

MEMORIA DE LA TAREA LMSGI04

**JOSÉ ANTONIO MARTÍNEZ
RUIZ**

2º DAM

EJERCICIO 1

1. Introducción

En este ejercicio se ha solicitado crear un esquema en XML para almacenar datos relacionados con las cárceles de España, incluyendo información sobre prisiones, funcionarios, reclusos y celdas. El diseño se realizó utilizando un **DTD** (Document Type Definition) para definir la estructura del documento XML y garantizar que los datos que se ingresen en el archivo cumplan con un formato adecuado.

2. Descripción del diseño del archivo DTD

El archivo **DTD** define la estructura y las reglas para los elementos que componen la información de las cárceles. En este caso, se han definido varios elementos principales:

- **Prisión:** Cada prisión tiene un **código nacional**, un **nombre**, la **información del director**, la **dirección**, la **fecha de inauguración**, la **capacidad máxima** y otros detalles.
- **Funcionario:** Para cada funcionario de la prisión, se almacena el **puesto**, la **fecha de incorporación** y el **nombre completo**.
- **Recluso:** Cada recluso tiene información sobre su **número de recluso**, **nombre completo**, **número de celda**, **fecha de ingreso**, **duración de la condena** y **clasificación de peligrosidad**.
- **Celda:** Cada celda tiene un **número de celda**, el **pabellón** y las **plazas disponibles**.

El diseño del DTD fue realizado de manera que se respete una estructura jerárquica clara, facilitando la organización y la validación de los datos.

Estructura del DTD:

- Utilicé elementos como **prisión**, **funcionario**, **recluso** y **celda**.
- Cada uno de estos elementos tiene atributos y subelementos obligatorios, como la **fecha de incorporación** o el **número de celda**.
- Se emplearon tipos de datos como **#PCDATA** (parsed character data) para cadenas de texto y **#REQUIRED** para asegurar que ciertos datos sean ingresados en todos los casos.

3. Descripción del diseño del archivo XML

El archivo **XML** contiene los datos estructurados siguiendo el formato definido por el DTD. Los datos se presentan en un formato que sigue estrictamente las reglas de validación del DTD.

- **Datos de las prisiones:** Se crearon dos prisiones de ejemplo, cada una con su código, nombre, director, dirección, capacidad y demás datos solicitados.
- **Datos de funcionarios y reclusos:** Se incluyeron dos funcionarios y dos reclusos en cada prisión, con la información detallada de cada uno.
- **Celdas disponibles:** Se describieron las celdas de cada prisión, especificando su número, pabellón y plazas disponibles.

4. Decisiones tomadas y justificación

A lo largo del proceso de diseño, se tomaron varias decisiones clave:

- **Elección del DTD:** Se optó por usar un **DTD externo** en lugar de un **DTD interno** para mantener el XML más limpio y modular. Esto permite reutilizar el DTD en otros proyectos o documentos.
- **Uso de elementos obligatorios:** Se definieron muchos elementos como **REQUIRED** en el DTD para garantizar que toda la información esencial sea provista, como el **nombre de la prisión, fecha de inauguración y número de recluso**.
- **Jerarquía clara:** Se estructuró la información en una jerarquía lógica, empezando por la prisión, y luego descendiendo hacia los funcionarios, reclusos y celdas. Esto facilita la lectura y organización de los datos.
- **Tipos de datos:** Para garantizar la validez de los datos, se definieron tipos de datos adecuados para cada campo, como fechas para **fecha de inauguración y fecha de nombramiento**, y texto libre para los **nombres**.

5. Conclusión

El diseño realizado cumple con los requisitos establecidos para almacenar la información de las prisiones, los funcionarios, los reclusos y las celdas. Se ha utilizado un DTD para definir un esquema que asegura la validez y consistencia de los datos almacenados en el archivo XML. El proceso de validación y estructuración se ha realizado con el fin de optimizar la claridad, el mantenimiento y la interoperabilidad de los datos.

Este ejercicio ha permitido profundizar en la creación de documentos XML bien estructurados y en la correcta utilización de herramientas como el DTD y la validación de XMLs.

EJERCICIO 2

1. Introducción

En este ejercicio se ha solicitado crear un esquema en XML para almacenar los datos de una tabla periódica. Para garantizar la validez y consistencia de los datos, se utilizó un archivo XSD (XML Schema Definition), el cual define la estructura de los datos y asegura que cualquier documento XML que lo utilice cumpla con las reglas establecidas. El esquema cubre diversos aspectos de la tabla periódica, como los elementos químicos (átomos), sus propiedades físicas (como punto de ebullición y densidad), y su estado físico (sólido, líquido o gaseoso).

