

Clase 18:

Presentación del

Integrador

Datos Importantes

Datos Importantes a tener en cuenta

Plazo de Entrega

- Tienen 7 días a partir de hoy para hacer la entrega del trabajo
- El día límite de entrega es el **Sábado 05 de Julio, 14 hrs horario Argentino.**
- **NO hay defensa del trabajo.** Lo envían por mail y tendrán un feedback de mi parte, pero no tendrán que exponer.

Datos Importantes a tener en cuenta

Es grupal o Individual?

- El trabajo es de carácter GRUPAL, si se adopta esta forma de trabajo, basta **con una sola** integrante **comparta el link del repositorio de gitHub** donde se encuentra alojado el proyecto. Recuerden tener visible el repositorio es decir, que sea público.
- Si eliges trabajar de **forma individual**, se respeta el paso anterior, pero... el archivo sería **tuNombreCompleto.js**

Datos Importantes a tener en cuenta

Que pasa si no apruebo?

- En el caso de no aprobar / no presentar el trabajo, irán a una instancia de **recuperación oral**.
- La fecha se determinara luego de la entrega del feedback del trabajo.
- En el oral se tomaran los **temas vistos en el curso**.

Datos Importantes a tener en cuenta

Criterios de Evaluación

Son 10 puntos en total, cada uno equivale a 1 punto y según el desarrollo de cada uno, será su nota final.

Además se tendrá en cuenta:

- Correcta implementación de todas las funcionalidades requeridas.
- Uso apropiado de los conceptos y características de JavaScript vistos en el módulo.
- Calidad y legibilidad del código.
- Utilización correcta de nombres en variables (estándares de nomenclatura).
- Utilizar una correcta estructura de archivo como se muestra en la imagen

```
JS Apellido_Nombre.js X
JS Apellido_Nombre.js
1 // ESTRUCTURA DEL ARCHIVO:
2
3 //Sistema de Gestión de Biblioteca
4
5 // PUNTO 1. Estructura de Datos
6 // a) Crear un array de objetos llamado libros que contenga al menos 10 libros.
7
8 // CODIGO DEL PUNTO
9
10 // b) Crear un array de objetos llamado usuarios con al menos 5 usuarios.
11
12 // CODIGO DEL PUNTO
13
14 // PUNTO 2. Funciones de Gestión de Libros
15 // a) Implementar una función agregarLibro(id, titulo, autor, anio, genero) que
16
17 // CODIGO DEL PUNTO
18
19 // b) Crear una función buscarLibro(criterio, valor) que permita buscar libros p
20 // el algoritmo de búsqueda lineal.
21
22 // CODIGO DEL PUNTO
23
24 // c) Desarrollar una función ordenarLibros(criterio) que ordene los libros por
25 // de ordenamiento burbuja (bubble sort) y luego muestre los libros ordenados en
26
27 // CODIGO DEL PUNTO
28
29 // d) Desarrollar una función borrarLibro(id) que elimine el libro que se le pas
30
31 // CODIGO DEL PUNTO
```

El Integrador

El Integrador

Sistema de Gestión de Biblioteca

Introducción

El siguiente trabajo práctico integrador está diseñado para evaluar tus conocimientos en los fundamentos de JavaScript a través de la implementación de un conjunto de tareas que deben ser resueltas en un solo programa. Asegúrate de aplicar las buenas prácticas y estándares de nomenclatura, además de seguir las instrucciones dadas.

Objetivo

Desarrollar un sistema de gestión para una biblioteca que permita administrar libros y usuarios, aplicando los conceptos fundamentales de JavaScript vistos en el módulo.

El Integrador Punto por Punto

El Integrador

Punto 1: Estructura de Datos

a) Crear un array de objetos llamado **libros** que contenga al menos 10 libros. Cada libro debe tener las siguientes propiedades:

- ✓ id (número)
- ✓ título (string),
- ✓ autor (string),
- ✓ año (número),
- ✓ género (string),
- ✓ disponible (booleano).

b) Crear un array de objetos llamado **usuarios** con al menos 5 usuarios. Cada usuario debe tener:

- ✓ id (número)
- ✓ nombre (string)
- ✓ email (string)
- ✓ librosPrestados (array de ids de libros).

El Integrador

Punto 2: Funciones de Gestión de Libros

- a) Implementar una función `agregarLibro(id, titulo, autor, anio, genero)` que agregue un nuevo libro al array libros.
- b) Crear una función `buscarLibro(criterio, valor)` que permita buscar libros por título, autor o género utilizando el algoritmo de búsqueda lineal.
- c) Desarrollar una función `ordenarLibros(criterio)` que ordene los libros por título o año utilizando el algoritmo de ordenamiento burbuja (bubble sort) y luego muestre los libros ordenados en la consola.
- d) Desarrollar una función `borrarLibro(id)` que elimine el libro que se le pase por parámetro.

