

# 04

# **Desenvolupament Web en entorn Client**

**Cicle: Desenvolupament d'aplicacions web**

**Curs 2025-2026**

©José Masip Alonso

## 1. Introducció

En aquest document aprendrem a utilitzar AJAX amb arxius de text i XML .

## 2. Contingut

### 2.1. AJAX.

Asynchronous JavaScript And XML

*No es un llenguatge de programació. Es un conjunt de tecnologies*

- *HTML*
- *JavaScript*
- *CSS*
- *JSON*
- *PHP/ASP.NET/JSP*
- *XML*

Avantatges:

- *utilitza tecnologies ja existents*
- *Suportat per la majoria de navegadors moderns*
- *interactivitat. L'usuari no cal que espere a que estiguen les dades del servidor*
- *Portable (no requereix de plug-ins dom flash o els applets de Java)*
- *Major velocitat. no es necessari carregar tota la pagina*

Inconvenients:

- *No existeix el concepte de tornar a la pagina anterior*
- *Si guardem en "FAVORITOS", no necessàriament vorem el que desitjàvem.*
- *No suportat en navegadors antics*
- *No funciona si l'usuari desactiva Javascript*
- *Els programadors tenen que conèixer moltes tecnologies*
- *Si el servidor esta saturat, experimentarem retrasos que poden desconcertar a l'usuari final.*

## 2.2. Mètodes

Mètode	Descripció
<code>new XMLHttpRequest()</code>	Crea un nou objecte <code>XMLHttpRequest</code>
<code>open( metode, URL, async )</code>	Especifica el tipus de sol·licitut, la URL, i si la petició es síncrona/asíncrona. <i>metode</i> : el tipus de petició: GET o POST <i>url</i> : la ubicació de l'arxiu en el servidor <i>async</i> : true (asíncrona) o falso (síncrona)
<code>send( cadena )</code>	Envia la sol·licitud fora del servidor. Sols s'utilitza per a peticions POST
<code>Send()</code>	Envia la sol·licitud fora del servidor. Sols s'utilitza per a peticiones GET

## 2.3. Propietats

Propietat	Descripció
<code>onreadystatechange</code>	Event al que se li assigna una funció que es crida automàticament cada volta que es modifica la propietat <code>readyState</code>
<code>readyState</code>	Manté l'estat de la <code>XMLHttpRequest</code> . Canvia de 0 a 4: 0: petició no inicialitzada 1: connexió de servidor establida 2: petició rebuda 3: sol·licitud de processament 4: acabada la sol·licitud i la resposta està llesta
<code>status</code>	200: "Está bien" 404: Página no encontrada
<code>responseText</code>	Obté les dades de resposta com a un string
<code>responseXML</code>	Obté les dades de resposta com a un XML

## 2.4. Hola mon amb un arxiu txt ajax.

Per a entendre el funcionament d'ajax vorem primer un exemple i posteriorment justificarem el seu us.

Recordar cal utilitzar un servidor php ja siga a un contenidor (docker) o instal·lat (xampp).

