Proyecto de Laboratorio:

Análisis y diseño

Carnet: 1076925 Nombre Completo: Marco Fabrizzio Donadio Castellanos

Carnet: 1032625 Nombre Completo: Sebastian Alexander Villeda Reyna

Acciones del programa:

Enumeración de las acciones que debe realizar el programa en cada sección o módulo:

a. Elección del personaje

1. Definir personajes
2. Mostrar opciones de personajes.
    a. “(1) Mago: 100 puntos de vida y 20 puntos de ataque.”
    b. “(2) Caballero: 70 puntos de vida y 30 puntos de ataque.”
    c. “(3) Arquera: 85 puntos de vida y 25 puntos de ataque.”
3. Solicitar al usuario que seleccione una opción.
4. Leer opción
5. Case opción 1:
    5.1 aventurero = “Mago”
    5.2 vida = 100;
    5.3 ataque = 20;
6. Case opción 2:
    6.1 Aventurero = “Caballero”
    6.2 vida = 70;
    6.3 ataque = 30;
7. Case opción 3:
    7.1 aventurero = “Arquero”
    7.2 vida = 85;
    7.3 ataque = 25;
8. Devolver aventurero

b. Selección del camino

1. Mostrar “Elige el camino en el que quieres comenzar...”
    1.1 Mostrar “===== CAMINOS A ESCOGER =======
    a. Mostrar “(1) Bosque oscuro (Se pueden hallar tesoros sorprendentes o caer en trampas)");

b. Mostrar "(2) Cueva sombría (Posibilidad de encontrar enemigos sigilosos. En este mapa los enemigos golpean primero)");
c. Mostrar (3) Camino de piedra (Es menos peligroso, pero con escasos recursos. Los cofres tienen un 25% de probabilidad de estar vacíos.)");

2. Leer mapaSeleccion
3. Devolver mapa

c. Menú Principal

1. Mostrar "=====MENU PRINCIPAL DEL AVENTURERO===="
2. Mostrar "El camino que elegiste fue: {mapa}"
3. Mostrar "Tu tienes los poderes de un/a {aventurero}"
4. Mostrar "Tus puntos de vida son: {vida}"
5. Mostrar "Tu poder de ataque es de: {ataque}"
6. Mostrar "Has derrotado a: {enemigosDerrotados}"
7. Preguntar al aventurero (usuario) qué quiere hacer
   7.1 Mostrar opción 1: Continuar la aventura.
   7.2 Mostrar opción 2: Rendirte.

8 Leer decisión del usuario.
   8.1 Si decisión == 1
      8.1.1 Entonces continúa al combate
      8.1.2 menu = "continuar"
   8.2 Si decisión == 2
      8.2.1 Entonces mostrar "No me cabe duda del gran aventurero que eres... pero será a la próxima."
      8.2.2 Mostrar "¡GAME OVER!"
      8.2.3 menu = "rendirse"

d. Combate

1. Mientras vida > 0 AND vidaEnemigo > 0
2. Mostrar "===INICIA EL COMBATE==="
   Ejecutar
   2.1 Mostrar "el enemigo a vencer es: "+tipoDeEnemigo
   2.2 Mostrar "Opciones de combate:"
      2.2.1 Mostrar "(1) Luchar"
      2.2.2 Mostrar "(2) Huir (-10 puntos de vida)"
3. Leer opciónCombate

   3.1 Si opciónCombate == 2
      3.1.1 vida = vida – 10
      3.1.2 Mostrar "Huiste del combate y volverás al menú principal."

   3.2 Si opciónCombate == 1
      3.2.1 Mostrar "Inicio de combate, es tu turno de atacar"
         3.2.1.1 Si camino == 1

3.2.1.2 Declarar aparicióTrampa
3.2.1.3 Declarar trampa
3.2.1.4 Asignar trampa = aparicionTrampa.Next (1,3)
3.2.1.4.1 Si trampa == 1 y enemigos derrotados< 3
3.2.1.4.1.1 vida = vida – 8
3.2.1.4.1.2 Mostrar “¡CARAY! cuidado por donde pisas aventurero, has caído en una trampa. Has perdido 8 puntos de vida... espero no te hagan falta.”
3.2.1.4.2 vidaEnemigo = vidaEnemigo – ataque
3.2.1.4.3 Mostrar “¡Has atacado a tu enemigo, su vida se redujo a “ + vidaEnemigo + puntos de vida”

