

Chapter 1: The Design Process 5

A high-level look at the design process 6

Research 7

The importance of research 8

Designing in an agile environment 10

Information architecture 10

Introducing flowchart development 11

Defining the shapes in flowcharts 12

Transitioning to wireframes 15

Usability testing 17

Visual design 17

Applying the visual layer 18

Delivery 19

# El proceso de diseño

Diseñar software puede ser una experiencia emocionante y satisfactoria. Pero, también puede ser un esfuerzo horriblemente caótico y frustrante. Habrá muchos desafíos mientras trabajamos para simplificar todas las complejidades de nuestro producto. Habrá muchas opiniones para considerar y comparar. Aunque desafortunados, algunos de nuestros compañeros de trabajo pueden intentar intimidarnos para que aceptemos su punto de vista sobre otro. También habrá momentos en que haya una falta total de opinión. A veces nadie puede ver lo que él o ella considera la solución obviamente correcta. Y, ocasionalmente, la visión del producto puede ser tan ambigua que nos deja sin una pista de lo que se supone que debemos diseñar.

La mejor defensa contra todas estas situaciones es un proceso de diseño bien definido y evangelizado. Este proceso nos permitirá contener parte de la confusión y la confusión que se producen naturalmente al crear un software. La única manera segura de tener éxito es trabajando juntos para resolver un conjunto definido de problemas en un orden lógicamente dirigido.

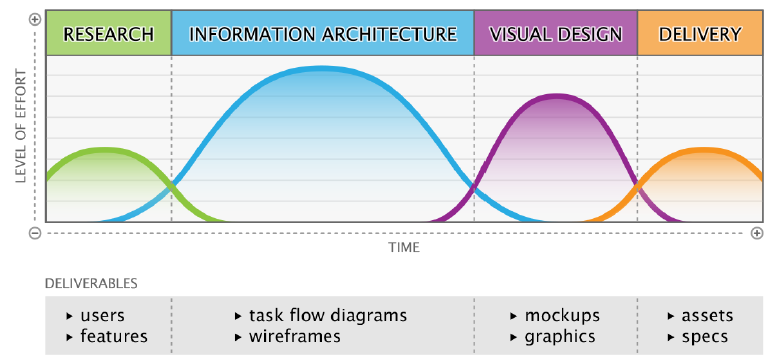
***La primera clave para emplear y mantener un proceso de diseño saludable es tener una comprensión de los pasos necesarios para el proyecto en el que estamos trabajando***. Necesitaremos averiguar ***qué técnicas nos ayudarán a obtener la información que estamos buscando***. También necesitaremos saber ***cómo medir cuándo es el momento adecuado para pasar de un paso a otro***. Será importante mantener la flexibilidad al evaluar cada nuevo proyecto. Para tener éxito, tendremos que adaptar el proceso de diseño para cada nuevo producto. Documentar y distribuir el proceso de diseño que pretendemos utilizar ayudará a establecer las expectativas. También ayudará a nuestro intento de generar estimaciones precisas de fecha de entrega que los gerentes de proyecto y los clientes esperen que entreguemos.

## Tópicos

* La importancia de la investigación en el proceso de diseño.
* Cómo trazar la estructura y el flujo de tareas de un sitio o aplicación.
* El proceso de wireframing de contenido, diseño y navegación específicos de la página requeridos para soportar las tareas que un usuario desea completar
* Directrices generales de diseño visual sobre la conversión de wireframes en maquetas de píxeles perfectos
* Lo que los desarrolladores de software necesitarán una vez que los diseños estén completos y listos para el desarrollo.

# Una mirada de alto nivel al proceso de diseño.

Las etapas de un proceso de diseño típico y el nivel de esfuerzo generalmente experimentado en cada paso se ilustran en el siguiente gráfico. Otros diseñadores pueden dividir esto de manera un poco diferente o pueden aplicar diferentes títulos a las etapas. Independientemente de esas pequeñas diferencias, existe un consenso general con respecto al flujo y la metodología comunes del proceso de diseño de UX.



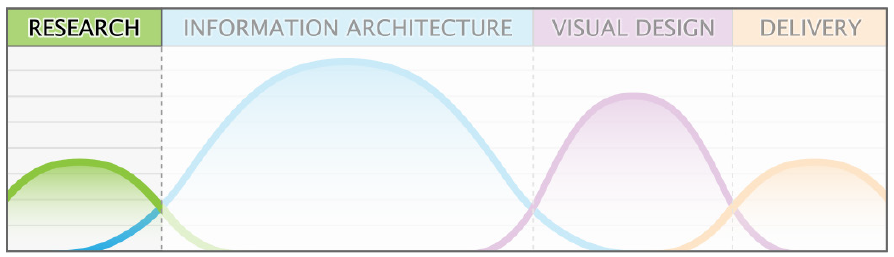
Por supuesto, el nivel real de esfuerzo dependerá de cada proyecto específico y del equipo con el que estemos trabajando. Sin embargo, esto debería darnos una idea general del esfuerzo requerido para producir los entregables enumerados después de cada etapa de este capítulo.

