

Primera entrega- Proyecto Análisis de Algoritmos

David Enrique Palacios García¹

Karen Sofia Coral Godoy¹

¹Departamento de Ingeniería de Sistemas, Pontificia Universidad Javeriana
Bogotá, Colombia
{david_palacios, corallg_ksofia}@javeriana.edu.co

4 de octubre de 2022

Resumen

En este documento se presenta la formalización de la propuesta de la interfaz del juego *FlowFree* correspondiente a la primera entrega del proyecto del curso Análisis de Algoritmos, el cual busca dar solución al problema de conectar puntos del mismo color dibujando caminos entre ellos de modo que toda la matriz esté ocupada por caminos.

Palabras clave: matriz, interfaz, formalización, caminos, puntos.

Índice

1. Introducción	2
2. ¿Cómo jugar?	2
2.1. Instrucciones	2
2.2. Condiciones para ganar	2
2.3. Restricciones del juego	3
3. Diagrama de clases	3
3.1. Clase: Flow Free	3
3.2. Clase: Tablero	4
3.3. Clase: Camino	4
3.4. Clase: Punto	4
4. Funciones importantes	4
4.1. Flujo principal	4
4.2. Realizar Movimiento	6
4.2.1. Validar Movimiento	7
4.3. Deshacer Movimiento	8
4.3.1. Validar Deshacer Movimiento	10
4.4. Verificar Victoria	11
5. Glosario	12

1. Introducción

FlowFree es un juego que consiste en una cuadrícula de cuadrados (matriz) con pares de puntos de colores que ocupan algunos de los cuadrados. El objetivo es conectar puntos del mismo color mediante caminos que no pueden cruzarse entre ellos. La dificultad está determinada principalmente por las dimensiones de la matriz, que va desde 5×5 hasta 9×9 . En este documento se presenta el diseño de una interfaz sencilla para que un humano pueda jugar, con el objetivo de mostrar: las instrucciones para interactuar con la interfaz mencionada (sección 2.1), las diferentes clases que componen el juego (sección 3) y el flujo de los principales casos de uso (sección 4).

2. ¿Cómo jugar?

En el juego *FlowFree* tenemos una matriz que puede ser de distintos tamaños (para esta interfaz desde 55 hasta 99). En dichas casillas hay puntos de colores distribuidos en todo el tablero de forma aleatoria y se deben unir las parejas del mismo color. La cantidad de colores está determinada por el resultado de restarle a la dimensión de la matriz una unidad. Las uniones no se pueden cruzar y es necesario rellenar todas las casillas, es decir, no pueden quedar casillas vacías en el tablero.

2.1. Instrucciones

1. Ejecutar programa "*FlowFree.py*" en una terminal
2. Ingresar dimensión del tablero deseada
3. En cada iteración, escoger una acción:
 - Realizar movimiento
 - a) Ingresar fila del punto inicial (Se debe partir desde un punto original)
 - b) Ingresar columna del punto inicial (Se debe partir desde un punto original)
 - c) Ingresar fila del punto final (Debe corresponder a un punto vacío)
 - d) Ingresar columna del punto final (Debe corresponder a un punto vacío)
 - Deshacer movimiento
 - a) Ingresar fila del punto inicial (Se debe partir desde un punto no vacío y debe corresponder al último por donde se continuo el camino)
 - b) Ingresar columna del punto inicial (Se debe partir desde un punto no vacío y debe corresponder al último por donde se continuo el camino)
 - c) Ingresar fila del punto final (Debe corresponder a un punto no vacío y no original)
 - d) Ingresar columna del punto final (Debe corresponder a un punto no vacío y no original)
 - Ver puntos originales

2.2. Condiciones para ganar

Básicamente para que exista una victoria se debe cumplir con el objetivo del juego previamente descrito, unir todas las parejas de colores mediante caminos que no se cruzan, sin dejar casillas vacías en el tablero.

Específicamente la victoria es alcanzada cuando se cumplen las siguientes condiciones:

1. Todos los caminos que unen las parejas de colores están completos, es decir, el número de caminos correctos es igual al número de colores del tablero
2. El porcentaje de llenado del tablero es igual al 100 %

2.3. Restricciones del juego

1. Únicamente se permiten realizar y deshacer movimientos horizontales o verticales (NO diagonales)
2. El número de colores es igual a la dimensión del tablero menos uno
3. Sólo están disponibles matrices de dimensiones desde 5×5 hasta 9×9

3. Diagrama de clases

El juego *FlowFree* está compuesto por 4 grandes clases definidas en el siguiente diagrama de clases (Ver figura 1) que trazan claramente la estructura del sistema que se intenta modelar mediante atributos, operaciones y relaciones entre clases.

