



Lenguajes de marcas

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
  <head>
    <meta name="TITLE" content="Voyant"/>
    <meta name="KEYWORDS" content="Voyant"/>
    <meta name="DESCRIPTION" content="A web-based interface for reading, analyzing, and visualizing large texts."/>
    <link rel="stylesheet" type="text/css" href="https://voyant-tools.org/voyant.css"/>
    <script language="javascript" type="text/javascript" src="https://voyant-tools.org/voyant.js"/>
  </head>
  <body bgcolor="#ffffcc" width="100%" height="100%"/>
    <div style="display: flex; align-items: center; justify-content: center; height: 100%; width: 100%; margin: auto; border: 1px solid black; padding: 10px; background-color: #f0f0f0; border-radius: 10px; font-family: sans-serif; font-size: 14px; color: black;">
      <div style="text-align: center; width: 100%;">
        <img alt="Voyant logo" style="width: 100px; height: 100px;"/>
        <h1 style="margin: 0;">Voyant
        <p style="margin: 0;">A Text Analysis Platform
      </div>
    </div>
  </body>
</html>
```



Tema 2. Elementos básicos HTML

1. Introducción

En el cuerpo de un documento HTML encontramos el contenido real que será visible al usuario. Toda esta información deberá estar incluida entre las etiquetas **<body>** y **</body>**.

En HTML la mayoría de elementos del cuerpo se pueden clasificar en:

- Elementos de bloque: Son elementos que crean estructuras más grandes que pueden contener a los elementos de línea y a otros elementos de bloque.
- Elementos de línea: Éstos solo pueden contener datos y a otros elementos de línea.

EJERCICIO 1

- Vas a crear un directorio de carpetas que te servirá como base para cualquier proyecto, práctica o ejercicio que vayas a realizar a lo largo del curso:
 - Crea una carpeta principal llamada “ProyectoEjemplo”. El nombre de esta carpeta cambiará cada vez que la copies para un proyecto nuevo.
 - Dentro de “ProyectoEjemplo” crea las siguientes carpetas (Llámala como quieras):
 - una carpeta que será la que contenga los documentos html.
 - Una segunda carpeta para las imágenes.
 - Otra carpeta para los documentos de script.
 - Otra carpeta más, esta vez para los documentos css.
 - Una última carpeta para vídeos.

2. Manejo de texto

2.1. Elementos para establecer encabezados

Los elemento **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>** y **<h6>** sirven para establecer encabezados, es decir, el tamaño de los títulos. Como se ve hay seis niveles, ordenados estrictamente por tamaño, donde h1 es el más grande. Los encabezados son elementos de bloque.

Las etiquetas de apertura y cierre son obligatorias.

Cada vez que se introduzca un encabezado, el navegador creará un espacio de línea tanto arriba como abajo.

Las buenas prácticas dictan que se deben usar los títulos por orden y sin omitir ninguno. Si se quiere reducir el tamaño de la fuente se usarían estilos y no se pone otro tipo de encabezado.



Este tipo de elementos admiten los atributo globales.

```
<h1>Primer encabezado</h1> <h2> 2º encabezado </h2>  
<h3>Este es el tercero</h3><h4>El 4º tiene texto normal</h4>  
<h5>Quinto encabezado</h5><h6>El texto del 6º es muy pequeño</h6>
```

2.2. Párrafo

De la misma forma que se reparte la información mediante párrafos en un documento de texto tipo Word, las líneas de texto en HTML suelen agruparse en un elemento `<p>`.

Cada párrafo en html se compone de una única línea de texto, para cambiar esto hay que introducir explícitamente un retorno de carro con la etiqueta `
`. El navegador será el encargado de que la línea ocupe todo el ancho disponible, y si el ancho cambia, el texto se adapta al nuevo ancho ocupando más o menos líneas.

Todo texto de un documento HTML, que no sea encabezado, lista o celda debería estar dentro del elemento `<p>`.

La etiqueta de cierre es opcional pero por buenas prácticas es mejor ponerla.

En los textos que se introduzcan, se deberían tener en cuenta unas apreciaciones sobre el espacio en blanco. Si se quiere insertar un espacio más del normal entre dos palabras, el navegador ignorará todos los que se incluyan utilizando la barra espaciadora. Será necesario introducir el carácter ` ` por cada espacio adicional que se vaya a introducir.

EJERCICIO 2

- Confecciona una página que muestre nombres de ciudades separados por un guión.
- Añade, a la página anterior, un párrafo que muestre, uno por línea, los nombres de ciudades del ejercicio anterior.
- Añade dos nuevos párrafos con información de una ciudad cada uno. En el primero agrega varios saltos de línea.
- Crea una nueva página que contenga un título de primer nivel `<h1>` y luego dos títulos de nivel `<h2>`. Definir un párrafo para cada título de segundo nivel.

2.3. Separadores de texto

`
`

Los saltos de línea en el texto hay que introducirlos con este elemento, ya que cualquier salto introducido con enter será ignorado por el navegador.

Este elemento no tiene etiqueta de cierre.

`<hr>`

Este separador nos introduce un línea visible horizontal. Tampoco tiene etiqueta de cierre.



2.4. Estilos de fuente

HTML tiene una serie de elementos con los que es posible formatear el texto. Se trata de elementos de línea que aún se pueden ver en páginas antiguas pero que deberían evitarse ya que están obsoletos. Por ejemplo, ****, *<i>*, **<big>**, *<small>*, **<s>**, ****, etc.

Para establecer un texto con formato previo, existe la etiqueta **<pre>**. Ésta permite que el navegador visualice el texto tal y como aparece en el contenido de esta etiqueta, respetando tabulaciones, espacios y saltos de línea. La única excepción se da en caso de que se trate de otros elementos HTML, los cuales se ejecutarán también. Para que los caracteres < y > no se interpreten como etiquetas han de incluirse tal que así : **<** y **>** .

EJERCICIO 3

- Crea, utilizando los elementos HTML vistos hasta el momento, una página con el siguiente aspecto:

The screenshot shows a web browser window with the title 'Coronavirus'. The address bar displays '127.0.0.1:5500/codigo/ejercicio2.html'. The page content is as follows:

Medidas contra el coronavirus

Manténgase al día de la información más reciente sobre el brote de COVID-19, a la que puede acceder en el sitio web de la OMS y a través de las autoridades de salud pública pertinentes a nivel nacional y local.

Medidas de higiene

Lavar las manos frecuentemente

Lávese las manos con frecuencia con un desinfectante de manos a base de alcohol o con agua y jabón.

Adoptar medidas de higiene respiratoria

Al toser o estornudar, cubrase la boca y la nariz con el codo flexionado o con un pañuelo; tire el pañuelo inmediatamente y lávese las manos con un desinfectante de manos a base de alcohol, o con agua y jabón.

Mantener el distanciamiento social

Mantenga al menos 1 metro y medio de distancia entre usted y las demás personas, particularmente aquellas que tosan, estornuden y tengan fiebre.

Pregunta a la OMS

¿Debo evitar dar la mano a las personas por el nuevo coronavirus?

Si. Los virus respiratorios pueden transmitirse al darse la mano y tocarse los ojos, la nariz y la boca. Es mejor saludar con un gesto de la mano, una inclinación de la cabeza o una reverencia.

¿Usar guantes de goma cuando se está en público es una forma eficaz de prevenir la infección por el nuevo coronavirus?

No. Lavarse las manos con frecuencia proporciona más protección frente al contagio de la COVID-19 que usar guantes de goma. El hecho de llevarlos puestos no impide el contagio, ya que si uno se toca la cara mientras los lleva, la contaminación pasa del guante a la cara y puede causar la infección.



3. Listas

En HTML hay dos tipos fundamentales de listas: de elementos y de definiciones. A su vez, las listas de elementos pueden estar ordenadas o no.

3.1. Listas de elementos

Listas de elementos no ordenados (utiliza símbolos para marcar cada elemento de la lista). Se construyen con el elemento `` y su contenido está formado exclusivamente por elementos ``. Éstos últimos serán los que contengan el texto.

Son elementos con cierre.

Lista de elementos ordenados (utiliza números para marcar cada elemento de la lista). Se construyen con el elemento `` y su contenido está formado exclusivamente por elementos ``. Éstos últimos serán los que contengan el texto.

Son elementos con cierre.

Elemento de lista. Se trata de un objeto de lista, que puede contener otros elementos de bloque como párrafos o directamente el texto.

El cierre es opcional.

3.2. Listas anidadas

La mayor utilidad de las lista surge cuando se combinan entre sí, es decir, cuando cada elemento de una lista es a su vez otra lista. En estos casos, las buenas practica marcan que los niveles de anidación deben ser de tipos distintos, para evitar la confusión del usuario.

3.3. Listas de definiciones

Se trata de listas que no tienen índices ni símbolos de ninguna clase, como las anteriores, si no que se componen de dos elementos: el término y la definición, a modo de diccionario. Todo el texto aparece con sangría.

<dl> Elemento principal con el que se construyen estas listas. Contendrá solo elementos de tipo `<dt>` o `<dd>`. Las etiquetas de apertura y cierre son obligatorias.

<dt> En este elemento incluimos el término que se quiere definir. Se trata de un elemento de línea.

<dd> Aquí se introduce la definición del término anterior. Normalmente es un texto mayor que el de la etiqueta `dt` y se trata de un elemento de bloque.

**EJERCICIO 4**

- Crea una página HTML con el siguiente contenido:
- Habrá un encabezado principal que ponga “Tareas pendientes”.
 - Dos encabezados secundarios que indiquen “Tareas para hoy” uno, y “Tareas para mañana” el otro.
 - Para ese primer apartado de tareas para hoy se incluirá un párrafo donde habrá 3 tareas que se tengan que hacer hoy. Cada tarea comenzará en una línea diferente.
 - En el segundo apartado añadir una lista de definiciones donde se incluyan 3 tareas y dos pequeñas descripciones para cada una.
- Una vez terminado lo anterior, agrega debajo un texto donde se muestre el fragmento de código Java que se incluye a continuación. Se busca que el navegador respete tabulaciones y saltos de línea tal cual se escriban.

```
public class HolaMundo {  
  
    public static void main(String[] args) {  
  
        System.out.println("Hola Mundo");  
  
    }  
  
}
```

- Por último incorpora una lista anidada para ver claramente las asignaturas que del módulo DAW de forma ordenada y contenidos de cada una(vale con poner dos) sin ordenar. Para esta última lista, busca con que atributo y como se puede cambiar la numeración o los símbolos que aparecen por defecto. Prueba tipos diferentes.



4. Enlaces

Los enlaces o hipervínculos son los elementos más característicos de html, ya que fueron el motivo que inspiró su desarrollo. Un hipervínculo es en realidad una zona del documento que, al realizar una acción sobre ella, redirecciona a otro punto del mismo documento o a otro documento distinto.

El hipervínculo en principio es invisible, lo que se ve son los elementos que se insertan en su contenido, como texto, imágenes, objetos, etc.

<a>

Con este elemento se establece el origen o destino del hipervínculo. Se trata de un elemento en línea, por lo que puede estar dentro de un párrafo, lista, etc., y a su vez puede contener otros elementos en línea excepto a si mismo.

Algunos de sus atributos (además de los globales) son los siguientes:

- href: indicar la URL o fragmento de URL del recurso enlazado.
 - URL
 - ```
Developer mozilla
```
  - Fragmento de URL: Especifica una ubicación interna. Será un nombre precedido por #

```
Pulsar para ver resultado
<p id="idParrafo">Resultado de la operación 5</p>
```
- target : establece el marco de destino, es decir, donde se abrirá el documento enlazado. Sus posibles valores son:
  - \_blank: Abre el link en una nueva ventana o pestaña

```
Google
```
  - \_self: Valor por defecto. Abre el link en el mismo marco.

