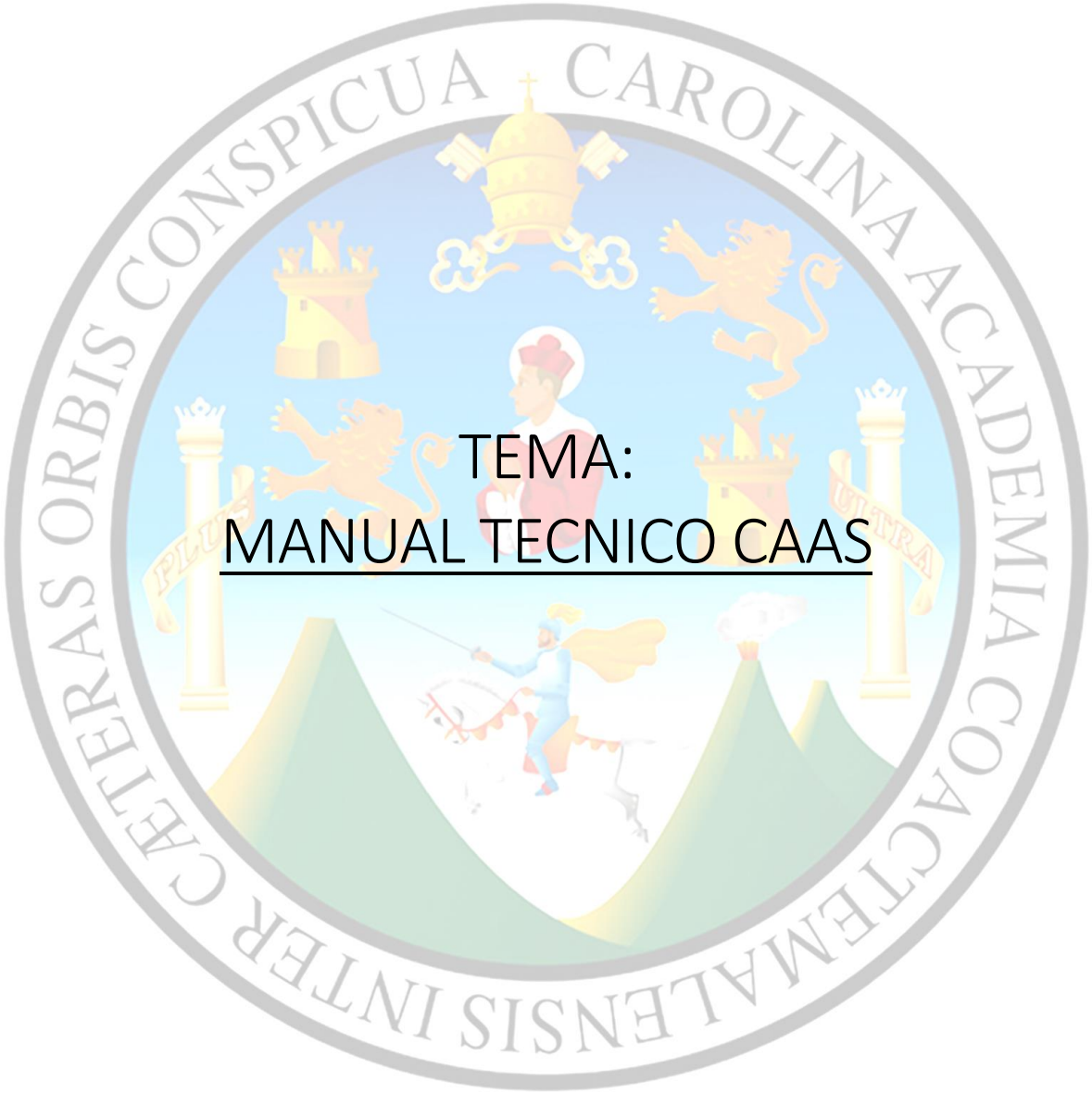
UNIVERSIDAD DE SAN CARLOS DE GUTEMALA
FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS
ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 2
PRIMER SEMESTRE 2019
ING. BYRON LÓPEZ



TEMA:
MANUAL TECNICO CAAS

NOMBRE: Christian Adolfo Real Ixcayau CARNET: 201513700

DESCRIPCION DE LA APLICACIÓN

Se trata de desarrollar una aplicación que permita monitorear y gestionar los procesos de un servidor Linux, por medio de una interfaz web de fácil acceso desde el navegador de una computadora o de un dispositivo móvil como teléfono o Tablet.

