

Edier Santiago Moyano

Aprendices



Instructor:
CESAR ARTURO ESQUIVEL CORTES

SENA SERVICIO NACIONAL DE APRENDIZAJE

ADSO 2901879

IBAGUÉ – TOLIMA

Centro De Industria y la construcción

Contenido

Preparación	3
Creación de la estructura del proyecto	3
Compilación del proyecto	8
Ejecución de la aplicación	8

Preparación

Requisitos previos

- Go instalado (versión 1.16 o superior)
- Un editor de texto (como Visual Studio Code, Sublime Text, etc.)
- Una terminal o línea de comandos

Verificar la instalación de Go

1. Abre una terminal (Command Prompt o PowerShell en Windows)
2. Escribe el siguiente comando y presiona Enter:

go version

3. Deberías de ver algo como :

go version go1.21.0 windows/amd64

Creación de la estructura del proyecto

Crear el directorio principal

1. Abre una terminal (win+r CMD)
2. Navega a donde quieras crear el proyecto (por ejemplo, tu Escritorio)

```
C:\Users\edier>cd C:\xampp\htdocs
```

3. Crea un nuevo directorio para el proyecto:

```
cs>mkdir calculadora-web-go
```

4. Entra en ese directorio:

```
htdocs>cd calculadora-web-go
```

Inicializar el modulo Go

1. 1.Dentro del directorio del proyecto ejecuta:

```
\calculadora-web-go>go mod init ejemplo/calculadora-web
```

Esto crea un archivo go.mod en su directorio

Crear la estructura de carpetas

1. Crea un directorio para las plantillas HTML:

```
o\htdocs\calculadora-web-go>mkdir templates
```

2. Crea un archivo en el apartado principal llamado [main.go](#) y pon este codigo:

```
package main

import (
    "fmt"
    "html/template"
    "log"
    "net/http"
    "strconv"
)

// Estructura para pasar datos a la plantilla HTML
type CalculadoraData struct {
    Num1          string
    Num2          string
    Operacion     string
    OperacionSimbolo string
    Resultado     float64
    MostrarResultado bool
    Error         bool
    MensajeError  string
}

func main() {
    // Cargar la plantilla HTML
    tpl, err := template.ParseFiles("templates/index.html")
    if err != nil {
        log.Fatalf("Error al cargar la plantilla: %v", err)
    }

    // Manejador para la página principal
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        data := CalculadoraData{
            MostrarResultado: false,
        }
        tpl.Execute(w, data)
    })

    // Manejador para procesar el cálculo
    http.HandleFunc("/calcular", func(w http.ResponseWriter, r *http.Request) {
        // Verificar que el método sea POST
        if r.Method != http.MethodPost {
```

```

// Obtener los valores del formulario
num1Str := r.FormValue("num1")
num2Str := r.FormValue("num2")
operacion := r.FormValue("operacion")

// Convertir los valores a números
num1, err1 := strconv.ParseFloat(num1Str, 64)
num2, err2 := strconv.ParseFloat(num2Str, 64)

// Preparar los datos para la respuesta
data := CalculadoraData{
    Num1:          num1Str,
    Num2:          num2Str,
    Operacion:     operacion,
    MostrarResultado: true,
}

// Verificar errores de conversión
if err1 != nil || err2 != nil {
    data.Error = true
    data.MensajeError = "Por favor, ingresa números válidos."
    tpl.Execute(w, data)
    return
}

// Realizar la operación seleccionada
switch operacion {
case "suma":
    data.Resultado = num1 + num2
    data.OperacionSimbolo = "+"
case "resta":
    data.Resultado = num1 - num2
    data.OperacionSimbolo = "-"
case "multiplicacion":
    data.Resultado = num1 * num2
    data.OperacionSimbolo = "x"
case "division":
    if num2 == 0 {
        data.Error = true
        data.MensajeError = "No se puede dividir por cero."
        tpl.Execute(w, data)
        return
    }
}

```

```

        data.MensajeError = "No se puede dividir por cero."
        tmpl.Execute(w, data)
        return
    }
    data.Resultado = num1 / num2
    data.OperacionSimbolo = "÷"
default:
    data.Error = true
    data.MensajeError = "Operación no válida."
}

// Mostrar el resultado
tmpl.Execute(w, data)
})

// Iniciar el servidor web
fmt.Println("Servidor iniciado en http://localhost:8080")
log.Fatal(http.ListenAndServe(":8080", nil))
}

