CENTRE ESPECÍFIC
D´EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

UD 05.
PROGRAMACIÓN FUNCIONAL

Desarrollo Web en entorno cliente
CFGS DAW

Ejercicios

Álvaro Maceda Arranz
alvaro.maceda@ceedcv.es
2022/2023
Versión:221114.1316

Licencia

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

ÍNDICE DE CONTENIDO

UD05. Programación Funcional

Los ejercicios están pensados para que intentes aplicar las técnicas de programación funcional vistos en la unidad didáctica.

> Recuerda crear los tests para los ejercicios

1. Ejercicio 5: Validación de direcciones

Escribe un programa que filtre las direcciones de una lista para obtener únicamente las direcciones válidas. Para que una dirección sea válida debe cumplir las siguientes condiciones:

- Debe tener informados el país, la ciudad y la dirección
- Debe tener informado o el móvil o el fijo
- Debe incluir una región o un código postal (cp)

El filtrado se debe hacer en una única sentencia que ocupe una sola línea de código. Además, **no puedes utilizar `if` en ninguna parte del programa.**

Por ejemplo, con la siguiente lista de direcciones debería devolver únicamente la primera:

```
let direcciones = [
  {
    // Válido
    pais: 'España', region: '', cp: '46014',
    ciudad: 'Valencia', direccion: 'Carrer Misericòrdia, 34',
    complemento: '',
    movil: '', fijo: '961 20 69 90'
  },
  {
    // Inválido: no tiene movil o fijo
    pais: 'España', region: '', cp: '46960',
    ciudad: 'Aldaia', direccion: 'C/ Montcabrer, 22',
    complemento: 'Pol. Ind. La Lloma',
    movil: '', fijo: ''
  },
  {
    // Inválido: no tiene país
    pais: '', region: 'Alicante', cp: '',
    ciudad: 'Petrer', direccion: 'Los Pinos, 7',
    complemento: '',
    movil: '', fijo: '965 37 08 88'
  }
]
```

2. Ejercicio 6: Remap

Implementa una función `map(array, funcion)` que reciba un array y aplique la función a cada elemento, devolviendo un nuevo array.

No puedes usar bucles de ningún tipo ni las funciones `forEach` o `map` de los arrays

Por ejemplo:

```
map([1,2,3], x => x * 2) // [2,4,6]
```

3. Ejercicio 7: Evolución de Pokémon

Dada la cadena de evolución de un pokémon en formato JSON obtenida del PokéAPI (https://pokeapi.co/) obtén una cadena con la lista de los diferentes pokémon en los que evoluciona. Si un pokémon puede tener varias cadenas de evolución (por ejemplo, Eevee) devuelve sólo el contenido de la primera.

Por ejemplo, para Bulbasaur, se puede obtener la cadena de evolución de esta dirección: https://pokeapi.co/api/v2/evolution-chain/1. El programa debería devolver la cadena `"Bulbasaur-Ivysaur-Venusaur"` (ojo, que la primera letra está en mayúsculas)

Si quieres ver otras cadenas de evolución puedes cambiar el número final en la URL. Aquí tienes la documentación sobre la información de devuelve el endpoint:
https://pokeapi.co/docs/v2#evolution-section

> No es necesario que hagas la llamada para obtener los datos, sólo debes programar el proceso de la cadena. Ten en cuenta que para las pruebas no necesitarás la cadena entera sino sólo las partes relevantes para el ejercicio.

> Aunque este ejercicio te parezca complicado, se puede implementar con menos de 10 líneas de código. Usa la recursividad.

> Bonus: devuelve un array con todas las posibles cadenas de evolución

A continuación ofrecemos en orden alfabético el listado de autores que han hecho aportaciones a este documento:

- Álvaro Maceda Arranz