Comencemos con algunos de los detalles y examinando cada paso del proceso de diseño. Explicaré el objetivo de cada fase, le daré algunos consejos útiles, le señalaré algunas técnicas de uso común y describiré cómo determinar cuándo es el momento de pasar a la siguiente etapa del proceso.



# Investigación

Puede sorprender a algunos que el primer paso para diseñar no sea diseñar en absoluto, sino ***hacer preguntas***.



La presión para comenzar a diseñar lo antes posible está casi siempre vigente. Los diseñadores de software maduros, los desarrolladores y el personal de administración saben que ***el tiempo de investigación es una parte necesaria del proceso***. De hecho, es la manera de comenzar el proceso. Sin embargo, hay situaciones en las que incluso los profesionales experimentados olvidan la importancia de este primer paso. Quedan atrapados en el fervor para sacar el producto de la puerta y sucumbir a la presión de cortar esquinas al escatimar en la investigación que se requiere para establecer una base de información sólida para comenzar a construir nuestro software. Es esencial comenzar por obtener respuestas a varias ***preguntas clave***.

Estas preguntas son las siguientes:

* ¿Quién va a utilizar este software o sitio?
* ¿Qué tareas desea realizar el usuario?
* ¿Qué desea lograr el fabricante del software o del sitio? (No siempre lo mismo que la pregunta anterior)
* ¿Qué tecnología se utilizará? (¿Hay alguna limitación a considerar?)
* ¿Por qué el público usaría su software o sitio en lugar de otro?
* ¿Cuál es el contenido necesario para ayudar al usuario a lograr sus objetivos?

Si estamos rediseñando un sitio o aplicación existente, es probable que encontremos valioso buscar respuestas a estas preguntas adicionales:

* ¿Qué características o complejidades existentes obstaculizan o afectan negativamente la experiencia del usuario?
* ¿Qué funciones adicionales serían útiles para el usuario o editor en la ***próxima versión*** del producto?

Encontrar las respuestas a esta lista de preguntas puede requerir la aplicación de varias ***técnicas de investigación***. Nuestros esfuerzos de investigación pueden tomar la forma de un análisis competitivo para garantizar que nuestro producto tenga las características correctas o simplemente para entrevistar a quienes saben quiénes serán los usuarios finales esperados.

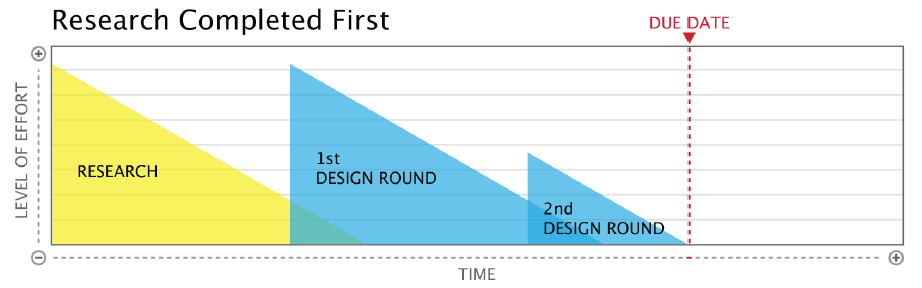
Algunas de las técnicas de investigación más utilizadas y efectivas se mencionan a continuación (consulte el tema 4, Técnicas de investigación para obtener más detalles):

* Clasificación de tarjetas
* Grupos de enfoque
* Encuestas de usuarios
* Entrevistas con partes interesadas (stakeholders)
* principios de diseño (tenets)
* Perfiles de usuario y Personas
* Investigación contextual
* Evaluaciones heurísticas.
* Análisis competitivo.

## La importancia de la investigación.

La calidad y la cantidad de investigación que completemos tendrá un impacto significativo en el éxito que le brindamos al usuario lo que necesita. También influirá en la cantidad de tiempo que lleva completar nuestros diseños.

Para ilustrar cuán constante es este problema, he incluido los siguientes dos gráficos, que creé hace unos 12 o 13 años. Aunque estaban dirigidos a abordar los problemas a los que me enfrentaba con un equipo específico, aún es relevante y vale la pena explicar a cualquier equipo o cliente con el que trabajará:



Este primer cuadro muestra cómo debería funcionar el proceso. La mayoría, si no toda la investigación, se ha completado por adelantado, es decir, antes de que comience el trabajo de diseño. Significa un ciclo de diseño bastante predecible. El diseñador conoce todos los problemas que necesita resolver. La revisión de la ***1ª RONDA DE DISEÑO*** generalmente produce algunos refinamientos necesarios, pero no más que eso. Las estimaciones de tiempo se cumplen, y todos son felices.

***Este segundo gráfico muestra cómo las cosas pueden salir mal y cómo se vencen las fechas de vencimiento***. Según mi experiencia, algunos clientes o partes interesadas no pueden pensar en todos los requisitos y características necesarios para comenzar un proyecto. Les hacemos todas las preguntas necesarias y ellos darán algunos de los detalles. Sin embargo, simplemente no pueden formular respuestas a las preguntas que estamos haciendo sin ver primero nuestra ronda inicial de diseños. Una vez que ven nuestro intento de disipar la ambigüedad en la sumisión con algunos bocetos o estructuras wireframing, se convierten en una verdadera fuente de información.

