

# Maquetador Web Avanzado

Módulo 1

# Roles

# ¿Qué rol cumple cada uno?

## Diseñador UI/UX

Realiza estudios y prototipos del desarrollo.

Lo hace pensando **en el usuario y debe generar tests**, acciones de estudio, así como **prototipos** pensados para cada dispositivo y contexto.

Cuenta con **conocimientos de UX, accesibilidad y diseño** suficientes para poder realizar tal tarea.

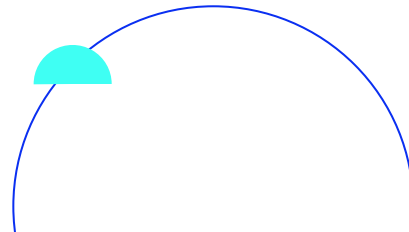


## Maquetador

**Arma el diseño** previamente generado y prototipado por el **diseñador UX**.

Muchas veces, debemos trabajar con herramientas de diseño para poder modificar elementos específicos para cada parte del desarrollo pero **no es nuestra tarea diseñar o prototipar**.

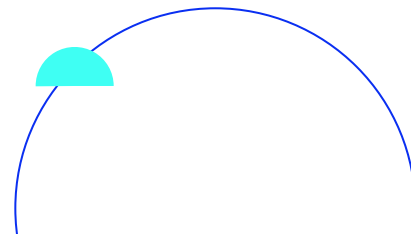
Generalmente su rol, se mezcla con el del **Desarrollador front**. Pero no siempre, depende del proyecto y envergadura de la empresa.



## Maquetador

**Arma el diseño** previamente generado y prototipado por el **diseñador UX**.

Muchas veces, debemos trabajar con herramientas de diseño para poder modificar elementos específicos para cada parte del desarrollo pero **no es nuestra tarea diseñar o prototipar**.



## Desarrollador Front

Se encarga del trabajo con lenguajes como **HTML, CSS y JavaScript** al igual que el maquetador, pero **profundiza en el trabajo con frameworks** de JavaScript tales como **Vue.js, Angular, React.js, Node.js, y MongoDB**.

En muchos desarrollos la tarea de **maquetador la desarrolla quien también trabaja en el desarrollo front**, pero al ser tareas bastante complejas y específicas lo ideal es que si queremos lograr un proyecto de calidad, se separen las tareas.

Si el **desarrollo y la maquetación lo hace la misma persona**, en el presupuesto freelance se debe segmentar el trabajo, ya que no es el mismo. **El costo debe constar por separado y también los plazos de entrega.**

Hoy en día se solicita muchas veces que el desarrollador también trabaje con **back** es decir, base de datos, por ejemplo, manejo de PHP o NodeJS.

# ¿Cómo comenzar?

# ¿Qué es un prototipo?

## Comienzo del desarrollo

Para empezar un proyecto que puede ser una interfaz de cualquier aplicación, desarrollo o un sitio web, necesitamos ya sea nosotros o quien nos da la tarea de realizar el maquetado, **trabajar sobre un prototipo.**

Toda actividad tiene primero un *brainstorming*, y luego se realiza un bosquejo de lo que será finalmente nuestra idea realizada.

En este caso, nuestro proceso, si es que nosotros mismos lo iniciamos, comenzará con un **mapa del sitio**. Esto no es más ni menos que una lista de páginas accesibles para los usuarios.



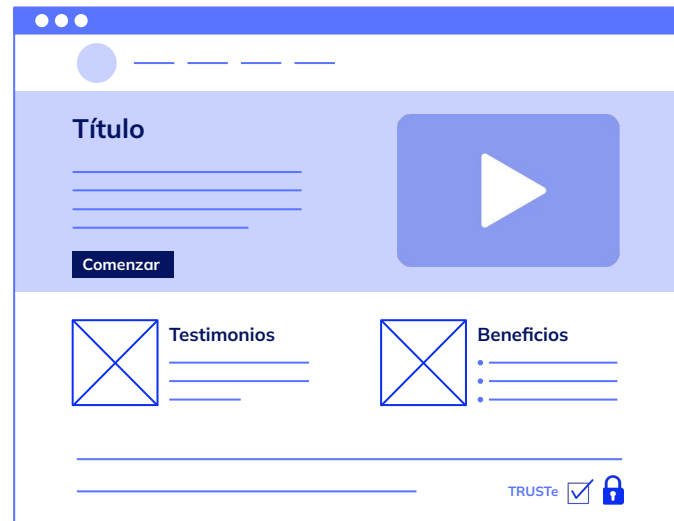


# Prototipos

El prototipo es un ejemplo de interfaz que va recorriendo diferentes niveles de abstracción hasta llegar al diseño final aprobado y que será maquetado por nosotros.

