# INFORME: Unidades de medidas responsivas

#### ¿Para que sirve?

Las unidades de medida responsivas sirven para crear diseños web que se adapten automáticamente a diferentes tamaños de pantalla y dispositivos. Estas unidades permiten que los elementos en una página web cambien de tamaño, posición y proporción en función del entorno en el que se visualizan, como en una computadora de escritorio, tableta o teléfono móvil. Esto asegura que el contenido sea legible, accesible y visualmente coherente en cualquier dispositivo, mejorando la experiencia del usuario

#### ¿Cuales hay?

· **Porcentaje (%)**: Ajusta el tamaño de un elemento en relación con su contenedor padre. Por ejemplo, un ancho de 50% hace que un elemento ocupe la mitad del ancho del contenedor.

· **Ventajas**:

* Flexible y fácil de usar para diseños fluidos.
* Ideal para ajustar elementos dentro de un contenedor, manteniendo proporciones relativas.

· **Desventajas**:

* Puede ser difícil de manejar en diseños complejos o en combinación con otras unidades responsivas.
* La dependencia del tamaño del contenedor padre puede generar problemas de diseño si este cambia inesperadamente.

.container {

width: 50%; /\* El contenedor ocupa el 50% del ancho de su padre \*/

}

· **Viewport Width (vw)**: 1vw equivale al 1% del ancho del viewport (la ventana del navegador). Se utiliza para dimensionar elementos en función del ancho de la pantalla.

· **Ventajas**:

* Permite un control preciso sobre el tamaño de los elementos en relación con el ancho de la pantalla.
* Ideal para elementos que deben mantener su proporción respecto a la pantalla completa.

· **Desventajas**:

* Puede generar problemas de legibilidad en dispositivos pequeños si no se maneja con cuidado.
* No tiene en cuenta la barra de desplazamiento, lo que puede causar desajustes

.full-width {

width: 100vw; /\* El elemento ocupa el 100% del ancho del viewport \*/

}

· **Viewport Height (vh)**: 1vh es igual al 1% de la altura del viewport. Permite ajustar la altura de los elementos según la altura de la pantalla.

· **Ventajas**:

* Útil para crear elementos que necesitan ocupar un porcentaje específico de la altura de la pantalla.
* Perfecto para layouts que requieren una altura dinámica, como pantallas completas.

· **Desventajas**:

* Similar a vw, puede causar problemas de visualización en pantallas pequeñas.
* La altura de la pantalla puede cambiar con la presencia de teclados virtuales o barras de herramientas en dispositivos móviles.

.half-height {

height: 50vh; /\* El elemento ocupa el 50% de la altura del viewport \*/

}

· **Ventajas**:

* Consistente y fácil de mantener, especialmente en proyectos grandes.
* Permite un control centralizado del tamaño de la fuente, lo que facilita ajustes globales.

· **Desventajas**:

* Menos flexible en comparación con em en situaciones donde se necesita más control granular.
* Puede complicar la creación de diseños altamente específicos si la base rem no está bien definida

· **Rem**: Relativa al tamaño de la fuente raíz (<html>). Se usa comúnmente para establecer tamaños de fuente y márgenes que escalen de manera consistente en toda la página.

· **Ventajas**:

* Más flexible que rem cuando se necesita que un tamaño se escale en relación con elementos padres específicos.
* Útil para crear componentes modulares con tamaños relativos dentro de un contexto específico.

· **Desventajas**:

* Puede ser más difícil de gestionar en proyectos grandes debido a su dependencia del contexto.
* Los tamaños pueden volverse impredecibles si no se controla bien la cascada de estilos.

body {

font-size: 16px; /\* Tamaño base de la fuente \*/

}

h1 {

font-size: 2rem; /\* 2 veces el tamaño de la fuente raíz \*/

}

· **Em**: Relativa al tamaño de la fuente del elemento padre. Similar a rem, pero más específica al contexto del elemento donde se aplica.

· **Ventajas**:

* Más flexible que rem cuando se necesita que un tamaño se escale en relación con elementos padres específicos.
* Útil para crear componentes modulares con tamaños relativos dentro de un contexto específico.

· **Desventajas**:

* Puede ser más difícil de gestionar en proyectos grandes debido a su dependencia del contexto.
* Los tamaños pueden volverse impredecibles si no se controla bien la cascada de estilos.

.text {

font-size: 1.5em; /\* 1.5 veces el tamaño de la fuente del elemento padre \*/

}