

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

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

Control de Calidad

| Tipo de prueba | Implementado | Observaciones |
|--------------------------------------|--------------|--|
| Unitarias (<code>*_test.go</code>) | Sí | Presentes en <code>pkg/</code> , <code>config/</code> , <code>database/</code> . |
| 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 <code>.env.example</code> . 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 (<code>.dev.yaml</code>). |
| Entrada Principal | <code>main.go</code> arranca router + dependencias. |

DevOps / Automatización

| Elemento | Estado / Observación |
|-------------------------|--|
| GitHub Actions / CI/CD | No se encontró <code>.github/workflows/</code> 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 | <code>golang:1.22-alpine</code> , <code>scratch</code> |
| Formato de build | Go Modules (<code>go.mod</code>) |

Revisión #13

Creado 4 junio 2025 18:24:01

Actualizado 4 junio 2025 21:14:04