

# Diseño de Interfaces Únicas

martes, 17 de enero de 2023 23:41

Mientras más compleja se vuelve la información, mucho más compleja se vuelve la interfaz; comunicar toda la información es lo más importante.

¿Interfaz? Es la conexión que se da de manera física y a nivel de utilidad entre dispositivos o sistemas.

## COMUNICACION EFECTIVA

¿De qué información requieren las diferentes interfaces para poder ser útiles para el usuario?

UX (User Experience): Se refiere a la experiencia, basada en los sentimientos, que un usuario tenga a la hora de interactuar con un producto o servicio.

La **experiencia** se **basa en los sentimientos** que genera

De la experiencia forma parte cualquier interacción que el usuario tenga con el flujo del sistema.

La funcionalidad (resolver el problema) es la cosa más importante de un producto digital o físico!

UI (User Interface): Se refiere a la vista, o, interfaz la cual permite interactuar al usuario con el sistema.

Las interfaces son la comunicación humano – maquina entre 2 o más sistemas y tienen un límite a la hora de interactuar.

Gestos: Formas con la cual se interactúa con la interfaz de no forma no convencional (apretar un botón)

Poseen una forma de entrada (interacción) y posteriormente la instrucción.

Física: Botón físico

Digital: Pantallas de celulares

Auditivas: Google home, Alexa

Estructura de flujo de una interfaz:

1. Se genera una entrada
2. Se declara una función
3. Se ejecuta una instrucción.

## PRODUCTO DIGITAL

Definición escrita del producto: Tener muy claro que va a hacer la aplicación ¿Cuál es el objetivo que se va a lograr?

¿Qué es mi producto? ¿Qué soluciona?

Productos complejos con soluciones simples

**Antes de estructurar el producto digital definir el flujo de interacción, para así poder ver de formas más clara las funcionalidades que este debería de cumplir.**

¿De dónde vienen mis usuarios?

## Foundations y principios:

**Foundations:** Anatomía de la aplicación, puntos bases fundamentales de nuestras interfaces: Colores, tipografía, espaciado, branding, border radius.

Tipografías: Consultar en <https://fonts.google.com>

Espaciado: Hace que todo tenga un orden y un ritmo

**Principios de diseño:** Definir principios que debe de seguir la aplicación (<https://principles.design>)

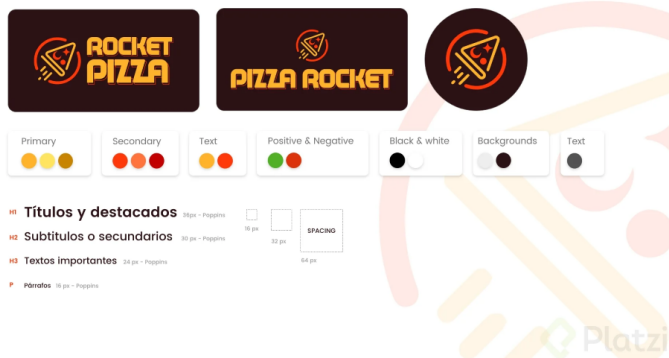
- Direct Manipulation: Manipulación directa del software, no hay procesos que el no puede controlar
- What u see is what u get: No esconder funcionalidades de la aplicación, mostrar al usuario las funcionalidades de la misma.
- Feedback and dialogue: Mantener al usuario informado con que acaba de pasar. Decirle al usuario como salirse de la situación siempre y cuando sea posible.

**Material Design** (<https://m3.material.io>)

Dentro de la creación de foundations esta{

1. Creación de logo
2. Colores: Primarios (colores que mas definen mi marca), Secundarios, Black & White, Positive & Negative (respuestas a peticiones / acciones del usuario), Backgrounds
3. Clasificación de textos: Tamaño y tipografía

}



Los foundations se van editando y enriqueciendo a medida del desarrollo de la aplicación, los foundations se terminan una vez el paquete de aplicación ya este creado.

Mapa de flujo del producto: (<https://coggle.it/?lang=es>)

- Usuarios finales que interactúan con la aplicación
- Que información puede llegar a necesitar el usuario
- Que secciones va a tener la aplicación
- Que procesos va a llevar a cabo la aplicación
- Definir cada una de las acciones que se llevan a cabo en el flujo

COMO DISEÑADOR SE TIENE LA MISION Y VISION DE PRESENTAR LA INFORMACION DE LA MANERA MAS FACIL PARA QUE EL USUARIO LA CONSUMA.



