

PRÁCTICA: DISEÑO DE APLICACIÓN/PÁGINA WEB EN FIGMA

INFORMACIÓN GENERAL

Modalidad: Individual o por parejas

Herramienta: Figma (cuenta gratuita suficiente)

OBJETIVO

Diseñar una aplicación web o móvil completa aplicando metodología profesional de diseño UX/UI, demostrando dominio de herramientas de Figma, sistemas de diseño, layout grids, componentes reutilizables e interacciones.

COMPETENCIAS A EVALUAR

- Organización profesional de archivos y estructura de trabajo
 - Proceso de diseño iterativo desde wireframes hasta diseño final
 - Uso correcto de Layout Grids con columnas, filas y márgenes
 - Creación de componentes reutilizables con variantes
 - Aplicación de sistema de color documentado y accesible
 - Tipografía y jerarquía visual bien estructurada
 - Interacciones y prototipado funcional
 - Calidad visual y profesionalismo en la ejecución
-

ESTRUCTURA DEL PROYECTO

Organización del Archivo Figma

Es fundamental mantener una estructura ordenada y profesional. Crear un archivo con las siguientes **páginas organizadas**:

1. **Cover** - Portada con información del proyecto (nombre, autores, descripción breve)
2. **Wireframes** - Bocetos de baja fidelidad que muestran la estructura básica
3. **Design System** - Colores, tipografías y componentes base documentados
4. **Diseño Final** - Pantallas de alta fidelidad con el diseño completo
5. **Prototipo** - Flujos e interacciones conectadas
6. **Referencias** - Inspiración, mood boards y documentación de decisiones

¿Por qué es importante esta organización?

Una estructura clara facilita la navegación, demuestra profesionalismo y permite que otros diseñadores (o tú mismo en el futuro) comprendan rápidamente el proyecto. En entornos profesionales, esta organización es estándar.

PROCESO DE TRABAJO (7 FASES)

FASE 1: PLANIFICACIÓN Y ORGANIZACIÓN

Actividades:

- Crear archivo Figma con estructura de páginas mencionada anteriormente
- Definir el tipo de aplicación que diseñarás (e-commerce, gestión de tareas, red social, etc.)
- Identificar el público objetivo de tu aplicación
- Crear página Cover con información del proyecto
- Establecer convenciones de nomenclatura para frames y elementos

Consejos prácticos:

- Usa nombres descriptivos y consistentes (ejemplo: "btn-primary", "card-product", "screen-home")
 - Define desde el inicio si diseñarás para móvil, desktop o ambos
 - Escribe en la página Cover el problema que resuelve tu aplicación
-

FASE 2: WIREFRAMES DE BAJA FIDELIDAD

Actividades:

- Crear bocetos simples de las 5 pantallas principales de tu aplicación
- Usar únicamente cajas grises, texto placeholder y elementos básicos
- Definir el layout y la distribución de elementos en cada pantalla
- Establecer el flujo de navegación básico entre pantallas
- Evitar colores finales, imágenes o tipografías definitivas

¿Qué son los wireframes y por qué son importantes?

Los wireframes son esquemas visuales que muestran la estructura y jerarquía de la información sin distracciones visuales. Son como el plano arquitectónico de una casa: muestran dónde irá cada elemento antes de decidir colores o decoración. Esto permite:

- Validar la estructura de información rápidamente
- Hacer cambios sin perder tiempo en detalles visuales

- Concentrarse en la usabilidad y funcionalidad
- Comunicar ideas de forma clara al equipo o cliente

Elementos comunes en wireframes:

- Rectángulos grises para imágenes
 - Líneas para texto
 - Cajas para botones
 - Símbolos básicos para iconos
-

FASE 3: SISTEMA DE DISEÑO

Actividades:

A. Paleta de colores:

- Definir color primario (identidad de marca, acciones principales)
- Definir color secundario (acciones complementarias, énfasis)
- Establecer escala de grises (fondos, textos, bordes)
- Crear colores de estado (éxito, error, advertencia, información)

B. Verificación de contraste:

- Usar herramientas como Contrast Checker para validar accesibilidad
- Asegurar contraste mínimo AA según WCAG: 4.5:1 para texto normal
- Esto garantiza que personas con problemas visuales puedan leer tu contenido

C. Documentación:

- Crear un mood board o página de referencias con inspiración visual
- Explicar la razón detrás de la elección de colores

D. Tipografías:

- Seleccionar máximo 2 familias tipográficas (una para títulos, otra para cuerpo)
- Definir escala tipográfica clara: H1, H2, H3, H4, H5, H6, body, caption
- Establecer pesos y tamaños para cada nivel

E. Crear estilos en Figma:

- Crear "Color Styles" para cada color de tu paleta
- Crear "Text Styles" para cada nivel tipográfico
- **Ventaja:** Al cambiar un estilo, se actualiza automáticamente en todo el proyecto

¿Por qué crear un sistema de diseño?

