

**Universidad de Costa Rica**

**Escuela de Ciencias de la Computación e Informática**

**CI-0114 Fundamentos de Arquitectura de Computadoras**

**Profesor:** Luis Araya Carballo

**Estudiantes:**

- Amber Villarreal Campos - C28481
- Liqing Yosery Zheng Lu - C38680

## **Propuesta de Proyecto Final: Programación en ensamblador MIPS**

### **Descripción**

“Llegar a 100” es un juego de dos jugadores, en donde en cada turno, se le puede sumar al contador un número del 1 al 10. El ganador es el jugador que, en su turno, logra llegar a 100.

En el caso de este programa, el jugador competirá contra una máquina, la cual estará programada con la intención de ganar la mayoría de partidas posibles.

### **Consideraciones técnicas:**

- El programa será desarrollado para que el usuario juegue contra una máquina.
- El usuario tomará el primer turno.
- Solo hay una partida conocida en la que la máquina no gana: entradas del usuario en orden = (1, 12, 23, 34, 45, 56, 67, 78, 89, 100).
- Se desarrollará una interfaz gráfica, en la cual se mostrará en una ventana el estado actual del contador y el número de turnos que ha durado la partida.
- El programa se terminará de ejecutar una vez el contador llegue a 100.
- Si se le ingresa un caracter diferente a entero, ocurrirá un error de syscall y el programa terminará de ejecutarse.

### **Restricciones del programa**

- Para la interfaz gráfica se debe tomar en cuenta las restricciones que se pueden enfrentar al programar en ensamblador MIPS.

- La máquina estará equipada con una lógica implementada para calcular el movimiento ganador de manera que siempre gane.

## **Resultados**

Imprime de quién es el turno al principio de cada uno. Si es el usuario, le pide que ingrese un número del 1 al 10. En caso de que se le ingrese un número fuera del rango pedido, se imprime un mensaje pidiendo otro número para continuar con la jugada. Una vez ingresado un número válido, se imprime un mensaje de confirmación sobre el número que ya se logró sumar al contador, además del estado del contador actual. En el turno de la máquina se imprime el número que se jugó y el estado del contador actual. Cuando el contador llegue a 100, el programa imprimirá una felicitación al usuario si este gana, de lo contrario, imprime un mensaje indicando que ganó la máquina.

## **Funciones Desarrolladas**

### **1. MANEJO\_TURNOS:**

- a. Diagrama: Incluido en el diagrama del proyecto.
- b. Parámetros: Recibe la cantidad de turnos, guardado en el registro \$t1.
- c. Retornos: No retorna nada.
- d. Funcionalidad: Si el turno es par, salta a TURNO\_HUMANO, de lo contrario, salta a TURNO\_MAQUINA.

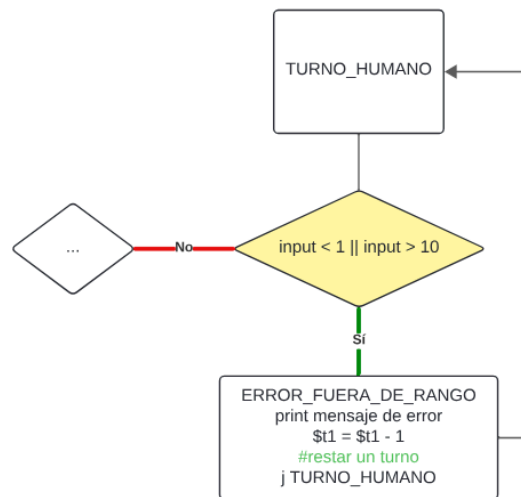
### **2. TURNO\_HUMANO:**

- a. Diagrama: Incluido en el diagrama del proyecto.
- b. Parámetros: Recibe la cantidad de turnos, guardado en el registro \$t1, y el contador guardado en \$a1.

- c. Retornos: Devuelve la cantidad de turnos sumada en 1 y el contador sumado en lo que el usuario haya ingresado.
- d. Funcionalidad: Pide que se ingrese un número del 1 al 10. Si el número ingresado no cumple con el rango, salta a la función ERROR\_FUERA\_DE\_RANGO. Le suma al contador el número ingresado y a la cantidad de turnos le suma 1.

### 3. ERROR\_FUERA\_DE\_RANGO:

- a. Diagrama:



- b. Parámetros: Recibe la cantidad de turnos guardado en le registro \$t1.
- c. Retornos: Devuelve la cantidad de turnos restada en 1.
- d. Funcionalidad: Imprime el mensaje de error, le resta 1 a la cantidad de turnos y salta a TURNO\_HUMANO.

### 4. TURNO\_MAQUINA:

- a. Diagrama: Incluido en el diagrama del proyecto.
- b. Parámetros: Recibe la cantidad de turnos, guardado en el registro \$t1, y el contador guardado en \$a1.
- c. Retornos: Devuelve la cantidad de turnos sumada en 1.

- d. Funcionalidad: Le suma 1 a la cantidad de turnos. Si el contador actual es igual a algún número crítico (1, 12, 23, 34, 45, 56, 67, 78, 89), salta a TIRADA\_RANDOM. Si el contador actual es menor o igual a 90, salta a MOVIMIENTO\_GANADOR. De lo contrario, salta a TIRADA\_X, siendo X el número crítico más cercano.

5. TIRADA\_X:

- a. Diagrama: Incluido en el diagrama del proyecto.
- b. Parámetros: Recibe el contador guardado en \$a1.
- c. Retornos: Devuelve el contador igual a X.
- d. Funcionalidad: Encuentra el número que se le debe sumar al contador para que sea igual a X, restando X con el contador actual. Imprime en pantalla el número encontrado y se lo suma al contador.
- e. Observación: Hay 8 métodos de TIRADA\_X, siendo las X los números 12, 23, 34, 45, 56, 67, 78, y 89.

6. TIRADA\_RANDOM:

- a. Diagrama: Incluido en el diagrama del proyecto.
- b. Parámetros: Recibe el contador guardado en \$a1.
- c. Retornos: Devuelve el contador sumado a algún número random entre 1 al 10.
- d. Funcionalidad: Genera un número random del 0 al 9. Le suma 1 al número random generado para que esté dentro del rango deseado. Imprime en pantalla el número random y se lo suma al contador.

7. MOVIMIENTO\_GANADOR:

- a. Diagrama: Incluido en el diagrama del proyecto.
- b. Parámetros: Recibe el contador guardado en \$a1.
- c. Retornos: Devuelve el contador igual a 100.

- d. Funcionalidad: Encuentra el número que se le debe sumar al contador para que sea igual a 100, restando 100 con el contador actual. Imprime en pantalla el número encontrado y se lo suma al contador.

8. GANADOR\_HUMANO:

- a. Diagrama: Incluido en el diagrama del proyecto.
- b. Parámetros: No recibe nada.
- c. Retornos: No retorna nada.
- d. Funcionalidad: Imprime el mensaje de ganador humano y termina la ejecución del programa.

9. GANADOR\_MAQUINA:

- a. Diagrama: Incluido en el diagrama del proyecto.
- b. Parámetros: No recibe nada.
- c. Retornos: No retorna nada.
- d. Funcionalidad: Imprime el mensaje de ganador máquina y termina la ejecución del programa.

## Diagrama de Flujo:

