

Lenguajes de marcas y sistemas de gestión de información.

UT 1. Introducción a o lenguajes de marcas.

PRÁCTICA 1. Introducción a XML.

1. Preparando el entorno.

Trabajar con lenguajes de marcas o formatos de texto se puede hacer usando simples editores de texto como nano o notepad, pero esto hace que el trabajo sea lento, tedioso y poco productivo.

Existen también programas específicos para trabajar con lenguajes de marcas, en especial en todo el sistema XML, pero en el día a día de un desarrollador o desarrolladora no suele ser lo habitual, ya que, si bien es una parte destacada de su trabajo, no es la principal.

Por último, se tienen los entornos de desarrollo (IDE's), herramientas que integran lo necesario para realizar las tareas, entre otras, de programación. Los IDE's actuales poseen, de forma nativa o con extensiones de herramientas para trabajar con XML, JSON, YAML y otros lenguajes/formatos. En el día a día los desarrolladores usan mayoritariamente este tipo de programas.

La herramienta principal para este curso va a ser el entorno de desarrollo Visual Studio Code (de ahora en adelante VSC o Code)(no confundir con Visual Studio, mucho más complejo y pesado), junto con algunos de las extesiones (plugings") que posee.

1.1. Instalando VSC.

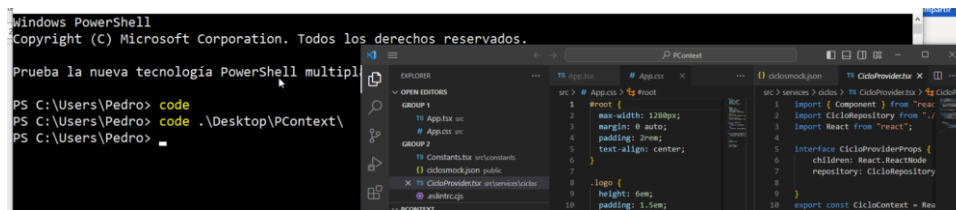
La instalación es sencilla, simplemente ir a la página web de VSC:

<https://code.visualstudio.com/>

Y descargar para el SO anfitrión:

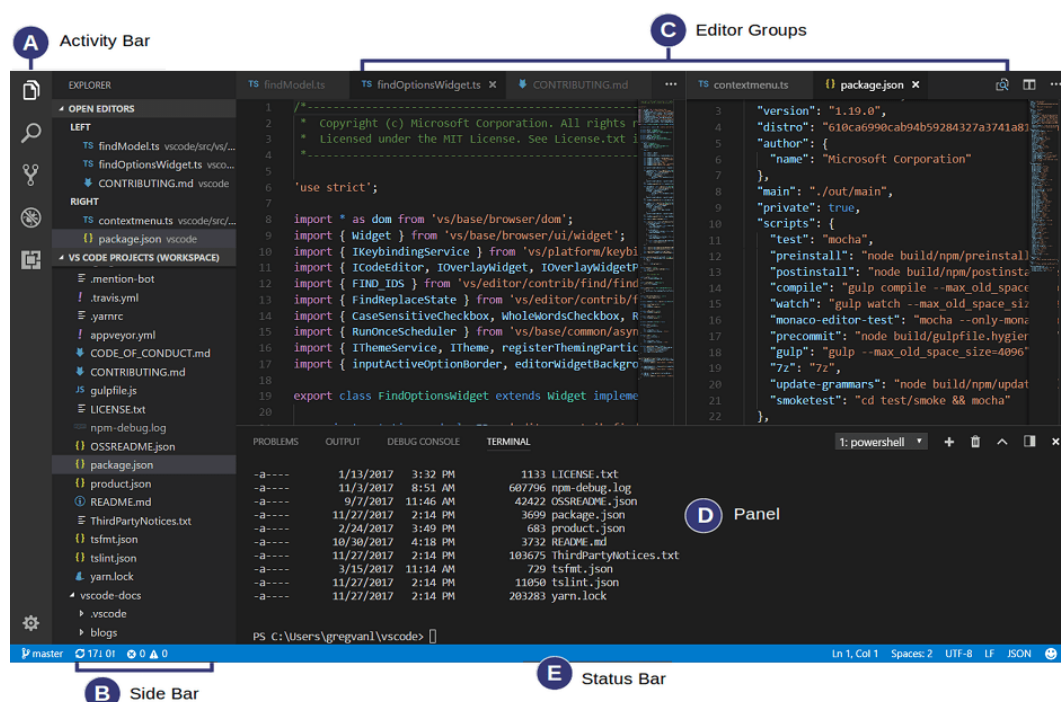


Una vez instalado se puede iniciar de forma clásica pulsando sobre el icono o desde la línea de comando usar code, si se desea abrir en el directorio actual code . O uno concreto code ruta.