Exemple d'us d'ajax:

```
<script type="text/javascript">
var peticio_http;

//funció que rep les dades, resposta petició ajax
function mostraContingut() {
// 0 No inicializado. 1 Cargando. 2 Cargado. 3 Interactivo. 4 Completado.
if(peticio_http.readyState == 4) {
    if(peticio_http.status == 200) { //si tot OK
        alert(peticio_http.responseText);
    }
    else{ //mostra el estat en format de text
        alert("error: "+peticio_http.statusText);
    }
}
}

//funció llança la petició ajax
function descarregaArxiu() {
    // Obtener instancia objeto XMLHttpRequest
    if(window.XMLHttpRequest) {
        peticio_http = new XMLHttpRequest();
    }
    else if(window.ActiveXObject) { //navegadors obsolets
        peticio_http = new ActiveXObject("Microsoft.XMLHTTP");
    }

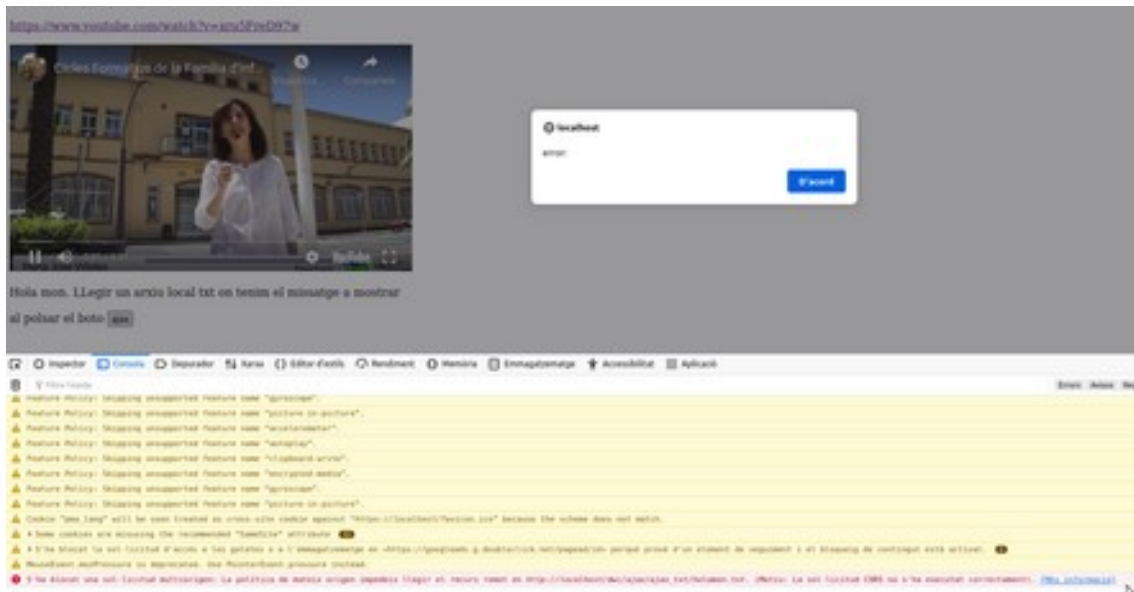
    // Prepara funcio rep dades del servidor
    peticio_http.onreadystatechange = mostraContingut;
    // Realizar peticio HTTP
    peticio_http.open('GET', 'http://localhost/ruta/holamon.txt', true);
    //comence el proces. envies les dades
    peticio_http.send(null);
}

window.onload = descarregaArxiu;
</script>
```

Cal crear un arxiu holamon.txt dins de la carpeta del servidor.

## Opcions:

- Si utilitzem el docker facilitat per al mòdul cal situar els arxius (html/js/txt) a la carpeta arxius (./dockerdwc/arxius o dockercompartida/arxius)
- si hem instal·lat xampp al nostre equip cal situar els arxius (html/js/txt) a la carpeta /opt/lampp/htdocs en linux o la ruta corresponen a Windows.



Si ens trobem un error: ***S'ha blocat una sol·licitud multiorigen: La política de mateix origen impedeix llegir el recurs remot en [http://localhost/dwc/ajax/ajax\\_txt/holamon.txt](http://localhost/dwc/ajax/ajax_txt/holamon.txt). (Motiu: La sol·licitud CORS no s'ha executat correctament)***. Es per la configuració de seguretat. Podem solucionar-ho canviant la ruta absoluta per una ruta relativa, així no considera que canviem de domini.

```
peticio_http.open('GET', 'http://localhost/ruta/holamon.txt', true);
peticio_http.open('GET', './holamon.txt', true);
```

**NOTA:** Error CORS

Aquest error el provoca el servidor. Es pot afegir unes línies al document PHP o al servidor per a solucionar aquest problema. Aquesta part no correspon al mòdul actual. Per tant, utilitzarem unes extensions de firefox per a solucionar el problema (sols per a nosaltres en local, no ho soluciona adequadament).

