

Desarrollo de un Componente de Login con QR

- Steven Cuello Vásquez - Arianna Espitia

Samir Banquez Humanez - Diana Humanez - Santiago Quintero

Desarrollo de un componente de inicio de sesión “Login con QR “: enfoque desde el producto

Introducción

El proyecto "Login con QR" surge como una respuesta a la problemática que enfrentan los usuarios en plataformas digitales, depender de credenciales como nombre de usuario y contraseña. Este método, aunque ampliamente utilizado, limita a ciertas poblaciones, especialmente a las personas con discapacidades visuales o usuarios propensos a olvidar sus contraseñas. Frente a esta necesidad, el proyecto busca ofrecer una solución innovadora, accesible y eficiente que simplifique el proceso de inicio de sesión mediante códigos QR. Este sistema no solo mejora la accesibilidad, sino que también optimiza tiempos de autenticación y refuerza la seguridad de los usuarios.

El desarrollo del componente se abordó en etapas a lo largo de los tres cursos de desarrollo de Software Educativo. Cada etapa corresponde a cada curso de software educativo: Desarrollo de Software educativo¹, 2 y 3. permitiendo construir de forma progresiva el sistema. Primeramente, se desarrolla la documentación base y pertinente del componente, luego, la implementación técnica del backend y, por último, el desarrollo visual y la experiencia de usuario en el frontend.

El componente Login con QR despliega su desarrollo y diseño acorde a su objetivo de mejorar la experiencia de usuario al ingresar a la plataforma CREAVI, principalmente para personas con discapacidades visual o que olvidan sus contraseñas.

Los requisitos funcionales del componente permiten lograr un inicio de sesión rápido y seguro en CREAVI mediante el escaneo de códigos QR, asegurando una experiencia de usuario sin obstáculos.

Producto final:

El sistema de login con QR para la plataforma CREAVI es un componente de autenticación diseñado para mejorar la seguridad y accesibilidad. El backend genera y valida códigos QR únicos y cifrados, verificando cada acceso en tiempo real y registrándolo en la base de datos para garantizar trazabilidad y seguridad. El frontend ofrece una interfaz intuitiva y accesible, compatible con lectores de pantalla y con opciones de exportación de códigos en PDF e imagen. Integrado de forma nativa en CREAVI, el sistema proporciona una experiencia de usuario fluida, optimizando el acceso a los recursos educativos con un enfoque en inclusión y protección de datos.

Requisito Funcional	Descripción
Iniciar sesión	Los usuarios ingresan sus credenciales y el sistema verifica su autenticidad para acceder a la plataforma.
Escanear Código QR	El usuario escanea su código QR y la plataforma lo autentica.
Generar código QR	La plataforma genera códigos QR únicos para cada usuario registrado.
Registros de inicio de sesión	Se registra la hora de ingreso de cada usuario.
Generar nuevo código QR (recuperar)	Los usuarios pueden generar un nuevo código QR si olvidan sus credenciales o tienen problemas para escanearlo.
Exportar el Código QR generado	Los usuarios pueden exportar o descargar su código QR en diferentes formatos de imagen.

Las restricciones del sistema incluyen el acceso limitado por roles específicos de la plataforma CREAVI, la necesidad de registro y aprobación previa por parte del administrador, configuraciones personalizadas como horarios de acceso específicos, requisitos técnicos que implican la necesidad de dispositivos con cámaras y políticas de seguridad como la caducidad de códigos QR o cambios regulares de contraseñas.

Figura 1: primera línea de código, Login Qr

```

91
92     if (fileInput.files.length === 0) {
93         alert('Por favor, selecciona un archivo.');
```

```

94         return;
95     }
96
97     const file = fileInput.files[0];
98     const formData = new FormData();
99     formData.append('file', file);
100
101     fetch('/upload_qr', {
102         method: 'POST',
103         body: formData
104     })
105     .then(response => response.json())
106     .then(data => {
107         if (data.success) {
108             resultado.innerHTML = '<div class="alert alert-success" role="alert">Inicio de sesión exitoso</div>';
109         } else {
110             resultado.innerHTML = '<div class="alert alert-danger" role="alert">Credenciales inválidas</div>';
111         }
112     })
113     .catch(error => console.error('Error:', error));
114 });
115 </script>
```

Este código implementa un sistema de inicio de sesión basado en la carga de un archivo, como un código QR. Primero valida que se haya seleccionado un archivo; si no, muestra una alerta. Luego, envía el archivo al servidor mediante una solicitud POST al endpoint

/upload_qr usando un objeto FormData. La respuesta del servidor se procesa en formato JSON: si el campo success es true, muestra un mensaje de éxito; de lo contrario, indica credenciales inválidas. Además, gestiona posibles errores y los registra en la consola.

