

Programación Lógica y funcional.

Luis Alberto Vargas González.

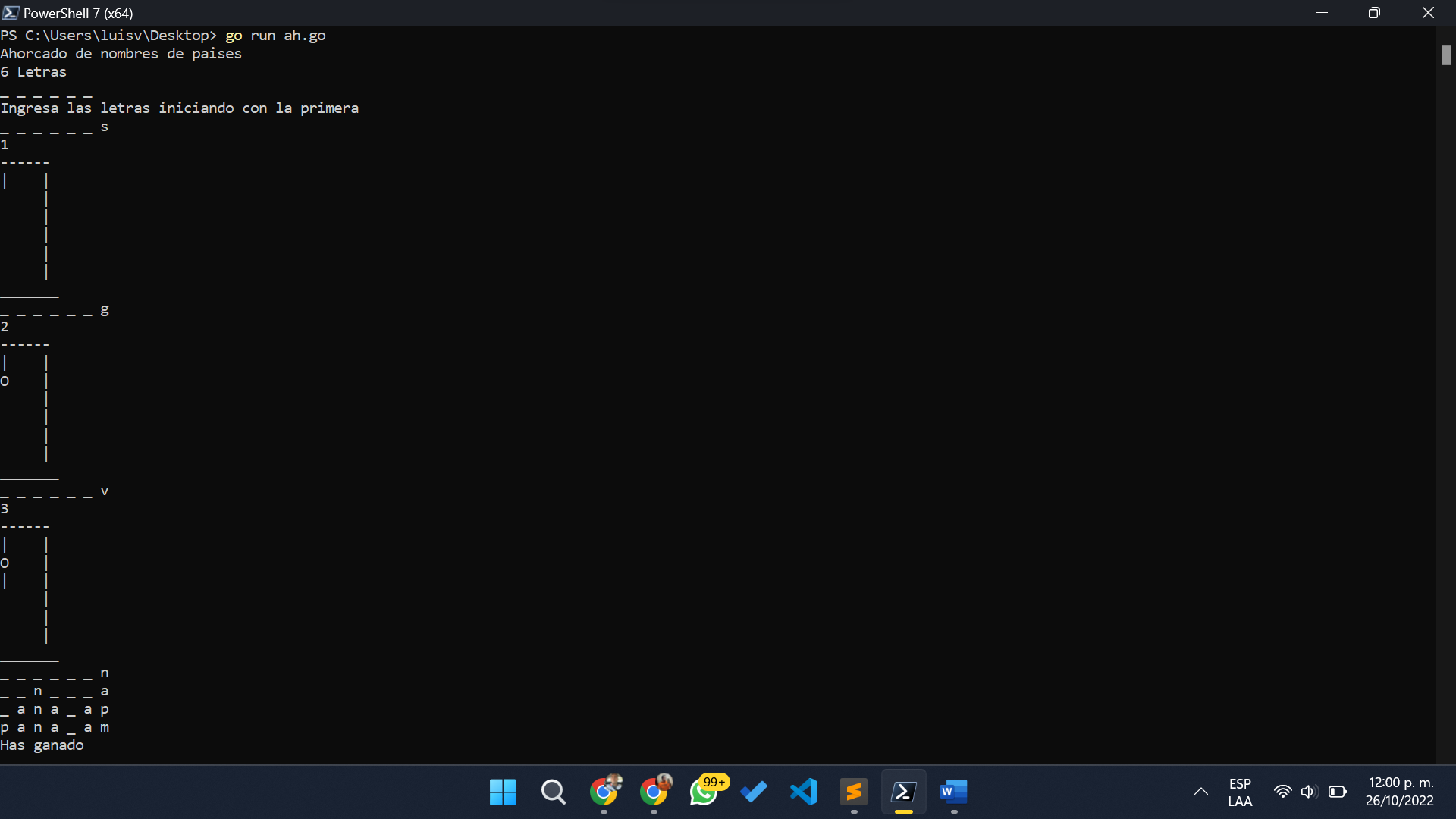
26/10/2022.

