Figura 5: Curva de percepción de alterada por la presencia de sonido (cortesía de J. Tost).

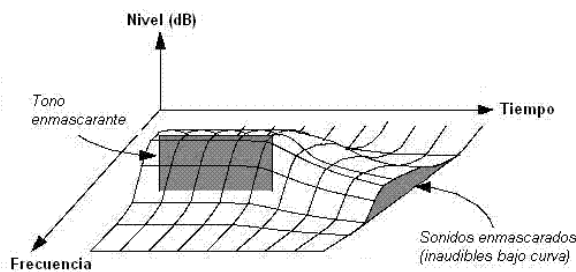
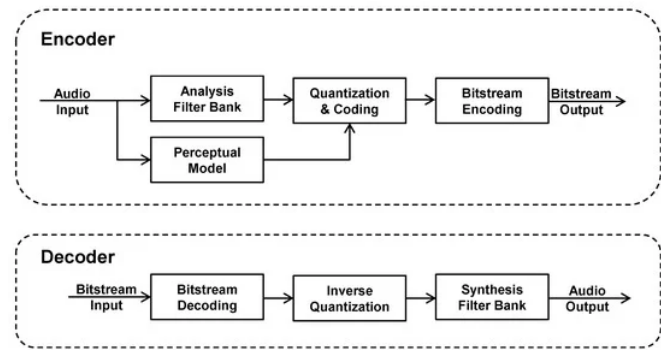


Figura 6: Esquema de la operativa funcional de un codec psicoacústico (de [4]).



de aspecto. La figura 5 muestra cómo las curvas de percepción que habíamos presentado en la figura 4 se ven alteradas por la presencia de tonos de determinadas frecuencias. Los pulsos representan tonos en determinadas frecuencias. Para que un tono cercano pueda percibirse, su intensidad tiene que ser muy alta para no ser enmascarado.

El efecto depende de la frecuencia del tono que enmascara. Por lo general, cuanto mayor es la frecuencia, más enmascaramiento. Empíricamente se demuestra que hasta 500Hz, el enmascaramiento ocupa una banda de unos 100Hz. A partir de 500Hz, aumenta linealmente desde 100Hz a 6KHz.

El oído parece portarse como un filtro pasa banda (deja pasar sólo un rango de frecuencias ignorando el resto). El filtro cambia dinámicamente en función del sonido que se está escuchando. Además, hay que tener en cuenta la persistencia del efecto: Un tono enmascarado no se percibe inmediatamente al desaparecer el enmascarador. Cuanto más cerca están los tonos en términos de frecuencia, más tardará en oírse el tono enmascarado. Cuanto más intenso es el tono enmascarador, más tardará en oírse el enmascarado.

Los codecs basados en modelos psicoacústicos cuantifican la señal en el dominio de la frecuencia. Cada banda de frecuencia se cuantifica empleando un número de bits diferente. Las tablas de cuantificación que determinan el número de bits por banda de frecuencia deben cambiar a lo largo del tiempo por los efectos psicoacústicos.

La operativa de estos codecs es la siguiente:

1. La señal se divide en ventanas que se consideran señales independientes. Se emplean entre 512 y 4096 ventanas. Se aplican diferentes filtros pasa banda para dividir la señal. Se obtienen coeficientes frecuenciales que deben ser cuantificados.
2. En paralelo, se analizan las componentes principales desde un punto de vista perceptual. Se cuantifica cada banda empleando un número de bits diferente. Las bandas que van a ser perceptualmente importantes tienen más bits. Las bandas enmascaradas tendrán menos bits.

La figura 6 muestra un diagrama funcional que representa la operativa de los codecs. Se filtra la señal en ventanas y se estima el número de bits a emplear para cada ventana. Para

descomprimir se reconstruye la señal correspondiente a cada ventana de sonido y se suman los canales correspondientes.

En los codecs psicoacústico el proceso de codificación es costoso porque hay que localizar las bandas enmascaradas y asignar códigos, pero el proceso de decodificación es muy rápido.

III-E. Compresión basada en el modelado de la fuente

Si se sabe cual es la fuente del sonido y cómo es el sonido que emite dicha fuente, no hace falta enviar o guardar la señal de audio como tal, sino que es mejor guardar los parámetros que gobiernan la fuente. Ejemplo, imaginemos una sirena que tiene dos tonos: el tono A y el tono B. Para enviar su sonido, en lugar de enviar el audio, podemos enviar información de los instantes de tiempo en los que está encendido o apagado. Si en el destino sabemos reproducir el sonido de la sirena, con 2 bits podemos codificar el sonido: 0 para indicar sirena apagada, 10 para indicar sirena encendida con tono A y 11 para indicar sirena encendida con tono B.

Para los instrumentos musicales existe el estándar llamado MIDI. Si tenemos un instrumento electrónico compatible MIDI podemos guardar en el ordenador la representación de una pieza musical que hayamos interpretado.

Para la voz también se utiliza este paradigma. En la voz humana hay partes sonoras y sordas. Las sonoras son resultado del paso de la vibración generada en las cuerdas vocales por el tracto vocal. Las sordas son el resultado del paso de un ruido. Los **vocoders** modelan el sistema de generación de voz y producen habla a partir de parámetros como entonación, fonemas, etc... De hecho, empleando un sistema de traducción texto voz, el texto puede ser considerado una representación del audio siempre que en el destino tengamos un sistema que sepa interpretarlo. Un sistema informático que genera audio a partir de texto se llama conversor de texto en voz.

IV. ESTÁNDARES DE CODIFICACIÓN DE AUDIO

MPEG-1 es un estándar de codificación de audio y vídeo. Los layers I, II y III se dedican a la compresión de audio. Los tres layers codifican por bandas de frecuencia y emplean un modelo psicoacústico para asignar más o menos bits a cada banda. La diferencia es que el layer 3, también llamado **MP3**, usa una transformada del coseno llamada MDCT frente a los layer 1 y layer 2 que usan la transformada de Fourier para el modelo psicoacústico. Cuantifica la salida del codificador con un código Huffman basado en entropías, por lo que requieren un buffer para asignar los códigos.

MPEG-2 incorpora sistema Surround (5 canales). Hay dos versiones: la **BC** (*Backward Compatible*) reservada para dar compatibilidad con MPEG-1 y la **AAC** (*Advanced Audio Coding*). AAC incorpora dos características que lo hacen más eficiente que MPEG-1. Por un lado en lugar de codificar cada canal por separado (L/R por ejemplo), primero elimina la redundancia identificando zonas en las que el audio en los canales es igual. Por otro lado aplica un módulo predictivo en el que se tiene en cuenta que las tramas de audio adyacentes pueden ser muy parecidas entre ellas. Este codec se emplea en DVD y HDTV.

La parte 3 del estándar MPEG-4 está dedicada al audio. La principal novedad es que permite adaptarse al tipo de audio, por ejemplo, incluye codecs específicos para voz. Existen diferentes perfiles del codec AAC que se aplican dependiendo de la aplicación. Es muy relevante porque puede incluir información de derechos de autor para evitar copias.

OGG Vorbis es un codec de libre disposición que incorpora las mejoras de compresión de más de 2 canales mejorando en calidad a MP3. Surge por iniciativa de xiph.org para evitar el inconveniente de tener que pagar derechos de autor por el uso de codecs. Si tu aplicación incorpora la posibilidad de guardar o reproducir MP3 debía pagar una licencia. El formato OGG, a diferencia de AAC, no permite incluir la protección de derechos de autor, de manera que se puede evitar que se reproduzcan en dispositivos no autorizados. Esto es usado por ejemplo por Apple en iTunes.

Todos los métodos anteriores utilizan compresión con pérdidas, lo que supone que se pierda calidad. El codec de compresión sin pérdida más extendido se denomina **FLAC** (**Free Lossless Audio Codec**), que como su nombre indica está disponible para su uso público. FLAC emplea predicción lineal de las muestras, lo que quiere decir que la muestra de audio actual se predice en función de las muestras anteriores. Lo que se codifica no es el valor de la muestra sino el valor del error de predicción. Al ser los errores números pequeños, se utiliza una codificación especializada (*Golomb coding*) y después se aplican métodos RLE para reducir secuencias largas como pueden ser los silencios.

Los estándares **ITU G.7xx** son codecs de audio para telefonía. Están pensados para situaciones donde el ancho de banda es limitado principalmente pensados para voz.

Los ficheros de audio aparecen en diferentes formatos como por ejemplo WAV, AIFF, WMA, MP3, MP4, OGG, FLAC etc... Estos ficheros y su formato correspondiente se llaman **contenedores de audio**. Es importante entender que el formato del contenedor y el codec que se emplea son conceptos que pueden estar disociados. Wav suele tener muestras PCM sin comprimir pero también puede tener audio comprimido. Además, los contenedores de audio permiten incluir pistas independientes que pueden contener otra información diferente del audio, por ejemplo transcripciones de texto.

V. CONCLUSIONES

El audio es un fenómeno que captamos y reproducimos mediante dispositivos electrónicos que nos permiten digitalarlo para su posterior procesamiento.

La señal de audio se caracteriza por una amplitud, medida en decibelios, y por un espectro que representa su variación a lo largo del tiempo. Una vez digitalizada, la señal de audio se caracteriza por la frecuencia de muestreo, número de bits por muestra y número de canales.

La compresión de audio es una necesidad para poder almacenar y enviar la señal reduciendo el espacio ocupado y el tiempo de envío. Existen diferentes técnicas de compresión, la más popular la que emplea métodos basados en la psicoacústica.

Existen estándares de compresión y codificación de audio, algunos de ellos permiten comprimir sin pérdidas. Existen

estándares para facilitar el intercambio de archivos entre plataformas.

MÁS INFORMACIÓN

Para profundizar en temas más avanzados de codificación de audio y compresión, [5] ofrece información bastante completa con el capítulo 11 dedicado enteramente a la compresión de audio.

El capítulo 11 de [6] profundiza en la importancia de la elección de buenos micrófono para tener audio de calidad y en proceso de captura de sonido.

El capítulo 4 de [7] profundiza sobre el MIDI y habla de cómo incluir audio en los proyectos multimedia.

CUESTIONES Y PROBLEMAS

1. Explique el principio físico de captura de una señal analógica de audio empleando imanes y bobinas.
2. Decimos que una señal sinusoidal pura es una nota simple. ¿De qué da idea la amplitud de esta señal? ¿Y el periodo del tono?
3. A qué nos referimos cuando hablamos de la información espectral de un sonido.
4. Explique en qué consiste el proceso de digitalización de una señal de audio.
5. A qué hace referencia la tasa de muestreo o sampling rate de sonido digital?
6. ¿A qué hace referencia el número bits por muestra de un sonido digital?
7. Si tenemos un sonido que dura tres segundos y queremos muestrearlo a 16.000 muestras por segundo y a 16 bits por muestra. ¿Cuánto ocuparía el fichero sin comprimir en mono?
8. Diferencia entre un sistema estéreo y otro mono.
9. ¿Cómo se define el sistema 5.1 de audio digital?
10. ¿Qué es PCM y a qué hace referencia?
11. ¿Qué es el sistema de audio espacial?
12. ¿Qué quiere decir que un fichero de audio digital está saturado?
13. ¿Cómo conseguiría el audio de un vídeo en Youtube?
14. ¿Cuál es la razón de que las técnicas de compresión de audio puedan ser más complejas que las técnicas de compresión de imagen o de vídeo?
15. Indique tres propiedades a tener en cuenta para comprimir y codificar la señal de audio.
16. ¿Qué se entiende por fidelidad en compresión de audio?
17. ¿Qué consecuencias tiene emplear un algoritmo de compresión de audio de alta complejidad frente a otros de menos complejidad?
18. Describa un ejemplo de algoritmos de compresión de audio digital basado en la forma de onda.
19. La figura incluye una curva de percepción auditiva. Para poder escuchar un tono a 100Hz hay que emitirlo a más o a menos intensidad que otro de 200Hz.
20. Explique brevemente en qué consiste el enmascaramiento en los modelos psicoacústicos.
21. Decimos que si conocemos la fuente del sonido podemos usar una codificación muy eficiente para los mismos. ¿por qué?
22. El sistema MP3 es un sistema de codificación psicoacústico. ¿Explique brevemente qué significa esto?
23. El sistema MP3 supera MPEG-1, de hecho, es la siguiente versión a MPEG-2. Comente esta afirmación.
24. Indique alguna mejora del sistema AAC frente al sistema MP3
25. ¿Qué codificación de audio utilizan los DVD y HDTV?
26. Indique tres aspectos a tener en cuenta a la hora de comprimir audio digital.
27. ¿En qué versión de MPEG se incluyen propiedades para gestionar los derechos de autor?
28. En compresión de audio, ¿Qué se entiende por fidelidad?
29. Indique la relación entre MP3 y MPEG-2. ¿Qué formatos de audio soporta MPEG-4?
30. Una tarjeta digitalizadora de audio permite capturar audio a una tasa de 41.000 muestras por segundo con 16 bits por muestra. Si la tarjeta captura en estéreo, ¿Cuántas muestras puede ocupar en un minuto sin comprimir?
31. Explique qué representa la curva de la figura. Indique qué es el pulso Masker y el pulso Maskee y qué efecto tienen en la codificación MP3
32. ¿Por qué decimos que la señal de audio es 1D, la señal de imagen es 2D y la señal de vídeo es 3D?
33. Una tarjeta digitalizadora de audio permite capturar audio a una tasa de 16.000 muestras por segundo con 16 bits por muestra. Si la tarjeta captura en mono, ¿Cuántos minutos de audio sin comprimir puede grabar en una tarjeta de 16MB?
34. La nota La es un sonido cuya frecuencia fundamental oscila a 440Hz. Si queremos digitalizar esta nota en calidad CD y almacenar dos minutos de sonido. ¿Cuántos bytes vamos a generar en la digitalización?

REFERENCIAS

- [1] P. A. Tipler and G. Mosca, *Physics for scientists and engineers*. Macmillan, 2007.
- [2] H. Gray, "Anatomy of the human body," *Annals of surgery*, vol. 68, no. 5, pp. 564–566, 1918.
- [3] C. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, jan 1949. [Online]. Available: <https://doi.org/10.1109/jrproc.1949.232969>
- [4] J. Herre and S. Dick, "Psychoacoustic models for perceptual audio coding—a tutorial review," *Applied Sciences*, vol. 9, no. 14, p. 2854, 2019.
- [5] P. Havaldar and G. Medioni, *Multimedia systems: algorithms, standards, and industry practices*. Course Technology Press, 2009.
- [6] V. Costello, *Multimedia foundations: Core concepts for digital design*. CRC Press, 2016.
- [7] T. Vaughan, "Multimedia: Making it work," *Yogyakarta: Penerbit Andi*, 2006.

El vídeo digital como recurso multimedia

OBJETIVOS

El objetivo de este tema es presentar una serie de conceptos técnicos y definiciones básicas relacionadas con el vídeo digital. Después del tema, el lector, como resultados de aprendizaje deberá:

- Conocer los parámetros básicos del vídeo digital.
- Conocer las técnicas fundamentales de la compresión de vídeo.
- Conocer diferentes formatos de vídeo y los conceptos relacionados con las arquitecturas de vídeo.

ÍNDICE

I.	Características básicas	2
II.	Compresión de vídeo	4
II-A.	Codificación y decodificación	5
II-B.	Macrobloques	5
II-C.	Tipo de predicción	6
II-D.	Control de tasa	7
III.	Formatos de archivos de vídeo	7
III-A.	Nombres de formatos	7
III-B.	Estándares de codificación de vídeo	7
III-C.	La arquitectura de vídeo digital	8
IV.	Conclusiones	8
	Referencias	9

El vídeo digital es un medio de extraordinaria importancia debido a su uso masivo. Youtube, por ejemplo, es uno de los servicios de Internet más populares. Los canales de televisión bajo demanda se apoyan en servidores de vídeo digital desde los cuales el público accede a las películas y documentales. La televisión digital terrestre distribuye también vídeo digital.

No debemos confundir vídeo digital con animación generada por ordenador. En ambos casos el espectador percibe sensación de movimiento en la pantalla. Sin embargo, la forma de crear dicha sensación es muy diferente. En una animación, el ordenador redibuja una determinada escena en cada paso de la animación. En un vídeo digital hay una secuencia de imágenes o fotogramas que se muestran a rápida velocidad.

Lo que sí es cierto es que muchas animaciones generadas por animación acaban formando parte de un vídeo digital por ejemplo, los rótulos de las películas o los efectos especiales.

Los vídeos digitales pueden obtenerse de diversas maneras, por ejemplo, con cámaras de vídeo que graban escenas, con editores de vídeo digital que combinan vídeo con imágenes y rótulos para generar un nuevo vídeo digital o pueden ser también el resultado de una animación generada por ordenador como puede ser un vídeo con una simulación o un vídeo con una grabación de renders de animaciones de objetos 3D. Si pensamos en un videojuego, las animaciones que vemos mientras jugamos son animaciones generadas con el ordenador; si grabamos mientras jugamos, el resultado es un vídeo digital.

En este tema, primero vamos a presentar las características básicas del vídeo digital, poniendo el acento en el tratamiento de la profundidad del color. Después, nos centraremos en cuestiones relativas a la compresión de vídeo y el control de tasa de transmisión. Finalizamos presentando los formatos y estándares más utilizados.

Figura 1: El vídeo se compone por una serie de fotogramas proyectados a una velocidad lo suficientemente rápida como para que el sistema de visión los funda en un continuo (vídeo cortesía de T. Miroshnichenko vía Pexels.com CC BY-SA).



I. CARACTERÍSTICAS BÁSICAS

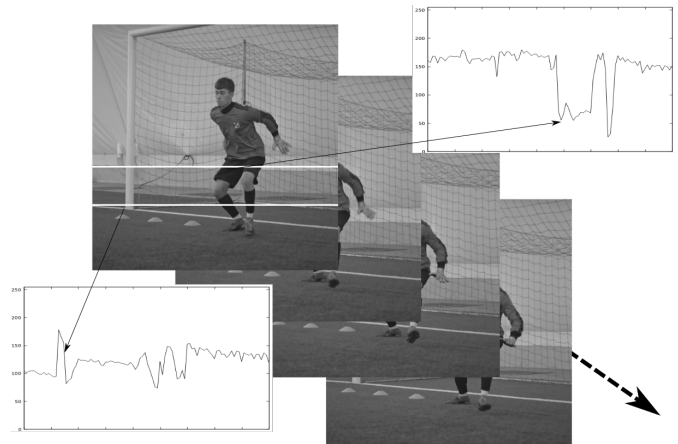
El vídeo es la representación de una secuencia discreta de imágenes que se muestran a gran velocidad (figura 1). A cada imagen que compone el vídeo se le llama *frame* o *fotograma*. Cada fotograma puede ser considerada como una imagen estática que se caracteriza por un **ancho**, un **alto** y un **pixel depth**. Como alternativa a especificar el alto y el ancho del vídeo se emplea el número de líneas. El número de líneas se corresponde con el alto del fotograma aunque hay que tener cuidado porque como veremos a lo largo de esta sección, el vídeo puede estar entrelazado. Generalmente encontraremos las opciones 480p, 720p o 1080p cuando se nos pregunta por la calidad de vídeo que queremos. El número se refiere al número de líneas. La “p” indica que el vídeo es progresivo; si en lugar de una “p” aparece una “i” el vídeo está entrelazado. Explicaremos a qué hace referencia el concepto de entrelazado a lo largo de esta sección.

El pixel depth o profundidad de la imagen permite representar el color del pixel empleando algún modelo de color, típicamente el modelo RGB. Estos atributos permanecen constantes para toda la película. Finalizamos la sección con una descripción sobre el tratamiento que se hace del color en el vídeo digital.

Además de estos atributos básicos hay tres atributos que también son muy importantes: el frame rate, el scanning format y el bit rate. El **frame rate** es la tasa a la que se muestran las imágenes. Las películas del cine se ven a 24 fotogramas por segundo (*frames per second* fps). La televisión analógica 30 fps en el sistema NTSC y 25 fps el sistema PAL. La televisión digital terrestre TDT de alta definición emplea 50 fps y la ultra alta definición 100 fps. Si el frame rate es lento, el ojo humano percibe un parpadeo. La tasa tiene que ser lo suficientemente alta como para poder beneficiarse de la **persistencia de la visión**: las imágenes persisten en la retina durante un periodo de tiempo breve. Si antes de desvanecer llega una nueva imagen similar el cerebro las funde como un continuo.

