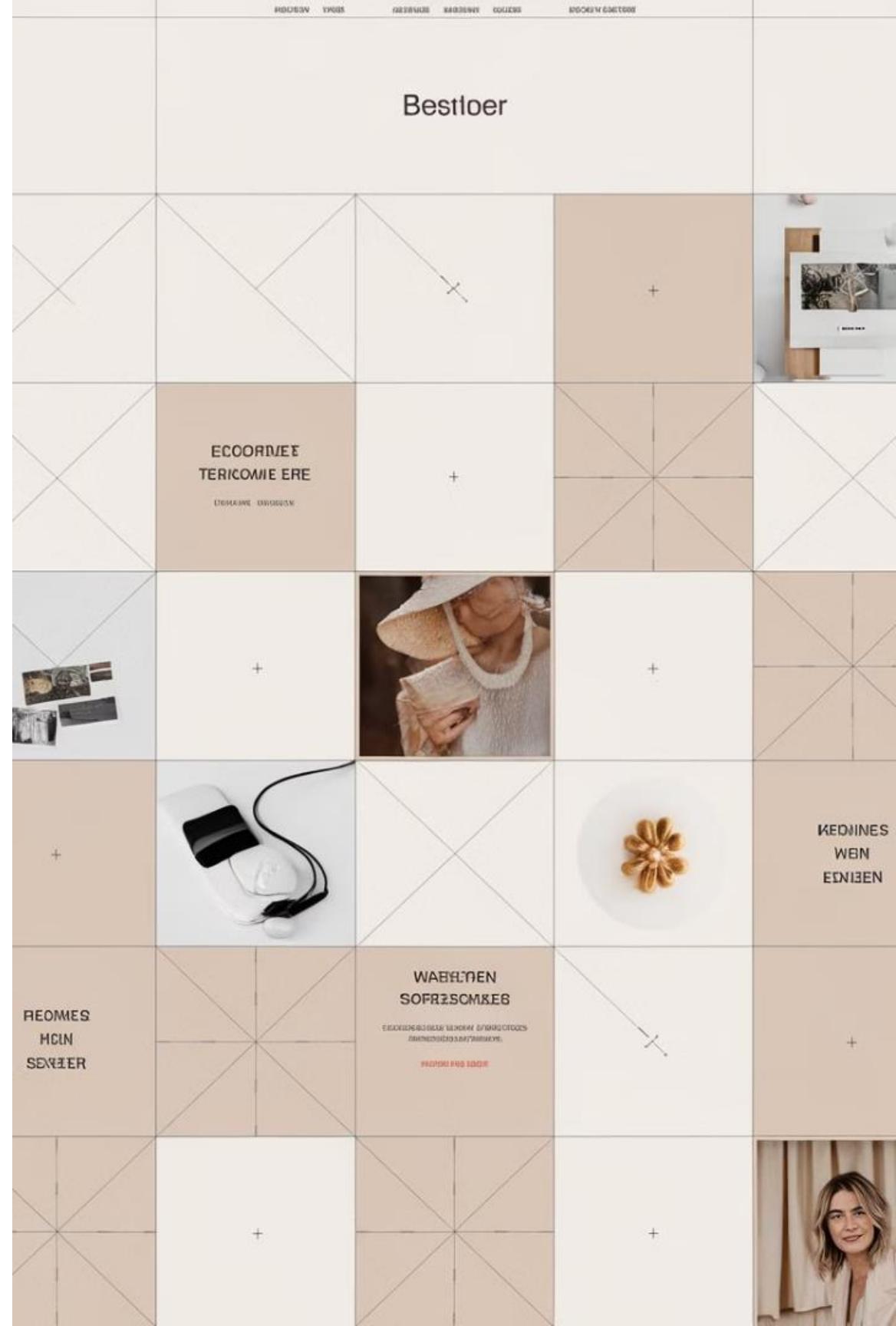


# Sistema de cuadrícula con CSS Grid (Parte 2)

Exploraremos el uso avanzado de CSS Grid, incluyendo técnicas como grid-template, grid-gap y la unidad fr, así como el desarrollo de plantillas de diseño complejas.