3.2.1.4.4 Si vidaEnemigo > 0
3.2.1.4.4.1 Mostrar “Es el turno del enemigo de atacar”
3.2.1.4.4.2 vida = vida – ataqueEnemigo
3.2.1.4.4.3 Mostrar "El enemigo te ataco y te hizo " + ataqueEnemigo + “puntos de daño."
3.2.1.4.4.4 Mostrar “Tu vida es de: “+ vida
3.2.1.4.4.5 Mostrar “¡Ahora es tu turno de atacar!”
3.2.1.4.4.6 vidaEnemigo = vidaEnemigo – ataque
3.2.1.4.4.7 Mostrar “¡Has atacado a tu enemigo, su vida se redujo a “ + vidaEnemigo + “puntos de vida”

3.2.1.4.5 Si vidaEnemigo <= 0 Entonces
3.2.1.4.5.1 enemigosDerrotados = enemigosDerrotados +1
3.2.1.4.5.2 Mostrar "Has vencido a " + enemigo

3.2.1.4.6 Si camino ==2
3.2.1.4.6.1 Mostrar “Inicia combate”
3.2.1.4.6.2 Mostrar” Turno del enemigo de atacar”
3.2.1.4.6.3 vida = vida - ataqueEnemigo
3.2.1.4.6.4 Mostrar "El enemigo ataque del enemigo te hizo " + ataqueEnemigo + " puntos de daño."
3.2.1.4.6.5 Mostrar “Tu vida es de: “+vida
3.2.1.4.6.6 Mostrar “¡Ahora es tu turno de atacar!”
3.2.1.4.6.7 vidaEnemigo = vidaEnemigo - ataque
3.2.1.4.6.8 Mostrar “¡Has atacado a tu enemigo, su vida se redujo a “ + vidaEnemigo + “puntos de vida”

3.2.1.4.7 Si vidaEnemigo > 0
3.2.1.4.7.1 vida = vida - ataqueEnemigo
3.2.1.4.7.2 Mostrar "El enemigo te ataco y te hizo " + ataqueEnemigo + " puntos de daño."

3.2.1.4.7.3 Mostrar “Tu vida es de: “+vida

3.2.1.4.8 Si vidaEnemigo <= 0 Entonces
3.2.1.4.8.1 enemigosDerrotados = enemigosDerrotados +1
3.2.1.4.8.2 Mostrar "Has vencido a " + enemigo

3.2.1.4.9 Si vidaEnemigo > 0
3.2.1.4.9.1 vida = vida - ataqueEnemigo
3.2.1.4.9.2 Mostrar "El enemigo te ataco y te hizo " + ataqueEnemigo + " puntos de daño."
3.2.1.4.9.3 Mostrar “Tu vida es de: “+vida

3.2.1.4.10 Si camino ==3
3.2.1.4.10.1 vidaEnemigo = vidaEnemigo – ataque
3.2.1.4.10.2 Mostrar “¡Has atacado a tu enemigo, su vida se redujo a “ + vidaEnemigo + “puntos de vida”

3.2.1.4.11 Si vidaEnemigo > 0
3.2.1.4.11.1 Mostrar “Es el turno del enemigo de atacar”
3.2.1.4.11.2 vida = vida – ataqueEnemigo
3.2.1.4.11.3 Mostrar "El enemigo te ataco y te hizo " + ataqueEnemigo + " puntos de daño."
3.2.1.4.11.4 Mostrar “Tu vida es de: “+vida

3.2.1.4.12 Si vidaEnemigo <= 0 Entonces
3.2.1.4.13 enemigosDerrotados = enemigosDerrotados +1
3.2.1.4.14 Mostrar "Has vencido a " + enemigo

e. Manejo de cofres

1. Declarar decisionAperturaCofres
2. Declarar Random contenidoCofre ()
3. Declarar cofre
   3.1.1 Mostrar “¡Encontraste un cofre! Ahora, ¿deseas abrirlo? (Si/No)
4. Leer decisionAperturaCofres
5. Si decisionAperturaCofres == “Si”
6. cofre = contenidoCofre.Next(1,4)

6.1. Si cofre==1 y mapa ==3
Mostrar “Que mala suerte has tenido, el cofre está vacío”

6.2 switch (cofre)

6.2.1 case 1:
6.2.1.1 Si vidaBase > vida
6.2.1.1.1 vida = vida + 10

6.2.1.1.2 Mostrar "Has ganado 10 puntos de vida."
6.2.1.1.3 break;

6.2.2 case 2:
6.2.2.1.1 ataque = ataque + 7
6.2.2.1.2 Mostrar "Tu ataque ha aumentado en 7 puntos."
6.2.2.1.3 break;

6.2.3 case 3:
6.2.3.1.1 vida = vida – 5
6.2.3.1.2 Mostrar "El cofre estaba envenenado. Pierdes 5 puntos de vida."
6.2.3.1.3 break;

6.3 Si mapa == 1
6.3.1 Cofre = contenidoCofre.Next(1,5)
6.3.2 case 1:
6.3.2.1 Si vidaBase > vida
6.3.2.2 vida = vida + 10
6.3.2.3 Mostrar "Has ganado 10 puntos de vida."
6.3.2.4 break;