Podemos hablar de ciertos prototipos más abstractos, por ejemplo, los **wireframes**, como podemos visualizar en la imagen de la derecha.

**Nota:** antes de avanzar en la lectura, se recomienda leer el manual “**Prototipado y Roles**” de este mismo módulo.



## Ventajas de trabajar con diseño responsive

Las ventajas de trabajar con diseños responsivos, en sí es que **cualquier usuario puede acceder desde cualquier dispositivo y entender el contenido**, lo cual genera **mayor conversión** ya sea de servicios, o de productos, pero su necesitamos resumir son las siguientes:

- Mejora la **experiencia del usuario** (usabilidad y conversión).
- Mejora **mantenimiento**.
- **Evita contenido duplicado**.
- Mejora el **posicionamiento web**.

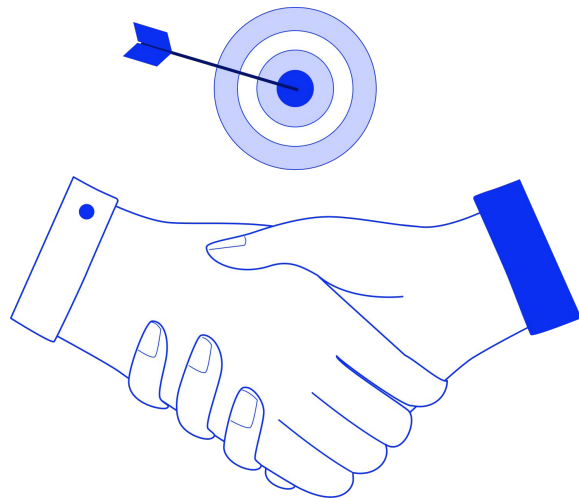


## Diseño y maquetado: trabajo colaborativo

Tanto los diseñadores como los maquetadores estamos interconectados a conceptos comunes, pero el principal, base de todo el maquetado responsivo es el uso de la **grilla de 12 columnas**, que veremos en detalle más adelante.

Otros elementos que conceptualmente deben estar claros para ambos son:

- **Imágenes o multimedia flexible.**
- ***Media Queries.***
- **Tipografías.**
- **Medidas de longitud.**



# Nexo entre el diseñador y el maquetador

En principio nuestro diseñador UX/UI, nos entregará los prototipos esperados para poder trabajar. Para realizar dichos prototipos, **el diseñador inicia su trabajo con una investigación** relacionada, entre otras cosas, con **dos conceptos fundamentales:**

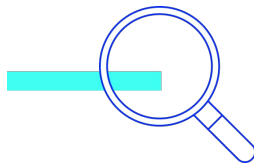
- *User Research*
- *Arquitectura de la Información.*

## User Research

Investigación del usuario (gustos, costumbres, interpretación, cultura, etc.).

## Arquitectura de la Información

También llamada **AI**, permite al diseñador UX/UI entender **cómo organizar la información del sitio** según el público y objetivo de la interfaz, como mencionamos anteriormente.



## ¿Por qué es mejor el diseño responsivo?

Muchas veces, al comenzar el proceso de aprendizaje responsivo, encontramos la falsa ilusión de que el trabajo con % solucionará nuestro trabajo de maquetación.

Sin embargo, si bien el uso de % forma parte del proceso general, **los diseños elásticos, fluidos o líquidos representan una maquetación donde el contenido crece junto al ancho del diseño para rellenar la pantalla.**

Por ejemplo, en televisores las imágenes se estiran mucho y en teléfonos los textos son difíciles de leer, más allá de que conceptualmente estamos simplemente estirando y agrandando elementos sin realmente priorizar el contenido de la interfaz, así como el diseño original de la misma, **deformando todo** solamente para poder hacer qué esta funcione a toda costa en cualquier medio.