Funcionalidades:

1. Grafica de recursos: La aplicación web permite visualizar gráficas dinámicas que muestren el uso del CPU y de la memoria RAM del servidor.
2. Administrador de procesos: La aplicación web permite mostrará la información básica de los procesos que se ejecutan como los que se describen a continuación:
    a. Número total de procesos: la cantidad total de los procesos registrados.
    b. Procesos en ejecución: la cantidad de procesos en estado de ejecución (running)
    c. Procesos suspendidos: la cantidad de procesos en estado suspendido (sleeping)
    d. Procesos detenidos: la cantidad de procesos en estado detenido (stoped)
    e. Procesos zombie: la cantidad de procesos en estado zombie
    f. y permite terminar los procesos(kill) que se encuentran en ejecución.

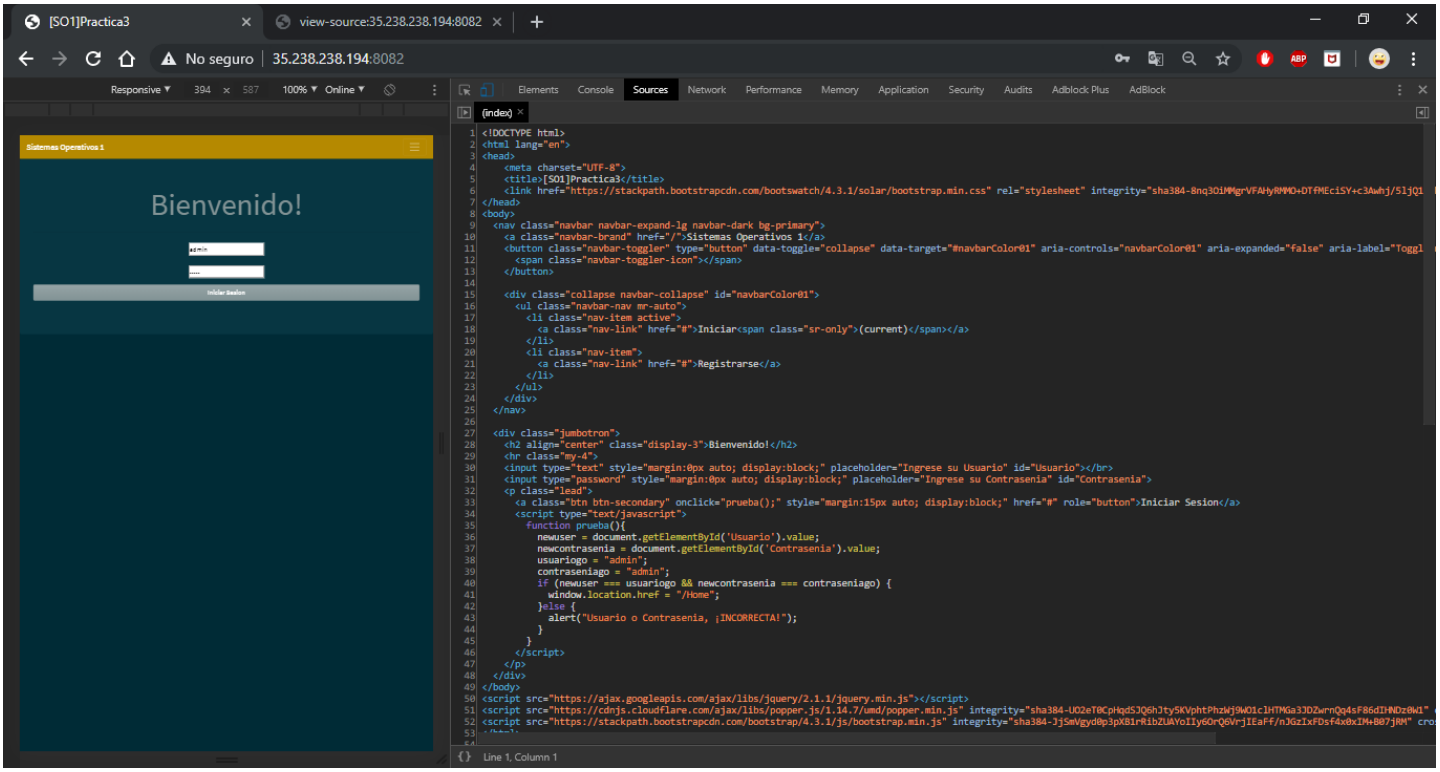
Características

1. La solución web debe ser desarrollada utilizando golang (GO).
2. La solución web debe contar con:
    a. una página de login, en donde solamente será válido usuario admin.
    b. Una vez validada la sesión, el panel de control tendrá tres secciones:
        i. Listar los procesos en el sistema
        ii. Graficar el uso de CPU
        iii. Graficar el uso de memoria RAM

LOGIN

Para hacer responsiva la página y darle una vista profesional se decide utilizar una plantilla que nos provee estas características, haciendo uso de un CDN en el que se encuentran alojados todos los archivos que permiten que esta plantilla funcione.

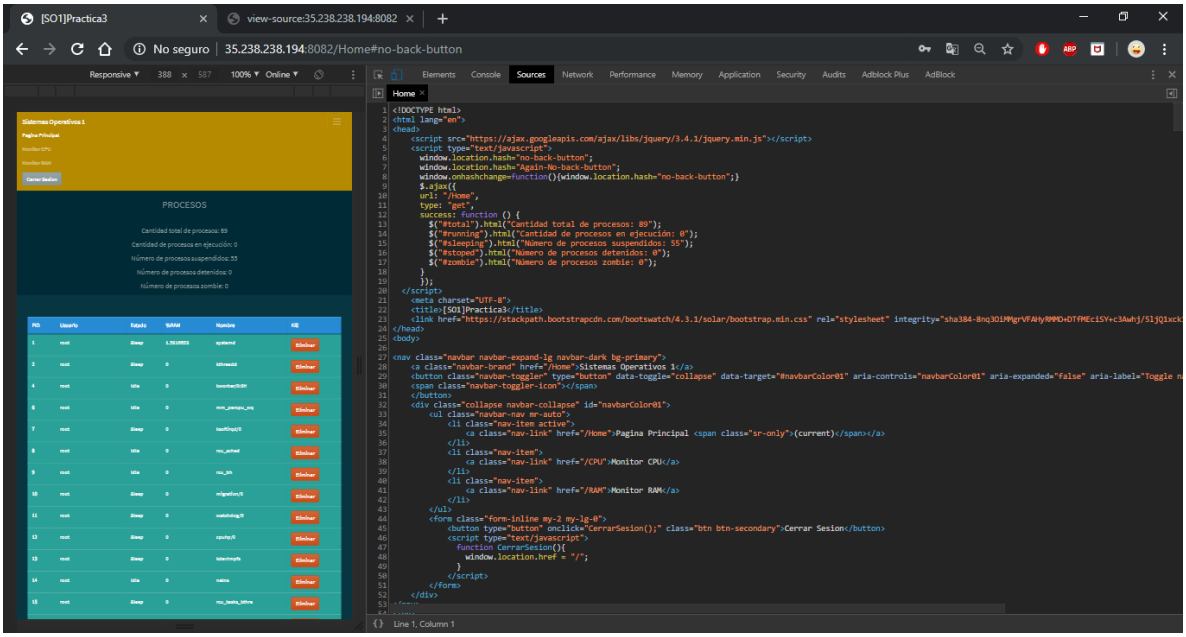
En la parte del html se realizó la validación de que el usuario permitido por default era admin y que su contraseña es admin



PAGINA PRINCIPAL

Para hacer responsiva la página y darle una vista profesional se decide utilizar una plantilla que nos provee estas características, haciendo uso de un CDN en el que se encuentran alojados todos los archivos que permiten que esta plantilla funcione.

