Recuerden poner a grabar la clase

Clase 16

Librerias de JavaScript

Diplomatura UNTREF

Temario

UNTREF
UNIVERSIDAD NACIONAL DE TRES DE FEBRERO

Librerias en JavaSCript:

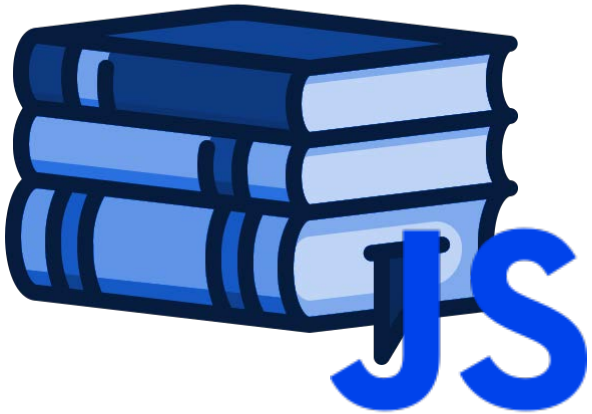
- qué son las librerías
- concepto de minificación de archivos
- Sweet Alert (simple - con múltiples botones - el método then())
- La librería qrcode (abrir la cámara fotográfica - leer un qr - decodificar un qr)

Que son las librerías

Librerías

En **JavaScript, una "biblioteca" (o "librería")** se refiere a un conjunto de funciones, métodos y objetos predefinidos y reutilizables que se crean para resolver tareas comunes y facilitar el desarrollo de aplicaciones web.

Estas **bibliotecas** se desarrollan para ayudar a los programadores a escribir código de manera más eficiente y reducir la necesidad de reinventar la rueda cada vez que se aborda una tarea común.

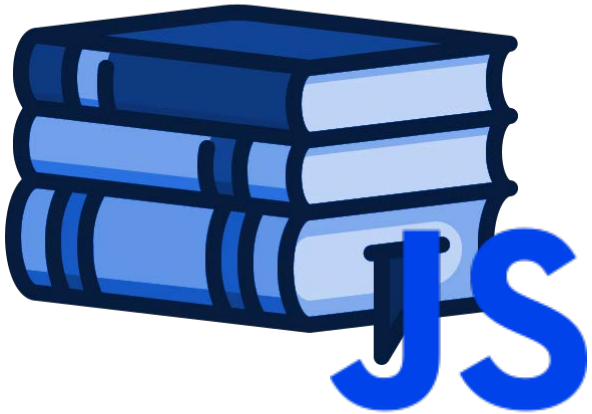


Librerías

Algunas caracteristicas de las librerias son:

Reutilización de Código: Una biblioteca proporciona una colección de funciones y utilidades que se pueden usar en diferentes proyectos. Esto fomenta la reutilización de código y ahorra tiempo y esfuerzo en el desarrollo.

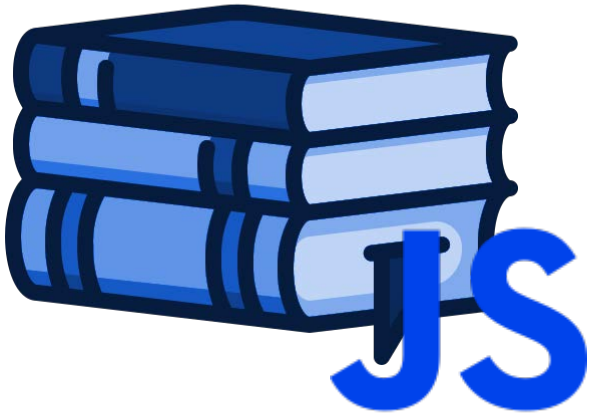
Resolución de Tareas Comunes: Las bibliotecas se crean para resolver tareas comunes, como manipulación del DOM, manejo de eventos, manipulación de fechas, animaciones, llamadas a API, gráficos, validación de formularios y mucho más.



