



**UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE**  
**UNIDAD DE EDUCACIÓN A DISTANCIA**



**Asignatura:**

**Aplicación de Tecnologías Web**

**Ing.**

**Angel Geovanny Cudco Pomagualli**

**TEMA:**

**Gestión de Post y Comentarios en una red social**

**NOMBRES:**

**Carlos Ramiro Yáñez Yazán**

**Andres Alejandro Toledo Rojas**

**Luis David Flores Pillajo**

**NRC:**

**1382**

**Quito...17...enero de 2025**

## Gestion de Post y Comentarios en una Red Social

### Objetivo general

Desarrollar una aplicación web interactiva que permita gestionar publicaciones y comentarios en una red social ficticia, utilizando jQuery, funciones arrow de JavaScript y CSS para estilos. La actividad busca integrar conocimientos avanzados de manipulación del DOM, eventos y diseño modular en el desarrollo de aplicaciones web.

### Objetivos Específicos

- Desarrollar funcionalidades interactivas que permitan a los usuarios crear, editar y eliminar publicaciones y comentarios en tiempo real, utilizando jQuery para la manipulación dinámica del DOM y funciones arrow de JavaScript para una sintaxis más concisa y moderna.
- Diseñar una interfaz web responsiva y atractiva, empleando CSS avanzado para garantizar que la aplicación sea visualmente consistente y funcional en diferentes dispositivos y tamaños de pantalla.
- Implementar una estructura modular de código separando los archivos HTML, CSS y JavaScript, incluyendo módulos especializados para la gestión de publicaciones y comentarios, para fomentar la utilización del código

### Descripción y Comprobación de Funcionalidad del aplicativo

Para iniciar con la comprobación de las funcionalidades partiremos a través de los requisitos que se solicitan en la actividad.

### 1. Pagina General de la red social



The screenshot displays a web browser window with a dark green header bar containing navigation links for Gmail, YouTube, Maps, and Adobe Acrobat. The main content area has a light blue background with a dark green header section titled "NET GLOBAL RED". Below the header is a search bar labeled "Filtrar publicaciones...". A message "No hay publicaciones disponibles" is displayed below the search bar. Further down, there is a section titled "Crea tu nueva publicación" which contains two input fields: "Titulo" and "Descripción". Below these fields is a light blue button labeled "Agregar Post".

## 2. Pagina como sección principal del aplicativo web

```
index.html U X
index.html > html > body > div.container > header
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>NET GLOBAL RED</title>
7   <link rel="stylesheet" href="css/style.css">
8 </head>
9
10 <body>
11   <div class="container">
12     <header>
13       <h1>NET GLOBAL RED</h1>
14       <input type="text" id="filterKeyword" placeholder="Filtrar publicaciones..." />
15     </header>
16
17     <section id="postsContainer">
18       <!-- Los posts se cargarán aquí dinámicamente -->
19     </section>
20
21     <section id="newPost">
22       <h2>Crea tu nueva publicación</h2>
23       <input type="text" id="postTitle" placeholder="Título" />
24       <textarea id="postDescription" placeholder="Descripción"></textarea>
25       <button id="addPostButton">Agregar Post</button>
26     </section>
27   </div>
28
29   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
30   <script src="js/post.js"></script>
31   <script src="js/comments.js"></script>
32   <script src="js/main.js"></script>
33 </body>
34 </html>
```

Se estableció un índice como archivo principal definiendo el uso de estructuras js, jQuery para limitar las funcionalidades que se nos pide en la actividad.

## 3. Creación de un archivo main.js

Se establece como un archivo general un main el cual maneja las opciones para el sistema posts.js y comments.js

```
JS main.js U X
js > JS main.js > ready() callback > on(click) callback
1 // scripts.js
2 $(document).ready(() => {
3   // Event listener para agregar un post
4   $('#addPostButton').on('click', () => {
5     const title = $('#postTitle').val();
6     const description = $('#postDescription').val();
7     const date = new Date().toLocaleDateString(); // Fecha actual
8
9     // Validar que los campos no estén vacíos
10    if (title && description) {
11      addPost(title, description, date); // Llamar a la función para agregar un post
12      $('#postTitle').val(''); // Limpiar el campo de título
13      $('#postDescription').val(''); // Limpiar el campo de descripción
14    } else {
15      alert('Por favor, complete todos los campos');
16    }
17  });
18
19  // Event listener para filtrar posts por palabra clave
20  $('#filterKeyword').on('input', () => {
21    const keyword = $('#filterKeyword').val();
22    renderPosts(keyword); // Llamar a la función para renderizar posts filtrados
23  });
24
25  // Renderizar los posts al cargar la página
26  renderPosts();
27 });
```

