



Res. No. 16740, 2017-2021.



Vigilante MinEduación.

## TABLA DE CONTENIDO

<b>Objetivos de aprendizaje</b>	<b>1</b>
<b>Análisis</b>	<b>2</b>
<b>Diseño</b>	<b>2</b>
<b>Implementación</b>	<b>2</b>
<b>Verificación</b>	<b>2</b>
<b>Corte 2 - Caso 3 : Juego tragamonedas</b>	<b>2</b>
Fase 1 y 2: análisis y diseño	3
Pruebas de escritorio	5
Fase 3: implementación	5
Fase 4: verificación	6
Entregables	6

## OBJETIVOS DE APRENDIZAJE

1. Aplicar la metodología vista en clase para resolver problemas usando algoritmos.
2. Aplicar las estructuras de control de flujo condicional en la solución de problemas.
3. Aplicar las estructuras de control de flujo repetitivo en la solución de problemas.
4. Crear una propuesta de solución para un problema haciendo uso de los flujogramas.
5. Usar el lenguaje de programación Python para implementar una propuesta de solución que involucra estructuras de control de flujo condicional y repetitivo.



#### Análisis

Comprender la funcionalidad del programa, así como identificar las restricciones que se tengan sobre la operación del mismo.



#### Diseño

Crear una propuesta de solución mediante el uso de flujogramas. Transformar entradas en salidas esperadas.



#### Implementación

Transcribir la propuesta a un lenguaje de programación.



#### Verificación

Ejecución bajo condiciones controladas.

## CORTE 2 - CASO 3 : JUEGO TRAGAMONEDAS



Jugar a las maquinitas tragamonedas es una de las formas más populares de juegos de azar en el mundo y potencialmente, una de las más dañinas.

Las siguientes son las instrucciones que usted debe tener en cuenta para crear un programa que permita simular el juego de tragamonedas.

- Inicialmente, el jugador indica con cuántas monedas empieza a jugar, debe ser un valor positivo mayor a cero.
- En cada tirada, al jugador se le cobra(apuesta) una moneda.
- En cada tirada, el programa muestra tres números al azar del 1 al 6.
- Si los tres números son distintos, el jugador no gana nada (es decir, pierde la moneda que apostó).

- Si salen dos números iguales, el jugador gana dos monedas (es decir, gana en total una moneda).
- Si salen tres números iguales, el jugador gana cinco monedas (es decir, gana en total cuatro monedas).
- Si el jugador se queda sin monedas, la partida termina.
- Tras cada tirada, el programa retroalimenta al usuario indicando la cantidad de monedas que ha ganado en la partida y el acumulado de estas. el jugador indica si quiere seguir jugando o no.
- Al terminar la partida, el programa indica cuántas monedas se han ganado o perdido.

Recuerde que el proceso principal es el encargado de pedir los datos al usuario y mostrar las salidas.

A continuación, se le entrega el análisis modular y el diseño del **principal** (proceso integrador que pide los datos de entrada al usuario, llama a los subprocesos y finalmente muestra las salidas)



## FASE 1 Y 2: ANÁLISIS Y DISEÑO

Identifique y defina los datos de entrada, restricciones y salidas para el proceso principal.

Trabajo realizado por JUAN PABLO LENIS y NICOLAS ROMERO GUARNIZO

Entradas			Salidas		
Tipo	Descripción	Variable	Tipo	Descripción	variable
int	Numero de monedas	Moneda	Stri	Retorna mensaje de	mensajeEr