Extensions:

- CORS Everywhere (Permet activar i desactivar peticions CORS amb un botó). Simula que el servidor admet peticions des de qualsevol origen.
- Allow CORS: Access-Control-Allow-Origin (afegeix l'encapçalament Access-Control-Allow-Origin: \* a les respostes).

Quan utilitzen un servidor php podrem afegir aquesta capçalera directament als nostres arxius.

### 2.5. Justificació ajax.

Per a entendre el funcionament d'ajax primer intentarem fer el mateix sols amb php.

Aquesta pagina reproduceix un video de youtube. Mentrestant intentarem fer una petició php que simplement ens retornaria el contingut de l'arxiu holamon.txt a la pàgina original.

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>AJAX</title>
  <meta name="description" content="descripcio">
  <meta name="author" content="Jose Masip">
  <link rel="stylesheet" href="css/estilos.css">
</head>
<body>
  <h1>pagina php - sense ajax</h1>

  <a href="https://www.youtube.com/watch?v=xru5FreD97w">
https://www.youtube.com/watch?v=xru5FreD97w</a>
  <p>
    <iframe width="560" height="315"
src="https://www.youtube.com/embed/xru5FreD97w" title="YouTube video
player" frameborder="0" allow="accelerometer; autoplay; clipboard-write;
encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
  </p>
  <p>Hola mon. Llegir un arxiu local txt on tenim el missatge a mostrar</p>
  <p>al pulsar el boto </p>
  <form name="f" method="get" action="ajax-redirect.php" >
    <input type="submit" value="php">
  </form>
</body>
</html>
```

## Contingut de ajax-redirect.php

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>AJAX</title>
  <meta name="description" content="descripcio">
  <meta name="author" content="Jose Masip">
  <link rel="stylesheet" href="css/estilos.css">
  <script type="text/javascript">
    location.href="/ajax-video.php"
  </script>
</head>
<body>
</body>
</html>
```

Com podem observar cada volta que executarem la petició aconseguirem que la reproducció del video seria interrompuda i tornaria a escomençar des del principi. Al video acompanya aquesta unitat teniu demostració.

## Petició ajax.

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>AJAX</title>
  <meta name="description" content="descripcio">
  <meta name="author" content="Jose Masip">
  <link rel="stylesheet" href="css/estilos.css">
<script type="text/javascript">
  var peticio_http;

  function mostraContingut() {
    // 0 No inicializado. 1 Cargando. 2 Cargado. 3 Interactivo. 4 Completado.
    if(peticio_http.readyState == 4) {
      if(peticio_http.status == 200) { //si tot OK
        alert(peticio_http.responseText);
      }
    }
  }
```

```
        else{ //mostra el estat en format de text
            alert("error: "+peticio_http.statusText);
        }
    }
}

function descarregaArxiu() {
// Obtenir instancia objete XMLHttpRequest
if(window.XMLHttpRequest) {
    peticio_http = new XMLHttpRequest();
}
else if(window.ActiveXObject) { //navegadors obsolets
    peticio_http = new ActiveXObject("Microsoft.XMLHTTP");
}
// Prepara funcio rep dades del servidor
peticio_http.onreadystatechange = mostraContingut;
// Realizar peticio HTTP
peticio_http.open('GET', './ajax_txt/holamon.txt', true);
//comence el proces. envies les dades
peticio_http.send(null);
}
</script>
</head>
<body>
    <h1>pagina amb ajax</h1>
    <a href="https://www.youtube.com/watch?v=xru5FreD97w">
https://www.youtube.com/watch?v=xru5FreD97w</a>
    <p>
        <iframe width="560" height="315"
src="https://www.youtube.com/embed/xru5FreD97w" title="YouTube video
player" frameborder="0" allow="accelerometer; autoplay; clipboard-write;
encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
        </p>
        <p>Hola mon. LLegir un arxiu local txt on tenim el missatge a mostrar</p>
        <p>al pulsar el boto <input type="button" onclick="descarregaArxiu()"
value="ajax"> </p>
    </body>
</html>
```

Amb ajax podem fer mes d'una petició asíncrona al servidor que ens retorna les dades i no interrompim l'execució del vídeo.



## 2.6. Plantilla.

Model disponible per als exercicis i examen.

