



Universidad Nacional Autónoma de México

Facultad de Ingeniería

División de Ingenierías Eléctrica

Asignatura:
Cómputo Móvil

Actividad #2

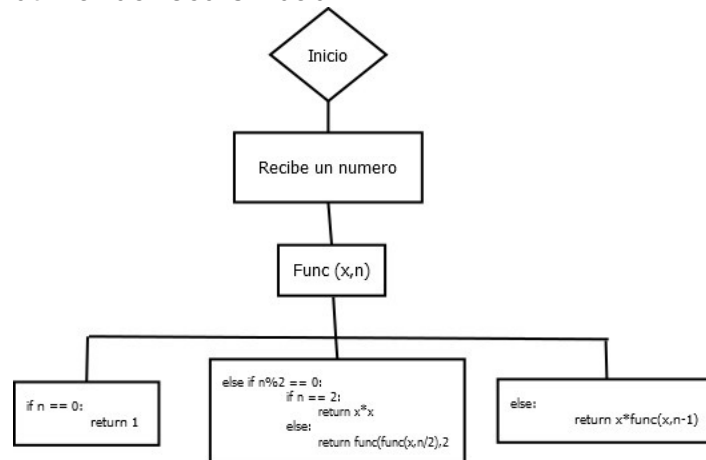
Alumno:
Ortega Moreno, Omar Orlando

Semestre 2021-1
Fecha de entrega: 06 de enero del 2021



1. Ejercicio 1:

- a. Desarrolle el algoritmo que calcule 2 a la N donde N es un número entero mayor o igual a cero, el algoritmo debe ser desarrollado exclusivamente utilizando recursividad.



b.

```

import Foundation

func readData() {
    let x: Int = 2
    print("Ingresa el numero entero para calcular la potencia: ")
    let input = readLine()!
    let n = Int(input)
    print("El resultado es: \n")
    print(power(x: x, n: n!))
}

func power(x: Int, n: Int) -> Int {
    if n == 0 {
        print("El resultado es: ")
        return 1
    }
    if n % 2 == 0 {
        if n == 2 {
            return x*x
        } else {
            return power(x: power(x: x, n: n / 2), n: 2)
        }
    } else {
        return x * power(x: x, n: n - 1)
    }
}
  
```

c.

```

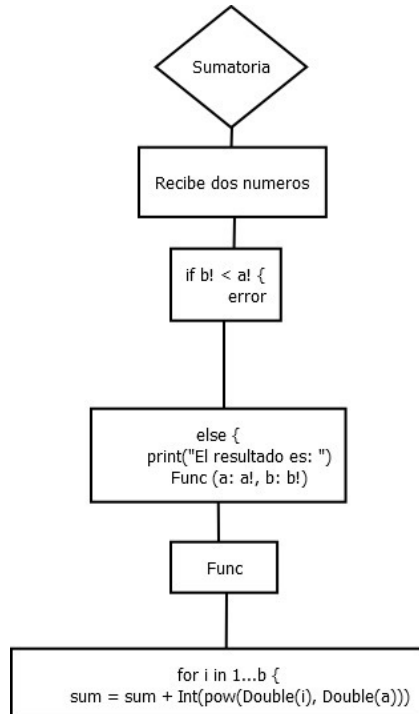
Ingresa el numero entero para calcular la potencia:
8
El resultado es:

256
Program ended with exit code: 0
  
```

d.

2. Ejercicio 2:

- a. Dado a y b como números enteros, realizar la sumatoria de: $1^a + 2^a + 3^a \dots b^a$



b.

```

// Sumatoria.swift
// Program2
//
// Created by Orlando Ortega on 30/12/20.
//

import Foundation

func readData() {
    print("Ingresa el primer número para calcular la sumatoria: ")
    let input = readLine()!
    let a = Int(input)
    print("Ingresa el segundo número: ")
    let input2 = readLine()!
    let b = Int(input2)
    if b! < a! {
        print("El segundo número no puede ser menor a el primero.")
        readData()
    } else {
        print("El resultado es: ")
        sumatory(a: a!, b: b!)
    }
}

func sumatory(a: Int, b: Int) {
    var sum: Int = 0
    for i in 1...b {
        sum = sum + Int(pow(Double(i), Double(a)))
    }
    print(sum)
}
  
