



Bootcamp de Desarrollo Web

Clase 20

Figma - parte 2


1



¿Qué es la usabilidad?

La usabilidad en UX es un **atributo** de calidad que evalúa la facilidad de uso de las interfaces de usuario.

La palabra "**usabilidad**" también se refiere a métodos para mejorar la facilidad de uso durante el proceso de diseño.



Introducción a la programación

2



Usabilidad

- Si un producto es difícil de usar, lo que quiere decir que va a tener muchas fricciones al momento de interactuar, la persona va a sentir de forma inevitable un estrés, frustración y malas emociones que lograrán que se aleje de nuestra solución y deje de usarlo.
- Si los usuarios se pierden en un Sitio Web, una Aplicación o un Software, simplemente se van.



Componentes de la usabilidad

La usabilidad se define por 6 componentes de calidad principales:

Utilidad: ¿Hace las funciones que los usuarios necesitan?

Capacidad de aprendizaje: ¿Qué tan fácil es para los usuarios realizar tareas básicas la primera vez que se encuentran con el diseño?

Eficiencia: Una vez que los usuarios han aprendido el diseño, ¿con qué rapidez pueden realizar las tareas?



Componentes de la usabilidad

Memorabilidad: Cuando los usuarios vuelven al diseño después de un período de no usarlo, ¿con qué facilidad pueden restablecer la competencia?

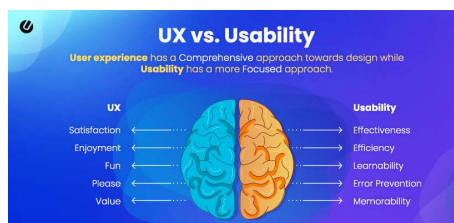
Errores: ¿Cuántos errores cometen los usuarios, qué tan graves son estos errores y con qué facilidad pueden recuperarse de los errores?

Satisfacción: ¿Qué tan agradable es usar el diseño?



Usabilidad vs Ux?

La **usabilidad** se refiere a la facilidad de uso de un producto: la facilidad con la que puede realizar una tarea determinada con el producto. En cambio, **UX** se refiere a la experiencia general que los usuarios tienen con el producto, de principio a fin.





Bootcamp de Desarrollo Web

Clase 20

Design Thinking

7



Design Thinking

El **Design Thinking** es una metodología orientada a la generación de soluciones a partir de un reto. Se divide en cinco fases:

Empatía, Definición, Ideación, Prototipado y Validación. Una vez terminado el proceso, puedes volver a cualquier de las fases desde la que empezar una nueva iteración.

Una de las características más importantes de esta herramienta es que está plenamente orientada al usuario. Por eso, los productos y servicios generados de forma adecuada a través del **Design Thinking** aportan valor a las personas. Ya que han sido diseñados con este fin.

8



Empatizar

Es la primera etapa del proceso, y la base del mismo. En la fase de Empatizar, definimos el arquetipo de usuario al que vamos a dirigirnos, establecemos nuestros objetivos de investigación y, a partir de ellos, decidimos qué técnicas de recopilación de información utilizaremos.

El objetivo de esta fase es identificar deseos y necesidades relevantes para el usuario.

Técnicas propias de la fase de Empatizar son la entrevista en profundidad, la observación y el Focus Group, entre otras.

Introducción a la programación



Definir

En ella, organizamos toda la información recopilada para identificar todas las áreas de oportunidad desde la que podemos ofrecer soluciones relevantes para los deseos y necesidades para el usuario.

La técnica más habitual en esta parte del proceso es el **Clusterizado o Saturar y Agrupar**, que consiste en primer lugar en el volcado de la información que hemos recopilado en post its.

En segundo, en la agrupación de esos post its de acuerdo a su contenido. Y, en tercero y último, en encontrar una frase que **sintetice** la información de cada uno de los grupos.

Introducción a la programación



Idear

Una vez establecido el reto (que puede ser también reformulado o refinado) pasamos a la parte de diseño de la solución. En esta parte del proceso, la primera fase por la que pasaremos será la de idear.

En la fase de Idear, o Ideación, se trata de dar el mayor número de ideas posibles que respondan al reto planteado.

Es el momento del brainstorming, y de otras herramientas que nos impulsen hacia el encuentro del mayor número de ideas posibles.

Introducción a la programación

11



Prototipar

A partir de las ideas generadas, se lleva a cabo una selección, y éstas pasan a prototiparse. La fase de prototipado es aquella en la que damos forma a las ideas, las tangibilizamos. De esa forma, se las podremos mostrar al usuario, quien podrá darnos feedback y decir en qué medida la solución que estamos diseñando para él se ajusta a sus necesidades o deseos.