Un sistema de diseño es como una guía de construcción que asegura consistencia visual en todo el proyecto. Permite:

- Mantener coherencia entre todas las pantallas
 - Acelerar el proceso de diseño (no reinventar cada elemento)
 - Facilitar el trabajo en equipo
 - Simplificar futuras actualizaciones
-

FASE 4: LAYOUT GRIDS Y ESTRUCTURA

Actividades:

- Configurar grid para móvil: 4 columnas, margen 16-20px, gutter 16-20px
- Configurar grid para desktop: 12 columnas, margen 60-80px, gutter 20-24px
- Aplicar grids a todos los frames principales
- Alinear todos los elementos importantes al grid

¿Qué es un layout grid y por qué usarlo?

Un layout grid es una retícula invisible que divide la pantalla en columnas, filas y márgenes. Funciona como una cuadrícula que ayuda a:

- Alinear elementos de forma consistente
- Crear diseños balanceados y profesionales
- Adaptarse fácilmente a diferentes tamaños de pantalla
- Mantener proporciones armónicas

Terminología:

- **Columnas:** Divisiones verticales del espacio
- **Margen:** Espacio entre el borde de la pantalla y el contenido

- **Gutter:** Espacio entre columnas
- **Filas:** Divisiones horizontales (opcional, útil para alineación vertical)

Ejemplo práctico: En móvil con 4 columnas, un elemento puede ocupar 2 columnas (mitad del ancho), 3 columnas (tres cuartos) o las 4 columnas (ancho completo).

FASE 5: COMPONENTES REUTILIZABLES

Ejemplos de componentes obligatorios:

A. Botones:

- Crear variantes: Default, Hover, Disabled, Pressed
- Configurar Auto Layout para que el botón se ajuste al texto
- Establecer padding interno consistente

B. Campos de formulario:

- Crear variantes: Empty, Filled, Error, Focused
- Incluir estados de validación con mensajes
- Mantener altura y estilos consistentes

C. Cards o Tarjetas:

- Diseñar para contenido repetitivo (productos, publicaciones, perfiles)
- Usar Auto Layout para adaptabilidad
- Incluir variantes si es necesario (con/sin imagen, destacada, etc.)

Configuración de componentes:

- Crear componentes principales (Main Components) en la página Design System
- Usar Auto Layout para elementos que necesiten ajustarse dinámicamente
- Establecer constraints (restricciones) correctos para el responsive
- Nombrar variantes de forma clara y descriptiva

¿Por qué usar componentes?

Los componentes son elementos reutilizables que se actualizan automáticamente en todas sus instancias. Si cambias el color de un botón en el componente principal, todos los botones del proyecto se actualizan instantáneamente. Esto:

- Ahorra tiempo de diseño
- Garantiza consistencia absoluta
- Facilita mantenimiento y actualizaciones
- Simula cómo funcionan los componentes en código real

Auto Layout explicado: Es una función de Figma que permite que los elementos se ajusten automáticamente. Por ejemplo, si un botón tiene Auto Layout y cambias el texto de "Enviar" a "Enviar mensaje", el botón se expande automáticamente para acomodar el texto nuevo manteniendo el padding.

FASE 6: DISEÑO DE ALTA FIDELIDAD

Actividades:

- Aplicar el sistema de color definido previamente
- Insertar tipografía siguiendo la jerarquía establecida
- Usar instancias de los componentes creados (nunca duplicar elementos)
- Añadir imágenes reales o placeholders de alta calidad
- Mantener consistencia visual entre todas las pantallas
- Cuidar meticulosamente el espaciado y la alineación usando los grids

Aspectos clave:

- **Espaciado consistente:** Usar múltiplos de 4 u 8 píxeles (4px, 8px, 16px, 24px, 32px)
- **Jerarquía visual:** El elemento más importante debe destacar más
- **Balance:** Distribuir elementos de forma equilibrada en el espacio
- **Respiración:** Dejar espacio en blanco suficiente, no saturar la pantalla

Imágenes e iconos:

- Usar imágenes de alta resolución
 - Mantener estilo consistente en iconos (mismo grosor de línea, mismo estilo)
 - Optimizar tamaño de imágenes para diferentes contextos
-

FASE 7: INTERACCIONES Y PROTOTIPADO

Actividades:

- Conectar el flujo principal entre pantallas usando la herramienta Prototype
- Crear interacciones en componentes (hover effects en botones)
- Configurar transiciones apropiadas (tipo de animación y duración)
- Establecer el punto de inicio del flujo (pantalla inicial)
- Probar el prototipo completo en modo presentación

Tipos de interacciones básicas:

- **On Click:** Al hacer clic, navega a otra pantalla
- **On Hover:** Al pasar el cursor, cambia el estado del elemento
- **On Drag:** Permite arrastrar elementos (útil para sliders)
- **After Delay:** Acción automática después de X tiempo

Transiciones recomendadas:

- **Instant:** Sin animación, cambio inmediato
- **Dissolve:** Fundido entre pantallas (sutil y profesional)
- **Move In/Out:** Deslizamiento direccional (útil para navegación jerárquica)
- **Push:** Una pantalla empuja a la otra (común en apps móviles)

Duración sugerida: 200-400ms (animaciones rápidas y fluidas)

¿Por qué prototipar?

Un prototipo interactivo permite:

- Testear la experiencia de usuario antes de programar
 - Presentar ideas de forma convincente a clientes o stakeholders
 - Identificar problemas de navegación o usabilidad
 - Simular la aplicación real para recibir feedback
-

ESPECIFICACIONES TÉCNICAS

Layout Grids

Dispositivo	Columnas	Margen	Gutter
Móvil	4	16-20px	16-20px
Tablet	8	40px	20px
Desktop	12	60-80px	20-24px

Tipografía

Elemento	Tamaño
H1	32-48px
H2	24-32px
H3	20-24px
Body	16px
Caption	12-14px

Contraste Mínimo (Accesibilidad WCAG)

- **Texto normal:** 4.5:1 (Nivel AA)
- **Texto grande (más de 18px):** 3:1 (Nivel AA)
- **Elementos no textuales:** 3:1 (iconos, botones, bordes)

¿Por qué es importante el contraste?

Un contraste adecuado asegura que personas con discapacidades visuales, daltonismo o simplemente bajo condiciones de luz desfavorable puedan leer y usar tu aplicación sin dificultad. Es un requisito legal en muchos países y una buena práctica de diseño inclusivo.

ENTREGA

Archivos requeridos:

1. Archivo Figma descargado:

- En Figma, ir a: Menú principal (esquina superior izquierda) → File → Save local copy (.fig)
- Subir el archivo .fig descargado a tu repositorio de GitHub
- Incluir el enlace al repositorio en la entrega

2. Link del prototipo:

- Compartir enlace en modo presentación: Share → Get link → Set to "Anyone with the link can view"
- Formato: [https://figma.com/proto/...](https://figma.com/proto/)

Verificación antes de entregar:

- Comprobar que los enlaces son públicos o están compartidos con el instructor
 - Probar el prototipo completo en modo presentación
 - Revisar que todas las páginas estén correctamente nombradas y organizadas
-

EVALUACIÓN

La práctica se evaluará según la siguiente rúbrica:

Criterio	Peso
Organización del archivo (Páginas y Secciones)	10%
Proceso de diseño (Wireframes → Final)	15%
Layout Grids (Columnas, Filas, Márgenes)	15%
Sistema de Color	10%
Tipografía y Jerarquía	10%
Componentes Reutilizables	20%
Interacciones y Prototipado	15%
Calidad Visual y Profesionalismo	5%

Niveles de evaluación:

- **Excelente (90-100%):** Cumple todos los requisitos con calidad profesional, demostrando dominio avanzado de las herramientas y conceptos.
 - **Bueno (70-89%):** Cumple la mayoría de requisitos con buena calidad, con pequeñas áreas de mejora en ejecución o consistencia.
 - **Aceptable (50-69%):** Cumple requisitos básicos con calidad limitada, presenta inconsistencias o fallos en algunos aspectos importantes.
 - **Insuficiente (<50%):** No cumple requisitos mínimos, muestra falta de comprensión de conceptos fundamentales o ejecución deficiente.
-

RÚBRICA DETALLADA DE EVALUACIÓN

1. Organización del archivo (Páginas y Secciones) - 10%

Excelente (90-100%): Archivo perfectamente organizado con páginas claramente nombradas (Cover, Wireframes, Diseño Final, Componentes, Referencias). Frames ordenados y agrupados lógicamente. Convenciones de nomenclatura consistentes en todo el proyecto.

