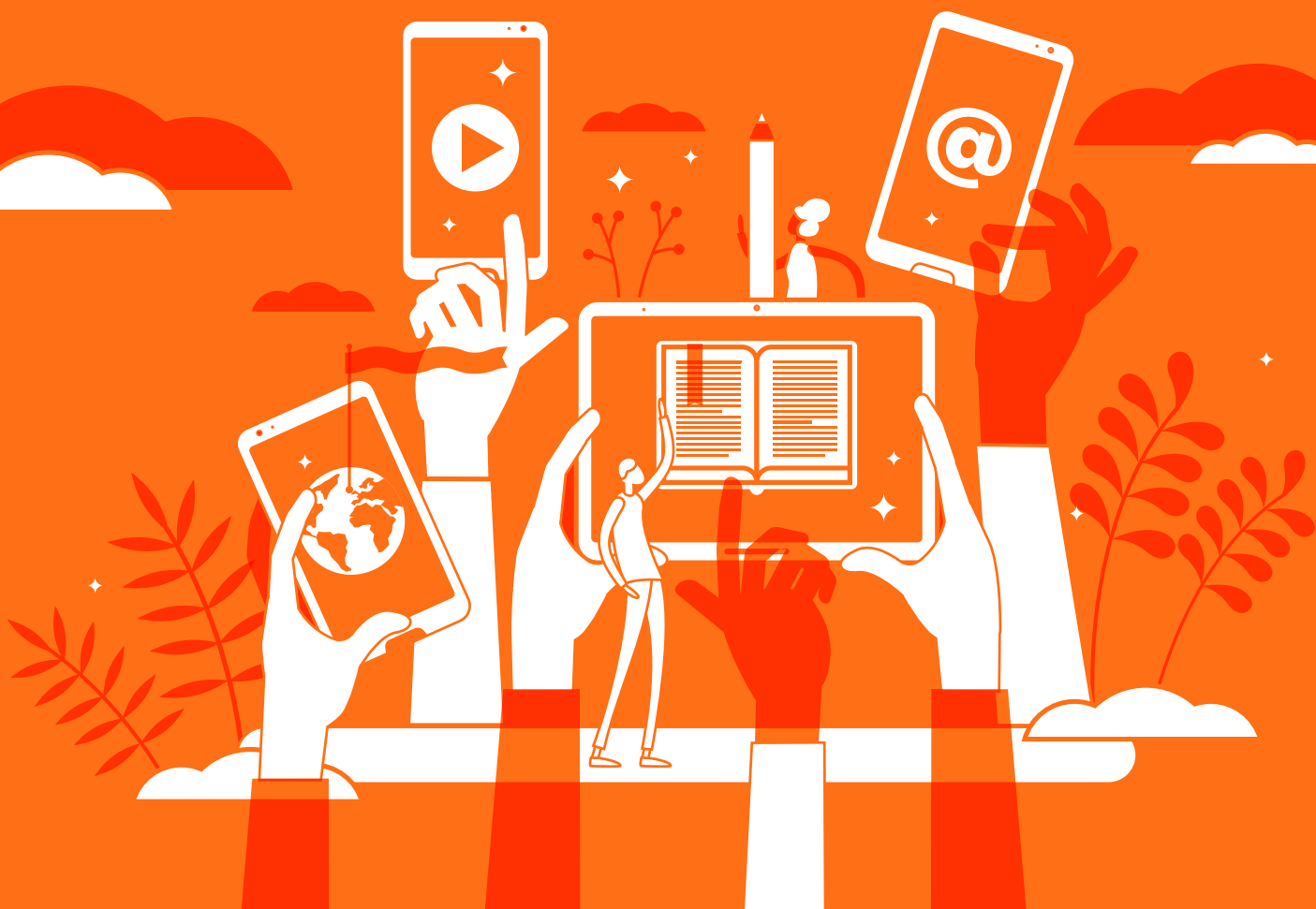


3

Creación de contenidos digitales





Formación en
Competencias
Digitales



Creación de
contenidos digitales

Nivel A2





Creación de contenidos digitales

ÍNDICE

3.1. DESARROLLO DE CONTENIDOS

- *Idoneidad de los formatos digitales*
- *Enriquecimiento de texto*
- *Diseño de imágenes: herramientas de dibujado, color y modificación*
- *Modificadores de capas*
- *Herramientas básicas para el diseño de imágenes vectoriales*
- *Herramientas para la compresión de imágenes*
- *Esquema de conexiones para la grabación y edición de vídeo*
- *Parámetros de ajuste para la compresión de vídeo*
- *Parámetros de ajuste para la compresión de audio*
- *Correcta visualización de medios digitales en los medios sociales más populares*

3.2. INTEGRACIÓN Y REELABORACIÓN DE CONTENIDO DIGITAL

- *Creación de presentaciones efectivas*
- *Representación gráfica de los datos incluidos en una hoja de cálculo*
- *Adquisición y uso de contenidos existentes*

3.3. DERECHOS DE AUTOR Y LICENCIAS DE PROPIEDAD INTELECTUAL

- *Licencias Creative Commons*
- *Incumplimientos de la Propiedad intelectual*

3.4. PROGRAMACIÓN

- *Técnicas de construcción de algoritmos*
- *Pseudocódigo*
- *Divide y vencerás*
- *Lenguajes de programación vs Lenguajes Naturales*
- *Intérpretes VS Compiladores*
- *Concepto de programa en Scratch*
- *Expresiones y asignación*
- *Mini-guía de estilo en Scratch*



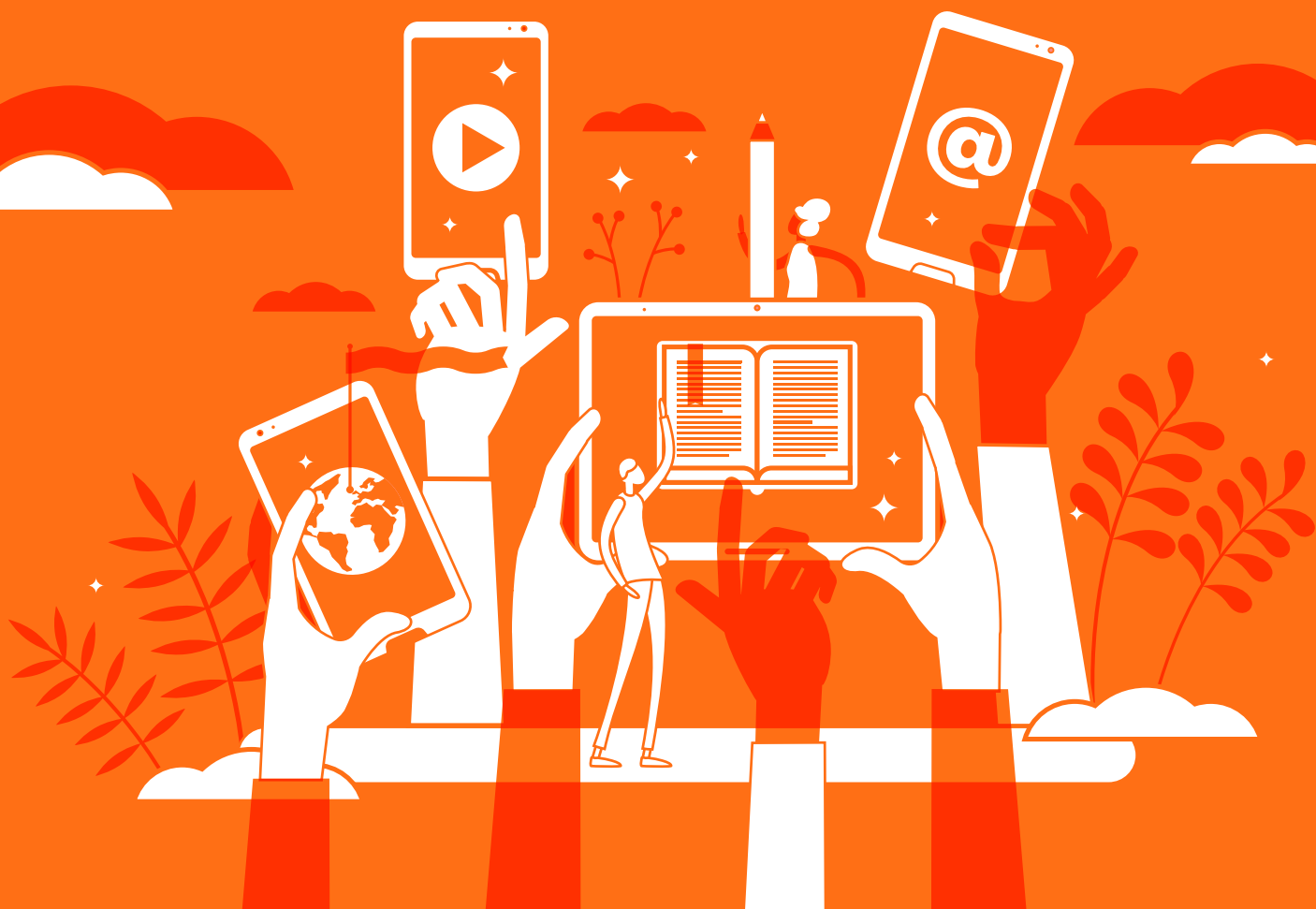


DigitAll

Creación de
contenidos digitales

3.1

DESARROLLO DE CONTENIDOS





Creación de
contenidos digitales

Nivel A2 3.1 Desarrollo
de contenidos

Idoneidad de los formatos digitales





Idoneidad de los formatos digitales

Formatos digitales

Los formatos digitales son una herramienta fundamental en el mundo de la información y la tecnología. Suponen una forma de almacenar y transmitir información y son utilizados en una amplia variedad de aplicaciones, desde la edición de documentos y la visualización de imágenes hasta la reproducción de audio y vídeo. La elección del formato digital adecuado puede ser crucial para garantizar que un contenido digital se vea y se comparta correctamente y su elección dependerá del propósito del contenido digital y de la compatibilidad del formato con los dispositivos y programas que se van a utilizar para acceder a él.

La necesidad de elegir entre formatos digitales dentro de cada categoría se debe a que cada formato tiene características y capacidades específicas. Por ejemplo, algunos formatos son mejores para la visualización de imágenes en alta resolución, mientras que otros son más adecuados para la reproducción de audio de alta calidad. Además, cada formato digital tiene sus propios requisitos de sistema y puede ser compatible con un conjunto diferente de dispositivos y aplicaciones.

Criterios de elección

En función del propósito de los contenidos digitales, es importante elegir un formato que cumpla con las necesidades específicas del contenido. Por ejemplo, si se desea compartir un archivo de audio a través de Internet, es importante elegir un formato de audio que sea compatible con la mayoría de los dispositivos y que permita una reproducción fluida en línea. De la misma manera, si se desea crear una presentación de diapositivas que se compartirá en línea, es importante elegir un formato que permita la visualización de las diapositivas de manera clara y precisa en diferentes dispositivos.





Algunos criterios que pueden ayudar a elegir entre distintos formatos digitales son:

- **La calidad del contenido:** algunos formatos ofrecen una mejor calidad de imagen o sonido que otros, por lo que pueden ser adecuados para contenidos multimedia como vídeos o audios.
- **La compatibilidad:** es importante asegurarse de que el formato elegido sea compatible con los dispositivos y programas que se van a utilizar para acceder al contenido.
- **La facilidad de uso:** algunos formatos son más fáciles de utilizar que otros, por lo que pueden ser más adecuados para ciertos propósitos.
- **La capacidad de almacenamiento:** algunos formatos ocupan menos espacio en disco que otros, por lo que pueden ser más adecuados para contenidos que se van a almacenar en dispositivos de almacenamiento limitado.
- **La facilidad de compartir:** aunque todos los formatos se comparten de la misma manera, algunos formatos pueden ser más adecuados para contenidos que se van a distribuir a través de internet o de otras vías de comunicación que requieren inmediatez o algunas características particulares.





Características de los formatos

Es importante elegir el formato digital adecuado en función del propósito del contenido y de las características del dispositivo o programa que se va a utilizar para acceder a él. A continuación, se presenta una tabla con criterios orientativos que pueden ayudar a elegir entre diferentes formatos digitales:

CARACTERÍSTICAS DE LOS FORMATOS PRINCIPALES

Formato	Calidad	Compatibilidad	Almacenamiento	Uso
JPG	Compresión elevada	Estándar internacional con alta compatibilidad	Tamaño ligero y práctico	Distribución de contenido de imágenes finales
PNG	Permite transparencias y una mayor calidad	Compatible con la mayoría de visores de imagen	Tiende a ocupar más espacio en disco, lo que supone su principal debilidad	Ideal para imágenes con vectores y tipografías que pretenden ser converidas a píxeles
GIF	Solo almacena 8-bits de información y 256 colores, no pierde calidad cuando se guarda	No almacena tipografías de forma ligera, permite la incorporación de animaciones pero no todos los visores de imagen las reproducen	Ocupa poco espacio	Imágenes digitales animadas e imágenes donde la limitación de color no sea un problema
TIFF	Muy elevada calidad sin compresión agresiva	No todos los visores de imagen permiten abrir este formato	Ocupa mucho espacio	Imágenes de alta resolución para imprimir con calidad
AVI	Video y audio de muy alta calidad	Compatible con muchos reproductores de vídeo	Compresión sin pérdidas, considerado contenedor oficial de vídeo	Ideal para archivar archivos multimedia
MP4	Calidad alta	Compatible con navegadores y la mayoría de dispositivos	Ocupa poco espacio	Ideal para compartir archivos multimedia
WAV	Distinto rango de calidad	Compatible con distintos sistemas operativos y reproductores	Ocupa más espacio que otros formatos en función de la calidad del audio	Adecuado para su uso en audio profesional
MP3	Calidad inferior en función del grado de compresión	Universalmente utilizado	Poco espacio	Adecuado para la mayoría de archivos de audio

Al elegir entre diferentes formatos digitales, es importante considerar la calidad que se necesita, la compatibilidad con los dispositivos y programas que se van a usar, el espacio de almacenamiento disponible y el propósito del contenido.

Saber más

Si deseas saber más, puedes consultar el libro **Formatos Digitales** de Jorge Franganillo Fernandez (ISBN 9788491809470).



Creación de
contenidos digitales

Nivel A2 3.1 Desarrollo
de contenidos

Enriquecimiento de texto





Enriquecimiento de texto

El enriquecimiento de texto se refiere al uso de diversas técnicas para mejorar la presentación visual del texto con el objetivo de hacerlo más atractivo y fácil de leer. El objetivo principal del enriquecimiento de texto es mejorar la velocidad de lectura y la comprensión, ya que permite al lector identificar rápidamente las ideas principales y las relaciones entre ellas.

Herramientas comunes

Algunas de las herramientas básicas más comunes para el enriquecimiento del texto son las que se comentan a continuación:

- **Negrita:** se utiliza para resaltar palabras o frases importantes dentro del texto.
- **Cursiva:** se utiliza para dar énfasis a palabras o frases, o indicar un término en otro idioma.
- **Subrayado:** se utiliza para resaltar títulos o frases importantes. Suele ser utilizado también para resaltar un enlace hipertexto o una palabra clave.
- **Cambio de tamaño de fuente:** se utiliza para hacer que el título de una sección sea más grande que el resto del texto o diferenciar títulos y subtítulos de párrafos.
- **Listas:** se utilizan para organizar el contenido en puntos o números.
- **Encabezados:** se utilizan para dividir el texto en secciones y darle estructura al contenido.
- **Interlineado:** se utiliza para aumentar o disminuir el espacio entre las líneas del texto.
- **Justificación:** se utiliza para alinear el texto a la izquierda, derecha o centrado.
- **Colores:** se utilizan para resaltar palabras o frases en el texto.
- **Imágenes:** se utilizan para añadir ilustraciones o gráficos al texto.





Todas estas herramientas ayudan a que el lector identifique rápidamente aquello que se quiere resaltar. Se pueden usar independientemente o combinadas entre sí.

Otra herramienta importante para el enriquecimiento del texto es el uso de **diferentes tipografías**. Es una forma de hacer que el texto sea más atractivo proporcionando contrastes visuales. La combinación de diferentes tipografías puede ayudar a diferenciar entre los títulos y los subtítulos, y puede mejorar la legibilidad del texto. Ayuda a dar sensación de **jerarquía** al texto. Sin embargo, es importante tener en cuenta que la combinación de diferentes tipografías debe ser coherente y estética, ya que un uso excesivo puede confundir al lector y dificultar la lectura. Por ejemplo, se puede utilizar una tipografía más grande y en negrita para los títulos y subtítulos, y una tipografía más pequeña y regular para el cuerpo del texto. Esto ayuda a guiar al lector a través del texto y a identificar rápidamente los diferentes elementos. Otra ventaja del uso de diferentes tipografías es que pueden ayudar a dar una sensación de **personalidad** y estilo al texto. Por ejemplo, se puede utilizar una tipografía más moderna y minimalista para un texto sobre tecnología, o una tipografía más tradicional y elegante para un texto sobre literatura.

A continuación, te presentamos un ejemplo de un texto plano y como queda transformado con el enriquecimiento para que valores la utilidad de estas herramientas.

Ejemplo de enriquecimiento de texto

Texto plano

Cambio climático.

Fenómenos atmosféricos.

El cambio climático se refiere a los cambios en el clima a nivel global debido a la actividad humana, como la emisión de gases de efecto invernadero. Los fenómenos atmosféricos incluyen eventos como las tormentas, las sequías y los huracanes. Estos fenómenos son causados por el cambio climático y pueden tener un impacto significativo en la vida humana y en el medio ambiente. Para más información www.un.org/what-is-climatechange.





Texto enriquecido

Cambio climático

Fenómenos atmosféricos

El cambio climático se refiere a los cambios en el clima a nivel global debido a la actividad humana, como la emisión de gases de efecto invernadero. Los fenómenos atmosféricos incluyen eventos como las tormentas, las sequías y los huracanes. Estos fenómenos son causados por el cambio climático y pueden tener un impacto significativo en la vida humana y en el medio ambiente. Para más información

www.un.org/what-is-climate-change.

Uso de plantillas

Por último, es importante mencionar la posibilidad de utilizar plantillas para aplicar formatos automáticamente al texto. Esto puede ahorrar tiempo y esfuerzo, y garantizar que el texto tenga una presentación consistente y profesional. Las plantillas son predefinidas con un conjunto de formatos y estilos, lo que permite a los usuarios aplicar rápidamente el enriquecimiento de texto sin tener que hacerlo manualmente. Esto es especialmente útil para documentos de gran tamaño o para aquellos que requieren un formato específico. Estas plantillas pueden ser utilizadas para crear documentos con un diseño y formato específico, como cartas comerciales, informes, boletines y mucho más. Incluyen elementos como el tamaño y el tipo de letra, el color de fondo, los márgenes y los estilos de párrafo.

Para utilizar plantillas, primero debes seleccionar la plantilla que deseas utilizar de las posibilidades que te ofrecen los programas de procesamiento de texto, editar una de las que te ofrecen o crear de inicio una. Luego, puedes editar el contenido del documento (texto plano) para adaptarlo a tus necesidades. Una vez que hayas terminado de editar el contenido, puedes guardar el documento como una nueva plantilla o como un archivo de texto normal y así recuperarías el formato enriquecido a tu gusto.





Las plantillas son utilizadas para aplicar estilos automáticamente al texto. Por ejemplo, si deseas que todos los títulos en tu documento tengan un tamaño de fuente más grande y sean negritas, puedes crear un estilo de título en tu plantilla y aplicarlo automáticamente a todos los títulos en tu documento.

En definitiva, el enriquecimiento de texto: mejora la comprensión del contenido a los lectores, aumenta la interacción y la retención del conocimiento, hace más atractivo el contenido por lo que aumenta la probabilidad de que el lector lea la información completa o la distribuya y recomiende a otros lectores potenciales.

Saber más

Para más información sobre el uso de plantilla puedes ver el vídeo **A3C31A2V03**.



APLICACIÓN DE PLANTILLAS DE DOCUMENTOS

e.digitall.org.es/A3C31A2V03



Creación de
contenidos digitales

Nivel A2 3.1 Desarrollo
de contenidos

Diseño de imágenes: herramientas de dibujado, color y modificación





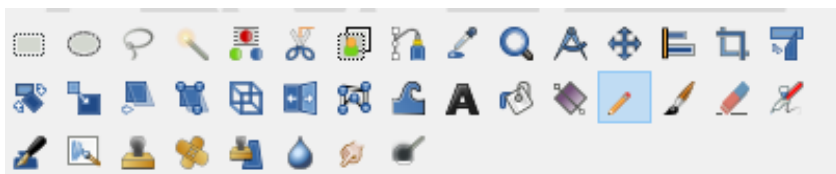
Diseño de imágenes: herramientas de dibujo, color y modificación

En este documento vamos a explicar las herramientas de dibujo, color y modificación. Para ello utilizaremos el programa Gimp (gimp.org.es). Es el programa de edición y retoque de gráficos más popular de software libre. No obstante, cualquier otro programa de edición de imágenes contiene herramientas similares.

Activación de las herramientas de dibujo

Para activar cada una de las utilidades de dibujo que vamos a ver a continuación podremos:



- 1| Seleccionar Herramientas → Herramientas de pintura → Lápiz/Pincel/etc.
- 2| Pulsar el icono de la herramienta en el desplegable.





- 3| Pulsando atajos de teclado (como N).

Opciones

En las **Herramientas de dibujo** puedes encontrar varias opciones de dibujo, como la herramienta de pincel, lápiz, aerógrafo, tinta, etc. Para usarlas simplemente debes seleccionar la herramienta deseada y comenzar a dibujar en una imagen abierta o un fondo. La característica que tienen en común es que todas se usan moviendo el puntero sobre la muestra de la imagen, creando pinceladas. Su uso intenta asemejar el movimiento natural que se realiza al pintar sobre un papel real.