Para explicar el **scanning format** hay que conocer algo de los fundamentos de vídeo analógico. Aunque el video digital puede ser considerado como una señal 3D —una 2D imagen que cambia en el tiempo— el vídeo analógico se convierte en una señal 1D empleando las líneas de cada imagen. A estas líneas se les denomina líneas de rastreo (*scan lines*) y

Figura 2: Esquema de transformación de los fotogramas en una señal 1D mediante las líneas de rastreo.



son la unidad básica de trabajo en vídeo analógico. El video se transmitía línea a línea; después de la última línea de un fotograma se pasa a la primera línea del siguiente fotograma (ver figura 2). La conversión a línea de scan se hace para garantizar la compatibilidad con monitores de tubo de rayos catódicos CRT. En CRT, un cañón de electrones proyecta un rayo catódico que incide en una pantalla de puntos de fósforo. Cuando el rayo catódico incide con un punto de fósforo, éste se enciende. Con el tiempo el fósforo se apaga por lo que hay que refrescar haciendo que el rayo vuelva a pasar los cada punto tantas veces como indica la tasa de fotogramas por segundo.

Cada línea de rastreo se separa mediante **señales de sincronización** vertical y horizontal. Las señales de sincronización permiten que el rayo catódico vaya de izquierda a derecha y de arriba abajo sistemáticamente. El CRT, cuando recibe una señal de sincronización hace que el rayo catódico cambie de línea o que vuelva al comienzo de la pantalla (ver figura 4). El escaneado se hace de forma progresiva o **entrelazada**. El escaneado entrelazado genera imágenes de vídeo de peor calidad (ver figura 3). El proceso debe repetir a una frecuencia de 60Hz NTSC o 50Hz en PAL. Para hacerlo más eficaz, en el escaneado entrelazado, cada frame se divide en dos: por un lado, las líneas pares y por otro, las impares. En contraposición con el escaneado entrelazado está el **escaneado progresivo**, donde las líneas se rastrean en orden, sin separar las pares y las impares consiguiendo mayor calidad.

Los monitores actuales no necesitarían el rastreo, pero al video digital se llega desde una larga evolución y se heredan aspectos de la tecnología analógica. Los estándares son NTSC (National Television Systems Committee) y PAL (Phase Alternating Line) y SECAM (Système Electronique Couleur Avec Mémoire)

Por último, el **bit rate** se refiere a la cantidad de bits por unidad de tiempo que debemos enviar por un canal de comunicación para poder visualizar la imagen en destino. El bit rate va a depender del ancho y alto del vídeo, de la profundidad y del frame rate. En formatos entrelazados, para

Figura 3: Rastreo entrelazado y progresivo: en el esquema entrelazado se envían las líneas pares de un fotograma y las impares del siguiente. En el esquema progresivo se envían todas las líneas en orden.

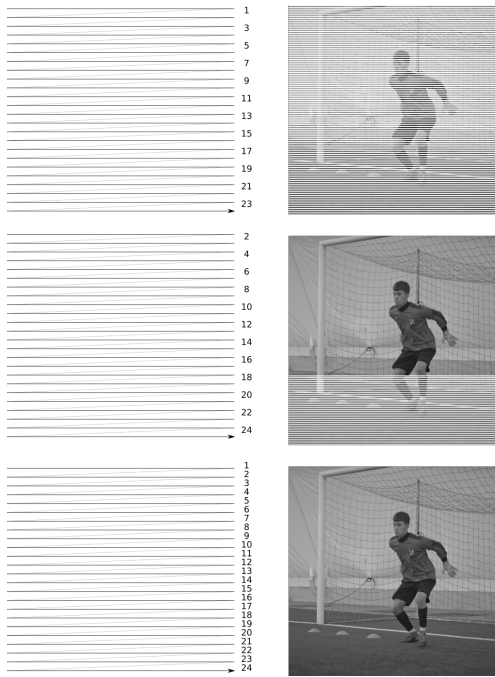
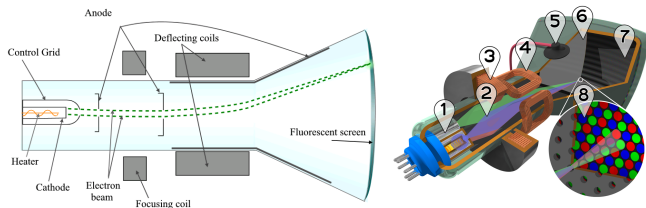


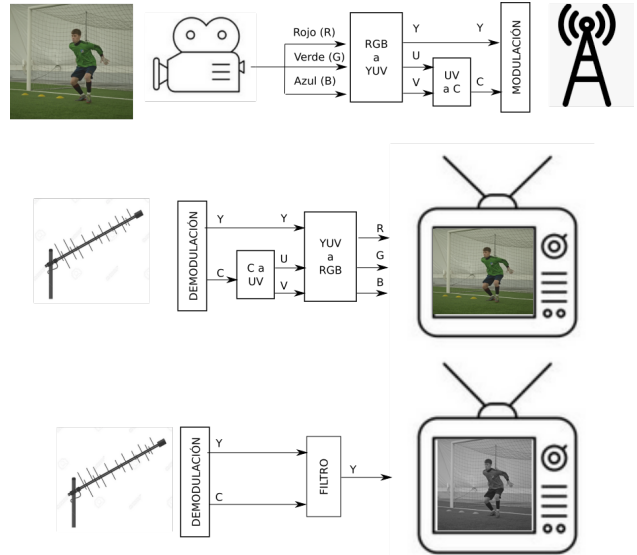
Figura 4: Esquema de funcionamiento de un tubo de rayos catódicos CRT (imágenes de T. Knoth y S. Peo CC BY-SA vía Wikipedia). A la izquierda el mecanismo para monitores en blanco y negro y a la derecha el mecanismo para monitores en color. (1) son los emisores de electrones, (2) son los rayos de electrones, (3) son las bobinas de enfoque (4) las bobinas de deflexión (5) los ánodos terminales (6) la máscara para separar los rayos (7) la capa de fósforo y (8) una ampliación para ver las zonas de rojo, verde y azul en la capa de fósforo.



cada fotograma se envía la mitad de la información y por lo tanto el bit rate se reduce a la mitad. El bit rate también está fuertemente relacionado con el formato de compresión empleado porque el vídeo que se envía por un canal de comunicación está convenientemente comprimido.

Con respecto al **color**, las cámaras fotográficas y de vídeo captan y recogen señal de color que está separada en componentes RGB. Los monitores o dispositivos de visualización también utilizan RGB para visualizar las imágenes y el vídeo. Sin embargo, cuando se transmite una señal de vídeo entre una cámara y un monitor, se emplea un modelo diferente al RGB.

Figura 5: Esquema de conversión de representaciones RGB para el color en representaciones YUV.



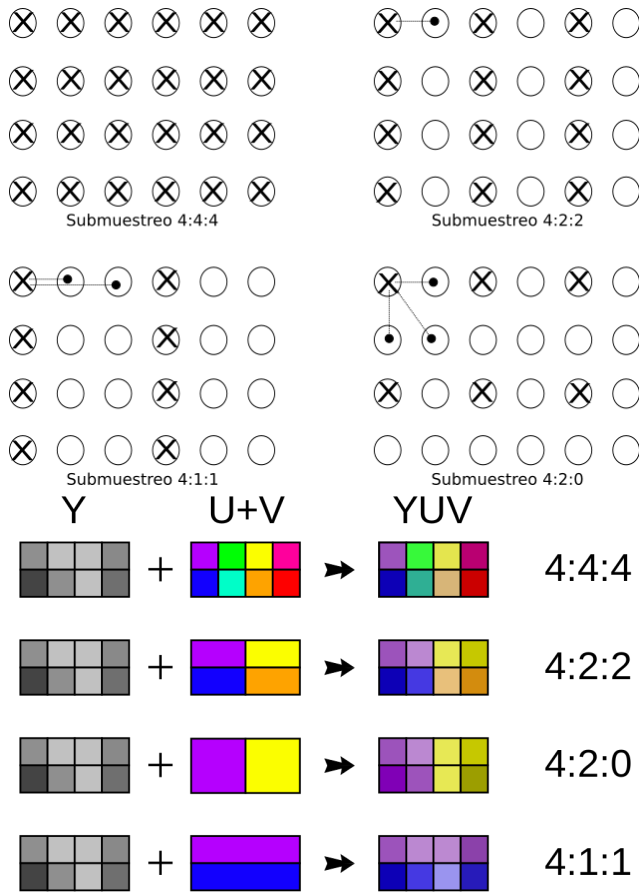
La información de color de los puntos del vídeo se convierten en componentes YUV antes de ser enviada o guardada. En el modelo YUV la información de Y es la luminancia de cada pixel y UV, juntas codifican el color. Se hace esta conversión debido a las siguientes razones:

1. Se desacopla la intensidad o luminancia (componente Y) de la información de color o cromaticidad (componentes UV) porque la componente Y es una buena aproximación de la imagen en escala de grises que se emplear directamente en monitores en blanco y negro.
2. La cromaticidad puede comprimirse más porque el ojo humano es menos sensible a los cambios de color que a los cambios de intensidad. Al comprimir la componente de color se reduce el ancho de banda de transmisión o el tamaño de almacenamiento.

Como hemos apuntado, la información de cromaticidad es menos importante para la visión humana. Como consecuencia, se reduce la calidad de la componente de cromaticidad UV para reducir el ancho de banda necesario para la transmisión. En vídeo digital esto se traduce en reducir el número de bits dedicados a la cromaticidad. Para reducir el número de bits se aplican esquemas de **submuestreo**. Los esquemas de submuestreo más utilizados son:

- Esquema 4:4:4 cada pixel tiene 8 bits para luminancia y (8 bits para U and 8 bits para V), resultando 24 bits por pixel
- Esquema de submuestreo 4:2:2 la crominancia se almacena para cada dos pixels reduciendo el número de bits por pixel a 16.
- Esquema de submuestro 4:1:1 La crominancia se almacena cada cuatro pixels en cada fila. Resulta 12 bits por pixel.
- En el esquema 4:2:0 se almacena la media de valores U para un área de 2x2 pixels U y similar para los valores de V.

Figura 6: Esquema de submuestreo de pixels en vídeo digital (autor de la imagen en color Stevo Domio Público vía Wikipedia).



Como sólo hay 1 bit U y 1 bit V para cada cuatro valores de luminancia, el equivalente en bits por pixel es de 12.

Ejercicio: Un fichero de vídeo MPG tiene un tamaño de 25MB. El ancho de fotograma es de 1280 pixels y el alto de 720 pixels. La velocidad de fotograma es de 29 fotogramas/segundo. Dura 30 segundos. Tiene 2 canales de audio (estéreo) a 44kHz 16 bits por muestra. ¿Cuánto ocuparía el archivo si no hubiera compresión?

Cada fotograma sería tendría 1280×720 pixels. Dependiendo de la codificación de color podemos tener 24, 16 o 12 bits por pixel. Si empleara 24, tendríamos $1280 \times 720 \times 3/1024/1024 = 2,6MB$ por fotograma. Como hay 29 fotogramas en un segundo y el vídeo dura 30 segundos tenemos un tamaño para el vídeo de $2,6 \cdot 29 \cdot 30 = 2294MB$. Además tenemos el audio, que sería $44000 \cdot 2 \cdot 2 \cdot 30 = 5MB$. Como vemos, el audio es lo de menos, tenemos un total de $2299MB$ que se reducen casi a la mitad en función del esquema de color.

II. COMPRESIÓN DE VÍDEO

La tabla I ilustra las necesidades de tamaño de almacenamiento y de tiempo de transmisión de diferentes videos

Cuadro I: Tabla comparativa de distintos tipos de vídeo que ilustra la necesidad de comprimir.

Multimedia video data	NTSC video	QCIF video	CIF video	HDTV (progressive) video	HDTV (interlaced) video
Frame size	720 × 486	176 × 144	352 × 288	1280 × 720	1920 × 1080
Bits/pixel	16 bpp	12 bpp	12 bpp	12 bpp	12 bpp
Frame rate	29.97	30	30	59.94	29.97
Uncompressed frame size (bytes)	700 KB	38 KB	152 KB	1.38 MB	3.11 MB
Uncompressed data produced per second (bits per second)	167.79 Mbps	9.12 Mbps	36.5 Mbps	662.89 Mbps	745.75 Mbps
Disk space to store 1 min of video in bytes	12.6 GB	68.4 MB	273 MB	4.96 GB	5.59 GB
Transmission time for 1 min. of video (56 K Modem)	49.94 hours	2.71 hours	10.8 hours	Not worth it	Not worth it
Transmission time for 1 second of video (780 Kb DSL)	7.31 hours	22.31 minutes	0.775 hours	14.16 hours	15.93 hours
Transmission time for 1 second of video (1 Gb fibre optic)	10.07 seconds	0.55 seconds	2.19 seconds	39.72 seconds	44.7 seconds

digitales sin comprimir. Si la compresión de información es una necesidad en imagen y en audio, en vídeo lo es aún más porque la cantidad de información necesaria es mucho mayor para este medio.

Ejercicio: En las propiedades de un fichero de vídeo encontramos que la Velocidad de datos es de 5942kbs para un vídeo con fotogramas de 1280x720 pixels y 29 fotogramas por segundo. Si originalmente suponemos que la profundidad de color es de 24 bits por pixel ¿Qué podemos afirmar de la tasa de compresión de vídeo?

La tasa de compresión indica la relación entre los datos sin comprimir y los datos comprimidos. En un segundo se generarían 1280×720 pixels/fotograma $\times 29$ fotograma/segundo $\times 24$ bits por pixel/1000Kbps= $641443Kbps$. Como se usa una tasa de $5942Kps$ tenemos una relación de compresión de $641443:5942$ o lo que es lo mismo de $108:1$. Significa que es necesario que en 1 bit del archivo comprimido se represente la información de más de 100 bits del fichero sin comprimir.

Las imágenes se comprimen explotando la redundancia espacial y espectral. Un vídeo es una secuencia de imágenes o frames. Una primera aproximación para comprimir vídeo puede ser comprimir cada imagen de forma individual. Esto es lo que hace la técnica conocida como **Motion JPEG (M-JPEG)**. La cámara toma imágenes, las comprime con JPEG y las envía. El decodificador recibe las imágenes JPEG las decodifica y las muestra. Se utiliza en pequeñas aplicaciones de videoconferencia doméstica donde se envían unas pocas imágenes por segundo. También lo usaban algunos teléfonos móviles, videovigilancia, cámaras de portátiles etc...

La compresión de vídeo explota la **redundancia temporal**: las imágenes consecutivas del vídeo son similares entre sí. Cuando la cámara se mueve, el fondo se desplaza en una dirección. Cuando los objetos se mueven, su movimiento no tiene por qué afectar al resto de la imagen (ver figura 7). Al codificar el vídeo, sólo se codifica la información que

Figura 7: Redundancia temporal: entre distintos fotogramas de vídeo hay parecidos que son predecibles si se identifican las partes móviles o se predice el movimiento de la cámara.

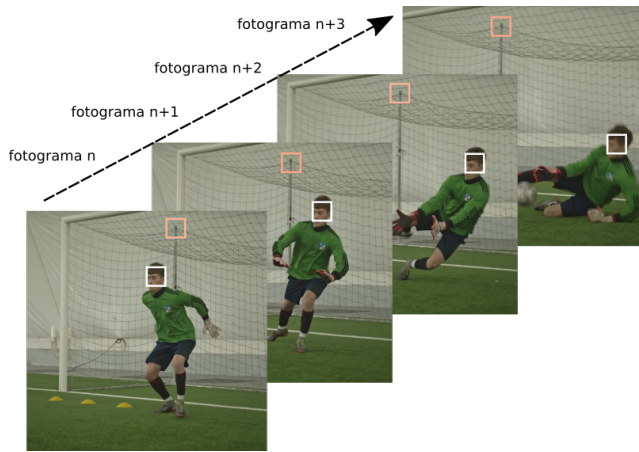
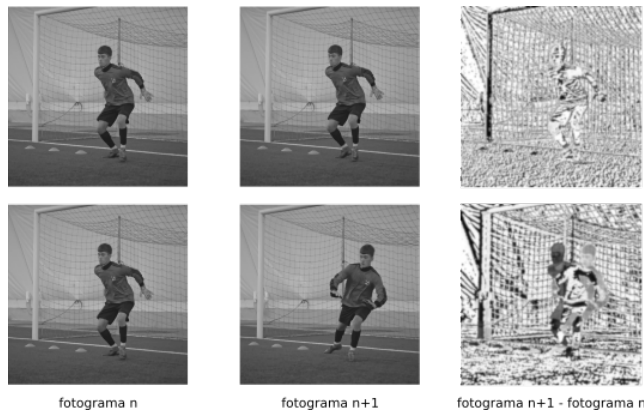


Figura 8: Redundancia temporal: calcular la diferencia entre frames lleva a una nueva imagen y no resuelve el problema. Sólo lo haría si no hay cambios entre imágenes.



cambia entre frames consecutivos, lo que se denomina de forma genérica DPCM (*Differential Pulse Code Modulation*).

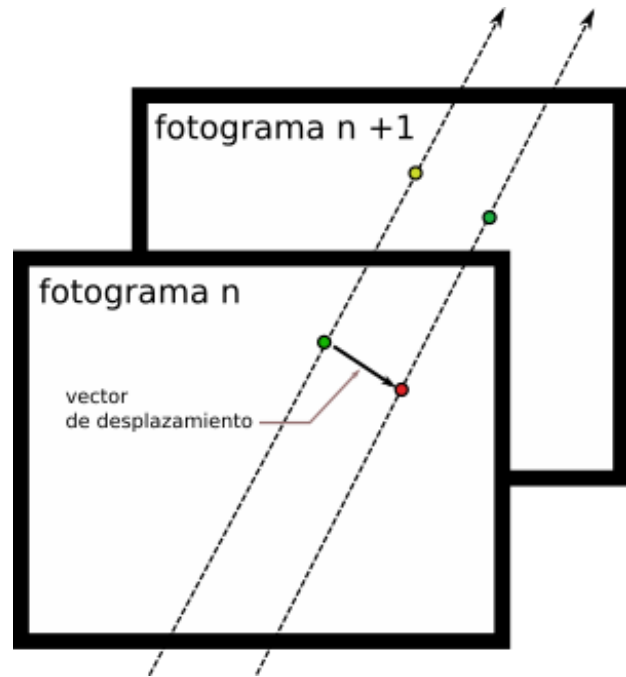
La idea fundamental de la compresión basada en redundancia temporal consiste en intentar codificar sólo los cambios. Podríamos codificar la diferencia entre frames pero no es eficaz. Sólo sería efectivo cuando no hay cambios (ver figura 8). La redundancia temporal se modela matemáticamente estudiando cuánto y porqué cambia cada pixel entre frames.

Sea un pixel en una posición dada (x, y) en el frame n , con unos valores de color de $C_n(x, y)$. En el siguiente frame $n+1$, el pixel en la misma posición, conocido como *colocated pixel*, tiene un color dado por $C_{n+1}(x, y)$.

Podemos predecir el color $C_{n+1}(x, y)$ basándonos en los pixels conocidos del frame n . La relación entre $C_{n+1}(x, y)$ y $C_n(x, y)$ puede representarse de múltiples formas.

Si no hay cambio y la cámara está estática: $C_{n+1}(x, y) = C_n(x, y)$. Si el objeto se mueve o la cámara no está fija o hay cambio de iluminación no estamos en el caso anterior. Las cámaras suelen trabajar a 15 fps o 25 fps por lo que

Figura 9: Redundancia temporal: el color de los pixels se desplaza entre los fotogramas.



los cambios siempre son suaves. El pixel $C_{n+1}(x, y)$ puede no ser igual a $C_n(x, y)$ pero sí igual a algún $C_n(x', y')$ del vecindario.

De esta manera podemos tener que: $C_{n+1}(x, y) = C_n(x', y') = C_n(x + d_x, y + d_y)$ donde d_x, d_y es el vector que muestra cómo se movió el pixel. El vector (d_x, d_y) se llama el **vector de movimiento** (*vector motion*). Debido al ruido por errores de cuantificación en la predicción, podemos definir la ecuación como: $C_{n+1}(x, y) = C_n(x + d_x, y + d_y) + e(x, y)$, donde $e(x, y)$ es el pequeño error de diferencia.

II-A. Codificación y decodificación

La figura 10 muestra el esquema seguido para codificar vídeo y decodificarlo. En codificación, se dispone del fotograma n y el fotograma $n+1$. Con esos dos fotogramas se calculan los vectores de compensación de movimiento. Al aplicar los vectores de movimiento al fotograma n obtenemos una versión aproximada del frame $n+1$ que llamas fotograma predicho. Con el fotograma real en $n+1$ y el fotograma predicho, puede calcularse el error, que se envía como JPEG. Por otro lado se envían los vectores de movimiento.

