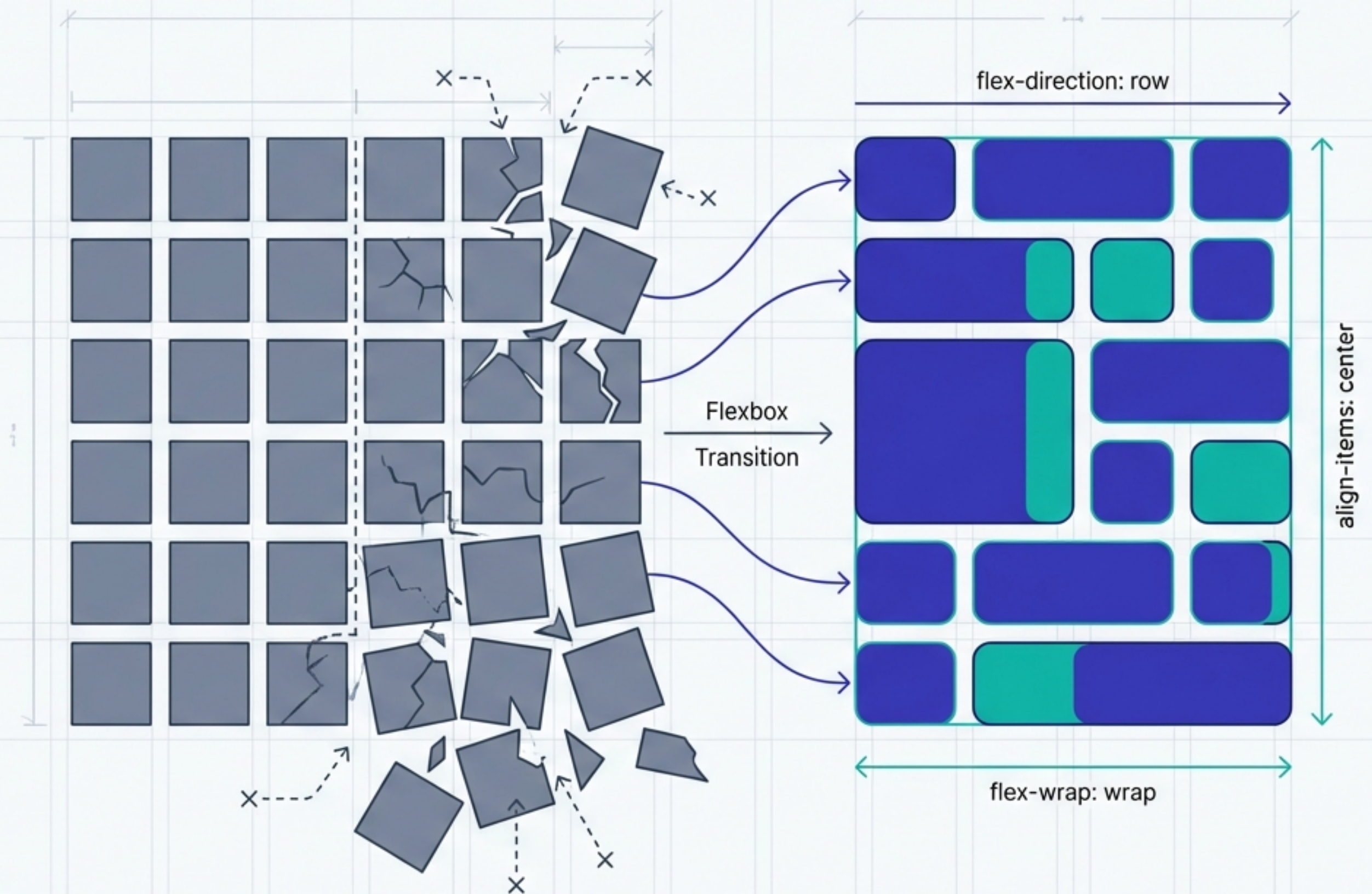


CSS

Unidad 6: Flexbox

Una nueva arquitectura
para diseños flexibles

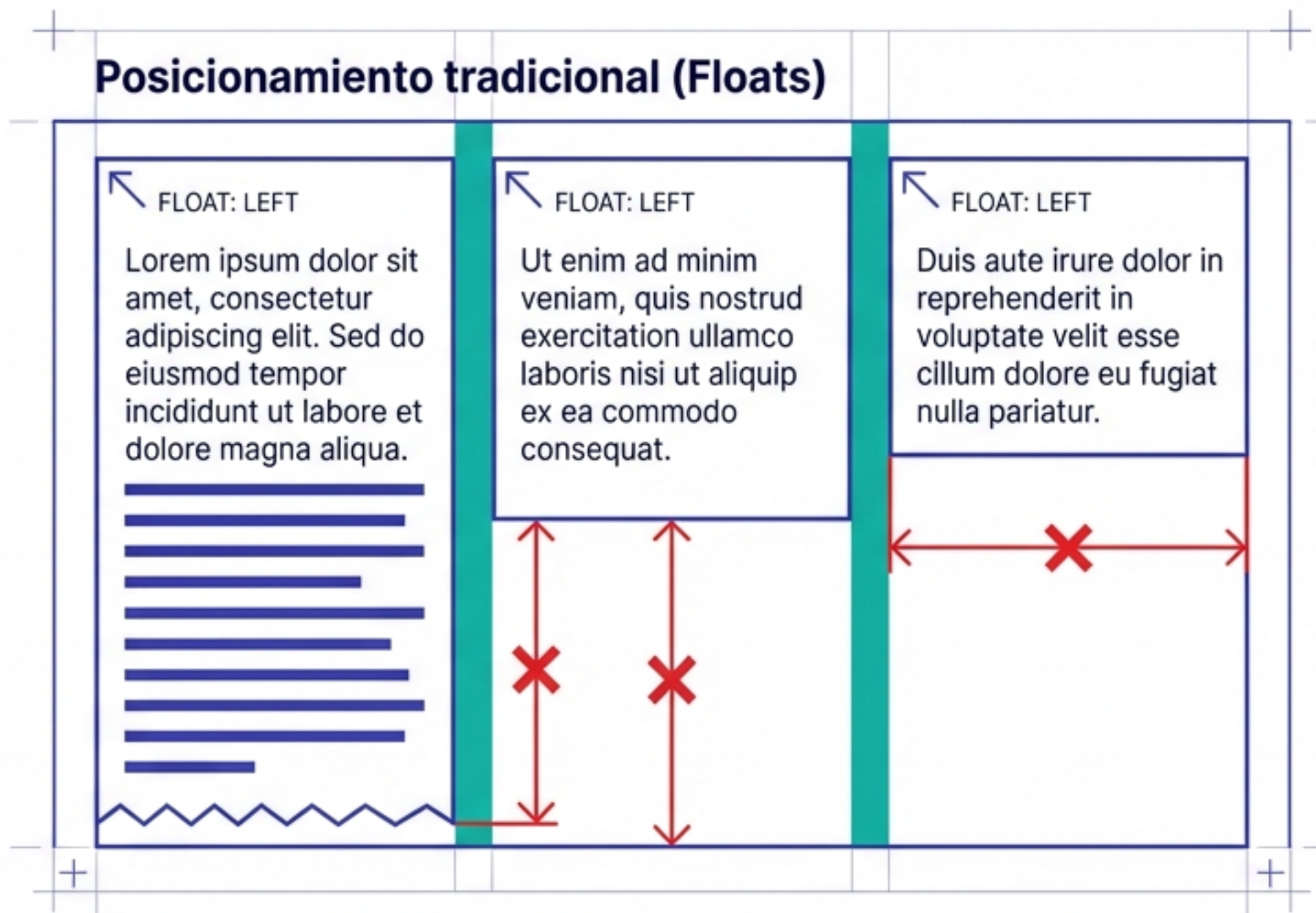


Basado en el material de Fernando Ruiz Rico

Las limitaciones de las herramientas antiguas

Durante mucho tiempo, los 'floats' y el posicionamiento fueron las únicas opciones. Era difícil o imposible lograr requisitos modernos:

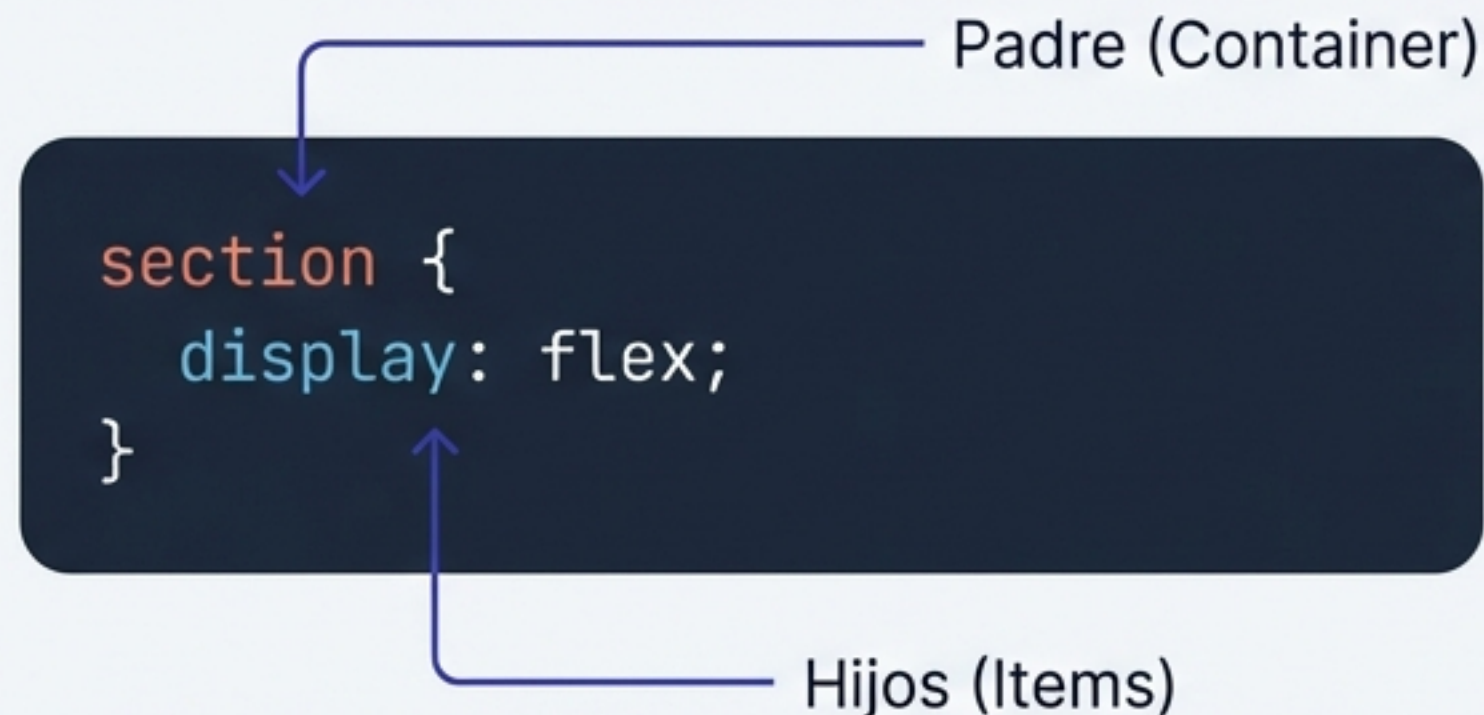
1. **Centrado Vertical:** Centrar un bloque de contenido dentro de su padre.
2. **Igualdad de Espacio:** Distribución equitativa del ancho y alto disponible.
3. **Igualdad de Altura:** Columnas con la misma altura visual independientemente del contenido.