2. Descripción del diseño del archivo XSD

El archivo XSD define la estructura y las reglas para los elementos que componen la tabla periódica. Los principales elementos definidos en el esquema son:

- **tabla_periodica:** Este es el elemento raíz que contiene una lista de átomos.
- **atomo:** Cada átomo tiene varios subelementos, tales como nombre, simbolo, numero_atomico, peso_atomico, punto_ebullicion, y densidad. También tiene un atributo llamado estado, que indica el estado físico del átomo (sólido, líquido o gaseoso).
- **punto_ebullicion y densidad:** Son elementos complejos que contienen un valor numérico y un atributo para especificar la unidad de medida.
- **atributos:** El atributo estado define el estado físico del átomo (gaseoso, sólido, líquido). El atributo unidades se utiliza para especificar las unidades de medida, como "kelvin" para el punto de ebullición o "gramos/centímetros cúbicos" para la densidad.
- **maxOccurs="unbounded":** Este atributo permite que el elemento atomo se repita tantas veces como sea necesario para representar todos los átomos de la tabla periódica.

Decisiones clave en el diseño del XSD:

- **Uso de tipos de datos adecuados:**
 - **xs:string** para los elementos nombre, simbolo, y estado porque son cadenas de texto.
 - **xs:int** para numero_atomico ya que se trata de un valor entero.
 - **xs:decimal** para peso_atomico, punto_ebullicion y densidad porque son valores numéricos con decimales.
- **Estructura jerárquica clara:** Los átomos están representados como elementos dentro del elemento raíz tabla_periodica, lo que facilita la organización y lectura del archivo XML.

3. Descripción del diseño del archivo XML

El archivo XML fue creado siguiendo la estructura definida en el archivo XSD. En este archivo se almacenan datos de ejemplo sobre los elementos químicos de la tabla periódica.

Estructura y contenido del XML:

- **Nombre, símbolo, número atómico, peso atómico:** Para cada átomo, estos elementos están representados con los valores correspondientes.
- **Punto de ebullición y densidad:** Estos valores se especifican junto con las unidades correspondientes (por ejemplo, "kelvin" para el punto de ebullición y "gramos/centímetros cúbicos" para la densidad).
- **Estado físico:** Se especifica si el átomo está en estado sólido, líquido o gaseoso.

Decisiones tomadas y justificación:

- **Selección de elementos y atributos:** Cada átomo se representa de forma completa con su nombre, símbolo, número atómico, peso atómico, punto de ebullición, densidad y estado físico. Los atributos estado y unidades permiten añadir información adicional sin necesidad de crear nuevos elementos.
- **Reutilización de unidades:** El valor de las unidades para punto_ebullicion y densidad se especifica en los atributos correspondientes. Esto asegura que los datos sean fáciles de interpretar y comparar.

4. Validación de los archivos

Una vez creados el archivo XSD y el archivo XML, se procedió a la validación para asegurarse de que el archivo XML cumpla con la estructura definida en el archivo XSD.

- **Validación del archivo XML:** Utilizando un editor de texto como **Visual Studio Code**, se validó que el XML sea correcto y cumpla con el esquema establecido en el archivo XSD. Si el archivo XML no se ajustaba al XSD, el validador proporcionaba un error que indicaba la ubicación del problema.
- **Evidencia de la validez:** La validación exitosa del archivo XML asegura que los datos están bien estructurados y son consistentes con el esquema definido en el XSD.

5. Conclusión

El ejercicio fue completado con éxito, y el diseño de los archivos XML y XSD asegura que los datos de la tabla periódica estén bien organizados y sean válidos. El uso del XSD permitió crear un esquema reutilizable y modular que valida cualquier documento XML relacionado con la tabla periódica, mientras que el archivo XML contiene los datos de los elementos de manera clara y estructurada. Este ejercicio ha permitido profundizar en el uso de esquemas XML (XSD) y en la creación de archivos XML válidos, lo cual es fundamental para garantizar la calidad y la interoperabilidad de los datos en sistemas informáticos.

EJERCICIO 3

1. Introducción

En este ejercicio se ha solicitado crear un archivo XML y su correspondiente esquema XSD para gestionar una agenda de contactos. La agenda está compuesta por varias fichas de contacto, donde cada ficha contiene datos como el nombre, los apellidos, números de teléfono, direcciones de correo electrónico y direcciones postales. El objetivo es estructurar esta información de forma que sea válida, consistente y fácilmente accesible mediante la definición de un esquema XSD (XML Schema Definition).