Cuando nuestros intentos de investigación rinden muy poco, es probable que involucremos al tomador de decisiones en la creación de algunas sesiones de esbozo. Entonces, haga estos bocetos rápidos, luego ensucie, pero asegúrese de que el cliente esté involucrado en el proceso. Si intentamos completar una ronda formal de diseños con información incompleta, es probable que nos demos cuenta de que hemos perdido el tiempo.

Hay tanto que necesita ser considerado al diseñar software. Cuando alguien se demora en presentar nuevos requisitos o funciones en el proceso, puede sentir que todo se debe desechar y comenzar de nuevo. Podemos ahorrarnos algo de la agonía asegurándonos de que la investigación se haya realizado y documentado a fondo. Luego, presentamos los resultados al cliente y al equipo para obtener su aprobación y aceptación. Asegurarse de que todos estén en la misma página desde el principio, con suerte, limitará la cantidad de sorpresas y cambios que se presentarán más adelante. Y, cuando lo hagan, será con el entendimiento de que estas solicitudes están alterando las expectativas existentes.

De esta manera, los cambios de programación se pueden discutir como una consecuencia natural.

## Diseñar en un entorno ágil.

Algunos diseñadores pueden encontrar difíciles de trabajar con metodologías de desarrollo ágil mientras diseñan soluciones integrales más grandes. ***Agile es una metodología de desarrollo iterativo que intenta hacer que un equipo de desarrollo produzca más rápido al reducir la cantidad de documentación y otros gastos generales***, recopilados históricamente antes de que pueda comenzar el desarrollo. Es una reacción a la antigua metodología de la cascada, que tradicionalmente el producto había sido diseñado en su mayor parte o totalmente diseñado y pensado antes de comenzar la producción. Este método requería mucha discusión y documentación que ralentizaba significativamente la producción. Aunque la metodología de cascada todavía está en uso, ha perdido el favor debido a su ritmo de entrega más lento.

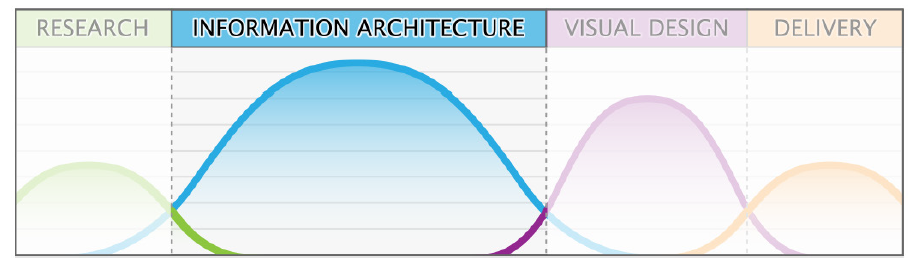
Con proyectos más pequeños, no debería haber un gran problema para que nuestra investigación se resuelva al principio. Sin embargo, los proyectos más grandes y complejos pueden ser un desafío. El diseño en un entorno ágil generalmente requiere obtener una buena ventaja para completar nuestros resultados de investigación y diseño antes de que el equipo de desarrollo lo necesite. Cuanto más lejos estemos, más tiempo tendremos para examinar y optimizar nuestro trabajo antes de entregarlo al equipo de desarrollo.

Para resumir, la calidad y cantidad de nuestra investigación tendrá un impacto directo y relacional en la calidad de la solución que creamos. Apresurarse a diseñar una solución sin detalles clave, como quién es nuestra audiencia o qué funciones podrían necesitar, significará muchas conjeturas que pueden o no tener éxito. Siempre me gusta pensar que si lo quieres mal, lo pillas mal.

***Independientemente de la metodología con la que estemos trabajando, es esencial que incluyamos el tiempo de investigación en nuestro plan de desarrollo y diseño***.

## Arquitectura informacional

Una vez que hemos respondido las grandes preguntas en la fase de investigación, pasamos a la parte de la ***arquitectura de la información*** del proceso de diseño:



Si bien estos pasos se han dividido en etapas diferentes, es natural que nuestra investigación continúe por un tiempo a medida que comenzamos a cambiar el enfoque. ***No necesariamente tiene que haber una ruptura limpia de un paso a otro***. Dependiendo del alcance y la complejidad del proyecto, podemos esperar que tengamos diferentes partes del proyecto en diferentes fases del proceso de diseño en cualquier momento. La excepción a esto es el primer punto en la siguiente lista. Nuestra investigación inicial debe estar dirigida a obtener suficiente información para ***trazar un diagrama completo de las tareas*** que los usuarios desean realizar mientras visitan el sitio o usan la aplicación.

Los ***objetivos*** de esta etapa son los siguientes:

1. Cree un mapa de alto nivel del sitio o aplicación.
2. Haga un mapa de las tareas que se encuentran en cada página o pantalla.
3. Defina el contenido requerido para soportar cada tarea.
4. Vetar y probar nuestros diseños.
5. Refinar nuestras soluciones de diseño.
6. Documentar los patrones de UX.