1.2. El entorno. (Extraído de la documentación de VSC).

El entorno es el clásico de cualquier aplicación, si bien a medida que se añaden extensiones, el aspecto puede cambiar. Los elementos que se encuentran en la instalación por defecto son:

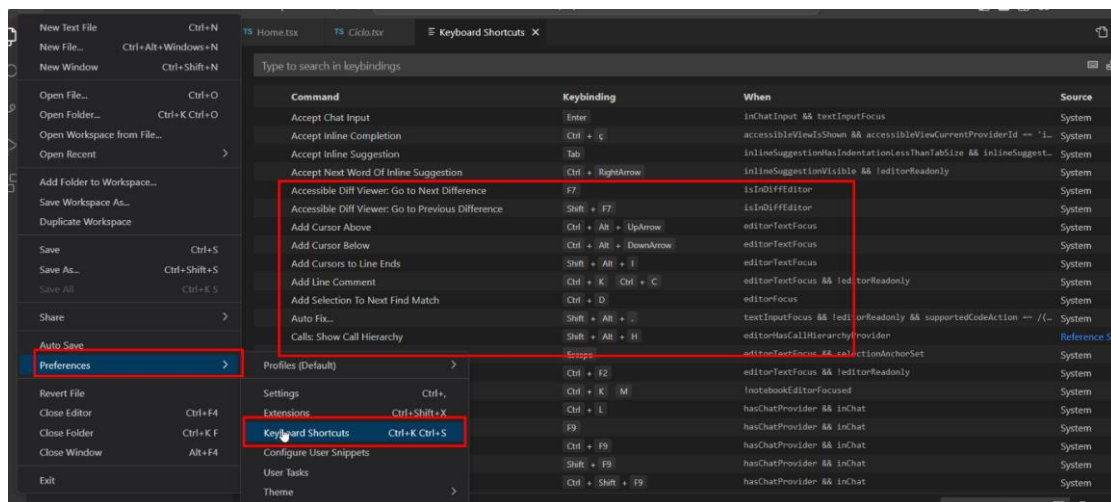


- (A) Barra lateral principal: contiene diferentes vistas, como el Explorador, para ayudar mientras se trabaja en el proyecto, o el uso de Git.
- (B) Panel: un espacio adicional para vistas debajo de la región del editor. De forma predeterminada, alberga resultados, información de depuración, errores y advertencias, y una terminal integrada. El panel también se puede mover hacia la izquierda o hacia la derecha para obtener más espacio vertical.
- (C) Editor: el área principal para editar archivos. Puede abrir tantos editores como desee, uno al lado del otro, vertical y horizontalmente.
- (D) Barra de actividad: ubicada en el extremo izquierdo, le permite cambiar entre vistas y le brinda indicadores adicionales específicos del contexto, como la cantidad de cambios salientes cuando Git está habilitado, los errores o un termina.

- (E) Barra de estado: información sobre el proyecto abierto y los archivos que edita.

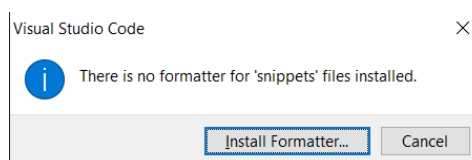
1.3. Atajos de teclado.

El uso del teclado es mucho más productivo que el ratón y teclado (de ahí el éxito en parte de Vim y Emacs), por defecto se define los clásicos atajos: Ctrl+C, Ctrl+V o Ctrl+A, siendo posible borrar, modificar y definir atajos propios pulsando File->Prefereces->Keyboard ShortCuts.



