# Ejercicio combinado donde con el cual se realiza búsquedas parametrizadas.

// DataSet

const personajes = [{

        nombre: "Spectra",

        rol: "Francotirador",

        nivel: 13,

        salud: 889,

        dano: 70,

        velocidad: 3.4,

        habilidades: ["crítico", "cura", "dash"],

        faccion: "Sombra",

        activo: false

    },

    {

        nombre: "Titan",

        rol: "Soporte",

        nivel: 22,

        salud: 1760,

        dano: 165,

        velocidad: 3.1,

        habilidades: ["cura", "dash", "teletransporte"],

        faccion: "Centinelas",

        activo: false

    },

    {

        nombre: "Volt",

        rol: "Explorador",

        nivel: 33,

        salud: 3247,

        dano: 113,

        velocidad: 2.6,

        habilidades: ["misil", "regeneración", "barrera"],

        faccion: "Aegis",

        activo: false

    },

    {

        nombre: "Shadow",

        rol: "Francotirador",

        nivel: 38,

        salud: 2954,

        dano: 188,

        velocidad: 2.7,

        habilidades: ["escudo", "campo eléctrico", "invisibilidad", "barrera"],

        faccion: "Aegis",

        activo: false

    },

];

const predicados = {

    "===": (a, b) => (a === b) ? true : false,

    "!==": (a, b) => (a !== b) ? true : false,

    ">=": (a, b) => (a >= b) ? true : false,

    "<=": (a, b) => (a <= b) ? true : false,

    ">": (a, b) => (a > b) ? true : false,

    "<": (a, b) => (a < b) ? true : false,

    "and": (a, b) => Boolean(a && b),

    "or": (a, b) => Boolean(a || b),

    "include": (a, b) => Array.isArray(a) && a.includes(b)

}

// Reglas de busquedas...

const reglas = [{

        "operador": "===",

        "atributo": "rol",

        "valor": "Francotirador",

    },

    {

        "operador": "<=",

        "atributo": "nivel",

        "valor": "30",

    },

    {

        "operador": "include",

        "atributo": "habilidades",

        "valor": "cura"

    }

]

**const cumpleTodas = (personaje, reglas) => {**

**return reglas.every(**

**({ operador, atributo, valor }) => {**

**return predicados[operador](personaje[atributo], valor)**

**}**

**)**

**};**

**const resutado = personajes.filter(p => cumpleTodas(p, reglas));**

**console.log(resutado);**