

**UNIVERSIDAD TECNOLÓGICA DE  
TEHUACÁN**

**NOMBRE DEL ALUMNO:  
OSCAR FLORES CERQUEDA**

**MATERIA  
DWP-8A**

**MAESTRO  
José Miguel Carrera Pacheco**

**Título  
Investigacion individual**

## **1. Diferencia entre Página Web y Aplicación Web**

### **1.1 Página Web**

Una página web es un documento digital que proporciona información al usuario de manera principalmente estática. Sus características principales son:

#### **Características:**

- Contenido principalmente estático o informativo
- Construida con HTML, CSS y JavaScript básico
- Interactividad limitada (formularios básicos, enlaces)
- Objetivo principal: informar y presentar contenido
- Actualización menos frecuente del contenido
- No requiere autenticación de usuario generalmente
- Menor complejidad técnica

#### **Ejemplos típicos:**

- Blogs personales o corporativos
- Portafolios profesionales
- Páginas institucionales
- Sitios informativos de empresas

### **1.2 Aplicación Web**

Una aplicación web es un software que se ejecuta en un navegador web, permitiendo a los usuarios realizar tareas específicas y complejas. Sus características principales son:

#### **Características:**

- Altamente interactiva y dinámica
- Requiere tecnologías más complejas (frameworks, bases de datos)
- Procesamiento del lado del servidor y del cliente
- Objetivo principal: permitir que los usuarios realicen acciones
- Actualización en tiempo real
- Generalmente requiere autenticación de usuario
- Mayor complejidad técnica

## Diferencias Clave:

Aspecto	Página Web	Aplicación Web
Propósito	Informar	Realizar tareas
Interactividad	Limitada	Alta
Complejidad	Baja	Alta
Tecnologías	HTML, CSS, JS básico	Frameworks, bases de datos, APIs
Actualización	Manual, menos frecuente	Automática, en tiempo real
Acceso	Generalmente público	Requiere credenciales
Desarrollo	Más rápido y sencillo	Más tiempo y recursos

## 2. Ejemplos Reales de Aplicaciones Web Profesionales

### 2.1 Aplicaciones de Productividad

#### Google Docs

- Procesador de texto basado en la nube
- Colaboración en tiempo real entre múltiples usuarios
- Guardado automático en Google Drive
- Sincronización entre dispositivos
- Millones de usuarios activos diariamente

#### Trello

- Gestión de proyectos basada en metodología Kanban
- Organización visual de tareas
- Colaboración en equipos
- Integración con múltiples servicios

#### Notion

- Plataforma todo-en-uno para organización
- Combina notas, bases de datos y calendarios
- Ideal para planificación de contenidos
- Colaboración entre equipos

### 2.2 Aplicaciones de Comunicación y Colaboración

#### Gmail

- Servicio de correo electrónico
- Organización automática de mensajes
- Integración con Google Workspace

- Aplicación de una sola página (SPA)

#### **Slack** (mencionado en contexto empresarial)

- Comunicación empresarial
- Canales de equipo
- Integración con herramientas de productividad

### **2.3 Aplicaciones de Diseño y Creatividad**

#### **Canva**

- Herramienta de diseño gráfico
- Interfaz de arrastrar y soltar
- Procesa 150 diseños por segundo globalmente
- 125 millones de usuarios activos mensuales
- Biblioteca de plantillas prediseñadas
- ROI promedio de 380% para empresas

### **2.4 Aplicaciones Educativas**

#### **Coursera**

- Plataforma educativa en línea
- Cursos de universidades reconocidas
- Uso de CDN para optimización de streaming
- Almacenamiento en caché de videos y materiales

### **2.5 Aplicaciones de Entretenimiento**

#### **Spotify**

- Streaming de música
- Personalización basada en IA
- Recomendaciones según comportamiento del usuario
- Aprendizaje automático para engagement

#### **YouTube**

- Plataforma de videos
- Millones de usuarios simultáneos
- Procesamiento en tiempo real