Librerías

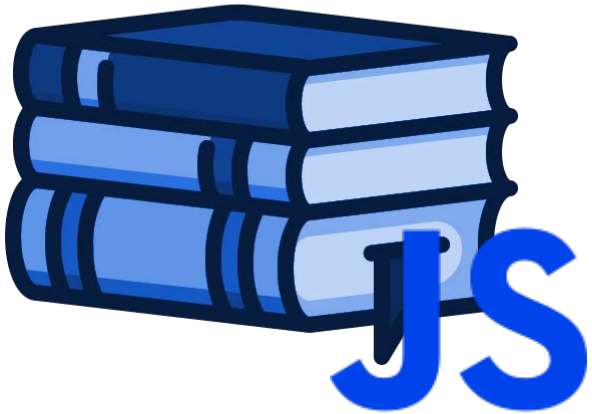
Documentación: Las bibliotecas suelen estar bien documentadas, lo que facilita su uso y comprensión. Los desarrolladores pueden consultar la documentación para aprender cómo utilizar las funciones y métodos proporcionados por la biblioteca.

Comunidad y Mantenimiento: Las bibliotecas populares a menudo tienen una comunidad activa de desarrolladores que contribuyen con mejoras y correcciones de errores. Esto asegura que la biblioteca se mantenga actualizada y sea compatible con las últimas versiones de JavaScript y los navegadores web.



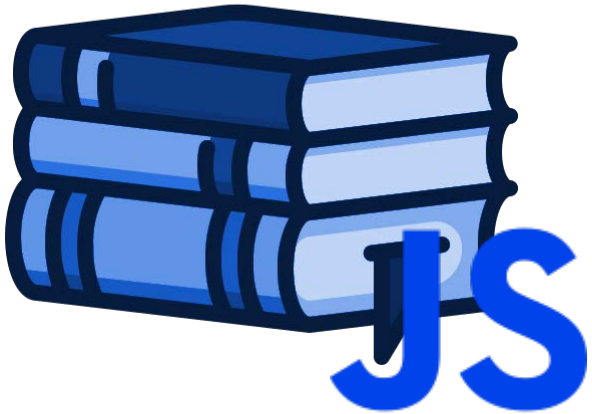
Librerías

Ejemplos y Ejemplos de Uso: Muchas bibliotecas también ofrecen ejemplos de uso y demos que ayudan a los desarrolladores a comprender cómo aplicar la biblioteca en situaciones del mundo real.



Librerías

Ejemplos y Ejemplos de Uso: Muchas bibliotecas también ofrecen ejemplos de uso y demos que ayudan a los desarrolladores a comprender cómo aplicar la biblioteca en situaciones del mundo real.



Minificación de archivos

Minificación de archivos

La minificación de archivos es un proceso común en el desarrollo web que implica reducir el tamaño de los archivos de código fuente, como HTML, CSS y JavaScript, eliminando caracteres y estructuras innecesarios sin afectar la funcionalidad del código.

El objetivo principal de la minificación es mejorar el rendimiento y la eficiencia de una aplicación web al reducir el tiempo de carga de la página y el ancho de banda necesario para transferir archivos.

Minificación de archivos: aspectos claves

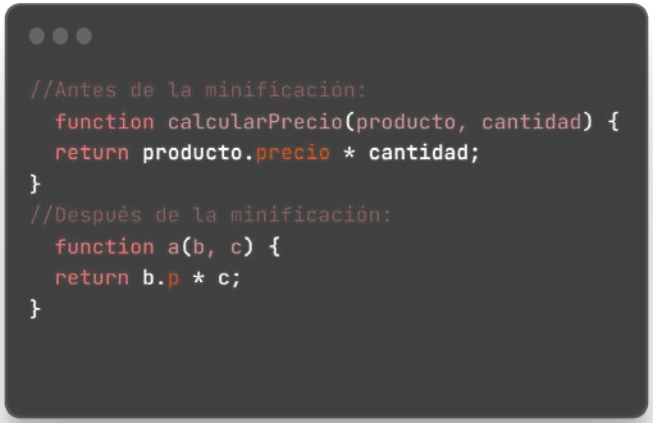
UNTREF
UNIVERSIDAD NACIONAL DE TRES DE FEBRERO

Eliminación de Espacios en Blanco y Comentarios:

Uno de los pasos más simples en la minificación es eliminar espacios en blanco, saltos de línea y comentarios del código fuente. Estos elementos son útiles para los desarrolladores durante la fase de desarrollo, pero no son necesarios en la versión final de una aplicación web.

