

## Práctica 05

DOCENTE	CARRERA	CURSO
MSc. Vicente Enrique Machaca Arceda	Escuela Profesional de Ingeniería de Software	Fundamentos de Lenguajes de Programación

PRÁCTICA	TEMA	DURACIÓN
05	Ensamblador	3 horas

### 1. Datos de los estudiantes

- GIT: GIT-Repo
- Integrantes:
  - Roberto Heredia Garland

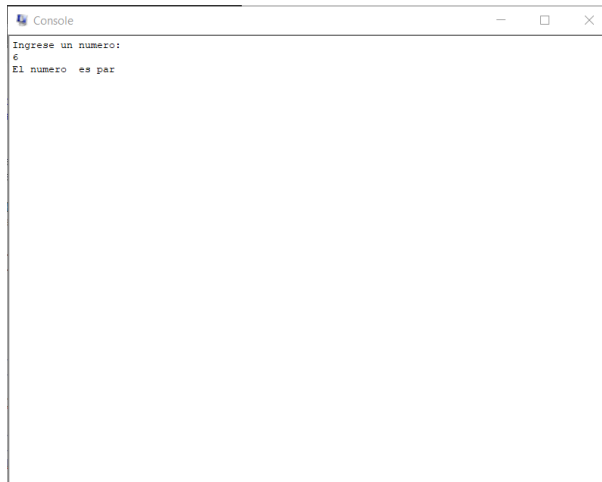
### 2. Ejercicios

1. Escribir un programa que pida al usuario un número entero y muestre por pantalla si es par o impar.

```
1 .data
2 num:.asciiz  "Ingrese un numero:\n"
3 even:.asciiz "El numero es par "
4 oddr:.asciiz "El numero es impar "
5
6 .text
7
8 main:
9 addi $t0, $0, 2
10 la $a0, num
11 li $2, 4
12 syscall
13
14 li $2, 5
15 syscall
16
17 div $2, $t0
18 mfhi $t1
19 beq $t1, $0, print_1
20 la $a0, odd
21 li $2, 4
22 syscall
23
24 j fin
25 print_1: la $a0, even
26 li $2, 4
```

```
27 syscall
28
29 j fin
30 fin: li $2, 10
31 syscall
32
33 jr $ra
```

---

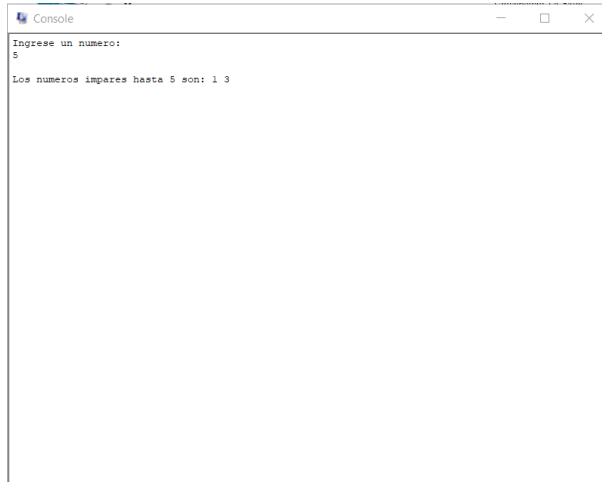


2. Escribir un programa que pida al usuario un número entero positivo y muestre por pantalla todos los números impares desde 1 hasta ese número.

---

```
1 .data
2 num: .asciiz "Ingrese un numero: \n"
3 odds: .asciiz "\nLos numeros impares hasta "
4 num1: .asciiz " son: "
5 .text
6 main:
7     li $v0, 4
8     la $a0, num
9     syscall
10
11     li $v0, 5
12     syscall
13     move $t0, $v0
14
15     li $v0, 4
16     la $a0, odds
17     syscall
18
19     li $v0, 1
20     move $a0, $t0
21     syscall
22
23     li $v0, 4
24     la $a0, num1
25     syscall
26
27     li $t1, 1
28     loop1:
29         bge $t1, $t0 end_loop1
30
31         li $v0, 1
32         move $a0, $t1
33         syscall
34
35         li $a0, 32
36         li $v0, 11
37         syscall
38
39         add $t1, $t1, 2
40         j loop1
41
42     end_loop1:
43         li $v0, 10
44         syscall
```

---

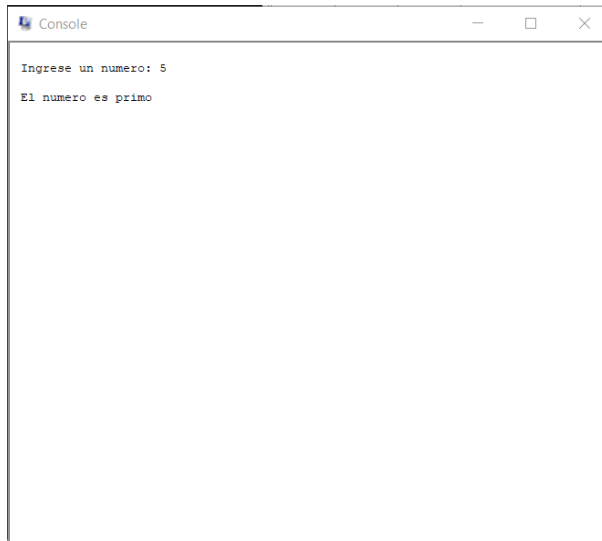


```
Console
Ingrese un numero:
5
Los numeros impares hasta 5 son: 1 3
```

3. Escribir un programa que pida al usuario un número entero y muestre por pantalla si es un número primo o no.
- 

```
1
2 .data
3 num: .asciiz "\n Ingrese un numero: "
4 noprimo: .asciiz "\n El numero no es primo"
5 primo: .asciiz "\n El numero es primo"
6
7 .text
8
9 main:
10     li $v0, 4
11     la $a0, num
12     syscall
13     li $v0, 5
14     syscall
15     move $t0, $v0
16     li $t1, 2
17
18 loop1:
19     beq $t0, $t1 si_primo
20     div $t0, $t1
21     mfhi $t2
22     beqz $t2, no_primo
23     addi $t1, $t1 1
24     j loop1
25
26 no_primo:
27     li $v0, 4
28     la $a0, noprimo
29     syscall
30     j exit
31
32 si_primo:
33     li $v0, 4
34     la $a0, primo
35     syscall
36     j exit
37
38 exit: li $v0, 10
39     syscall
```

---



```
Console
Ingrese un numero: 5
El numero es primo
```