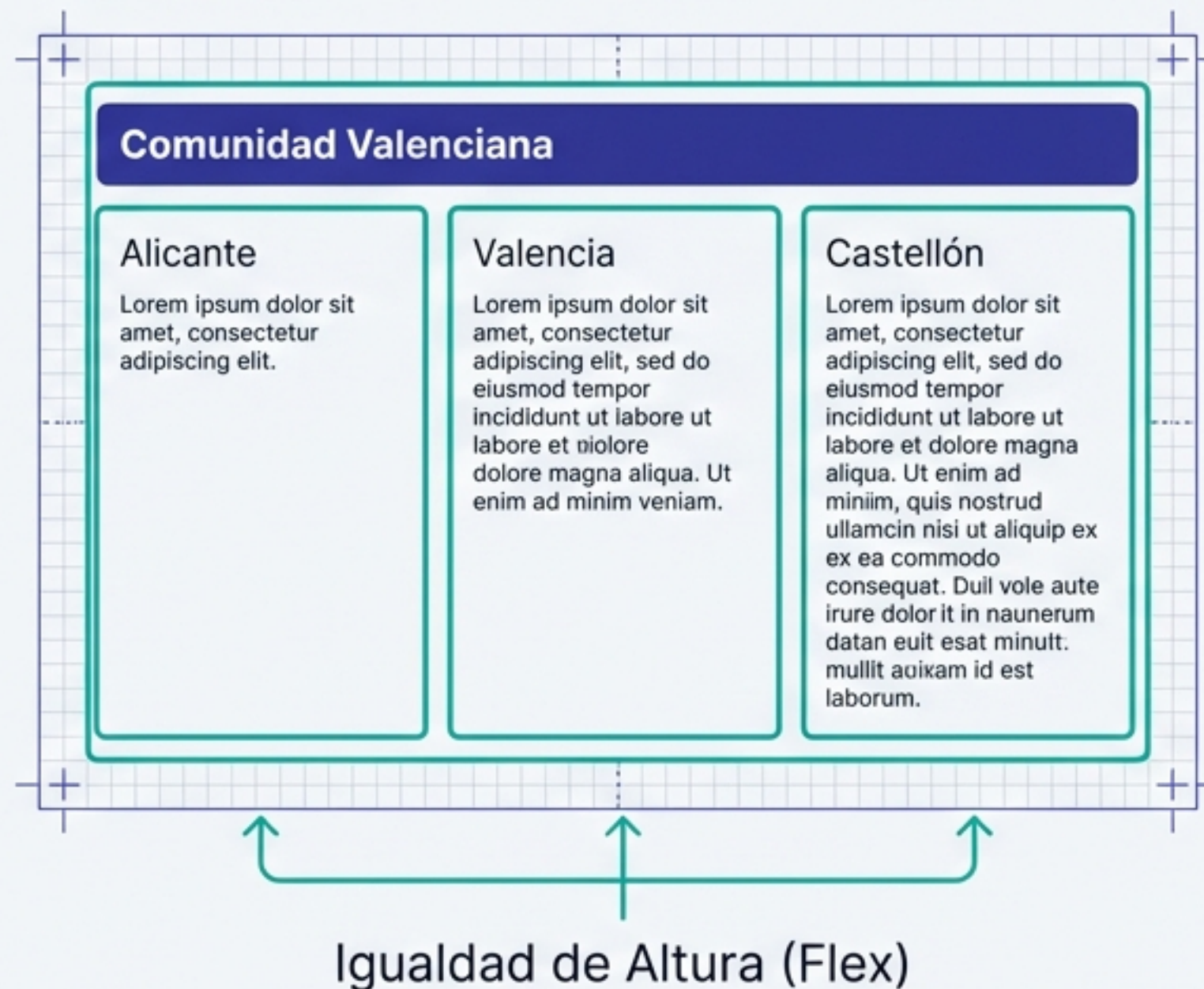
Visualización del layout fallido debido a la falta de control sobre la altura y el alineamiento vertical.

El contenedor flexible

Una sola línea de código convierte al elemento padre en un Contenedor Flex.