En decodificación, asumimos que tenemos una versión del fotograma n . Aplicamos los vectores de movimiento al fotograma n para obtener una versión del fotograma $n+1$. A esa versión se le añade la imagen JPEG con la diferencia para obtener una aproximación del fotograma $n+1$.

II-B. Macrobloques

En realidad, no se trabaja por pixels sino por bloques en lo que se conoce como **predicción de frames basada en**

Figura 10: Esquemas de codificación de vídeo (arriba) y decodificación (abajo).

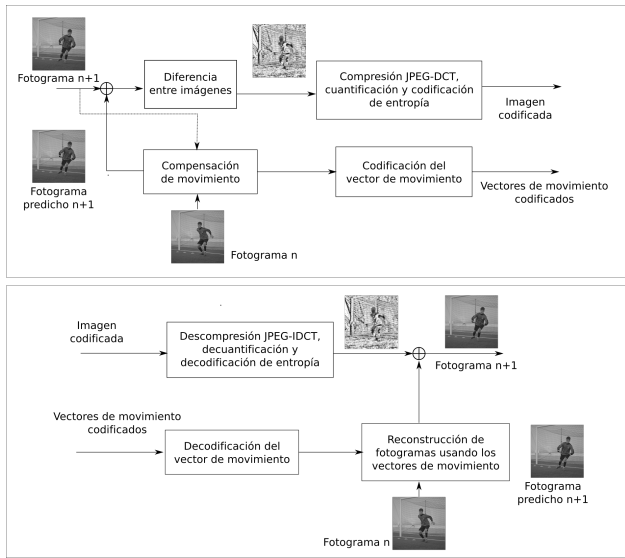


Figura 11: Predicción de tramas P a partir de las tramas I en secuencia.

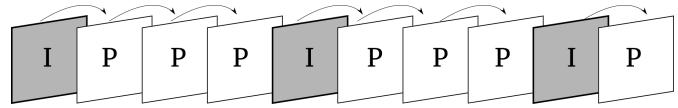
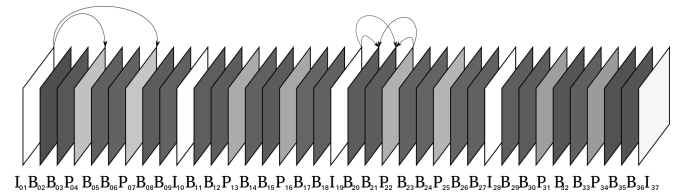


Figura 12: Las flechas muestran las dependencias necesarias para poder generar las tramas. Para poder generar las tramas en orden de visualización es necesario reordenarlas y almacenarlas temporalmente hasta disponer de la información necesaria. Los subíndices muestran el orden de visualización. El orden de transmisión de las tramas sería $I_{01} P_{04} B_{02} B_{03} P_{07} B_{05} B_{06} I_{10} B_{08} B_{09} P_{13} \dots$



bloques. Utilizar los píxeles no es eficiente, son demasiados cálculos. Trabajar por bloques permite reducir el espacio de búsqueda. A continuación, detallamos el procedimiento.

El nuevo frame se llama *target frame* y debe ser predicho a partir del *reference frame*. El target se divide en bloques llamados **macrobloques**. Se buscan los desplazamientos para cada uno de los macrobloques.

El proceso de construcción de fotogramas mediante frames de referencia y macrobloques se llama **compensación de movimiento** (motion compensation) En codificación, el objetivo es encontrar el mejor vector de movimiento para cada bloque. En decodificación, hay que predecir la imagen o crear el target frame motion-compensated. En codificación, primero se deben encontrar los macrobloques más parecidos entre sí y luego calcular el error de predicción de la imagen. Después, se comprime la imagen de error empleando JPEG (lossy) y los vectores de movimiento empleando codificación de entropía (lossless).

El cálculo efectivo de estos vectores es estratégico porque cuanto mejor estén calculados, menor será el error y por tanto más fácil de comprimir será la información del error. Para cada macrobloque del fotograma target, el problema consiste en encontrar el bloque que mejor confronta en el frame anterior. La búsqueda se limita a un área cercana a cada bloque para reducir cálculos.

Determinar el tamaño de los bloques es otro parámetro a tener en cuenta:

- Los bloques pequeños implican más vectores de movimiento a computar. Esto implica más bits para codificar los vectores, pero los bloques pequeños tienden a tener error menores, necesitando menos bits para codificar el error.
- Los macrobloques más grandes llevan a menos vectores de movimiento a comprimir, pero tienden a incrementar el error de predicción. Esto es así porque las áreas más

grandes pueden incluir más áreas que se mueven.

La mayoría de estándares ISO e ITU usan un tamaño de macrobloque de 16x16 que parece que ofrece un buen compromiso. Sin embargo, los estándares más modernos como H.264, utilizan tamaño variable.

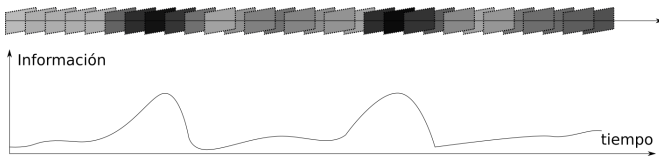
II-C. Tipo de predicción

En función de la relación que haya entre el fotograma target y el fotograma de referencia, hablamos de predicción intramodo e intermodo. En **predicción intramodo** el frame se codifica utilizando el pipeline JPEG. En **predicción intermodo** el frame se predice empleando una referencia. Tal y como hemos presentado el proceso de predicción, podemos pensar que cada fotograma toma su referencia del fotograma anterior. Sin embargo, esto no tiene por qué ser así. Cuando se dispone de todo el material grabado y se quiere guardar el archivo, la referencia puede ser una referencia pasada o una referencia futura.

Estos tres tipos de fotogramas (intramodo, intermodo pasado e intermodo futuro) se intercalan dentro del fichero de vídeo. Para referirnos a ellos usamos las letras I, P, B. Los I frames (intra frame coded) utilizan JPEG y se intercalan dentro del stream de video. Son más grandes, pero no precisan tanto tiempo de codificación al no hacer la búsqueda de los motion vectors. Es una opción interesante en aplicaciones de tiempo real.

Los P-frames hacen referencia a los *predicted coded*. Utilizan la redundancia temporal con respecto al frame anterior, que puede ser de tipo I o de tipo P. La predicción se realiza macrobloque a macrobloque. Se crea una imagen de error que también se codifica. La tabla de cuantificación empleada para el I frame es cuadruple de la tabla empleada para cuantificar el error del P frame. Los fotogramas P son caros de computar

Figura 13: En la parte superior, los fotogramas más oscuros representan fotogramas con más bits. En la parte inferior, el eje de abscisas representa la cantidad de bits que se deben transmitir a lo largo del tiempo.



pero necesarias para la compresión e introducen una relación de dependencia hacia delante, lo cual quiere decir que los errores cometidos en una trama P se van a arrastrar hacia adelante hasta que tengamos un nuevo fotograma I tal y como se describe en la imagen 11. El criterio que se sigue para insertar los fotogramas tipo I es, típicamente, donde el vídeo haga cambios bruscos, que es donde la tasa de compresión predictiva no sea tan buena.

Los B-frames son *bidirectional predicted coded*, lo que significa que utilizan como referencia el frame previo y posterior. A veces el frame pasado no ayuda como puede ser cuando hay un cambio de escena. En estos casos, ayudarse del fotograma siguiente puede ayudar.

Emplear B-frames supone incrementar el coste computacional ya que será necesario comparar si es mejor usar un B-frame o un P-frame. El codificador busca el mejor macrobloque en el frame anterior y también el mejor macrobloque en el frame posterior. Finalmente codifica el macrobloque usando una de estas opciones:

- Forward prediction: usa el macrobloque del siguiente frame.
- Backward prediction: usa el macrobloque del frame anterior.
- Interpolated prediction: usa la media de ambos macrobloques.

Además, Los macrobloques se pueden buscar empleando varios frames futuros y pasados, no necesariamente el anterior y el siguiente. De esta manera, se incrementa la precisión de la predicción pero también el tiempo de cálculo. Esta opción se llama **Multiframe prediction**.

El uso de B-frames implica reordenar los fotogramas. Las tramas llegan al codificador en *display order*, que es el orden en el que se captan por la cámara y es el orden en el que deben reproducirse. El uso de B-frames implica la necesidad de disponer del P-frame para poder codificar y decodificar el B-frame. Esto hace necesario reordenar y emplear buffers de almacenamiento temporal. La figura 12 muestra las dependencias que se generan entre tramas. Para reproducir un frame, hay que haber recibido todas aquellas tramas de las cuales depende la primera. Se envían en orden para que puedan reproducirse lo antes posible teniendo en cuenta las dependencias.

II-D. Control de tasa

Debido al método de compresión de los ficheros de audio, a lo largo del tiempo, el stream de vídeo puede tener más o

menos datos en función de la información a transmitir (ver figura 13). Cuando hay mucho movimiento en el vídeo, será necesario incluir más tramas de tipo I y también habrá más información de desplazamientos. Esto implica más bits por unidad de tiempo.

El control de tasa, es el conjunto de técnicas que se emplean para enfrentarse a la necesidad variable de bps en función del tipo de imágenes que componen el vídeo. Hay dos tipos de estrategia, la estrategia de tasa de bits variable, y de tasa de bits constante.

En la estrategia de tasa de bits variable (VBR *Variable Bit Rate*) se asume que puede ocurrir y no es un problema, simplemente se utilizan buffers en los que va guardando información antes de reproducirla. Si es necesario esperar a que llegue nueva información, por ejemplo, porque la red en la que se transmite el vídeo es muy lenta, se espera a que llegue nueva información. Si la red es rápida, se va almacenando en un buffer temporal a la vez que se reproduce.

La estrategia de tasa de bits constante (CBR *Constant Bit Rate*) es más realista en determinadas aplicaciones como puede ser una videoconferencia. En una videoconferencia, cuando la red es lenta, no se puede esperar a que lleguen los bits del vídeo porque se perdería la comunicación. En estos casos, la tasa de transmisión es constante independientemente de que en el vídeo haya más o menos necesidades de bits porque sea una imagen con mucho movimiento o con poco movimiento. CBR exige soluciones de codificación con pérdidas: en partes del vídeo donde haya mucho movimiento la calidad será muy menor que en partes en las que apenas haya movimiento.

Si la capacidad del canal (los bps que se pueden transmitir) es baja, entonces la calidad de la imagen en CBR será también baja. En VBR cuando la capacidad del canal es baja, tendremos que esperar a haber recibido un número de fotogramas suficientes como para empezar a reproducir, pero la calidad de la imagen podría no verse afectada.

III. FORMATOS DE ARCHIVOS DE VÍDEO

III-A. Nombres de formatos

En la tabla II se muestran los nombres que reciben algunos de los formatos de vídeo más empleados. La lista no es exhaustiva porque algunas aplicaciones establecen sus propios formatos y esquemas de codificación. En la misma tabla presentamos los formatos de vídeo analógico. En comparación, en vídeo analógico era mucho más pobre en lo que se refiere al número de líneas y pixels por línea.

III-B. Estándares de codificación de vídeo

Destacamos los siguientes estándares de codificación de vídeo:

- H.261 Estándar de la ITU para transmisión de videoconferencia en RDSI (ISBN) con tasa de transmisión constante a 64Kbps. Los tamaños de frame son 176x144 QCIF y 352x288 CIF. Emplea vídeo no entrelazado y sólo usa frames I+P.
- H.263 Estándar de la ITU para anchos de banda variable entre 33.6 y 28 Kbps Los tamaños de frame son 704x576

Cuadro II: Diferentes formatos de vídeo digital (arriba) y analógico (abajo).

Format name	Lines per frame	Pixels per line	Frames per second	Support for interlaced format	Subsampling scheme	Image asp ratio
CIF	288	352		N	4:2:0	4:3
QCIF	144	176		N	4:2:0	4:3
SQCIF	96	128		N	4:2:0	4:3
4CIF	576	704		N	4:2:0	4:3
SIF-525	240	352	30	N	4:2:0	4:3
SIF-625	288	352	25	N	4:2:0	4:3
CCIR 601 NTSC (DV, DVB, DTV)	480	720	29.97	Y	4:2:2	4:3
CCIR 601 PAL/SECAM	576	720	25	Y	4:2:0	4:3
EDTV (576p)	480/576	720	29.97	N	4:2:0	4:3/16:9
HDTV (720p)	720	1280	59.94	N	4:2:0	16:9
HDTV (1080i)	1080	1920	29.97	Y	4:2:0	16:9
HDTV (1080p)	1080	1920	29.97	N	4:2:0	16:9
Digital cinema (2K)	1080	2048	24	N	4:4:4	47:20
Digital cinema (4K)	2160	4096	24	N	4:4:4	47:20
Property	NTSC	PAL	SECAM			
Frame rate	30	25	25			
Number of scan lines	525	625	625			
Number of active lines	480	576	576			
Aspect ratio	4:3	4:3	4:3			
Color model	YIQ	YUV	YDbDr			
Primary area of usage	North America (USA and Canada), Japan	Asia	France and Russia			

4CIF y 1408x1152 16CIF Emplea vídeo no entrelazado y tramas I+P+B

MPEG-1 Estándar de OSI para dar calidades similares al vídeo VHS. Emplea vídeo no entrelazado y frames IPB. Es el formato de VCD con tamaños de frame 240x352 o 288x352. Está pensado para anchos de banda de hasta 1,5Mbps.

MPEG-2 Estándar de OSI para uso en DVDs y TDT. Formato entrelazado con predicción de frames IPB Incorpora codificación escalable que se adapta a las necesidades del decodificador y del canal. Usa paquetes que pueden incluir vídeo, audio y como novedad código ejecutable. Trabajan en anchos de banda de 4 a 9 Mbps.

H.264 AVC Estándar de ITU para productos avanzados como vídeo bajo demanda y BlueRay. Es la parte 10 de MPEG-4. Utiliza multi-referencia en predicción de movimiento. Para la codificación intraframe emplea bloques de 4x4. Mejora también la codificación de entropía de los vectores de movimiento. Para la predicción de movimiento bloque emplea bloques de tamaño variable El resultado es la eliminación del efecto bloque.

III-C. La arquitectura de vídeo digital

Una arquitectura de vídeo digital se compone de:

- Un algoritmo para la compresión y codificación del vídeo y del audio o codec. Alguno de los vistos en la sección anterior para el vídeo y para el audio alguno de los vistos en el tema anterior.

- Un contenedor en el que poner los contenidos comprimidos. Serían la secuencia de bits que contiene los datos del vídeo.
- Un reproductor que reconozca y reproduzca los contenidos de los contenedores utilizando los codecs.

Los tipos de contenedor típicos son: OGG (.ogg, Theora para vídeo y Vorbis para audio), Flash Video (.flv), MPEG (.mp4), QuickTime (.mov), Windows Media Format (.wmv), WebM (.webm), Real Media (.rm) etc. . .

Un contenedor puede incluir los datos del stream de vídeo comprimidos con diferentes codecs. Un reproductor debe reconocer el formato del contenedor y disponer del decodificador (decoder). El contenedor puede incluir otros medios que no son ni vídeo ni audio.

MPEG-4 es el contenedor estándar de la ISO que incorpora 2D y 3D vectorial. Permite guardar múltiples streams, lo que significa que permite transmitir varios simultáneamente. Puede permitir, por ejemplo, ver una carrera de automóviles desde diferentes vehículos si estos disponen de una cámara. Este contenedor incluye consideraciones de derecho de autor, lo que fue un fuerte incentivo para que Vorbis desarrollara OGG como competidor para hacer un contenedor de libre disposición.

IV. CONCLUSIONES

El vídeo digital tiene, como características básicas que dependen de la calidad de cada fotograma, del número de fotogramas por segundo, del formato de scanning y de la técnica de codificación del color.

En vídeo digital, la compresión es una necesidad que se deriva del gran volumen de información que hay que representar. Existe diferentes formatos y estándares de compresión que explotan principalmente la redundancia temporal. La técnica básica se apoya en el concepto de la compensación de movimiento que busca macrobloques en fotogramas previos o en fotogramas siguientes.

Como resultado de la compresión se pueden enviar bits a tasa constante o a tasa variable. Para visualizar el contenido del stream de vídeo se emplea una arquitectura de vídeo que incluye el contenedor con las muestras, el codec para descomprimirlas y el visor para reproducirlas.

MÁS INFORMACIÓN

Los contenidos de esta sección son una adaptación del capítulo 3 de [1]. Otros libros más básicos sobre multimedia también incluyen descripciones del vídeo digital. Así, el capítulo 8 de [2] incluye referencias sobre el problema de reproducir imágenes en movimiento, y el capítulo 12 aborda diferentes cuestiones sobre captura de vídeo. El capítulo 6 de [3] está dedicado enteramente a este tema aportando algunas claves sobre cómo realizar captura de vídeo de forma eficiente.

CUESTIONES Y EJERCICIOS

1. Qué son los sistemas NTSC, PAL y SECAM?
2. Explique la diferencia entre los esquemas de muestreo 4:4:4, 4:2:2, 4:1:1 y 4:2:0
3. Si empleamos 30 frames por segundo con frames de 480×300 en un esquema de color YUV 4:1:1, ¿Cuánto tamaño ocupan 2 minutos de video?
4. Para qué sirven y en qué se diferencian los procesos de pan/scan y de letter box.
5. En qué consiste el Motion-JPEG.
6. A qué hace referencia la redundancia temporal en la compresión de vídeo.
7. Qué es un vector motion y qué tiene que ver con la compresión de vídeo.
8. Explique brevemente en qué consiste la predicción de frames basada en bloques.
9. Despliegue un diagrama de bloques de los sistemas de compresión por compensación de movimiento.
10. Relación entre el tamaño de los bloques de un esquema de compresión de video por compensación de movimiento y el tamaño del fichero final.
11. Diferencias entre predicción intramodo e intermodo.
12. En el stream de video digital ¿qué es un I frame y qué ventaja tiene su uso?
13. Qué son los P-frames y cuál es su papel dentro del stream de video digital.
14. Qué es un B-frame y cuál es su papel dentro del stream de video digital.
15. ¿Por qué es necesario emplear buffers de almacenamiento cuando se emplean B-frames?
16. En qué consiste la predicción multiframe y cuáles son sus ventajas.
17. Por qué se produce la tasa de transferencia variable en video digital y como se soluciona.
18. Indique cual es la solución general a los problemas de tasas de transmisión constante CBR en transmisión de video digital.
19. ¿El formato VCD está relacionado con MPEG-1 o con MPEG-2?
20. ¿El formato de video digital de la TDT está relacionado con MPEG-1 o con MPEG-2?
21. ¿El formato de video digital de los DVDs está relacionado con MPEG-1 o con MPEG-2?
22. ¿Cuál es la principal diferencia entre MPEG-4 y sus precedentes?
23. ¿El formato de vídeo del Blu-ray tiene que ver con H.264 o con H.263?
24. Enumere los tres componentes de una arquitectura de video digital
25. El video de Youtube se recibe en condiciones de control de tasa VBR, ¿Cómo puede verse el vídeo sin discontinuidades?
26. En video digital, comente el papel de los bloques en los esquemas de compresión. ¿Cuál es la relación el tamaño de los bloques y el tiempo de compresión?
27. Indique qué es un contenedor de vídeo y cuál es la diferencia entre un códec y un contenedor de vídeo.
28. Enumere tres formatos de contenedor de vídeo digital diferentes e indique quienes son sus propietarios.
29. El video de Youtube, ¿cree que se emite en condiciones VBR o CBR? Justifique su respuesta.
30. Indique qué son los vector motion y su papel en compresión de vídeo.
31. Explique el papel de los buffers en la transmisión y visualización de vídeo digital en condiciones CBR y VBR.
32. Realice el mismo cálculo suponiendo que emplea un esquema de color 4:1:1
33. Si tiene una línea de telecomunicaciones de 10Mbps y un vídeo cuya velocidad de datos una vez comprimido es de 12542Kbps, ¿Qué opciones tiene para visualizarlo on-line?
34. A qué hacen referencia los términos VBR y CBR en transmisión de vídeo digital.

REFERENCIAS

- [1] N. Chapman and J. Chapman, *Digital multimedia*. Wiley Publishing, 2009.
- [2] V. Costello, *Multimedia foundations: Core concepts for digital design*. CRC Press, 2016.
- [3] T. Vaughan, "Multimedia: Making it work," *Yogyakarta: Penerbit Andi*, 2006.