Si bien esto soluciona el problema de los textos ilegibles, mantiene todos los demás inconvenientes. Por esa razón pensar un diseño responsivo es salir de la caja y entender que **debemos utilizar una combinación de técnicas orientadas a inteligentemente mejorar la maquetación del sitio o interfaz.**



## ¿Por qué es importante ser responsivo?

Actualmente hay estudios donde sabemos que **la mayoría de los accesos a las web o interfaces se realizan de dispositivos que no son de escritorio.**

Entendiendo, entonces, que la mayoría de los usuarios acceden desde tablets, celulares y otros dispositivos móviles, diseñar para todos ellos es imperativo al momento de pensar una web de la cual esperamos conversiones, ya sean estas hacer conocer nuestra marca, producto, vender o lograr suscripciones.



# ¿Cómo comenzar el camino responsivo?

En el Responsive Web Design el diseño y el contenido se adaptan a cada pantalla, entregando una experiencia de usuario muy similar en resoluciones bajas, altas o en formatos de distintas pulgadas.

## Características a tener en cuenta:

- **Ancho y alto** de la ventana del navegador.
- **Orientación** del dispositivo.
- **Proporción** entre el alto y ancho de la pantalla.
- **Resolución del dispositivo**, es decir, la precisión del detalle en las imágenes de mapa de bits.



# Origen del Responsive Web Design

Si hacemos un poco de historia, es interesante mencionar que el diseñador y autor norteamericano **Ethan Marcotte** creó y difundió esta técnica a partir de una serie de artículos en ***A List Apart***, una publicación en línea especializada en **diseño y desarrollo web**, idea que luego extendería en su libro ***Responsive Web Design***.



Ethan Marcotte

## Content first

Más allá de lograr un **maquetado responsivo**, lo importante, incluso al trabajar con elementos semánticos de HTML es entender que el origen de un proyecto está signado por las ideas u objetivos que nuestro cliente esté esperando de nosotros es fundamental para no perder el foco principal de nuestro trabajo.

**El camino del diseño inicia en una evaluación general de varios elementos que forman parte de la creación o generación de un sitio web.** Por ejemplo, entender a qué target apunta la interfaz.

Cómo estos usuarios interactúan con la página así como también conocer qué conocimientos previos tienen y el *journey* o viaje que -estimamos- harán por nuestra web o *app*.

**Los maquettadores no somos diseñadores,** incluso esos roles están claramente separados en la línea de producción de una interfaz. Sin embargo, **es importante entender la relación entre ambos roles** y cómo generamos el nexo con nuestro diseñador de forma colaborativa.

## Herramientas de trabajo colaborativo

Si bien existen diferentes herramientas de diseño, como Illustrator o Photoshop, en los últimos años fueron apareciendo herramientas más específicas y orientadas al prototipado de interfaces.

Entre las más representativas nos encontramos con **Sketch**, y aunque esta herramienta en principio era solo utilizable desde MacOS, actualmente posee una app web, disponible para los demás sistemas operativos.

Sin embargo, en cuanto a trabajo colaborativo y uso de *plugins* **se vio ampliamente desplazado primero, por AdobeXD, y luego por Figma.**

Entre AdobeXD y Figma las principales características o diferencias son que, en principio AdobeXD no contenía la potencialidad del trabajo colaborativo y, fundamentalmente, es una aplicación paga de la empresa Adobe; mientras que Figma, recientemente adquirida por la misma empresa, continúa teniendo una versión gratuita.



**Figma llevó la delantera en trabajo colaborativo remoto online y hoy se posiciona como una de las mejores herramientas de prototipado.**

Figma no solo tiene herramientas de trabajo colaborativo en cuanto al diseño, sino también en trabajo Agile y Jam.

Esto permite la **integración de diversas áreas dentro del “flow” del desarrollo**, por tanto contiene tanto en su versión web como de escritorio diversas herramientas y *plugins* que extienden funcionalidades muy interesantes a la hora de establecer un nexo entre el diseñador, el maquetador y también otras áreas como el área Back, UX Writing, y de comunicación de la empresa vinculada con la interfaz en cuestión.

Figma contiene herramientas de diseño fáciles de entender, que podemos encontrar en cualquier programa de diseño vectorial, como Illustrator o Corel Draw.

También ofrece una amplia variedad de *plugins*, librerías y templates, así como el **concepto de componente**, que nos permite reutilizar un mismo elemento para múltiples diseños o interfaces.

También brinda un eficaz trabajo con diferentes **frames o tamaños de dispositivo**, lo cual permite al diseñador y al maquetador establecer un nexo directo.

### La **integración con un sistema de grillas**

opcional, así como la cantidad de columnas y filas elegidas le da un marco útil y eficaz para el trabajo en equipo.