Renombrado de Variables y Funciones:

En JavaScript, una técnica común de minificación implica renombrar variables y funciones a nombres más cortos y crípticos, lo que reduce la longitud del código fuente. Esto se hace sin cambiar el comportamiento del código, ya que los navegadores no se preocupan por los nombres de las variables.



Minificación de archivos: aspectos claves

UNTREF
UNIVERSIDAD NACIONAL DE TRES DE FEBRERO

Optimización de Código:

Además de reducir el tamaño del código, la minificación también puede incluir optimizaciones para hacer que el código sea más eficiente. Por ejemplo, se pueden reemplazar expresiones costosas por versiones más eficientes o eliminar código inaccesible.

Compresión de Recursos:

Después de la minificación, los archivos se pueden comprimir aún más utilizando algoritmos de compresión, como Gzip o Brotli, para reducir aún más el tamaño del archivo durante la transferencia a través de Internet.

Minificación de archivos: beneficios

UNTREF
UNIVERSIDAD NACIONAL DE TRES DE FEBRERO

Mejora el rendimiento:

Los archivos más pequeños se cargan más rápido en el navegador, lo que reduce los tiempos de carga de la página.

Ahorra ancho de banda:

La transferencia de archivos más pequeños ahorra ancho de banda tanto para el servidor como para los usuarios.

Mejora el SEO:

Un sitio web más rápido y eficiente puede mejorar la clasificación en los motores de búsqueda.

Experiencia del usuario:

Las páginas que se cargan más rápido brindan una mejor experiencia al usuario.

SweetAlert

SweetAlert

Es una popular biblioteca de JavaScript que se utiliza para crear ventanas modales de diálogo y notificaciones personalizadas en sitios web y aplicaciones web.

Esta biblioteca es ampliamente utilizada porque proporciona una forma sencilla y elegante de mejorar la experiencia del usuario al mostrar mensajes de alerta, confirmación o información de manera más atractiva y amigable que las ventanas emergentes del navegador estándar.

https://sweetalert2.github.io/

https://sweetalert.js.org/

SweetAlert: ventajas

Diseños Atractivos y Personalizables: SweetAlert ofrece una variedad de estilos y diseños predefinidos para ventanas modales, y también es altamente personalizable a través de CSS para que puedas adaptarlas al diseño de tu sitio web.

Fácil de Usar: La sintaxis de SweetAlert es simple y amigable para los desarrolladores. Puedes crear ventanas modales con solo unas pocas líneas de código.

SweetAlert: ventajas

Mensajes Personalizables: Puedes personalizar fácilmente el contenido de las ventanas modales, incluyendo el título, el texto del mensaje y los botones. Esto te permite crear mensajes de alerta, confirmación o información específicos para tu aplicación.

Integración con Promesas: SweetAlert se integra bien con las promesas de JavaScript, lo que facilita la ejecución de acciones después de que el usuario interactúa con la ventana modal.

SweetAlert: ventajas

Interacciones de Usuario Avanzadas: Puedes mostrar ventanas modales de entrada de texto, cargar contenido externo, crear secuencias de ventanas modales y más.

Licencia de Código Abierto: SweetAlert es de código abierto y se distribuye bajo la licencia MIT, lo que significa que es gratuito para su uso en proyectos comerciales y no comerciales.

SweetAlert: diferencia de versiones

SweetAlert2 (también conocido como Swal2) es una versión mejorada y más moderna de la biblioteca original SweetAlert (SweetAlert 1).

A lo largo del tiempo, **SweetAlert2** se ha convertido en la versión más recomendada y ampliamente utilizada debido a sus mejoras y características adicionales en comparación con SweetAlert 1.