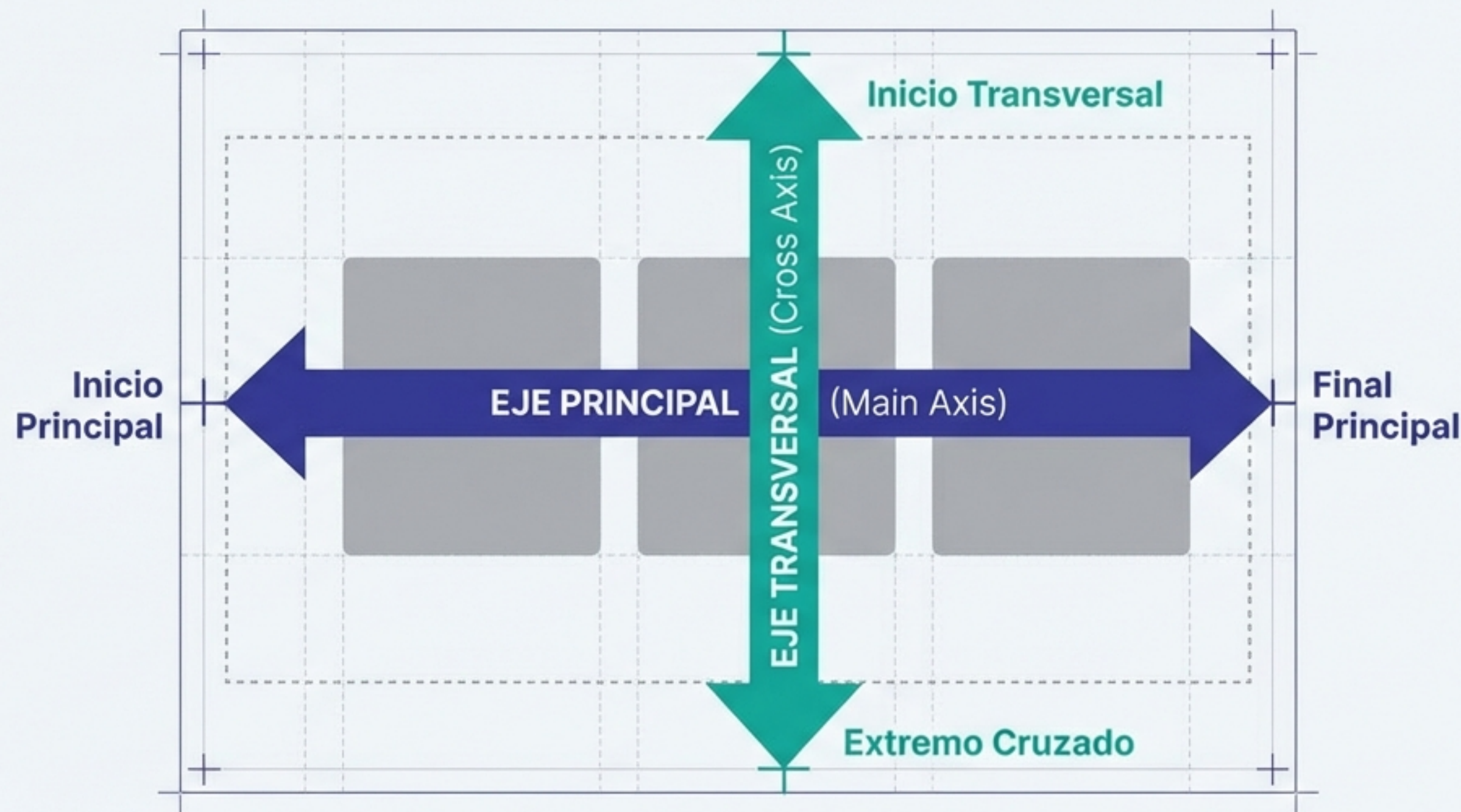


Resultado Renderizado



La geometría del modelo flexible

Los elementos no flotan; se distribuyen respecto a dos ejes invisibles.



Comprender estos ejes es la clave para dominar la alineación.

Controlando la dirección

La propiedad **flex-direction** define el eje principal.

flex-direction: row;
(Por defecto)

flex-direction: row; (Por defecto)



flex-direction: column;

flex-direction: column;

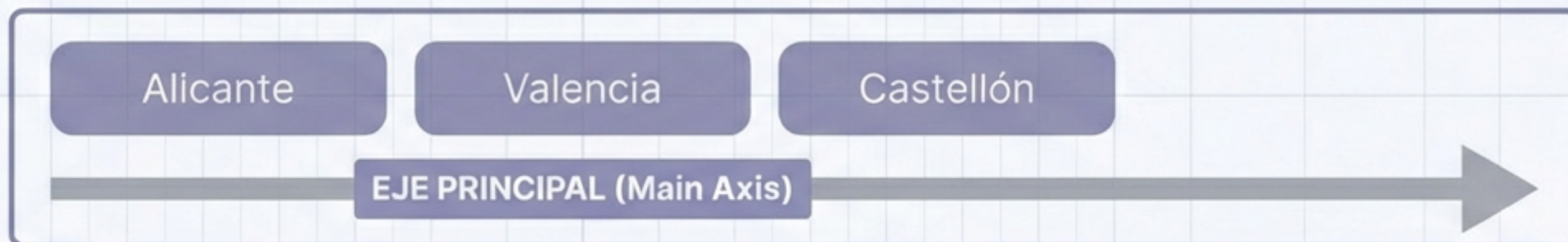


Al cambiar a column, el eje principal rota 90 grados.

Invirtiendo el flujo

```
section { display: flex; flex-direction: row-reverse; }
```

flex-direction: row;
(Por defecto)



flex-direction: row-reverse;



También existe column-reverse, que apila los elementos desde abajo hacia arriba.

El problema del desbordamiento

Cuando el contenido excede el ancho del contenedor.

Sin 'wrap', los hijos se comprimen o desbordan.

nowrap (default)



La solución es permitir el salto de línea.

flex-wrap: wrap;

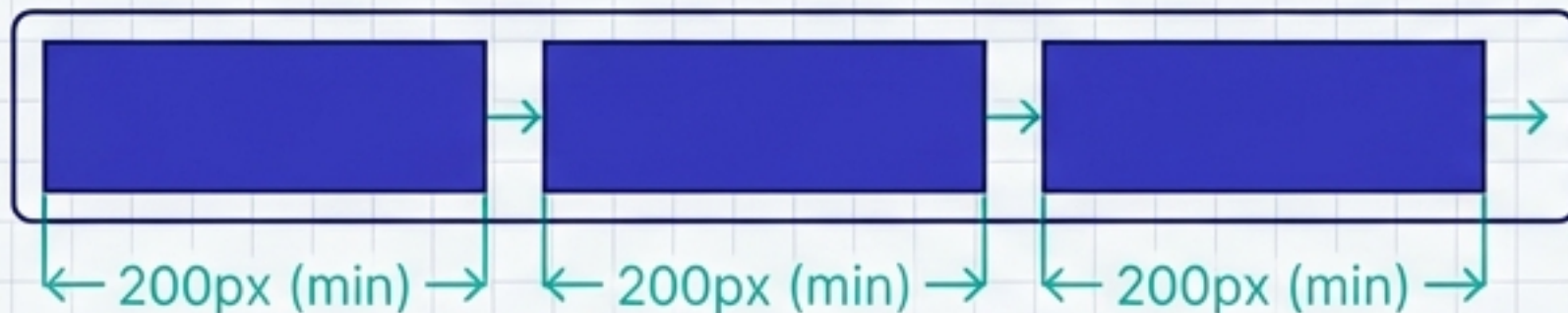


flex-wrap: wrap;