El **Lápiz**  es una herramienta valiosa para dibujantes y usuarios que buscan precisión y control en sus trabajos. Su capacidad para crear líneas finas y precisas lo hace ideal para trabajar con imágenes pequeñas o detalladas. El **pincel**  en comparación con el lápiz, produce bordes difusos.

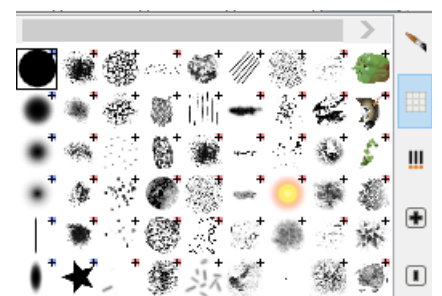
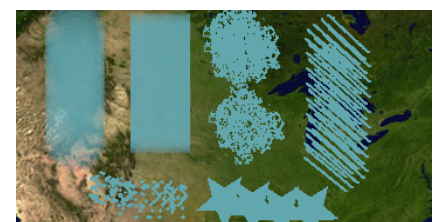
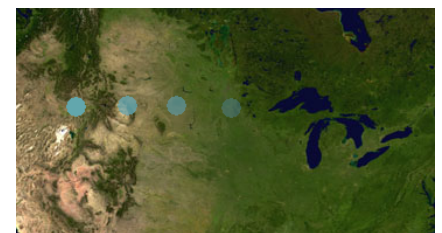


La herramienta **aerógrafo**  pintará áreas suaves de color. La **tinta**  simula una pluma estilográfica con plumín. A continuación, tienes un ejemplo sobre la misma imagen de los resultados del trazado con cada una de estas herramientas de dibujo.

Puedes cambiar el tamaño, color y la forma de las herramientas de dibujo desde las opciones de configuración en la barra de herramientas.

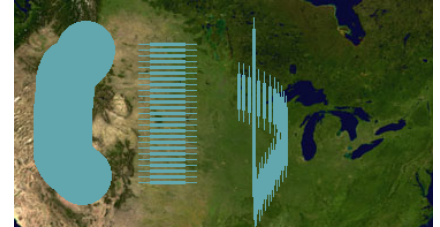
Todas las herramientas de pintura tienen algunas opciones compartidas como son modo, opacidad, pincel, tamaño, proporción de aspecto, ángulo o espaciado. A continuación describimos su funcionamiento utilizando como fondo una imagen de superficie terrestre:


- **Modo:** controla cómo se mezclan los colores en la capa actual. Puedes elegir entre diferentes opciones como "Normal", "Mezclar", "Multiplicar", entre otros. En la siguiente imagen aparece un trazado de lápiz en el que hemos ido usado los modos normal, disolver, clarear solo, oscurecer solo, solapar, o diferencia sobre una imagen.
- **Opacidad:** controla la transparencia de la capa actual. Un valor de 100% significa que la capa es completamente opaca, mientras que un valor de 0% significa que la capa es completamente transparente. En la siguiente imagen aparece un trazado de lápiz en el que hemos ido usado las opacidades 100%, 75%, 50%, 25% y 0% bajo un modo normal de lápiz de izquierda a derecha. Véase que la opacidad al 0% implica que no se traza nada sobre la imagen.
- **Pincel:** controla el tipo de pincel que se usa para dibujar. Puedes elegir entre diferentes opciones como "Redondo", "Punto", "Línea", entre otros. En el siguiente ejemplo hemos utilizado distintos tipos de bloques y formas. Al lado del ejemplo encuentras todas las posibilidades de trazos disponibles. Además, puedes descargar gratuitamente otros tipos de pinceles y añadirlos a tu panel (o editar y crear nuevos a partir de cero).





- **Tamaño:** controla el tamaño del pincel. Un valor mayor significa que el pincel es más grande, mientras que un valor menor significa que el pincel es más pequeño.
- **Proporción de aspecto:** controla la relación entre el ancho y la altura del pincel. Un valor de 1 significa que el pincel es cuadrado, mientras que un valor mayor significa que el pincel es más alto que ancho, y un valor menor significa que el pincel es más ancho que alto. En la siguiente imagen tenemos un trazo sobre la superficie con proporción de aspecto 1, +20 y -20 de izquierda a derecha.
- **Ángulo:** controla el ángulo del pincel. Un valor de 0 significa que el pincel está en posición vertical, mientras que un valor mayor significa que el pincel está inclinado hacia la derecha, y un valor menor significa que el pincel está inclinado hacia la izquierda.
- **Espaciado:** controla la distancia entre los trazos del pincel. Un valor mayor significa que los trazos están más separados, mientras que un valor menor significa que los trazos están más juntos.



Por último, para poder cambiar el color del trazo del dibujo que se está realizando, simplemente con realizar doble click sobre el icono  podremos cambiar el color del primer plano a nuestro gusto.

Saber más

Si quieres profundizar más en el manejo de estas herramientas puedes acceder al manual de usuario en docs.gimp.org/2.10/es. Esperamos que hayas profundizado en los tipos de trazos en los dibujos y cómo modificar sus características principales.





Creación de
contenidos digitales

Nivel A2 3.1 Desarrollo
de contenidos

Modificadores de capas





Modificadores de capas

En los programas de edición de imagen nos vamos a encontrar paquetes de herramientas que incluyen una serie de modificaciones predefinidas comunes que nos van a ayudar a modificar nuestras capas de forma rápida en función de las capas que hay debajo. En general, nos vamos a encontrar estos modificadores de capas en el menú derecho de los programas de edición agrupados por temáticas y también los podemos encontrar por la denominación *modos de fusión* o *modos de capas*.

Modos de capa en GIMP

A continuación, dejamos un listado de estos modificadores, agrupados en diferentes grupos, junto con una breve descripción de los efectos que producen. Algunos de estos se muestran en el **vídeo A3C31A2V05**.



**EFFECTOS DE LOS
MODIFICADORES
DE CAPAS**

e.digitall.org.es/A3C31A2V05

Modificadores de capas en GIMP

Grupo	Modificador	Descripción
Normal	Normal	Es el predeterminado: la capa superior cubre las de abajo, aunque lo podemos regular mediante la opacidad.
	Disolver	Disuelve la capa superior en la de abajo usando áreas transparentes.
	Borrado de color	Toma los colores de la capa superior y los elimina de la capa inferior.
	Borrar	Elimina los píxeles de la capa superior siempre que haya píxeles en la capa inferior.
	Combinar	Combina dos imágenes, rellenando los huecos si los hubiera.
	Partir	Sustrae la capa superior de la inferior.
Aclarar	Clarear solo	Compara cada píxel y deja los más claros (el negro no tiene efecto) basándose en la luminosidad.
	Aclarar solo la luma/ luminancia	Muy similar al anterior, pero basa la comparación en los calores de luma o luminancia.
	Pantalla	Busca el valor de cada píxel y multiplica los colores contrarios, dando lugar a una imagen aclarada.
	Blanquear	Se usa para incrementar la exposición en las zonas oscuras de la imagen.
	Suma	Los píxeles de las capas superior e inferior se suman.



Grupo	Modificador	Descripción
Oscurecer	Oscurecer solo	De la comparación de píxeles entre capas se eligen los de valores más pequeños.
	Oscurecer solo la luma/ luminancia	Muy similar al anterior, pero basa la comparación en los calores de luma o luminancia.
	Multiplicar	Multiplifica los píxeles de cada capa y divide el resultado por 255 dando una imagen normalmente más oscura.
	Ennegrecer	Similar al anterior. Se usa para que aparezcan detalles de zonas claras.
	Ennegrecido lineal	Similar a multiplicar solo que suma los píxeles de cada capa y le resta 255, obteniendo de nuevo una imagen más oscura.
Contraste	Solapar	Se modifican los píxeles oscureciendo la imagen, pero no tanto como en "multiplicar".
	Claridad suave	En ocasiones, idéntico al modo "Solapar", generando imágenes más suavizadas y brillantes.
	Claridad fuerte	Se obtienen colores brillantes y bordes definidos.
	Luz puntual	Va a aclarar los píxeles más claros y a oscurecer los más oscuros, los grises desaparecen.
	Luz lineal	Similar a los anterior solo que hace los claros más brillantes y los oscuros menos brillantes.
	Mezcla fuerte	Suma los valores de los colores primarios, si el resultado es mayor de 255 deja ese color, en caso contrario lo sustituye por blanco
Inversión	Diferencia	Se resta el valor de los píxeles de la capa superior a la inferior.
	Exclusión	Muy similar al anterior solo que hace una combinación menos intensa.
	Sustraer	Similar a "Diferencia" pero dando lugar a imágenes más oscuras.
	Extraer granulado	Puede servir para dar relieve.
	Combinar granulado	Opuesto a "Extraer granulado".
	Dividir	Produce imágenes con aspecto quemado, se puede usar para eliminar un color determinado.
Cancelación	Tono HSV	Toma el tono de la capa superior y lo combina con la intensidad y saturación de la inferior.
	Saturación HSL	Toma la saturación de la capa superior y el tono y la intensidad de la inferior.
	Color HSL	Toma el tono y la saturación de la capa superior y la luminosidad de la inferior.
	Valor HSV	Toma la intensidad de la capa superior y el tono y la saturación de la inferior.



Grupo	Modificador	Descripción
Componente	Tono LCh	Mantiene el tono de la superior y la luminosidad y el croma de la inferior.
	Chroma LCh	Mantiene el croma de la capa superior y le asigna el tono y la luminosidad de la inferior.
	Color LCh	Toma el tono y el croma de la capa superior y la luminosidad de la inferior.
	Luminosidad LCh	Mantiene la luminosidad de la capa superior y el croma y el tono de la inferior.
	Luminancia	Combina la luminancia de la capa superior con el tono y el croma de la inferior.

Saber más

Puedes consultar la documentación oficial de GIMP en el siguiente enlace:

e.digitall.org.es/gimp-layers





Creación de
contenidos digitales

Nivel A2 3.1 Desarrollo
de contenidos

Herramientas básicas para el diseño de imágenes vectoriales





Herramientas básicas para el diseño de imágenes vectoriales

Clasificación de herramientas básicas para el diseño de imágenes vectoriales

Existen diversas herramientas básicas que se utilizan en el diseño de imágenes vectoriales. Estas herramientas se pueden agrupar en varias categorías, como **herramientas de dibujo**, **herramientas de formato** y **herramientas de edición**. Cada categoría incluye un conjunto de herramientas específicas que se utilizan para realizar tareas concretas en el diseño de imágenes vectoriales.

- **Herramientas de dibujo:** se utilizan para crear y modificar los elementos básicos de una imagen vectorial, como líneas, formas y texto. Estas herramientas suelen incluir opciones para definir el grosor y el color de las líneas, así como la capacidad de aplicar rellenos y patrones a las formas dibujadas.
- **Herramientas de formato:** se utilizan para darle estilo y formato a los elementos de una imagen vectorial. Estas herramientas pueden incluir opciones para cambiar el tamaño y la orientación de los objetos, así como para aplicar efectos como sombreado y degradado a los mismos.
- **Herramientas de edición:** se utilizan para manipular y organizar los elementos de una imagen vectorial. Estas herramientas pueden incluir opciones para mover, copiar y eliminar.

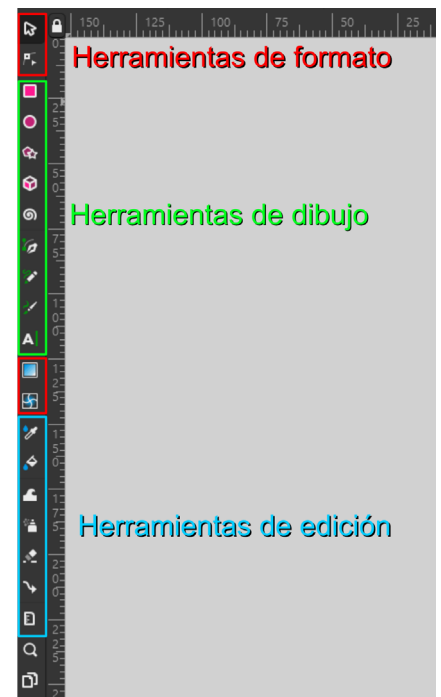


Figura 1. Herramientas agrupadas.

En la Figura 1 (a la derecha) se muestran estas herramientas agrupadas.

Algunas herramientas básicas

Algunas de las herramientas básicas más comunes son:

- **Herramienta de selección:** permite seleccionar y manipular objetos individuales o grupos de objetos en una imagen. Para utilizar la herramienta de selección, primero hay que seleccionarla en la barra de herramientas del



programa de diseño que se esté utilizando. Después, se puede hacer clic en un objeto individual en el diseño para seleccionarlo. También se puede utilizar la herramienta de selección para dibujar un rectángulo alrededor de varios objetos para seleccionarlos a todos a la vez. Una vez seleccionado uno o más objetos, se puede utilizar la herramienta de selección para moverlos, cambiar su tamaño, rotarlos, o aplicar cualquier otro efecto que se desee. Esto es útil para realizar cambios precisos y controlados en el diseño sin afectar a otros elementos. La herramienta de selección también puede ser utilizada para seleccionar elementos específicos dentro de un grupo de objetos. Por ejemplo, si en un grupo de círculos y cuadrados en el diseño, se puede utilizar la herramienta de selección para seleccionar solo los círculos y aplicarles un efecto, mientras que los cuadrados permanecerán intactos.

- **Herramienta de línea:** permite dibujar líneas rectas y curvas precisas en una imagen. Para utilizar la herramienta de línea en un programa de diseño de imágenes vectoriales, primero hay que seleccionar la herramienta en la barra de herramientas del programa. A continuación, hay que hacer clic en el punto de inicio de la línea en la imagen y arrastrar el ratón mientras se mantiene el botón del ratón presionado hasta el punto de finalización de la línea. Esto dibujará una línea recta entre los dos puntos. Si se desea dibujar una curva, se puede utilizar la herramienta de línea curva. Se dibujará del mismo modo, pero tras dibujarla, se podrá hacer clic en el punto final de la curva y arrastrar el ratón mientras se mantiene el botón del ratón presionado hasta el punto de finalización de la curva. Esto dibujará una curva suave entre los tres puntos.
- **Herramienta de forma:** permite dibujar formas geométricas básicas como círculos, rectángulos y elipses en una imagen. Esta herramienta suele incluir opciones para definir el grosor y el color de las líneas que forman la forma, así como la capacidad de aplicar rellenos y patrones a la misma. Una vez seleccionada la herramienta y la forma específica que se quiere dibujar, se puede hacer clic en algún punto del diseño y desplazar el cursor para dibujar la forma geométrica deseada. Una vez creada, se puede modificar su tamaño y resto de características.





- **Herramienta de texto:** permite agregar texto a una imagen. Una vez seleccionada, se puede hacer clic en la posición en la imagen donde se desea agregar el texto y comenzar a escribir. La herramienta de texto permitirá ajustar la fuente, el tamaño y el estilo del texto, así como su color y alineación. Una vez que el texto se ha agregado a la imagen, se puede mover y ajustar su posición utilizando la herramienta de selección. También se pueden aplicar efectos como sombras y reflejos para darle más profundidad y dimensiones al texto.
- **Herramienta de pincel:** permite dibujar líneas suaves y curvas de manera más natural que con la herramienta de línea. Una vez seleccionada, se puede hacer clic en cualquier punto en la imagen y arrastrar el cursor para dibujar una línea en la dirección deseada. La herramienta de pincel permite ajustar el tamaño y el estilo del pincel, así como su color y transparencia, para crear diferentes efectos y estilos de pintura. También permite dibujar formas más complejas y precisas. Para hacer esto, primero se debe seleccionar la opción “forma” en la barra de herramientas, luego hacer clic en el punto de inicio de la forma y arrastrar el cursor en la dirección deseada. Mientras se arrastra el cursor, se pueden agregar puntos de control adicionales para crear formas más precisas y detalladas.
- **Herramienta de relleno:** permite aplicar colores o patrones a las formas dibujadas en una imagen. Para utilizar esta herramienta, primero se debe seleccionar una forma o un objeto en la imagen utilizando la herramienta de selección. Luego, se selecciona la herramienta de relleno. Esta permitirá elegir un color, un patrón o una textura para aplicar al objeto seleccionado. Una vez que se ha aplicado el relleno, se puede ajustar su posición y su apariencia utilizando las opciones disponibles en la herramienta de relleno. Por ejemplo, se puede cambiar el color del relleno, ajustar su transparencia o agregar efectos.
- **Herramienta de alineación:** que permite alinear y distribuir objetos en una imagen de manera precisa. Esta herramienta es muy útil para asegurarse de que las formas y objetos en una imagen estén organizados de manera estructurada. En primer lugar, se deben seleccionar



las formas u objetos que se desea alinear utilizando la herramienta de selección. La herramienta de alineación permitirá elegir cómo alinear los objetos seleccionados, ya sea en el centro, en la parte superior, en la parte inferior, a la izquierda o a la derecha. Una vez que los objetos se han alineado, se pueden mover y ajustar su posición utilizando la herramienta de selección. También se pueden agrupar los objetos alineados utilizando la herramienta de agrupación.

Saber más

Si deseas saber más sobre diseño de vectores, puedes consultar **El gran libro de Illustrator**. El vector del diseño de Eduardo Guarniz Izquierdo, ISBN 8426732771.





Creación de
contenidos digitales

Nivel A2 3.1 Desarrollo
de contenidos

Herramientas para la compresión de imágenes





Herramientas para la compresión de imágenes

Existen varias herramientas para la compresión de imágenes, tanto de software libre como privativo. Su utilidad consiste en reducir el tamaño de las imágenes sin perder demasiada calidad, lo que permite que las imágenes se carguen más rápido en la web y se utilicen en aplicaciones móviles o archivar las imágenes sin ocupar tanta memoria.

Ejemplos de herramientas

Algunas de las herramientas más populares incluyen:

- **Adobe Photoshop:** es uno de los editores de imágenes más utilizados y cuenta con una gran variedad de herramientas de compresión, incluyendo opciones para guardar imágenes en formato JPG, PNG y GIF.
- **GIMP:** es un software libre y gratuito, similar a Photoshop, que también cuenta con herramientas de compresión de imágenes.
- **ImageOptim:** es una aplicación para Mac que permite optimizar el tamaño de las imágenes de forma automática, utilizando diferentes algoritmos de compresión.
- **Compressor.io:** es una herramienta en línea que permite comprimir imágenes en formato JPG, PNG y GIF, y también permite ajustar el nivel de compresión.
- **TinyPNG:** es una herramienta en línea que optimiza automáticamente las imágenes PNG y JPG, reduciendo su tamaño sin perder calidad.
- **Figma:** es una herramienta en línea de diseño colaborativo que permite a los diseñadores colaborar y trabajar en proyectos en tiempo real.



Opciones de compresión

En cuanto a cómo estas herramientas están incluidas en las propias aplicaciones de diseño, generalmente se encuentran en la opción de “guardar” o “exportar” y permiten seleccionar el formato de imagen y el nivel de compresión deseado.



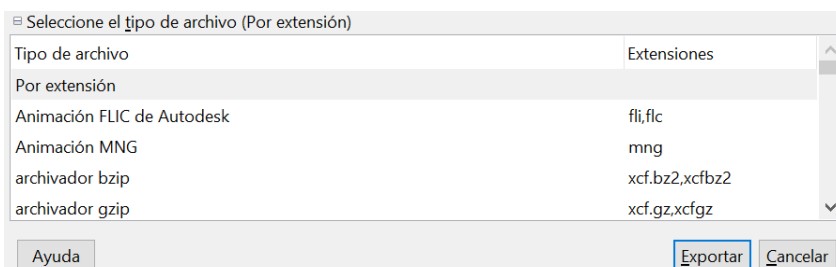
Por ejemplo, en Adobe Photoshop, la herramienta de compresión de imágenes se encuentra en el menú “Archivo” y se llama “Guardar para web”. Esta herramienta permite seleccionar el formato de la imagen (como JPEG, PNG o GIF), la calidad de la imagen y el tamaño de archivo deseado. Una vez que se han seleccionado estas opciones, se puede guardar la imagen comprimida en el disco duro.

En otras aplicaciones de diseño, como Sketch o Figma, las herramientas de compresión de imágenes se encuentran en la sección “Exportar” o “Exportar como”. Estas herramientas permiten seleccionar el formato de la imagen, la calidad de la imagen y el tamaño de archivo deseado. Una vez que se han seleccionado estas opciones, se puede exportar la imagen comprimida a su disco duro.

Paso a paso en GIMP

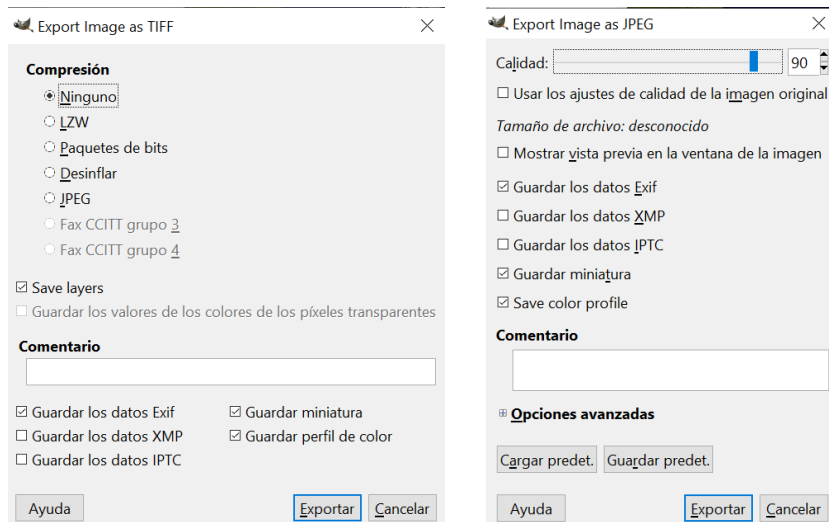
A continuación, se describen los pasos que se deben seguir en GIMP, software libre para la edición de imágenes, para guardar los diseños con distintos niveles de compresión.

