



**INSTITUTO SUPERIOR
TECNOLÓGICO QUITO**

Excelencia en Educación Superior

Guía general de DISEÑO DE INTERFACES



DESDE 1984 - QUITO
**SABER
HACER**
Bien
SUPERIOR TECNOLÓGICO QUITO

ISBN: 978-9942-7328-4-2



9 789942 173284 2



GUÍA GENERAL DE DISEÑO DE INTERFACES

AUTOR: GABRIEL HOYOS

PRIMERA EDICIÓN

AÑO: 2024

TRABAJO EN EDICIÓN:



EDITOR INTERNO: DIEGO J BASTIDAS

EDITOR EXTERNO: DAVID FABIAN CEVALLOS SALAS

Este material está protegido por derechos de autor. Queda estrictamente prohibida la reproducción total o parcial de esta obra en cualquier medio sin la autorización escrita de los autores y el equipo editorial. El incumplimiento de esta prohibición puede conllevar sanciones establecidas en las leyes de Ecuador.

Todos los derechos están reservados.

ISBN: 978-9942-7328-4-2



SOBRE EL AUTOR



Gabriel Hoyos Ingeniero de sistemas graduado en la mención desarrollo de aplicaciones para la gestión en la Universidad Politécnica Salesiana en el año 2018. Su carrera ha estado marcada por una vasta trayectoria en proyectos relevantes en el manejo, supervisión, soporte e implementación de ERP's así como desarrollo y diseño de los mismos. Su experiencia y habilidades técnicas le han permitido destacarse en el ámbito de la informática, desarrollo y la tecnología gracias a una sólida comprensión de los sistemas de planificación de recursos empresariales (ERP), con experiencia en la implementación, configuración y personalización de varias plataformas ERP para satisfacer las necesidades específicas de las organizaciones.

He liderado y gestionado proyectos de implementación de ERP desde la fase de planificación hasta la ejecución y el soporte post-implementación, asegurando el cumplimiento de plazos y presupuesto soy experto en trabajar con stakeholders para identificar y documentar requisitos de negocio, lo que me permite adaptar el ERP para mejorar la eficiencia operativa y apoyar los objetivos estratégicos de la empresa utilizando mi conocimiento en ERP's para analizar y optimizar procesos de negocio, reduciendo costos y aumentando la productividad mediante la automatización y la mejora continua.

Además de proporcionar soporte técnico experto y de capacitación a usuarios finales, ayudándolos a maximizar el uso del sistema ERP y resolver cualquier problema técnico que pueda surgir mediante un profundo conocimiento de las tecnologías y herramientas utilizadas en el desarrollo de interfaces de software, incluyendo lenguajes de programación, bibliotecas, y frameworks de front-end como HTML, CSS, JavaScript, React, Angular, entre otros.

Al ser capaz de descomponer conceptos técnicos complejos en términos más simples que permitan facilitar la comprensión de los estudiantes, utilizando ejemplos claros y relevantes desarrollando planes de estudio que aborden tanto los fundamentos como las tendencias actuales en el desarrollo de interfaces, asegurando que los estudiantes adquieran conocimientos actualizados y aplicables. Lo cual me permite ofrecer orientación y apoyo a los estudiantes, ayudándolos a superar desafíos y a desarrollar sus habilidades de resolución de problemas y pensamiento crítico.

Esto a su vez le ha permitido mantenerse al día con las últimas tendencias y avances en el desarrollo de interfaces de software, incorporando nuevos conocimientos y tecnologías en la enseñanza siendo esta la pauta para comprometerse con el campo del desarrollo y trabajar con otros departamentos y disciplinas que me han permitido integrar el diseño, desarrollo e implementación de interfaces en un contexto más amplio, mostrando cómo se relaciona con otros aspectos dentro del desarrollo de software y la tecnología.



CONTENIDO

GUÍA GENERAL DE DESARROLLO DE INTERFACES	7
1. DESCRIPCIÓN DE LA ASIGNATURA	7
2. BIBLIOGRAFÍA	7
2.1. Básica	7
2.2. Complementaria	8
3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS	9
4. OBJETIVO GENERAL	9
5. FORMACIÓN CIUDADANA, VALORES Y HABILIDADES BLANDAS	9
6. NORMAS DE CLASE	10
7. SISTEMA DE EVALUACIÓN	11
8. UNIDADES	11
UNIDAD 1 (Introducción a Diseño de Interfaces)	11
Temas y Subtemas	11
Orígenes de la Relación Persona-Ordenado	12
Hitos Importantes en la Historia de la Interacción Persona-Ordenador	13
Iteración en el diseño web	14
Interacción por comandos	14
Interacción basada en formularios	15
Primera Iteración: Prototipo Inicial	15
Pruebas de Usuario y Recopilación de Comentarios	18
Segunda Iteración: Ajustes Basados en Comentarios	19
Nuevas Pruebas y Validación	19
Tercera Iteración: Optimización Final	19
Implementación y Lanzamiento	19
Autoevaluación 1	23
Resumen de la Unidad 1	26
UNIDAD 2 (Experiencia de Usuario)	28
Temas y Subtemas	28
Historia y Origen de UX	28
Por qué se llama UX	29
Evolución de UX	29
UX (Experiencia de Usuario)	30
UI (Interfaz de Usuario)	31
Importancia de UX	39



CTA (Call To Action) Llamada de acción en el diseño web	40
Diseño Visual	40
Importancia del CTA	42
Características de CTA	43
Visibilidad y Detectabilidad:	43
Claridad y Brevedad.....	43
Relevancia y Contexto:	43
Urgencia.....	43
Modelo AIDA.....	44
Modelo AIDA aplicado a CTA.....	44
Pruebas y Ajustes.....	45
Tipos de CTA	45
CTA de Conversión Directa	45
CTA de Generación de Leads	46
CTA de Información o Aprendizaje.....	46
CTA de Redes Sociales	46
CTA de Retención o Fidelización.....	46
CTA de Retroalimentación.....	47
Autoevaluación 2.....	48
Resumen de la Unidad 2.....	51
UNIDAD 3 (HTML, CSS y JS).....	53
Temas y Subtemas.....	53
HTML: El Lenguaje Base de la Web	53
HTML (HyperText Markup Language)	53
Elementos Clave de HTML.....	53
Estructura básica de un documento HTML	54
Elementos HTML Importantes.....	54
Ejemplo Completo de un Documento HTML.....	55
CSS: El Maquillaje de la Web	57
¿Por qué es importante CSS?	57
Elementos clave de CSS	57
La sintaxis básica.....	58
Ejemplo de un documento CSS.....	58
¿Qué es JS o JavaScript?	62
Fundamentos de JavaScript.....	62

Características Principales de JavaScript.....	62
Estructuras de JavaScript.....	63
Características Distintivas.....	63
Aplicaciones de JavaScript.....	64
¿Qué es la validación de formularios?	64
¿Por qué es importante la validación de formularios?	64
Reglas de validación	65
Mensajes de error	65
Mecanismos de validación	65
Ejemplo de validación en JavaScript.....	65
Bootstrap: Un Framework Fundamental en el Desarrollo Web Frontend.....	67
Características	67
Beneficios de su uso	68
Integración de Bootstrap a proyectos web	68
Categorías Principales de Clases y Atributos.....	69
Ejemplo Práctico 1	70
Ejemplo Práctico 2	71
Personalización	74
Plantillas y Temas	74
Integración de Bootstrap en Proyectos React: Una Guía Técnica.....	76
¿Por qué usar Bootstrap en React?.....	76
Dependencias	77
Métodos de Integración	77
➤ Instalación.....	77
➤ Importación y Uso	77
Autoevaluación 3.....	79
Resumen de la Unidad 3.....	81
9. ANEXOS.....	83

ÍNDICE DE FIGURAS

Figura 1: Herramienta FIGMA Área de Diseño (Drafts).....	16
Figura 2: Herramienta de diseño de prototipado FIGMA	16
Figura 3: Herramienta FIGMA y sus componentes	17
Figura 4: Herramienta FIGMA en modo Desarrollador	18
Figura 5: Proceso de edición de modelado para la presentación de gestión de proyectos en FIGMA.....	19





Figura 6: Proceso Lean UX	20
Figura 7: Representación de la Experiencia de Usuario (UX)	34
Figura 8: Anatomía de un producto Digital Deseable	35
Figura 9: Pirámide de Diseño UX	37
Figura 10: Modelo de Trabajo sobre la Pirámide de UX.....	38
Figura 11: Metodología de Diseño UX (iteración)	38
Figura 12: Importancia de UX para las Empresas.....	39
Figura 13: Call to Action Simple (CTA) en página de Salesforce.....	41
Figura 14: Call to Action Smart CTA en página de venta de Vehículos	42
Figura 15: Modelo AIDA para Diseño de CTA.....	44
Figura 16: Interprete CodeOpen del Ejemplo planteado	56
Figura 17: Mejora estética con Estilos CSS	58
Figura 18: BootstrapMade plantillas prefabricadas para diseño.	74
Figura 19: Previsualización de Diseño BootstrapMade.....	74
Figura 20: StartBootstrap Plantillas prefabricadas para diseño.....	75
Figura 21: Previsualización de Diseño StartBootstrap	75

ÍNDICE DE TABLAS

Tabla 1: Solucionario unidad 1	83
Tabla 2: Solucionario unidad 2	84
Tabla 3: Solucionario unidad 3	85



GUÍA GENERAL DE DESARROLLO DE INTERFACES

1. DESCRIPCIÓN DE LA ASIGNATURA

El Diseño de Interfaces es una disciplina que se encarga de la creación y desarrollo de interfaces visuales y funcionales para distintos tipos de sistemas, aplicaciones y dispositivos. Estas interfaces son la conexión entre los usuarios y los sistemas, por lo que su diseño es fundamental para garantizar una experiencia de usuario satisfactoria.

En esta asignatura exploraremos los fundamentos del Diseño de Interfaces y su importancia en el mundo actual, así como las técnicas y herramientas necesarias para crear interfaces atractivas, intuitivas y fáciles de usar. También estudiaremos los diferentes aspectos que influyen en el diseño de interfaces, como la usabilidad, la accesibilidad, la experiencia de usuario y la interacción.

A lo largo del curso, trabajaremos en proyectos prácticos que nos permitirán aplicar los conceptos y técnicas aprendidos, y desarrollar habilidades en el diseño de interfaces para distintos dispositivos y sistemas. Además, aprenderemos a evaluar y mejorar interfaces existentes para hacerlas más efectivas y atractivas.

2. BIBLIOGRAFÍA

2.1. Básica

- Wood, D. (2015). *Diseño de interfaces: (1 ed.)*. Parramón Paidotribo S.L.
<https://elibro.net/es/ereader/itq/226898?page=18>
- Experiencia de Usuario: Principios Fundamentales

Este texto aborda los principios fundamentales para diseñar con éxito interfaces de usuario la mismas deben tener una comunicación visual clara y eficaz. El diseño de interfaces ayuda a conseguirlo mediante una serie de casos de estudio incisivos, entrevistas con diseñadores profesionales y consejos prácticos para producir en distintas interfaces digitales diseños front-end centrados en el usuario. El autor Dave Wood presenta los principales elementos del diseño gráfico para medios digitales, incluyendo la composición, el color, la iconografía, las imágenes y la tipografía, y explica cómo se combinan estos fundamentos de la comunicación visual para generar experiencias interactivas positivas. Este libro es una introducción imprescindible al desarrollo de interfaces que gusten a los usuarios, lleno de consejos prácticos para mejorar la





comprensión del trabajo de diseñadores y programadores a su vez ofrece un interesante punto de vista sobre el diseño interactivo para los cinco sentidos.

- Allanwood Gavin & Beare Peter (2021). *Diseño de experiencias de usuario I. Londres-Reino Unido.: Parramón; 1er edición (8 Septiembre 2021)*

Este texto contiene actividades de diseño lo que representa el paso ideal para la explicación clara de los conceptos y métodos fundamentales que deben aplicarse al diseñar experiencias de usuario en el entorno web , móvil y diseños que marcan tendencias multiplataforma. cubriendo así temas esenciales como la investigación del usuario y el diseño de experiencias pasando por la estética, los estándares y los prototipos, diseños en base a las experiencias de usuarios demuestran por qué los métodos centrados en el usuario son hoy en día fundamentales para garantizar el éxito de una amplia variedad de proyectos de diseño. Basándose en los estándares de los servicios digitales, la integración y el mapeado de situaciones hipotéticas. Es decir, esto constituye un manual básico para estudiantes y no diseñadores que necesiten una introducción al pensamiento contemporáneo y a los enfoques comunes en torno a la experiencia de usuario. Diseñando específicamente para los recién llegados las actividades prácticas: plantillas, entrevistas de profesionales, notas de colaboradores y fuentes de consulta para quienes inician en el diseño.

2.2. Complementaria

- Córcoles Tendero, J. E. (2015). *Diseño de interfaces web: (ed.). RA-MA Editorial.*
<https://elibro.net/es/lc/itq/titulos/62487>

Los contenidos del texto están orientados para una clara comprensión de casos prácticos sobre el diseño de interfaces y como ha evolucionado a lo largo del tiempo. De tal forma que al finalizar cada unidad, usted encontrará ejercicios que le ayudarán a identificar su avance en la comprensión de los temas técnicos con respecto a la materia además es un clásico sobre usabilidad web y ofrece consejos prácticos para crear interfaces intuitivas.

- Steve Krug (2006). *No me hagas pensar. Una aproximación a la usabilidad en la Web.* Madrid: Editorial Pearson Educación.

Los contenidos están estratégicamente organizados para presentar una gran cantidad de ejemplos aplicables de los métodos empleados en el diseño web, y sobre todo, el uso y





aplicación de los elementos recogidos en una investigación del autor. De esta manera, el lector adquiere la capacidad de analizar los resultados obtenidos de un diseño y su respectivo levantamiento de requerimientos.

- Tidwell , Jenifer (2010). *Designing Interfaces: Patterns for Effective Interaction Design*. USA.: Editorial O'Reilly Media; N.º 2 edición

Este texto se centra especialmente en patrones de diseño prácticos para crear interfaces efectivas, con ejemplos aplicables a una variedad de plataformas y los temas relacionados con el diseño web. En él, encontrará una serie de ejercicios propuestos, así como ejercicios resueltos explicados paso a paso.

- Norman, Donald (2024). *El diseño de las cosas cotidianas Madrid*: Editorial: CAPITAN SWING

Este texto contiene esta guía de diseño centrado en el ser humano muestra que la usabilidad es tan importante como la estética además de tener ejercicios propuestos para adquirir destrezas en el diseño de sitios web que el lector puede consolidar y complementar su estudio proporcionando una visión general del diseño de interfaces.

3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS

Desarrollar aplicativos informáticos mediante el uso de plataformas actuales de programación.

4. OBJETIVO GENERAL

Capacitar a los estudiantes en la planificación, diseño y evaluación de interfaces de usuario intuitivas, eficientes y estéticamente agradables para aplicaciones de software. Los estudiantes adquirirán una comprensión sólida de los principios de diseño de interfaces, así como la capacidad de aplicar herramientas y técnicas modernas de diseño de forma efectiva. Además, se fomentará la consideración de las necesidades y preferencias del usuario final, así como la optimización de la experiencia del usuario en diferentes plataformas y dispositivos.

5. FORMACIÓN CIUDADANA, VALORES Y HABILIDADES BLANDAS

EDUCACIÓN AMBIENTAL: CONSUMO Y PRODUCCIÓN SOSTENIBLES

VALORES & HABILIDADES BLANDAS: AMOR: COMUNICACIÓN ASERTIVA Y ESCUCHA ACTIVA



VALORES & HABILIDADES BLANDAS: COMPROMISO SOCIAL: ADAPTABILIDAD

VALORES & HABILIDADES BLANDAS: CULTURA: CREATIVIDAD

VALORES & HABILIDADES BLANDAS: GRATITUD: RESILIENCIA

VALORES & HABILIDADES BLANDAS: JUSTICIA: RESOLUCIÓN DE PROBLEMAS

VALORES & HABILIDADES BLANDAS: LEALTAD: LIDERAZGO

VALORES & HABILIDADES BLANDAS: OPTIMISMO: PLANIFICACIÓN Y GESTIÓN DEL TIEMPO

VALORES & HABILIDADES BLANDAS: ORGULLO NACIONAL: PENSAMIENTO CRÍTICO

VALORES & HABILIDADES BLANDAS: RESPETO: INTELIGENCIA EMOCIONAL.

VALORES & HABILIDADES BLANDAS: SOLIDARIDAD: TRABAJO EN EQUIPO

VALORES & HABILIDADES BLANDAS: TOLERANCIA: FLEXIBILIDAD

VALORES & HABILIDADES BLANDAS: VERDAD Y HONRADEZ: PROACTIVIDAD

6. NORMAS DE CLASE

Con relación a las normas de clase, es importante destacar que la evaluación de los componentes de gestión académica se compone de tres notas sumativas, cada una con una puntuación máxima de 6.60/6.60, así como un proyecto práctico, como evaluación formativa que se valora con 3.40/3.40, lo que da un total de 10/10 para la calificación del módulo. Los parciales se califican en una escala de hasta 6.60 puntos, representando cada uno el 2.22 de la calificación total de 6.6 puntos. Para presentarse al proyecto final, el estudiante debe haber obtenido al menos 4.50 puntos sumando las tres primeras notas. En caso de no alcanzar este mínimo en el proyecto, se otorga una oportunidad de recuperación dentro de las 48 horas laborables siguientes, según el calendario académico oficial. La nota mínima acumulada requerida para aprobar la asignatura es 7/10, y es esencial mantener al menos un 70% de asistencia a las clases. Los docentes deben informar a los estudiantes sobre sus notas individuales antes de registrarlas en el sistema, y se espera que los alumnos confirmen su aceptación y conformidad con estas calificaciones. Además, los docentes deben entregar un reporte de notas y asistencia a través del SGA y notificado a la coordinación de carrera y registrar las calificaciones en el sistema en un plazo máximo de 5 días posteriores a la recepción del proyecto final.



7. SISTEMA DE EVALUACIÓN

Instrumento	Porcentaje	Puntos
PARCIAL II	22.0	2.2
PARCIAL III	22.0	2.2
PARCIAL I	22.0	2.2
PROYECTO FINAL	34.0	3.4

8. UNIDADES

- 1) Introducción a Diseño de Interfaces
- 2) Experiencia de Usuario
- 3) HTML, CSS y JS (React)

UNIDAD 1 (Introducción a Diseño de Interfaces)

Aplicar conceptos de diseño web mediante la implementación de herramientas HTML, CSS, JavaScript y Bootstrap para la creación de páginas web alojadas en un servidor.

Implementar páginas web bajo cumplimientos de estándares W3C que garanticen un alto grado de usabilidad.

Demostrar participación formando parte de equipos de trabajo, interiorizando el perfil profesional del trabajo colaborativo.

Temas y Subtemas





Tema 1 : Persona-Ordenador

Tema 2 : Estilos de Iteración

Tema 3 : Desarrollo de modelos de interfaces con herramienta de modelado

La relación entre personas y ordenadores, a menudo referida como Interacción Humano-Computadora (HCI, por sus siglas en inglés), comenzó a desarrollarse a medida que los ordenadores se volvieron más accesibles y capaces de realizar tareas diversas que requerían interacción directa con los usuarios.

Orígenes de la Relación Persona-Ordenado

En la era de la Computación la interacción persona-ordenador comenzó en las décadas de 1940 y 1950, cuando los primeros ordenadores electrónicos, como el ENIAC, fueron desarrollados. Estos ordenadores eran enormes podían llegar a ocupar una habitación entera y se operaban principalmente mediante tarjetas perforadas y paneles de control, con una interacción muy limitada y técnica. A medida que las computadoras se volvieron más avanzadas y comenzaron a utilizarse en aplicaciones más diversas, surgió la necesidad de interfaces más intuitivas que permitieran a las personas utilizarlas de manera efectiva. Esto llevó al desarrollo de nuevos paradigmas de interacción, como interfaces gráficas y dispositivos de entrada más intuitivos.

A continuación, remarcamos esta relación de Persona-Ordenador por 10 Hitos que han permitido tanto al hombre como a la maquina evolucionar en su forma de interactuar a lo largo de los años tanto así que ahora podemos tener minicomputadores en la palma de nuestra mano. Realizar transacciones mediante la web y manejar o administrar nuestro contacto con otras personas.





