***Notas del Curso.***

***Curso de Sistemas de diseño.***

**¿Qué es un Sistema de Diseño?**

Los sistemas de diseño están en todas partes, su función tiene que ver con la planeación y la construcción de productos *(no necesariamente digitales)*.

Hace muchos años, un hombre llamado Malcom McLean transportaba mercancías en sus barcos, pero la descarga de todos estos objetos era muy compleja, todo tenía tamaños y pesos distintos. Para resolver este problema, McLean trabajo con un grupo de ingenieros y crearon los *containers* o contenedores, cajas de metal que agilizan este proceso gracias a la estandarización de todos los sistemas y medios de transporte de estos contenedores.

Esto mismo sucede cuando trabajamos con software, buscamos una manera de agilizar nuestros desarrollo estandarizando todos estos procesos. Los sistemas de diseño son conjuntos de reglas para un producto que se modifica dependiendo de las necesidades del mismo, son una decisión de negocio. Cualquier persona del equipo de producto debe estar involucrada en este proyecto y poder consultar toda esta documentación.

Para crear un sistema de diseño podemos usar diferentes herramientas como **Bird**, **Notion** o **Evernote**, lo importante es que podamos escribir y organizar estos sistemas.

Partes importantes de un Design System:

* **Principios**: marcan la pauta y las reglas de nuestro producto
* **Sistema**: Cual va a ser el flujo de creación (tiempos y forma)
* **Foundations**: Bases de todo lo visual
* **Componentes**: Partes que forman un todo

**Principios del Sistema de Diseño**

Los principios del diseño son la base fundamental de los paradigmas y metodologías que aplicaremos en nuestro sistema. Todas las decisiones que tomemos deben tener en cuenta y fundamentarse en estos principios, así evitamos cometer errores o generar conflictos, ya que todos los caminos serian correctos.

Principios del sistema de diseño:

* **Accesibilidad**: Nuestro producto debería ser usable para cualquier usuario, debemos construir productos **perceptibles**, **operables**, **entendibles** y **robustos**.
* **Consistencia**: Todo el equipo debe tener los mismos objetivos y deben hablarle de la misma forma a los usuarios, sea visualmente o por escrito.
* **Reusabilidad**: Cada hora que invirtamos trabajando en algún componente debe servir para construir otros componentes y evitar que tu o alguien más tenga que reconstruir nuestro trabajo.
* **Shareable**: Debemos construir y trabajar con herramientas que nos permitan compartir todo nuestro trabajo.

Podemos añadir otros principios que se adecuen correctamente a nuestra empresa, pero debemos tener cuidado de que todos estos principios sean compatibles entre sí.

Por ejemplo, los principios de ***User Control*** *(el usuario debe tener la sensación de control sobre el producto)*, ***Forgiveness*** *(debemos permitir que el usuario pueda cambiar de opinión o volver a empezar algún proceso)* y ***Perceived stability*** *(aún con plataformas robustas y complejas, el usuario debe percibir los procesos tan simples y familiares como sea posible)*.

[***https://principles.design/examples/co-op-design-principles***](https://principles.design/examples/co-op-design-principles)

# Niveles de Sistematización

El **Design System Workflow** nos ayuda a entender mucho mejor cómo entra, por dónde sale y por cuáles puntos pasa nuestro trabajo cuando trabajamos con sistemas de diseño.

También vamos a aprender sobre **Building Design Systems**, donde construimos herramientas para que todos puedan usar e implementar las guías y los estilos que estamos trabajando. Para organizarnos y sistematizar estos procesos podemos implementar el modelo solitario, dónde tú, el diseñador haces todo el trabajo para que alguien más lo consuma, pero existen otros modelos como el centralizado o el confederado donde trabajamos con otras personas o incluso otros equipos y pedimos Feedback dependiendo de la organización de nuestras empresas.

Los sistemas de diseño tienen niveles y cada nivel abarcan diferentes partes del equipo.  
Un sistema de diseño es un equipo que convierte lo que se está pensando se convierta en software. No importa todo lo que tenemos en ui kit, wireframe, prototipo, si al final no es creado en software.  
En un sistema de diseño intervienen todo tipo de perfiles (Ux, User research, frontend, backend, etc).

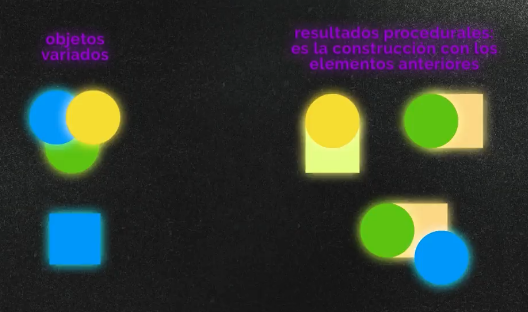
**Design System Workflow**  
En este punto me parece interesante trabajar junto al equipo de “Procesos” de la empresa ya que tendremos que definir el Workflow de trabajo, desde que llega la necesidad hasta que la solución se convierte en Software. tendríamos un modelo Centralizado.

**Building Design System:** En este punto es darle la capacidad a los demás para poder construir. Para organizarnos podemos utilizar el **Modelo solitario** nos puede ayudar a impulsar de manera más rápida la construcción dentro de nuestros equipos. Consiste en que una persona construye y los demás consumen.  
También tenemos el\*\* Modelo Centralizado\*\* este modelo es conveniente cuando contamos con un equipo.

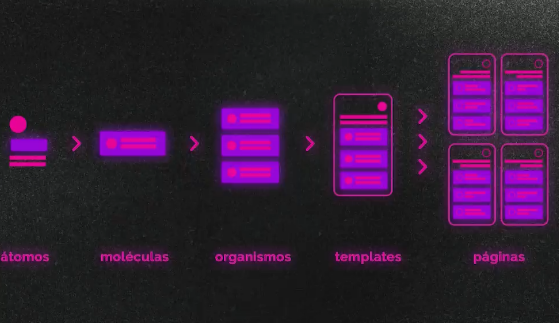
**Modelo Confederado**, contamos con equipos de otros departamentos y todos construyes el sistema hacia dentro.

# Paradigmas: Diseño atómico, diseño procedural y DRY

**Procedural:** Permite la construcción de funciones que a futuro se van a reconstruir a sí mismas. Se basa en objetos ya hechos para obtener resultados diferentes. Por ejemplo, cuando se usa código se utiliza el concepto procedural.



**Atómico:** Le da una organización a los elementos: átomos, moléculas, organismos, templates, páginas. Es decir, ir desde los elementos más sencillos como las etiquetas más básicas, hasta los elementos mayores que corresponden a la unión de varios átomos, moléculas, organismos y templates para obtener una página.



**DRY:** Don’t Repeat Yourself. No repetir los mismos elementos.

**¿Qué es un componente?**

Los componentes forman parte de un todo, así como las velas de un pastel, el piso de los edificios o las partes de un motor cada una con su funcionalidad. Lo mismo pasa con las interfaces, vamos a construir diferentes elementos y herramientas para que los usuarios logren cumplir sus objetivos.

Todos los componentes tienen una entrada y una salida, el usuario realiza una acción y los elementos deben responder de alguna forma *(Feedback)*, esta es la forma de comunicamos con los usuarios, trabajamos haciendo conversaciones para informar que todo esta funcionando correctamente y qué pasos deben seguir a continuación. Este proceso de comunicación lo conocemos como **interacción**.

