EcomLabs

GUÍA PASO A PASO

Container queries en CSS: una tarjeta de producto en cualquier hueco

El HTML y el CSS completos de una tarjeta que pasa de una a tres columnas según el ancho de su contenedor, un script de Playwright para comprobarlo en tres anchos y una hoja de referencia de container-type, @container y las unidades cqi.

PARA	Temas y tiendas con la misma tarjeta de producto en colección, barra lateral y carrito
PROBADO CON	Chromium 151.0.7922.34 · Playwright 1.62.0 · Node 22.22.3 · Windows 11
ACTUALIZADA	12 de noviembre de 2026
DOCUMENTACIÓN	developer.mozilla.org/en-US/docs/Web/CSS/container-type
ARTÍCULO	ecomlabs.dev/g/container
AUTOR	Luis Morales · @luismoralesdev

Índice

Antes de empezar	Qué vas a montar
Paso 01	El HTML
Paso 02	El CSS
Paso 03	Hoja de referencia
Paso 04	Comprobarlo en tres anchos
Al terminar	Checklist

ANTES DE EMPEZAR

Qué vas a montar

Una tarjeta de producto que decide su diseño por el ancho del hueco donde está, no por el de la ventana: una columna en un hueco de 240 px, dos en uno de 480 px y tres en uno de 720 px. Lo probé el 12 de noviembre de 2026 con Playwright 1.62.0 y Chromium 151.0.7922.34, en una ventana fija de 1280 px.

PASO 01

El HTML

La tarjeta vive dentro de un hueco, `.card-slot`. El hueco es el contenedor y lo declara quien coloca la tarjeta (la colección, el carrusel, el carrito):

```
<div class="card-slot">
  <article class="card">
    <img class="card__img" src="/camiseta.jpg" alt="Camiseta de algodón" width="400" height="300">
    <h3 class="card__title">Camiseta de algodón</h3>
    <button class="card__cta" type="button">Agregar</button>
  </article>
</div>
```

PASO 02

El CSS

```
.card-slot {
  container: card / inline-size;
}

.card {
  display: grid;
  grid-template-areas: "img" "title" "cta";
  gap: 0.75rem;
  padding: 0.75rem;
  border: 1px solid #bbb;
}

.card__img {
  grid-area: img;
  width: 100%;
  height: auto;
  aspect-ratio: 4 / 3;
  object-fit: cover;
}
```

```
.card__title {
  grid-area: title;
  margin: 0;
  font-size: 1rem;
}

.card__cta {
  grid-area: cta;
}

@container card (min-width: 24rem) {
  .card {
    grid-template-columns: 8rem 1fr;
    grid-template-areas: "img title" "img cta";
    align-items: start;
  }
}

@container card (min-width: 36rem) {
  .card {
    grid-template-columns: 12rem 1fr auto;
    grid-template-areas: "img title cta";
    align-items: center;
  }

  .card__title {
    font-size: clamp(1.125rem, 3cqi, 1.5rem);
  }
}
```

Es el CSS que ejecuté. Con la fuente base de 16 px, `24rem` son 384 px y `36rem` son 576 px.

PASO 03

Hoja de referencia

PIEZA	QUÉ HACE
`container-type: inline-size`	Consulta el ancho; el ancho viene del contexto, no de los hijos
`container-type: size`	Consulta ancho y alto; colapsa sin tamaño propio
`@container card (min-width: 24rem)`	Consulta con nombre; estiliza a los descendientes, no al contenedor (un `outline` sobre `.card-slot` no se aplicó)
`cqi`, `cqw`, `cqb`, `cqh`	1% del contenedor; sin contenedor caen a la ventana (`10cqi` dio 128 px en una ventana de 1280 px)
Compatibilidad	Baseline "Widely available" desde febrero de 2023, según MDN (12 de noviembre de 2026)

PASO 04

Comprobarlo en tres anchos

Guarda esto como `check-card.mjs`. Abre una página con una tarjeta, cambia el ancho del hueco y cuenta las columnas de la cuadrícula:

```
import { chromium } from "@playwright/test";

const browser = await chromium.launch();
const page = await browser.newPage({ viewport: { width: 1280, height: 900 } });
await page.goto("http://localhost:3000/demo-card"); // tu página con un .card-slot

for (const width of [240, 480, 720]) {
  await page.$eval(".card-slot", (el, w) => { el.style.width = w + "px"; }, width);
  const columns = await page.$eval(".card", (el) => getComputedStyle(el).gridTemplateColumns.split("
").length);
  console.log(width, "px ->", columns, "column(s)");
}
await browser.close();
```

Con una página local de la tarjeta (abierta con `file://`) imprimió `240 px -> 1 column(s)`, `480 px -> 2 column(s)` y `720 px -> 3 column(s)`. Con `@media` en lugar de `@container`, las tres copias dieron tres columnas.

AL TERMINAR

Checklist

- [ ] El contenedor es el hueco, no la tarjeta, y tiene `container: nombre / inline-size`.
- [ ] El hueco tiene un ancho que viene del contexto o de un `width`.
- [ ] Cada `@container` lleva el nombre del contenedor.
- [ ] El diseño base, sin consultas, funciona en una columna.
- [ ] La imagen tiene `width`, `height` y `aspect-ratio`.
- [ ] Cada `cqi` está dentro de un contenedor.
- [ ] Probé la tarjeta en tres anchos de hueco con la ventana fija.