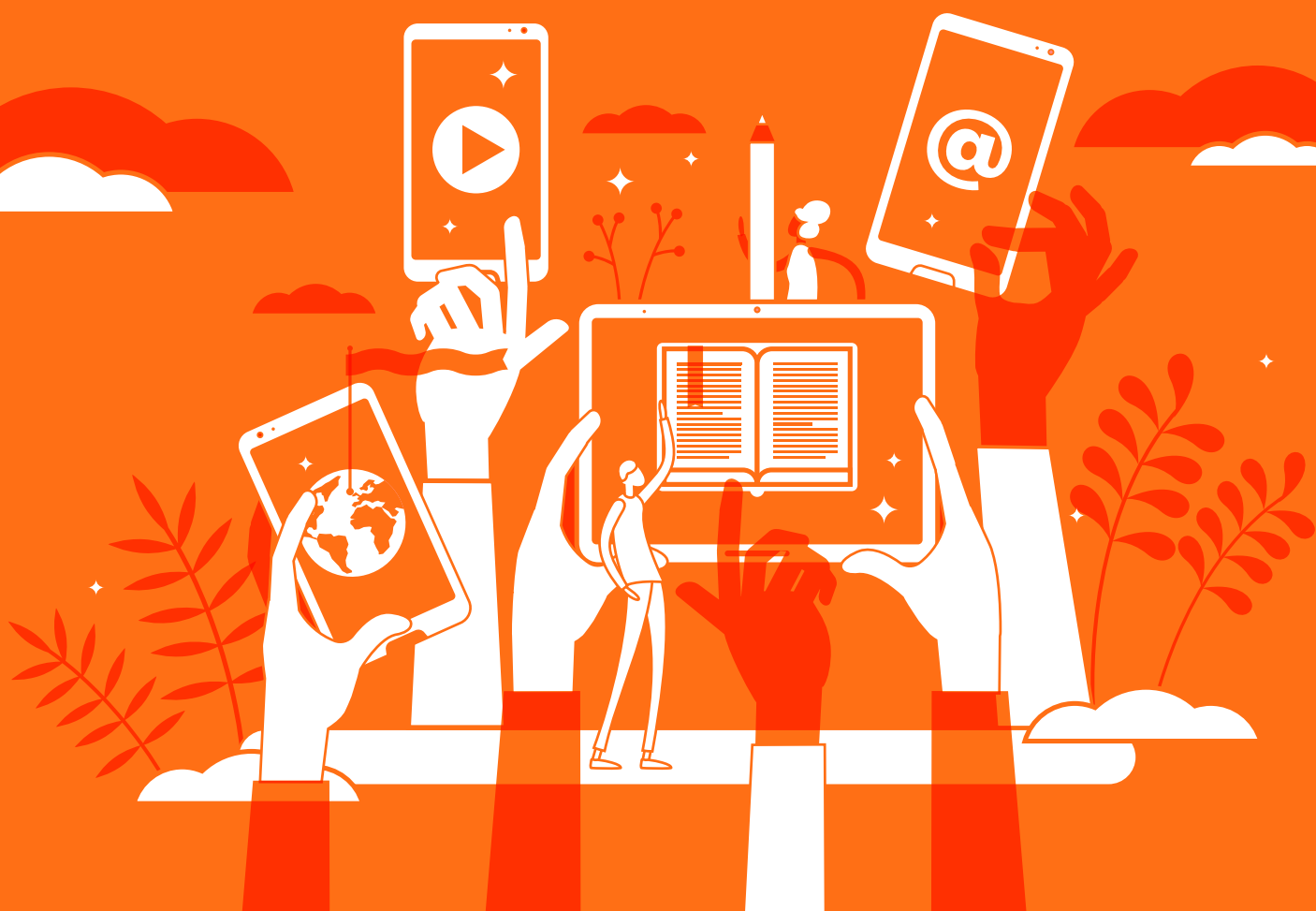




Formación en
Competencias
Digitales

3

Creación de contenidos digitales





Formación en
Competencias
Digitales



Creación de
contenidos digitales

Nivel B2





ÍNDICE

3.1. DESARROLLO DE CONTENIDOS

- *Uso avanzado de los procesadores de texto: conexión con otras herramientas ofimáticas*
- *Uso de herramientas habituales en procesadores de texto libres y online*
- *Efectos aplicados sobre texto*
- *Creación de imágenes animadas*
- *Herramientas de diseño gráfico libres y en la nube*
- *Compresión de imágenes y vídeo por lotes*
- *Alternativas para la creación de sitios web*
- *Niveles de adecuación de WCAG*

3.2. INTEGRACIÓN Y REELABORACIÓN DE CONTENIDO DIGITAL

- *Transiciones y animaciones avanzadas en presentaciones*
- *Remasterización de vídeo y audio*

3.3. DERECHOS DE AUTOR Y LICENCIAS DE PROPIEDAD INTELECTUAL

- *Dotando de copyright a una obra*
- *Software libre, freeware, shareware, adware*

3.4. PROGRAMACIÓN

- *Aspectos generales de Python como lenguaje*
- *Tipos de datos en Python*
- *Aspectos generales de variables y constantes en Python*
- *Funciones y paso de parámetros en Python*
- *Diccionarios: definición y usos principales*
- *Manejo básico de archivos en Python*
- *¿Qué es el software 2.0?*



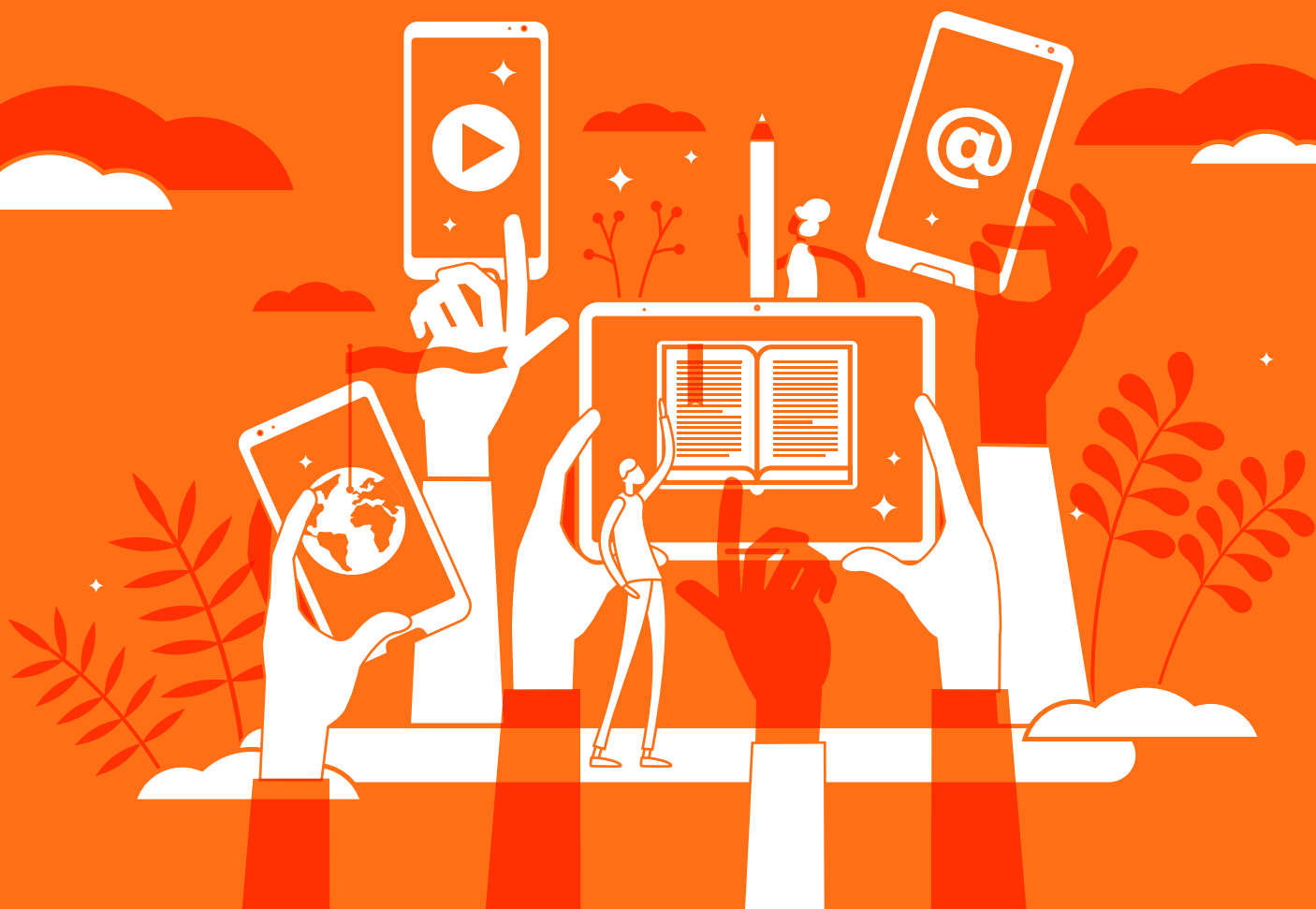


DigitAll

Creación de
contenidos digitales

3.1

DESARROLLO DE CONTENIDOS





Creación de
contenidos digitales

Nivel B2 3.1 Desarrollo
de contenidos

**Uso avanzado de
los procesadores
de texto: conexión
con otras
herramientas
ofimáticas**





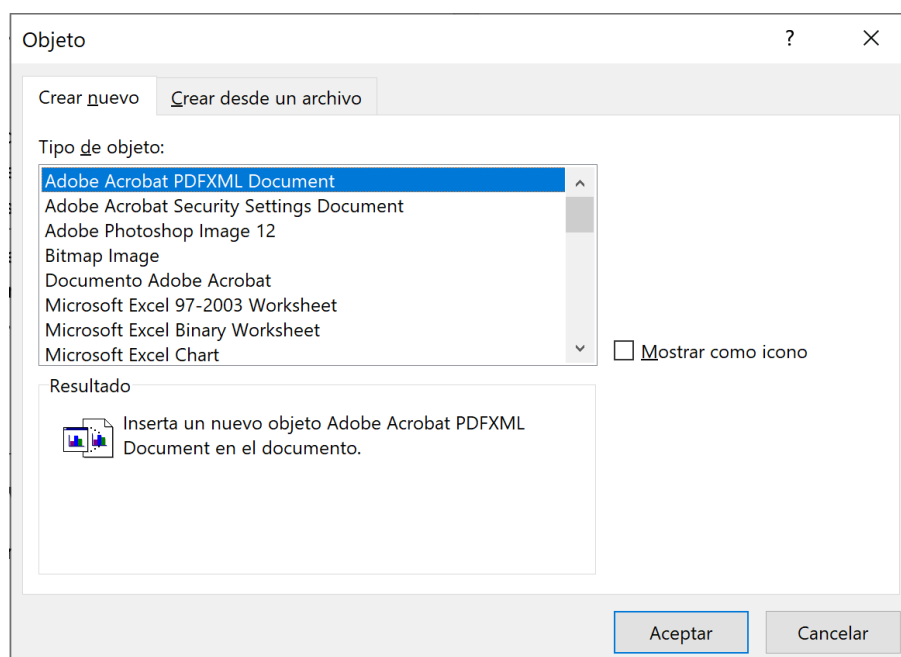
Uso avanzado de los procesadores de texto: conexión con otras herramientas ofimáticas

Herramientas ofimáticas conectables

Los procesadores de texto (nosotros usaremos Word de Microsoft como ejemplo) tienen la capacidad de conectarse con otras herramientas de ofimática para mejorar la productividad y eficiencia en el trabajo. Aquí te presentamos algunas alternativas como hojas de cálculo, presentaciones o archivos PDF.

Pasos para vincular otras herramientas a Word

Conectar los archivos es muy sencillo. solo se debe seleccionar **“Objeto”** en la pestaña **“Insertar”**.



En el desplegable que se abre podremos seleccionar herramientas de distintos tipos. Podremos añadirlas a Word creándolos de nuevas (“crear nuevo”) o creándolos a partir de un archivo ya existente. Una vez creado el vínculo con el objeto y teniéndolo ya insertado en el documento de texto podremos, haciendo doble clic sobre el objeto, acceder a todas las utilidades de la herramienta que haya sido insertada.



Hoja de cálculo en procesador de texto

Para conectar una hoja de cálculo de Microsoft Excel o Google Sheets en Word, también se pueden “vincular” los datos de esta manera:

- 1| Abre el documento de Word y la hoja de cálculo de Excel que deseas vincular.
- 2| Selecciona en la hoja de Excel la celda o rango de celdas que deseas vincular.
- 3| Haz clic derecho en la selección y selecciona la opción “Copiar”.
- 4| Vuelve al documento de Word y posiciona el cursor en el lugar donde deseas insertar los datos vinculados.
- 5| Haz clic derecho y selecciona “Pegado especial”.
- 6| En la ventana emergente, selecciona “Pegar vínculo” y haz clic en “Hoja de Cálculo”.

¡Listo! Los datos de la hoja de Excel se han vinculado al documento de Word.

Esto permite que se puedan actualizar los datos de la hoja de cálculo directamente desde el documento de Word e insertar tablas, gráficos y otros elementos que requieren cálculos o análisis.

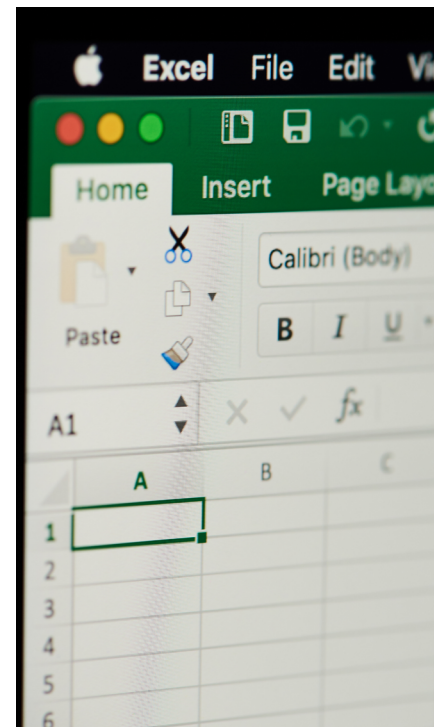
Utilidad: Si eres un profesor o estudiante y necesitas realizar una presentación en Word que incluya datos numéricos, podrías vincular estos datos a una hoja de cálculo de Excel para que se actualicen automáticamente en el documento de Word cada vez que se hagan cambios en la hoja de Excel.

Archivo PDF en procesador de texto

Los pasos a seguir son los que aparecen en el punto 1.1 de este documento.

La utilidad de vincular un PDF en un documento de texto queda ejemplificada en los siguientes ejemplos:

- 1| Imagina que estás escribiendo un informe y necesitas incluir algunos gráficos o imágenes que se encuentran en un archivo PDF. En lugar de tener que copiar y pegar cada imagen, puedes simplemente insertar el archivo PDF en tu documento de Word y tener acceso a todas las imágenes en un solo lugar.





2 Al insertar un archivo PDF en tu documento de Word, puedes estar seguro de que el formato se mantendrá igual, ya que el archivo PDF es una imagen fija y se incluye la totalidad del archivo.

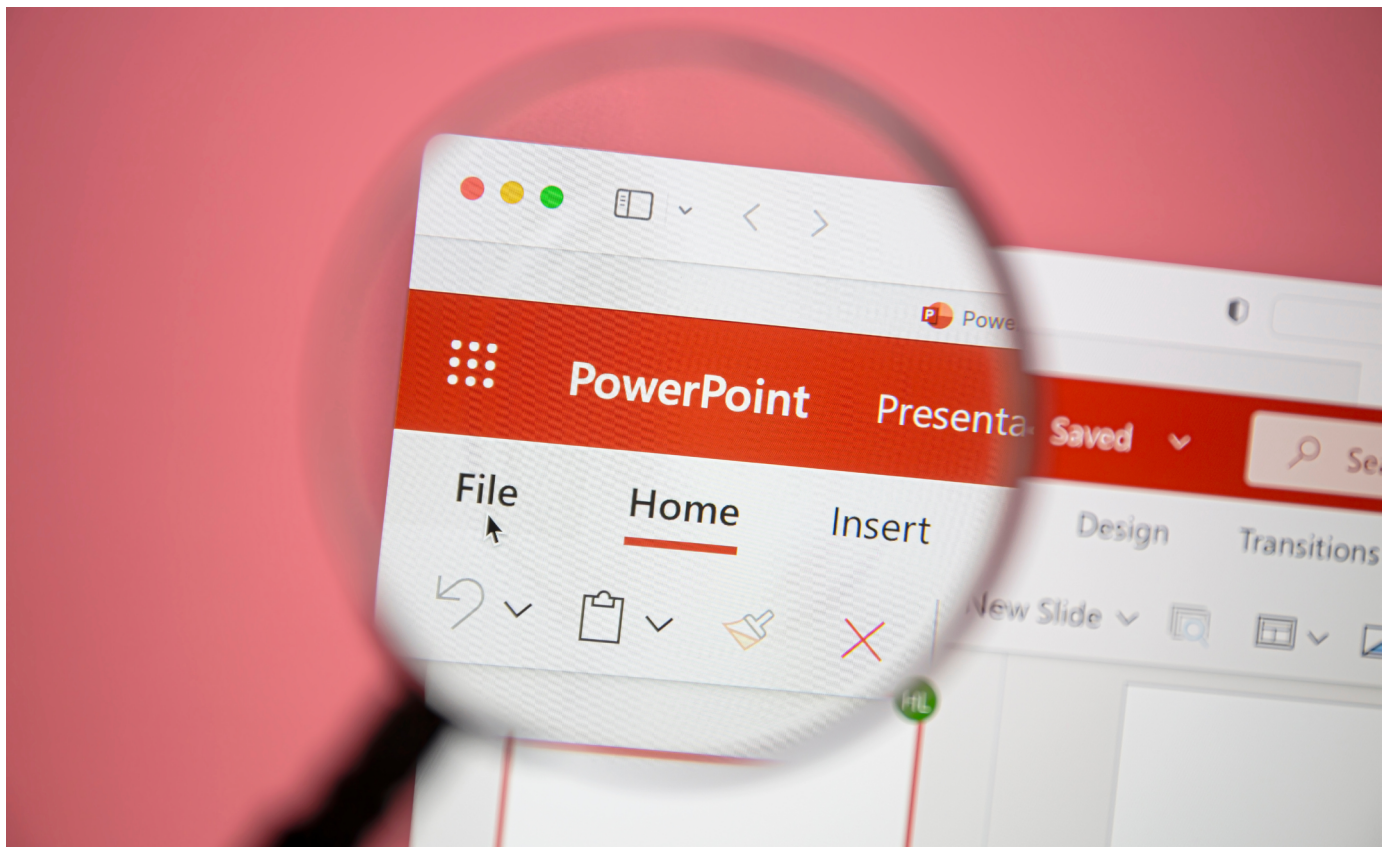
Añadir presentaciones en el procesador de texto

Por último, las herramientas para elaborar presentaciones como Microsoft PowerPoint también pueden conectarse con procesadores de texto como Word mediante la opción de "Insertar objeto", donde se puede insertar una presentación completa en un documento de Word.