```
function () {  
  // 0 No inicializado. 1 Cargando. 2 Cargado. 3 Interactivo. 4 Completado.  
  if(peticio_http.readyState == 4) {  
    if(peticio_http.status == 200) { //si tot OK  
  
    }  
    else{ //mostra el estat en format de text  
      alert("error: "+peticio_http.statusText);  
    }  
  }  
}  
  
function (event) {  
  // Obtener instancia objeto XMLHttpRequest  
  peticio_http = new XMLHttpRequest();  
  // Prepara funcio rep dades del servidor  
  peticio_http.onreadystatechange = ;  
  peticio_http.open("", true);  
  //indiquem que les dades estan codificades com un formulari  
  //peticio_http.setRequestHeader("Content-Type","application/x-www-form-  
urlencoded");  
  //comence el proces. envies les dades  
  peticio_http.send();  
}
```

## 2.7. Formatar l'eixida.

Per a formatar l'eixida tenim que treballar amb ella com a cadena de text i afegir les etiquetes HTML.

Exemple mostrar h1 dins d'un div:

```
function mostraContingut() {  
  // 0 No inicializado. 1 Cargando. 2 Cargado. 3 Interactivo. 4 Completado.  
  if(peticio_http.readyState == 4) {  
    if(peticio_http.status == 200) { //si tot OK  
      document.getElementById("capa").innerHTML =  
"<h1>"+peticio_http.responseText+"</h1>";  
    }  
    else{ //mostra el estat en format de text  
      alert("error: "+peticio_http.statusText);  
    }  
  }  
}
```

Com aquest contingut el rebem com a una cadena de text caldrà formatar-la des de javascript (**no des de php**). Per exemple, si llegim un arxiu en moltes línies hi ha que separar aquest per el «intro» o «/n».

## 2.8. Hola mon amb XML.

Podem fer peticions ajax a arxius XML. Donat el següent arxiu d'exemple:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<saludo>
  <hola>
    hola mon
  </hola>
</saludo>
```

Modificaríem el nostre ajax per a que apunte a l'arxiu xml en volta del txt.

```
peticio_http.open('GET', './ajax_xml/holamon.xml', true);
```

Quan rebem les dades de la petició ajax obtindrem:

```
function mostraContingut() {
  if(peticio_http.readyState == 4) {
    if(peticio_http.status == 200) { //si tot OK
      //alert(peticio_http.responseText); //contingut arxiu amb etiquetes XML

      //alert(peticio_http.responseXML); //objecte XML
      //alert(peticio_http.responseXML.getElementsByTagName("hola")); //vector en totes les
      etiquetes hola
      //alert(peticio_http.responseXML.getElementsByTagName("hola")[0]); //primera
      etiqueta hola
      alert(peticio_http.responseXML.getElementsByTagName("hola")[0].firstChild.data+" 0");
      //text de l'etiqueta
      alert(peticio_http.responseXML.getElementsByTagName("hola")[0].innerHTML+" 1");
      //text de l'etiqueta
    }
    else{ //mostra el estat en format de text
      alert("error: "+peticio_http.statusText);
    }
  }
}
```

Podem utilitzar el `responseText` per a verificar que rebem bé les dades. Per a poder tractar-lo com a etiquetes necessitem utilitzar el `responseXML`.

