



UNIVERSIDAD TECNOLÓGICA DE TEHUACÁN

Nombre del Alumno:

Manelic Alitzel Reyes Torres

Alejandro Contreras Martinez

Emmanuel Castro Salvador

**Carlos Eduardo Hernandez
Camacho**

**Danna Paola Tobon
Mosqueda**

Docente:

José Miguel Carrera Pacheco

**ING EN DESARROLLO DE
SOFTWARE
MULTIPLATAFORMA**

Diferencia entre Página Web vs. Aplicación Web

Danna Paola Tobón Mosqueda.

Aunque ambas corren en un navegador, sus propósitos y arquitecturas son radicalmente distintos.

Característica	Página Web (Sitio Web)	Aplicación Web (Web App)
Objetivo Principal	Informar (Consumo de contenido).	Realizar tareas (Interacción y creación).
Interacción	Unidireccional (El usuario lee/ve).	Bidireccional (El usuario manipula datos).
Complejidad	Baja (HTML/CSS, poco JS).	Alta (Lógica de negocio compleja, bases de datos).
Estado	Generalmente "Stateless" (no guarda sesión compleja).	Mantiene estado (Sesiones, carritos, perfiles).
Ejemplos	Blogs, Portafolios, Landing Pages, Sitios de noticias.	Gmail, Trello, Facebook, Banca en línea.

Manelic Alitzel Reyes Torres

Desde el liderazgo técnico, debemos distinguir si estamos construyendo un vehículo de información o un motor de transacciones.

Característica	Página Web (Sitio Web)	Aplicación Web (Web App)
Objetivo	Consumo de contenido e información.	Ejecución de tareas y creación de datos.
Interacción	Unidireccional (Lectura/Visualización).	Bidireccional (Manipulación de datos).
Complejidad	Baja; stack básico de HTML/CSS.	Alta; lógica de negocio y bases de datos.
Gestión de Estado	Generalmente "Stateless".	Mantiene sesiones, carritos y perfiles.

Casos de Uso	Blogs, Portafolios, Landing Pages.	Gmail, Facebook, Banca en línea.
--------------	------------------------------------	----------------------------------

Alejandro Contreras Martinez

- Página web: Una Página Web es un documento digital que se muestra en un navegador web, y están compuestas por una combinación de texto, imágenes, videos y otros recursos multimedia, así como por código informático que define la estructura, el diseño y la funcionalidad de la página
- Aplicación web: Una aplicación web es un software que se ejecuta en el navegador web. Las empresas tienen que intercambiar información y proporcionar servicios de forma remota. Utilizan aplicaciones web para comunicarse con los clientes cuando lo necesiten y de una forma segura. Las funciones más comunes de los sitios web, como los carros de compra, la búsqueda y el filtrado de productos, la mensajería instantánea y los canales de noticias de las redes sociales, tienen el mismo diseño que las aplicaciones web.

Le permiten acceder a funcionalidades complejas sin la necesidad de instalar o configurar un software. las páginas web son estáticas, no hay cambios frecuentes en ellas. Con la web app sucede todo lo contrario, están constantemente actualizándose por lo que el contenido es más novedoso, además son más dinámicas.

Carlos Eduardo Hernández Camacho.

Una página web tiene como objetivo principal informar, mostrando contenido mayormente estático y con poca interacción por parte del usuario.

Una aplicación web permite la interacción constante con el sistema, manejo de usuarios autenticados, estados dinámicos, flujos complejos y persistencia de datos.

El proyecto SGIM se clasifica como una aplicación web porque:

- Maneja autenticación de usuarios
- Utiliza roles (reportante y encargado)
- Permite crear, consultar y modificar información
- Administra estados de los reportes
- Requiere flujos de navegación complejos

Emmanuel Castro Salvador

Para el SGIM, se ha determinado que el desarrollo debe ser una Aplicación Web profesional.