- 1 | Abra la imagen en GIMP.
- 2 | Haga clic en “Archivo” y seleccione “Exportar como” para guardar una copia de la imagen en un formato diferente.
- 3 | En la ventana de “Exportar como”, seleccione el formato de imagen deseado del desplegable “Seleccione el tipo de archivo (por extensión)” como se ve en la figura de abajo. Los formatos populares incluyen JPEG, PNG, GIF y BMP. Sin embargo, el programa ofrece otras múltiples opciones como:





4 Una vez seleccionado el tipo de archivo se abre un cuadro de diálogo en el que se elige el método de compresión. El cuadro de diálogo es un poco diferente en función del tipo de archivo (véanse los dos ejemplos a continuación).



5 Si hay opciones avanzadas, seleccione el nivel de compresión deseado. El nivel de compresión más alto significa que la imagen se comprimirá más, lo que a su vez significa que se perderá mayor calidad de imagen.

6 Haga clic en "Aceptar" para aplicar la compresión.

7 Escriba un nombre de archivo y seleccione una ubicación para guardar la imagen comprimida.

8 Haga clic en "Exportar" para guardar la imagen comprimida.

i Saber más

Es importante elegir el formato de imagen adecuado según el tipo de imagen que se está guardando. Todos los programas de compresión de imagen te darán la posibilidad de elegir el formato óptimo y las características de compresión para él.



Creación de
contenidos digitales

Nivel A2 3.1 Desarrollo
de contenidos

Esquema de conexiones para la grabación y edición de vídeo





Esquema de conexiones para la grabación de vídeo

En este apartado vamos a describir una de las posibles configuraciones que podría adquirir un estudio de grabación de vídeo, representada en la figura 1. Por lo general, estos estudios van a variar dependiendo de los recursos disponibles y de las necesidades de grabación del usuario, pero todos cuentan con una serie de elementos comunes que se pueden sumar o restar según las necesidades.

En el esquema presentado en la Figura 1 se muestra en la parte central un ordenador desde donde se integrará toda la grabación. Esta configuración facilita bastante la edición posterior de la grabación ya que, mediante el uso de un programa como OBS, que ya hemos mencionado previamente, vamos a poder predefinir diversas escenas de grabación e ir pasando de una a otra. Además, estas funciones se pueden hacer de forma más profesional gracias a los **mezcladores de vídeo**.

Esta configuración nos va a permitir grabar lo que estamos haciendo en el ordenador y además nos va a permitir integrar una o varias **cámaras**. Estas pueden ser webcams, que se conectan al equipo principal vía USB, o bien cámaras o videocámaras externas, las cuales la mayoría incorpora una salida de vídeo que podremos conectar a través de un cable HDMI a una capturadora de vídeo y de ahí a nuestro equipo. Una capturadora de vídeo no es más que un aparato que nos va a permitir recoger el material emitido por otro dispositivo (una cámara, una videoconsola...) en nuestro ordenador.

Por otro lado, tendremos los **micrófonos**, que emplearemos para mejorar la calidad del audio e incorporaremos a nuestro equipo a través de una conexión USB generalmente. En la imagen se muestra un micrófono de condensador con filtro anti-pop, pero recuerda que en el mercado hay disponibles micrófonos de muchas características.

Finalmente, tendremos la **iluminación**. En la imagen se muestran dos focos de tipo LED colocados de forma simétrica para poder obtener luz difusa. Normalmente, este tipo de focos nos va a permitir regular la cantidad de luz con la que queremos trabajar, así como emplear distintos tipos de filtros.



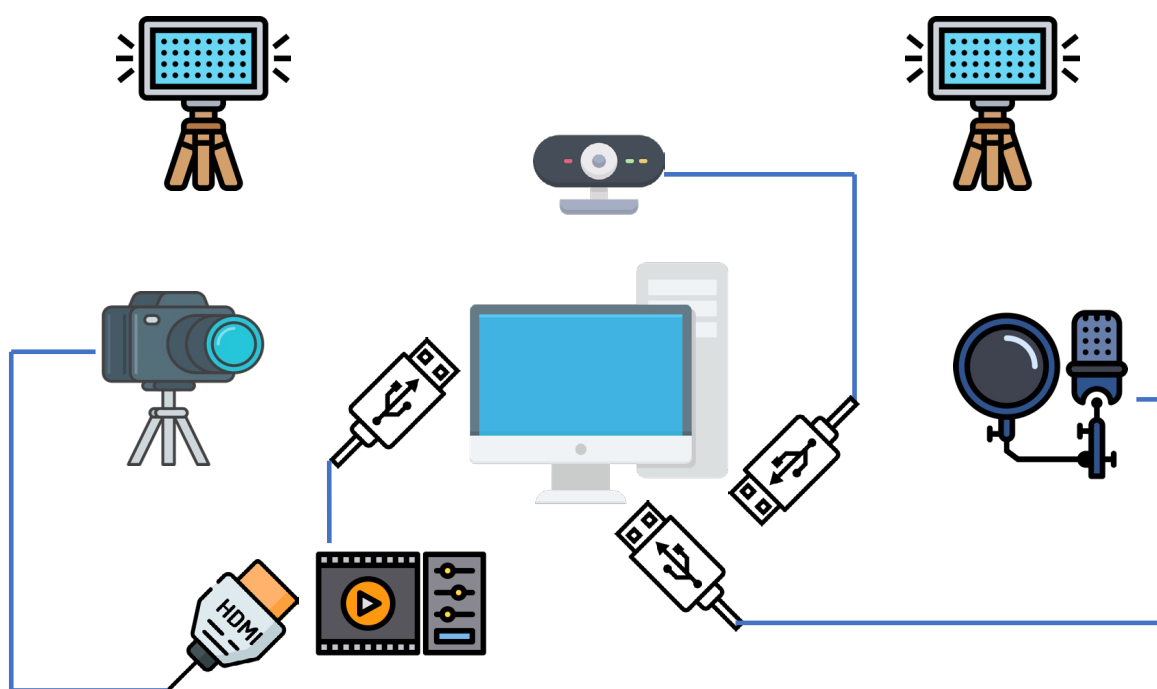


Figura 1. Representación de una posible configuración para un estudio de grabación de vídeo.

Saber más

Puedes acceder a las recomendaciones para grabar contenido de Udemy a través del siguiente enlace:

e.digitall.org.es/udemy





Creación de
contenidos digitales

Nivel A2 3.1 Desarrollo
de contenidos

Parámetros de ajuste para la compresión de vídeo





Parámetros de ajuste para la compresión de vídeo

Cuestiones previas

Como ya hemos comentado en otras ocasiones que hemos trabajado el concepto de compresión, la compresión de vídeo es un proceso en el que vamos a reducir el tamaño de nuestro archivo. Dado que el tamaño de los archivos representa la cantidad de información que contienen, al reducir su tamaño, vamos a perder información.

Existen bastantes parámetros de los que depende la compresión de un vídeo, empezando por las características del propio vídeo, que determinan cuánta información vamos a perder en ese proceso, es decir, cuánta calidad va a perder nuestro vídeo. Por lo general, se busca que el proceso de compresión, mediante diferentes técnicas, elimine información redundante que afecte lo menos posible a la calidad de nuestro archivo. No obstante, en otras ocasiones decidiremos sacrificar la calidad del archivo porque debemos cumplir algún requisito, por ejemplo, que para enviar un vídeo no podemos superar un determinado tamaño, por lo que necesitaremos comprimir lo máximo posible, aunque perdamos mucha información por el camino.

Parámetros para la compresión de vídeo

A continuación, vamos a describir la influencia de algunos parámetros que podemos ajustar a la hora de comprimir un vídeo:

- **Resolución:** para comprimir un vídeo podemos perder resolución. Esta medida puede parecer muy drástica puesto que todos relacionamos intuitivamente la resolución con la calidad de un vídeo, pero hemos de tener en cuenta que queremos comprimir un vídeo en base a una necesidad. Por ejemplo, si tenemos un vídeo a una resolución de 4K (3840x2160 píxeles) y queremos que se reproduzca de forma óptima en móviles, generalmente no va a ser apreciable la pérdida de calidad con una





compresión a resolución Full HD (1920x1080 píxeles), que es disminuir 4 veces la resolución de nuestro vídeo, porque, para la gran mayoría de los móviles actuales, Full HD es la máxima resolución a la que pueden trabajar.

- **Escala:** en ocasiones, en vez de ajustar la resolución, vamos a ajustar un parámetro análogo que es la escala del vídeo de salida. En el ejemplo que hemos puesto anteriormente, representado en la figura 1, estaríamos seleccionando una escala de 0,5 (cuatro veces más pequeña que el original).
- **Tasa de bits:** o *bitrate*, usando el término inglés, es la cantidad de bits por segundo que se transmiten en el vídeo. Generalmente hablaremos de **Mbps** (*megabites per second*, del término en inglés). Dado que estamos hablando de flujo de información por unidad de tiempo, va a ser un factor muy importante a la hora de determinar la calidad de un vídeo y, por tanto, con gran influencia sobre la compresión de éste. De hecho, la tasa de bits es el factor que determina principalmente el tamaño de un vídeo. Aunque es cierto que, a mayor tasa de bits, mayor calidad de vídeo, una tasa de bits demasiado elevada no nos va a mejorar la calidad a la vez que nos puede generar vídeos innecesariamente pesados. Por otro lado, esta afirmación es cierta cuando estamos comparando vídeos de la misma resolución. Si vamos a comprimir dos vídeos de diferente resolución a una misma tasa de bits, pongamos por ejemplo el vídeo 4K y el Full HD del ejemplo anterior, dado que la cantidad de información que se está procesando es la misma (misma tasa de bits), el nivel de compresión deberá ser mucho mayor para el vídeo de mayor resolución, por lo que, en definitiva, al comparar los resultados tras la compresión, va a tener peor calidad que el vídeo que en origen tenía mucha más resolución, puesto que su proceso de compresión ha debido ser mucho más agresivo. Dicho con otras palabras, para mantener la calidad de un vídeo de elevada resolución tras la compresión debemos usar una elevada tasa de bits para tener la misma calidad de salida que tendríamos con un vídeo originariamente de menor calidad.

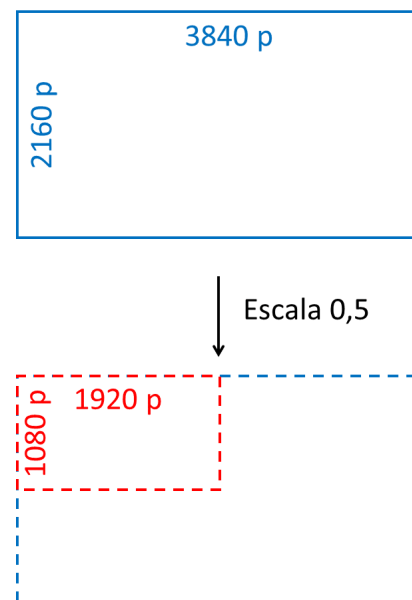


Figura 1. Ejemplo de reducción de escala.



- **Tasa de fotograma:** los vídeos están formados por una sucesión de imágenes o fotogramas. A la velocidad a la que van pasando esas imágenes es a lo que llamamos tasa de fotograma. En el estándar europeo o sistema PAL, los vídeos deben tener 25 fotogramas por segundo (fps). Como podemos intuir, a mayor cantidad de fotogramas por unidad de tiempo percibiremos una imagen muy fluida, mientras que si el número de fps es bajo percibiremos que los movimientos se suceden de forma brusca. Por lo tanto, a hora de comprimir un vídeo, no sería deseable bajar de esos 25 fps para evitar ver los movimientos a “tirones”. No obstante, la tecnología actual ha permitido que esta tasa pase de ser una limitación a algo convencional.
- **Calidad del vídeo:** en algunos programas vamos a tener, además de todas las demás opciones, una opción genérica denominada “calidad de vídeo”, que nos va a permitir seleccionar una compresión de *elevada calidad*, en la que vamos a obtener mejores calidades de vídeo, pero mayores tamaños, de *baja calidad*, donde perderemos lógicamente calidad para ganar efectividad en la compresión, o bien valores intermedios. La idea es que especifiquemos la calidad de vídeo que queremos para todo el vídeo y el sistema optimice la tasa de bits para ofrecernos esa calidad constante, reduciendo el tamaño de los datos para las escenas de poco movimiento y utilizando más bits para las escenas de mucho movimiento.
- **Códec:** en la actualidad, uno de los códecs más empleados es H.264 (o MPEG-4 parte 10) por proporcionar buenas tasas de compresión manteniendo una buena calidad de imagen. Recordemos que los códecs son programas que usamos para comprimir y descomprimir vídeo y audio para así poder reducir el tamaño de los archivos por lo que van a ser también una pieza fundamental del proceso pues influyen no solo en la calidad, sino también en otros aspectos como la compatibilidad del vídeo de salida. Por ejemplo, como vemos en la figura 2, la mayoría de las cámaras de nuestros móviles usan el códec H.264 para comprimir vídeo de forma eficiente y generar archivos de alta compatibilidad, aunque también podemos usar el códec H.265, que proporciona una mayor compresión de vídeo, pero tiene una menor compatibilidad con los reproductores.

NOTA

¿Sabías que? La tecnología disponible en los comienzos del cine solo permitía gabar a unos 10 fps, por eso ¡en las películas de cine mudo percibimos los movimientos muy bruscos.

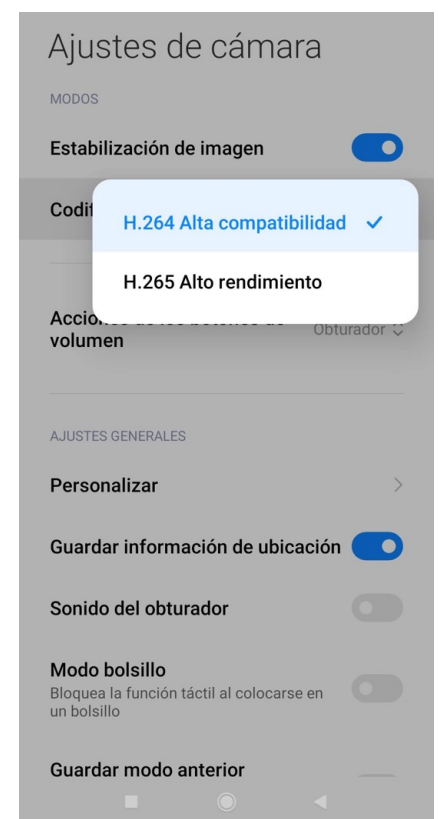


Figura 2. Se muestra el detalle de los ajustes de selección de códec para la cámara de un móvil.

**NOTA**

Recuerda que: además, en este proceso también podemos manipular el audio de nuestro vídeo. En este caso, algunos de los parámetros que podríamos ajustar serían la tasa de bits, el número de canales, la tasa de muestreo o el propio códec. Por supuesto, también podríamos eliminar el audio si no nos interesara para mejorar la compresión. No olvides consultar el documento **“Parámetros de ajuste para la compresión de audio”** para ampliar esta información.

**PARÁMETROS
DE AJUSTE PARA
LA COMPRESIÓN
DE AUDIO**

Documento referenciado:
A3C31A2D09

Saber más

Puedes ampliar esta información con las recomendaciones de la plataforma de vídeo Vimeo para comprimir vídeo y audio a través del siguiente enlace:

e.digitall.org.es/vimeo





Creación de
contenidos digitales

Nivel A2 3.1 Desarrollo
de contenidos

Parámetros de ajuste para la compresión de audio



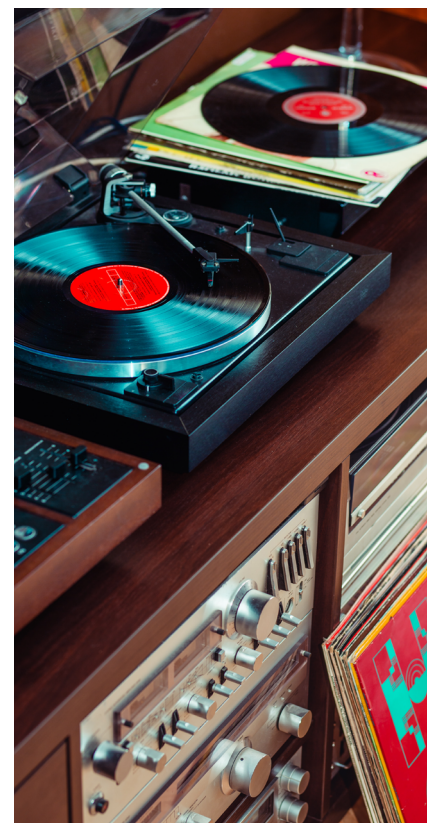


Parámetros de ajuste para la compresión de audio

Compresión de audio

La compresión de audio es un proceso que permite reducir el tamaño de un archivo de audio sin afectar significativamente su calidad. Esto se logra mediante el uso de ciertos parámetros de ajuste que permiten determinar qué partes del audio se mantienen y cuáles se descartan, sin afectar la calidad de sonido percibida por el oyente. Esto se logra mediante la eliminación de redundancias y la codificación de la señal de audio de una manera más eficiente.

Existen **diferentes algoritmos y técnicas de compresión de audio**, cada uno con sus propios parámetros de ajuste. Algunos de los parámetros más importantes en la compresión de audio son: la **tasa de bits**, la **frecuencia de muestreo**, el **tipo de codificación** y el **tipo de compresión**.



Tasa de bits

La **tasa de bits** se refiere a la cantidad de información que se utiliza para codificar el audio y tiene un impacto directo en la calidad del archivo comprimido. Cuanto mayor sea la tasa de bits, mejor será la calidad del audio, pero también aumentará el tamaño del archivo. La tasa de bits puede ser variable, lo que significa que se ajusta dinámicamente en función de la complejidad de la señal de audio, o fija, lo que significa que se utiliza una tasa de bits constante para toda la señal de audio. Una forma de lograr la compresión del audio es reduciendo la tasa de bits, lo que implica eliminar cierta información del audio original. Esto se puede hacer de forma imperceptible para el oído humano, lo que permite obtener archivos más pequeños sin sacrificar demasiado la calidad del sonido. Sin embargo, si se reduce demasiado la tasa de bits, la calidad del sonido puede verse afectada de forma perceptible. Por lo tanto, es importante encontrar un equilibrio entre la calidad del sonido y el tamaño del archivo al comprimir audio. Las tasas más utilizadas son la de 16 bits, que permite codificar un total de 65.536 niveles posibles para cada muestra, y la de 24 bits, que alcanza los 16.777.216 niveles.



Frecuencia de muestreo

La **frecuencia de muestreo** se refiere a la cantidad de veces que se toma una muestra de la señal de audio por segundo. Cuanto mayor sea, mejor será la calidad del sonido, pero también se necesitará más espacio para almacenar el archivo. La frecuencia de muestreo típica para un audio de calidad es de 44.1 kHz o 48 kHz (estándares introducidos con la estandarización del formato CD). Dado que la máxima frecuencia audible por el oído humano sano es de 20 kHz, una frecuencia de muestreo por encima de 40 kHz garantiza, en teoría, una calidad suficiente para que la señal no se perciba comprimida.

Tipo de codificación

El **tipo de codificación** se refiere a la forma en que se representa la información del audio en el archivo comprimido para que pueda ser transmitido o almacenado de manera eficiente. Algunos de los tipos de codificación más comunes son MP3, AAC y WAV. Cada uno de estos tipos de codificación tiene sus propias ventajas y desventajas en cuanto a calidad y tamaño del archivo. La elección del tipo de codificación dependerá de la finalidad de su uso, pues el tipo de codificación afecta directamente al tamaño del archivo resultante y la calidad del sonido.

Tipo de compresión

El **tipo de compresión** se refiere a la forma en que se lleva a cabo el proceso de compresión. Los tipos de compresión más comunes son la compresión con pérdida y la compresión sin pérdida.

- La **compresión sin pérdida** reduce el tamaño de un archivo de audio sin sacrificar la calidad del sonido. Un ejemplo de este tipo de compresión es el formato FLAC (*Free Lossless Audio Codec*), que permite reducir el tamaño de un archivo de audio sin pérdida de calidad.
- La **compresión con pérdida** reduce el tamaño de un archivo de audio sacrificando cierta calidad del sonido. Un ejemplo de este tipo de compresión es el formato MP3, que es muy común en la reproducción de música en dispositivos móviles y en la descarga de música en internet.





- La **compresión con pérdida adaptativa** es similar a la compresión con pérdida, pero permite ajustar el nivel de compresión en función de la complejidad de la señal de audio. Esto permite lograr una mejor calidad de sonido en señales con una mayor complejidad, como voces humanas, sin sacrificar demasiado la calidad de sonido en señales más simples, como el sonido de un instrumento. Un ejemplo de este tipo de compresión es el formato Ogg Vorbis.

La compresión con pérdida permite obtener archivos más pequeños, pero también puede causar una cierta pérdida de calidad en el audio. La compresión sin pérdida, por su parte, mantiene la calidad del audio original, pero no permite obtener archivos tan pequeños. Es conveniente encontrar un equilibrio en función de las necesidades.

Saber más

Si quieres saber si puedes distinguir entre compresión con y sin pérdida en distintos archivos de audio, te puede interesar el libro **Compresión de audio y vídeo** de Cliff Wootton. (Editorial Anaya Multimedia, ISBN: 978-84-415-1972-5).