**El trabajo colaborativo puede establecerse de manera sincrónica**, brindando una real sensación de cercanía con nuestro equipo.

El sistema de capas, imágenes de posición, tipografías, colores, y formas es típica y familiar, por lo que pasar de otro programa de diseño a Figma será una tarea sencilla y **mejorará la calidad del proyecto y, sobre todo, los tiempos.**



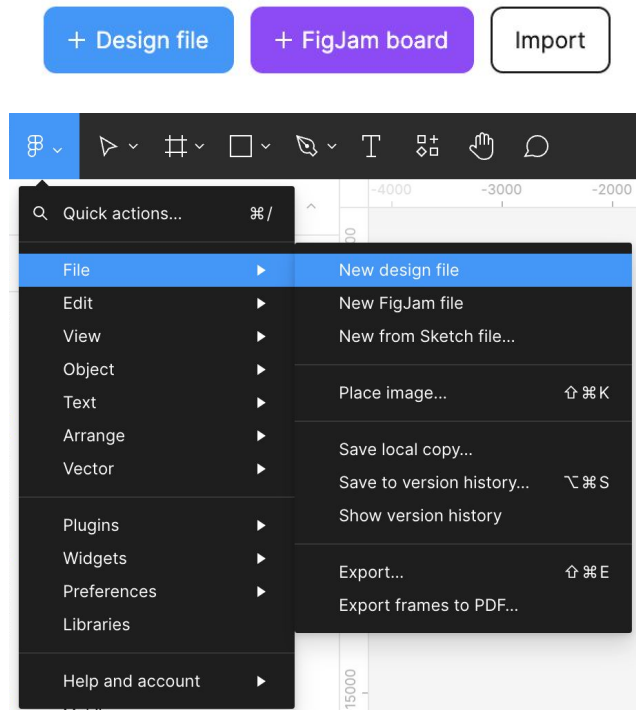
Para saber más sobre esta herramienta recomendamos realizar el curso [UI: Interfaz de Usuario](#), de la Carrera de UX/UI.

También te recomendamos visitar la [página oficial de Figma](#).

# Figma: ¿Cómo comenzar?

## Crear un archivo de diseño

1. Para acceder, debemos ingresar a la página oficial de Figma o descargar su versión de escritorio y loguearnos con nuestra cuenta de e-mail.
2. Luego vamos a elegir la opción **Design file**, para poder acceder a la pantalla principal.

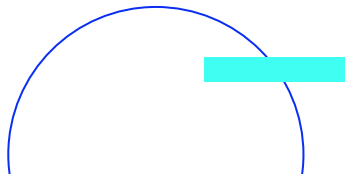
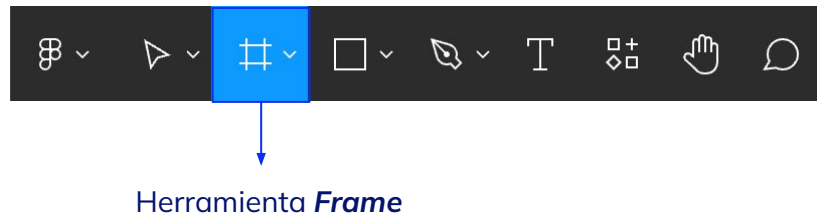