```
google </Aa>
```
- download: con este atributo se establece que la url de enlace es un archivo para descargar del servidor. Si no se indica el nombre del archivo, se descargará con el que tenga en el servidor.

```
Texto
```
- rel: Especifica la relación del documento actual con el link. El valor será una lista de tipo enlace, separados por espacios en caso de haber más de uno. Los valores que puede tomar los podemos ver en [https://www.w3schools.com/TAGS/att\\_a\\_rel.asp](https://www.w3schools.com/TAGS/att_a_rel.asp)



## 4.1. Enlace a la misma página

Como hemos visto antes, se enlazará con otra parte de nuestra página a través del id del elemento a enlazar. Se utilizará # para indicar que el destino es un marcador de posición interno al documento.

```
<h1 id="primerEncabezado"> Consultar calificaciones </h1>
<! - -Introducimos el grueso de la página - ->
 Volver arriba
```

## 4.2. Enlace a otra página del mismo servidor

Si se pretende acceder a otra página que se encuentre alojada en el mismo servidor que el documento, se indica de la siguiente manera:

```
 Ir a la página 2
```

Si lo que se quiere es enlazar un punto concreto de otro documento se hará a través del identificador que tenga el elemento al que se quiera dirigir. El id se referenciará con el símbolo #:

```
 Ir la definición de XML
 Ir la definición de HTML
```

Y en el documento 2 tendremos:

```
<li id="idXML"> XML
<li id="idHTML"> HTML
```

## 4.3. Enlace en otro servidor

En este caso hay que especificar la URI completa de la página de destino.

```
 HTML
 XML
```

La estructura de la URI completa será:

Protocolo	:	Parte jerárquica	?	Solicitud	#	Fragmento
-----------	---	------------------	---	-----------	---	-----------

Donde:

- Protocolo: Hay muchos, como file, ftp, http, https, mailto, etc.
- Parte jerárquica: dominio, subdominio, carpeta, subcarpeta, recurso, puerto y las barras(/) para el nivel de jerarquía. (Si el puerto es 80 se omite)
- Solicitud: variable \$\_GET que se pasa al recurso (Suelen utilizarse para páginas dinámicas con PHP y similares)
- Fragmento: Posición del recurso dentro del documento.



Características de un URI:

- La solicitud y el fragmento no son obligatorios.
- El protocolo y el (sub.) dominio de la parte jerárquica son sensibles a mayúsculas y minúsculas y por lo tanto el navegador las convierte en minúsculas.
- La ruta de la parte jerárquica, la solicitud y el fragmento, en general, si son sensibles a mayúsculas y minúsculas. Depende del servidor
- Una **URL** consta de las tres primeras partes de una URI. No tiene fragmento.

Se puede ver una explicación mas en detalle en el siguiente enlace:

[https://developer.mozilla.org/es/docs/Web/HTTP/Basics\\_of\\_HTTP/Identificaci%C3%B3n\\_recursos\\_en\\_la\\_Web](https://developer.mozilla.org/es/docs/Web/HTTP/Basics_of_HTTP/Identificaci%C3%B3n_recursos_en_la_Web)

Por ejemplo, distintas URIs y URLs, son las siguientes:

<ftp://ftp.rediris.es>

<http://www.w3c.org>

[http://www.uterra.com/codigo\\_php/codigo\\_php.php?ref=mi\\_primera\\_pagina\\_en\\_php](http://www.uterra.com/codigo_php/codigo_php.php?ref=mi_primera_pagina_en_php)

#### EJERCICIO 5

- De las listas creadas en el ejercicio 4, pasa la lista anidada a un nuevo documento html que se llame listaAsignaturas.
- En el primer documento, al final del cuerpo, introduce dos enlaces internos. Uno a las “tareas de hoy” y otro a las “tareas de mañana”.
- Añade ahora un tercer enlace que lleve al documento listaAsignaturas.
- En el documento listaAsignaturas, enlaza alguno de los contenidos con una web donde haya un manual de ayuda para ese contenido concreto.



## 5. Agrupación de contenido

A diferencia de los elementos anteriores, que nos servían para agregar contenido o formato al documento, los dos siguientes que vamos a ver nos van a permitir darle una estructura a ese contenido.

### 5.1. <div>

Sirve para agrupar elementos dentro de un bloque y así poder asignarle nombre, clase y estilo. Se trata de un elemento de bloque. Al ser elemento de bloque, el navegador introducirá un salto de línea automáticamente después del cierre del elemento.

Cada bloque creado con div define un área dentro del documento, cuyas medidas dependen principalmente del contenido que hayamos insertado en el bloque.

Distintos elementos div podrían ocultarse y solaparse entre si, por lo que se puede ver denominado como capa. Mediante hojas de estilo podremos establecer posición en pantalla, alto, ancho, orden de solapamiento, etc...:

- left, top, position: para la posición en pantalla.
- width, height: Para ancho y alto.
- background-color, etc: Para dar color.

<div> acepta los atributos globales y el atributo align que en html5 queda obsoleto.

En el siguiente ejemplo vemos como con el elemento <div> separamos el contenido del documento en tres partes bien diferenciadas:

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8">
 <title>Etiquetas div</title>
 </head>
 <body>
 <div style="border: 1px solid red;">Capa 1</div>
 <div style="border: 1px solid blue;">Capa 2</div>
 </body>
</html>
```

