

Análisis Técnico del Repositorio: cc-institutions-mock-api

Información General

Nombre del Proyecto: cc-institutions-mock-api

Repositorio: github.com/ogticrd/cc-institutions-mock-api

API Mockup para proveer data fake de las Instituciones para la App Carpeta Ciudadana.

Carpeta / Archivo	Descripción breve
<code>.github/workflows/</code>	Contiene workflows de GitHub Actions para CI/CD (clean, development, staging).
<code>cmd/seeders</code>	Probablemente incluye scripts para sembrar datos en la base de datos.
<code>config/config.go</code>	Archivo de configuración global de la app (env vars, puertos, etc).
<code>database/mongodb.go</code>	Inicialización y conexión a MongoDB.
<code>docs/</code>	Contiene documentación Swagger (<code>.json</code> , <code>.yaml</code>) y archivo auxiliar.
<code>internals/</code>	Lógica principal dividida en submódulos según entidad o institución.
<code>models/</code>	Definiciones de modelos de datos usados por las APIs y la base de datos.
<code>router/router.go</code>	Inicializa las rutas HTTP del servidor.
<code>utils/utils.go</code>	Funciones auxiliares reutilizables (helpers).
<code>.dockerignore</code>	Archivos que se deben excluir del contexto de Docker.
<code>.env.example</code>	Variables de entorno de ejemplo para configurar localmente.
<code>Dockerfile</code>	Imagen Docker para compilar y ejecutar el backend.
<code>Makefile</code>	Tareas automatizadas (build, lint, run, etc).

Carpeta / Archivo	Descripción breve
<code>README.md</code>	Documentación inicial del proyecto.
<code>go.mod / go.sum</code>	Manejadores de dependencias de Go.
<code>main.go</code>	Punto de entrada principal de la aplicación Go.

Módulos por dominio (internals/handlers y routes)

Institución / Módulo	Handler (<code>handlers/</code>)	Rutas (<code>routes/</code>)	Modelo (<code>models/</code>)	Funcionalidad esperada
Educación	<code>educacion.go</code>	<code>educacion.go</code>	<code>educacion.go</code>	Consultas al Ministerio de Educación
Intrant	<code>intran.go</code>	<code>intran.go</code>	<code>intran.go</code>	Licencias, tránsito y vehículos
JCE	<code>jce.go</code>	<code>jce.go</code>	<code>jce.go</code>	Datos de cédula y registro civil
MESCyT	<code>mescyt.go</code>	<code>mescyt.go</code>	<code>educacion.go</code>	Becas y títulos universitarios
Migración	<code>migracion.go</code>	<code>migration.go</code>	<code>migracion.go</code>	Entrada/salida del país
SIPEN	<code>sipen.go</code>	<code>sipen.go</code>	<code>pension.go</code>	Pensiones, AFP
SISALRIL	<code>sisalril.go</code>	<code>sisalril.go</code>	<code>sisalril.go</code>	Seguro familiar de salud
SNS	<code>sns.go</code>	<code>sns.go</code>	<code>sns.go</code>	Centros de salud, afiliación
Superseguros	<code>superseguros.go</code>	<code>superseguros.go</code>	<code>citizen.go</code> (?)	Información de seguros

Documentación y APIs

Archivo	Propósito
<code>docs/swagger.yaml</code>	Definición OpenAPI de las rutas y modelos
<code>docs.go</code>	Generación o integración de Swagger posiblemente con comentarios Go
<code>README.md</code>	Introducción al proyecto y guía de uso

Tecnología

Elemento	Descripción
Tecnología Backend	Go (Golang)
Framework	Ninguno especificado; se utiliza Go nativo para implementar la lógica del backend.
Base de Datos	MongoDB
Archivos de Configuración	Archivos como <code>config.go</code> y <code>database.go</code> sugieren la configuración de la base de datos y otros parámetros del sistema.
Entorno de Desarrollo	El repositorio contiene un <code>Makefile</code> , <code>Dockerfile</code> , y archivos de configuración como <code>.env.example</code> que permiten configurar y ejecutar el entorno de desarrollo.
Contenedores	El repositorio contiene un <code>Dockerfile</code> que permite construir una imagen Docker para el backend.
CI/CD	El repositorio incluye workflows de CI/CD en <code>.github/workflows/</code> , como <code>clean.yml</code> , <code>development.yml</code> , <code>staging.yml</code> , que gestionan despliegues automáticos.
Dependencias	<code>go.mod</code> y <code>go.sum</code> indican que el proyecto usa Go Modules para gestionar las dependencias del proyecto.
Servicios	El backend parece estar dividido en varios servicios, como <code>citizenapi</code> , con distintos manejadores en <code>handlers</code> que se encargan de diferentes aspectos del servicio.
Documentación	Se incluyen archivos de documentación como <code>swagger.json</code> y <code>swagger.yaml</code> , que pueden describir las APIs del servicio.

DevOps / Automatización

Área	Herramienta/Archivo	Comentarios
CI/CD	<code>.github/workflows</code> (presente)	GitHub Actions configurado (ej. limpieza de caché, tests)
Contenedores	<code>Dockerfile</code>	Contenedor de pruebas configurado
Seguridad	<code>dependabot.yml</code>	Monitoreo automático de versiones y alertas

Análisis de Workflows CI/CD (`.github/workflows/`)

Archivo	Descripción
<code>development.yml</code>	Desplegar entorno de desarrollo en Cloud Run
<code>staging.yml</code>	Desplegar a entorno de preproducción (staging)
<code>clean.yml</code>	Eliminar recursos temporales de PR cerrados

Revisión #18

Creado 4 junio 2025 18:24:26

Actualizado 4 junio 2025 21:09:01