Gestión del tamaño de los elementos

Base Fija

```
article {  
  flex: 200px;  
}
```

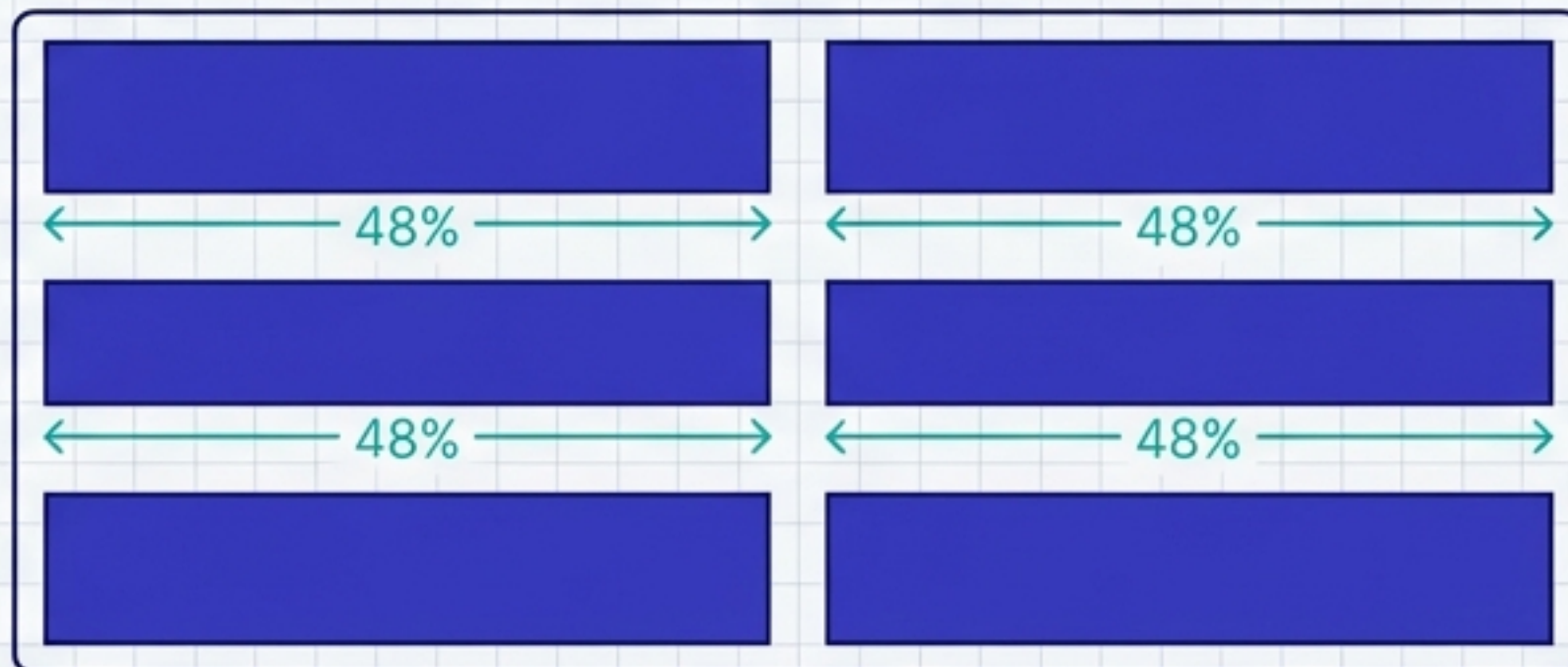


```
article {  
  flex: 200px;  
}
```

Cada tarjeta tiene al menos 200px.

Porcentajes (Grid)