**Wireframing** (esta entre el boceto y el producto final): Definir de forma rápida el flujo de un producto, sin preocuparse demasiado en pantallas finales, para poder tomar decisiones sobre los flujos (si estan bien o mal)

**No se usan colores (va a blanco y negro), solo se definen posiciones. Se diseñan únicamente las cosas necesarias para comunicar mi idea.**

1. El Wireframe tiene la función de definir la **jerarquía**, ¿Qué elementos son más o menos importantes dentro de la aplicación?

Se puede generar jerarquía a través de: El tamaño, el color, la posición.

2. También define la **estructura**: Todos los componentes de la aplicación deben de tener un nombre, una forma, una función definida (una estructura)

3. Componentes: Deben de ser reutilizables y tener una forma en común (tener una forma estándar)

4. Foundations

5. El wireframe es estático (no tiene animaciones), solo define la arquitectura visual y la arquitectura de la información de la aplicación.

1. Niveles de resolución: Dependiendo de la necesidad de comunicación al equipo

- Low level
- Medium level
- High level

2. Arquitectura de la información:

- Jerarquía
- Textos: Saber en qué contexto y cuanta cantidad de texto va a haber en el cuadro
- Grupos / componentes

3. Flujos

- Information flow: Como viaja la información para el usuario
- Visual flow: Transiciones de pantalla a pantalla
- Function review: Que funciones va a tener el componente

PROTOTIPO

El prototipo debiera de comunicar la:

- Animación: Comunicar cual será la entrada y salida de un componente, flujo o animación.
- Comportamiento

- Interacción y retroalimentación: Que pasa cuando el usuario accione de una u otra forma con la aplicación (on key press, on enter press, on key rage).
- Estados: Diferentes estados que puede tomar el componente y como los comunicamos.
- **Funcionalidad y flujo**

Tratar de que todos los botones tengan un lugar hacia donde ir

#### PATRONES Y COMPONENTES

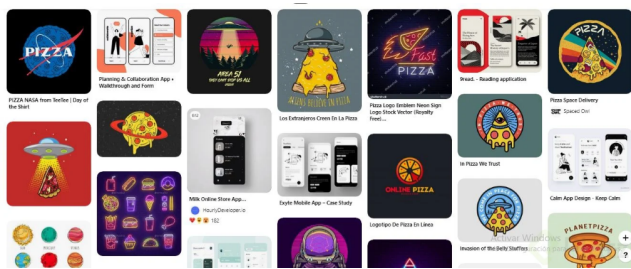
**Patrones UI:** Formas más comunes de aplicar una función en una interfaz, estructura ya definida para la funcionalidad.

**Componentes:** Software que hace funciones pequeñas dentro de todo el software

Los componentes deben de tener:

- Nombre
- Descripción
- Comportamientos e interacciones
- Estados

Hacer un moodboard con referencias visuales y conceptos para el diseño de la interfaz, se puede hacer uso de pinterest.



- Ejemplo de Moodboard

El ámbito visual debería de comunicar que quiere lograr esa aplicación

**MICROINTERACCIONES:** Interacción que tiene la persona con el Software, informar al usuario que está pasando en el software a partir de la interacción con él.

Fases:

- Triggers / disparadores: Desencadenadores de las micro interacciones
- Reglas: Cosas que se tiene que cumplir para la ejecución de la micro interacción
- Retroalimentación: Información que da el elemento al usuario una vez ejecutada la micro interacción.
- Bucles o modos

#### JERARQUIA VISUAL:

¿Cuál es el elemento que más quiero que vean?

- Posición
- Tamaño
- Color

#### ORDEN DE PANTALLAS Y BUENAS PRACTICAS

Secciones de forma vertical y flujos de forma horizontal, o, secciones de forma horizontal y flujos de forma vertical.

**Uso de gradientes:** Escalas o espectros de tonos entre 2 colores de forma progresiva.

#### Diseño de iconos.

Medir la efectividad de los componentes y el diseño por medio de KPI's y mediciones de las interacciones de los usuarios con el componente.

¿Qué variable o concepto evalúa el éxito de lo que estoy diseñando?

**UI Kit:** Documento con todos los elementos de la UI.

**Starter:** Documento para que otros diseñadores comiencen a diseñar desde cero siguiendo unas reglamentaciones definidas.

Documentar cada uno de los componentes y copiarlos del UI Kit al Starter

Para estudiar:  
Sistemas de diseño  
Documentación Avanzada

Resumen de los pasos:

1. Bocetado
2. Construcción del flujo de la aplicación
3. Wireframing
4. Moodboard, definicion de estilo de la aplicacion
5. Selección de color, tipografia, espaciado, etc (Mockup estatico)
6. Prototipado
7. Selección final del estilo de la interfaz