Texto e hipertexto en multimedia

OBJETIVOS

El objetivo de este tema es presentar una serie de conceptos técnicos y definiciones básicas relacionadas con el texto y el hipertexto dentro de las aplicaciones y proyectos multimedia. Después del tema, el lector, como resultados de aprendizaje deberá:

- Conocer qué son los códigos alfanuméricos y su papel en la compatibilidad de sistemas.
- Conocer la importancia del uso de las fuentes.
- Conocer el papel del hipertexto y de la hipermedia en la definición de contenidos multimedia.

ÍNDICE

I.	Códigos alfanuméricos	2
I-A.	ASCII	2
I-B.	Unicode	2
I-C.	UTF-8	3
II.	El arte de la tipografía	3
II-A.	Terminología básica	3
II-B.	Anatomía de una fuente	4
II-C.	Diferentes tipografías	4
III.	Tipografías en tecnologías de la información	5
IV.	Separación entre el mensaje y la fuente	6
V.	Texto con formato, hipertexto e hipermedia	6
VI.	Conclusiones	7
	Referencias	9

El texto es importante en comunicación audiovisual porque permite transmitir mensajes de forma directa a los potenciales receptores. El texto es un recurso multimedia que permite transmitir mensajes tanto de forma explícita con el contenido del mensaje como de forma implícita a través de la tipografía, colores, formas etc. . .

Si en imagen digital la unidad elemental de información es el pixel, en audio digital la unidad elemental es la muestra de audio y en vídeo digital hablamos de fotogramas, en texto digital la unidad elemental es el carácter. Con muchos matices, un carácter se puede corresponder con una letra, un grafema o un glifo. Una secuencia de caracteres forma una cadena de caracteres o “string”. Los mensajes escritos se componen por secuencias ordenadas de caracteres.

Los caracteres se representan internamente en el ordenador mediante números o códigos binarios. La representación de estos números en formas visualmente entendibles por los usuarios de los sistemas informáticos requiere del uso de fuentes tipográficas. En la sección I mostramos los códigos de caracteres alfanuméricos más utilizados. En las secciones II y III presentamos algunas cuestiones relativas con tipografías en general y con el uso de tipografías en sistemas informáticos en particular. En la sección IV se detalla la conexión entre los códigos alfanuméricos por un lado y las tipografías por otro a la hora de renderizar los mensajes escritos. La ventaja del texto con respecto a otros medios es el pequeño tamaño de los archivos. No es necesario ahora profundizar en temas relativos a la compresión de texto. Finalizamos el tema con una sección sobre usos particulares de ficheros de texto de cara a potenciar el renderizado de mensajes y a la inclusión de otros medios dentro de los documentos.

Figura 1: Códigos ASCII (elaboración propia).

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
00	00	00	NUL	32	20	40	 	Space	64	40	100	@	@
01	01	01	SOH	33	21	41	!	!	65	41	101	A	A
02	02	02	STX	34	22	42	"	"	66	42	102	B	B
03	03	03	ETX	35	23	43	#	#	67	43	103	C	C
04	04	04	EOF	36	24	44	$	\$	68	44	104	D	D
05	05	05	ENQ	37	25	45	%	%	69	45	105	E	E
06	06	06	ACK	38	26	46	&	&	70	46	106	F	F
07	07	07	BEL	39	27	47	'	'	71	47	107	G	G
08	08	08	BS	40	28	50	((72	48	110	H	H
09	09	11	TAB	41	29	51))	73	49	111	I	I
10	0A	12	LF	42	2A	52	*	*	74	4A	112	J	J
11	0B	13	VT	43	2B	53	+	+	75	4B	113	K	K
12	0C	14	FF	44	2C	54	,	,	76	4C	114	L	L
13	0D	15	CR	45	2D	55	-	-	77	4D	115	M	M
14	0E	16	SO	46	2E	56	.	.	78	4E	116	N	N
15	0F	17	SI	47	2F	57	/	/	79	4F	117	O	O
16	10	20	DLE	48	30	60	0	0	80	50	120	P	P
17	11	21	DC1	49	31	61	1	1	81	51	121	Q	Q
18	12	22	DC2	50	32	62	2	2	82	52	122	R	R
19	13	23	DC3	51	33	63	3	3	83	53	123	S	S
20	14	24	DC4	52	34	64	4	4	84	54	124	T	T
21	15	25	NAK	53	35	65	5	5	85	55	125	U	U
22	16	26	SYN	54	36	66	6	6	86	56	126	V	V
23	17	27	ETB	55	37	67	7	7	87	57	127	W	W
24	18	30	CAN	56	38	70	8	8	88	58	130	X	X
25	19	31	EM	57	39	71	9	9	89	59	131	Y	Y
26	1A	32	SUB	58	3A	72	:	:	90	5A	132	Z	Z
27	1B	33	ESC	59	3B	73	;	:	91	5B	133	[[
28	1C	34	FS	60	3C	74	<	<	92	5C	134	\	\
29	1D	35	GS	61	3D	75	=	=	93	5D	135]]
30	1E	36	RS	62	3E	76	>	>	94	5E	136	^	^
31	1F	37	US	63	3F	77	?	?	95	5F	137	_	_

I. CÓDIGOS ALFANUMÉRICOS

En informática, cada carácter alfanumérico está vinculado a un número o código representativo de dicho carácter. Esto significa que donde nosotros vemos una "a", el ordenador ve un número. El número en concreto que se asocia a la "a" depende del código alfanumérico que se emplea. Después la "a" puede representarse con fuentes muy diferentes, pero para el ordenador, lo esencial es saber el número asociado a dicha "a" para distinguirla de otros caracteres diferentes. El código ASCII es el código más extendido hasta la aparición de Unicode. En esta sección presentamos las diferencias entre los distintos códigos y la justificación de los códigos UTF-8 de uso masivo hoy en día en tecnologías de la información.

I-A. ASCII

ASCII es el acrónimo de American Standard Code for Information Interchange. El código ASCII emplea números de 7 bits para representar los caracteres. Esto implica que como máximo hay 128 caracteres diferentes ya que $2^7 \geq 128$. El código ASCII representaba minúsculas y mayúsculas para escribir en inglés, números, algunos símbolos y 32 caracteres especiales como los separadores, tabulados y saltos de línea.

El código ASCII hay que entenderlo en un contexto en el que la memoria de los ordenadores era muy cara, por lo que economizar en el número de bits era importante. Con la expansión universal de la informática surge el problema de que a ASCII le faltan símbolos para representar caracteres empleados en otros idiomas como la "ñ" del español. El consenso al que se llega es el de extender ASCII de manera que los códigos nuevos siguen siendo compatibles con ASCII.

Nace así el ASCII extendido en el que se utilizan 8 bits para representar los caracteres. Esto significa que ahora podemos disponer de hasta 256 símbolos ($2^8 = 256$). Para el español, se incluyen otros caracteres como la "ñ" y las vocales acentuadas. El problema es que las reglas para asignar códigos ASCII extendidos no son estándar y cada fabricante podía establecer su propia extensión. De esta manera, por ejemplo, 165 es una cosa en Macintosh y otra en diferente en Windows. También pueden depender del idioma que se esté utilizando. El que

Figura 2: Ejemplos de caracteres representables utilizando Unicode (Autor N. Sancks CC SA-BY vía Wikipedia).

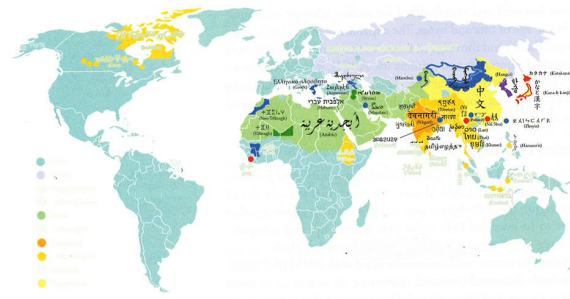


Figura 3: Creación de un carácter compuesto en Unicode (autor M. Regueria Dominio Público vía Wikipedia).

$$\tilde{\text{n}} \equiv \text{n} + \tilde{\text{}} \\ \text{U+00F1} \quad \text{U+006E} \quad \text{U+0303}$$

usamos en España y en la mayoría de países europeos se denomina **ISO-Latin-1**.

I-B. Unicode

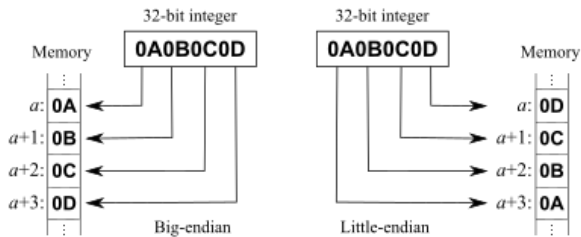
Lo cierto es que a pesar de la ampliación a 8 bits, los códigos de 8 bits seguían siendo muy problemáticos. Muchos fabricantes utilizaban su propia versión ASCII extendido (el comando *iconv*, empleado para convertir entre formatos de texto admite más de 1000 formatos diferentes) e incluso los estándares aceptados como el ISO-Latin-1 tenían carencias como la falta de símbolos muy extendidos como el símbolo del euro €.

Unicode es un estándar que viene a solventar los problemas de los códigos de 8 bits pasando a emplear, en su versión más básica 16 bits. Los principios en los que apoya UNICODE son los de universalidad: debe poder representarse cualquier carácter; eficiencia, con sistemas de codificación específicos; y no ambigüa, donde cada código debe representar un único carácter.

Con 16 bits, hay espacio para tantos caracteres como sea necesario, con lo que ahora el problema era sólo llegar a un consenso. De hecho, no se emplean sólo 16 bits, sino que los 17 planos que se han definido hasta la fecha se resuelven con 20 bits. Los que más se emplean son los 16 bits primeros por lo que es común decir que Unicode es un sistema de codificación de 16 bits. Con Unicode se da soporte a todos los lenguajes conocidos en el planeta e incluso se reservan símbolos para iconos e incluso para alfabetos muertos.

Igual que siempre, a cada carácter le corresponde un número. Se distinguen diferentes planos para organizar los caracteres. Cada **plano** emplea 16 bits. El primer plano (plano 0) es para la mayor parte de alfabetos modernos y alfabetos históricos de uso común; el segundo (plano 1) es para alfabetos históricos y usos técnicos; el tercer plano (plano 2) contiene ideogramas y caracteres raros. Hay un total de 17 planos

Figura 4: Esquema de funcionamiento del almacenamiento y transmisión de información empleando esquemas big-endian y little-endian (Autor R. Shaw Dominio Público vía Wikipedia).



siendo los dos últimos para uso privado de los fabricantes de software.

Para expresar un carácter en Unicode se emplea el hexadecimal. Al menos hay que escribir 4 dígitos. Se emplea la secuencia U+ para referirnos a cada carácter. El carácter U+6C34 es una letra en japonés que en HTML, se escribe `水`. 6C34 es la representación en hexadecimal de 16 bits, 4 bits por cada código.

Unicode permite componer caracteres mediante el uso de secuencias de caracteres. La “ñ” por ejemplo, puede componerse empleando el código del carácter “n” seguido del símbolo del carácter “~” (ver figura 3). Los primeros 256 caracteres de Unicode coinciden con los caracteres de ISO-Latin-1.

I-C. UTF-8

UTF es una familia de formatos de codificación para Unicode. Los problemas de la codificación del texto no terminan con el uso de 16 bits por carácter. Según la arquitectura del ordenador, podemos emplear palabras (unidades mínimas de tratamiento) de 8, 16, 32 o 64 bits. Cuando trabajamos con ordenadores de más de 16 bits podemos pensar que codificar caracteres es tan sencillo como asignar un carácter a cada palabra del ordenador. Sin embargo, hay que tener en cuenta que la información debe ir por la red, donde las unidades mínimas de transmisión son los bits y los bytes. Cuando por la red llega una palabra de 16 bits, debemos saber si los primeros bits que llegan se corresponden con la parte más significativa de la palabra o con la menos significativa. Las arquitecturas de ordenador se clasifican en *big-endian* y *little-endian*. Cuando las palabras tienen más de 8 bits, en las arquitecturas *big-endian*, los bytes más significativos se guardan o envían primero. En las *little-endian* los bytes menos significativos se envían primero. La figura 4 ilustra este comportamiento.

Para evitar problemas con la **serialización** de los caracteres Unicode (envío en serie de bytes por la red), surge UTF (Unicode Transformation Format). UTF-32 emplea 32 bits para enviar cada carácter Unicode. Como los caracteres Unicode tienen siempre longitud menor que 32, es el formato menos problemático: sólo hay que indicar si se emplea codificación *little-endian* o *big-endian*. Es el formato menos problemático pero el más costoso en requisitos de memoria porque un simple mensaje como la palabra “hola”, que en ASCII extendido

ocupa 4×8 bits, en UTF-32 ocuparía 4×32 bits. Los ficheros de texto son 4 veces más grande empleando UTF-32.

La codificación UTF-16 está pensada para enviar ficheros de texto que incluyan sólo caracteres del plano 0. En este caso, el tamaño sólo aumenta el doble con respecto a ASCII. Además, hay que indicar si empleamos *little-endian* o *big-endian*. Para indicar si se emplea uno u otro formato se añade un símbolo llamado BOM (byte of order) al comienzo del fichero.

El formato UTF-8 resuelve la compatibilidad entre ASCII y Unicode y evita además guardar más bits de los necesarios. Es un sistema de codificación, no una tabla de caracteres: emplea más o menos bits dependiendo de los caracteres a representar. Los caracteres más frecuentes se representan con 1 byte. Para otro tipo de caracteres más raros pueden llegar a utilizar hasta 6 bytes. La letra “ñ” por ejemplo necesita 2 bytes, mientras que el resto de letras empleadas en español (exceptuando las letras con acento) emplean un único byte.

Ejercicio: El Lazarillo de Tormes descargado de gutemberg.org/ en formato UTF-8 con BOM ocupa en disco 131.396 bytes y tiene 128.728 caracteres contando espacios en blanco y saltos de línea. Haga una estimación de cuántos caracteres no ASCII-estándar hay en la obra.

Como carácter ASCII estándar ocuparía 1 byte, si llamamos X al número de estos caracteres e Y al número de caracteres no ASCII, se debe cumplir que $X + Y = 128728$ y que $1 + X + 2 \cdot Y = 131396$, asumiendo que los caracteres no ASCII ocupan 16 bits cada uno. Habría 2667 caracteres de un tipo y 126061 del otro.

II. EL ARTE DE LA TIPOGRAFÍA

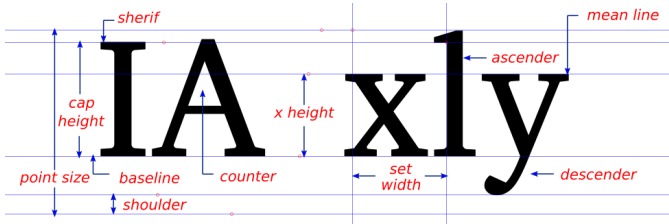
La aparición de las tecnologías de la información en el mundo de la impresión ha revolucionado las tipografías, sin embargo existen aspectos importantes que son heredados de las antiguas técnicas de impresión y que conviene conocer.

II-A. Terminología básica

Las antiguas civilizaciones ya empleaban pequeños objetos para estampar patrones en diferentes soportes como papiros, tablas de arcilla etc. . . con los que configuraban los mensajes. Con la aparición de la imprenta, se denomina **tipo** o **glifo**, en inglés *type* o *glyph*, a cada una de las formas elementales con las que los impresores componen el mensaje que va a ser impreso. Un mensaje escrito se compone como una secuencia de **caracteres**. Un glifo es la representación de un carácter, de varios caracteres o de alguna parte de un carácter. Un impresor podía requerir un glifo para cada vocal y un glifo especial para las tildes. En alfabetos asiáticos un mismo glifo puede representar varios caracteres a la vez. Cuando el glifo tiene forma de letra se denomina **letterform**. La **tipografía** es el arte que se encarga de diseñar y ordenar los tipos.

Una **tipografía**, en inglés *typeface* o también *font family*, es un conjunto de una o varias fuentes, cada una de ellas compuesta por glifos que comparten alguna característica de diseño común. Una **fuentes**, en inglés *font* es una colección de glifos (letras, números y otros. . .) que comparten tamaño,

Figura 5: Anatomía de las fuentes.



grosor y estilo. **Estilos** típicos de fuente son *italic*, *condensed*, *extended*, *thin*, *light*, *bold* etc...

En tipografía digital, fuente y tipografía con sinónimos de manera que Times New Roman italic y Times New Roman bold se consideran la misma fuente, aunque estrictamente no lo sean. Estrictamente, Helvética, Times New Roman y Courier, por ejemplo, serían typefaces y especificando estilo y tamaño, por ejemplo Times New Roman 12-point italic, sería una fuente. Una colección de typefaces relacionadas, es una **familia** (typeface family). Por ejemplo la familia Century: Century Nova, Century Old Style, Century School Book

El ordenador puede añadir otros **atributos de estilo** como pueden ser subrayado o el tachado. No forman parte de la fuente. El editor puede incluirlos si tiene programado cómo utilizarlos.

Antes, el texto se componía letra a letra en las imprentas. Las fuentes estaban guardadas en dos cajones o en inglés *cases*. Esto ha supuesto que buena parte de la terminología sea heredada: El cajón de arriba *upper tray* para las mayúsculas; el cajón de abajo *lower tray* para las minúsculas; *uppercase* se utiliza hoy para referirnos a las mayúsculas y *lowercase* a las minúsculas; a veces, como al escribir los password, el sistema es sensible al hecho de usar mayúsculas o minúsculas: *case sensitive*.

En relación a las mayúsculas se emplea el término *Capital letters* para referirnos a ellas, y de aquí se llega a la escritura en *intercap* que es popular entre programadores, empleando letras mayúsculas para separar palabras como en *WebStar*, o en *holaMundo*. De nuevo el término aparece cuando distinguimos entre AllCaps y Smallcaps.

II-B. Anatomía de una fuente

También relacionado con su origen etimológico están los términos *stroke*, *contrast* y *stress*. Los **stroke** o trazos son los componentes del typeform y pueden ser rectos como en la “l, k, m” o curvos como en la “s” o la “o”. Cuando el *stroke* cambia dentro del typeform se habla de **contrast**. El **stress** es la posición o el ángulo de cambio desde palo fino a palo grueso. En función del grosor **weight** del *stroke* hablamos de Regular, Boldface y Light (Bold incrementa el tamaño del palo). El término **Posture** distingue entre roman, oblique and italic. Roman son rectas, oblique inclinan las respectivas roman e italic incluye aspectos relativos a escritura manual.

El número de puntos no describe bien el tamaño: Dos fuentes pueden tener el mismo tamaño pero distinto **x-height**. Los ordenadores añaden automáticamente bajo el **descender**

un **shoulder** para hacer un espaciado (o leading) entre líneas. La figura 5 muestra estos detalles.

II-C. Diferentes tipografías

Existe una evolución histórica de las tipografías que condiciona diferentes estilos. De más antiguas a más recientes distinguimos las siguientes variantes:

Blackletters : Empleadas en las primeras imprentas. El principal problema es que eran difíciles de leer.

Roman : Utilizan geometrías inspiradas en templos romanos. Los strokes imitan las columnas con sus bases y sus capiteles.

Italic : Se introducen las cursivas por una razón de tipo práctico que no es otra que el ahorro de espacio. Empleando letra cursiva pueden incluirse más palabras por línea sin perder legibilidad.

Old Style : Este tipo de fuentes se caracterizan por el poco contraste entre strokes y normalmente emplean serif grueso.

Transitional : A diferencia de las anteriores, hay más contraste y el serif es más delgado.

Modern : Se imponen los serif muy delgados y grandes contrastes.

Sans Serif Se elimina el serif principalmente para mejorar el impacto visual de los títulos. Una variedad son los **slab-serif** que emplean segmentos rectos en los serif. Están asociadas a la revolución industrial y a la necesidad de hacer anuncios y cartelera.

Geometric : Es un sans serif inspirado en formas geométricas.

Humanistic : Suaviza las geometrías

Ejercicio: Busque ejemplos de cada uno de los tipos de tipografías enumerados anteriormente. Para cada uno de ellos analice las diferencias en los caracteres A mayúscula y minúscula.

Una forma sencilla de clasificar las fuentes es la que distingue entre serif y sans serif. El serif es un pequeño adorno que acompaña a las letras, típicamente un rabito. Son fuentes serif Times, New Century, Schoolbook, Bookman... y sans serif la Helvetica, Verdana, Arial... Tradicionalmente se considera que las fuentes serif son más apropiadas para documentos escritos en papel porque el serif ayuda a la vista a seguir la línea de texto. En el mundo de los monitores, la definición es de 72dpi y las fuentes sans serif son más fáciles de leer y atractivas lo cual es especialmente importante cuando se emplean fuentes pequeñas. Para títulos, una serif puede tener una connotación interesante pero generalmente no se utilizan. La correcta elección depende del soporte final. Pare ver si una fuente es apropiada para un uso final conviene imprimir porque a veces no se usan sistemas WYSIWYG (What you see is what you get) y el resultado puede ser diferente al esperado.

Con la incorporación de las fuentes tipográficas en las tecnologías de la información, la variedad de las fuentes se ha incrementado. Ahora tenemos nuevos términos como son **Decorative typefaces**, diseñadas para atraer visualmente, aportan connotación emocional. **Script typefaces** que intentan imitar

la escritura manual; son muy usadas para anuncios impresos, muy difíciles de integrar en un display. Las **Symbol Typefaces** que permiten emplear caracteres especiales o pequeños iconos.

Ejercicio: Busque ejemplos de cada uno de los tipos de tipografías enumerados anteriormente.

III. TIPOGRAFÍAS EN TECNOLOGÍAS DE LA INFORMACIÓN

Los monitores primitivos de los ordenadores personales estaban pensados para mostrar caracteres de texto. Esto implicaba que la resolución era baja y que los pixels eran alargados (optimizados para texto pixelado). Las impresoras también estaban pensadas para imprimir documentos de texto, ya que la ofimática era el principal uso de los ordenadores hasta el momento. Los gráficos se consideraban aún una aplicación de futuro.

1984 se considera el año en el que se produjo el gran cambio. Apple lanza el Macintosh y como respuesta Microsoft lanza el sistema operativo Windows para PCs con tarjetas gráficas VGA. Con estos sistemas se consiguen resoluciones mucho mayores (72 pixels por pulgada en el Macintosh y 96 pixels por pulgada en VGA), color RGB y pixel cuadrado. Con estas resoluciones se da la posibilidad de trazar gráficos en el display por un lado y se avanza hacia el principio de WYSIWYG (What you see is what you get). El principio WYSIWYG es buscado en el ámbito de la impresión ya que supone que lo que se ve en la pantalla será igual o similar a lo que se imprimirá en la impresora.

En 1985, el mercado de la edición digital explota con la aparición de los editores de texto (Word, WordPerfect, Page Maker), los ordenadores Apple Mac especializados en edición digital y las impresoras láser 300dpi. Adobe había creado el lenguaje PostScript. Es un lenguaje gráfico que dibuja con curvas de Bézier toda la información de las páginas. Las impresoras láser incluyen un módulo hardware PostScript que interpreta el lenguaje PostScript y lo traduce en órdenes de la impresora. Entre la información gráfica que se traduce en PostScript está el texto. PostScript define, para cada carácter, el contorno del mismo con curvas de Bézier. Este contorno es empleado por la impresora para renderizar el texto. La figura

Figura 6: Ordenadores personales que empiezan a necesitar tipografías para explotar los interfaces gráficos. A la izquierda en Machintosh de Apple y a la derecha el PC con VGA de IBM (autores A. Schaelss GNU FDL y R. de Rijke CC BY-SA vía Wikipedia).



Figura 7: Iconos de las fuentes TrueType, Poscript y OpenType a la izquierda. Ejemplo de fuente MetaFont a la izquierda (los iconos de la izquierda son propiedad de Windows).



6 muestra el aspecto de los ordenadores Apple y Microsoft mencionados.

Hasta la fecha los ordenadores trabajaban con versiones bitmap de las fuentes de texto. En las fuentes bitmap, cada letra está representada en una matriz de pixels. Esto era ideal para terminales e impresoras optimizadas para representar texto, pero con la aparición de los display gráficos y las impresoras gráficas las fuentes tipo bitmap dejan de ser atractivas. Microsoft y Apple querían representar fuentes PostScript pero Adobe no se lo permitió.

Ejercicio: Busque una imagen en Internet en el que aparezcan los puntos de control de un carácter Postscript y de otro TrueType. ¿Puede observar la diferencia?

La especificación de PostScript es abierta. Sin embargo Adobe se reserva los **hints** en las fuentes Type 1. Los hints tienen que ver con la forma de renderizar las fuentes cuando hay poca resolución. Los fabricantes dependen de Adobe para hacer impresoras compatibles con PostScript. También los sistemas operativos dependen de Adobe para poder representar texto en las ventanas de los programas de edición.

Los hints tienen que ver con la manera en la que los textos son renderizados. El efecto **jaggie** tienen que ver con el aspecto pixelado de los gráficos que se observa al ampliar su tamaño. Los hints son algoritmos que evitan en los jaggies a la hora de renderizar el texto.

En 1989, Macintosh y Microsoft anuncian un esfuerzo conjunto para hacer fuentes de mejor calidad visual que las fuentes PostScript y que permitan ser representadas de forma más rápida. Nacen las fuentes **TrueType**. Emplean curvas de Bézier cuadráticas y no cúbicas (un punto de control por cada segmento y no dos como es habitual). Imprimen en terminales de 72 dpi y 96dpi. Las fuentes TrueType son de Apple con licencia Microsoft.

OpenType es el formato más utilizado actualmente en la mayoría de sistemas operativos. Nacen de la colaboración de Microsoft y Adobe para librarse de las licencias de Apple de TrueType.

El proyecto **Freetype** nació en 1996 para ofrecer programas informáticos de libre disposición que permitan representar cualquier tipo de fuente. Apple no dejó incluir los hints en estos programas por cuestiones de patentes. En 2010 los derechos de Apple caducaron, de manera que los hits de renderizado de fuentes pueden emplearse sin restricciones. En 2013 Adobe contribuyó al proyecto Freetype con su motor de rendering de fuentes postscript. Android, Linux, Apple,

Java, PlayStation reconocen ser usuarios de Freetype para representar texto en sus aplicaciones.

Debemos hacer aquí una mención especial a las fuentes **MetaFonts** empleadas en el editor de texto *Latex*. En las fuentes PostScript, TrueType y OpenType se dibuja el contorno de las letras (el *outline*). En las MetaFonts, por contra, se representa el recorrido que hace una pluma al trazar cada letra, incluida la inclinación. El resultado son fuentes estilizadas que son muy valoradas en la escritura de textos científicos.

La consecuencia de la diversidad de formatos es que hay una alta dependencia entre fuentes y dispositivo en los que se visualizan. Un mismo texto, usando la misma fuente, puede verse diferente en Windows y en Apple porque pueden usar un fichero de especificación de fuente diferente, porque pueden usar hints o no usarlos o porque pueden usar un motor de render diferente.

Antes de ser utilizada, una fuente tiene que estar instalada en el sistema operativo. Esto significa que el sistema operativo debe tener acceso a la información necesaria para renderizar cada uno de los caracteres de la fuente. Si el sistema operativo no tiene acceso al **fichero de la fuente** correspondiente empleará una similar. Para solventar este problema en el mundo de la navegación web se creó el **Web open font format** que es un formato de fuente optimizado para ser transferido al navegador en el momento en el que deba visualizar una página web. Cada vez que se accede a una página web que contenga texto en un formato concreto, el navegador puede acceder al fichero de fuente correspondiente (que puede estar en un servidor remoto) y representar el texto con precisión evitando así el indeseable efecto de la sustitución de fuente.

Las empresas especializadas en la creación y distribución de fuentes se denominan **font foundry**. Algunas de ellas son www.typequarry.com, www.oldfonts.com, www.linotype.com, www.myfonts.com, www.bitstream.com, www.will-harris.com. Para crear fuentes nuevas se utilizan herramientas específicas como fontographer o fontforge de libre distribución.

IV. SEPARACIÓN ENTRE EL MENSAJE Y LA FUENTE

Hay que entender que en todo mensaje de texto visualizado en el contexto de las tecnologías de información tiene una naturaleza dual: de un lado están los códigos asociados con cada carácter, de otro está la forma y aspecto de cada uno de los caracteres.

En la sección I hemos visto que cada cadena de caracteres tiene asociada una secuencia de números o códigos asociados a cada uno de los caracteres. En las secciones II y III hemos visto cómo el aspecto de los mensajes de texto dependen de la tipografía y fuente empleados para renderizarlos.

Cuando hablamos de fichero de texto o de cadenas de caracteres, generalmente hacemos referencia a la secuencia de códigos alfanuméricos asociado. Para visualizar el mensaje en sí, es necesario que un algoritmo interprete esta secuencia y renderice cada carácter en el símbolo o glifo correspondiente. Para poder mostrar el contenido de una cadena de caracteres de forma legible, es necesario pues convertir los números en símbolos legibles, en caracteres gráficos o en glifos. El responsable de realizar esta conversión será el sistema operativo o el

programa que muestre la información textual, por ejemplo un editor de textos, un navegador o un entorno de programación. Para que estos programas conviertan los códigos alfanuméricos en glifos es necesario indicar qué fuente debe utilizarse y cómo utilizar la información de la fuente para renderizar los glifos.

Es muy importante entender que en algún momento hay que decir a los programas que visualizan texto qué fuentes deben utilizar. La **vinculación de las fuentes** a los mensajes de texto se le llama asignación de la fuente y es una operación extraordinariamente importante.

La información relativa a las fuentes está pues separada del contenido de los mensajes. Suele estar disponible en archivos suplementarios a los que accede el programa que renderiza los mensajes a través del sistema operativo. Un error muy típico es el **error de fuente desconocida** que se produce cuando editamos un mensaje de texto con un programa determinado (por ejemplo Powerpoint) para que se visualice con una fuente determinada, pero, al mostrar el mensaje en otro ordenador diferente la fuente no está disponible. Si la fuente empleada en el ordenador de edición no está instalada en el ordenador de visualización, el sistema operativo no la va a encontrar y utilizará otra fuente diferente **font substitution**, que con suerte, será similar. Aunque pueden existir tablas de correspondencia entre fuentes, el resultado puede ser muy malo.

En el caso de cambio de sistema operativo, por ejemplo porque estamos editando en Windows y después visualizamos en Mac, el problema se complica porque en este caso, las fuentes, aunque se llamen igual pueden ser ligeramente diferentes, bien porque el diseño es diferente, bien porque el motor de render puede utilizar hints diferentes dibujando lo mismo pero de diferente forma.

Una forma de evitar este problema es utilizar **fuentes incrustadas** (*embedded fonts*). En este caso, el programa responsable de mostrar el mensaje se asegura que las fuentes van a estar disponibles porque las añade al fichero en el que está el texto. Esta es la solución que utiliza Acrobat PDF para evitar problemas con las fuentes. Incrustar una fuente tiene el inconveniente de que el tamaño del archivo puede llegar a crecer bastante ya que hay que guardar en el mismo fichero tanto el texto como las fuentes.

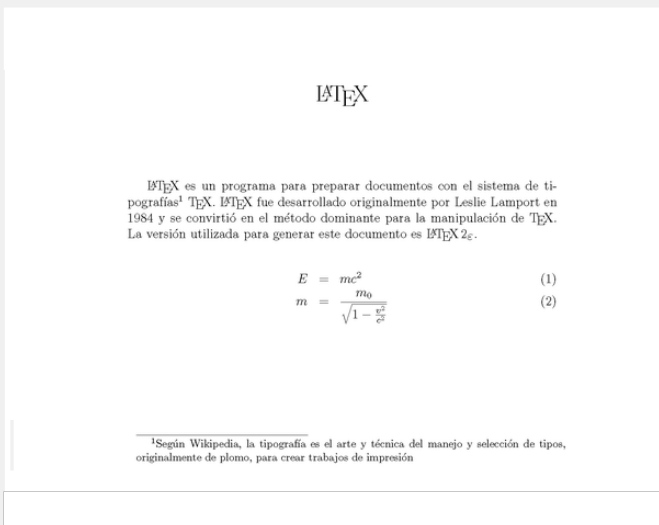
Otra solución consiste en emplear **fuentes online**. En este caso, cada vez que se necesite visualizar un mensaje de texto se accede a la información de la fuente por internet.

Una tercera solución consiste en renderizar el mensaje en el fichero. En este caso, el texto pasa a convertirse en imágenes de manera que la dependencia de las fuentes se pierde. Esta solución tiene el problema de que editar de nuevo el texto es imposible.

V. TEXTO CON FORMATO, HIPERTEXTO E HIPERMEDIA

En ocasiones encontramos que en el mismo mensaje de texto aparecen símbolos especiales que se utilizan para indicar el estilo en el que los mensajes deben renderizarse o interpretarse. Hablamos entonces de **texto con formato**. Un lenguaje recoge el conjunto de combinaciones posibles. Latex, HTML, openOffice son ejemplos de lenguaje que permiten definir ficheros de texto con formato (ver figura 8).

Figura 8: Ejemplo de texto con formato Latex que después se traduce en una imagen con el documento final (autor Shooke Dominio Público vía Wikipedia).

código	resultado
<pre> \documentclass[12pt]{article} \usepackage[spanish]{babel} \usepackage{amsmath} \title{\LaTeX} \date{} % Este es un comentario, no será mostrado en el documento final. \begin{document} \maketitle \LaTeX{} es un programa para preparar documentos con el sistema de tipografias\footnote{%nota al pie de página} Según Wikipedia, la tipografía es el arte y técnica del manejo y selección de tipos, originalmente de plomo, para crear trabajos de impresión } %fin nota al pie de página \TeX{}. \LaTeX{} fue desarrollado originalmente por Leslie Lamport en 1984 y se convirtió en el método dominante para la manipulación de \TeX. La versión utilizada para generar este documento es \LaTeXe. \newline % El siguiente código muestra la calidad de la tipografía de LaTeX \begin{align} E &= mc^2 \\ m &= \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}} \end{align} \end{document} </pre>	 <p style="text-align: center;">LATEX</p> <p>LATEX es un programa para preparar documentos con el sistema de tipografías¹ TEX. LATEX fue desarrollado originalmente por Leslie Lamport en 1984 y se convirtió en el método dominante para la manipulación de TEX. La versión utilizada para generar este documento es LATEX 2_ε.</p> $E = mc^2 \quad (1)$ $m = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}} \quad (2)$ <p><small>¹Según Wikipedia, la tipografía es el arte y técnica del manejo y selección de tipos, originalmente de plomo, para crear trabajos de impresión</small></p>

En los ficheros de texto con formato encontramos **etiquetas, marcas, macros o comandos** que sirven para indicar el papel del texto dentro del mensaje final. Por ejemplo en latex podemos escribir `\section{Titulo 1}` para indicar el comienzo de una sección nueva en un documento y en HTML `<H1>Titulo </H1>` para el mismo fin.

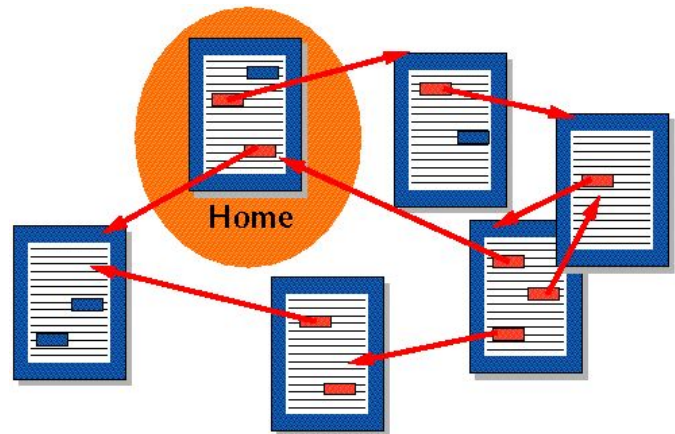
El **hipertexto** consiste en la vinculación de fragmentos de texto con otros fragmentos de texto o de documentos. Desde un determinado fragmento de texto (por ejemplo desde una palabra), podemos acceder a otra parte del texto o a otro documento diferente mediante un **enlace**. La disposición de diferentes enlaces en diferentes puntos de múltiples ficheros crea una red de vínculos que permite acceder a la información de manera no secuencial. El hipertexto es la base de la información disponible en web. El lenguaje **HTML** (*hypertext markup language*) es el lenguaje en la que declaran los documentos y los vínculos. La figura 9 muestra un ejemplo de documentos enlazados entre sí utilizando hipertexto.

Hablamos de **hipermedia** cuando se reservan marcas o etiquetas para colocar, dentro de un documento de texto, contenido multimedia como imágenes, vídeos o sonido. El lenguaje HTML incluye marcas especiales para incluir enlaces a estos recursos multimedia que serán incrustados en la posición correspondiente por el programa de renderizado, generalmente por el **navegador**. La figura 10 muestra un documento que integra texto e imagen especificando la composición en hipertexto.

Para separar la información de formato del contenido se creó el lenguaje **css** (*Cascading Style Sheets*). Dentro del código HTML de cada página web se indica qué hoja de estilo css utilizar para representar cada elemento del documento. Las hojas de estilo sirven para hacer explícita la separación entre contenido y formato que hemos presentado en la sección IV.

OpenOffice es una especificación abierta en formato hipermedia de los documentos creados con algún programa del fa-

Figura 9: El hipertexto permite enlazar entre sí diferentes fragmentos de texto (autor A. Riverac, vía Wikipedia).



moso paquete Office de Microsoft (Excel, Word, Powerpoint). Los documentos office aparecen descritos como documentos de texto en formato XML. El **formato XML**, al igual que HTML se basa en marcas que emplean los símbolos `>` y `<`. En el caso de XML se incluye una definición de documento para poder representar una gran variedad de tipos de información, incluyendo bases de datos completas.

VI. CONCLUSIONES

Los textos tienen una doble naturaleza: numérica (los códigos) y estética (las fuentes). Existen diferentes códigos ASCII, ASCII extendido y Unicode. Unicode tiene diferentes codificaciones. Responden a la necesidad de ahorrar tamaño.

La elaboración de fuentes es un proceso sofisticado y costoso con una terminología asociada que es importante co-

Figura 10: El concepto de hipermedia tiene que ver con la posibilidad de emplear enlaces de hipertexto para incluir otro tipo de medios como imágenes, vídeos o sonidos.



nocer. Los opentype y webfonts han estabilizado la evolución tecnológica.

El uso de texto con formato y el hipertexto, junto con un motor de render apropiado permite crear layouts multimedia.

MÁS INFORMACIÓN

En el capítulo 9 de [1] puede encontrarse una buena descripción de las cuestiones relativas a las tipografías. También en este libro se dedica un capítulo a hablar de HTML (capítulo 7)

Un similar tratamiento del tema se hace en [2] donde el capítulo 2 presenta detalles sobre codificación de texto y tipografías y los capítulos 12 y 13 profundizan sobre HTML.

Existe una amplia bibliografía sobre tipografías que aborda el tema fundamentalmente desde el punto de vista del diseño, pero que incluye toda la terminología que hemos presentado en este tema [3].

CUESTIONES Y EJERCICIOS

1. Escriba una definición personal de tipografía. ¿Por qué es tan importante?
2. Enuncie la diferencia entre typeface y font.
3. ¿Qué es una familia de typefaces?
4. Enumere tres estilos que pueden caracterizar una fuente.
5. En qué se diferencia una fuente oblicua de una fuente itálica.
6. ¿En qué se diferencia una fuente de palo fino de una fuente de palo grueso?
7. ¿Qué indica que una fuente tenga un tamaño de 16 puntos?
8. ¿Qué es el leading? ¿De dónde viene este nombre?
9. Indique las diferencias en el uso de las fuentes serif y sans serif.
10. En ocasiones se habla de typefaces de tipo Humanist, Italic, Old Style, Transitional... ¿A qué se debe esta distinción?
11. ¿Qué es un decorative typeface? Indique sus usos.
12. ¿Qué es un script typeface? Indique cuáles son sus usos.
13. ¿A qué hace referencia el término case?
14. Comente la afirmación: "Fijo el tipo de letra para asegurarme de que todos los que vean mi documento en el ordenador vean lo mismo".
15. Indique el nombre de las tres empresas más importantes que estuvieron implicadas en la llamada "Guerra de las fuentes".
16. ¿Qué es PostScript y quién lo inventó?
17. ¿Qué es un Metafont? ¿Qué es una font foundry?
18. ¿Por qué existe el ASCII Extendido? ¿Qué es ISO-Latin1?
19. ¿Qué es UNICODE y por qué existe?
20. ¿Qué es UTF-8 y para qué se utiliza? ¿Qué es ISO-Latin1 y cuál es su relación con UTF-8?
21. ¿En qué consiste la técnica de Font substitution?
22. Rodrigo ha utilizado un decorative typeface bastante particular para el título de su presentación. La tipografía es tan rara que no espera que haya muchos ordenadores que la tengan instaladas. ¿Qué puede hacer para garantizar que su presentación va a ser correctamente visualizada en otros ordenadores?
23. ¿Qué es una fuente incrustada o embebida? ¿Qué problema pretende resolver? ¿Cómo se utilizan en Flash?
24. El profesor me ha dicho "Debe entender que la fuente y el contenido textual están separados. Así es y así debe ser". ¿Qué recurso tenemos en HTML para conseguir esto? ¿Y en Flash?
25. En tipografía ¿Qué son los jaggies y por qué aparecen?
26. ¿Qué es UNICODE. Indique las diferencias entre ASCII y UNICODE
27. Imagine que le ha tocado montar el máster de un proyecto multimedia Flash que carga diferentes archivos SWF en una única película. Al cargar un determinado fichero SWF en el máster, el compañero de trabajo que ha creado dicho archivo se da cuenta de que el texto es diferente a como él lo había diseñado en su ordenador (la fuente no se corresponde). ¿Qué ha pasado? Hay dos formas de evitar esto. Indique cuales estas son estas formas y las implicaciones que tienen desde el punto de vista del tamaño de la aplicación final.
28. ¿Qué es UNICODE. Indique las diferencias entre UTF-8 y UNICODE
29. Un fichero de texto de 2000 caracteres es guardado en tres formatos de texto diferentes: ASCII, UTF-8 y UNICODE ¿Qué puede decir del tamaño de cada uno de los ficheros resultantes sin comprimir?
30. Un fichero de texto contiene 2000 caracteres. Al guardarlo en formato ASCII extendido el tamaño es de 2000 bytes, sin embargo al guardarlo en UTF-8 resulta que es algo mayor. ¿Por qué pasa esto? ¿Puede haber algún caso en el que no pase esto?
31. Defina los términos intercap y case sensitive e indique la etimología de los mismos.
32. Diferencia entre fuentes bitmap y trueType. ¿Qué papel jugaron ambos tipos de fuentes en el avance de la informática doméstica?