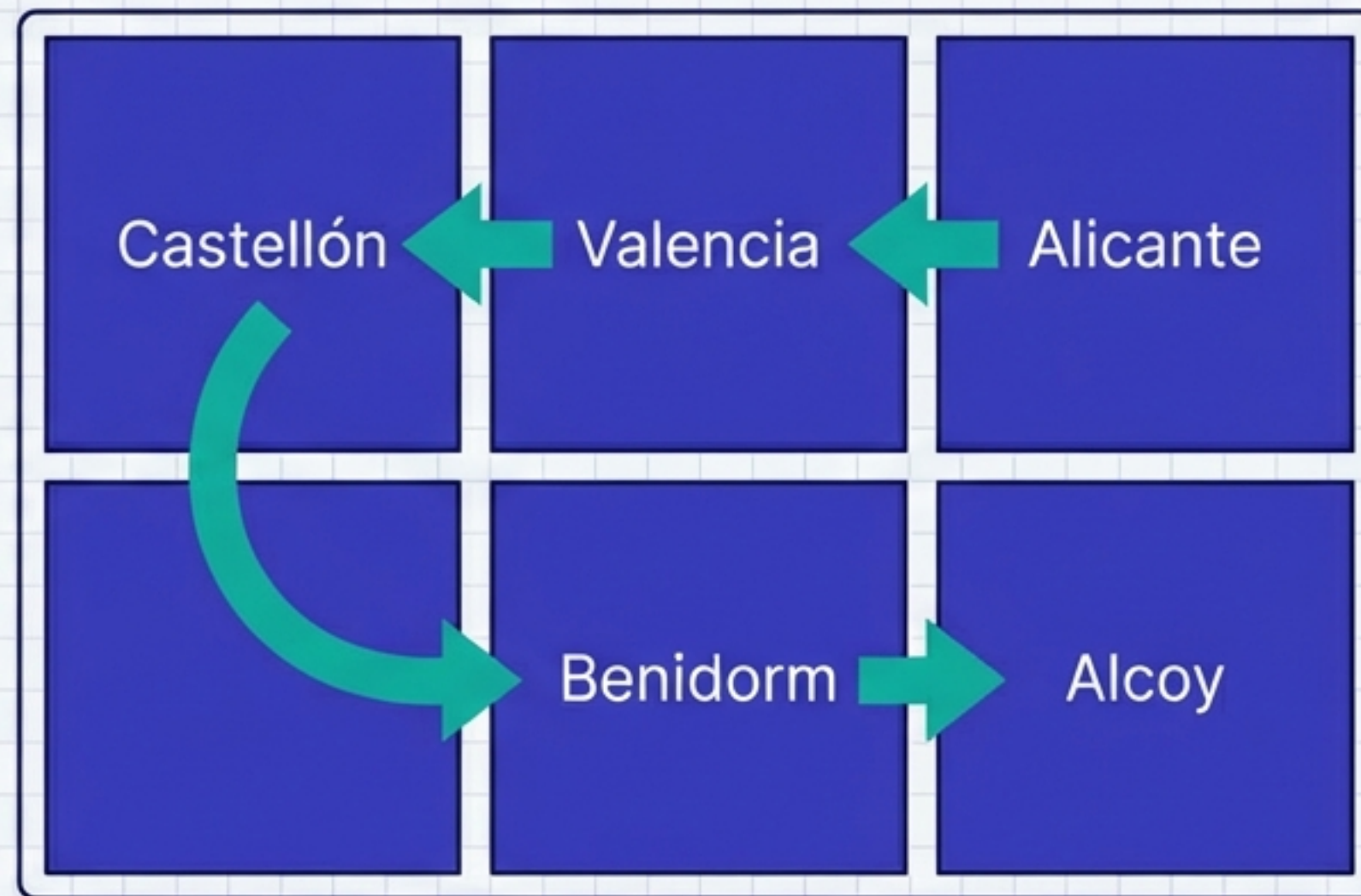
```
article {  
  flex: 48%;  
}
```



Crea una cuadrícula estricta de dos columnas.

Caso de Estudio: Grid Inverso

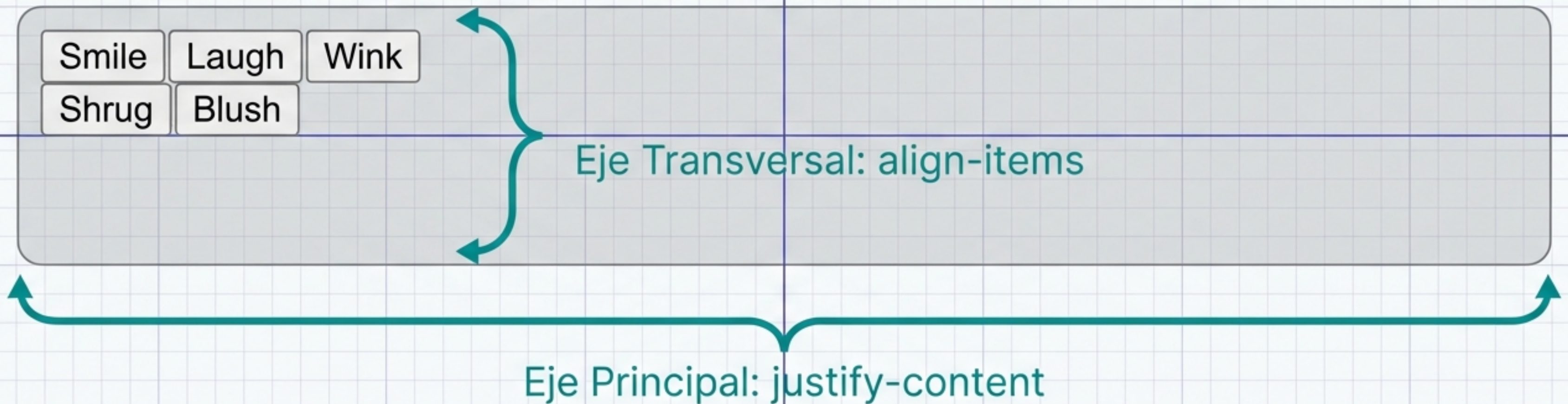
```
section {  
  display: flex;  
  flex-wrap: wrap;  
  flex-direction: row-reverse;  
}
```



El flujo comienza desde la esquina opuesta, pero mantiene la integridad de la cuadrícula.

Alineación y Precisión

Controlando la posición exacta en ambos ejes para UI.



Preparamos el escenario para una barra de navegación perfecta.

Justify Content (Eje Principal)

Define cómo se distribuye el espacio sobrante alrededor de los elementos.