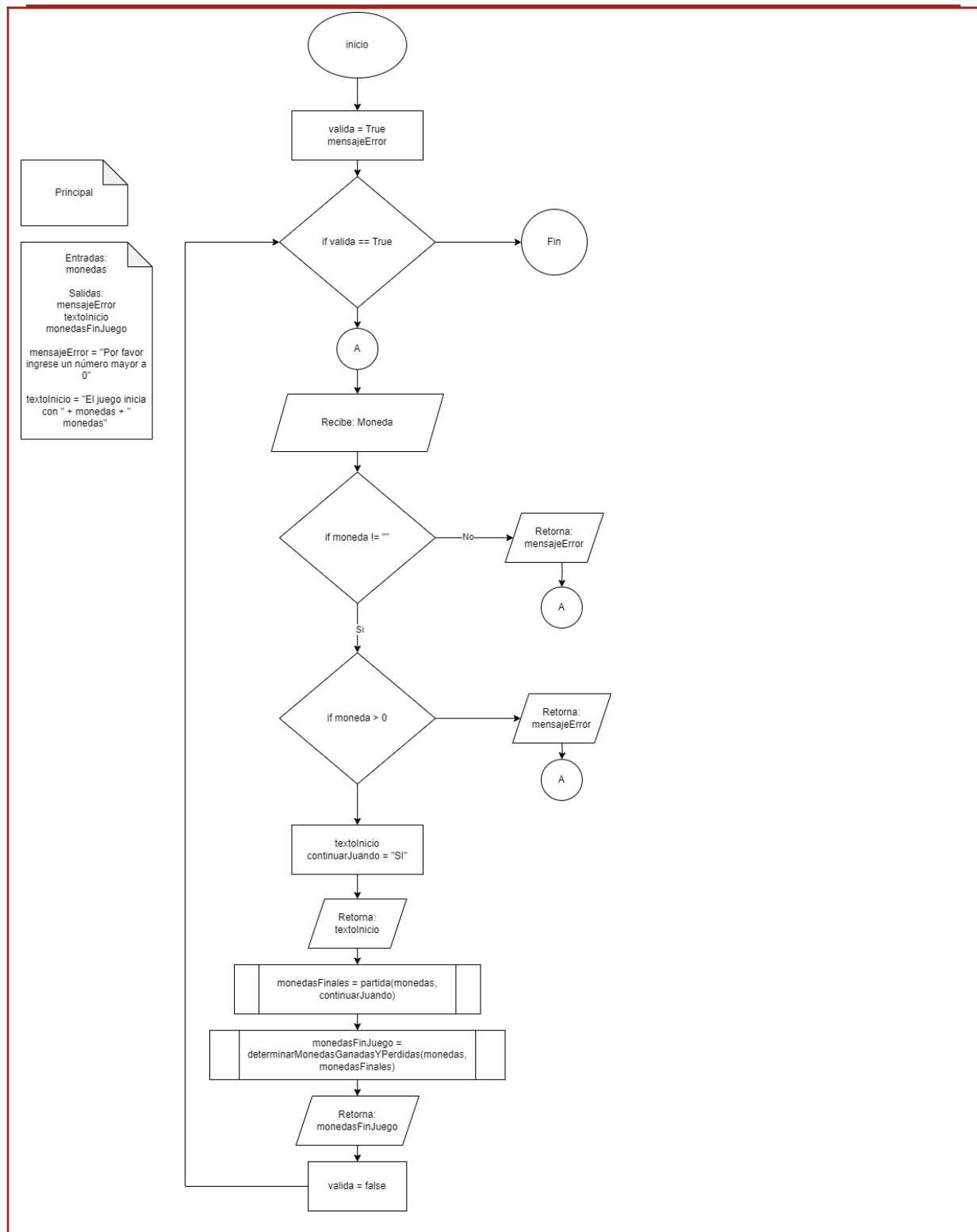
	del usuario		ng	error indicando su respuesta correcta	ror
			Stri ng	Indica con cuantas monedas inicia el usuario	textoInicio
			Stri ng	Retorna cuantas monedas gano o perdio cuando decide no jugar más	monedasF inJuego

**Restricciones:**

- En cada tirada, al jugador se le cobra una moneda
- Si los tres números son distintos, el jugador no gana nada y pierde una moneda
- Si salen dos números iguales, el jugador gana dos monedas
- Si salen tres números iguales, el jugador gana cinco monedas
- Si el jugador se queda sin monedas, el juego termina



