

EXAMEN RECUPERACIÓN (UD1-UD3)

Los RA's que evaluaremos son:

RA1 – RA2 – RA3 – RA4

ARCHIVO ZOOLOGICO

En un zoológico informatizado, queremos llevar un registro digital de los **animales** y los **hábitats** en los que viven. Cada animal tiene características como su nombre, especie, número de patas y nivel de fuerza. Algunos animales son salvajes, con capacidad de ataque, y otros son domésticos, con distintos niveles de obediencia.

Tu objetivo es crear este sistema desde cero usando **Programación Orientada a Objetos**.

Ejercicio 1: Implementa la clase Animal

- Constructor: (nombre, especie, patas, nivelFuerza)
- nombre: String. Mínimo 3 caracteres.
- especie: String.
- patas: Número. No puede ser inferior a 2.
- nivelFuerza: Número. No puede superar 100.
- descripcion(): Devuelve toda la información del animal.
- Crea los **setters** y **getters** necesarios. Aplica validaciones en los **setters**.

Ejercicio 2: Implementa la clase AnimalSalvaje

- Hereda de Animal.
- Constructor: (nombre, especie, patas, nivelFuerza, tipoAtaque, nivelAgresividad)
- tipoAtaque: String.
- nivelAgresividad: Número. Debe ser menor que el nivelFuerza.
- descripcion(): Devuelve toda la información del animal.

Ejercicio 3: Implementa la clase AnimalDomestico

- Hereda de Animal.
- Constructor: (nombre, especie, patas, nivelFuerza, obediencia, esCariñoso)
- obediencia: Número. Debe ser menor que el nivelFuerza.
- esCariñoso: Booleano.
- descripcion(): Devuelve toda la información del animal.

Ejercicio 4: Implementa la clase Habitat

- Constructor: (nombre, ubicacion, nivelPeligro)
- nombre: String. Mínimo 4 caracteres. Si es menor, se añade "ZZZ".
- ubicacion: String.
- nivelPeligro: Número.

- resumen(): Devuelve la información del hábitat.

Ejercicio 5: Implementa la clase ArchivoZoologico

- Constructor sin parámetros.
- Atributos privados:
 - animales: Array de objetos tipo Animal.
 - habitats: Array de objetos tipo Habitat.
- Métodos:
 - añadirAnimal(animal): Añade el animal si no está repetido.
 - añadirHabitat(habitat): Añade el hábitat si no está repetido.
 - mostrarAnimales (): Devuelve los animales.
 - mostrarHabitats(): Devuelve la lista de hábitats.
 - resumen(): Devuelve el total de animales y hábitats.

Ejercicio 6: Creación y uso de objetos

- Crea al menos:
 - 2 animales (uno salvaje y uno doméstico).
 - 2 hábitats.
- Usa los métodos del ArchivoZoologico para añadirlos.
- Muestra en pantalla y consola:
 - El resumen del archivo.
 - La lista de animales.
 - Los hábitats registrados.

Ejercicio / Criterio	Puntos
Ejercicio 1: Clase Animal	1,5
- Constructor con validaciones correctas	0,5
- Setters y getters implementados correctamente	0,5
- Método descripcion() funcionando	0,5
Ejercicio 2: Clase AnimalSalvaje	1,5
- Herencia desde Animal aplicada correctamente	0,5
- Atributos y validaciones específicas	0,5
- Método descripcion() ampliado correctamente	0,5
Ejercicio 3: Clase AnimalDomestico	1,5
- Herencia desde Animal aplicada correctamente	0,5
- Atributos y validaciones específicas	0,5
- Método descripcion() ampliado correctamente	0,5
Ejercicio 4: Clase Habitat	1,5
- Validaciones sobre nombre, ubicación y nivelPeligro	0,5
- Método resumen() funcionando correctamente	0,5

Ejercicio / Criterio	Puntos
- Ajuste del nombre si es menor de 4 caracteres	0,5
Ejercicio 5: Clase ArchivoZoologico	2,0
- Métodos annadirAnimal() y annadirHabitat() correctos	0,5
- Métodos mostrarAnimales(), mostrarHabitats()	0,5
- Método resumen() funcional y claro	0,5
- Gestión de arrays y no duplicados	0,5
Ejercicio 6: Uso de objetos en script.js	1,0
- Creación de animales y hábitats	0,5
- Muestra en consola y pantalla según se indica	0,5
Calidad del código y buenas prácticas	0,5
- Claridad en la estructura	0,2
- Comentarios útiles y buena legibilidad	0,2
- Uso adecuado de ES6 y módulos	0,1

TOTAL	10