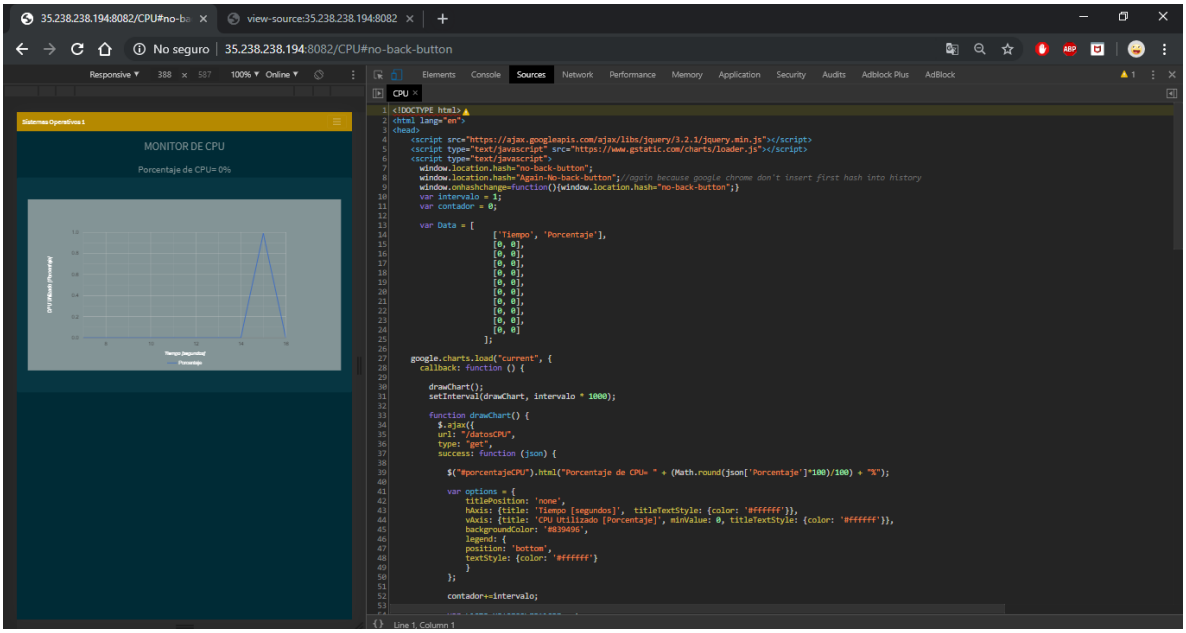
En este html se cuenta con un método Ajax que obtiene de parte del backend la información de los procesos y sus detalles que luego pasaran a ser renderizadas de forma automática con el contenido de estas.



MONITOR CPU

Para hacer responsiva la página y darle una vista profesional se decide utilizar una plantilla que nos provee estas características, haciendo uso de un CDN en el que se encuentran alojados todos los archivos que permiten que esta plantilla funcione.

