

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Sistemas Operativos 2 Sección A

Primer Semestre 2023



Nombre: William Alejandro Borrayo Alarcón

Carné: 201909103

Contenido

Descripción de la solución.....	2
Requerimientos mínimos del entorno de desarrollo.....	2
Diccionario de clases	3
Diccionario de métodos y funciones	4
Comandos utilizados	4
Diagramas que ilustran el flujo funcional del programa.....	5
Bibliografía	6

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. Los porcentajes de uso de CPU y de disco son mostrados 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

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. **recursos.go:** Este archivo contiene funciones y estructuras necesarias para obtener detalles de los recursos de hardware del sistema. Las estructuras tienen los campos a leer respectivos al recurso y son enviados a la interfaz gráfica mediante el formato JSON.

Estructuras:

```
// CPU struct
type CPU struct {
    Porcentaje int `json:"Porcentaje"`
    Nucleos int `json:"Nucleos"`
}

// Disco struct
type Disco struct {
    Porcentaje int `json:"Porcentaje"`
    Usado string `json:"Usado"`
    Disponible string `json:"Disponible"`
    Total string `json:"Total"`
}

// Recursos struct
type Recursos struct {
    CPU CPU `json:"CPU"`
    Disco Disco `json:"Disco"`
}
```

Diccionario de métodos y funciones

1. **colorPorcentaje:** Esta función del archivo frontend/src/App.jsx se encarga de retornar el color correspondiente al tercio del porcentaje sobre 100 del valor recibido, siendo celeste, naranja o rojo respectivamente.
2. **getCPUInfo:** Esta función del archivo recursos.go se encarga de obtener el porcentaje de uso y los núcleos del CPU del sistema.
3. **getDiskInfo:** Esta función del archivo recursos.go se encarga de obtener el porcentaje de uso del dispositivo de almacenamiento secundario, también la cantidad de espacio disponible, utilizado y total.
4. **getRecursos:** Esta función del archivo recursos.go llama a las funciones getCPUInfo y getDiskInfo para enviarlas a la interfaz gráfica, por medio de un evento.
5. **startup:** Esta función del archivo app.go se encarga de obtener la información de los recursos constantemente y se ejecuta al iniciar la aplicación.

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
```

2. 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
```

3. Instalación de Wails

go install github.com/wailsapp/wails/v2/cmd/wails@latest

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

4. 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:

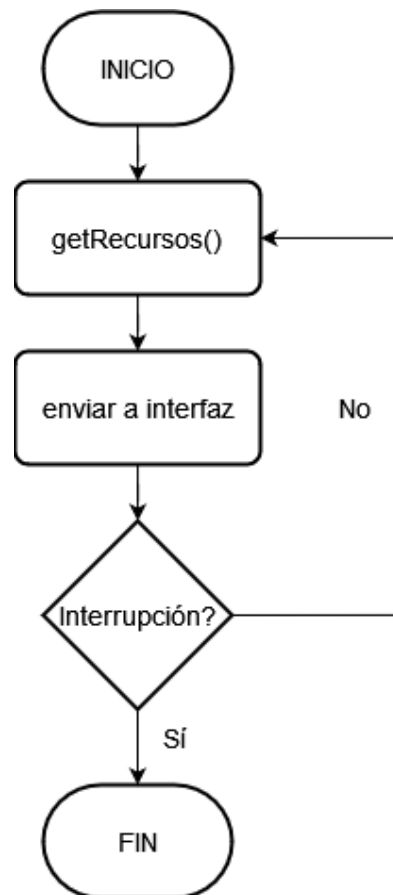
```
wails dev
```

Crear ejecutable de la aplicación:

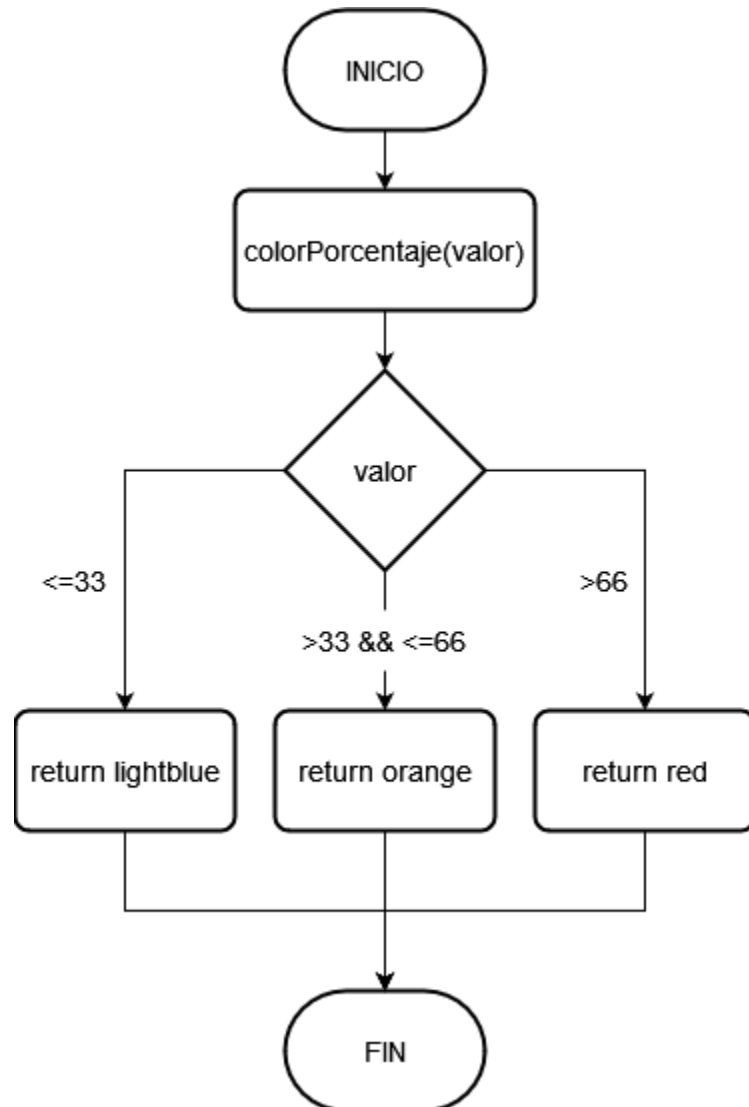
```
wails build
```

Diagramas que ilustran el flujo funcional del programa

1. Lectura de datos de CPU y Disco Duro



2. Color correspondiente al porcentaje



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>