

Tarea 2: Documentos XML y vistas HTML mediante XSLT

Universidad de Málaga

Asignatura: Estándares de Datos en Bioinformática y Salud

¿Qué hemos hecho?

En esta segunda tarea hemos transformado la base de datos diseñada en la primera tarea (MongoDB + JSON) a un conjunto completo de documentos XML, sus correspondientes esquemas XSD y plantillas XSLT para generar visualizaciones HTML. Además, hemos desarrollado un script en Python para poblar automáticamente los XML a partir de los datos estructurados.

El objetivo principal es garantizar interoperabilidad y claridad estructural, los cuales son aspectos fundamentales en la bioinformática.

Diseños de los distintos esquemas:

Según el orden del flujo de trabajo que hemos seguido:

1. Esquemas XSD

En primer lugar, creamos los esquemas XSD que definen:

- El tipo de datos.
- Las relaciones jerárquicas.
- Los elementos (que pueden ser obligatorios u opcionales).
- Las referencias entre entidades.

2. Creación de los documentos XML:

Partiendo de los esquemas XSD, generamos los documentos XML correspondientes, replicando la estructura de cuatro niveles de anidamiento utilizada en la primera tarea.

3. Población de los documentos XML:

Diseñamos el script `json_to_xml.py` que:

- Toma como entrada los datos JSON estructurados, exportados de MongoDB.
- Genera XML válidos según sus esquemas XSD.
- Permite actualizar los XML fácilmente cuando los datos cambian.

4. Transformación con XSLT:

Por último, desarrollamos plantillas XSLT para transformar los esquemas XML a HTML, permitiendo así visualizar los datos de una forma clara y ordenada.

Conclusión

Esta segunda tarea completa la transición del modelo MongoDB a un conjunto de estándares abiertos validados. Mediante el uso de XML, XSD y XSLT podemos asegurar interoperabilidad y claridad estructural, aspectos fundamentales en el ámbito de la bioinformática.

Esta segunda tarea completa la transición del modelo MongoDB a un conjunto de estándares abiertos y validados. Mediante el uso de XML, XSD y XSLT podemos asegurar:

- Interoperabilidad: los datos pueden ser compartidos entre sistemas distintos.
- Claridad estructural: los datos son legibles y bien organizados.
- Automatización y mantenimiento sencillo: mediante el script Python.

Este flujo de trabajo establece una base sólida para futuras tareas en bioinformática y salud, donde el manejo eficiente de datos estructurados es crítico.

Repositorio: <https://github.com/Husakaa/data-standards-project>