#### 4. Creación de archivo Post para manejo de sistemas de (agregación eliminación y edición)

Para poder crear un post se creó un archivo netamente dedicado a la formulación como post.js donde se busca simplificar la interacción del usuario con la aplicación a través de eventos y manipulación dinámica del DOM además de Implementa buenas prácticas con validaciones básicas y actualizaciones inmediatas de la interfaz.

```
JS post.js U X
js > JS post.js > ...
1 // posts.js
2 let posts = []; // Array para almacenar los posts
3
4 // Función para agregar un post
5 const addPost = (title, description, date) => {
6   const newPost = { title, description, date, id: posts.length };
7   posts.push(newPost);
8   renderPosts(); // Vuelve a renderizar los posts
9 };
10
11 // Función para eliminar un post
12 const deletePost = (postId) => {
13   posts = posts.filter(post => post.id !== postId);
14   renderPosts(); // Vuelve a renderizar los posts
15 };
16
17 // Función para renderizar los posts
18 const renderPosts = (keyword = '') => {
19   $('#postsContainer').empty(); // Limpiar el contenedor de posts
20
21   // Filtrar los posts por la palabra clave
22   const filteredPosts = posts.filter(post => post.title.toLowerCase().includes(keyword.toLowerCase()));
23
24   // Si no hay posts, mostrar un mensaje
25   if (filteredPosts.length === 0) {
26     $('#postsContainer').html('<p>No hay publicaciones disponibles</p>');
27   }
28
29   // Renderizar los posts
30   filteredPosts.forEach(post => {
31     const postHtml =
32       <div class="post" data-id="${post.id}">
33       <h3>${post.title} <small>${post.date}</small></h3>
```

#### 5. Creación del archivo comments para (agregación eliminación y edición de comentarios).


Se establece un archivo comments.js para que los comentarios se almacenen en un array global llamado comments, que actúa como una base de datos en memoria para todos los comentarios de la aplicación.

```
JS comments.js U X
js > JS comments.js > ...
1 // comments.js
2 let comments = []; // Array para almacenar los comentarios
3
4 // Función para agregar un comentario
5 const addComment = (postId) => {
6   const commentText = $('#commentInput-${postId}`).val();
7   if (commentText) {
8     const newComment = { postId, text: commentText, id: comments.length };
9     comments.push(newComment);
10    renderComments(postId); // Vuelve a renderizar los comentarios para este post
11  } else {
12    alert('Por favor, ingrese un comentario');
13  }
14 };
15
16 // Función para eliminar un comentario
17 const deleteComment = (commentId) => {
18   comments = comments.filter(comment => comment.id !== commentId);
19   renderAllComments(); // Vuelve a renderizar todos los comentarios
20 };
21
22 // Función para editar un comentario
23 const editComment = (commentId, newText) => {
24   const comment = comments.find(comment => comment.id === commentId);
25   if (comment) {
26     comment.text = newText;
27     renderAllComments(); // Vuelve a renderizar todos los comentarios
28   }
29 };
30
31 // Función para renderizar los comentarios de un post
32 const renderComments = (postId) => {
33   const postComments = comments.filter(comment => comment.postId === postId);
```

Dentro de estos sistemas se uso sistemas de renderizado dinámico, manejo del Dom manipulación de sistemas de agregación eliminación o búsqueda clave tanto para el post y comments los cuales a través del sistema main tienen actividad esto se describe de manera mas explicita dentro de lo que es el archivo **README.md**.