# Uso avanzado de Grid: grid-template

## Definir filas y columnas de manera flexible

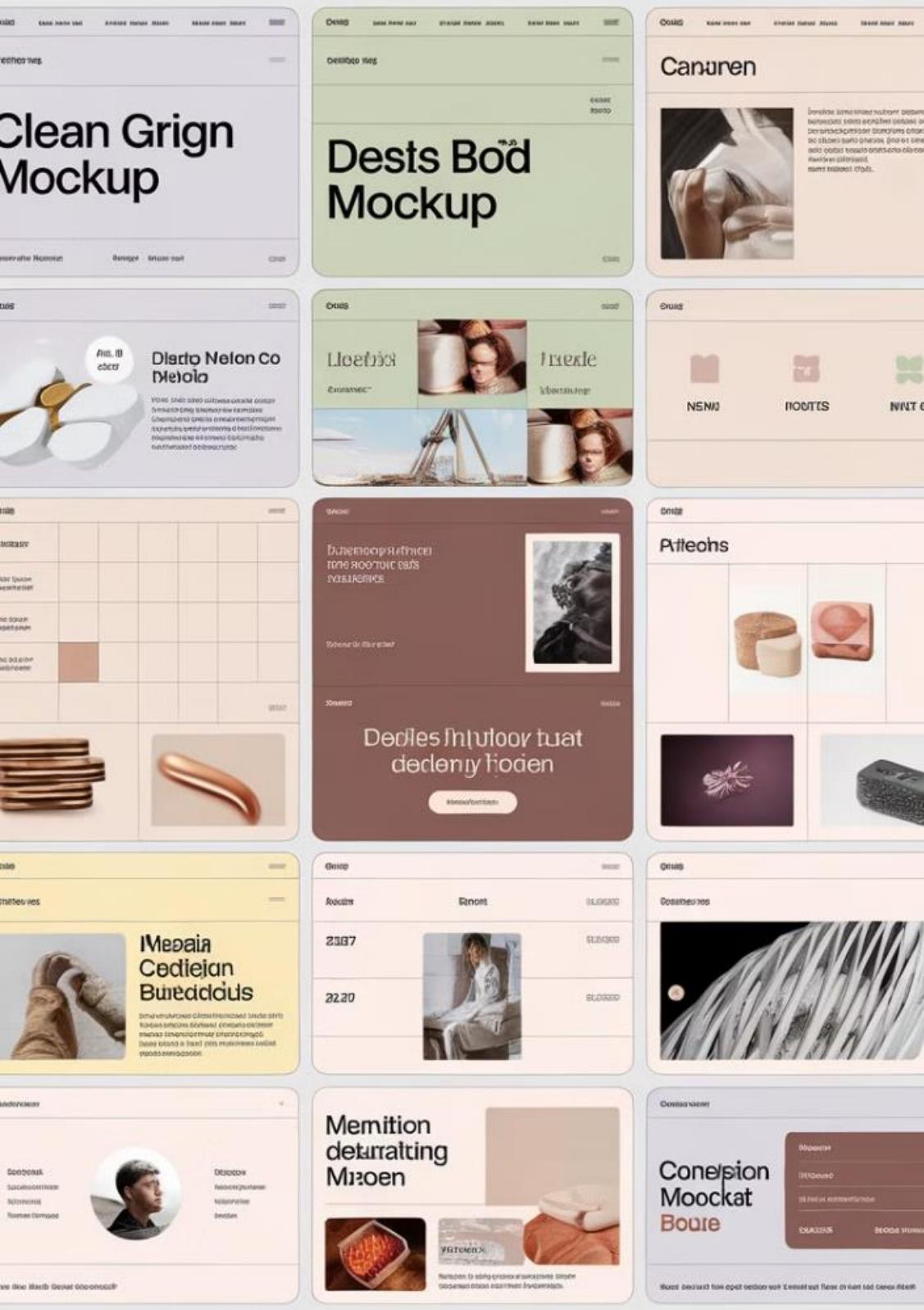
La propiedad **grid-template** permite definir de manera flexible la estructura de filas y columnas de un sistema de cuadrícula. Esto brinda mayor control y personalización del diseño.

## Utilizar unidades "fr" para distribuir el espacio

La unidad **fr** (del inglés "fraction") se usa para dividir el espacio disponible de manera proporcional entre las filas y columnas del grid.

## Combinar unidades fijas y variables

Es posible combinar unidades fijas como **px** o **em** con las unidades variables **fr** para lograr diseños más complejos y adaptables.



# Definir filas y columnas de manera flexible



## Filas Flexibles

Utiliza la propiedad **grid-template-rows** para definir el tamaño de las filas de manera flexible, adaptándose al contenido y al espacio disponible.



## Columnas Flexibles

Ajusta el tamaño de las columnas con la propiedad **grid-template-columns**, permitiendo que se expandan o contraigan según sea necesario.



## Diseño Responsivo

Las filas y columnas flexibles facilitan la adaptación del diseño a diferentes tamaños de pantalla, logrando un aspecto fluido y adaptable.

# CSS Grid Layout

Website Design

## Uso avanzado de Grid: grid-template

### 1 Distribuir el espacio con unidades "fr"

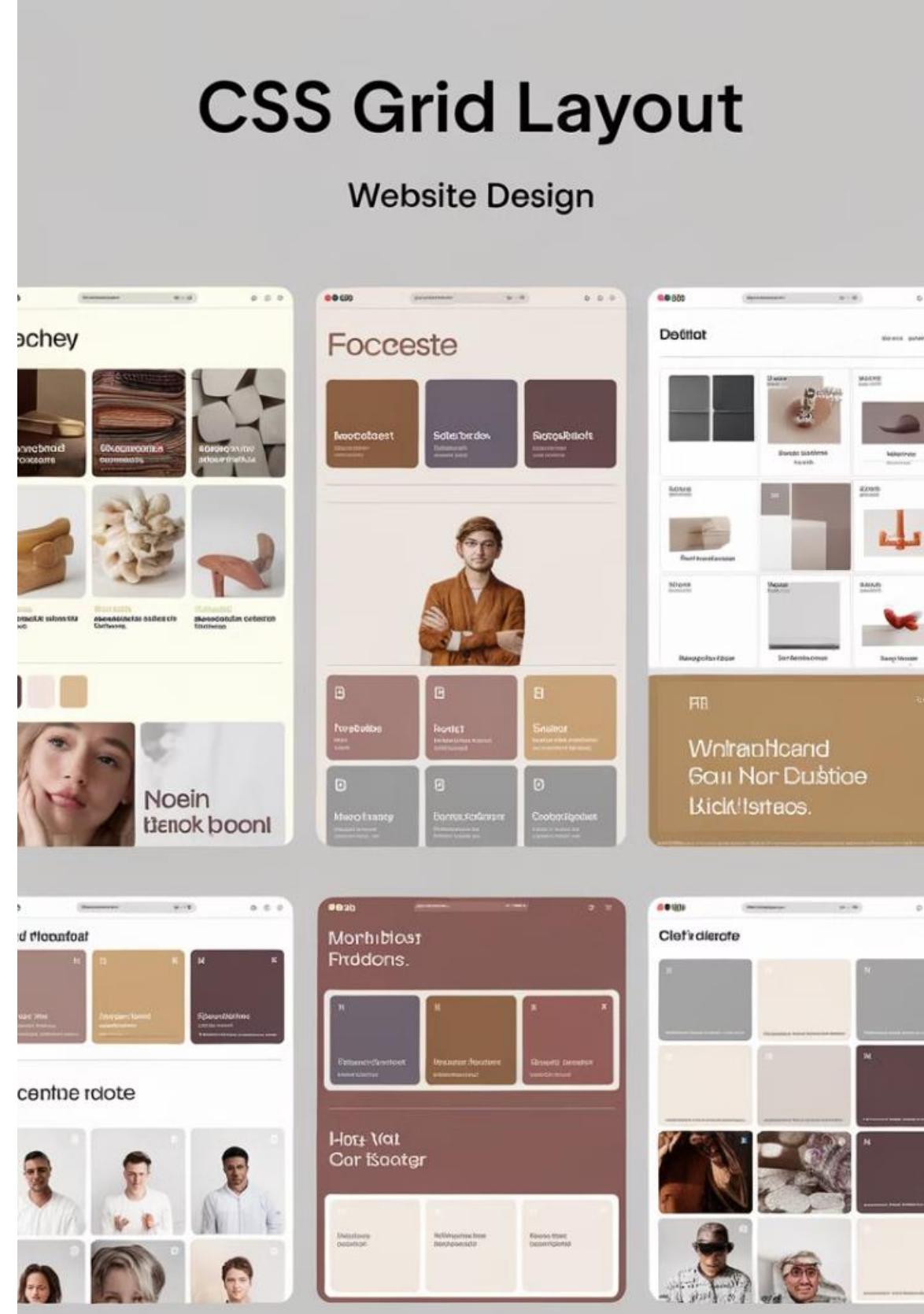
Las unidades "fr" (fracción) permiten distribuir el espacio disponible de manera proporcional entre las filas y columnas, sin tener que especificar medidas fijas.