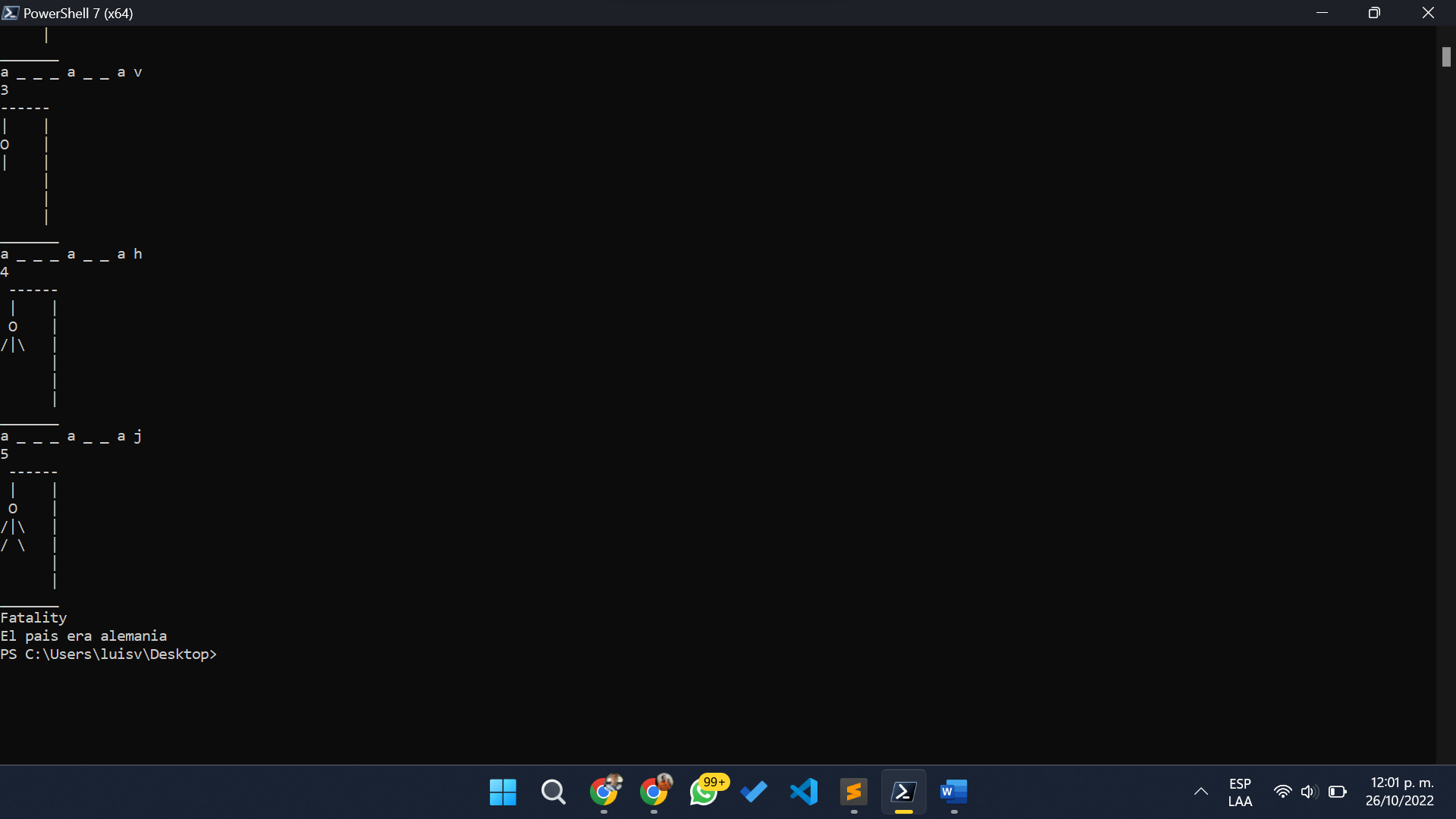
Practica 2.5 Ahorcado en Go.