**Flujograma**



Realice la propuesta de solución para el subproceso **determinarMonedasGanadas**

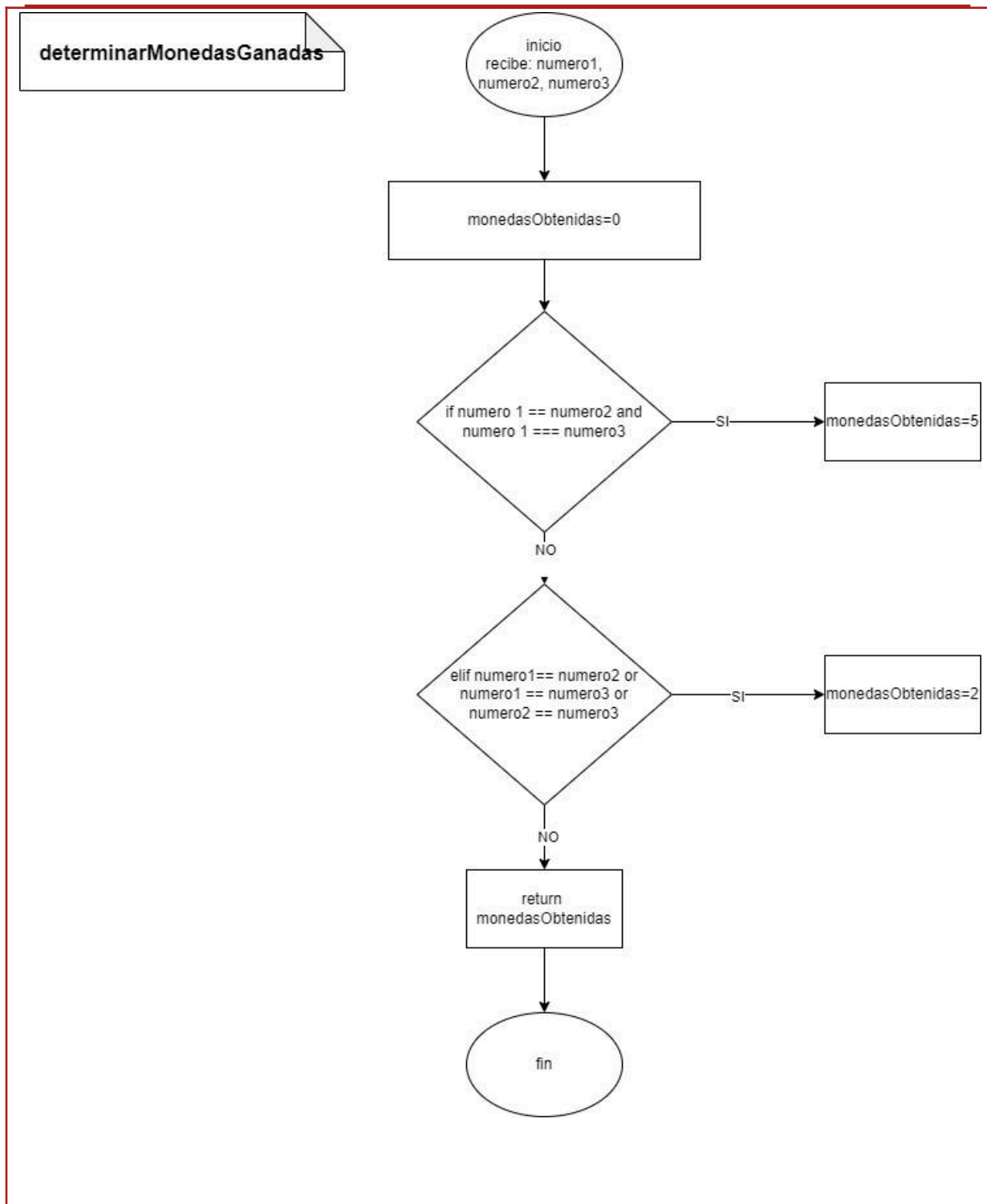


### Subproceso : determinarMonedasGanadas

Entradas			Salidas		
Tipo	Descripción	Variable	Tipo	Descripción	Variable
int	número al azar 1	numero1	int	número de monedas ganadas	monedasObtenidas
int	número al azar 2	numero2			
int	número al azar 3	numero3			



### Flujograma subproceso:





Realice la propuesta de solución para el subproceso **determinarResultado**.

