Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Sistemas Operativos 2 Sección A

Primer Semestre 2023



**Proyecto**

**Monitoreo de memoria y manejo de puertos USB**

**Manual Técnico**

Nombre: William Alejandro Borrayo Alarcón

Carné: 201909103

Contenido

[Descripción de la solución 2](#_Toc128088687)

[Requerimientos mínimos del entorno de desarrollo 2](#_Toc128088688)

[Diccionario de clases 3](#_Toc128088689)

[Diccionario de métodos y funciones 4](#_Toc128088690)

[Comandos utilizados 4](#_Toc128088691)

[Diagramas que ilustran el flujo funcional del programa 5](#_Toc128088692)

[Bibliografía 6](#_Toc128088693)

# Descripción de la solución

La aplicación de escritorio fue desarrollada con Wails, porque permite proporcionar una vista amigable al usuario y obtener información relacionada a los recursos del sistema mediante instrucciones y procedimientos en Golang. Wails proporciona la comunicación correcta entre la interfaz y el código funcional de Golang, mostrando detalles como el porcentaje de uso de CPU, el espacio de disco ocupado, el espacio en disco disponible, y el espacio de disco total en tiempo real, esta comunicación en tiempo real se realiza mediante eventos, con las instrucciones Event Emit desde Golang para enviar un mensaje de actualización y Event On desde la interfaz para que funcione como un Listener. La interfaz fue realizada con el Framework React, porque permite crear diseños muy llamativos y fáciles de implementar. El porcentaje de uso de la memoria RAM se muestra mediante barras de progreso circulares, que cambian su color a celeste, anaranjado o rojo dependiendo a qué tercera parte del 100% tienen su valor, respectivamente.

# Requerimientos mínimos del entorno de desarrollo

* Una distribución de Linux Ubuntu (Ubuntu 22.04.2 LTS recomendado).
* Golang en versión compatible con Wails (go1.20 linux/amd64 recomendado)
* Node.js en versión compatible con Wails (v19.7.0 recomendado)
* Wails CLI (v2.3.1 recomendado)
* IDE/Editor de código (Visual Studio Code Recomendado)
* Procesador: Intel Celeron Processor N4000
* Memoria RAM: 4 GB
* Espacio de disco duro disponible: 6 GB

En equipos con características similares o mejores debe funcionar sin problema.

# Diccionario de clases

## Administrador de tareas

1. **App.css:** En este archivo de la carpeta frontend/src se colocaron estilos para los diferentes componentes utilizados en App.jsx
2. **app.go:** Este archivo generado automáticamente con el comando “wails init” contiene métodos y funciones importantes con Wails como lo es “startup” que se ejecuta al iniciar la aplicación, también estas funciones pueden ser llamadas desde el frontend.
3. **App.jsx:** En este archivo de la carpeta frontend/src se creó todo el diseño de la interfaz gráfica.
4. **main.go:** Este archivo generado automáticamente con el comando “wails init” es donde se ejecuta el código de Golang necesario.
5. **memoria.go:** Este archivo contiene funciones y la estructura necesaria para obtener detalles del uso de memoria RAM en el sistema. La estructura tiene los campos a leer respectivos son enviados a la interfaz gráfica mediante el formato JSON.

Estructura:

Texto

Descripción generada automáticamente

## Administrador de puertos USB

1. **bitácora:** Este archivo contiene funciones y estructuras necesarias para la correcta lectura y escritura del archivo de bitácora, que almacena el historial de archivos copiados hacia y desde el USB.
2. **main.go:** Este es el archivo principal para la ejecución de este programa, aquí se realiza la lectura de los dispositivos USB conectados al computador, se comparan los archivos con los que están la computadora, y se comparan fechas de creación para saber el origen y destino de los archivos copiados.

# Diccionario de métodos y funciones

1. **compararTiempo:** Esta función del archivo AdministradorUSB/main.go se encarga de comparar los tiempos de creación de dos archivos para deducir cuál fue creado antes.
2. **obtenerArchivosUSB:** Esta función del archivo AdministradorUSB/main.go se encarga de listar los archivos que están los dispositivos USB conectados al computador, incluyendo carpetas y subcarpetas.
3. **comprararArchivos:** Esta función del archivo AdministradorUSB/main.go compara el contenido de dos archivos para saber si son iguales o no.
4. **verificarArchivosCopiados:** Esta función del archivo AdministradorUSB/main.go inicia todo el proceso de verificación de archivos copiados
5. **AgregarBitacora:** Este método del archivo AdministradorUSB/bitácora/bitacora.go se encarga de agregar un nuevo log al archivo de bitácora.
6. **getMemoriaInfo:** Esta función del archivo memoria.go se encarga de obtener la información del uso de la memoria RAM en el sistema.

# Comandos utilizados

1. **Instalación de Golang**

sudo apt-get update

sudo apt-get -y upgrade

mkdir temp

cd temp

wget <https://dl.google.com/go/go1.20.linux-amd64.tar.gz>

sudo tar -xvf go1.20.linux-amd64.tar.gz

sudo mv go /usr/local

export GOROOT=/usr/local/go

export GOPATH=$HOME/go

export PATH=$GOPATH/bin:$GOROOT/bin:$PATH

source ~/.profile

1. **Instalación de Node.js**

sudo apt install curl

cd ~

curl -sL https://deb.nodesource.com/setup\_16.x | sudo bash -

cat /etc/apt/sources.list.d/nodesource.list

sudo apt -y install nodejs

1. **Instalación de Wails**

go install [github.com/wailsapp/wails/v2/cmd/wails@latest](mailto:github.com/wailsapp/wails/v2/cmd/wails@latest)

- comandos mostrados por “wails doctor” para ver qué otras herramientas son necesarias.

1. **Uso de Wails**

Crear un proyecto con React como Framework de frontend:

wails init -n proyecto -t preact

Ejecutar el Proyecto en modo de desarrollo:

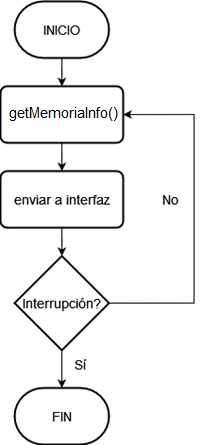
wails dev

Crear ejecutable de la aplicación:

wails build

# Diagramas que ilustran el flujo funcional del programa

1. **Lectura de datos de CPU y Disco Duro**

****

1. **Bloquear o desbloquear puertos USB**

**Diagrama

Descripción generada automáticamente**

1. **Analizar archivos copiados**

**Diagrama

Descripción generada automáticamente**

# Bibliografía

1. Harnanta, K. J. (2022, 30 marzo). How to monitor your CPU and Memory usage in MacOS with Go Programming Language. Medium. <https://medium.com/macoclock/how-to-monitor-your-cpu-and-memory-usage-in-macos-with-go-programming-language-2e9537178d78>
2. Get amount of free disk space using Go. (2013, 20 noviembre). Stack Overflow. <https://stackoverflow.com/questions/20108520/get-amount-of-free-disk-space-using-go>
3. R. (2023, 3 febrero). *Installing Go 1.19 on Ubuntu 22.04 | Level Up Coding*. Medium. <https://levelup.gitconnected.com/installing-go-on-ubuntu-b443a8f0eb55>
4. B. (s. f.). GitHub - benjamin-thomas/wails-async-demo2. GitHub. <https://github.com/benjamin-thomas/wails-async-demo2>