Saber más

Conectar Word con otras herramientas, como hojas de cálculo en Excel o presentaciones en PowerPoint, facilita la creación de informes completos y profesionales. Los usuarios pueden importar datos y gráficos directamente desde otras aplicaciones, lo que agiliza el proceso de creación y mejora la precisión de la información presentada. Esto ahorra tiempo y reduce errores. Para más información puedes visitar la siguiente web:

e.digitall.org.es/objetos-vinculados





Creación de
contenidos digitales

Nivel B2 3.1 Desarrollo
de contenidos

**Uso de
herramientas
habituales
en procesadores
de texto libres
y online**





Uso de herramientas habituales en procesadores de texto libres y online

Muchas de las herramientas habituales en los procesadores de texto también están disponibles en los procesadores de texto libres y en línea, lo que significa que no es necesario comprar software costoso para tener acceso a estas funciones. Aquí se presentan algunos ejemplos de herramientas comunes que se pueden encontrar en los procesadores de texto en línea y libres.

Herramientas comunes

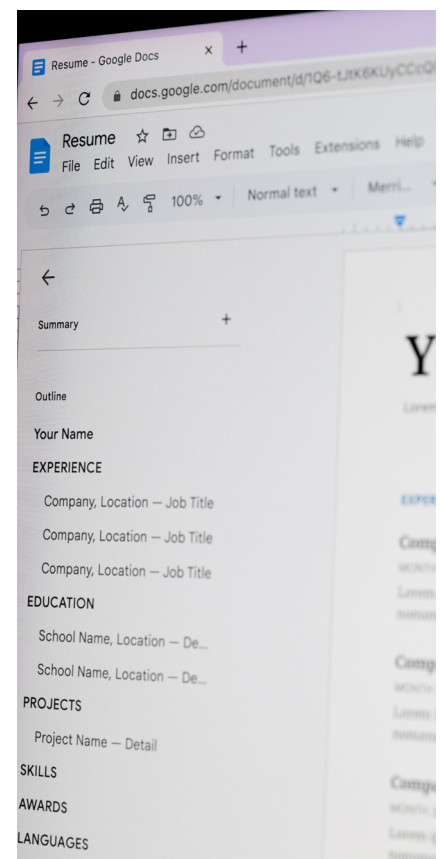
1 | Edición de texto: permiten crear, editar y dar formato a documentos de texto. Se pueden cambiar fuentes, tamaños, estilos, colores, alineación del texto, así como también se puede aplicar negrita, cursiva, subrayado y otros efectos de texto.

2 | Herramientas de formato: tanto los procesadores de texto libres como los procesadores de texto en línea tienen herramientas de formato que permiten ajustar el diseño del documento, como márgenes, espaciado entre líneas y párrafos, sangrías, entre otros.

3 | Insertar imágenes y otros elementos multimedia: permiten insertar imágenes, gráficos, tablas y otros elementos multimedia en los documentos. Estos elementos se pueden editar, ajustar, redimensionar y mover dentro del documento según sea necesario.

4 | Revisión de texto: tanto los procesadores de texto libres como los procesadores de texto en línea tienen herramientas de revisión de texto que permiten corregir la ortografía y gramática, así como también realizar seguimiento de cambios y comentarios en el documento.

5 | Guardar y compartir documentos: también permiten guardar documentos en diferentes formatos, como DOC, DOCX, PDF, RTF, entre otros. Además, se pueden compartir documentos con otros usuarios para colaborar en tiempo real o para revisión y comentarios.





6 | Plantillas y estilos: tanto los procesadores de texto libres como los procesadores de texto en línea ofrecen plantillas y estilos predefinidos que permiten dar un aspecto profesional y consistente a los documentos. Estas plantillas y estilos se pueden personalizar según las necesidades del usuario.

7 | Herramientas de búsqueda y reemplazo: ambos tipos de procesadores de texto tienen herramientas de búsqueda y reemplazo que permiten encontrar y reemplazar palabras o frases en el documento, lo que facilita la edición y corrección de documentos largos.

8 | Auto-guardado y sincronización en la nube: los procesadores de texto en línea suelen tener la ventaja de la sincronización en la nube, lo que permite que los documentos se guarden automáticamente en línea y se puedan acceder desde diferentes dispositivos. Esto facilita el trabajo colaborativo y el acceso a los documentos desde cualquier lugar y en cualquier momento.

Procesadores libres y on-line

Algunos ejemplos de procesadores de texto libres son LibreOffice Writer y Apache OpenOffice Writer, que son parte de las suites de código abierto y se pueden descargar y usar de forma gratuita en computadoras locales. Algunos ejemplos de procesadores de texto en línea son Google Docs, Microsoft Office Online, que son aplicaciones web que se pueden acceder a través de un navegador web y ofrecen herramientas de procesamiento de texto en la nube.

En la siguiente tabla se encuentran algunos programas libres de edición de texto y la página web donde localizarlos.

Alternativas de procesadores de texto

Programa	Compatibilidad con Office	Web
OpenOffice Writer	visualizar	openoffice.org
LibreOffice Writer	visualizar / modificar documentos	es.libreoffice.org
Google Docs	visualizar / modificar documentos	docs.google.com
WPS Office Writer	visualizar / modificar documentos	wps.com/office



i *Saber más*

Explora todas las capacidades de las herramientas de procesamiento de texto online. No sólo permiten la edición y el formateo de texto, sino también el almacenamiento en la nube, la colaboración en tiempo real, y la integración con otras aplicaciones útiles.



En esta página web puedes encontrar 12 alternativas gratuitas a Microsoft Word.

e.digitall.org.es/freeword-processors



Creación de
contenidos digitales

Nivel B2 3.1 Desarrollo
de contenidos

Efectos aplicados sobre texto



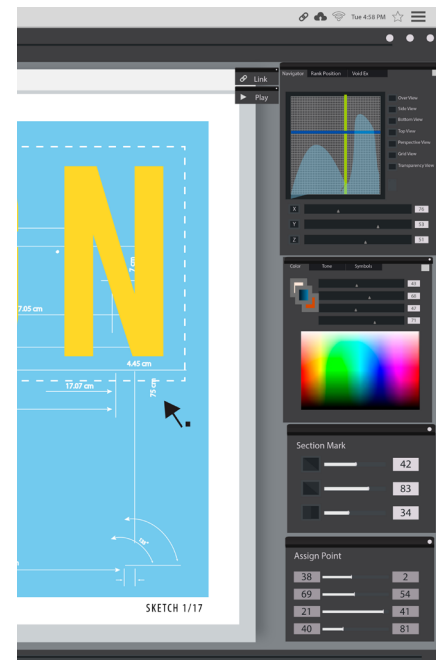


Efectos aplicados sobre texto

El texto en diseño gráfico no es simplemente una cuestión de escribir palabras en un lienzo, puede ser una herramienta poderosa y dinámica que transforma el trabajo y lo hace más atractivo.

Las herramientas de diseño gráfico, como Adobe Photoshop, Gimp o Illustrator, nos permiten trabajar con efectos de texto asombrosos, que van más allá de lo que un procesador de texto estándar, como Microsoft Word, puede hacer. Aunque Word ofrece una gama básica de efectos de texto, las herramientas de diseño gráfico ofrecen una mayor profundidad y flexibilidad.

A continuación, se describen algunas de las opciones de diseño de texto y sus principales utilidades.



Definiciones de algunos efectos comunes

1 | Bisel y Relieve.

El bisel y relieve son efectos que le dan al texto una apariencia 3D. Estos efectos pueden hacer que el texto parezca que está sobresaliendo o hundiéndose en la página. Los ajustes permiten controlar la profundidad del bisel y la dirección de la luz que crea la sombra y el relieve. El bisel crea una transición entre el texto y el fondo, generalmente haciendo que los bordes del texto parezcan levantarse por encima del fondo. El relieve, por otro lado, es similar al bisel, pero crea una apariencia de profundidad desde dentro del texto hacia fuera, lo que puede dar la impresión de que el texto está tallado o presionado en el fondo.

2 | Sombra paralela.

También conocida como sombra de caída o arrojada, la sombra paralela añade una sombra detrás del texto que da la impresión de que el texto está flotando encima del fondo. Se puede ajustar la distancia, el ángulo y la difuminación de la sombra para crear diferentes efectos.



3 | Iluminación Interior y Exterior (resplandor).

Estos dos efectos son esenciales para dar al texto una sensación de textura y dimensionalidad. Manipulan la iluminación del texto para dar la sensación de que hay una fuente de luz en la escena. La iluminación interior da la impresión de que la luz viene desde el centro del texto, mientras que la iluminación exterior crea la apariencia de que la luz viene desde fuera del texto. Estos efectos son útiles para crear texto metálico o de neón.

4 | Degradado (superposición de degradado).

Este efecto se utiliza para crear una transición suave de un color a otro dentro del texto. Puede ser utilizado para agregar profundidad, para crear un efecto de luz y sombra, o simplemente para hacer que el texto sea más interesante y atractivo visualmente.

Ejemplo en Adobe Photoshop y GIMP

A continuación, podremos un texto plano en un lienzo y modificaremos cada una de las herramientas explicadas arriba usando el desplegable de Estilo de capa de Adobe Photoshop (Imagen 1 - a la derecha).

De esta manera pasaremos de la Imagen 2 a la 3 (abajo) usando: sombra paralela normal con una opacidad de 77% y ángulo de 150°, resplandor interior con modo de fusión trama y opacidad del 75%, un bisel con inglete interior y profundidad de 300% y una superposición de degradado del 42%.

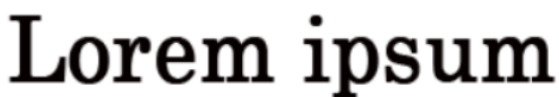


Imagen 2. Texto inicial.



Imagen 3. Texto editado.

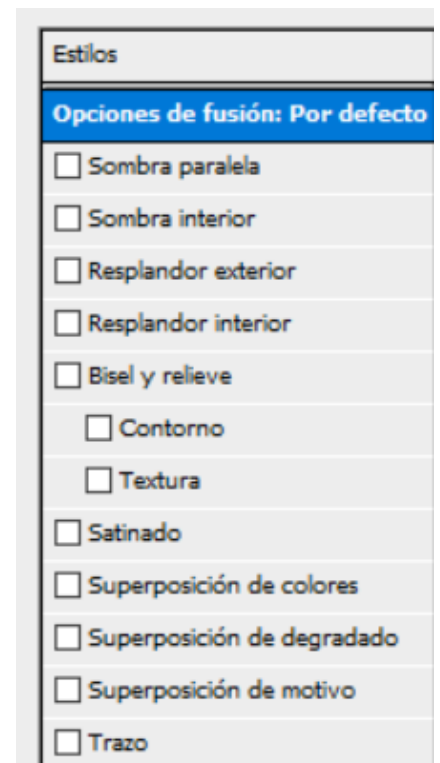


Imagen 1. Estilos de capa.



Para hacer lo propio en GIMP usaremos la pestaña de Filtros, con la que podremos añadir distintas características al texto como son la iluminación, sombra o texturas.

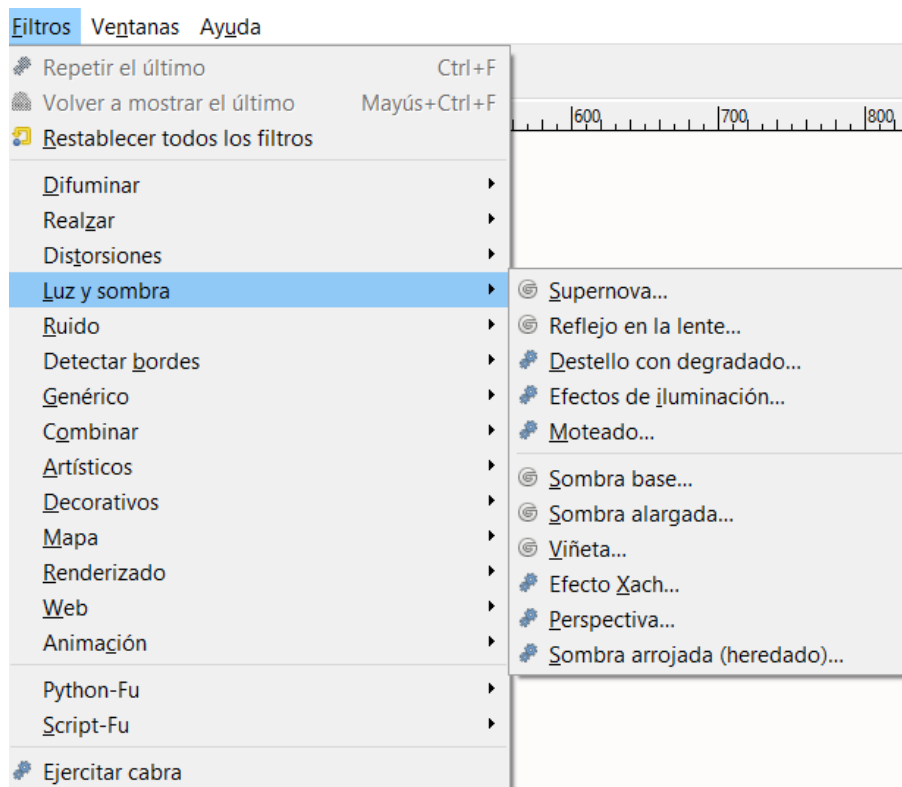


Imagen 4. Pestaña filtros.

Saber más

La experimentación es clave. Las herramientas de diseño gráfico suelen tener opciones preestablecidas para estos efectos, pero también permiten un gran grado de personalización. Es importante recordar que a veces menos es más. Los efectos de texto deben usarse para mejorar el mensaje que se está transmitiendo y NO para distraer de él.

Es recomendable visitar la página oficial de **Adobe** (adobe.com), o en **GIMP** (docs.gimp.org) donde se pueden encontrar tutoriales y guías paso a paso muy útiles.



Creación de
contenidos digitales

Nivel B2 3.1 Desarrollo
de contenidos

Creación de imágenes animadas





Creación de imágenes animadas

La creación de imágenes animadas es una disciplina que requiere práctica y paciencia. Sin embargo, con las herramientas adecuadas y el conocimiento correcto, puedes crear animaciones impresionantes que capturarán la atención de tu público.

A continuación, proponemos un paso a paso de cómo realizar estas imágenes.

Elegir el Software Adecuado

Una de las funciones más versátiles y útiles de GIMP es su capacidad para generar imágenes en formato GIF o PNG animado. Estos formatos son ampliamente utilizados en la web y otros medios digitales debido a su versatilidad y la capacidad de soportar animación. En este informe, describiremos paso a paso cómo generar este tipo de imágenes utilizando GIMP.

Procedimiento

Creación de capas para la animación

Cada cuadro de tu animación corresponderá a una capa en GIMP. Para crear una nueva capa, ve al menú 'Capa', luego a 'Nueva Capa'. Se abrirá una ventana donde podrás asignarle un nombre a la capa (por ejemplo, "Cuadro1"), definir el tamaño (si es distinto al de la imagen original) y el color de fondo.

Repite este proceso para cada cuadro de tu animación. Recuerda que el orden de las capas en el panel de capas (generalmente a la derecha del lienzo) determinará el orden de los cuadros en la animación. La capa superior corresponde al primer cuadro, y la capa inferior al último cuadro. Cada cuadro de tu animación será una capa separada en GIMP. Haz clic en 'Capa' > 'Nueva capa' para comenzar a crear tus cuadros.



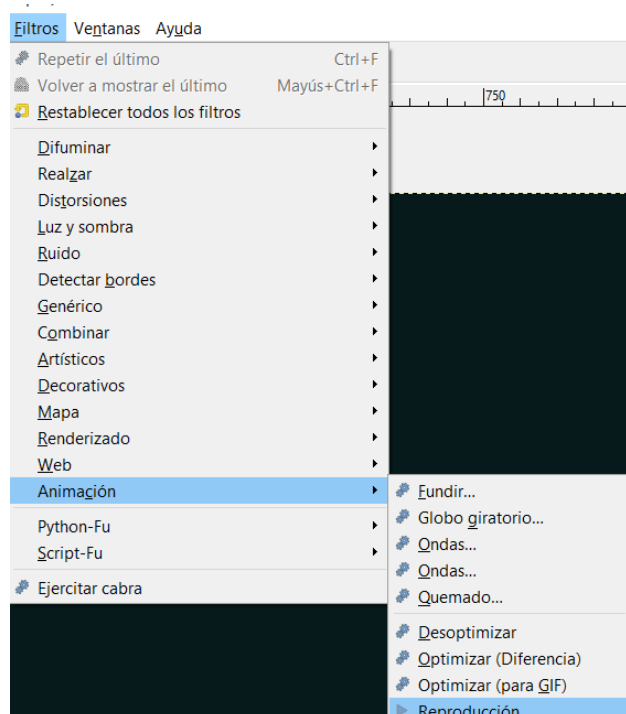


Creación de capas para la animación

Con las capas creadas, puedes empezar a diseñar cada cuadro de tu animación. Selecciona la capa que quieres editar en el panel de capas y utiliza las herramientas de GIMP para crear tu diseño. Por ejemplo, si quieres dibujar una línea en el primer cuadro, selecciona la capa "Cuadro1", elige la herramienta 'Lápiz' del panel de herramientas, elige el color y dibuja tu línea. Puedes incluir herramientas de dibujo, relleno, texto y más.