Dependiendo del momento en el que nos encontremos, el prototipado será más o menos básico. Existen muchas formas de prototipar, como la prueba de concepto, el Storyboard o la Maqueta física.

Prototipamos para fallar rápido y barato, buscando que el usuario, cuanto antes, nos indique si el camino que estamos tomando en el diseño de la solución es adecuado o no..

Introducción a la programación

12



Validación

Es el momento en el que mostramos nuestro prototipo de solución al arquetipo para el que estamos diseñando. La fase de validación exige preparación. Tendremos que establecer los objetivos, construir la guía y, por último, mostrar al usuario nuestra solución.

En esta fase, es importante que entendamos que no estamos vendiendo. Se trata de aprender del feedback del usuario para hacer posteriormente una nueva versión mejorada de nuestra solución.

Introducción a la programación

13



Bootcamp de Desarrollo Web

Clase 20

Figma – MVP

14



¿Qué es un Minimum Viable Product (MVP)?

Un Minimum Viable Product (MVP) es un producto con el conjunto mínimo de funcionalidades que facilite a la empresa aprender lo máximo posible con el mínimo esfuerzo sobre sus usuarios potenciales, cómo resolver sus necesidades, y si estarían dispuestos a pagar por un producto que lo consiguiera.



Introducción a la programación

15



¿Qué es un Minimum Viable Product (MVP)?

Esta metodología permite el desarrollo incremental e iterativo de productos, con ciclos de desarrollo breves basados en la verificación de hipótesis con **feedback** e investigación de usuarios.

La principal funcionalidad del MVP es la de permitir al equipo de desarrollo de producto probar la viabilidad en el mercado de una idea de forma rápida y con un coste reducido. También es útil para recopilar información y el aprendizaje para saber hacia dónde ha de ir el desarrollo del producto.

Introducción a la programación

16



¿Cómo se define un Minimum Viable Product (MVP)?

1. Identificar los objetivos de negocio del MVP: Esto implica comprender los objetivos a largo plazo del producto. Se establecen metas concretas y medibles utilizando metodologías como SMART, y se definen criterios para validar las hipótesis iniciales.

2. Investigar las posibles implementaciones del MVP: Se realiza una investigación de usuario para identificar a quiénes se quiere ayudar, cómo interactúan actualmente con el producto, cuáles son sus puntos de dolor y posibles soluciones. Se generan ideas de funcionalidades que permitan cumplir con los objetivos establecidos.

Introducción a la programación

17





¿Cómo se define un Minimum Viable Product (MVP)?


3. Priorizar la funcionalidad del MVP: Se utilizan matrices de priorización para determinar qué funcionalidades se implementarán primero en el MVP, considerando el impacto en la experiencia del usuario y el coste de implementación.

4. Planificar el desarrollo: Se decide cómo llevar a cabo la construcción del MVP, teniendo en cuenta los recursos disponibles y la inversión deseada. Se puede optar por herramientas no code para acelerar el desarrollo inicialmente. Se establece un plan de desarrollo para evaluar el tiempo necesario para tener el producto listo en el mercado.

Introducción a la programación

18







Bootcamp de Desarrollo Web

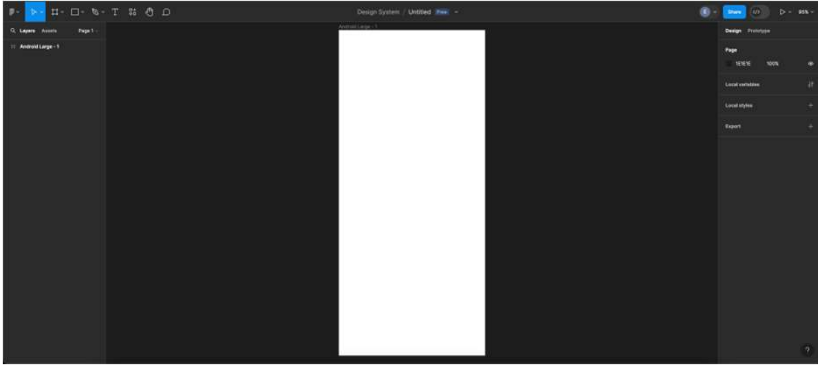
Clase 20

Figma - prototipo navegable

19



Crear nuevo proyecto



Introducción a la programación

20



Diseñar pantallas

Antes de crear un prototipo, necesitas diseñar las pantallas individuales de tu aplicación o sitio web. Utiliza las herramientas de diseño de Figma para crear elementos como botones, imágenes, campos de texto, etc.