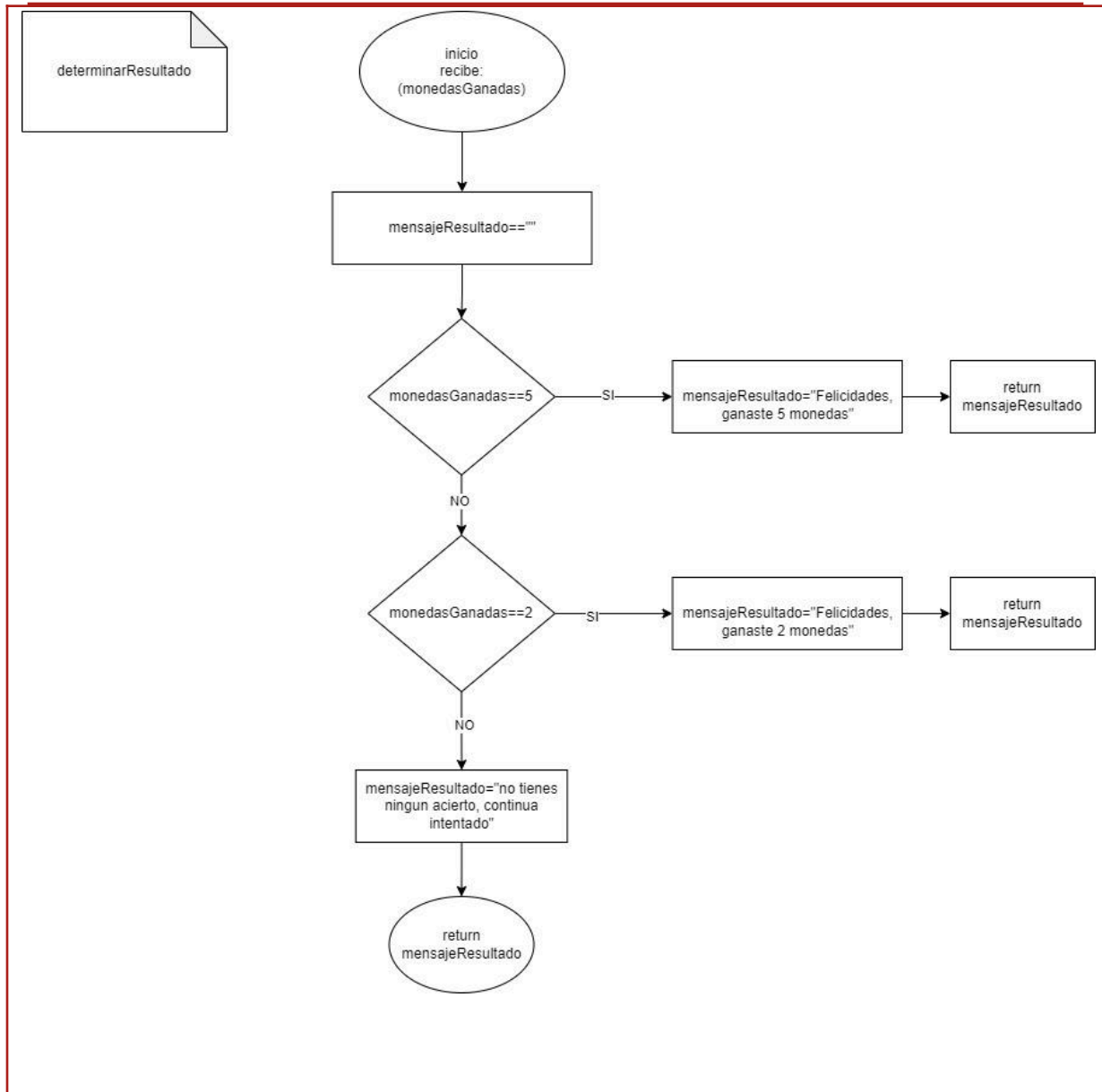


### Subproceso : determinarResultado

Entradas			Salidas		
Tipo	Descripción	Variable	Tipo	Descripción	Variable
int	monedas ganadas por el usuario	monedasGanadas	int	Mensaje del resultado en la partida	mensajeResultado