6.3.3 case 2:
6.3.3.1 ataque = ataque + 7
6.3.3.2 Mostrar "Tu ataque ha aumentado en 7 puntos."
6.3.3.3 break;

6.3.4 case 3:
6.3.4.1 vida = vida - 5
6.3.4.2 Mostrar "El cofre estaba envenenado. Pierdes 5 puntos de vida."
6.3.4.3 break;

6.3.5 case 4:
6.3.5.1 ataque = ataque + 14

6.3.5.2 “Has encontrado un tesoro sorprendente, tu ataque ha aumentado en 14 puntos”

6.3.5.3 break;

6.3.6

7. Si decisionAperturaCofres == “No”

7.1 Mostrar “Con que estás confiado eh, vuelves al menú principal”

f. Finalización del juego

1. Si vida <= 0
   1.1 Entonces Mostrar “GAME OVER”
2. Si enemigosDerrotados == 3
   2.1 Entonces Mostrar “¡Has ganado la aventura!”
3. Sino
   3.1 Mostrar “La aventura ha llegado a su fin”

Datos del programa:

Definiremos la información o datos necesarios para completar los subprocesos. Es posible que un subproceso requiera **datos de entrada** (solicitados al usuario) y **datos principales** (no solicitados al usuario). Por ejemplo, los *puntos de salud de un bandido* es clave para llevar a cabo un combate, pero al tener siempre valor de 20 no requerimos solicitarlo al usuario.

Adicionalmente, deben decidir cuáles datos almacenarán como **variables** y nombrarlas. Por ejemplo, algunos datos como los *puntos de salud del monstruo* o *veneno del cofre* son claves, pero al tener siempre el mismo valor usted podría decidir emplear una variable/constante, o colocar su monto directamente en el código. **No olvide utilizar la notación correcta para nombrar variables**.

a. Datos globales

<Aquí incluiremos datos que requerimos en varios subprocesos simultáneamente.>

Entrada/Principal	Descripción	Tipo	Variable
Principal	Cantidad de puntos de salud del personaje elegido.	int	saludPoints
Principal	Cantidad de puntos de salud del personaje elegido (valor referencial fijo).	int	saludDefinida
Principal	Cantidad de poder ataque del personaje elegido.	int	attackPower
Entrada	Tipo de aventurero	String	aventurero
Entrada	Nombre del jugador (nombre de usuario)	String	idJugador
Principal	Enemigos derrotados	int	defeteadEnemigos
Principal	Vida de los enemigos	int	vidaEnemigos

b. Elección del personaje

<Aquí incluiremos datos que requerimos únicamente en este subproceso.>

Entrada/Principal	Descripción	Tipo	Variable
Entrada	Opción de aventurero ingresada por usuario	String	opcion
Entrada	Aventurero seleccionado	int	aventureroSeleccionado
Principal	Asignar aventurero	String	aventurero

c. Selección del camino

<Aquí incluiremos datos que requerimos únicamente en este subproceso.>

Entrada/Principal	Descripción	Tipo	Variable
Entrada	Opción de mapa ingresada	String	opcionMapa

Principal	Valor de la opción de mapa ingresado	int	mapaSeleccion

d. Menú principal

<Aquí incluiremos datos que requerimos únicamente en este subproceso.>

Entrada/Principal	Descripción	Tipo	Variable
Emtrada	Valor ingresado por usuario para decidir si continua la aventura o no.	string	decisionContinuacion
Principal	Asignación de la decisión para continuar la aventura, o al contrario.	int	decision

e. Combate

<Aquí incluiremos datos que requerimos únicamente en este subproceso.>

Entrada/Principal	Descripción	Tipo	Variable
Principal	Puntos de vida del enemigo	int	vidaEnemigo
Principal	Puntos de ataque del enemigo	int	ataqueEnemigo
Principal	Obstáculo con 25% de probabilidad de aparecer	int	trampa
Principal	Probabilidad de aparición de trampas	Random	aparicionTrampa

f. Manejo de cofres

<Aquí incluiremos datos que requerimos únicamente en este subproceso.>

Entrada/Principal	Descripción	Tipo	Variable
Entrada	Decisión del usuario de abrir o no un cofre	string	decisionAperturaCofres
Principal	Tipo del contenido del cofre	Random	contenidoCofre
Principal	Asignación de contenido al cofre	int	cofre
Principal	Tipos de cofres y probabilidades en mapas	int	mapa

g. Finalización del juego

<Aquí incluiremos datos que requerimos únicamente en este subproceso.>

Entrada/Principal	Descripción	Tipo	Variable
Principal	Finalización	string	menu

Restricciones y fórmulas

Enlistaremos las condiciones o fórmulas que usaremos en cada subproceso (si las hay). Por ejemplo, durante el manejo de cofres una restricción es que el usuario no puede recuperar una cantidad de puntos de salud que sobrepase la capacidad máxima del personaje.