Previsualizar la animación

Una vez que hayas terminado de diseñar todas tus capas, puedes previsualizar tu animación. Ve al menú 'Filtros', luego a 'Animación' y selecciona 'Reproducir'. Se abrirá una ventana de previsualización donde podrás ver cómo se ve tu animación. Si necesitas realizar algún ajuste, cierra la ventana de previsualización y regresa a tus capas para hacer las modificaciones necesarias.





Exportar como GIF o PNG animado

Cuando estés satisfecho con tu animación, es hora de exportarla. Ve al menú 'Archivo' y selecciona 'Exportar como'. En la ventana de diálogo que se abre, elige el formato que desees (GIF o PNG), asigna un nombre a tu archivo y haz clic en 'Exportar'. En el siguiente cuadro de diálogo, asegúrate de marcar la casilla 'Como animación'. También puedes ajustar otros parámetros como el tiempo de retraso entre los cuadros y si la animación se repetirá indefinidamente. Una vez que hayas terminado, haz clic en 'Exportar'.



Saber más

La animación es un arte, y como todo arte, hay ciertos principios que se deben seguir para crear trabajos efectivos y atractivos. Los 12 principios de la animación, establecidos por los animadores de Disney Ollie Johnston y Frank Thomas, son un excelente punto de partida y son considerados la "Biblia" de la animación.

Si quieres más detalles te recomendamos ver los tutoriales del propio programa.

gimp.org.es/tutoriales/animaciones



Creación de
contenidos digitales

Nivel B2 3.1 Desarrollo
de contenidos

Herramientas de diseño gráfico libres y en la nube





Herramientas de diseño gráfico libres y en la nube

Introducción al diseño de imágenes

El diseño de una imagen es un tema central en muchas situaciones cotidianas o laborales que nos va a permitir mejorar desde nuestras propias fotos, si somos aficionados a la fotografía, hasta conseguir, a través del diseño gráfico, un currículum más atractivo que nos pueda diferenciar de otros candidatos para acceder a un puesto de trabajo.

Aunque para hacer un buen diseño gráfico siempre podremos recurrir a un profesional, existen en la actualidad una serie de herramientas que nos van a facilitar bastante la tarea a la hora de diseñar imágenes. Estas herramientas, por lo general, tienen en común que son capaces de proporcionar al usuario una serie de ideas de diseño predefinidas, en forma de plantilla, y permiten editarlas de forma sencilla, lo que facilita enormemente el diseño de nuestra imagen incluso si solo tenemos nociones básicas de diseño gráfico.

En este documento vamos a dividir estas herramientas en dos grandes bloques, las libres que podemos instalar en nuestro equipo y las que están disponibles en la web.

Herramientas libres para el diseño de imágenes 2D

Las herramientas libres para el diseño de imágenes 2D ofrecen una alternativa asequible, personalizable y de alta calidad a las herramientas de software propietario, lo que las hace una opción popular para diseñadores y artistas.

Entre las grandes ventajas de estas herramientas nos encontramos que son gratuitas para descargar y utilizar, y que son de código abierto, lo que permite que los usuarios trabajen en comunidad proponiendo mejoras o nuevas funcionalidades. En este campo, ya hemos trabajado con las dos seguramente más empleadas:

- **GIMP:** un editor de imágenes con una amplia gama de herramientas para la edición de fotos, la creación de gráficos y la ilustración digital.





- **Inkscape:** un software de gráficos vectoriales que se utiliza para crear ilustraciones, gráficos, logotipos y diseños complejos.

Sus alternativas más empleadas entre el software privativo son Adobe Photoshop y Adobe Illustrator, respectivamente.

Herramientas para trabajar en la nube

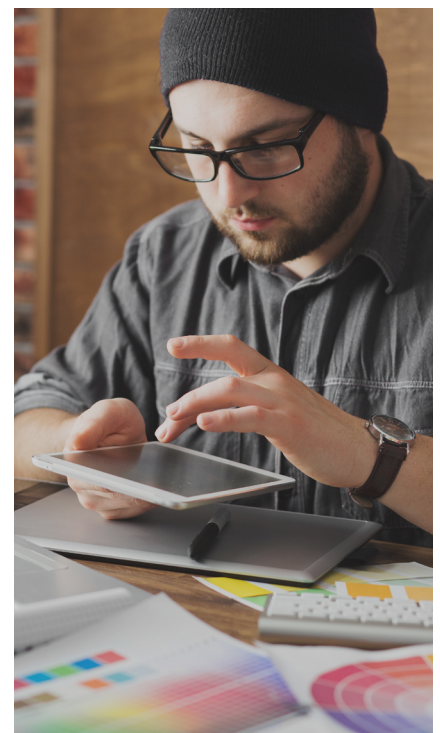
En la nube disponemos de multitud de herramientas para trabajar el diseño de imágenes. Aunque cada una tiene sus particularidades, podemos describir algunas características generales de todas ellas a continuación:

- **Plantillas:** cuentan con una amplia variedad de plantillas prediseñadas para crear diseños en diferentes formatos, como publicaciones en redes sociales, presentaciones, tarjetas de visita, entre otros.
- **Biblioteca de recursos:** ofrecen una amplia biblioteca de recursos, como imágenes, iconos y fuentes, para personalizar y mejorar los diseños.
- **Editor fácil de usar:** son intuitivos y fáciles de usar, con herramientas simples para agregar y editar elementos, ajustar colores, cambiar fuentes y más.
- **Colaboración:** permiten que los usuarios compartan y colaboren en proyectos, lo que es ideal para equipos de diseño o para trabajos en grupo.
- **Almacenamiento en la nube:** almacenan los diseños en la nube, lo que permite el acceso y la edición desde cualquier dispositivo con una conexión a Internet.
- **Multitud de formatos de salida:** permiten exportar nuestros diseños a multitud de formatos de imagen. Algunos permiten exportar a otros formatos como PDF o generar incluso presentaciones o vídeos cortos.

De todas las disponibles, seguramente Canva sea una de las más empleadas por lo que vamos a describir sus características principales brevemente.

NOTA

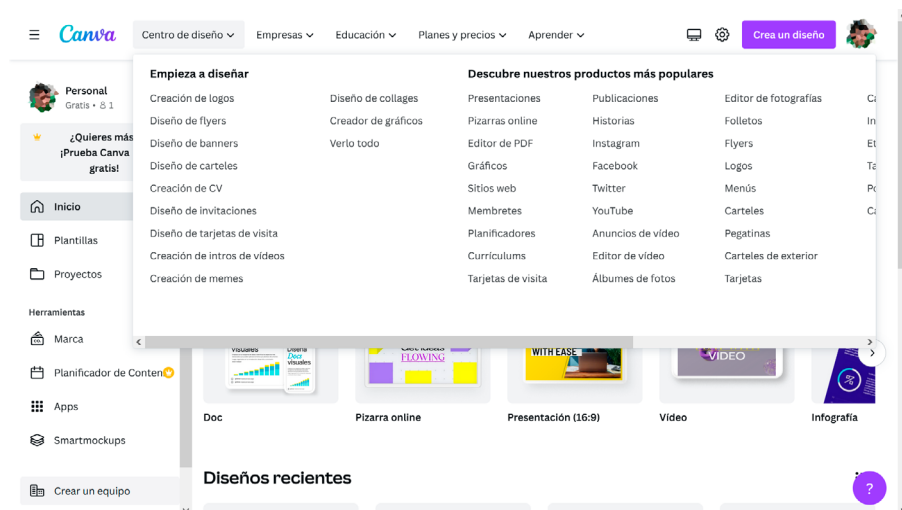
Recuerda que, además de estas herramientas, tienes disponibles bancos de imágenes libres, como Pixabay, que ponen a tu disposición multitud de imágenes de calidad.





Diseño de imágenes con Canva

Se trata de una de las herramientas de diseño más populares. No solo nos va a permitir diseñar imágenes de forma sencilla, sino que, además, como vemos en la imagen, posee una amplia biblioteca con plantillas que nos va a ayudar a crear nuestra imagen desde cero. Así, Canva nos ofrece multitud de plantillas agrupadas en temáticas (por ejemplo, Redes sociales) que a su vez se subdividen en otras más concretas (por ejemplo, Historias de Instagram). Por último, también nos permite, en vez de seleccionar una temática, seleccionar el tipo de documento que queremos crear (por ejemplo, una Infografía) de un amplio abanico y, posteriormente, nos ofrece plantillas para editar de forma sencilla este documento proporcionándonos por tanto una idea con una estética agradable. Además, dispone de varios planes de pago que mejoran las características de la cuenta gratuita ampliando las funcionalidades ya existentes que pueden resultar interesantes a nivel profesional. Por ejemplo, la cuenta gratuita proporciona 5 Gb de almacenamiento mientras que la profesional incrementa el espacio disponible hasta 1 Tb.



Detalle de la herramienta Canva en la que se muestran las plantillas ordenadas por bloques dentro del Centro de diseño.



Conclusiones

En resumen, estas herramientas para el diseño de imágenes son populares debido a su facilidad de uso, flexibilidad y accesibilidad. Particularmente, las herramientas en línea para el diseño de imágenes en la nube tienen la ventaja añadida de que los diseñadores pueden trabajar en sus diseños desde cualquier lugar con una conexión a Internet y también pueden compartir y colaborar en proyectos con otros diseñadores.

Aunque pueda parecer poco relevante, la colaboración en línea es una cualidad importantísima de estas herramientas, ya que permite a los miembros de un equipo que trabajan desde distintas localizaciones colaborar en un proyecto a tiempo real, lo que mejora la eficiencia, ya que los miembros del equipo pueden participar desde cualquier lugar con una conexión a internet, y la calidad del trabajo, ya que los miembros del equipo pueden aportar diferentes habilidades y perspectivas.

Saber más

Puedes animarte a realizar tus primeros diseños en la nube. Nosotros te recomendamos que uses Canva pues cumple con todas las características que le permiten ser una herramienta de éxito en este campo y, al ser muy empleada, es sencillo encontrar información e ideas nuevas que probar. Pero te dejamos aquí cinco alternativas que son también muy interesantes:

- **Adobe Express:** adobe.com/es/express
- **Desygner:** desygner.com/es
- **Piktochart:** piktochart.com
- **Visme:** visme.co/es
- **Vista Create:** create.vista.com/es



Creación de
contenidos digitales

Nivel B2 3.1 Desarrollo
de contenidos

Compresión de imágenes y vídeo por lotes





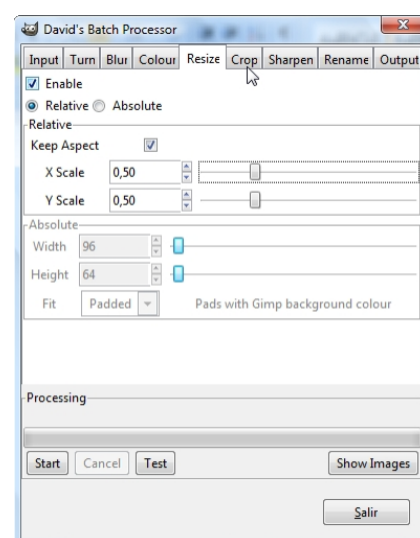
Compresión de imágenes y vídeo por lotes

Compresión de imágenes y vídeo por lotes

En caso de querer comprimir varias imágenes, es conveniente realizar esta tarea de manera simultánea para optimizar el tiempo y el trabajo. Una manera de hacerlo es comprimiendo por lotes. Esta tarea, se puede realizar mediante los siguientes pasos:

- 1 | Organizar las imágenes:** crear una carpeta o directorio en el ordenador y colocar todas las imágenes que se desean comprimir en ese directorio. Hay que asegurarse de que las imágenes estén en un formato compatible con la compresión, como JPEG o PNG.
- 2 | Elegir una herramienta de compresión por lotes:** existen varias herramientas disponibles que permiten comprimir imágenes en lotes de manera rápida y sencilla. Algunas opciones populares son:

- **Adobe Photoshop:** si se dispone de acceso a Adobe Photoshop, se puede utilizar su función "Procesador de imágenes" para comprimir imágenes en lotes. Simplemente creando una acción que realice la compresión y luego aplicándola a todas las imágenes en el directorio.
- **GIMP:** ya se ha hablado y trabajado en este curso con esta alternativa de software libre de edición de imágenes gratuito y de código abierto. Se puede utilizar la función "Procesador de lotes" para comprimir las imágenes deseadas. Es necesario configurar el procesador para aplicar la compresión a todas las imágenes en el directorio seleccionado.
- **Utilidades en línea:** existen numerosas herramientas en línea que permiten comprimir imágenes por lotes de forma gratuita. Algunas opciones comúnmente utilizadas incluyen *TinyPNG* (tinypng.com), *Compressor.io* (compressor.io) y *Optimizilla* (imagecompressor.com/es). Estas herramientas permiten cargar y comprimir múltiples imágenes al mismo tiempo.



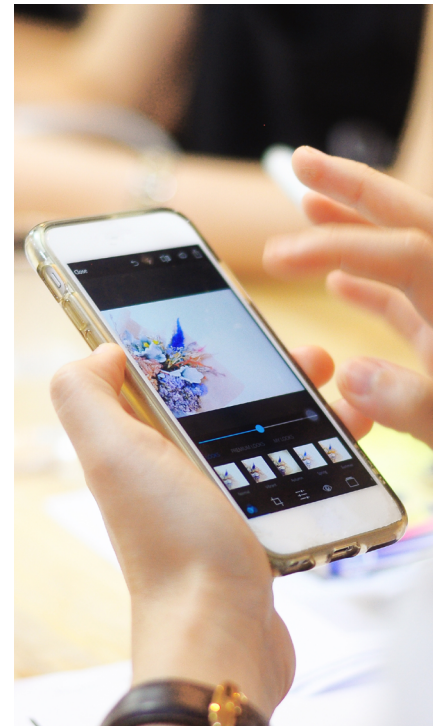
Procesado por lotes en GIMP. Permite redimensionar y comprimir un conjunto de imágenes y exportarla en una sola acción.



3 | Configurar la compresión: una vez seleccionada la herramienta a utilizar, es probable que haya que configurar los ajustes de compresión. Se puede elegir el nivel de compresión deseado, tal que, como ya se ha explicado en este curso, un mayor nivel de compresión reduce más el tamaño del archivo, pero también puede afectar la calidad de la imagen. Es necesario encontrar un equilibrio entre la calidad y el tamaño de archivo que sea adecuado para las necesidades.

4 | Iniciar el proceso de compresión: una vez configurados los ajustes de compresión, hay que iniciar el proceso de compresión por lotes. Esto puede implicar cargar las imágenes en la herramienta en línea o ejecutar el procesador de lotes en el software de edición de imágenes. La herramienta procesará todas las imágenes en el directorio seleccionado y las comprimirá según los ajustes.

5 | Guardar las imágenes comprimidas: una vez que se complete el proceso de compresión, la herramienta proporcionará las imágenes comprimidas. Será necesario guardarlas en una ubicación específica para que no se confundan con las imágenes originales. Para ello es recomendable crear un nuevo directorio o agregar un sufijo a los nombres de archivo para distinguir las imágenes comprimidas.



Es recomendable hacer una copia de seguridad de las imágenes originales antes de realizar cualquier compresión para evitar pérdida de datos en caso de que ocurra algún problema durante el proceso.

Es necesario encontrar un equilibrio entre la calidad y el tamaño de archivo que sea adecuado para las necesidades.



Compresión de vídeo por lotes

Comprimir varios vídeos en lotes puede ser una tarea útil para reducir el tamaño de los archivos y facilitar su distribución. Los pasos básicos para comprimir vídeos en lotes serán los mismos que con las imágenes:

1 | Organiza los vídeos: al igual que con las imágenes, será conveniente crear una carpeta o directorio en el ordenador y colocar todos los vídeos que se desean comprimir en ese directorio. Es recomendable asegurarse de que los vídeos estén en formatos compatibles con la compresión, como MP4, AVI o MKV.

2 | Elegir una herramienta de compresión por lotes: hay varias herramientas disponibles que permiten comprimir vídeos en lotes de manera eficiente. Algunas opciones son:

- **HandBrake:** es una herramienta gratuita y de código abierto que permite comprimir vídeos en lotes. Se pueden configurar los ajustes de compresión, como el tamaño, la calidad y el formato de salida, y luego aplicarlos a todos los vídeos en el directorio seleccionado.
- **FFmpeg:** es una solución de línea de comandos muy potente y versátil para procesar vídeos. Se puede escribir un script de lote que utilice FFmpeg para comprimir todos los vídeos en el directorio deseado. El script puede contener los comandos necesarios para ajustar los parámetros de compresión, como la tasa de bits, la resolución y el formato de salida.
- **Programas de edición de vídeo:** algunos programas de edición de vídeo, como Adobe Premiere Pro o Final Cut Pro, también tienen la capacidad de comprimir vídeos en lotes. Se pueden importar todos los vídeos en el programa, configurar los ajustes de compresión y luego exportarlos como un lote.

3 | Configurar los ajustes de compresión: una vez seleccionada la herramienta de compresión, se deben configurar los ajustes de compresión según las necesidades. Se pueden ajustar parámetros como la resolución, la tasa de bits, el códec de compresión y otros



aspectos que afecten la calidad y el tamaño del archivo resultante. Al igual que en las imágenes, es importante encontrar un equilibrio entre la calidad visual y el tamaño de archivo deseado.

