Tehuacán, Puebla
Fecha: 18 / 01 / 2026
TL: Oscar Espinoza Landeta
Matricula: 3523110665
Carrera: TICS DSM

Bloque 1

Páginas y Aplicaciones Web

Resumen

Es crucial que profesionales y empresas entiendan la distinción entre un **sitio web informativo** y una **aplicación web**:

- **Sitios Web Informativos:** Sirven principalmente como herramientas de *marketing* y a menudo se gestionan fácilmente con sistemas CMS (Content Management Systems) como WordPress o Wix.
- **Aplicaciones Web:** Constituyen el núcleo operativo y funcional de un negocio.

Por consiguiente, el desarrollo de *software* basado en tecnologías web (aplicaciones) mantiene una altísima demanda para desarrolladores con experiencia específica en campos críticos como la banca, la inteligencia artificial y las grandes plataformas tecnológicas.

Definiciones y Estructura Técnica

Para comprender el panorama actual, es necesario precisar los conceptos técnicos que a menudo se confunden en el lenguaje coloquial:

- **Página Web:** Es la unidad mínima; un documento escrito en lenguaje HTML que puede incluir CSS, JavaScript, imágenes y videos. Cada pantalla individual (Inicio, Cursos, Blog) es una página.
- **Sitio Web:** Es el conjunto de páginas web estructuradas bajo un mismo dominio (ejemplo: **`ed.team`**). Técnicamente, tanto un sitio informativo como una aplicación web entran en esta categoría, aunque su propósito difiere.
- **Dominio:** Es la dirección o sección que identifica a un sitio en internet (ejemplo: **`google.com`**, **`netflix.com`**).

Comparativa: Páginas Web vs. Aplicaciones Web

La diferencia principal no radica solo en la tecnología, sino en el **objetivo de negocio** y la **complejidad**.

Característica	Página Web	Aplicación Web
Objetivo Principal	Marketing y comunicación.	Es el núcleo del negocio (Software).
Función	Informar sobre servicios, capturar leads, vender productos externos.	Ejecutar procesos internos, gestionar usuarios, interacciones complejas.
Ejemplos	Restaurantes, clínicas, colegios, diarios.	Netflix, Facebook, Slack, Notion, EdTeam.
Herramientas	CMS (WordPress), Builders (Wix, HubSpot).	Librerías y Frameworks (React, Angular, Vue, Svelte).
Perfil Profesional	Profesionales de marketing / Desarrolladores básicos.	Desarrolladores web profesionales (Frontend/Backend).

El Sitio Web Informativo

Su propósito es integrarse en una estrategia de marketing digital. No es el negocio en sí, sino una herramienta para que el cliente conozca la empresa. Al ser fácilmente reemplazable por perfiles de redes sociales o constructores visuales, su valor depende de la capacidad de generar ventas o prospectos, no de la complejidad del código.

La Aplicación Web

Es un software construido con tecnologías web. Es donde ocurre la actividad principal de la empresa:

- **Interacción:** Los usuarios comentan, ven historiales, se suscriben, cancelan servicios y consumen contenido directamente.
- **Personalización:** Requiere soluciones a medida que no pueden ser cubiertas por un CMS estándar.
- **Plataforma Universal:** La web actúa como la base para visualizar datos (incluso en Inteligencia Artificial) y servicios complejos.

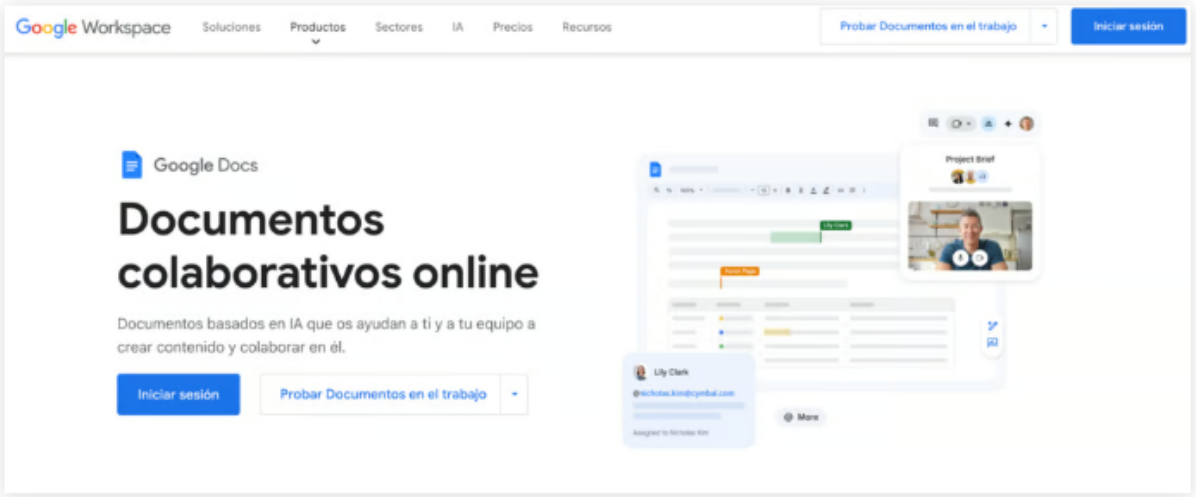
Desarrollo Técnico y Escalabilidad

El proceso de creación varía drásticamente según la naturaleza del proyecto:

1. **Enfoque CMS (Sin código o bajo código):** Ideal para sitios de marketing o medios de comunicación cuyo valor principal es el contenido.
2. **Enfoque Profesional (Código personalizado):** Requiere el uso de tecnologías de vanguardia:
    - **Backend:** Node.js, Go, Python, Java, APIs (REST o GraphQL).
    - **Infraestructura:** Computación en la nube (Cloud computing).
3. **Transición y MVP:** Es posible iniciar una aplicación web utilizando un CMS como Producto Mínimo Viable (MVP) para validar el mercado

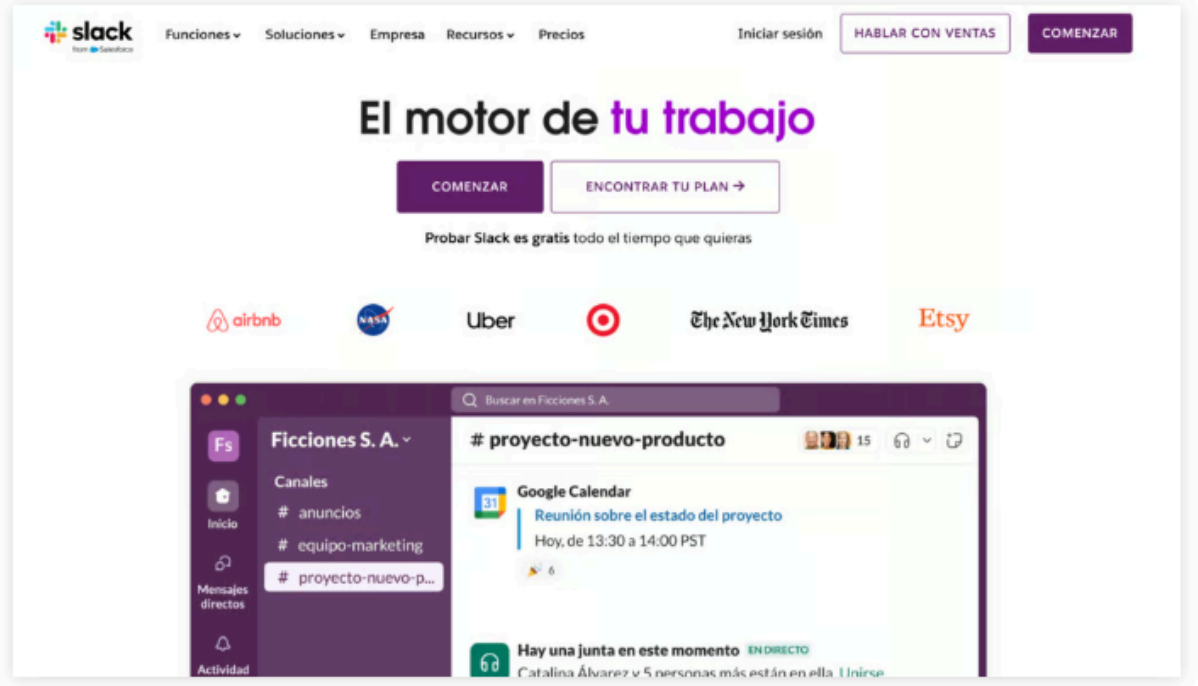
Ejemplos de Páginas Web Profesionales

1. Google Docs



Google Docs es una aplicación de procesamiento de textos basada en la nube que permite a los usuarios escribir, editar y colaborar en tiempo real.
Como parte de Google Workspace, se utiliza ampliamente para todo, desde tareas escolares hasta informes empresariales, lo que la convierte en una herramienta esencial para estudiantes, profesionales y equipos.
Docs funciona con la infraestructura en la nube de Google. El guardado automático en Google Drive permite que varios usuarios colaboran sin problemas y garantiza que no se pierda ningún progreso si se corta la conexión a Internet.

2. Slack



Slack es una plataforma de mensajería en la nube diseñada para mejorar la comunicación y la colaboración en el lugar de trabajo.
Las empresas y los equipos remotos la utilizan para ordenar y agilizar los debates, gestionar proyectos e integrar datos y documentos de herramientas de productividad.
Slack actualiza dinámicamente los hilos de mensajes y las reacciones en tiempo real. Utiliza el protocolo WebSocket para permitir la mensajería bidireccional entre los usuarios y el servidor.

¿Qué tipo de problemas se resuelven con software?

Problemas Resueltos con Software

El software, particularmente en forma de aplicaciones web y plataformas especializadas, se ha convertido en la solución primordial para optimizar, automatizar y transformar una amplia gama de desafíos en casi todos los sectores. Los problemas que resuelve pueden clasificarse en varias categorías fundamentales:

1. Problemas de Eficiencia y Automatización

Estos son los problemas relacionados con tareas repetitivas, lentas o propensas a errores humanos.

- **Automatización de Tareas Repetitivas:** El software se utiliza para ejecutar procesos de forma autónoma (ej. facturación automática, envío programado de correos electrónicos, *bots* de atención al cliente).
- **Gestión de Datos a Gran Escala:** Los sistemas de bases de datos y el software de análisis (BI) permiten manejar, procesar y extraer conocimiento de volúmenes masivos de información (Big Data) que serían imposibles de gestionar manualmente.
- **Optimización de Procesos:** Plataformas como los ERP (Enterprise Resource Planning) o los sistemas de gestión de inventario (WMS) eliminan los silos de información, coordinando departamentos y acelerando los flujos de trabajo.

2. Problemas de Comunicación y Colaboración

El software conecta personas, equipos y sistemas sin importar su ubicación geográfica.