### 2 Flexibilidad y adaptabilidad

Al usar "fr", el diseño se ajusta automáticamente cuando se modifica el tamaño del contenedor o se agregan/eliminan elementos, manteniendo una distribución equilibrada.

### 3 Combinar unidades fijas y variables

Es posible combinar unidades "fr" con medidas fijas como px o % para lograr diseños más complejos y controlados.



# Combinar unidades fijas y variables

## Unidades fijas

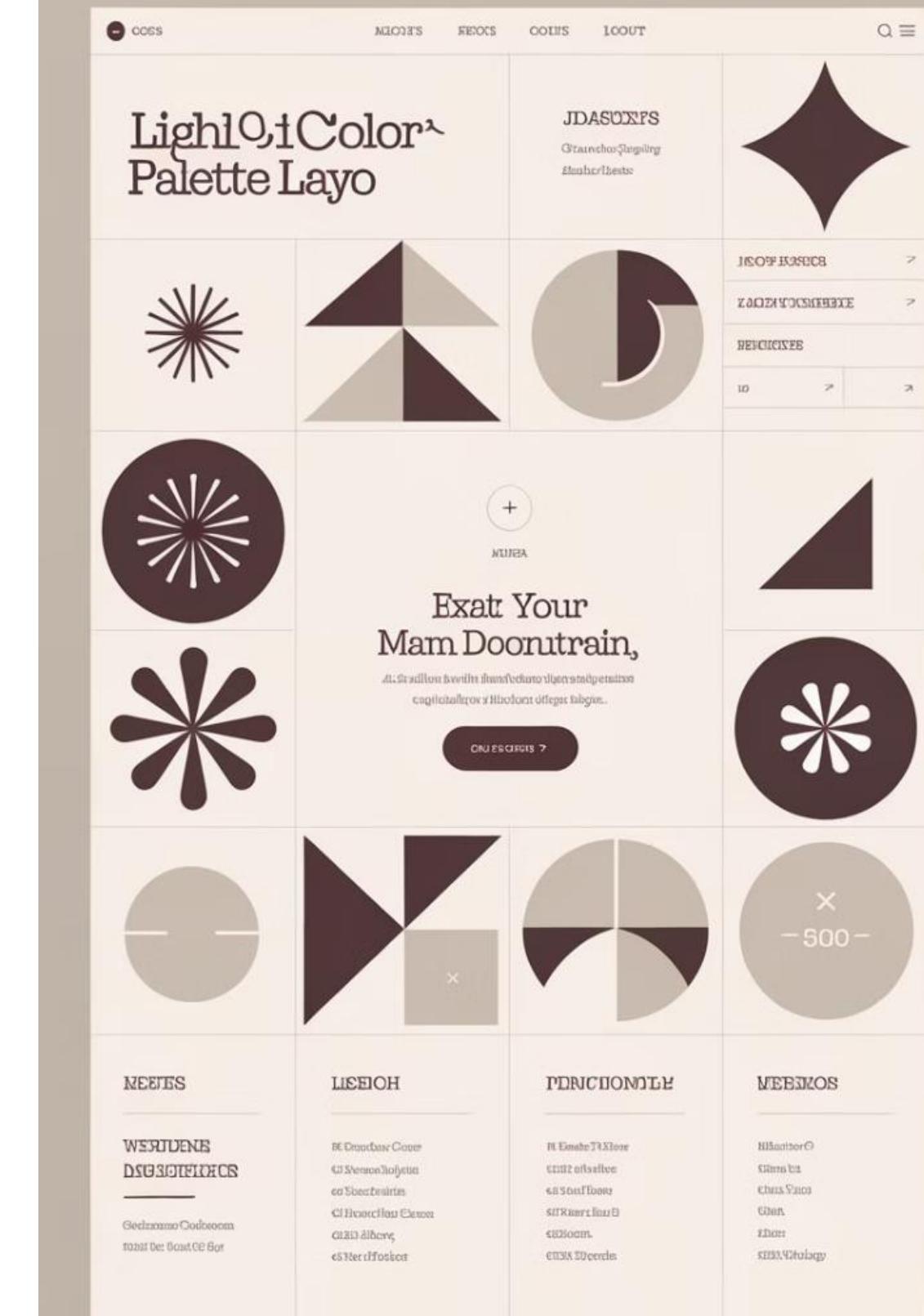
Las unidades fijas como píxeles (px) o puntos (pt) permiten definir un tamaño preciso en el diseño.

