

CURSO : DESARROLLO DE APLICACIONES MÓVILES II
PROFESOR : Jorge Luis Mayta Guillermo
SEMESTRE : 2024 – I
CICLO : SEXTO
SECCIÓN : T6AN
GRUPO : 01
FECHA : 15/10/2024

NOTA

ALUMNO (A) : BRENDA LUZ ALEJANDRO BECERRA

EVALUACION DE LABORATORIO 1

Logro

Tenga presente que el logro de esta evaluación consiste en que usted los conceptos fundamentales de programación orientada a objetos en Swift, tales como la creación y manipulación de estructuras o clases, así como la implementación de funciones para realizar operaciones básicas sobre los datos.

Consideraciones generales

- Considerar el orden, la limpieza y la claridad de las respuestas.
- La evaluación es personal, queda totalmente prohibido realizar consultas a compañeros.
- Cualquier intento de plagio o incumplimiento de las normas anulara completamente el examen.

Consolidado

Pregunta	Puntaje		Llenar solo en caso de recalificación justificada	
	Máximo	Obtenido	Sustento	Puntaje
1	10			
2	10			
Nota recalificada				

Pregunta 01

Escriba una función en Swift que tome un arreglo de enteros y devuelva un arreglo que contenga solo los números pares del arreglo original.

CODIGO:

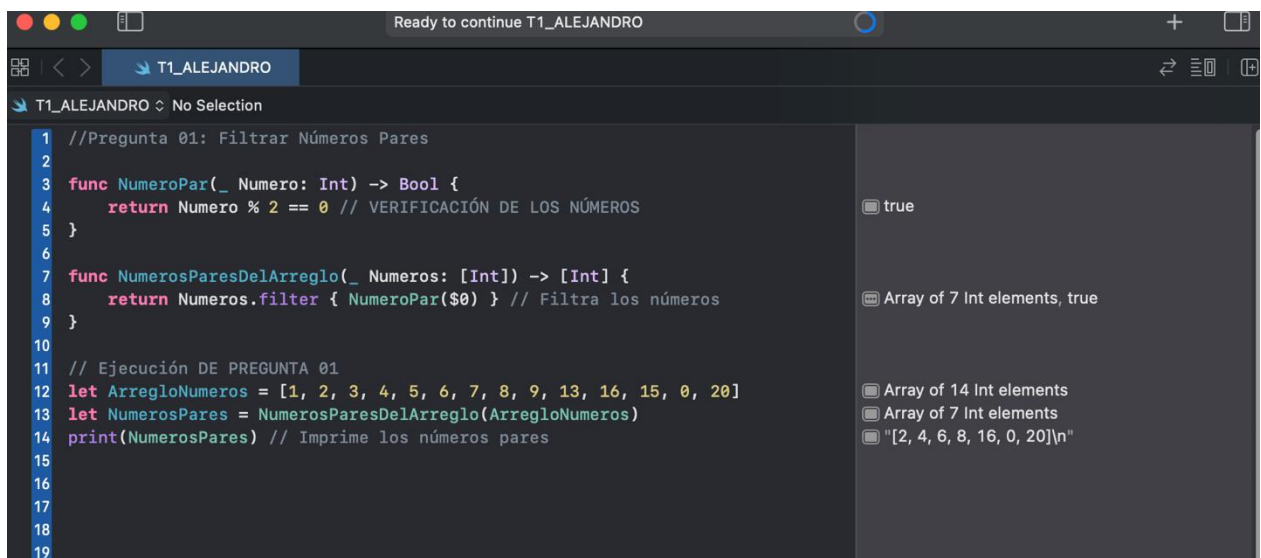
//Pregunta 01: Filtrar Números Pares

```
func NumeroPar(_ numero: Int) -> Bool {
    return numero % 2 == 0 // VERIFICACIÓN DE LOS NÚMEROS
```

```
func NumerosParesDelArreglo(_ numeros: [Int]) -> [Int] {
    return numeros.filter { NumeroPar($0) } // Filtra los números
```

// Ejecución DE PREGUNTA 01

```
let ArregloNumeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 16, 15, 0, 20]
let NumerosPares = NumerosParesDelArreglo(ArregloNumeros)
print(NumerosPares) // Imprime los números pares
```



Pregunta 02

Escriba una función en Swift que tome un arreglo de enteros y compruebe si el primer elemento y el último elemento son iguales. La longitud del arreglo debe ser mayor o igual a 1.

CODIGO:

//Pregunta 02: Comprobar Primer y Último Elemento

```
func VerificarElementos (_ array: [Int]) -> Bool {
    guard array.count > 1 else {
        print "La lista está vacía." // LISTADO
        return false
    }

    let first = array.first // PRIMER NÚMERO
    let last = array.last // ÚLTIMO NÚMERO

    if first == last {
        print "El primer elemento ( \(first) ) es igual al último elemento ( \(last) )."
    } else {
        print "El primer elemento ( \(first) ) no es igual al último elemento ( \(last) )."
    }
}
```

// Ejecución: COMPRUEBA SI EL PRIMER Y ÚLTIMO NÚMERO SON IGUALES

```
let EjecucionDeAreglo = [1, 2, 3, 1, 2, 3, 4, 6, 4]
VerificarElementos(EjecucionDeAreglo)
```

```

12 let ArregloNumeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 16, 19, 0, 20]
13 let NumerosPares = NumerosParesDelArreglo(ArregloNumeros)
14 print(NumerosPares) // Imprime los números pares
15
16
17 //Pregunta 02: Comprobar Primer y Último Elemento
18
19 func VerificarElementos(_ Numeros: [Int]) {
20     guard Numeros.count >= 1 else {
21         print("La lista está vacía.") // LISTADO
22         return
23     }
24
25     let NumeroUno = Numeros.first! // PRIMER NÚMERO
26     let NumeroUltimo = Numeros.last! // ÚLTIMO NÚMERO
27
28     if NumeroUno == NumeroUltimo {
29         print("El primer elemento \(NumeroUno) es igual al último elemento
30             \(NumeroUltimo).")
31     } else {
32         print("El primer elemento \(NumeroUno) no es igual al último
33             elemento \(NumeroUltimo).")
34     }
35 }
36
37 // Ejecución: COMPRUEBA SI EL PRIMER Y ÚLTIMO NÚMERO SON IGUALES
38 let EjecucionDeAreglo = [1, 2, 3, 1, 2, 3, 4, 6, 4]
39 VerificarElementos(EjecucionDeAreglo)

```

Execution Output:

- Array of 14 int elements
- Array of 7 Int elements
- "[2, 4, 6, 8, 16, 0, 20]\n"
- 1
- 4
- "El primer elemento (1) no es igual al último elemento (4).\n"
- Array of 9 Int elements

Line: 16 Col: 1

ARCHIVO DE TRABAJO:

[T1_BRENDA LUZ ALEJANDRO BECERRA.zip](#)

Rúbrica				
Pregunta	Puntaje	10 puntos	6 puntos	0 puntos
1	10 puntos	La función está implementada correctamente y devuelve los números pares esperados.	La función está implementada parcialmente.	La función no está implementada correctamente.
2	10 puntos	La función está implementada correctamente y devuelve el resultado esperado, manejando adecuadamente los casos de longitud del arreglo.	La función está implementada correctamente pero no maneja adecuadamente los casos de longitud del arreglo.	La función no está implementada correctamente.