- **Coordinación Remota:** Herramientas como Slack, Zoom o Google Workspace permiten a los equipos trabajar juntos en tiempo real y compartir recursos de manera instantánea, superando las barreras de la distancia.
- **Servicio al Cliente (CS):** Los CRM (Customer Relationship Management) y las plataformas de soporte técnico (ej. Zendesk) centralizan las interacciones con los clientes, garantizando respuestas más rápidas y un seguimiento personalizado.
- **Transparencia de la Información:** Las plataformas compartidas aseguran que todos los miembros de un equipo o la organización trabajen con la versión más reciente y precisa de los documentos y los datos.

3. Problemas de Acceso y Distribución

El software permite la distribución de servicios, productos y conocimiento a una escala masiva y global.

- **Comercio Electrónico:** Las aplicaciones web de *e-commerce* resuelven el problema de vender productos y servicios más allá de una ubicación física, gestionando catálogos, pagos, inventarios y logística.
- **Educación y Contenido:** Plataformas como EdTeam, Netflix o Coursera democratizan el acceso al conocimiento y al entretenimiento, permitiendo el consumo bajo demanda.
- **Servicios Financieros (Fintech):** Las aplicaciones bancarias y de pagos facilitan transacciones seguras y el acceso a servicios financieros básicos (préstamos, inversiones) desde cualquier dispositivo.

4. Problemas de Personalización y Experiencia de Usuario

Las soluciones de software se diseñan para ofrecer experiencias adaptadas a las necesidades individuales.

- **Recomendación:** Algoritmos de IA en plataformas de *streaming* o *e-commerce* resuelven el problema de la "sobrecarga de opciones" ofreciendo contenido o productos relevantes a cada usuario.
- **Interacción a Medida:** Las aplicaciones web complejas, como las de salud (registros médicos electrónicos) o ingeniería (software CAD), se adaptan a flujos de trabajo muy específicos que no pueden ser cubiertos por herramientas genéricas.

En resumen, el software resuelve el problema fundamental de la **escalabilidad**, permitiendo que las operaciones de una pequeña empresa crezcan para servir a millones de usuarios o gestionar miles de millones de datos sin un aumento lineal en la mano de obra.

Arquitectura General de Aplicaciones Web

1. Frontend (La cara del usuario)

Dado que necesitas que funcione en la obra (caseta) y en la calle (repartidores), lo ideal es una **PWA (Progressive Web App)**. No requiere descarga de la App Store y funciona como una app nativa.

- **Framework: React.js** o **Next.js**. Son rápidos de desarrollar y tienen librerías de sobra para manejar cámaras y QRs.
- **Estilos: Tailwind CSS**. Te permite crear una interfaz profesional y responsiva (celular/tablet) en tiempo récord.
- **Librerías Clave:**
    - html5-qrcode: Para que el vigilante escanee usando la cámara del navegador.
    - speakeasy o otplib: Para generar el **QR Dinámico** (algoritmo TOTP).
    - framer-motion: Para que la experiencia se sienta como una app de alta calidad.

---

2. Backend (El cerebro y los datos)

Necesitas algo que maneje tiempo real (notificaciones al residente) y que sea muy seguro.

- **Opción Recomendada (BaaS): Firebase (Google)**.
    - **Firestore:** Base de datos NoSQL en tiempo real. Perfecta para que el residente vea cuando alguien entra "al instante".
    - **Firebase Auth:** Maneja el registro de usuarios, validación de teléfono (SMS) y seguridad sin que tú programes el login desde cero.
    - **Cloud Functions:** Micro-servicios para la lógica pesada (ej. validar si un código QR ya expiró o enviar una alerta de tiempo excedido).
    - **Cloud Storage:** Para guardar las fotos de perfil, las fotos del INE y las fotos de las placas.

---

3. Infraestructura y Entornos

Para un MVP profesional, debes separar donde "juegas" de donde "vives" (Producción).

Entornos:

1. **Development (Local):** Tu computadora donde escribes el código.
2. **Staging/Testing:** Un entorno en la nube idéntico al real donde pruebas que el escáner no falle antes de dárselo a un vigilante.
3. **Production:** El sistema real usado por los fraccionamientos en Tehuacán.

Hosting / Despliegue:

- **Vercel** o **Netlify**: Para hospedar el Frontend. Se conectan a tu GitHub y despliegan cambios automáticamente.

- **Dominio:** Un nombre profesional (ej. paserapido.mx o accesotehuacan.com).

---

4. Diagrama de Flujo de Datos (Arquitectura)

$$\text{Usuario (Repartidor)} \xrightarrow{\text{Genera QR TOTP}} \text{Frontend PWA}$$
$$\text{Vigilante (Caseta)} \xrightarrow{\text{Escanea QR}} \text{Backend (Cloud Function)}$$
$$\text{Backend} \xrightarrow{\text{Valida y registra}} \text{Base de Datos (Firestore)}$$
$$\text{Base de Datos} \xrightarrow{\text{Notificación Push}} \text{App del Residente}$$

---

5. El Modelo de Datos (Estructura de la BD)

Para que tu sistema sea rápido, organiza tu base de datos de esta forma:

Colección: Usuarios

- uid: ID único.
- nombre_completo: String.
- foto_url: Link a la foto en Storage.
- tipo: "repartidor" | "visitante".
- placas: String (opcional).
- verificado: Booleano (si ya revisaste su INE).
- secret_2fa: La semilla para generar su QR dinámico.