## Unidades variables

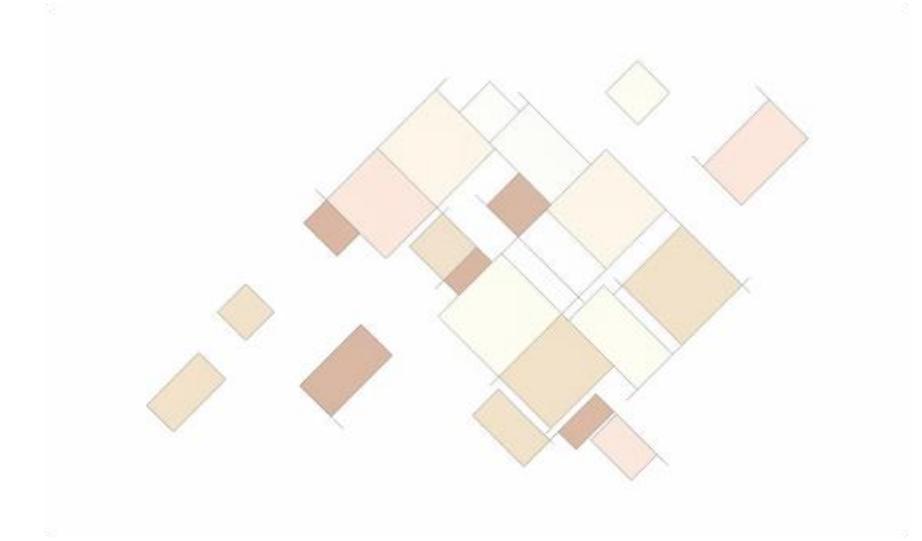
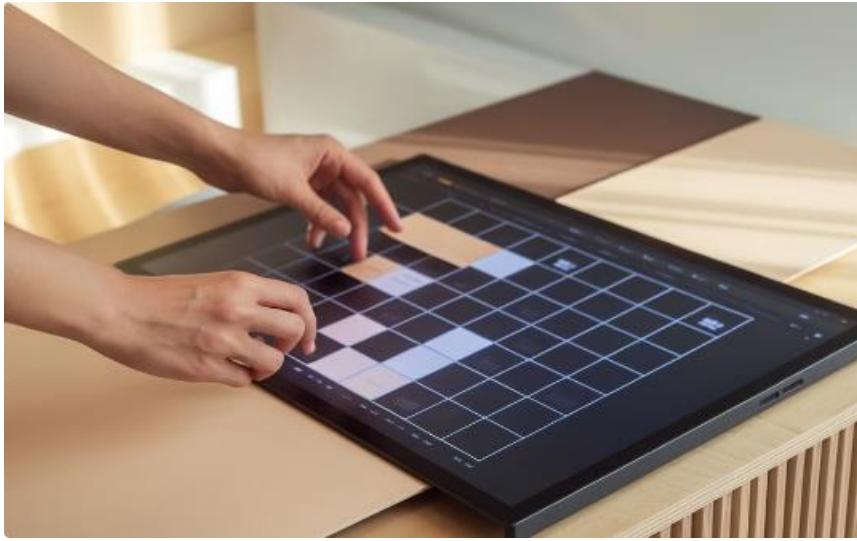
Las unidades variables como porcentajes (%) o la unidad "fr" (fracción) permiten crear diseños flexibles y responsivos.

## Combinación efectiva

Combinar unidades fijas y variables en grid te permite crear estructuras sólidas con elementos ajustables al espacio disponible.



# Uso avanzado de Grid: grid-gap



## Controlar el espacio entre filas y columnas

La propiedad grid-gap permite definir el espacio entre los elementos de la cuadrícula, ya sea de manera uniforme o con valores diferentes para filas y columnas.

## Aplicar diferentes espaciados a filas y columnas

Utilizando grid-row-gap y grid-column-gap, se puede ajustar de forma independiente el espacio entre filas y columnas para obtener diseños más personalizados.

## Combinar gaps horizontales y verticales

Al combinar grid-row-gap y grid-column-gap, se puede lograr una división más precisa del espacio dentro de la cuadrícula, adaptándose a necesidades específicas del diseño.

