1. Indica en alguna parte del sitio cuáles fueron los componentes AJAX que utilizaste. (Utiliza una arquitectura MVC.)
    - AJAX es utilizado en nuestro proyecto para eliminar a los usuarios creados (también lo implementaremos con los roles):

      https://github.com/M4urici02002/Dream_Crafters/tree/develop/Prototipo

      Una vez que se da clic en eliminar la tarjeta del usuario correspondiente desaparece de la interfaz sin necesidad de recargar la página.

    - Los componentes utilizados fueron los siguientes:

      Prototipo/controllers/gestionUsuarios.controller.js

```
exports.post_eliminar = (request, response, next) => {
   console.log("Username a eliminar:", request.body.username); // Añade esto para depuración
   Usuario.eliminar(request.body.username)
      .then(() => {
         return Usuario.fetchAll();
      }).then(([usuarios, fieldData]) => {
         return response.status(200).json({usuarios: usuarios});
      }).catch((error) => {console.log(error)})
};
```

      Prototipo/models/usuario.model.js

```
static eliminar(username) {
         return db.execute('DELETE FROM usuario WHERE username = ?', [username]);
   }
```

      Prototipo/routes/gestionUsuarios.routes.js

```
router.post('/gestionUsuarios/eliminar', isAuth, gestionUsuariosController.post_eliminar);
```

      Prototipo/views/gestionUsuarios.ejs

```
<button onclick="eliminar('<%= usuario.username %>')" type="button" class="btn text-tarjeta-usuario btn-dark btn-modificar-eliminar">

<script>
   const eliminar = (username) => {
      const respuesta = confirm("¿Deseas eliminar este registro de manera permanente?");
```

```
            if (respuesta) {
                console.log("Eliminando usuario:", username);
                const csrf = document.getElementById('_csrf').value;
                fetch('/gestionUsuarios/eliminar', {
                    method: 'POST',
                    headers: {
                        'Content-Type': 'application/json',
                        'csrf-token': csrf,
                    },
                    body: JSON.stringify({username: username})
                }).then((result) => {
                    if (result.ok) {
                        document.getElementById(`usuario-${username}`).remove();
                        document.getElementById('notification').innerHTML = `
                            <div class="alert alert-success" role="alert">
                                El usuario ${username} fue eliminado correctamente.
                            </div>
                        `;
                    } else {
                        alert('Hubo un error al intentar eliminar el usuario.');
                    }
                }).catch((error) => {
                    console.error('Error:', error);
                    alert('Error al eliminar el usuario.');
                });
            }
        }
    </script>
```

2. ¿Qué importancia tiene AJAX en el desarrollo de RIA's (Rich Internet Applications?
    - AJAX (Asynchronous JavaScript and XML) es una técnica fundamental en el desarrollo de Rich Internet Applications (RIAs) porque permite actualizar partes específicas de una página web sin necesidad de recargarla por completo. Esto proporciona una experiencia de usuario más fluida y dinámica, similar a las aplicaciones de escritorio. Con AJAX, los desarrolladores pueden enviar y recibir datos del servidor en segundo plano, lo que permite la carga de contenido de manera asincrónica, lo que mejora la velocidad y la interactividad de la aplicación web.

3. ¿Qué implicaciones de seguridad tiene AJAX? ¿Dónde se deben hacer las validaciones de seguridad, del lado del cliente o del lado del servidor?
    - Las implicaciones de seguridad de AJAX están relacionadas con la posibilidad de que los ataques como Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF) puedan explotarse de manera más fácil si no se implementan adecuadamente medidas de seguridad. En cuanto a las

validaciones de seguridad, es importante realizarlas tanto en el lado del cliente como en el lado del servidor:

- Validaciones del lado del cliente: Son **útiles** para proporcionar una respuesta instantánea al usuario y mejorar la experiencia de usuario, pero no son suficientes por sí solas para garantizar la seguridad, ya que pueden ser eludidas o manipuladas fácilmente por un atacante.
- Validaciones del lado del servidor: Son **esenciales** para garantizar la seguridad de la aplicación, ya que los datos enviados desde el cliente pueden ser manipulados antes de ser enviados al servidor. Las validaciones del lado del servidor deben implementarse para verificar la integridad y la legitimidad de los datos recibidos, evitando así posibles ataques.

4. ¿Qué es JSON?
    - JSON es un formato de intercambio de datos ligero y fácil de leer que se utiliza comúnmente en aplicaciones web para transmitir datos entre el cliente y el servidor. Es legible tanto por humanos como por máquinas, y se basa en la sintaxis de objetos de JavaScript. JSON se utiliza habitualmente en combinación con AJAX para el intercambio de datos asincrónico entre el cliente y el servidor. Es una alternativa más liviana y flexible a XML para el intercambio de datos estructurados.