Hitos Importantes en la Historia de la Interacción Persona-Ordenador

1950s - Interfaz de Tarjetas Perforadas: Las tarjetas perforadas fueron uno de los primeros métodos de interacción, permitiendo a los usuarios programar y operar computadoras.

1963 - Sketchpad por Ivan Sutherland: Considerado el primer programa gráfico interactivo, Sketchpad demostró la posibilidad de usar una pantalla y un lápiz óptico para la interacción directa con gráficos.

1968 - La "Madre de Todas las Demos": Douglas Engelbart presentó un sistema que incluía el ratón, hipervínculos, y ventanas, mostrando el potencial de la interacción gráfica y multimedia.

1973 - Xerox Alto: Primera computadora en implementar una interfaz gráfica de usuario (GUI) completa con ventanas, íconos, y el uso de un ratón.

1984 - Apple Macintosh: Popularizó la GUI con ventanas, íconos, menús y el ratón, haciendo las computadoras más accesibles al público en general.

1989 - Introducción de la World Wide Web: Tim Berners-Lee creó la WWW, transformando la manera en que las personas interactúan con la información en las computadoras a través de navegadores web.

1993 - Lanzamiento de Mosaic: El primer navegador web popular que soportaba gráficos, lo que facilitó la navegación por Internet y contribuyó a su expansión masiva.

2007 - iPhone y Pantallas Táctiles Multitáctiles: Introdujo una nueva forma de interacción con dispositivos móviles mediante el uso de gestos táctiles.

2010s - Asistentes Virtuales: La llegada de asistentes como Siri, Alexa, y Google Assistant cambió la interacción persona-ordenador al incorporar interfaces de voz.

Realidad Virtual y Aumentada: Tecnologías como Oculus Rift y Microsoft HoloLens han expandido las posibilidades de interacción, proporcionando experiencias inmersivas que mezclan lo digital y lo físico.



Estos hitos muestran la evolución de la interacción humano-computadora, desde métodos rudimentarios hasta interfaces avanzadas que permiten una interacción más natural e intuitiva con la tecnología.

Iteración en el diseño web

Los estilos de iteración dentro del diseño de interfaces web son las maneras y métodos en que los usuarios pueden interactuar con una aplicación o sitio web. Los mismo que determinan cómo el o los usuarios navegan, ingresan información y utilizan las funcionalidades del sitio web o aplicativo multiplataforma que son tan comunes hoy en día permitiendo así que la creación de diseños más amigables, responsivos y llamativos al uso y función de los procesos destinados nuestra web o aplicativo móvil por ende son considerados herramientas multiplataforma es decir que independientemente del dispositivo que los usuarios manejan deben presentarse los datos e información en forma ordena e intuitivas al uso para ello se presentan tres estilos de iteración más utilizados para poder determinar el desarrollo de estas aplicaciones

Interacción directa: Este estilo implica que los usuarios manipulen directamente los elementos de la interfaz, generalmente a través de clics, arrastres y toques. Es intuitivo y facilita un control preciso. (Krug, 2014)

Ejemplo:

- Mapas interactivos: Servicios como Google Maps, Waze permiten a los usuarios arrastrar el mapa, hacer zoom con gestos de pinza y tocar para obtener más información sobre una ubicación específica.
- Edición de imágenes: Herramientas como Canva permiten a los usuarios arrastrar elementos gráficos y texto directamente en la interfaz para ajustar sus diseños.

Interacción por comandos: En este estilo, los usuarios interactúan con la interfaz mediante comandos de texto. Es común en interfaces más técnicas donde el usuario necesita un control detallado sobre las acciones.

Ejemplo:

- Consolas de comandos: Las herramientas de desarrollo web a menudo incluyen consolas donde los desarrolladores pueden ingresar comandos para ejecutar tareas específicas.
- Chatbots: Plataformas como Slack o Discord permiten interacciones por comandos para realizar acciones como configurar recordatorios o buscar información.



Interacción basada en formularios: Este estilo se centra en la recopilación de información mediante campos de entrada. Es ideal para tareas que requieren que los usuarios proporcionen datos estructurados.

Ejemplo:

- Formularios de registro: Sitios web que requieren crear una cuenta (como redes sociales) utilizan formularios donde los usuarios ingresan su nombre, correo electrónico, y contraseña.
- Cajas de búsqueda: Tiendas en línea como Amazon, eBay y Mercado Libre utilizan formularios de búsqueda para permitir a los usuarios encontrar productos específicos ingresando palabras clave.

Estos estilos de iteración pueden combinarse dentro de una misma aplicación para ofrecer una experiencia de usuario más completa y flexible. ***En el diseño de interfaces, la iteración es un proceso clave que se centra en mejorar continuamente la experiencia del usuario mediante la repetición y refinamiento de los diseños.***

A continuación, un ejemplo de cómo se puede aplicar la iteración en el desarrollo de interfaces web: Se solicita diseñar o rediseñar el panel de usuario para una aplicación de Gestión de Proyectos

Primera Iteración: Prototipo Inicial

Crear un panel intuitivo que permita a los usuarios ver y gestionar sus proyectos de manera eficiente, el mismo debe permitir al usuario navegar entre proyectos permitiéndole así ver, crear, editar y eliminar el proyecto seleccionado.

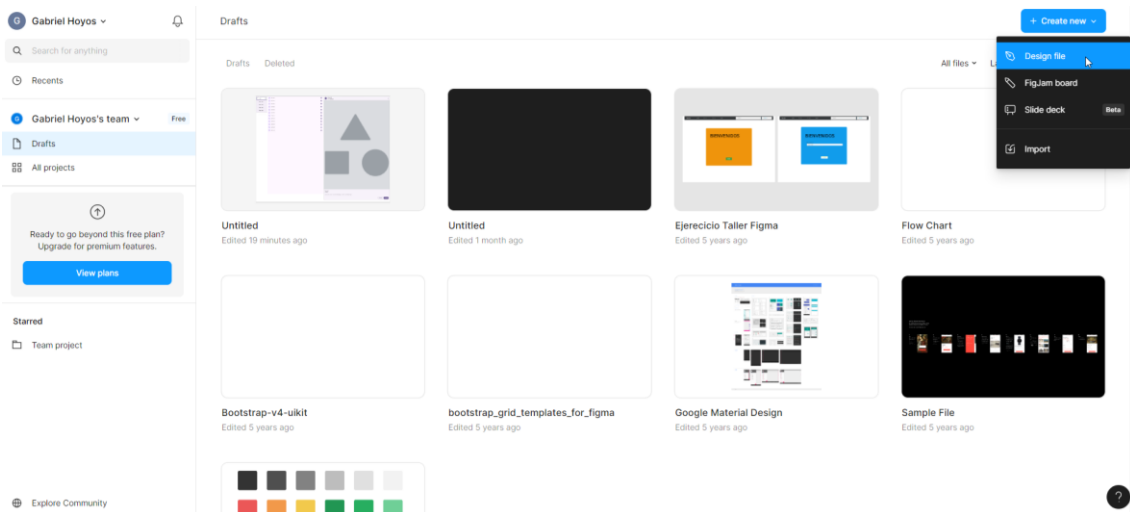
Ejecución:

Se desarrolla un prototipo de baja fidelidad que incluye una barra lateral de navegación, una vista de lista para proyectos, y botones para agregar, editar y eliminar proyectos.

Se utiliza una herramienta como Figma, Sketch o Fluid para esbozar el diseño básico que tendrá el panel de usuario.



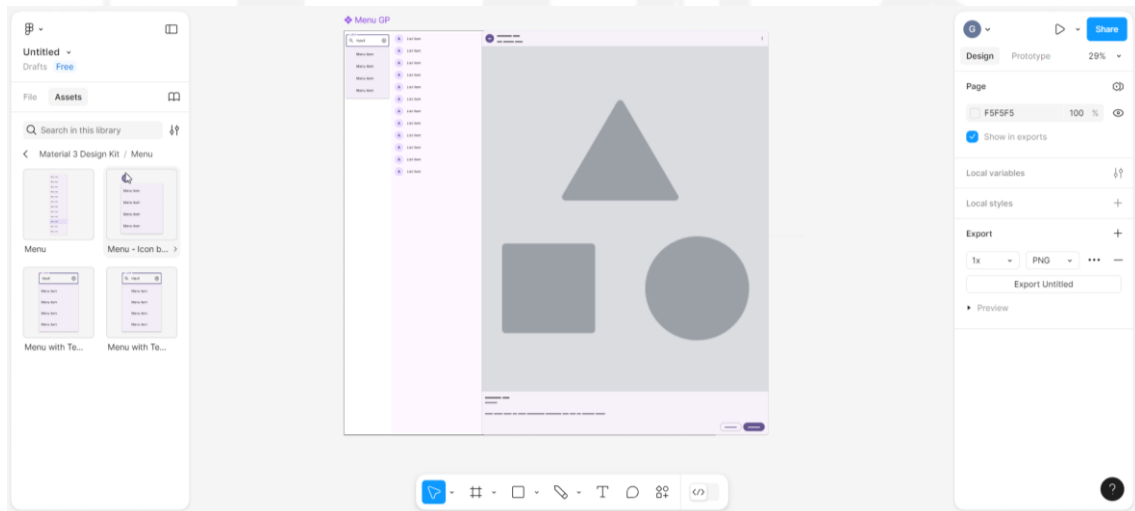
Figura 1:
Herramienta FIGMA Área de Diseño (Drafts)



Elaboración: Autor

En la Figura 1 podemos ver la herramienta de prototipado FIGMA en el área de trabajo Draft encontrara cada diseño realizado también se pueden crear nuevos diseños en el mismo.

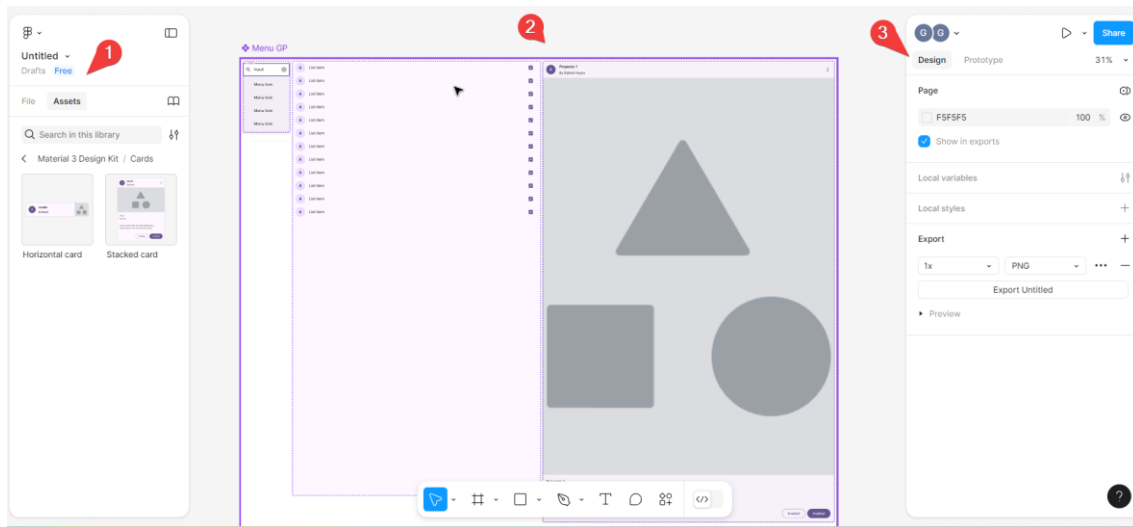
Figura 2:
Herramienta de diseño de prototipado FIGMA



Elaboración: Autor

En la Figura 2 podemos ver la herramienta de prototipado FIGMA con la resolución del ejemplo planteado, cabe recalcar que este es solo un prototipo.

Figura 3:
Herramienta FIGMA y sus componentes



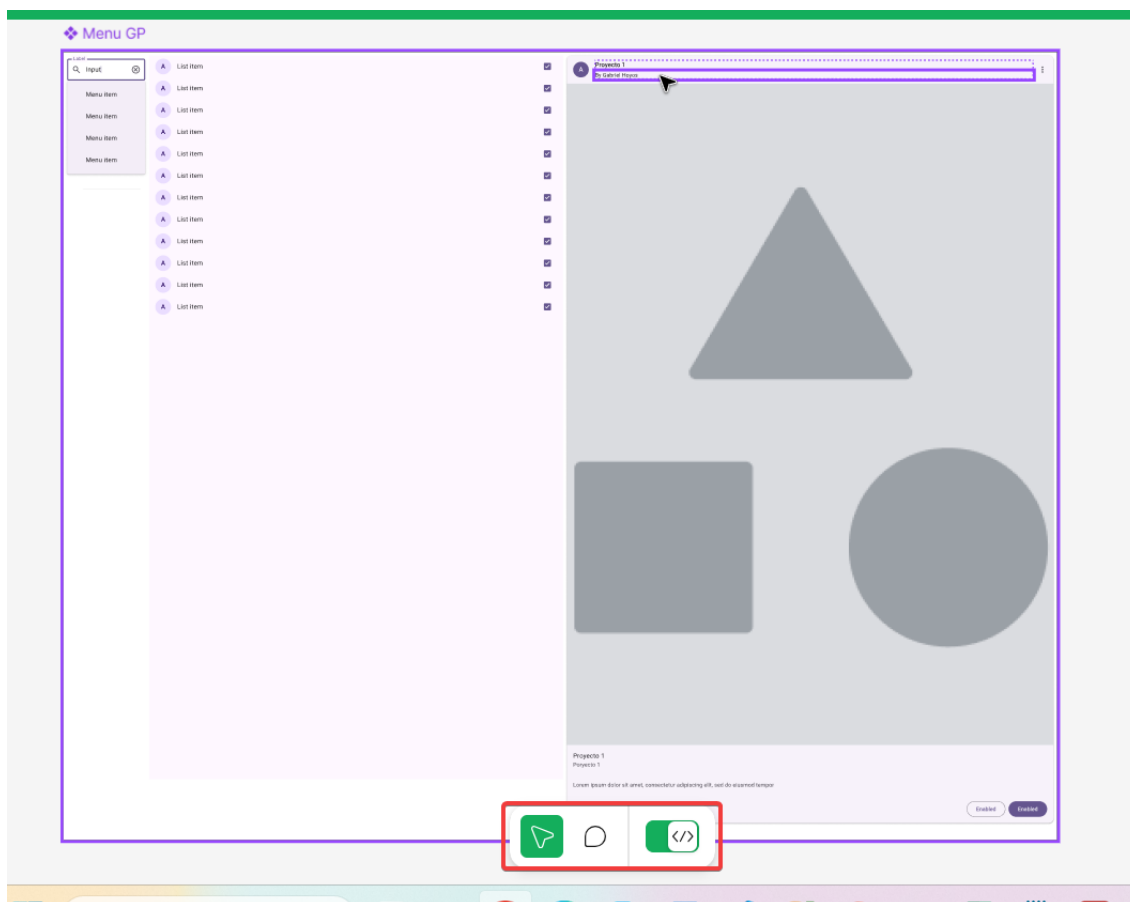
Elaboración: Autor

En la Figura 3 podemos ver las configuraciones de la herramienta de prototipado FIGMA:

- 1) Assests podemos encontrar elementos de diseño que se prefabricados y que se pueden presentar
- 2) Es el área de diseño donde debemos elegir el tipo de dispositivo en el que vamos a diseñar el prototipo
- 3) Configuraciones de cada elemento del área de trabajo.

Todo este entorno representa nuestra área de trabajo definida en un marco el mismo puede ser de un equipo personal (laptop, desktop, Tablet) y también podemos escoger otros tipos de marcos de trabajo para el diseño de prototipos de desarrollo de aplicaciones móviles, así como herramientas que se ajustan a este proceso de diseño.

Figura 4:
Herramienta FIGMA en modo Desarrollador



Elaboración: Autor

En la Figura 4 podremos revisar la herramienta en su modo desarrollo en el mismo se pueden ver cómo funciona el prototipo con los datos facilitados, a su vez este se puede desactivar en la parte inferior del navegador para regresar al modo edición.

Pruebas de Usuario y Recopilación de Comentarios

Ejecución: Realizar pruebas de usabilidad con un grupo pequeño de usuarios objetivo.

Los usuarios no deben encontrar confusa la ubicación de los elementos como botones, barras de tareas y navegación, así como menús para agregar proyectos a La lista de proyectos al tener una previa prueba con el usuario generamos un feedback que retroalimentara el diseño de la aplicación y lo que se espera de la misma.

Segunda Iteración: Ajustes Basados en Comentarios

Se puede adaptar el o los botones a una posición más prominente y accesible, como la parte superior derecha del panel. Se mejora la vista de lista para incluir más detalles sobre cada proyecto, como la fecha de vencimiento y el estado.

Nuevas Pruebas y Validación

Ejecución: Generar una segunda ronda de pruebas con usuarios finales mismos que ahora encuentran el botón de agregar más intuitivo, pero sugieren que la barra lateral podría incluir categorías o filtros para facilitar la búsqueda de proyectos.

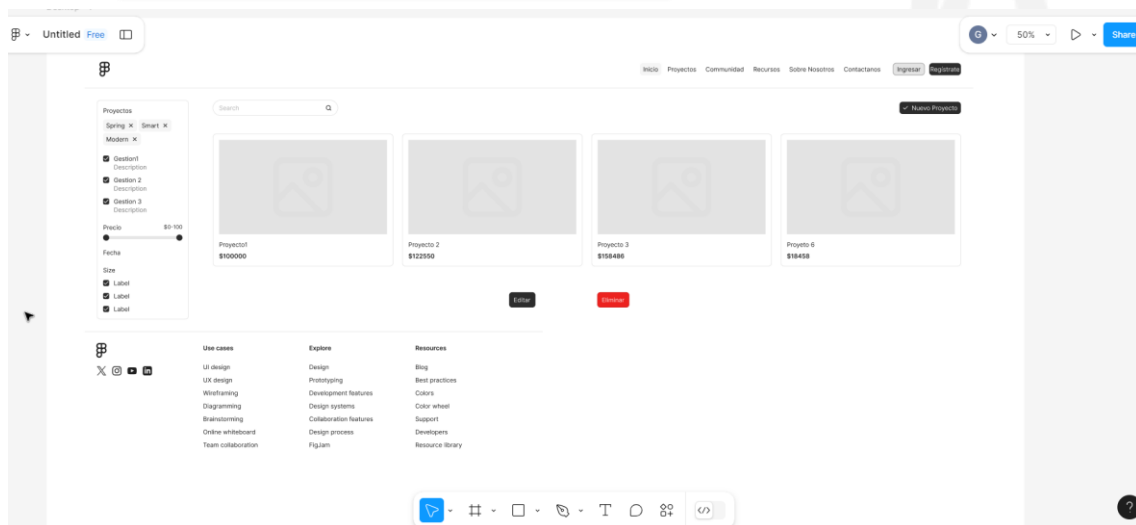
Tercera Iteración: Optimización Final

Se añade una funcionalidad de filtro en la barra lateral que permite a los usuarios clasificar proyectos por categoría, prioridad o estado. Se optimiza el diseño visual para mejorar la estética y la claridad, utilizando un esquema de colores más contrastante y tipografía más legible.

Implementación y Lanzamiento

Después de iterar y ajustar, el diseño se desarrolla e implementa en la aplicación, proporcionando una experiencia de usuario mejorada y más eficiente. Este proceso iterativo no solo garantiza que el producto final cumpla con las necesidades de los usuarios, sino que también permite al equipo de diseño y desarrollo abordar problemas y mejoras de manera proactiva, basándose en datos reales y comentarios directos de los usuarios.

*Figura 5:
Proceso de edición de modelado para la presentación de gestión de proyectos en FIGMA*



Elaboración: Autor

En la Figura 5 podemos evidenciar el resultado del proceso iterativo expuesto en el ejemplo anterior la idea es tener una web que permita gestionar los proyectos de manera eficiente, además la herramienta proporciona la visión necesaria del alcance final mediante un prototipo que puede ser interpretado por el cliente o usuario.

Conclusión: En nuestra unidad hemos revisado el método iterativo y sus cualidades, el mismo debe tener al menos 3 iteraciones antes de implementar y enviar el trabajo colaborativo al área de desarrollo para su creación, a su vez esto permite tener una perspectiva de retroalimentación con los usuarios finales que utilizarán la aplicación permitiendo así evitar la resistencia al cambio, ya que el usuario final fue incluido en el proceso de diseño y va a hacer más fácil e intuitivo el manejo de una aplicación nueva en la que él haya colaborado.

