



TECNOLÓGICO  
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO<sup>®</sup>  
de Pabellón de Arteaga

**TEC**

28-5-2020

# Ingeniería de software

## RESEÑA

Profesor:  
**Eduardo Flores Gallegos**

Alumno:  
**José Yovani Loera Lara**

Carrera:  
**Ingeniería en Tecnologías de la Información y  
Comunicaciones**



## PRIMER LECTURA: Button Design - UI component series

Todo lo que necesita saber sobre el diseño de botones, uno de los principales bloques de construcción interactivos para crear una interfaz de usuario.

Para diseñar bien tenemos que mirar hacia atrás en la historia y los orígenes de los botones físicos, un componente de interfaz de usuario tan utilizado en todos los productos digitales de hoy. Los botones son asombrosos. El toque de un dedo que pone en movimiento un electrodoméstico, un automóvil o un sistema, incluso si el usuario no comprende los mecanismos o algoritmos subyacentes.

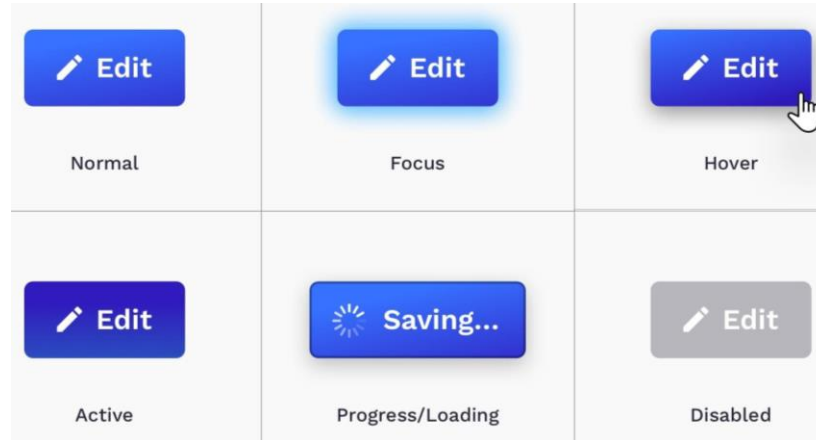


### Botones y Enlaces

Los botones comunican acciones que los usuarios pueden realizar. Por lo general, se colocan en toda la interfaz de usuario, en lugares como: cuadros de diálogo, formularios, barras de herramientas, etc.

Diferencia de enlaces y botones es:

- Los enlaces se utilizan cuando navega a otro lugar, como: página "ver todo", perfil "Roger Wright", etc.
- Los botones se utilizan cuando realiza una acción, como: "enviar", "fusionar", "crear nuevo", "cargar", etc.



Normal: Comunica que el componente es interactivo y está habilitado.

Focus: Comunica que el usuario ha resaltado un elemento, utilizando un teclado u otro método de entrada.

Hover: Se comunica cuando un usuario ha colocado un cursor sobre un elemento interactivo.

Active: El estado presionado comunica que el usuario había tocado el botón.

Progress / Loading: Se utiliza cuando la acción no se realiza de inmediato y comunica que el componente está en progreso de completar la acción.

Disabled: Comunica que el componente actualmente no es interactivo, pero se puede habilitar en el futuro.



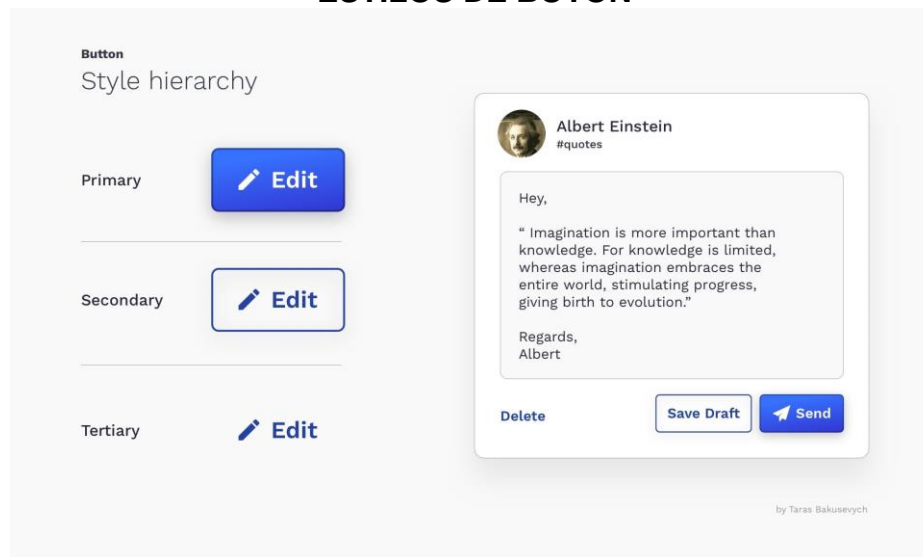
TECNOLÓGICO  
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO<sup>®</sup>  
de Pabellón de Arteaga