Colección: Accesos (Bitácora)

- id_acceso: ID único.
- usuario_id: Referencia al usuario que entró.
- fraccionamiento_id: A qué fraccionamiento entró.
- casa_destino: Número de casa.
- timestamp_entrada: Fecha y hora.
- timestamp_salida: Fecha y hora (nulo hasta que salga).
- status: "dentro" | "fuera" | "sospechoso".

Análisis de 2 plataformas reales similares a la idea del equipo

1. Los Competidores Principales

A. LobbyFix / Colonos

Son de las más usadas en México y Latinoamérica.

- **Lo que hacen:** Permiten generar invitaciones QR, tienen comunicación con administración y registro de pagos de mantenimiento.
- **Su debilidad:** Son sistemas "cerrados". Cada fraccionamiento es un mundo aparte. Si tú como repartidor vas a 5 fraccionamientos, tendrías que estar registrado en 5 sistemas diferentes o esperar a que cada residente te mande un QR distinto.

B. Proteget (Muy fuerte en México)

- **Lo que hace:** Enfocada 100% en seguridad. El vigilante escanea identificaciones y placas.
- **Su debilidad:** Es muy burocrática. El proceso en caseta sigue siendo lento porque está diseñado para "fiscalizar" más que para "agilizar".

C. Parrot (Enfocada en repartidores)

- **Lo que hace:** Algunas apps de delivery tienen sus propios sistemas de acceso, pero solo funcionan para sus propios empleados.

---

2. ¿Cómo diferenciar tu idea? (Tu "Ventaja Competitiva")

Aquí es donde tu proyecto puede ganarles, especialmente en una ciudad como **Tehuacán**:

1. El concepto de "Pasaporte Universal"

- **Competencia:** El visitante es un "extraño" hasta que el residente lo invita.
- **Tu Idea:** El visitante/repartidor es un **usuario verificado de tu plataforma**.
- **Diferencia:** Si un repartidor de "Tacos El Chino" ya se registró una vez para entrar a *Lomas de la Soledad*, cuando vaya a *Residencial San Lorenzo*, su QR ya funciona. El vigilante solo escanea y valida. **Ahorras tiempo real.**

2. Enfoque en el Repartidor Local (No solo Apps)

- **Competencia:** Se enfocan en Uber Eats o Rappi.
- **Tu Idea:** En Tehuacán hay muchísimos negocios propios (farmacias, ferreterías, comida local) que reparten por su cuenta.
- **Diferencia:** Tu app les da una "identidad profesional" a estos repartidores locales, dándoles estatus y confianza ante los fraccionamientos.

3. Tecnología de QR Dinámico (Antifraude)

- **Competencia:** La mayoría usa QRs estáticos que se pueden enviar por captura de pantalla (fáciles de robar).
- **Tu Idea:** Tu QR cambia cada 30 segundos (tipo Token bancario).
- **Diferencia:** Vendes **"Seguridad de Grado Bancario"**. Si alguien roba un celular o toma una foto al QR, para cuando lleguen a la caseta, ese código ya no sirve.

4. Costo de Implementación "Light"

- **Competencia:** Suelen vender hardware caro (cámaras especiales, tótems, escáneres fijos).
- **Tu Idea:** Todo funciona desde el celular del repartidor y la tablet del vigilante.
- **Diferencia:** Es un software que cualquier fraccionamiento puede empezar a usar **mañana mismo** sin comprar equipo costoso.

---

3. Matriz de Diferenciación Rápida

Característica	Apps Tradicionales	Tu Proyecto (MVP)
Registro	Por cada fraccionamiento.	**Único y Global.**
Velocidad	Media (esperar invitación).	**Alta (llegar y escanear).**
Seguridad QR	Estática (Captura de pantalla).	**Dinámica (Token temporal).**
Costo Caseta	Alto (Hardware especial).	**Bajo (Tablet/Celular).**

Bloque 2

1. Arquitectura de Información (AI)

Es la forma en que organizas y etiquetas el contenido para que el usuario encuentre lo que busca sin pensar.

- **En tu App:** No mezcles el registro de perfil con el visor de QR.
- **Estructura Sugerida:**
    - **Vista Repartidor:** Botón gigante para mostrar QR + Historial de accesos.
    - **Vista Vigilante:** Escáner central + Lista de personas "Dentro ahora".

    - **Vista Residente:** Notificaciones + Mi Casa (gestión de quién está llegando).

2. Jerarquía de Contenido

Se refiere a cómo guías el ojo del usuario hacia lo más importante usando tamaño, color y contraste.

- **Nivel 1 (Crítico):** El código QR dinámico y la foto del perfil en la tablet del vigilante.
- **Nivel 2 (Importante):** Nombre del usuario y número de casa.
- **Nivel 3 (Secundario):** Placas del vehículo o marca de la moto.
- **Aplicación:** Usa colores de contraste (ej. un botón verde brillante para "Autorizar") y fuentes grandes para datos que se leen a distancia.

3. Patrones de Navegación Web

Son rutas familiares que los usuarios ya conocen.

- **Barra de Navegación Inferior (Tab Bar):** Crucial para el repartidor. Poner los botones de "Inicio", "Perfil" y "Ayuda" al alcance del pulgar.
- **Dashboard (Panel de Control):** Para el vigilante. Una pantalla limpia que muestre indicadores rápidos: *"X personas dentro"*, *"1 alerta de tiempo"*.