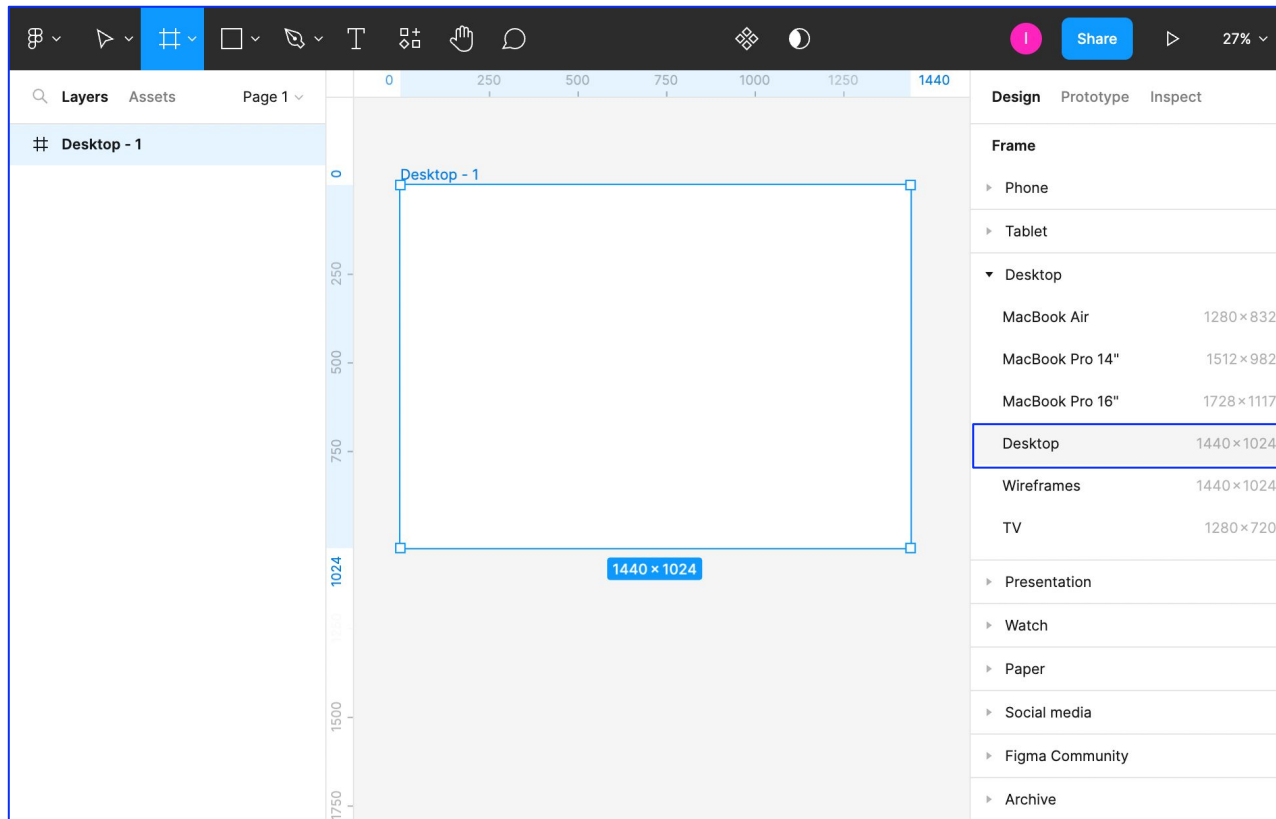
### Crear un frame

3. Allí elegiremos el *frame* o pantalla con la que deseamos trabajar (este proceso es realizado por los diseñadores, pero es interesante mostrar el paso a paso en un contexto de trabajo colaborativo).

Para ello, utilizaremos la herramienta **Frame**, que se muestra en la imagen de la derecha.

En la siguiente pantalla se muestra un ejemplo genérico de pantalla creada al elegir la opción *Desktop* (1440 x 1024) en el panel lateral **Frame**.

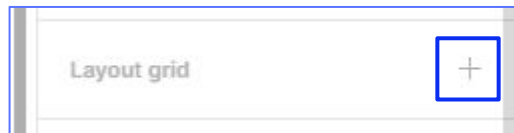




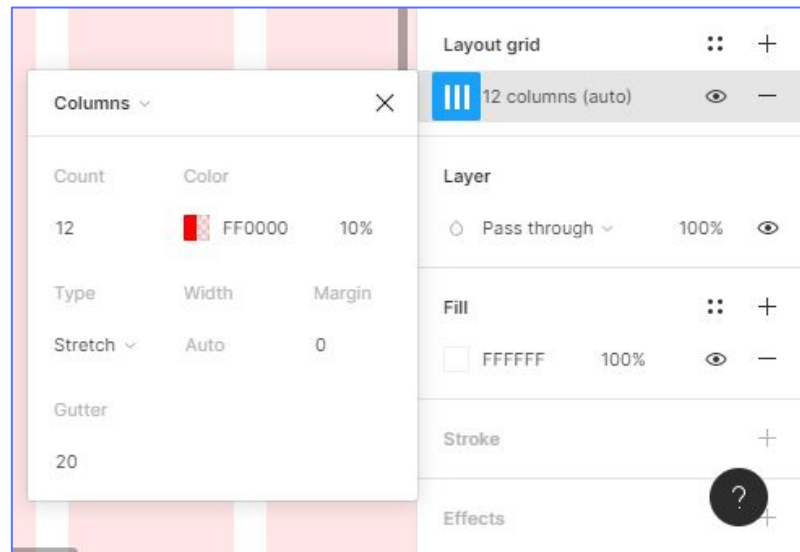


## Crear una grilla

4. Desde el panel de propiedades del *frame*, vamos a elegir **Layout Grid**, como vemos en las imágenes de referencia.



Esto permitirá seleccionar la cantidad de columnas y las demás características deseadas al momento de trabajar. Por ejemplo, decidir si esta grilla tendrá o no espacio (*gutter*) es importante para implementar correctamente las propiedades y valores correspondientes en CSS.