TEC

## ESTILOS DE BOTON



### A veces no hay "defecto"

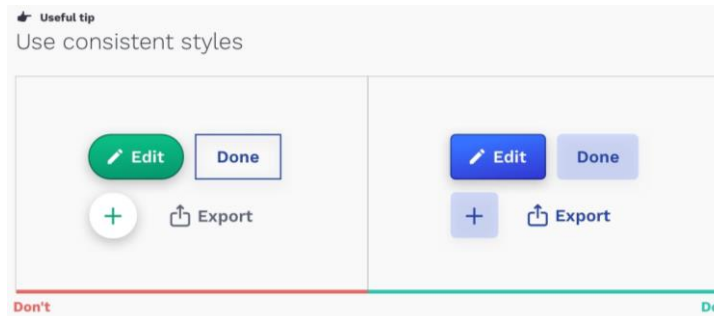
En general, desea hacer que el botón seleccionado más comúnmente sea el "predeterminado" (usar estilos principales) y ponerlo en un estado enfocado. Esto ayuda a la mayoría de los usuarios a terminar sus tareas más rápido y los orienta en la dirección correcta.

### La consistencia mejora la velocidad y la precisión.

"La coherencia es uno de los principios de usabilidad más poderosos: cuando las cosas siempre se comportan igual, los usuarios no tienen que preocuparse por lo que sucederá". - Jakob Nielsen

