

Ejercicio 3: Traducción JSON a XML

- a. Com has manejat el valor null en l'element age de l'Anna en la traducció a XML? És aquesta la millor manera de representar la falta d'informació? Proposa totes les alternatives possibles.

Per manejar el valor null a l'element "age" d'Anna en la traducció a XML, s'ha utilitzat un element d'edat buit (`<age></age>`). Aquesta no és la millor manera de representar la falta d'informació, ja que un element d'edat buit pot ser confós amb un valor d'edat vàlid de 0. Per representar la falta d'informació, és millor fer servir un atribut o un element separat per indicar que la informació no està disponible. Alternatives possibles per representar la falta d'informació podrien ser: Fer servir un atribut o Utilitzar un element separat.

- b. Què haurem de tenir en compte quan tenim elements repetits com ara les mascotes o els amics? S'ha mantingut la consistència en la traducció?

Quan tenim elements repetits com ara les mascotes o els amics, és important mantenir la consistència en la traducció. En aquest cas, sembla que s'ha mantingut la consistència, ja que els elements de mascotes i amics estan emmagatzemats de manera similar dins dels seus respectius contenidors (`<pets>` i `<friends>`). Cada mascota o amic és representat com un subelement amb el mateix nom (`<pet>` i `<friend>`), i s'ha mantingut l'estructura jeràrquica del document XML.

Ejercicio 4: Traducción XML a JSON

- a. Explica què s'ha convertit en objectes, i què arrays i per què has pres aquestes decisions.

S'han convertit els objectes que representen cada dispositiu (computer, tablet, smartphone) amb els seus atributs com a propietats. Aquests objectes estan continguts dins d'un array anomenat "devices". S'ha pres aquesta decisió perquè cada element de "devices" representa un dispositiu individual, i agrupar-los en un array facilita l'accés i la gestió de múltiples dispositius.

- b. Explica què has fet per tal de mantenir junta la informació del preu amb el tipus de moneda pagada. Com has transformat, en aquest cas, els atributs de l'XML a JSON i per què?

He utilitzat un únic camp "price" per emmagatzemar el preu, i un camp "currency" per indicar la moneda. Aquesta decisió s'ha pres per mantenir junta la informació del preu amb la moneda associada i per simplificar la representació de les dades. En aquest cas, s'ha preferit fer servir dos camps separats per claredat i per evitar confusions.

- c. Hi ha alguna etiqueta en l'XML que no s'ha traduït directament a JSON? Creus que això significa que s'ha perdut informació?

L'etiqueta <internal_info> de l'XML no s'ha traduït directament a JSON. Això no implica necessàriament que s'ha perdut informació, ja que podria ser una informació interna o específica de la implementació que no té sentit ser representada en el JSON de manera directa.

- d. Com has gestionat els caràcters especials com les cometes dobles en la traducció? Com afecta això la llegibilitat del JSON?

Els caràcters especials com les cometes dobles s'han escapçat amb barres invertides (") per gestionar-los correctament en JSON. Això assegura la validesa del JSON i evita problemes amb la interpretació dels caràcters especials. Afecta la llegibilitat en el sentit que pot fer que el JSON sigui una mica més difícil de llegir per humans, però és una pràctica comuna per garantir la integritat de les dades.

- e. Explica com has tractat els elements sense informació o amb dades opcionals. Has optat per deixar el camp buit, per fer servir el valor null o per ometre el camp? Explica quina creus que és la millor decisió i per què.

Pels elements sense informació o amb dades opcionals, s'ha optat per deixar el camp buit o no incloure'l si no és rellevant. Aquesta decisió depèn del context i de si la informació és realment opcional o si la seva absència pot tenir un significat específic. En general, és preferible utilitzar el valor null per indicar la falta d'informació, ja que això proporciona una representació explícita de l'absència de dades.

- f. Quina estructura de dades has utilitzat per representar les característiques de "P50 Pocket"? Explica si hi ha alternatives i per què has pres aquesta decisió.

He utilitzat un array anomenat "characteristics" per representar les característiques de "P50 Pocket". Aquesta estructura permet gestionar múltiples característiques de manera ordenada. Alternatives podrien ser l'ús d'un objecte amb propietats per cada característica, però s'ha optat per un array, ja que la informació sembla ser una llista sense associacions específiques amb noms.