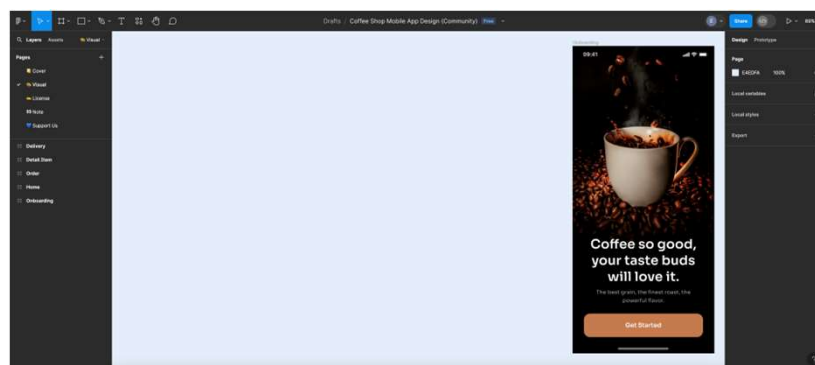
Cada grupo de pantallas deberá estar en una página separada dentro de tu proyecto para crear proyectos con orden.

Introducción a la programación

21



Diseñar pantallas



Introducción a la programación

22



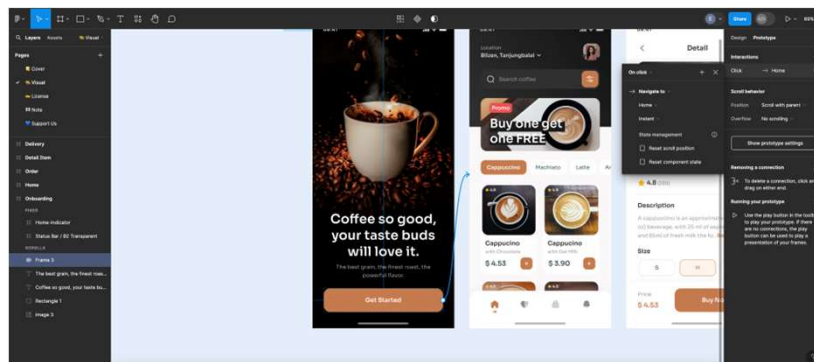
Crea enlaces entre pantallas

Para crear un prototipo, necesitas establecer vínculos entre las diferentes pantallas. Esto es lo que permitirá a los usuarios navegar entre ellas. Para hacerlo, sigue estos pasos:

- Selecciona un elemento en una pantalla que desees convertir en un enlace, como un botón.
- En el panel derecho, verás una pestaña llamada "Prototype" o "Prototipo". Haz clic en ella.
- Haz clic en el botón "Add Interaction" o "Agregar interacción" que se representa con un +.
- Selecciona la pantalla a la que desees vincular. Puedes elegir una pantalla específica o simplemente "Previous" para volver a la pantalla anterior.
- Define la acción de activación (por ejemplo, "On Tap").
- Ajusta las transiciones y animaciones según tus preferencias.



Crea enlaces entre pantallas





Compartir prototipo

Ahora que ya sabemos cómo hacer un prototipo en Figma y lo tenemos preparado, puedes compartirlo con otros colaboradores o clientes para que puedan visualizar y testear el resultado. Figma te permite generar un enlace, acceso público o agregar personas a tu proyecto para que puedan ver y comentar sobre el diseño.

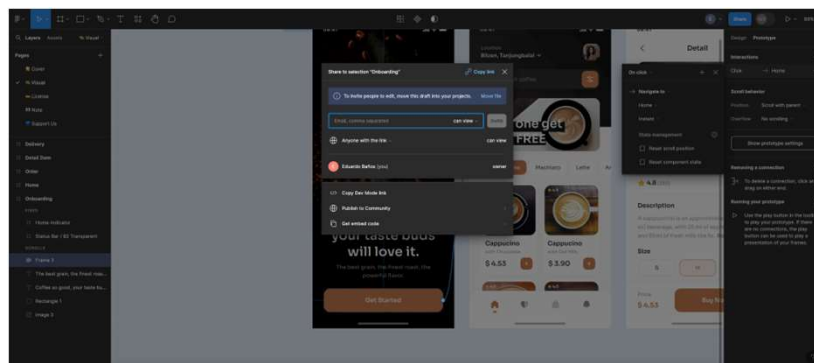
Para general el link simplemente tenemos que dirigirnos al botón Share ubicado en la parte superior derecha de la interfaz. Al hacer clic podemos acceder a la siguiente pantalla dónde podemos invitar a otros usuarios o generar un link para su visualización.

Introducción a la programación

25



Compartir prototipo



Introducción a la programación

26



Bootcamp de Desarrollo Web

Clase 20

Figma – parte 2