# Uso avanzado de Grid: grid-gap

## Controlar el espacio entre filas y columnas

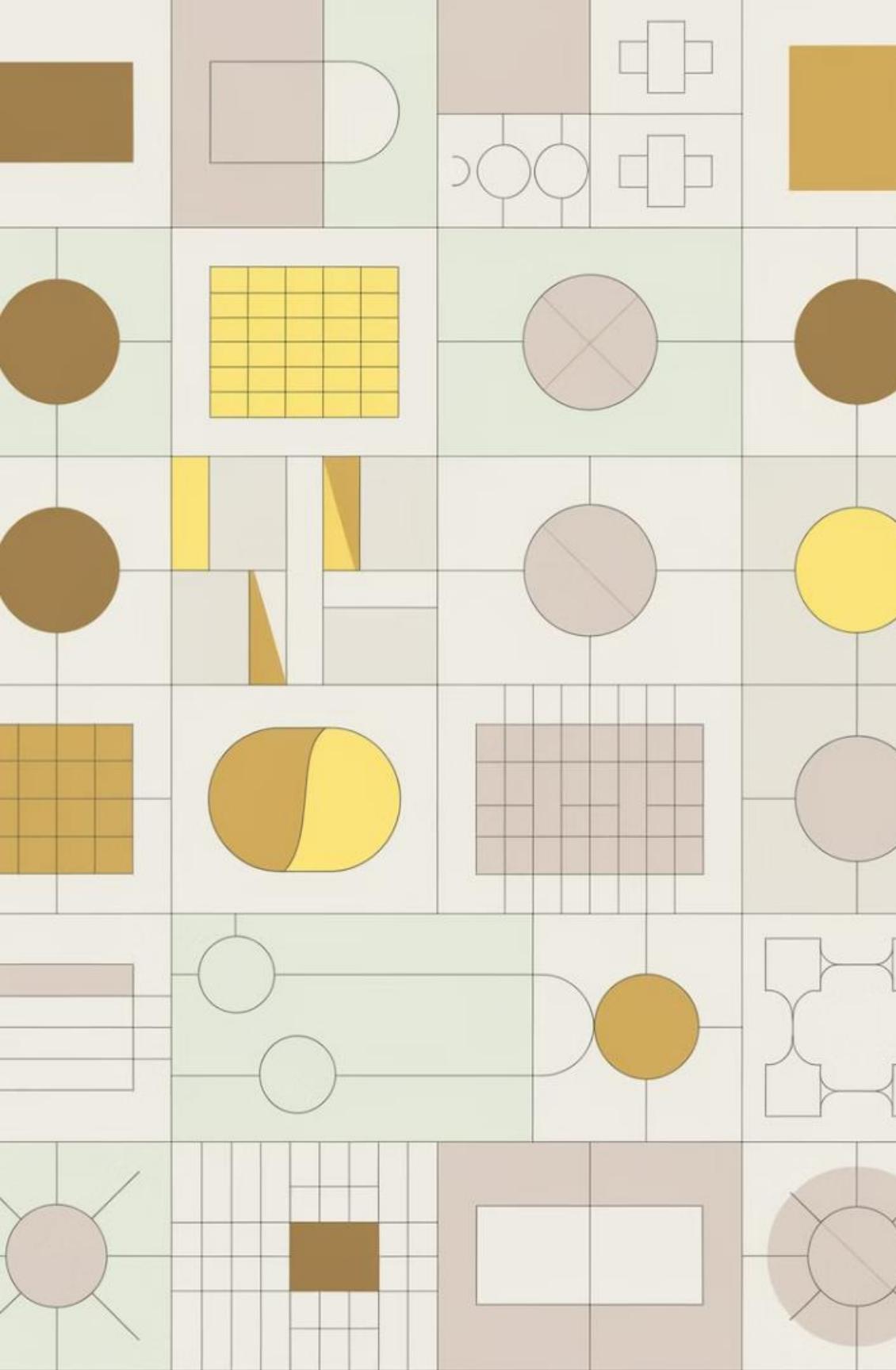
CSS Grid permite definir el espacio entre los elementos de la cuadricula, tanto horizontal como verticalmente, a través de la propiedad **grid-gap**.

## Aplicar diferentes espaciados a filas y columnas

Incluso puedes establecer diferentes valores para el espacio entre filas y entre columnas, usando **grid-row-gap** y **grid-column-gap**.

## Combinar gaps horizontales y verticales

Si quieres ajustar el espacio de manera más precisa, puedes utilizar la propiedad shorthand **gap** que permite definir el espacio entre filas y columnas en un solo lugar.



# Uso avanzado de Grid: grid-gap



## Espaciado de columnas

Permite controlar el espacio entre las columnas de la cuadrícula, definiendo un valor único o diferente para cada columna.



## Espaciado de filas

De manera similar, se puede establecer un espaciado personalizado entre las filas de la cuadrícula.



## Combinando gaps

Es posible aplicar diferentes espaciados a filas y columnas de forma independiente, creando un diseño más complejo y personalizado.



# Combinar gaps horizontales y verticales

## Espaciado Personalizado

Utiliza las propiedades **row-gap** y **column-gap** para definir el espacio entre filas y columnas de manera independiente.