```

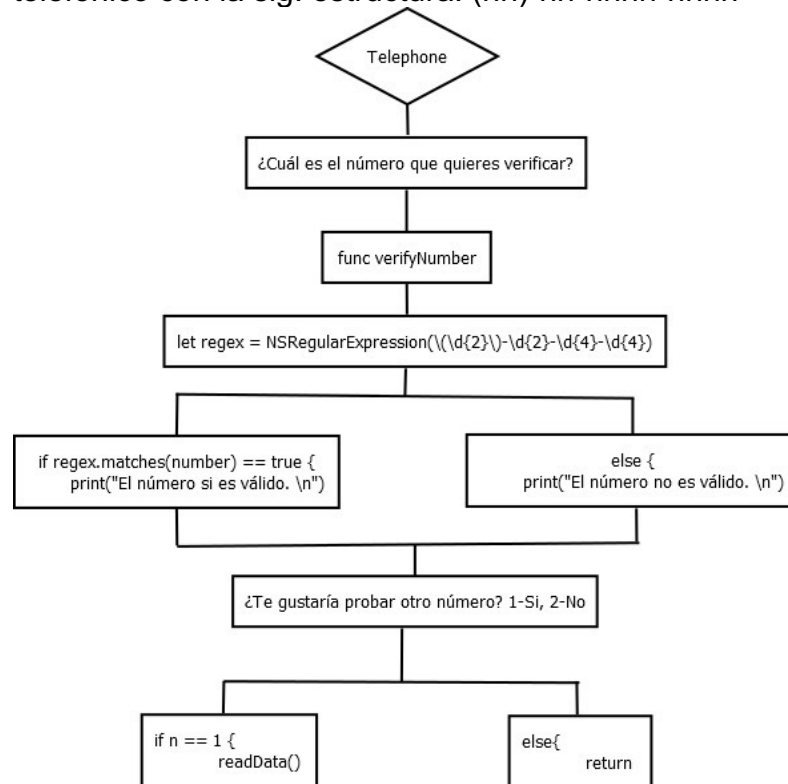
c.

```

Ingresa el primer número para calcular la sumatoria:
3
Ingresa el segundo número:
4
El resultado es:
100
Program ended with exit code: 0

```

- d.
3. Ejercicio 3:
- a. Dada una cadena introducida por el usuario validar si es un número telefónico con la sig. estructura: (nn)-nn-nnnn-nnnn



b.

```

import Foundation

func readData() {
    print("¿Cuál es el número que quieres verificar?")
    guard let input = readLine() else { return }
    verifyNumber(number: input)
}

func verifyNumber(number: String) {
    // En swift 5 ya no es necesario escapar con doble backslash.
    // Hay un problema en Xcode donde no toma de forma correcta los parentesis escapados.
    let regex = NSRegularExpression("#\\(\\d{2}\\)-\\d{2}-\\d{4}-\\d{4}"#)
    if regex.matches(number) == true {
        print("El número si es válido. \n")
        print("¿Te gustaría probar otro número? 1-Si, 2-No")
        let input = readLine()!
        let n = Int(input)!
        if n == 1 {
            readData()
        } else {
            return
        }
    } else {
        print("El número no es válido. \n")
        print("¿Te gustaría probar otro número? 1-Si, 2-No")
        let input = readLine()!
        let n = Int(input)!
        if n == 1 {
            readData()
        } else {
            return
        }
    }
}

```

- c.
- d. En Xcode no reconoce los paréntesis escapados, la corrida del código es desde repl.it:

```

➤ ./main
¿Cuál es el número que quieres verificar?
(55)-55-5555-5555
El número si es válido.

¿Te gustaría probar otro número? 1-Si, 2-No
1
¿Cuál es el número que quieres verificar?
55-55-5555-5555
El número no es válido.