---

4. Accesibilidad y Navegación Pro (Sin Mouse)

Muchos vigilantes en casetas usan computadoras de escritorio o tablets con teclados físicos. Diseñar para ellos acelera el proceso.

Orden de Tabulación (Tab Order)

Es el camino que sigue el cursor cuando presionas la tecla Tab.

- **Regla:** El orden debe ser lógico (de izquierda a derecha y de arriba a abajo).
- **En tu App:** Si el vigilante registra un acceso manual, el Tab debe ir de: *Nombre -> Casa -> Placas -> Botón Guardar*. Nunca debe saltar a ciegas.

Navegación por Teclado y Atajos

- **Inputs enfocados:** Cuando el vigilante abra la pantalla de registro manual, el cursor debe estar **automáticamente** en el campo "Nombre" o "Placas".
- **Hotkeys:** Programar que la tecla Enter confirme el acceso y Esc cierre el escáner. Esto reduce el tiempo en caseta de 30 segundos a 10.

Accesibilidad sin Mouse (A11y)

- **Áreas de toque (Touch Targets):** Los botones deben medir al menos **44x44 píxeles**. El vigilante suele usar guantes o tener los dedos grandes; si los botones son pequeños, habrá errores.
- **Contraste de Color:** Asegura que el texto sea legible bajo el sol intenso de Tehuacán. Usa fondos blancos con texto negro puro, evitando grises claros.

---

5. Checklist de Usabilidad para tu MVP

Concepto	Pregunta de Validación
Jerarquía	¿Es la foto del repartidor lo más grande en la pantalla del vigilante?
Navegación	¿Puede el repartidor llegar a su QR en menos de 2 clics al abrir la app?
Tabulación	¿Puede el vigilante llenar el registro usando solo el teclado de su computadora?
Feedback	¿Hay un sonido o vibración clara cuando el QR es aceptado o rechazado?

ABRAHAM CERVANTES ROMERO

Bloque 1 – Fundamentos del proyecto Antes de proponer cualquier idea, investiga y documenta: Diferencia entre

Concepto	Definición Profesional	Ejemplo Real
Página Web	Enfocada en **consumo de información**. Estática o con poca interacción.	Blog de noticias, Portfolio.
Aplicación Web	Enfocada en **interacción y procesos**. Manipula datos y tiene estados de usuario.	Trello, Gmail, Spotify Web.

- Qué tipo de problemas se resuelven con software
    - 1. Automatización: Eliminar tareas repetitivas (ej. enviar correos de recordatorio).
    - 2. Centralización: Tener una "única fuente de verdad" para los datos.
    - 3. Escalabilidad: Procesar 10 o 10,000 registros con el mismo esfuerzo humano.

- Arquitectura general de aplicaciones web: o Frontend o Backend o Infraestructura / entornos

1. Arquitectura de Información (AI) y Jerarquías

La AI es la disciplina de organizar, etiquetar y estructurar el contenido para que el usuario encuentre lo que busca sin esfuerzo mental.

- Jerarquías de Contenido: Es el orden de importancia que le das a los elementos. Como ingeniero, esto se traduce en cómo anidas tus componentes y etiquetas HTML.
    - Visual: Uso de tamaños, pesos de fuente y colores.
    - Estructural: Uso correcto de etiquetas de encabezado (h1, h2, h3). Un error de QA común es saltarse niveles (pasar de h1 a h3), lo cual rompe la jerarquía para los lectores de pantalla.

- Sitemap (Mapa del sitio): Es el plano arquitectónico. Define la relación entre las páginas (Padre/Hijo/Hermano) y ayuda a planificar las rutas del backend.

2. Patrones de Navegación Web

Son soluciones estándar que los usuarios ya conocen. No "inventes el hilo negro" si quieres que tu sistema sea profesional.

- Barra Lateral (Sidebar): Ideal para aplicaciones complejas (Dashboards) con muchas herramientas (ej. Propuesta 1 de SkillSync).
- Barra Superior (Navbar): Ideal para sitios con 3 a 5 secciones principales (ej. Propuesta 2 de MediTrack).
- Migas de Pan (Breadcrumbs): Indispensables en sistemas con jerarquías profundas para que el usuario sepa dónde está y cómo volver atrás.

3. Accesibilidad Técnica (Navegación sin Mouse)
Aquí es donde un QA Engineer pone a prueba la calidad del código. Un sistema profesional debe ser operable únicamente con el teclado.

Orden de Tabulación (Focus Order)

Cuando un usuario presiona la tecla Tab, el "foco" debe moverse en un orden lógico (generalmente de arriba a abajo y de izquierda a derecha).

- El Error del Junior: Usar z-index o posicionamiento absoluto que visualmente mueve un botón, pero mantiene el foco en otro lugar.
- La Regla de Oro: El orden del DOM (el código HTML) debe coincidir con el orden visual.

Navegación por Teclado y Estados de Foco
No es suficiente que se pueda navegar; el usuario debe ver dónde está.

- Indicador de Foco: Nunca quites el outline de CSS sin poner un reemplazo visual claro. Un usuario sin mouse que no ve dónde está el foco es un usuario perdido.
- Atajos: Las aplicaciones profesionales suelen usar teclas como Esc para cerrar modales o Enter/Espacio para activar botones.

Accesibilidad sin Mouse (A11y)
Se basa en principios WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Applications).

- Roles: Si haces un "botón" usando un, los lectores de pantalla no sabrán qué es. Debes usar la etiqueta o el atributo role="button".
- Alt Text: Las imágenes que transmiten información deben tener descripciones. Si son decorativas, deben marcarse como tales para que el software de asistencia las ignore.