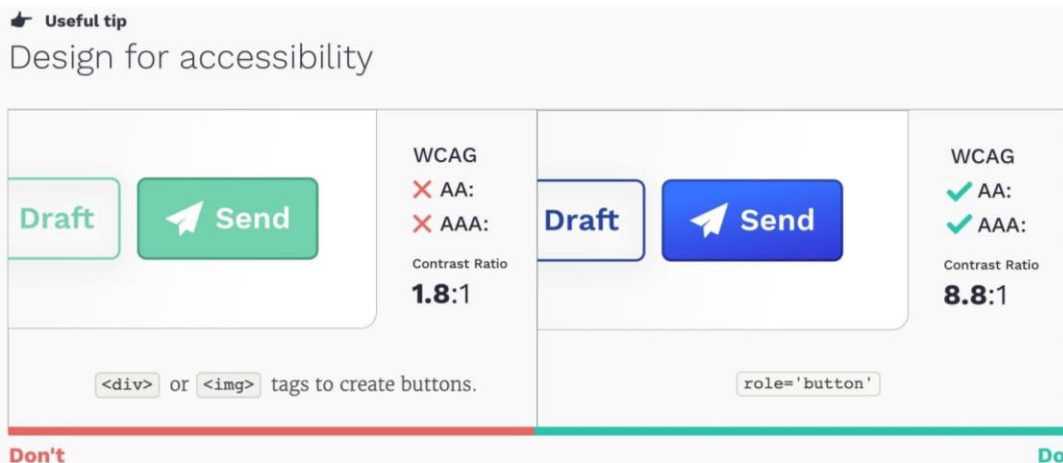


## Diferentes usos de estilos



## Diseño para accesibilidad

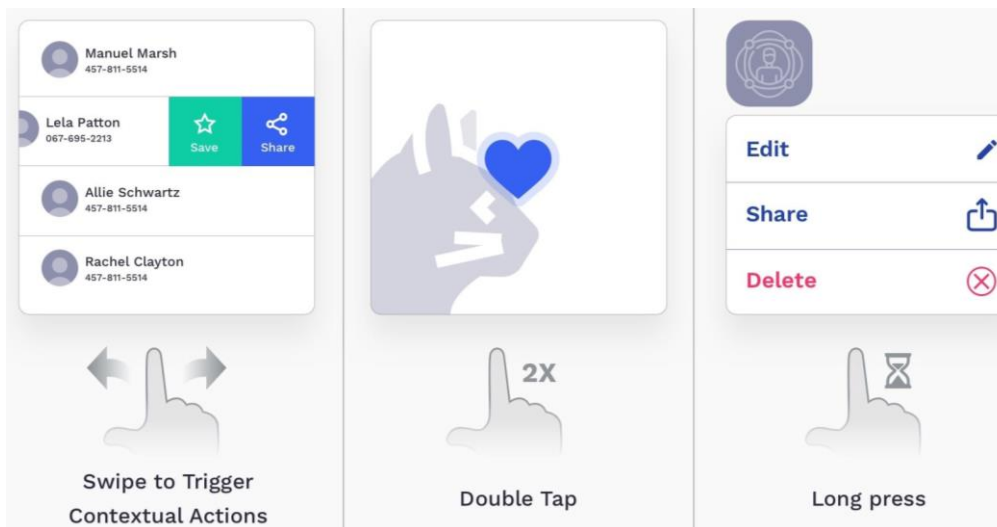
Esta recomendación debe repetirse para cada componente. El tamaño del área objetivo fue uno de los factores que afectaron la accesibilidad. Otros son tamaño de fuente, color y contraste. Hay un montón de herramientas que pueden ayudarlo a verificar fácilmente el rendimiento de sus componentes.



Los diseñadores deben trabajar en estrecha colaboración con los equipos de desarrollo para asegurarse de que los botones funcionan con los lectores de pantalla.



Los gestos permiten a los usuarios interactuar con la aplicación mediante el tacto. Usar el tacto como otra forma de realizar una tarea puede ahorrar tiempo y dar un control táctil. Como algunos gestos, como deslizar para activar acciones contextuales, tocar dos veces para dar me gusta o mantener presionado, y se usan más ampliamente todos los días, todavía no son muy aparentes para el usuario promedio.



### OK – Cancelar

- Tener una acción OK primero apoya el orden de lectura natural. Puede ayudar ahorrar algo de tiempo si sabemos que lo más probable es que esto sea lo que seleccionarán los usos. Windows pone OK primero
- Listado OK último mejora el flujo. Algunos pueden argumentar Ok ya que el siguiente botón moverá al usuario hacia adelante. Poner OK al final, ayuda a los usuarios a evaluar todas las opciones antes de tomar medidas y ayuda a evitar errores y tomar decisiones apresuradas. Apple pone OK al final

### Botones deshabilitados

Todos estaban en esta situación antes. Estar atrapado en la pantalla durante varios segundos o minutos, tratando de descubrir por qué su progreso está bloqueado por un botón deshabilitado y qué debe hacer para que esto vuelva a la vida). Los botones deshabilitados se usan porque quitar el botón de su ubicación nativa y revelarlo en un contexto posterior podría confundir a los usuarios.



👉 Useful tip  
Disabled buttons suck

Create user

Full Name

Email

Cancel

Create

Create user

Full Name

Email

Email is required to create new user

Cancel

Create

Don't

Do

## SEGUNDA LECTURA: ¿CUÁL ES LA DIFERENCIA ENTRE WIREFRAME, MOCKUP Y PROTOTYPE?

A menudo escuchas términos como sketch, wireframe, mockup y prototype mientras trabajas con desarrolladores, pero ¿entiendes realmente lo que significan estos términos? ¿Sabrías cuándo usar una estructura metálica o cuándo usar un prototipo?

Comencemos con las razones por las que debe comenzar con un sketch, wireframe, mockup o incluso un prototype al comenzar a tosa:

- Hacer una lluvia de ideas o descubrir lo que quieres construir. Están diseñados para ayudarte a definir tus expectativas.
- Ahorre dinero en desarrolladores y describa claramente lo que necesita para construir.
- Úselos para lanzar inversores, primeros clientes y cofundadores.



## **# 1 Sketch**

Básicamente es solo un dibujo a mano alzada sin formato en una hoja de papel, que le brinda una representación de baja fidelidad de su aplicación. Es la forma más rápida de preparar su idea para una lluvia de ideas. Incluso un simple sketch puede describir su idea mejor que las palabras. Genere ideas, cambie detalles, visualice lo que tiene en mente; Todo depende de tu imaginación.

## **# 2 ¿Qué es una wireframe?**

Una wireframe es equivalente al esqueleto o estructura simple de su sitio web / aplicación. Cada uno se utiliza para describir la funcionalidad de un producto, así como las relaciones entre las vistas (lo que sucederá cuando haga clic en un botón determinado). Las decisiones sobre qué (contenido / características) y dónde colocar en el sitio web o la aplicación generalmente se toman durante esta etapa.