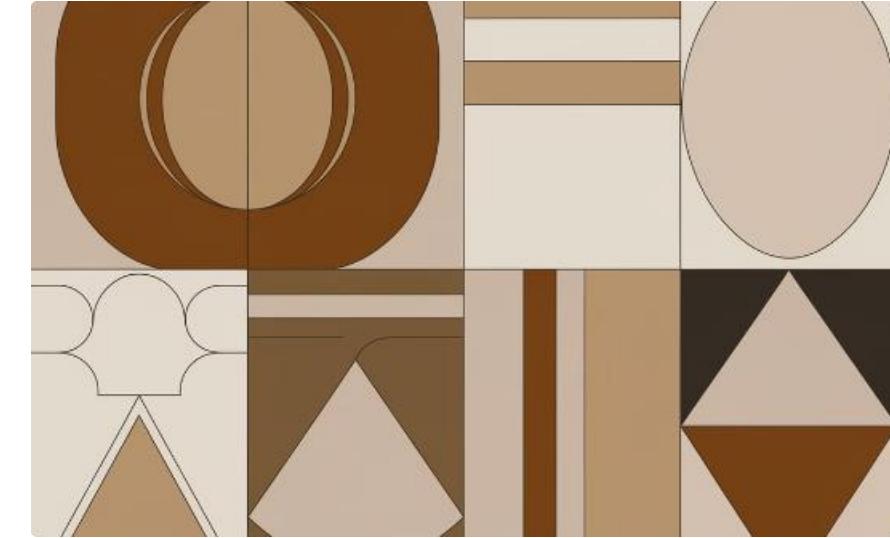
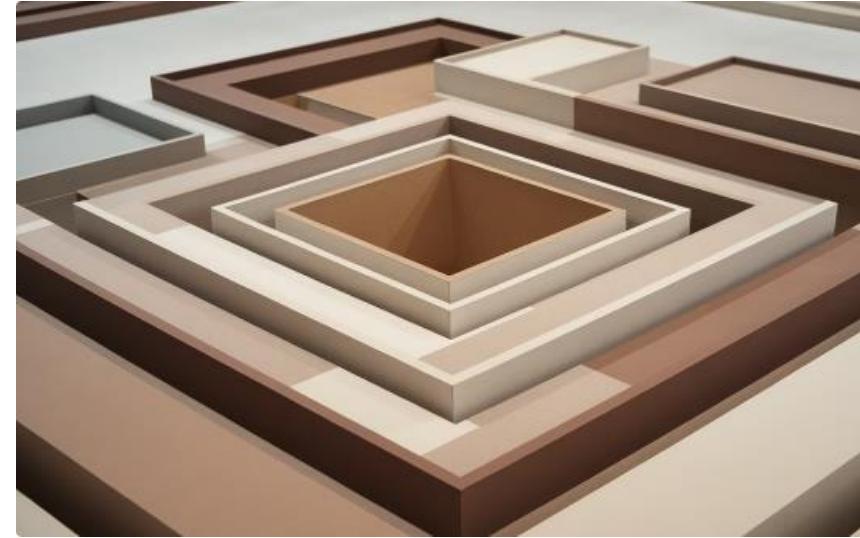
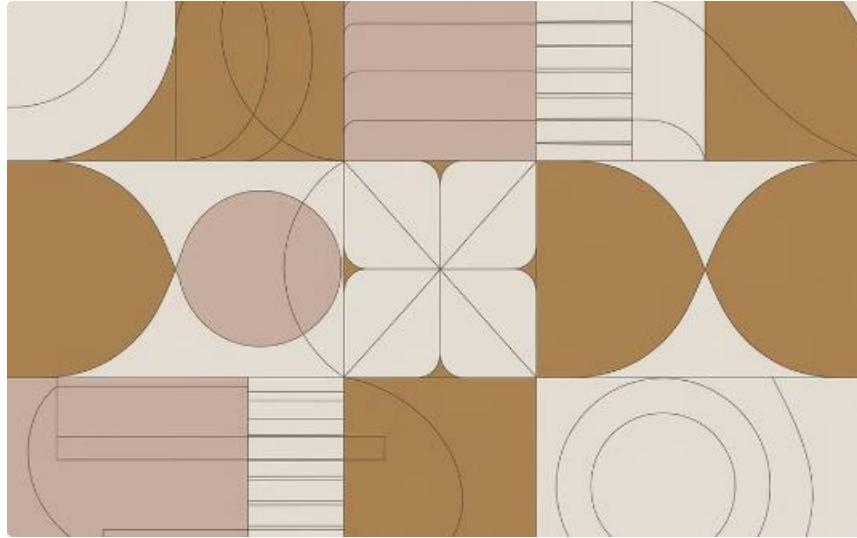
## Alineación de Elementos

Al combinar gaps horizontales y verticales, puedes ajustar la posición de los elementos dentro de la cuadrícula para crear diseños más equilibrados.

## Diseños Complejos

Esta técnica te permite construir diseños asimétricos y creativos, superponiendo elementos y creando una sensación de profundidad.

# Plantillas de diseño complejas



## Diseños Asimétricos

Los diseños de cuadrícula asimétricos permiten crear composiciones dinámicas y visualmente interesantes, rompiendo con la simetría tradicional.

## Anidación de Contenedores

Al anidar contenedores de cuadrícula, se pueden lograr estructuras de diseño complejas con múltiples capas y superposiciones de elementos.

## Superposición de Elementos

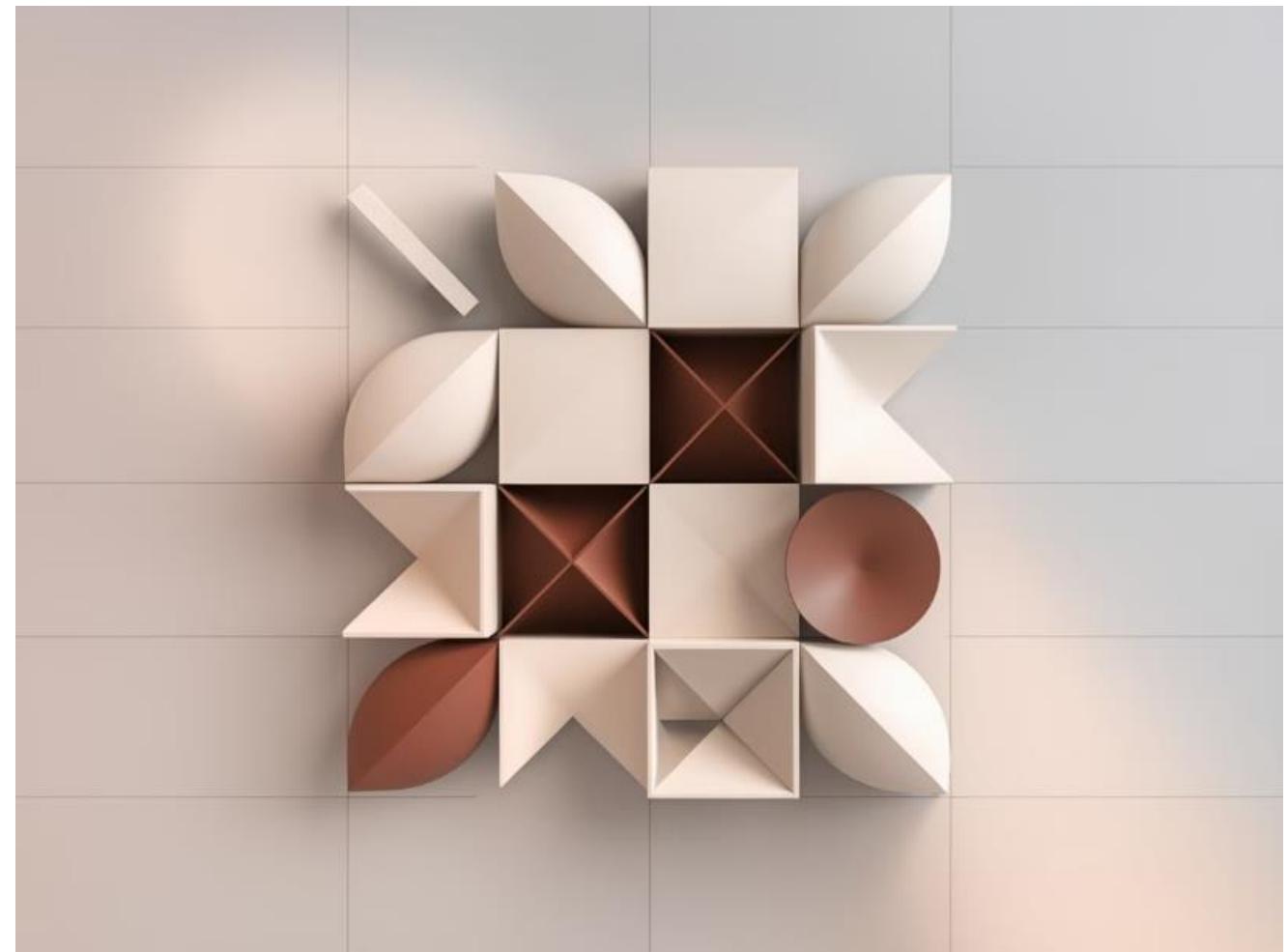
La superposición estratégica de elementos dentro de la cuadrícula agrega profundidad visual y crea composiciones impactantes y dinámicas.