a. **Elección del personaje**
    a. Validar valores de entrada y definir parámetro: 1, 2, o 3.
    b. Dependiendo de la selección, se procede a asignar el poder de ataque y salud del personaje seleccionado.
    c. Iteración (ciclo) que permita al programa seguir solicitando los datos hasta que las condiciones se cumplan.

b. **Selección del camino**
    a. Validar valores de entrada y definir parámetro: 1, 2, o 3.
    b. Asignar las características y probabilidades de los mapas.
        i. Cueva sombría: enemigos atacan primero.
        ii. Camino de piedra: 25% de probabilidad de que un cofre esté vacío.
    c. Nuevamente, una iteración (ciclo) que permita al programa seguir solicitando los datos hasta que las condiciones se cumplan.

c. **Menú Principal**
    a. Validar valores de entrada y definir parámetro: 1 AND 2
    b. Mostrar datos generales del jugador, el mapa seleccionado, enemigos derrotados y etc.
    c. Relación de datos que permita identificar lo que el usuario quiere hacer, entre: continuar con la aventura o rendirse.
    d. Un ciclo que permita validar el ingreso al menú principal después de cada batalla, por lo menos hasta que el aventurero pase al jefe final o sea derrotado.

d. **Combate**
    a. Validar valores de entrada y definir parámetro: 1 AND 2
    b. Operaciones aritméticas que permitan el desenvolvimiento de la batalla, substrayendo puntos de vida de ataques directos entre contrincantes.
    c. Un ciclo que permita la continuación de la batalla hasta que la vida de uno de los contrincantes sea 0.
    d. Si el aventurero es el vencedor procede a la siguiente batalla, o si es la batalla del jefe final, gana el juego.
    e. La vida de ningún personaje puede ser mayor a la establecida en las variables principales.

e. **Manejo de cofres**
    a. Cerciorarse que los valores ingresados estén dentro de “si” y “no”
    b. La salud del jugador no puede superar el límite máximo de salud de su personaje

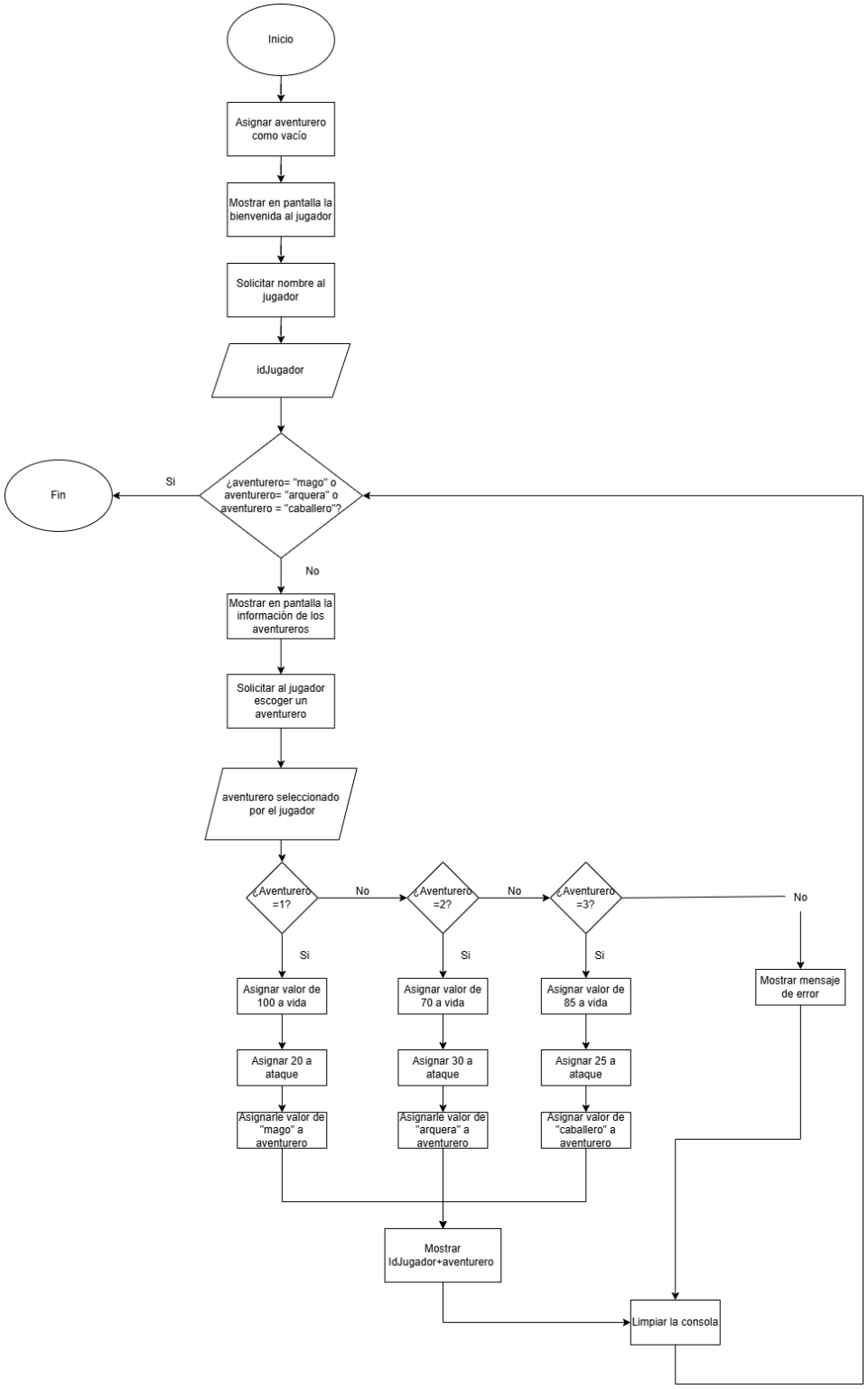
c. Para el mapa 2 el contenido aleatorio de los cofres es de {1,4}
d. Para el mapa 1 el contenido aleatorio de los cofres es de {1,5} (considerando el tesoro sorprendente)
e. Para el mapa 3 el contenido aleatorio de los cofres es de {1,5}, considerando que 1 de cada 4 cofres está vacío (25% de probabilidad de que lo estén en este mapa)