## Introducción al desarrollo de diagramas de flujo.

Esta fase está dedicada al esfuerzo de hacer que se diseñe la estructura de nuestro sitio o aplicación. Cuanto más complejo sea nuestro proyecto, más importante será pasar el tiempo necesario para trazar la estructura de la página y el flujo de tareas antes de pasar a otros pasos. Si estamos creando un sitio web de estilo folleto simple o una aplicación pequeña, se reduce la necesidad de una investigación exhaustiva y la documentación del flujo de tareas.

Sin embargo, es un buen hábito y ayuda a comunicar nuestro plan al cliente y/o al equipo. Si estamos trabajando en un sitio web complejo, una aplicación web u otras aplicaciones, es absolutamente fundamental que primero planifiquemos el flujo de tareas y las interacciones con las que se enfrentará el usuario cuando intente completar una tarea.

Debemos considerar la creación de un diagrama de flujo de tareas holístico o un mapa del sitio del producto, una de nuestras primeras preocupaciones principales. Si es necesario, podemos cerrar la puerta de nuestra oficina y producir este mapa solo en base a la investigación que hemos completado hasta la fecha. Hay situaciones en las que es mejor dejar de lado el ruido de la opinión para que podamos procesar todo para encontrar una solución recomendada. Sin embargo, recomendaría llamar a las partes interesadas y miembros importantes del equipo para una sesión de lluvia de ideas.

Descubrí que agiliza enormemente el proceso de mapeo cuando tenemos a todos en la misma sala hablando sobre posibles soluciones.

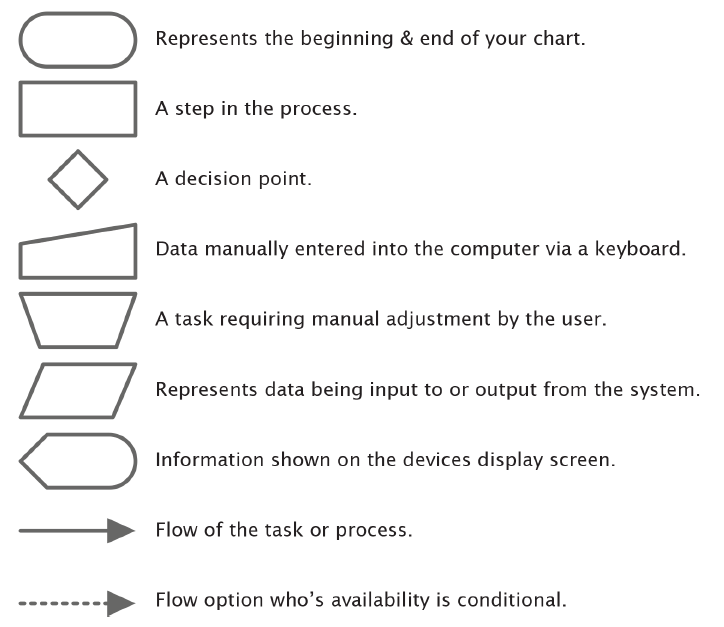
Puede ser difícil otorgar el crédito adecuado a los creadores de ciertas técnicas UX comúnmente utilizadas. Sin embargo, sabemos que el diagrama de flujo de procesos fue desarrollado originalmente por Frank Gilbreth Sr. y presentado a la Sociedad Americana de Ingenieros Mecánicos en 1921 (<http://en.wikipedia.org/wiki/Frank_Bunker_Gilbreth,_Sr>.).

El Sr. Gilbreth tiene una historia particularmente fascinante. Trabajó para refinar el mundo físico como lo hacen los diseñadores de UX en el mundo virtual. Su metodología de cartografía ha sido adoptada y modificada desde entonces para su uso en muchas industrias diferentes. Jesse James Garrett inventó la primera metodología de diagrama de flujo estandarizada específica para el diseño de UX en 2000. Se pueden encontrar más detalles en línea en el sitio web del Sr. Garrett (<http://jjg.net/ia/visvocab/>).

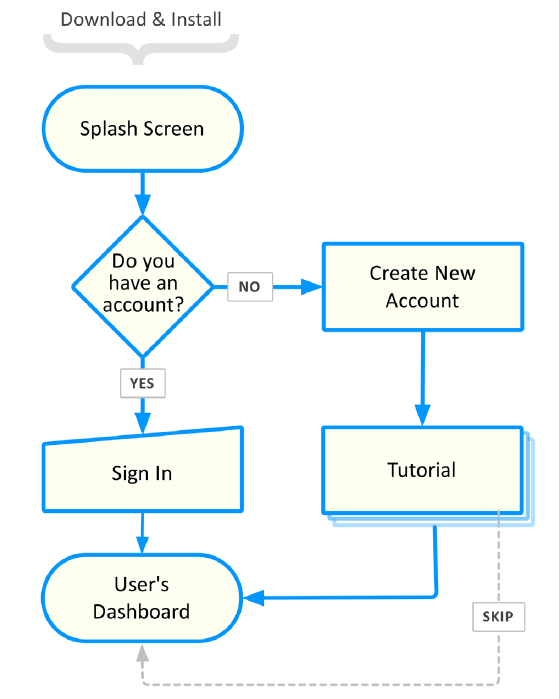
## Definiendo las formas en diagramas de flujo.