A diferencia de una página web (cuyo propósito es principalmente informativo y estático),

nuestra solución requiere:

- Gestión de Estados Dinámicos: El sistema debe rastrear el ciclo de vida de una incidencia
- (Pendiente -> En Proceso -> Resuelto).
- Interactividad Compleja: Los usuarios no solo consumen información, sino que generan
- datos (reportes, carga de imágenes) que activan procesos en el servidor.
- Autenticación y Autorización: Implementación de niveles de acceso estrictos donde un
- alumno solo puede reportar y ver sus reportes, mientras que un administrador tiene
- permisos de edición y asignación

Ejemplos Reales de Aplicaciones Web Profesionales

Danna Paola Tobón Mosqueda

- Figma: Una herramienta de diseño vectorial compleja que corre en el navegador. Reemplazó software de escritorio instalable.
- Airbnb: Gestiona reservas, pagos, mensajería en tiempo real, mapas y perfiles de usuario. Conecta oferta y demanda (Marketplace).
- Slack/Discord (Versión Web): Comunicación en tiempo real usando WebSockets. No requiere refrescar la página para recibir mensajes.

Manelic Alitzel Reyes Torres

- Figma: Sustituye al software de escritorio con diseño vectorial complejo en el navegador.
- Airbnb: Un marketplace que gestiona pagos, geolocalización y mensajería en tiempo real.
- Slack/Discord: Implementación de WebSockets para comunicación persistente sin refresco de página.

Alejandro Contreras Martinez.

Algunos ejemplos de páginas web profesionales:

- Jira: es una herramienta de software de Atlassian para la gestión ágil de proyectos y seguimiento de tareas/incidencias,
- Google Drive: es un servicio de almacenamiento en la nube que te permite guardar, organizar, compartir y acceder a archivos
- shopify: es una plataforma de comercio electrónico (e-commerce) todo en uno que permite a emprendedores y empresas crear, gestionar y hacer crecer una tienda online de forma sencilla, sin necesidad de conocimientos técnicos avanzados.

Carlos Eduardo Hernández Camacho

Durante la investigación se analizaron plataformas reales de gestión de incidencias:

- Zendesk: Sistema profesional de tickets con seguimiento y control por estados.
- Freshdesk: Plataforma de soporte con roles diferenciados y panel administrativo.

Estas aplicaciones resuelven problemas similares al SGIM y utilizan:

- Formularios claros
- Navegación estructurada
- Paneles según rol
- Seguimiento del estado de cada incidencia

Emmanuel Castro Salvador

- Jira Service Management: Investigué su flujo de tickets. Aunque es extremadamente robusto, noté que para una institución educativa es demasiado complejo. De aquí extraje la necesidad de tener un historial de reportes claro, pero decidí que nuestra interfaz debe ser mucho más directa para reducir el tiempo de reporte del alumno.
- FixMyStreet: Analicé cómo esta plataforma ciudadana gestiona reportes de baches o luces fundidas. Me inspiré en su flujo de "evidencia visual" y transparencia. Gracias a este análisis, determinamos que el SGIM debe permitir subir una fotografía obligatoria para evitar reportes falsos o poco claros, optimizando así el tiempo del técnico de mantenimiento

¿Qué problemas se resuelven con software?

Danna Paola Tobón Mosqueda.

- En la industria, el software es una herramienta para eliminar fricción. Generalmente se resuelven problemas en estas categorías:
- Ineficiencia / Automatización: Tareas manuales repetitivas que consumen tiempo humano
- Conectividad / Comunicación: Barreras geográficas o de tiempo

Manelic Alitzel Reyes Torres

- Acceso a Información / Transparencia: Datos que están ocultos o son difíciles de encontrar
- Gestión de Transacciones: Seguridad y confianza en intercambios de dinero o bienes

Alejandro Contreras Martínez.

- Repetición (se hace muchas veces).
- Datos (se guardan, se consultan, se cruzan).

- Reglas (validaciones, estados, permisos).

Carlos Eduardo Hernández Camacho

- Escala (más usuarios, más procesos).
- Necesidad de trazabilidad (historial, auditoría).
- Coordinación (varias personas/roles).

Arquitectura General de Aplicaciones Web

Danna Paola Tobón Mosqueda.