REFERENCIAS

- [1] V. Costello, *Multimedia foundations: Core concepts for digital design*. CRC Press, 2016.
- [2] T. Vaughan, "Multimedia: Making it work," *Yogyakarta: Penerbit Andi*, 2006.
- [3] A. Haslam and P. Baines, *Tipografía. Función, forma y diseño*, 2005.

Técnicas de animación generada por ordenador

ÍNDICE

I. Fundamentos	2
I-A. Animación en tiempo real vs frame a frame	2
I-B. Animación bajo nivel vs alto nivel . . .	2
I-C. Aplicaciones	3
II. Animación tradicional	3
III. Animación asistida por ordenador	4
III-A. Edición de cuadros	4
III-B. Interpolación	5
III-C. Movimiento a lo largo de trayectorias .	6
IV. Animación generada por ordenador	6
IV-A. Animación basada en la forma	7
IV-B. Animación paramétrica de cuadros clave	7
IV-C. Animación capturada	7
IV-D. Animación procedural	8
V. Control automático de animación	8
V-A. Cinemática inversa y restricciones posicionales	8
V-B. Control de movimiento empleando dinámica	10
V-C. Considerar el impacto del entorno . . .	10
V-D. Planificación de tareas	10
V-E. Comportamiento asociado a los actores	10
VI. Conclusiones	11
Referencias	12

El término **animación por computadora** se refiere al conjunto de técnicas que emplean el ordenador para la generación de escenas con las que se produce sensación de movimiento. **Animación** se refiere al proceso de generación de imágenes donde cada imagen es una alteración de la anterior. La presentación de estas imágenes a una velocidad suficiente produce la sensación de movimiento.

La animación por computadora es un recurso de gran utilidad porque consigue efectos visuales a los que no llegan otros medios. Mediante animación por ordenador podemos sintetizar secuencias animadas difíciles de encontrar en la realidad o simplemente imposibles.

La animación por ordenador tiene sus orígenes en las técnicas de apoyo que necesitaban los dibujantes de animación tradicional para realizar su trabajo más rápido. Las primeras películas de dibujos animados se remontan a principios del siglo XX. En 1932 Walt Disney consigue el premio Óscar al mejor corto de animación. En 1974 una película generada por ordenador gana el festival de Cannes (*Hanger/La Faim* de Peter Foldes). Hoy en día el uso de animación generada por ordenador es masivo en la industria cinematográfica para crear efectos especiales.

Las animaciones se han convertido en un elemento a veces imprescindible en determinados ámbitos. Podríamos pensar que las animaciones son un accesorio en lo que hace referencia a la representación visual de escenas, sin embargo, debemos tener en cuenta que existen procesos físicos a los que sólo se puede acceder con animación por ordenador: simulación de accidentes o de reacciones físicas. Además, la animación aporta información importante no siempre es fácil representarla con iconos visuales. Por ejemplo, las escenas de las predicciones meteorológicas, las animaciones ofrecen información de la dinámica de los sistemas.

En este tema primero presentamos los fundamentos en los que se apoya la generación de animaciones. Después, el tema se desarrolla en cuatro bloques principales: las técnicas básicas de animación tradicional, la animación asistida por ordenador, la animación generada por ordenador, y la animación controlada automáticamente.

I. FUNDAMENTOS

La animación por ordenador se basa en los mismos principios que el cine o el vídeo: proyectar una secuencia de imágenes a una tasa o frecuencia determinada produciendo con ello la sensación de continuo. Cada una de estas imágenes se denomina cuadro o fotograma.

Los **cuadros, marcos o frames** se graban en fotogramas de una película y se proyectan a una velocidad alta. Para que no se produzca parpadeo la sucesión de imágenes debe hacerse lo suficientemente rápido. En el cine, por ejemplo, se emiten fotogramas a una tasa de 24 imágenes por segundo.

La figura 1 muestra ocho fotogramas correspondientes a una secuencia animada. La proyección de estos fotogramas a velocidad suficiente provoca la sensación de movimiento.

En animación tradicional se crea una ilusión de movimiento mediante la visualización de una secuencia de dibujos individuales. Esta ilusión se produce gracias a una característica de nuestro sistema de visión que se llama **persistencia**. Gracias a la persistencia, las imágenes quedan registradas en nuestra retina por un breve periodo de tiempo. Si llega otra imagen antes de que se desvanezca la anterior, el cerebro las funde y las percibe como un continuo.

Anatómicamente el ojo puede compararse a una cámara fotográfica, en la que la retina realiza las funciones de película fotosensible. Cuando la retina es iluminada por un breve haz luminoso, la imagen llega a los centros nerviosos con un ligero retardo debido al fenómeno de persistencia luminosa de unos 0.25 segundos. Este hecho nos permite percibir las secuencias de fotogramas estáticos como un continuo.

Ya en el siglo XVII, Isaac Newton demostró este hecho utilizando un disco circular dividido en ocho sectores. Cada sector estaba pintado con uno de los colores del arco iris y al hacer girar el disco a una velocidad adecuada se volvía completamente blanco. Los estímulos visuales sucesivos se perciben como estímulos diferentes dependiendo de la frecuencia con las que se producen. La sensación de continuidad es mayor con frecuencias mayores. Por debajo de un cierto umbral, científicamente identificado como el **umbral crítico de fusión**, los estímulos son captados de forma individual percibiéndose como una secuencia de imágenes estáticas. Por encima de este umbral, la secuencia de imágenes se percibe como un único estímulo produciéndose la sensación de movimiento. El valor del umbral depende de la intensidad de cada estímulo, de su duración y del observador. Para una imagen televisada, si los estímulos llegan a intervalos de 0.2 segundos las imágenes parecen sucederse y si, en cambio, están separados por menos de 0.06 segundos dan la impresión de un movimiento continuo. En cine se utilizan 24 cuadros por segundo.

I-A. Animación en tiempo real vs frame a frame

La animación puede ser generada en tiempo real o en modo frame a frame. En tiempo real implica que las imágenes son generadas a una velocidad lo suficientemente rápida como para generar persistencia. Para propósito general, la tasa capaz de producir esta percepción es habitualmente de 24 imágenes por segundo. La tasa real depende de los tipos de imágenes que

Figura 1: Cuadros de la animación de un personaje bailando[1].



estamos visualizando y de las condiciones de visualización. Se habla de animación en tiempo real cuando se generan escenas a frecuencias superiores a 5 frames por segundo.

Cuando las escenas que componen la animación no pueden ser generadas con la suficiente velocidad, se generan las frames individuales y se graban en algún medio persistente, de manera que una vez generadas todas las frames que componen la animación, se reproduzcan a las tasas necesarias.

El uso de una u otra técnica, depende del tiempo necesario para calcular el movimiento (**control del movimiento**) y para hacer generar de la escena (**rendering**). Debemos entender que son cosas completamente distintas. Una simulación de un proceso meteorológico suele implicar la resolución de complejos sistemas de ecuaciones que pueden tardar minutos. Sin embargo, la visualización de los datos resultantes puede ser inmediata. Sin embargo, una animación simple como el desplazamiento de un objeto en una escena compuesta de multitud de formas complejas, no implica ninguna dificultad de cálculo de movimientos, pero el render de las escenas puede tardar segundos dependiendo del número de polígonos del objeto.

Otra distinción que debe tenerse en cuenta es la que hace referencia a la **tasa de reproducción** y a la **tasa de actualización**. La tasa de reproducción es la tasa a la cual se ven las frames en el dispositivo de visualización. La tasa de actualización es la tasa a la que se calcula el movimiento. En dibujos animados, por ejemplo, la tasa de actualización puede llegar a ser de 8 frames por segundo a pesar de que la tasa de reproducción sea de 24 frames por segundo en el estándar del cine.

I-B. Animación bajo nivel vs alto nivel

En este documento distinguiremos, a la hora de especificar movimiento, entre técnicas de bajo nivel (que ayudan al animador a especificar movimiento de forma precisa) y modelos de alto nivel (técnicas utilizadas para definir el movimiento).

Las **técnicas de bajo nivel** son un conjunto de técnicas como las técnicas de interpolación, que ayudan al animador a

completar los detalles del movimiento una vez que se tienen suficientes detalles de cómo está definido. Cuando se utilizan este tipo de técnicas, el animador tiene una idea clara de lo que quiere.

Las **técnicas de alto nivel** son típicamente algoritmos o modelos empleados para generar un movimiento empleando un conjunto de reglas o restricciones. El animador establece las reglas del modelo o unos parámetros iniciales, y el movimiento se controla por el algoritmo del modelo.

Esta distinción inicial, tiene objetivos puramente pedagógicos, ya que lo que nos encontramos en la realidad es un abanico de técnicas y procedimientos que se extienden entre ambos métodos. Todas las técnicas requieren un esfuerzo por parte del animador y otro esfuerzo por parte del ordenador. En las técnicas de bajo nivel, el animador tiene más trabajo que en las técnicas de alto nivel. Por otro lado, el uso de técnicas de alto nivel implica un coste computacional mayor.

Otro factor que ayuda a separar ambas técnicas es el nivel de abstracción al que trabaja el animador. En las primeras, el animador trabaja a un nivel muy bajo de representación. De hecho, puede que para realizar una animación tenga que modificar los valores de los pixels de las imágenes o la posición de los vértices. En las segundas, el animador se olvida de los niveles inferiores. Una orden para generar animación empleando estas últimas técnicas podría ser: "haz que el perro se mueva". Un nivel de abstracción alto, libera al animador de tener que considerar determinados detalles. Un nivel de abstracción bajo, permite al animador ser muy preciso especificando exactamente lo que quiere ver en cada momento. El objetivo de las herramientas de animación es diseñar aplicaciones que permitan al usuario trabajar sobre los distintos niveles de abstracción.

I-C. Aplicaciones

Existen gran variedad de usos de la animación por computadora. La animación por computadora ha encontrado su sitio en campos de aplicación que a priori parecen muy distantes. A continuación describimos una lista no exhaustiva de aplicaciones:

Educación La posibilidad de mover objetos y de resaltar conceptos mediante el uso de animaciones permite también incrementar las alternativas para explicar con claridad conceptos complejos. Existe una gran variedad de aplicaciones educativas que permiten interactuar con mapas y recursos que mejoran las explicaciones de los libros incluyendo dinamismo.

Entretenimiento Todos los juegos de ordenador utilizan animación por ordenador. La gama va desde juegos donde un simple *sprite* o imagen se mueve por la pantalla, hasta los más modernos donde se realizan renders fotorrealistas en tiempo real.

Cine, vídeo y televisión El uso de ordenadores en cine es masivo. Los efectos especiales que se consiguen con la animación basada en ordenadores son imposibles de conseguir con métodos tradicionales. Muchas presentaciones de programas televisivos, se hacen con escenarios sintéticos. Uno de los usos más extendidos de la animación

por computadora es la publicidad en televisión. Muchos de los objetos que aparecen en los anuncios televisivos son, en realidad, modelos 3D animados.

Simulaciones Los simuladores de vuelo y otros simuladores como los de carros de combate son empleados con frecuencia en los entrenamientos de militares. El ahorro que se consigue empleando sistemas que se apoyan en modelos 3D es considerable.

Visualización científica Las técnicas de visualización realizan animaciones a partir de una serie de datos obtenidos de los experimentos. La cantidad de datos suele ser tan grande, que, si no se visualizan en un ordenador, pueden ser intratables. En ocasiones los científicos crean modelos realistas de procesos físicos y emplean la simulación para obtener datos y las técnicas de visualización para comprobar resultados. La visualización de datos es en sí una técnica que evoluciona paralela a la animación por computador.

Las animaciones más complejas requieren el uso de **métodos de disciplinas muy variadas**: robótica, dinámica analítica, dinámica de fluidos, biología, sistemas caóticos, psicología e inteligencia artificial. Para conseguir generar sensación de realismo es necesario considerar factores relativos a todos estos ámbitos.

II. ANIMACIÓN TRADICIONAL

Antes de la llegada de los ordenadores, todas las animaciones se hacían a mano. Cada uno de los cuadros de la animación debía ser dibujada a mano. Si consideramos 24 cuadros por cada segundo de la animación, podemos imaginar la inmensa cantidad de trabajo que suponía realizar una película de una hora de duración.

Ejercicio: La película de Blancanieves (1939) de Walt Disney tiene una duración de 83 minutos. ¿Cuántos fotogramas debieron dibujarse?

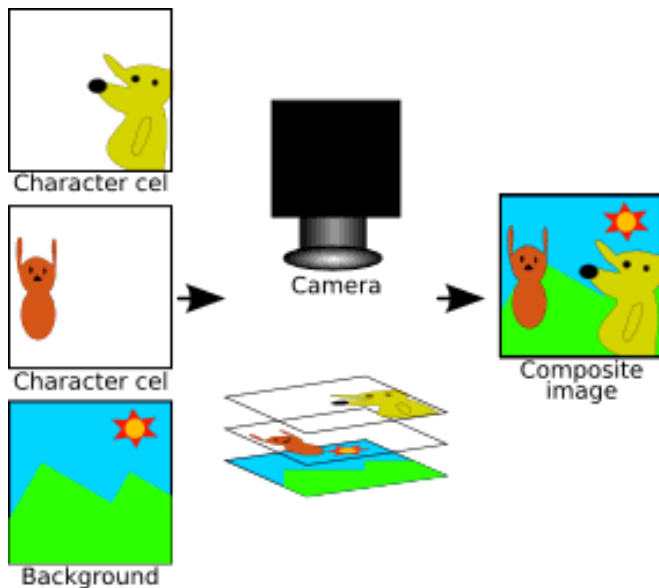
Si suponemos una tasa de 24 fotogramas por segundo, serían necesarios 83·60·24 fotogramas, esto es 119.520 fotogramas pintados y coloreados a mano.

Para reducir este trabajo, se desarrollaron algunas técnicas que describimos a continuación:

Animación por cuadros clave Debido a la inmensa cantidad de trabajo que conlleva hacer una animación, el dibujo y la pintura de los diferentes cuadros se realiza por un equipo de personas. Después de crear el guion de la película, un dibujante de mayor categoría, dibuja los cuadros principales de la animación. Estos cuadros principales son, típicamente, los cuadros en los que ocurren los mayores cambios. Son los puntos clave de la animación y por esto se denominan **cuadros clave**.

Una vez dibujados, un equipo de dibujantes de menor categoría se dedica a realizar los **cuadros intermedios**, tantos como exija la velocidad de la escena. Este trabajo de crear los cuadros intermedios a partir de los cuadros clave, se denomina **in-betweening**. De este modo, el tiempo de producción se distribuye, permitiendo reducir considerablemente los tiempos de desarrollo.

Figura 2: Ejemplo de la técnica de cel animation. Desplazando los acetatos se consigue generar la ilusión de desplazamiento (imagen cortesía de G. Albright CC BY-SA).

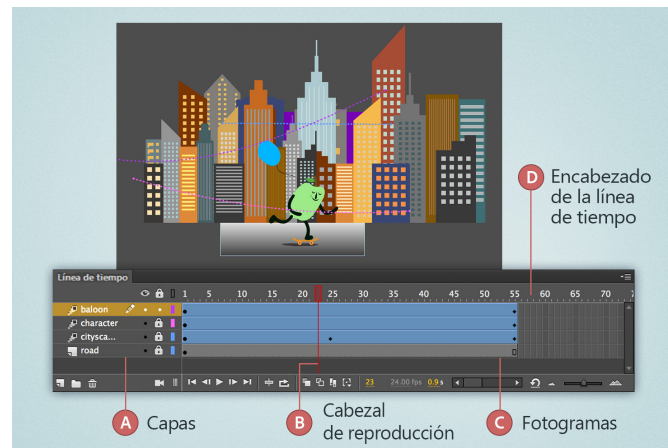


Animación por capas Otra técnica que reduce considerablemente el tiempo de trabajo es la técnica llamada *cel animation*. Cuando creamos una animación empleando este método, cada personaje se dibuja en un papel transparente separado del resto. El fondo de la animación también se dibuja en un papel separado. Son papeles de acetato transparente (u otro material de propiedades similares) que se superponen los unos a los otros. Cuando llega la hora de realizar la animación, los distintos personajes se superponen sobre el papel que contiene el fondo. Este método supone un ahorro considerable de tiempo a los artistas que no tienen que dibujar todos los personajes que participan en cada uno de los cuadros. En ocasiones se emplean las hojas transparentes para dibujar partes concretas de los personajes. Así por ejemplo si un personaje habla, se emplean acetatos diferentes para la boca, ojos u otras partes que se pueden mover mientras el personaje habla; el resto del personaje permanece estático.

Rotoscopía Empleamos rotoscopía cuando se copian imágenes previamente grabadas en un vídeo para realizar los fotogramas de la animación. Si queremos animar el salto de una rana, se graba en un vídeo a una rana saltando para aislar aquellas partes de la secuencia que resulte difíciles de visualizar al animador. Un dibujante copia las partes del movimiento más relevantes.

Estos procesos supusieron el paso hacia la **producción industrial** de los dibujos animados. Hablamos de producción industrial porque permiten reutilizar, dividir tareas, especializar trabajos y solapar actividades en el tiempo. Los computadores entran a formar parte del proceso de producción de animación tradicional para reducir el tiempo de trabajo en lo que se llama animación asistida por computador, a la que dedicamos el siguiente capítulo.

Figura 3: Timeline en un programa de animación (Imagen propiedad de Adobe[2])



III. ANIMACIÓN ASISTIDA POR ORDENADOR

Existen dos categorías principales de animación por ordenador: animación asistida por ordenador y animación generada por ordenador. La **animación asistida por ordenador** se refiere al empleo de ordenadores en el proceso de producción de animación tradicional: típicamente dibujos animados. En la **animación generada por ordenador**, por contra, el ordenador no es sólo una herramienta de apoyo al animador sino que se convierte en un elemento fundamental para la generación de animaciones: permite hacer secuencias animadas que es imposible generar manualmente.

Cuando el ordenador puede servir de apoyo en el trabajo de los animadores que emplean métodos tradicionales, el objetivo es liberar a los dibujantes de las actividades más tediosas. La incorporación de los ordenadores como herramientas encaminadas a sustituir algunos de los pasos tradicionales de la animación convencional, originó el nombre de animación asistida por ordenador. A continuación presentamos algunos de los procesos en los que el ordenador es más útil.

III-A. Edición de cuadros

En primer lugar, los ordenadores sirven de apoyo a la hora de **generar los cuadros clave**. El dibujante hace un boceto del dibujo, lo escanea, y con ayuda de un programa de diseño gráfico termina el trabajo. También se ofrecen **librerías de personajes** o de partes de personajes para componer nuevos actores.

La técnica de **animación por capas** se puede emplear en la mayoría de los programas informáticos de diseño gráfico y de tratamiento de imagen. Estos programas utilizan capas para realizar los dibujos o para procesar las imágenes. Esta técnica de trabajo por capas permite realizar un trabajo equivalente al realizado en las técnicas de cel animation.