### Flujograma subproceso:




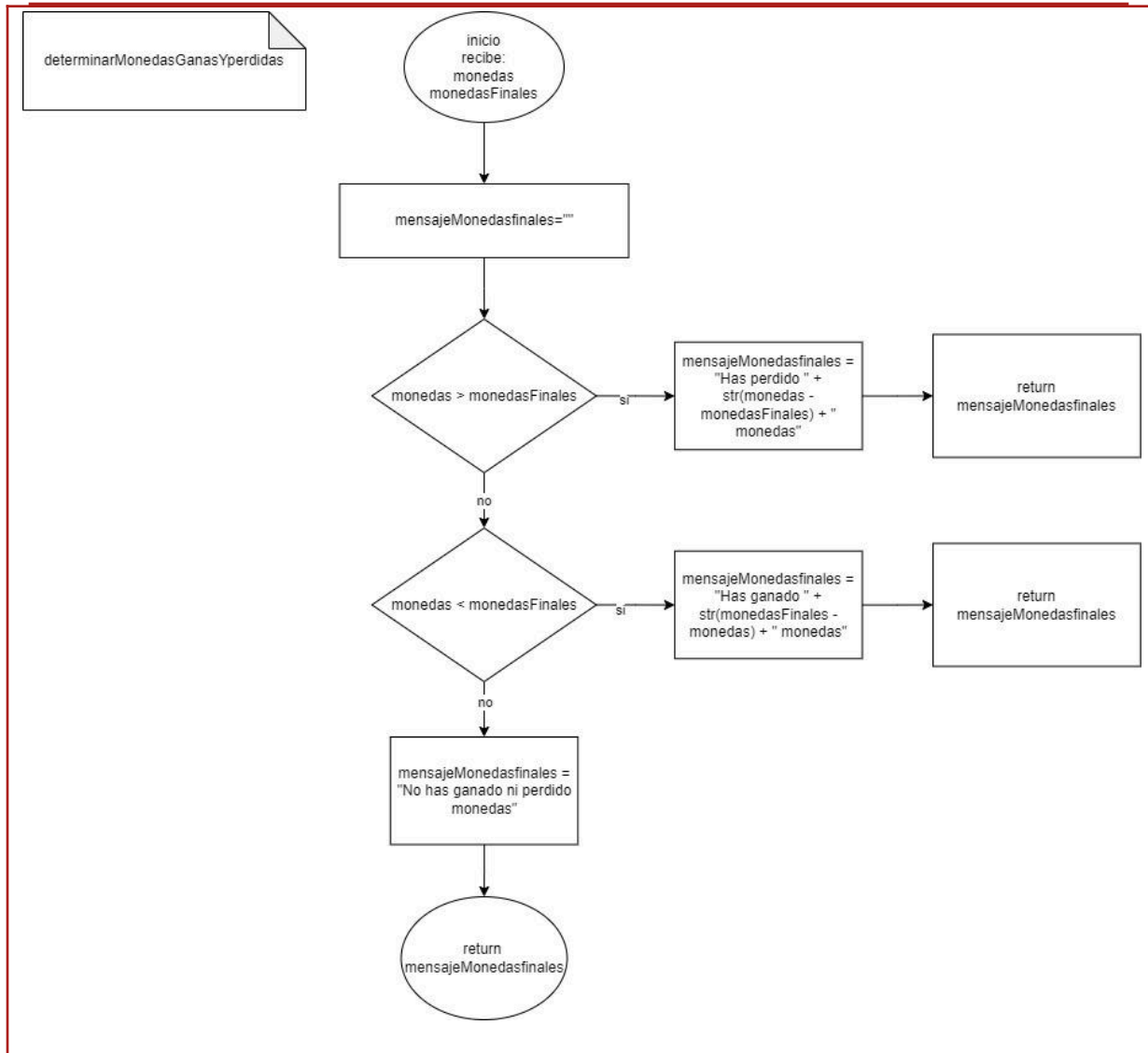
Realice la propuesta de solución para el subproceso **determinar Monedas Ganadas y/o pérdidas**