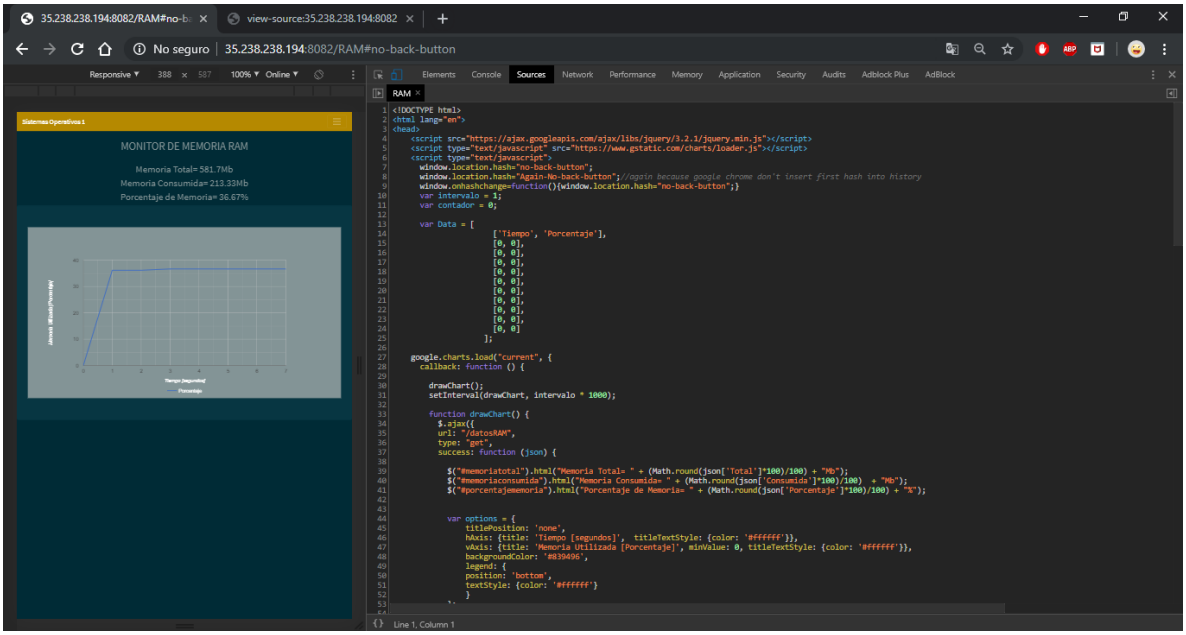
Este html cuenta con un método Ajax al igual que la vista anterior pero con la variante que esta petición se realiza de manera constante para que se vea que la grafica es en tiempo real del uso del CPU.



MONITOR RAM

Para hacer responsiva la página y darle una vista profesional se decide utilizar una plantilla que nos provee estas características, haciendo uso de un CDN en el que se encuentran alojados todos los archivos que permiten que esta plantilla funcione.

Este html cuenta con un método Ajax al igual que la vista anterior pero con la variante que esta petición se realiza de manera constante para que se vea que la gráfica es en tiempo real del uso de la memoria RAM.



BACKEND

Para proceder a realizar el backend se procedió a definir las rutas para esta ocasión se definieron estas:

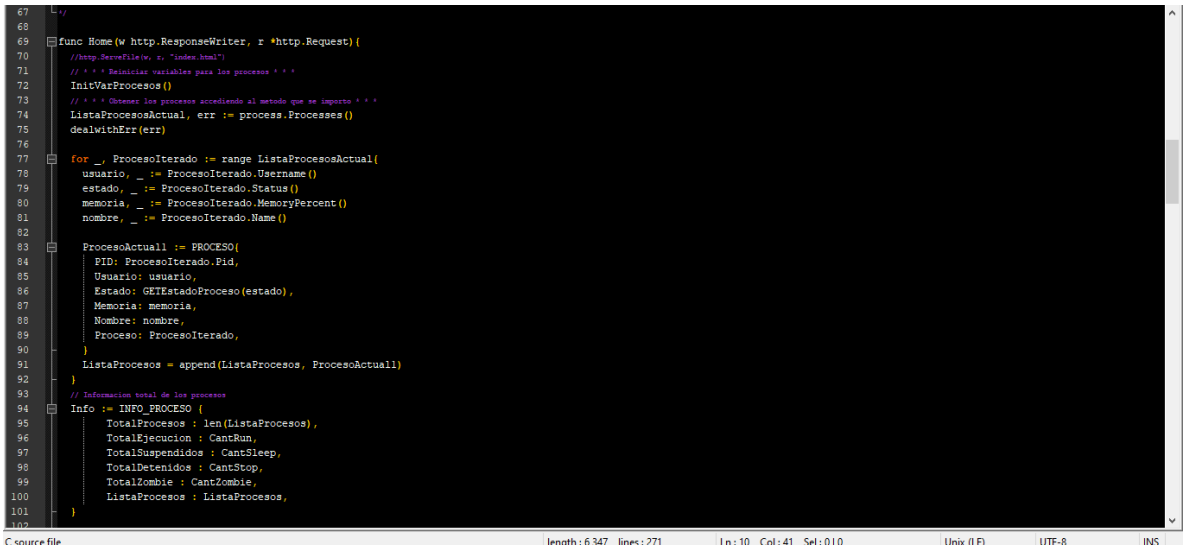
1. / : Aquí se encuentra alojado el login
2. /Home : Aquí se encuentra alojado el menú principal
3. /CPU : Aquí se encuentra alojado el uso del cpu
4. /RAM : Aquí se encuentra alojado el uso de la ram