Algunas de las principales diferencias entre SweetAlert2 y SweetAlert 1:

SweetAlert: diferencia de versiones

UNTREF
UNIVERSIDAD NACIONAL DE TRES DE FEBRERO

Diseño y Personalización:

SweetAlert2 ofrece una mayor flexibilidad en términos de personalización y diseño. Puedes modificar fácilmente la apariencia de las ventanas modales mediante CSS y personalizar los estilos de forma más avanzada.

Mensajes HTML:

SweetAlert2 permite el uso de HTML en los mensajes y títulos de las ventanas modales. Esto significa que puedes incluir texto formateado y elementos HTML en tus mensajes.

Mejoras en la Interfaz de Usuario:

SweetAlert2 presenta una interfaz de usuario más moderna y elegante en comparación con SweetAlert 1. Las animaciones y transiciones son más suaves, lo que mejora la experiencia del usuario.

SweetAlert: diferencia de versiones

UNTREF
UNIVERSIDAD NACIONAL DE TRES DE FEBRERO

Comportamiento Mejorado:

SweetAlert2 ofrece un comportamiento mejorado en términos de manipulación de promesas y gestión de eventos. Esto hace que sea más fácil trabajar con las ventanas modales y ejecutar código después de que el usuario interactúa con ellas.

Compatibilidad con Promesas:

SweetAlert2 se integra mejor con el modelo de programación basado en promesas de JavaScript, lo que facilita la ejecución de acciones después de que el usuario interactúa con una ventana modal.

Soporte Activo:

SweetAlert2 está en desarrollo activo y recibe actualizaciones y correcciones de errores regulares. La comunidad de desarrollo está más comprometida con SweetAlert2.

SweetAlert: instalación

Para utilizar SweetAlert en tu proyecto, primero debes incluir la biblioteca en tu código a través de una etiqueta <script> o utilizando una herramienta de administración de paquetes como npm o yarn si estás trabajando en un entorno de desarrollo moderno.

Luego, puedes comenzar a crear ventanas modales personalizadas según tus necesidades.

SweetAlert: instalación CDN

Si estás trabajando en un proyecto HTML simple sin un sistema de construcción, puedes cargar SweetAlert2 directamente desde un CDN en tu página HTML:

Agrega los siguientes enlaces a tu archivo HTML dentro de la etiqueta <head>:

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/sweetalert2@11/dist/sweetalert2.min.css">
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
```

Luego, en tu código JavaScript, puedes usar SweetAlert2 como sigue:

```
Swal.fire("¡Hola, SweetAlert2!");
```

SweetAlert: instalación CDN

Si estás utilizando un proyecto basado en Node.js, puedes instalar SweetAlert2 usando npm o yarn desde la terminal de la siguiente manera:

```
npm install sweetalert2
```

```
yarn add sweetalert2
```

Después de la instalación, puedes importar SweetAlert2 en tu código JavaScript o TypeScript:

```
import Swal from 'sweetalert2';
```

SweetAlert: Ejemplos

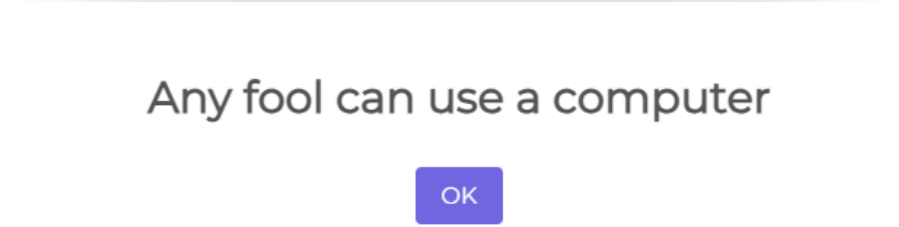
SweetAlert2 ofrece varios componentes y funciones que se pueden utilizar para crear ventanas modales personalizadas y mensajes interactivos en tus aplicaciones web.

SweetAlert: Ejemplos

Swal.fire(): Este es el método principal para crear ventanas modales personalizadas con SweetAlert2.

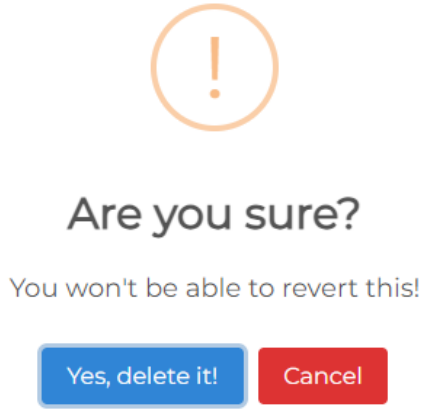
Se utiliza para mostrar mensajes de alerta, confirmación, información o cualquier tipo de ventana modal personalizada.