Bueno (70-89%): Archivo organizado con la mayoría de páginas requeridas. Algunos frames sin orden óptimo. Nomenclatura generalmente consistente con pequeñas inconsistencias.

Aceptable (50-69%): Organización básica presente pero confusa. Faltan algunas páginas importantes. Nomenclatura inconsistente que dificulta la navegación.

Insuficiente (<50%): Sin organización clara. Todo en una página. Sin estructura de nomenclatura. Imposible navegar eficientemente.

2. Proceso de Diseño (Wireframes → Final) - 15%

Excelente (90-100%): Evolución clara desde wireframes de baja fidelidad hasta diseño final. Muestra iteraciones y mejoras progresivas. Wireframes funcionan como blueprint efectivo del diseño final.

Bueno (70-89%): Wireframes presentes y diseño final completo. Evolución visible pero con algunos saltos. Wireframes algo básicos pero funcionales.

Aceptable (50-69%): Wireframes muy básicos o incompletos. Conexión débil entre wireframe y diseño final. Falta documentar proceso iterativo.

Insuficiente (<50%): Sin wireframes o directamente diseño final. No se evidencia proceso de diseño iterativo.

3. Layout Grids (Columnas, Filas, Márgenes) - 15%

Excelente (90-100%): Grids perfectamente configurados según tipo de dispositivo (4 col móvil, 12 col desktop). Márgenes y gutters consistentes en todo el proyecto. Elementos perfectamente alineados al grid.

Bueno (70-89%): Grids configurados correctamente. Algunos elementos no alineados perfectamente. Márgenes y gutters mayormente consistentes.

Aceptable (50-69%): Grids presentes pero mal configurados. Muchos elementos ignoran el grid. Inconsistencia en márgenes.

Insuficiente (<50%): Sin grids o configuración incorrecta completa. Elementos sin alineación. Diseño caótico.

4. Sistema de Color - 10%

Excelente (90-100%): Paleta definida con colores primarios, secundarios y neutros. Excelente contraste (WCAG AA mínimo). Referencias documentadas. Uso consistente en todo el proyecto. Estilos de color creados en Figma.

Bueno (70-89%): Paleta definida y mayormente consistente. Buen contraste general. Referencias documentadas básicamente. Algunos estilos creados.

Aceptable (50-69%): Paleta básica pero inconsistente en su aplicación. Problemas de contraste menores. Referencias incompletas. Sin estilos.

Insuficiente (<50%): Sin paleta definida. Colores aleatorios. Sin considerar accesibilidad. Sin documentación.

5. Tipografía y Jerarquía - 10%

Excelente (90-100%): 1-2 familias tipográficas bien elegidas. Escala tipográfica clara (H1-H6, body, caption). Jerarquía visual evidente. Excelente legibilidad. Estilos de texto creados.

Bueno (70-89%): Tipografías apropiadas. Jerarquía presente pero mejorable. Buena legibilidad. Algunos estilos de texto creados.

Aceptable (50-69%): Demasiadas tipografías o muy pocas variaciones. Jerarquía débil. Legibilidad aceptable. Sin estilos.

Insuficiente (<50%): Tipografías inadecuadas o inconsistentes. Sin jerarquía. Problemas de legibilidad. Caótico.

6. Componentes Reutilizables - 20%

Excelente (90-100%): Mínimo 3 componentes bien diseñados (botones, inputs, cards). Uso de variantes para diferentes estados (default, hover, disabled). Propiedades bien configuradas. Componentes usados consistentemente en todo el diseño.

Bueno (70-89%): 2-3 componentes creados con algunas variantes. Configuración básica correcta. Uso mayormente consistente.

Aceptable (50-69%): 1-2 componentes muy básicos. Sin variantes o mal configuradas. Uso inconsistente.

Insuficiente (<50%): Sin componentes o copias repetidas sin convertir a componente. No aplica concepto de reutilización.

7. Interacciones y Prototipado - 15%

Excelente (90-100%): Flujo completo conectado entre todas las pantallas. Interacciones en componentes (hover states). Transiciones apropiadas y fluidas. Prototipo testeable y funcional. Uso inteligente de triggers y acciones.

Bueno (70-89%): Flujo principal conectado. Algunas interacciones en componentes. Transiciones básicas. Prototipo funcional.

Aceptable (50-69%): Conexiones básicas entre pantallas. Pocas interacciones. Transiciones genéricas. Prototipo limitado.

