

3. Responde a las siguientes cuestiones sobre la traducción del ejercicio 1 (traducción JSON — XML).

a. ¿Cómo has manejado el valor null en el elemento age de Anna en la traducción a XML? ¿Es ésta la mejor manera de representar la falta de información? Propone todas las alternativas posibles.

- El valor null en la etiqueta “age” de Anna representa un valor vacío.
- Otras alternativas:
- <age/>
- <age></age>

b. Qué deberemos tener en cuenta cuando tenemos elementos repetidos como las mascotas o amigos? ¿Se ha mantenido la consistencia en la traducción?

Cuando hay elementos repetidos como mascotas y amigos, que serían representados como arrays en JSON, en XML son representados mediante atributos y elementos. Para realizar la traducción se ha de modificar la traducción.

4. Responde a las siguientes cuestiones sobre la traducción del ejercicio 2 (traducción XML-JSON).

a. Explica qué se ha convertido en objetos, y qué en arrays y por qué has tomado estas decisiones.

- Se realizó una conversión de arrays para las etiquetas “type” y características del P50 pocket y los objetos son las demás etiquetas como: name, Company, Price.

b. Explica qué has hecho para mantener junta la información del precio con el tipo de moneda pagada. ¿Cómo has transformado, en este caso, los atributos del XML a JSON y por qué?

- Se ha utilizado el objeto “Price”, que guarda la información de “amount”, “currency” y “discount”

c. ¿Existe alguna etiqueta en XML que no se ha traducido directamente a JSON? ¿Crees que esto significa que se ha perdido información?

- La etiqueta device que separa los elementos (items_count=4). Pues al realizar un array en JSON, los identifico claramente por “type”.

d. Cómo has gestionado los caracteres especiales como las comillas dobles en la traducción? ¿Cómo afecta esto a la legibilidad de JSON?

- Hay simbología que no se puede introducir en JSON y esta se lee tal cual como en XML. Por ende, tuve que investigar un poco sobre como introducir simbología en el JSON. Para representar las comillas dobles, estas deben ir acompañadas con un \.

e. Explica cómo has tratado los elementos sin información o con datos opcionales. ¿Has optado por dejar el campo vacío, por utilizar el valor null o por omitir el campo? Explica cuál crees que es la mejor decisión y por qué.

- En mi caso, yo utilice el valor null, porque visualmente se indica que es un valor que no tiene información ingresada.

f. Qué estructura de datos has utilizado para representar las características de "P50 Pocket" Explica si existen alternativas y por qué has tomado esta decisión.

- Ha sido representado como un array, ya que no tienen características específicas. Otra alternativa, sería crear un objeto características con información específica de cada característica.

g. Si JSON resultante no tiene el campo "items_count", crees que se ha perdido información? ¿Crees que es útil tener esa información en un campo?

- No pierde información como tal. El item_count determina el número de ítems que contendrá el documento. En el caso que el documento sea más largo, sería útil tener este dato, pero al ser solo de 4 datos, no es indispensable.

6.

a.

```
fun getUnitatMesuraAltura(pokemon) {  
    return pokemon["altura"].split()[1]  
}
```

b.

```
fun isSegonMovimentDeContacte(pokemon):  
    return pokemon["movimiento"][1]["contacto"]
```

c.

```
fun getSumaEstadistiques(pokemon):  
    estadisticas = pokemon["estadística"]  
    return suma(estadisticas.values())
```

d.

```
def getMitjanaEstadistiques(pokemon):  
    estadistica = pokemon["estadística"]  
    return suma(estadistica.values()) / len(estadistica)
```

e.

```
fun getPes(lista3pokemon):  
    pesos = [float(pokemon["peso"].split()[0]) for pokemon in  
lista3pokemon]  
    return suma(pesos)
```

f.

```
fun isEvolucioPossible(pokemon, nivell):  
    if len(pokemon["evoluciones"]) > 0:  
        nivell_evolucion = pokemon["evoluciones"][0]["nivel"]  
        return nivel >= nivel_evolucion  
    else:  
        return False
```

g.

```
def getPotenciaMesAlta(pokemonsList):  
    def calcularPotenciaTotal(pokemon):  
        potencia_movimientos = [calcularPotenciaMovimientos(movimientos)  
for movimientos in pokemon["movimientos"]]  
        return sum(potencia_movimientos)  
  
    return max(pokemonsList, key=calcularPotenciaTotal)
```