2. Descripción del diseño del archivo XSD

El archivo XSD define la estructura de la agenda de contactos y las reglas que deben seguir los elementos y atributos dentro del documento XML. A continuación, se describen los principales componentes del archivo XSD:

- **Tipos de datos simples:**
 - **telefono_nacional:** Define un número de teléfono nacional, restringido a un patrón específico de dígitos (empieza con 6, 7, 8 o 9 seguido de 8 dígitos).
 - **telefono_internacional:** Define un número de teléfono internacional, restringido al patrón de "+XX XXXXXXXXXX" (donde XX es el código del país y X es el número del teléfono).
 - **email:** Define la estructura de una dirección de correo electrónico utilizando una expresión regular que valida el formato estándar de un email.
 - **codigo_postal:** Define un código postal español, restringido a números entre 10000 y 99999.
- **Tipos de datos complejos:**
 - **direccion_postal:** Este tipo complejo agrupa los elementos calle, piso, letra, codigo_postal, ciudad y pais para representar una dirección completa. El piso y la letra son opcionales.
 - **ficha:** Cada ficha de contacto contiene los elementos nombre, apellido1, apellido2, telefono_nacional, telefono_internacional, email y direccion_postal. Además, cada ficha tiene dos atributos:
 - **categoria:** Define el tipo de contacto (por ejemplo, empresa, particular, comercial).
 - **zona:** Define una zona adicional asociada al contacto, que es opcional.
- **Elemento principal:**
 - **agenda:** Este es el contenedor principal que puede contener múltiples elementos ficha, cada uno representando un contacto en la agenda.

Decisiones clave en el diseño del XSD:

- Se utilizaron **tipos de datos simples** (como telefono_nacional, telefono_internacional, email, etc.) para garantizar que los datos se ajusten a los formatos específicos que son esperados.
- La estructura **compleja** de ficha permite incluir información de contacto detallada, que incluye nombres, apellidos, números de teléfono, correos electrónicos y direcciones postales, todo organizado de manera jerárquica.
- La utilización de **atributos** en lugar de elementos para categoría y zona facilita la representación de estos valores como propiedades del contacto sin necesidad de un nodo adicional.

3. Descripción del diseño del archivo XML

El archivo XML contiene los datos de la agenda, conforme al esquema XSD previamente definido. Este archivo sigue la estructura jerárquica y las restricciones establecidas en el XSD.

Estructura y contenido del archivo XML:

- Cada **ficha** de contacto contiene información sobre:
 - **Nombre y apellidos:** Obligatorios.
 - **Teléfonos:** Se permiten hasta 5 teléfonos nacionales y 2 internacionales por contacto, aunque no todos los contactos usan la cantidad máxima.
 - **Correo electrónico:** Se permiten hasta 2 correos electrónicos por ficha.
 - **Dirección postal:** Puede incluir la dirección con el nombre de la calle, código postal, ciudad y país, y puede contener hasta tres direcciones postales por contacto.

Decisiones tomadas y justificación:

- **Estructura modular:** La organización de la información en elementos de **ficha** y subelementos como telefono_nacional, telefono_internacional, email, etc., permite representar adecuadamente los diferentes atributos de cada contacto de manera clara.
- **Categorías y zonas:** La inclusión de atributos como categoría y zona facilita la clasificación de los contactos, y permite extender el formato para agregar más información en el futuro sin alterar la estructura básica.

4. Validación de los archivos

Para garantizar la validez del archivo XML según el esquema XSD, se utilizó una herramienta de validación. Esto permitió verificar que los datos se ajustan a las restricciones definidas en el XSD, como el formato de los números de teléfono, el correo electrónico y los códigos postales.

Validación exitosa: Después de realizar la validación, se comprobó que tanto el archivo XML como el XSD están correctamente estructurados y cumplen con las reglas establecidas. En caso de errores de validación (como un formato incorrecto de teléfono o email), la herramienta de validación proporcionó detalles precisos sobre los problemas.

5. Conclusión

Este ejercicio ha permitido crear una agenda estructurada y validada de contactos mediante el uso de un archivo XML y un esquema XSD. El diseño fue cuidadosamente realizado para asegurar que se cumplieran las restricciones necesarias, como los formatos de teléfono y email. Además, se hizo un uso eficiente de los tipos de datos simples y complejos para representar de manera clara y modular la información de cada contacto. La validación del archivo XML garantizó la consistencia y calidad de los datos, lo que asegura su utilidad y fiabilidad en sistemas informáticos.

Este ejercicio ha servido para profundizar en el uso de **XML** y **XSD** para la creación de documentos estructurados y validados, y para aprender cómo garantizar la integridad de los datos en aplicaciones de gestión de información.