### **2.6 Aplicaciones de Comercio Electrónico**

#### **Amazon**

- Comercio electrónico global
- Recomendaciones personalizadas

- Gestión de inventario en tiempo real
- Procesamiento de pagos

### **Shopify / WooCommerce**

- Plataformas de tiendas online
- Gestión de productos y carrito
- Procesamiento de transacciones

## **2.7 Aplicaciones de Traducción e Idiomas**

### **Google Traductor**

- Traducción a más de 100 idiomas
- Procesa 100 mil millones de palabras diariamente
- 94% de precisión en idiomas principales
- Traducción de páginas web completas

## **2.8 Aplicaciones Empresariales**

### **HubSpot**

- Automatización de marketing
- CRM integrado
- Gestión de contactos y campañas
- Análisis en tiempo real

## **3. Problemas que Resuelve el Software**

### **3.1 Problemas Operativos y de Gestión**

#### **Gestión de Inventario**

- Control preciso de stock
- Prevención de sobre-almacenamiento
- Evita faltantes de productos
- Optimiza el flujo de caja
- Reduce pérdidas por obsolescencia
- Planificación basada en demanda

#### **Gestión de Información**

- Centralización de datos
- Acceso en tiempo real
- Eliminación de duplicados
- Versionado controlado

- Compartición segura de información

### **Gestión de Producción**

- Planificación de materiales
- Control de procesos
- Seguimiento de rendimiento
- Comparación planificado vs. real
- Alertas de reabastecimiento

### **3.2 Problemas Financieros y Contables**

#### **Control Financiero**

- Estados financieros en tiempo real
- Registro contable automatizado
- Control de costos
- Gestión de precios
- Análisis de rentabilidad
- Facturación automatizada

#### **Reducción de Pérdidas**

- Identificación de desviaciones
- Control de mermas
- Detección de artículos obsoletos
- Prevención de robos

### **3.3 Problemas de Comunicación y Colaboración**

#### **Trabajo en Equipo**

- Colaboración en tiempo real
- Comunicación entre departamentos
- Compartición de documentos
- Sincronización de equipos

#### **Atención al Cliente**

- Gestión de tickets
- Seguimiento post-venta
- Historial de interacciones
- CRM integrado

### **3.4 Problemas Organizacionales**

#### **Estructura y Procesos**

- Sistematización de actividades
- Definición de roles y funciones
- Supervisión de estándares
- Automatización de tareas repetitivas

#### **Escalabilidad**

- Manejo de alto volumen de datos
- Crecimiento sin limitaciones físicas
- Distribución de carga
- Optimización de recursos

### **3.5 Problemas Tecnológicos**

#### **Lentitud del Sistema**

- Optimización de rendimiento
- Limpieza de archivos temporales
- Gestión de recursos

#### **Seguridad**

- Protección de datos
- Autenticación de usuarios
- Cifrado de información
- Prevención de ataques

#### **Compatibilidad**

- Funcionamiento multiplataforma
- Integración entre sistemas
- Actualización de versiones

### **3.6 Problemas de Accesibilidad**

#### **Disponibilidad**

- Acceso desde cualquier dispositivo
- Funcionamiento 24/7
- No requiere instalación
- Actualizaciones automáticas

#### **Ubicuidad**

- Trabajo remoto
- Acceso desde múltiples ubicaciones
- Sincronización entre dispositivos

## 4. Arquitectura General de Aplicaciones Web

### 4.1 Visión General

La arquitectura de aplicaciones web define cómo interactúan los diferentes componentes de una aplicación. Se estructura típicamente en capas que separan las responsabilidades y facilitan el mantenimiento y escalabilidad.

### 4.2 Frontend (Capa de Presentación)

**Definición:** El frontend es la parte visual y con la que interactúan directamente los usuarios. También se conoce como "lado del cliente" o interfaz de usuario (UI).

