



TECNOLÓGICO
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO
de Pabellón de Arteaga

ITEC

Instituto Tecnológico de Pabellón de Arteaga, Ags.

Lecturas (Unidad 4)

Tristán Nathaniel Huerta Valdivia

ITIC's 4to Semestre

Ingeniería de Software

Mtro. Eduardo Flores Gallegos

INDICE

LECTURA NÚM. 1 3

DISEÑO DE BOTONES: SERIE DE COMPONENTES UI..... 3

LECTURA NÚM. 2 4

¿CUÁL ES LA DIFERENCIA ENTRE WIREFRAME, MOCKUP Y PROTOTYPE? 4

LECTURA NÚM. 3 5

7 RAZONES PARA MODELAR EL SOFTWARE ANTES DE PROGRAMAR 5

Lectura Núm. 1

Diseño de botones: serie de componentes UI

Para diseñar la interacción correcta, debemos revisar el historial y el origen de los botones físicos, que son los predecesores directos de los componentes de la interfaz de usuario que se utilizan ampliamente en todos los productos digitales de hoy. Estos botones son geniales. Incluso si el usuario no comprende el mecanismo o algoritmo básico, el toque del dedo hará que el aparato eléctrico, el automóvil o el sistema se muevan. En "Botón de encendido", Rachel Plotnick rastrea los orígenes de la cultura de botones de hoy y describe la forma en que se presionan los botones como un medio de comando digital, lo que garantiza un control fácil, cuidadoso e infalible.

Algunas cosas a tomar en cuenta es el diferenciar un botón de un enlace, los enlaces se utilizan cuando navega a otro lugar y los botones se utilizan cuando realiza una acción, otra cosa es el estado del botón que ayuda a indicar a el usuario que pasa con el botón, existen al menos seis estados: Normal, Focus (Enfocado), Hover (Cuando se pasa el puntero sobre el botón), Active (Activo, es decir cuando le das clic), Prosses/Loading (Cargando) y Disabled (Deshabilitado). Además de esto los botones tienen muchas formas colores y estilos estos tienen importancia ya que por ejemplo: si usas un color rojo en algún botón le estas comunicando al usuario que si presionas ese botón pasara algo con un nivel alto de importancia como un botón rojo para borrar, también los estilos indican la importancia o el objetivo principal del sitio, por ejemplo si vas a publicar algo el botón que tiene que resaltar más debería ser el de publicar y los otros no deberían resaltar mas que este ya que seria raro y un error de diseño si por ejemplo el botón de editar resaltase más. Otro punto a tomar en cuenta es el hacer los

botones lo mas parecidos a un botón ya que es difícil distinguir si algo es o no un botón y eso seria otro error de diseño y por ultimo hacer los botones lo suficientemente grandes ya que esto facilita al usuario el uso de estos.

Lectura Núm. 2

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

Cuando trabajas con desarrolladores, a menudo escuchas términos como sketch, Wireframe, mockup, y prototipo, pero ¿entiendes realmente el significado de estos términos? ¿Sabes cuándo usar Wireframes o cuándo usar prototipos?

Comencemos con la razón por la que comenzamos con bocetos, estructuras alámbricas, modelos e incluso prototipos cuando comenzamos a diseñar

Sketch (Bosquejo): Básicamente, este es solo un dibujo a mano alzada ordinario en una hoja de papel que proporciona una representación de baja fidelidad para su aplicación. Esta es la forma más rápida de lluvia de ideas. Incluso un simple boceto puede describir su idea mejor que las palabras. Haga una lluvia de ideas, cambie los detalles y visualice sus ideas; todo depende de su imaginación. Este paso es crucial para la etapa de Wireframe.

Wireframe (estructura metálica): Un Wireframe es equivalente al marco o estructura simple de su sitio web / aplicación. Cada uno se utiliza para describir la función del producto y la relación entre las vistas (qué sucede cuando se hace clic en un botón). Por lo general, es en esta etapa que decide qué hacer (contenido / función) y dónde se encuentra en el sitio web o la aplicación. Este paso no cubre el diseño del producto.

Mockup (maqueta): En Brainhub, nunca comenzamos a crear aplicaciones antes de que se complete el Muckup. Con esta representación, puede comenzar el proceso de desarrollo y el desarrollador puede hacer realidad su Muckup.

Cualquier modelo proporcionará una representación de fidelidad media. Agregue colores, fuentes, texto (Lorem ipsum), imágenes, logotipos y otras formas que pueden afectar a Wireframe. El resultado es un mapeo estático de la aplicación. Al configurar este paso, tenga en cuenta las prácticas de interfaz de usuario. Si no tiene la capacidad de mover la Wireframe a la siguiente etapa, externalícela.

Prototype (Prototipo): Los prototipos proporcionan un rendimiento de alta fidelidad para sus aplicaciones. Es como una maqueta que contiene partes de UX, interacciones, animaciones y cualquier otra cosa que desee experimentar al hacer clic en un botón. Este paso no siempre es necesario para crear una aplicación. Si no es desarrollador, le sugiero que prepare un prototipo para presentar sus ideas a amigos, familiares y potenciales inversores. Lo único que falta es la función. Puede darle la sensación de usar aplicaciones reales, pero son solo imágenes conectadas entre sí.

Lectura Núm. 3

7 RAZONES PARA MODELAR EL SOFTWARE ANTES DE PROGRAMAR

Sé lo que muchos de ustedes están pensando. ¿Quién pierde el tiempo dibujando cuadros y flechas al escribir JavaScript? Si usted es uno de ellos, le daré cinco razones (más las dos "recompensas" que acaba de persuadir) para que pueda restaurar libros UML y actualizar sus conocimientos de modelado y diseño de software.

1. Las técnicas de modelado pueden aumentar la productividad de los equipos de desarrollo.
2. Reduce los errores o defectos durante la creación del código.
3. Con los modelos, puede explorar diferentes alternativas de diseño y arquitectura antes de comenzar a usar la tecnología de simulación para el desarrollo.

4. El modelo simplifica el mantenimiento y la evolución del proyecto.
5. El modelo puede capturar efectivamente el conocimiento del sistema.
6. Muchos clientes le pedirán que proporcione modelos de software desarrollados para ellos.
7. Los que hacen dibujitos cobran más que los que programan.