

# Pràctica 7: JSON (part 1)

## Objectius

La pràctica inicial amb JSON té com a objectiu introduir els fonaments essencials d'aquest format de dades, centrant-se en la comprensió de la seva sintaxi, la creació de documents senzills i la manipulació d'objectes i arrays.

D'altra banda, es practicarà la traducció de dades entre JSON i XML, destacant la correspondència entre les estructures d'ambdós formats. Mitjançant exercicis progressivament més complexos, es pretén que els estudiants s'habituen a treballar amb documents JSON amb múltiples nivells d'anidament i altres característiques avançades. La pràctica també fomenta la reflexió sobre les diferències entre JSON i XML, així com l'avaluació de la comprensió dels estudiants mitjançant tasques d'avaluació pràctiques.

## Lliuraments

Aquesta pràctica està formada per la Part 1 i la Part 2. El lliurament es farà en una sola entrega una vegada ambdues parts estiguin realitzades. Per aquesta raó, **de moment no cal lliurar aquesta pràctica**.

Els resultats d'aquesta part de la pràctica s'hauran d'entregar mitjançant GIT i hauran d'estar en el següent tipus de fitxers:

- Exercici 1: un fitxer .xml
- Exercici 2: un fitxer .json
- Exercici 3 i 4: un fitxer PDF

## 3. Respon les següents questions sobre la traducció de l'exercici 1 (traducció JSON -

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.

Decidimos utilizar <age/> para mostrar que la etiqueta existe pero no contiene ningún valor, también podríamos haber no colocado la etiqueta para no mostrar su existencia en ese «friend» o haber colocado un atributo en friend como age = null, aunque este último cambiaría la representación de datos.

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?

Al ser una lista que representa un mismo tipo de elemento decidimos colocar una etiqueta en común como «friend» y «pet» con la consistencia en plural, así definir cada pet y friend como independiente dentro de «pets» y «friends».

4. Respon les següents questions sobre la traducció de l'exercici 2 (traducció XML-JSON).

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

En principio los devices se convierte en un array de objetos json, los cuales cada uno representa un device con sus respectivas propiedades, price se convierte en un objeto que contiene «currency» y «value», characteristics se convierte en un array de strings que contiene cada característica tomamos la decisión de convertirlo en array porque tiene más de una característica.

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?

Price se vuelve un objeto con currency y value en la cual se guarda el atributo original currency y el valor que contenía en value.

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?

Devices, device, y characteristic son etiquetas que no se colocaron ya que en la estructura json se sobreentiende su existencia. Y no creo que se haya perdido información.

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

Usamos \ para especificar que seguía un carácter que no debía procesarse, no afecta a la legibilidad.

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.

Si estaba presente como una etiqueta decidimos usar null o un array vacío, en cambio si no existía la etiqueta no se representó.

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.

Las characteristics se representaron como un array de Strings cada uno representando cada characteristic, también se podía utilizar un String separado por comas.

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?

En el caso de json al usar un array se crean los indices de cada objeto dentro por tanto el items\_count no es necesario. Json naturalmente ya contiene un length para los arrays, no se pierde información.

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 se 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 dades 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:

```
function getMovimentPrimerPokemon(pokemonsList) {  
    primerPokemon = pokemonsList[0]  
    primerMoviment = primerPokemon["moviments"][0]  
    return primerMoviment["nom"]  
}  
  
//Solució alternativa  
//return pokemonsList[0]["moviments"][0]["nom"]
```

a. 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.

```
function getUnitatMesuraAltura(pokemon) {  
    mesura = pokemon["altura"]  
    if (mesura == 0.8) return "m"  
    else return ""  
}
```

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

```
function isSegonMovimentDeContacte(pokemon) {  
  if (pokemon["moviments"].length < 2) return false  
  else return pokemon["moviments"][1]["contacte"]  
}
```

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

```
function getSumaEstadistiques(pokemon) {  
  velocidad = pokemon["estadistiques"]["velocidad"]  
  fortaleza = pokemon["estadistiques"]["fortaleza"]  
  precision = pokemon["estadistiques"]["precision"]  
  resistencia = pokemon["estadistiques"]["resistencia"]  
  
  return velocidad + fortaleza + precision + resistencia  
}
```

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

```
function getMitjanaEstadistiques(pokemon) {  
  suma = getSumaEstadistiques(pokemon)  
  
  return suma / 4  
}
```

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

```
function getPes(llista3pokemon) {  
  suma = 0  
  for (i = 0; i < 3; i++) {  
    suma += llista3pokemon[i]["pes"]  
  }  
  return suma  
}
```

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.

```
function isEvolucioPossible(pokemon, nivel) {  
    return pokemon["evolucions"][0]["nivel"] <= nivel  
}
```

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)

```
function getPotencia(pokemon) {  
    suma = 0  
    for (i = 0; i < pokemon["moviments"].length; i++) {  
        suma += pokemon["moviments"][i]["potencia"]  
    }  
    return suma  
}  
  
function getPotenciaMesAlta(pokemonsList) {  
    mas_potente = null  
    mayor_potencia = 0  
    for (i = 0; i < pokemonsList; i++) {  
        pokemon = pokemonsList[i]  
        potencia = getPotencia(pokemon)  
  
        if (potencia > mayor_potencia) {  
            mas_potente = pokemon  
            mayor_potencia = potencia  
        }  
    }  
    return mas_potente  
}
```