

Noms: **Miquel Martínez Casanovas** i **Angel Munera Escobar**

## 7.1

activitat 3:

**a.1** Hem posat l'etiqueta `<age>` i hem posat null i la hem tancat `</age>`

**b.1** Haurem de tenir en compte que estan tots a la mateixa altura i que haurem de inventar-nos un key al XML com "friend" pero que al JSON no hem posat per ahorrarnos caràcters. Sí, s'ha mantingut la consistència.

activitat 4:

**A.2** En objectes s'han convertit totes les etiquetes i arrays han estat totes les que tenen més informació que era del mateix pare com device que te nom, company, inches... i characteristics, que te dintre characteristic que serian els objectes.

**B.2** Ho hem posat tot junt es un objecte `{"currency": "value", "preu": "value"}`

**C.2** No s'ha perdut, tan sols que en JSON com hem d'intentar resumir el contingut es poden treure pero la estructura segueix sent la mateixa, com en characteristics que no fa falta posar characteristic al JSON

**D.2** Doncs hem posat en les cometes dobles "key": "value" "caracter amb cometes dobles". Afecta la llegibilitat i no podries pensar que son cometes dobles

**E.2** Hem optat per posar la key i enla value posar-hi null. Creiem que és la millor decisió perquè com no hi ha informació anulem el camp.

**F.2** Un objecte. No hi ha altre alternativa i hem pres aquesta decisió perquè és un valor normal i que no pot ser llista per ell sol

**G.2** Sí que s'hauria perdut informació porque la final és un atribut del root que serveix per tot al JSON.

## 7.2

activitat 6:

a. def obtenir\_unitat\_mesura(altura):

```
    if float(altura) == 0.8: return "m"
```

```
    else: return "El valor no es 0.8"
```

b. def isSegonMovimentDeContacte(pokemon):

```
    return pokemon["moviments"][1]["contacte"] == "Sí"
```

c. def suma\_estadistiques(pokemon):

```
    estadistiques = pokemon.get('estadístiques', {})
```

```
    suma = sum(estadistiques.values())
```

```
    return suma
```

d. def mitjana\_estadistiques(pokemon):

```
    estadistiques = pokemon.get('estadístiques', {})
```

```
    mitjana = sum(estadistiques.values()) / len(estadistiques)
```

```
    return mitjana
```

e. def suma\_pesos(pokemon):

```
    suma = 0
```

```
    for pokemon in lista_pokemon:
```

```
        pes = pokemon["mesures"]["pes"]
```

```
        pes_numero = float(pes.replace("kg", "").replace(",", ""))
```

```
        suma += pes_numero
```

```
    return suma
```

**f.** def nivellsuperiorigualprimeraevolucio(pokemon):

    evolucions = pokemon.get("evoluciones", [])

    if evolucions:

        primera\_evolucio = evolucions[0].get("pre-evolucio", {})

        nivel\_requerido = int(primera\_evolucio.get("nivell\_evolucio", 0))

        nivel\_pokemon = int(pokemon.get("nivell\_evolucio", 0))

        return nivel\_pokemon >= nivel\_requerit

    return False

**g.** def potencia\_pokemon(pokemon):

    moviments = pokemon.get('movimientos', [])

    potencia = sum(moviment.get('potencia', 0) for moviment in moviments)

    return potencia

def pokemon\_mes\_potent(pokemons):

    pokemon\_mes\_potent = None

    potencia\_maxima = 0

    for pokemon in pokemons:

        potencia = potencia\_pokemon(pokemon)

        if potencia > potencia\_maxima:

            pokemon\_mes\_potent = pokemon

            potencia\_maxima = potencia

    return pokemon\_mes\_potent