Si tuviéramos que buscar en Internet el significado de las formas de diagrama de flujo, encontraríamos miles de ejemplos y posiblemente algunas interpretaciones diferentes de lo que significa cada forma y calidad de línea. Adoptar y aplicar un vernáculo visual más profundo puede ampliar enormemente la cantidad de información que podemos incluir en nuestros mapas de interacción. Dicho esto, no deberíamos considerarlo un requisito para adoptar estos lenguajes de gráficos en su totalidad. Es bueno estar familiarizado con los estándares de la industria para la creación de diagramas de flujo, y si adoptamos o modificamos es perfectamente aceptable, siempre y cuando el flujo de información esté claramente definido y sea fácil de comprender de un vistazo. Entender los principios básicos de la creación de flujos de tareas debería ser suficiente para que comencemos.

Este es un ejemplo de algunas de las formas de diagrama de flujo más comunes y sus significados:



A continuación, muestro un diagrama de ejemplo:



***Este ejemplo de diagrama de flujo documenta la experiencia esperada al instalar un software.*** La tarea principal aquí es determinar si el usuario final tiene una cuenta existente o si necesita crear una nueva cuenta.

Como podemos ver en el diagrama anterior, cada rectángulo representa una página o tarea. Comienza en la parte superior del diagrama con la opción Descargar e instalar de la aplicación. El lector del documento simplemente tiene que seguir las flechas para ver las opciones disponibles para el usuario y los pasos subsiguientes que encuentran al tomar decisiones e ingresar datos.

Aquí, podemos ver la experiencia derivada cuando se pregunta al usuario si tiene una cuenta existente. Si lo hacen, se les pide que inicien sesión y se los lleva a su panel de control.

Si no tienen una cuenta existente, se les pedirá que creen una. Luego se les lleva a un tutorial para aprender a usar la aplicación. Parece que el tutorial consta de varias páginas, y al usuario se le dará la oportunidad de saltar e ir directamente a su panel de control. Al usar una línea discontinua, el gráfico parece sugerir que saltarse el tutorial no es el camino preferido que se desea que el usuario tome, pero está disponible.

Aunque solo es un pequeño fragmento de una experiencia más amplia, podemos comenzar a ver cuánta información se puede transmitir de un vistazo. Esto es particularmente importante cuando se trata de la bifurcación de decisiones. Cuantas más opciones ofrezcamos, más complicado será nuestro mapa. La experiencia comienza a complicarse exponencialmente si cada respuesta a una pregunta conduce a más preguntas. Agregue algunas de estas preguntas de derivación en una secuencia y nuestra experiencia sería extremadamente difícil de transmitir con una explicación basada en texto.

Examinemos la experiencia mundana de ingresar a una casa en el siguiente diagrama de flujo:



Comenzamos por entrar en la casa. Una vez en, inmediatamente tenemos muchas opciones para hacer.

Todos dependen de la dirección en la que decidimos movernos. Una vez que hemos tomado una decisión, nos espera otra serie de opciones únicas. Tómese un momento para pensar cómo describiríamos la misma experiencia utilizando solo texto. Ciertamente, se puede hacer, pero tomaría mucho más tiempo y procesamiento mental para que el lector lo entienda. La figura anterior ofrece una solución visual que puede entenderse de un vistazo.

Hace poco recibí un documento de especificación funcional de un compañero de trabajo que estaba administrando un proyecto en el que se esperaba que trabajara mi equipo de diseño. Explicó, con moderado detalle, cómo funcionaría el producto utilizando solo texto. Aunque no es un documento particularmente largo, nos tomó medio día leerlo en un intento por entender el proceso que estaba describiendo. Al final, ninguno de nosotros había comprendido completamente el proceso que estaba intentando expresar. Terminamos abandonando la experiencia y decidimos reunirnos con él para hablar sobre el tema. Después de conversar con él para obtener una imagen clara del flujo de tareas que pretendía, trazamos la misma experiencia en una sola página. Recortamos aproximadamente el 80 por ciento del texto, y terminamos con un documento fácilmente comprensible que pesaba una fracción del tamaño que tenía inicialmente.