Figura 2: segunda línea de código, Login Qr

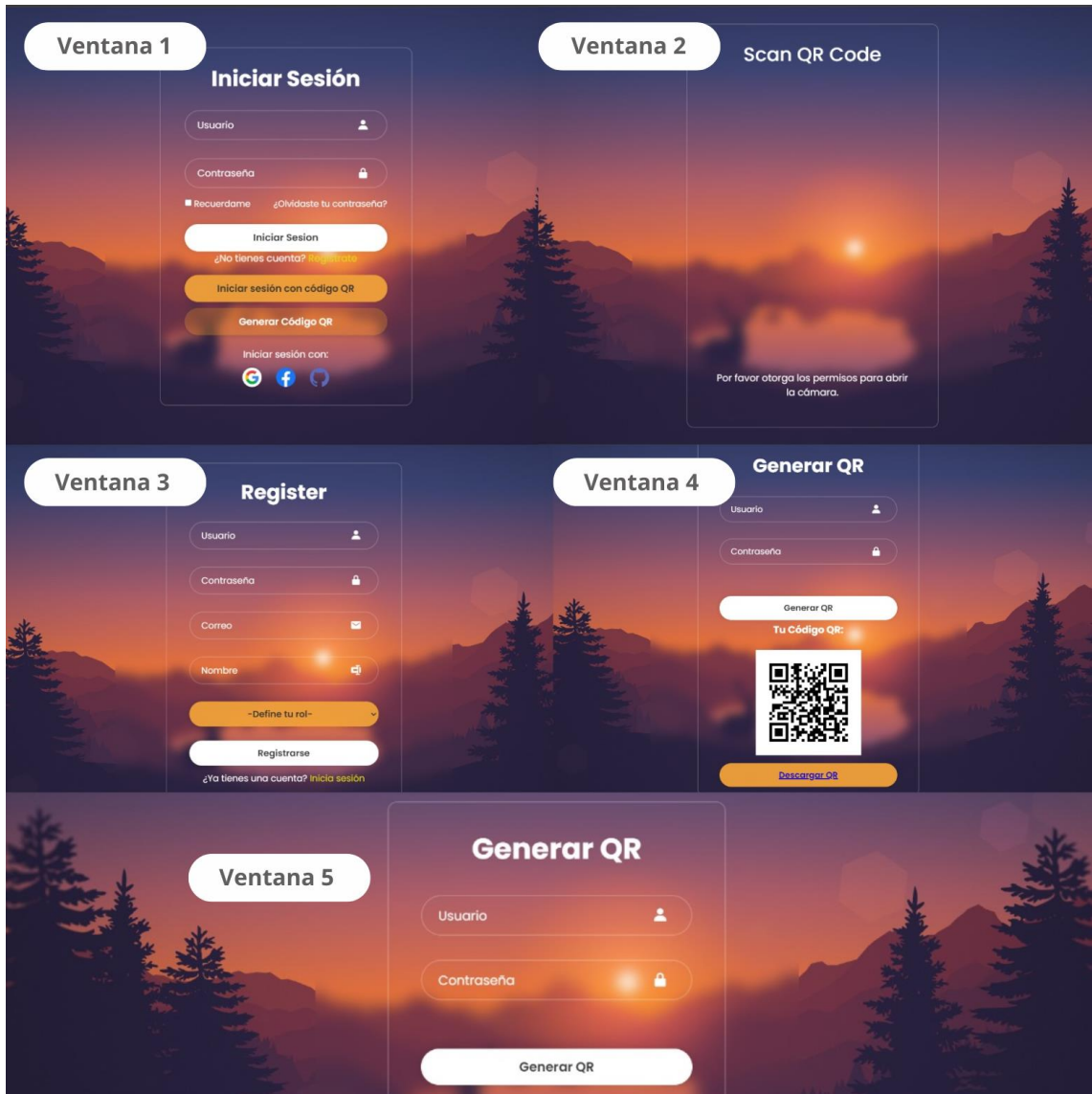
```
61         },
62         body: JSON.stringify({ encrypted_data: code.data })
63     })
64     .then(response => response.json())
65     .then(data => {
66         if (data.success) {
67             resultado.innerHTML = '<div class="alert alert-success" role="alert">Inicio de sesión exitoso</div>';
68         } else {
69             resultado.innerHTML = '<div class="alert alert-danger" role="alert">Credenciales inválidas</div>';
70         }
71     })
72     .catch(error => console.error('Error:', error));
73     } else {
74         requestAnimationFrame(scanQR);
75     }
76     } else {
77         requestAnimationFrame(scanQR);
78     }
79     }
80
81     scanQR();
82 })
83 .catch(function(err) {
84     console.error('Error al acceder a la cámara: ', err);
85 });
86
87
88 document.getElementById('subirArchivo').addEventListener('click', function() {
89     const fileInput = document.getElementById('fileInput');
90     const resultado = document.getElementById('resultado');
91
```