Además, se utilizó un sistema de estilo css propio para este aplicativo:

## 6.Estilo /CSS

```
# style.css U X
css > # style.css > 
1 * {
2   margin: 0;
3   padding: 0;
4   box-sizing: border-box;
5 }
6
7
8
9 body {
10  font-family: Arial, sans-serif;
11  background-color: #f4f4f4;
12  background: #DBE6F6; /* fallback for old browsers */
13  background: -webkit-linear-gradient(to right, #C5796D, #DBE6F6); /* Chrome 10-25, Safari 5.1-6 */
14  background: linear-gradient(to right, #C5796D, #DBE6F6); /* W3C, IE 10+, Edge, Firefox 16+, Chrome 26+, Opera 12+,
15
16 }
17
18 .container {
19  width: 80%;
20  margin: 0 auto;
21  padding-top: 20px;
22 }
23
24 header {
25  text-align: center;
26 }
27
28 h1, h2 {
29  color: #333333
30 }
```

## Función de la Aplicación

### 1.Pantalla General

Como primer tendremos un vistazo general del aplicativo dentro del navegador el cual incluye las opciones para agregar un post donde se define un Título y una breve descripción.



### 2.Generacion de Posts

Se establecerá un título y una breve descripción de algunos posts de manera que se valide el ingreso de los mismos en el aplicativo dando clic en el botón **“Agregar Post”**.



Como se observa se pudo añadir el título y una breve descripción de las mismas teniendo también la opción de agregar comentarios como si fuera una red social abierta para los usuarios.

### 3. Generación de Comentarios.

Los usuarios pueden agregar comentarios y estos se pueden ir desplazando hacia abajo esto se realizará a través de la opción “Agregar Comentario”.



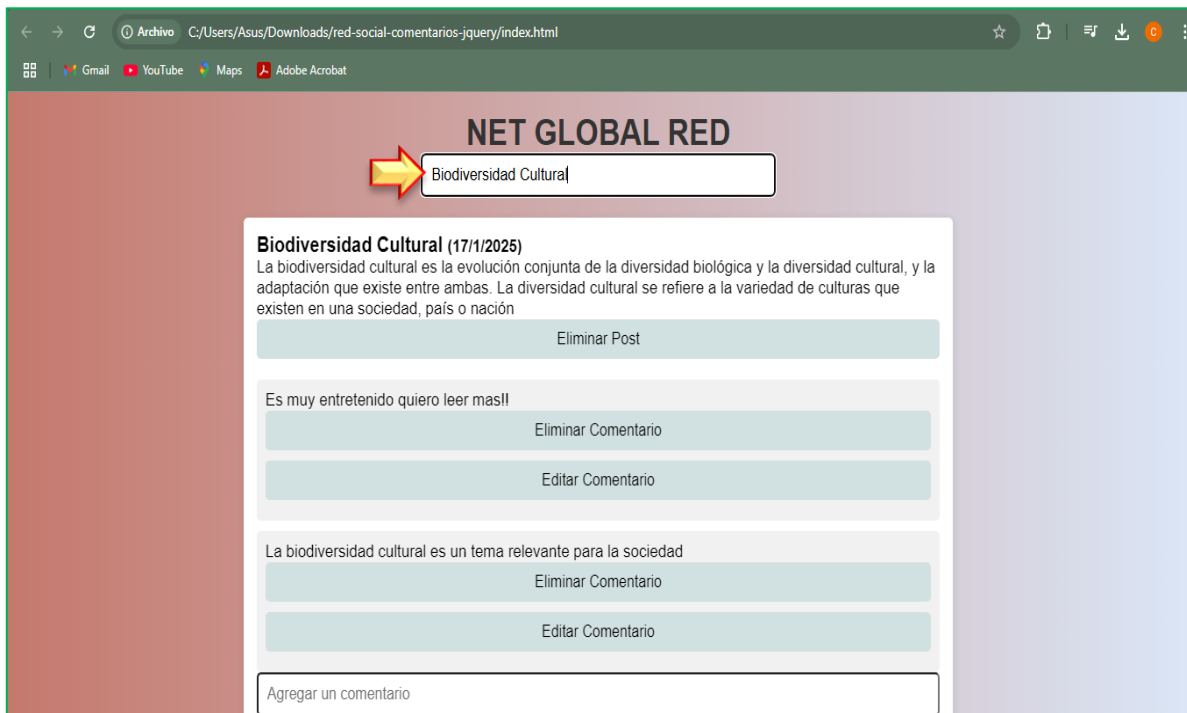
Ahora, hasta aquí se ha establecido las opciones para lo que es la creación de post y agregación de comentarios ahora dentro de la actividad se nos solicitaba una funcionalidad la cual **filtre las publicaciones a través de la palabra clave que viene a ser el título**, esto podemos describirlo de la siguiente manera.