#### Componentes principales:

1. **Interfaz de Usuario (UI)**
  - Elementos visuales (botones, formularios, menús)
  - Diseño y layout
  - Gráficos e imágenes
  - Mensajes y notificaciones
2. **Experiencia de Usuario (UX)**
  - Flujos de navegación
  - Interactividad
  - Respuesta a acciones del usuario
  - Retroalimentación visual

#### Tecnologías principales:

- **HTML (HyperText Markup Language)**
  - Estructura del contenido
  - Definición de elementos de página
- **CSS (Cascading Style Sheets)**
  - Diseño visual
  - Layout y posicionamiento
  - Animaciones y transiciones
  - Diseño responsivo
- **JavaScript**
  - Interactividad
  - Manipulación del DOM
  - Validación de formularios
  - Comunicación con el backend (AJAX)

#### Frameworks y librerías modernas:

- **React**



- Componentes reutilizables
- Virtual DOM
- Ecosistema extenso
- **Angular**
  - Framework completo
  - TypeScript
  - Arquitectura MVC
- **Vue.js**
  - Progresivo y flexible
  - Fácil integración
  - Reactivo
- **Next.js**
  - Renderizado del lado del servidor
  - Optimización automática
  - Enrutamiento integrado

#### **Responsabilidades:**

- Renderizar la interfaz visual
- Capturar entradas del usuario
- Enviar solicitudes HTTP al backend
- Recibir y mostrar respuestas
- Gestionar el estado de la aplicación en el cliente
- Proporcionar retroalimentación inmediata

### **4.3 Backend (Capa de Lógica de Negocio)**

**Definición:** El backend es la parte "detrás de escena" donde se gestiona la lógica de negocio, el procesamiento de datos y la comunicación con bases de datos. También se conoce como "lado del servidor".

#### **Componentes principales:**

1. **Servidor de Aplicaciones**
  - Procesa solicitudes HTTP
  - Ejecuta la lógica de negocio
  - Gestiona autenticación y autorización
  - Coordina servicios
2. **Lógica de Negocio**
  - Reglas empresariales
  - Procesamiento de transacciones

- Validación de datos
- Cálculos y transformaciones

### 3. **APIs (Application Programming Interfaces)**

- Endpoints para comunicación
- REST APIs
- GraphQL
- WebSockets para tiempo real

### 4. **Middleware**

- Autenticación
- Validación
- Logging
- Manejo de errores
- Compresión de respuestas

## **Tecnologías principales:**

### **Lenguajes de programación:**

- **JavaScript/Node.js**
  - Asíncrono y orientado a eventos
  - NPM ecosistema
  - Frameworks: Express, NestJS
- **Python**
  - Sintaxis clara
  - Frameworks: Django, Flask, FastAPI
  - Excelente para IA/ML
- **Java**
  - Robusto y escalable
  - Frameworks: Spring Boot
  - Amplio uso empresarial
- **PHP**
  - Frameworks: Laravel, Symfony
  - WordPress, Drupal
- **Ruby**
  - Framework: Ruby on Rails
  - Convención sobre configuración
- **Go (Golang)**

- Alto rendimiento
- Concurrencia nativa
- Microservicios

#### **Responsabilidades:**

- Procesar solicitudes del cliente
- Ejecutar lógica de negocio
- Validar datos
- Gestionar seguridad
- Interactuar con bases de datos
- Coordinar microservicios
- Generar respuestas

#### **4.4 Base de Datos (Capa de Persistencia)**

**Definición:** La capa de datos es responsable del almacenamiento, recuperación y gestión de la información de la aplicación.

#### **Tipos de bases de datos:**

##### **1. Bases de Datos Relacionales (SQL)**

- **MySQL**
  - Código abierto
  - Ampliamente adoptado
  - Buen rendimiento
- **PostgreSQL**
  - Avanzadas capacidades
  - Extensible
  - Cumplimiento ACID
- **Microsoft SQL Server**
  - Integración con ecosistema Microsoft
  - Herramientas empresariales
- **Oracle**
  - Soluciones empresariales
  - Alta disponibilidad

##### **2. Bases de Datos NoSQL**

- **MongoDB**
  - Documentos JSON
  - Flexible y escalable

- Esquema dinámico
- **Redis**
  - Base de datos en memoria
  - Caché de alto rendimiento
  - Estructuras de datos avanzadas
- **Cassandra**
  - Distribuida
  - Alta disponibilidad
  - Escalabilidad lineal
- **Firebase/Firestore**
  - Tiempo real
  - Integración con servicios de Google
  - Backend as a Service

#### **Responsabilidades:**

- Almacenar datos persistentes
- Garantizar integridad de datos
- Optimizar consultas
- Realizar backups
- Gestionar transacciones
- Mantener relaciones entre datos

## **4.5 Infraestructura y Entornos**

**Definición:** La infraestructura comprende todos los recursos de hardware, software y servicios necesarios para alojar, ejecutar y mantener la aplicación web.

