Análisis Técnico del Repositorio: Carpeta Ciudadana Landing

Información General

Nombre del Proyecto: Carpeta Ciudadana Landing

Repositorio: github.com/ogticrd/carpeta-ciudadana-landing

Este repositorio parece estar orientado a crear una aplicación de **landing page** con funcionalidad para múltiples secciones. Está basado principalmente en **Vue.js** y utiliza **Quasar Framework** para facilitar el desarrollo y el diseño.

Carpeta/Archivo	Descripción
public/	Archivos estáticos como iconos, imágenes y recursos multimedia.
src/	Código fuente de la aplicación.
assets/	Recursos estáticos utilizados en la app, como imágenes.
components/	Componentes reutilizables de la interfaz de usuario.
layouts/	Plantillas (layouts) utilizadas en las páginas.
pages/	Páginas del sitio web (e.g., `FAQsPage.vue`, `IndexPage.vue`).
router/	Configuración de rutas de la aplicación.
stores/	Gestión del estado global usando **Pinia**.

Tecnología

Elemento	Descripción
Framework	Quasar (Vue.js 3) para el desarrollo de la aplicación.
SSR (Server-Side Rendering)	Configurado para SSR con Quasar, mejorando rendimiento y SEO.

Elemento	Descripción
Vite	Utiliza Vite como bundler para optimizar el desarrollo y la construcción.
Gestión de Estado	Usa **Pinia** para gestionar el estado global de la aplicación.
Ruteo	Usando **Vue Router** para gestionar la navegación.

Dependencias

Dependencia	Descripción
Dependencias principales	
`vue`, `vue-router`	Framework y librería de enrutamiento para Vue.js.
`quasar`	Framework para crear aplicaciones con diseño consistente y responsivo.
`pinia`	Gestión de estado para Vue 3 (alternativa a Vuex).
`axios`	Para realizar solicitudes HTTP.
Dependencias de desarrollo	
`eslint`, `prettier`	Herramientas de linting y formateo de código.
`@intlify/unplugin-vue-i18n`	Plugin para la internacionalización en Vue.js.
`vite-plugin-checker`	Plugin para verificaciones de tipos y linting.

Internacionalización (i18n)

Elemento	Descripción
i18n	Configurado para soportar múltiples idiomas.
Archivos de idioma	Se encuentra la configuración de idioma para **es-DO**.
vue-i18n	Utiliza el plugin **vue-i18n** para internacionalización.

Scripts en `package.json`

Script	Descripción
`dev`	Inicia la aplicación en modo de desarrollo con SSR.

Script	Descripción
`build`	Construye la aplicación para producción con un dominio específico.
`host`	Despliega la aplicación en **Firebase Hosting** después de construirla.
`lint`	Ejecuta ESLint en los archivos `.js` y `.vue` del proyecto.
`format`	Ejecuta Prettier para formatear el código según las configuraciones.

Control de Calidad

Elemento	Descripción
Pruebas	No se configuran pruebas unitarias ni de integración.

DevOps / Automatización

Elemento	Descripción
Integración Continua	No se mencionan explícitamente flujos de integración continua (CI). Sin embargo, la presencia de configuraciones de despliegue y scripts como `build` y `host` sugieren un enfoque manual para integración y despliegue.
Despliegue Continuo	El despliegue se realiza utilizando el script `host`, que ejecuta el despliegue a **Firebase Hosting**. Sin embargo, no hay un flujo automático o un archivo específico para gestionar el despliegue continuo (CD).
Faltan archivos de CI/CD	No hay archivos de configuración específicos para plataformas de CI/CD como GitHub Actions.
Automatización de Despliegue	El script `host` despliega manualmente la aplicación al entorno de Firebase.

Revisión #6
Creado 2025-06-04 18:23:15 UTC
Actualizado 2025-06-04 20:56:23 UTC