1.4. Formato.

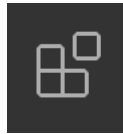
El código fuente ha de estar formateado de forma correcta ya que facilita el análisis y el manteniendo, la principal característica, aunque no la única es la de un tabulado correcto, pudiendo realizarse de forma manual (pérdida de tiempo) o de forma automática, estableciéndose en la mayoría de los IDE's la combinación Ctr+Alt+F o en el caso de VSC Shift+Alt+F, siendo la F de format. Cada lenguaje o incluso organización o programador/a puede definir su estilo de formato, en caso de que VSC no sepa como formatear un tipo concreto, pregunta si se desea instalar uno de los repositorios:



1.5. Extensiones.

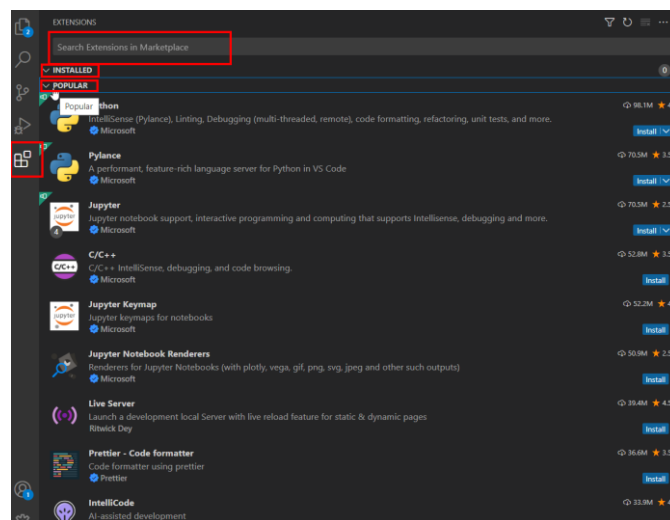
Las extensiones han sido uno de los puntos principales para que VSC sea uno de los IDE's más utilizados, sino el que más. Permiten agregar idiomas, depuradores y herramientas a la instalación.

Posee una tienda/repositorio de extensiones, ya sea usando la web o desde el mismo entorno con el icono:



Posee un buscador sencillo y 3 secciones:

- Instalas.
- Más populares
- Recomendadas.

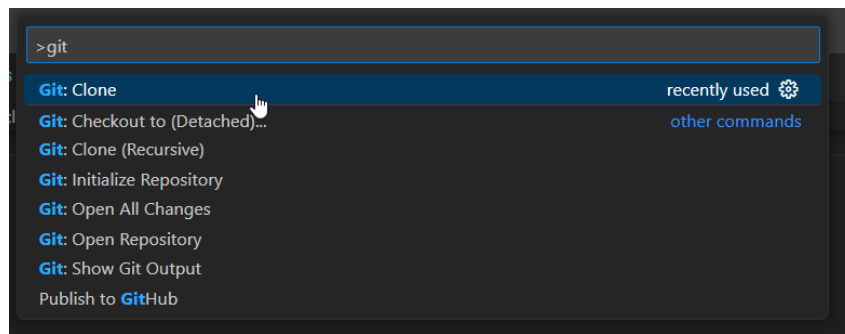


Se puede ver que entre las más descargadas se encuentra Python, C/C++ o Prettier(formateador de código). Al hacer “click” sobre una extensión se abre un panel en el que se tiene información de la extensión (comandos, ficheros de configuración, funcionamiento...) y un botón para instalar. Dependiendo de la extensión añadirá elementos o no al entorno, como comandos, opciones de botón derecho o diferentes paneles y vistas, por ejemplo, al añadir Docker (gestión de contenedores y virtualización) se añaden diferentes elementos.

1.6. Paleta de comandos.

La paleta de comando es uno de los elementos más usados del entorno, esta permite acceder a toda la funcionalidad del IDE, sin necesidad de tener decenas de ventanas y pestañas, sino haciendo uso de comandos.