4 | Llevar a cabo el proceso de compresión en lotes:

después de configurar los ajustes de compresión, hay que iniciar el proceso de compresión en lotes utilizando la herramienta seleccionada. Esto puede implicar cargar los vídeos en la herramienta y aplicar los ajustes de compresión en cada uno de ellos, o ejecutar un script de lote si se está utilizando alguna herramienta de línea de comandos.

5 | Guardar los vídeos comprimidos: una vez que se complete el proceso de compresión, la herramienta proporcionará los vídeos comprimidos. Una vez más, es necesario guardarlos en una ubicación específica para que no se mezclen con los vídeos originales.

La compresión de vídeo implica una pérdida de calidad, por lo que es recomendable hacer pruebas previamente con un único vídeo y ajustar los parámetros de compresión para encontrar el equilibrio adecuado entre calidad y tamaño de archivo según las necesidades.

Saber más

Si deseas saber más, puedes consultar el libro *"Compresión de audio y vídeo"* de Cliff Wootton (Editorial Anaya).





Creación de
contenidos digitales

Nivel B2 3.1 Desarrollo
de contenidos

Alternativas para la creación de sitios web





Alternativas para la creación de sitios web

Introducción

Antes de la existencia de las herramientas que vamos a describir, la mayor limitación que existía a la hora de crear una página web era que se necesitaba tener conocimientos de programación. Escribir código puede ser una habilidad compleja de aprender, y muchas personas y empresas no podían permitirse la contratación de desarrolladores web para crear su sitio web.

Sin embargo, con el surgimiento de plataformas accesibles y fáciles de usar para la creación de sitios web, esta limitación disminuyó significativamente. Ahora, cualquier persona con un ordenador (o incluso un móvil) y una conexión a Internet puede crear una página web de aspecto profesional y personalizada sin tener que aprender a codificar desde cero.

Posibilidades actuales

A continuación, te presentamos algunas de las opciones más populares para crear páginas web sin tener que escribir código desde cero. Cada una de ellas tiene sus aspectos positivos y negativos, y la opción óptima dependerá de las necesidades de cada usuario. En general, las podemos agrupar en dos categorías:

- **Plataformas online de construcción de sitios web:** son herramientas alojadas en la nube que normalmente se basan en sistemas de "agarrar y soltar" (del inglés, *drag and drop*), es decir, que los usuarios pueden agregar y personalizar elementos en su sitio web simplemente arrastrándolos y colocándolos donde deseen que aparezcan. Además, suelen presentar plantillas prediseñadas que los usuarios pueden personalizar. Aquí encontramos herramientas como Wix, Weebly, Squarespace, entre otras.





- **Sistemas de Gestión de contenidos (CMS, por sus siglas en inglés):** a diferencia de los anteriores, los CMS requieren un poco más de conocimiento técnico e incluso habilidades de codificación para explotar todo su potencial. Wordpress es el más popular y ampliamente utilizado, ofrece una gran cantidad de temas y plugins, es decir, una amplia gama de opciones, que nos van a permitir personalizar nuestro sitio web proporcionándole un aspecto profesional. Por ejemplo, Wordpress ofrece el plugin “Elementor” que nos permite editar de forma visual (mediante *drag and drop*) sin necesidad de escribir código. WordPress es una plataforma de creación de sitios web más adecuada para sitios web más grandes y complejos que requieren más flexibilidad y escalabilidad.

Como podemos suponer, estas herramientas han permitido tanto a particulares como a empresas tener una presencia en línea y expandir su alcance sin tener que invertir en habilidades técnicas costosas o recurrir a desarrolladores web.

Saber más

Direcciones de interés:

- **Wix:** es.wix.com
- **Weebly:** weebly.com/es
- **Squarespace:** es.squarespace.com
- **Wordpress:** wordpress.com
- **Visual Studio:** visualstudio.microsoft.com/es
- **Eclipse:** eclipse.org/ide
- **NetBeans:** netbeans.apache.org
- **IntelliJ IDEA:** jetbrains.com/idea

Entornos profesionales

Los desarrolladores web profesionales suelen trabajar en **entornos de desarrollo** (IDE, por sus siglas en inglés), que son aplicaciones que proporcionan a los desarrolladores herramientas para escribir, probar y depurar código. Básicamente, un IDE es un programa que permite a los desarrolladores crear aplicaciones, sitios web, software y otros programas. Los IDEs suelen incluir un editor de texto, herramientas de depuración, un compilador, un depurador, un generador de código y otras herramientas que ayudan a los desarrolladores a trabajar de manera más eficiente. Aunque



presentan menos limitaciones que los CMS, se necesitan tener conocimientos de programación para poder aprovechar al máximo sus funcionalidades. Algunos ejemplos de entornos de desarrollo populares son Visual Studio de Microsoft, Eclipse, NetBeans y IntelliJ IDEA.

Conclusiones

Hoy en día existen diversas alternativas para la creación de sitios web que van más allá de la creación de código desde cero. Los gestores de contenidos, como WordPress, ofrecen una manera más accesible y fácil para la creación de sitios web sin necesidad de tener conocimientos avanzados en programación. Además, existen otros entornos de desarrollo más avanzados para aquellos que posean conocimientos técnicos y deseen desarrollar sitios web más complejos y personalizados. En definitiva, estas alternativas permiten una mayor flexibilidad y versatilidad en la creación de sitios web, lo que resulta en una mayor satisfacción y experiencia de usuario.



NOTA

Un **tema** en WordPress es un conjunto de archivos que determinan el diseño y la apariencia de un sitio web. Los temas se descargan e instalan en la plataforma de WordPress y se pueden personalizar fácilmente para adaptarse a las necesidades y preferencias individuales de un sitio web.

Los **plugins** son pequeñas piezas de software que se pueden descargar e instalar en la plataforma de WordPress para agregar funcionalidades y características adicionales a un sitio web, como formularios de contacto, galerías de imágenes, integración de redes sociales, etc.

Saber más

Para entender mejor el sistema de Wordpress, puedes acceder a la documentación que proporciona la plataforma tanto para resolver las primeras dudas como para cuestiones más avanzadas.



es.wordpress.org/support



Creación de
contenidos digitales

Nivel B2 3.1 Desarrollo
de contenidos

Niveles de adecuación de WCAG



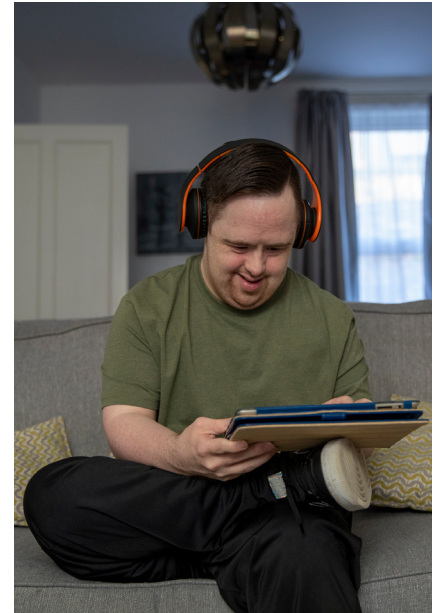


Niveles de adecuación de WCAG

Los niveles de conformidad del **WCAG (Web Content Accessibility Guidelines)** (e.digitall.org.es/niveles-WCAG) son una forma de clasificar y evaluar la accesibilidad de un sitio web. Se introdujeron en la versión 2.0 de las pautas de la WCAG, que fueron publicadas por el World Wide Web Consortium (W3C) en diciembre de 2008. Esta versión actualizada reemplazó el enfoque de prioridades de la versión 1.0 con un sistema de niveles de adecuación.

Estas pautas proporcionan recomendaciones detalladas y técnicas para hacer que el contenido web sea más accesible. En la versión 2.0 del WCAG, se establecieron un total de 12 pautas principales, agrupadas en cuatro principios fundamentales. Estos principios son:

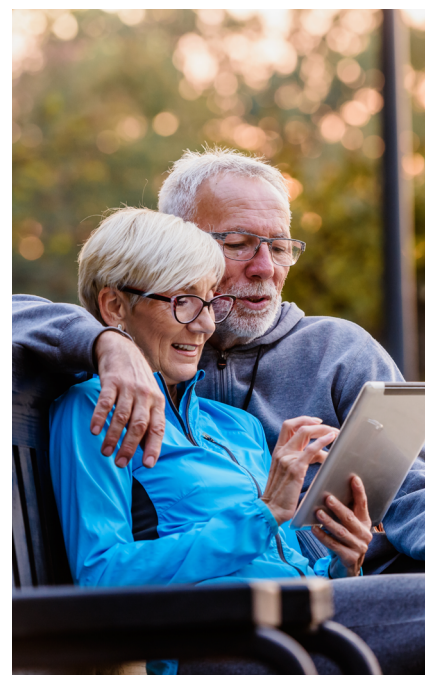
- 1 | Perceptible:** el contenido web debe presentarse de una manera que sea perceptible para todos los usuarios, incluidos aquellos con discapacidades visuales o auditivas. Las pautas de este principio abordan aspectos como el contraste de color, el uso de alternativas textuales para imágenes y medios, y la capacidad de controlar la presentación del contenido.
- 2 | Operable:** el contenido web debe ser operable para todos los usuarios, incluidos aquellos que tienen dificultades motoras o de navegación. Las pautas de este principio se centran en la navegación y la interacción, incluyendo elementos como la accesibilidad del teclado, la capacidad de omitir contenido repetitivo y el tiempo suficiente para leer y utilizar el contenido.
- 3 | Comprensible:** el contenido web debe ser comprensible para todos los usuarios, incluidos aquellos con discapacidades cognitivas o de comprensión. Las pautas de este principio abordan la claridad del lenguaje, la estructura y la organización del contenido, y la capacidad de proporcionar ayuda y orientación claras.
- 4 | Robusto:** el contenido web debe ser robusto y compatible con diferentes tecnologías y navegadores, para garantizar que pueda ser accedido por una amplia gama de usuarios. Las pautas de este principio se centran en el uso de estándares web, la compatibilidad con tecnologías de asistencia y la creación de contenido que sea resistente a errores y cambios en la tecnología.





Cada una de estas pautas se desglosa en criterios de conformidad específicos, que indican los requisitos técnicos y funcionales que deben cumplirse para lograr los niveles de adecuación (A, AA y AAA). Hay que tener en cuenta que las pautas y los criterios del WCAG continúan evolucionando y se han lanzado versiones posteriores, como la versión 2.2 publicada en mayo de 2022 y la versión 3.0 disponible y en continua evolución. Cada versión introduce nuevas pautas y criterios para abordar los desafíos de accesibilidad emergentes y mejorar la experiencia de los usuarios con discapacidades en la web.

Los criterios de conformidad específicos, que indican los requisitos técnicos y funcionales que deben cumplirse para lograr los niveles de adecuación, son los niveles A, AA y AAA.



Niveles de conformidad

El objetivo principal de introducir los niveles de adecuación fue proporcionar una estructura más clara y flexible para evaluar y mejorar la accesibilidad web. Los niveles de adecuación permiten un enfoque progresivo, donde los desarrolladores y diseñadores pueden trabajar hacia la mejora continua de la accesibilidad en diferentes etapas. Estos niveles están organizados en tres categorías: **A, AA y AAA**. Cada nivel tiene una serie de criterios de satisfacción que se deben cumplir para lograr ese nivel de conformidad. A continuación, se explican en detalle cada nivel y sus criterios de satisfacción:

Nivel de conformidad A

El nivel A establece los criterios mínimos de accesibilidad que se deben cumplir. Estos criterios son los requisitos básicos para lograr una cierta medida de accesibilidad y abordar las barreras más comunes que enfrentan las personas con discapacidades. Cumplir con el nivel A implica tomar medidas esenciales para mejorar la accesibilidad y garantizar que el contenido sea accesible para algunas personas con discapacidades.



Algunos criterios de satisfacción son:

- Proporcionar alternativas textuales para las imágenes.
- Utilizar una estructura de encabezados lógica para organizar el contenido.
- Garantizar que los formularios sean accesibles y puedan ser completados por teclado.

Nivel de conformidad AA

El nivel AA va más allá del nivel A y establece criterios adicionales más rigurosos. Cumplir con estos criterios proporciona una mayor accesibilidad y mejora la experiencia para un rango más amplio de personas con discapacidades. El nivel AA es el estándar recomendado y ampliamente adoptado, ya que proporciona un alto grado de accesibilidad y aborda muchas barreras comunes que enfrentan las personas con discapacidades visuales, auditivas, cognitivas y motoras.

Algunos criterios de satisfacción son:

- Proporcionar suficiente contraste entre el texto y el fondo.
- Proporcionar transcripciones de audio para los contenidos multimedia.
- Asegurar que los enlaces sean descriptivos y significativos.

Nivel de conformidad AAA

El nivel AAA es el nivel más alto de conformidad y establece los criterios más estrictos y completos para la accesibilidad. Cumplir con los criterios de nivel AAA lleva la accesibilidad a un nivel superior y garantiza una experiencia inclusiva para una amplia gama de discapacidades. Alcanzar el nivel AAA puede requerir cambios significativos en el diseño, la estructura y la funcionalidad del contenido web, y algunos de los criterios pueden ser difíciles de lograr en ciertos contextos.

Algunos criterios de satisfacción son:

- Proporcionar una navegación clara y consistente en todo el sitio web.
- Permitir a los usuarios ajustar el tiempo de espera antes de que el contenido cambie automáticamente.
- Proporcionar una versión simplificada del contenido para facilitar su comprensión.





Conclusiones

El nivel AA es generalmente considerado como el estándar recomendado y es ampliamente adoptado. Cumplir con los criterios de satisfacción del nivel AA proporciona una accesibilidad sólida y mejora la experiencia para la mayoría de los usuarios con discapacidades. El nivel AAA, aunque más riguroso, puede ser más difícil de alcanzar y algunos de sus criterios pueden ser difíciles de cumplir en ciertos contextos. Los niveles de adecuación no son obligatorios por ley en general (sí lo son en portales de administración pública), pero cumplir con ellos es muy recomendable y puede ser requerido en ciertos contextos legales o gubernamentales. Especialmente en docencia, el cumplimiento de los niveles de adecuación ayuda a garantizar una experiencia inclusiva para una amplia gama de usuarios, dando ejemplo y proporcionando beneficios tanto para las personas con discapacidades como para otros usuarios.

Saber más

Si quieres conocer en detalle los contenidos de la WCAG, puedes revisarlos en su página web.

[w3.org/TR/WCAG22](https://www.w3.org/TR/WCAG22)



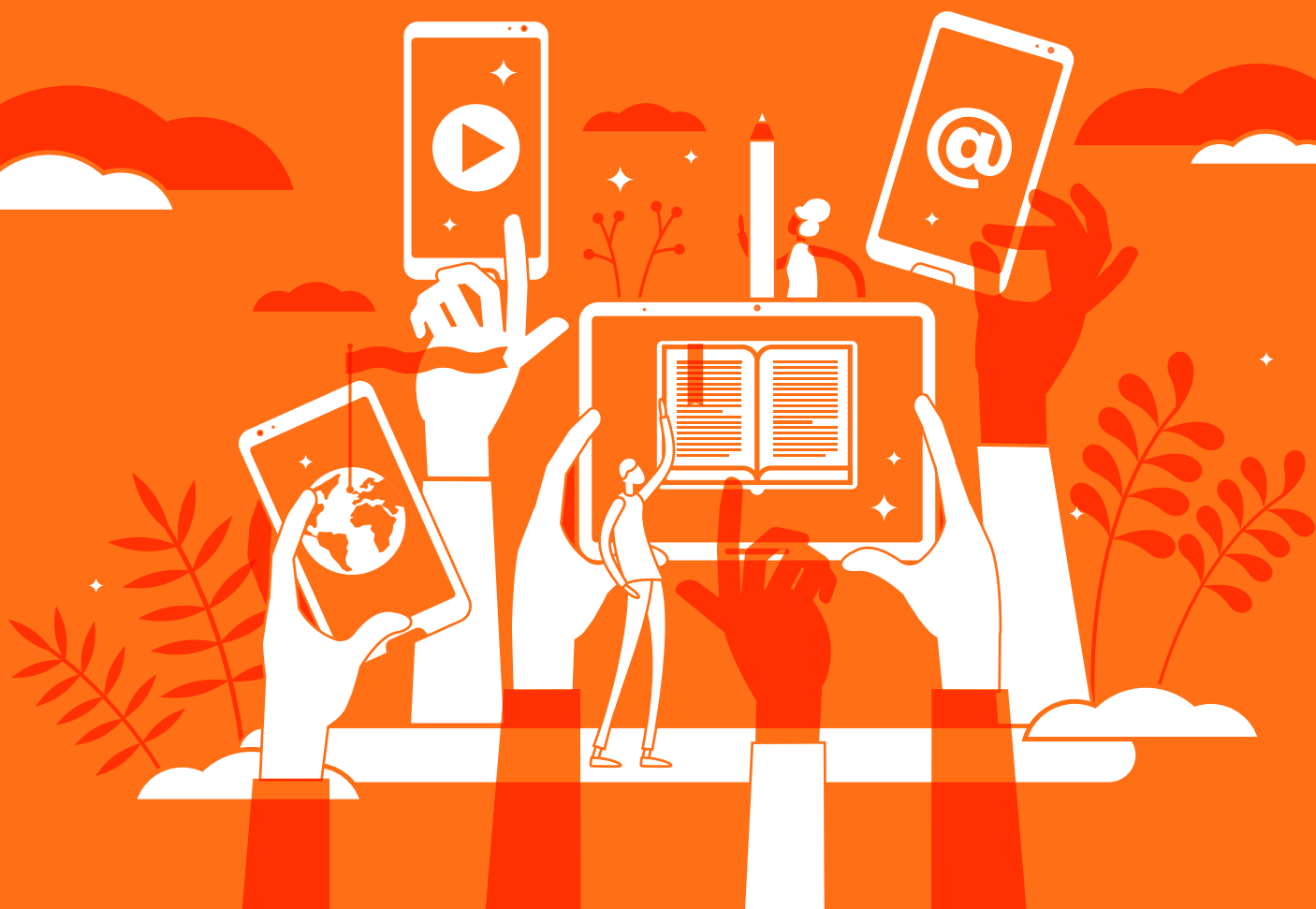


DigitAll

Creación de
contenidos digitales

3.2

INTEGRACIÓN Y REELABORACIÓN DE CONTENIDO DIGITAL





Creación de
contenidos digitales

Nivel B2 3.2 Integración y reelaboración
de contenido digital

Transiciones y animaciones avanzadas en presentaciones





Transiciones y animaciones avanzadas en presentaciones

Para que una presentación sea **efectiva**, esta debe incorporar elementos que impriman **estructura y ritmo** a sus diapositivas. En este sentido, las herramientas de creación de presentaciones ponen a disposición de sus usuarios tres elementos principales: **transiciones, animaciones y botones de acción**.