```

4. Ejercicio 4:

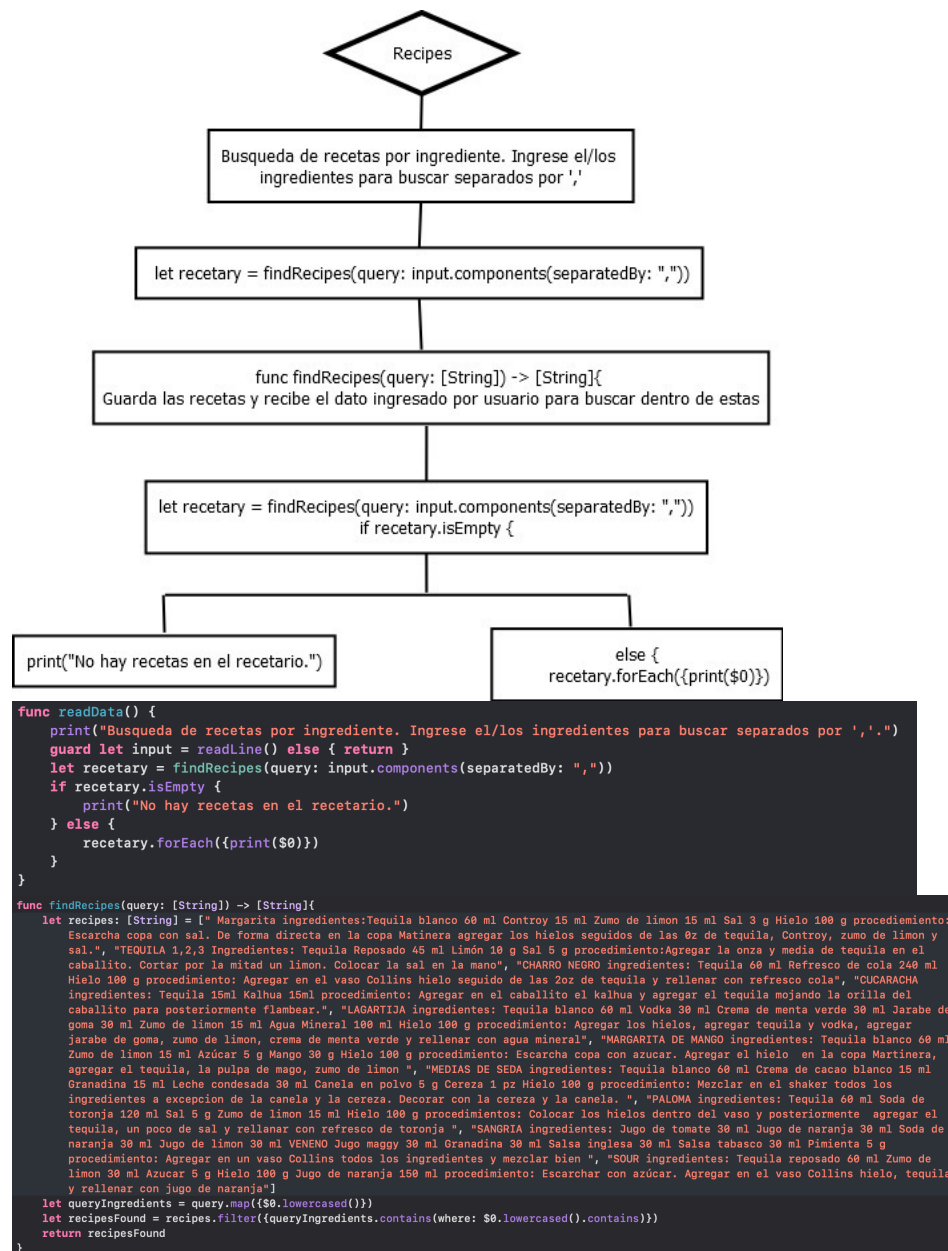
- Dada una lista de recetas (mínimo 10 recetas – texto) encontrar las recetas que contengan alguno o algunos ingredientes. El usuario solo verá las recetas que contengan los ingredientes que el captura desde la línea de comandos. Es importante que las recetas no solo indiquen instrucciones, sino también ingredientes. El usuario puede hacer la siguiente captura:

Introduce ingredientes: huevos

Introduce ingredientes: res, huevos, sal

Introduce ingredientes: harina, sal, leche

Y solo se deberá mostrar las recetas que contengan estos ingredientes.