Partes de un componente:

* **Nombre** *(así evitamos diferentes definiciones y establecemos los objetivos y funciones de nuestros componentes)*
* **Descripción** y solución *(en qué problema estamos trabajando y cómo deberíamos implementar estas soluciones)*
* ***Behavior*** *(el comportamiento de nuestros componentes dentro del sistema)*
* **States** *(las variaciones y distintos comportamientos que pueden tener nuestros componentes dependiendo de su contexto)*



**Foundations**

Los sistemas de diseño son un conjunto de reglas que organizamos con nuestros equipos, y los fundations son las partes más básicas que podemos configurar en nuestro sistema. Vamos a repasar cada una de estas bases mientras escribimos la documentación en Notion:

* Tipografía
* Colores
* *Layout* y *spaces* *(son las formas y espacios fundamentales que utilizamos para ordenar los elementos de nuestro sistema)*
* Iconografía
* *Styles (nuestra marca puede presentarse con estilos juguetones o realistas, lo importante es definir tan claro como sea posible qué intentamos transmitir)*
* Tono *(con qué personalidad o de qué forma debemos hablar con nuestra audiencia)*

# Tipografía

Cuando trabajamos con tipografía para software debemos tener en cuenta las implicaciones que estas pueden generar si trabajamos para dispositivos móviles o los requerimientos técnicos de alguna pantalla en particular. Un buen lugar para encontrar tipografías listas para el desarrollo y diseño de nuestros productos es [Google Fonts](https://fonts.google.com).

Para definir y clasificar las características y peculiaridades en los elementos de nuestro diseño podemos basarnos las etiquetas de HTML para títulos y encabezados (*H1*, *H2*, *H3*, *H4*, *H5* y *H6*), párrafos (*p*) y párrafos más pequeños (*small*). Podemos definir que los títulos se trabajen en negrita y con tamaños de fuente más grandes, lo importante es que estas reglas se acomoden al sistema de diseño que estamos trabajando

# Bonus: Cómo evitar colores contrastantes.

Cuando utilizamos colores demasiado contrastantes podemos generar efectos visuales algo molestos, problemas en el render o líneas blancas y negras en los bordes de los elementos. Para solucionar estos problemas visuales debemos elegir nuestros colores con mucho más cuidado, moviendo los colores hacia alguna tonalidad para evitar colores demasiado contrarios.

# Paleta de colores.

Los colores también deben llevar algún tipo de clasificación ya que definen el estilo visual de nuestro sistema. La paleta de colores define el estilo de tu sistema de diseño y ayuda a definir el sistema de diseño, diferenciando la identidad de la marca y del resto de efectos y elementos visuales.

Muchos sistemas de diseño clasifican sus colores según sus productos (un color para ventas, otro para servicio al cliente, etc), otros utilizamos Material Design y clasificamos los colores como actions colors, \_secondary colors \_ y otros niveles de clasificación. En realidad, no importa, podemos tomar cualquier otro paradigma de fundamentos de color para nuestros sistemas, lo importante es clasificar el uso de estos colores y asegurarnos de que todo el equipo los entiende.

Para escoger estos colores debemos tener en cuenta los colores de la marca, los colores principales (colores de acción o call to actions), colores secundarios, los grises y los colores de fondo. El trabajo se puede complicar un poco cuando tenemos en cuenta todos estos colores, pero el resultado será mucho mejor, los colores funcionaran correctamente entre todos.

# Reglas de espaciado

Las reglas de espaciado son muy útiles cuando trabajamos nuestros sistemas de diseño para anticipar la visualización de nuestras interfaces en diferentes dispositivos.

* **Ritmo**: Qué tan seguido aparecen elementos arriba o debajo de otros.
* **Padding**: Es el espacio dentro de nuestros elementos, nos ayuda a enfocar y transmitir los estilos de nuestra marca.
* **Margin**: Es el espacio entre elementos pero hacia afuera, nos permite generar ritmo visual o para separar elementos de secciones diferentes.
* **Border**: Trabajamos con el borde de los elementos, recuerda que las dimensiones o medidas de nuestros objetos se verán afectadas al aplicar o no aplicar estos bordes.
* **Layout**: Vamos a definir la forma general en que combinamos las columnas, headers y barras de navegación de nuestras plataformas, gracias a estas reglas podemos garantizar que nos adaptamos a todos los tamaños y dispositivos.



# Animación

Los 12 principios de la animación son un conjunto de reglas creadas por Disney para la animación de personajes, pero muchos de estos principios son muy útiles para ciertos aspectos de la animación de elementos de nuestras interfaces:

* **Anticipación**: Vamos a a preparar a los usuarios para la acción que viene a continuación, por ejemplo, antes de desconectar la señal de un teléfono podemos animar el icono del avión y transmitir que no señal durante algún tiempo.
* **Estirar y encoger**: Nos ayuda a generar drama sobre algún elemento, por ejemplo, para dramatizar la animación de un botón cuando el usuario ha cometido un error o alguna acción incorrecta.
* **Entradas y salidas lentas**: Nos permite introducir o remover elementos de la plataforma, tal vez conozcas estas animaciones como Fade In y Fade Out.
* **Acciones secundarias**: Todas las micro interacciones trabajan con este principio, estas animaciones nos ayudan a transmitir o dar información adicional como respuesta a alguna acción de los usuarios.
* **Timing**: Mientras más detalles añadimos a las animaciones, más rápidas o lentas se pueden percibir. Podemos utilizar estos efectos visuales para transmitir apuro o tranquilidad mientras la plataforma esta cargando.
* **Exageración**: Así como las acciones secundarias, podemos exagerar los movimientos de nuestros elementos para transmitir alguna sensación, por ejemplo, cuando el usuario quiere eliminar su cuenta.

También vamos a utilizar [Animate.css](https://daneden.github.io/animate.css/), una librería para trabajar con todo tipo de animaciones sin mucho trabajo, también nos facilita el proceso de documentación de las reglas de animación de nuestro sistema.



[***https://www.youtube.com/watch?v=uDqjIdI4bF4***](https://www.youtube.com/watch?v=uDqjIdI4bF4)

**Voz y tono.**

Los textos o *copies* que acompañan nuestros diseños también hace parte de la forma en que te comunicas con tus usuarios. Al incluir la voz y el tono de nuestros productos en el sistema de diseño conseguimos que todo el equipo tenga una guía sobre la personalidad de la marca, recuerda que toda esta documentación debe contribuir a la consistencia y comunicación con nuestros equipos.

Para definir la voz y el tono de nuestros sistemas debemos tener en cuenta los siguientes aspectos:

* **Buzzwords**: las palabras más usadas de nuestro producto

* **Phrases**: vamos a construir una librería de frases clave de nuestro producto



* **Objetivo**: el objetivo o la misión de nuestro producto en términos de voz y tono ayuda a que el equipo tenga orientación a la hora de crear nuevos textos o frases



* **Características**: definimos la personalidad de la marca, así tendremos mucho más contexto y podremos crear mejores *copies* para nuestra marca

# Iconografía

Los iconos son muy útiles para comunicar o resaltar características y funcionalidades de nuestros productos, podemos construir nuestras propias librerías de iconos personalizados o utilizar librerías como [Font Awesome](https://fontawesome.com/) y completar los iconos que hagan falta si no tenemos un equipo tan grande.

Para definir las reglas y la documentación de una iconografía consistente debemos tener en cuenta los siguientes aspectos:

* Grid: Las medidas y lineamientos que deben seguir todos los iconos
* Shapes: Las formas o figuras que podemos utilizar
* Size: Qué tamaño deben tener nuestros iconos para estar alineados con la tipografía y el resto de la plataforma
* Styles: Cómo deben estar construidos visualmente nuestros iconos, podemos utilizar colores planos o podemos trabajar en iconos mucho más realistas, etc, lo importante es estar alineados con las reglas de nuestro sistema

# Hitos

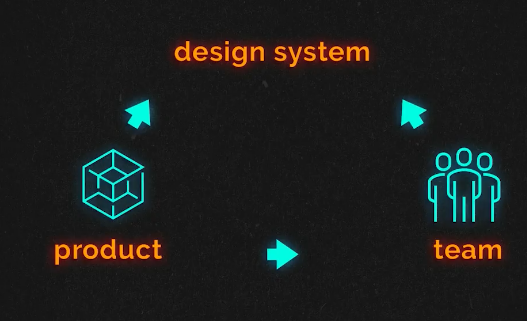
Los **hitos** son características muy particulares o incluso personales de nuestros productos, las mejores marcas incluso pueden definirse y marcar su presencia sin necesidad de utilizar logos o nombres, tal es el caso de las animaciones de carga de Google o Slack.