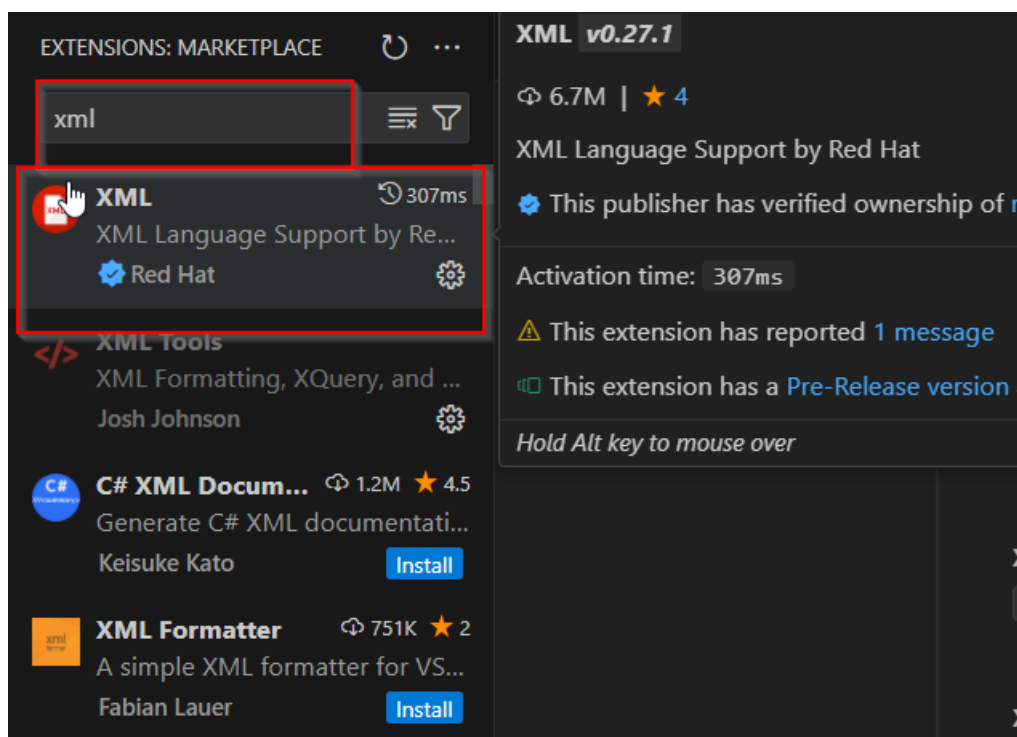
Para acceder a la paleta de comandos usar la combinación de teclas Ctrl+Shift+P, apareciendo en la parte superior central un cuadro de búsqueda con todos los comandos disponibles, por ejemplo, los comandos relacionados con Git:



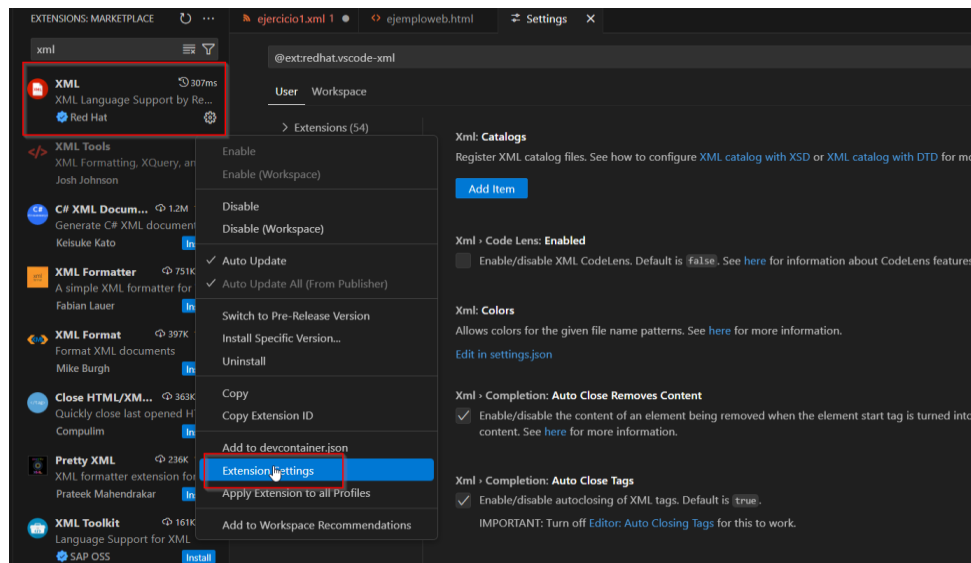
Además, aparece el atajo de teclado (si existe) para ejecutar el comando de forma directa.