### Propiedades CSS width y height

Los elementos HTML de bloque, por defecto, ocupan todo el ancho que le ofrece el navegador o dispositivo desde el que se muestre la página. De la misma forma, la altura del elemento se adaptará al contenido del mismo.

Para variar este comportamiento, se puede establecer tanto el ancho como el alto del elemento mediante las propiedades CSS width y height. Las dos formas mas utilizadas para darles valor son:



- Mediante valores absolutos de longitud: En pixels(relativo a la pantalla), en em(relativa respecto del tamaño de letra empleado), en ex(relativa respecto de la altura de la letra x minúscula del tipo de letra empleado), centímetros, milímetros, pulgadas, etc.
- Mediante porcentaje respecto del elemento que contiene al elemento que se le quiera establecer.

### ***Propiedades CSS float y clear***

Dos de las propiedades CSS más utilizadas para maquetar una página html son float y clear.

Con la propiedad float, se puede indicar que un elemento flote saliéndose de su posición por defecto. Al posicionar de esta forma un elemento, el resto de elementos de la página se repositionarán para adaptarse a la posición marcada para dicho elemento. Los valores más utilizados de los que puede tomar son:

- left: El elemento se posicionará a la derecha dentro de su contenedor.
- right: El elemento se posicionará a la izquierda dentro de su contenedor.

Esta propiedad float se suele combinar con las propiedades para establecer alto y ancho del elemento.

Por su parte, la propiedad clear, permite controlar el comportamiento de elementos que sea adyacentes a elementos flotantes.

La propiedad clear indica el lado o lados de una caja que no debe ser adyacente a un elemento posicionado de forma flotante con la propiedad float. Los valores mas utilizados son:

- left: hace que la caja sobre la que se aplica baje hasta que su borde superior esté por debajo del borde inferior de cualquier elemento flotado a la izquierda.
- right: hace que la caja sobre la que se aplica baje hasta que su borde superior esté por debajo del borde inferior de cualquier elemento flotado a la derecha.
- both: hace que la caja sobre la que se aplica baje hasta que su borde superior esté por debajo del borde inferior de cualquier elemento flotado a la izquierda o a la derecha.

Para evitar que un contenedor “ignore” a su contenido y parezca que se ha cerrado antes de incorporar los elementos que se indican, se le asignará la propiedad css *overflow: hidden;*. Esta propiedad se explicará mas adelante en la unidad dedicada a CSS.

### **EJERCICIO 6**

- Crea una página web que tenga:
- Un encabezado de tipo h1.
  - Un contenedor de tipo div que contendrá otro tres contenedores div. Cada uno de esos contenedores div incluirá, a su vez, un párrafo con un texto cualquiera.