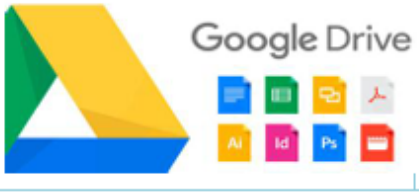
LILIA HERNANDEZ TUN

Bloque 1 – Fundamentos del proyecto

- Diferencia entre:
    - Página web: Un documento en la World Wide Web, que consiste en un archivo de hipertexto y cualquier archivo relacionado con scripts y gráficos, y a menudo enlazado con hipervínculos a otros documentos en el Web.
    - Aplicación web: Son programas de software que se ejecutan en un navegador web, como Google Chrome o Firefox. Estas aplicaciones no se descargan en nuestros dispositivos, sino que se ejecutan en un servidor y se transmiten a través de Internet al navegador de nuestro dispositivo.

Página web	Aplicación Web
Proporciona contenido estático, principalmente para leer o navegar.	Permite a los usuarios completar tareas, objetivos o acciones específicas.
Puede construirse utilizando sólo HTML y CSS; puede incluir JavaScript, o un sistema de gestión de contenidos como WordPress.	Requiere marcos avanzados, tecnologías del lado del servidor y API para una funcionalidad dinámica.

- Ejemplos reales de aplicaciones web profesionales

	Canva es una aplicación que permite crear infografías, posters, logotipos y una gran cantidad de diseños desde su sitio web con herramientas propias, sin necesidad de conocimientos técnicos o profesionales. Cuenta con la alternativa de colaborar con equipos.
	Esta es una aplicación que permite subir documentos en PDF y ejecutar varias funciones como comprimirlos hasta 75 % menos peso; reordenar, eliminar o separar páginas; combinar documentos o convertirlos a otros formatos.
	Edita y organiza archivos de Documentos, Hojas de cálculo y Presentaciones de Google, de Microsoft Office y PDFs en tiempo real.

- Qué tipo de problemas se resuelven con software

El software permite resolver una gran variedad de problemas relacionados con la gestión de información, la automatización de procesos y la optimización del tiempo y recursos. Pueden solucionar problemas como:

Organización y almacenamiento de datos: Permite guardar grandes volúmenes de información de forma segura y ordenada, facilitando su consulta y actualización.

Automatización de procesos: Reduce tareas repetitivas y manuales, como cálculos, registros o generación de reportes, disminuyendo errores humanos.

Comunicación y acceso a la información: Facilita la interacción entre usuarios y sistemas desde cualquier lugar y en cualquier momento.

Control y seguimiento: Ayuda a monitorear actividades, procesos o servicios en tiempo real, mejorando la toma de decisiones.

Mejora de la eficiencia y productividad: Optimiza el uso de recursos, permitiendo realizar tareas de forma más rápida y eficaz.

- Arquitectura general de aplicaciones web
  La arquitectura de una aplicación web define cómo se organiza el sistema y cómo interactúan sus componentes para ofrecer un servicio funcional y eficiente. Se divide en tres partes principales:
  - Frontend Es la parte visible de la aplicación, con la que el usuario interactúa directamente. Incluye el diseño, la interfaz gráfica y la experiencia de usuario. Se encarga de mostrar la información de forma clara y accesible, así como de captar las acciones del usuario, como clics, formularios o navegación.
  - Backend Es la parte interna de la aplicación que procesa la lógica del sistema. Se encarga de gestionar la información, procesar solicitudes del frontend, aplicar reglas de negocio y comunicarse con las bases de datos. El backend asegura que los datos sean correctos, seguros y estén disponibles cuando se necesiten.
  - Infraestructura / Entornos Incluye los recursos tecnológicos necesarios para que la aplicación funcione correctamente. Abarca servidores, bases de datos, redes, servicios en la nube y los diferentes entornos de trabajo, como desarrollo, pruebas y producción. Esta parte garantiza la disponibilidad, seguridad y rendimiento de la aplicación web.

- Análisis de 2 plataformas reales similares a la idea del equipo

Bloque 2 - Arquitectura de información y accesibilidad

- Arquitectura de información:
  Es una metodología que ayuda a decidir la organización de las partes de un algo a fin de que sea comprensible para quien recibe la información.

  Existen dos metodologías de la arquitectura de información:

  Método de arquitectura ancha: el usuario no ejecuta muchos clics en la página web para llegar a un nivel detallado, pero cuenta con muchas opciones en cada nivel visitado.

  Método de arquitectura profunda: es el caso opuesto al anterior. No cuenta con muchos niveles aunque requiere de muchos clics para llegar al cometido.

- **Jerarquías de contenido**:

  Se refiere a la estructura y organización de la información en una página web. Es la forma en la que se presenta el contenido para que los usuarios puedan entender y navegar fácilmente por la página. Esta estructura se basa en la importancia y relevancia de cada elemento en la página, ayuda a guiar la atención del usuario y a comunicar de manera efectiva.

  Elementos de la jerarquía de contenido

  Título principal (H1): Define el tema principal de la página y es el elemento más importante dentro de la jerarquía. Debe ser claro y descriptivo.

  Encabezados secundarios (H2 y H3): Se utilizan para dividir el contenido en secciones y subsecciones, facilitando la lectura y comprensión del texto.

  Contenido principal: Incluye textos, imágenes, videos u otros elementos multimedia. Debe estar organizado de manera coherente y alineado con los encabezados para mantener una estructura clara.