Figura 6:
Proceso Lean UX



Nota: En la Figura 6 se representa a una metodología poderosa para diseñar productos que se alinean más estrechamente con las necesidades del usuario

Fuente: <https://es.slideshare.net/slideshow/introduccion-a-ux-110275116/110275116>

El desarrollo de modelos de interfaces con herramienta de modelado en la actualidad permite tener un panorama claro al usuario de que esperar de su desarrollo final, este es un primer paso para tener una aplicación agradable a la vista que le permite al usuario palpar de mejor manera por el producto que está pagando. Dado que el desarrollo de software y sus respectivas interfaces nacen de la creatividad y de un proceso de creación que puede tomar cierto tiempo relativamente aún no definido, pero mediante estas herramientas podemos pre visualizar el



parte del producto final , si bien un prototipo no es funcional por completo permite tener la perspectiva más cerca del usuario y por ende se puede llegar a acuerdos en base a lo expuesto sobre el modelado de estos diseños realizados ya que podremos tener un consenso entre el usuario final quien va a utilizar la aplicación , el diseñador de la web y el área de desarrollo de software todos son parte de este proceso, a continuación se enumeran una lista de las herramientas más utilizadas para realizar el diseño de interfaces con sus respectivas características , ventajas y desventajas:

Figma

Características:

- Basada en la web, permite la colaboración en tiempo real.
- Ofrece herramientas de diseño vectorial y de interfaz de usuario.
- Facilita la creación de prototipos interactivos.
- Permite el uso de componentes reutilizables y estilos compartidos.
- Soporta extensiones y plugins para aumentar la funcionalidad.

Ventajas:

- Colaboración en tiempo real desde cualquier lugar.
- Interfaz intuitiva y fácil de usar.
- Soporte para componentes reutilizables y estilos compartidos.
- Extensa biblioteca de plugins para extender la funcionalidad.
- Diseños y transiciones basadas en links
- Proceso de creación claro en marcos de trabajo
- Modelado Multiplataforma

Desventajas:

- Requiere conexión a Internet para la mayoría de las funcionalidades.
- Las características avanzadas pueden ser limitadas en comparación con herramientas más especializadas.

Adobe XD

Características:





- Integración con otros productos de Adobe.
- Permite diseñar y prototipar experiencias de usuario de manera interactiva.
- Ofrece herramientas de diseño vectorial y de wireframing.
- Soporte para animaciones y transiciones.
- Proporciona capacidades de colaboración y revisión.

Ventajas:

- Integración fluida con otros productos de Adobe.
- Soporte para diseño y prototipado interactivo.
- Herramientas de diseño vectorial y animación potentes.
- Fácil exportación de recursos para desarrolladores.

Desventajas:

- Requiere suscripción a Adobe Creative Cloud.
- Algunas características avanzadas pueden ser menos intuitivas para nuevos usuarios.

Sketch

Características:

- Popular entre diseñadores de interfaces de usuario y aplicaciones móviles.
- Amplio ecosistema de plugins y complementos.
- Soporte para símbolos reutilizables y bibliotecas compartidas.
- Funcionalidad de diseño vectorial poderosa.
- Exportación de recursos optimizada para desarrolladores.

Ventajas:

- Fuerte enfoque en el diseño de interfaces y experiencia de usuario.
- Amplia comunidad y ecosistema de plugins.
- Ofrece símbolos reutilizables y bibliotecas compartidas.
- Exportación de recursos optimizada para desarrolladores.

Desventajas:





- Solo está disponible para macOS.
- No tiene colaboración en tiempo real integrada sin plugins de terceros.
- Diseños pesados y difícil alcance para otros ambientes de desarrollo

InVision

Características:

- Focalizada en la creación de prototipos y la colaboración con otras herramientas.
- Permite realizar pruebas de usabilidad con prototipos interactivos.
- Ofrece herramientas de diseño básico y de comentarios en tiempo real.
- Facilita la creación de flujos de trabajo y maquetas.
- Incluye InVision Studio para diseño y animación avanzados.

Ventajas:

- Excelente para prototipado y pruebas de usabilidad.
- Herramientas de colaboración en tiempo real.
- Integración con herramientas de diseño populares como Sketch y Photoshop.
- Ofrece InVision Studio para diseño y animación avanzados.

Desventajas:

- La curva de aprendizaje puede ser empinada para usuarios nuevos.
- Algunas funcionalidades están limitadas en el plan gratuito.
- Membresía paga no cuenta con tutoriales.

Autoevaluación 1

1. ¿Qué es el método de Iteración?
 - a. Es la usabilidad con prototipos interactivos.
 - b. La iteración es un proceso clave que se centra en mejorar continuamente la experiencia del usuario mediante la repetición y refinamiento de los diseños.
 - c. Es una metodología que permite a los usuarios ver y gestionar sus proyectos de manera eficiente
 - d. Es el diseño de páginas web.





2. ¿Indique 2 herramientas de modelado para el diseño web?
 - a. Sierralcon/GrandPoot
 - b. DisignEvent/Adobe XD
 - c. Figma/Sketch
 - d. Flutter/Angular
 - e. Android/IOS

3. ¿Cuántas Iteraciones mínimas es recomendable en el proceso de diseño?
 - a. 1 iteración
 - b. 3 iteraciones
 - c. 8 iteraciones
 - d. Ninguna

4. ¿Indique cuál de los enunciados es una ventaja de Figma?
 - a. Bueno para pruebas de usabilidad.
 - b. Diseñar y prototipar experiencias de usuario de manera interactiva.
 - c. Limitada biblioteca de plugins para extender la funcionalidad.
 - d. Facilidad de diseño web
 - e. Modelado Multiplataforma
 - f. Manejo de iteración basado en métricas
 - g. Edición de plantillas base

5. ¿Qué es la Iteración basada en formularios?
 - a. Se centra en la recopilación de información mediante campos de entrada. Es ideal para tareas que requieren que los usuarios proporcionen datos estructurados.
 - b. Se centra en la información mediante datos recopilados. Es ideal para tareas que requieren que los usuarios proporcionen estructuras.
 - c. Interfaces con herramienta de modelado en la actualidad para usuarios.
 - d. Recopilación de información mediante campos.





6. ¿Indique cuáles son los ejemplos de una iteración directa?
 - a. Google Maps / Canva
 - b. Facebook/Instagram
 - c. Waze / X
 - d. IA/ Amazon
 - e. Pinterest/IP

7. ¿Seleccione el año y la empresa que popularizo el uso de la GUI con ventanas, íconos, menús y el ratón, haciendo las computadoras más accesibles?
 - a. 1984 - Apple Macintosh
 - b. 1989 – Macintosh
 - c. 2007 – Facebook
 - d. 1990 – Apple Macintosh

8. ¿Quién creo la World Wide Web y en que año fue creada?
 - a. Tim Berners-Lou - 1980
 - b. Tim Barnes-Ben - 1998
 - c. Tim Berners-Lee - 1987
 - d. Tim Berners-Lee -1989

9. ¿Indique cuál de los siguientes enunciados es una desventaja de Skechth?
 - a. Exportación de recursos optimizada para desarrolladores.
 - b. Amplio ecosistema de plugins y complementos.
 - c. Funcionalidad de diseño vectorial poderosa.
 - d. Solo está disponible para macOS.
 - e. Integración con herramientas de diseño populares como Photoshop.
 - f. Herramientas de colaboración en tiempo real.

10. ¿Indique si los siguientes enunciados son una característica de InVision ?
 - a. Realiza métricas para su usabilidad con prototipos interactivos.
 - b. Limitado prototipado y pruebas de usabilidad.



- c. Herramientas de colaboración no en tiempo real.
- d. La curva de aprendizaje puede ser empinada para usuarios nuevos.
- e. Algunas funcionalidades están limitadas en el plan gratuito.
- f. Focalizada en la creación de prototipos y la colaboración con otras herramientas.

Resumen de la Unidad 1

El trabajo a lo largo de los años entre el ordenador y las personas ha ido mejorando y volviéndose cada vez más habitual hoy en día todo se realiza en nuestros equipos de cómputo ya sea una laptop o una computadora de escritorio y hoy en día hasta el mismo celular proveen un sinfín de herramientas a nuestra disposición esto no fuera posible sin el diseño de interfaces y la mejora continua que la iteración ha proveído a la creación de estas aplicaciones.

En esta unidad abarcamos los conceptos básicos sobre el diseño de interfaces esta breve introducción nos indica la historia del cómo y el por qué la interacción entre el ordenador y el humano a avanzado a pasos agigantados a lo largo de las décadas permitiendo así la mejora continua de las interfaces desde el mismo punto de vista de un producto cambia y se renueva así mismo lo han hecho los Sistemas Operativos que manejamos actualmente en nuestro equipos , esto es debido a la iteración un proceso en el cual se realizan una serie de n pasos para llegar a un avance sobre el diseño de nuestro software , lo usual es que estos pasos sean realizados al menos tres veces ya que en este proceso incluiremos al usuario final sus dudas y nuevos requerimientos serán tomados en cuenta para que vaya de la mano hacia la nueva creación del diseño lo cual permite mitigar la resistencia al cambio de un nuevo software, para ello podemos utilizar cualquiera de nuestros tipos de iteración que hemos indicado en la guía siempre tomando en cuenta que va depender mucho del objetivo final de nuestro software.

Ahora bien, en el transcurso de la unidad se presentó un ejemplo de la creación de un diseño de interfaz de un panel intuitivo que permita a los usuarios ver y gestionar sus proyectos, al principio podemos definir nuestra herramienta que es Figma para la creación de este prototipo y que a su vez nos ha permitido crearlo a la necesidad del usuario este paso de prediseño antes del diseño final permite al usuario final tener una perspectiva del entregable final que vendría a ser el desarrollo de la página que le permita administrar los proyectos, a su vez hemos generado recomendaciones sobre donde se pueden trabajar estos prototipos como por ejemplo lo hemos realizado en Figma pero no es la única herramienta donde se puede trabajar este tipo de primeros entregables en un desarrollo de software si no también lo podremos realizar en Fluid,



Skech, Adobe XD e InVision que son herramienta alternas que también me permiten la creación de prototipos con distintas características y sistemas operativos de trabajo.

Algo importante a recalcar de estas herramientas de diseño es su simplicidad de uso, las características que brindan para el diseño de interfaces desde el punto de multiplataforma hasta el manejo de plugins que permitan una mejor comprensión del proceso de diseño y que a si mismo facilitan el diseño final de nuestro prototipo. Esto a su vez ha generado que las mismas herramientas se enfoquen en el diseño free o gratuito o en su defecto que lo realicen de tal manera que se vuelve privativo ya que son pocas funciones las que encontramos en el modo gratuito. Las herramientas de diseño hoy en día proporcionan el enlace perfecto entre el diseñador/desarrollador y el usuario final que permite realizar presentaciones de prototipos y que sobre estas presentaciones se analicen cambios y mejoras dependiendo de la necesidad permitiendo así evaluar el desempeño y desarrollo de nuestro software sean aplicables a una serie de estándares de diseño, así como a un mejor desenvolvimiento una vez que este ya salga a producción. El diseño de interfaces con herramientas de modelado permite crear representaciones visuales de cómo se verá y funcionará un producto antes de su desarrollo completo.

Estas herramientas ayudan a los diseñadores a planificar la disposición de elementos, la navegación y la interactividad de la interfaz. Proporcionan una forma de experimentar y validar ideas con usuarios y equipos de desarrollo, asegurando que el diseño cumpla con las expectativas y necesidades del usuario final. Las herramientas de modelado también facilitan la colaboración entre diseñadores, desarrolladores y otros interesados. Así como el campo de interacción Persona-Ordenador (HCI, por sus siglas en inglés) se centra en el diseño, evaluación e implementación de sistemas informáticos interactivos que son utilizados por personas.

Este enfoque estudia cómo los humanos interactúan con las computadoras y busca mejorar la usabilidad, accesibilidad y experiencia del usuario. La HCI combina principios de psicología, diseño, informática y ergonomía para crear interfaces que sean intuitivas y efectivas. El objetivo es facilitar interacciones que sean naturales y eficientes, mejorando así la satisfacción y productividad del usuario. Para ello podemos también utilizar los estilos de interacción se refieren a las diferentes formas en que los usuarios pueden interactuar con un sistema informático. Estos incluyen, entre otros, la interacción basada en menús, formularios, comandos, iconos y dispositivos de entrada como el teclado, ratón o pantallas táctiles. Cada estilo tiene sus propias ventajas y desventajas dependiendo del contexto de uso, las características del usuario y las tareas a realizar.



Elegir el estilo de interacción adecuado es crucial para asegurar una experiencia de usuario fluida y eficaz, permitiendo que las acciones del usuario sean claras y directas. Por último, tenemos el despliegue en producción después de iterar y ajustar, el diseño se desarrolla e implementa en la aplicación, donde se provee una experiencia de usuario mejorada y más eficiente. A su vez este proceso iterativo no solo garantiza que el producto final cumpla con las necesidades de los usuarios, sino que también permite al equipo de diseño y desarrollo abordar problemas y mejoras de manera proactiva, basándose en datos reales y comentarios directos de los usuarios finales. Como bien se ha mencionado a lo largo de la unidad esto nos permite mitigar la resistencia al cambio dado que permite al usuario final tener un gran porcentaje de confiabilidad en el sistema.

UNIDAD 2 (Experiencia de Usuario)

Maneja conceptos de IDEs y Servidores para desarrollo web Realiza una investigación sobre los pasos para tener una buena experiencia de usuario.

Temas y Subtemas

Conceptos Generales sobre la Experiencia de Usuario (UX)

Interfaz de Usuario (UI)

Call to action Navegación de las Interfaces Web (CTA)

Historia y Origen de UX

Creación del Término "UX" Donald Norman: El término "Experiencia de Usuario" fue popularizado por Donald Norman, un ingeniero, investigador y científico cognitivo que trabajó



en Apple en los años 90. Aunque no inventó el concepto de enfocar el diseño en el usuario, fue uno de los primeros en usar el término "Experiencia de Usuario" de manera formal.

El término "experiencia de usuario" fue popularizado por Donald Norman, en diseño de productos y arquitectura cognitiva, cuando trabajaba en Apple en la década de 1990. Norman buscaba describir todos los aspectos de la experiencia de un usuario con un sistema, desde el primer contacto hasta el último. Según él, UX se refiere a "todas las interacciones que tiene el usuario con la empresa, sus servicios y productos" (Norman D. , 2013)

Por qué se llama UX:

El término "experiencia de usuario" se utiliza porque abarca todos los aspectos del sistema desde la perspectiva del usuario, no solo el diseño visual o la interfaz. Implica una comprensión holística de la interacción del usuario con el producto. (Garret, 2011)

La razón por la cual Norman eligió el término "Experiencia de Usuario" es porque quería abarcar todos los aspectos de la interacción de una persona con una compañía, sus servicios y productos, no solo la interfaz o la estética. Buscaba un término que incluyera el sentimiento general y la satisfacción del usuario final.

Evolución de UX

La evolución de la experiencia de usuario (UX) ha sido un proceso continuo a lo largo de las décadas, adaptándose a los cambios en tecnología, expectativas de los usuarios y metodologías de diseño

Orígenes del Diseño Centrado en el Usuario (1940-1960)

La experiencia de usuario comenzó a tomar forma con el desarrollo de la ergonomía y la ingeniería humana durante la Segunda Guerra Mundial. Estos campos se centraban en cómo los humanos interactúan con las máquinas y cómo optimizar esa interacción para mejorar la eficiencia y reducir errores. (Krug, 2014)

Interacción Humano-Computadora (1970)

Con la introducción de las computadoras personales, la interacción humano-computadora (HCI) se convirtió en un área de interés académico. Se exploraron temas de interfaz y diseño de sistemas informáticos que facilitaran su uso. (Carroll, 2003)

Aparición de Interfaces Gráficas de Usuario GUI (1980)



Apple y Xerox fueron pioneros en el uso de interfaces gráficas de usuario, que permitieron interacciones más intuitivas con las computadoras. Esto llevó a una mayor consideración de los aspectos visuales y de interacción en el diseño. (Norman D. A., 1988)

Formalización del Término UX (1990)

Donald Norman popularizó el término "experiencia de usuario" en Apple, destacando la necesidad de considerar todos los aspectos de la interacción del usuario con un producto o sistema. Este período también vio la profesionalización del diseño de UX como disciplina. (Norman D. , 2013)

Durante esta época, el diseño centrado en el usuario comenzó a ganar popularidad, especialmente en el desarrollo de software y la tecnología. La necesidad de interfaces más amigables y efectivas se hizo evidente con la expansión de los ordenadores personales.

A medida que la tecnología avanzó, las interfaces digitales se volvieron más complejas, lo que aumentó la necesidad de comprender cómo interactúan los usuarios con estos sistemas. A esto se le llamó el desarrollo tecnológico intangible que llevó a la adopción y formalización de métodos de investigación y diseño centrados en el usuario.

UX en la Web y Dispositivos Móviles (2000)

Con el auge de Internet, la usabilidad web se convirtió en un foco central. El diseño responsivo y las aplicaciones móviles impulsaron la necesidad de experiencias de usuario consistentes en múltiples plataformas. (Garret, 2011)

UX Holístico y Multicanal (2010 – al presente)

Hoy en día, la experiencia de usuario se ve como un componente integral de la estrategia empresarial. El diseño de experiencias abarca todo el recorrido del cliente, utilizando datos y análisis para personalizar y mejorar continuamente las interacciones. (Morville & Rosenfeld, 2006)

UX (Experiencia de Usuario): Se refiere a cómo una persona se siente al interactuar con un sistema o producto. Esto incluye la usabilidad, eficiencia, y satisfacción que el usuario experimenta al utilizar un sitio web, aplicación o cualquier producto digital cuenta con los siguientes objetivos:

- **Satisfacción del Usuario:** Verificar que el usuario se sienta satisfecho sobre sus expectativas después de usar el producto.



- **Facilidad de Uso:** Diseños intuitivos donde los usuarios puedan lograr sus objetivos con el menor esfuerzo posible.
- **Accesibilidad:** Garantizar que el sistema sea usable por personas con diversas capacidades.
- **Interdiscipliniedad:** Permite incluir múltiples disciplinas, como la psicología, el diseño industrial, la interacción humano-computadora (HCI), y el diseño gráfico. Esto lo convierte en un campo multidisciplinario que abarca todos los aspectos de la interacción del usuario

Ejemplos:

- **Navegación Intuitiva:** el diseño debe contar con menús claros y bien organizados para ayudar a los usuarios a encontrar rápidamente la información que necesitan.
- **Feedback Efectivo:** Proporcionar mensajes claros cuando se completa una acción, como una notificación que confirma la recepción de un formulario.
- **Diseño Responsivo:** Asegurar que el sitio funcione bien en dispositivos móviles y de escritorio, ajustándose automáticamente al tamaño de la pantalla.

Dentro de la UX podemos tener métodos de investigación para entender las necesidades y comportamientos de los usuarios a través de entrevistas y encuestas. A su vez podemos también presentar prototipos y pruebas de baja fidelidad y realizar pruebas con usuarios reales para identificar problemas antes de la implementación final todo esto a través del proceso de Iteración o mejora continua basado en la retroalimentación obtenida de los mismos usuarios y sus pruebas

UI (Interfaz de Usuario): Es el espacio donde ocurren las interacciones entre humanos y máquinas. Se centra en el diseño visual de la aplicación y la manera en que se presentan los elementos gráficos y textuales al usuario tiene como principales objetivos:

- **Atractivo Visual:** Crear interfaces visualmente atractivas y coherentes que representen la identidad de la marca.
- **Consistencia:** Usar estilos y patrones uniformes a lo largo de toda la interfaz.
- **Claridad:** Asegurar que los elementos visuales comuniquen de manera efectiva sus funciones.

Ejemplos:





- **Botones Atractivos:** Diseñar botones con efectos de sombra y animación para hacerlos más visibles y atractivos.
- **Tipografía Clara:** Usar una jerarquía tipográfica que guíe al usuario a través del contenido, con tamaños y pesos adecuados.
- **Paleta de Colores Coherente:** Aplicar colores que se alineen con la marca y sean agradables a la vista, facilitando la navegación.