flex-start



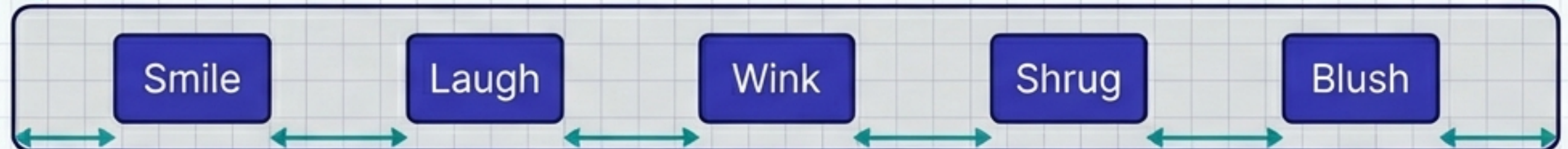
center



space-between



space-around



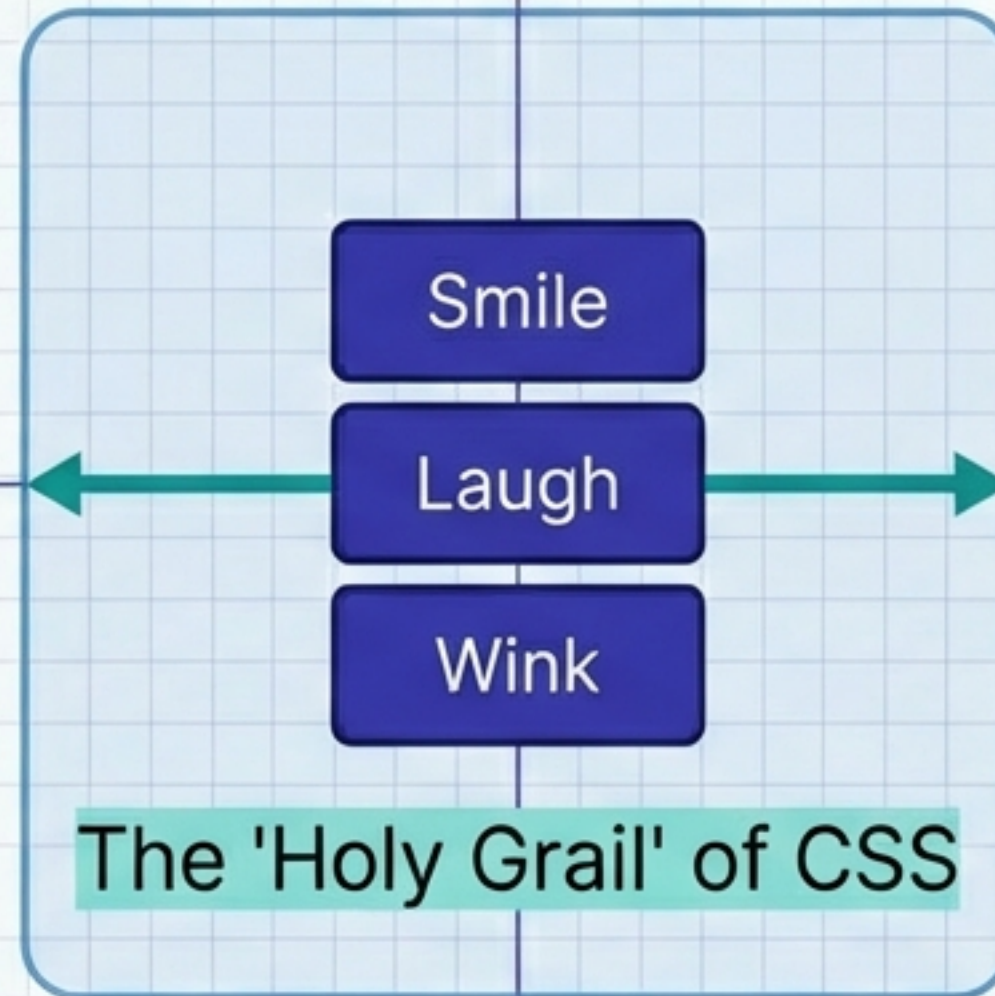
Align Items (Eje Transversal)

Controla la alineación de los elementos a lo largo del eje secundario (vertical).

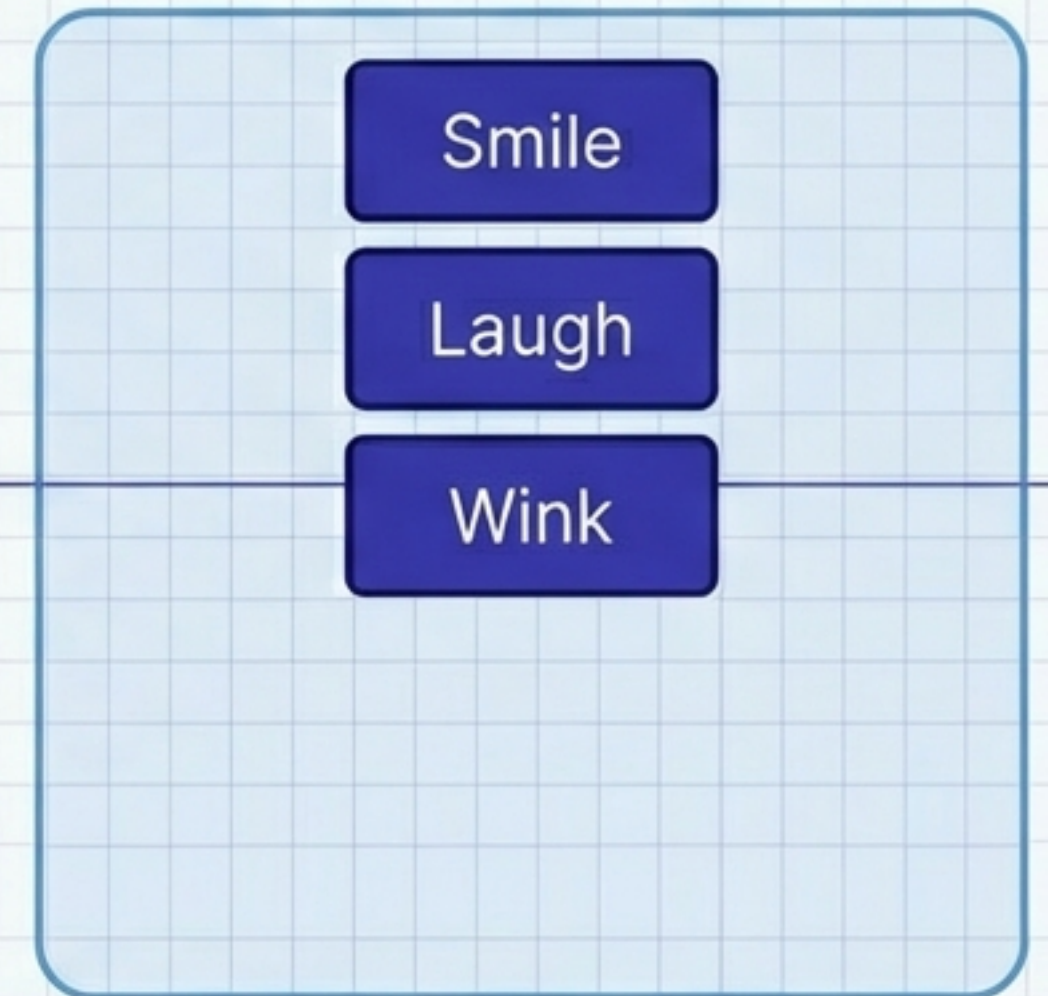
stretch (Default)