La figura 3 muestra el interfaz del programa Animate de Adobe, especializado en la creación de animaciones 2D. El programa no sólo realiza las posiciones intermedias de los **sprites** (elementos gráficos que se desplazan o transforman

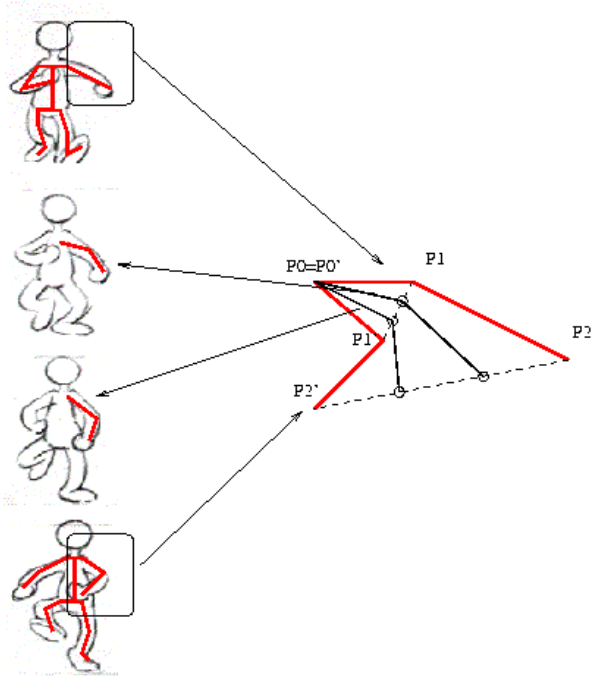


Figura 4: Interpolación. Entre los puntos \bar{P} y \bar{P}' se generan valores intermedios muestreando la recta que los une.

a lo largo del tiempo en las animaciones) sino que también permite diseñarlos.

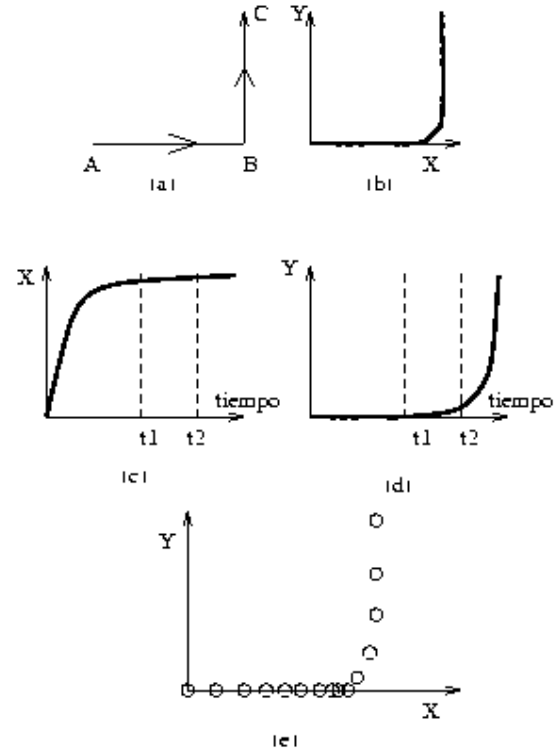
Los editores de vídeo permiten guardar los fotogramas de los vídeos en ficheros de imagen estática. Tras un proceso de vectorización, se consigue obtener dibujos de los cuadros a partir de las imágenes de los fotogramas de forma casi automática. Sería una aportación de los ordenadores al apoyo en técnicas de rotoscopia.

III-B. Interpolación

Existen técnicas que realizan el **in-betweening** de forma automática. Siempre que sea posible, el ordenador debe calcular de forma automática los cuadros intermedios entre los cuadros clave. Utilizando algoritmos que se apoyan en técnicas de interpolación, se consiguen transiciones suavizadas en la mayoría de los casos. Desafortunadamente, estas técnicas sólo se pueden realizar cuando existe una correspondencia origen destino entre los trazos de los dibujos que componen los cuadros clave. Si esto no es así (por ejemplo, un personaje que se da la vuelta) el trabajo no se puede realizar automáticamente (en el caso 2D).

Para generar los cuadros intermedios entre cuadros clave se emplean técnicas de interpolación. Calculamos los cuadros intermedios promediando información de los cuadros clave. Cada cuadro intermedio difiere ligeramente del anterior desde el cuadro clave inicial al final. La información que se interpola no tiene por qué ser necesariamente posicional, sino que puede ser de varios tipos: la textura, el tamaño, etc. . .

Figura 5: Movimiento a lo largo de trayectoria. (a) Un objeto se mueve desde A hasta C pasando por B; (b) la trayectoria no es una trayectoria recta sino que en el codo es ligeramente suavizada. En (e) vemos la trayectoria seguida en (a) enriquecida por la posición de las claves; entre cada clave la distancia temporal es la misma. (c) y (d) muestran la evolución temporal de X e Y tal y como se expresa en (e).



La figura 3 muestra el **timeline** de un programa de animación asistida por ordenador. El timeline permite dibujar los diferentes elementos que componen la animación en capas que se superponen las unas sobre las otras, simulando los tradicionales *cels*. Además, se pueden distinguir fotogramas clave y el programa realiza las transiciones entre claves (*in-betweening*) realizando interpolación.

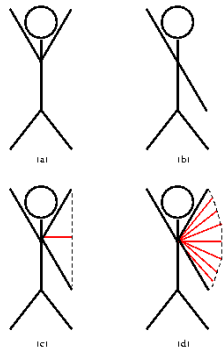
Sea \bar{P} una propiedad cualquiera de un objeto dado (por ejemplo su posición). Llamamos \bar{P}_o al valor de esa propiedad en el cuadro clave de origen y \bar{P}_f al valor de la misma en el cuadro clave final. Así, si queremos generar cuadros intermedios aplicando interpolación lineal, debemos aplicar la fórmula genérica:

$$\bar{P}_\lambda = \bar{P}_o + (\bar{P}_f - \bar{P}_o)\lambda \quad \lambda \in [0, 1] \quad (1)$$

Cuando $\lambda = 0$ estamos en el punto \bar{P}_o . Cuando $\lambda = 1$ estamos en el punto \bar{P}_f . Dando valores a λ entre 0 y 1 obtenemos valores intermedios entre \bar{P}_o y \bar{P}_f .

Para generar n cuadros entre \bar{P}_o y \bar{P}_f , muestreamos λ tomando los valores: $\lambda = \{0, \frac{1}{n+1}, \frac{2}{n+1}, \dots, \frac{n}{n+1}, 1\}$ El número de cuadros intermedios dependerá del sistema de visualización que empleemos y de la duración que queramos que dure el

Figura 6: Movimiento a lo largo de trayectoria. El brazo baja siguiendo una trayectoria curva y utilizando un movimiento uniformemente acelerado. Para ello, el extremo del brazo sigue una curva-P específica.



movimiento. Así, si quiero que la animación de la figura 4 tarde en realizarse medio segundo en un sistema que emplee 30 cuadros por segundo, deberé generar 15 cuadros intermedios.

Ejercicio: Queremos hacer una animación en la que el segmento AB definido por los puntos $A=(0,4,5)$ y $B=(5,8,9)$ se desplace a los puntos $A'=(3,5,7)$ y $B'=(4,6,8)$. Si queremos hacer la animación empleando 10 cuadros en total, escriba los valores de los puntos intermedios.

Si utilizamos la fórmula vista anteriormente, los puntos entre A y A' sería $P_\lambda = A + \lambda \cdot A'$ con $\lambda = \{0, \frac{1}{9}, \frac{1}{9}, \frac{2}{9}, \dots, \frac{8}{9}, 1\}$. Para los puntos entre B y B' se hace el cálculo análogo.

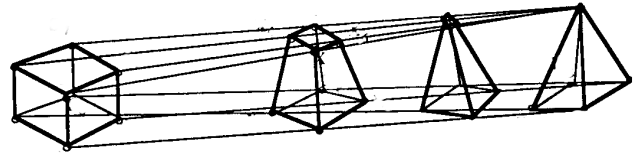
Las interpolaciones no tienen por qué ser necesariamente lineales. Podemos especificar la trayectoria empleando curvas (por ejemplo un *spline*). Si la curva está definida de forma paramétrica $Q(t)$ con $t \in [0, 1]$, y queremos generar n valores intermedios, dividiremos el intervalo $[0, 1]$ en n intervalos iguales.

III-C. Movimiento a lo largo de trayectorias

Para representar el movimiento a lo largo de una trayectoria, la interpolación lineal no es adecuada primero porque los movimientos pueden no hacerse a lo largo de trayectos rectos. Además, porque los cuadros intermedios están separados por un intervalo uniforme a lo largo del tiempo. Si además se espacian de forma uniforme en el espacio tenemos velocidad constante. Ningún objeto se mueve a velocidad uniforme cuando parte del reposo o cuando termina parándose. Siempre hay una aceleración o deceleración a lo largo del movimiento.

Las curvas P son un artificio que permite especificar trayectorias introduciendo información espacial y temporal en la misma gráfica. Las líneas denominadas **P-curvas** establecen la trayectoria que siguen en la escena los distintos elementos de la animación y también la velocidad. Sobre la curva aparecen símbolos superpuestos, típicamente círculos. Estos símbolos representan los cuadros de la animación que están equiespaciados en el tiempo. A mayor distancia espacial entre

Figura 7: Morphing: el cubo cambia su forma para convertirse en una pirámide.



símbolos mayor velocidad porque la distancia temporal es la misma (ver figura 5).

Por ser dibujos 2D que evolucionan a lo largo de una trayectoria, esta técnica se conoce como animación 2D1/2. Aunque están pensadas para representar trayectorias, también se pueden utilizar para hacer in-betweening como se muestra en la figura 6

IV. ANIMACIÓN GENERADA POR ORDENADOR

La animación generada por ordenador se diferencia de las técnicas animación asistida por ordenador en que el ordenador deja de ser únicamente un soporte encaminado a sustituir al hombre en determinadas tareas y se convierte en un elemento clave para el modelado y la animación de mundos tridimensionales. Ahora, con el ordenador realizamos animaciones que no pueden hacerse manualmente.

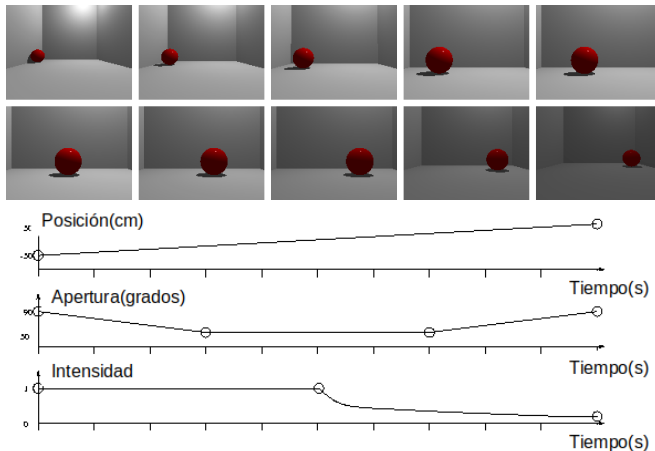
En esta sección sólo distinguimos los tipos de animación generada por computador en función de las características que evolucionan y de cómo se especifica la evolución. Cuando queremos generar animaciones más complejas, hemos de recurrir al empleo de técnicas más sofisticadas que se agrupan bajo el nombre de control automático de la animación que describimos en la siguiente sección.

En animación generada por computador se generan las escenas con los métodos de rendering conocidos de informática gráfica y se introducen movimientos en dichas escenas. En animación generada por ordenador o animación 3D primero se especifican las escenas, dando valores a las características de las entidades que las componen y después se especifica la evolución temporal de los valores de dichas características. Para los instantes de tiempo en los que se deben dibujar los cuadros, se calculan los valores de las características y se hace el rendering de la escena con dichos valores. Cada rendering es un fotograma de la animación final.

En función de qué características de qué entidades se modifiquen, distinguimos los siguientes tipos de animación:

- **Objetos**
 - Posición, orientación: los objetos cambian su posición a lo largo del tiempo con traslaciones o rotaciones.
 - Forma: pueden cambiar el tamaño o alguna característica al objeto como puede ser la altura de una pirámide o el número de lados de una pirámide.
 - Color, transparencia; una animación puede implicar también variaciones en el tiempo del color, transparencia o textura de la superficie de los objetos.
- **Cámara:**

Figura 8: Animación paramétrica de cuadros clave.



- **Posición:** es posible generar animaciones por computadora al cambiar los parámetros de la cámara, como su posición al hacer movimientos tipo *pan*.
- **Punto hacia el que apunta:** cambiamos la orientación de la cámara cuando el objeto focalizado cambia de posición.
- **Apertura del angular:** cambiando la longitud focal hacemos zoom sobre los objetos.
- **Fuentes de iluminación:**
 - **Posición:** las fuentes de iluminación pueden cambiar de posición para representar por ejemplo el movimiento del sol a lo largo del día o el de los focos de un vehículo al desplazarse.
 - **Intensidad, color:** por ejemplo, cuando un foco se desvanece.

En función de cómo se especifique la evolución de las características que cambian a lo largo del tiempo distinguimos entre animación basada en la forma de los objetos clave, animación paramétrica de cuadros clave, animación capturada y animación procedural. A estos tipos de animación dedicamos los siguientes apartados.

IV-A. Animación basada en la forma

La metamorfosis de imagen o más comúnmente *morphing* o *shape animation* es un método para generar animaciones en el que un objeto 2D o 3D se convierte en otro diferente en sucesivos pasos. Puede partirse de objetos 3D o 2D para generar la animación, y también de imágenes digitales convenientemente tratadas.

Es necesario establecer una **correspondencia entre puntos** de origen y de destino, y calcular los puntos intermedios. Los puntos intermedios se calculan aplicando técnicas de interpolación como las vistas en las secciones previas. La figura 7 muestra el proceso de generación de cuadros basado en la forma. La presentación de los cuadros en intervalos de tiempo breves, genera la ilusión de movimiento.

Elegir una buena correspondencia entre puntos es fundamental para obtener un buen resultado. Cuando las figuras son

muy diferentes, este método da unos resultados que pueden ser imprevisibles. Por ejemplo, si el número de puntos en la figura original y en la figura destino no son los mismos, es difícil determinar la mejor correspondencia entre puntos.

Cuando se aplica a imágenes digitales, la interpolación se hace entre pixels. Cuando hay muchas diferencias entre imágenes empleamos otra técnica adicional denominada **warping**. Mediante el warping, una figura común envuelve la zona de interés de la imagen de origen y de destino y la correspondencia entre puntos se hace con las figuras envolventes. Los pixels de las imágenes cambian ajustando sus colores con respecto a las figuras envolventes.

Se localizan las zonas en las imágenes u objetos A y B entre las que queremos que se produzca la transformación. Estas zonas definen una correspondencia entre pixels de su interior. Se realiza una interpolación de la posición de los puntos y del color de los pixels color. Un ejemplo de morphing aplicado a imágenes puede ser el cambio de una cara en otra o de un gesto en otro de la misma cara. El warping nos permite elegir las zonas de la cara que más nos interese.

IV-B. Animación paramétrica de cuadros clave

Para poder controlar la evolución de los parámetros a lo largo del tiempo de forma precisa, se emplean los **gráficos de evolución temporal**, en los que se despliega una gráfica que muestra los valores de cada parámetro en cada instante de tiempo. En estas gráficas se especifican los valores que tiene un determinado parámetro en los cuadros clave; después, el ordenador estima el valor que se corresponde en los cuadros intermedios utilizando las curvas.

La parte superior de la figura 8 muestra algunos cuadros clave de una animación simple en la que una esfera avanza de izquierda a derecha a la vez que la cámara hace zoom y la luz se desvanece. En la parte de abajo vemos la evolución temporal de los parámetros que intervienen en la animación: posición en el eje x, apertura del angular e intensidad de la fuente de luz. En cada una de las gráficas se especifican las claves y con ellas la forma de la curva antes y después del cuadro clave.

IV-C. Animación capturada

Podemos generar las trayectorias de los objetos en movimiento haciendo un seguimiento de acciones reales. En las técnicas de captura de movimiento se conecta algún tipo de indicador o dispositivo en los puntos clave del cuerpo de una persona o del objeto que se quiere rastrear. Rastreando las posiciones de los indicadores, podemos obtener las posiciones de los puntos clave de una persona. Por ejemplo, se pueden conectar pequeños puntos de luz y las posiciones de esos puntos se graban desde distintas posiciones para obtener la posición 3D de cada punto clave. El actor puede ver sus propios movimientos y obtener un *feedback* para corregirlos en tiempo real. El movimiento puede ser capturado para aplicarlo más tarde en el movimiento de otros personajes.

La figura 9 muestra un dispositivo electromecánico de captura de movimiento. Los sensores recogen la posición de

Figura 9: Captura de movimiento: empleando marcadores ópticos, se capturan las posiciones de las articulaciones durante el movimiento de un actor. Esas posiciones quedan registradas y se emplean para articular después avatares sintéticos.

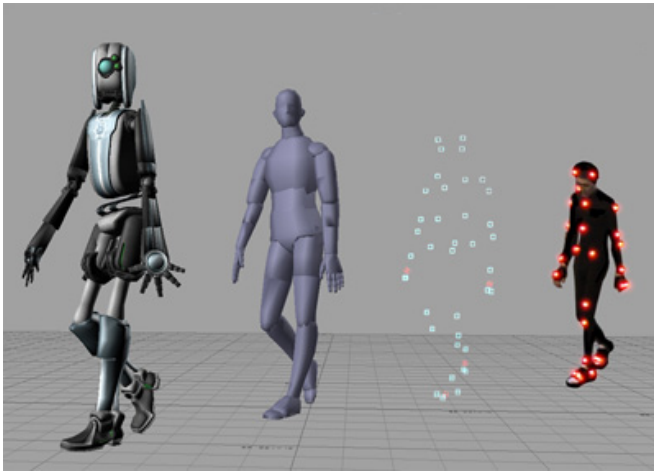
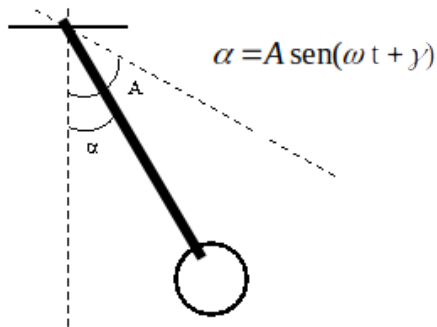


Figura 10: Ejemplo de modelo empleado para realizar animación procedural.



una serie de puntos clave. La información de posición de los puntos clave es utilizada para mover el avatar 3D.

Otro tipo de mecanismo es el guante empleado en dispositivos de realidad virtual. El movimiento se captura por unos sensores y se visualiza en tiempo real en el interior de una escena.

IV-D. Animación procedural

Movimiento procedural es un término amplio que recoge las técnicas que generan animación apoyándose en un determinado algoritmo.

Se establecen una serie de reglas que rigen el movimiento del sistema y se asigna un estado inicial para los objetos del sistema. En cada cuadro se calcula la composición de la escena en función de la composición del cuadro anterior y de las acciones especificadas en el procedimiento. Un **controlador** o programa o función software es la encargada de ir generando los valores intermedios necesarios para realizar la animación empleando el modelo.

La principal diferencia con el resto de métodos es que no es necesario definir cuadros clave. Con este tipo de animación se puede generar movimiento muy realista con relativamente poca aportación por parte del animador.

Como el aspecto de los cuadros sucesivos tiene su origen en los cuadros anteriores, este tipo de animación tiene el inconveniente del escaso control que se tiene de la misma.

V. CONTROL AUTOMÁTICO DE ANIMACIÓN

Cuando hablamos de control automático de la animación, las distintas entidades que componen la escena están gobernadas por una serie de métodos o programas que rigen su movimiento. En cada cuadro, los métodos intercambian información con otros procedimientos asociados a otras entidades. El movimiento se basa en módulos de inteligencia artificial denominados actores.

Para que un sistema de **actores** permita crear escenas que den sensación de realismo se han de resolver 5 problemas:

Paso 1: Cinemática inversa

Paso 2: Control de movimientos empleando dinámica

Paso 3: Considerar el impacto con el entorno

Paso 4: Planificación de tareas

Paso 5: Asociar un comportamiento a los actores

A continuación describimos cada uno de estos problemas.

V-A. Cinemática inversa y restricciones posicionales

La cinemática se encarga del estudio del movimiento independientemente de las fuerzas que lo provocan. Una descripción cinemática del movimiento tendría en cuenta las trayectorias lineales y angulares de los objetos a lo largo del tiempo. No tendría en cuenta, por ejemplo, el peso, la forma ni las fuerzas que se ejercen sobre ellos.