## Tips para el diseño responsivo

**Muchas veces lo mejor que podemos hacer para pensar en “responsivo” es simplificar nuestro diseño.**

Si optamos por ser *responsive* y no *adaptables* (ver contenido anterior), y utilizamos el mismo diseño o template inicial para todos los dispositivos un recurso común es entender que terminarán “stackeandose” o apilando elementos que previamente estaban encolumnados.

También es común utilizar diversos tipos de recursos para rápidamente mejorar la calidad y rapidez con la cual se diseña y se maqueta una interfaz responsiva.

Imágenes que pueden reducir su peso, así como esconder o hacer invisible elementos como publicidades, modales o contenedores que pueden dificultar la visibilidad desde un dispositivo móvil son recursos comunes a la hora de diseñar.

## El menú hamburguesa

Otra idea común en el diseño responsivo, es el **ícono de menú hamburguesa** (llamado así por su forma, donde utiliza tres rayas horizontales para indicar la existencia de un menú principal) en pantallas en general menores a 768px.

**De este modo se evita el trabajo extra de adaptar completamente toda una barra de navegación.** Es decir, el menú que se utiliza es el mismo, solo que cambia su estilo y disposición en pantallas pequeñas.

En definitiva, el éxito de este elemento es que ha sido creado específicamente para adaptarse de forma muy fácil para versiones de dispositivos móviles.



## Entender la grilla CSS

En maquetación se trabaja con una grilla de 12 columnas. Estas 12 columnas equivalen al 100% del viewport.

Por ejemplo, si un elemento toma dos columnas, su width será de 16.66%. Se realizó el siguiente cálculo:

$$\frac{2 \text{ (las columnas que se ocupan)} \times 100}{12 \text{ (total de columnas)}} = 16.66\%$$

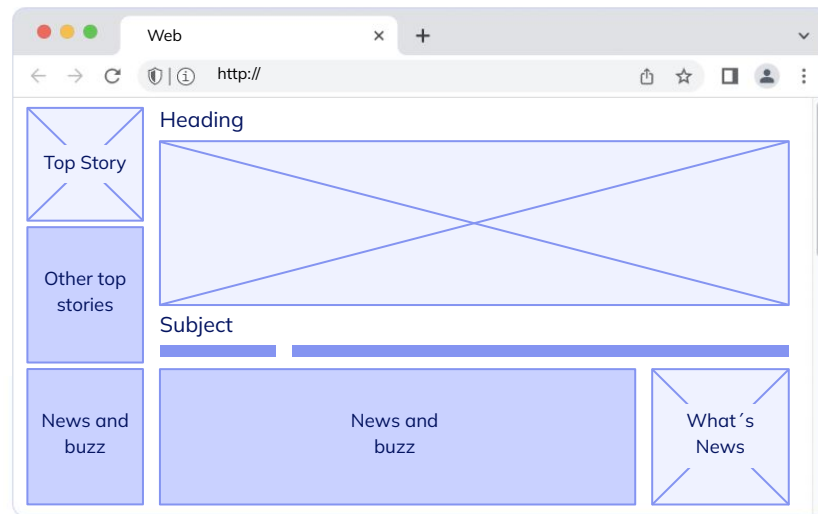
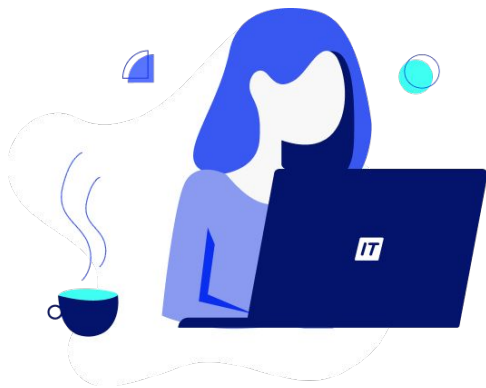
Veamos un ejemplo más claro donde, en el primer caso, el contenido ocupa 3 columnas (25%) y el segundo contenido ocupa 9 columnas (75%)





## Wireframe

El *wireframe* es el **prototipo más básico** para trabajar y nos permite ubicar los elementos que formarán parte de nuestra interfaz.