### **4.5.1 Componentes de Infraestructura**

#### **1. Servidores**

- **Servidores Web**
  - Nginx: Alto rendimiento, proxy inverso
  - Apache: Flexible, ampliamente usado
  - Caddy: Configuración automática HTTPS
- **Servidores de Aplicaciones**
  - Ejecutan el código del backend
  - Manejan lógica de negocio
  - Ejemplos: Tomcat, Uvicorn, Gunicorn

#### **2. Equilibradores de Carga (Load Balancers)**

- Distribuyen tráfico entre servidores
- Previenen sobrecarga
- Garantizan alta disponibilidad
- Ejemplos: HAProxy, AWS ELB, Google Cloud Load Balancing

### **3. CDN (Content Delivery Network)**

- Distribución de contenido estático
- Reducción de latencia
- Caché geográfica
- Ejemplos: Cloudflare, AWS CloudFront, Akamai

### **4. Caché**

- **Memcached**
  - Caché en memoria distribuida
  - Simple y rápido
- **Redis**
  - Estructura de datos avanzada
  - Persistencia opcional
- **Varnish**
  - Caché HTTP
  - Proxy inverso

#### **4.5.2 Modelos de Infraestructura**

##### **1. On-Premise (Infraestructura Local)**

- Servidores físicos propios
- Control total
- Mayor inversión inicial
- Mantenimiento propio

##### **2. IaaS (Infrastructure as a Service)**

- Infraestructura virtualizada
- Escalabilidad bajo demanda
- Ejemplos: AWS EC2, Google Compute Engine, Azure VMs

##### **3. PaaS (Platform as a Service)**

- Plataforma de desarrollo completa
- Gestión automática de infraestructura
- Ejemplos: Heroku, Google App Engine, AWS Elastic Beanstalk

#### **4. BaaS (Backend as a Service)**

- Backend preconfigurado
- Servicios listos para usar
- Ejemplos: Firebase, Back4app, Supabase

#### **5. CaaS (Containers as a Service)**

- Gestión de contenedores
- Orquestación automática
- Ejemplos: Kubernetes, Docker Swarm, AWS ECS

#### **6. Serverless (Función como Servicio)**

- Ejecución basada en eventos
- Pago por uso
- Ejemplos: AWS Lambda, Google Cloud Functions, Azure Functions

#### **4.5.3 Entornos de Desarrollo**

##### **1. Desarrollo (Development)**

- Entorno local del desarrollador
- Pruebas iniciales
- Iteración rápida
- Configuración flexible

##### **2. Pruebas (Testing/QA)**

- Validación de funcionalidades
- Pruebas de integración
- Pruebas de usuario
- Detección de bugs

##### **3. Staging (Pre-producción)**

- Réplica del entorno de producción
- Pruebas finales
- Validación de despliegues
- Entrenamiento de usuarios

##### **4. Producción (Production)**

- Entorno en vivo
- Usuarios reales
- Alta disponibilidad
- Monitoreo continuo

#### **4.5.4 Servicios de Cloud Computing**

##### **Amazon Web Services (AWS)**

- EC2 (servidores virtuales)
- RDS (bases de datos gestionadas)
- S3 (almacenamiento de objetos)
- Lambda (serverless)
- CloudFront (CDN)

##### **Google Cloud Platform (GCP)**

- Compute Engine
- Cloud SQL
- Cloud Storage
- Cloud Functions
- Cloud CDN

##### **Microsoft Azure**

- Virtual Machines
- Azure SQL Database
- Blob Storage
- Azure Functions
- Azure CDN