Es el modelo de especificación de movimiento utilizado en la mayoría de los sistemas de animación generada por ordenador. La cinemática es una técnica empleada para mover **figuras articuladas**, que son aquellos compuestos por una serie de elementos rígidos unidos entre sí por una serie de articulaciones. Ejemplos de figuras articuladas de este tipo son los sistemas mecánicos, y también los animales vertebrados. Objetos más sencillos que incluyan articulaciones como una caja con tapadera también entrarían dentro de la categoría de figuras articuladas. La figura 11 muestra tres ejemplos de figuras articuladas.

En todas ellas hay enlaces rígidos unidos por articulaciones. Se imponen una serie de **restricciones** y unas dependencias en el movimiento de los distintos elementos que tienen su origen en la estructura de la figura articulada. Las **dependencias** se establecen fijando una relación padre hijo, de manera que al mover un elemento padre, los hijos heredan el movimiento. Por ejemplo, en un brazo articulado, al mover la base el resto de los elementos se mueven heredando el movimiento. Las **restricciones** permiten indicar que un elemento no puede moverse de cualquier manera: puede tener limitados determinados grados de libertad o determinados tipos de movimiento. Por ejemplo, un cuello de un ser humano no puede girar 360 grados ni desplazarse con respecto al tronco al que está unido.

Figura 11: Ejemplos de estructuras articuladas. Arriba, la biela es movida por la rueda cuando ésta rota. En el medio los elementos del brazo articulado heredan el movimiento de los elementos con los que están unidos. Abajo un avatar de una humanoide, cuyo esqueleto determina los movimientos articulados (la imagen inferior es cortesía de Autodesk).

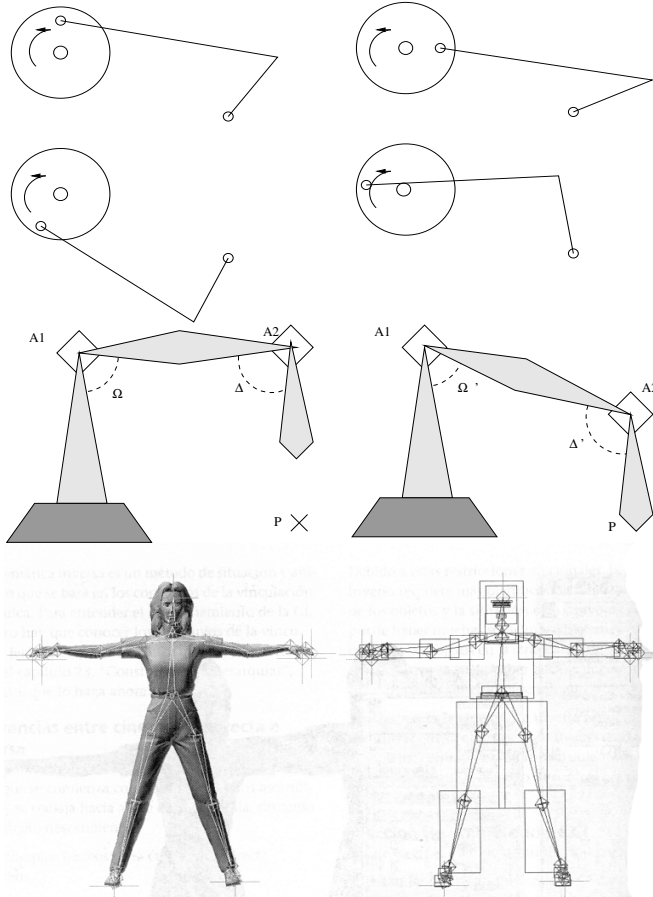


Figura 12: Los sistemas articulados son ejemplos típicos en los que se puede emplear cinemática directa o cinemática inversa para realizar la animación. Empleando cinemática inversa, el movimiento se especifica posicionando el elemento terminal de la cadena cinemática, en este caso el talón. Empleando cinemática directa, este mismo movimiento se realiza especificando primero la rotación en la cadera, y luego en la rodilla (Imagen cortesía de Autodesk Softimage).

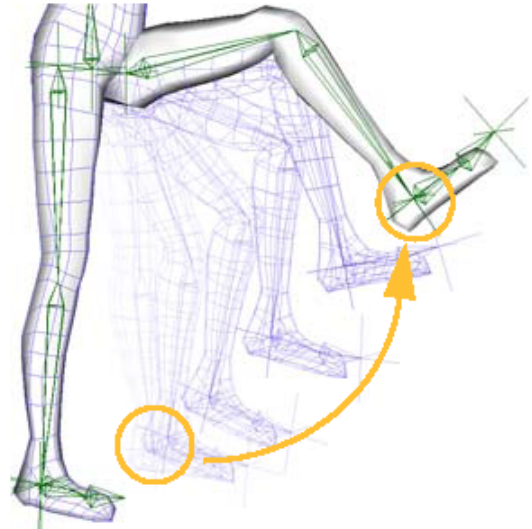
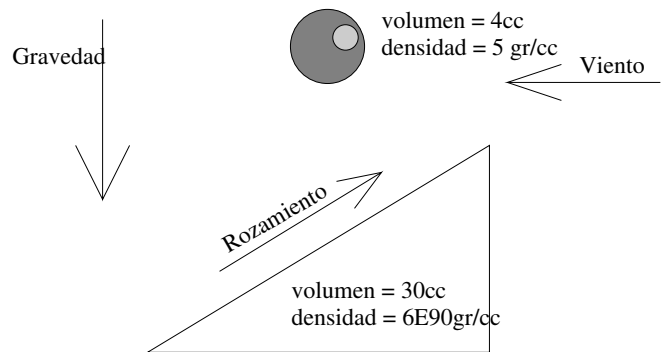


Figura 13: Ejemplo de configuración de una escena para hacer una animación basada en la dinámica



La forma natural de mover una figura articulada es desplazando el elemento final de la estructura de manera que éste arrastre al resto de elementos en el desplazamiento. Éste es el problema que intenta resolver la **cinemática inversa**: para una posición final del elemento extremo de una figura articulada calcular las posiciones del resto de elementos. En una definición de un movimiento de **cinemática directa**, a partir del movimiento de los elementos padre que tenga en cuenta sus restricciones se recalcula la posición de los elementos hijo. Por contra en una definición de **cinemática inversa** se cambia la posición de los elementos hijo y se calculan las posiciones de los elementos padre para cumplir las restricciones.

En la figura 12, se plantea el problema de llevar el extremo del esqueleto articulado hasta el punto de destino. Empleando cinemática directa, iríamos asignando valores nuevos a los ángulos de rotación de las articulaciones. Comenzando por los elementos más lejanos al extremo y teniendo en cuenta que al desplazarlos el movimiento se transmite a través de toda la estructura, iríamos acercando el punto extremo al

punto de destino. En una aproximación de cinemática inversa, simplemente especificamos el punto de destino, y el sistema evoluciona de forma automática calculando los valores de las articulaciones para alcanzar dicho elemento final.

Definir un problema empleando cinemática inversa tiene la ventaja de que es mucho más fácil de definir para el animador. Por contra, al emplear cinemática inversa, se generan problemas no deterministas, ya que las soluciones pueden ser múltiples. Esto ocurre cuando el extremo puede llegar al destino, optando por varios caminos. Además, los cálculos a realizar para resolver el problema de la cinemática inversa son complicados y suelen requerir abundante tiempo de cálculo computacional.

V-B. Control de movimiento empleando dinámica

La dinámica es la rama de la física que describe cómo se mueven los objetos teniendo en cuenta las fuerzas que los provocan. La animación por ordenador basada en estas técnicas emplea propiedades físicas de los objetos (densidad, volumen, coeficiente de rozamiento) y del entorno (resistencia del aire, gravedad) para calcular el camino animado que recorrerá el objeto. Las reglas que rigen el movimiento son las leyes de Newton y las de la mecánica de fluidos.

Estas técnicas se emplean para simular fenómenos físicos como: gravedad, momento, colisiones, fricción, flujos de fluidos (turbulencia), solidez flexibilidad, elasticidad, fracturas . . .

Una definición completa de una animación empleando dinámica, aunque resultaría muy realista, suele implicar un alto coste computacional. Primero porque el tiempo necesario para simular todos los condicionantes físicos es mucho y segundo porque en ocasiones interesa definir animaciones que no cumplan estas leyes: por ejemplo, una caja que sube por la pared.

V-C. Considerar el impacto del entorno

El movimiento de un objeto tiene un impacto con el entorno y viceversa. Empleando una buena planificación de trayectorias podrían evitarse colisiones, sin embargo en muchas aplicaciones las colisiones son importantes. Si pensamos por ejemplo en el ámbito de los videojuegos, detectar colisiones entre sprites es fundamental por ejemplo para saber cuándo un personaje es golpeado por otro.

Detectar colisiones tiene un alto coste computacional porque puede haber muchos elementos en la escena que colisionen entre sí, y hay que realizar tests de colisiones por cada par de elementos. Además, las geometrías de estos elementos pueden ser muy complejas, y para detectar las colisiones puede ser necesario analizar cada plano.

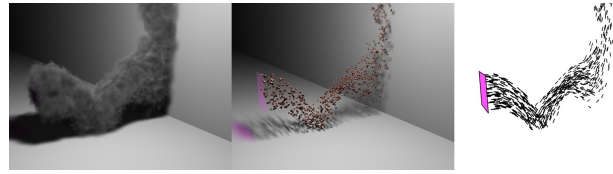
Para evitar el problema de la complejidad de las geometrías de los objetos que pueden colisionar, se suelen utilizar **envolventes**, que son geometrías simplificadas a los objetos que se emplean para hacer el cálculo de las colisiones. Es habitual emplear esferas que envuelven los objetos porque saber si dos esferas se tocan entre sí es un cálculo sencillo.

Para que las colisiones sean realistas, también hay que hacer que las reacciones al choque sean coherentes. Si los elementos que colisionan son sólidos rígidos, la reacción será diferente a la que tienen objetos blandos. Para los objetos no rígidos, se emplean deformaciones que afectan directamente a los vértices de los objetos.

V-D. Planificación de tareas

La planificación de tareas es un ámbito más cercano a la inteligencia artificial que a la animación por ordenador. El objetivo es descomponer la especificación de una tarea en acciones elementales que pueda realizar el actor. Ejemplo: una figura humana debe reaccionar ante la orden “responde al teléfono”. La tarea debe descomponerse en una serie de acciones tales como: Calcula la posición del teléfono, Busca un camino hacia el teléfono, Avanzar, Coger el teléfono y Decir “hola”

Figura 14: Ejemplo de animación empleando un sistema de partículas (cortesía de T. Wusa CC BY-SA).



Un ejemplo de animaciones en las que se necesita planificación son los **sistemas de partículas**. Un sistema de partículas es una gran colección de objetos, que puestos juntos, representan objetos difusos. Estas partículas, se modelan con una vida finita, de manera que en una animación pueden emplearse cientos de miles de partículas, pero sólo unos cientos estarán activas en cada momento. Para evitar el orden excesivo en el sistema, se introducen factores aleatorios para modelar y producir partículas.

Para calcular cada uno de los cuadros se siguen los siguientes pasos:

- Se generan las partículas que han nacido durante el cuadro.
- Se asignan atributos a cada nueva partícula.
- Cualquier partícula que haya excedido su tiempo de vida es eliminada.
- Para las partículas que quedan, se modifican sus atributos de acuerdo al proceso de control.
- Se visualizan las partículas.

Cada partícula activa en el sistema de partículas, es controlada a lo largo de su vida. Esto incluye no sólo su posición y velocidad, sino también sus atributos de visualización: forma, color, transparencia. Para animar la posición de las partículas, se calcula la velocidad después de aplicar factores aleatorios. Otros factores físicos como la gravedad, la velocidad del viento, las colisiones con otros objetos etc. . . también se emplea para actualizar los cálculos. La transparencia y el color pueden ser función del tiempo de vida. La forma puede variar con la velocidad.

V-E. Comportamiento asociado a los actores

El comportamiento es función de la naturaleza del objeto. El comportamiento se refiere a propiedades diversas que tienen que ver con cómo se desplazan y cómo se comportan ante otros actores. Los comportamientos suelen apoyarse en módulos de inteligencia artificial.

El uso de actores es una forma de control procedural de alto nivel. Un actor es un programa que se invoca una vez por cuadro para determinar las características de un objeto en la animación. Un actor se corresponde a un objeto en el sentido de la programación orientada al objeto. Un actor, durante la secuencia de ejecución una vez por cuadro puede enviar mensajes a otros actores para controlar su comportamiento.

Podríamos utilizar este método para controlar el movimiento de un tren. La máquina se corresponde con un actor que se ejecuta una vez cada cuadro y que cambia su posición de

acuerdo a las vías del tren y a la velocidad que lleve. A su vez, envía un mensaje al vagón que lleva detrás, indicándole que ponga su parte delantera inmediatamente detrás de su trasera. Este vagón a su vez, enviará el mismo mensaje a vagón que le precede y así sucesivamente.

Este concepto ha evolucionado ampliamente, incluyendo actores que realizan amplios rangos de comportamientos que pueden ejecutar en función de las circunstancias. Los objetos interactúan entre sí hasta tal punto que existen aplicaciones en las que los objetos simulan vivir, en el sentido de que nacen, crecen, se reproducen y mueren. La inteligencia artificial tiene mucho que decir en estos sistemas, donde se introduce una red de reglas causa-efecto para determinar el movimiento.

Un ejemplo son los **sistemas de bandadas**. Estos sistemas se caracterizan por tener un número medio de participantes cada uno de los cuales es controlado por típicamente un conjunto reducido de reglas que operan de forma local. Los participantes muestran un comportamiento de inteligencia limitada y están limitados por un conjunto de restricciones físicas reducido. Las restricciones físicas que se incorporan son típicamente para evitar colisiones, la gravedad y el arrastre.

Aunque el nombre del método se refiere a grupos de pájaros, el método se hace extensivo a cualquier conjunto de objetos de tamaño medio que se desplazan en grupo.

En estos sistemas existen dos fuerzas principales que hacen que se mantengan las bandadas: evitar las colisiones y permanecer en la bandada. Son fuerzas que compiten y que deben de balancearse.

Para determinar la trayectoria de cada elemento, se calculan en cada cuadro tres vectores velocidad distintos: el determinado por el módulo que se encarga de evitar colisiones, el del módulo encargado de mantener el grupo unido, y por último el determinado por la velocidad del grupo. La velocidad final del elemento se calcula como la media ponderada de estos tres vectores.

El principal problema con este tipo de sistema es que la complejidad es N^2 siendo N el número de elementos.

VI. CONCLUSIONES

La animación generada por ordenador se apoya en los fundamentos de animación tradicional reinterpretando los términos de cuadros clave, in-betweening, cel animation y rostoscopia. El ordenador puede servir de apoyo en todas estas tareas.

Existen una serie de técnicas fundamentales para realizar animación generada por ordenador como son el morphing, la animación paramétrica, la animación capturada y la animación procedural. Algunas técnicas más sofisticadas son el establecimiento de vínculos cinemáticos y restricciones, el control de movimiento empleando sistemas dinámicos y el control de colisiones.

El mayor nivel de sofisticación viene con el uso de los actores. Se trata de entidades autónomas que gobiernan su propio movimiento en función del entorno.

MÁS INFORMACIÓN

Sobre animación tradicional son recomendables las lecturas de cualquier edición de los libros de Laybourne [3] y de los libros de Disney [4].

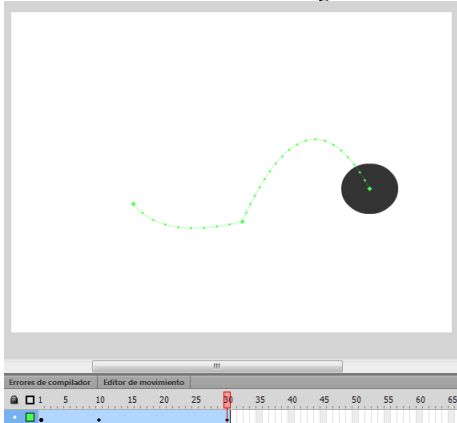
Los libros básicos de referencia sobre informática gráfica, todos suelen incluir un capítulo o varios sobre transformaciones geométricas. Por ejemplo, Hearn y Baker tratan los temas relativos a animación en el capítulo 16 de [5], [6] y 11 de [7]. Foley aborda el tema en el capítulo 35 de [8]. Shirley en el capítulo 16 de [9]. Watt dedica la cuarta parte de su libro [10] a describir técnicas avanzadas de animación.

En libros de multimedia suele aparecer también bastante información sobre técnicas de especificación de movimiento. Por ejemplo, el capítulo 5 de [11], el capítulo 8 de [12].

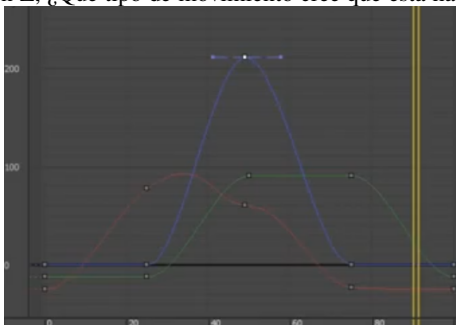
La mejor forma de aprender sobre animación es descargándose un programa específico como 3D Studio o Animate CC y empezar a probar.

EJERCICIOS Y CUESTIONES

1. Por qué se dice que las técnicas de animación por computador no son sólo un accesorio para enriquecer la comunicación visual.
2. Indique cinco usos de la animación por ordenador al margen del cine.
3. En qué se distingue la animación en tiempo real de la animación frame a frame.
4. En la generación de fotogramas en animación generada por ordenador, en qué se diferencia el tiempo de rendering del tiempo de control de movimiento.
5. Indique alguna diferencia entre técnicas de animación de bajo nivel y de alto nivel.
6. Defina cuadro clave en el contexto de la animación por ordenador.
7. ¿Por qué se dice que las técnicas de rotoscopía, cel animation e in-betweening abren el camino a la producción industrial de animaciones?
8. ¿En qué se diferencian los términos de animación asistida por ordenador y animación generada por ordenador?
9. Relacione tareas de animación asistida por ordenador relacionadas con la generación de cuadros.



10. Explique la fórmula de la interpolación lineal aplicada a animación asistida o generada por ordenador.
11. ¿Queremos hacer una animación en la que un cubo se desplace en una dirección dada. Si la animación dura 3s y queremos generar 15 fotogramas por segundo, cuantas posiciones debemos recalcular?
12. Dado un movimiento representado por la curva-P de la figura, marque qué posición ocupa el objeto en el tiempo $t=0,4s$ si es una animación realizada a 15 fps (fotogramas por segundo) y en total dura 2 segundos.
13. Las curvas paramétricas de la figura representan el movimiento de una esfera (rojo posición en X, verde posición en Y y azul posición en Z, ¿Qué tipo de movimiento cree que está haciendo



la esfera?

14. Indique qué es el warping dentro de las técnicas de morphing.
15. Enumere los cinco problemas fundamentales que debe resolver el control automático de la animación.
16. Explique las diferencias entre una definición de movimiento basada en la cinemática y una definición de movimiento basada en la dinámica.
17. Dado un modelo de figura articulada, explique las diferencias en la aplicación de movimiento empleando cinemática directa y cinemática inversa.

REFERENCIAS

- [1] J. Plateau, "Draft for a phenakistiscope disc representing a pirouetting dancer," *Corresp. Math. Phys.*, vol. VII, p. 291, 1832.
- [2] Adobe, *Animate CC User Guide*, 2020.
- [3] K. Laybourne, "The animation book: A complete guide to animated," *EUA: Three Rivers Pr*, 1998.
- [4] F. Thomas, O. Johnston, and F. Thomas, *The illusion of life: Disney animation*. Hyperion New York, 1995.
- [5] D. Hearn and M. P. Baker, *Gráficas por Computadora*, segunda ed. PHH Prentice Hall, 1995.
- [6] —, *Computer Graphics, C Version*, segunda ed. Prentice Hall, 1997.
- [7] D. Hearn, M. P. Baker et al., *Computer graphics with OpenGL*. Upper Saddle River, NJ: Pearson Prentice Hall, 2004.
- [8] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, E. Angel, and J. Hughes, *Computer graphics: principles and practice*. Addison-Wesley Professional, 2014.
- [9] S. Marschner and P. Shirley, *Fundamentals of Computer Graphics*, 5th ed. CRC Press, Taylor and Francis Group, 2022.
- [10] A. Watt and M. Watt, *Advanced Animation and Rendering Techniques*. acm Press, Addison-Wesley, 1992.
- [11] T. Vaughan, "Multimedia: Making it work," *Yogyakarta: Penerbit Andi*, 2006.
- [12] N. Chapman and J. Chapman, *Digital multimedia*. Wiley Publishing, 2009.