#### 4. Filtrar publicaciones por palabra clave en el título

Para esto nos vamos a dirigir a la opción superior Filtrar Publicaciones donde a través del título de cada Publicación podremos filtrar el tema y encontrarlo como un sistema de búsqueda.



Dentro de esta opción añadiremos la palabra clave o título del post que estamos buscando abarcando un punto que se nos solicita en a la actividad.



Con esto hemos cumplido con las principales características que se nos solicita ahora también se solicita opciones de agregación, edición y eliminación de post y comentarios, estos se presentan de la siguiente manera.

#### 5. Agregación y Eliminación de Posts.

##### 5.1 Eliminación

Para este punto vamos a eliminar un Post y vamos a comprobarlo filtrando su búsqueda para validar que este ya no exista en la red social.



Una vez dando clic en **“Eliminar post”** filtramos su búsqueda para validar la opción de que este no existe en la red social desarrollada.



Se comprueba que dicho Post no existe, ahora para agregarlo es de la misma manera como explicamos con anterioridad damos en el botón agregar Post y dicha publicación volverá a aparecer.

## 5.2 Agregación

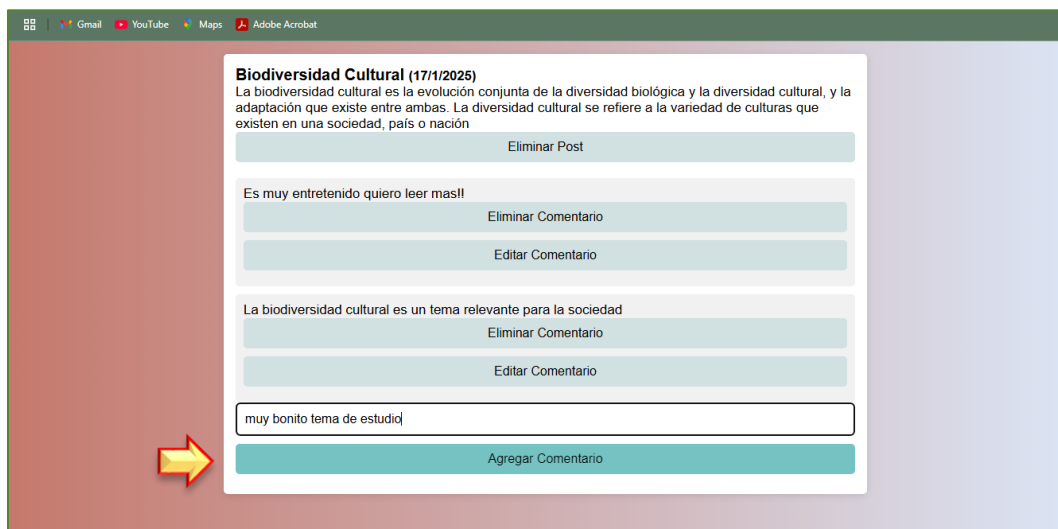




Ahora también se observa que por cada Posts publicado tenemos comentarios estos si pueden ser modificados de la siguiente manera.