#### **4.5.5 DevOps e Integración Continua**

##### **Control de Versiones**

- Git
- GitHub, GitLab, Bitbucket

##### **CI/CD (Integración y Despliegue Continuo)**

- Jenkins
- GitHub Actions
- GitLab CI/CD
- CircleCI
- Travis CI

##### **Contenedores y Orquestación**

- Docker (contenedorización)
- Kubernetes (orquestación)
- Docker Compose (multi-contenedor local)

## **Monitoreo y Logging**

- Prometheus (métricas)
- Grafana (visualización)
- ELK Stack (Elasticsearch, Logstash, Kibana)
- New Relic
- Datadog

## **Gestión de Configuración**

- Ansible
- Terraform
- Chef
- Puppet

## **4.6 Comunicación entre Capas**

### **Protocolos y Tecnologías:**

#### **1. HTTP/HTTPS**

- Protocolo de transferencia
- Base de la comunicación web
- Métodos: GET, POST, PUT, DELETE, PATCH

#### **2. REST (Representational State Transfer)**

- Arquitectura de API
- Endpoints organizados por recursos
- Stateless (sin estado)
- Uso de métodos HTTP

#### **3. GraphQL**

- Lenguaje de consulta
- Cliente especifica datos exactos
- Reducción de over-fetching

#### **4. WebSockets**

- Comunicación bidireccional
- Tiempo real
- Mantiene conexión abierta

#### **5. gRPC**

- Protocol Buffers
- Alto rendimiento



- Microservicios

## **4.7 Patrones Arquitectónicos Modernos**

### **1. Arquitectura Monolítica**

- Una sola aplicación
- Todo en un mismo código base
- Despliegue completo
- Más simple para proyectos pequeños

### **2. Microservicios**

- Servicios independientes
- Responsabilidad única
- Escalabilidad individual
- Despliegue independiente

### **3. Backend for Frontend (BFF)**

- Backend dedicado por frontend
- Optimización específica
- Mejor rendimiento por plataforma

### **4. Serverless**

- Funciones independientes
- Ejecución basada en eventos
- Escalabilidad automática
- Pago por uso

### **5. JAMstack**

- JavaScript, APIs, Markup
- Pre-renderizado
- Desacoplamiento frontend/backend
- Alto rendimiento

## **4.8 Seguridad en la Arquitectura**

### **Capas de Seguridad:**

#### **1. Frontend**

- Validación de entrada
- Prevención XSS
- CORS configurado

#### **2. Backend**

- Autenticación (JWT, OAuth)
- Autorización basada en roles
- Validación de datos
- Prevención de inyección SQL
- Rate limiting

### 3. Base de Datos

- Cifrado en reposo
- Cifrado en tránsito
- Backups regulares
- Control de acceso

### 4. Infraestructura

- Firewalls
- SSL/TLS
- VPN
- Segmentación de red

## Arquitectura de información (AI)

Organiza y estructura el contenido para que las personas encuentren lo que buscan de manera rápida y lógica. Incluye clasificación, etiquetado, jerarquías y flujos de interacción.

### Jerarquías de contenido

Es la forma de priorizar y ordenar la información según su importancia. Ayuda a que el usuario sepa qué ver primero y qué es secundario, usando títulos, tamaño de texto, colores y posición.

### Patrones de navegación web

Modelos comunes de navegación que facilitan el uso porque el usuario ya los conoce (menú superior, menú lateral, breadcrumb, hamburguesa, pestañas).

### Orden de tabulación

Secuencia en la que los elementos interactivos reciben foco al presionar **TAB**. Un buen orden sigue la lógica visual de la página (arriba → abajo, izquierda → derecha).

### Navegación por teclado

Permitir que toda la interfaz sea usable solo con teclado: TAB para avanzar, SHIFT+TAB para retroceder, ENTER/SPACE para activar, flechas para menús.

### Accesibilidad sin mouse

Consiste en diseñar para personas que no pueden usar un mouse: teclados físicos, lectores de pantalla, navegadores de voz. Requiere foco visible, roles ARIA, semántica correcta y controles accesibles.