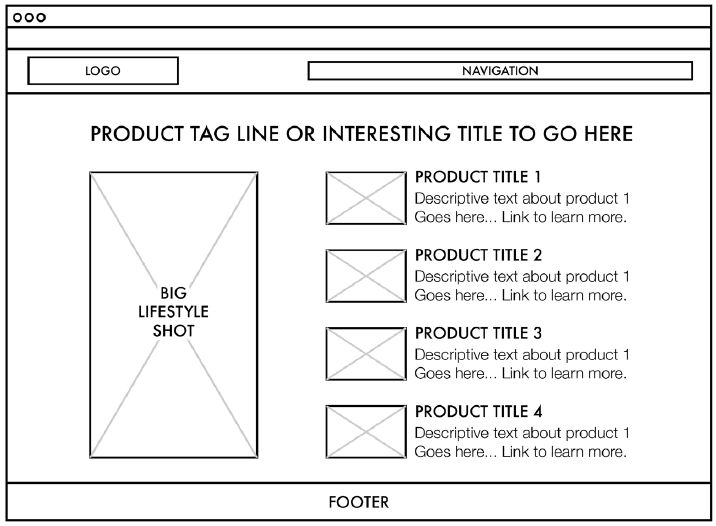
## Transición a wireframes

Una vez que las partes interesadas del proyecto han visto nuestro diagrama de flujo de tareas y están de acuerdo en que es el modelo con el que desean proceder, es hora de pasar a la etapa de wireframes (<https://es.wikipedia.org/wiki/Wireframe_(dise%C3%B1o_web)>.

Un wireframe es el modelo básico que ilustra la forma y la función central que se encuentra en una sola pantalla de su página web o aplicación. La fidelidad de estos wireframes aumentará en detalle a medida que los refinemos. Sin embargo, es probable que nuestra primera versión solo utilice contornos y formas en blanco y negro básicos para indicar dónde se colocarán en la pantalla los elementos de navegación, el texto y los gráficos. La recopilación de estos esquemas de alambre debe brindar una visión esquelética integral de todo nuestro producto.

En el diseño web, un wireframe o un diagrama wireframe es una representación visual en escala de grises de la estructura y funcionalidad de una sola página web o pantalla de aplicación móvil. Los wireframes se usan en las primeras etapas del proceso de desarrollo con el fin de establecer la estructura básica de una página antes de agregar el contenido y el diseño visual, y se puede crear con papel, directamente en HTML/CSS o con aplicaciones de software.

Aquí hay un ejemplo de un primer borrador de la página de inicio de un sitio web:



Como podemos ver al examinar la estructura alámbrica anterior, el contenido de la página admite una tarea principal: dirigir al usuario para que encuentre el producto sobre el que desea obtener más información.

Para apoyar esta tarea, hemos creado lo que llamaremos "puntos de acceso" a los diferentes productos, que se muestran aquí como imágenes, encabezados y enlaces. Sin embargo, no sabemos qué dirá el texto, qué contendrá la barra de navegación o cómo se verán los gráficos. Todo esto requiere más discusión y exploración, por lo que solo bloquearemos un espacio para ello y seguiremos adelante.

Este proceso puede ser mucho más fácil si estamos rediseñando un sitio o una aplicación existente, ya que generalmente se puede reutilizar gran parte del contenido. Sin embargo, si esta es la primera versión de nuestro producto, no debemos molestarnos con demasiados detalles para empezar. Solo imagine el tipo de contenido que será necesario para soportar las tareas que deben aparecer en la página.

A medida que comenzamos a iterar las versiones progresivas de estos wireframes al definir e ingresar el contenido de la página, la fidelidad y el detalle de nuestros wireframes aumentarán. A medida que avancen los wireframes, comenzaremos a ver dónde necesitamos solicitar o crear contenido. También tendremos que definir e incluir el modelo de navegación óptimo y las taxonomías de contenido en nuestros refinamientos de estructura alámbrica.

Ahora sería el momento de reunirse con el equipo de desarrollo para explicar los detalles del plan del proyecto actual y cualquier consideración técnica especial o características inusuales.

En este punto, tendremos que averiguar si planeamos que nuestro sitio optimice su diseño para el dispositivo específico en el que se está viendo (computadora de escritorio, tableta, teléfono u otros dispositivos móviles). **Esto se conoce como diseño responsivo**. Se ha convertido en el método estándar para crear sitios web. Significa que es probable que definamos cómo se cambiará el diseño y el diseño de nuestra página para cada tipo de pantalla.

El sitio web de ejemplo que he incluido en el siguiente lección está diseñado teniendo en cuenta la computadora de escritorio tradicional. Sin embargo, el aumento en el uso de dispositivos móviles hace que muchos centren sus esfuerzos de diseño en una metodología de "primero móvil". Esto significa que comienzan creando un diseño optimizado para un dispositivo móvil y luego expanden sus diseños para la optimización de escritorio en segundo lugar. Este método solo será más relevante a medida que aumente el uso del dispositivo móvil. Independientemente de su elección de la cual buscar primero, es probable que considere un diseño receptivo cuando diseñe sus wireframes.

Se ha escrito mucho sobre el tema del diseño responsivo y una técnica similar llamada diseño adaptativo en los últimos años. Hay muchos tutoriales en línea y tutoriales en video sobre el tema que pueden ayudarlo a comprender mejor el tema. Una búsqueda de "técnicas de diseño receptivo" debería ayudarlo a comenzar a aprender más.