Insuficiente (<50%): Sin conexiones o muy pocas. Sin interacciones en componentes. Prototipo no funcional.

8. Calidad Visual y Profesionalismo - 5%

Excelente (90-100%): Diseño pulido y profesional. Espaciado consistente. Alineación perfecta. Imágenes e iconos de alta calidad. Coherencia visual total. Demuestra atención al detalle.

Bueno (70-89%): Diseño profesional con pequeños detalles mejorables. Buena coherencia visual. Espaciado generalmente bueno.

Aceptable (50-69%): Diseño funcional pero básico. Varios problemas de alineación o espaciado. Coherencia limitada.

Insuficiente (<50%): Diseño descuidado. Múltiples problemas visuales. Sin coherencia. Aspecto no profesional.

CONSEJOS PARA EL ÉXITO

Buenas prácticas:

- **Nombra todo correctamente:** Layers, componentes y páginas con nombres descriptivos y consistentes facilitan enormemente la navegación y el trabajo colaborativo.
 - **Usa Auto Layout estratégicamente:** Para elementos que necesitan ajustarse dinámicamente al contenido (botones, cards, listas). Esto simula el comportamiento del código real.
 - **Aprovecha los estilos de Figma:** Crear Color Styles y Text Styles ahorra tiempo y garantiza consistencia. Un cambio en el estilo se refleja automáticamente en todo el proyecto.
 - **Documenta tus decisiones:** Explica en la página Referencias por qué elegiste ciertos colores, tipografías o enfoques de diseño. Esto demuestra pensamiento estratégico.
 - **Prueba tu prototipo exhaustivamente:** Navega por todos los flujos posibles antes de entregar. Verifica que no haya enlaces rotos o comportamientos inesperados.
 - **Trabaja con múltiplos de 4 u 8:** Para espaciados y dimensiones (4px, 8px, 16px, 24px, 32px). Esto crea ritmo visual y facilita la implementación en código.
 - **Usa constraints (restricciones) correctamente:** Para que tus diseños sean responsive y se adapten a diferentes tamaños de pantalla.
-

Errores comunes a evitar:

- **Usar demasiadas tipografías:** Máximo 2 familias. Más tipografías crean confusión visual y parecen poco profesionales.
 - **Ignorar los grids:** Elementos flotando sin alineación clara destruyen la armonía visual y dificultan la lectura.
 - **No crear componentes:** Copiar y pegar elementos repetitivos es ineficiente y rompe la consistencia cuando necesitas hacer cambios.
 - **Colores sin contraste suficiente:** Hace tu diseño inaccesible y dificulta la lectura, especialmente en dispositivos móviles bajo luz solar.
 - **Prototipo incompleto:** Enlaces rotos, pantallas sin conexión o flujos que no tienen sentido lógico confunden al usuario.
 - **Archivo desorganizado:** Sin estructura clara, páginas sin nombres descriptivos, layers sin agrupar. Dificulta enormemente la evaluación y el mantenimiento.
 - **Espaciado inconsistente:** Usar valores aleatorios (13px, 19px, 27px) en lugar de un sistema coherente crea caos visual.
 - **Olvidar los estados interactivos:** Botones sin hover, campos de formulario sin estado de error o focus. Los usuarios esperan feedback visual.
-

GLOSARIO DE TÉRMINOS

Auto Layout: Función de Figma que permite que los elementos se ajusten automáticamente según su contenido, similar a Flexbox en CSS.

Component (Componente): Elemento reutilizable que se puede instanciar múltiples veces. Los cambios en el componente principal afectan a todas sus instancias.

Constraints (Restricciones): Reglas que definen cómo se comportan los elementos cuando cambia el tamaño del frame (izquierda, derecha, centro, escala, etc.).

Gutter: Espacio entre columnas en un layout grid.

Instance (Instancia): Copia vinculada de un componente principal. Hereda las propiedades del componente pero puede tener overrides.

Layout Grid: Sistema de columnas, filas y márgenes que ayuda a alinear elementos de forma consistente.

Margin (Margen): Espacio entre el borde del frame y el inicio del contenido.

Prototype (Prototipo): Versión interactiva del diseño que simula la navegación y comportamiento de la aplicación final.

Variants (Variantes): Diferentes versiones de un mismo componente (ejemplo: botón default, hover, disabled).

WCAG: Web Content Accessibility Guidelines - Estándares internacionales para accesibilidad web.

Wireframe: Boceto de baja fidelidad que muestra la estructura y jerarquía sin detalles visuales finales.