3. Respon les següents qüestions 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.

   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?

a. Per manejar el valor null de l'element age de l'Anna, hem fet <age></age>, es a dir, l'hem deixat buit
b. Per als elements repetits, hem fet una subetiqueta per guardar les etiquetes. Per exemple friends, hem creat friend per guardar cada amic de manera individual dins de friends, i en el cas de pets, l'etiqueta pet.

4. Respon les següents qüestions 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.

   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?

   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?

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

   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 qué.

   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.

   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?

a. S'ha convertit en en arrays els devices , el preu i els tipus de device, i en list s'ha convertit les characteristics

b. S'ha transformat en un array per poder guardar els descomptes, moneda i preu

c. S'han perdut algunes etiquetes com ara, l'etiqueta "item_countt" que només ens mostra tots els ítems que hi ha, no és una pèrdua important, ja que no és un valor necessari i a més a més es pot calcular fàcilment

d. el que hem fet per a poder fer visibles les cometes entre altres caràcters especials, ha estat incloure "/" abans d'expressar les cometes dobles
d'aquesta forma l'arxiu mostra les cometes adequadament per a poder imprimir les cometes finals

e. Hem deixat el camp amb el valor null, per si en el futur s'ha de posar algun valor.

f. Hem posat una llista, perquè es pot posar qualsevol característica, i dependrà del dispositiu les característiques que vols posar.

g. En el nostre cas no hem conservat l'etiqueta, ja que és fàcilment calculable i no aporta informació útil

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 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"]
```

   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.

```
fun getUnitatMesuraAltura(pokemon)
```

   b. La funció ha de retornar un booleà que indiqui si el segon moviment de la

A.

```
fun getUnitatDeMesura(pokemon){
    pokemonAlcada = pokemon[4];
    alcadaStrip = pokemonAlcada.strip();
    String letter="";
    for(i=0; i<alcadaStrip.lenght; i++){
        if(Character.isLetter(alcadaStrip.charAt(i))){
            letter+=alcadaStrip.charAt(i);
        }
    }
    return letter;
}
```

B.

```
fun isSegonMovimentDeContacte(pokemon){
    moviments = pokemon[7];
    segonMoviment= moviments[1];
    contacte= segonMoviment[3];
    return contacte;
}
```

C.

```
fun getSumaEstadistiques(pokemon){
    pokemonEstadistiques = pokemon[5];
    int sumaEstadistiques =0;
    for(i=0; i<pokemonEstadistiques.lenght; i++){
        sumaEstadistiques = sumaEstadistiques + pokemonEstadistiques[i];
    }
    return sumaEstadistiques;
}
```

D.

```
fun getsSumaEstadistiques (pokemon){
    primerpokemon = pokemonList[0];
    pokemonEstadistiques = primerpokemon[5];
    int sumaEstadistiques = 0;
    for (i=0; i < pokemonEstadistiques.lenght;i++){
        sumaEstadistiques = sumaEstadistiques + pokemonEstadistiques[i];
    }
    Int resultat = sumaEstadistiques / pokemonEstadistiques.lenght;
    return resultat;
}
```

E.

```
fun getPes (llista3pokemons){
     primerpokemon = pokemonList[0];
     segonpokemon = pokemonList[1];
     tercerpokemon = pokemonList[2];
     int pesPrimerpokemon = primerpokemon[3];
     int pesSegonpokemon = segonpokemon[3];
     int pesTercerpokemon = tercerpokemon[3];
     int sumaPesos = pesPrimerpokemon + pesSegonpokemon + pesTercerpokemon;
     return sumaPesos;
}
```

F.

```
fun isEvolucioPossible(pokemon, nivell){
    pokemonEvolucio = pokemon[8];
    pokemonPrimeraEvolucio = pokemonEvolucio[0];
    nivellEvolucio= pokemonPrimeraEvolucio[2];
    if(nivell >= nivellEvolucio){
        return true;
    }
    else{
        return false;
    }
}
```

G.

```
fun getPotenciaMesAlta(pokemonList){
    String pokemon= "";
    for(int i=0; i< pokemonList.length; i++){
        int sumaMoviments=0;
        int sumaMax=0;
        pokemonActual = pokemonList[i];
        moviments = pokemonActual[7];
        for(int y=0; y< moviments.length; y++){
            movActual=moviments[i];
            potencia=movActual[2];
            sumaMoviments+=potencia;
        }
        if(pokemon.isEmpty()){
            pokemon = pokemonActual[i];
        }
        else if(sumaMoviments>sumaMax){
            pokemon="";
            pokemon = pokemonActual[i];
        }
    }
    return pokemon;
}
```