2. Trabajando con XML.

Existen multitud de extensiones para trabajar con XML en VSC, normalmente los “plugins” con más descargas suelen ser los más útiles (cualquier persona puede crear y publicar “plugins” para VSC). En el caso de XML se tienen 2 XML de Red Hat y XML Tools de Josh Johnson, optando en esta práctica por la primera opción.

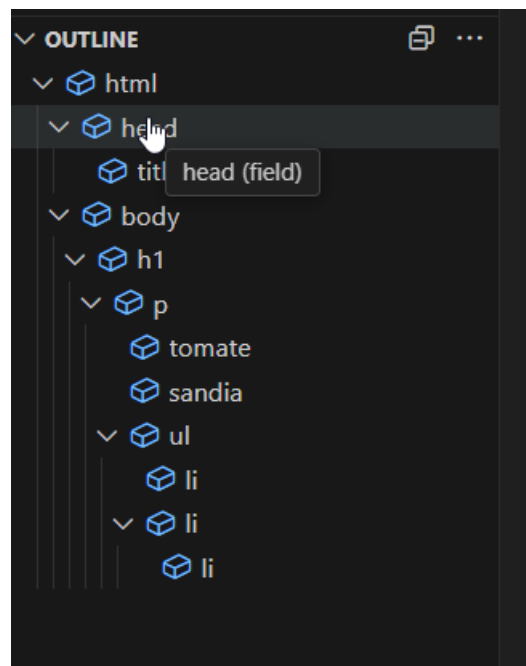


Es posible que sea necesario configurar algunas opciones de la extensión, para ello ir a la sección de plugins y en el pulsar la opción de configuración y abrir la ventana de “Extension Settings”.



Para crear un archivo ir a “File->New File”, indicar en el nombre del fichero la extensión en este caso xml.

En la parte izquierda si se ha activado ver el árbol XML aparece la pestaña OUTLINE, en el que se puede ver el mismo.



2.1. Ejercicio. (CE 1a)

Indicar las reglas para que un documento XML se encuentre bien formado. ¿Qué significa que está bien formado? (Se puede copiar de Internet, pero no en el examen o en una entrevista de trabajo)

2.2. Ejercicio. (1a,1g,1h)

Se tiene el siguiente documento XML, indicar si se encuentra o no bien formado y los cambios necesarios para que este bien formado. (Lo puede hacer Chapgpt, pero la empresa también puede pagar a Chapgpt y no a ti).

```
<playlist>
  <song title=La flaca">
    

  <song title="Si antes te hubiera conocido">

  <song>
    
    <song title="Peligrosa">
</playlist>
  </song>
```

Pueden existir diferentes soluciones, con lo que se crea ambigüedad, por ejemplo, toda canción debe tener un título y una imagen o una lista se compone de 0 o más canciones., a pesar de que el documento se encuentre bien formado. ¿Cómo se podría solucionar esa ambigüedad?

2.3. Ejercicio 2. (CE 1a, b,e, g, h)

Crear un documento XML con tus iniciales, por ejemplo, si te llamas Maximo Decimo Meridio, el nombre del fichero sera MDM.xml, usando VSC para un reproductor de música, siguiendo con el ejercicio anterior, el documento posee:

- Una lista de canciones, esta lista posee un nombre y una fecha.
- La lista posee un conjunto de canciones (insertar los 5 más populares en Spotty, por ejemplo).
- Cada canción ha de poseer:
 - Un título.
 - Un nombre.
 - Un artista o grupo.
 - Una imagen.
 - De forma opcional un comentario.

Realizar una captura de pantalla del árbol y otra del fichero.

2.4. Ejercicio. (C· 1a, 1b, 1e, 1g, 1h)

Ampliar el fichero anterior, de forma que ahora se pueda gestionar conjuntos de listas de reproducción, por ejemplo: Rock, Pop, Infantil, todo dentro del mismo fichero.

Realizar una captura de pantalla del nuevo árbol y del fichero.

