

Practica 1

Sistemas Distribuidos

Grado en Ingeniería Telemática

Introducción a GO

Introducción

La paquetería de GO estará disponible en los ordenadores del laboratorio. Si lo deseas en tu ordenador sigue estos pasos:

- Ve a la página <https://golang.org/dl/>
- Descarga el binario de tu SO
- Haz doble clic en el instalador y clic en Ejecutar
- Una vez finalizada la instalación, puedes verificarla abriendo el terminal y escribiendo

```
$ go version
```

Go es muy diferente a otros lenguajes orientados a objetos. Esta respaldado por Google y lo usan grandes compañías como IBM, Intel, Medium y Adobe:

<https://github.com/golang/go/wiki/GoUsers>

Compilación y ejecución

Para compilar un código llamado *fichero* en GO pueden usar:

```
$ go build fichero.go
```

Si solo se quiere ejecutar sin que se guarde el archivo ejecutable:

```
$ go run fichero.go
```

Códigos

En esta sección se tiene que realizar las describir los siguientes códigos hecho en GO para ser representados en un diagrama **UML** (diagramas de flujo, diagrama de actividades, diagramas de clases o diagrama de casos de uso) y en **pseudocódigo**

NOTA

Los diagramas de UML deben estar hechos en algun software de ordenador. No se admitira ningún trabajo hecho a mano. Se recomienda usar draw.io o Dia, los dos instalados en los ordenadores del laboratorio.

1. Describir el siguiente código en un diagrama UML y un pseudocódigo

```
package main
import "fmt"
func main() {
    const LENGTH int = 10
    const WIDTH int = 5
    var area int
    area = LENGTH * WIDTH
    fmt.Printf("Valor del área: %d", area)
}
```

2. Describir el siguiente código en un diagrama UML y un pseudocódigo

```

package main
import "fmt"
func main() {
    var grade string = "B"
    var marks int = 90
    switch marks {
        case 90: grade = "A"
        case 80: grade = "B"
        case 50, 60, 70: grade = "C"
        default: grade = "D"
    }
    switch {
        case grade == "A" :
            fmt.Printf("Excellent!\n")
        case grade == "B", grade == "C" :
            fmt.Printf("Well done\n")
        case grade == "D":
            fmt.Printf("You passed\n")
        case grade == "F":
            fmt.Printf("Better try again\n")
        default:
            fmt.Printf("Invalid grade\n")
    }
    fmt.Printf("Your grade is %s\n", grade)
}

```

3. Describir el siguiente código en un diagrama UML y un pseudocódigo

```

package main
import "fmt"
func sum(nums ...int) {
    fmt.Print(nums, " ")
    total := 0
    for _, num := range nums {
        total += num
    }
    fmt.Println(total)
}
func main() {
    sum(1, 2)
    sum(1, 2, 3)
    nums := []int{1, 2, 3, 4}
    sum(nums...)
}

```

4. Describir el siguiente código en un diagrama UML y un pseudocódigo

```

package main
import "fmt"
func main() {
    var a[5] int
    fmt.Println("emp:", a)
    a[4] = 100
    fmt.Println("set:", a)
    fmt.Println("get:", a[4])
    fmt.Println("len:", len(a))
    b := [5] int{1, 2, 3, 4, 5}
    fmt.Println("dcl:", b)
    var twoD [2][3] int
    for i := 0; i < 2; i++ {
        for j := 0; j < 3; j++ {
            twoD[i][j] = i + j
        }
    }
    fmt.Println("2d: ", twoD)
}

```

Programas

Realizar el código en GO, el diagrama de flujo y pseudocódigo

1. Implementa una función para inicializar las posiciones del array
2. Implementa una función para imprimir las posiciones del array
3. Implementa una función que devuelva la media
4. Implementa una función que devuelva el valor máximo
5. Implementa una función que devuelva el valor mínimo
6. Implementa una función que dado un valor, devuelva la posición del array
7. Escribe un programa en el que te declares un array y desde el invoques a las anteriores funciones a modo de menú

• Indicaciones adicionales

Elabora un archivo en LibreOffice Writer con las soluciones de los enunciados anteriores.

Guarda el archivo con un nombre como `Practica_1_Nombre_SS00_dist.odt` . Donde nombre ha de ser tu nombre de usuario.

Luego convierte ese documento a formato PDF en el menu exportar del software, y subelo a la actividad en aula virtual.