El Integrador

Punto 3: Gestión de Usuarios

- a) Implementar una función `registrarUsuario(nombre, email)` que agregue un nuevo usuario al array usuarios.
- b) Implementar una función `mostrarTodosLosUsuarios()` que me devuelva el array completo de usuarios
- c) Crear una función `buscarUsuario(email)` que devuelva la información de un usuario dado su email.
- d) Implementar una función `borrarUsuario(nombre, email)` que elimine el usuario seleccionado.

El Integrador

Punto 4: Sistema de Préstamos

- a) Desarrollar una función `prestarLibro(idLibro, idUsuario)` que marque un libro como no disponible y lo agregue a la lista de libros prestados del usuario.
- b) Implementar una función `devolverLibro(idLibro, idUsuario)` que marque un libro como disponible y lo elimine de la lista de libros prestados del usuario.

El Integrador

Punto 5: Sistema de Préstamos

- a) Crear una función `generarReporteLibros()` que utilice métodos avanzados de arrays (`.map()`, `.filter()`, `.reduce()`) para generar un reporte con la siguiente información:
- ✓ Cantidad total de libros.
 - ✓ Cantidad de libros prestados.
 - ✓ Cantidad de libros por género.
 - ✓ Libro más antiguo y más nuevo

El Integrador

Punto 6: Identificación Avanzada de libros

- a) Implementar una función `librosConPalabrasEnTitulo()` que identifique y muestre todos los libros cuyo título contiene más de una palabra (no títulos que contengan números ni otros caracteres). La función debe devolver un array con los títulos de esos libros y mostrarlo en la consola.



En este punto además de colocar los métodos, tendrán que pensar muy bien la forma de filtrar para que solo sean letras.

Piensen muy bien su lógica

El Integrador

Punto 7: Cálculos Estadísticos

- a) Desarrollar una función `calcularEstadisticas()` que utilice el objeto Math para calcular y mostrar:
- ✓ Promedio de años de publicación de los libros.
 - ✓ Año de publicación más frecuente.
 - ✓ Diferencia en años entre el libro más antiguo y el más nuevo.

El Integrador

Punto 8: Manejo de Cadenas

- a) Crear una función `normalizarDatos()` que utilice métodos de strings para:
- ✓ Convertir todos los títulos a mayúsculas.
 - ✓ Eliminar espacios en blanco al inicio y final de los nombres de autores.
 - ✓ Formatear los emails de los usuarios a minúsculas.

El Integrador

Punto 9: Interfaz de Usuario por Consola

- a) Implementar una función `menuPrincipal()` que muestre un menú de opciones al usuario y permita interactuar con el sistema utilizando prompt(). El menú debe incluir opciones para todas las funcionalidades anteriores y utilizar estructuras de control (if, switch, ciclos) para manejar la lógica.



ueden utilizar la estructura que gusten, sin embargo les recomiendo el uso de `switch` ya que hace al código más limpio y claro a la hora de escribirlo.

El Integrador

Punto 10: Comentando mi código

- a) Se tomará como último punto a evaluar la correcta utilización de comentarios explicando paso por paso su código.
- b) Deberán seccionar el código punto por punto y con una explicación corta y simple de que hicieron en cada punto.

Ejemplo:

//Punto 1:

// a). Consigna

// código y comentarios de lo que crean
necesario explicar

```
JS Apellido_Nombre.js X
JS Apellido_Nombre.js
1 // ESTRUCTURA DEL ARCHIVO:
2
3 //Sistema de Gestión de Biblioteca
4
5 // PUNTO 1. Estructura de Datos
6 // a) Crear un array de objetos llamado libros que contenga al menos 10 libros.
7
8 // CODIGO DEL PUNTO
9
10 // b) Crear un array de objetos llamado usuarios con al menos 5 usuarios.
11
12 // CODIGO DEL PUNTO
13
14 // PUNTO 2. Funciones de Gestión de Libros
15 // a) Implementar una función agregarLibro(id, titulo, autor, año, género) que
16
17 // CODIGO DEL PUNTO
18
19 // b) Crear una función buscarLibro(criterio, valor) que permita buscar libros p
20 // el algoritmo de búsqueda lineal.
21
22 // CODIGO DEL PUNTO
23
24 // c) Desarrollar una función ordenarLibros(criterio) que ordene los libros por
25 // de ordenamiento burbuja (bubble sort) y luego muestre los libros ordenados en
26
27 // CODIGO DEL PUNTO
28
29 // d) Desarrollar una función borrarLibro(id) que elimine el libro que se le pas
30
31 // CODIGO DEL PUNTO
```

Llegó el momento de poner a prueba lo aprendido!

Vamos a Codear!