- g. Si el JSON resultant no té el camp "items_count", creus que s'ha perdut informació? Creus que és útil tenir aquesta informació en un camp?

Si el JSON no té el camp "items_count", no s'ha perdut informació crítica, ja que aquesta informació pot ser inferida comptant els elements en l'array "devices". No obstant això, tenir el camp "items_count" pot ser útil per a optimitzacions i facilitar la recuperació ràpida del nombre total de dispositius sense haver de recórrer tot l'array. No és absolutament necessari, però pot ser convenient en certes situacions.

6. Proposa un pseudocodi d'una funció per obtenir les dades que es demanen partint del JSON que acabes de generar. Fes servir la capçalera que es suggereix i recorda que l'objectiu d'aquest exercici és trobar l'estructura òptima del JSON. Per aquesta raó, la majoria d'aquests exercicis s'han de resoldre accedint a les de manera quasi directa a les dades.

Considera que els índexs de les llistes comencen en 0. Tampoc no cal tenir en compte els possibles errors, com trobar-se llistes buides o elements inexistents.

A tall d'exemple, considera una funció que retorni el nom del pokémon. Una possible solució seria

```
fun getPokemonName(pokemon) {  
  
    return pokemon["name"]  
  
}
```

Un exemple diferent pot ser una funció que retorni el nom del primer moviment del primer pokémon d'una llista de pokémons. La solució:

```
fun getMovimentPrimerPokemon(pokemonsList) {  
  
    primerPokemon = pokemonsList[0]  
  
    primerMoviment = primerPokemon["moviments"][0]  
  
    return primerMoviment["nom"]  
  
}
```

//solució alternativa:

```
// return pokemonsList[0]["moviments"][0]["nom"]
```

- Implementa una funció que retorna la unitat de mesura l'altura del pokémon. Si el pokémon mesura 0.8 m, la funció ha de retornar "m". Recorda que no cal processar les dades, sinó que és millor tenir-les ben estructurades.

```
fun getUnitatMesuraAltura(pokemon) {  
  
    return pokemon["mesura"].substring(pokemon["mesura"].length - 1)  
  
}
```

- b. La funció ha de retornar un booleà que indiqui si el segon moviment de la llista de moviments del pokémon és de contacte o no.

```
fun isSegonMovimentDeContacte(pokemon) {  
  
    return pokemon["moviments"][1]["esDeContacte"]  
  
}
```

- c. Una funció que retorni la suma de totes les estadístiques del pokémon (velocitat, força, precisió, resistència)

```
fun getSumaEstadistiques(pokemon) {  
  
    return pokemon["estadistiques"]["velocitat"] +  
  
        pokemon["estadistiques"]["força"] +  
  
        pokemon["estadistiques"]["precisió"] +  
  
        pokemon["estadistiques"]["resistència"]  
  
}
```

- d. La funció ha de retornar la mitjana de totes les estadístiques del pokémon.

```
fun getMitjanaEstadistiques(pokemon) {  
  
    return getSumaEstadistiques(pokemon) / 4  
  
}
```

- e. Donada una llista de 3 pokémons, la funció ha de retornar la suma dels pesos d'aquests pokémons.

```
fun getPes(llista3pokemon) {  
  
    return llista3pokemon[0]["pes"] + llista3pokemon[1]["pes"] + llista3pokemon[2]["pes"]  
  
}
```

- f. Donat un pokémon i un nivell, la funció ha d'indicar si el nivell és igual o superior al nivell requerit per fer la primera evolució del pokémon.

Per exemple, si la funció rep nivell=4, i el nivell de la primera evolució és 3, la funció ha de retornar true.

```
fun isEvolucioPossible(pokemon, nivell) {  
    return nivell >= pokemon["evolucions"][0]["nivellRequerit"]  
}
```

- g. Donada una llista de pokémons, la funció ha de retornar el pokémon amb la potència més alta. Pots fer una funció auxiliar que calculi la potència d'un pokémon que és la suma de les potències dels seus moviments (opcional)

```
fun getPotenciaMesAlta (pokemonsList)
```