Creación de
contenidos digitales

Nivel A2 3.1 Desarrollo
de contenidos

**Correcta
visualización
de medios
digitales en los
medios sociales
más populares**

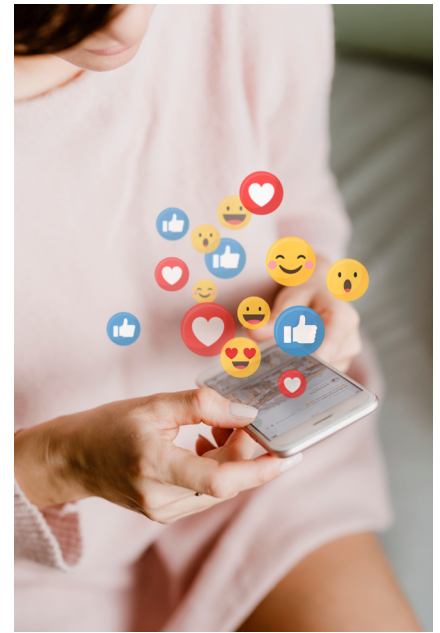




Correcta visualización de medios digitales en los medios sociales más populares

Introducción

En este documento vamos a repasar las características que deben cumplir los contenidos digitales que queramos compartir a las redes sociales más populares. En algunos casos, estas características serán de obligado cumplimiento, como el tipo de archivo aceptado, en otras, una recomendación a tener en cuenta para obtener una adecuada difusión del contenido y que otros usuarios interactúen con nuestro contenido adecuadamente.



Imagen

Empezaremos por las imágenes para realizar publicaciones como vemos en la siguiente tabla. Para otro tipo de publicaciones las recomendaciones son ligeramente diferentes, por ejemplo, para la sección de noticias de Facebook, la imagen que usemos puede ser mucho más pequeña o para las historias estará invertida.

Características recomendadas para las imágenes en publicaciones en las principales redes sociales

	Twitter	Facebook	Instagram
Resolución mínima recomendada	1200x1200 o 1200x698	1080x1080	1080x1080
Aspecto recomendado	1:1 o 1,91:1	1,91:1	1:1
Tipos de archivo	PNG o JPEG	PNG o JPEG	PNG o JPEG
Tamaño máximo	5 MB	30 MB	30 MB



Vídeo

Para los vídeos tenemos las recomendaciones descritas en la siguiente tabla. Debe tenerse en cuenta que estos datos no se ajustan a todos los formatos que ofrecen estas redes, por ejemplo, el tiempo máximo permitido de un vídeo en una historia de Instagram es de 15 segundos.

Características recomendadas para los vídeos en publicaciones en las principales redes sociales

	Twitter	Facebook	Instagram	YouTube
Resolución mínima recomendada	Desde 32x32 hasta: 1920x1200	1080x1080	1080x1080	Desde 426x240 hasta 7680x4320
Aspecto recomendado	Rango de 1:2.39 a 2.39:1	1:1 o 4:5 (solo móviles)	9:16	16:9
Tipos de archivo	MP4 o MOV	MP4, MOV o GIF	MP4, MOV o GIF	MP4 (recomendado)
Tamaño máximo	512 MB	4 GB	250 MB	256 GB
Duración máxima	2:20 min	240 min	60 min	12 horas

NOTA

Recuerda que: ya hemos trabajado los motivos por los cuales existen estas limitaciones de tamaño (y duración en el caso de los vídeos). Puedes refrescar esta información en el vídeo **Contenidos digitales para su publicación e internet**.



CONTENIDOS DIGITALES PARA SU PUBLICACIÓN EN INTERNET

e.digitall.org.es/A3C31A1V11

Audio

Y finalmente nos encontramos con las recomendaciones para compartir audio recogidas en la siguiente tabla.

Características recomendadas para los archivos de audio en las principales plataformas de contenido

	YouTube	iVoox	Spotify
Tasa de bits recomendada	64 kbps (mínima)	16 kbps (mínima)	-
Tipos de archivo	MP3/WAV (MP3), WAV (PCM), MOV (ACC) o FLAC	MP3, M4A o WAV	MP3, M4A, WAV, MPG, MOV o MP4
Tamaño máximo	256 GB	300 MB	250 MB



i Saber más

En la siguiente guía podrás ver una comparativa entre distintos micrófonos disponibles en el mercado:

e.digitall.org.es/microfonos





DigitAll

Creación de
contenidos digitales

3.2

INTEGRACIÓN Y REELABORACIÓN DE CONTENIDO DIGITAL





Creación de
contenidos digitales

Nivel A2 3.2 Integración y reelaboración
de contenido digital

Creación de presentaciones efectivas





Creación de presentaciones efectivas

En el vídeo **A3C32A2V02 “Integración de contenidos digitales”** hemos visto cómo combinar contenidos digitales en presentaciones. En este documento vamos a aprender algunas pautas para hacerlas más efectivas.

El primer paso para preparar una presentación es responder a una serie de **preguntas previas: ¿Qué pretendo transmitir?, ¿Qué le interesa a la audiencia? y ¿Cuánto tiempo tengo?** Estas son tres cuestiones vitales para **organizar y seleccionar** con éxito el contenido de la presentación.

Piensa en el tema y determina su **interés** para la audiencia. De este análisis selecciona el **mensaje** a transmitir. Organiza tus ideas y focalízate en aquello que la **audiencia quiere** o **necesita saber** sobre el tema en cuestión, dejando fuera de la presentación los contenidos más generales.

Con estas ideas, crea una **historia**. Utiliza un **hilo conductor** para poner en una secuencia las ideas que pretendes transmitir y conéctalas. Las historias son **agradables de escuchar y sencillas de recordar**, por lo que si consigues elaborar una a partir de tus ideas, conseguirás que tu comunicación sea **efectiva**.

No olvides que toda buena historia debe tener un buen **final**. Por eso, asegúrate de cerrar tus presentaciones con un **resumen, conclusiones** o incluso alguna **reflexión personal** que pueda dejar abierto el final de tu historia.

Dicho lo anterior no es suficiente con tener **una buena historia**. También hay que **contarla bien**. Una presentación **complementa** al conferenciante, aportándole a este tipo de acto teatral el **escenario** donde se desarrolla la historia. La presentación define el hilo conductor de las ideas. Las ideas fluyen una tras otra y una o varias diapositivas transmiten una idea. Nunca se deberían mezclar varias ideas en una diapositiva.

Los elementos esenciales que contiene las diapositivas son texto, imágenes y videos. A continuación listamos algunas recomendaciones de carácter general que pueden ser tenidas en cuenta para diseñar presentaciones efectivas. Las organizamos por tipo de contenido.





Recomendaciones sobre el texto

- 1 | Usa muy poco texto.** El público no debe estar leyendo las diapositivas mientras intenta escuchar lo que estás diciendo. El texto debe enfatizar, es la negrilla de la palabra hablada. Se debe emplear frases directas y cortas. Un fallo común en el diseño de las presentaciones es la **sobrecarga cognitiva** causada por un exceso de información, habitualmente por una cantidad de texto mayor que el que se es capaz de leer y procesar.
- 2 | No leas** la presentación. La presentación debe ser una conversación con la audiencia.
- 3 | Usa nuevas fuentes** y que el **tamaño de la fuente** sea lo suficientemente **grande** para leerse desde cualquier punto de la sala. Se debe usar fuentes fáciles de leer como la tipografía San Serif (las letras carecen de terminaciones o remates). Ejemplos de esta familia son Gill Sans, Arial, Calibri, Futura, Helvética, Tahoma o Verdana. No te limites a la Arial o Calibri.
- 4 | Emplea colores del texto que contrasten** con el fondo.
- 5 | Formula algunas informaciones** en forma de **preguntas**. Esto activa una parte diferente del cerebro para mantener el interés del público.
- 6 | Cuidado con las erratas.** El medio es el mensaje. Se transmite descuido e improvisación.
- 7 | Como carácter general busca la máxima legibilidad** y la diapositiva estará terminada cuando no se pueda eliminar ningún elemento de la misma.



Recomendaciones sobre imágenes

- 1 | Imágenes simples** para explicar **conceptos**. El dicho popular que *una imagen vale más que mil palabras* sintetiza un motivo fundamental del uso de imágenes. Permite una economía en el lenguaje. Pero para que la imagen funcione su diseño debe ser simple de modo que se minimice el ruido visual. El fallo común es dar demasiada información para procesar también está presente en la inclusión de gráficos.
- 2 |** Que transmita **emociones**. Otra finalidad es la de apuntalar el relato. Una estructura que funciona en la presentación es la introducción de un problema, de un conflicto, de un contraste, de un “malo”. Las imágenes son la forma más directa de llevar estas emociones.
- 3 |** Usar imágenes a **pantalla completa** y con **alta resolución**. Un fallo habitual es presentar varias imágenes en una diapositiva.
- 4 |** Que sean **significativas**. Existen librerías de imágenes que ponen a la libre disposición de los usuarios, sin coste alguno, muchas imágenes digitales para que sean utilizadas con cualquier fin. No se requiere ningún permiso. Hay que emplear tiempo en seleccionar las imágenes adecuadas a la idea que se desea transmitir, buscando la analogía y/o el contraste.
- 5 | Combinar imágenes** con **texto**. Es la forma más efectiva. Se estima que la comunicación oral tiene una tasa de transmisión de un 10%, la imagen de un 35% y la suma de oral más imagen alcanza el 65%.





Recomendaciones sobre videos

1 Deben ser **significativos**. El objetivo del uso de un video es ilustrar o destacar un concepto. El conferenciante debe tener nítidamente claro que el video es imprescindible en el relato. Que merece la pena ceder la voz a otro para que prosiga con la historia. Hay que tener cuidado que el otro (el video) no se apropie de la historia y se las lleve a sitios no pensados por el presentador.

2 Emplearlo para **conceptos difíciles** que hayan sido abordado con éxito.

3 Al igual que el texto exige la legibilidad el video exige que se vea y escuche bien. Hay que revisar los elementos tecnológicos, la sala y la propia calidad del video.



INTEGRACIÓN DE CONTENIDOS DIGITALES

e.digitall.org.es/A3C32A2V02

Saber más

Si quieres profundizar en la planificación, estructura, diseño y exposición de presentaciones te recomendamos que consultes las siguientes referencias:

- **El arte de presentar.** Gonzalo Álvarez Marañón. Ediciones Gestión. ISBN: 9788498752205.
- **Slide:ology.** Nancy Duarte. O'Reilly Media, Inc. ISBN: 9780596522346.



Creación de
contenidos digitales

Nivel A2 3.2 Integración y reelaboración
de contenido digital

Representación gráfica de los datos incluidos en una hoja de cálculo





Representación gráfica de los datos incluidos en una hoja de cálculo

En el **documento A3C32A1D02 “Hojas de Cálculo: representación y cálculo con datos”** se introdujeron las hojas de cálculo como herramientas software que permiten la visualización y el cálculo con datos. En esta sección se plantea un caso práctico de utilización de una hoja de cálculo. Este ejemplo ilustra su uso para la gestión de las **calificaciones de una asignatura**, con tres objetivos principales. El primero es la automatización de los cálculos para obtener la calificación final, el segundo es la elaboración de un listado de calificaciones para comunicar los resultados a los alumnos y el tercero es facilitar la reflexión del docente sobre el propio proceso de evaluación a través de la visualización de los resultados. Estos tres objetivos definen la estructura del **libro** a utilizar. La **primera hoja** está destinada a almacenar los **datos en bruto**, la **segunda hoja** recoge el listado de calificaciones que se publicará entre los alumnos, por lo que contiene los **datos elaborados** y se añadirán dos nuevas hojas al libro en las que se situarán sendos **gráficos**, visualizando los resultados del proceso de evaluación. En el **documento A3C32A1V03 “Generación de gráficos a partir de un conjunto de datos”** tienes un breve recorrido por los gráficos que puedes emplear.

El docente dispone de un listado de alumnos con los ítems de evaluación: i) prueba 1, ii) prueba 2, iii) trabajos, iv) examen de prácticas y v) dossier de prácticas. Estos datos constituyen los **datos brutos** situados en la primera hoja, denominada **DATOS_BRUTOS**, con un aspecto como el mostrado en la Tabla 1. Para ahorrar espacio la tabla contiene solamente dos registros de los ciento cincuenta y dos registros existentes.

Tabla 1: Hoja 1 – DATOS_BRUTOS

NOMBRE	APELLIDOS	PRUEBA 1 (SOBRE 50)	PRUEBA 2 (SOBRE 20)	TRABAJOS	EXAMEN PRÁCTICAS	DOSSIER PRÁCTICAS
ELENA	GARCÍA ROMÁN	47	20	10	7,2	10
JUAN	MEGÍA PEREA	22	8	7	7	5



HOJAS DE CÁLCULO: REPRESENTACIÓN Y CÁLCULO CON DATOS

Documento referenciado:
A3C32A1D02



GENERACIÓN DE GRÁFICOS A PARTIR DE UN CONJUNTO DE DATOS

Documento referenciado:
A3C32A1V03



La segunda hoja, denominada **LISTADO**, emplea **fórmulas** y los datos brutos para obtener los datos elaborados mostrados en la Tabla 2. Se observa que aparece una nueva columna EX. TEORIA donde se almacena la nota de la parte de teoría de la asignatura obtenida a partir de las Pruebas 1 y 2.

Tabla 2: Hoja 2 - LISTADO

NOMBRE	APELLIDOS	EX. TEORIA	TRABAJOS	EXAMEN PRÁCTICAS	DOSSIER PRÁCTICAS	NOTA FINAL
ELENA	GARCÍA ROMÁN	9,7	10	7,2	10	9,57
JUAN	MEGÍA PEREA	4,2	7	7	5	5,1

La celdilla C2 (EX. TEORIA) de la alumna ELENA GARCÍA ROMÁN se calcula con la fórmula:

```
=(DATOS_BRUTOS!C2/5+DATOS_BRUTOS!D2/2)/2
```

Esta expresión convierte a una escala sobre 10 las puntuaciones obtenidas en las Pruebas 1 y 2, que estaban evaluadas respectivamente en una escala de 50 y 20. Posteriormente se calcula la media aritmética.

La celdilla D2 de la misma alumna se calcula tomando el dato de la misma celdilla, pero recogido en la hoja de DATOS_BRUTOS. Esto se expresa mediante la fórmula:

```
=DATOS_BRUTOS!D2
```

Finalmente, la NOTA FINAL se calcula como una media ponderada de los distintos ítems de evaluación mediante la fórmula $=0,5*C2+0,15*D2+0,1*E2+0,25*F2$. ¡Notar que como los datos se encuentran en la misma hoja se ha omitido la referencia la hoja DATOS_BRUTOS!

Las siguientes dos hojas están destinadas a la visualización de los resultados. El primer objetivo es analizar el proceso de evaluación. De entre los distintos gráficos disponibles se debe elegir aquellos tipos que agreguen la información. Como la tabla consta de más de un centenar de alumnos, entonces los gráficos de columna, barras, líneas, circulares



no son adecuados porque representaría individualmente las calificaciones de todos los alumnos. Se ha elegido un gráfico de tipo histograma porque permite juntar los alumnos cuya calificación cae en un determinado intervalo. Se han considerado 5 intervalos. En el eje Y se representa el número de alumnos en dicho intervalo. El resultado se muestra en la Ilustración 1. El gráfico facilita la comprensión del fenómeno al hacer patrones de alumnos y cuantificar su importancia. Este gráfico le permite al profesor medir cuantos alumnos abandonan la asignatura (intervalo $[0,2]$), suspenden (intervalo $[2,4]$), alumnos de especial atención (intervalo $[4,6]$) o alumnos que aprueban con solvencia la asignatura (intervalos $[6,8]$ y $[8,10]$).

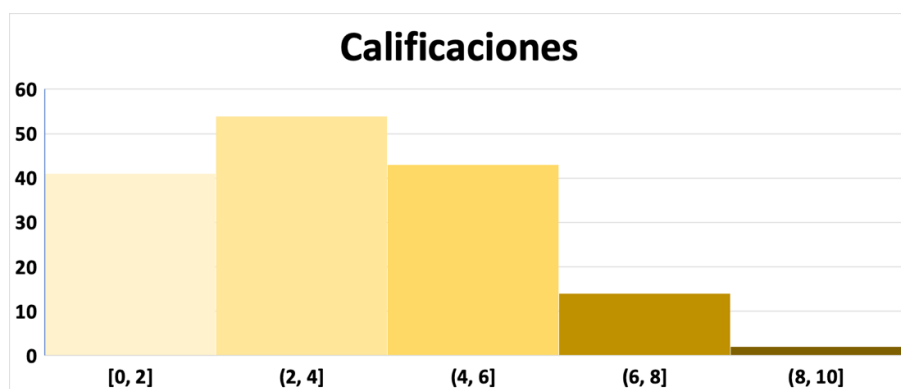


Ilustración 1. Histograma de la variable NOTA FINAL.

El siguiente objetivo es analizar el efecto de cada ítem de evaluación. Por este motivo se grafica simultáneamente todos los ítems de evaluación para que se puedan comparar entre sí. Se ha empleado el diagrama de bigotes porque permite visualizar el rango, la media, la mediana y sobre todo como se distribuye la parte central de la clase. Los rectángulos marcan el rango de la mitad de la clase, eliminados el 25% con mejores y peores resultados. De esta forma se eliminan los valores extremos, ya sean por ser demasiado grandes o por ser demasiado pequeños, ya que pueden influir en el análisis. El gráfico resultante se muestra en la Figura 2. Se observa que los ítems que más discriminan son el DOSSIER PRÁCTICA y el EXAMEN DE TEORIA (el rectángulo tiene una mayor amplitud y por tanto contiene más variedad de notas). Se observa también que el lado superior del rectángulo amarillo se alinea con las calificaciones más bajas. Esto quiere decir que este ítem de evaluación asigna calificaciones más bajas que el



resto de las pruebas evaluación. Por tanto, si hubiese que mejorar el rendimiento del proceso de evaluación se debería de incidir en los contenidos teóricos de la asignatura y su evaluación (EX. TEORIA).

Cómo conclusión, la visualización de los datos permite obtener información de estos. La visualización actúa como una capa entre los datos y el usuario de modo que se facilita la **interpretación** y la **evaluación** de los resultados. En este ejemplo hemos detectado y medido cuantos alumnos serían susceptibles de mejorar cualitativamente su resultado académico y detectado el ítem de evaluación clave sobre el que se debe actuar.

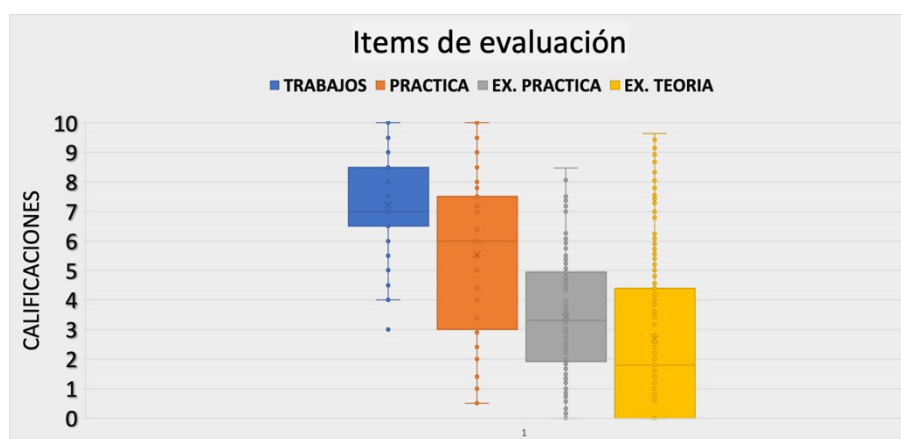


Ilustración 2. Diagrama de bigotes de los diferentes ítems de evaluación.



GENERACIÓN DE GRÁFICOS A PARTIR DE UN CONJUNTO DE DATOS

Vídeo referenciado: **A3C32A1V03**

Saber más

Análisis eficaz de datos. Pierre Rigollet. Ediciones ENI.
ISBN: 978-2-409-03154-0

ATENCIÓN

Tanto en el ámbito particular como en el empresarial, la **productividad** es esencial para la optimización del tiempo.

Las hojas de cálculo te permitirán organizar, editar, analizar y visualizar tus datos. Tanto a nivel personal como profesional una buena **estructuración de los datos** brutos y elaborados, vislumbrando desde el inicio su relación en la organización, planificación, dirección y control en tu negocio o proyecto, incrementará sustancialmente la productividad.



Creación de
contenidos digitales

Nivel A2 3.2 Integración y reelaboración
de contenido digital

Adquisición y uso de contenidos existentes





Adquisición y uso de contenidos existentes

En el **vídeo A3C32A2V04 “Fuentes de imágenes, vídeos y audios”** hemos aprendido que existen diferentes plataformas en la red para la adquisición gratuita o de pago de contenidos digitales que podemos usar en nuestros proyectos. En este documento vamos a aprender cómo podemos descargar contenido de una de estas plataformas para poder incorporarlo en plataformas de edición.

Para ello, a modo de ejemplo, veamos el proceso necesario para obtener contenidos sobre una de estas plataformas explicadas en el vídeo. En concreto, nos centraremos en la plataforma **Pixabay** (pixabay.com/es), por ser una de las plataformas gratuitas más utilizadas en la actualidad. No obstante, el proceso seguido será muy similar en el resto de las plataformas, tanto gratuitas como de pago.