f. Finalización del juego

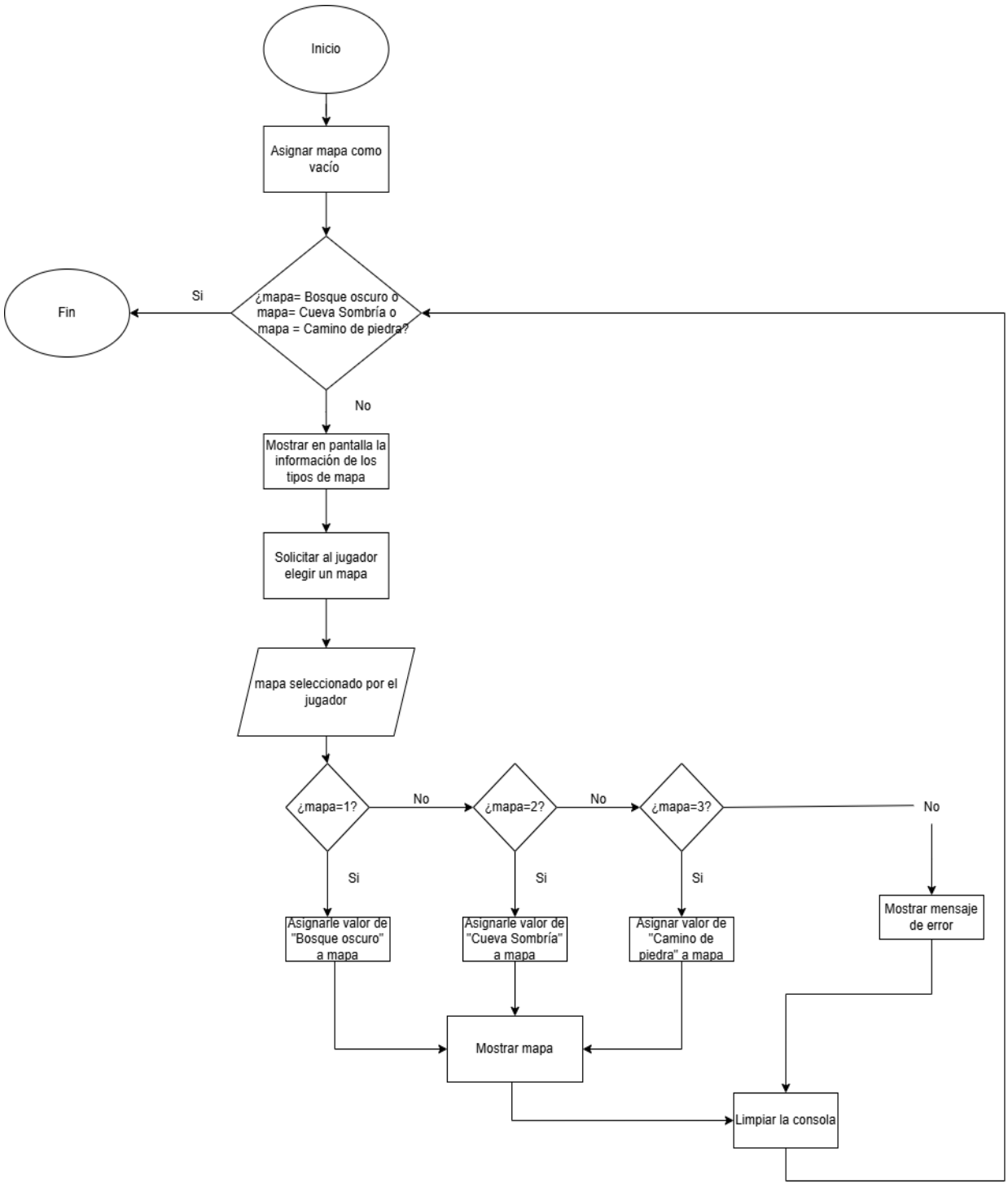
a. Para que el jugador gané tiene que vencer sí o sí a tres enemigos.
b. La vida del jugador debe ser mayor a cero, de lo contrario el jugador fue eliminado.
c. El jugador puede rendirse luego de cada uno de los primeros dos combates, incluso antes, si este lo decide.