# Un sistema basado en personas

Un sistema de diseño se asegura que todo lo que estás diseñando se convierta en software, debemos asegurarnos de que todos los elementos estén conectados para hacer nuestro trabajo más eficiente.

Los sistemas de diseño nos ayudan a construir software en equipo, involucrando al equipo de desarrollo, de marketing, de finanzas, etc, todos aquellos que contribuyen a la hora de crear un nuevo software y ayudan a que el trabajo fluya y sea más eficiente.





**Iteremos**

El proceso de creación de software no es tan definitivo como la creación de otros tipos de producto, son productos vivos que siguen mutando y desarrollándose en el tiempo. Los sistemas de diseño también pueden prever la forma de iterar, es decir, mejorar todos los días las construcciones y la forma de construir del equipo, lo más importante es que entre todos nos pongamos de acuerdo de manera fácil, rápida y efectiva.

Para organizar el proceso de iteraciones debemos documentar todos los cambios y problemas que nos encontramos para que todo el equipo este actualizado y logremos entender el trabajo de la misma forma. Vamos a seguir los siguientes pasos:

* Darle un nombre a la iteración
* Describir por qué estamos iterando, ¿qué problema vamos a resolver?
* ¿Qué solución encontramos a estos problemas?
* Cambios de estimación de conflictos, la duración y la dificultad estimada para trabajar esta iteración

***Curso de diseño para programadores.***

**El proceso creativo**

El proceso creativo consta de algunos pasos:

1. **Preparación:** Investigar, recopilar información relativa a un problema.
2. **Incubación:** Experimentar, sintetizar, ver cómo alguien ha solucionado un problema.
3. **Iluminación:** Idear, imaginar.
4. **Evaluación:** Criticar, replantear. Definir si las soluciones son prácticas o viables.
5. **Implementación:** Construir, trabajar.

Éste proceso se aplica a problemas de la vida cotidiana.



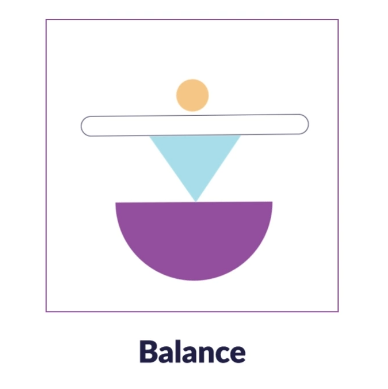
# Conceptos básicos de diseño

# Existen 6 conceptos de composición de diseño que nos ayudaran a crear no solamente composiciones web sino, editoriales, ilustraciones, etc.

1. **Balance**

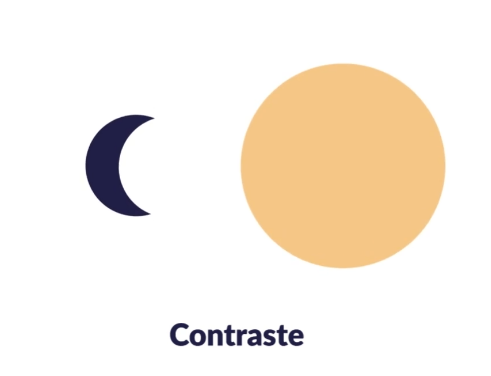
Se refiere a la estructura y estabilidad de una composición, en donde la posición de cada elementos está dada por su peso visual.

El balance puede ser simétrico o asimétrico.



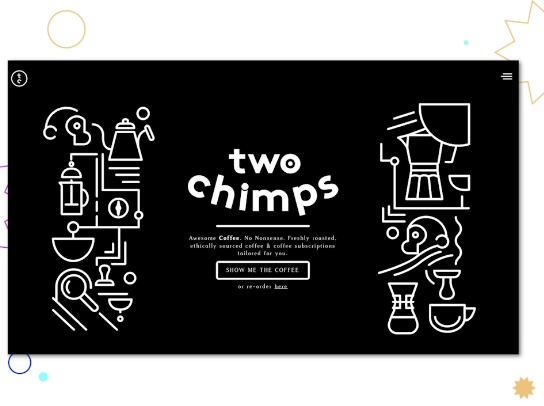
1. **Contraste**

El contraste se puede utilizar para señalar un concepto marcado por la diferencia entre dos elementos. (ejemplo: claro y oscuro, grande y pequeño, antiguo y nuevo)



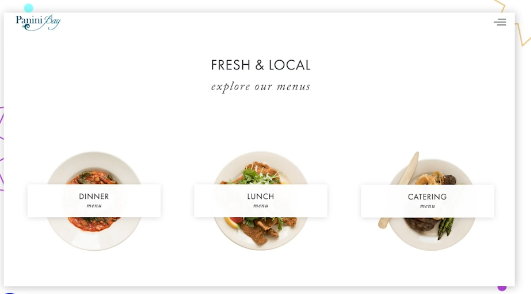
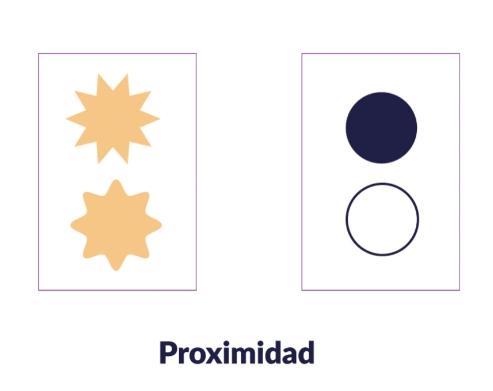
1. **Alineación**

Nos permite crear una ruta visual entre diferentes elementos que le queremos señalar al usuario.



1. **Proximidad**

Nos permite agrupar diferentes elementos. Se pueden agrupar elementos dependiendo de varias categorías.



1. **Repetición**

Es muy utilizada para generar concepto de marca (estilos repetitivos en sus interfaces)



1. **Espacio**

Se puede aprovechar el espacio de una interfaz de muchas maneras.

# Diseño responsivo.

# Es una metodología que nos permite crear diferentes diseños que sean adaptables a diferentes dispositivos.

# ¿Cómo garantizamos que nuestro diseño sea responsivo? 1º Tenemos que empezar por dispositivos móviles. Esto nos garantiza que partimos desde el contenido básico hasta el contenido más complejo. 2º Separando las capas de contenido con la de funcionalidad. Así aseguramos que todo el contenido este accesible a todos los usuarios. 3º Utilizar sistemas de Grilla y columnas. Las columnas son referencia para dividir el ancho de la página. Así es más fácil el diseño.

# Metodologías.

# 

# 

# 

# 

Estas metodologías afectan tanto al diseño como a la funcionalidad.



# Accesibilidad y diseño.

# Una parte importante del diseño es la accesibilidad. La accesibilidad es la posibilidad de que cualquier persona pueda acceder a la web, no importa la incapacidad que tenga. Esta accesibilidad es obligatoria en todas la Webs.

# 

# 

# Brief y requerimientos técnicos

# El primer elemento que debemos tener claro para construir una aplicación es el BRIEF. ¿Qué es el BRIEF? Es la hoja de ruta para empezar a diseñar. Es lo que se habló en los pasos del proceso creativo: investigar y recopilar toda la información relativa con el proyecto, con el problema a solucionar. Es donde apuntamos esos datos que hemos recopilado. Las secciones más comunes son: • Descripción del cliente o producto. • Objetivos o retos. • El target o audiencia al que va destinado ese producto o servicio. • La competencia que puede haber. • Como será la distribución. Es importante que este documento sea bonito, este organizado y podamos ver bien la jerarquía de contenidos.