center



flex-start

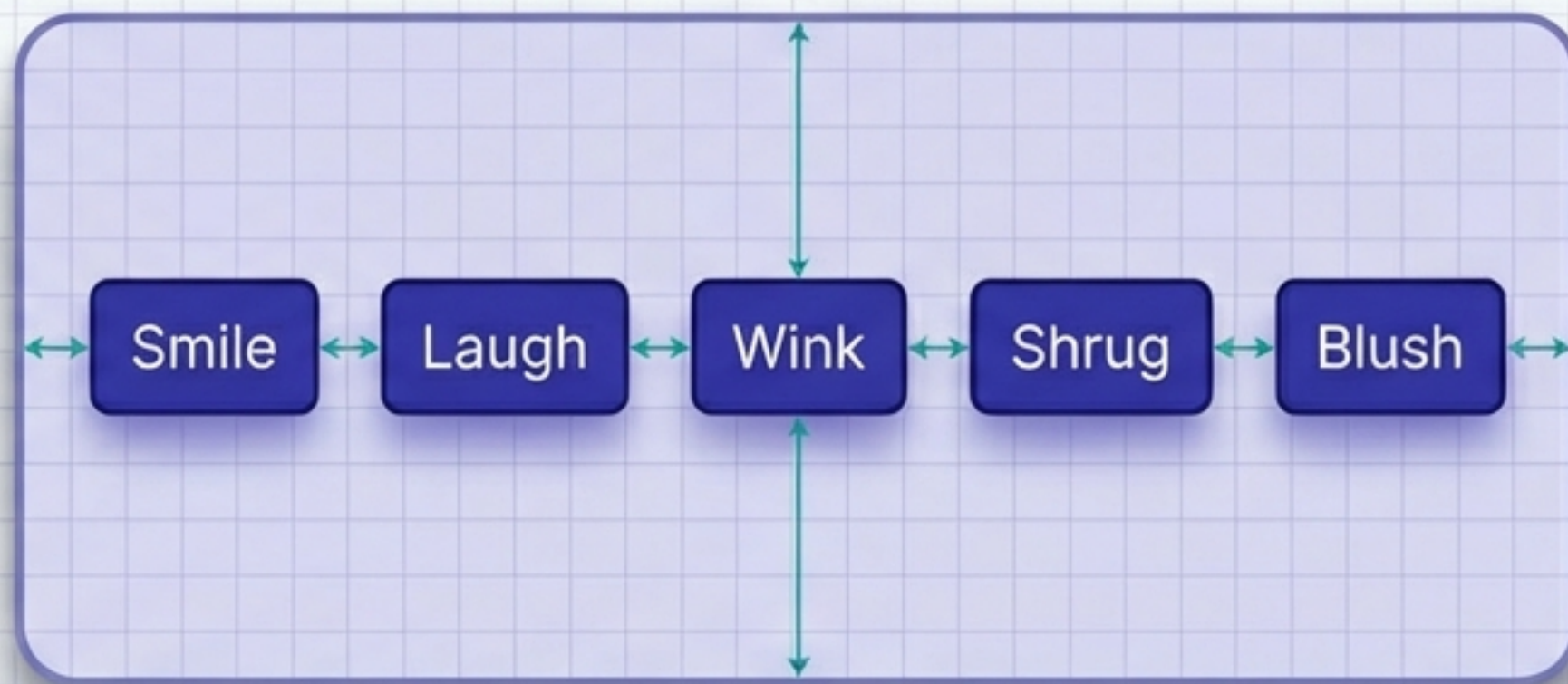


align-items: center es la solución moderna para el centrado vertical.

Síntesis: La barra de herramientas perfecta

```
div {  
  display: flex;  
  align-items: center;  
  justify-content: space-around;  
  height: 100px;  
}
```

Con solo tres propiedades, hemos logrado un diseño robusto y flexible.



Referencia Rápida de Flexbox

Icono	Propiedad	Valor(es)	Descripción
	display	flex	Activa el modo Flexbox.
	flex-direction	row / column	Define el eje principal.
	flex-wrap	wrap / nowrap	Gestiona el desbordamiento.
	justify-content	flex-start / center / space-between	Alineación en Eje Principal.
	align-items	stretch / center / flex-start	Alineación en Eje Transversal.
	flex	200px / 50%	Tamaño del elemento hijo.

Una nueva forma de pensar el diseño

Flexbox no es solo una herramienta de alineación; es un sistema de distribución inteligente. Permite que el contenido dicte el diseño, asegurando que la interfaz funcione tanto en pantallas grandes como como en dispositivos móviles.

“Los elementos flexibles facilitan mucho las tareas de distribución.”

— Fernando Ruiz Rico