  Enlaces y llamadas a la acción (CTA): Permiten guiar al usuario hacia acciones específicas como registrarse, comprar un producto o contactar con la empresa.

  Espacio en blanco: Ayuda a mejorar la legibilidad y comprensión del contenido, evitando que la página se vea saturada o confusa.

- Patrones de navegación web: Los patrones de navegación web son modelos comunes que ayudan a los usuarios a moverse dentro de un sitio o aplicación de forma intuitiva. Estos patrones permiten una experiencia más fluida y accesible.

- Orden de tabulación: Define el recorrido que sigue el cursor al navegar con el teclado. Un orden lógico de tabulación mejora la accesibilidad y facilita el uso del sitio para personas que no utilizan mouse.

- Navegación por teclado: Permite que los usuarios interactúen con todos los elementos del sitio utilizando únicamente el teclado, como enlaces, botones y formularios.

- Accesibilidad sin mouse: Garantiza que todas las funcionalidades del sitio puedan ser utilizadas sin necesidad de un mouse, beneficiando a personas con discapacidades motoras o visuales.

Referencias Bibliográficas

- Farlex. (2021, 23 de abril). Web page. The Free Dictionary. https://web.archive.org/web/20210423170632/https://www.thefreedictionary.com/web+page un • Hostinger. (2024, 15 de abril). ¿Cuál es la diferencia entre una aplicación web y un sitio web?

- Hostinger Tutoriales. https://www.hostinger.com/es/tutoriales/diferencias-entre-aplicacion-web-y sitio-web

- KeepCoding. (2023, 10 de octubre). Jerarquía de contenido en una aplicación web. KeepCoding Blog. https://keepcoding.io/blog/jerarquia-de-contenido-en una-aplicacion-web/

- Ondho. (s.f.). Arquitectura de la Información. Diccionario de Marketing. https://ondho.com/diccionario-de-marketing/term/arquitectura-de-la informacion/

- Platzi. (2022, 14 de julio). ¿Qué son las aplicaciones web y cómo funcionan? Platzi Blog. https://platzi.com/blog/que-son-aplicaciones-web/

- Sociment. (2023, 22 de mayo). ¿Qué es una aplicación web y qué beneficios tiene? Sociment. https://www.sociment.com/que-es-aplicacion-web/

FRANCISCO XAVIER GIL GINEZ

FUNDAMENTOS DEL PROYECTO
Diferencia entre Página web/Aplicación web:

Una página web es un sitio cuyo propósito principal es mostrar información al usuario. Su contenido suele ser estático o con cambios mínimos y está orientado a la lectura o consulta, como textos, imágenes o videos. Normalmente no requiere que el usuario inicie sesión ni que interactúe de forma compleja con el sistema. Un ejemplo típico sería un blog personal o una página institucional donde solo se consulta información.

Una aplicación web, en cambio, está diseñada para que el usuario interactúe activamente con ella. Permite realizar acciones como registrarse, iniciar sesión, crear, modificar o eliminar información, y suele estar conectada a una base de datos. Funciona de manera similar a un programa, pero desde el navegador, como ocurre con Gmail o Google Docs. Su contenido es dinámico y responde constantemente a las acciones del usuario.

Ejemplos reales de aplicaciones web profesionales

Las aplicaciones web profesionales son sistemas utilizados diariamente por millones de personas y empresas. Un ejemplo claro es Gmail, que permite administrar correo electrónico de manera eficiente, segura y accesible desde cualquier lugar. Google Docs, que resuelve la necesidad de crear y editar documentos de forma colaborativa en tiempo real. En el ámbito empresarial, Trello ayuda a organizar proyectos y tareas, mientras que Shopify permite a negocios crear y administrar tiendas en línea sin desarrollar todo desde cero. Todos estos sistemas son aplicaciones web porque funcionan en el navegador, manejan datos dinámicos y requieren interacción constante del usuario.

Qué tipo de problemas se resuelven con software

El software se utiliza para resolver problemas que serían lentos, costosos o imposibles de manejar de forma manual. Un ejemplo seria automatizar procesos repetitivos como el registro de usuarios, el cálculo de pagos, la generación de reportes o el control de inventarios. También permite almacenar y consultar grandes volúmenes de información de manera segura, mejorar la comunicación entre personas o equipos, y reducir errores humanos mediante validaciones automáticas. En el contexto empresarial, el software ayuda a optimizar tiempo, reducir costos y tomar mejores decisiones a partir de datos.

Arquitectura general de aplicaciones web:

En cuanto a la arquitectura general de una aplicación web, esta se divide normalmente en tres capas principales.

El frontend es la parte visible para el usuario y se ejecuta en el navegador. Aquí se encuentran las interfaces gráficas, formularios, botones y vistas con las que el usuario interactúa, tecnologías comunes en esta capa incluyen HTML, CSS y frameworks de

JavaScript como React o Vue, cuyo objetivo principal es ofrecer una experiencia de uso clara y rápida.

El backend es la capa lógica del sistema y se ejecuta en el servidor. Se encarga de procesar las solicitudes del frontend, validar datos, aplicar reglas de negocio, gestionar la autenticación de usuarios y comunicarse con la base de datos. En esta parte suelen utilizarse lenguajes y frameworks como Node.js con Express, Java con Spring o Python con Django. El backend es esencial para que la aplicación funcione de forma segura y consistente.