El primer paso, si no lo hemos hecho ya, será registrarnos en la plataforma seleccionada. Para ello, buscaremos el botón de registro y crearemos una cuenta con nuestro correo electrónico y una contraseña única para esta página web. La cuenta creada será completamente gratuita y nos permitirá descargar cualquier contenido de la página, así como llevar un registro del contenido descargado o incluso subir nuestro propio contenido a la plataforma.

A continuación, buscaremos la pieza deseada. Para ello, en el cuadro de búsqueda introduciremos las palabras clave que describan el contenido deseado y seleccionaremos el tipo de contenido que busquemos: Imágenes, Vídeos, Música, Efectos de sonido o GIF. En este punto es interesante probar con distintas combinaciones de palabras en el buscador hasta encontrar el elemento que más se adecue a nuestras necesidades. En muchos casos, puede ser necesario introducir la búsqueda en inglés, ya que estas plataformas tienen más cantidad de contenido en este idioma. Retomando el ejemplo del vídeo anterior, vamos a introducir en el cuadro de búsqueda “Estación de tren” o “Train station” y seleccionaremos “Vídeos” en el selector de tipo de contenido.



FUENTES DE
IMÁGENES, VÍDEOS
Y AUDIOS

e.digitall.org.es/A3C32A2V04





Una vez elegido el contenido deseado, realizamos clic sobre el para abrirlo. La plataforma nos mostrará la información técnica del vídeo, como por ejemplo: El autor, el tipo de licencia, formato, fecha de publicación, resolución (si es una imagen o vídeo), duración (si es un vídeo o sonido) e incluso en algunas plataformas el número de descargas. Por ejemplo, imaginemos que elegimos el siguiente vídeo: e.digitall.org.es/video-escalera.

En este punto, es imprescindible revisar si la información técnica de la pieza se corresponde con el contenido deseado, así como revisar la licencia de esta. Si la pieza se adecua a nuestras necesidades, podremos seccionar el botón de descarga (en las plataformas gratuitas) o realizar la compra de la licencia de la pieza (en las plataformas de pago) y posteriormente descargarla. Antes de iniciar la descarga, se nos pedirá seleccionar la resolución deseada para la pieza, ya que este tipo de plataformas suelen ofrecer distintas resoluciones para las piezas de imagen o vídeo. Una vez realizada la descarga está se guardará en la carpeta de descargas de nuestro ordenador.



FUENTES DE IMÁGENES, VÍDEOS Y AUDIOS

e.digitall.org.es/A3C32A2V04

⚠ ATENCIÓN

No olvides revisar la licencia del contenido que vayas a utilizar. Aunque el contenido descargado de plataformas gratuitas suele tener una licencia permisiva, es recomendable revisarla antes de descargar la foto, ya que puede ser necesario atribuir la autoría de la pieza. Además, algunas licencias no permiten el uso de los contenidos para fines comerciales. Te recomendamos que en este punto revises el material de "Propiedad intelectual de contenidos digitales".

👁 NOTA

Es recomendable ser metódico y tener un flujo de trabajo bien definido a la hora de trabajar con el contenido descargado de estas plataformas. Una buena idea es almacenar las piezas descargadas de manera organizada en varias carpetas o subcarpetas por cada proyecto. Por ejemplo, podemos definir un primer nivel de carpetas con el tipo de contenido: Fotografías, Vídeo, Efectos de sonido, etc. y otro segundo nivel con la temática de cada una de las piezas. De esta manera será mucho más fácil encontrar las piezas después e incorporarlas rápidamente a nuestros proyectos.

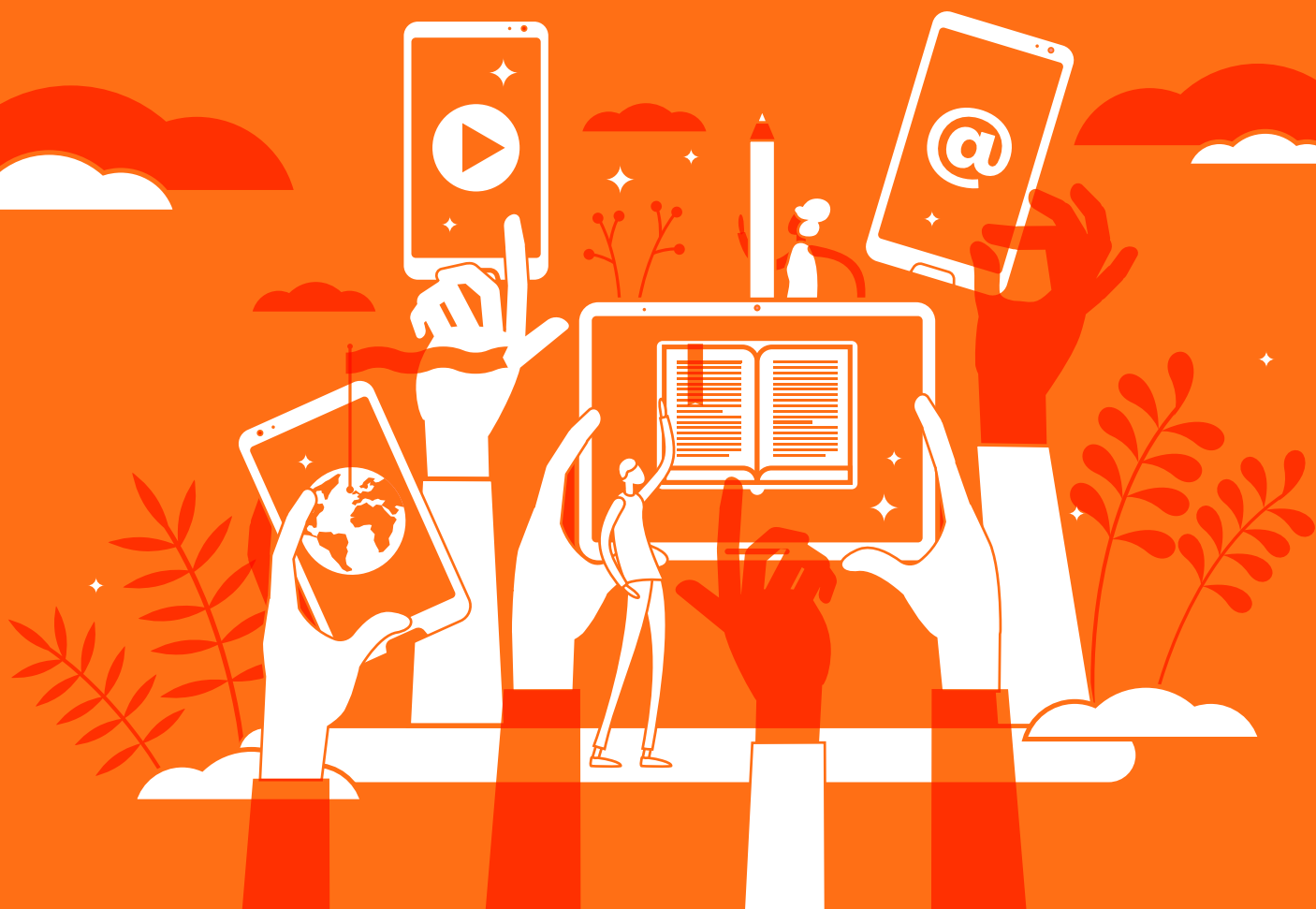


DigitAll

Creación de
contenidos digitales

3.3

DERECHOS DE AUTOR Y LICENCIAS DE PROPIEDAD INTELECTUAL





Creación de
contenidos digitales

Nivel A2 3.3 Derechos de autor y licencias
de propiedad intelectual

Licencias Creative Commons





Licencias Creative Commons

Los autores de una obra son los propietarios de todos los derechos sobre dicha obra, y cualquier persona física o jurídica que desee utilizar, modificar, distribuir o vender la obra está obligado a ser autorizado por el autor.

Mediante las Licencias Creative Commons (LCC), los autores u otros titulares de derechos sobre una obra autorizan y/o ceden algunos de los derechos sobre dicha obra bajo ciertas condiciones, pudiendo mantener otra parte de dichos derechos.



Las LCC se pueden aplicar a cualquier tipo de contenidos creativos, como fotografías y música, y también a las bases de datos. Sin embargo, las LCC no son de habitual aplicación al software y al hardware.

Por lo tanto, cada licencia Creative Commons (CC) indica qué derechos cede el autor, bajo qué condiciones y qué pueden hacer los usuarios con la obra respecto a su posible distribución, modificación o venta. Para ello, Creative Commons considera cuatro condiciones básicas que se combinan en sus licencias:

- Reconocimiento, obliga a que todas las licencias deben indicar la referencia al autor original, ya que es su derecho moral irrenunciable.
- Compartir Igual, que permite crear obras derivadas bajo la misma licencia o similar a la obra original.
- No Derivadas, que no permite modificar la obra original de ninguna manera.
- No Comercial, que prohíbe que la obra sea utilizada con fines comerciales directos o indirectos.

Su combinación genera seis Licencias Creative Commons diferentes, entre las que el autor puede elegir para permitir su uso por terceros. En orden de menos restrictiva a más restrictiva, las seis LCC son las siguientes:



- **Reconocimiento:**

La Licencia BY permite **compartir** (copiar y redistribuir) y **adaptar** (remezclar, transformar y construir a partir del material), incluso para fines comerciales; siempre que **se reconozca la autoría de la obra original** de forma adecuada.



- **Reconocimiento + Compartir Igual:**

La Licencia BY-SA permite **compartir** (copiar y redistribuir) y **adaptar** (remezclar, transformar y construir a partir del material), incluso para fines comerciales; siempre que **se reconozca la autoría de la obra original** de forma adecuada y **se licencie la nueva creación** (en su caso) **bajo condiciones idénticas**.



- **Reconocimiento + Sin Obra Derivada:**

La Licencia BY-ND permite **compartir** (copiar y redistribuir), incluso para fines comerciales; siempre que **se reconozca la autoría de la obra original** de forma adecuada y **no se distribuyan modificaciones de la obra original**.



- **Reconocimiento + No Comercial:**

La Licencia BY-NC permite **compartir** (copiar y redistribuir) y **adaptar** (remezclar, transformar y construir a partir del material); siempre que **se reconozca la autoría de la obra original** de forma adecuada y **no se utilice para fines comerciales**.



- **Reconocimiento + No Comercial + Compartir Igual:**

La Licencia BY-NC-SA permite **compartir** (copiar y redistribuir) y **adaptar** (remezclar, transformar y construir a partir del material); siempre que **se reconozca la autoría de la obra original** de forma adecuada, **no se utilice para fines comerciales**, y **se licencie la nueva creación** (en su caso) **bajo condiciones idénticas**.



- **Reconocimiento + No Comercial + Sin Obra Derivada:**

La Licencia BY-NC-ND permite **compartir** (copiar y redistribuir); siempre que **se reconozca la autoría de la obra original** de forma adecuada, **no se utilice para fines comerciales**, y **no se distribuyan modificaciones de la obra original**. Es la licencia más restrictiva de las seis.





En definitiva, todas las LCC sin excepción incluyen la condición BY, que exige el reconocimiento del autor y la fuente.

Cada LCC es representada por medio de las siguientes figuras gráficas para su fácil identificación:

A continuación, se presenta una tabla-resumen de los seis tipos de LCC y cuáles son los permisos y las condiciones que se conceden. Hacemos uso de esta tabla indicando que está regulada por una Licencia Creative Commons de tipo BY. Por lo tanto, estamos autorizados a compartirla (copiarla y redistribuirla) siempre que indiquemos, como hacemos a continuación, que ha sido elaborada por los autores Ulises Hernández y Marcela Hernández.

Permisos y condiciones en Licencias

		Usted es Libre de...			Siempre y cuando...		
		Acceder	Distribuir	Modificar	Vender	Reconozca	Licencie Igual
+Abierto +Cerrado		✓	✓	✓	✓	✓	
		✓	✓	✓	✓	✓	
		✓	✓	✓	✓	✓	✓
		✓	✓	✓	✗	✓	
		✓	✓	✓	✗	✓	✓
		✓	✓	✗	✓	✓	
		✓	✓	✗	✗	✓	
	✓	✗	✗	✗	✓		

<http://creativecommons.org/licenses/?lang=es>

Ulises Hernandez y Marcela Hernandez
v2 Febrero de 2015

<http://karisma.org.co/>

Podrás complementar esta información en el video “*Licencias Creative Commons*” de este mismo nivel, así como en:

e.digitall.org.es/creative-commons

**LICENCIAS
CREATIVE COMMONS**

e.digitall.org.es/A3C33A2V02



Creación de
contenidos digitales

Nivel A2 3.3 Derechos de autor y licencias
de propiedad intelectual

Incumplimientos de la propiedad intelectual





Incumplimiento de la propiedad intelectual

La utilización de obras y creaciones digitales, literarias, musicales, fotográficas, cinematográficas, etc. **sin autorización** de sus autores o titulares de derechos de propiedad intelectual puede suponer la vulneración de esos derechos, y provocar un importante menoscabo económico que redundará en la desaparición de empresas y en la destrucción de puestos de trabajo. Asimismo, esa vulneración de derechos de propiedad intelectual es un acto de competencia desleal, notablemente perjudicial para el desarrollo de la industria cultural legal. Ante dichas vulneraciones de derechos, el titular de los derechos o su representante legal pueden ejercitar las acciones civiles (de reparación de daños y perjuicios, etc.) y penales previstas en la ley contra estos delitos, teniendo graves consecuencias para el infractor de ese incumplimiento. Podemos distinguir tres tipos de delitos:



- 1 | Delito de tipo básico:** es el acto de fabricar, importar, almacenar o comercializar obras protegidas por derechos de autor sin contar con la autorización expresa del titular. Es más grave si existe ánimo de lucro. Las penas pueden llegar a ser de prisión de 6 meses a 3 años.
- 2 | Delito contra la propiedad intelectual online:** se produce cuando, a través de internet, alguien sube, descarga o facilita el acceso a una obra protegida sin contar con la autorización expresa del titular. Las penas pueden llegar a ser de prisión de 6 meses a 3 años. La normativa europea obliga desde 2019 a que las plataformas de contenidos dispongan de herramientas para bloquear contenidos plagiados, duplicados o protegidos por derechos de autor.
- 3 | Delitos contra la distribución o comercialización ambulante u ocasional:** se produce cuando alguien vende o alquila ejemplares protegidos por derechos de autor sin contar con la autorización expresa del titular de forma ocasional, siendo este por ejemplo el caso de los famosos "manteros". La pena prevista para este delito es de multa de seis a doce meses.



No obstante, existe la posibilidad de realizar **copias privadas**: se permite a una persona realizar la grabación o copia de una obra ya divulgada para uso privado sin ánimo de lucro siempre que haya tenido acceso legítimo al original. Su existencia surgió a petición de los fabricantes de equipos (reproductores, grabadores) y soportes (CD, DVD, tarjetas almacenamiento) para la copia de material protegido con el fin de permitir su venta. A cambio, todos pagamos un canon compensatorio a los titulares de derechos de propiedad intelectual. La copia privada **excluye los programas informáticos, el software y las bases de datos**. No se debe confundir la copia privada con la copia de seguridad de los programas informáticos. Por otra parte y desde el auge de las tecnologías digitales se vienen desarrollando las denominadas medidas tecnológicas de protección. En determinados casos, el único medio práctico para impedir la copia de obras es recurrir a sistemas de protección o de gestión como, por ejemplo, la utilización de dispositivos técnicos que impiden la copia o consiguen que la calidad de ésta sea tan mala que no puedan ser utilizadas. Otro ejemplo son los medios técnicos para impedir la recepción de programas de televisión comerciales codificados para los que se precisa la utilización de mecanismos de descodificación.

Podrás complementar esta información en el video **"Introducción a derechos de autor y licencias"** de este mismo nivel, así como en la normativa de referencia:

- **Texto Refundido de la Ley de Propiedad Intelectual** (e.digitall.org.es/ley-propiedad)
- **Código Penal** (e.digitall.org.es/codigo-penal)



**INTRODUCCIÓN
A DERECHOS
DE AUTOR Y LICENCIAS**

e.digitall.org.es/A3C33A2V01

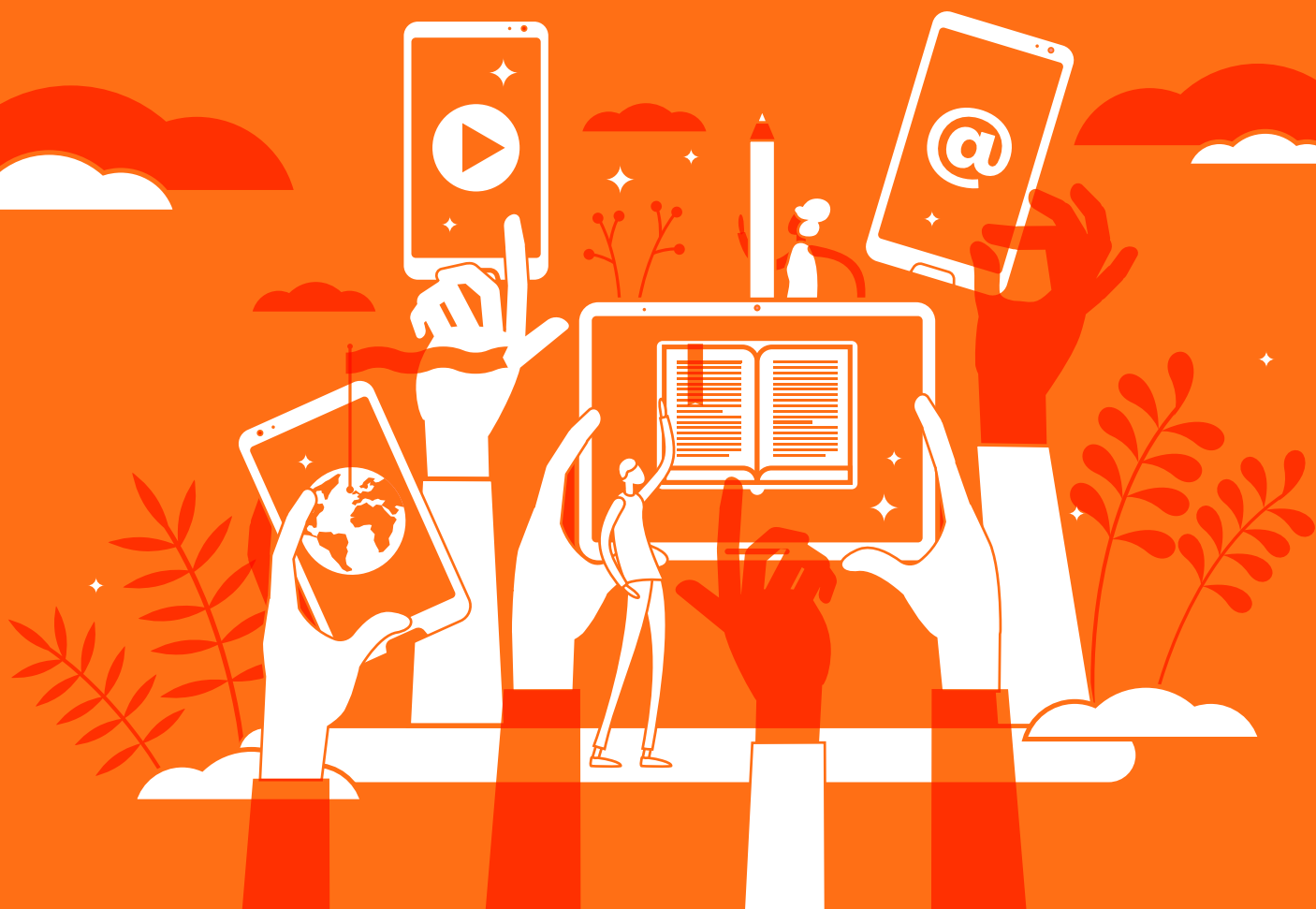


DigitAll

Creación de
contenidos digitales

3.4

PROGRAMACIÓN





Creación de
contenidos digitales

Nivel A2 3.4 Programación

Técnicas de construcción de algoritmos



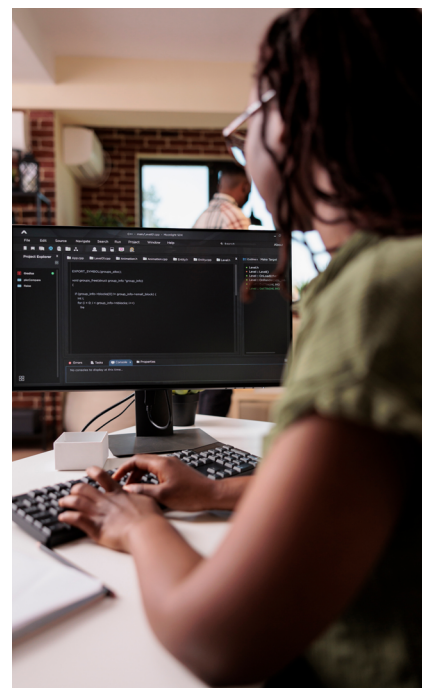


Técnicas de construcción de algoritmos

Una de las mayores dificultades a la hora de resolver un problema con un ordenador no es la escritura del programa sino el paso previo: la construcción del algoritmo, es decir su diseño y su definición. Para ello, no basta con conocer las herramientas que existen para su especificación (como el pseudocódigo o los diagramas de flujo) sino que es indispensable identificar el cuerpo del algoritmo, esto es, los datos de entrada y de salida, y la secuencia de instrucciones que hay que realizar para obtener los resultados deseados. El principal inconveniente es que no se pueden aplicar fórmulas ni recetas por lo que se requieren ciertas dosis de creatividad para buscar soluciones. No obstante, podemos beneficiarnos de algunas técnicas o estrategias que facilitan la tarea cuando la solución del problema no es trivial ni sencilla.