# Los objetivos es la parte más importante, porque de ahí vamos a partir para crear nuestro producto.

# También es bueno tener un cronograma donde vamos a poner las fechas de las entregas.

# 

# Definición de diseño UX

1. **Investigación** ⇒ Recopilar información para conocer que ocupan los usuarios principalmente. Es bueno apoyarse de aplicaciones ya existentes para crear un buen diseño UX
2. **Análisis** ⇒ Una vez recopilada la información se analiza y se obtienen los puntos importantes que debemos tener en cuenta al momento de crear el diseño
3. **Diseño** ⇒ Se crean prototipos o sketches para visualizar el resultado
4. **Pruebas de usuario** ⇒ Por lo general se llevan a cabo en los sketches para poder realizar ajustes antes de llevar el diseño a código.

# 

# Diagramas de flujo.

# El primer paso para realizar nuestro diseño UX son los diagramas de flujo.

# Ejemplo.

# 

# Un diagrama de flujo básico, muy sencillo, es el Site Map que nos permite ver un mapa de todas las secciones del sitio. Este tipo de diagrama nos permite ver cuales secciones son principales , cuales son las secundarias y cuales son las externas y que serán un link desde nuestra pagina.

# 

# Otro tipo de diagrama, más complejo, son los User Flow. Este nos permite ver el flujo general que debe completar un usuario para hacer su pedido.

# 

# Wireframes y componentes.

# Los Wireframes son los planos de nuestra aplicación. En ellos hacemos nuestros bocetos en papel de todos los componentes y pantallas que nos salieron en los flujos que hemos creado antes.

# Los WIREFRAMES DE BAJA FIDELIDAD son dibujados a mano, pero con ellos podemos empezar a hacer pruebas de usuarios.

# 

# Podemos hacer estas pruebas de muchas formas. Lo importante es ver la usabilidad de los componentes y pantallas, si son muy complejos para los usuarios finales. Los WIREFRAMES DE ALTA FIDELIDAD son los creados con programas de diseño.

# 

# Esta parte es importante en diseño y también en desarrollo, porque en esta etapa los desarrolladores van a ver el tamaño de la aplicación final, y poder hacer estimaciones, además de decidir con que herramientas lo va a programar. También es importante que lo vea el cliente para que se haga una idea de que secciones va a tener su producto. En esta etapa no se usan las paletas de colores. Solo es un boceto en blanco y negro o grises.

# Definición de diseño.

# UX (User Experience) / UI (User Interface)

# 

# Moodboard y línea gráfica.

# Es un documento o tablero donde pondremos referencias visuales que se usan como inspiración para un diseño. No hay ninguna regla para crearlo.

# 

# 

# Teoría del color.

# Para saber que colores son los adecuados a un diseño aplicamos la teoría del color.

# 

# 

# La idea es usar la teoría del color para elegir los colores de la aplicación según el problema que resuelve.

# Otros tips para seleccionar los colores:

# • Utiliza color RGB y hexadecimales.

# • Crea un código de color consistente. Que todos los componentes de la misma categoría tengan el mismo color.

# • Menos, es más. No satures de colores el sitio.

# • Asegura que el color elegido es accesible.

# • Define una paleta de color. Esto es un grupo de colores que vamos a usar en toda nuestra aplicación.

# 

# Paletas de color.

# Tipos de Colores:

# Para construir una paleta de color, tenemos que tener claro que tipo de colores existen:

# • Colores primarios: Amarillo, azul y rojo.

# • Colores secundarios: Son los que resultan de la combinación de los colores primarios.

# • Colores terciarios: Son los colores intermedios que hay entre un color primario y un color secundario.

# 

# Tipos de paletas:

# • Combinación monocromática: Diferentes tonos de un mismo color. (diferente opacidad).

# 

# • Combinación análoga: Resulta de combinar un color primario con uno secundario y uno terciario que estén seguidos en el circulo cromático.

# 

# • Combinación complementaria: Consiste en combinar un color primario con un color secundario que no estén seguidos en el circulo cromático. Estos colores estan opuestos en el circulo.

# 

# • Combinación triádica: Se trata de combinar tres colores que estén en el circulo cromático de forma que forman un triángulo dentro del circulo.

# 

# • Combinación tétrada: Combinamos 4 colores elegidos al formar un rectángulo dentro del circulo cromático. Se eligen dos primarios y dos secundarios.

# 

**Página de paleta de colores.**

<https://color.adobe.com/es/create/color-wheel>

**Cuando ponemos colores en CSS utilizamos RGB, RGBA o hexadecimales**

RGB y RGBA

**RGB** significa: **RED**— **GREEN**— **BLUE**  
**RGBA** significa: **RED**— **GREEN**— **BLUE**—**ALPHA(Opacidad)**

Los hexadecimales

Son un grupo de 6 caracteres (números y letras) que representan un color.  
Esos 6 caracteres se agrupan en grupos de 2: el primer grupo representa el rojo, el segundo el color verde y el tercero el color azul.

Los números y letras de los hexadecimal.

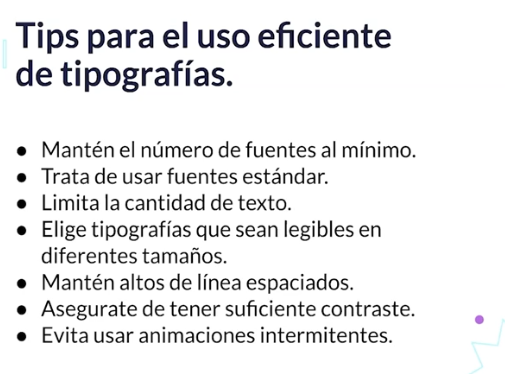
Para crear un color hexadecimal podemos utilizar los siguientes caracteres:

* 0 1 2 3 4 5 6 7 8 9 A B C D E F  
  Los números y letras representan la intensidad del color, entre más a la izquierda, menos intenso el color, entre más a la derecha, más intenso el color.  
  Con esto podemos crear diferentes intensidades de un color, por ejemplo:
* #27ECC0

Este hexadecimal tiene una intensidad de rojo 27, una intensidad de verde EC y una intensidad de azul C0.

# Tipografía.

# Otra de las cosas esenciales en una aplicación es el tipo de letra.



# Tipos de Fuente:

# 

# 

# 

# Es importante definir estas características estándar de nuestro sitio en el CSS.

# 

# Pagina de fuentes.

# *https://fontawesome.com/*

# Layout y sistemas de grillas.

Es un sistema de columnas creado por nosotros de acuerdo a las necesidades para ajustar nuestros elementos y componentes.  
La mejor herramienta de CSS para crear una grilla es Grid.  
Sass es un pre-procesador de CSS te ayuda a escribir CSS de una manera más rápida y más fácil.  
Mixin es una clase que tiene dentro del pre-procesador, que te ayuda a manejar mejor los Breakpoints.  
Tenemos que configurar nuestros BreakPoints (Tipos de pantalla).  
• $xs: 360px. Para móviles pequeños.  
• $s: 440px. Para móviles con la pantalla más grande.  
• $m: 768px. Para tablets.  
• $l: 1280px. Para Ordenadores pantalla normal.  
• $xl: 1440px. Monitores de alta calidad.

Podemos usar un Mixin para manejar los distintos breakpoints más fácilmente.  
Después vamos a configurar nuestras variables dependiendo de los breakpoints.  
–columns. Es el número de columnas que vamos a poner.  
–column-gap es el espacio entre las columnas.  
Creamos después una clase Grid para configurar cada uno de los elementos contenedores donde vamos a incluir nuestros componentes.  
Finalmente le añadimos un display grid, que nos permite que toda esta configuración surta efecto. Y definimos un grip-column-gap para la distancia entre columnas y un grid-template-column que nos permite tener esta estructura.  
Para definir el tamaño de un componente, se pone grid-column: 1/5, donde uno es la columna donde empieza, y 5 es la columna donde termina.

# 

# Themes y customizaciones.

# Otra herramienta que podemos hacer uso en nuestra aplicación son los themes.

# ¿Qué es un themes? Es una capa de color y estilos que se añade por encima de nuestra aplicación y nos permite tener distintas variaciones sin hacer cambios drásticos en nuestro código. En otras palabras, theme es un archivo donde tenemos configurado la presentación visual de toda la aplicación. Teniendo dos o más theme podremos tener dos o más presentaciones visuales.

