

# Proyecto **FPMAD***digital*

*Recursos digitales y multimedia para Formación Profesional*



**Comunidad  
de Madrid**

Dirección General  
de Educación Secundaria,  
Formación Profesional  
y Régimen Especial

CONSEJERÍA DE EDUCACIÓN,  
UNIVERSIDADES, CIENCIA  
Y PORTAVOCÍA



Unión Europea

Fondo Social Europeo

*“El FSE invierte en tu futuro”*

**Financiado como parte de la respuesta  
de la Unión a la pandemia de COVID-19**

## CFGS Desarrollo de Aplicaciones Multiplataforma

módulo profesional

0486 - Acceso a Datos

unidad didáctica

02 Manejo de XML

resultados de aprendizaje

05 Desarrolla aplicaciones que gestionan la información almacenada en bases de datos nativas XML evaluando y utilizando clases específicas.



Comunidad  
de Madrid

Dirección General  
de Educación Secundaria,  
Formación Profesional  
y Régimen Especial

CONSEJERÍA DE EDUCACIÓN,  
UNIVERSIDADES, CIENCIA  
Y PORTAVOCÍA



Unión Europea

Fondo Social Europeo

“El FSE invierte en tu futuro”

Financiado como parte de la respuesta  
de la Unión a la pandemia de COVID-19

# **Resultados de aprendizaje y unidades didácticas**

# Resultados de aprendizaje y unidades didácticas

| RESULTADOS DE APRENDIZAJE |   |   |   |   |   | UNIDAD DIDÁCTICA                                  |
|---------------------------|---|---|---|---|---|---|
| 1                         | 2 | 3 | 4 | 5 | 6 |   |
| X                         |   |   |   |   |   | 1.- Ficheros, colecciones y data frames           |
|                           |   |   |   | X |   | 2.- Manejo de XML                                 |
|                           | X | X |   |   |   | 3.- Procesamiento de BBDD Relacionales            |
|                           |   |   | X |   |   | 4.- Uso de BBDD NoSQL                             |
|                           |   | X | X |   | X | 5.- Programación de componentes de acceso a datos |

# **Unidades didácticas y materiales asociados**

# Unidades didácticas y materiales multimedia

| RRAA |   |   |   |   |   | UDD   | Material Multimedia                              |
|------|---|---|---|---|---|---|--|
| 1    | 2 | 3 | 4 | 5 | 6 |   |  |
| X    |   |   |   |   |   | 1.- Ficheros, colecciones y data frames           | 1.1 Contenidos básicos<br>1.2 Ejemplos aplicados |
|      |   |   |   | X |   | 2.- Manejo de XML                                 | 2.1 Contenidos básicos<br>2.2 Ejemplos aplicados |
|      | X | X |   |   |   | 3.- Procesamiento de BBDD Relacionales            | 3.1 Contenidos básicos<br>3.2 Ejemplos aplicados |
|      |   |   | X |   |   | 4.- Uso de BBDD NoSQL                             | 4.1 Contenidos básicos<br>4.2 Ejemplos aplicados |
|      |   | X | X |   | X | 5.- Programación de componentes de acceso a datos | 5.1 Contenidos básicos<br>5.2 Ejemplos aplicados |

# **Repositorios de materiales y prácticas**

# Repositorio de materiales y prácticas

**Todos los proyectos mostrados, así como otros materiales utilizados en las unidades didácticas los podrás encontrar completos en:**

**<https://github.com/joseluisgs/FP-NextGen-AccesoDatos>**

Cualquier error o propuestas de mejora se publicarán en el repositorio indicado.  
Gracias por tu colaboración.



# Contenidos

1. XML
2. Esquemas
3. DOM
4. Librerías



# **XML**



# XML

Los ficheros **XML** permiten el intercambio de información entre aplicaciones utilizando para ello un fichero de texto plano al que se le pueden añadir etiquetas para darle significado a cada uno de los valores que se almacenan.

XML es el acrónimo de Extensible Markup Language, es decir, es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos.

Los documentos XML forman una estructura de tipo árbol, comenzando desde la raíz, con ramas hacia las hojas.

Todos los elementos pueden tener contenido textual y atributos.

```
1 <Users>
2   <User id="1">
3     <firstName>Pepe</firstName>
4     <lastName>Perez</lastName>
5     <age>28</age>
6     <gender>Male</gender>
7   </User>
8   <User id="2">
9     <firstName>John</firstName>
10    <lastName>Cena</lastName>
11    <age>45</age>
12    <gender>Male</gender>
13  </User>
14  <User id="3">
15    <firstName>Tom</firstName>
16    <lastName>Cruise</lastName>
17    <age>40</age>
18    <gender>Male</gender>
19  </User>
20 </Users>
```

# XML

**Namespaces:** proporcionan una forma de evitar conflictos de nombres entre elementos. Los nombres de elementos pueden coincidir entre diferentes documentos XML.

**XPath:** es un lenguaje que permite construir expresiones que recorren y procesan un documento XML.

**XSLT:** son las siglas de eXtensible Stylesheet Language Transformations, y es el lenguaje de hojas de estilo de XML. Con XSLT se pueden añadir/eliminar elementos y atributos del archivo de salida. Se pueden también reordenar elementos entre otras cosas o marcar elementos gracias a XPath.

