

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

<code>dev</code>	Inicia la aplicación en modo de desarrollo con SSR.
<code>build</code>	Construye la aplicación para producción con un dominio específico.
<code>host</code>	Despliega la aplicación en Firebase Hosting después de construirla.
<code>lint</code>	Ejecuta ESLint en los archivos <code>.js</code> y <code>.vue</code> del proyecto.
<code>format</code>	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 <code>build</code> y <code>host</code> sugieren un enfoque manual para integración y despliegue.
Despliegue Continuo	El despliegue se realiza utilizando el script <code>host</code> , 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 <code>host</code> despliega manualmente la aplicación al entorno de Firebase.

Revisión #6

Creado 4 junio 2025 18:23:15

Actualizado 4 junio 2025 20:56:23