Para construir una base sólida y escalable, se deben entender las tres capas principales:

A. Frontend (El Cliente)

- Es lo que el usuario ve y toca. En una aplicación moderna, el Frontend no solo "muestra cosas", sino que gestiona lógica de presentación.
- Tecnologías: React, Vue, Angular (Frameworks), HTML5, CSS3.
- Responsabilidad: UX/UI, accesibilidad, llamadas al servidor (API requests)

B. Backend (El Servidor)

- El cerebro de la operación. El usuario nunca lo ve, pero sin esto, la aplicación no funciona.
- Lógica de Negocio: Reglas del sistema (ej. "No permitir comprar si no hay saldo").
- Base de Datos: Donde persiste la información (SQL como PostgreSQL, o NoSQL como MongoDB).
- API (Application Programming Interface): El puente que permite al Frontend pedirle datos al Backend.

C. Infraestructura / Entornos

- Dónde vive el código. No es suficiente con que funcione en "mi computadora".
- Entorno de Desarrollo: Tu computadora local.
- Entorno de Staging (Pruebas): Un espejo del real para probar antes de lanzar.
- Entorno de Producción: Lo que usan los clientes reales.
- CI/CD (Integración y Despliegue Continuo): Automatización para pasar el código de tu PC a producción sin romper nada.

Análisis de Plataformas Similares

Ejemplo A: Zendesk / Jira Service Management

- Son los líderes mundiales en gestión de tickets para empresas.

- Lo bueno: Flujos de trabajo potentes (Ej: "Si el reporte es de 'Infraestructura', manda correo automático a Mantenimiento").
- Lo malo: Son demasiado complejos, caros y se sienten "corporativos". No están diseñados para que un alumno reporte algo en 30 segundos desde su celular en el pasillo.

Ejemplo B: Aplicaciones de Reporte Ciudadano (Ej. "SeeClickFix")

- Apps que usan los ayuntamientos para que ciudadanos reporten baches o luminarias rotas.
- Lo bueno: Uso excelente de geolocalización y subida de fotos.
- Lo malo: Generalmente la información es pública.
- El riesgo para el proyecto: Un reporte de "acoso" o "mal trato de un profesor" NO puede tratarse igual que una "silla rota".
- Lección: se necesitan Niveles de Privacidad.

Manelic Alitzel Reyes Torres

Para que nuestro software sea escalable, dividimos las responsabilidades en tres pilares fundamentales:

A. Frontend (El Cliente) No es solo "maquetación"; es la gestión de la lógica de presentación.

- Tecnologías: React, Vue, Angular, HTML5, CSS3.
- Foco: UX/UI, accesibilidad y consumo eficiente de APIs.

B. Backend (El Servidor) Es el cerebro donde reside la integridad del sistema.

- Lógica de Negocio: Definición de reglas (ej. validación de saldos o permisos).
- Persistencia: Bases de datos SQL (PostgreSQL) o NoSQL (MongoDB).
- Interfaz: La API actúa como el puente de comunicación con el cliente.

C. Infraestructura y Ciclo de Vida El código debe ser portable y seguro a través de diferentes entornos:

1. Desarrollo: Entorno local del ingeniero.
 2. Staging: Espejo de producción para control de calidad (QA).
 3. Producción: Entorno final servido al cliente.
- CI/CD: Automatización para garantizar despliegues continuos sin interrupciones de servicio.
3. Análisis Competitivo y Lecciones Aprendidas (Benchmarking)

Al analizar plataformas similares, identificamos riesgos y oportunidades críticas para nuestro desarrollo:

- Zendesk / Jira: Aunque potentes en automatización, fallan en la experiencia de usuario para contextos rápidos (movilidad) debido a su excesiva complejidad corporativa.
- SeeClickFix (Reporte Ciudadano): Excelentes en geolocalización, pero su modelo de información pública es un riesgo de seguridad para nosotros.

Alejandro Contreras Martinez:

La arquitectura general de aplicaciones web es el plano fundamental que organiza y define la interacción entre los componentes de un sistema en línea, dividiéndose principalmente en Front-end (cliente) (interfaz de usuario en el navegador), Back end (servidor) (lógica de negocio, bases de datos) y la Base de Datos, usando patrones como capas o microservicios para asegurar escalabilidad, rendimiento y mantenimiento, gestionando la comunicación entre el usuario y el servidor.