Por último, la infraestructura y los entornos son los componentes que permiten que la aplicación esté disponible en internet. Esto incluye servidores, servicios en la nube, bases de datos, sistemas de despliegue y entornos de desarrollo, pruebas y producción. Plataformas como AWS o Google Cloud facilitan el alojamiento, la escalabilidad y la seguridad de las aplicaciones web modernas, permitiendo que estas soporten desde pocos usuarios hasta millones de conexiones simultáneas.

En conjunto, estos elementos forman la base de cualquier aplicación web profesional utilizada actualmente en entornos académicos, empresariales y comerciales.

ANÁLISIS DE 2 PLATAFORMAS REALES SIMILARES A LA IDEA DEL EQUIPO

Cluster (Control de acceso para fraccionamientos)

Plataforma en la nube para gestionar accesos a fraccionamientos, oficinas o edificios mediante códigos QR generados por residentes.

Fortalezas: Permite crear accesos desde una app y enviarlos por WhatsApp. Evita filas en caseta: el visitante ya llega con QR listo.

Debilidades: No está enfocada en repartir una “identidad universal” del repartidor (cada fraccionamiento sigue siendo independiente). No hay QR dinámico tipo TOTP (solo QR estático tradicional).

Comparación con tu idea: Ambos usan QR, pero tu propuesta supera en seguridad dinámica (cambia cada 30 s) y el pasaporte universal que sirve en todos lados.

Gate Sentry (Software de Visitor Management)

Herramienta de gestión de accesos para comunidades cerradas, edificios y propiedades que no requiere hardware costoso y funciona con tablets o dispositivos móviles.

Fortalezas: Sin hardware costoso, todo por software en la nube. Ofrece control de visitantes y monitoreo desde un dashboard. Escalabilidad y seguridad robusta.

Debilidades: No está optimizado específicamente para repartidores locales y su dinámica frecuente. No hay QR dinámico tipo TOTP por defecto; más bien escaneo tradicional.

Comparación con tu idea: Más parecido a una solución completa empresarial, pero tu propuesta destaca en enfoque al repartidor local y QR dinámico con más seguridad antifraude.

ARQUITECTURA DE INFORMACIÓN Y ACCESIBILIDAD

Conceptos clave para diseñar sistemas usables y profesionales:

Arquitectura de información se refiere a la forma en que se organiza, estructura y presenta la información dentro de un sistema digital para que los usuarios puedan encontrarla y entenderla fácilmente. Una buena arquitectura de información define cómo se agrupan los contenidos, cómo se nombran las secciones y cómo se conectan entre sí, reduciendo la confusión y el esfuerzo cognitivo del usuario. Su objetivo principal es que cualquier persona pueda navegar por el sistema de forma intuitiva, incluso sin instrucciones previas.

Las jerarquías de contenido están directamente relacionadas con la arquitectura de información y consisten en establecer niveles de importancia dentro del contenido. Esto se logra mediante títulos, subtítulos, tamaños de texto, colores y espaciados que guían visualmente al usuario. Una jerarquía bien definida permite identificar rápidamente qué información es principal y cuál es secundaria, facilitando la lectura, el escaneo del contenido y la toma de decisiones dentro de la interfaz.

Los patrones de navegación web son estructuras de navegación ya conocidas por los usuarios, como menús superiores, menús laterales, breadcrumbs o barras de navegación fija. Utilizar estos patrones ayuda a que el sistema sea predecible y fácil de usar, ya que el usuario no necesita aprender desde cero cómo moverse dentro del sitio. La consistencia en la navegación es clave para ofrecer una experiencia profesional y reducir errores de uso.

El orden de tabulación es un aspecto fundamental de la accesibilidad y define el recorrido que sigue el foco cuando el usuario navega usando la tecla Tab. Este orden debe ser lógico y seguir la estructura visual de la página, pasando primero por los elementos más importantes como menús, formularios y botones principales. Un mal orden de tabulación puede hacer que el sistema sea confuso o inutilizable para personas que no usan mouse.

La navegación por teclado permite que todas las funcionalidades del sistema puedan ser utilizadas únicamente con el teclado. Esto incluye acceder a enlaces, completar formularios, enviar información y navegar entre secciones. Es especialmente importante para personas con discapacidades motoras, visuales o usuarios que utilizan lectores de pantalla, y es un requisito básico de accesibilidad web profesional.

La accesibilidad sin mouse amplía este concepto al asegurar que la aplicación sea completamente funcional sin depender de dispositivos apuntadores. Esto implica botones accesibles, indicadores visibles de foco, atajos de teclado y compatibilidad con tecnologías asistivas. Estas prácticas están alineadas con las recomendaciones de W3C,

específicamente con las pautas WCAG, y no solo benefician a personas con discapacidad, sino también a usuarios avanzados que prefieren una interacción más rápida y eficiente.

Referencias

World Wide Web Consortium. (2018). Web Content Accessibility Guidelines (WCAG) 2.1. https://www.w3.org/TR/WCAG21/ W3C

Rosenfeld, L., Morville, P., & Arango, J. (2015). Information architecture for the web and beyond (4th ed.). O'Reilly Media.

Nielsen, J., & Loranger, H. (2006). Prioritizing web usability. New Riders Publishing. Nielsen Norman Group

Krug, S. (2014). Don't make me think, revisited: A common sense approach to web usability (3rd ed.). New Riders.

International Organization for Standardization. (2018). ISO 9241-210: Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems. ISO. ISO

Lazar, J., Goldstein, D. F., & Taylor, A. (2015). Ensuring digital accessibility through process and policy. Morgan Kaufmann.