La primera de ellas es la del uso de la **abstracción** para que, de forma jerárquica, y mediante refinamientos sucesivos, se pueda descomponer un problema en subproblemas. De este modo, según evoluciona el diseño del algoritmo, cada nivel de su estructura representa un refinamiento en el nivel de abstracción. Por ejemplo, si salimos de ver una película del cine y nos encontramos con alguien que nos pregunta de qué iba la película, le haremos un resumen de la misma. En ese momento, se está realizando un proceso de abstracción porque se destacan las principales ideas del argumento y se ignoran detalles de escenas concretas. Si después nos pregunta por una subtrama o similar, profundizaremos en ella, dando más detalles de la subtrama e ignorando las demás. De esta forma, vamos contando la película mediante un proceso “jerárquico”.

En el contexto de la programación, la abstracción puede ser de distintos tipos: la evolución de los lenguajes de programación depende del nivel de abstracción del ordenador; y los paradigmas de programación surgen al aplicar la abstracción a los datos y/o a las instrucciones. Estos temas se tratan con más detalle posteriormente. Ahora es importante describir cómo hacer uso de la abstracción para realizar la descomposición de un problema complejo en subproblemas más simples. De esta forma, se puede abordar la resolución de cada uno de ellos de forma independiente de



⚠ ATENCIÓN

Abstraer es un proceso mental para entender procesos complejos que consiste en:

- **Destacar** los detalles relevantes del objeto en estudio.
- **Ignorar** los detalles irrelevantes del objeto en ese momento.



los demás, de forma similar a cuando cuentas una película. Precisamente, esta estrategia consistente en descomponer un problema en subproblemas es la base de la técnica “Divide y Vencerás”, utilizada para el diseño de algoritmos “rápidos” de ordenación de multiplicación de enteros grandes o para multiplicar matrices. Un poco más adelante se explica con más profundidad esta estrategia.

Otra técnica, muy útil para resolver problemas de optimización, esto es, aquellos en los que se pretende maximizar o minimizar algún aspecto de estos, es la técnica **voraz** (o *greedy*, en inglés). Un ejemplo típico en el que se suele aplicar esta técnica en la vida real es el de la devolución de cambio con el menor número de monedas posible. La estrategia voraz consiste en elegir lo mejor en cada momento, en base a la información de la que se dispone, de forma que las decisiones son inamovibles. Por ejemplo, en el caso de la devolución del cambio, en cada momento se elige la moneda de mayor valor disponible que no supere la cantidad a devolver. Para poder aplicar la estrategia voraz, la solución del problema debe cumplir dos requisitos: ser un subconjunto de un conjunto de elementos (en la devolución del cambio, la solución es un subconjunto de las monedas disponibles) y poder construirse por etapas (la elección de una moneda constituye una etapa). En general, los algoritmos voraces suelen seguir el esquema que se ilustra en el listado siguiente:

Algoritmo Voraz**Datos**

Candidatos: Conjunto de elementos

S: solución C: Elemento

InstruccionesS $\leftarrow \emptyset$ **Mientras** Candidatos $\neq \emptyset$ **Y** S NoEsSolución **Hacer**C \leftarrow ElegirMejorCandidato de los CandidatosCandidatos \leftarrow Candidatos $\setminus \{C\}$ **Si** C puede incluirse en SEntonces S \leftarrow S $\cup \{C\}$ **Fin_Si****Fin_Mientras****Devolver** S**Fin_Voraz**



Como ves, en el algoritmo se distinguen una serie de elementos:

- **Objetivo** para optimizar
- **Candidatos** a formar parte de la solución
- Candidatos **seleccionados**
- Criterio de **solución**, que decide si los elementos seleccionados constituyen la solución al problema
- Criterio de **factibilidad**, que decide si un candidato se puede incluir en la solución
- Criterio de **selección**, que elige el mejor candidato de los posibles.

En el ejemplo de la devolución del cambio, el objetivo es minimizar el número de monedas que forman parte de la solución; los candidatos son las monedas de las que disponemos; el criterio de solución es que el valor de las monedas seleccionadas coincida con el valor del cambio; el de factibilidad es el no considera las monedas que, al incluirlas en la solución, superan el valor del cambio; y el criterio de selección elige la moneda de mayor valor entre las disponibles. Esta estrategia es óptima con nuestro sistema de monedas. Sin embargo, si las monedas de 10 céntimos fueran de 11, por ejemplo, y no existieran las monedas de 2 céntimos, la estrategia no es óptima. Por ejemplo, para devolver 15 céntimos, usaría 1 de 11 y 4 de 1, 5 monedas en total, mientras que la mejor solución sería devolver 3 monedas de 5.

Los algoritmos voraces suelen ser muy sencillos y rápidos (complejidad polinomial). Por eso, aunque la solución óptima no está garantizada en algunos casos, se utilizan porque se prefiere una solución buena en poco tiempo que la mejor en mucho tiempo. Otro problema típico en el que se aplica esta técnica es el de la mochila. En él se tiene una mochila de capacidad $M(>0)$ y $N(>0)$ objetos fraccionables (es decir, que se pueden partir) con un peso y un valor determinado. El objetivo es llenar la mochila maximizando su valor. Como ves, se ajusta a su resolución voraz: es un problema de optimización y la solución es un subconjunto de los objetos, que se puede construir por etapas. Además, se pueden identificar los elementos voraces:

- **Objetivo:** maximizar el valor de la mochila.
- **Candidatos:** los N objetos.





- **Candidatos seleccionados (S):** los que se han incluido en la mochila.
- **Criterio de solución:** el peso de los elementos en S igual a la capacidad M.
- **Criterio de factibilidad:** no sobrepasar la capacidad de la mochila al incluir en S el nuevo elemento seleccionado.
- **Criterio de selección:** elegir el objeto de mayor relación valor/peso, que proporciona una solución óptima.

Otros problemas en los que la estrategia voraz se aplica con éxito son los de la planificación de tareas para maximizar rendimiento o en los de recorrido de una serie de puntos con la menor distancia posible.

Saber más

Brassard, G., Bratley, P., & Giner, R. G. B. (1997). *Fundamentos de algoritmia* (Vol. 3, No. 5.1). Madrid: Prentice Hall.

Guerequeta, R., & Vallecillo, A. (1998) *Técnicas de Diseño de Algoritmos*. Servicio de Publicaciones de la Universidad de Málaga. Disponible online: <http://www.lcc.uma.es/~av.Libro/indice.html>





Creación de
contenidos digitales

Nivel A2 3.4 Programación

Pseudocódigo





Pseudocódigo

Introducción

El pseudocódigo, como su nombre indica, hace referencia al concepto de código “falso”, es decir, código que no es procesable por una computadora. Es en realidad una descripción a alto nivel de los elementos y pasos que intervienen en un algoritmo o programa. Por lo tanto, al igual que los diagramas de flujo, el pseudocódigo se utiliza para entender de forma sencilla su funcionamiento. La principal diferencia entre ambos reside en que el diagrama de flujo es una representación gráfica y el pseudocódigo hace uso de un conjunto de palabras y/o símbolos.

El pseudocódigo es una herramienta potente, ya que permite describir con palabras cercanas al lenguaje natural cómo se va a resolver un determinado algoritmo.

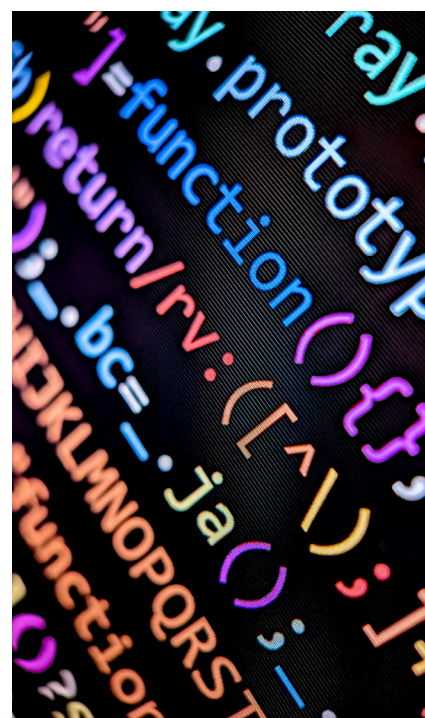
Según las preferencias de la persona encargada de escribir en pseudocódigo se puede utilizar un idioma u otro; en nuestro caso podemos utilizar palabras en español.

Ventajas del pseudocódigo

Una de las principales ventajas que presenta el pseudocódigo es la capacidad de abstraerse de un lenguaje de programación en particular. Esto significa que no es necesario tener un alto conocimiento de la tecnología que se va a utilizar posteriormente, ya que el pseudocódigo es independiente del lenguaje de programación utilizado para su implementación.

Un programa en pseudocódigo ofrece una visión global del algoritmo, lo que facilita detectar posibles errores conceptuales antes de ser implementado.

Esta detección, en las primeras fases de desarrollo de aplicaciones, es fundamental para conseguir un ahorro de costes, recursos y producir software de calidad. Es por esta razón por la que se utiliza en etapas tempranas del diseño





de aplicaciones. Además, puede ser escrito y entendido sin necesidad de tener experiencia en programación, lo que facilita el ser compartido por personas con diferentes roles.

Características y elementos

A pesar de ser flexible y no estar sujeto a una sintaxis en concreto, hay que tener presente cuales son las construcciones y operaciones más destacables en el desarrollo de un algoritmo. A continuación, se describen algunas de las más frecuentes:

- **Palabras clave.** Son palabras reservadas, utilizadas para indicar una determinada acción. Son el caso de las siguientes:
 - Comienzo y fin del algoritmo.
 - Comienzo y fin de un bloque.
 - Lectura y escritura de datos.
- **Operaciones matemáticas.**
 - Aritméticas.
 - De asignación.
 - Lógicas.
- **Condicionales.** Un condicional es un fragmento del algoritmo (secuencia de pasos) que debe ejecutarse si se cumple una determinada condición.
- **Iteraciones y bucles.** Se trata de un conjunto de pasos que deben repetirse o un determinado número de veces o bien hasta que se cumple una condición.

Recomendaciones de uso

Aunque no existe una forma concreta de escribir un algoritmo en pseudocódigo, es recomendable seguir una serie de recomendaciones para que pueda ser legible de forma universal:

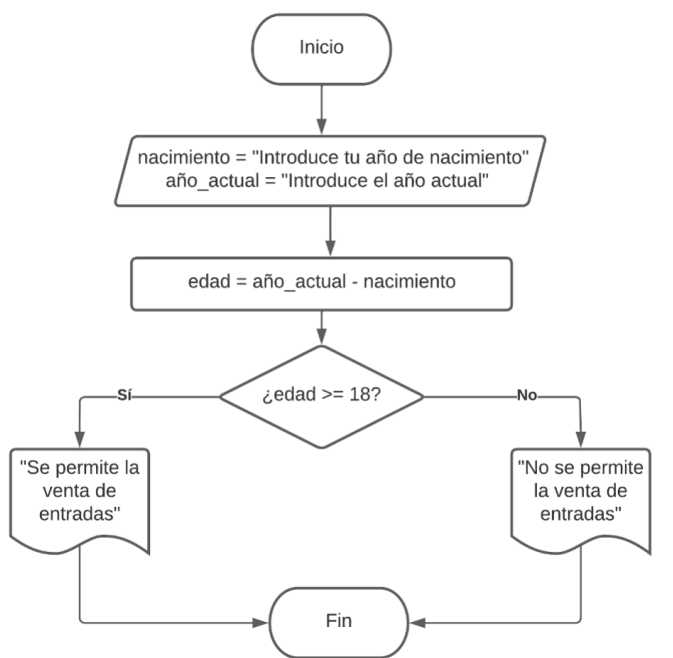
- Ser conciso y esquemático, identificado los pasos realmente necesarios y eliminando información innecesaria.
- Escribir una sola sentencia por línea.
- Reflejar la jerarquía de los pasos. Si un paso o conjunto de pasos está contenido en una sección o bloque, es recomendable utilizar sangrado (indentación).



- Capitalizar la primera letra de aquellas palabras que tengan un significado especial (palabras clave).
- Utilizar como nombre de datos y algoritmos aquellas palabras con cuya lectura se pueda entender lo que representan.

Para comprender el funcionamiento del pseudocódigo, se va a partir del ejemplo de diagrama funcional expuesto en temas anteriores. En él se planteaba un algoritmo para controlar la venta de entradas a un concierto. Su resolución utilizando lenguaje algorítmico sería el siguiente:

Diagrama de flujo



Pseudocódigo

Algoritmo venta_de_entradas

Datos

nacimiento, edad, año_actual: Entero

Instrucciones

Obtener por teclado el valor de nacimiento

Obtener por teclado el valor de año_actual

edad ← año_actual - nacimiento

Si edad >= 18 **Entonces:**

Escribir "Se permite la venta de entradas"

Si no

Escribir "No se permite la venta de entradas"

Fin_venta_de_entradas



A continuación, se analizan cada una de las líneas escritas anteriormente.

Algoritmo venta_de_entradas	La palabra clave "Algoritmo" indica el comienzo de este. El nombre del algoritmo es descriptivo ya que indica el procedimiento a realizar.
Datos	La palabra clave "Datos" indica que va a comenzar un bloque en el que se muestran los datos implicados en la resolución del programa.
nacimiento, edad, año_actual: Entero	Los datos a utilizar son de tipo entero. Se vuelve a emplear nombres descriptivos y un sangrado para indicar la pertenencia al bloque "Datos".
Instrucciones	Palabra clave empleada para indicar el comienzo de la secuencia de pasos necesarios para la resolución del problema. Desde la siguiente línea y hasta que finalicen las instrucciones, se realiza un sangrado para definir este bloque.
Obtener el valor de nacimiento	Se solicita y se guarda el año de nacimiento.
Obtener el valor de año_actual	De igual forma, se solicita y se almacena el año actual.
edad ← año_actual - nacimiento	Se realizan dos operaciones matemáticas: una operación aritmética (resta) y una operación de asignación (el resultado se asigna al dato "edad").
Si edad >= 18 Entonces:	Comienza un condicional. Si la edad es mayor o igual que 18 años entonces se continúa con las líneas mostradas en el bloque del condicional.
Escribir "No se permite la venta de entradas"	Se muestra al usuario la frase "Se permite la venta de entradas".
Si no:	Si no se cumple la condición indicada anteriormente, es decir, si el cliente tiene menos de 18 años, se continúa por los pasos definidos en este bloque.
Escribir "Se permite la venta de entradas"	Se muestra al usuario la frase "No se permite la venta de entradas".
Fin_venta_de_entradas	Fin del algoritmo.

NOTA

Independientemente de las palabras claves utilizadas, estructuras u operadores, lo más importante de un programa escrito en pseudocódigo es que debe de ser descriptivo y claro. El objetivo final consiste en que se pueda entender cada uno de los pasos que llevan a resolver el problema.

Saber más

El uso del pseudocódigo: Instrucciones en palabras sencillas.
ISBN: 9781538334119, 1538334119. Jonathan Bard.



Creación de
contenidos digitales

Nivel A2 3.4 Programación

Divide y vencerás





Divide y Vencerás

La técnica conocida como Divide y Vencerás consiste en descomponer un problema complejo en subproblemas más simples, independientes entre sí, para construir la solución del problema inicial mediante la combinación de las soluciones de los subproblemas. Este método se puede esquematizar de la siguiente manera:

- Si el problema es **sencillo**, solucionarlo;
- Si el problema es **complejo**:
 - 1 | **Dividir** el problema en subproblemas más pequeños
 - 2 | **Resolver** cada subproblema, aplicando la misma estrategia
 - 3 | **Combinar** dichas soluciones para obtener la solución del problema original

Un problema se considera “sencillo” cuando se puede resolver fácilmente, lo que normalmente se da cuando el tamaño del problema es pequeño. Además, los subproblemas deben ser menos costosos de resolver que el problema original. También deben ser disjuntos, es decir, que la solución de cada uno de ellos debe obtenerse independientemente de los otros. Por último, el número de subproblemas debe ser mayor o igual que 2; si fuera 1, la técnica se denomina *reducción* o *simplificación*. Así, la forma general de este tipo de algoritmos es la que se ilustra en el listado siguiente.

Algoritmo Divide y Vencerás

Datos

TamañoProblema: Número
S, SP1, ..., SPk: Solución

Instrucciones

Si TamañoProblema es Pequeño

Entonces S ← ResolverProblema

Sino

SP1 ← Divide y Vencerás (SubProblema1)

...

SPk ← Divide y Vencerás (SubProblemaK)

S ← Combinar (SP1, ..., SPk)

Fin_Si

Devolver S

Fin_Divide y Vencerás



Respecto a la eficiencia de este tipo de algoritmos, hay que insistir en que los subproblemas sean independientes entre sí, es decir, que no existan “solapes” entre ellos, porque dispararía la complejidad a órdenes exponenciales. En los casos en los que haya solapamiento entre subproblemas, existen otras técnicas más eficientes para resolverlos, como es el caso de la programación dinámica. Además, es importante que los subproblemas tengan tamaños similares para que no se produzcan desequilibrios que afecten a su rendimiento. Cuando se cumplen todas estas restricciones, la complejidad de este tipo de algoritmos suele ser polinomial, lo que los hace muy eficientes.

Un ejemplo sencillo que ilustra esta técnica es la búsqueda del menor valor de una colección de números. Si la colección solo tiene un número, el problema es trivial: el menor valor es el de dicho número. Si la colección tiene más de un número, se puede dividir la colección en dos mitades de tamaño similar. Así, una vez obtenido el menor elemento de cada una de las dos mitades, bastará con quedarnos con el menor de ellos para tener el menor elemento de la colección completa. El algoritmo *MenorColeccion* del listado siguiente describe la idea. La Figura 1 ilustra dos ejemplos del funcionamiento del algoritmo. El de la izquierda, es la aplicación a dos casos: cuando el problema es sencillo ($P=1$) y cuando no lo es ($P>1$). El de la derecha, describe cómo se obtiene la obtención del menor de una de las mitades (para la otra mitad sería similar) aplicando la misma técnica. Para el caso de que la colección tiene más de un elemento, se descompone siempre en dos mitades, y el proceso se repite hasta que el tamaño de la colección queda reducido a 1, que es el que constituye el caso en el que el problema tiene una solución inmediata. Obsérvese que, para cada mitad, la obtención del menor valor se obtiene aplicando la misma técnica. Por último, la complejidad de este algoritmo es lineal, $O(n)$, ya que se comparan todos los elementos de la colección.

```
<img class="mini" alt="img" background="" />
<td class="tableCell">
  <img class="mini" alt="img" background="" />
</td>
</tr>
</table>
</div>
<div id="pages" class="pages">
</div>
</div>
</td>
</tr>
</table>
</div>
<td style="background: url(images/www_07.gif); width: 25px; height: 25px; vertical-align: top; background-image: none;" colspan="3" style="vertical-align: top; background-image: none;">
  <div class="contentRight contentRightTop">
    <div class="bigPhotoDiv">
      <table cellpadding="0" cellspacing="0" width="100%">
        <tr>
          <td valign="middle">
            
            
          </td>
        </tr>
      </table>
    </div>
  </div>
</td>
</tr>
</table>
</div>
<div id="background" class="bigPhotoDiv">
  <table cellpadding="0" cellspacing="0" width="100%">
    <tr>
      <td valign="middle">
        <div id="bigImages" class="bigImages">
          <img id="bigImage1" alt="" src="" />
          <img id="bigImage2" alt="" src="" />
          <img id="bigImage3" alt="" src="" />
          <img id="bigImage4" alt="" src="" />
          <img id="bigImage5" alt="" src="" />
          <img id="bigImage6" alt="" src="" />
        </div>
      </td>
    </tr>
  </table>
</div>
```

**Algoritmo** MenorColeccion**Datos**

P, N1, N2, ..., NP: Número

Menor, MenorI, MenorD: Número

Instrucciones**P** ← TamañoColección**Si** P = 1 **Entonces** Menor ← NP**Sino**

K ← P/2

MenorI ← MenorColeccion (N1, ..., Nk)

MenorD ← MenorColeccion (Nk+1, ..., NP)

Si MenorI < MenorD **Entonces** Menor ← MenorI **Sino** Menor ← MenorD **Fin_Si****Fin_Si** S**Devolver** Menor**Fin_MenorColeccion**

P=1

[7] → Menor = 7

P>1

[4, 7, 3, 6, 1, 5]

MenorI = Menor [4, 7, 3] = 3

MenorD = Menor [6, 1, 5] = 1

Como MenorI > MenorD

Menor = MenorD = 1

Menor [4, 7, 3] (P>1)

MenorI = Menor [4] (P=1) = 4

MenorD = Menor [7, 3] (P>1)

MenorI = Menor [7] = 7

MenorD = Menor [3] = 3

Como MenorI > MenorD

Menor = MenorD = 3

MenorD = Menor [7, 3] = 3

Como MenorI > MenorD

Menor = MenorD = 3

Figura 1. Ejemplo de aplicación del algoritmo para dos colecciones de números (izda) y desarrollo de uno de los casos (dcha).

La técnica de Divide y Vencerás ha sido utilizada con éxito para reducir la complejidad de los algoritmos tradicionales de ordenación de elementos, que en sus versiones más sencillas (inserción, selección, burbuja) son $O(n^2)$. Así, los algoritmos de ordenación rápida (*quicksort*) o de ordenación por mezcla (*mergesort*), que aplican “divide y vencerás” son ambos $O(n \log n)$. Veamos uno de ellos, el *quicksort*, cuya idea es la siguiente:

- Si la colección solo tiene un elemento, el problema es trivial: ya está ordenada.



