

Taura Birra - Plan de mejoras

Mejoras propuestas

Refactorización a React.js

- **Objetivo:** Mejorar la mantenibilidad del código, permitir la creación de componentes reutilizables y facilitar futuras ampliaciones del proyecto.
- **Herramientas y tecnologías:** React 18, Vite, React Router, CSS Modules o Tailwind, ESLint, Prettier.

Implementación de criterios de accesibilidad

- **Objetivo:** Aumentar la accesibilidad del sitio web según estándares WCAG, asegurando que sea usable por personas con distintas capacidades (lectores de pantalla, navegación por teclado, contraste adecuado, etc.).
- **Herramientas y tecnologías:** Chrome Lighthouse, axe-core, WAVE, contrast checker.

Integración de sistema de gestión de contenido

- **Objetivo:** Permitir que el contenido del sitio (descripciones de cervezas, imágenes, información de contacto, etc.) pueda actualizarse sin modificar el código, reduciendo la dependencia del desarrollador.
- **Herramientas y tecnologías:** Netlify CMS, Strapi, Contentful o Sanity, fetch/axios en React.

Implementación de formulario de pedidos vía WhatsApp

- **Objetivo:** Facilitar que los clientes puedan realizar pedidos directamente desde el sitio web de manera rápida y simple, mejorando la experiencia del usuario y potencialmente aumentando las ventas.
- **Herramientas y tecnologías:** WhatsApp API URL, HTML5, JS, Firebase/Firestore o Google Sheets (opcional), React Forms.

Etapas de desarrollo y entregas

1. Análisis de la situación actual y rediseño de la estructura del sitio

Duración estimada: 2 semanas

Actividades Detalladas

- Revisar toda la estructura de archivos HTML, CSS y JS existentes.
- Mapear secciones y funcionalidades actuales del sitio (inicio, sobre nosotros, cervezas, bar, contacto).
- Identificar elementos que se pueden convertir en componentes React reutilizables.
- Evaluar contenido que debería ser dinámico y gestionable mediante CMS.
- Revisar la accesibilidad actual:
 - Contraste de colores y tipografía.
 - Navegación mediante teclado.

- Uso de etiquetas alt y roles ARIA.
- Definir estructura de carpetas y archivos en React (components, pages, assets, services).
- Diseñar plan de flujo de datos y props entre componentes.
- Crear documento de planificación con prioridades para refactorización y accesibilidad.

Entrega

- Documento de planificación completo con análisis de situación actual, plan de refactor y accesibilidad.
- Identificación clara de componentes reutilizables y contenido dinámico.
- Accesibilidad evaluada y priorizada para mejoras.

2. Refactorización en React.js e implementación inicial de herramientas de accesibilidad

Duración estimada: 2 semanas

Actividades Detalladas

- Crear el proyecto base en React con la estructura definida en la etapa 1.
- Transformar páginas y secciones HTML existentes en componentes React.
- Integrar CSS existente en los componentes o migrarlo a CSS modular/Styled Components.
- Implementar navegación y rutas con React Router (si aplica).
- Incorporar accesibilidad inicial:
 - Añadir roles ARIA y etiquetas adecuadas en elementos interactivos.
 - Ajustar navegación por teclado.
 - Revisar y mejorar contraste de colores y tipografía.
- Validar que cada componente se comporta correctamente y mantiene responsividad.
- Realizar pruebas iniciales de funcionamiento de componentes y navegación.

Entrega

- Sitio React funcional replicando la navegación y funcionalidades existentes.
- Componentes reutilizables y accesibles según estándares básicos WCAG.
- Responsividad garantizada en distintos dispositivos.

3. Integración de CMS headless

Duración estimada: 2 semanas

Actividades detalladas

- Seleccionar y configurar Headless CMS (Netlify CMS, Strapi, Sanity o Contentful).
- Crear colecciones y definir campos para cervezas, secciones del sitio, contacto y otros contenidos dinámicos.

- Conectar React con el CMS para renderizar contenido dinámico en los componentes correspondientes.
- Ajustar componentes React para mostrar correctamente datos provenientes del CMS.
- Probar creación, edición y eliminación de contenido en CMS sin modificar código.
- Validar que el sitio sigue siendo responsive y accesible con contenido dinámico.

Entrega

- Sitio web React con contenido dinámico gestionable desde CMS y componentes preparados para futuras actualizaciones.

4. Formulario de pedidos vía WhatsApp y ajustes finales

Duración estimada: 2 semanas

Actividades Detalladas

- Diseñar y maquetar el formulario de pedidos en React (selección de cervezas, cantidad, nombre, teléfono, dirección, método de retiro).
- Validar datos en frontend (HTML5 + JS) y, opcionalmente, almacenar pedidos en backend ligero o serverless (Firebase/Firestore, Google Sheets).
- Configurar botón “Enviar por WhatsApp” con mensaje prellenado usando API URL.
- Implementar confirmación visual al usuario (modal o pantalla resumen).
- Ajustar accesibilidad del formulario (labels, roles ARIA, navegación por teclado).
- Realizar pruebas de interacción y flujo completo del sitio.
- Corregir errores y optimizar experiencia de usuario y responsividad.

Entrega

- Sitio web completo con formulario de pedidos vía WhatsApp, accesibilidad optimizada y contenido dinámico gestionable desde CMS.