# Plantillas de diseño complejas

## Crear diseños asimétricos y dinámicos

CSS Grid permite crear diseños complejos y asimétricos que se adaptan de manera fluida a diferentes tamaños de pantalla.

Esto se logra combinando diferentes unidades de medida, como píxeles, porcentajes y la unidad "fr" (fracción).

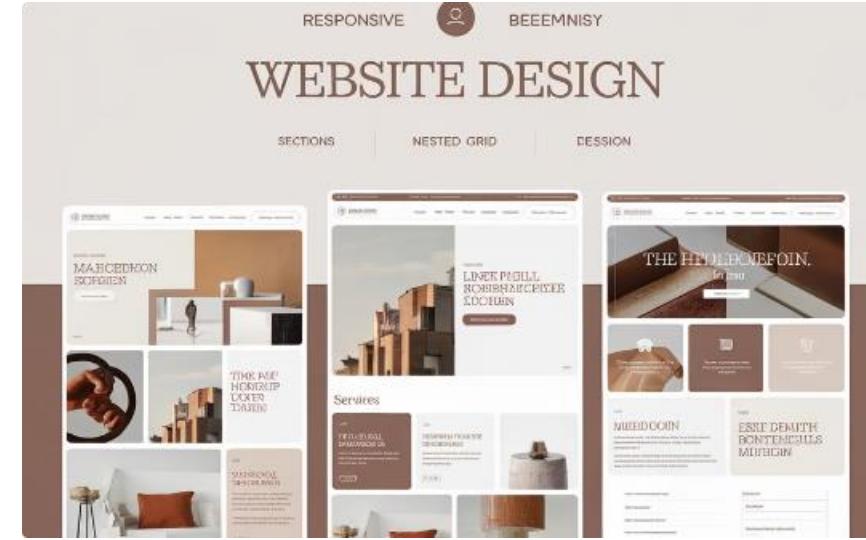


# Anidación de contenedores Grid



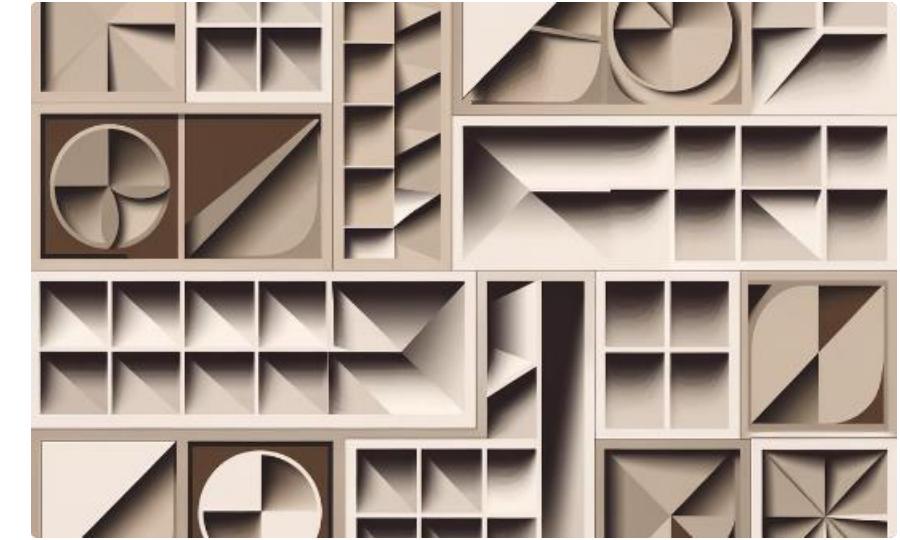
## Layouts anidados

La anidación de contenedores Grid permite crear diseños complejos y dinámicos al anidar grillas dentro de otras grillas, ofreciendo control granular sobre la estructura y distribución de los elementos.