**Subproceso :**

Entradas			Salidas		
Tipo	Descripción	Variable	Tipo	Descripción	Variable
int	Total al iniciar el juego	monedas	str	Determina si perdiste y ganaste monedas	mensajesMonedasfinales
int	Total monedas al finalizar juego	monedasFinales			


**Flujograma subproceso:**

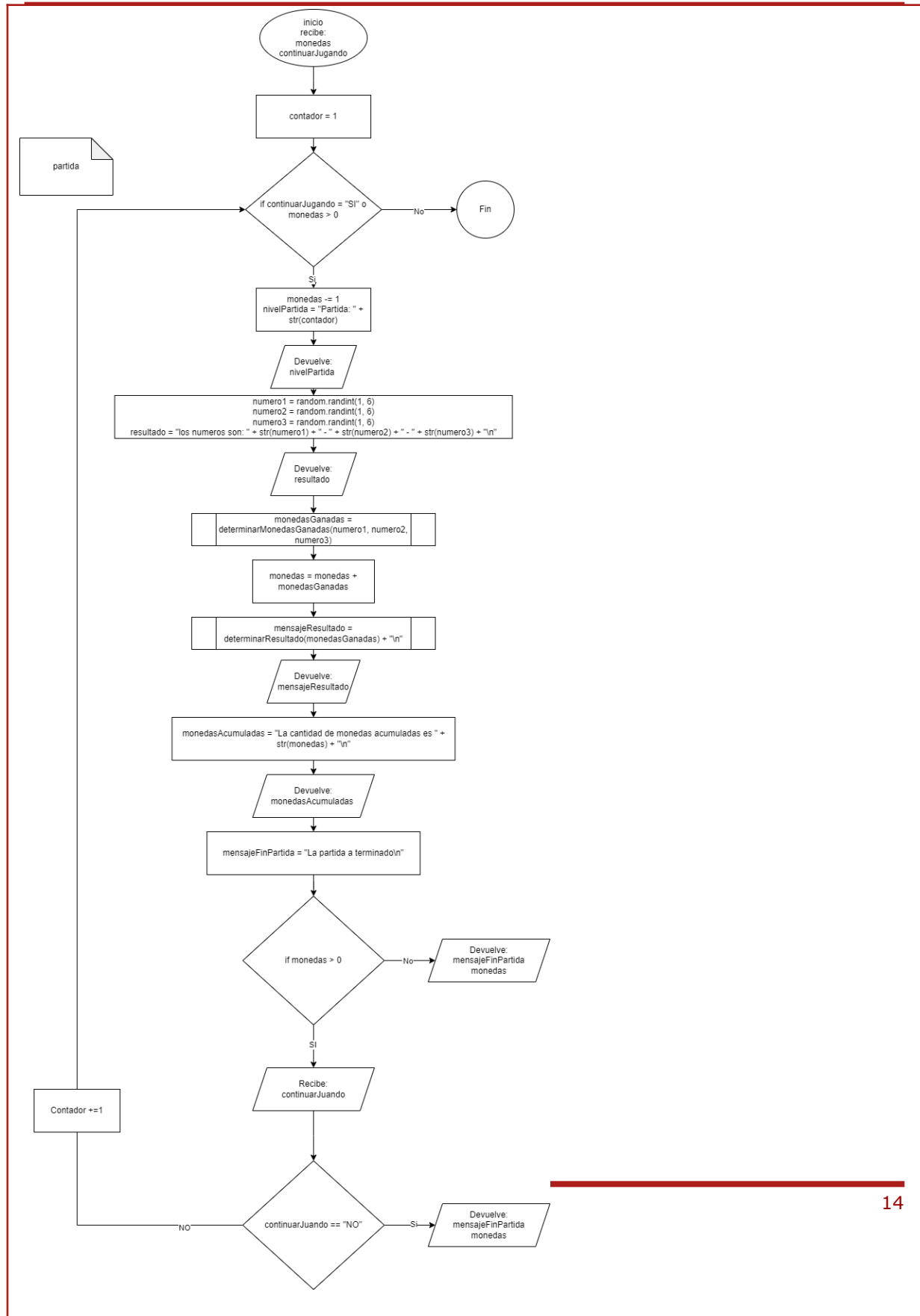


### Subproceso : partida

Entradas			Salidas		
Tipo	Descripción	Variable	Tipo	Descripción	Variable
int	Cantidad inicial de monedas que el usuario tiene	monedas	String	Indica al usuario el número de la partida	nivelPartida
String	Decisión del usuario de jugar una partida	continuarJutando	String	Devuelve al usuario los número que se generaron automáticamente	resultado
			String	Se le indica al usuario cuantas monedas ganó	mensajeResultado
			String	Le indica el total de monedas que tiene actualmente el usuario	MonedasAcumuladas
			string	Mensaje que le indica al usuario que la partida terminó	mensajeFinPartida
			int	Devuelve el total de monedas disponibles por el usuario	monedas



