

# Examen de Programación en JavaScript (2º DAW)

**Duración:** 3 horas

**Modalidad:** Individual — sin conexión a internet

**Lenguaje:** JavaScript (ES6+)

**Puntuación total:** 10 puntos

---

## Contexto general: “Gestor de Biblioteca Digital”

Una pequeña biblioteca local quiere digitalizar su sistema de préstamos y gestión de libros mediante una aplicación en JavaScript que funcione desde consola o navegador (sin interfaz gráfica obligatoria).

Tu tarea consiste en **desarrollar un prototipo funcional del sistema**, aplicando de forma integrada los conceptos aprendidos durante el curso: estructuras de control, manipulación de strings, colecciones de datos (arrays, sets, maps), objetos y clases, manejo de errores, depuración y modularización con ES Modules.

El ejercicio no está dividido en apartados: tú decidirás la mejor forma de estructurar el código, aplicando los fundamentos de JavaScript de manera razonada, coherente y limpia.

---

## Objetivo general del programa

Diseña un programa que permita **gestionar un conjunto de libros y usuarios** de una biblioteca. El sistema debe permitir, al menos:

1. **Registrar libros** (título, autor, género, año y número de copias disponibles).
  2. **Registrar usuarios** (nombre, correo y número de socio).
  3. **Permitir préstamos y devoluciones de libros**, actualizando los datos correctamente.
  4. **Registrar el historial de préstamos y controlar duplicados o inconsistencias**.
  5. **Generar resúmenes y estadísticas** (por ejemplo, libros más prestados, usuarios con más actividad, géneros disponibles, etc.).
- 

## Recomendaciones finales

- Estructura el código con comentarios claros y significativos.
- Prioriza la claridad y la legibilidad antes que la complejidad innecesaria.
- Cada decisión técnica debe tener una justificación razonable.
- Se valorará la **creatividad**, la **autonomía** y la **capacidad de integrar múltiples conceptos en una solución coherente**.

## Criterios de evaluación (10 puntos totales)

Criterio	Descripción	
1. Lógica y control de flujo	Uso correcto y razonado de condicionales, bucles y validaciones.	1.5
2. Manipulación de cadenas y datos	Transformación, búsqueda y normalización de texto y colecciones.	1
3. Uso de estructuras (Array, Set, Map)	Correcta elección y manejo de las estructuras de datos.	1.5
4. Funciones y métodos funcionales	Diseño de funciones reutilizables y uso de métodos como map/filter/reduce.	1
5. POO en JavaScript	Creación y uso adecuado de clases, herencia, encapsulación y polimorfismo.	2
6. Gestión de errores	Manejo de excepciones, validaciones y robustez del código.	1
7. Depuración y diagnóstico	Uso de las herramientas de consola para seguimiento y optimización.	1
8. Modularización y organización del código	Uso correcto de import/export y estructura clara del proyecto.	1