Diagramas de flujo

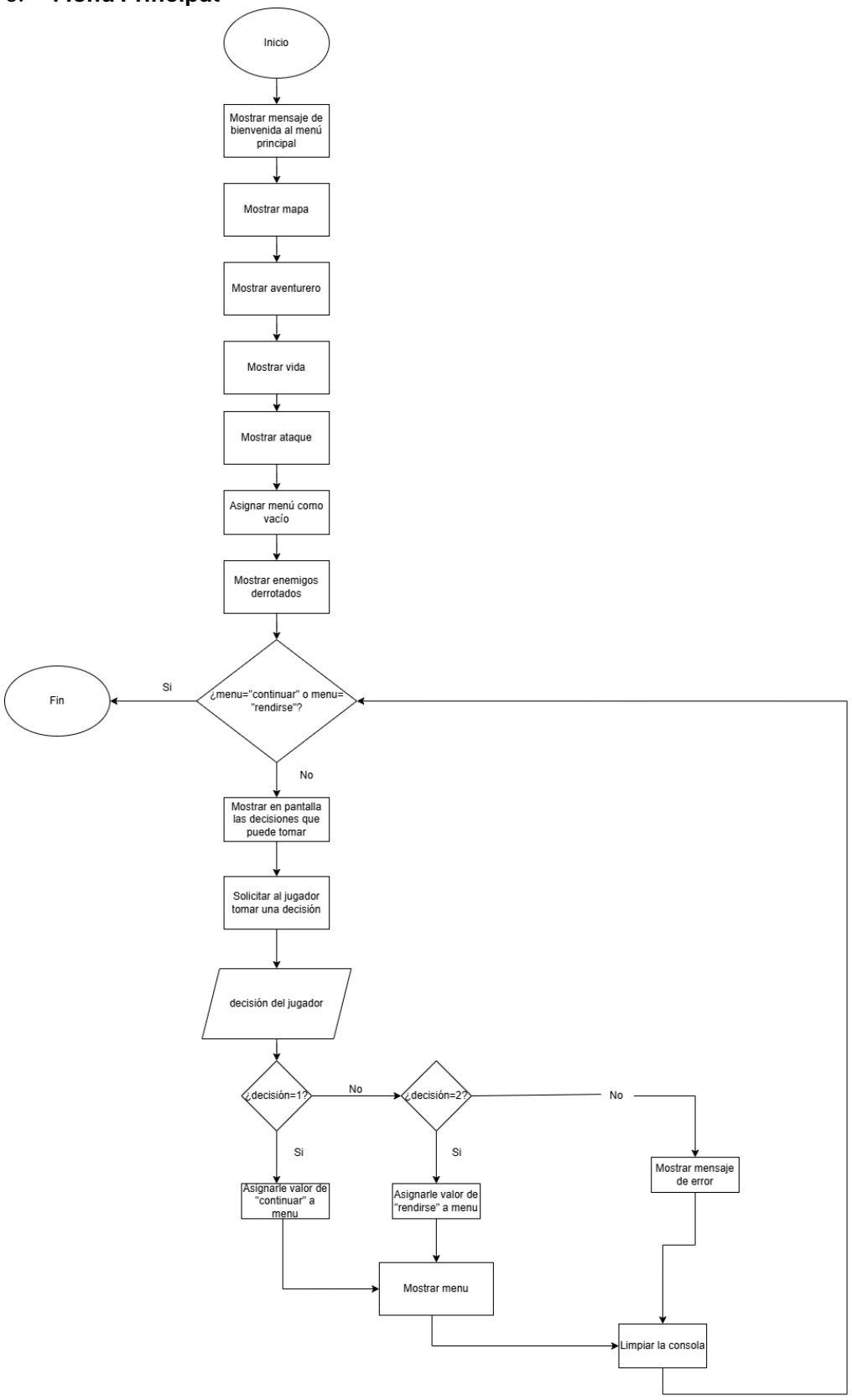
a. Elección del personaje



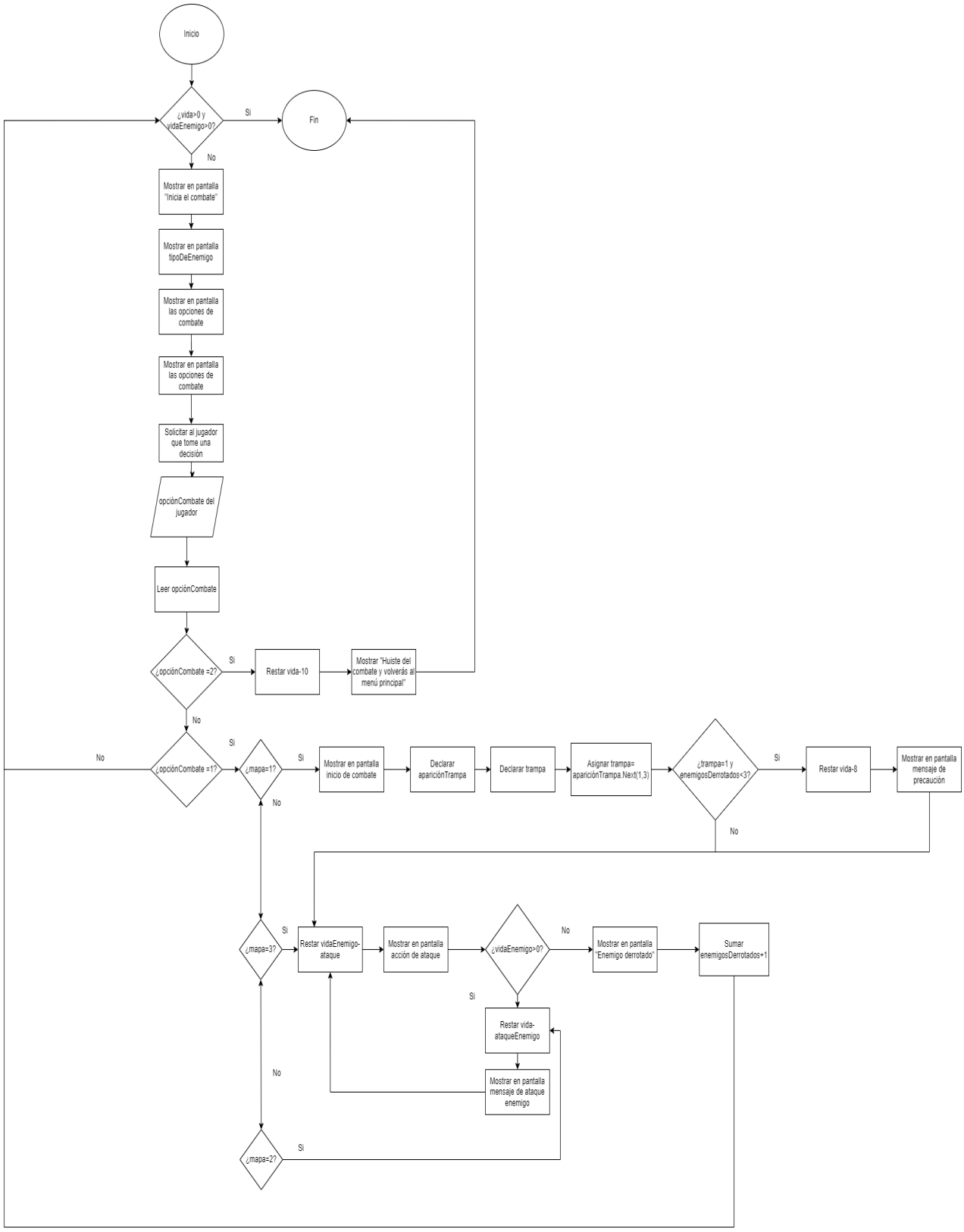
b. Selección del camino



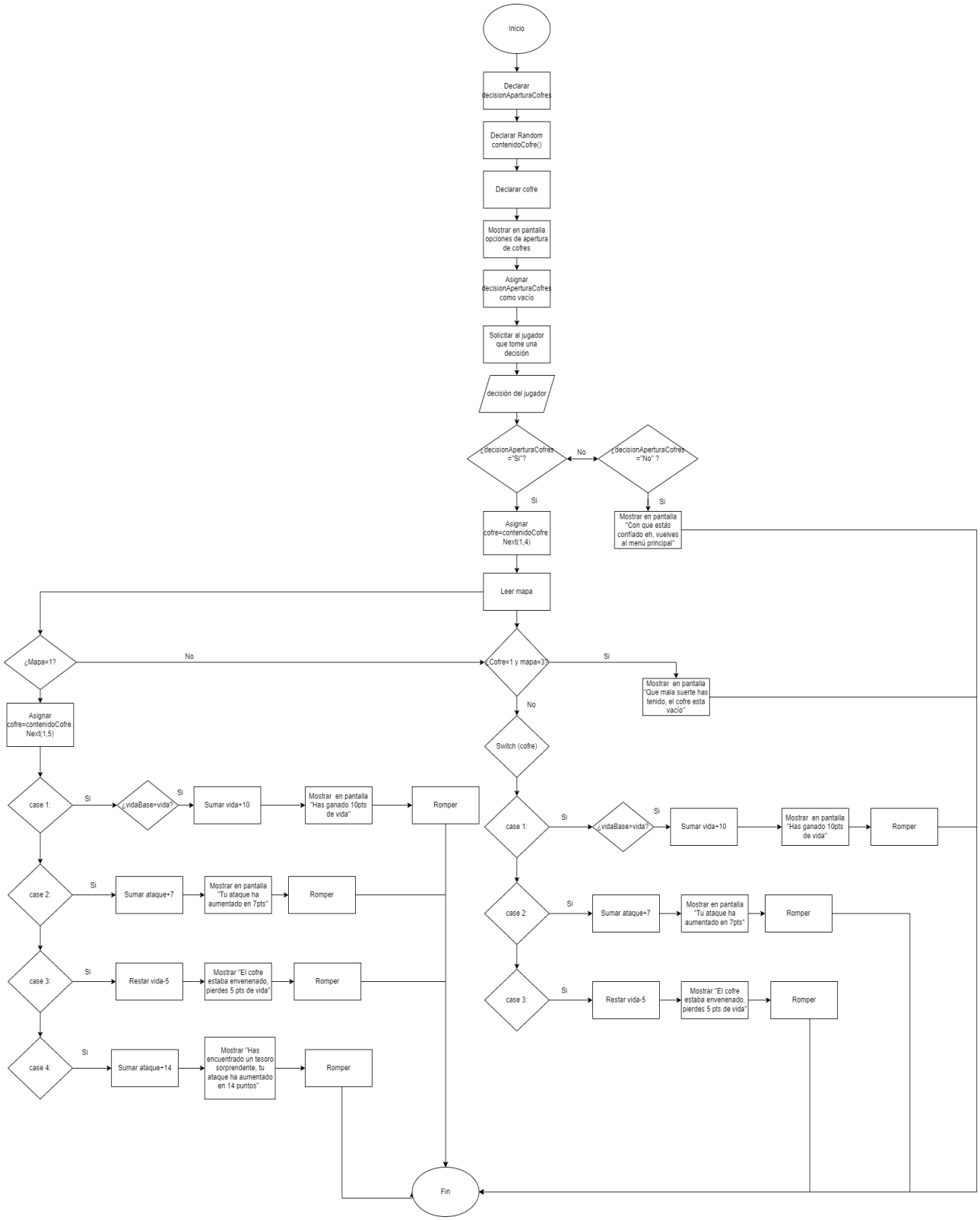
c. **Menú Principal**



d. Combate



e. Manejo de cofres


Inicio
Declarar decisionAparturaCofres
Declarar Random contenidoCofre()
Declarar cofre
Mostrar en pantalla opciones de apertura de cofres
Asignar decisionAperturaCofres como vacío
Solicitar al jugador que tome una decisión
decisión del jugador
¿decisionAperturaCofres ="Si"?
No
¿decisionAperturaCofres ="No" ?
Si
Si
Asignar cofre=contenidoCofre. Next(1,4)
Mostrar en pantalla "Con que estás confiado eh, vuelves al menú principal"
Leer mapa
¿Mapa=1?
No
Cofre=1 y mapa=3
Si
Mostrar en pantalla "Que mala suerte has tenido, el cofre esta vacío"
No
Asignar cofre=contenidoCofre. Next(1,5)
Switch (cofre)
case 1:
Si
¿vidaBase>vida?
Si
Sumar vida+10
Mostrar en pantalla "Has ganado 10pts de vida"
Romper
case 2:
Si
Sumar ataque+7
Mostrar en pantalla "Tu ataque ha aumentado en 7pts"
Romper
case 3:
Si
Restar vida-5
Mostrar "El cofre estaba envenenado, pierdes 5 pts de vida"
Romper
case 4:
Si
Sumar ataque+14
Mostrar "Has encuentrado un tesoro sorprendente, tu ataque ha aumentado en 14 puntos"
Romper
Fin

f. **Finalización del juego**