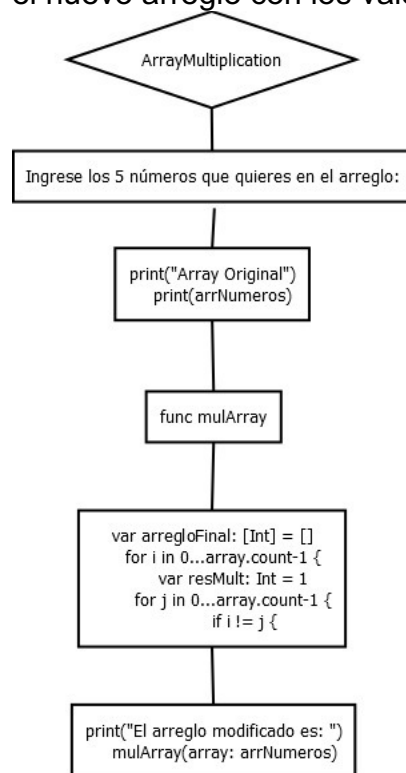
```

Busqueda de recetas por ingrediente. Ingrese el/los ingredientes para buscar separados por ','.
limon
Margarita ingredientes: Tequila blanco 60 ml Controy 15 ml Zumo de limon 15 ml Sal 3 g Hielo 100 g
procedimiento: Escarcha copa con sal. De forma directa en la copa Matinera agregar los
hielos seguidos de las 0z de tequila, Controy, zumo de limon y sal.
TEQUILA 1,2,3 Ingredientes: Tequila Reposado 45 ml Limón 10 g Sal 5 g procedimiento: Agregar la
onza y media de tequila en el caballito. Cortar por la mitad un limon. Colocar la sal en la
mano
LAGARTIJA ingredientes: Tequila blanco 60 ml Vodka 30 ml Crema de menta verde 30 ml Jarabe de goma
30 ml Zumo de limon 15 ml Agua Mineral 100 ml Hielo 100 g procedimiento: Agregar los hielos,
agregar tequila y vodka, agregar jarabe de goma, zumo de limon, crema de menta verde y
rellenar con agua mineral
MARGARITA DE MANGO ingredientes: Tequila blanco 60 ml Zumo de limon 15 ml Azúcar 5 g Mango 30 g
Busqueda de recetas por ingrediente. Ingrese el/los ingredientes para buscar separados por ','.
canela, cola
CHARRO NEGRO ingredientes: Tequila 60 ml Refresco de cola 240 ml Hielo 100 g procedimiento:
Agregar en el vaso Collins hielo seguido de las 2oz de tequila y rellenar con refresco cola
MEDIAS DE SEDA ingredientes: Tequila blanco 60 ml Crema de cacao blanco 15 ml Granadina 15 ml
Leche condensada 30 ml Canela en polvo 5 g Cereza 1 pz Hielo 100 g procedimiento: Mezclar en el
shaker todos los ingredientes a excepcion de la canela y la cereza. Decorar con la cereza y la
canela.
Program ended with exit code: 0

```

5. Ejercicio 5:

- Dado un arreglo de números enteros, regresar un nuevo arreglo tal que en la posición "i" del nuevo arreglo es el producto de todos los números del arreglo original menos el de la posición i, por ejemplo: [1,2,3,4,5] nos dara el nuevo arreglo con los valores [120,60,40,30,24]



b.

```

import Foundation

func readData() {
    var arrNumeros: [Int] = []
    print("Ingrese los 5 números que quieres en el arreglo: ")
    for i in 1...5 {
        print("Número \(i): ")
        let input = readLine()!
        arrNumeros.append(Int(input)!)
    }
    print("Array Original")
    print(arrNumeros)
    print("El arreglo modificado es: ")
    mulArray(array: arrNumeros)
}

func mulArray(array: [Int]) {
    var arregloFinal: [Int] = []
    for i in 0...array.count-1 {
        var resMult: Int = 1
        for j in 0...array.count-1 {
            if i != j {
                resMult *= array[j]
            }
        }
        arregloFinal.append(resMult)
    }
    print(arregloFinal)
}

```

c.

```

Ingrese los 5 números que quieres en el arreglo:
Número 1:
1
Número 2:
2
Número 3:
3
Número 4:
4
Número 5:
5
Array Original
[1, 2, 3, 4, 5]
El arreglo modificado es:
[120, 60, 40, 30, 24]
Program ended with exit code: 0

```

d.