## Pruebas de usabilidad

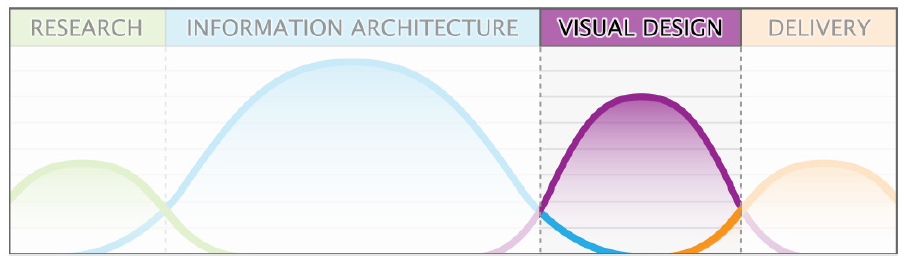
Aunque a menudo se guardan hasta después de que se hayan generado las maquetas, ahora es el momento de comenzar a probar la usabilidad de nuestros diseños. Si decidimos probar nuestros esfuerzos con prototipos en papel (consulte el tema 5, Arquitectura de la información y técnicas de diseño visual para obtener más detalles) o algo más formal, es importante examinar nuestras ideas mientras haya tiempo suficiente para cambiarlas. Si esperamos probar nuestros diseños hasta que se hayan desarrollado completamente en forma de maqueta o se hayan desarrollado por completo, a menudo hay muy poco que podamos hacer para cambiar la funcionalidad principal.

Aquí se mencionan algunas de las técnicas de cableado efectivas y de uso común (consulte el Lección 5, Arquitectura de la información y técnicas de diseño visual para obtener más detalles):

* Mapeo de la realidad
* Diagramas del mapa del sitio
* Diagramas de flujo de tareas basados ​​en la persona.
* Mapas de interacción de captura de pantalla
* Prototipos de papel.

# Diseño visual

Una vez que todos estemos de acuerdo con el diseño del flujo de tareas, la navegación y el diseño general de la página, pasaremos a la parte del diseño visual del proceso de diseño:



Dependiendo de cómo haya decidido llegar a este punto, ahora es generalmente el momento de hacer la transición de su aplicación de armado de alambre (Axure, Omnigraffle, Visio u otros) y abrir Photoshop para crear sus maquetas.

Las ***maquetas (mockups)*** creadas en este punto deben ser un intento de representar una representación de píxeles perfectos del producto final. Todo el contenido y los gráficos deben definirse y ponerse en marcha. Debo señalar aquí que el concepto de píxel perfecto está empezando a cambiar con la adopción de un diseño receptivo y una mayor interactividad del sitio web.

Cuando los sitios web eran un poco más estáticos y menos interactivos, era mucho más fácil crear maquetas que se tradujeran perfectamente de Photoshop a un sitio web. Aunque esto es algo por lo que hay que esforzarse, debe entenderse que las animaciones, las transiciones y las características interactivas crearán un objetivo móvil que será cada vez más difícil de capturar en cualquier aplicación de diseño que produzca imágenes estáticas.

## Aplicando la capa visual.

Como se mencionó anteriormente, el ***diseño UX*** es una carrera multidisciplinaria. A algunas empresas les resulta más fácil dividir el proceso de diseño mediante la contratación de arquitectos de información que obtienen los detalles en su lugar. Luego pasan sus archivos a los diseñadores gráficos que ajustan los diseños mediante el diseño de la capa visual.

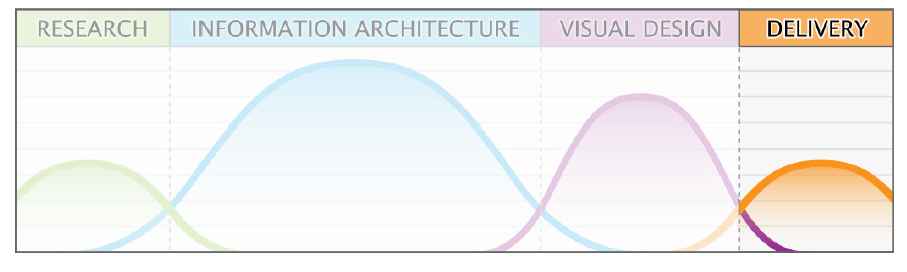
Cuando el mismo diseñador aplica los wireframes y el diseño visual, puede ser más fácil refinar los wireframes a un nivel más alto de fidelidad. Cuando nuestros wireframes comienzan a adoptar algunas de las cualidades finales de la maqueta, la transición a la fase de diseño visual puede ser mucho más fácil. La generación de maquetas es solo una extensión natural de lo que ya se ha definido. Sin embargo, si se va a dividir el trabajo, recomendaría dejar algo de espacio para que el diseñador gráfico explore soluciones visuales que se alejen un poco de los esquemas. Una buena manera de hacer esto es marcar los elementos cuya ubicación o propiedades no deben alterarse y dejar que el diseñador gráfico tenga dominio sobre el resto.

Los cambios de contenido en esta etapa son comunes. El texto y los gráficos se explorarán y actualizarán a medida que se refinen las maquetas. Sin embargo, ofrecería un consejo sobre las características adicionales y los cambios de funcionalidad que se producen durante la fase de diseño visual. Puede ser muy difícil volver al escenario de la estructura alámbrica una vez que hemos comenzado a producir maquetas. Es tentador seguir haciendo nuestro píxel perfecto.