# Para hacer los themes fácilmente podemos hacerlo por medio de variables de css o variables de Sass. Las variables de Sass son más sencillas.

# Para tener varios themes, se crean un archivo de css por cada theme, pudiéndose intercambiar una por otra desde una archivo donde se configuran todos los archivos importados, sin tener que cambiar toda la programación de la aplicación. Simplemente con cambiar el nombre de un archivo por otro se realiza el cambio de theme.

# 

# Imágenes para web.

# Al momento de elegir imágenes para mi sitio debo asegurarme de que estas imágenes aporten al contenido y en las que el usuario se vea reflejado. Tambien hay que tener en cuenta que las imágenes sean consistentes con mi paleta de colores.

# 

# 

# 

**Pagina para imágenes.**

<https://unsplash.com/>

<https://www.freepik.es/>

<https://pixabay.com/>

<https://morguefile.com/photos>

<https://picjumbo.com/>

<https://www.pexels.com/es-es/>

**Recomendaciones para el uso de animaciones.**

* Una página muy animada es una página que saca rápido al usuario.
* Procurar que no se reproduzca automáticamente y que no tengan sonido.
* Que las animaciones no tengan flashes.
* Si la animación aporta al contenido, es necesario añadir transcripciones.
* Evitar que la animaciones bloqueen la lectura básica del contenido.
* Recordar que las animaciones y vídeos afectan el rendimiento de la página.

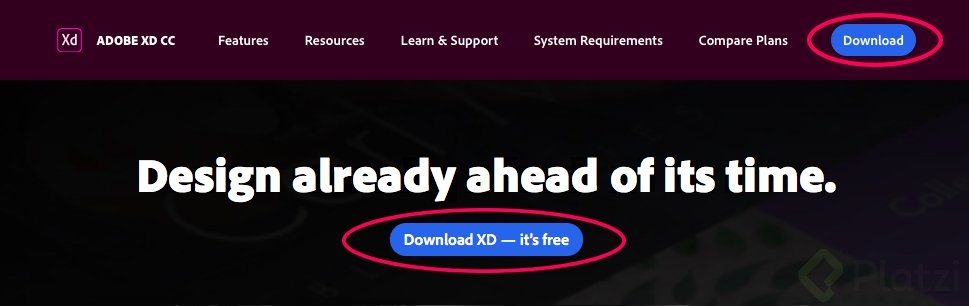
***Curso de Adobe XD***

**Instalación**

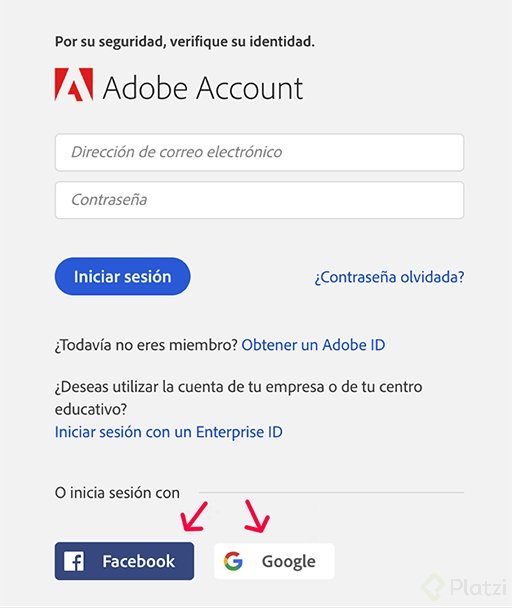
Ya que tienes los conceptos básicos sobre la creación de un producto digital, vamos a comenzar con el proceso de diseño visual y la definición de la estructura. Para esto vamos a instalar el software XD de Adobe, con este podremos crear nuestras propias interfaces.

1- Entra a: <https://www.adobe.com/es/products/xd.html>

2- Da click en el botón de descarga

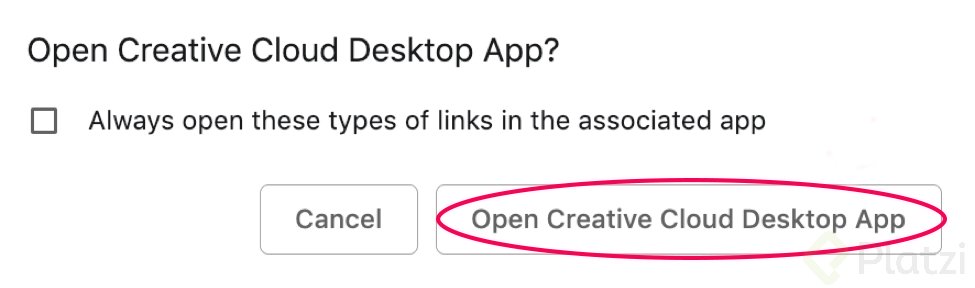


3- Inicia sesión con Facebook o Google.

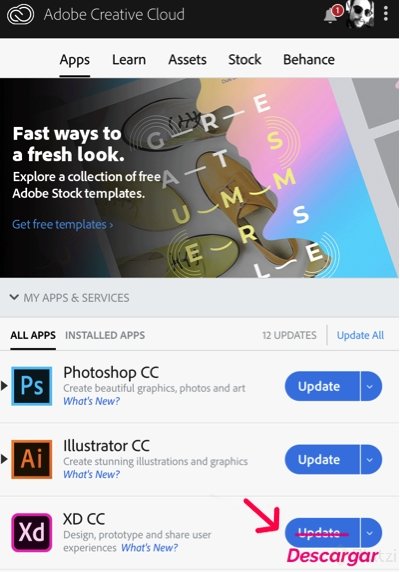


4- Después de haber iniciado sesión en tu computadora se instalará Adobe Creative Cloud. Este software te permite gestionar los programas de Adobe.

5- Da click en “Open Creative Cloud Desktop App".



6- Al instalar Adobe Creative Cloud verás esta ventana donde se pueden gestionar todos los programas de Adobe.



7- Para nuestro curso solo debemos seleccionar la descarga de Adobe XD. Al finalizar el proceso de descarga, estás listo para usar Adobe XD.



Por último, quiero recordarte que Adobe XD tiene una versión gratuita y una versión paga.

La versión paga te permite publicar todos los prototipos y diseños que hagas. Si te quedas con la versión gratuita podrás diseñar todo lo que haremos en el curso, pero la publicación de los diseños solo podrás hacerla una vez.

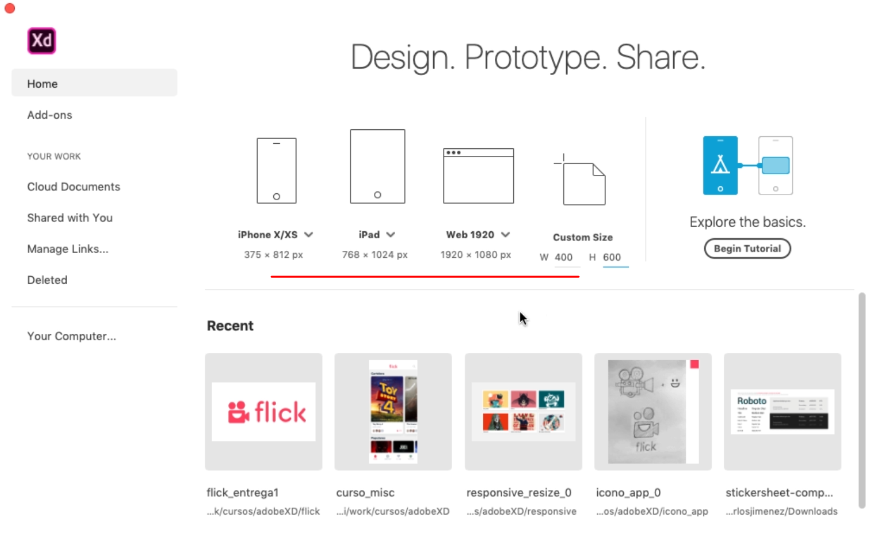
**¿Qué es Adobe XD?**

Adobe XD, o Adobe Experience Design, es una herramienta de edición de gráficos que funciona para crear interfaces de páginas web y de aplicaciones. Permite al diseñador enfocarse en la experiencia del usuario al navegar, con un rango mínimo de error y en el menor tiempo posible.

