Gobackup — Respaldo sencillo de archivos

Logotipo

El contenido generado por IA puede ser incorrecto.

Curso: Laboratorio de Sistemas Operativos II

Universidad Privada de Santa Cruz (UPSA)

Docente: Tito Zúñiga

Integrantes: Mateo Lopez, Henry Lopez, Sebastian Lichtenstein

Fecha de presentación: 15/08/2025

# Resumen Ejecutivo

Gobackup es una herramienta desarrollada en Go para realizar respaldos (copias) de archivos modificados   
desde una carpeta de origen hacia una carpeta de respaldo. Ofrece dos modos de uso: línea de comandos (CLI)   
y un servidor web local con panel visual. El propósito principal es simplificar el respaldo de archivos   
a usuarios sin conocimientos técnicos, detectando cambios y copiando concurrentemente para optimizar el tiempo.  
  
Resuelve el problema de respaldos manuales complejos y poco eficientes, brindando una solución configurable   
y fácil de usar. Las tecnologías principales empleadas son Go 1.24, Gin para el servidor web, Cobra para   
la CLI y HTML/CSS para la interfaz del panel. El alcance incluye funcionamiento multiplataforma,   
configuración por JSON y registro de actividad (logs).

# Introducción

La pérdida de información es un problema recurrente tanto para usuarios domésticos como para empresas.   
Realizar respaldos periódicos ayuda a prevenir esta situación, pero muchas veces las soluciones existentes   
son complejas o requieren conocimientos técnicos avanzados.  
  
Gobackup surge para cubrir esta necesidad, ofreciendo una herramienta simple, configurable y multiplataforma   
para realizar respaldos eficientes. El objetivo general es desarrollar un sistema que permita   
respaldar archivos modificados mediante un proceso concurrente con opción de interfaz web.  
  
Objetivos específicos:  
1. Implementar el modo CLI para respaldos rápidos.  
2. Implementar el modo servidor web para interacción visual.  
3. Configurar parámetros mediante archivo JSON.  
4. Generar registros detallados de cada respaldo.

# Marco Teórico

El proyecto se basa en conceptos como concurrencia, sincronización y manipulación de sistemas de archivos.   
La concurrencia en Go se gestiona con goroutines y canales, optimizando el rendimiento en la copia de múltiples   
archivos. El servidor web se implementa con Gin, un framework HTTP rápido y minimalista. La interfaz de línea   
de comandos se desarrolla con Cobra, que permite definir comandos, subcomandos y parámetros fácilmente.  
  
Fundamentos clave:  
- Lectura y escritura en sistemas de archivos.  
- Arquitectura cliente-servidor.  
- API REST para comunicación entre frontend y backend.  
  
Referencias:  
- Documentación de Go.  
- Manual de Gin y Cobra.

# Metodología

Se utilizó una metodología de desarrollo iterativa incremental, permitiendo avanzar por fases y validar   
funcionalidades a medida que se implementaban.  
  
Fases:  
1. Análisis de requerimientos.  
2. Diseño de arquitectura y componentes.  
3. Implementación de la CLI.  
4. Implementación del servidor web.  
5. Pruebas funcionales y de rendimiento.  
6. Documentación final.  
  
Herramientas utilizadas:  
- VS Code como entorno de desarrollo.  
- Git y GitHub para control de versiones.  
- Go Modules para manejo de dependencias.

# Desarrollo del Proyecto

La arquitectura del sistema está compuesta por dos modos: CLI y Web.   
El CLI ejecuta respaldos directamente en la terminal, mientras que el servidor web ofrece   
un panel visual con progreso y estadísticas.  
  
Componentes principales:  
- CLI (`cli.go`): escaneo de archivos y copia concurrente.  
- Web (`web.go`): servidor Gin, rutas `/status` y `/backup`.  
- Backup: lógica de respaldo, detección de cambios.  
- Logger: registro de eventos y errores.  
  
Se emplean goroutines para la copia concurrente de archivos, controladas por un límite   
de concurrencia configurable. Las pruebas incluyeron respaldos de carpetas pequeñas y medianas,   
midiendo tiempos y verificando integridad de los datos.

# Resultados

- Capturas: ejecución en CLI mostrando archivos copiados y panel web con barra de progreso.  
- Métricas: tiempos de copia reducidos gracias a la concurrencia.  
- Análisis: el uso de concurrencia optimiza el rendimiento; la interfaz web facilita el uso   
por usuarios sin conocimientos técnicos.

# Conclusiones

Se logró implementar un sistema de respaldo funcional con dos modos de uso, configuración flexible y   
registro detallado. Se aprendió a manejar concurrencia en Go, implementar APIs REST con Gin y diseñar   
interfaces CLI con Cobra.  
  
Limitaciones: el sistema no incluye compresión de archivos ni respaldo remoto.

# Recomendaciones

- Agregar soporte para compresión antes de copiar.  
- Añadir opción de sincronización incremental.  
- Integrar con almacenamiento en la nube.

# Bibliografía

- Go Documentation: https://go.dev/doc/  
- Gin Documentation: https://gin-gonic.com/docs/  
- Cobra Documentation: https://cobra.dev/

# Anexos

- Código fuente disponible en GitHub: https://github.com/Henry-Lopez/Gobackup  
- Diagramas de arquitectura

Inicio

↓

Cargar configuración (archivos YAML/JSON, variables de entorno)

↓

Inicializar logger y reportes

↓

Verificar rutas de backup y recursos disponibles

↓

Iterar sobre tareas de backup configuradas

↓

Para cada tarea:

- Preparar fuente y destino

- Ejecutar copia de seguridad

- Registrar resultados y errores

↓

Finalizar backup y generar resumen

↓

Enviar notificaciones (opcional: email, Slack, etc.)

↓

Fin

- Resultados de pruebas de rendimiento. Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.  
Imagen de la pantalla de un computador

El contenido generado por IA puede ser incorrecto.