## **# 3 ¿Qué es una mockup?**

En Brainhub, nunca comenzamos a construir una aplicación antes de que se completen las maquetas. Con esta representación, puede comenzar a trabajar en el proceso de desarrollo y el desarrollador puede hacer realidad sus mockups. Cualquier mockup proporcionará una representación de fidelidad media. Agregue colores, fuentes, texto (Lorem ipsum), imágenes, logotipos y cualquier otra cosa que dé forma a su wireframe. Su resultado es un mapa estático de la aplicación.

## **# 4 prototype**

Los prototypes ofrecen una representación de alta fidelidad de su aplicación. Es como una mockup enriquecida con piezas UX, interacciones, animación y cualquier otra cosa que le gustaría experimentar al hacer clic en los botones. Este paso no siempre es necesario para crear una aplicación. Si no es un desarrollador, le recomiendo tener un prototype para presentar su idea a amigos, familiares y posibles inversores.





TECNOLÓGICO  
NACIONAL DE MÉXICO

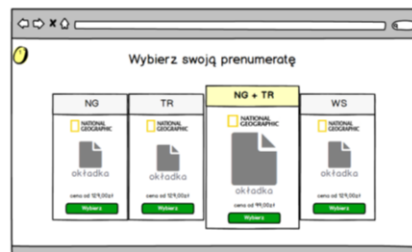


INSTITUTO TECNOLÓGICO<sup>®</sup>  
de Pabellón de Arteaga  
**TEC**

## LA DIFERENCIA ENTRE WIREFRAME & MOCKUP

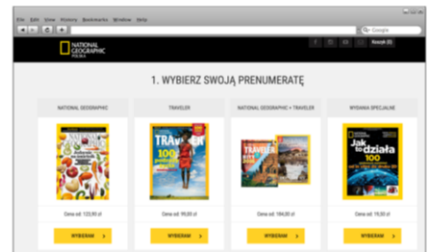
### WIREFRAME

Structure + Functions + Content



### MOCKUP

Style + Colours + Right Content



Si está buscando excelentes herramientas para crear su primer sketch, wireframe, mockup o prototype, solo consulte mi última publicación sobre cómo crear un prototype de una aplicación como no diseñador . Una vez allí, encontrará excelentes herramientas y recursos gratuitos que lo ayudarán a construir su primer prototipo de aplicación.

## TERCER LECTURA: 7 RAZONES PARA MODELAR EL SOFTWARE ANTES DE PROGRAMAR

Las técnicas de modelado mejoran la productividad del equipo de desarrollo. Una vez tienes el modelo de datos, de comportamiento... los puedes usar para generar código automáticamente. Para aplicaciones “típicas” (tipo “data-entry” con muchos formularios tipo CRUD) puedes llegar a generar todo el código para otras una parte considerable.

Reduce los defectos en el código. Hay un buen número de herramientas para verificar, testear y validar tus modelos que permiten detectar errores analizando sólo los modelos. Y ya sabes, cuánto antes detectes un defecto más barato es repararlo.

Los modelos te permiten explorar diferentes alternativas de diseño y arquitectónicas antes de empezar a desarrollar usando técnicas de simulación.

Los modelos simplifican el mantenimiento y la evolución del proyecto (por ejemplo, ayudando a localizar rápidamente las partes del código a tocar después de un cambio en los requisitos). También facilitan la reutilización del software en proyectos futuros (el modelado mejora el ROI especialmente cuando la empresa desarrolla proyectos similares).



Los modelos capturan eficientemente el conocimiento del sistema. Los modelos son la mejor documentación. Lo que simplifica la integración de nuevos miembros al equipo.

¿Todavía vas diciendo que no con la cabeza? Pues dos otras razones, más “pragmáticas”:

Muchos de tus clientes te van a pedir los modelos del software que desarrolles para ellos. En muchas administraciones públicas, seguir un proceso de ingeniería de software bien formalizado (y por tanto con sus fases de análisis y diseño y los modelos correspondientes) es obligatorio para participar en los concursos públicos. Recuerda que el modelado no está reñido con el desarrollo ágil.

Los que hacen dibujitos cobran más que los que programan. A lo mejor dibujar es menos “sexy” pero ser bueno analizando sistemas software (y expresando estos análisis mediante el uso de modelos software) está mejor pagado que dominar los últimos frameworks JavaScript. Hay excepciones pero ésta es la regla habitual.