Diseña y rueda estos cambios al mismo tiempo. Esto se puede hacer, y podría ser lo más inteligente si los cambios son menores. Sin embargo, una vez que comencemos a realizar cambios significativos en la arquitectura de la información, sería más rápido y más fácil detener nuestros esfuerzos de maquetas y examinar estos cambios en otro conjunto de estructuras de alambre. La razón es sobre todo la velocidad de ejecución. La fase de diseño gráfico tiene que ver con marcar los detalles visuales, lo que puede requerir un esfuerzo y tiempo significativos. Examinar los cambios de características al mismo tiempo puede ralentizar significativamente el proceso.

# Delivery

La fase de entrega del proceso de diseño puede tener lugar una vez que hayamos desarrollado nuestro contenido y las maquetas aprobadas por los interesados del proyecto:



Esta etapa básicamente se divide en tres tareas:

* Optimización de los gráficos para su uso en el sitio web o aplicación.
* Crear documentos de especificaciones que ayuden a los desarrolladores a construir lo que hemos diseñado.
* Revisar el trabajo de desarrollo completado para verificar que coincida con los diseños

Este último paso es, con mucho, el más difícil de los tres. Es probable que haya algunas diferencias visuales significativas entre los diseños y lo que se ha desarrollado.

Incluso cuando hayamos suministrado documentos de especificación que llamen los márgenes, el kerning (Kerning es el proceso de adición o eliminación de espacio entre pares de caracteres concretos. Tracking es el proceso de creación de expansión o compresión de un bloque de texto.), el liderazgo y otros atributos, las cosas serán ligeramente diferentes. El hecho del asunto es que el nivel de control que tiene sobre esas cosas en Photoshop es mucho mayor que el que tiene en un navegador web. Se ha ofrecido mucho más control a HTML5 y CSS3, pero a menudo aún no alcanzan lo que necesitas.

Este problema realmente ha llevado a una nueva trayectoria profesional en la industria del software llamada desarrollador de UX. Es para esa persona rara que tiene tanto la capacidad de codificar la interfaz como un ojo para el diseño. Si descubrimos que nuestro equipo tiene problemas importantes con la traducción de las maquetas al diseño final, podemos considerar contratar a alguien para que lo ayude en esta capacidad.

Dado que este es un problema bastante común, podemos esperar que todos los ojos estén fijos en el resultado final.

Podríamos argumentar que es responsabilidad de todos garantizar que el producto desarrollado coincida con las maquetas lo más cerca posible. Después de todo, había muchos ojos en los diseños a medida que se producían. Se expresaron muchas opiniones durante su creación y se llegó a un acuerdo final sobre el contenido, la navegación y su apariencia general. Sin embargo, la mayoría de las veces recae en el diseñador para supervisar los esfuerzos del intento del equipo de desarrollo de recrear lo que se representa en las maquetas.

En este punto, muchas personas tienen opiniones firmes sobre los detalles y matices del producto; sin embargo, estos parecen desvanecerse de las mentes de quienes los retuvieron una vez que el producto entró en la fase de desarrollo.

***El truco para resolver algo de esto antes de que comience es incluir a los desarrolladores antes en el proceso de diseño***. Tiende a ser natural que se acumule una pared entre los equipos de desarrollo y diseño. Después de todo, hablan idiomas completamente diferentes y se les llama en diferentes etapas del proceso de desarrollo de software. Será de gran beneficio para todos los interesados ​​si los incluimos temprano y con frecuencia.

Además, querremos asegurarnos de que participen en las primeras discusiones para que puedan influir en la tecnología o las tecnologías que deben utilizarse. Una discusión sobre las características deseadas y nuestras ideas iniciales sobre cómo pensamos que intentaremos crear la experiencia del usuario debería darles suficiente información para decidir qué tecnología usar. Su decisión debería darnos una mejor idea desde el principio en cuanto a qué limitaciones podemos tener y qué opciones tenemos a nuestra disposición.

Más allá de esto, debemos incluir al equipo de desarrollo en revisiones de diseño posteriores.

Esto les ayudará a comprender por qué se tomaron ciertas decisiones y señalar la importancia de ciertas partes de la interfaz que no deben modificarse. Asignar un punto de contacto principal del equipo de desarrollo que se incluye en las sesiones de lluvia de ideas y las revisiones de los diseñadores puede ayudar a nuestros equipos a mantenerse en la misma página sin interrumpir el programa completo del equipo de desarrollo.

Todo esto puede ayudar a evitar que el problema más grave de los diseños y las funciones se altere significativamente o se reduzca sin previo aviso. La excusa común es: "Sé que estás muy ocupado" y "No quería molestarte". Establezca desde el principio la expectativa con todo el equipo de que le gustaría involucrarse con cualquier cambio que se realice en la función, el flujo, la apariencia, y así sucesivamente. Es posible que haya estado documentando las decisiones sobre el producto hasta este momento, pero hubo muchos ojos en el trabajo y la aprobación es otorgada por todos. Si hay un cambio en lo que se aprobó, deberá ser discutido con las partes interesadas.