```
Swal.fire("Título", "Mensaje de texto", "success");
```



SweetAlert: Ejemplos

Botones Personalizados: Puedes agregar botones personalizados a tus ventanas modales y asignarles funciones específicas cuando se hacen clic.




```
Swal.fire({
  title: "¿Estás seguro?",
  text: "Esta acción no se puede deshacer.",
  icon: "warning",
  showCancelButton: true,
  confirmButtonText: "Sí, eliminar",
  cancelButtonText: "Cancelar",
}).then((result) => {
  if (result.isConfirmed) {
    // Aquí puedes ejecutar la acción de eliminación
    Swal.fire("Eliminado", "El elemento ha sido eliminado.", "success");
  }
});
```

SweetAlert: Ejemplos

Ventanas Modales de Entrada de Texto: SweetAlert2 permite mostrar ventanas modales que solicitan al usuario que ingrese texto. Esto es útil para formularios de entrada.

```
Swal.fire({
  input: "text",
  inputLabel: "Ingrese su nombre",
  inputPlaceholder: "Ejemplo: Juan",
}).then((nombre) => {
  if (nombre.value) {
    Swal.fire(`¡Hola, ${nombre.value}!`, "", "success");
  }
});
```

SweetAlert: Ejemplos

Ventanas Modales de Carga de Contenido Externo: Puedes cargar contenido HTML externo dentro de una ventana modal de SweetAlert2.

```
Swal.fire({
  title: "Cargando contenido externo...",
  html: '<div style="text-align:center;"><img src="spinner.gif" alt="Cargando..." />
</div>',
  showCancelButton: true,
  showConfirmButton: false,
});
```

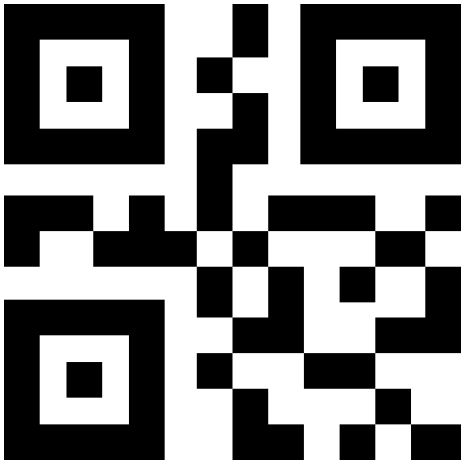
Qr Code

QrCode

La librería qrcode es una biblioteca de JavaScript que permite generar códigos QR en aplicaciones web.

Los códigos QR son una forma popular de codificar información en una imagen bidimensional que se puede escanear con la cámara de un dispositivo móvil o una aplicación de escaneo de códigos QR.

Esta librería facilita la creación de códigos QR personalizados que pueden contener diversos tipos de datos, como URL, texto, información de contacto, números de teléfono y más.



https://github.com/soldair/node-qrcode

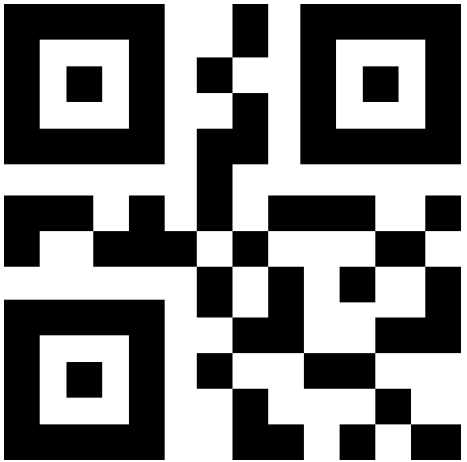
QrCode

Interacción con la Cámara:

Puedes integrar la librería qrcode con la cámara del dispositivo para permitir a los usuarios escanear códigos QR desde la cámara de su teléfono o tableta.

Compatibilidad con Diferentes Formatos de Imagen:

La librería puede generar códigos QR en varios formatos de imagen, como PNG, SVG o incluso en un elemento HTML canvas, lo que brinda flexibilidad en cómo deseas mostrar o guardar los códigos generados.

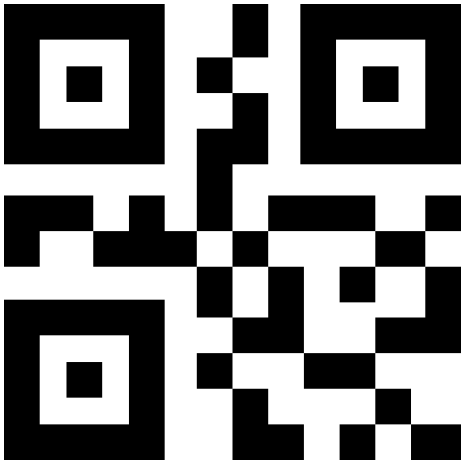


QrCode

UNTREF
UNIVERSIDAD NACIONAL DE TRES DE FEBRERO

Uso en Aplicaciones Web:

La librería qrcode es adecuada para su uso en aplicaciones web, ya que es fácil de integrar en proyectos basados en JavaScript y HTML.



QrCode: instalación

Abre una terminal o línea de comandos en el directorio de tu proyecto o donde desees instalar la librería y listo esta funcional.

Ejecuta el siguiente comando para instalar qrcode utilizando npm:

```
npm install qrcode
```

Si prefieres usar Yarn, puedes ejecutar este comando en su lugar:

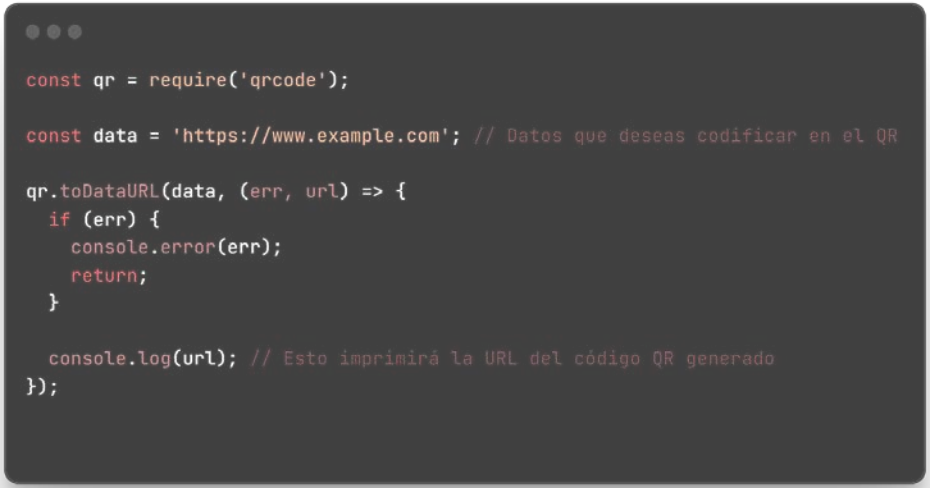
```
yarn add qrcode
```

QrCode: ejemplos

UNTREF
UNIVERSIDAD NACIONAL DE TRES DE FEBRERO

Generación de Códigos QR:

La función principal de la librería es generar códigos QR a partir de datos que proporcionas, como URLs, texto, números de teléfono, direcciones de correo electrónico, información de contacto y más.