- Si la colección tiene más de un elemento:

- 1 | Elegir un elemento X (llamado pivote) de la colección
- 2 | Mover los menores que X a su izquierda y los mayores que X a su derecha. El elemento X ha quedado colocado en su sitio:

<X	X	>X
----	---	----
- 3 | Ordenar aplicando la misma idea la izquierda de X
- 4 | Ordenar aplicando la misma idea la derecha de X

El listado siguiente describe el algoritmo:

Algoritmo QuickSort

Datos

X:Elemento; N, PosX:Número;

Instrucciones

N ← TamañoColección

Si N > 1

Entonces X ← ElegirPivote

PosX ← Posición de X al mover menores y mayores

QuickSort ← (hasta PosX)

QuickSort ← (desde PosX)

Fin_Si

Fin_QuickSort

La Figura 2 ilustra un ejemplo de funcionamiento. En este caso, se ha elegido como pivote el elemento que ocupa la primera posición, aunque podría haber sido cualquier otro (el último o el de la posición central).

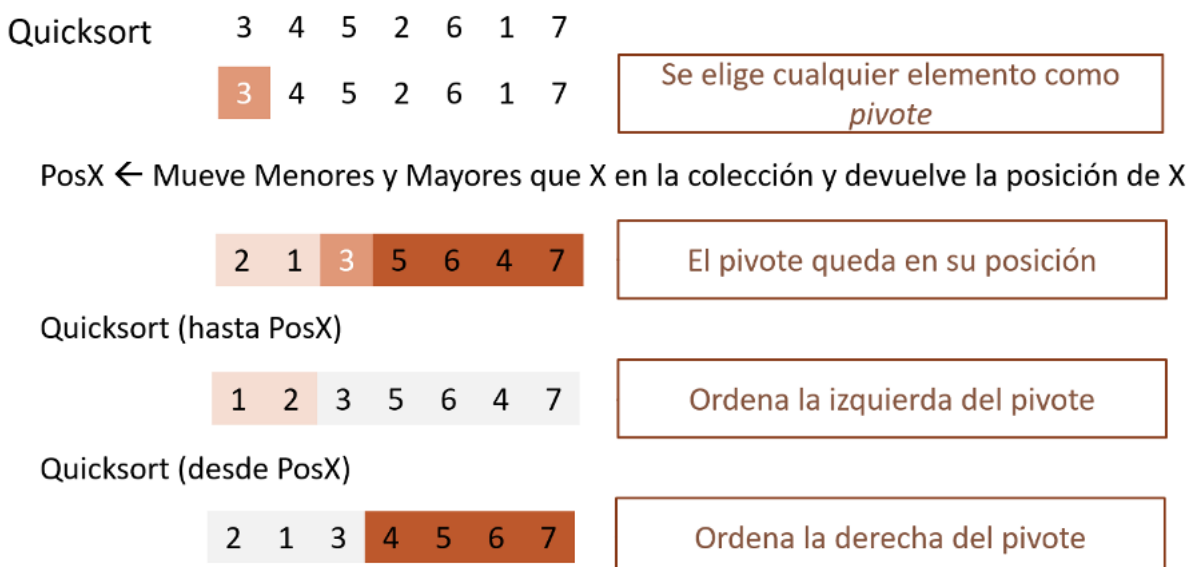


Figura 2. Ejemplo de funcionamiento del algoritmo QuickSort.



La elección del pivote puede afectar a la complejidad del algoritmo, pero no se profundizará en este problema aquí. Por otro lado, el algoritmo que mueve los menores y mayores que el pivote, a su izquierda y derecha, respectivamente, y proporciona la posición en la que ha quedado, no se incluye para no confundir al lector o a la lectora. La Figura 3 ilustra la idea, que consiste en recorrer la secuencia desde la izquierda hacia la derecha (izq) hasta que se encuentre un elemento mayor que el pivote, y desde la derecha hasta la izquierda (der) hasta encontrar un elemento menor que el pivote. En ese momento, se intercambia uno por otro. El proceso se repite hasta que se crucen izq y der. La posición de der será la posición ordenada del pivote, por lo que se realiza el intercambio entre ambas.

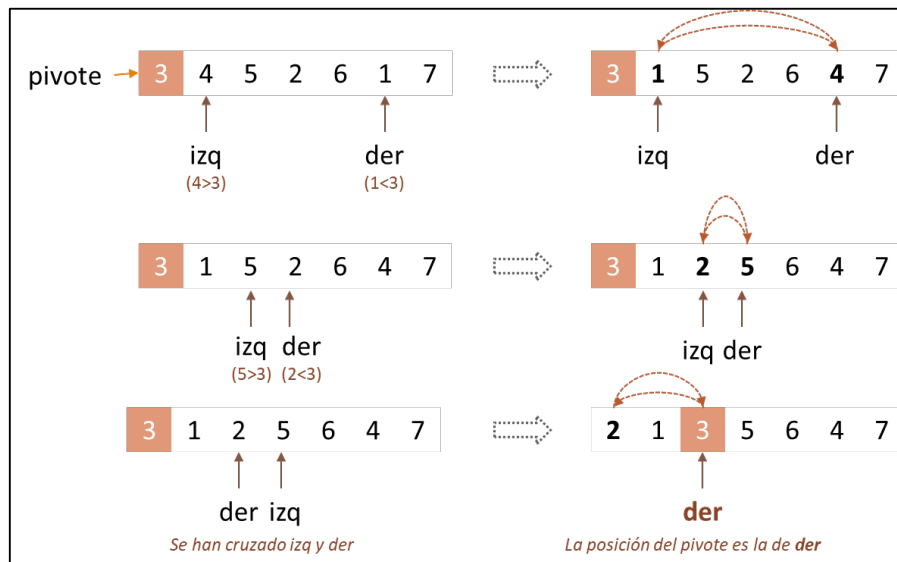


Figura 3. Ejemplo de colocación del pivote.

i Saber más

Además del Quicksort, existen otros algoritmos de ordenación que utilizan la técnica de Divide y Vencerás, como es el caso del MergeSort. Si quieres comprobar cómo se ejecutan cada uno de ellos paso a paso, puedes hacerlo a través de la plataforma **Visualgo.net** (visualgo.net/en/sorting)



Creación de
contenidos digitales

Nivel A2 3.4 Programación

Lenguajes de programación vs. Lenguajes naturales





Lenguajes de programación vs. Lenguajes Naturales



LENGUAJES DE PROGRAMACIÓN: ¿EN QUÉ CONSISTEN?

Se presenta la diferencia entre los lenguajes de programación y los lenguajes naturales, empleando los elementos definidos anteriormente: léxico, sintaxis y semántica

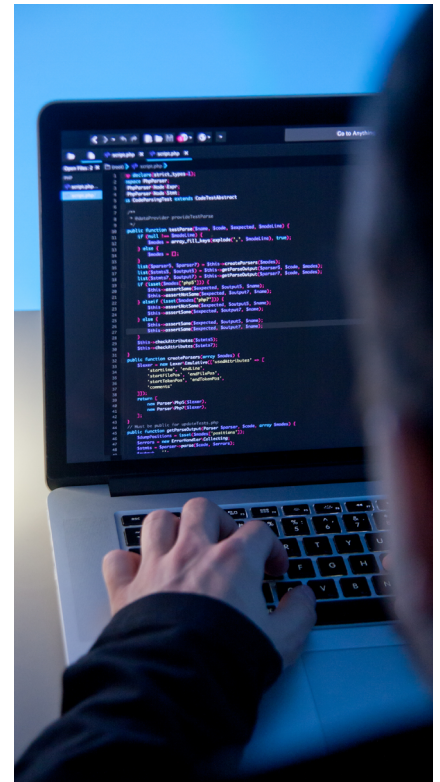
e.digitall.org.es/A3C34A2V04

El lenguaje natural es el lenguaje utilizado normalmente en la comunicación entre humanos. Existen distintos lenguajes para comunicarse entre humanos como el español, el inglés, el francés... y cada uno tiene un conjunto de palabras diferentes, es decir, un vocabulario o léxico diferente. En español, por ejemplo, existen palabras como: "ciudad", "desierta", "la", "estaba". Palabras se pueden clasificar en: verbos, sustantivos, adjetivos, determinantes, pronombres, adverbios, conjunciones, preposiciones.

La sintaxis permite ordenar las palabras del léxico para formar frases correctamente escritas. Así, por ejemplo, el sujeto se suele formar por un determinante seguido de un sustantivo y suele aparecer delante del verbo y un adjetivo para formar una frase. Siguiendo estas reglas, con las palabras anteriores, se puede formar la frase: "la ciudad estaba desierta".

La semántica permite entender el significado de esa frase. Si una persona le dice que a otra que "la ciudad estaba desierta", esta entendería que no había nadie en la ciudad. En general, el lenguaje natural suele ser muy ambiguo y requiere conocer el contexto para entender frases.

De forma análoga, los lenguajes de programación también se emplean para permitir la comunicación, pero en este caso, entre el ordenador y la persona que desarrolla un programa informático. Al igual que los lenguajes naturales cada lenguaje de programación tendrá su léxico o vocabulario concreto junto con sus categorías léxicas que es necesario conocer. Normalmente es un vocabulario pequeño, que contiene un número limitado de términos fácilmente memorizable. Así, por ejemplo, es fácil encontrar en la mayoría de los lenguajes categorías léxicas como: identificador, números, símbolo de asignación, operador aritmético, operador lógico, operador





relacional, separadores como el punto y como o la coma, agrupadores como las llaves o los paréntesis y palabras reservadas como: "if", "else", "for" o "while" por citar alguna.

De forma similar al lenguaje natural, la sintaxis indica al programador, la forma en la cual se estructuran las frases que un ordenador puede entender. Estas frases normalmente son las instrucciones que puede ejecutar un ordenador. Un ejemplo de una posible instrucción sintácticamente correcta sería: "c:=2+3;", que corresponde con la estructura: identificador símbolo de asignación expresión, siendo esta última número operador de suma número.

La semántica asociada a esta instrucción sería sumar los valores 2 más 3 y su resultado se guardaría con el identificador c.

Al contrario que el lenguaje natural, los lenguajes de programación son lenguajes formales con unas reglas gramaticales precisas que determinan el léxico y sintaxis del lenguaje y con una semántica perfectamente definida, carentes de ambigüedad.

Saber más

Univerisdad de Pensilvania. From Natural Language to Programming Language. e.digitall.org.es/university-pensilvania

OpenWebinars. Qué es un lenguaje de programación. e.digitall.org.es/open-webinars



Creación de
contenidos digitales

Nivel A2 3.4 Programación

Intérpretes vs. Compiladores





Intérpretes vs. Compiladores



LENGUAJES DE PROGRAMACIÓN: ¿EN QUÉ CONSISTEN?

Presentar el trabajo que realizan los intérpretes y los compiladores empleando los elementos definidos anteriormente: léxico, sintaxis y semántica.

e.digitall.org.es/A3C34A2V04

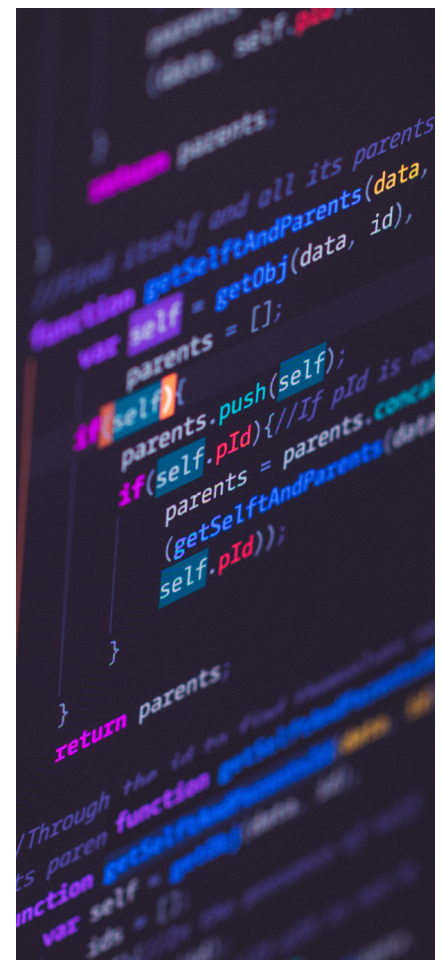
Cuando dos personas se comunican suelen emplear el mismo idioma o lenguaje. En caso de no hablarlo, deberían emplear a alguien o algo que tradujera lo que uno dice en su idioma al idioma que el otro pueda entender. Por ejemplo, imagina que te tienes que comunicar con un alemán, y no conoces su idioma, sin duda alguna en ese acto de comunicación necesitarás de un traductor de español a alemán y al contrario.

¿Para qué sirven los intérpretes y los compiladores?

En el mundo de la programación existe una necesidad de comunicación similar: un programador informático tiene que decirle a un ordenador lo que tiene que hacer. Para ello debe usar el lenguaje máquina que es complejo (formado por ceros y uno) o usar un lenguaje de programación más cercano al humano, sin embargo, este último no lo entiende la máquina.

Es un caso de comunicación en el que cada uno habla o entiende su propio lenguaje, el ordenador solo comprende instrucciones escritas en el lenguaje máquina y el programador solo es capaz de comunicar las instrucciones que debe ejecutar el ordenador por medio de cadenas de un lenguaje de programación de alto nivel.

El problema está en hacer que el ordenador entienda las instrucciones dadas en el lenguaje de programación. Por tanto, es necesario usar un "traductor" que transforme las instrucciones expresadas con un lenguaje de programación, a instrucciones en lenguaje máquina que pueda entender el ordenador. Hay dos posibles tipos de "traductores" de este tipo: los compiladores y los intérpretes.





Ambas herramientas deben comprobar primero que las cadenas escritas en el lenguaje de programación son correctas: a nivel léxico, es decir estén formadas por palabras del vocabulario del lenguaje; a nivel sintáctico, es decir tengan una estructura correcta; y a nivel semántico, esto es tengan un significado. Esto es lo que se denomina fase de análisis del traductor o procesador del lenguaje.

Una vez comprobado que todas las instrucciones están formadas por términos correctos del vocabulario del lenguaje, así como que la estructura de las instrucciones es correcta según las reglas gramaticales que determinan las cadenas válidas del lenguaje, entonces se analiza a nivel semántico. Este análisis semántico implica dos cosas: por un lado, acaba de hacer comprobaciones de más alto nivel, por ejemplo, comprobar que una variable se ha declarado antes de ser usada o que no se asigna a una variable un valor diferente al que puede almacenar (i.e. asignar a una variable un valor numérico cuando se ha declarado como una variable para almacenar cadenas de texto). Por otro lado, lo que hace es entender lo que se está comunicando con la cadena escrita en el lenguaje de alto nivel con el objetivo de comunicárselo a la máquina.

Con lo que se ha entendido que se está ordenando con la instrucción escrita en lenguaje de alto nivel, se genera la instrucción para la máquina en su propio lenguaje. Esto es lo que se denomina fase de síntesis del traductor o procesador del lenguaje. La traducción generalmente no es uno a uno, es decir que una instrucción del lenguaje de alto nivel por lo general implicará varias instrucciones del lenguaje máquina. A estos tipos de traductores también se les denomina procesadores de lenguajes, ya que implican analizar y sintetizar.



Diferencias entre intérpretes y compiladores

La principal diferencia entre un intérprete y un compilador está en su objetivo. Un compilador tiene como objetivo analizar y traducir completamente el programa escrito por un programador en lenguaje de programación de alto nivel a sentencias del lenguaje máquina. Como consecuencia el texto traducido a lenguaje máquina podrá ser almacenado en un fichero o archivo, pudiéndose referir a él como el programa compilado. Por el contrario, un intérprete tiene como objetivo analizar sentencia a sentencia el programa escrito por el programador y ejecutar cada sentencia una vez analizada. No se genera un resultado que se pueda almacenar en un fichero.

Cada enfoque tiene sus propias ventajas e inconvenientes: un programa que ha sido compilado se podrá ejecutar en la máquina cuando se quiera, siendo esto una de las ventajas del compilador. Por el contrario, el programa interpretado debe analizarse y ejecutarse cada vez que se dese ejecutar en la máquina destino. Esto además implica mayor rapidez en la ejecución de los programas compilados frente a los interpretados, ya que estos últimos siempre van a tener que ser analizados en su ejecución, los compiladores solo realizan el análisis una vez.

Por el contrario, el intérprete, al analizar y ejecutar una a una cada instrucción, facilita la depuración, ya que permite comprobar el resultado de cada instrucción, y si hay algún error, saber en qué punto exacto del programa se ha cometido. Este proceso es más complejo con un compilador y requiere de entornos de desarrollo. Además, los intérpretes al no generar un programa compilado favorecen la portabilidad de los programas. Cada máquina tendrá su propio intérprete del lenguaje de alto nivel, de manera que en cada ejecución hay una ejecución en la máquina en la que se ejecuta el intérprete.



Saber más

Keepcoding. Qué es un intérprete y qué es un compilador.
e.digitall.org.es/interprete-compilador

ResearchGate. Lovello, J. M. C. Conceptos básicos de procesadores de lenguaje. Servitec. e.digitall.org.es/conceptos-basicos



Creación de
contenidos digitales

Nivel A2 3.4 Programación

Concepto de programa en Scratch





Concepto de programa en Scratch



SCRATCH

Se presentan los principales elementos del lenguaje de programación Scratch.

e.digitall.org.es/A3C34A2V05

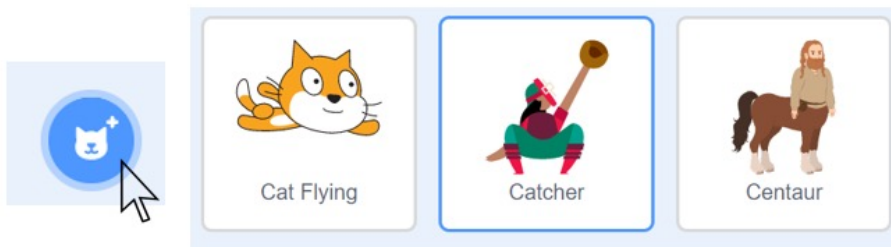
Scratch

Scratch es un lenguaje muy útil para contar historias de personajes que puede interactuar entre ellos. Sigue una metáfora en la que un programa informático es una historia en la que una serie de **personajes** realizan una serie de **acciones** sobre un **escenario**.

Veamos un pequeño ejemplo de programa en el que un gato vuela por el cielo.

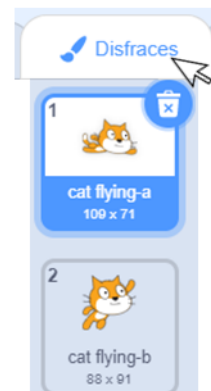
Personajes

Los personajes que intervienen en un programa reciben el nombre de **objetos** o **sprites**. Si pinchas sobre el icono del gato, aparecerá una galería con múltiples personajes. Seleccionaremos como personaje al *gato volador*.



Disfraces

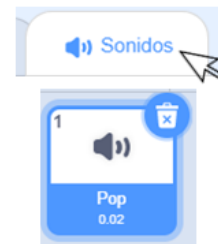
Este personaje tiene dos disfraces, uno cuando vuela para coger altura y otro para desplazarse una vez que ya ha cogido altura, es decir, cuando ya está en el cielo. Selecciona la pestaña **Disfraces** para ver todos los disfraces o crear los tuyos propios.





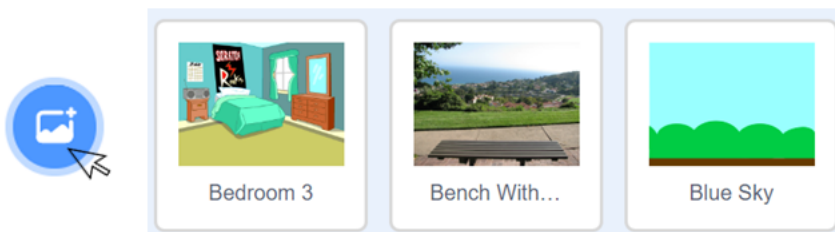
Sonidos

Todos los personajes pueden emitir sonidos. Este personaje por defecto trae un sonido, pero tú puedes crear distintos sonidos.



Escenario

El gato volador se moverá por el escenario con el fondo que elijas. Como queremos que el gato vuele, seleccionaremos el fondo del cielo de entre todos los fondos.



Bloques de instrucciones

Las acciones del gato se seleccionarán de entre el grupo de instrucciones de Código que tiene Scratch.

Estas instrucciones están agrupadas por colores según su tipo. Pueden ejecutar movimientos de los personajes, que los personajes emitan sonidos, digan frases, cambien de posición o tomen decisiones según las circunstancias de la historia.

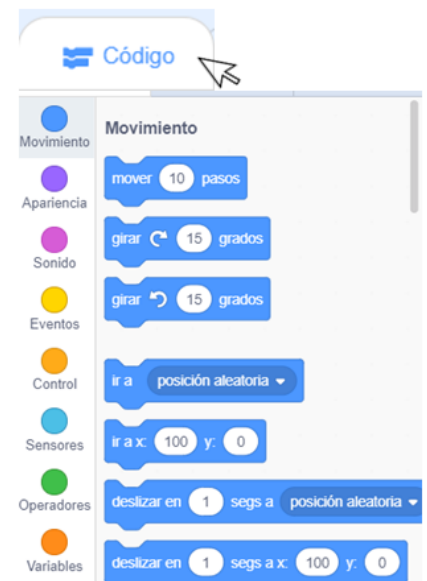
En este caso, nuestro gato debe volar, o lo que lo mismo, moverse por el escenario, mientras comenta cómo ve la ciudad desde el aire.