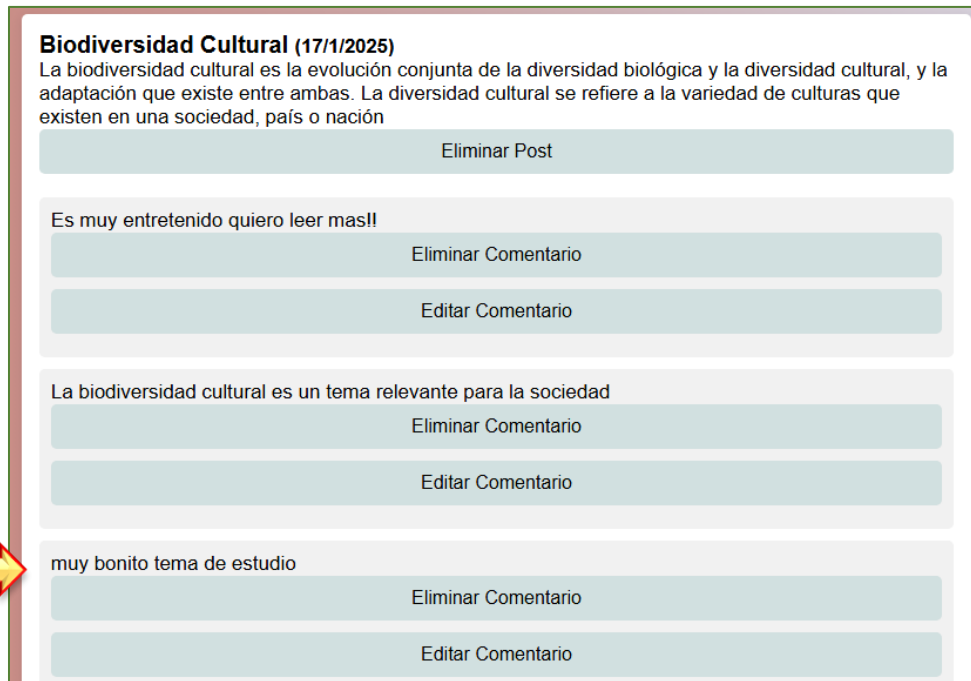
## 6. Agregación, Eliminación y Edición de Comentarios.

Para este caso vamos a utilizar un claro ejemplo para validar los 3 aspectos en este caso de la siguiente publicación.



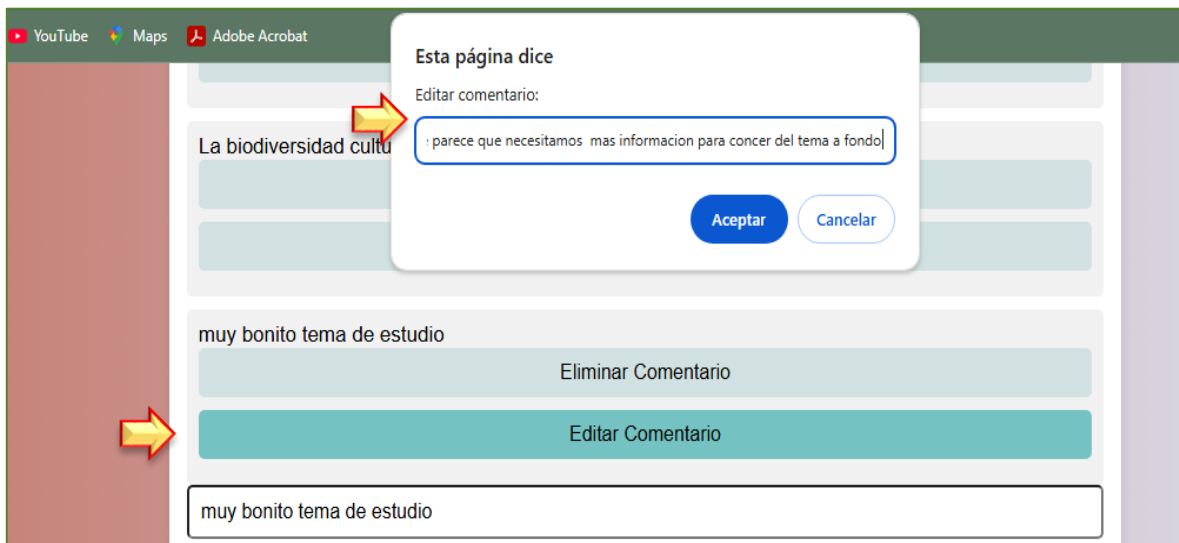
### 6.1Agregacion

Se da clic en la sección agregar comentario una vez que este ya este escrito dentro del post que se encuentra publicado de la siguiente manera.



## 6.2 Edición de Comentario

Se utiliza el mismo comentario para validar el respectivo cambio donde se dará clic en el botón editar comentario para dicha modificación dentro de la publicación