Todos los servicio se acceden por medio del puerto 8082 que fue el que se definió dentro del código fuente

```
package main
/**
' -------------------IMPORTACIONES-----------------
*/
import (
    "encoding/json"
    "github.com/gorilla/mux"
    "html/template"
    "net/http"
    "github.com/shirou/gopsutil/process"
    "github.com/guillermo/go.procmeminfo"
    "github.com/shirou/gopsutil/cpu"
    "math"
    "log"
    "fmt"
    "regexp"
    "strconv"
)
var router = mux.NewRouter()
/**
' -------------------ESTRUCTURAS-----------------
*/
type USUARIOS struct{
  Usuario string
  Contrasenia string
}

type PROCESO struct{
  PID int32
  Usuario string
  Estado string
  Memoria float32
  Nombre string
  Proceso *process.Process
}
```

C source file | length : 6,347 lines : 271 | Ln : 10 Col : 41 Sel : 0 | 0 | Unix (LF) | UTF-8 | INS



```
*/

func Home(w http.ResponseWriter, r *http.Request){
  //http.ServeFile(w, r, "index.html")
  // * * * Reiniciar variables para los procesos * * *
  InitVarProcesos()
  // * * * Obtener los procesos accediendo al metodo que se importo * * *
  ListaProcesosActual, err := process.Processes()
  dealwithErr(err)

  for _, ProcesoIterado := range ListaProcesosActual{
    usuario, _ := ProcesoIterado.Username()
    estado, _ := ProcesoIterado.Status()
    memoria, _ := ProcesoIterado.MemoryPercent()
    nombre, _ := ProcesoIterado.Name()

    ProcesoActual1 := PROCESO{
      PID: ProcesoIterado.Pid,
      Usuario: usuario,
      Estado: GETEstadoProceso(estado),
      Memoria: memoria,
      Nombre: nombre,
      Proceso: ProcesoIterado,
    }
    ListaProcesos = append(ListaProcesos, ProcesoActual1)
  }
  // Informacion total de los procesos
  Info := INFO_PROCESO {
        TotalProcesos : len(ListaProcesos),
        TotalEjecucion : CantRun,
        TotalSuspendidos : CantSleep,
        TotalDetenidos : CantStop,
        TotalZombie : CantZombie,
        ListaProcesos : ListaProcesos,
  }
```

C source file | length : 6,347 lines : 271 | Ln : 10 Col : 41 Sel : 0 | 0 | Unix (LF) | UTF-8 | INS

DESCRIPCION DE HERRAMIENTAS UTILIZADAS

GOLANG [BACKEND]

Golang es un lenguaje de programación moderno conocido por su escalabilidad, concurrencia y verificación de errores, la creación de este lenguaje se basa en tres Python, Java, C++.

Su principal característica es ser un lenguaje de programación compilado de tipado estático diseñado por Google, sintácticamente similar a C pero con énfasis en una mayor simplicidad y seguridad. Algunas de las características con las que cuentas y razón para distinguir entre los demás lenguajes de programación es la seguridad de memoria, garbage collector, concurrencia de comunicación secuencial de procesos.

HTML [FRONTEND]

El lenguaje de marcado de hipertexto (HTML) es el lenguaje de marcado estándar para documentos diseñados para mostrarse en un navegador web . Puede ser asistido por tecnologías como hojas de estilo en cascada (CSS) y lenguajes de script como JavaScript .

GOOGLE CHART [GRAFICAS]