Adobe crea Adobe XD porque ningún programa de adobe se especializaba en diseño de interfaces.

**Creación de un documento**

Los **Artboards** son tamaños de documentos que nos proporciona Adobe XD con sus respectivos usos como móvil, tablet, web y una opción personalizada.



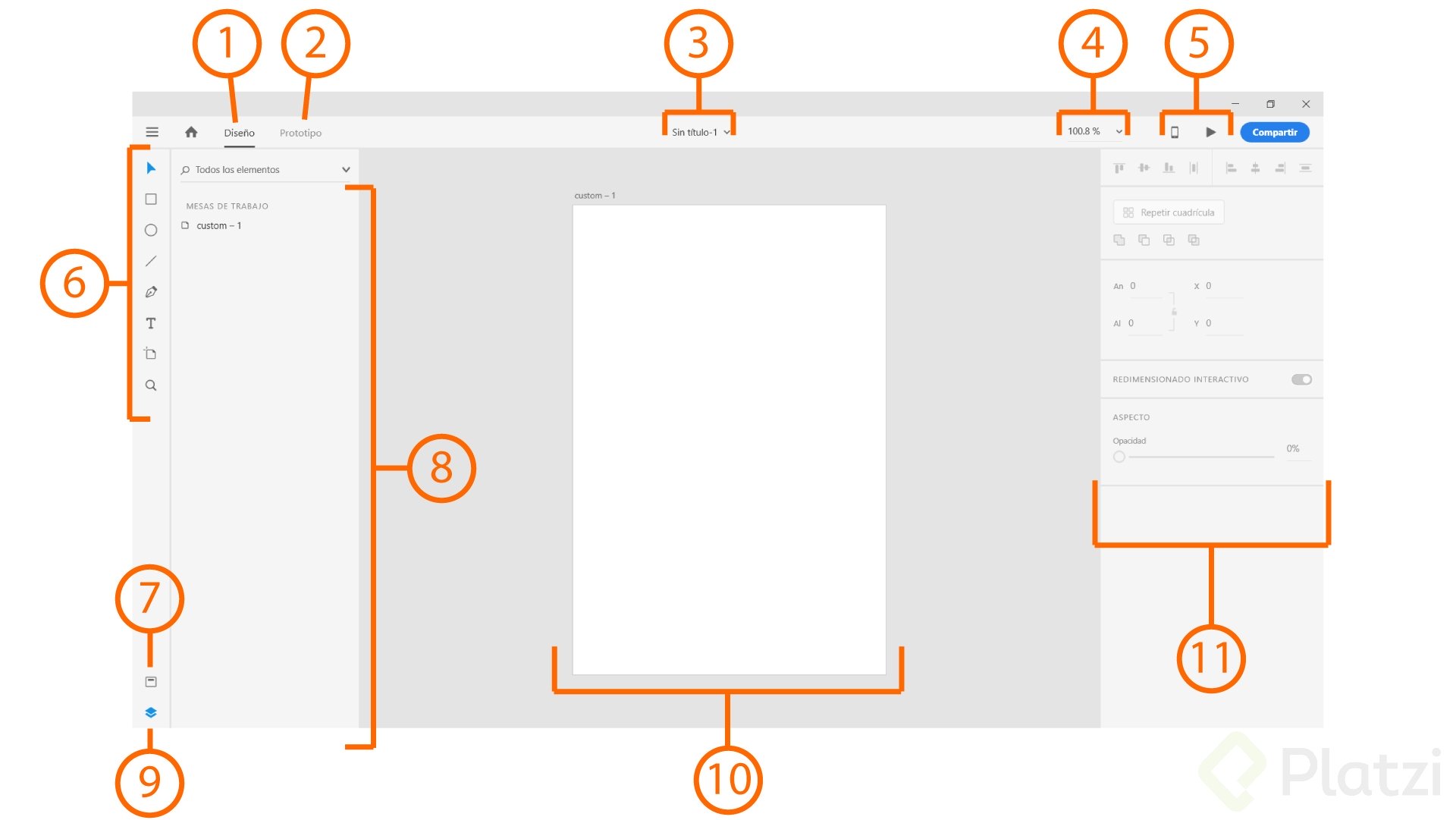
**Interfaz y navegación**

El **zoom** se puede encontrar en dos lugares: arriba a la derecha o a la izquierda como última opción en la barra de herramientas. También podemos seleccionarla usando la tecla **Z**.

Una vez seleccionada damos click para hacer un **zoom in** y click manteniendo oprimido **Alt** para hacer **zoom out**.

**Cmd/Ctrl + 1**: Lleva el zoom al 100%  
**Cmd/Ctrl + 2**: Lleva el zoom a 200%  
**Cmd/Ctrl + 3**: Ver centrado en pantalla un objeto seleccionado.

Para poder seleccionar y arrastrar nuestro Artboard usaremos la Tecla **Espaciado.**



1. **Modo Diseño**: en este modo trabajaremos nuestro diseño. Encontraremos las herramientas necesarias para crear las pantallas, capas y *artboards* creadas y panel de propiedades de la herramienta o elemento seleccionado.
2. **Modo Prototipo**: te permite seleccionar objetos del diseño y cablear interacciones entre mesas de trabajo.
3. **Nombre del proyecto**: nombre bajo el cual identificamos el proyecto en el que estamos trabajando.
4. **Zoom**: ajustar el tamaño del *viewport* de trabajo.
5. **Vista previa**: Encontramos dos opciones, ver en **dispositivo móvil** y **vista previa en el ordenador**. Podremos probar las interacciones creadas en el modo prototipo.
6. **Barra de herramientas**: encontraremos las herramientas de trabajo con las cuales poder realizar el diseño.
7. **Activos**: en esta sección/cajón hallaremos la paleta de colores, estilos de carácter y símbolos.
8. **Capas**: cuando no hay nada seleccionado visualizaremos los *artboards* y elementos fuera de ellos. Cuando hay un *artboard* seleccionando veremos las capas/elementos creados en ellos.
9. **Mostrar/ocultar menú de Capas**.
10. ***Artboard***: mesa de trabajo en el cual diseñaremos nuestras pantallas. En la esquina superior izquierda se encontrará el nombre de la mesa. Podremos cambiar el nombre para un mayor orden.
11. **Menú contextual (Propiedades)**: dependiendo del elementos seleccionado nos mostrará las propiedades que podremos modificar para personalizarlo.

# Artboards

Los Artboards son este contenedor fundamental creado desde la pantalla de bienvenida, es el tamaño del dispositivo sobre el cual vamos a incluir elementos, diseños y más.

Las propiedades principales de un artboard son **tamaño** y **posición.**

# Grids y layouts

**Grids y Layouts** son las últimas características agregadas por Adobe XD para los artboards y son útiles cuando queremos trabajar en interfaces bien estructuradas, con elementos alineados que tengan una lectura y proporción clara.

Grid que normalmente usamos en diferentes dispositivos:  
**Web**: 12 columnas  
**Tableta**: 8 columnas  
**Móvil**: 4 columnas

**Formas**

En el diseño de interfaces todos los elementos con los que interactúas son formas sencillas como: rectángulos, círculos, triángulos y líneas.

* **Rectángulo**. Puedes crear rectángulos o un **cuadrado** manteniendo presionada la tecla **Shift**.
* **Elipse**. Puedes crear elipses o un **círculo perfecto** manteniendo presionada la tecla **Shift**.
* **Triángulo** Puedes crear triángulos o un **triángulo equilátero** manteniendo presionada la tecla **Shift**.
* **Línea** Puedes crear líneas en cualquier dirección o líneas **horizontales, verticales y a 45 grados** presionada la tecla **Shift**.