**XLink:** se utiliza para crear hiperenlaces en documentos XML.

**XPointer:** permite a los enlaces apuntar a partes específicas de un documento XML usando XPath.

# Definición de XML

# Definición de XML

**DTD:** Document Type Definition, que son reglas que definen los elementos y atributos permitidos. Un DTD permite que un grupo de personas acuerden utilizar un DTD estándar para el intercambio de datos y así poder asegurar una uniformidad.

**XML Schemas:** Un esquema XML o XML Schema describe la estructura de un documento XML, como un DTD. Es una alternativa a DTD basada en XML. Los XML Schemas son más potentes que los DTD porque están escritos en XML, se pueden extender y soportan tipos de datos y namespaces.

```
1 <!DOCTYPE pelicula
2 [
3 <!ELEMENT pelicula (titulo,estreno,director,reperto)>
4 <!ELEMENT titulo (#PCDATA)>
5 <!ELEMENT estreno (#PCDATA)>
6 <!ELEMENT director (#PCDATA)>
7 <!ELEMENT reperto (#PCDATA)>
8 ]>
```

```
1 <xs:element name="pelicula">
2   <xs:complexType>
3     <xs:sequence>
4       <xs:element name="titulo" type="xs:string"/>
5       <xs:element name="director" type="xs:string"/>
6       <xs:element name="reperto" type="xs:string"/>
7     </xs:sequence>
8   </xs:complexType>
9 </xs:element>
```



# DOM



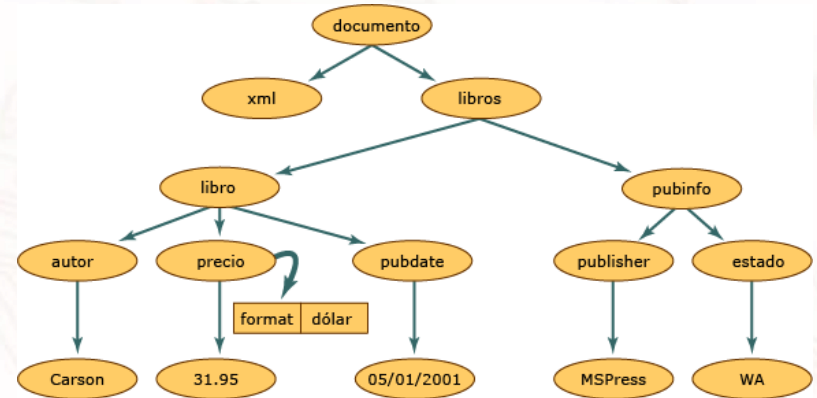
# DOM

Document Object Model o DOM ('Modelo de Objetos del Documento' o 'Modelo en Objetos para la Representación de Documentos')

El responsable del DOM es el World Wide Web Consortium (W3C).

El DOM permite el acceso dinámico a través de la programación para acceder, añadir y cambiar dinámicamente contenido estructurado en documentos con lenguajes de programación.

```
1 <?xml version="1.0"?>
2 <libros>
3   <libro>
4     <autor>Carson</autor>
5     <precio format="dollar">31.95</precio>
6     <pubdate>05/01/2001</pubdate>
7   </libro>
8   <pubinfo>
9     <publisher>MSPress</publisher>
10    <estado>WA</estado>
11  </pubinfo>
12 </libros>
```





The background of the slide is a light gray circuit board pattern with various colored lines (green, red, yellow) and small circular components.

# Librerías

A large, solid red rectangular box occupies the bottom half of the slide, likely intended for additional text or graphics.

# Librerías

**DOM Parser:** carga el fichero en memoria, creando la representación del mismo. Trabajamos nodo a nodo. Recomendable para ficheros pequeños.

**SAX Parser:** no carga el fichero en memoria ni crea representación del mismo. Los analizadores SAX están basados en un modelo de eventos: a medida que el parser recorre el documento éste informa de la ocurrencia de eventos, tales como el comienzo de un elemento XML o el final del documento, a un manejador de eventos (event handler).

**JDOM Parser:** nos permite leer, editar y escribir XML muy fácilmente y nos permite poder elegir entre SAX Parser, DOM Parser, STAX Event Parser, y STAX Stream Parser.

# Librerías

**JAXB:** Java Architecture for XML Binding (JAXB) define como los objetos son convertidos en JAVA desde/a XML y viceversa.

Gracias a su proceso de serializar o “marshalling” (convertir un objeto Java/Kotlin en XML) o deserializar o “unmarshalling” un XML (convertir un fichero XML en un objeto de Java/Kotlin) podemos trabajar orientados a objetos con este tipo de documentos.

La definición de la estructura del XML se puede hacer en base de un esquema XSD o de anotaciones sobre las propias clases de Java o Kotlin.

# Conclusiones



# ¡Vamos con la práctica!

"XML no es más lenguaje de programación que unas notas sobre una servilleta de papel"

- Charles Simonyi



**Comunidad  
de Madrid**

Dirección General  
de Educación Secundaria,  
Formación Profesional  
y Régimen Especial

CONSEJERÍA DE EDUCACIÓN,  
UNIVERSIDADES, CIENCIA  
Y PORTAVOCÍA



**Unión Europea**

Fondo Social Europeo

*“El FSE invierte en tu futuro”*

**Financiado como parte de la respuesta  
de la Unión a la pandemia de COVID-19**