De la mano de la experiencia de usuario viene el manejo de las interfaces para el usuario por ello la relación entre UX (User Experience) y UI (User Interface) es cercana y complementaria, aunque abarcan diferentes aspectos del diseño y desarrollo de productos digitales. Mientras UX (User Experience) es la experiencia global del usuario al interactuar con un producto o servicio que incluye todos los aspectos de la interacción del usuario, como la facilidad de uso, la eficiencia, y la satisfacción general el UI (User Interface) se enfoca en el diseño visual y los elementos con los que el usuario interactúa directamente lo cual incluye botones, menús, iconos, tipografía, colores, y disposición gráfica.

Es decir, la UX implica investigación sobre el usuario, creación de flujos de usuario, prototipos y pruebas de usabilidad dado que los diseñadores de UX buscan comprender las necesidades y expectativas del usuario para mejorar la funcionalidad y accesibilidad del producto digital y la UI se centra en el aspecto estético y el diseño detallado de cada elemento de la interfaz. Los diseñadores de UI trabajan para hacer que la interacción sea visualmente atractiva y coherente por lo tanto UX busca optimizar la experiencia completa, asegurando que el producto sea útil, fácil de usar, agradable y funcional y UI trata de crear una interfaz atractiva que facilite la interacción y que sea visualmente intuitiva. Aunque UX y UI tienen enfoques diferentes, son interdependientes una excelente UI puede mejorar la experiencia del usuario (UX), y un buen diseño UX puede guiar la estructura y diseño de la interfaz (UI). Un buen producto digital necesita tanto de un diseño UX sólido como de una UI efectiva para lograr el éxito en el mercado y satisfacer a los usuarios.

En conclusión, UX y UI son partes esenciales del diseño de productos digitales que, al trabajar juntas, garantizan que los usuarios tengan una interacción efectiva y agradable dentro del desarrollo de un producto digital.





- **Complementariedad:** UX y UI son dos componentes esenciales del diseño de productos digitales que trabajan juntos para elaborar una experiencia de usuario efectiva y atractiva. Mientras que UX se enfoca en el proceso y la satisfacción del usuario frente al producto digital, UI se concentra en los elementos visuales y la presentación.
- **Colaboración:** Los diseñadores de UX y UI suelen colaborar estrechamente para asegurarse de que la investigación de usuarios y la arquitectura de la información se traduzcan en interfaces visualmente atractivas y funcionales.

UX: Se centra en el proceso de interacción del usuario con el producto lo cual incluye investigación de usuarios, diseño de la estructura del producto, y pruebas de usabilidad por ende procura mejorar la experiencia completa del usuario, asegurando que sea intuitiva, eficiente y satisfactoria. Al incluir estos aspectos como la arquitectura de la información, la accesibilidad, la investigación del usuario, y la creación de prototipos nos permite obtener una comprensión profunda del usuario mediante la investigación y el análisis, se entienden las necesidades y comportamientos del usuario. Por ende, es primordial la optimización de la funcionalidad para que se pueda evidenciar una mejora en la eficiencia y efectividad del producto al hacer que el flujo de trabajo sea lógico y sin fricciones.

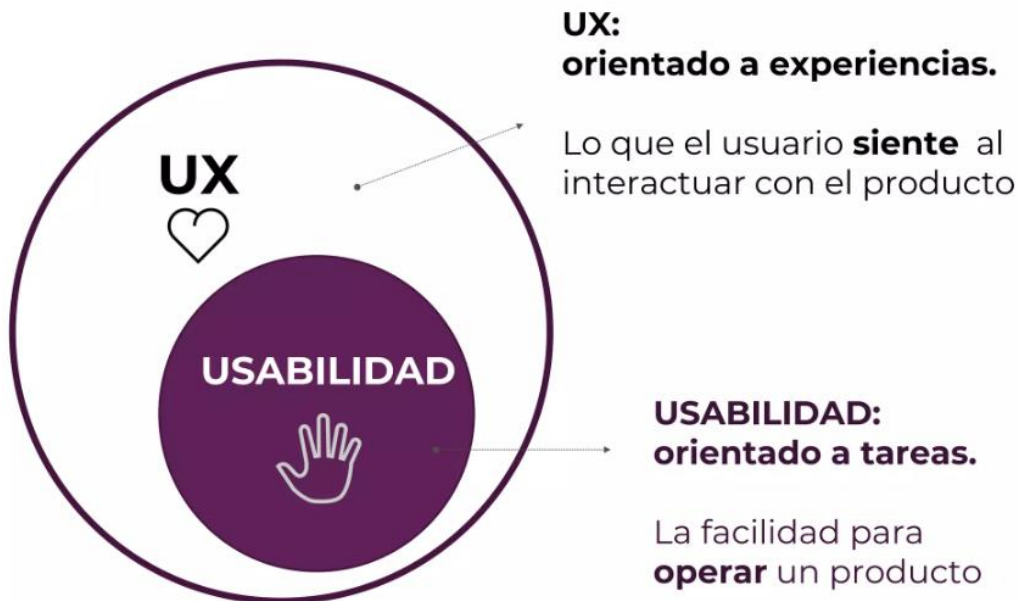
Como se indicó en la unidad anterior la capacidad de iterar y mejorar continuamente a través de pruebas con usuarios reales garantizara la correcta retroalimentación para el desarrollo del diseño debido a que el UX se enfoca en la apariencia y la disposición de los elementos gráficos con los que interactúa el usuario por tanto promueve que el diseño sea estéticamente agradable y que los elementos visuales sean claros y coherentes para ello el usuario presenta sus dudas y recomendaciones sobre el diseño realizada las misma que serán tomadas en cuenta para la próxima iteración y así mismo la UI se ocupa del diseño gráfico, la tipografía, la paleta de colores, y el diseño de elementos interactivos que cuentan con un atractivo visual por ejemplo carruseles o transiciones que se encuentran dentro de las interfaces que son visualmente atractivas y que captan la atención del usuario. Además, se asegura de que todos los elementos visuales sean coherentes y refuercen la identidad de la marca. En resumen, UX y UI son disciplinas interrelacionadas que, al unirse, crean productos que no solo son útiles y funcionales, sino también visualmente atractivos y fáciles de usar.





Figura 7:
Representación de la Experiencia de Usuario (UX)

UX ≠ USABILIDAD



Nota: la Figura 7 representa la Usabilidad dentro del entorno de UX.

Fuente: <https://es.slideshare.net/slideshow/introduccion-a-ux-110275116/110275116>

La Experiencia de Usuario (UX) es el conjunto de percepciones y reacciones que experimentan los usuarios al interactuar con un producto, sistema o servicio. Se centra en entender y mejorar cómo los usuarios perciben la utilidad, facilidad de uso y eficiencia del producto mediante sus siguientes características:

- Enfoque en el Usuario
- Usabilidad
- Accesibilidad
- Arquitectura de la Información
- Diseño Iterativo
- Satisfacción
- Feedback del Usuario
- Diseño Visual

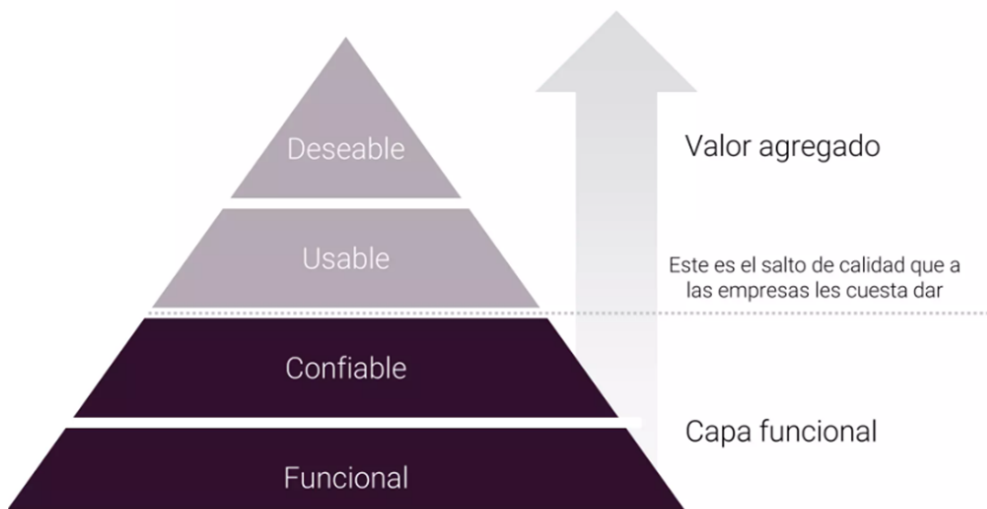




Proceso de un producto digital deseable:

Figura 8:
Anatomía de un producto Digital Deseable

ANATOMIA DE UN PRODUCTO DESEABLE



Nota: la Figura 8 representa la anatomía en que se considera un producto deseable y debe converger en cada una de sus capas.

Fuente: <https://es.slideshare.net/slideshow/introduccion-a-ux-110275116/110275116>

Estándares que se manejan en UX / UI según las características que hemos revisado durante el desarrollo de esta unidad:

- Crear bosquejos básicos de la estructura de la página para definir la ubicación de los elementos clave.
- Desarrollar prototipos de alta fidelidad con todos los detalles visuales, incluidos los colores y tipografía.
- Colaborar con desarrolladores para implementar el diseño y realizar evaluaciones de usabilidad para garantizar que la interfaz sea fácil de usar.
- Enfoque de un diseño web simplificado en el uso de usuario.
- Diseño centrado en las necesidades, deseos y limitaciones de los usuarios.
- Involucrar a los usuarios en el proceso de diseño mediante pruebas y retroalimentación
- Verificar que el producto sea fácil de usar y entender.
- Minimizar la curva de aprendizaje para los usuarios nuevos.



- Mejorar continuamente a través de ciclos de pruebas, retroalimentación y ajustes.
- Proporcionar retroalimentación inmediata y clara al usuario sobre sus acciones.
- Organizar la información de manera que guíe al usuario a través del contenido de forma lógica.

Usabilidad: Se refiere a la facilidad con la que los usuarios pueden aprender a usar un sistema, la eficiencia con la que pueden realizar tareas, y su satisfacción general al hacerlo. Un producto con alta usabilidad es intuitivo y permite a los usuarios lograr sus objetivos de manera efectiva. (Nielsen, 2012)

Accesibilidad: Es la capacidad del usuario para encontrar información o características dentro del producto. Un diseño de UX eficaz organiza la información de manera lógica y predecible, facilitando la navegación. (Morville & Rosenfeld, 2006)

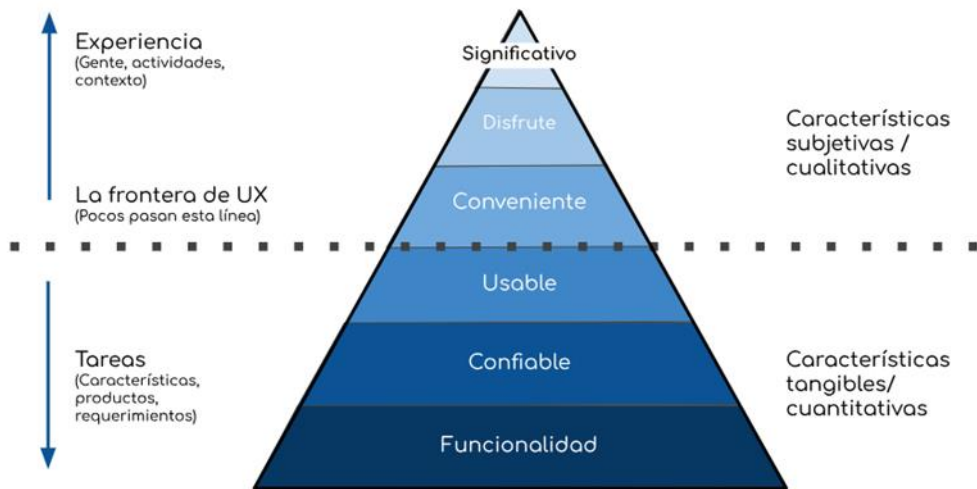
Implica diseñar productos que puedan ser utilizados por personas con diversas capacidades, incluyendo aquellas con discapacidades. Esto abarca el uso de tecnologías asistidas y el cumplimiento de pautas de accesibilidad. (Thatcher, y otros, 2006)

Diseño Visual e Iterativo: La experiencia de usuario también debe ser atractiva desde el punto de vista emocional. Esto incluye el diseño visual y el impacto estético, lo cual puede influir en cómo los usuarios perciben el valor y el atractivo de un producto. (Norman D. A., 2004)

Credibilidad: La percepción de la credibilidad afecta la confianza que los usuarios tienen en el producto. Esto incluye aspectos de seguridad, fiabilidad de la información y la percepción de la marca. (Fogg, 2003)

Utilidad: Un producto debe cumplir una necesidad o resolver un problema específico para el usuario. La utilidad se centra en la funcionalidad del producto y cómo esta satisface las necesidades del usuario. (Krug, 2014)

Figura 9:
Pirámide de Diseño UX



Nota: la Figura 9 representa la pirámide de diseño de UX.
Fuente: <https://www.tesseractspace.com/blog/la-piramide-de-diseno-de-ux/>

En la Figura 9 podemos divisar las características generales de UX aplicadas al modelo de trabajo para obtener un producto digital deseable valorado de manera cuantitativa y cualitativa de tal forma que sea un diferenciador al generar nuestra aplicación utilizando este modelo, la base será siempre imprescindible para que nuestras aplicaciones sean consideradas un éxito. Como diseñadores/desarrolladores web debemos encontrar crucial el proceso para equilibrar ambos aspectos, UX y UI, para crear productos digitales exitosos.

Durante el proceso de diseño, es esencial mantener una comunicación constante con los clientes y usuarios finales, asegurando que sus expectativas y necesidades sean siempre priorizadas. Además, la iteración y la mejora continua, basadas en la retroalimentación del usuario, son prácticas clave para lograr un diseño que no solo sea funcional, sino también hermoso y agradable de usar.

Figura 10:
Modelo de Trabajo sobre la Pirámide de UX

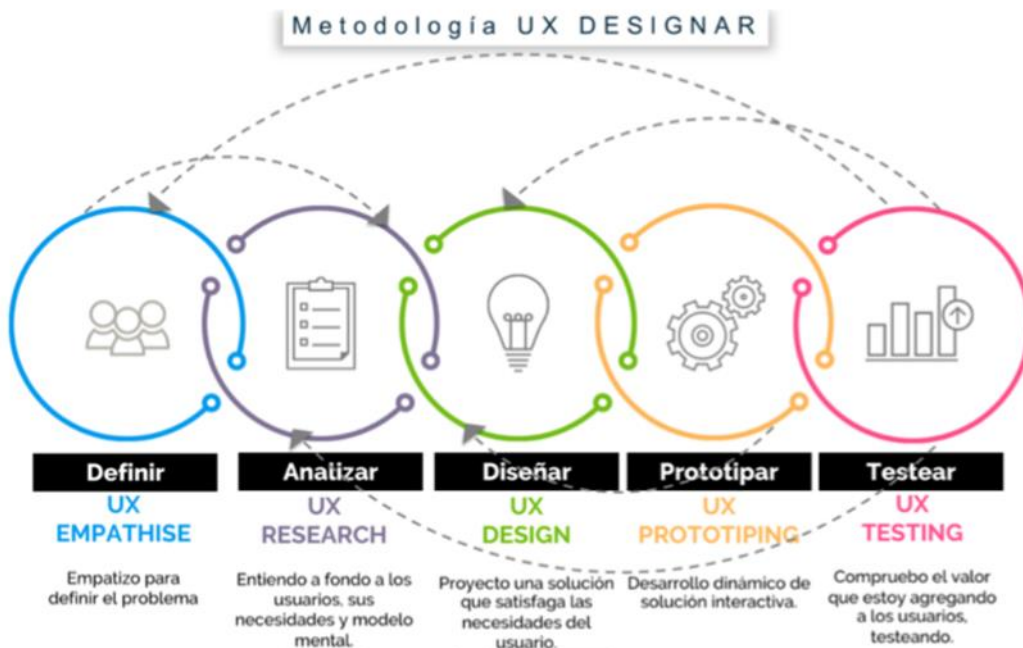


Es mejor trabajar la pirámide de diseño de UX de manera lateral en lugar de forma vertical. **Crédito:**
Aarron Walter y Jussi Pasanen

Nota: la Figura 10 representa el modelo de trabajo de la pirámide de diseño de UX.
Fuente: <https://www.tesseractspace.com/blog/la-piramide-de-diseno-de-ux/>

En la Figura 10 se puede evidenciar como debe aplicarse el modelo de trabajo de forma vertical es decir que debe ir aumentando las mejoras, los entregables a medida que se van cumpliendo con cada una de las características del modelo de trabajo.

Figura 11:
Metodología de Diseño UX (iteración)



Nota: la Figura 11 representa el modelo de diseño UX aplicado para la resolución de un proyecto.
Fuente: <https://www.revistaespacios.com/a20v41n01/20410127.html>

En la Figura 11 podemos ver aplicada la metodología UX con el método de iteración y respaldado en la pirámide del diseño UX donde empezaremos definiendo el problema y levantando los requerimientos para poder diseñar una solución que satisfaga las necesidades presentadas en los requerimientos una vez teniendo claro el panorama podemos realizar un prototipo interactivo de la futura solución a entregar, facilitando así la comprensión de nuestro diseño y su usabilidad.

Importancia de UX

Un buen diseño de UX mejora la satisfacción del cliente, lo que puede aumentar la lealtad y la retención como cliente potencial para futuros desarrollos, los productos digitales bien diseñados permiten a los usuarios completar tareas de manera más rápida y eficiente y esto a su vez se traduce en una mejor manera de identificar problemas de usabilidad en etapas tempranas del desarrollo puede reducir significativamente los costos de desarrollo y mantenimiento del software la UX sigue siendo un componente esencial en el diseño de productos digitales modernos, asegurando que la tecnología no solo sea funcional, sino también accesible y placentera de usar por tal motivo cumple con los siguientes criterios : Satisfacción del Cliente, Eficiencia, Reducción de Costos.

Figura 12:
Importancia de UX para las Empresas



Nota: En la Figura 12 podemos divisar la importancia que toma el diseño UX para las empresas en general.
Fuente: <https://fastercapital.com/es/tema/comprender-la-importancia-del-dise%C3%B1o-ux-en-la-optimizaci%C3%B3n-de-conversiones.html>

En la Figura 12 nos presenta la importancia que genera el diseño de UX en distintas ramas que giran en torno al usuario ya que conlleva una especie de fidelización digital entre usuario y empresa permitiéndole a la empresa tener un mejor margen al obtener nuevos y potenciales clientes y de esa forma fidelizarlos en el proceso.

CTA (Call To Action) Llamada de acción en el diseño web

Es un elemento valioso dentro del diseño de páginas web, cuyo propósito es guiar a los usuarios hacia una acción específica que los diseñadores o las empresas desean que realicen. Estas acciones pueden variar desde hacer clic en un enlace, suscribirse a un boletín, comprar un producto, descargar un archivo, o cualquier otra interacción que convierta a un visitante pasivo en un participante activo de la web. (Chaffey & Ellis-Chadwick, 2019)

Los CTA están diseñados para captar la atención del usuario y dirigirlos hacia una acción deseada. Son fundamentales para la conversión y el éxito de una página web, ya que sin una guía clara los usuarios pueden no realizar las acciones que se espera de ellos. (Caddick & Cable, 2011)

Diseño Visual:

- **Colores y Contrastes:** Los CTA suelen utilizar colores que contrastan con el resto del diseño de la página para destacar. Esto hace que sean fácilmente visibles y atraigan la atención del usuario.
- **Tamaño y Forma:** Deben ser lo suficientemente grandes para ser notados, pero no tan grandes que se conviertan en una distracción. Las formas redondeadas o con bordes suaves son populares porque resultan más atractivas y amigables.
- **Texto Claro y Directo:** El texto del CTA debe ser breve, claro, y orientado a la acción. Por ejemplo, "Compra ahora", "Descargar gratis", "Suscríbete", etc. Este texto debe comunicar exactamente lo que sucederá cuando el usuario haga clic.
- **Posicionamiento Estratégico:** Los CTA deben ubicarse en lugares estratégicos dentro de la página web, como la parte superior de la página, después de secciones clave de contenido, o al final de un artículo o post. El objetivo es colocarlos donde los usuarios ya estén predispuestos a actuar.
- **Pruebas y Optimización:** Es importante realizar pruebas A/B para determinar qué diseño, texto o ubicación de CTA funcionan mejor. Esto implica crear dos versiones diferentes de un CTA y medir cuál tiene una mejor tasa de conversión.