## Referencias

- Alboka Soft. (2024, octubre 2). *Aplicaciones web: qué son, tipos y ejemplos*. <https://albokasoft.com/adoramos-las-aplicaciones-web/>
- Amazon Web Services. (s.f.). *Front End frente a back-end: diferencia entre el desarrollo de aplicaciones*. <https://aws.amazon.com/es/compare/the-difference-between-frontend-and-backend/>
- Appleute. (2025, mayo 2). *Los problemas más comunes en el desarrollo de software*. <https://www.appleute.de/es/app-entwickler-bibliothek/problemas-habituales-desarrollo-software/>
- Aplyca. (s.f.). *Website vs. Web app*. <https://www.aplyca.com/blog/blog-website-vs-webapp-como-determinar-la-mejor-opcion-proyecto>
- AppMaster. (2023, marzo 24). *Arquitectura web back-end*. <https://appmaster.io/es/blog/arquitectura-web-back-end>
- Axarnet. (2023, diciembre 28). *Web Apps: qué son y qué tipos hay [+Ejemplos]*. <https://axarnet.es/blog/web-apps>
- Back4app. (2024, octubre 24). *Infraestructura de backend: tipos y cómo escoger la ideal*. <https://blog.back4app.com/es/que-es-una-infraestructura-backend/>
- Batec Digital. (2024, marzo 19). *Los 5 Problemas Informáticos Más Comunes en la Empresa y Cómo Solucionarlos*. <https://www.batecdigital.com/administracion-de-sistemas/los-5-problemas-informaticos-mas-comunes-en-la-empresa-y-como-solucionarlos/>
- Brandesign. (2022, noviembre 19). *10 ejemplos originales de aplicaciones webs a medida*. <https://brandesign.es/desarrollo-web/10-ejemplos-originales-de-aplicaciones-webs-a-medida/>
- Código en Casa. (2024, noviembre 18). *Arquitectura Backend for Frontend (BFF)*. <https://codigoencasa.com/arquitectura-backend-for-frontend-bff/>
- Cyberclick. (2025, noviembre 19). *¿Qué es una aplicación web y cuáles son las mejores para profesionales de marketing?* <https://www.cyberclick.es/numerical-blog/que-es-una-aplicacion-web-y-cuales-son-las-mejores-para-profesionales-de-marketing>
- Daslaf Dev. (s.f.). *Arquitectura CLEAN para el frontend*. <https://daslaf.dev/posts/arquitectura-clean-react>
- EDteam. (s.f.). *¿Cuáles son las diferencias entre página web, sitio web y aplicación web?* <https://ed.team/blog/cuales-son-las-diferencias-entre-pagina-web-sitio-web-y-aplicacion-web>
- Evolve IT. (2023, febrero 21). *¿Qué deficiencias podrán resolver con un software para empresas?* <https://www.evolve-it.com.mx/que-deficiencias-podran-resolver-con-un-software-para-empresas/>
- Guayahack. (s.f.). *Introducción a la arquitectura frontend-backend o cliente-servidor*. <https://guayahack.co/community/member/davidadarme/reto-arquitectura-frontend-backend-cliente-servidor/>
- Hostinger. (2025, enero 17). *Arquitectura de Aplicaciones Web: Desglosando una App Web*. <https://kinsta.com/es/blog/arquitectura-aplicaciones-web/>
- Hostinger. (2025, diciembre 18). *Los 10 mejores ejemplos de aplicaciones web*. <https://www.hostinger.com/es/tutoriales/ejemplos-de-aplicaciones-web>
- Hostinger. (2025, marzo 4). *¿Cuál es la diferencia entre una aplicación web y un sitio web?* <https://www.hostinger.com/es/tutoriales/diferencias-entre-aplicacion-web-y-sitio-web>
- HubSpot. (2025, agosto 21). *Qué son las aplicaciones web y 8 ejemplos*. <https://blog.hubspot.es/website/que-es-aplicacion-web>
- IDA Chile. (2022, abril 5). *Diferencias entre un sitio web y una aplicación web*. <https://blog.ida.cl/estrategia-digital/diferencias-aplicacion-web-sitio-web/>