Cualquiera de las formas anteriores puede ser trazada desde el centro presionando las teclas **Shift + Alt/Opt** y comparten las propiedades:

* Opacidad
* Border radius (excepto elipses y líneas)
* Fill (excepto las líneas)
* Border
* Shadow

**Shortcuts.**

Para centrar y ver todo lo del proyecto: ***Control + 0***

Para centrar la vista de algún elemento oprimimos **Ctrl + 3** en Windows.

Para generar una forma uniforme a partir del punto donde se clickea: ***Control + ALT***

Para posicionar adelante o atrás un elemento: ***Control +******Tecla [*** */* ***Tecla ]***

Para hacer un grupo seleccionando los elemento: ***Control + G***

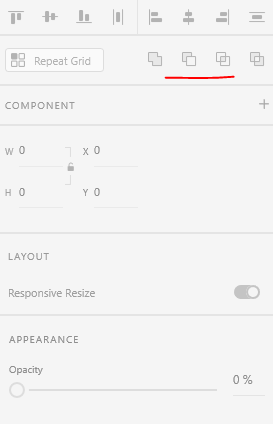
Para quitar temporalmente elementos: ***Control + Q*** Volver a traer: ***Control + B***

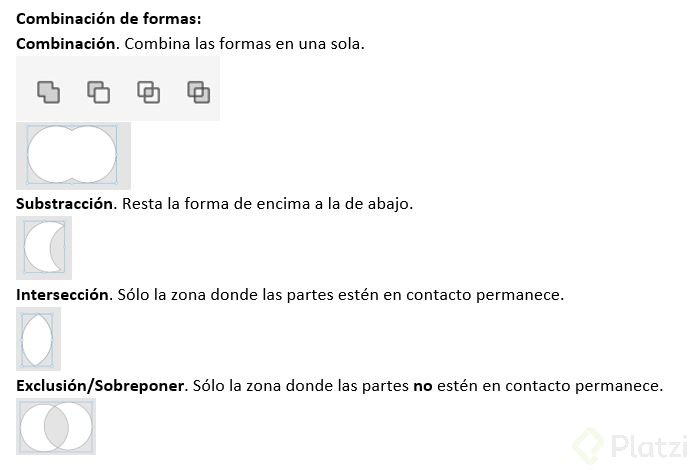
Para duplicar un elemento: ***Control + D*** o también: ***ALT + EXTRAER FIGURA CON EL MOUSE***

**Alinear y combinar formas**

Para **duplicar** cualquier forma puedes hacer **Cmd/Ctrl + D** o mantener **Alt** oprimido mientras haces clic y arrastras.  
Combinación de formas:

* **Combinación**. Combina las formas en una sola.
* **Substracción**. Resta la forma de encima a la de abajo.
* **Intersección**. Sólo la zona donde las partes estén en contacto permanece.
* **Exclusión/Sobreponer**. Sólo la zona donde las partes **no** estén en contacto permanece.





***La alineación de varios objetos seleccionados se hace en relación a la posición media de todos.***

# Diseñar icono.

# Podemos crear formas complejas a partir de formas básicas y haciendo uso de operaciones booleanas.

# 

**Cmd/Ctrl + 8** te ayudará a convertir un objeto compuesto a una forma sencilla.

# Textos y tipografía

El atajo de teclado para la herramienta de texto es la **tecla T**.  
Puedes trabajar con dos tipos de texto: texto de tipo punto y texto de área.

Para aumentar el tamaño de la tipografía seleccionen (en Windows) **Control+Shift+punto** y para disminuir el tamaño sería **Control+Shift+coma**

# Imágenes y máscaras.

# Las máscaras se crean con imágenes y formas.

Si queremos utilizar la opción de máscara en Windows, seleccionamos la imagen, la forma simple y usamos el comando Ctrl + Shift + M.

Ctrl + Shift + G para desagrupar la máscara.



# Pluma

La pluma te ayudará a crear paths que pueden ser formas regulares o irregulares. La encuentras en la barra de herramientas y su shortcut o atajo de teclado es la **Tecla P**

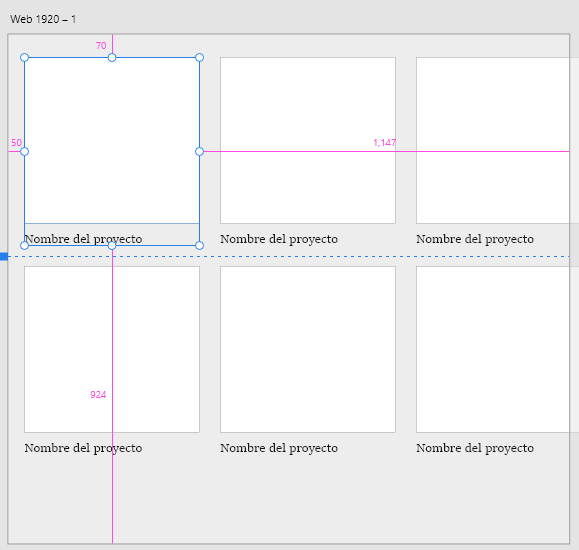
**Para practicar:**

[*https://bezier.method.ac/*](https://bezier.method.ac/)

# Repeat grid tool

En el diseño de interfaces es normal encontrarse con estructuras repetitivas y por esa razón **Adobe XD** creó una herramienta llamada **Repeat Grid Tool** para facilitar este trabajo.

Con un elemento seleccionado y **ALT** puedo ver sus dimensiones en comparación con los otros elementos.



# Cargar contenido en repeat grid tool

# De manera intuitiva si luego de crear un grid soltamos por ejemplo las imágenes sobre el contenedor automáticamente las llenara. De igual forma funcionará con textos, pero este texto deberá estar separados por texto de *Tipo Punto* que es textos cortos de una línea.

# Cuando hablamos de texto de párrafo hablamos de texto de *Tipo Area*

# 

# 

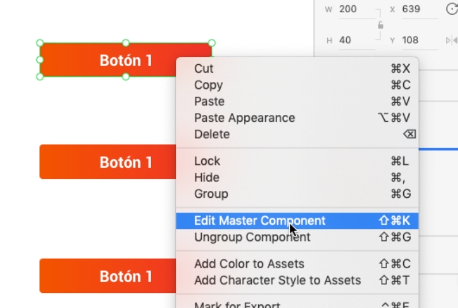
# *Texto Tipo Punto*

# 

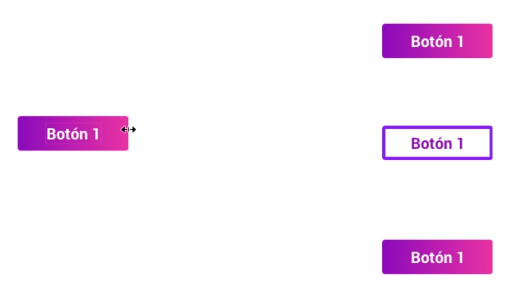
# Componentes

Los componentes son elementos reutilizables a través de un diseño.  
Puedes crear un componente con **Ctrl/Cmd + K. A partir de ahí podre crear más componentes hijos.**

**En caso de que el elemento padre se eliminé se podrá invocar nuevamente seleccionando un elemento hijo + Click Derecho + Edit Master Component**

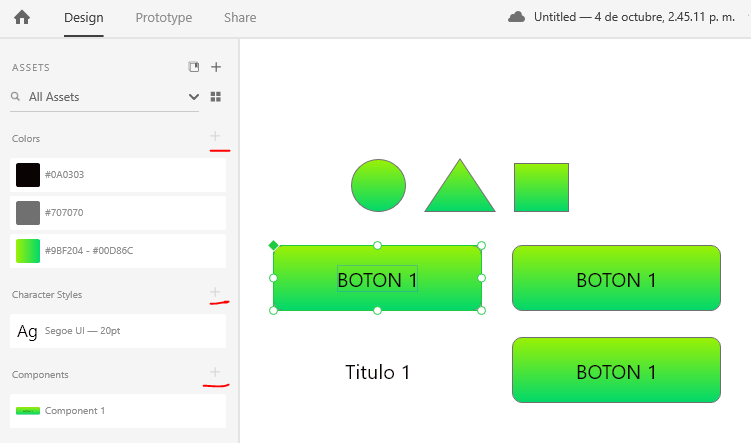


Puedo cambiar una instancia y seguir heredando los estilos que no fueron modificados de su padre.



