

|  |
| --- |
| VERÓNICA BONIS MARTÍN |
| MARIA CARMEN CORREA HERAS |
| ÁNGEL SÁNCHEZ-SIERRA CRUZ |
| JOSÉ MARÍA TENREIRO EIRANOVA |
| JUAN RAMON VARÓ NÚÑEZ  **REPOSITORIO GITHUB** |
| https://github.com/ITTDAW/Actividad2LdM.git |

Actividad 2. XML, DTD y XSD

lenguaje de marcas

**Vamos a elaborar una estructura de base de datos en XML que permita almacenar los datos de una biblioteca en la red. Para ello se pide que elabores un DTD que permita validar los documentos XML con las siguientes características:**

* **Existen tres tipos de documentos almacenados en la biblioteca: libros, revistas y periódicos. Todos los documentos están identificados por el atributo Id.**
  + **Para los libros este atributo empieza con la letra “L” seguido de 4 dígitos identificativos.**
  + **Para los periódicos este atributo empieza por la letra “P” seguido de los 4 dígitos identificativos.**
  + **En el caso de las revistas empieza por la letra “R”.**
* **Los libros a su vez son clasificados en novela, infantil o didáctico. Cada libro contiene un atributo identificativo de su clase denominado tipo\_clase. Dentro de cada libro se tiene un título, varios capítulos con el título en su interior, un índice y una sinopsis. Tanto en libro como en capítulo existe un atributo que contiene el número de páginas del libro o del capítulo, según corresponda. Cada capítulo contiene un elemento denominado contenido, en el que se tiene un atributo con el enlace a la información.**
* **Las revistas a su vez son clasificadas en: informática, corazón, coches, investigación y otras. Cada revista tiene el atributo tipo\_clase identificativo de la clase a la que pertenece. Dentro de cada revista tenemos el título, el número de la revista, un índice de contenido y las secciones. En cada sección y en la revista se tiene un atributo que contiene el número de páginas. Además, en cada sección se tiene la parte denominada contenido, en la que se tiene un atributo con un enlace a la información. De igual forma que en los anteriores, la última etiqueta del árbol debe ser el contenido con un atributo que referencia al contenido.**
* **Los periódicos se clasifican en nacionales e internacionales. Cada periódico contiene el atributo tipo\_clase identificativo de la clase a la que pertenece y un atributo que incluye la fecha de publicación.**
* **Dentro de los periódicos tenemos secciones y un índice. Cada sección debe contener un atributo identificativo del tipo de sección, que puede ser: económica, opinión, deportes, nacional o internacional. Las secciones se dividen en artículos, en donde se define en un atributo el autor. Finalmente, el contenido será el último elemento del árbol, que necesita un atributo que referencie a la información.**

**Requerimiento 1**

1. **Crear un XML con el modelo de datos indicado en el enunciado.**
2. **Elaborar un DTD que permita validar el documento XML.**
3. **Validar el documento con alguna aplicación externa.**

***IMPORTANTE: Cada alumno propondrá una solución de XML y DTD. Una vez todos los alumnos hayan puesto todas sus soluciones, se cotejará con el resto para poner la solución final.***

Para completar la tarea, cada persona trabaja por separado su versión de XML. Se hacen unas consideraciones previas:

* Trabajar en Visual Studio Code. Se recomienda instalar el plugin [*XML Language Support* by Red Hat](https://marketplace.visualstudio.com/items?itemName=redhat.vscode-xml) que nos facilitará la tarea de dar formato al archivo XML, corrección de errores, asignar sangrías automáticamente a los nodos, y que nos parece muy buena opción para trabajar.
* Trabajar en de nuevo en con GIT, subiendo cada comentario y documentos a la carpeta compartida.

Una vez preparado el entorno de trabajo y el método, comenzamos a analizar el problema que nos plantea el ejercicio. Para ello nos parece interesante seguir este flujo:

1. Crear una “estructura de árbol “en la que identifiquemos claramente la estructura jerárquica de los elementos que se incluirán en el XML (nodo raíz, subnodos, atributos…) que nos permita ir definiendo mejor las etiquetas.
2. Definir el nodo raíz y los subnodos mediante etiquetas.
3. Ir rellenando los atributos requeridos en cada elemento.