Transiciones

A la hora de cambiar de diapositiva cuando se muestra una presentación, es común visualizar efectos de movimiento entre una diapositiva y la siguiente. Estos efectos permiten dar continuidad a la presentación y a su contenido.

La **transición** de diapositivas es el **efecto visual** que se aplica sobre el espacio entre una diapositiva y la siguiente.

Añadir una transición a una diapositiva en una presentación PowerPoint es muy sencillo. Para ello, el usuario deberá realizar los siguientes pasos:

- 1 | Seleccionar** la diapositiva sobre la que se aplicará la transición.
- 2 | Clic** en la pestaña **Transiciones**. Elegir la transición deseada entre las disponibles. La transición "Ninguna" permite eliminar la transición de una diapositiva.
- 3 | Elegir** la dirección de la transición en el botón **Opciones de Efectos**.
- 4 | En** el grupo **Intervalos**, ajustar más opciones sobre la transición seleccionada: duración de la transición, aplicar la transición al hacer clic o al transcurrir un tiempo determinado, agregar un sonido a la transición o aplicar la transición a todas las diapositivas de la presentación.





Animaciones

Las animaciones dentro de una diapositiva fueron introducidas en el **documento A3C32B1D02 “Animaciones en presentaciones”**. En una diapositiva, es posible que un objeto, como un texto, imagen, gráfico o forma, contenga más de una animación. Cuando en una diapositiva hay múltiples animaciones, es común utilizar el **panel de animación**.



ANIMACIONES EN PRESENTACIONES

Documento referenciado:
A3C32B1D02

El **panel de animación** es el espacio que permite **administrar** las propiedades de las animaciones aplicadas a una diapositiva.

Para **trabajar con el panel de animación** en Microsoft PowerPoint, se deben realizar los siguientes pasos (Figura 1):

- 1 | Seleccionar** el objeto que se pretende animar dentro de la diapositiva.
- 2 |** En la pestaña **Animaciones**, seleccionar la opción **Panel de Animación**.
- 3 |** Cada animación aplicada sobre un objeto se **añadirá** al Panel de Animación.
- 4 |** Para **administrar** las **propiedades** de una animación, es posible hacer clic en la flecha que aparece a la derecha de la animación en cuestión.
 - a. Al hacer clic:** reproduce la animación al hacer clic con el ratón.
 - b. Iniciar con anterior:** reproduce la animación al mismo tiempo que el efecto anterior.
 - c. Iniciar después de Anterior:** reproduce la animación después de reproducir el efecto anterior.
 - d. Intervalos:** permite ajustar la duración de la animación y otras propiedades.

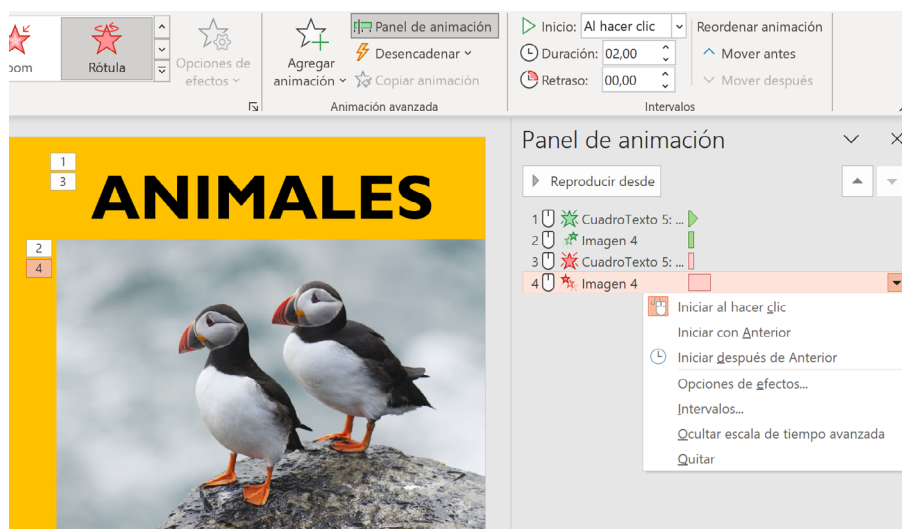


Figura 1. Principales opciones del panel de animación.

Botones de Acción

En muchas ocasiones es necesario que el presentador **interactúe** con las **diapositivas** durante la impartición de la presentación. Esto puede ocurrir, por ejemplo, para abrir un archivo local del presentador, acceder a una página web a través de un hipervínculo, reproducir una pista de audio del dispositivo o reproducir un vídeo externo. Para facilitar esta comunicación e interacción entre el presentador y la diapositiva, son muy útiles los botones de acción.

Un **botón de acción** es un objeto, concretamente una **forma**, que se inserta en una diapositiva y que realiza una **acción** cuando se pasa el ratón o se hace clic sobre él.

Para **insertar botones de acción** dentro de una diapositiva PowerPoint, han de seguirse los siguientes pasos:

- 1 | Seleccionar** la diapositiva donde insertar el botón de acción.
- 2 | Hacer clic** en la **Pestaña Insertar** y, después, en el botón **Formas**.
- 3 | Seleccionar** el botón deseado dentro del grupo **Botones de acción**. Dibujar la forma.



4 | Se abre un cuadro de diálogo. Seleccionar la **pestaña “Hacer clic con el mouse”** o **“Pasar el mouse por encima”** según cómo se quiere que se ejecute la acción asociada al botón.

5 | **Configurar la acción del botón** dentro del listado de acciones. Las más importantes son: crear un hipervínculo a cualquier diapositiva, archivo externo o dirección web; ejecutar un programa o reproducir un sonido.

Saber más

Transiciones de diapositivas en PowerPoint.

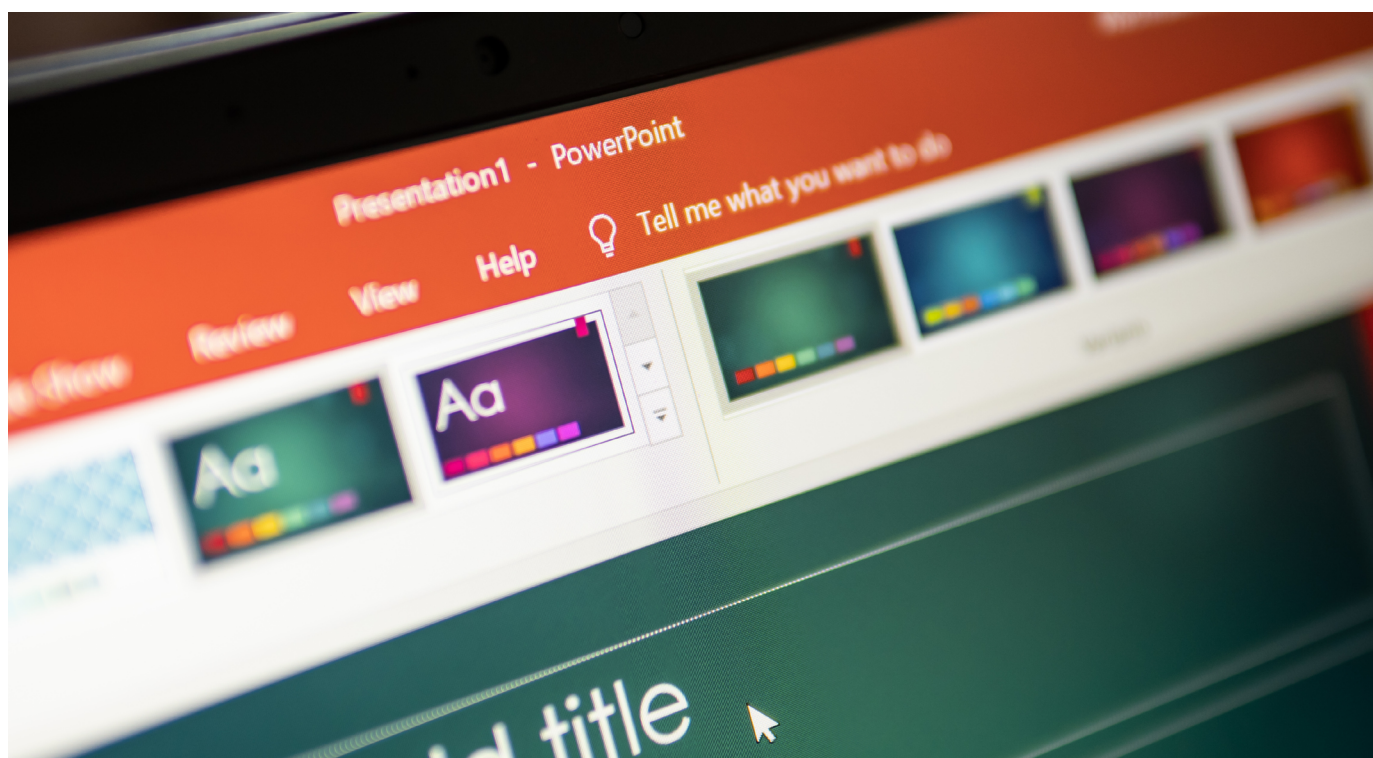
e.digitall.org.es/transiciones-diapositivas

El panel de animación en PowerPoint.

e.digitall.org.es/efectos-animaciones

Botones de acción en PowerPoint.

e.digitall.org.es/botones-accion





Creación de
contenidos digitales

Nivel B2 3.2 Integración y reelaboración
de contenido digital

Remasterización de vídeo y audio





Remasterización de vídeo y audio

Como se ha estudiado en los niveles anteriores, la calidad de los componentes de audio y vídeo es muy importante para generar una buena sensación en los consumidores de contenidos digitales. Sin embargo, como se ha visto en el **vídeo A3C32B2V02 “Análisis de calidad para la mejora de contenidos digitales”**, los contenidos pueden tener varias deficiencias que reducen la calidad de este, como por ejemplo un vídeo con baja resolución, un encuadre inadecuado, que el sonido sea difícil de percibir, etc. A continuación, se muestra cómo las técnicas de remasterización pueden ayudar a solventar estas problemáticas.

La remasterización es el proceso mediante el cual se modifica una obra sonora o audiovisual previamente grabada para mejorar la calidad de la grabación original.

⚠ ATENCIÓN

No se debe confundir la **remasterización** con la **reelaboración de contenido digital**, ya que en el proceso de remasterización el contenido no se modifica para obtener una versión diferente. La finalidad principal siempre será la de mejorar la calidad del sonido o imagen de la pieza audiovisual.

Remasterización de Audio

Antiguamente, los estudios utilizaban cintas para almacenar las grabaciones realizadas. Estas cintas eran posteriormente procesadas y remasterizadas para generar el denominado archivo **máster** que posteriormente se utilizaba para generar las copias que eran distribuidas. El problema con las grabaciones realizadas en cinta es que estas poseen ciertas limitaciones en cuanto a la calidad de sonido que permiten almacenar, así como el ruido que se genera al grabar en este formato. Por ello, en la actualidad, cuando se almacenan estas grabaciones antiguas en digital, estas son sometidas a un proceso de remasterización en el que se **elimina el ruido de fondo de las grabaciones**, se **reducen los defectos** producidos por deterioros en las cintas, y se **mejora el sonido** de los





distintos instrumentos. En el **vídeo A3C32BIV04 “Herramientas basadas en Inteligencia Artificial para modificar contenido digital”** se mostró cómo eliminar el ruido de fondo de grabaciones usando la herramienta **Podcastle.ai** (e.digitall.org.es/podcastle).

En la actualidad, los álbumes de estudio son grabados utilizando pistas separadas para cada uno de los instrumentos, lo que da lugar a piezas con más de 48 pistas en muchos casos. A continuación, los técnicos del estudio mezclan cada una de estas pistas de la manera deseada para obtener un fichero de dos únicas pistas (correspondientes a altavoces situados a izquierda y derecha) en un formato denominado estéreo. Al tener diferentes pistas para cada instrumento, se puede remasterizar la grabación para **acentuar distintos instrumentos, modificar la percepción de los instrumentos en el espacio**, o modificar las frecuencias de la señal para **ecualizar el sonido**. Una herramienta online muy útil es **Audioalter** (audioalter.com).

Finalmente, otro de los usos principales de la remasterización de audio consiste en **recuperar la definición de audio perdida durante la compresión** de este para su almacenamiento de forma eficiente. Esta técnica puede ser muy interesante para uso doméstico, ya que los ficheros de audio comerciales en la primera década de los 2000 utilizaban formatos con una alta tasa de compresión para reducir al máximo el tamaño del fichero (sacrificando la definición de sonido). El motivo de esta compresión era la reducida capacidad de almacenamiento de los dispositivos de reproducción de la época. La herramienta **Podcastle.ai** (e.digitall.org.es/magic-dust) también permite aumentar la calidad de las grabaciones.

Remasterización de Vídeo

La remasterización de vídeo es ampliamente utilizada en la actualidad, principalmente en la industria cinematográfica. Sus usos son muy variados ya que permite modificar una gran cantidad de aspectos sobre la grabación original. Por ejemplo, se pueden **modificar los planos de las tomas** para lograr diferentes encuadres o corregir una cámara mal alineada con respecto al horizonte. También pueden utilizarse para **modificar la saturación del color**, el contraste y el brillo, corrigiendo así las tomas demasiado oscuras o sobreexpuestas.



**HERRAMIENTAS
BASADAS EN
INTELIGENCIA
ARTIFICIAL PARA
MODIFICAR
CONTENIDO DIGITAL**

e.digitall.org.es/A3C32BIV04

Saber más

Una imagen **sobreexpuesta** se obtiene cuando la cantidad de luz que alcanza el sensor de la cámara es excesiva, lo que impide apreciar con claridad detalles de las partes más iluminadas.



Otro uso muy interesante de la remasterización de vídeo es la aplicación de herramientas de **coloreado de grabaciones antiguas en blanco y negro**. Mediante técnicas basadas en inteligencia artificial se logra “adivinar” cuál es el color de cada una de las partes de la imagen en cada fotograma del vídeo. Estas técnicas resultan muy llamativas, ya que logran traer al presente escenas de películas muy antiguas. Destaca, por ejemplo, la herramienta **DeOldify** (deoldify.ai) que utiliza redes neuronales profundas para colorear imágenes y vídeos. La Figura 1 muestra un ejemplo del uso de esta técnica.



Figura 1. “Toffs and Toughs” by Jimmy Sime (1937). Imagen obtenida usando DeOldify.

También es común aplicar técnicas para **aumentar la resolución de los vídeos**, adaptando las grabaciones con baja resolución a resoluciones FullHD o 4K. Esto permitiría proyectar el vídeo original en pantallas con una resolución mucho mayor evitando que el resultado se vea pixelado. Dentro de estas técnicas, se pueden usar herramientas como **Gigapixel AI** (topazlabs.com/gigapixel-ai) de Topaz Labs, la cual usa redes neuronales profundas para aumentar la resolución de los vídeos. Esta aplicación es usada por empresas como Netflix, Disney, Apple, Warner Bros, entre otros.



Otros usos de técnicas de remasterización de vídeo incluyen el **interpolado de fotogramas**, lo que permite dar suavidad a los vídeos al pasar estos de 24 o 30 fotogramas por segundo a 60 fotogramas por segundo. Destacamos la herramienta **Smoother AI** (e.digitall.org.es/smoothier-ai) que permite aplicar esta técnica fácilmente. También es común aplicar técnicas de **reducción de ruido** para evitar el ruido generado al almacenar vídeos antiguos en sistemas de almacenamiento que puedan tener pérdidas o estar dañados. Hay multitud de herramientas para lograr esta reducción de ruido, como por ejemplo **Neat Video** (neatvideo.com).

Como puede verse, las posibilidades en este campo son muy amplias. Además, el auge de las técnicas de inteligencia artificial en estos últimos años ha multiplicado las posibilidades y el número de herramientas disponibles para remasterizar audio y vídeo. Animamos al lector a que investigue sobre estas herramientas con sus propios vídeos, logrando así materiales de mayor calidad y supliendo las deficiencias comentadas en el **vídeo A3C32B2V02 “Análisis de calidad para la mejora de contenidos digitales”**.



**ANÁLISIS DE CALIDAD
PARA LA MEJORA
DE CONTENIDOS
DIGITALES**

e.digitall.org.es/A3C32B2V02





DigitAll

Creación de
contenidos digitales

3.3

DERECHOS DE AUTOR Y LICENCIAS DE PROPIEDAD INTELECTUAL





Creación de
contenidos digitales

Nivel B2 3.3 Derechos de autor y licencias
de propiedad intelectual

Dotando de copyright a una obra





Dotando de copyright a una obra

El contenido de este documento brinda conocimientos sobre el mecanismo de registro de la propiedad intelectual.

El Registro de la Propiedad Intelectual es único en todo el territorio nacional. Sus competencias están distribuidas entre el Estado y las Comunidades Autónomas. Por esta razón, las inscripciones realizadas en cualquiera de los Registros tienen la misma validez en todo el territorio español.

El Registro Central forma parte de la Administración General del Estado y depende del Ministerio de Cultura y Deporte. En todas las capitales de provincia de las Comunidades Autónomas de Islas Baleares, Canarias, Cantabria, Castilla y León, Castilla-La Mancha y Navarra, así como en las Ciudades Autónomas de Ceuta y Melilla existe una Oficina del Registro Central.