2.5. Ejercicio. (1d, 1i)

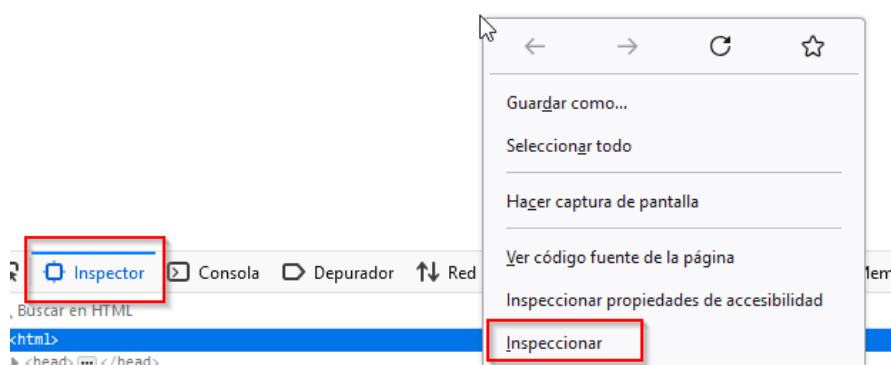
Un documento XML además de bien formado ha de ser válido ¿Qué significa esto? Relacionar con el ejercicio 1 y 2 en cuanto a la ambigüedad y los requisitos.

2.6. Ejercicio. (1c, 1d, 1f)

HTML es uno de los lenguajes de marcas más usados en la actualidad, utilizado para páginas web. Crear un nuevo fichero con tus iniciales+web.html con el siguiente código:

```
<html>
<head>
  <title> Página web de TUSINICIALES</title>
</head>
<body>
  <h1>Hola mundo+TUS INICIALES
  <p> En un lugar de la mancha...
    <tomate>Grande</tomate>
    <sandia>Mediana</sandia>
    <ul>
      <li>Uno</li>
      <li>Dos
      <li>Tres</li>
    </ul>
  </body>
</html>
```

¿Es un documento bien formado? Abrir con el navegador, botón derecho y ver el árbol. Explicar qué ha sucedido.



2.7. Ejercicio. (CE 1i)

Explicar qué son los espacios de nombres.

2.8. Ejercicio. (CE 1c, 1d, 1e, 1f, 1i)

El lenguaje Math sirve para definir de expresiones matemáticas, es un lenguaje definido usando XML y añadiendo una serie de reglas (se añaden con DTD o Schemas, se tratarán en temas posteriores).

Un ejemplo de fichero MathML:

```
<?xml version="1.0" encoding="UTF-8"?>
<math>
  <mrow>
    <mi>x</mi>
    <mo>=</mo>
    <mfrac>
      <mrow>
        <mi>-b</mi>
        <mo>±</mo>
        <msqrt>
          <mrow>
            <msup>
              <mi>b</mi>
              <mn>2</mn>
            </msup>
            <mo>-</mo>
            <mn>4</mn>
            <mo>×</mo>
            <mi>a</mi>
            <mo>×</mo>
            <mi>c</mi>
          </mrow>
        </msqrt>
      </mrow>
      <mrow>
        <mn>2</mn>
        <mo>×</mo>
        <mi>a</mi>
      </mrow>
    </mfrac>
  </mrow>
</math>
```

Al abrir el fichero con el navegador, se pinta la expresión de una ecuación de segundo grado:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- Crear un fichero tusinicialesMathML.xml con el contenido anterior y ver que funciona en el navegador.
- Introducir ahora una etiqueta como primera hija de “math” de la forma <tusiniciales>nombre completo</tusiniciales>. ¿El editor da algún fallo?

- Abrir en el navegador. ¿Qué sucede? ¿Existe la etiqueta <tusiniciales> en MathML?
- Añadir el espacio de nombre <http://www.w3.org/1998/Math/MathML> y las reglas para que solo pueda tener etiquetas de MathML y en el formato correcto: <http://www.w3.org/Math/XMLSchema/mathml2/mathml2.xsd>.
- ¿Qué sucede ahora en el editor con la etiqueta <tusiniciales>? ¿El fichero es un documento bien formado y válido?

3. Entrega.

En Aules.

- Las memorias e informes en formato PDF. **Si se encuentran en otro formato no se corregirán.**
- En las imágenes tiene que aparecer el nombre del fichero en el que aparecen las iniciales, en caso contrario se dan por erróneas.
- Memorias con solo capturas de pantalla no se corregirán.
- Carpeta con los ficheros creados.
- Fichero comprimido con el formato TUSINCIALES.zip.
- No se admiten retrasos.