InvGate. (2025, febrero 4). *Los 14 problemas más comunes de soporte técnico y cómo resolverlos*. <https://blog.invgate.com/es/problemas-mas-comunes-de-soporte-tecnico>

INVID Group. (2025, septiembre 16). *3 diferencias clave entre un sitio web y una aplicación web*. <https://invidgroup.com/3-diferencias-clave-entre-un-sitio-web-y-una-aplicacion-web/>

Kiwop. (2023, septiembre 27). *Tipos de aplicaciones web*. <https://www.kiwop.com/blog/tipos-de-aplicaciones-web>

Los Creativos. (2025, julio 17). *Conoce 17 diferencias entre página web y aplicación web*. <https://loscreativos.co/marketing-digital/diferencias-entre-pagina-web-aplicacion-web/>

Microsoft Learn. (s.f.). *Patrón de backends para frontends - Azure Architecture Center*. <https://learn.microsoft.com/es-es/azure/architecture/patterns/backends-for-frontends>

Microtech. (s.f.). *Solución de problemas de logística con un software ERP*. <https://www.microtech.es/blog/soluci%C3%B3n-de-problemas-de-logistica-con-un-software-erp>

Modyo Docs. (s.f.). *Backend for Frontend (BFF)*. <https://docs.modyo.com/es/architecture/patterns/bff>

Ordenatech. (2020, enero 10). *5 consejos para solucionar problemas de software en tu empresa*. <https://www.ordenatech.es/5-consejos-para-solucionar-problemas-de-software-en-tu-empresa/>

Platzl. (s.f.). *¿Qué es una aplicación web? Cómo funcionan y ejemplos*. <https://platzl.com/blog/que-son-aplicaciones-web/>

Profile. (2025, septiembre 1). *Tipos de desarrollo de aplicaciones web: ejemplos y características*. <https://profile.es/blog/desarrollo-aplicaciones-web/>

R2 Docuo. (2024, marzo 18). *Las 4 formas (ahora 5) de resolver un problema en informática de gestión*. <https://www.r2docuo.com/es/las-4-formas-de-resolver-un-problema-en-informatica-de-gestion>

Red Hat. (s.f.). *¿Qué es una arquitectura de aplicaciones? Tipos de arquitecturas de aplicaciones*. <https://www.redhat.com/es/topics/cloud-native-apps/what-is-an-application-architecture>

Rootstack. (s.f.). *¿Qué problemas resuelve un software ERP?* <https://rootstack.com/es/blog/que-problemas-resuelve-un-software-erp>

Saint Leo University. (s.f.). *¿Cuáles son las diferencias entre sitio web y aplicación web?* <https://worldcampus.saintleo.edu/blog/cuales-son-las-diferencias-entre-sitio-web-y-aplicacion-web>

Scriptcase Blog. (s.f.). *Aplicaciones Web (WebApp) vs Websites – Entiende la diferencia*. <https://scriptcaseblog.net/es/development-es/aplicaciones-web-webapp-vs-websites-entienda-la-diferencia/>

Sistemas Catalunya. (2024, junio 20). *5 Problemas de software y sus soluciones*. <https://www.sistemas-catalunya.com/5-problemas-de-software-y-sus-soluciones/>

Telefónica. (2022, diciembre 14). *5 aplicaciones Web 3.0 y ejemplos que debes conocer*. <https://www.telefonica.com/es/sala-comunicacion/blog/5-aplicaciones-web-3-0-y-ejemplos-que-debes-conocer/>

Thomson Reuters. (s.f.). *6 problemas que tu pyme puede solucionar con un software ERP*. <https://www.thomsonreuters.com.ar/es/soluciones-fiscales-contables-gestion/blog-empresas/6-problemas-que-tu-pyme-puede-solucionar-con-un-software-erp.html>

Weblabor. (s.f.). *Diferencias entre una aplicación web y una página web*. <https://weblabor.mx/blog/diferencia-entre-una-aplicacion-web-y-una-pagina-web>