Este código implementa un sistema de inicio de sesión basado en el escaneo de códigos QR. Utiliza la cámara para capturar el código QR y extraer sus datos. Luego, envía estos datos en formato JSON a un servidor mediante una solicitud POST. Si el servidor responde con éxito (success: true), muestra un mensaje indicando un inicio de sesión exitoso; en caso contrario, indica credenciales inválidas. Además, el sistema reintenta continuamente el escaneo mediante requestAnimationFrame si no se detecta un QR válido, y gestiona errores relacionados con el acceso a la cámara o la solicitud al servidor.

Figura 3: Tercera línea de código, Login Qr

```
29 <script>
30 document.getElementById('escanear').addEventListener('click', function() {
31     const video = document.getElementById('video');
32     const resultado = document.getElementById('resultado');
33
34     navigator.mediaDevices.getUserMedia({ video: { facingMode: 'environment' } })
35     .then(function(stream) {
36         video.srcObject = stream;
37         video.play();
38
39         const canvas = document.createElement('canvas');
40         const context = canvas.getContext('2d');
41
42         function scanQR() {
43             if (video.readyState === video.HAVE_ENOUGH_DATA) {
44                 canvas.width = video.videoWidth;
45                 canvas.height = video.videoHeight;
46                 context.drawImage(video, 0, 0, canvas.width, canvas.height);
47
48                 const imageData = context.getImageData(0, 0, canvas.width, canvas.height);
49                 const code = jsQR(imageData.data, imageData.width, imageData.height);
50
51                 if (code) {
52                     resultado.innerText = `Resultado: ${code.data}`;
53                     // Detener el escaneo después de encontrar un código QR
54                     stream.getTracks().forEach(track => track.stop());
55
56                     // Enviar los datos del QR al servidor para autenticar al usuario
57                     fetch('/authenticate_qr', {
58                         method: 'POST',
59                         headers: {
60                             'content-type': 'application/json'
61                         }
62                     })
63                     .then(response => response.json())
64                     .then(data => {
65                         if (data.success) {
66                             resultado.innerHTML = '<div class="alert alert-success" role="alert">Inicio de sesión exitoso</div>';
67                         } else {
68                             resultado.innerHTML = '<div class="alert alert-danger" role="alert">Credenciales inválidas</div>';
69                         }
70                     })
71                     .catch(error => console.error('Error:', error));
72                 } else {
73                     requestAnimationFrame(scanQR);
74                 }
75             }
76         }
77         scanQR();
78     })
79     .catch(function(err) {
80         console.error('Error al acceder a la cámara: ', err);
81     });
82 }
83
84 document.getElementById('subirArchivo').addEventListener('click', function() {
85     const fileInput = document.getElementById('fileInput');
86     const resultado = document.getElementById('resultado');
87
```

El código implementa un escáner de código QR para un sistema de autenticación. Al hacer clic en un botón, se accede a la cámara del dispositivo (usando la API `getUserMedia`) para mostrar la transmisión en un elemento `<video>`. Un `<canvas>` captura un cuadro de la transmisión y, con la librería `jsQR`, se analiza y decodifica el código QR. Si se detecta un código válido, se muestra el resultado en pantalla, se detiene la cámara, y se envía el dato al servidor mediante una solicitud HTTP POST para autenticar al usuario.



El componente de Login con QR está compuesto por cinco ventanas:

Ventana 1: Iniciar Sesión

Es la pantalla principal de acceso. Ofrece campos para ingresar usuario y contraseña, con opciones adicionales como recordarme y recuperación de contraseña. Además, permite iniciar sesión mediante redes sociales (Google y Facebook) o utilizando un código QR.

Ventana 2: Escanear Código QR

Al seleccionar la opción de iniciar sesión con QR, esta ventana solicita permisos para acceder a la cámara del dispositivo y escanear el código QR generado previamente.

Ventana 3: Registro

Esta ventana permite crear una cuenta nueva ingresando información como usuario, contraseña, correo electrónico y nombre. También incluye un selector de roles para definir el perfil del usuario antes de completar el registro.

Ventana 4: Generar Código QR

Ofrece la funcionalidad de generar un código QR personalizado para iniciar sesión en futuras ocasiones. Requiere que el usuario ingrese sus credenciales y, una vez autenticado, genera el código QR correspondiente, con una opción para descargarlo.

Ventana 5: Generar QR Simplificado

Similar a la ventana anterior, pero con un enfoque más sencillo, esta ventana solicita usuario y contraseña para generar el código QR de acceso, facilitando una experiencia rápida y eficiente.