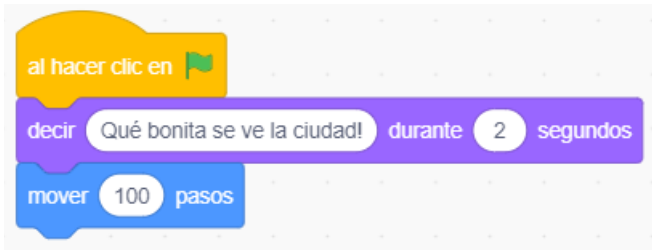
Programa

Gracias a las instrucciones, podemos ordenar los eventos que van a pasar en nuestra historia. Las tres acciones que vemos en este ejemplo representarían el siguiente algoritmo:

- Iniciar el programa
- El gato dice la frase: "Qué bonita se ver la ciudad!"
- El gato se mueve varios pasos sobre el fondo del cielo para que dé la sensación de volar.

Y el resultado de este programa sería nuestro gato en movimiento, surcando los cielos y comentando qué bonita se ve la ciudad.





i Saber más

Editor online de Scratch. Página oficial de editor.

scratch.mit.edu/projects/editor

Proyectos Scratch. Repositorio de proyectos compartidos en lenguaje

Scratch. scratch.mit.edu/ideas



Creación de
contenidos digitales

Nivel A2 3.4 Programación

Expresiones y asignación

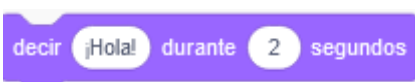




4 | Tras la creación de las tres variables, añadir a la zona de programación el evento que le indica al programa que comience la ejecución. En este ejemplo, se utiliza:

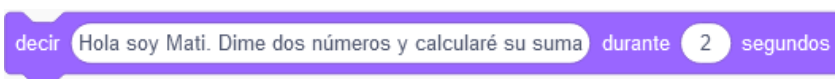


5 | Lo primero que tiene que hacer *Mati* es presentarse. En la categoría de color morado, etiquetada como *Apariencia*, se encuentran los bloques que permiten controlar la apariencia de los elementos visuales de Scratch. Pinchar y arrastrar a la zona de programación el bloque:



Con este elemento, el personaje escribirá en un bocadillo el texto introducido durante el tiempo indicado.

6 | Modificar el texto anterior para que el personaje diga "¡Hola, soy Mati! Dime dos números y calcularé su suma".



7 | A continuación, *Mati* debe de pedir el primer operando a sumar. Para ello, se utiliza el elemento:



que está contenido en el código Sensores. Con este bloque, el programa hará una pregunta o solicitará un dato y esperará hasta que el usuario introduzca la respuesta.

8 | Modificar la pregunta por la que se desea que aparezca, en este caso: "Dime el primer operando, por favor.".

9 | La respuesta introducida por el usuario será el primer operando de la suma. Ahora toca guardar este valor en una variable para poder consultarla posteriormente. Para ello, se asigna la respuesta del usuario a la variable **operando1** de la siguiente manera:

a. Seleccionar el bloque: 

y añadirlo debajo del bloque creado en el punto 7.

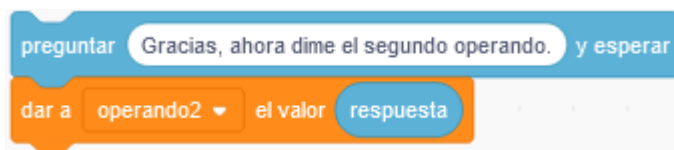
b. La variable a la que se le asignará la respuesta no es **mi variable** sino **operando1**, por lo que hay que seleccionar esta última del desplegable.




c. El valor asignado tampoco es 0 sino: **respuesta** que se encuentra dentro de la categoría *Sensores*. Arrastrar **respuesta** al óvalo ocupado por el valor 0.



10 | De la misma forma que en los pasos del 7 al 9, *Mati* tiene que solicitar el segundo operando de la suma y asignar la respuesta a la variable **operando2**.



11 | Es hora de crear la expresión encargada de sumar los dos operandos. La suma se encuentra dentro de la categoría *Operadores*. Seleccionar el operador:  y arrastrarlo hasta la zona de programación.

12 | Los operandos de esta suma son las variables **operando1** y **operando2** que se encuentran dentro de la categoría *Variables*. Seleccionar una por una y arrastrarla a la posición correspondiente del operador.



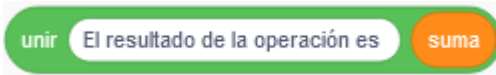
13 | Con la expresión anterior se realizará la suma, pero ¿dónde se almacena el resultado? El resultado debe asignarse a la variable **suma** por lo que se procederá de forma similar a los pasos explicados en el punto 9. La diferencia reside en que la variable elegida es **suma** y el valor es el resultado de la expresión del punto 12.



14 | Una vez calculada la suma, hay que mostrar al usuario el resultado de la operación. Se podría mostrar el valor como un número, pero para que la respuesta sea más natural se puede formar una frase del tipo "El resultado de la operación es (resultado)". Como se puede observar, la frase puede dividirse en dos partes. La primera parte es "El resultado de la operación" y la segunda el valor de la variable **suma**. Por lo tanto, se puede utilizar la unión de ambas partes para formar la frase completa.



a. Dentro de *Operadores* arrastrar a la zona de programación el operador **unir**. Como primer operando, escribir “El resultado de la operación es” y el segundo operando será la variable **suma**.



b. Después de realizar la concatenación de ambas cadenas, el resultado de esta expresión se utilizará en el bloque *decir*.



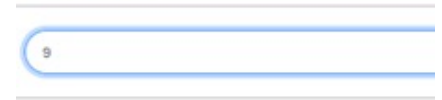
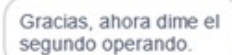
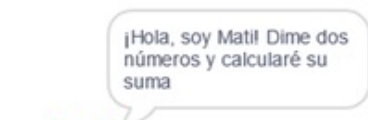
Una vez completados todos los pasos, los bloques del programa se ejecutarán de forma secuencial al pulsar sobre la bandera verde:



Partiendo de este ejemplo, se pueden crear diferentes programas que realicen el cálculo de otras expresiones, o bien ampliar el ejercicio con nuevas operaciones, ¿te animas a crear nuevas expresiones?

Saber más

Ejemplo en Scratch. e.digitall.org.es/ejemplo-scratch



Secuencias de la ejecución del programa de cálculo de suma



Creación de
contenidos digitales

Nivel A2 3.4 Programación

Mini-guía de estilo en Scratch





Mini-guía de estilo en Scratch

Aunque Scratch sea un lenguaje de programación visual y fácil de comprender, resulta importante seguir una serie de buenas prácticas a la hora de trabajar con él. En esta guía se muestra un conjunto de buenas prácticas y convenciones que puedes seguir al escribir código en Scratch. Esto nos ayudará a escribir código limpio, legible y fácil de mantener, y nos evitará cometer errores que dificulten la comprensión y el mantenimiento del código.

Aprenderemos a elegir nombres adecuados para nuestras variables y bloques, cómo utilizar comentarios, y a cómo organizar y estructurar el código para que sea más legible.



Nombrado de variables y otros recursos

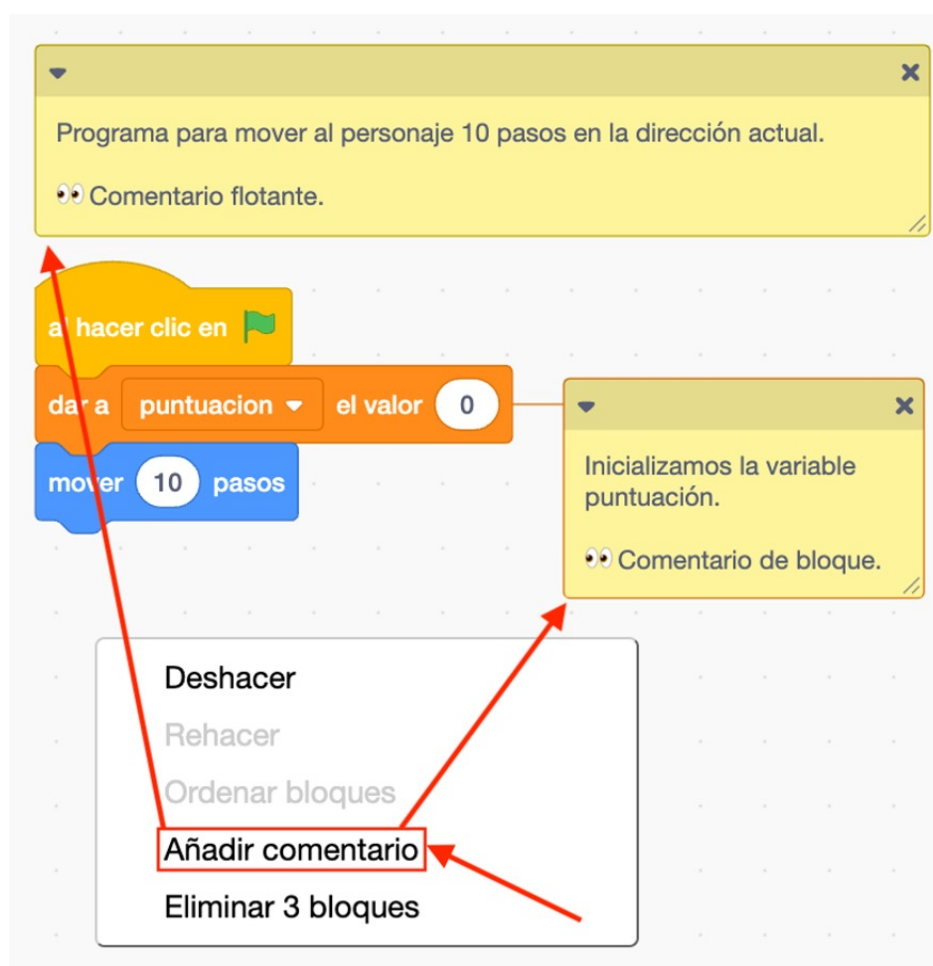
- Utilizar nombres descriptivos para las variables y recursos. Los nombres deben ser claros y concisos, y deben transmitir de manera efectiva el propósito de lo que representan. Por ejemplo, en lugar de utilizar un nombre como «x», podríamos utilizar «puntuación» para una variable que almacena la puntuación de un juego. Por otra parte, en lugar de tener un recurso con el nombre «Sprite113», podríamos darle un nombre más significativo como «PersonajePrincipal»
- Utilizar abreviaturas y acrónimos con cuidado. Si se utilizan abreviaturas o acrónimos en los nombres de las variables, debemos asegurarnos de que sean conocidos y fáciles de entender. De lo contrario, es mejor utilizar un nombre completo y descriptivo.
- Utilizar una convención de nombres estándar. Las convenciones de programación son un conjunto de directrices concreto que recomienda estilo, prácticas, y métodos de programación. En este caso, podemos utilizar una convención de nombres estándar para las variables y los bloques, de tal forma que nos permita organizar mejor el código y hacerlo más legible. Por ejemplo, podemos hacer que las variables globales (aquellas que se pueden utilizar en cualquier lugar del proyecto) empiecen siempre con una letra mayúscula, mientras que las variables locales (las que solo se pueden utilizar en un determinado



bloque) lo hagan siempre con una letra minúscula. De esta forma, resultaría más sencillo diferenciar entre variables locales y globales.

Uso de comentarios

- Añadir comentarios a los bloques para explicar su funcionamiento. Los comentarios son líneas de texto que se ignoran al ejecutar el programa, pero que nos permiten añadir explicaciones y notas al código para hacerlo más legible y fácil de entender. En Scratch, podemos añadir comentarios a los bloques de código o a la propia área de trabajo (llamados flotantes); en ambos casos, se añaden mediante click derecho y seleccionando la opción «Añadir comentario»:





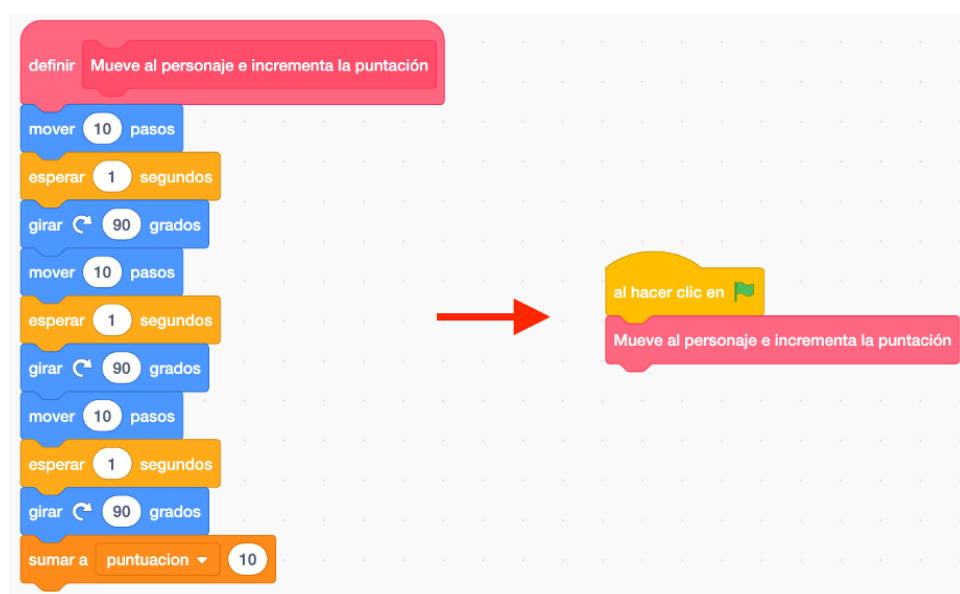
- Utilizar comentarios para documentar tus bloques personalizados. Cuando crees un bloque personalizado, es importante que añadas un comentario que explique su funcionamiento y su uso. De esta forma, otros usuarios que utilicen tu código podrán entender mejor cómo funcionan tus bloques personalizados y cómo utilizarlos en su propio proyecto.
- Utilizar comentarios para documentar las decisiones importantes que se hayan tomado al crear el programa código. Por ejemplo, si se ha utilizado un determinado algoritmo para resolver un problema o si se ha tomado una decisión específica en un punto del código, se puede añadir un comentario para explicar por qué se ha hecho así y cómo se ha llegado a esa decisión.
- Utilizar comentarios para documentar el propósito general del proyecto. Esto incluye información como el nombre del proyecto, su autor, la fecha en la que se ha creado y una breve descripción del propósito del proyecto. De esta forma, otros usuarios que utilicen el código podrán entender mejor su propósito y utilizarlo de manera adecuada.





Organización de código

- Utilizar bloques personalizados para organizar el código. Los bloques personalizados permiten agrupar ciertas partes del código en un solo bloque, lo que hace que el código sea más legible y fácil de mantener. Por ejemplo, se podría definir un bloque personalizado que moviera al personaje y modificara una variable, de tal forma que pudiéramos reutilizar ese bloque personalizado las veces que queramos sin tener que duplicar todos los bloques que forman dicha acción:



- Utilizar comentarios para separar los distintos bloques de código. Por ejemplo, insertando comentarios flotantes antes de los bloques que consideremos más importantes, podemos añadir descripciones a grupos de bloques. Podemos hacer esto por cada grupo de bloques que queramos comentar, de manera que nuestro código estará organizado en grupos lógicos y cada grupo tendrá un comentario que explique su función, lo que lo hará más fácil de leer y entender.



DigitAll

Formación en
Competencias
Digitales



Coordinación General

Universidad de Castilla-La Mancha
Carlos González Morcillo
Francisco Parreño Torres

Coordinadores de área

Área 1. Búsqueda y gestión de información y datos

Universidad de Zaragoza
Francisco Javier Fabra Caro

Área 2. Comunicación y colaboración

Universidad de Sevilla
Francisco Javier Fabra Caro
Francisco de Asís Gómez Rodríguez
José Mariano González Romano
Juan Ramón Lacalle Remigio
Julio Cabero Almenara
María Ángeles Borrueco Rosa

Área 3. Creación de contenidos digitales

Universidad de Castilla-La Mancha
David Vallejo Fernández
Javier Alonso Albusac Jiménez
José Jesús Castro Sánchez

Área 4. Seguridad

Universidade da Coruña
Ana M. Peña Cabanas
José Antonio García Naya
Manuel García Torre

Área 5. Resolución de problemas

UNED
Jesús González Boticario

Coordinadores de nivel

Nivel A1

Universidad de Zaragoza
Ana Lucía Esteban Sánchez
Francisco Javier Fabra Caro

Nivel A2

Universidad de Córdoba
Juan Antonio Romero del Castillo
Sebastián Rubio García

Nivel B1

Universidad de Sevilla
Francisco de Asís Gómez Rodríguez
José Mariano González Romano
Juan Ramón Lacalle Remigio
Montserrat Argandoña Bertran

Nivel B2

Universidad de Castilla-La Mancha
María del Carmen Carrión Espinosa
Rafael Casado González
Víctor Manuel Ruiz Penichet

Nivel C1

UNED
Antonio Galisteo del Valle

Nivel C2

UNED
Antonio Galisteo del Valle

Maquetación

Universidad de Salamanca
Fernando De la Prieta Pintado
Pilar Vega Pérez
Sara Alejandra Labrador Martín

Creadores de contenido

Área 1. Búsqueda y gestión de información y datos

1.1 Navegar, buscar y filtrar datos, información y contenidos digitales

Universidad de Huelva

Ana Duarte Hueros (coord.)
Arantxa Vizcaíno Verdú
Carmen González Castillo
Dieter R. Fuentes Cancell
Elisabetta Brandi
José Antonio Alfonso Sánchez
José Ignacio Aguaded
Mónica Bonilla del Río
Odriel Estrada Molina
Tomás de J. Mateo Sanguino (coord.)

1.2 Evaluar datos, información y contenidos digitales

Universidad de Zaragoza

Ana Belén Martínez Martínez
Ana María López Torres
Francisco Javier Fabra Caro
José Antonio Simón Lázaro
Laura Bordonaba Plou
María Sol Arqued Ribes
Raquel Trillo Lado

1.3 Gestión de datos, información y contenidos digitales

Universidad de Zaragoza

Ana Belén Martínez Martínez
Francisco Javier Fabra Caro
Gregorio de Miguel Casado
Sergio Ilarri Artigas

Área 2. Comunicación y colaboración

2.1 Interactuar a través de tecnología digitales

Iseazy

2.2 Compartir a través de tecnologías digitales

Universidad de Sevilla

Alién García Hernández
Daniel Agüera García
Jonatan Castaño Muñoz
José Candón Mena
José Luis Guisado Lizar

2.3 Participación ciudadana a través de las tecnologías digitales

Universidad de Sevilla

Ana Mancera Rueda
Félix Biscarri Triviño
Francisco de Asís Gómez Rodríguez
Jorge Ruiz Morales
José Manuel Sánchez García
Juan Pablo Mora Gutiérrez
Manuel Ortigueira Sánchez
Raúl Gómez Bizcocho

2.4 Colaboración a través de las tecnologías digitales

Universidad de Sevilla

Belén Vega Márquez
David Vila Viñas
Francisco de Asís Gómez Rodríguez
Julio Barroso Osuna
María Puig Gutiérrez
Miguel Ángel Olivero González
Óscar Manuel Gallego Pérez
Paula Marcelo Martínez

2.5 Comportamiento en la red

Universidad de Sevilla

Ana Mancera Rueda
Eva Mateos Núñez
Juan Pablo Mora Gutiérrez
Óscar Manuel Gallego Pérez

2.6 Gestión de la identidad digital

Iseazy

Área 3. Creación de contenidos digitales

3.1 Desarrollo de contenidos

Universidad de Castilla-La Mancha

Carlos Alberto Castillo Sarmiento
Diego Cordero Contreras
Inmaculada Ballesteros Yáñez
José Ramón Rodríguez Rodríguez
Rubén Grande Muñoz

3.2 Integración y reelaboración de contenido digital

Universidad de Castilla-La Mancha

José Ángel Martín Baos
Julio Alberto López Gómez
Ricardo García Ródenas

3.3 Derechos de autor (copyright) y licencias de propiedad intelectual

Universidad de Castilla-La Mancha

Gabriela Raquel Gallicchio Platino
Gerardo Alain Marquet García

3.4 Programación

Universidad de Castilla-La Mancha

Carmen Lacave Roderó
David Vallejo Fernández
Javier Alonso Albusac Jiménez
Jesús Serrano Guerrero
Santiago Sánchez Sobrino
Vanesa Herrera Tirado

Área 4. Seguridad

4.1 Protección de dispositivos

Universidade da Coruña

Antonio Daniel López Rivas
José Manuel Vázquez Naya
Martíño Rivera Dourado
Rubén Pérez Jove

4.2 Protección de datos personales y privacidad

Universidad de Córdoba

Aida Gema de Haro García
Ezequiel Herruzo Gómez
Francisco José Madrid Cuevas
José Manuel Palomares Muñoz
Juan Antonio Romero del Castillo
Manuel Izquierdo Carrasco

4.3 Protección de la salud y del bienestar

Universidade da Coruña

Javier Pereira Loureiro
Laura Nieto Riveiro
Laura Rodríguez Gesto
Manuel Lagos Rodríguez
María Betania Groba González
María del Carmen Miranda Duro
Nereida María Canosa Domínguez
Patricia Concheiro Moscoso
Thais Pousada García

4.4 Protección medioambiental

Universidad de Córdoba

Alberto Membrillo del Pozo
Alicia Jurado López
Luis Sánchez Vázquez
María Victoria Gil Cerezo

Área 5. Resolución de problemas

5.1 Resolución de problemas técnicos

Iseazy

5.2 Identificación de necesidades y respuestas tecnológicas

Iseazy

5.3 Uso creativo de la tecnología digital

Iseazy

5.4 Identificar lagunas en las competencias digitales

Iseazy



El material del proyecto DigitAll se distribuye bajo licencia CC BY-NC-SA 4.0. Puede obtener los detalles de la licencia completa en: <https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>