**Programa en ejecución.**

Con un intento exitoso.

Y un intento fallido del juego.





**Código fuente:**

package main

import "fmt"

import "math/rand"

import "time"

var lis1 [30]string

var lis2 [30]string

var contador int =0

var errores int =0

var crar int =0

func main () {

arreglo :=[33]string {"argentina","belice","rusia", "brasil","canada","colombia","cuba","chile","ecuador",

"francia","espania", "eua", "guatemala", "haiti","honduras","mexico","panama","paraguay",

"peru","italia","surinam","uruguay","venezuela","alemania","portugal","egipto","marruecos",

"grecia","turquia","inglaterra","suecia","china","japon"}

s := rand.New(rand.NewSource(time.Now().UnixNano()))

var dato string

ale:=s.Intn(33)

st:= arreglo[ale]

fmt.Print("Ahorcado de nombres de paises\n")

fmt.Print(len(st)," ", "Letras","\n")

for i:=1; i<=len(st); i++{

fmt.Print("\_"," ")

}

carac := []rune(st)

for i:=0; i <len(carac); i++{

ter:= string(carac[i])

lis2[i]=ter

lis1[i]="\_"

}

fmt.Print("\nIngresa las letras iniciando con la primera \n")

for i:=0; i<=100; i++{

for i:=0; i<len(carac);i++{

fmt.Print(lis1[i]," ")

}

fmt.Scan(&dato)

for i:=0; i<=len(carac);i++{

if dato==lis2[i]{

lis1[i]=dato

contador++

}else{

errores++

}

}

if contador==len(carac) {

fmt.Print("Has ganado","\n")

fmt.Print("palabara generada:",st)

break;

}

if errores==len(carac)+1{

errores=0

crar++

fmt.Println(crar)

//fmt.Println(errores)

dibujo(crar)

} else {errores=0}

if crar==5{

fmt.Println("Fatality")

fmt.Println("El pais era",st)

break;

}

if contador==len(carac){

fmt.Println("felicidades ganaste")

break

}

}

}

func dibujo(num int){

if num==1{

fmt.Println("------")

fmt.Println("| |")

fmt.Println(" |")

fmt.Println(" |")

fmt.Println(" |")

fmt.Println(" |")

fmt.Println(" |")

fmt.Println("\_\_\_\_\_\_\_")

}

if num==2{

fmt.Println("------")

fmt.Println("| |")

fmt.Println("O |")

fmt.Println(" |")

fmt.Println(" |")

fmt.Println(" |")

fmt.Println(" |")

fmt.Println("\_\_\_\_\_\_\_")

}

if num==3{

fmt.Println("------")

fmt.Println("| |")

fmt.Println("O |")

fmt.Println("| |")

fmt.Println(" |")

fmt.Println(" |")

fmt.Println(" |")

fmt.Println("\_\_\_\_\_\_\_")

}

if num==4{

fmt.Println(" ------")

fmt.Println(" | |")

fmt.Println(" O |")

fmt.Println("/|\\ |")

fmt.Println(" |")

fmt.Println(" |")

fmt.Println(" |")

fmt.Println("\_\_\_\_\_\_\_")

}

if num==5{

fmt.Println(" ------")

fmt.Println(" | |")

fmt.Println(" O |")

fmt.Println("/|\\ |")

fmt.Println("/ \\ |")

fmt.Println(" |")

fmt.Println(" |")

fmt.Println("\_\_\_\_\_\_\_")

}

}

**Conclusiones:**

Como se pudo constatar, la creación y desarrollo del algoritmo del ahorcado es un proceso más complejo que complicado en sí mismo, por lo que requiere de especial atención al momento de crear la lógica de programación para poder tanto generar las palabras aleatorias, evaluar las letras y palabras , asi como para reemplazar los guiones por las letras correctas ingresada por el usuario, asi como el evaluar si dicha palabra ingresada por el usuario corresponde con la palabra generada , e indicar según sea el caso, si el usuario perdió o ganó la partida.