## Grillas en CSS

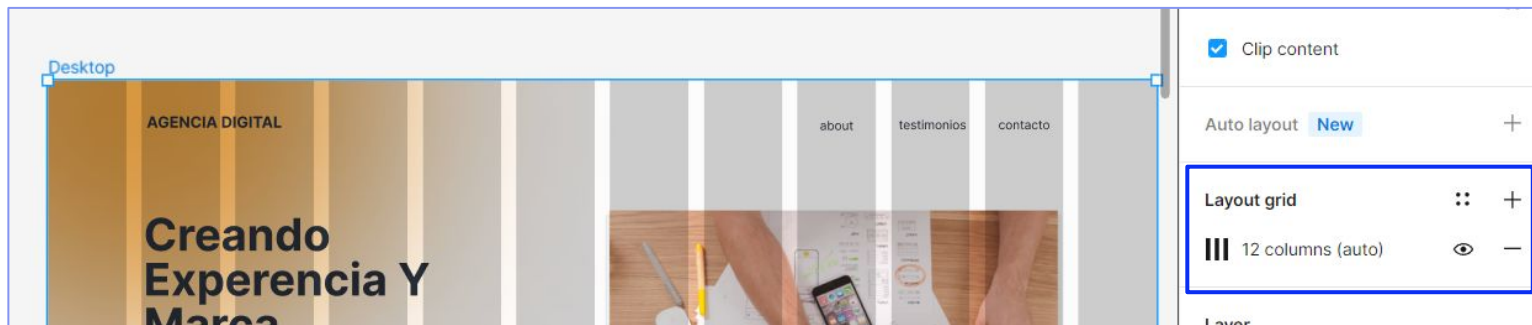
Hagamos un ejercicio de % según la cantidad de columnas, en el caso anterior `.col-3` debería ser implementado al primer elemento y `.col-9` al segundo:

```
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```



## Grillas en Figma

En Figma, podemos **ver las grillas** desde el panel de propiedades de la siguiente forma:



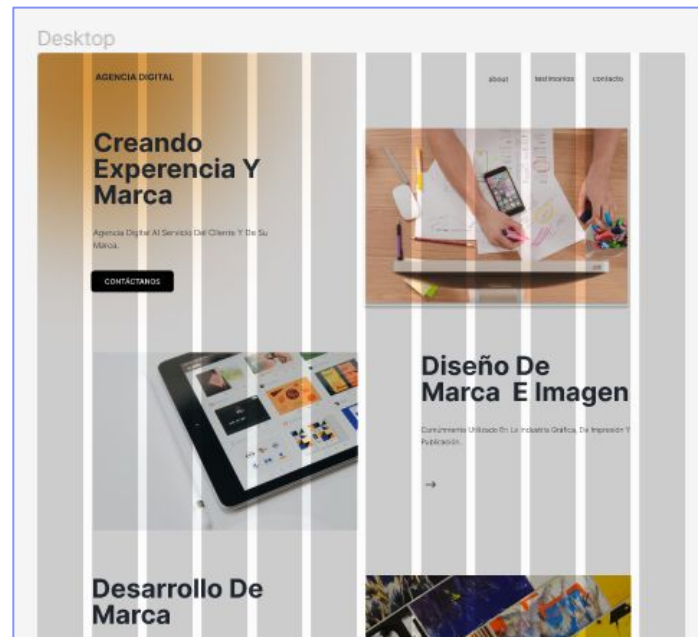


# Trabajo con grillas

## Seguir una grilla

Debemos seleccionar siempre **Layout grid > 12 columns > Auto** para poder lograr la grilla correcta dado que existen otros formatos pero no son los actualmente trabajados en maquetación.

Actualmente, se utiliza esta grilla para poder ubicar cada elemento en nuestro *layout*.



# Dev Mode o Modo de Desarrollo desde Figma

## Trabajo con Dev

Si bien existían herramientas o extensiones similares en Figma, actualmente se puede trabajar de forma simple (a través de un *slider*) con un Dev Mode. **Se asemeja a un inspector de elementos** (el que activamos con *F12* en nuestro navegador) que brinda datos muy interesantes al momento de hacer nuestro CSS.

Es importante entender que **el Dev Mode se utiliza para conocer más sobre el diseño** y brindarnos una herramienta de **ayuda en el proceso de interpretación del diseño**, pero no