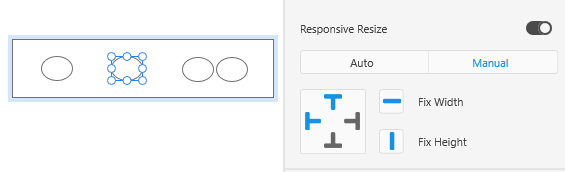
# Panel de assets

Al crear nuestro componentes automáticamente se añaden en nuestros Assets. Pero allí también podremos guardar nuestro tamaños y fuentes como los colores también. Solo hay que posicionarse sobre el elemento con la propiedad y añadirlo dependiendo de su categoría en el signo +.



# Responsive Resize

# Por medio de las opciones de Responsive Resize de Adobe XD podremos configurar el comportamiento en dimensión que tendrá un elemento hijo en cuanto a su elemento padre cambie de tamaño.



***Control + Shift + M*** en Firefox puedo cambiar entre la pagina con tamaño normal y el modo responsivo.

Las medidas estándar para hacer un diseño responsive son:

**Portátil estándar:** 1024px

**Tablet:** 768px

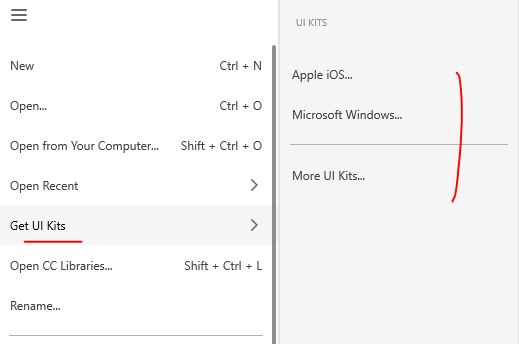
**Celular Landscape (Horizontal):** 480px

**Celular Portrait (Vertical):** 320px

# UI Kits.

Los UI Kits o User Interfaces Kits son elementos de diseños reusables para nuestros proyectos o proyectos ya existentes, como crear diseños para aplicaciones iOS o Android.

Para añadirlas debo ingresar:

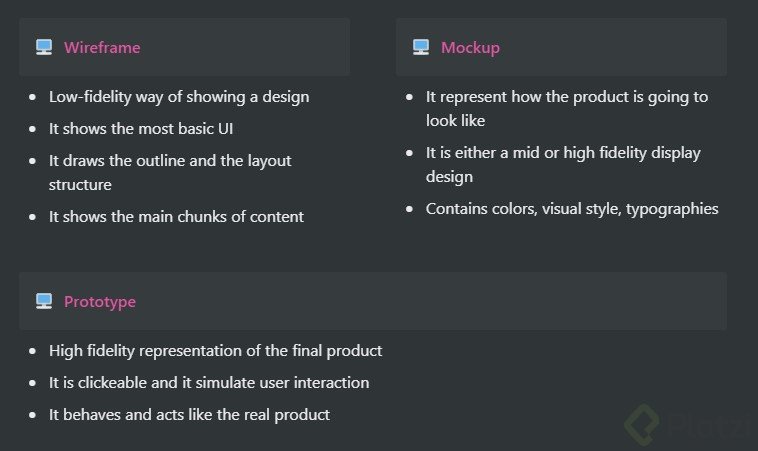


**Wireframes, Mockups y Prototipados.**

El **wirefrime** es la versión simplificada y estructural del proyecto.

El **mockup** es el diseño final de la interfaz de usuario.

El **prototipado** es el diseño que permite la interacción con él.



# Agregar contenido al mockup del proyecto.

# Para añadir imágenes de avatares a nuestros diseños podremos usar:

# [*https://uifaces.co/*](https://uifaces.co/)

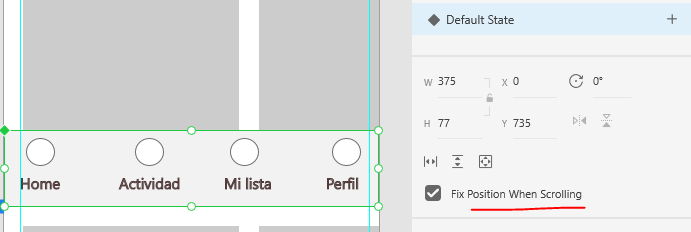
# 

# Prototipado.

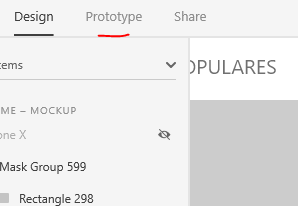
# Para probar nuestra interface con funcionalidad de scrolling podremos alargar nuestro artboard y nos marcara esa diferencia con una línea azul.

# 

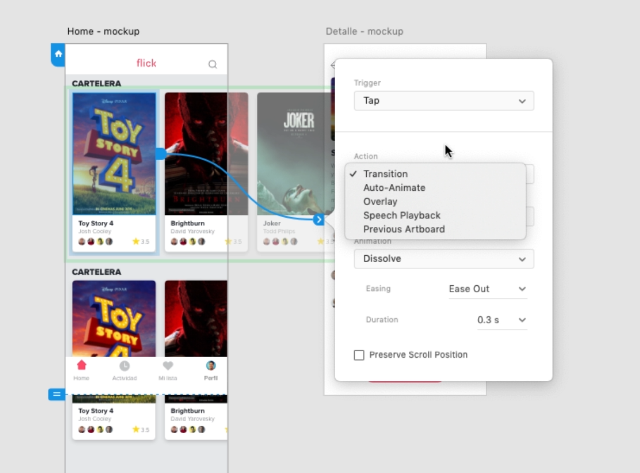
En este caso queremos dejar la barra de navegación pegada en la parte inferior. Para ello solo seleccionamos los elemento y al tenerlo ya como un componente seleccionaremos la opción *Fix Position When Scrolling*:



Cuando cambiemos a la pestaña de prototipo solo debemos seleccionar el elemento que desencadenará la otra vista (en este caso la vista de la película) y se une la línea azul con la vista que emergerá.



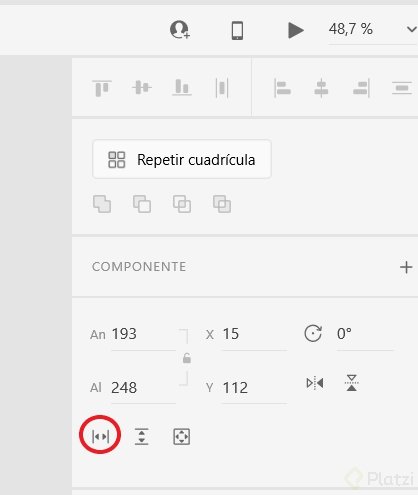
Completaremos algunas configuraciones como el desencadenador que en este caso es un **Tap** y el tipo de animación.



# Animaciones avanzadas

En las nuevas actualizaciones tenemos un scroll horizontal de una manera más fácil.

1. Seleccionamos los objetos.
2. Vamos al siguiente panel en la siguiente opción sin tener que duplicar el artboard para animar.



# Exportar.

# Para exportar los elementos tendremos dos maneras de hacerlo. Una es ir marcando los elementos para exportar como se ve en la siguiente imagen:

# 

# Luego iremos a Exportar y daremos en *Batch* para extraer todo los elementos que hemos marcado. La otra manera de exportarlo es seleccionar el elemento y daremos igualmente Exportar y esta vez seleccionaremos *Selected*

# 