- Los ejercicios que se detallan a continuación son todos para ordenar, de diferentes formas, los elementos especificados en el primer punto utilizando las propiedades CSS float, clear, width y height:
- Primer ejercicio:
  - El encabezado ocupará solo el 25% del ancho de la página.
  - El div principal ocupará el 55% de la página, estará centrado y debajo del encabezado.
  - Los divs secundarios se colocarán uno debajo de otro.
- Segundo ejercicio:
  - El encabezado ocupará solo el 25% del ancho de la página.
  - El div principal ocupará el 65% de la página, estará centrado y a la derecha del encabezado.
  - Los divs secundarios se colocarán uno al lado del otro, los tres paralelos.
- Tercer ejercicio:
  - El encabezado ocupará todo el ancho de la página.
  - El div principal ocupará el 80%.
  - De los tres divs secundarios, dos estarán en paralelo, uno a la derecha y otro a la izquierda, ocupando cada uno el 30% del espacio disponible y el tercero se situará debajo de los otros dos ocupando todo el ancho disponible.
- Cuarto ejercicio:
  - El encabezado ocupará todo el ancho de la página.
  - El div principal ocupará el 80%.
  - De los tres divs secundarios, dos estarán en paralelo, uno a la derecha y otro a la izquierda, ocupando cada uno el 30% del espacio disponible.
  - El tercero se situará debajo de los otros dos, centrado respecto de su contenedor y ocupando solo el 30% del ancho del mismo.

## 5.2. <span>

Sirve para identificar un elemento en línea y poder asignarle nombre, clase y estilo. Habitualmente se utiliza para crear subdivisiones dentro de una capa (div), y así poder aplicar estilos diferentes a cada subdivisión sin tener que crear nuevas capas.

Al ser elemento de línea, se pueden aplicar diferentes estilos a partes concretas del texto sin necesidad de cambiar de línea.

En el siguiente ejemplo vemos una combinación de las etiquetas <div> y <span>:

```
<!DOCTYPE HTML>
<html>
 <head>
 <title>Ejemplo div y span</title>
 </head>
 <body>
 <div style="border: 1px dotted blue;">
```



```
<p>
 Esto es un párrafo dentro de un div,
 y esto un span dentro de un párrafo.

</p>
</div>
</body>
</html>
```

\*\*NOTA: En estos dos últimos ejemplos hemos visto los estilos aplicados directamente en el elemento. Estos estilos, como hemos visto en apartados anteriores, se deberían meter en hojas de estilo aparte.

### EJERCICIO 7

- Crea un proyecto que contenga el documento fabricaEmbutidos.html.
- Introduce los elementos que consideres necesarios en la cabecera.
- Mediante las etiquetas div y span, en el cuerpo del documento, realiza lo siguiente, teniendo en cuenta las indicaciones que se incluyen a continuación de la imagen:



- Dada esa imagen anterior:
  - Embutido Guijuelo será un encabezado de nivel 1.
  - Los dos primeros bloques(azul y verde) estarán incluidos en elementos div, y, dentro de ellos, el texto que difiera del resto en cuanto a color y fuente será un elemento de tipo span.
  - La última línea que no se encuentra en ningún bloque, será un párrafo. El texto diferente estará incluido en un elemento tipo span.
  - Introducir los estilos en un documento aparte.
- Una vez tu página sea lo más parecida posible a la imagen anterior, haz las modificaciones necesarias para que quede de la siguiente manera:



Embutidos Guijuelo

Embutidos ibéricos

Embutidos Guijuelo presenta su nuevo jamón ibérico de curación larga

Características:

- Cerdos 100% ibéricos
- 36 meses en secadero **natural**.
- Salazón en refrigeración controlada

Solo disponible bajo pedido. Tiempo de espera 3 meses. *Se pagará por adelantado la mitad.*

No se admitirán cancelaciones pasados 15 días tras la realización del pedido



## 6. Imágenes

### 6.1. <img>

Esta etiqueta sirve para insertar una imagen en un documento html. Se trata de un elemento en línea y no tiene etiqueta de cierre.

