Análisis Técnico del Repositorio: totp-cc

Información General

Nombre del Proyecto: totp-cc

Repositorio: github.com/ogticrd/totp-cc

Este repositorio es una aplicación escrita en **Go (Golang)**, que parece estar diseñada para generar y validar **TOTP (Time-based One-Time Passwords)** para ciudadanos como parte de un sistema de autenticación o verificación de identidad.

Carpeta/Archivo	Descripción
`config/`	Contiene lógica de configuración general (`config.go`) y sus pruebas. Posiblemente gestiona valores de `.env` y configuración de entornos.
`database/`	Lógica de conexión a base de datos, específicamente MongoDB. Incluye pruebas (`mongodb_test.go`).
`docs/`	Archivos Swagger (`swagger.json`, `swagger.yaml`) para la documentación de API, y código Go relacionado (`docs.go`).
`internal/handlers/`	Manejadores HTTP para generación y validación de TOTP (`generate.go`, `validate.go`).
`internal/routes/`	Define las rutas HTTP para los endpoints de generación y validación.
`internal/models/`	Estructuras de datos y lógica del dominio como `citizentotp.go`, validación y estructuras de respuesta.
`pkg/cipher/`	Lógica criptográfica de cifrado AES-128 y pruebas unitarias.
`pkg/totp/`	Implementación central del algoritmo TOTP, con pruebas.
`pkg/utils/`	Funciones auxiliares o utilitarias generales.
`router/`	Define el router principal de la aplicación (`router.go`), usando probablemente un framework como Gin o Echo.

Carpeta/Archivo	Descripción
`.env.example`	Archivo de ejemplo de variables de entorno.
`Dockerfile`	Para construir la imagen Docker de la aplicación.
`docker-compose.dev.yaml`	Compose para entorno de desarrollo. Posiblemente incluye MongoDB.
`Makefile`	Comandos automáticos para tareas comunes como build, test o lint.
`main.go`	Punto de entrada de la aplicación. Suele iniciar servidor, cargar configuración, rutas, etc.

Control de Calidad

Tipo de prueba	Implementado	Observaciones
Unitarias (`*_test.go`)	Sí	Presentes en `pkg/`, `config/`, `database/`.
Integración	No	No se observan pruebas de flujo completo.
Cobertura con herramientas	No	No se detectó integración con coverage tools.

Seguridad

Elemento	Estado / Recomendación
Manejo de secretos	Usa `.env.example`. Se recomienda revisar claves críticas.
Cifrado	AES-128 implementado correctamente.
Variables en entorno	Bien gestionadas, pero falta validación de existencia obligatoria.

Resumen de Arquitectura

Componente	Implementación
Tipo de Aplicación	Microservicio en Go para generación/validación de TOTP.
Almacenamiento	MongoDB.
API	RESTful con documentación Swagger.

Componente	Implementación
Autenticación / Seguridad	Cifrado + TOTP (MFA o validación por usuario).
Despliegue Local	Docker Compose (`.dev.yaml`).
Entrada Principal	`main.go` arranca router + dependencias.

DevOps / Automatización

Elemento	Estado / Observación
GitHub Actions / CI/CD	No se encontró `.github/workflows/` ni ningún pipeline CI/CD configurado.
Docker	Dockerfile para una app en go.
Docker Compose	Existe un archivo para entorno de desarrollo.
Despliegue automatizado	No se identificó ningún script de despliegue automático.

Tecnología

Categoría	Tecnología / Herramienta
Lenguaje	Go (Golang)
Framework web	Fiber
Contenedor	Docker (multi-stage)
Base de imagen	`golang:1.22-alpine`, `scratch`
Formato de build	Go Modules (`go.mod`)

Revisión #13
Creado 2025-06-04 18:24:01 UTC
Actualizado 2025-06-04 21:14:04 UTC