- Frontend: es la parte de una aplicación o sitio web con la que el usuario interactúa directamente, es decir, todo lo visual y funcional que se muestra en el navegador, como botones, textos, imágenes, menús y formularios, asegurando una experiencia de usuario (UX) intuitiva y agradable.
- Backend: es la parte "detrás de escena" de una aplicación o sitio web, la lógica y la infraestructura que los usuarios no ven pero que hacen funcionar todo, incluyendo el servidor, la base de datos y las aplicaciones, encargándose del procesamiento de datos, la seguridad y la comunicación para entregar la información al frontend (la parte visual). Se enfoca en la lógica del negocio, la gestión de datos y la comunicación con el servidor
- Infraestructura / entornos: se refieren al conjunto de componentes físicos, virtuales y de software que permiten el funcionamiento, desarrollo, pruebas y despliegue de una aplicación web. Es la base tecnológica oculta detrás de la interfaz que ve el usuario, garantizando que el sitio sea rápido, seguro y confiable.

Análisis de 2 plataformas reales similares a la idea del equipo

- Zendesk: es una plataforma de software basada en la nube que centraliza la atención al cliente, permitiendo a las empresas gestionar todas las interacciones
- Jira Service Management: es una plataforma enfocada sobre todo a los equipos de TI, desarrollo y negocios a través de gestionar solicitudes ya sea por incidentes, problemas o cambios y acelerar la entrega de valor a través de un portal de autoservicio, automatización y flujos de trabajo colaborativo.

Emmanuel Castro Salvador

El equipo opta por una arquitectura Monolítica Modular que asegura la escalabilidad desde el día uno:

- Frontend: Una interfaz reactiva diseñada para la eficiencia operativa, priorizando la velocidad de carga en dispositivos móviles y de escritorio.
- Backend: Una API robusta que centraliza la lógica de negocio, validaciones de seguridad y la comunicación con la base de datos.
- Infraestructura: Uso de contenedores para garantizar que el entorno de desarrollo sea idéntico al de producción, eliminando fallos por diferencias de sistema operativo

Investiga conceptos clave para diseñar sistemas usables y profesionales:

- Arquitectura de información es la disciplina que organiza, estructura y etiqueta el contenido de productos digitales (sitios web, apps) para que los usuarios puedan encontrar, entender y navegar la información de manera intuitiva y eficiente, mejorando la usabilidad y la experiencia general. Se centra en la claridad de la navegación, categorías y sistemas de búsqueda, asegurando que la jerarquía del contenido tenga sentido para el usuario final, a diferencia de la arquitectura de datos que se enfoca en la infraestructura técnica.
- Jerarquías de contenido: es la organización estratégica de la información en una página web o documento, priorizando los elementos más importantes para facilitar la comprensión y navegación del usuario, usando técnicas visuales como tamaños de fuente (H1, H2, H3, etc.), colores, negritas y espaciado para guiar la atención desde lo general a lo específico, y es crucial para el SEO
- Patrones de navegación web: Los patrones de navegación web son soluciones reutilizables y probadas para organizar elementos y guiar a los usuarios de manera intuitiva a través de un sitio o aplicación, mejorando la experiencia (UX) y la usabilidad, como el botón de hamburguesa, las migas de pan (breadcrumbs) o el scroll infinito, resolviendo problemas comunes de diseño de forma eficiente y consistente.

Orden de tabulación se refiere a la secuencia en la que los elementos interactivos (como campos de formulario, enlaces) son enfocados al presionar la tecla Tab en una página web, formulario o documento, y es crucial para la accesibilidad, permitiendo una navegación lógica de izquierda a derecha y de arriba a abajo, siguiendo el orden visual del contenido para usuarios de teclado y lectores de pantalla.

- Navegación por teclado. La navegación por teclado es el uso exclusivo de teclas (como Tab, flechas, Intro, Espacio) para interactuar, moverse y ejecutar comandos en un software o sitio web, sin necesidad del ratón; es una función clave de accesibilidad para personas con discapacidades motoras o visuales, permitiéndoles acceder a todos

los elementos enfocables (enlaces, botones, formularios) y moverse entre ellos de forma secuencial o específica.

- Accesibilidad sin mouse La accesibilidad sin mouse se refiere a un conjunto de funciones en sistemas operativos y aplicaciones que permiten controlar la computadora completamente usando solo el teclado, comandos de voz, o dispositivos de seguimiento (como ojos o rostro) para compensar la falta de un ratón físico