Atributos desaprobados u obsoletos de esta etiqueta son: align, border, hspace, name, vspace.

Los siguientes atributos son algunos de los que se pueden incluir para esta etiqueta:

- src: Se trata de un atributo obligatorio. Enlaza a la imagen que se quiere mostrar en la página, ya sea mediante una URL relativa o absoluta, aunque lo más lógico es que se trate de una url relativa.
- alt: se trata de un texto con una breve descripción de la imagen, que se presentará al usuario en caso de que no se pueda mostrar la imagen por alguna razón.
- height, width: establecen la altura y ancho respectivamente. Cuando se especifican, el navegador debe modificar las medidas originales de la imagen. Se trata del alto o ancho de la imagen en píxeles CSS en HTML5, o píxeles o como porcentaje en HTML 4.

Si se necesitara alterar el tamaño de la imagen, se debería usar CSS.

- ismap: sirve para definir un mapa de imágenes.
- usemap: sirve para asociar a la imagen un mapa de imágenes (como un marcador).

```

```

### 6.2. <map>

La etiqueta map sirve para crear un mapa de imágenes en el lado del cliente. Para crear dicho mapa se usará de forma conjunta con los elementos img y area. Mediante un mapa de imágenes se pueden definir secciones en una imagen y convertirlas en hipervínculos que apuntan a otros recursos. Esta funcionalidad hace posible que se pueda crear un vínculo a partir de una o más secciones de la imagen (img) a la que se encuentran asociados.

En este caso si hay etiqueta de cierre.

La imagen que se utilice para un mapa de imágenes deberá ser de uno de estos tres formatos: .jpg, .gif o .png, que son los tres formatos recomendables para todos los navegadores.

Este elemento deberá llevar asociado el atributo name para indicar el nombre del mapa.

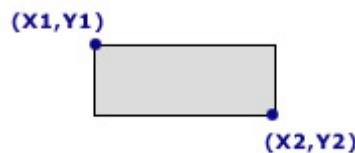
### 6.3. <area>

Con este elemento se puede especificar una región geométrica del mapa y el vínculo asociado a dicha región. Es decir, los mapas de imagen hacen posible la creación de un vínculo a partir de una o más secciones de la imagen (img) a la que se encuentran asociados.

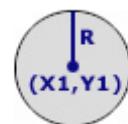
No cuenta con etiqueta final.

Atributos que puede contener, a parte de los globales, son shape, coords, href, nohref, alt, tabindex, accesskey, onfocus, y onblur:

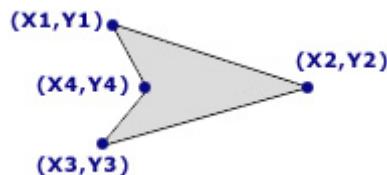
- **shape:** mediante este atributo se especifica el tipo de región con los valores:
  - **rect:** será una región de forma rectangular. Se ha de especificar el vértice superior izquierdo (x,y) y el vértice inferior derecho(x,y).



- **circle:** Región circular donde especificamos el centro y el radio.



- **poly:** Se trata de una región poligonal donde se han de especificar tantos puntos como sean necesarios. Para crearla debemos indicarle los diferentes puntos(x,y) del polígono que hagamos y de una forma ordenada.



- **coords:** indica la posición en pantalla, son valores de longitud partiendo de la esquina superior izquierda de la pantalla y separados por comas.
- **href:** indica el destino del área.

