

Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías



## Actividad 3. Goroutines

Suárez Álvarez Sergio

Sistemas concurrentes y distribuidos

Sección D02

Código del alumno: 217758497

Ingeniería en Computación

Profesor: Davalos Boites Michel

Fecha de entrega: 21 de abril de 2021

# Código

```
package main

import (
    "fmt"
    "time"
)

type Admin struct {
    Processes []Process
    exits []chan bool
    counter uint64
    c chan bool
    isShowingAll bool
}

type Process struct {
    id uint64
    quit chan bool
}

func (a *Admin) AddProcess() {
    exit := make(chan bool)
    process := Process{a.counter, make(chan bool)}
    a.Processes = append(a.Processes, process)
    a.exits = append(a.exits, exit)
    process.Start(a.c, exit)

    a.counter++
}

func (a *Admin) KillProcess(id uint64) {
    if id < 0 || int(id) > len(a.Processes) {
        fmt.Print("\nERROR. No existe proceso con ID ", id,
"\n\n")
    } else {
        a.exits[id] <- true
        fmt.Println("\nProceso con ID", id, "eliminado")
        fmt.Println()
    }
}
```

```
}
```

```
func (p *Process) Start(c chan bool, exit chan bool) {  
    fmt.Println("\nProceso con ID", p.id, "agregado!")  
    fmt.Println()  
    go p.Routine(c, exit)  
}
```

```
func (p *Process) Routine(c chan bool, exit chan bool) {  
    i := uint64(0)  
    for {  
        select {  
        case <- exit :  
            return  
        default:  
            shouldBeDisplayed := <- c  
            if shouldBeDisplayed {  
                fmt.Printf("id %d: %d\n", p.id, i)  
            }  
            i = i + 1  
            time.Sleep(time.Millisecond * 500)  
        }  
    }  
}
```

```
func (a *Admin) ControlShow(show chan bool) {  
    for {  
        if a.isShowingAll {  
            a.c <- true  
        }else {  
            a.c <- false  
        }  
    }  
}
```

```
func main() {  
    admin := Admin{c: make(chan bool)}  
    controlShow := make(chan bool)  
    shouldExit := false  
  
    go admin.ControlShow(controlShow)
```

```

for !shouldExit {
    showMenu()
    i := getIntFromUser()
    if i == 1 {
        admin.AddProcess()
    } else if i == 2 {
        if admin.isShowingAll {
            admin.isShowingAll = false
        } else {
            admin.isShowingAll = true
        }
    } else if i == 3 {
        fmt.Print("ID: ")
        admin.KillProcess(getIntFromUser())
    } else if i == 4 {
        shouldExit = true
    }
}

}

func showMenu() {
    fmt.Println("1. Agregar proceso")
    fmt.Println("2. Mostrar/Ocultar procesos")
    fmt.Println("3. Terminar proceso")
    fmt.Println("4. Salir")
    fmt.Print("Opcion: ")
}

func getIntFromUser() uint64 {
    var op uint64
    fmt.Scan(&op)

    return op
}

```