OPTIMIZACIÓN DE FUENTES

WOFF2 (Web Open Font Format 2) es una versión mejorada del formato WOFF, utilizado para incrustar fuentes en páginas web. WOFF2 aplica compresión adicional, lo que permite reducir el tamaño de los archivos de fuente hasta un **30%** en comparación con WOFF1 y otros formatos como TTF y OTF.

EJEMPLO DE REDUCCIÓN DE TAMAÑO:

Supongamos que tenemos una fuente llamada OpenSans en diferentes formatos:

Formato	Tamaño Original	Tamaño Comprimido	Reducción (%)
TTF	150 KB	150 KB	0%
OTF	140 KB	140 KB	0%
WOFF	100 KB	100 KB	33%
WOFF2	70 KB	70 KB	**53%**

🔹 **Resultado**: WOFF2 logra reducir el tamaño de OpenSans de 150 KB (TTF) a 70 KB, ahorrando ancho de banda y mejorando los tiempos de carga.

IMPLEMENTACIÓN DE WOFF2 EN CSS:

```
@font-face {
  font-family: 'OpenSans';
  src: url('fonts/OpenSans.woff2') format('woff2'),
       url('fonts/OpenSans.woff') format('woff');
  font-weight: normal;
  font-style: normal;
}
```

🔹 **Explicación**:

- Se carga primero la fuente en formato woff2.
- Si el navegador no admite woff2, se usa woff como respaldo.
- Navegadores antiguos que no admiten WOFF cargan el formato estándar (si está disponible).

VENTAJAS DE WOFF2:

- **Reducción de Tamaño**: Hasta un 30% más pequeño que WOFF y hasta 50% frente a TTF.
- **Carga Más Rápida**: Mejora el rendimiento web, reduciendo el tiempo de carga de fuentes personalizadas.

- **Compatibilidad**: Es compatible con la mayoría de los navegadores modernos (Chrome, Firefox, Edge, Opera).

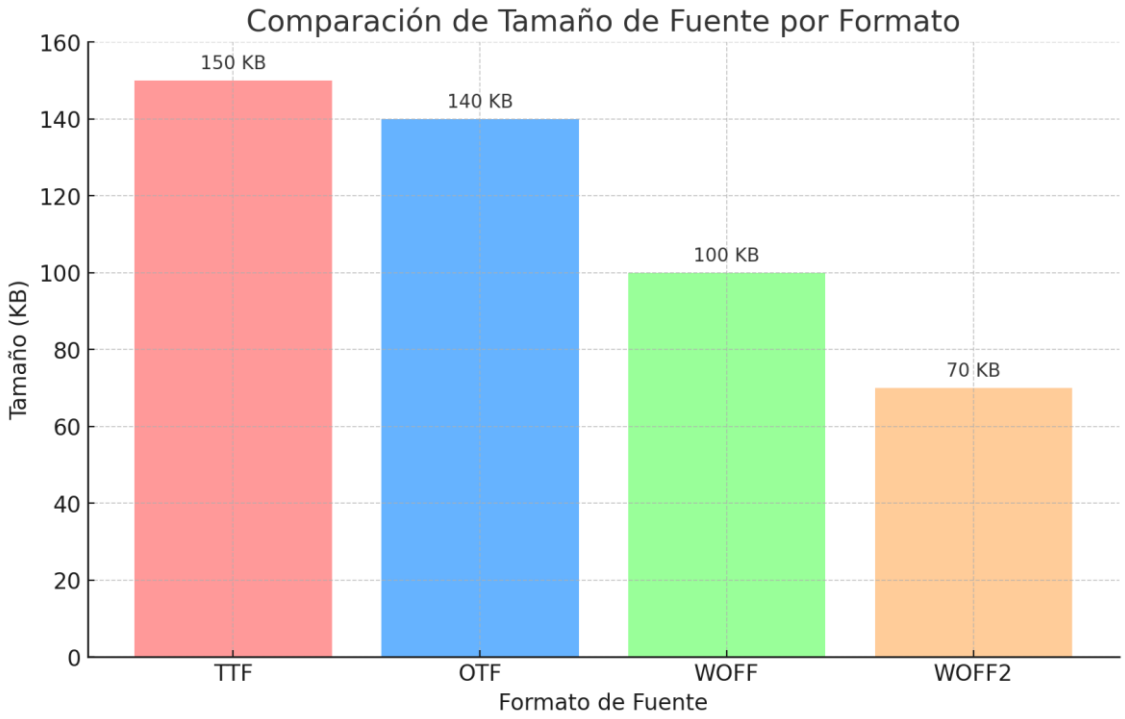
COMPATIBILIDAD DE NAVEGADORES PARA WOFF2:

Navegador	Soporta WOFF2
Chrome	Sí (desde v36)
Firefox	Sí (desde v35)
Edge	Sí (desde v14)
Safari	Sí (desde v12)
Opera	Sí (desde v26)

COMPARACIÓN VISUAL DE TAMAÑO:

Comparación del tamaño entre los diferentes formatos de fuente.

Comparación de Tamaño de Fuente por Formato



Aquí tienes una gráfica que muestra la reducción de tamaño entre diferentes formatos de fuente (TTF, OTF, WOFF y WOFF2). Como se observa, **WOFF2** ofrece la mayor compresión, reduciendo significativamente el tamaño del archivo, lo que mejora el rendimiento de la página.

Implementar carga condicional de fuentes y fallbacks garantiza que, si una fuente personalizada no se carga correctamente, el navegador utilice una alternativa. Esto mejora la experiencia del usuario y la accesibilidad.

1. IMPLEMENTACIÓN BÁSICA CON FALLBACKS (CSS):

```
body {
  font-family: 'Open Sans', Arial, sans-serif;
}
```

- **Explicación:**
  - Si Open Sans no está disponible, se cargará Arial.
  - Si Arial tampoco está disponible, se usará una fuente genérica sans-serif.

2. IMPLEMENTAR CARGA CONDICIONAL CON FONT-DISPLAY:

El uso de font-display permite controlar cómo y cuándo se muestra una fuente web mientras se carga.

```
@font-face {
  font-family: 'Roboto';
  src: url('fonts/Roboto.woff2') format('woff2'),
       url('fonts/Roboto.woff') format('woff');
  font-display: swap;
}
```

- **Opciones de font-display:**
  - **swap** – Usa la fuente de fallback hasta que la fuente personalizada esté disponible.
  - **fallback** – Usa el fallback si la fuente personalizada tarda demasiado en cargar.
  - **block** – Oculta el texto hasta que la fuente esté lista.
  - **optional** – Usa el fallback si la fuente no se carga rápidamente.

3. IMPLEMENTACIÓN DE FALLBACKS POR CLASE:

```
<h1 class="heading">Título de Ejemplo</h1>
.heading {
  font-family: 'Poppins', 'Helvetica Neue', Helvetica, Arial, sans-serif;
}
```

- **Ventaja**: Si Poppins falla, el navegador usará Helvetica Neue y continuará con Arial o sans-serif.

4. CARGA CONDICIONAL CON JAVASCRIPT (DYNAMIC LOADING):

Este método permite cargar fuentes de forma asíncrona.

```
<script>
  WebFont.load({
    google: {
      families: ['Roboto:400,700']
    },
```

```
    timeout: 2000 // Fallback después de 2 segundos
  });
</script>
```

- **Explicación:**
  - Si Roboto no se carga en 2 segundos, el navegador usa una fuente de respaldo.
  - **Ventaja**: Evita que el contenido permanezca invisible durante la carga.

5. FALLBACK CON VARIABLES CSS (MODERN APPROACH):

```
:root {
  --font-primary: 'Inter', sans-serif;
  --font-fallback: Arial, sans-serif;
}

body {
  font-family: var(--font-primary), var(--font-fallback);
}
```

- **Ventaja**: Usa una variable para gestionar fuentes principales y de respaldo.

VENTAJAS DE LA CARGA CONDICIONAL Y FALLBACKS:

- **Mejora el rendimiento**: Evita bloqueos durante la carga de fuentes.
- **Accesibilidad**: Garantiza que el contenido sea legible incluso si una fuente falla.
- **Compatibilidad**: Se adapta a diferentes dispositivos y navegadores.