En las Comunidades Autónomas de Andalucía, Aragón, Asturias, Cataluña, Comunidad Valenciana, Extremadura, Galicia, La Rioja, Madrid, Murcia y País Vasco y en las Ciudades de Ceuta y Melilla existen Registros Territoriales que dependen de cada una de estas Comunidades o Ciudades Autónomas.

Las solicitudes de inscripción que se presenten en el Registro Central y a través de las Oficinas Provinciales que dependen de él, se pueden realizar:

- **Presencialmente**, aportando:
 - Los impresos oficiales de solicitud de inscripción.
 - El ejemplar de la obra, actuación o producción, en la forma y modo indicados en los impresos oficiales.
 - La documentación habitual, que, según los casos, se requiera en virtud de la legislación es:
 - ◇ **Datos del solicitante y autores:** fotocopia del DNI o documento acreditativo de la identidad si fuera extranjero. Si fuera menor o incapacitado se deberá presentar fotocopia del libro de familia o documento que acredite la patria potestad, tutela o curatela
 - ◇ **Datos del representante, si existiese:** el Poder de Representación.
 - Justificante del abono de la tasa correspondiente. El pago se efectúa previa liquidación que efectúa el Registro en el impreso oficial.

⚠ ATENCIÓN

Cada Registro Territorial tiene sus propias instrucciones de presentación de solicitudes, es por ello por lo que, para presentar una solicitud en un Registro Territorial, se deberá contactar previamente con él para informarse sobre la forma de hacerlo.



- **Telemáticamente** mediante la aplicación telemática **RePI** (e.digitall.org.es/repi) disponible en la Sede Electrónica del Ministerio de Cultura y Deporte. El solicitante deberá disponer de firma electrónica para realizar el procedimiento de registro vía telemática. Una vez que haya realizado el proceso de identificación y autenticación se deberán seguir los siguientes pasos:
 - **Cumplimentar la solicitud.** Para ello se deben completar los datos del solicitante y del representado, en caso de que existiera. También se deben introducir los datos de la obra que se quiere registrar como por ejemplo: tipo de obra, título de la obra y fecha de publicación o divulgación, en caso de que se trate de una obra divulgada o publicada. Además, se deben incluir los datos referidos al autor o autores, si la obra corresponde a uno, o varios autores. En este último caso deberá indicarse expresamente. Finalmente, también se podrá adjuntar documentos adicionales a la solicitud, cuando así lo indique expresamente la misma solicitud o el interesado lo crea necesario.
 - **Pago de las tasas.** La solicitud de primera inscripción requiere efectuar el pago de una tasa antes de firmar y enviar la solicitud. El pago se podrá realizar de forma telemática o en el banco.
 - **Firma y envío de la solicitud.** Una vez introducidos los datos necesarios en la aplicación, se deberá “Enviar” y a continuación “Firmar” empleando la firma electrónica.

Si todo ha ido bien en la tramitación telemática, se indicará que la solicitud ha sido registrada con éxito y se mostrarán todos los datos introducidos en la solicitud, incluyendo también el número del expediente que se ha creado, el número de registro y la fecha.

Además, habrá que aportar una copia electrónica del ejemplar identificativo de la obra. Para ello, una vez que se ha registrado una obra, el sistema ofrecerá una dirección web para proceder a la aportación de la copia del ejemplar identificativo.

⚠ ATENCIÓN

Cuando se hace un registro hay que indicar claramente el tipo de OBRA. Se debe seleccionar el tipo y subtipo de obra. Para algunos tipos es posible introducir los detalles específicos de la obra. Estos detalles no hay que incluirlos para los siguientes tipos de obra: coreografía y pantomima, bases de datos, programas de ordenador, página web o multimedia, mera fotografía, derecho “sui generis” sobre una base de datos

**NOTA**

Cuando existan varios autores, a la hora de indicar los "Autores" se deberá indicar cuál de los tipos de autoría señalados en este apartado se atribuye a cada uno de ellos.

Saber más**Oficinas dependientes de los Registros Territoriales:**

e.digitall.org.es/registro-territorial

Oficinas dependientes del Registro Central:

e.digitall.org.es/registro-central

Legislación estatal:

- Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia (e.digitall.org.es/boe-8930).
- Real Decreto 281/2003, de 7 de marzo, por el que se aprueba el Reglamento del Registro General de la Propiedad Intelectual (e.digitall.org.es/boe-6247).
- **Legislación autonómica:** e.digitall.org.es/legislacion-autonomica





Creación de
contenidos digitales

Nivel B2 3.3 Derechos de autor y licencias
de propiedad intelectual

**Software libre,
freeware,
shareware,
adware**





Software libre, freeware, shareware, adware

El software también es un producto intelectual que puede y debe protegerse. Un programador que ha desarrollado un valioso y útil software se puede plantear qué hacer con su software, con su programa informático, para que pueda ser utilizada por terceros. Este permiso de utilización a otra persona, incluidas también la copia, la modificación, la distribución y/o la venta, con los límites y condiciones que tú quieras establecer, se formaliza mediante la **licencia del software**. Esta licencia es un contrato bajo el cual un titular de un elemento protegido por propiedad intelectual (por ejemplo, nuestro software) concede su autorización de uso a otra persona, a un tercero, con las condiciones establecidas en las cláusulas. Aunque si lo que desea es vender la titularidad de su programa deberá firmar un contrato de **CESIÓN** de software. El titular o autor del software es el licenciante y el usuario es el licenciatario.

¿Qué tipo de licencia se puede aplicar al software? En función de las condiciones o permisos o autorizaciones que se desee imponer a los licenciatarios, y **dependiendo de la plataforma en la que desear ubicarlo**, se distinguen los siguientes tipos de licencias:

- **Licencia GPL (GNU LGPL).** Es software libre y permite que se enlace con módulos no libres. Debe estar disponible y accesible para copias ilimitadas y a cualquier persona que lo solicite. En su primera versión se llamaba 'Licencia Pública General para Bibliotecas de GNU'.
- **Licencia BSD.** Es una licencia permisiva, pues casi no impone condiciones sobre lo que un usuario puede hacer: el software puede ser vendido y no es obligatorio incluir el código fuente.
- **Licencias MPL.** Son las de Mozilla Firefox, Mozilla Thunderbird y la mayoría de otros software de Mozilla y algo menos permisivas que las anteriores.





- **Licencia Debian** (Debian Free Software Guidelines-DFSG). De ella deriva la licencia de Open Source Initiative. Establece la exigencia de publicación del código fuente, la redistribución libre (también del trabajo derivado bajo la misma licencia del original) y que no puede discriminar a ninguna persona o grupo de personas, ni el software ni ninguna forma de su utilización.
- **Licencia Open Source Initiative.** Una de las licencias más conocidas y utilizadas, llamada también de fuentes abiertas o código abierto, que permite la libre distribución (también del código fuente), la modificación del mismo, los desarrollos derivados y su redistribución en las mismas condiciones que el software original. Además, Open Source establece la integridad del código fuente del autor, que no debe ser discriminatoria de persona alguna o grupos de personas, ni restringir la utilización del software a campos de dominio o actividad y debe ser neutral en relación con la tecnología.
- **Licencia de X.org** (X Windows System). Distribuida por el Consorcio X, como organización que controla su desarrollo, es usada en los sistemas operativos Linux y UNIX nacidos como alternativa a los sistemas gráficos como Microsoft. Es software libre, aunque sin adherirse al copyleft.
- **Licencia Freeware.** Autoriza el uso del software de forma libre y gratuita, pero en general a particulares y no a empresas u organismos oficiales. La **licencia Donationware** es similar, pero pide a cambio un donativo, aunque no sea obligatorio. La **licencia Postcardware** es similar, pero solicita el envío de una postal como confirmación de su utilización, en lugar del donativo.
- **Licencias de prueba o Shareware.** Solicitan un registro previo y autoriza el uso de un programa para que el usuario lo evalúe y posteriormente lo compre. La licencia de Demo es la sesión de un programa para su evaluación. Son muy populares en el caso de pruebas de juegos, por ejemplo.

Finalmente, si no se desea que el usuario del software no lo modifique, ni lo comparta, ni lo distribuya sin la autorización del creador, la **licencia sería de tipo privativo**, como son, por ejemplo, las licencias del sistema operativo macOS de Apple o Windows.



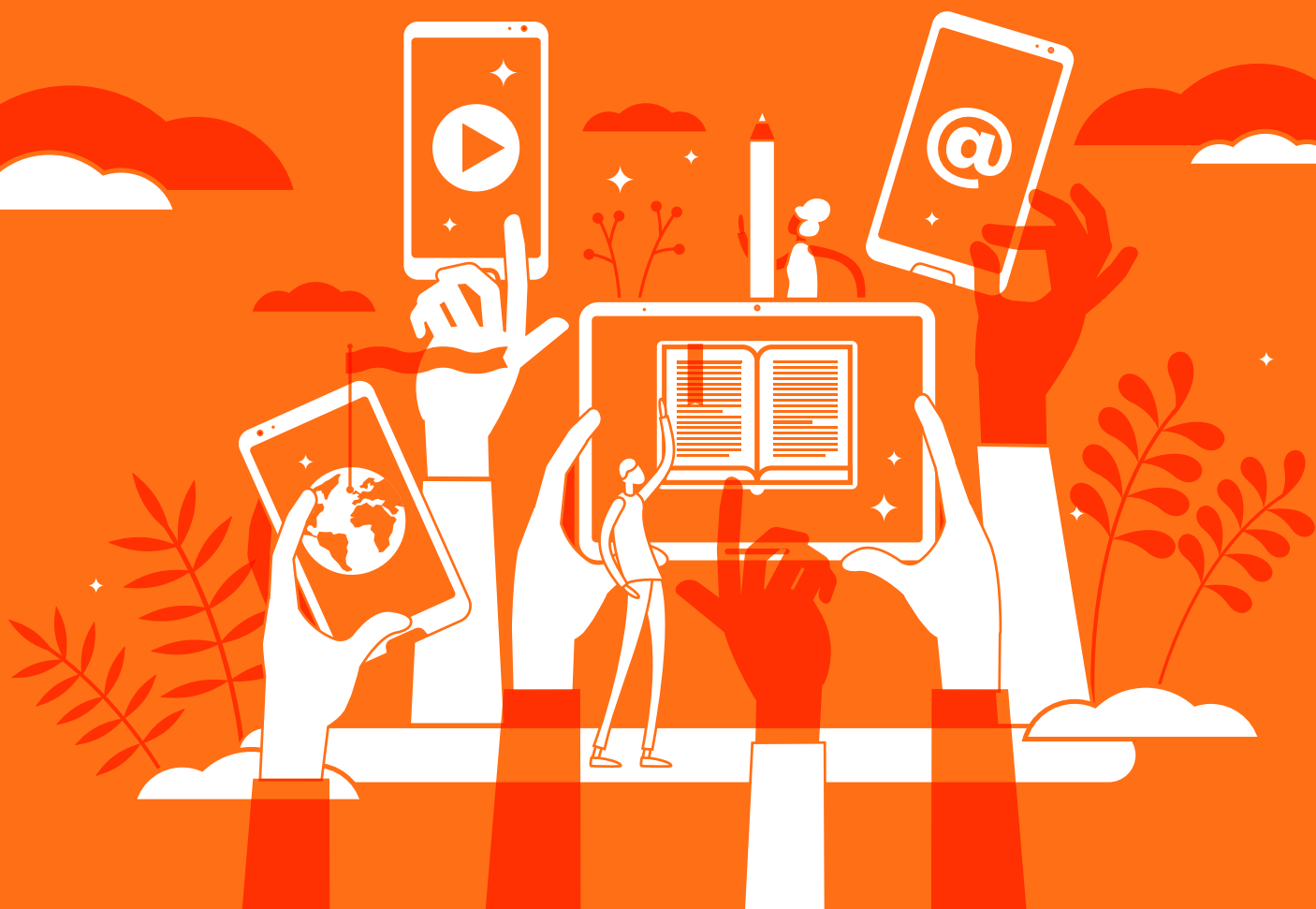


DigitAll

Creación de
contenidos digitales

3.4

PROGRAMACIÓN





Creación de
contenidos digitales

Nivel B2 3.4 Programación

Aspectos generales de Python como lenguaje





Aspectos generales de Python como lenguaje

Introducción

Cuando uno inicia su camino de aprendizaje en el mundo de la programación, una de las decisiones a tomar es “qué lenguaje de programación emplear”. Python puede ser una buena elección por varias razones: se trata de un lenguaje versátil, flexible, con una curva de aprendizaje relativamente suave en comparación con otros lenguajes, multiplataforma y totalmente gratuito.

En esta sección se introducirán los aspectos léxicos, sintácticos y semánticos de Python. Se presentará la estructura general de un programa en Python, introduciendo y familiarizando al estudiante con los diferentes niveles conceptuales que suelen verse en código Python. Finalmente, se discutirá el concepto de indentación en Python y la ausencia de llaves/sentencias para delimitar el alcance de instrucciones o estructuras de control.



Aspectos léxicos, sintácticos y semánticos de Python

Cuando se utiliza un lenguaje para expresarnos, se hace uso de las herramientas que proporciona como el alfabeto, el vocabulario con el conjunto de palabras que lo forman y su significado, y la gramática que representa el conjunto de reglas que determinan el orden en el que deben aparecer las palabras. En el caso de un lenguaje de programación, los elementos disponibles son similares.

El **léxico** de un lenguaje de programación, como Python, determina los símbolos y palabras que podemos emplear para construir las sentencias que se incluyen en los programas. El **análisis léxico** de Python desglosa un archivo (con el programa que hemos escrito) en *tokens* según reglas básicas que gobiernan el modo de escribir programas en este lenguaje.

A continuación, el **analizador sintáctico** de Python se encarga de analizar la estructura sintáctica del programa. Utiliza la salida del analizador léxico para construir un árbol de sintaxis abstracta (ASA) que representa la estructura sintáctica del programa de entrada. Con esta estructura, se pueden realizar varias tareas, la



detección de errores sintácticos o la generación de código objeto. Finalmente, el **analizador semántico** hace uso de la salida generada por el analizador sintáctico para detectar posibles errores que afectan a la semántica del lenguaje, como la verificación de tipos, acceso a variables y métodos, rango de valores, asignaciones válidas, variables no definidas u operaciones ilegales, entre otros.

Para que un programa en Python pueda ser ejecutado, éste debe estar libre de errores léxicos, sintácticos y semánticos.

Estructura general de un programa en Python

Un programa en Python suele tener la siguiente estructura general:

1 | Comentarios: son líneas de texto que explican el propósito del programa y cómo funciona. Se trata de un elemento opcional, a criterio del programador, y es ignorado por el intérprete de Python. Por tanto, no produce ningún tipo de error.

```
# Este es un comentario en una línea
"""
Este es un comentario en varias líneas
utilizando comillas triples
"""
```

2 | Declaración de importaciones: son líneas de código que permiten importar módulos o bibliotecas que contienen funciones y clases útiles para el programa.

```
import math
print(math.pi) # Imprime el valor de pi (3.141592653589793)
```

3 | Definición de funciones: son bloques de código que realizan una tarea específica y pueden ser llamadas varias veces en el programa. Las funciones se definen utilizando la palabra clave **“def”** seguida del nombre de la función y los parámetros entre paréntesis.

```
def suma(a, b):
    resultado = a + b
    return resultado
```



4 | Declaración de variables: son líneas de código que declaran y asignan valores a variables que se utilizarán en el programa. Estas variables pueden ser sencillas (almacenan un único valor) o estructuras de datos más complejas, como arrays, vectores, listas, o diccionarios, entre otros.

```
# Asignar valores a variables
numero = 10
nombre = "Jose Manuel"
es_activo = True
```

5 | Bloque principal: es el cuerpo principal del programa donde se realizan las operaciones y se llaman a las funciones definidas anteriormente. Tanto en el bloque principal como en las funciones, se suelen emplear estructuras de control de flujo, como las condicionales y los bucles.

Indentación en Python

La **indentación** en Python es el uso de espacios o tabulaciones al comienzo de una línea de código para indicar la jerarquía y estructura del programa. En Python, las indentaciones son utilizadas para indicar el alcance de las estructuras de control de flujo (como condicionales y bucles) y de las funciones.

La ausencia de llaves o sentencias para delimitar el alcance de instrucciones o estructuras de control en Python es una **característica distintiva del lenguaje**. En lugar de utilizar llaves para delimitar el alcance, tal como se hace en muchos otros lenguajes de programación, se utilizan indentaciones. A continuación, se muestra un ejemplo en el que se representa, en primer lugar, un sencillo fragmento de código con el uso de llaves para delimitar el bloque, y su equivalente en Python empleando las indentaciones:

```
if x > 0: {
    print("x es mayor a cero")
}
```

Código en Python:

```
if x > 0:
    print("x es mayor a cero")
```

⚠ ATENCIÓN

Las indentaciones en Python no son un aspecto puramente estético y, en el caso de no respetarlas, producirán errores de sintaxis. Se trata de una característica particular de Python que hace que el código sea más legible y fácil de comprender, facilitando, de forma clara, la visualización de la jerarquía y la estructura de los programas.



Creación de
contenidos digitales

Nivel B2 3.4 Programación

Tipos de datos en Python





Tipos de Datos en Python

Introducción

Como en cualquier otro lenguaje de programación, Python admite el almacenamiento de datos en variables, y cada variable puede tener un tipo de dato asociado. En la siguiente sección se ofrecerá una visión general de los tipos de datos básicos que ofrece Python y un ejemplo de uso de cada uno de ellos.

