



UNIVERSIDAD TECNOLÓGICA DE TEHUACÁN

S2 Arquitectura de información y navegación accesible

**Nombre del Alumno:
Manelic Alitzel Reyes Torres**

**Docente:
José Miguel Carrera Pacheco**

**Arquitectura de información y
navegación accesible**

**ING EN DESARROLLO
DE SOFTWARE
MULTIPLATAFORMA**

Estrategia de Producto: Página Web vs. Aplicación Web

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.

Benchmarks de Referencia:

- 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.

2. Arquitectura de Sistemas y Capas de Responsabilidad

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.

Decisión Técnica Crítica: No todos los reportes son iguales. Un problema de infraestructura (silla rota) no requiere el mismo nivel de confidencialidad que un reporte de acoso. Debemos implementar Niveles de Privacidad estrictos.