## 2.9. Exemple cdclub.

L'arxiu xml té el següent format:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CATALEG>
  <CD>
    <TITOL>cantar</TITOL>
    <ARTISTA>tamara</ARTISTA>
    <PAIS>espanya</PAIS>
    <PREU>2</PREU>
    <ANY>2010</ANY>
  </CD>
  <CD>
    <TITOL>manolo</TITOL>
    <ARTISTA>manolo escobar</ARTISTA>
    <PAIS>espanya</PAIS>
    <PREU>5</PREU>
    <ANY>1985</ANY>
  </CD>
  <CD>
    <TITOL>cd</TITOL>
    <ARTISTA>juanito</ARTISTA>
    <PAIS>francia</PAIS>
    <PREU>23</PREU>
    <ANY>2012</ANY>
  </CD>
  <CD>
    <TITOL>cd2</TITOL>
    <ARTISTA>juanito</ARTISTA>
    <PAIS>francia</PAIS>
    <PREU>32</PREU>
    <ANY>2013</ANY>
  </CD>
  <CD>
    <TITOL>recopilatori</TITOL>
    <ARTISTA>dj</ARTISTA>
    <PAIS>espanya</PAIS>
    <PREU>4</PREU>
    <ANY>2008</ANY>
  </CD>
</CATALEG>
```

El nostre arxiu html seria:

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>AJAX</title>
  <meta name="description" content="descripcio">
  <meta name="author" content="Jose Masip">
  <link rel="stylesheet" href="css/estilos.css">
  <script type="text/javascript">
    var peticio_http;

    function mostratitol() {
      if(peticio_http.readyState == 4) {
        if(peticio_http.status == 200) { //si tot OK
          //vector amb tots els cds
          var
cds=peticio_http.responseXML.getElementsByTagName("CD");
          //recorrer els cds per a mostrar valors
          for (i=0;i<cds.length;i++){
            //obtindre els distints camps
            tit=cds[i].getElementsByTagName("TITOL");
            titol=tit[0].firstChild.data;
            art=cds[i].getElementsByTagName("ARTISTA");
            artista=art[0].firstChild.data;
