


# Examen Javascript DWEC

## Estructura de Carpetas:

```
examen_red_social/  
├── assets/  
│   └── datos.js  
├── Usuario.js  
├── Publicacion.js  
├── RedSocial.js  
├── main.js  
└── index.html
```

Red Social HLanz 



## Instrucciones:

Este examen tiene como objetivo evaluar tus habilidades en JavaScript ES6 para el módulo de DWEC. Deberás utilizar clases y funciones constructoras para crear una red social y resolver algunas consultas específicas. Asegúrate de seguir la estructura de carpetas proporcionada y colocar los archivos en los lugares correspondientes.

### Parte 1: Archivo Usuario.js (Función Constructora para Usuarios) (1,25p)

En el archivo `Usuario.js`, crea una función constructora llamada `Usuario` que represente un usuario de la red social. La función constructora debe incluir las siguientes propiedades:

- `username`: El nombre de usuario del usuario.
- `nombreCompleto`: El nombre completo del usuario.
- `amigos`: Una lista de amigos del usuario.
- `publicaciones`: Una lista de publicaciones realizadas por el usuario.

Métodos:

- `agregarAmigo(amigo)`: Permite agregar un amigo al usuario.
- `agregarPublicacion(mensaje, fecha)`: Permite agregar una `publicación` al usuario.

**Parte 2: Archivo Publicacion.js (Función Constructora para Publicaciones) (1,25p)**

En un nuevo archivo llamado **Publicacion.js**, crea una función constructora llamada **Publicacion** que represente una publicación. La función constructora debe incluir las siguientes propiedades:

- **mensaje**: El mensaje de la publicación.
- **fecha**: La fecha de la publicación.
- **likes**: El número de likes de la publicación. Por defecto será cero al crear la publicación.

Métodos:

- Además, en la clase **Publicacion**, crea un método llamado **darLike** que permita dar 1 "like" a una publicación. Dicho método dará un like más cada vez que se invoque.

**Parte 3: Archivo RedSocial.js (Clase para la Red Social) (6,5p)**

En el archivo **RedSocial.js**, crea una clase llamada **RedSocial** que permita administrar la red social. La clase debe tener las siguientes propiedades y métodos:

Propiedades:

- **usuarios**: Una lista de usuarios de la red social.

Métodos:

- **agregarUsuario(usuario)**: Permite agregar un usuario a la red social. (0,75p)
- **obtenerUsuarios()**: Retorna la lista de usuarios de la red social. (0,75p) Este Método tiene que **estar documentado** (0,5p)
- **obtenerPublicacionesUsuario(username)**: Busca y retorna un **MAP** con las publicaciones realizadas por un usuario específico. (1p)
- **encontrarPublicacionMasLikes()**: Busca y retorna la publicación con más likes. La función debe retornar un objeto que incluye el mensaje, la fecha y el número de likes de esa publicación. (1,25p)
- **encontrarUsuarioMasLikes()**: Determina el usuario que ha recibido más likes en total por todas sus publicaciones. La función debe retornar el nombre completo del usuario. (1,25p)
- **guardarEnLocalStorage()**: Guarda toda la información de la red social en el **localStorage** bajo la clave "ExamenRedSocial". (1p)

**Parte 4: Archivo main.js (Implementación de la Red Social) (1p)**

En el archivo **main.js**, implementa las clases y funciones constructoras definidas en **Usuario.js**, **Publicacion.js** y **RedSocial.js**. Además, realiza consultas y pruebas relacionadas con la red social. Organiza la lógica de tu red social en este archivo.

**Estructura de Carpetas:**

Asegúrate de organizar tu proyecto siguiendo la estructura de carpetas proporcionada.

**Archivos Necesarios:**

1. **Usuario.js**: Contiene la función constructora **Usuario**.
2. **Publicacion.js**: Contiene la función constructora **Publicacion**.
3. **RedSocial.js**: Contiene la clase **RedSocial**.
4. **main.js**: Implementa la lógica de la red social.
5. **assets/datos.js**: Contiene la variable **examenRedSocial** con los datos de la red social para que te sirvan de **apoyo o ayuda** y poder ver como debería de quedar nuestra estructura de **Usuarios** que ejemplifica la red social.

Asegúrate de que el archivo **index.html** incluye los scripts para cargar tus clases y lógica JavaScript.

**Nota:** Puedes realizar las consultas a la estructura de datos proporcionada en **assets/datos.js** para obtener las respuestas a las partes 1, 2 y 3 del examen si lo estimas oportuno (aunque no es necesario).

Además se deben de documentar **Todas** las cabeceras de los ficheros javascript que se entreguen.

### Entrega:

- Debes presentar los archivos de acuerdo a las instrucciones. Para ello se comprimirá todo nuestra carpeta de examen a excepción de la carpeta **node\_modules**. Dicho comprimido con vuestro nombre se adjuntará a la tarea de Moodle habilitada para el examen.
- Esta prueba evalúa los *Resultados de Aprendizaje 2,3,4* del módulo de DWEC, a través de la implementación correcta de la sintaxis de Javascript, usando estructuras de datos, funciones constructoras y clases.