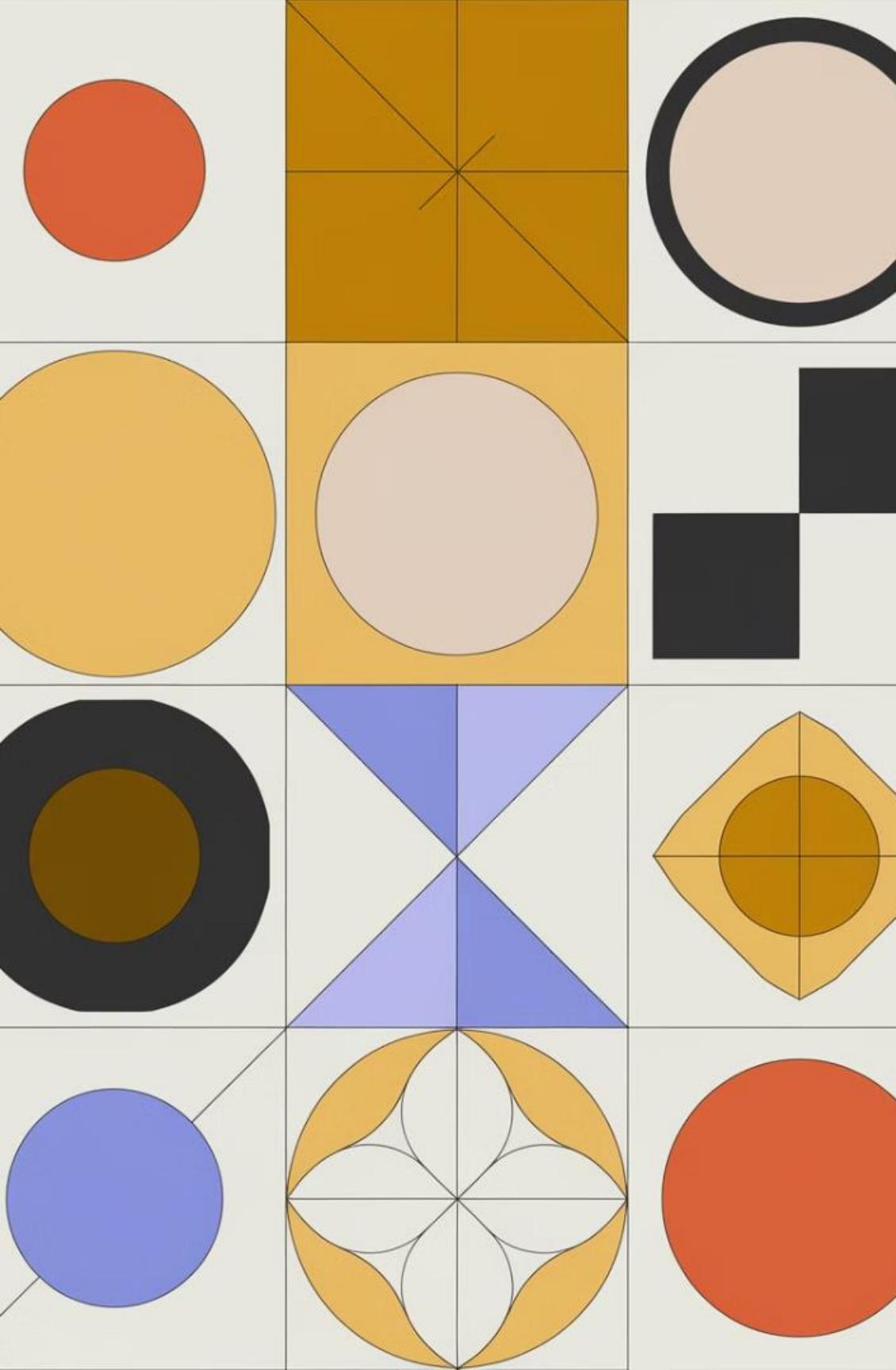
## Adaptabilidad responsive

Al anidar elementos Grid, se puede lograr una mayor capacidad de adaptación a diferentes tamaños de pantalla, optimizando la presentación del contenido en múltiples dispositivos.



## Diseños dinámicos

La anidación de grillas abre la puerta a diseños más asimétricos y creativos, permitiendo superposiciones, juegos de proporción y una mayor libertad compositiva.



# Superposición de elementos Grid



## Sobreposición

La superposición de elementos Grid permite crear diseños complejos y dinámicos, donde los elementos pueden apilarse y ocupar el mismo espacio.



## Orden de apilamiento

Puedes controlar el orden en el que se apilan los elementos utilizando la propiedad z-index, para definir qué elementos están por encima o por debajo.



## Alineación

Puedes alinear los elementos superpuestos utilizando las propiedades de alineación como justify-content y align-items.

# Ejercicio práctico 1: Diseño responsive con Grid

1

## Estructura Base

Definir un contenedor Grid con filas y columnas.

2

## Elementos Adaptables

Ubicar elementos dentro de la estructura Grid.

3

## Breakpoints Responsivos

Ajustar la cuadrícula según el tamaño de pantalla.

4

## Distribución Dinámica

Utilizar unidades flexibles para adaptar el diseño.

En este ejercicio práctico, exploraremos cómo crear un diseño web responsive utilizando la poderosa herramienta de CSS Grid. Aprenderemos a establecer una estructura base con filas y columnas, colocar elementos dentro de ella, y ajustar la cuadrícula de manera dinámica según los diferentes tamaños de pantalla.



## Ejercicio práctico 2: Mosaico de imágenes con Grid

1

### Definir una estructura de Grid

Crea un contenedor con display: grid y establece las propiedades grid-template-columns y grid-template-rows para definir la estructura de cuadrículas.

2

### Agregar imágenes

Inserta las imágenes que quieras mostrar en el mosaico y posíonalas dentro de las celdas de la cuadrícula.

3

### Ajustar espaciado y alineación

Usa las propiedades grid-gap, justify-items y align-items para controlar el espacio entre las imágenes y su posicionamiento dentro de las celdas.