La optimización continua del CTA puede aumentar significativamente la efectividad de una página web al mejorar las tasas de conversión.

Ejemplos:

Figura 13:
Call to Action Simple (CTA) en página de Salesforce



Nota: En la Figura 13 vemos como funciona los CTA simples en la página de Salesforce.
Fuente: <https://agenciaseonetbulb.com/noticias/call-to-action/>

En la Figura 13 podemos revisar un CTA o llamado de acción simple que invita al usuario de la página seguir revisando el contenido de esta o a su vez sugerirle que se suscriba o enrole a la membresía gratuita del servicio ofertado. Esto permite tener una fluidez y simplicidad para que los usuarios accedan a la información más relevante del sitio web esto permitirá obtener un mejor posicionamiento en buscadores de la misma forma que obtendremos unas estadísticas que nos permitan conocer el alcance de nuestro sitio es decir cuántos usuarios han ingresado y revisado las partes más relevantes del sitio web, en la misma se encuentra por lo general los servicios pagos que se ofertan o a su vez una prueba durante cierto tiempo que será renovada en si todo esto tiene una gran parte que ver con el marketing que se realiza dentro de la empresa ya que incluyen una serie de estrategias que deben llamar la atención del usuario.

Características de CTA

Visibilidad y Detectabilidad:

Los CTA deben ser fácilmente visibles y destacarse del resto del contenido de la página. Esto generalmente se logra mediante el uso de colores contrastantes, tamaños de fuente más grandes, y posiciones prominentes en la página. La idea es atraer la atención del usuario rápidamente y dirigir su foco hacia el CTA. (Chaffey & Ellis-Chadwick, 2019)

- Ejemplo: Un botón de "Suscríbete" en un blog puede ser de un color brillante como el rojo o el verde o el color corporativo de la empresa, que contraste con el fondo de la página para asegurarse de que los visitantes lo vean fácilmente.

Claridad y Brevedad:

El mensaje del CTA debe ser claro y conciso, indicando en forma efectiva lo que se espera que el usuario realice. Utilizar verbos de acción en como "Comprar ahora", "Registrarse", o "Descargar" ayuda a establecer expectativas claras. (Krug, 2014)

- Ejemplo: En lugar de un texto largo, un botón que diga simplemente "Descargar ahora" indica claramente lo que ocurrirá al hacer clic.

Relevancia y Contexto:

El CTA debe ser relevante para el contenido mismo que debe ser envolvente y aparecer en un contexto donde el usuario ya esté predispuesto a realizar la acción. Esto significa colocar CTA en lugares estratégicos después de haber proporcionado suficiente información o incentivos. (Caddick & Cable, 2011)

- Ejemplo: Un botón de "Agregar al carrito" colocado inmediatamente después de la descripción de un producto y sus ventajas asegura que los usuarios tomen decisiones informadas.

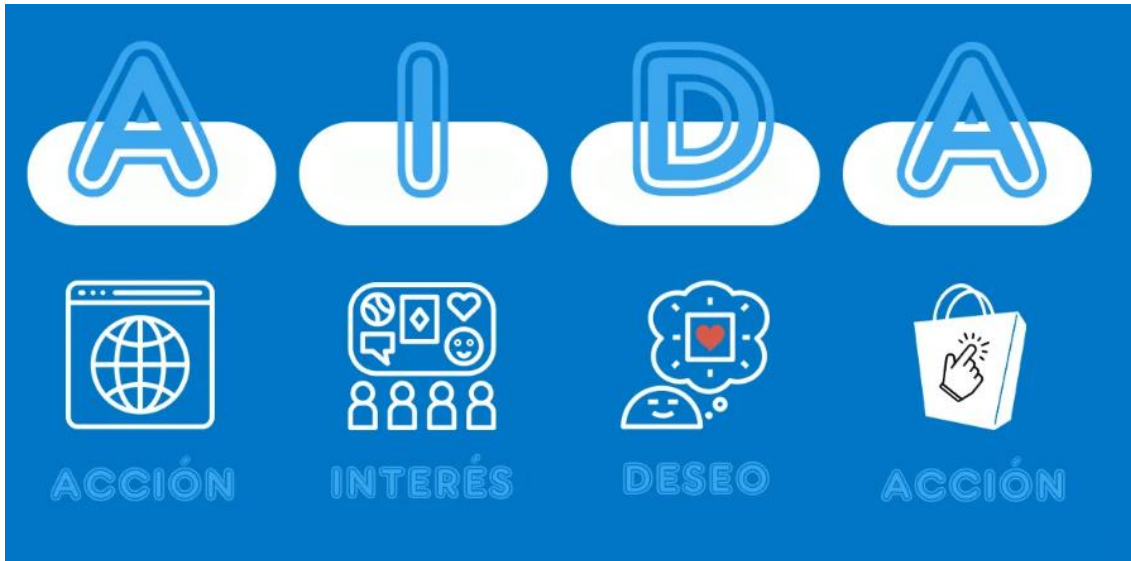
Urgencia:

Crear un sentido de urgencia puede motivar a los usuarios a actuar de inmediato. Esto se puede lograr mediante el uso de palabras que indiquen limitación de tiempo o disponibilidad, como "Oferta por tiempo limitado" o "Últimas unidades".

- Ejemplo: Un CTA que diga "Compra hoy y ahorra 20%" crea una sensación de urgencia que puede aumentar la probabilidad de conversión.

Modelo AIDA

Figura 15:
Modelo AIDA para Diseño de CTA



Nota: En la Figura 15 encontramos la definición del modelo AIDA e marketing digital.
Fuente: <https://blog.hubspot.es/sales/modelo-aida>

Es un marco clásico de marketing que describe las etapas que un consumidor atraviesa antes de realizar una compra. AIDA significa Atención, Interés, Deseo y Acción, y es especialmente relevante al diseñar Call to Action (CTA) efectivos en páginas web. (Kotler & Armstrong, 2018)

Modelo AIDA aplicado a CTA

- **Atención (Attention):** Captar la atención del usuario inmediatamente al llegar a la página mediante colores brillantes y contrastantes que hagan que el CTA destaque en la página. El uso de un diseño visual atractivo y una ubicación estratégica puede ayudar a atraer la atención. (Chaffey & Ellis-Chadwick, 2019)
- **Interés (Interest):** Mantener el interés del usuario proporcionando información relevante y valiosa. Ofrecer un texto de CTA que resuene con las necesidades o problemas del usuario. Puedes incluir beneficios o puntos de interés justo antes del CTA. (Chaffey & Ellis-Chadwick, 2019)
- **Deseo (Desire):** Crear un deseo en el usuario de obtener el producto o servicio. Usa palabras que apelen a las emociones y deseos del usuario. Resalta características únicas o beneficios que hagan que el producto o servicio sea deseable. (Cialdini, 2006)
- **Acción (Action):** Motivar al usuario a tomar acción de inmediato. El CTA debe ser directo y llevar al usuario a realizar la acción deseada sin obstáculos. Usa un lenguaje claro y llamativo. (Cialdini, 2006)

Pruebas y Ajustes: Realiza pruebas para ajustes de los elementos de AIDA, desde los colores y el texto hasta la posición del CTA, para optimizar las tasas de conversión. Una vez definido el modelo AIDA debemos asegurarnos de que cada paso del modelo AIDA fluya naturalmente dentro del diseño web. Es decir, cada elemento desde el diseño hasta el texto debe trabajar en conjunto para mover al usuario a través del embudo de conversión.

Empecemos a entender que un call to action (CTA) es un aviso, estímulo, invitación o llamada que se le hace al usuario para que realice una acción determinada, por ejemplo, en tus landing pages (páginas de aterrizaje, en general, tienen muchos menos elementos y enlaces de lo que tiene una home page de un sitio web normal.), pueden ser de tipo visual (colores, símbolos, etc.), en forma de texto e incluso mixtos, es decir, una combinación de ambos. Aunque esta no es una mala práctica necesariamente, es un texto muy genérico que no brinda contexto alguno a los usuarios, por tanto, es poco probable que motive a las personas a tomar dicha acción. Un buen CTA es aquel que invita a la acción conectando las necesidades de los usuarios con los objetivos de negocio.

Ejemplos del CTA en el enlace o Botón debe decir:

- “Domina Facebook Ads” para tutoriales, guías o cursos.
- “Obtén una asesoría gratuita” para servicios de consultoría
- “Descubre todas las funcionalidades sobre...” para plataforma todo en uno de marketing y ventas.

Al hacerlo de esta manera, las probabilidades de que los visitantes de tu landing tomen la acción objetivo, aumentan considerablemente al demostrarles que tendrán la solución a su problema o necesidad al hacer **una conversión es transformar un visitante en una oportunidad de negocio**, así de simple.

Tipos de CTA

CTA de Conversión Directa: Son aquellos CTA que están diseñados para convertir visitantes en clientes o suscriptores de inmediato. Son acciones que implican un compromiso significativo, como una compra o una suscripción.

Ejemplo:

- "Comprar ahora",
- "Regístrate"
- "Suscríbete".

Aplicación: Se colocan en páginas de productos, páginas de destino, o en secciones clave de una página web donde los usuarios ya han recibido suficiente información para tomar una decisión.

CTA de Generación de Leads: Son aquellos CTA que buscan obtener información de contacto del usuario, como su correo electrónico o número de teléfono, a cambio de algún incentivo.

Ejemplo:

- "Descargar eBook gratis"
- "Accede al webinar"
- "Obtén una cotización gratuita".

Aplicación: Se utilizan en blogs, páginas de destino y en contenido educativo o informativo donde el usuario está buscando más información.

CTA de Información o Aprendizaje: Estos CTA se encuentran diseñados para educar o informar al usuario, estos CTA dirigen a los usuarios a contenido adicional, como artículos, guías o videos.

Ejemplo:

- "Leer más"
- "Ver video"
- "Explora la guía completa".

Aplicación: Se encuentran comúnmente en blogs, artículos y recursos educativos dentro del sitio web.

CTA de Redes Sociales: Estos CTA fomentan la interacción en redes sociales, invitando a los usuarios a seguir, compartir o interactuar con contenido en plataformas sociales.

Ejemplo:

- "Síguenos en Facebook"
- "Comparte en Twitter"
- "Dale me gusta".

Aplicación: Se colocan en artículos de blogs, páginas de agradecimiento, y en el pie de página del sitio web.

CTA de Retención o Fidelización: Estos CTA son diseñados para mantener a los usuarios comprometidos y fomentar la lealtad y suelen invitar a los usuarios a regresar al sitio web o a participar en programas de fidelización.

Ejemplo:

- "Únete a nuestro programa de recompensas"
- "Visítanos nuevamente"
- "Explora más productos".

Aplicación: Se utilizan en campañas de email marketing, páginas de cuenta de usuario, y después de una compra.

CTA de Retroalimentación: Estos CTA buscan obtener la opinión del usuario sobre su experiencia, productos o servicios.

Ejemplo:

- "Déjanos tu opinión"
- "Completa la encuesta"
- "Califica nuestro servicio".

Aplicación: Se colocan en páginas de agradecimiento, correos post-compra, y en secciones de soporte al cliente.

En la actualidad el manejo de CTA dentro de una web que ofrece servicios, productos o información es fundamental para que tenga mayor visibilidad a nivel de buscadores que permitan a su vez la búsqueda generalizada de potenciales clientes en un entorno virtual hoy por hoy también parte de los CTA se integran con los chatbots que permiten luego un contacto directo con la empresa ya sea mediante un vendedor o agente de ventas.

Los tipos de CTA dependerán mucho del entorno en que fue desarrollado el sitio web y el diseño en las interfaces como se ha mencionado en las anteriores unidades con el método iterativo determinaremos cual utilizar para ser parte del proceso, el CTA potenciara la web con tus visitantes, prospectos y clientes que esperan poder conectarse contigo en tiempo real a través de chats en directo. La herramienta de chatbot ayudara a gestionar a tu equipo a calificar leads, programar reuniones, ofrecer respuestas a las preguntas técnicas más frecuentes.



Autoevaluación 2

1. ¿Qué significan las siglas UX/UI?
 - a. Usabilidad de usuarios externos / Interfaces Útiles
 - b. Diseño de interfaces de usuario interno / Diseño de experiencias
 - c. Experiencia de Usuario / Interfaz de Usuario
 - d. Es el diseño de páginas web.
2. ¿Indique la definición correcta de UX?
 - a. Se refiere a cómo una persona se siente al interactuar con un sistema o producto. Esto incluye la usabilidad, eficiencia, y satisfacción que el usuario experimenta al utilizar un sitio web, aplicación o cualquier producto digital.
 - b. Proceso en el cual una persona se siente como un sistema o producto utiliza un sitio web, aplicación o cualquier producto digital.
 - c. Formas de interactuar con un sistema o producto.
 - d. Es un equipo de cómputo que determina la usabilidad, eficiencia, y satisfacción que el usuario experimenta un sitio web, aplicación o cualquier producto digital.
3. ¿Que garantizan al trabajar juntas UX y UI?
 - a. Garantizaran que los usuarios tengan una interacción efectiva y agradable dentro del desarrollo de un producto digital
 - b. Conjunto de percepciones y reacciones que experimentan los usuarios al interactuar con un producto
 - c. Garantizar que los usuarios tengan una interacción efectiva y agradable dentro del desarrollo de un producto digital
 - d. Ninguna
4. ¿Indique cuál de los siguientes enunciados son una característica de CTA?
 - a. Visibilidad y Detectabilidad.
 - b. Prototipado en experiencias de usuario.
 - c. Funcionalidad y Contexto.
 - d. Facticidad de diseño web.
5. ¿Indique cuál es la importancia de UX?





- a. Un buen diseño de UX mejora la satisfacción del cliente, los productos digitales bien diseñados permiten a los usuarios completar tareas de manera más rápida y eficiente y esto a su vez se traduce en una mejor manera de identificar problemas de usabilidad en etapas tempranas del desarrollo puede reducir significativamente los costos de desarrollo y mantenimiento del software.
 - b. Se centra en la información mediante un buen diseño de UX mejora la satisfacción del cliente, lo que puede aumentar la lealtad y la retención como cliente potencial para futuros desarrollos.
 - c. Se focaliza en Interfaces con herramientas de los productos digitales bien diseñados esto permite identificar problemas de usabilidad en etapas tempranas del desarrollo puede reducir significativamente los costos de desarrollo y mantenimiento del software.
 - d. Recopilación para un buen diseño de UX mejora la satisfacción del usuario permitiéndole completar tareas de manera más efectiva y esto a su vez se traduce en una mejor manera del software.
 - e. Representa la importancia que genera el diseño de UX en distintas ramas que giran en torno al usuario ya que conlleva una especie de fidelización digital entre usuario y empresa permitiéndole a la empresa tener un mejor margen al obtener nuevos y potenciales clientes y de esa forma fidelizarlos en el proceso.
6. ¿Indique que significa las siglas de Método AIDA?
- a. Análisis / Inteligencia / Diseño / Autoridad
 - b. Atención / Interés / Deseo / Acción
 - c. Acción / Inteligencia / Diseño / Acción
 - d. Atención / Interés / Diseño / Acción
 - e. Antiguo/ Interés / Diseño / Acción
7. ¿Indique que cual es el principal propósito del CTA en una página web?
- a. Incentivar el consumo de clics en la web.
 - b. Motivar a los usuarios a actuar de inmediato. Con el fin de lograr mediante el uso de palabras que indiquen limitación
 - c. Desviar la atención del usuario y dirigirlos hacia una Interacción deseada





- d. Es guiar a los usuarios hacia una acción específica que los diseñadores o las empresas desean que realicen.
8. ¿Quién es el creador de la expresión UX y por qué?
- a. Donald Norman - Es porque quería abarcar todos los aspectos de la interacción de una persona con una compañía, sus servicios y productos.
 - b. Tim Barnes-Ben - Es porque quería abarcar todos los aspectos de la interacción de una persona con una compañía, sus servicios y productos.
 - c. Tim Burton - Es porque quería abarcar todos los aspectos del diseño en una persona con una compañía, sus servicios y productos.
 - d. David Norman - Es porque quería abarcar todos los aspectos de la interacción de una persona con una compañía, sus servicios y productos.
9. ¿Indique cuáles son los objetivos del Diseño UX?
- a. Satisfacción del Usuario
 - b. Facilidad de Uso
 - c. Accesibilidad
 - d. Interdisciplinariedad
 - e. Todas las Anteriores
 - f. Ninguna
10. ¿Indique la definición correcta de UI?
- a. Es el espacio donde ocurren las interacciones entre humanos y máquinas. Se centra en el diseño visual de la aplicación y la manera en que se presentan los elementos gráficos y textuales al usuario
 - b. Es el espacio donde ocurren las interacciones entre humanos y máquinas. Se centra en el diseño.
 - c. Herramientas que define el diseño de la aplicación.
 - d. Es el espacio donde ocurren las interacciones y diseño de interfaces.



Resumen de la Unidad 2

En la segunda unidad hemos abordado las tendencias y desafíos del diseño UX/UI el mismo que cuenta con diseño inclusivo permitiéndonos crear productos que sean accesibles para todos los usuarios, independientemente de sus capacidades. A su vez que tengan también un diseño emocional para así apelar a las emociones de los usuarios para crear conexiones más profundas con las marcas. En conclusión, el diseño UX/UI es un campo apasionante que combina creatividad, tecnología y psicología. Al dominar los fundamentos y las últimas tendencias, podrás crear productos digitales que sean tanto útiles como agradables de usar.

UX son las siglas de User Experience, que en español significa Experiencia de Usuario y se refiere a todos las sensaciones y pensamientos que experimenta una persona al interactuar con un producto, sistema o servicio en pocas palabras, el UX se enfoca en hacer que la interacción con un producto sea lo más sencilla, agradable y eficiente posible. Incluyendo su usabilidad: ¿Es fácil de usar? ¿Se encuentra lo que se busca rápidamente? En su utilidad: ¿Cumple el producto con las necesidades del usuario? A si mismo debemos cubrir religiosamente su deseabilidad: ¿Es atractivo visualmente? ¿Es entretenido de usar? Y si duda alguna su valor percibido: ¿Vale la pena el producto? ¿Qué beneficios aporta al usuario? no podemos olvidar la importancia de su accesibilidad: ¿Pueden usarlo todas las personas, sin importar sus capacidades? Una vez cubierto esto podemos decir que los usuarios felices son más propensos a volver y recomendar el producto. Los diseñadores UX son los profesionales que se encargan de crear experiencias de usuario excepcionales. Utilizan una variedad de herramientas y métodos para investigar a los usuarios, diseñar prototipos y evaluar la efectividad de sus diseños. En resumen, el UX es fundamental para el éxito de cualquier producto digital. Al centrarse en las necesidades y deseos de los usuarios, las empresas pueden crear productos que sean más útiles, atractivos y fáciles de usar.

UI en su defecto la Interfaz de Usuario (UI, por sus siglas en inglés) representa la capa tangible de un producto digital con la cual interactúa directamente el usuario. Es el punto de encuentro entre el ser humano y la tecnología, y como tal, desempeña un papel crucial en la experiencia general del usuario, el mismo cuenta con componentes clave como:

- Elementos visuales: Todos los componentes gráficos que conforman la interfaz, como botones, campos de texto, iconos, imágenes y tipografías. Estos elementos deben ser coherentes con la identidad visual de la marca y transmitir la información necesaria de manera clara y concisa.





- **Diseño de interacción:** Define cómo los usuarios interactúan con la interfaz a través de gestos, clics, desplazamientos y otros comandos. Un buen diseño de interacción garantiza que las acciones del usuario sean intuitivas y produzcan los resultados esperados.
- **Layout y maquetación:** La disposición de los elementos en la pantalla es fundamental para crear una interfaz visualmente atractiva y fácil de navegar. Se utilizan principios de diseño como la jerarquía visual, el equilibrio y el ritmo para organizar la información de manera efectiva.
- **Prototipado:** La creación de prototipos interactivos permite evaluar y refinar el diseño de la interfaz antes de su desarrollo final. Los prototipos pueden ser de baja o alta fidelidad, dependiendo de la etapa del proyecto.

Un diseño UI bien ejecutado ofrece múltiples beneficios por ende mejora la usabilidad y facilita que los usuarios encuentren la información que necesitan y completen las tareas de manera eficiente esto se trasmite en un aumento de la satisfacción del usuario ya que la interfaz estéticamente agradable y fácil de usar genera una experiencia positiva para el usuario.