            pai=cds[i].getElementsByTagName("PAIS");
            pais=pai[0].firstChild.data;
            pr=cds[i].getElementsByTagName("PREU");
            preu=pr[0].firstChild.data;
            an=cds[i].getElementsByTagName("ANY");
            any=an[0].firstChild.data;

            //mostrar taula amb datos
            document.getElementById("cdclub").innerHTML=
document.getElementById("cdclub").innerHTML+"<br><table
border='2'><tr><td rowspan='5'>caratula cd</td><td>"+titol+"</td> </tr>
<tr><td>"+artista+"</td></tr><tr><td>"+pais+"</td></tr><tr><td>"+preu+"
euros</td></tr><tr><td>"+any+"</td></tr></table>";
          }
        }
        else{ //mostra el estat en format de text
          alert("error: "+peticio_http.statusText);
        }
      }
    }

    function llegircds() {
      // Obtener instancia objeto XMLHttpRequest
      if(window.XMLHttpRequest) {
```

```

        peticio_http = new XMLHttpRequest();
    }
    // Prepara funcio rep dades del servidor
    peticio_http.onreadystatechange = mostratitol;
    // Realizar peticio HTTP
    peticio_http.open('GET', './ajax_xml/cdclub.xml', true);
    //comence el proces. envies les dades
    peticio_http.send(null);
}

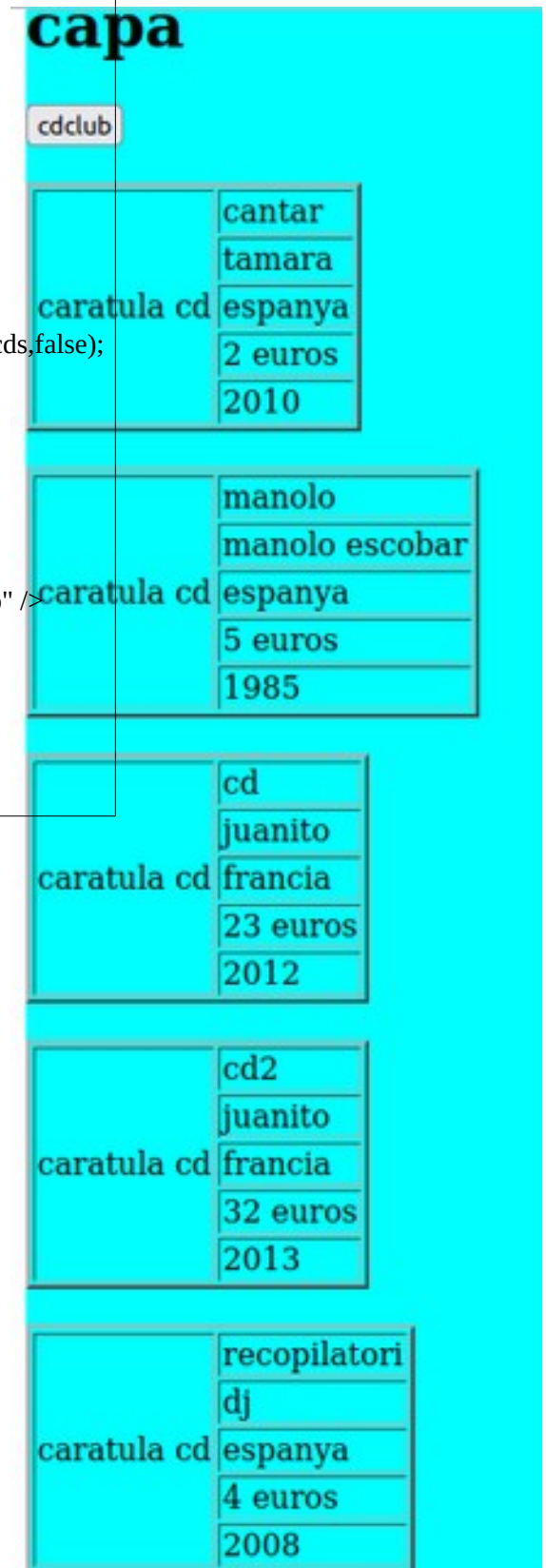
function inici(){

document.getElementById('boto').addEventListener('click',llegircds,false);
}

window.onload=inici;
</script>
</head>
<body>
    <div id="capa" style="background-color:cyan;"><h1>capa</h1>
        <input type="button" name="boto" id="boto" value="cdclub" />
        <br>
        <span id="cdclub"></span>
    </div>
    <p>Llegir les dades d'un xml. Mostra tota la cdteca</p>
</body>
</html>