con el propósito de automatizar el trabajo con CSS en un *copy/paste* mágico.

Es fundamental tener en cuenta que posee “errores”, por así llamarlos, dado que estima ciertas propiedades o valores que deberían implementarse pero no tiene el discernimiento necesario para interpretar al diseñador como podríamos hacerlo nosotros.

## Trabajo con Dev

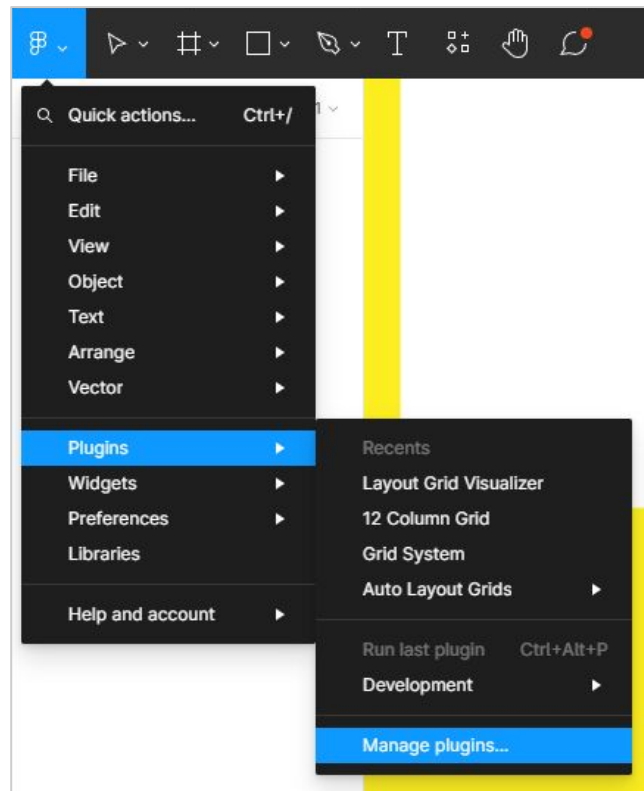


## Uso de plugins en Figma

Existen *plugins* para trabajar en Figma. Solo debemos ir al menú del lado izquierdo y buscar la opción *plugins*.

En este ejemplo, mostraremos cómo buscar un *plugin* llamado **Layout Grid Visualizer**. Este plugin permite también generar grillas en nuestros diseños.

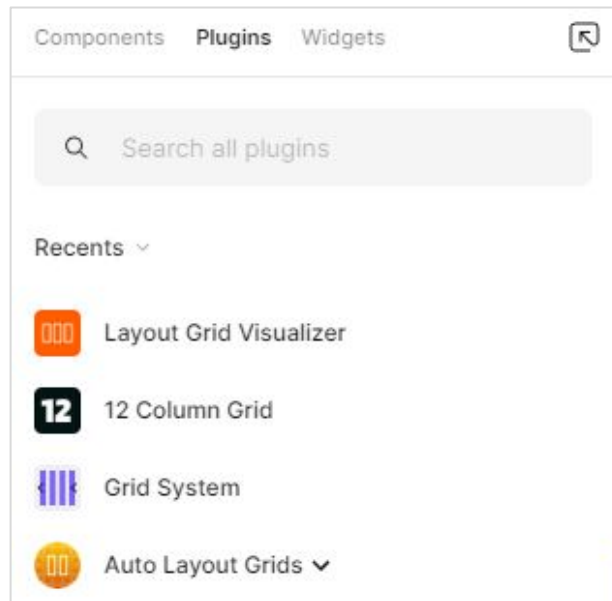
Para encontrar un *plugin*, buscaremos en el menú la opción **Plugin** y luego seleccionaremos **Manage Plugins**.



Escribiremos el nombre del *plugin* que estemos buscando, lo seleccionaremos y podremos activarlo (*run*) en nuestro Figma para utilizarlo.

**Nota:** revisa los tutoriales oficiales de Figma para conocer más sobre sus herramientas y usos. También, te sugerimos realizar el curso de [UI: Interfaz de Usuario](#) e [Introducción a UX](#) si deseas profundizar más en el diseño de Interfaces Web.

**Link oficial:** [Figma - YouTube](#)



**¡Sigamos  
trabajando!**