En primer lugar, debemos preocuparnos de que el XML **esté bien formado**, esto es:

* Los elementos deben seguir una *estructura jerárquica*. Cada elemento debe estar contenido dentro de otro, exceptuando el elemento raíz.
* Los elementos deben *estar alineados*; deben contener tabulaciones de forma que en la columna que tenemos la etiqueta de inicio de un elemento, debe estar también la de cierre.
* Los *elementos no pueden estar superpuestos*.
* Los *elementos deben estar cerrados*, es decir, todos los elementos deben contener su etiqueta de inicio y de cierre.
* *Solo puede existir un elemento raíz*, del que parten los demás.
* El *nombre de los elementos debe empezar siempre con un carácter no numérico* y nunca puede comenzar por XML o cualquier variación, es decir, no puede empezar por xml, Xml, XMl, xML, etc.
* Hay que *respetar las mayúsculas y minúsculas*. XML es un lenguaje case sensitive (sensible a mayúsculas y minúsculas).

Luego debemos comprobar que además de bien formado **es válido** (ver si cumple con las definiciones que se han incluido en el DTD y XSD).

**PASOS PARA TRABAJAR EN GRUPO**

La forma que ha tenido de trabajar el grupo ha sido mediante el repositorio GIT, en este repositorio se ha ido actualizando la página según cada miembro iba aportando algo a ella. Los pasos seguidos para ello en git han sido los siguientes:

1. Hay que decidir si se va a trabajar subiendo cambios con el perfil de Github del perfil creado (utilizar las contraseñas y nombre de usuario propias de ese perfil [ITTDAW@gmail.com](mailto:ITTDAW@gmail.com)) o utilizar nuestro propio perfil de Github y trabajar como colaborador(Se han enviado invitaciones a todos a los correos con los que cada persona se ha registrado en Github, hay que abrirlo y darle a aceptar)
2. Hacer un clone del repositorio en una carpeta de nuestro equipo

Git clone https://github.com/ITTDAW/Actividad1LdM.git

1. Entrar en la carpeta que se ha creado nueva dentro e iniciar el repositorio
2. Ahora estaremos en la rama master por defecto de nuestro repositorio local. Si se va a realizar cambios en el programa, se recomienda primero crear una rama propia para tener dos copias por si acaso.

Git checkout –b nombrerama

1. Ahora ya estaremos en una nueva rama de nuestro repositorio local. Realizar los cambios que consideremos en el programa que estamos haciendo, todas las pruebas que consideremos, es el momento de probar todo.
2. Cuando pensamos que hemos hecho algo valido para subir al repositorio, vamos a añadir nuestros cambios en nuestro repositorio local y rama propia.

Git add .

Git commit –m “Los cambios que hemos hecho”

1. Ahora ya tenemos nuestros cambios en nuestro repositorio local y rama propia, podríamos verlos con git log –oneline. Para subir estos cambios al repositorio remoto, es necesario que nos cambiemos a la rama master y fusionar nuestra rama personal con la master.

Git checkout master

Git merge nombredenuestrarama

1. En estos momentos nuestra rama master ya puede subirse al repositorio remoto. Si lo hacemos desde nuestro perfil de github, hemos aceptado el correo de aceptación de colaboradores y en nuestro equipo ya tenemos las credenciales de nuestro perfil de github, solamente con git push origin master se deberían de subir los cambios. Si trabajamos desde el perfil de ITTDAW tendremos que loguearnos cuando lo pida el gitbash y no tener guardadas nuestras credenciales en Windows de nuestro perfil personal.
2. CONSEJOS: Siempre antes de empezar a trabajar, hacer git pull en la rama master de tu repositorio local para ver si algún compañero ha hecho algún cambio, después fusionar esos cambios con tu rama personal y entonces si empezar a trabajar.
3. Si a alguien se le complica su repositorio local, que no se preocupe, se hace otro git clone en su equipo y vuelve a empezar desde el punto 2.