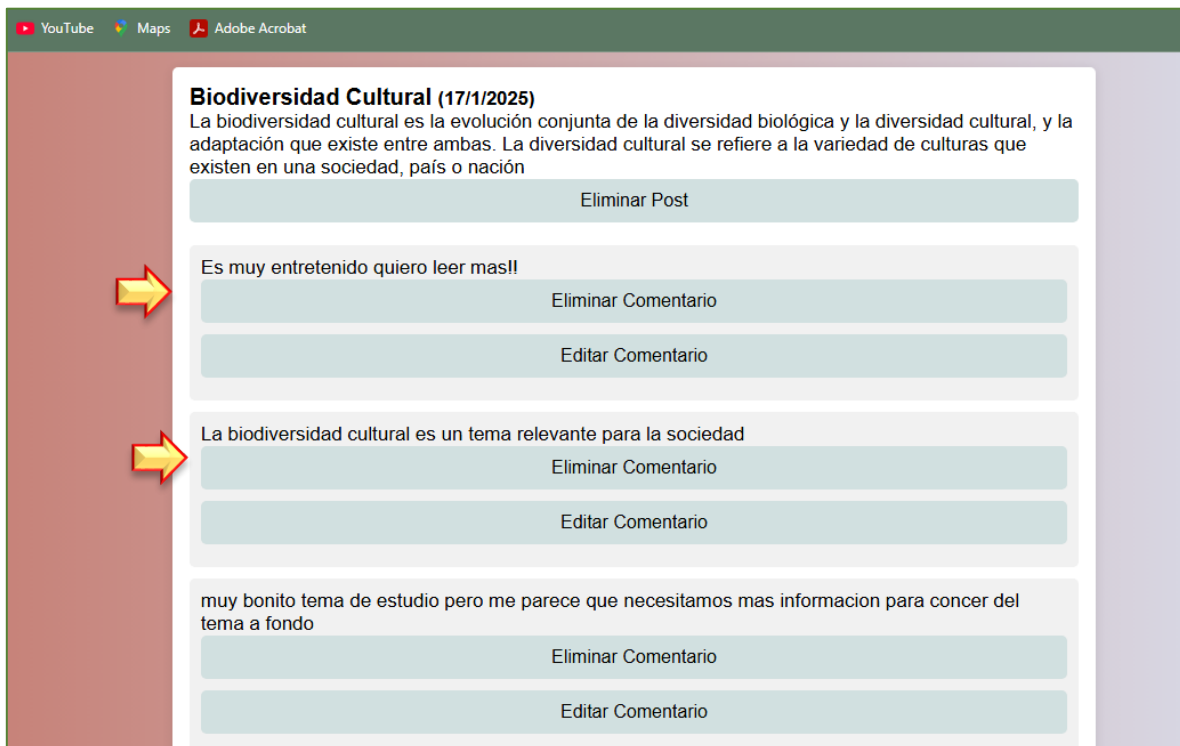


Se acepta la opción y el resultado se muestra de la siguiente manera denotando que dichos comentarios se deben haber actualizado en base a la edición que se realizó en los mismos.



### 6.3 Eliminación

Se hace uso de la opción **“Eliminar Comentario”** en este caso eliminaremos ambos comentarios para ver como se maneja la red social.



Una vez dado clic en la opción vamos a filtrar su búsqueda para ver si dichos comentarios han sido borrados de forma correcta.



Como se puede observar dichos comentarios fueron borrados y quedo solo uno cumpliendo con las ordenes que se nos piden dentro de la actividad tanto para los Posts y los respectivos comentarios denotando la funcionalidad de nuestra red social.

**(DENTRO DEL ARCHIVO README SE DESCRIBE LA FUNCIONALIDAD DE TODO EL CODIGO PROPUESTO)**

### Conclusiones

- La integración de jQuery y funciones arrow de JavaScript permite crear aplicaciones dinámicas y modernas que mejoran la experiencia del usuario al facilitar la interacción en tiempo real con publicaciones y comentarios.
- El diseño de una interfaz web responsiva con **CSS avanzado** garantiza la accesibilidad y funcionalidad en una amplia gama de dispositivos, contribuyendo a una experiencia de usuario consistente y profesional.
- La implementación de una estructura modular de código facilita el mantenimiento y la escalabilidad del proyecto

### Recomendaciones

- Validar las entradas de distintos datos para evitar sistemas incompletos o no deseados, asegurando la integridad de las publicaciones y comentarios.
- Optimizar el rendimiento de la aplicación al manejar grandes cantidades de datos dinámicos a través de JQUERY o de sistemas de actualización, eliminación y agregación.
- Documentar el aplicativo o red social con comentarios claros y concisos, lo que facilitará la comprobación de requerimientos por parte de la actividad a realizar.