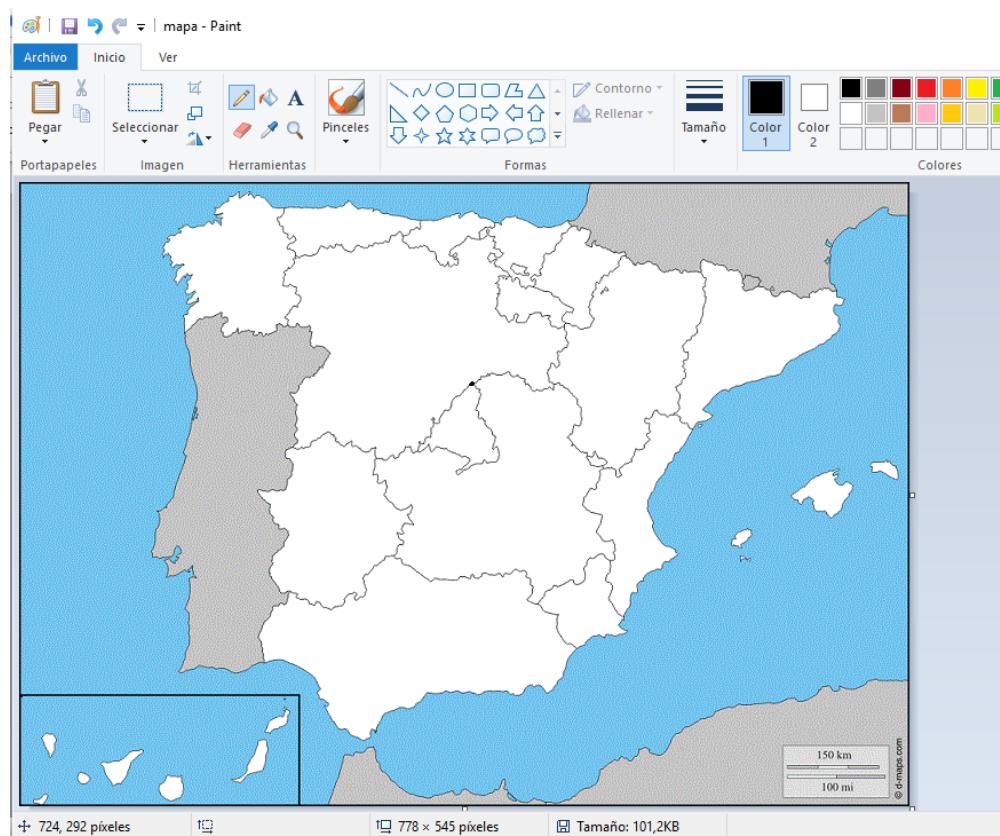
El formato, para cada opción, será el siguiente:

```
<area shape="RECT" coords="X1,Y1,X2,Y2" href="#">
```

```
<area shape="CIRCLE" coords="X1,Y1,R" href="#">
<area shape="POLY" coords=" X1,Y1, X2,Y2, X3,Y3, X4,Y4,....,XN,YN" href="#">
```

Para conocer las coordenadas que se deben introducir existen varias opciones:

- Al abrir la imagen con un editor de imágenes, por ejemplo paint, se pueden saber fácilmente. En la imagen siguiente se intuye un punto negro en la esquina superior de la comunidad de Madrid y en la parte inferior izquierda del editor se muestran las coordenadas del mismo.



- Para las áreas que vayan a ser de tipo poly, se puede utilizar un generador de mapas de imágenes, como por ejemplo <https://www.image-map.net/>
  - También se pueden ir viendo una a una en un editor de imágenes e ir apuntándolas pero es más tedioso.

### EJERCICIO 8

Continuando con el documento fabricaEmbutidos, introduce los siguientes cambios:

- Encapsuladas en un nuevo elemento div, introduce dos imágenes de jamones que se mostrarán de forma paralela. Se añadirán sendas descripciones. Las imágenes se mostrarán al final del documento.
- Para una de las imágenes pon en el enlace el nombre mal para poder ver que se muestra la descripción que se haya incluido.
- En la primera de las imágenes, crea un mapa de imágenes que se comportará de la



siguiente manera:

- Tendrá tres áreas diferentes, cada una de un tipo distinto.
- Una de ellas llevará a la página de información de Guijuelo en caso de que el usuario pinche sobre ella.
- La segunda de las áreas, enlazará a la página de una fábrica de embutidos cualquiera.
- La tercera, redirigirá al siguiente enlace:  
<https://www.enriquetomas.com/es/blog/propiedades-del-jamon-iberico-de-bellota-super-saludable/>