```

3. Dentro de la carpeta templates crea el archivo index.html e inserta el siguiente código:

```

<body>
<h1>Calculadora Web con Go</h1>

<div class="calculator">
<form method="POST" action="/calcular">
    <div class="form-group">
        <label for="num1">Primer número:</label>
        <input type="number" id="num1" name="num1" value="{{.Num1}}" required step="any">
    </div>

    <div class="form-group">
        <label for="operacion">Operación:</label>
        <select id="operacion" name="operacion">
            <option value="suma" {{if eq .Operacion "suma"}}selected{{end}}>Suma (+)</option>
            <option value="resta" {{if eq .Operacion "resta"}}selected{{end}}>Resta (-)</option>
            <option value="multiplicacion" {{if eq .Operacion "multiplicacion"}}selected{{end}}>Multiplicación (×)</option>
            <option value="division" {{if eq .Operacion "division"}}selected{{end}}>División (÷)</option>
        </select>
    </div>

    <div class="form-group">
        <label for="num2">Segundo número:</label>
        <input type="number" id="num2" name="num2" value="{{.Num2}}" required step="any">
    </div>

    <button type="submit">Calcular</button>
</form>

    {{if .MostrarResultado}}
    <div class="result" {{if .Error}}error{{end}}>
        {{if .Error}}
        <p><strong>Error:</strong> {{.MensajeError}}</p>
        {{else}}
        <p><strong>Resultado:</strong> {{.Resultado}}</p>
        <p>{{.Num1}} {{.OperacionSimbolo}} {{.Num2}} = {{.Resultado}}</p>
        {{end}}
    </div>
    {{end}}
</div>
</body>

```

estos son los estilos:

```

body {
  font-family: Arial, sans-serif;
  max-width: 600px;
  margin: 0 auto;
  padding: 20px;
  line-height: 1.6;
}
h1 {
  color: #333;
  text-align: center;
}
.calculator {
  background-color: #f5f5f5;
  border-radius: 8px;
  padding: 20px;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
}
.form-group {
  margin-bottom: 15px;
}
label {
  display: block;
  margin-bottom: 5px;
  font-weight: bold;
}
input[type="number"], select {
  width: 100%;
  padding: 8px;
  border: 1px solid #ddd;
  border-radius: 4px;
  box-sizing: border-box;
}
button {
  background-color: #4CAF50;
  color: white;
  padding: 10px 15px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 16px;
}
button:hover {
  background-color: #45a049;
}
.result {
  margin-top: 20px;
  padding: 15px;
  background-color: #e9f7ef;
  border-radius: 4px;
  border-left: 5px solid #4CAF50;
}
.error {
  background-color: #ffebee;
  border-left: 5px solid #f44336;
}

```

Verifica la estructura del proyecto

```
calculadora-web-go/  
├─ main.go  
├─ go.mod  
└─ templates/  
    └─ index.html
```

Compilación del proyecto

1. Asegúrate de estar en el directorio principal del proyecto (`calculadora-web-go`)
2. Ejecuta el siguiente comando:

```
s\calculadora-web-go>go build -o calculadora-web.exe
```

Este comando creará un archivo ejecutable llamado `calculadora-web.exe` en el directorio actual

Verifica que el archivo se haya creado correctamente:

```
ra-web-go>dir calculadora-web.exe  
label
```

Ejecución de la aplicación

Iniciar el servidor web

1. Asegúrate de estar en el directorio principal del proyecto
2. Ejecuta el archivo compilado:

```
ra-web-go>calculadora-web.exe
```

3. Deberías de ver el siguiente mensaje en la terminal:


```
Servidor iniciado en http://localhost:8080
```

****IMPORTANTE**:** Mantén esta ventana de terminal abierta mientras uses la aplicación. No la cierres.

Acceder a la aplicación web

1. Abre tu navegador web preferido (Chrome, Firefox, Edge, Opera, etc.)
2. En la barra de direcciones, escribe: `http://localhost:8080`