Tipos de datos básicos en Python

1 | Boolean Type (bool): en Python, el tipo de datos 'bool' representa un valor lógico, es decir verdadero (True) o falso (False). Este tipo de datos suele ser utilizado habitualmente en expresiones condicionales y operaciones lógicas. Un ejemplo sencillo podría ser el siguiente:

```
x = True
y = False
if x:
    print("x es verdadero")
else:
    print("x es falso")
```

2 | Text Type (str): el tipo de dato 'str' en Python representa una secuencia de caracteres (texto) y puede ser definido entre comillas simples o dobles. A continuación, se muestra un ejemplo sencillo en el que se crea una cadena de texto y se muestra por pantalla:

```
mensaje = "Hola Mundo"
print(mensaje)
```

3 | Numeric types (int, float): Python ofrece la posibilidad de representar tipos de datos numéricos. Los más habituales son 'int', que permite representar números enteros y 'float', que representa números con decimales. Se trata de un tipo de dato que permite la realización de operaciones matemáticas como suma, resta, multiplicación o división, entre otras. Algunos ejemplos sencillos de uso de estos dos tipos de datos son los siguientes:





```
num1 = 5
num2 = 10
resultado = num1 + num2
print(resultado)
```

```
num1 = 5.5
num2 = 10.1
resultado = num1 + num2
print(resultado)
```

Si se realizara una operación entre un entero y un número en punto flotante, el resultado sería un número en punto flotante.

4 | Sequence Types (list, tuple, range): los tipos de datos de secuencia en Python, tal como su propio nombre indica, representan secuencias de elementos. Estos tipos incluyen *list*, *tuple* y *range*.

- **List** es un tipo de secuencia que permite almacenar la secuencia ordenada de elementos de cualquier tipo. Las listas se definen entre corchetes `[]` y pueden ser modificadas después de su creación.

```
frutas = ["manzana", "banana", "pera"]
print(frutas)
```

- **Tuple** es similar a **list**, salvo que una tupla no puede ser modificada una vez creada. Las tuplas se definen entre paréntesis `()`.

```
dias_semana = ("lunes", "martes", "miércoles")
print(dias_semana)
```

- **Range** es un tipo de dato que permite crear una secuencia de números enteros y se suele utilizar con frecuencia en estructuras de control de flujo como los bucles *for*. El siguiente ejemplo generaría e imprimiría la secuencia de números enteros del 1 al 10.

```
numeros = range(10)
print(list(numeros))
```

En la última línea de código, el rango "numeros" es convertido a una lista para imprimir los elementos generados. También podrían ser almacenados para su manipulación, si así se deseara. A continuación, se muestra un sencillo ejemplo:

```
rango = range(1, 6)
lista = list(rango)
```




5 | Mapping Type (dict): en Python, el mapping se refiere a un tipo de objeto que almacena una colección de elementos en pares clave-valor, donde cada clave está asociada con un valor. En otros lenguajes de programación, este tipo de objeto a veces se conoce como un “diccionario” o “tabla hash”. Las claves son únicas (no podemos usar claves iguales para pares distintos) e inmutables (no se pueden modificar), y cada clave está asociada a un valor. Precisamente, estas claves se utilizan para acceder a los valores a los que están asociadas. Por ejemplo, podríamos crear un diccionario en el que las claves sean nombres de países y los valores asociados las capitales de dichos países. De esta forma, podríamos recuperar siempre de forma sencilla el nombre de la capital a partir del nombre del país:

```
capitales = {'España': 'Madrid', 'Francia': 'París', 'Alemania': 'Berlín',  
'Italia': 'Roma'}
```

Cabe recalcar que como valor podemos emplear diferentes clases de elementos, como listas, números enteros, o cualquier otro tipo de objeto mutable.

Una forma sencilla de mostrar por pantalla tanto las claves como los valores asociados es la siguiente:

```
print(list(capitales.keys()))  
print(list(capitales.values()))
```

6 | Set Types (set): en Python, set es un tipo de dato que permite almacenar una colección de elementos únicos y sin orden. Es similar a una lista o un diccionario, pero sin claves y con elementos únicos. Los sets se crean usando llaves {} o la función `set()`. Por ejemplo, podríamos definir de esta forma el conjunto de colores cuyos valores no procede que se modifiquen, en la mayor parte de ocasiones:

```
colores = {'rojo', 'verde', 'azul'}  
colores = set(['rojo', 'verde', 'azul'])
```

Este tipo de dato soporta operaciones matemáticas-lógicas como la unión, la intersección o la diferencia, entre otras.

Saber más

Para saber más sobre los tipos de datos en Python puedes consultar la documentación oficial en castellano en docs.python.org/es



Creación de
contenidos digitales

Nivel B2 3.4 Programación

Aspectos generales de variables y constantes en Python





Aspectos generales de variables y constantes en Python

Las variables, por convenio, se suelen nombrar con letras minúsculas para facilitar su identificación. Además, sus nombres o identificadores deben empezar por una letra o por un guion bajo (`_`) y pueden seguir con más letras, números o guiones bajos. Los identificadores no pueden incluir espacios en blanco, ni se pueden utilizar nombres de funciones ya existentes en Python. Si un identificador está formado por varias palabras, suele separarse con guiones por convenio.

Una variable se considera definida una vez que se le ha asignado un valor y, por tanto, está inicializada. Un ejemplo de variables bien definidas sería el siguiente:

```
precio_hora = 5
dias = 10
kilometros = 60
```

Las constantes no existen como tal en Python. Por ello, los programadores simplemente declaran una variable, le asignan un valor inicial y esperan que nadie modifique dicho valor. Existe una convención entre programadores para el nombrado de las constantes, para así identificarlas fácilmente y recordar que no deben ser modificadas. Al contrario que las variables, y atendiendo al convenio que se suele adoptar, su nombre debe escribirse completamente en mayúsculas.

Un ejemplo de constantes bien definidas sería el siguiente:

```
DIAS_AÑO = 365
MINUTOS_HORA = 60
SEGUNDO_HORA = 60
```



Asignación múltiple

Para definir una variable hay que inicializarla con el operador "=", como se ha visto anteriormente. Además, existen formas de inicializar variables conjuntamente. Por ejemplo, un valor puede ser asignado a varias variables a la vez:

```
a = b = c = 1 # Inicializa a, b y c con el valor 1
```

En caso de que se quieran inicializar varias variables a la vez, pero con distintos valores, la sintaxis sería como sigue:

```
a, b, c = 10, 50, 75 # Inicializa a con 10, b con 50 y c con 75
```





Creación de
contenidos digitales

Nivel B2 3.4 Programación

Funciones y paso de parámetros en Python





Funciones y paso de parámetros en Python

Python es un lenguaje de programación versátil que hace hincapié en la legibilidad del código y la sintaxis simple. Un elemento fundamental en cualquier lenguaje de programación, incluyendo Python, es la definición de **funciones**. A continuación, se discutirán las funciones de Python y se proporcionarán detalles sobre cómo se crean y se utilizan. Además, se examinará la diferencia entre argumentos y parámetros y se introducirá el concepto de parámetros por defecto.

Creación y uso de funciones

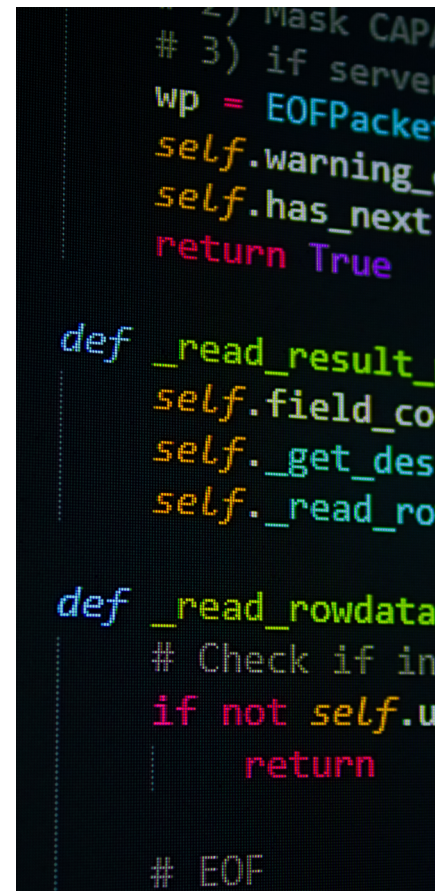
En Python, una función es un bloque de código reutilizable que realiza una acción específica. Las funciones proporcionan una forma de organizar el código en bloques lógicos, lo que permite una mayor claridad y facilidad de mantenimiento. Las funciones se crean utilizando la palabra clave **def**, seguida del nombre de la función y los paréntesis. Por ejemplo:

```
def mi_funcion():  
    print("¡Hola, mundo!")
```

Una vez definida, la función se puede llamar o invocar, para ejecutarla, utilizando su nombre seguido de paréntesis:

```
mi_funcion()
```

Ejecutar esta función mostraría por pantalla el texto «¡Hola, mundo!». Como se puede observar, una función puede estar formada por muchas otras expresiones que serán ejecutadas en orden cuando se invoque a dicha función. En este caso, la función que se ha definido ejecutaría una única expresión formada por la invocación a otra función llamada **print** para mostrar texto en pantalla.





Argumentos y parámetros

El término argumento y parámetro a menudo se utilizan indistintamente en programación, pero hay una diferencia clave. Un parámetro es una variable en la definición de la función, mientras que un argumento es el valor que se pasa a la función cuando se llama.

Por ejemplo, en la siguiente función, **nombre** es un parámetro:

```
def saludar(nombre):  
    print(f"¡Hola, {nombre}!")
```

Y cuando se invoca a la función, el valor que se le pasa, **Pedro**, es un argumento:

```
saludar("Pedro")
```

Pero, ¿qué sucedería si se invocara a la función del ejemplo sin ningún argumento? Por ejemplo, tal que así:

```
saludar()
```

En este caso, se ha definido una función que espera un argumento en su parámetro **nombre**, por lo que, si se invocara a la función sin pasarle dicho argumento, el intérprete de Python respondería con la siguiente salida de error por pantalla:

```
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
TypeError: saludar() missing 1 required positional argument: 'nombre'
```

Como indica el error, a la invocación de la función **saludar()** le falta un argumento, tal y como ésta espera que sea invocada.

NOTA

Python permite insertar variables directamente sobre en las cadenas de texto, tal y como se muestra en la invocación a la función **print()** del ejemplo anterior. Para ello, solo hay que prefijar la cadena con el carácter **f** y envolver entre llaves **{ }** las variables o expresiones que se deseen incluir en la cadena de texto.



Parámetros por defecto

Para terminar, Python permite definir parámetros con valores por defecto en las funciones. Esto significa que, si al invocar la función no se proporciona un valor para ese parámetro, se utilizará el valor por defecto que se haya indicado al definir la función. Por ejemplo:

```
def saludar(nombre="Mundo"):
    print(f"¡Hola, {nombre}!")
```

En este caso, si se invoca a la función sin ningún argumento mediante `saludar()`, se mostrará el texto «¡Hola, Mundo!» por pantalla ya que el parámetro `nombre` tiene como valor por defecto asignado la cadena de texto «Mundo». Pero si se le pasa un argumento, como en `saludar("Pedro")`, se mostrará «¡Hola, Pedro!» en su lugar. Esta técnica es muy útil para darle a las funciones un comportamiento por defecto y evitar así errores como el mostrado en la sección anterior. Por ejemplo, se indican a continuación algunos casos de uso:

- **Dotar a las funciones de un comportamiento flexible:** los parámetros por defecto permiten que las funciones se comporten de manera diferente dependiendo de cuánta información se proporcione. Por ejemplo, una función `potencia(base, exponente=2)` podría calcular el cuadrado de un número si solo se proporciona un argumento, pero calcularía la potencia especificada si se proporcionan dos argumentos.
- **Simplificar las invocaciones de la función:** en funciones que requieren muchos parámetros, proporcionar valores por defecto puede simplificar la invocación a la función, ya que solo se necesitarán argumentos para unos pocos parámetros críticos u obligatorios.



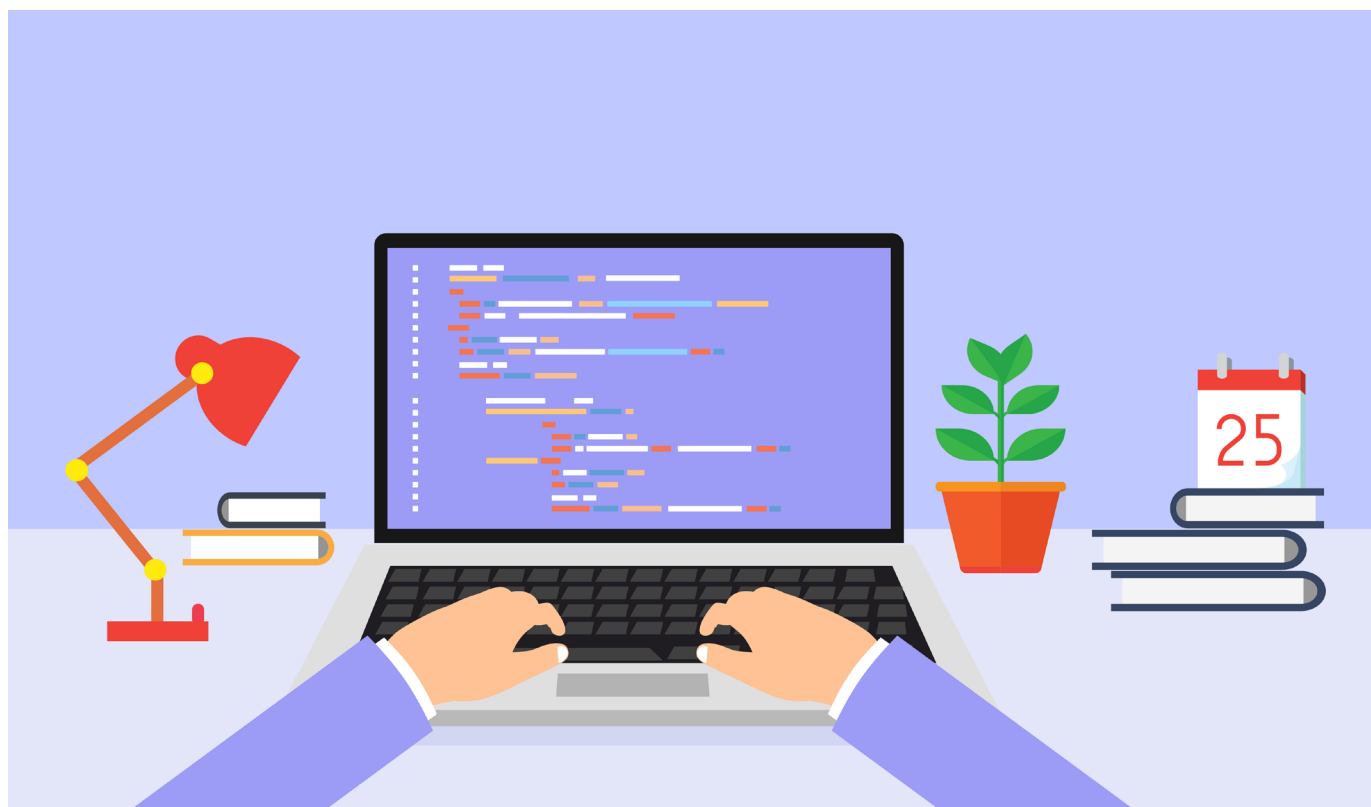
Conclusión

Los beneficios de utilizar funciones en Python son múltiples. En primer lugar, permiten la reutilización del código, lo que mejora la eficiencia y la coherencia; en lugar de tener que repetir el mismo código en diferentes partes del programa, se puede definir una función una vez y luego llamarla siempre que se necesite realizar esa operación específica.

Además, las funciones en Python contribuyen a la organización y la legibilidad del código; al agrupar segmentos de código relacionados en funciones, hacemos que los programas sean más fáciles de entender y de mantener. Este beneficio se amplifica aún más por las capacidades de Python para definir parámetros por defecto y proporcionar argumentos a las funciones por su nombre, lo que mejora la flexibilidad y la claridad de las interacciones con las funciones.

Saber más

Para saber más sobre las funciones en Python, puedes consultar la documentación oficial en castellano: e.digitall.org.es/definir-funciones





Creación de
contenidos digitales

Nivel B2 3.4 Programación

Diccionarios: definición y usos principales





Diccionarios: definición y usos principales

Los diccionarios en Python representan otra estructura de datos muy conveniente para ciertos tipos de problemas. Al contrario de lo que ocurre con una lista, accesibles a través de la posición que ocupa un determinado elemento dentro de la misma, un diccionario es accesible mediante una clave. En este punto, es importante remarcar que las claves de un diccionario en Python deben ser de un tipo inmutable (int, float, decimal, bool, string, tuple, y range). Típicamente, el tipo de las claves de un diccionario será int o string.

A la hora de considerar el uso de diccionarios en Python, estos se pueden entender como conjuntos de pares clave-valor. Así, las claves de un diccionario deben ser únicas. Por ejemplo, si se utiliza un diccionario para manejar información de un conjunto de personas de forma que la clave sea el DNI, entonces no es posible almacenar dos personas en el diccionario con el mismo DNI.

Como el lector puede imaginar en este punto, dos son las operaciones que más frecuentemente se utilizan a la hora de manejar un diccionario: i) añadir un nuevo valor con la correspondiente clave y ii) extraer un valor del diccionario empleando para ello la clave que permite indexarlo. Evidentemente, también es posible eliminar elementos de un diccionario.

El siguiente fragmento de código muestra un ejemplo de uso básico de un diccionario desde el intérprete interactivo de Python:

```
>>> d = {}
>>> d['David'] = 39
>>> d['Alberto'] = 34
>>> d
{'David': 39, 'Alberto': 34}
>>> d['David']
39
>>> del d['David']
>>> d
{'Alberto': 34}
```





Es posible iterar sobre los elementos de un diccionario, siendo incluso posible recuperar, de manera simultánea, la clave y el valor de cada elemento del diccionario. El siguiente ejemplo muestra esta característica, a la vez que expone otra forma diferente, y más sintetizada, de crear un diccionario en Python:

```
>>> d= {'David': 39, 'Alberto': 34, 'Javier': 33, 'Cristina': 30, 'Elena': 23}
>>> for k, v in d.items():
...     print(k, v)
...
David 39
Alberto 34
Javier 33
Cristina 30
Elena 23
```

Como se ha introducido previamente, los diccionarios son estructuras de datos ideales para almacenar datos donde existe una relación clara entre una referencia o propiedad única de cada elemento de la estructura y su valor correspondiente.