```
const qr = require('qrcode');

const data = 'https://www.example.com'; // Datos que deseas codificar en el QR

qr.toDataURL(data, (err, url) => {
  if (err) {
    console.error(err);
    return;
  }

  console.log(url); // Esto imprimirá la URL del código QR generado
});
```

QrCode: ejemplos

UNTREF
UNIVERSIDAD NACIONAL DE TRES DE FEBRERO

Personalización de Estilos:
Puedes personalizar la apariencia de los códigos QR generados mediante la configuración de colores, tamaños y niveles de corrección de errores.

```
const qr = require('qrcode');

const data = 'https://www.example.com';

const opciones = {
  color: {
    dark: '#000',  // Color de las partes oscuras del código QR
    light: '#fff', // Color de las partes claras del código QR
  },
  width: 300,      // Ancho del código QR en pixeles
  errorCorrectionLevel: 'H', // Nivel de corrección de errores (H, M, Q o L)
};

qr.toDataURL(data, opciones, (err, url) => {
  if (err) {
    console.error(err);
    return;
  }

  console.log(url);
});
```

QrCode: ejemplos

UNTREF
UNIVERSIDAD NACIONAL
DE TRES DE FEBRERO

Generación de Códigos QR Personalizados:
Puedes generar códigos QR personalizados incorporando imágenes o logotipos en el centro del código QR.

QrCode: ejemplos

UNTREF
UNIVERSIDAD NACIONAL DE TRES DE FEBRERO

Generación de Códigos QR en Diferentes Formatos:

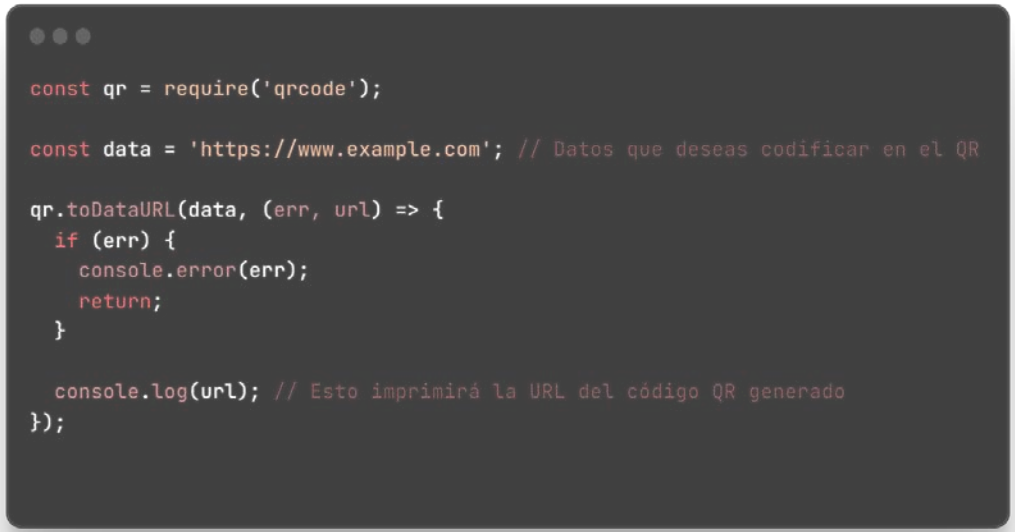
La librería qrcode puede generar códigos QR en varios formatos, como PNG, SVG, y en elementos HTML canvas.

```
const qr = require('qrcode');

const data = 'https://www.example.com';

qr.toFile('codigoqr.png', data, (err) => {
  if (err) {
    console.error(err);
    return;
  }

  console.log('Código QR guardado como códigoqr.png');
});
```

QrCode: ejemplos

UNTREF
UNIVERSIDAD NACIONAL DE TRES DE FEBRERO

qr.toDataURL() para generar un código QR a partir de una URL y luego imprime la URL del código QR generado en la consola.



```
const qr = require('qrcode');

const data = 'https://www.example.com'; // Datos que deseas codificar en el QR

qr.toDataURL(data, (err, url) => {
  if (err) {
    console.error(err);
    return;
  }

  console.log(url); // Esto imprimirá la URL del código QR generado
});
```

¿Dudas o consultas?

¡Muchas Gracias!