Esto permite reforzar la identidad de marca la cual debe ser una imagen coherente y memorable. Si bien UI y UX están estrechamente relacionados, se enfocan en aspectos diferentes:

- **UI:** Se centra en la apariencia visual y la interacción directa con el usuario.
- **UX:** Abarca la experiencia completa del usuario, desde el descubrimiento del producto hasta su uso a largo plazo.

En conclusión, el diseño UI es una disciplina fundamental en el desarrollo de productos digitales. Un diseño UI eficaz no solo mejora la estética de un producto, sino que también contribuye a su éxito comercial al proporcionar una experiencia de usuario positiva y memorable. Ahora durante esta unidad hemos revisado un elemento esencial en la conversión digital este se le conoce como una llamada a la acción (CTA, por sus siglas en inglés) es un elemento de diseño web diseñado estratégicamente para incitar a un usuario a realizar una acción específica y medida. En términos más técnicos, es un componente interactivo que guía al usuario a lo largo del embudo de conversión, convirtiendo a un visitante casual en un cliente potencial o cliente fidelizado el mismo cuenta con las siguientes características: Claridad, Concisión, Relevancia Urgencia y Visibilidad un CTA bien diseñado es una herramienta poderosa para impulsar las conversiones y alcanzar los objetivos de marketing digital. Al comprender los principios fundamentales detrás de los CTA y aplicarlos de manera estratégica, las empresas pueden mejorar significativamente su rendimiento en línea.





UNIDAD 3 (HTML, CSS y JS)

Maneja técnicas de programación del lado del cliente como HTML, CSS, JS y Bootstrap. Maneja conceptos de páginas web estáticas basadas en estándares Bootstrap.

Temas y Subtemas

Importancia de Validación de Formularios HTML, CSS y JS Pagina con formularios y validación de datos

Uso de plantillas Bootstrap

Instalación Bootstrap en React

HTML: El Lenguaje Base de la Web

HTML (HyperText Markup Language) es el lenguaje de marcado de hipertexto utilizado para estructurar y presentar contenido en la World Wide Web. Es el componente fundamental de las páginas web, definiendo la semántica y la estructura de un documento considerado como un lenguaje estándar utilizado para crear y estructurar páginas web. (Mozilla FireFox, 2024)

Desarrollado originalmente en 1991 por Tim Berners-Lee, HTML permite a los desarrolladores definir la estructura y el contenido de una página web mediante el uso de elementos y etiquetas. Cada elemento HTML tiene una función específica, ya sea para formatear texto, crear enlaces, incrustar imágenes, o definir la estructura de la página.

Elementos Clave de HTML:

- **Etiquetas:** Son palabras clave encerradas entre corchetes angulares (< y >) que indican al navegador cómo interpretar el contenido. Por ejemplo, la etiqueta <h1> define un encabezado de nivel 1.





- **Atributos:** Son características adicionales que se añaden a las etiquetas para proporcionar información sobre el elemento. Por ejemplo, el atributo href en una etiqueta <a> especifica la URL a la que enlaza.
- **Elementos:** Son las unidades básicas de una página HTML, formados por una etiqueta de apertura, contenido y una etiqueta de cierre. Por ejemplo, <p>Este es un párrafo</p>.

Estructura básica de un documento HTML:

<!DOCTYPE html>: Declara el tipo de documento HTML.

<html>: Raíz del documento HTML.

<head>: Contiene metadatos sobre la página, como el título, codificación de caracteres y enlaces a hojas de estilo.

<body>: Contiene el contenido visible de la página.

Elementos HTML Importantes:

- Encabezados: <h1>, <h2>, ..., <h6>. Definen el nivel de importancia de un texto.
- Párrafos: <p>. Define un párrafo de texto.
- Enlaces: <a>. Crea un hipervínculo a otra página o recurso.
- Imágenes: . Inserta una imagen en la página.
- Listas: (listas no ordenadas), (listas ordenadas), (elementos de lista).
- Tablas: <table>, <tr>, <th>, <td>. Organizan datos en filas y columnas.
- Formularios: <form>, <input>, <button>. Permiten la interacción del usuario con la página.
- Divs y Spans: <div> y son contenedores genéricos para agrupar elementos y aplicar estilos.

HTML es el pilar fundamental de la creación de páginas web. Comprender sus elementos y su estructura es esencial para cualquier desarrollador web. Al dominar HTML, puedes construir páginas web sólidas y semánticamente correctas, lo que mejora la accesibilidad y la optimización para motores de búsqueda, cabe mencionar que dentro de la estructura de HTML este debe abrirse y cerrarse es decir que todos los elementos que se crean en la estructura deben cerrarse para no presentar inconvenientes. (Duckett, 2011)





HTML no es un lenguaje de programación, sino un lenguaje de marcado que estructura el contenido en la web. Cada elemento HTML está compuesto por etiquetas que encierran el contenido y le dicen al navegador cómo debe interpretarlo. (Robbins, 2018)

Ejemplo Completo de un Documento HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mi Primera Página Web</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>Bienvenidos a Mi Página Web</h1>
    <p>Este es un ejemplo de un sitio web sencillo utilizando HTML.</p>
  </header>
  <nav>
    <ul>
      <li><a href="#inicio">Inicio</a></li>
      <li><a href="#sobre-mi">Sobre mí</a></li>
      <li><a href="#contacto">Contacto</a></li>
    </ul>
  </nav>
  <section id="inicio">
    <h2>Inicio</h2>
    <p>Esta es la sección de inicio de mi página web.</p>
    
  </section>
  <section id="sobre-mi">
    <h2>Sobre mí</h2>
    <p>Información sobre el autor del sitio web.</p>
  </section>
  <footer id="contacto">
    <h2>Contacto</h2>
    <p>Puede contactarme a través del siguiente email : <a
href="mailto:email@ejemplo.com">email@ejemplo.com</a>.</p>
  </footer>
</body>
</html>
```

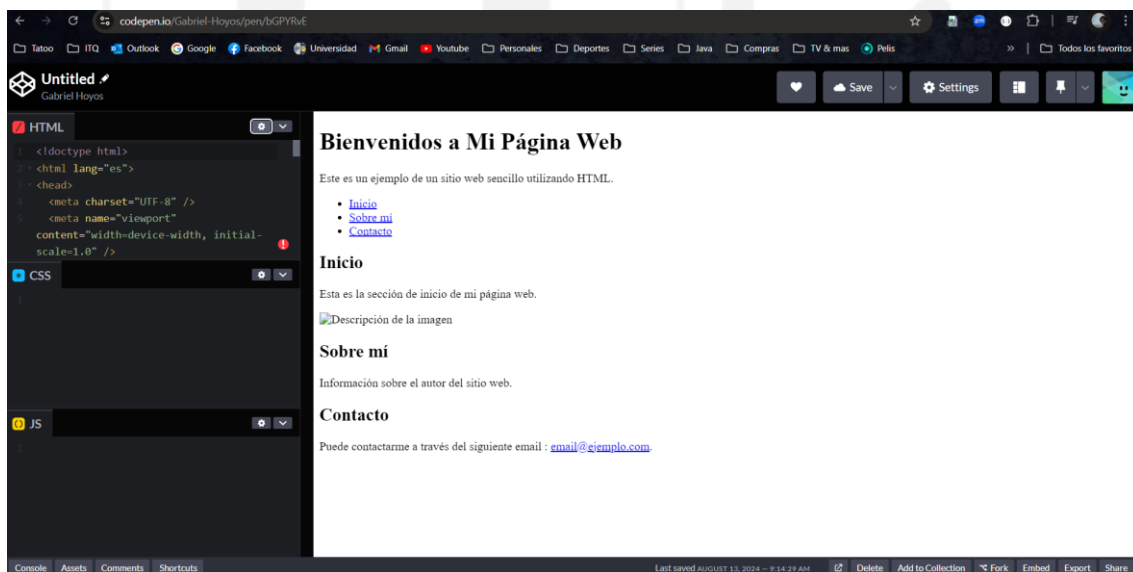
Explicación del Ejemplo:

A continuación indicaremos que realizan cada uno los atributos y etiquetas que se utilizaron en este documento HTML para ello hemos utilizado la herramienta de diseño codeOpen que permite visualizar e interpretar el código HTML ingresado.



- `<!DOCTYPE html>`: declaración dentro del documento donde se indica que utiliza HTML5.
- `<html lang="es">`: Indica que el idioma del contenido a presentar en la página es español.
- `<head>`: Contiene el título de la página y enlaces a hojas de estilo externas.
- `<body>`: Contiene el contenido visible para los usuarios, estructurado en secciones (`<header>`, `<nav>`, `<section>`, `<footer>`).
- Navegación (`<nav>`): Crea un menú de navegación con enlaces a diferentes secciones de la página.
- Imágenes (``): Inserta una imagen con un texto alternativo para accesibilidad.
- Enlaces (`<a>`): Incluye enlaces a diferentes secciones y un correo electrónico de contacto.

Figura 16:
Interprete CodeOpen del Ejemplo planteado



Elaboración: Autor

En la Figura 16 podemos ver el ejemplo interpretado por CodeOpen donde en la parte superior izquierda hemos copiado la estructura del ejemplo y automáticamente el intérprete nos muestra en pantalla la salida de los datos ingresados, en la herramienta en su apartado HTML y así mismo tenemos espacio de trabajo para trabajar con lenguajes de modelado como es CSS y un lenguaje de programación como lo es JS.



CSS: El Maquillaje de la Web

CSS (Cascading Style Sheets, o Hojas de Estilo en Cascada) es un lenguaje de diseño que se utiliza para dar estilo y formato a las páginas web. Si HTML es el esqueleto de una página, CSS es su maquillaje. Define cómo se ven los elementos HTML, desde los colores y las fuentes hasta el diseño y el posicionamiento en la pantalla del elemento al que se hace referencia dentro de la hoja de estilo, como bien se indica este puede referenciarse para que tome atributos en la presentación o visualización de este.

¿Por qué es importante CSS?

- **Separación de contenido y presentación:** Permite mantener el código HTML limpio y organizado, facilitando la actualización y el mantenimiento del sitio web.
- **Consistencia visual:** Permite aplicar estilos a múltiples elementos de una página o incluso a todo un sitio web, asegurando una apariencia uniforme.
- **Responsivo:** Permite crear diseños que se adapten a diferentes tamaños de pantalla, como computadoras de escritorio, tablets y teléfonos móviles.
- **Personalización:** Ofrece un gran control sobre la apariencia de una página web, permitiendo crear diseños únicos y personalizados.

Elementos clave de CSS:

- **Selectores:** Son las partes de una regla CSS que especifican a qué elementos HTML se aplicarán los estilos. Pueden ser selectores simples (por ejemplo, p para todos los párrafos), selectores de clase (por ejemplo, destacado) o selectores de ID (por ejemplo, #mi-encabezado).
- **Propiedades:** Son las características que se pueden aplicar a los elementos seleccionados, como color, font-size, background-color, margin, padding, etc.
- **Valores:** Son los valores asignados a las propiedades para definir el estilo exacto. Por ejemplo, color: blue establece el color del texto a azul.
- **Reglas CSS:** Una regla CSS consta de un selector y una declaración de una o más propiedades.



La sintaxis básica es:

```
selector {  
    propiedad1: valor1;  
    propiedad2: valor2;  
}
```

Ejemplo de una regla CSS:

```
p {  
    color: blue;  
    font-size: 16px;  
}
```

Esta regla aplicará el color azul y un tamaño de fuente de 16 píxeles a todos los párrafos de la página.

Ejemplo de un documento CSS:

A continuación, realizaremos el diseño del ejemplo anterior mejorándolo visualmente con los estilos de CSS lograremos que se vea estéticamente mejor para una presentación así mismo se procede a explicar el proceso del ejemplo

*Figura 17:
Mejora estética con Estilos CSS*



Elaboración: Autor

En la Figura 17 podemos evidenciar el cambio radical que ha sufrido la página una vez que se ha realizado el llamado a la referencia de una hoja de estilos CSS esto dentro del documento principal HTML en la sección encabezado head encontraran como se está llamando a los estilos de CSS.



Hoja de Estilos (CSS):

```
/* Estilos generales */
```

```
body {  
    font-family: Arial, sans-serif;  
    line-height: 1.6;  
    margin: 0;  
    padding: 0;  
    background-color: #f4f4f4;  
    color: #333;  
}
```

```
/* Encabezado */
```

```
header {  
    background: #4CAF50;  
    color: white;  
    padding: 20px;  
    text-align: center;  
}
```

```
/* Navegación */
```

```
nav {  
    background: #333;  
    color: white;  
    padding: 10px;  
    text-align: center;  
}
```

```
nav ul {  
    list-style: none;  
    margin: 0;  
    padding: 0;  
}
```

```
nav ul li {  
    display: inline;  
    margin: 0 10px;  
}
```





```
nav ul li a {  
    color: white;  
    text-decoration: none;  
    font-weight: bold;  
}  
  
nav ul li a: hover {  
    text-decoration: underline;  
}  
  
/* Secciones */  
section {  
    margin: 20px;  
    padding: 20px;  
    background: white;  
    border-radius: 5px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}  
  
section img {  
    max-width: 100%;  
    height: auto;  
    display: block;  
    margin: 0 auto;  
    border-radius: 5px;  
}  
  
/* Pie de página */  
footer {  
    background: #333;  
    color: white;  
    text-align: center;  
    padding: 10px;  
    margin-top: 20px;  
}
```



```
footer a {  
    color: #4CAF50;  
    text-decoration: none;  
}
```

```
footer a:hover {  
    text-decoration: underline;  
}
```

Explicación del Código

En este apartado del diseño va a existir un documento CSS (styles.css) mismo que contendrá el código y las referencias y el cual debe ser instanciado o referenciado en el documento HTML para que puedan visualizarse los cambios:

- Estilos Generales: Se establecen estilos básicos para el body, como la fuente, el color del texto, y el fondo general de la página.
- Estilo del Encabezado (header): El encabezado tiene un fondo verde (#4CAF50), texto centrado y con un color blanco.
- Estilo de la Navegación (nav): La navegación tiene un fondo oscuro, con enlaces que se resaltan al pasar el cursor sobre ellos.
- Estilo de las Secciones (section): Cada sección tiene un fondo blanco con sombras suaves y bordes redondeados para darle un aspecto más moderno y agradable.
- Las imágenes dentro de las secciones se ajustan para ser responsivas y centradas.
- Estilo del Pie de Página (footer): El pie de página mantiene el esquema de color oscuro de la navegación y destaca los enlaces con un color verde que hace juego con el encabezado.

Ahora con esto podemos realizar el desarrollo y diseño de una página web a su vez debemos tener en cuenta que la estructuras tanto de HTML 5 así como CSS se manejan como hemos visto a lo largo de la unidad esto se realiza a través de un editor de texto en el mismo vamos a poder crear nuestros documentos de forma ordenada y referenciarlos según la posición donde se encuentren almacenados



¿Qué es JS o JavaScript?

JavaScript es un lenguaje de programación interpretado que se utiliza principalmente para crear páginas web interactivas. A diferencia de HTML, que estructura el contenido, y CSS, que define el estilo, JavaScript añade comportamiento a las páginas web. Permite que los elementos de una página respondan a las acciones del usuario, como hacer clic en un botón, mover el ratón o cargar una página.

Fundamentos de JavaScript:

- Variables: Almacenan datos que pueden cambiar a lo largo de la ejecución del programa.
- Tipos de datos: JavaScript maneja diversos tipos de datos, como números, cadenas de texto, booleanos (verdadero o falso), objetos, etc.
- Operadores: Permiten realizar operaciones matemáticas, lógicas y de comparación.
- Control de flujo: Estructuras como if, else, for, while, etc., permiten controlar el flujo de ejecución del programa.
- Funciones: Son bloques de código reutilizables que realizan tareas específicas.
- Objetos: Son colecciones de propiedades (atributos) y métodos (funciones) que representan entidades del mundo real.

Características Principales de JavaScript:

- Interpretado: El código JavaScript se ejecuta directamente por el navegador, sin necesidad de compilarlo previamente.
- Débilmente tipado: No es necesario declarar el tipo de dato de una variable antes de usarla.
- Basado en prototipos: Los objetos heredan propiedades y métodos de otros objetos a través de prototipos.
- Multiplataforma: JavaScript se ejecuta en la mayoría de los navegadores web y en otros entornos.
- Asíncrono: Permite ejecutar tareas de forma independiente, sin bloquear la ejecución del programa principal.





Estructuras de JavaScript:

- **Variables:** Se utilizan para almacenar datos y se declaran con las palabras clave var, let o const.
- **Tipos de datos:** Incluye números, cadenas, booleanos, objetos, arrays, funciones, etc.
- **Operadores:** Se utilizan para realizar operaciones aritméticas, lógicas y de comparación.
- **Control de flujo:** Estructuras como if, else, for, while y switch controlan el flujo de ejecución del código.
- **Funciones:** Son bloques de código reutilizables que realizan una tarea específica.
- **Objetos:** Son instancias de clases y tienen propiedades y métodos.
- **Arrays:** Son colecciones ordenadas de elementos.

Es decir, JavaScript es un lenguaje de programación interpretado, de alto nivel, orientado a objetos y de propósito general, comúnmente utilizado para crear interfaces de usuario dinámicas e interactivas en páginas web. Se ejecuta directamente en el navegador del cliente, sin requerir una compilación previa del código.

Características Distintivas:

- **Orientación a objetos basada en prototipos:** JavaScript utiliza un modelo de objetos basado en prototipos, lo que significa que los objetos heredan propiedades y métodos de otros objetos a través de una cadena de prototipos.
- **Tipado dinámico:** No es necesario declarar el tipo de dato de una variable antes de usarla, lo que proporciona flexibilidad, pero puede introducir errores si no se maneja con cuidado.
- **Funcionalidad:** JavaScript es un lenguaje funcional, lo que significa que las funciones son ciudadanos de primera clase y pueden asignarse a variables, pasarse como argumentos y devolverse como valores.
- **Asincronía:** JavaScript utiliza un modelo de ejecución asíncrono basado en la devolución de promesas, lo que permite realizar tareas en segundo plano sin bloquear la ejecución del código principal.
- **DOM (Document Object Model):** JavaScript interactúa con el DOM para manipular el contenido y el estilo de una página web.



Aplicaciones de JavaScript:

- **Creación de interfaces de usuario interactivas:** Formularios dinámicos, animaciones, efectos visuales.
- **Desarrollo de aplicaciones web de una sola página (SPA):** Aplicaciones web que se cargan una sola vez y se actualizan dinámicamente.
- **Desarrollo de juegos:** Creación de juegos simples y complejos utilizando frameworks como Phaser o Three.js.
- **Desarrollo de aplicaciones móviles:** Utilizando frameworks como React Native o Ionic.
- **Desarrollo del lado del servidor:** Con Node.js, JavaScript se puede utilizar para crear aplicaciones del lado del servidor.

Por lo tanto, JavaScript es un lenguaje de programación versátil y poderoso que ha revolucionado el desarrollo web. Su capacidad para crear experiencias de usuario ricas y dinámicas lo ha convertido en una tecnología esencial para cualquier desarrollador web.

Por ende, en esta sección del documento utilizaremos JavaScript y otros elementos que nos permitan validar los formularios que vamos a ir desarrollando.

¿Qué es la validación de formularios?

La validación de formularios es el proceso de verificar que los datos ingresados por un usuario en un formulario web sean correctos y cumplan con ciertos criterios establecidos. Esto es crucial para garantizar la integridad de los datos que se envían a un servidor y para evitar errores en el procesamiento de la información. Al ser un aspecto fundamental cuando se crea una página web que capturará información, y más aún si esta información será almacenada en una base de datos. La validación consiste en verificar que la información ingresada, sea correcta y coherente de acuerdo con lo que se requiere en cada campo. Es decir, se verifica como si un campo que es obligatorio este vacío, si un campo que debe contener valores numéricos tiene un número asignado, si los valores ingresados en un campo están en cierto rango, etc.

¿Por qué es importante la validación de formularios?

- **Seguridad:** Evita la inyección de código malicioso.
- **Integridad de datos:** Asegura que los datos sean consistentes y útiles.
- **Experiencia del usuario:** Proporciona retroalimentación inmediata al usuario sobre errores en los datos ingresados, evitando frustraciones.



- Optimización de procesos: Reduce la carga en el servidor al filtrar datos inválidos antes de enviarlos.

Reglas de validación:

- Requeridos: Campos que deben ser rellenados obligatoriamente.
- Formato: Verificación de que los datos tengan el formato correcto (ej: email, número telefónico).
- Longitud: Limitación del número de caracteres permitidos.
- Rango: Aseguramiento de que los valores estén dentro de un rango específico.
- Patrones: Uso de expresiones regulares para verificar patrones complejos (ej: contraseñas fuertes).
- Comparaciones: Comparación de valores entre diferentes campos (ej: confirmación de contraseña).

Mensajes de error:

- Claros y concisos: Explican al usuario qué está mal con los datos ingresados.
- Ubicados cerca del campo: Facilitan la identificación del error.
- Personalizados: Adaptados a cada tipo de error.

Mecanismos de validación:

1. **Validación en el lado del cliente:** Se realiza en el navegador del usuario, utilizando JavaScript, para proporcionar una respuesta inmediata al usuario.
2. **Validación en el lado del servidor:** Se realiza en el servidor, como una medida de seguridad adicional, para verificar nuevamente los datos antes de almacenarlos.

Ejemplo de validación en JavaScript:

```
function validarFormulario() {  
  const nombre = document.getElementById('nombre').value;  
  const email = document.getElementById('email').value;  
  if (nombre=== "") {  
    alert("Por favor, ingresa tu nombre.");  
    return false;  
  }  
}
```



```
if (!isEmailValido(email)) {  
    alert('Por favor, ingresa un correo electrónico válido.');
```



```
    return false;  
}  
// Resto de las validaciones  
return true;  
}  
function isEmailValido(email) {  
    // Expresión regular para validar correos electrónicos  
    const regex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;  
    return regex.test(email);  
}
```

Una de las características más importantes de los controles de formulario de HTML5 es la capacidad de validar la mayoría de los datos de usuario sin depender de JavaScript. Esto se realiza mediante el uso de atributos de validación en los elementos del formulario como:

- **required:** Especifica si un campo de formulario debe completarse antes de que se pueda enviar el formulario.
- **minlength** y **maxlength:** Especifican la longitud mínima y máxima de los datos de texto (cadenas).
- **min** y **max:** Especifican los valores mínimo y máximo de los tipos de entrada numéricos.
- **type:** Especifica si los datos deben ser un número, una dirección de correo electrónico o algún otro tipo de reajuste específico.
- **pattern:** Especifica una expresión regular que define un patrón que los datos que se introduzcan deben seguir.

Si los datos que se introducen en un campo de formulario siguen todas las reglas que especifican los atributos anteriores, se consideran válidos. Si no, se consideran no válidos.

En resumen, la validación de formularios es una práctica fundamental para asegurar la calidad y la seguridad de las aplicaciones web. Al implementar reglas de validación claras y mensajes de error útiles, se mejora significativamente la experiencia del usuario y se protegen los datos de la aplicación.



Bootstrap: Un Framework Fundamental en el Desarrollo Web Frontend

Es un framework de desarrollo web frontend de código abierto altamente popular, diseñado para acelerar y simplificar el proceso de creación de interfaces de usuario responsivas y visualmente atractivas. Al proporcionar un conjunto coherente de componentes prediseñados, utilidades de JavaScript y unas potentes herramientas de diseño, Bootstrap ha revolucionado la forma en que los desarrolladores abordan proyectos web. Es decir que Bootstrap se considera como un marco de trabajo frontend de referencia para la creación de aplicaciones web. (Deramond, Mark, & Thornton, 2024)

Fue creado por dos desarrolladores de Twitter, Mark Otto y Jacob Thornton, con el objetivo de simplificar y agilizar el proceso de diseño y desarrollo web. Al ser un framework frontend de código abierto altamente popular, diseñado para acelerar el desarrollo de aplicaciones web responsivas y visualmente atractivas y así proporcionar una extensa colección de componentes prediseñados, estilos y utilidades, Bootstrap simplifica la creación de interfaces de usuario consistentes y adaptables a diversas plataformas y dispositivos.

Características:

- **Responsividad integrada:** Asegura que las interfaces de usuario se adapten de manera fluida a diferentes tamaños de pantalla y dispositivos, optimizando la experiencia del usuario en diversos contextos. Es decir, es responsivo ya que permite crear diseños que se ajusten automáticamente a diferentes dispositivos y orientaciones, gracias a sus clases de utilidad y sistema de cuadrícula.
- **Consistencia visual:** Establece un lenguaje de diseño unificado, promoviendo la coherencia estética en toda la aplicación y facilitando la creación de interfaces intuitivas.
- **Modularidad:** Permite a los desarrolladores seleccionar y combinar componentes individuales para construir interfaces personalizadas, adaptándose a las necesidades específicas de cada proyecto.
- **Extensibilidad:** Ofrece una amplia gama de opciones de personalización, permitiendo a los desarrolladores modificar los estilos predeterminados y crear componentes personalizados.
- **Rendimiento optimizado:** Está construido con un enfoque en el rendimiento, utilizando técnicas de optimización para garantizar una carga rápida de las páginas y una experiencia de usuario fluida.





- **Sistema de cuadrícula flexible:** Facilita la creación de diseños responsivos y adaptables a diferentes tamaños de pantalla, mediante un sistema de cuadrícula basado en filas y columnas.
- **Componentes UI predefinidos:** Ofrece una amplia gama de elementos de interfaz de usuario, como botones, formularios, tablas, navegaciones, carruseles, etc., que pueden ser personalizados fácilmente.
- **Utilidades CSS:** Incluye una serie de clases CSS para aplicar estilos rápidamente a elementos HTML, como márgenes, padding, colores, tipografías y más.
- **Extensibilidad:** Se puede personalizar y ampliar con temas, plugin y componentes adicionales.

Beneficios de su uso:

- **Ahorro de tiempo y recursos:** Los componentes prediseñados y la estructura modular de Bootstrap reducen significativamente el tiempo de desarrollo, permitiendo a los equipos centrarse en la lógica de la aplicación.
- **Mejora de la calidad del código:** Al proporcionar una base sólida y consistente, Bootstrap fomenta la escritura de código limpio y mantenible.
- **Mayor accesibilidad:** Los componentes de Bootstrap están diseñados teniendo en cuenta los estándares de accesibilidad, lo que garantiza que las aplicaciones sean accesibles para todos los usuarios.
- **Amplia comunidad:** Cuenta con una comunidad activa de desarrolladores que contribuyen a su evolución y ofrecen soporte a través de foros, documentación y recursos en línea.

Integración de Bootstrap a proyectos web

- **Descarga directa:** Descarga los archivos CSS y JavaScript y agrégalos a tu proyecto.
- **CDN (Content Delivery Network):** Utiliza un CDN para cargar los archivos de Bootstrap desde un servidor externo.
- **Gestor de paquetes:** Instala Bootstrap a través de un gestor de paquetes como npm o yarn.





Categorías Principales de Clases y Atributos

➤ Sistema de Cuadrícula:

- ❖ **Filas y columnas:** row, col-sm-6, col-md-4, etc. Definen la estructura básica de un layout, dividiendo el contenido en filas y columnas.
- ❖ **Offsets:** offset-md-3, desplazan elementos dentro de una fila.
- ❖ **Orden:** order-first, order-last, controlan el orden de los elementos en diferentes tamaños de pantalla.
- ❖ **Justificación y alineación:** justify-content-center, align-items-center, controlan la alineación de los elementos dentro de un contenedor.

➤ Tipos de Contenido:

- ❖ **Textos:** text-center, text-primary, font-weight-bold. Aplican estilos a los textos, como alineación, color y peso de la fuente.
- ❖ **Imágenes:** img-fluid, rounded, rounded-circle. Controlan el tamaño, forma y otros aspectos de las imágenes.
- ❖ **Botones:** btn, btn-primary, btn-lg. Crean botones con diferentes estilos y tamaños.
- ❖ **Formularios:** form-control, form-label, form-check. Estilizan los elementos de un formulario.
- ❖ **Tablas:** table, table-striped, table-bordered. Crean tablas con diferentes estilos.
- ❖ **Listas:** list-group, list-group-item. Estilizan listas.

➤ Utilidades:

- ❖ **Espaciado:** m-3, p-2. Controlan los márgenes y rellenos.
- ❖ **Texto:** text-uppercase, text-lowercase. Modifican el estilo del texto.
- ❖ **Fondo:** bg-primary, bg-gradient. Aplican colores de fondo y degradados.
- ❖ **Tamaño:** w-50, h-100. Controlan el ancho y alto de los elementos.
- ❖ **Visibilidad:** d-none, d-md-block. Controlan la visibilidad de los elementos en diferentes tamaños de pantalla.

➤ Componentes:

- ❖ **Navbar:** navbar, navbar-expand-lg, navbar-dark. Crean barras de navegación.
- ❖ **Card:** card, card-body, card-title. Crean tarjetas para presentar contenido.





- ❖ **Modal:** modal, modal-dialog, modal-content. Crean ventanas modales.
- ❖ **Carousel:** carousel, carousel-item. Crean carruseles para mostrar imágenes o contenido deslizable.

Ejemplo Práctico 1: Crear un botón primario y una fila con tres columnas además de crear una tarjeta con una imagen

HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
    integrity="sha384-
    rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65"
    crossorigin="anonymous">
</head>
<body>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
    kenU1KfDble4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4"
    crossorigin="anonymous"></script>
  <button type="button" class="btn btn-primary">Click me</button>
<div class="row">
  <div class="col-md-4">Columna 1</div>
  <div class="col-md-4">Columna 2</div>
  <div class="col-md-4">Columna 3</div>
</div>
<div class="card">
  
  <div class="card-body"></div>
</div>
</body>
</html>
```





Ejemplo Práctico 2: Crear un formulario en HTML con 5 campos de ingreso para el registro de pacientes en una clínica el mismo debe solicitar Nombre, Apellido, DNI , Fecha de Nacimiento y Correo , la fecha de nacimiento debe desplegarse con un calendario, utiliza Bootstrap

HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Formulario
de Registro de Pacientes</title>
  <link rel="stylesheet" href="style.css"> </head>
<body>
  <div class="container">
    <div class="row justify-content-center">
      <div class="col-md-6">
        <form>

          <div class="form-group">
            <label for="nombre">Nombre:</label>
            <input type="text" class="form-control" id="nombre" placeholder="Ingrese
su nombre">
          </div>

          <div class="form-group">
            <label for="apellido">Apellido:</label>
            <input type="text" class="form-control" id="apellido" placeholder="Ingrese
su apellido">
          </div>

          <div
class="form-group">
            <label for="dni">DNI:</label>
            <input type="text" class="form-control" id="dni" placeholder="Ingrese su DNI">
          </div>

          <div class="form-group">
            <label for="fechaNacimiento">Fecha
de Nacimiento:</label>
```





```
<input type="date" class="form-control" id="fechaNacimiento">
</div>
<div class="form-group">

  <label for="correo">Correo Electrónico:</label>
  <input type="email" class="form-control" id="correo" placeholder="Ingrese
su correo electrónico">
</div>

  <button type="submit" class="btn btn-primary">Registrar</button>
</form>
</div>
</div>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
C8ylhuAYhalmU7vzZbGr2jqoN9WMol3+315/ijv+UQnjE+oI05d+r/0Q+C7+94Q"
crossorigin="anonymous"></script>
</body>
</html>
```

STYLE.CSS

CSS

```
/* Estilos básicos de Bootstrap */
```

```
body {
```

```
  font-family: Arial, sans-serif;
```

```
}
```

```
.container {
```

```
  margin-top: 50px;
```

```
}
```

```
.form-control {
```

```
  border-radius: 0;
```

```
}
```



Explicación:

En la estructura HTML se utiliza el framework Bootstrap para organizar los elementos del formulario en una cuadrícula responsiva según los atributos y elementos que revisamos en esta unidad a su vez se agregaron los campos de entrada donde se crean los campos para cada uno de los datos solicitados (nombre, apellido, DNI, fecha de nacimiento y correo) coincidan con su elemento correspondiente. El calendario tiene un tipo de entrada date en el campo de fecha de nacimiento automáticamente muestra un selector de fecha en la mayoría de los navegadores modernos.

Se incluye el archivo JavaScript de Bootstrap para habilitar funcionalidades como el diseño responsivo y los componentes predefinidos. Además de estilos CSS incluyen los estilos básicos de Bootstrap para dar formato al formulario. Puedes personalizar estos estilos aún más para adaptarlos a tu diseño para el tema de la validación es recomendable agregar validación a los campos del formulario para asegurarte de que los datos ingresados sean correctos. Puedes utilizar JavaScript o bibliotecas de validación para esto. En su caso para utilizar Bootstrap utilizamos un CDN para incluirlo dentro de nuestro desarrollo, pero es preferible instalarlo y así puedes reemplazar la línea donde se vincula el archivo CSS por el enlace al CDN correspondiente.

Ahora esto nos permite tener el ahorro de tiempo y recursos ya que los componentes prediseñados y la estructura modular de Bootstrap se van a cargar directamente en el proyecto por ende reducen significativamente el tiempo de desarrollo, permitiendo a los equipos centrarse en la lógica de la aplicación. Además de mejorar la calidad del código al tener una base sólida, consistente y ordenada ya que Bootstrap fomenta la escritura de código limpio y mantenible.

Lo cual se traduce en una mayor escalabilidad de los componentes que están siendo diseñados teniendo en cuenta los estándares de accesibilidad, lo que garantiza que las aplicaciones sean accesibles para todos los usuarios. Por ello Bootstrap ofrece una variedad de plantillas y temas prediseñados que sirven como puntos de partida para diferentes tipos de proyectos. Estas plantillas proporcionan estructuras de diseño completas, incluyendo la disposición de elementos, estilos y esquemas de color, lo que agiliza aún más el proceso de desarrollo.

Cabe recalcar que ahora en la actualidad no solo se utiliza Bootstrap como frontend a mejorar en las aplicaciones, sino que también se combinan con los frontend actuales con un proceso de instalación por contexto para que pueda ser desarrollado en Angular, React o VueJs.



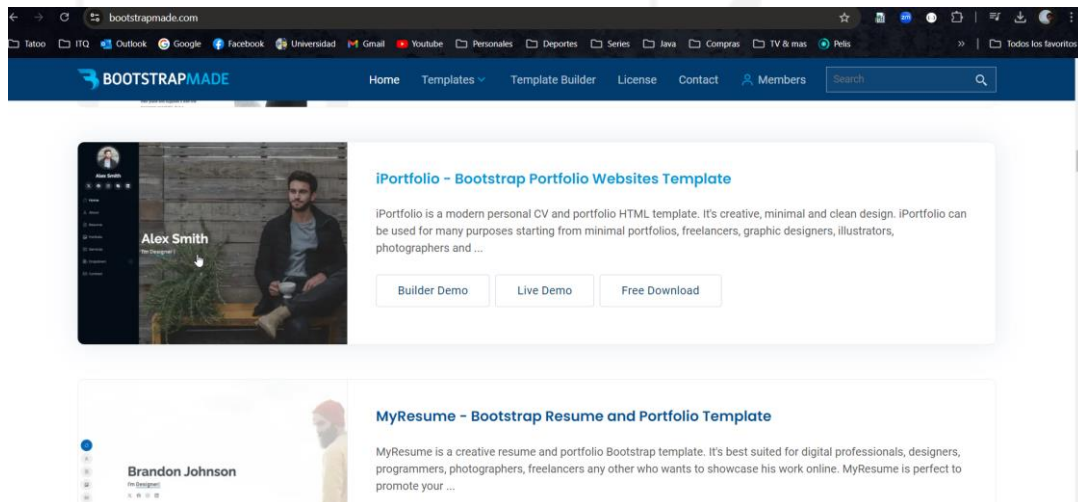
Personalización

Bootstrap permite personalizar los estilos predefinidos mediante variables CSS, mixins y temas. Esto te brinda la flexibilidad para crear diseños únicos y adaptados a tus necesidades.

Plantillas y Temas: Bootstrap ofrece una variedad de plantillas y temas prediseñados que sirven como puntos de partida para diferentes tipos de proyectos. Estas plantillas proporcionan estructuras de diseño completas, incluyendo la disposición de elementos, estilos y esquemas de color, lo que agiliza aún más el proceso de desarrollo. Hoy en día podemos encontrar infinidad de páginas que cumple un prediseño y son de mucha utilidad por ejemplo podemos encontrar paginas como:

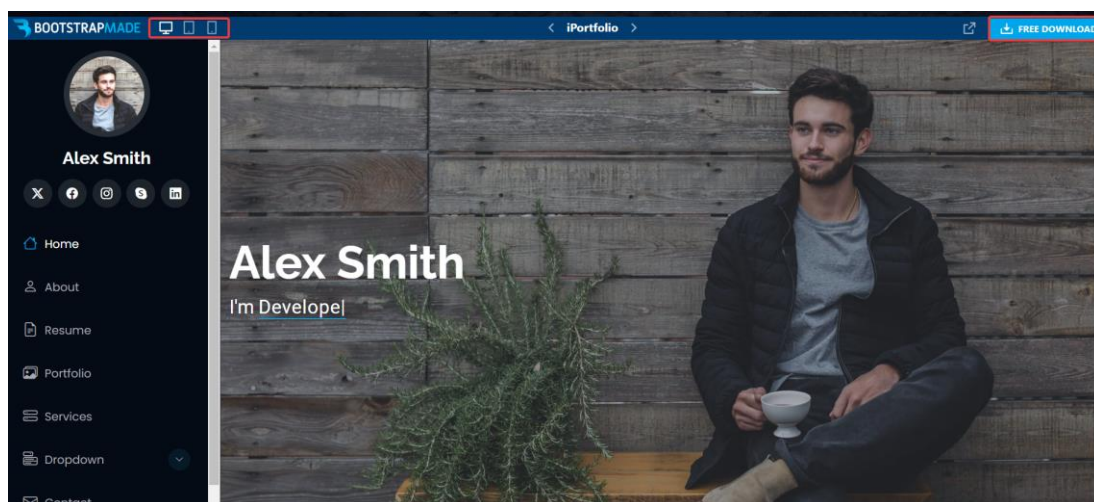
➤ <https://bootstrapmade.com/>

Figura 18:
BootstrapMade plantillas prefabricadas para diseño.



Fuente: <https://bootstrapmade.com/>

Figura 19:
Previsualización de Diseño BootstrapMade

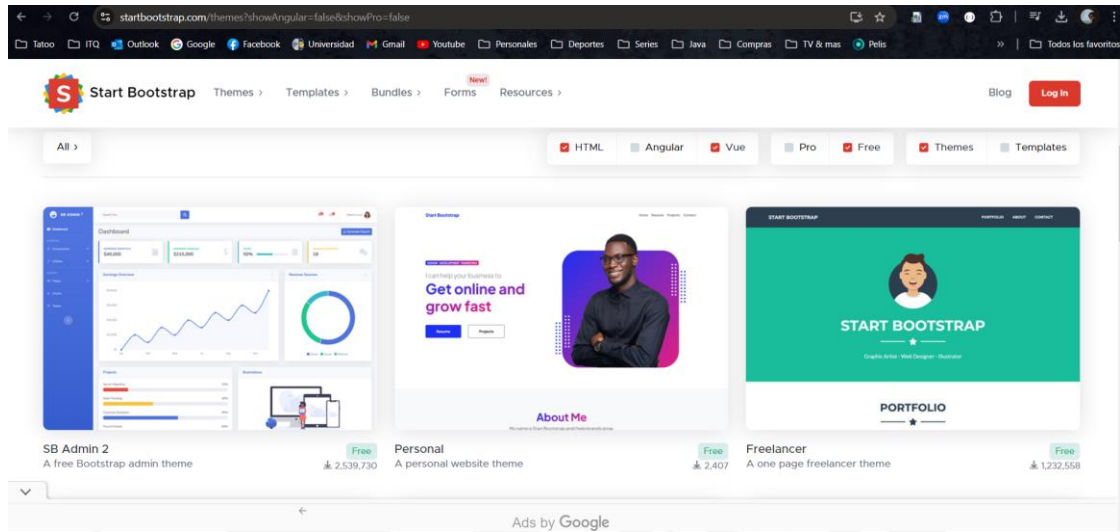


Fuente: <https://bootstrapmade.com/>

En la Figura 18 y 19 podemos ver que BootstrapMade es una página que permite tener una plantilla de diseño prefabricada, la misma debemos editarla para agregar los diseños o elementos que deseamos, cabe indicar que es de uso libre por lo que debemos realizar solo la descarga también existen plantillas de pago, pero son mucho más avanzadas o cuenta con una funcionalidad en específico.

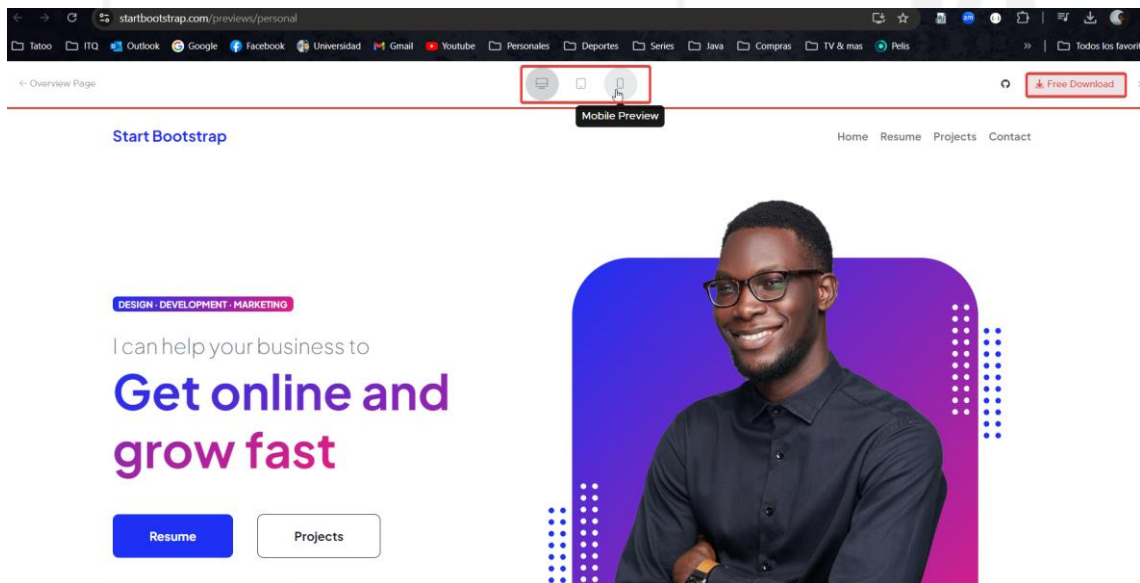
➤ <https://startbootstrap.com/>

Figura 20:
StartBootstrap Plantillas prefabricadas para diseño.



Fuente: <https://startbootstrap.com/>

Figura 21:
Previsualización de Diseño StartBootstrap



Fuente: <https://startbootstrap.com/>

En la Figura 20 y 21 podemos ver StartBootstrap es otra página que permiten tener unas plantillas de diseño prefabricada, la misma debemos editarla para agregar los diseños o elementos que deseamos, cabe indicar que es de uso libre por lo que debemos realizar solo la descarga también existen plantillas de pago, pero son mucho más avanzadas o cuenta con una funcionalidad en específico.

En ambas herramientas podremos tener una previsualización de la página que deseamos utilizar como plantilla de nuestro desarrollo una vez que ya tengamos claro lo que deseamos de este diseño prefabricado bastará con darle clic en bajar o descargar para que se descargue un proyecto con los directorios correspondientes a su funcionalidad la misma que será en HTML, JS y en este caso Bootstrap, aunque también puede existir el apartado de CSS.

De la misma forma en la plataforma que escojamos se puede ver los modos a lo cuales estará sujeta la información es decir que podremos previsualizar la página para ver como se mira en un equipo de escritorio, laptop, Tablet o celular. En conclusión, Bootstrap se ha consolidado como una herramienta esencial para cualquier desarrollador web frontend que busque crear aplicaciones web modernas y responsivas. Su enfoque, modularidad, consistencia y la facilidad de uso lo convierte en una elección popular tanto para proyectos pequeños como para grandes aplicaciones empresariales.

En resumen, Bootstrap es una herramienta esencial para cualquier desarrollador web que busque crear interfaces de usuario modernas, responsivas y consistentes. Al proporcionar un conjunto sólido de componentes y utilidades, Bootstrap simplifica el proceso de desarrollo y permite a los desarrolladores centrarse en la lógica de la aplicación.

Integración de Bootstrap en Proyectos React: Una Guía Técnica

La integración de Bootstrap, uno de los frameworks CSS más populares, en proyectos React, un framework de JavaScript para crear interfaces de usuario, **es una práctica común para acelerar el desarrollo y mejorar la apariencia de las aplicaciones web.** Esta guía detalla los dos métodos principales para llevar a cabo esta integración.

¿Por qué usar Bootstrap en React?

Bootstrap es un framework CSS popular que proporciona una gran variedad de componentes prediseñados para crear interfaces de usuario responsivas y atractivas. Al integrarlo en tu proyecto React, puedes acelerar significativamente el desarrollo **y obtener una apariencia consistente en toda tu aplicación.**





Dependencias: Al utilizar Bootstrap directamente, es posible que se requieran dependencias adicionales como jQuery y Popper.js, aunque React Bootstrap las maneja de forma interna.

Métodos de Integración: Existen 2 métodos de integrar Bootstrap con los proyectos de React el primero de forma manual utilizando instalaciones de contexto de Bootstrap y el segundo es el método más utilizado que es React Bootstrap pero que no está actualizado en su versión por esta razón en esta unidad revisaremos el primer método que ofrece mayor control sobre los estilos, pero requiere una configuración más manual es altamente recomendado debido a su integración con React, proporciona componentes diseñados específicamente para este framework.

➤ **Instalación:** En una terminal mediante Bash digitemos el siguiente comando

❖ `npm i --save bootstrap @popperjs/core`

➤ **Importación y Uso:** para poder utilizar Bootstrap en el proyecto necesitamos importarlo mediante JavaScript como se indica a continuación:

❖ `import 'bootstrap/dist/css/bootstrap.min.css';`

❖ **Ejemplo de uso:**