⚠ ATENCIÓN

A la hora de elegir una estructura de datos en Python para guardar elementos que pertenecen a un conjunto, es posible dudar entre utilizar una lista o un diccionario. Con carácter general, es importante recordar que el diccionario es más rápido que la lista en Python. Además, tanto la búsqueda de elementos como su recorrido es más eficiente en un diccionario. Sin embargo, la lista mantiene una relación de orden entre sus elementos, mientras que esto no ocurre en un diccionario.

Ejemplo de uso del tipo diccionario en Python

En esta sección se muestra un sencillo ejemplo de uso de diccionarios en Python. En esencia, se hace uso de un diccionario para almacenar y manipular información relativa a capitales del mundo.





```
>>> capitales = {'España': 'Madrid', 'Portugal': 'Lisboa', 'Francia': 'París',
'Italia': 'Roma'}
>>> print(len(capitales))
4
>>> if 'España' in capitales:
...     print(capitales ['España'])
...
Madrid
>>> print(capitales ['Reino Unido'])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'Reino Unido'
>>> 'Grecia' in capitales
False
>>> capitales.pop('Italia')
'Roma'
>>> print(capitales)
{'España': 'Madrid', 'Portugal': 'Lisboa', 'Francia': 'París'}
>>> capitales ['Portugal'] = 'Oporto'
>>> print(capitales)
{'España': 'Madrid', 'Portugal': 'Oporto', 'Francia': 'París'}
```

Cuando se trata de acceder a un diccionario mediante una clave que no está contenida en el mismo, como ocurre en el ejemplo anterior cuando *capitales* se indexa mediante *Grecia*, el intérprete arroja una excepción indicando que existe un error con dicha clave. Es posible comprobar si una clave está contenida en un diccionario mediante el operador **in**.

Por otro lado, en el ejemplo anterior también se ilustra el uso de la operación **pop()**, empleada para eliminar una entrada del diccionario a partir de una clave.

También es posible actualizar una entrada del diccionario. Para ello, simplemente ha de indexarse a través de la clave, ya existente en el mismo, indicando el nuevo valor. Este caso concreto se ilustra, en el ejemplo, con la actualización de la capital de *Portugal*.

Finalmente, es posible manejar operaciones relativas a la ordenación de elementos en un diccionario. Por ejemplo, se puede aplicar la función **sorted()** para obtener, en orden ascendente, una lista que contenga las claves de un diccionario:

```
>>> sorted(capitales)
['España', 'Francia', 'Portugal']
```



Creación de
contenidos digitales

Nivel B2 3.4 Programación

Manejo básico de archivos en Python





Manejo básico de archivos en Python

Python incluye varias características integradas para el manejo de archivos. Estas funciones permiten a los programadores crear, leer, escribir, y manipular archivos de texto de una manera sencilla y eficiente. A continuación, se explorará el proceso de crear y gestionar archivos en Python, con énfasis en operaciones básicas como la apertura, lectura, escritura y cierre de archivos.

Crear un archivo de texto con varias líneas de contenido

Python utiliza la función `open()` para abrir un archivo.

Si el archivo especificado no existe, Python lo creará automáticamente. Para escribir en un archivo, se utiliza el método `write()`. Por ejemplo, así se crearía un archivo de texto con varias líneas:

```
# Crear un archivo llamado 'ejemplo.txt'
archivo = open('ejemplo.txt', 'w')

# Escribir varias líneas en el archivo
archivo.write('Esta es la primera línea.\n')
archivo.write('Y esta es la segunda línea.\n')
archivo.write('Finalmente, esta es la tercera línea.\n')

# Cerrar el archivo
archivo.close()
```

Si se abriera ahora el archivo `ejemplo.txt` con un editor de texto, se mostraría el siguiente contenido:

```
Esta es la primera línea.
Y esta es la segunda línea.
Finalmente, esta es la tercera línea.
```

El argumento `'w'` en la función `open()` indica que el archivo se abre para escritura (*write*). Si se volviera a escribir en ese archivo empleando este enfoque, el archivo se sobrescribiría con el nuevo contenido, eliminando cualquier contenido existente. Para evitar este comportamiento, se puede configurar la apertura del archivo para agregar contenido a un archivo existente usando el argumento `'a'` (*append*). Por ejemplo:



```
# Abrir el archivo llamado 'ejemplo.txt'
archivo = open('ejemplo.txt', 'a')

# Añadir dos líneas al archivo
archivo.write('Esto es una línea nueva.\n')
archivo.write('Y esta es otra línea añadida al archivo.\n')

# Cerrar el archivo
archivo.close()
```

Ahora, al abrir el archivo con un editor de texto, se mostraría el siguiente contenido:

```
Esta es la primera línea.
Y esta es la segunda línea.
Finalmente, esta es la tercera línea.
Esto es una línea nueva.
Y esta es otra línea añadida al archivo.
```

Operaciones básicas: Apertura, Lectura, Escritura y Cierre

Apertura

Como se mencionó anteriormente, se utiliza la función `open()` para abrir un archivo. Esta función tiene dos argumentos, el nombre del archivo y el modo:

```
# Abrir un archivo para lectura
archivo = open('ejemplo.txt', 'r')
```

Cabe destacar, que el segundo argumento de la función `open()` es opcional y que, si no se especifica, tomará por defecto el valor `'r'`.

Lectura

Después de abrir un archivo, se pueden usar varios métodos para leer su contenido. El método `read()` lee todo el contenido del archivo y lo guarda en una variable que indiquemos:

```
# Leer todo el contenido del archivo
contenido = archivo.read()
print(contenido)
```




Escritura

Para escribir en un archivo, primero se debe abrir en modo de escritura ('w') o de agregación ('a'). Luego, se utiliza el método `write()`:

```
# Abrir un archivo para escritura
archivo = open('ejemplo.txt', 'w')

# Escribir en el archivo
archivo.write('Hola, mundo!')
```

Cierre

Después de trabajar con un archivo, es importante cerrarlo para liberar recursos del sistema y asegurarse de que el archivo se guarda correctamente en el sistema. Para ello, se utiliza el método `close()`:

```
# Cerrar el archivo
archivo.close()
```

Para no olvidar cerrar el archivo cuando se finalice el trabajo con él, se recomienda utilizar la sentencia `with`. Esta sentencia permite abrir un archivo y asegurarse de que se cierre correctamente una vez se ha terminado de trabajar con él, incluso si ocurren errores durante el proceso. Por ejemplo:

```
# Ejemplo de apertura de un archivo con with
with open("archivo.txt", "r") as archivo:
    contenido = archivo.read()
    print(contenido)
```

Cuando finalice la ejecución del bloque indentado por la sentencia `with`, el archivo se cerrará automáticamente sin necesidad de tener que invocar al método `close()`.

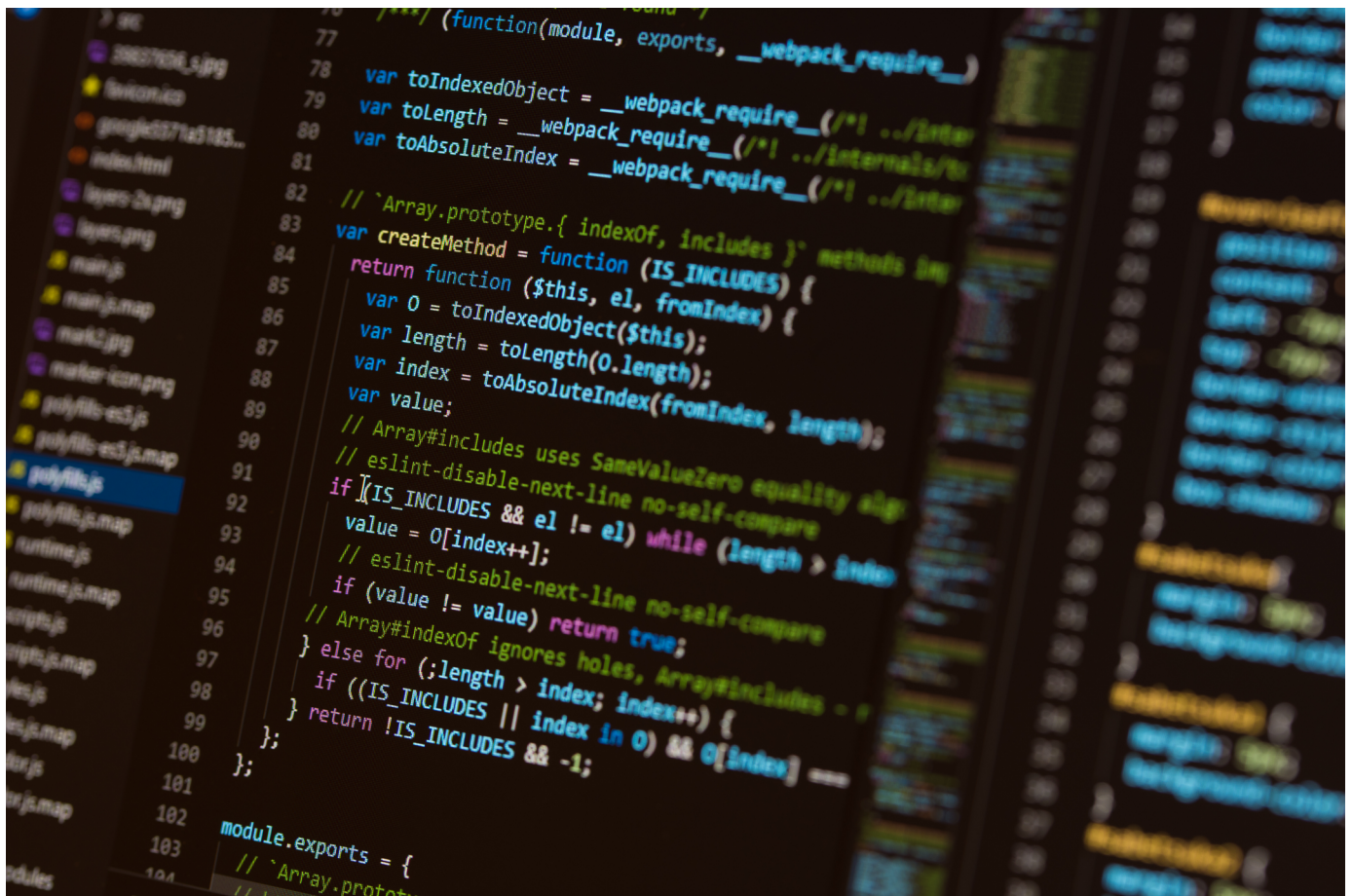


Conclusión

El manejo de archivos es una tarea esencial y común en la programación. Python ofrece funciones y métodos integrados que facilitan la lectura, escritura, apertura y cierre de archivos. Aunque los aspectos discutidos anteriormente solo cubren las operaciones básicas, Python ofrece muchas más funciones avanzadas para trabajar con archivos. Asegúrate de explorar más a fondo la documentación oficial de Python para obtener una comprensión más completa de estas funciones.

Saber más

Para saber más sobre el manejo de archivos en Python, puedes consultar la documentación oficial en castellano: e.digitall.org.es/archivos-python





Creación de
contenidos digitales

Nivel B2 3.4 Programación

¿Qué es el
software 2.0?





¿Qué es el software 2.0?

La Ingeniería Software es una disciplina que consta de una serie de métodos, técnicas y herramientas que permiten el desarrollo de software. Hasta el momento, el único protagonista de dicho desarrollo de software ha sido el programador. La calidad y eficiencia de los programas desarrollados siempre han dependido del arte y buen hacer de los programadores. Esta concepción del desarrollo de software, centrada en expertos programadores, es lo que se conoce como **software 1.0**. Sin embargo, dada la revolución de los últimos años que ha sufrido la disciplina de la *Inteligencia Artificial*, especialmente áreas concretas como el *Deep Learning* o *Aprendizaje Profundo*, este enfoque de desarrollo de software está cambiando, dando paso a nuevos actores que colaborarán en el desarrollo de programas informáticos. Estos actores son máquinas capaces de desarrollar software de forma autónoma, o al menos colaborar en algunas de las tareas de desarrollo, facilitando así la labor del programador clásico.

El término **software 2.0** fue acuñado por Adrej Karpathy, responsable del área de Inteligencia Artificial de la empresa Tesla. La forma de desarrollo de software tradicional es lenta y laboriosa; por ello, son muchos los errores que cometen los desarrolladores. Sin embargo, las técnicas de Aprendizaje Automático dentro de la Inteligencia Artificial permiten automatizar parte de este desarrollo. Para ello, los ordenadores aprenden a programar mediante ejemplos, es decir, a partir de programas ya existentes bien implementados son capaces de entender su estructura y replicarla. Este aprendizaje permite la automatización de tareas de programación, acelerando el proceso de desarrollo y reduciendo considerablemente los errores habitualmente cometidos.

Resumiendo, la fundamentación del software 1.0 se basa en el análisis del problema, el diseño de los algoritmos y componentes necesarios y, la implementación y pruebas de los mismos por parte del desarrollador. Por otro lado, el software 2.0 se basa en la recopilación de ejemplos de tareas de desarrollo que sirven para que un ordenador, mediante técnicas de Aprendizaje Automático, aprenda a replicarlas.



Ejemplo

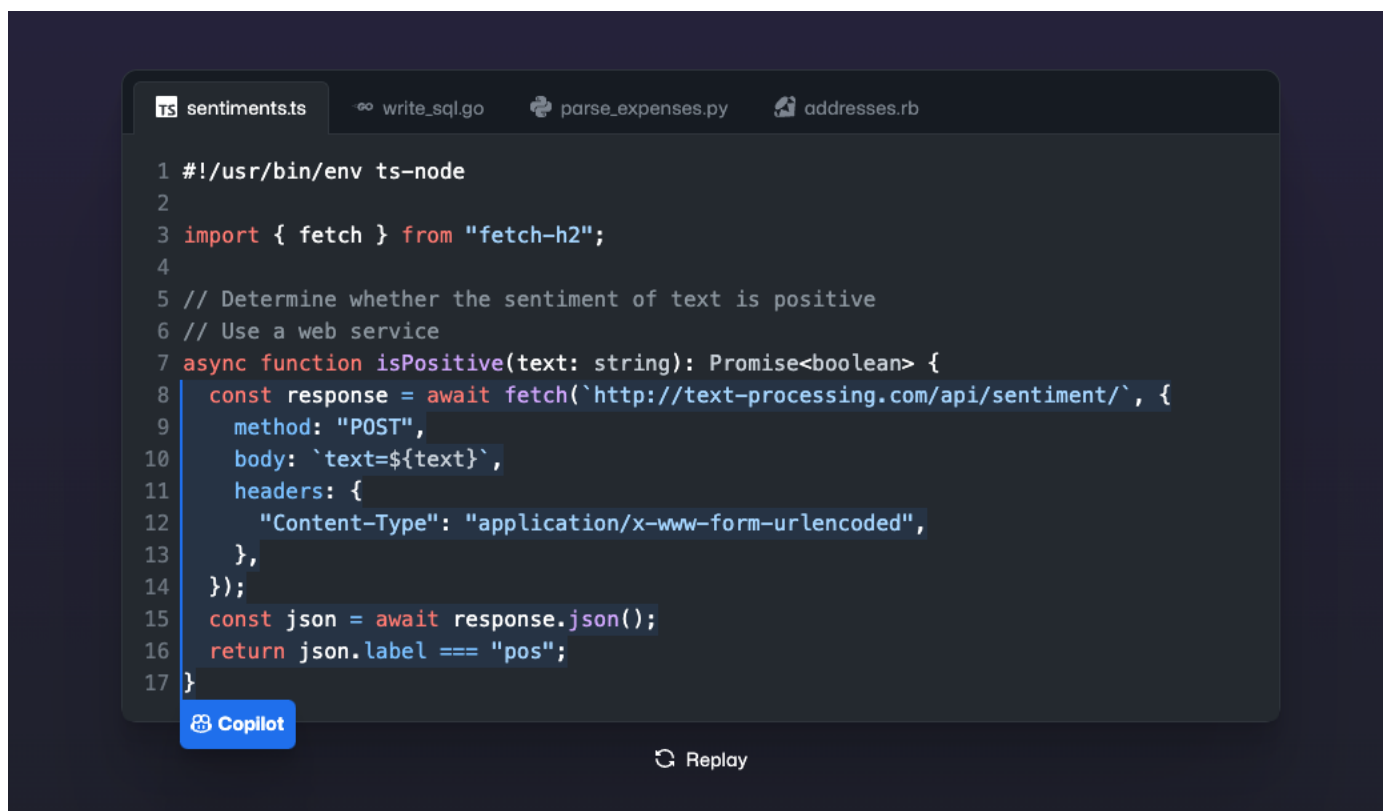
Son muchas las aplicaciones que se irán encontrando en los próximos años asociadas al software 2.0 en áreas como el reconocimiento visual y de voz, traducción automática, juegos, bases de datos, entre otros.

Un ejemplo es **Copilot** (github.com/features/copilot), una herramienta que permite al usuario describir en lenguaje natural una funcionalidad que necesita implementar, y esta traduce a código fuente la funcionalidad indicada en el lenguaje de programación que se le pida.


Saber más


Aprendizaje Profundo. A. Shrestha and A. Mahmood, "Review of Deep Learning Algorithms and Architectures," in *IEEE Access*, vol. 7, pp. 53040–53065, 2019, doi: 10.1109/ACCESS.2019.2912200.

Definición Software 2.0: e.digitall.org.es/software-2



```
sentiments.ts write_sql.go parse_expenses.py addresses.rb
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch(`http://text-processing.com/api/sentiment/`, {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17 }
```

 Copilot

 Replay

Captura de pantalla de Copilot. Fuente: github.com/features/copilot.



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 Rodero
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>