```

El resultat el podem vore a la foto.



### 3. Activitats.

- E01. Tenim un arxiu on guardem les notícies. Les notícies les guardem com es mostra a la taula següent. Cal formatar mostrant el nom de l'usuari en un h1, a continuació la data en un h3 i el comentari en un paràgraf.

```
data:usuari:comentari de la noticia escrit per usuari
27-12-2012:juanito:comentari de la noticia
1-12-2012:manuel:asdf asdf asdf asdf asdf asdf asdf asdf asdf
11-11-2010:juanito:asdf jklñ jklñ jklñ jklñ jklñ jklñ jklñ
```

- E02. Idem amb el següent format d'arxiu. Descartarem la línia dels «-----».

```
data
usuari
comentari de la noticia escrit per usuari
-----
27-12-2012
juanito
comentari de la noticia
-----
1-12-2012
manuel
asdf asdf asdf asdf asdf asdf asdf asdf asdf asdf
-----
11-11-2010
juanito
asdf jklñ jklñ jklñ jklñ jklñ jklñ jklñ jklñ
-----
```

- E03. Tenim un arxiu on guardem les notícies amb format XML. Les notícies les guardem com es mostra a la taula següent. Cal formatar mostrant el nom de l'usuari en un h1, a continuació la data en un h3 i el comentari en un paràgraf.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<NOTICIARI>
  <NOTICIA>
    <DATA>27-12-2123</DATA>
    <AUTOR>juanito</AUTOR>
    <COMENTARI>COMENTARI de la NOTICIA</COMENTARI>
  </NOTICIA>
  <NOTICIA>
    <DATA>1-12-2045</DATA>
    <AUTOR>manuel</AUTOR>
    <COMENTARI>asdf asdf asdf asdf asdf asdf asdf asdf asdf asdf asdf asdf asdf
asdf</COMENTARI>
  </NOTICIA>
  <NOTICIA>
    <DATA>11-11-1945</DATA>
    <AUTOR>juanitosss</AUTOR>
    <COMENTARI>asdf jklñ jklñ jklñ jklñ jklñ jklñ jklñ</COMENTARI>
  </NOTICIA>
  <NOTICIA>
    <DATA>11-11-1785</DATA>
    <AUTOR>juanitosss</AUTOR>
    <COMENTARI>asdf jklñ jklñ jklñ jklñ jklñ jklñ jklñss</COMENTARI>
  </NOTICIA>
</NOTICIARI>
```

- E04. Idem amb l'anterior amb la modificació de que el camp AUTOR pot no apareixer. Si no tenim AUTOR a una noticia utilitzarem un Anonim per a aquest valor.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<NOTICIARI>
  <NOTICIA>
    <DATA>27-12-2123</DATA>
    <AUTOR>juanito</AUTOR>
    <COMENTARI>COMENTARI de la NOTICIA</COMENTARI>
  </NOTICIA>
  <NOTICIA>
    <DATA>1-12-2045</DATA>
    <COMENTARI>asdf asdf asdf asdf asdf asdf asdf asdf asdf asdf asdf asdf
asdf</COMENTARI>
  </NOTICIA>
  <NOTICIA>
    <DATA>11-11-1945</DATA>
    <COMENTARI>asdf jklñ jklñ jklñ jklñ jklñ jklñ jklñ</COMENTARI>
  </NOTICIA>
  <NOTICIA>
    <DATA>11-11-1785</DATA>
    <AUTOR>juanitosss</AUTOR>
    <COMENTARI>asdf jklñ jklñ jklñ jklñ jklñ jklñ jklñss</COMENTARI>
  </NOTICIA>
</NOTICIARI>
```

#### 4. **Activitats entregables.**

Entrega les següents activitats:

- E02
- E04

#### 5. **Bibliografia.**

- [https://www.w3schools.com/xml/ajax\\_intro.asp](https://www.w3schools.com/xml/ajax_intro.asp)
- <https://www.w3.org/TR/XMLHttpRequest1/>

#### 6. **Aclariments**

No intentes copiar exercicis d'altres companys o internet. ni tans sols pegar una miradeta. Si que pots consultar la sintaxis d'una funció, etc. Però no et serveix per a res intentar buscar la solució.

És el major error que pot realitzar qui comença a programar. Sols et deixarà una falsa sensació d'aprenentatge i després no sabràs solucionar els problemes per tu mateix.

Cal intentar que el resultat que genere el vostre codi siga el mes paregut possible a l'exercici (si hi ha imatge d'exemple).

Com els Navegadors no son compatibles entre sí al 100% jo corregire els examens amb FIREFOX. Per tant, es recomanable utilitzar aquest per als nostres exercicis.

Crea uns arxius per a cada exercici o apartat. Així un error a un no afectarà a la resta.



Per a entregar els exercicis crear una carpeta amb el vostre nom i comprimiu aquesta (el nom de l'arxiu comprimit serà elteunom.zip o elteunom.tar.gz). Si entregueu arxius solts o arxius genèrics sense identificar el vostre nom puc confondre'm i no puntuar-vos algun exercici al no saber de qui és.

Cal entregar tots els arxius necessaris per a l'exercici (fotos, llibreries, etc).