```
import { Formulario } from './components/Formulario'
```

```
import { Home } from './components/Home'
```

```
import { useState } from 'react'
```

```
import './App.css'
```

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

```
function App() {
```

```
  const [user, setUser] = useState([])
```

```
  return (
```

```
    <div>
```

```
      {
```

```
        !user.length > 0
```

```
        ?<Formulario setUser = {setUser}/>
```

```
        :<Home user = {user} setUser = {setUser}/>
```

```
      }
```

```
    </div>
```

```
  )
```

```
}
```

```
export default App
```



Pasos a seguir:

- **Crea un nuevo proyecto React:** Si aún no tienes un proyecto, puedes usar Create React App
 - ❖ `npx create-react-app my-app` (Antiguo/Deprecado)
 - Navega al directorio del proyecto: `cd my-app`
 - ❖ `npm create vite@latest` (Actual)
 - Selecciona React como método de creación del proyecto
 - Digita el nombre del proyecto `my-app`
- **Ubicarse en el directorio del proyecto:** `cd my-app`
- **Instala las dependencias:** Ejecuta el comando de instalación correspondiente (según el método elegido).
- **Importa los componentes:** Importa los componentes de Bootstrap en tus componentes React.
- **Utiliza los componentes:** Utiliza los componentes de Bootstrap en tu JSX.

Esto permite que los componentes sean optimizados para React lo cual facilita la integración con otras librerías debido a su amplia documentación y comunidad. Por lo tanto, existe una mayor flexibilidad en la personalización de estilos. Tanto React Bootstrap como Bootstrap ofrecen mecanismos para personalizar los estilos. Ambos métodos de instalación son recomendados, pero tener en cuenta que con React Bootstrap no se encuentra actualizado en la última versión de Bootstrap y es por ese motivo que hemos preferido la instalación manual del mismo ya que permite que el marco de trabajo se realice con las últimas versiones estables de la herramienta.

La elección del método de integración dependerá de las necesidades específicas del proyecto y del nivel de personalización requerido. React Bootstrap ofrece una integración más sencilla y optimizada pero desactualizada, mientras que Bootstrap directo proporciona mayor flexibilidad. En ambos casos, Bootstrap aporta un sólido conjunto de componentes y utilidades para crear interfaces de usuario atractivas y responsivas en aplicaciones React.



Autoevaluación 3

1. ¿Qué significa las siglas CSS?
 - a. Hojas de Estilo en Cascada
 - b. La iteración es un proceso clave
 - c. posicionamiento en la pantalla del elemento
 - d. Es el esqueleto de una página

2. ¿Indique las palabras a completa el enunciado “HTML no es un....., sino un que estructura el contenido en la web.?”
 - a. Proceso estructurado/Bootstrap
 - b. CSS/JavaScript
 - c. Lenguajes de marca de Hipertexto/Hojas de estilo en cascada
 - d. Lenguaje de desarrollo/React
 - e. Lenguaje de programación / lenguaje de marcado

3. ¿Qué es la validación de formularios?
 - a. No verifica obligatoriedad en los campos vacíos
 - b. Aspecto no tan fundamental cuando se crea una página web que capturará información, y más aún si esta información será almacenada en una base de datos.
 - c. Es el proceso de verificar que los datos ingresados por un usuario en un formulario web sean correctos y cumplan con ciertos criterios establecidos.
 - d. Ninguna

4. ¿Indique cuál de los siguientes enunciados son características/reglas de la validación de datos?
 - a. Requeridos (required)
 - b. Formato (Type)
 - c. Longitud (maxlength, minlength)
 - d. Patrones (Pattern)
 - e. Todas las anteriores





5. ¿Indique cuál de los siguientes enunciados son características de CSS?
 - a. Diseño/Personalización
 - b. Estructuras/Recopilados.
 - c. Interfaz/Modelado
 - d. Personalización/Responsivo

6. ¿Indique cuáles son las principales características de JavaScript?
 - a. Rápido / DOM
 - b. Multiplataforma/Asíncrono
 - c. Versátil / Interpretado
 - d. Atributos / Diseño
 - e. Personalización/Interfaces

7. ¿Qué es Bootstrap?
 - a. Es un framework de desarrollo web frontend de código abierto altamente popular, diseñado para acelerar y simplificar el proceso de creación de interfaces de usuario responsivas y visualmente atractivas.
 - b. Bootstrap es una gran herramienta, a pesar de todo, pues hay una serie de escenarios en los que Bootstrap puede ser de gran ayuda. Como cuando tienes que construir un prototipo extremadamente rápido (por ejemplo, en un día). O si vas a crear una aplicación donde la interfaz no tiene importancia (Por ejemplo, una aplicación para uso interno).
 - c. Bootstrap proporciona un conjunto de componentes y plantillas CSS, HTML y JavaScript que cualquiera puede utilizar o modificar de manera paga/suscrita.
 - d. Es una interfaz de desarrollo web que provee código abierto altamente popular, diseñado para cambiar el proceso de creación de interfaces de usuario responsivas y visualmente atractivas.
 - e. Bootstrap es una gran herramienta de prototipado, para construir un prototipo extremadamente rápido (por ejemplo, en un día). O si vas a crear una aplicación donde la interfaz no tiene importancia (Por ejemplo, una aplicación para uso interno).





8. ¿Quién creo Bootstrap?
 - a. Mark Otto y Jacob Thornton
 - b. Mark Griffin y James Thornton
 - c. Mark Zuckerberg y Jacob Norton
 - d. Tim Berners-Lee y Jacob Marley

9. ¿Indique cuál de los siguientes enunciados representan las características de Bootstrap?
 - a. Responsividad integrada
 - b. Modularidad
 - c. Extensibilidad
 - d. Rendimiento optimizado
 - e. Todas las Anteriores

10. ¿Indique si los siguientes enunciados son correctos para la instalación de Bootstrap en React?
 - a. `npm i --save bootstrap popperjs`
 - b. `npm r --save bootstrap @popperjs/core`
 - c. `npm installer--save bootstrap @popperjs/core`
 - d. `import 'bootstrap/dist/css/bootstrap.min.css';`
 - e. `import 'bootstrap/dist/css;`
 - f. `import 'bootstrap.min.css';`
 - g. `npm i --save bootstrap @popperjs/core`

Resumen de la Unidad 3

En esta unidad se trataron los siguientes puntos a profundidad y aquí se va a proporcionar un pequeño detalle de estos: HTML, CSS, JavaScript y Bootstrap: Fundamentos del Desarrollo Web Frontend donde estas tecnologías son pilares fundamentales en el desarrollo de interfaces de usuario web (frontend). Cada una cumple un rol específico y se complementan para crear experiencias de usuario dinámicas y visualmente atractivas.



HTML (HyperText Markup Language): Es el lenguaje estándar para estructurar el contenido de una página web. Define la semántica y la organización de los elementos en un documento, utilizando etiquetas como encabezados (<h1>, <h2>), párrafos (<p>), imágenes (), enlaces (<a>), y muchas otras. HTML proporciona la base sobre la cual se construyen las páginas web.

CSS (Cascading Style Sheets): Se utiliza para aplicar estilos a los elementos HTML, controlando su apariencia visual. Define propiedades como color, fuente, tamaño, espaciado, diseño de la página, y disposición de los elementos en la pantalla. CSS permite separar la presentación del contenido, haciendo que las páginas web sean más mantenibles y escalables.

JavaScript: Es un lenguaje de programación interpretado que se utiliza para agregar interactividad a las páginas web. Permite manipular el DOM (Document Object Model), responder a eventos del usuario (como clics, desplazamientos), realizar cálculos, y comunicarse con servidores. JavaScript es esencial para crear aplicaciones web dinámicas y ricas en características.

Bootstrap: Es un framework de desarrollo front-end que proporciona una colección de herramientas y utilidades para crear sitios web responsivos y modernos de forma rápida y eficiente. Incluye un sistema de cuadrícula flexible, componentes prediseñados (como botones, formularios, navegación), y clases CSS para estilos comunes. Bootstrap acelera el proceso de desarrollo al proporcionar una base sólida y un conjunto de herramientas reutilizables.

En resumen:

- HTML define la estructura del contenido.
- CSS define la presentación visual.
- JavaScript agrega interactividad y comportamiento dinámico.
- Bootstrap es una herramienta que facilita la creación de interfaces de usuario atractivas y responsivas.

Relación entre las tecnologías:

- HTML proporciona la estructura base.
- CSS se aplica a los elementos HTML para darles estilo.
- JavaScript interactúa con los elementos HTML y CSS para crear comportamientos dinámicos.
- Bootstrap proporciona componentes prediseñados y un sistema de cuadrícula para acelerar el desarrollo con HTML y CSS.

En conjunto, estas tecnologías permiten crear sitios web complejos y sofisticados, desde simples páginas estáticas hasta aplicaciones web interactivas.

9. ANEXOS

Solucionario 1.-

Tabla 1: Solucionario unidad 1

# pregunta	Respuesta	Retroalimentación
1	b	La Iteración es un proceso clave para el diseño de interfaces ya que en cada paso vamos a mitigar la resistencia al cambio.
2	c	Figma y Sketch proporcionan herramientas para el prototipado de interfaces.
3	b	Lo mínimo requerido serian 3 iteraciones para obtener cambios y puntos de vista del cliente para mejorar, así como parte de nuestro asesoramiento.
4	e	Figma cuenta con un modelado multiplataforma
5	a	Es el manejo de datos desde los elementos de un formulario
6	a	Son herramientas que cuentan con una manera fácil de manejar netamente intuitivas con dar clic o arrastrar ya podemos estar utilizándolas.
7	a	Apple – Macintosh fueron de las pioneras en realizar interfaces agradables e intuitivas para el uso ya sea del ordenador o móvil.
8	d	Tim Berners Lee padre fundador de la World Wide Web
9	d	Esta herramienta es muy útil pero solo se puede utilizar a nivel de sistema operativa MacOS
10	f	Se centra en las colaboraciones para que se pueda familiarizar con el trabajo en equipo.



Solucionario 2.-

Tabla 2: Solucionario unidad 2

# pregunta	Respuesta	Retroalimentación
1	c	Experiencia de Usuario / Interfaz de Usuario
2	a	Se refiere a cómo una persona se siente al interactuar con un sistema o producto. Esto incluye la usabilidad, eficiencia, y satisfacción que el usuario experimenta al utilizar un sitio web, aplicación o cualquier producto digital.
3	c	Garantizar que los usuarios tengan una interacción efectiva y agradable dentro del desarrollo de un producto digital
4	a	Visibilidad y Detectabilidad.
5	a	Un buen diseño de UX mejora la satisfacción del cliente, los productos digitales bien diseñados permiten a los usuarios completar tareas de manera más rápida y eficiente y esto a su vez se traduce en una mejor manera de identificar problemas de usabilidad en etapas tempranas del desarrollo puede reducir significativamente los costos de desarrollo y mantenimiento del software.
6	b	Atención / Interés / Deseo /Acción
7	d	Es guiar a los usuarios hacia una acción específica que los diseñadores o las empresas desean que realicen.
8	a	Donald Norman - Es porque quería abarcar todos los aspectos de la interacción de una persona con una compañía, sus servicios y productos.
9	e	Son todas las Anteriores
10	a	Se centra en el diseño visual de la aplicación y la manera en que se presentan los elementos gráficos y textuales al usuario

Solucionario 3.-

Tabla 3: Solucionario unidad 3

# pregunta	Respuesta	Retroalimentación
1	a	Hojas de Estilo en Cascada
2	e	Lenguaje de programación / lenguaje de marcado
3	c	Es el proceso de verificar que los datos ingresados por un usuario en un formulario web sean correctos y cumplan con ciertos criterios establecidos.
4	e	Todas las Anteriores son parte de las características y reglas de validación de los formularios.
5	d	El modo responsivo y la personalización son características fundamentales para la elaboración de hojas de estilo CSS
6	b	Unas de las principales características de JS o JavaScript es q permite trabajar de forma asíncrona y podemos utilizarlo de modo multiplataforma, es decir en cualquier S. O
7	a	Es un framework de desarrollo web frontend de código abierto altamente popular, diseñado para acelerar y simplificar el proceso de creación de interfaces de usuario responsivas y visualmente atractivas.
8	a	Mark Otto y Jacob Thornton
9	e	Son todas las Anteriores
10	g	<code>npm i --save bootstrap @popperjs/core</code> es el comando de instalación por el terminal de Bootstrap en React.

Referencias

Caddick, R., & Cable, S. (2011). *Communicating the user experience: A practical guide for creating useful UX documentation*. John Wiley & Sons.

Carroll, J. M. (2003). *HCI models, theories, and frameworks: Toward a multidisciplinary science*. Morgan Kaufmann. doi:<https://doi.org/10.1016/B978-1-55860-808-5.X5000-X>

Chaffey, D., & Ellis-Chadwick, F. (2019). *Digital marketing: Strategy, implementation and practice*. Pearson.

Cialdini, R. B. (2006). *Influence: The psychology of persuasion*. . Harper Business.

Deramond, J., Mark, O., & Thornton, J. (14 de 08 de 2024). *Bootstrap*. Obtenido de Bootstrap: <https://getbootstrap.com/>

Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. California: John Wiley & Sons.

Fogg, B. J. (2003). *Persuasive technology: Using computers to change what we think and do*. Morgan Kaufmann.

Garret, J. J. (2011). *The Elements of User Experience: User-Centered Design for the Web and Beyond*. . California, San Francisco: New Riders.

Kotler, P., & Armstrong, G. (2018). *Principles of marketing*. Pearson.

Krug, S. (2014). *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. California, San Diego: New Riders.

Morville, P., & Rosenfeld, L. (2006). *Information architecture for the World Wide Web: Designing large-scale web sites*. California: O'Reilly Media.

Mozilla FireFox. (12 de 08 de 2024). *Mozilla Developer Network*. Obtenido de Mozilla Developer Network: Mozilla Developer Network

Nielsen, J. (2012). *Usability 101: Introduction to Usability*. Nielsen Norman Group, 1.

Norman, D. (2013). *The Design of Everyday Things*. California, San Diego: Basic Books. doi:9780465050659

Norman, D. A. (1988). *The psychology of everyday things*. . California San Diego: Basic Books.

Norman, D. A. (2004). *Emotional design: Why we love (or hate) everyday things*. California, San Diego: Basic Books.





Robbins, J. N. (2018). Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. . O'Reilly Media.

Thatcher, J., Burks, M. R., Heilmann, C., Henry, S. L., Kirkpatrick, A., Lauke, P. H., & Waddell, C. D. (2006). Web accessibility: Web standards and regulatory compliance. Friends of Ed. friends of ED. doi:9781590596388

