

RESEÑAS DE LECTURAS UNIDAD 4

Jaqueline Garcia Luevano ITIC4
Eduardo Flores Gallegos

LECTURA 1: Diseño de botones: serie de componentes UI

La lectura nos muestra un tema importante y quizás uno en el que no muchos ponen atención, los botones, estos nos han facilitado mucho del trabajo, como usuarios, de algún aparato o incluso sistema que los tenga. Los botones son los que realizan las acciones que nosotros deseamos, por otro lado, tenemos los enlaces, estos son solo una redirección a algo en específico. El botón a su vez tiene características que lo distinguen, si bien mencione antes que nos facilita el trabajo y ciertas acciones, sus características físicas nos permiten detectarlos más fácilmente, tales como que estos tienen una tarjeta de texto en la cual se explica que realiza este, en ocasiones un icono, puede ser de forma rectangular y con esquinas redondeadas entre otras más ya que como también lo sabemos hay una variedad en cuanto al diseño de estos. Los botones al hacer acciones también tienen estados, como bien lo menciona la lectura, unos de ellos son el estado normal, el de enfoque, el de desplazamiento que es cuando se coloca un cursor en él, el activo que es cuando se presiona, el de progreso y el desactivado. Otra característica de los botones es que tienen estilos, estos se aplican dependiendo de la importancia del botón lo cual es relevante ya que así mostrara de forma destacada el que más comúnmente se utiliza. Como bien lo mencione antes los botones nos facilitan el trabajo como usuarios por tanto estos no deben dejar duda sobre si es o no un botón o sobre que es específicamente lo que hace además de tener una consistencia en todos ellos. El tamaño también juega un papel importante ya que es más fácil presionar un botón de un tamaño considerable a uno muy pequeño. En general la lectura nos muestra información muy amplia del cómo las características físicas de los botones nos permiten la usabilidad de los mismos.

LECTURA 2: ¿Cuál es la diferencia entre Wireframe, Mockup y Prototype?

El proceso de diseño de un sistema requiere dedicarle tiempo ya que, prácticamente es con lo que el usuario va a interactuar. Si este no incluye todas las funcionalidades que el usuario pidió además de que no sea intuitivo, estaríamos a mi parecer, en un serio aprieto como desarrolladores. El proceso de diseño pasa por diversas partes a lo largo de su evolución, el primero con el que contamos es el sketch, el cual, prácticamente es un dibujo a mano en una hoja de papel, si bien no es un diseño complicado, sirve de base para el siguiente paso el cual es el wireframe, el cual es una representación un poco más compleja que la anterior, este también es sencillo, pero ya se puede apreciar un poco más visualmente lo que contendrá el sistema. Posteriormente tenemos el mockup, con este ya empieza a tomar aún mas forma el sistema ya que se agregan elementos como colores, fuentes, imágenes, logo, con esto ya se figura como una representación de lo que en realidad será el sistema. Después de todos los anteriores tenemos el prototipo el cual ya contiene todas o al menos la gran mayoría de las funcionalidades del sistema, a su vez este es iterativo, es decir, ya se puede navegar por él usando los botones que se colocan en el mismo, sin embargo, esto no quiere decir que sea funcional ya que como bien lo dice su nombre solo es un prototipo. Las diferencias son claras entre una parte del diseño y otra, y me atrevo llamarlas partes ya que como mencioné anteriormente, el diseño va evolucionando cuando se pasa de uno a otro. Se comienza por lo básico que es en papel, después se pasa a un boceto, posteriormente a algo que se podría considerar como un boceto más avanzado a mi parecer, ya que se agregan más elementos visuales a este y posteriormente a una representación iterativa, muy útil a mi parecer y que podría servir de mucho cuando le queremos presentar un proyecto a un cliente que no tiene tanta experiencia en el manejo de estos sistemas.

LECTURA 3: 7 RAZONES PARA MODELAR EL SOFTWARE ANTES DE PROGRAMAR

Esta lectura se centra en las razones del por qué es útil el modelado y diseño de software, entre ellas se encuentran: mejorar la productividad del equipo, esto es debido a que al hacer un modelado y basarse en ese, los distintos miembros del equipo pueden distinguir fácilmente que y cómo es lo que se está haciendo; disminuir los errores en el código: al tener bien estructurado el sistema a través de los diagramas permite observar de manera “física”, si es que se puede llamar así, el lugar donde está flaqueando el sistema de esta manera se pueden prevenir un poco más la aparición de errores al implementar código; formas de diseño: dado a que se está planeando una opción cabe la posibilidad de que ciertas partes puedan cambiarse dando espacio a una manera diferente pero igual de eficaz de implementar el sistema; agilizan la evolución del software: dado a que se tiene estructurado y plasmado en un documento el modelado y diseño permitirá identificar más fácilmente las características a mejorar; documentación simplemente aquí se capta todo acerca del modelado y diseño del sistema, la cual claramente es muy útil. Adicionalmente en la lectura se destacan otros dos puntos por los cuales el modelado y diseño de software son fundamentales, una de ellas es que muchos de los clientes van a requerir de esta documentación y otra es que el análisis de sistemas es bien pagado. Personalmente creo que el modelado y diseño de software a través de UML nos facilita un poco el camino, como lo explique en los puntos anteriores, por lo cual a pesar de requerir un poco más de tiempo a que solo dirigirse a programar, nos traerá beneficios a largo plazo.

Referencias

Bakusevych, T. (s.f.). *UX Collective* . Obtenido de UX Collective : <https://uxdesign.cc/button-design-user-interface-components-series-85243b6736c7>

Cabot, J. (7 de Noviembre de 2018). *Ingeniería de software* . Obtenido de Ingeniería de software : <https://ingenieriadesoftware.es/7-razones-modelar-software-programar/>

Warcholinski, M. (s.f.). *Brainhub*. Obtenido de Brainhub: <https://brainhub.eu/blog/difference-between-wireframe-mockup-prototype/>