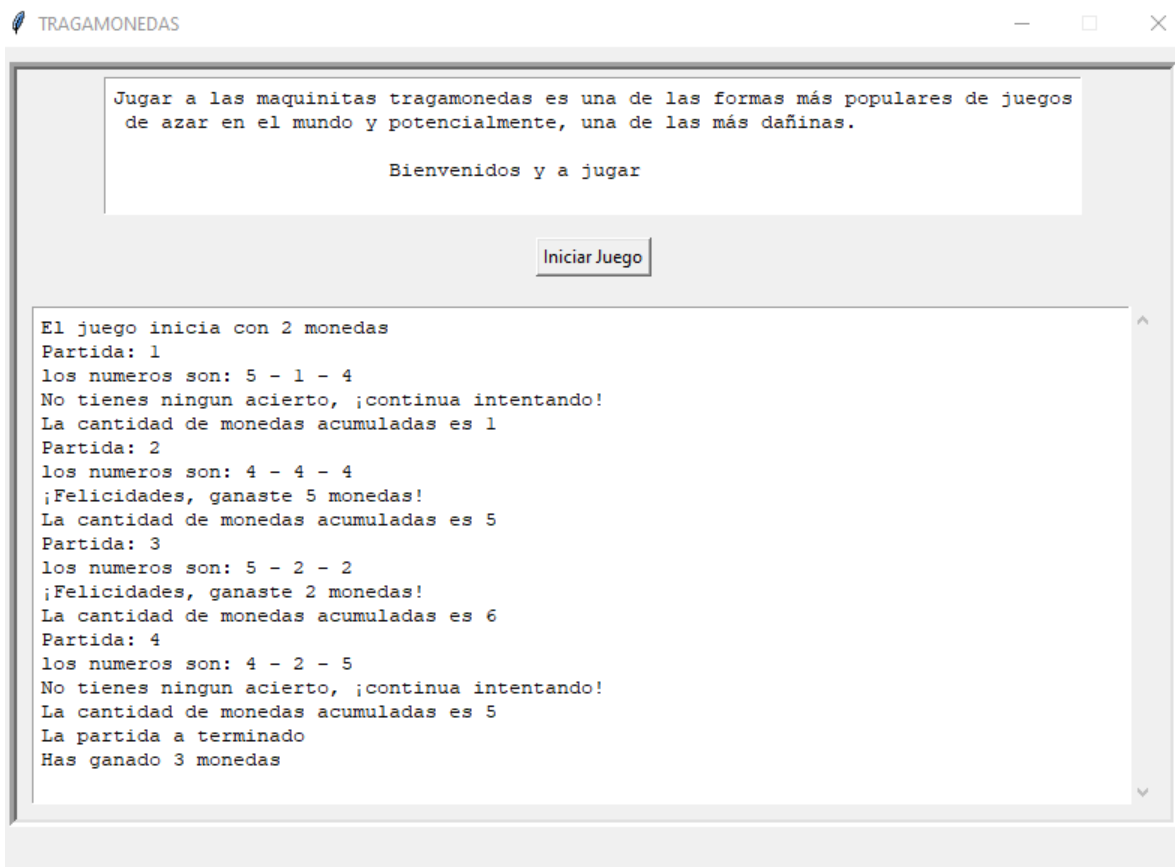
### **Flujograma subproceso:**





### PRUEBAS DE ESCRITORIO

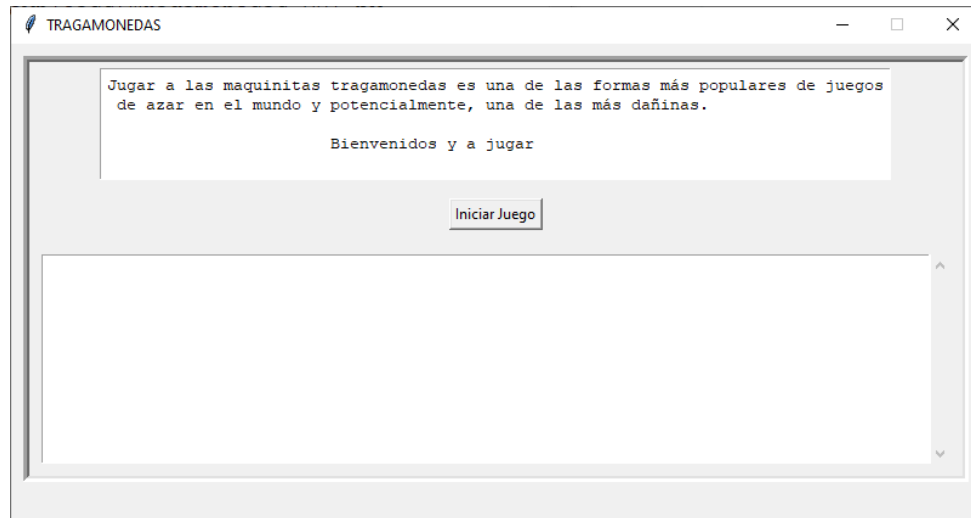
Realice pruebas de escritorio para un jugador que inicia la partida con 2 monedas:



### FASE 3: IMPLEMENTACIÓN

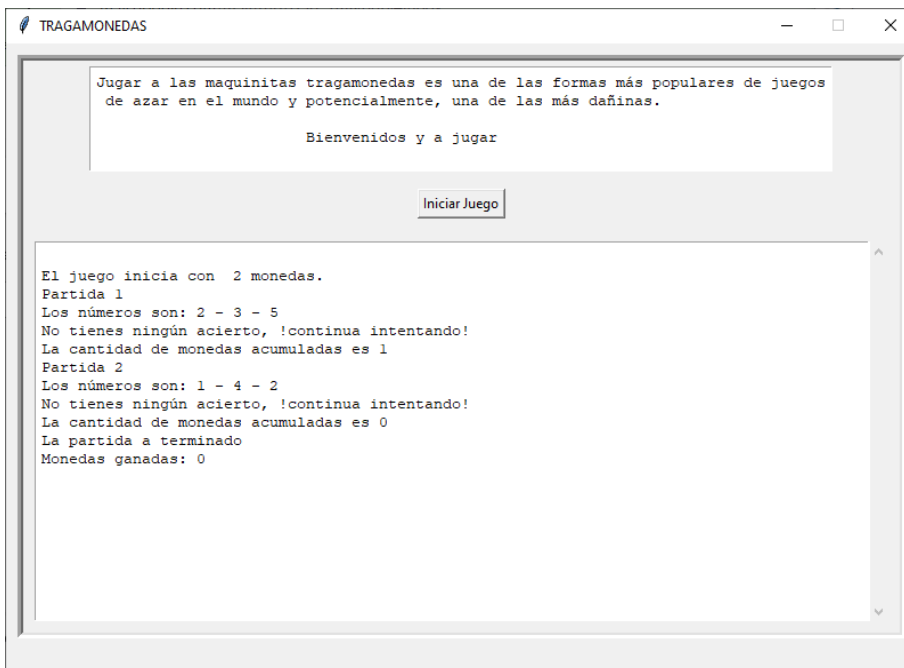
Implemente su propuesta de solución usando la siguiente [interfaz gráfica](#).





#### FASE 4: VERIFICACIÓN

La siguiente imagen muestra un juego



## ENTREGABLES

Archivo .pdf con el análisis modular por flujogramas  
Archivo .py con la interfaz gráfica y la implementación del principal  
Archivo .py con la implementación de los subprocesos