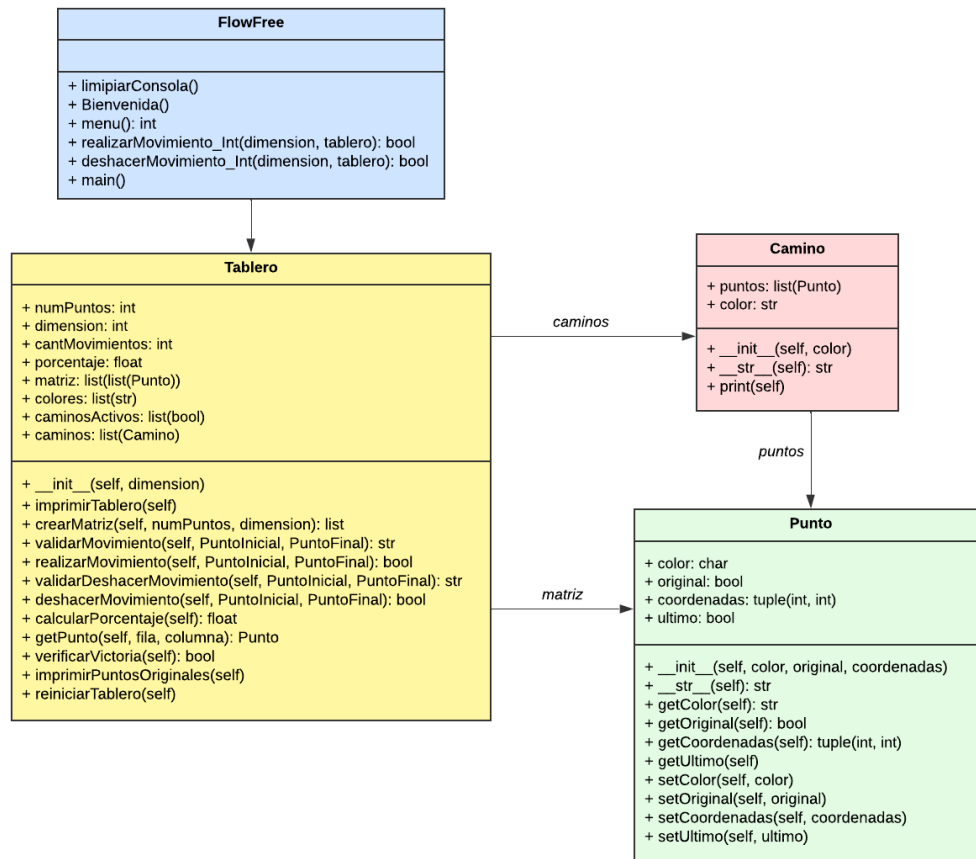


Figura 1: Diagrama de clases

3.1. Clase: Flow Free

Es la clase que ejecuta el main del juego y llama a los métodos de las demás clases. Esta clase recibe todas las entradas del usuario durante las iteraciones, valida las dimensiones de la matriz, llama a diferentes métodos de la clase *Tablero* como su creación e impresión, y permite como tal, la iteración de acciones mientras no exista una victoria.

3.2. Clase: Tablero

La clase *Tablero* compone la lógica del juego, contiene la matriz de puntos y la lista de caminos, adicionalmente sus demás atributos determinan e informan características importantes del juego como: Número de puntos, dimensión del tablero, cantidad de movimientos hechos, porcentaje de llenado, colores disponibles, etc. En cuanto a sus métodos, corresponden a la creación e impresión del tablero de juego, realización de movimientos y deshacerlos siguiendo unas respectivas validaciones previas y finalmente la verificación de la existencia de una victoria.

3.3. Clase: Camino

La clase *Camino* es muy sencilla, está compuesta por una lista de puntos que forman un camino en el tablero y un color del camino, así mismo, cuenta con operaciones sencillas de inicialización, asignación e impresión de un camino.

3.4. Clase: Punto

La clase *Punto* define el objeto principal que compone las demás clases, un punto pertenece a un color, puede ser o no original, cada punto se encuentra en una coordenada única respecto a los otros puntos, y también guarda un estado que indica si el punto fue o no el último marcado en el tablero que servirá para las validaciones de los movimientos. Presenta métodos de inicialización, asignación y obtención.

4. Funciones importantes

En esta sección se presentarán algunos diagramas de flujo de los procesos más importantes del juego.

4.1. Flujo principal

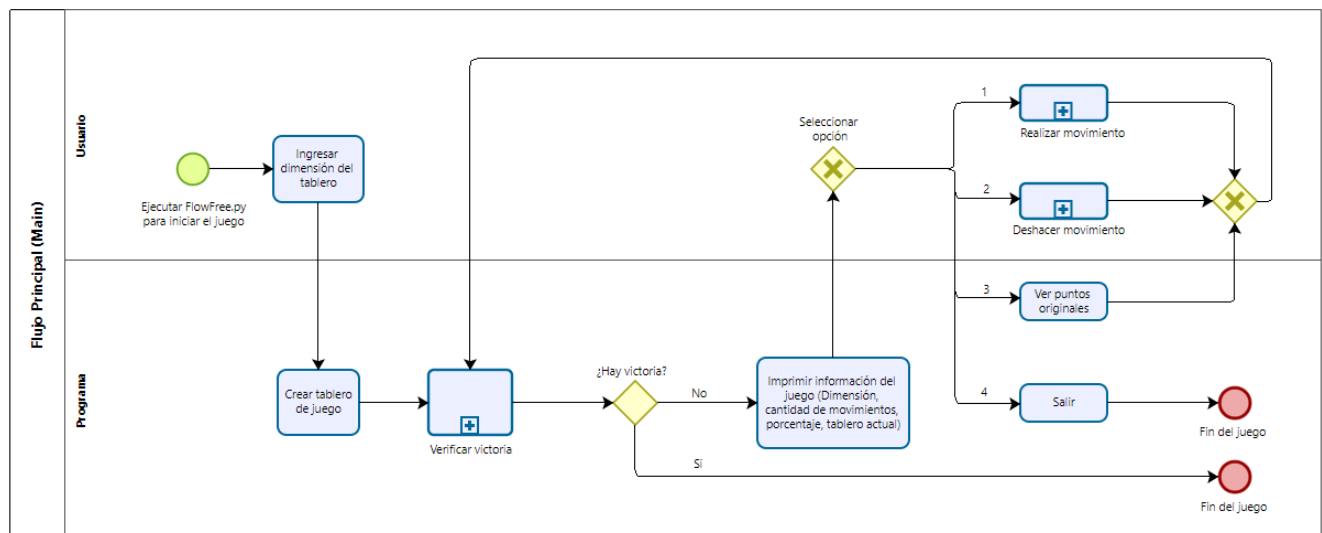


Figura 2: Flujo principal

El flujo principal determina el procedimiento general que permite jugar el juego de *FlowFree*, comienza con la ejecución del programa escrito en lenguaje *Python*, se debe abrir una terminal e ingresar el comando `python FlowFree.py`. El sistema mostrará la siguiente interfaz:

```

Primer Proyecto: Analisis de algoritmos
Profesor: Leonardo Florez Valencia
Desarrollado por: David Palacios y Sofía Coral
Bienvenido a Flow Free
Dimension del tablero-> numero de Puntos
5. 5x5 -> 4 puntos
6. 6x6 -> 5 puntos
7. 7x7 -> 6 puntos
8. 8x8 -> 7 puntos
9. 9x9 -> 8 puntos
Ingrese la dimension del tablero:

```

Figura 3: Interfaz de usuario- Inicio

Donde se le solicita al usuario ingresar las dimensiones del tablero de juego disponibles (Desde 5×5 hasta 9×9). Una vez se ingresa la entrada solicitada, el sistema procede a la creación de un tablero con las especificaciones definidas en la clase *Tablero*, distribuyendo las $|dimensión - 1|$ parejas de puntos originales de colores aleatoriamente por el tablero. Los colores estan representados por la primera letra del color, de la siguiente manera:

- R = Rojo
- V = Verde
- A = Azul
- N = Negro
- M = Morado
- C = Cyan
- B = Blanco
- G = Gris
- T = Turquesa

```

Dimension: 6 x 6
Cantidad de movimientos: 0
Porcentaje: 27.78 %
Tablero actual:
  0 1 2 3 4 5
0      M
1
2 V      M V
3  N      R
4  A      A
5      N R
Opciones:
1. Realizar movimiento
2. Deshacer movimiento
3. Ver puntos originales
4. Salir
Ingrese una opcion:

```

Figura 4: Interfaz de usuario- Tablero creado

El usuario puede realizar una de las 4 acciones presentadas en cada iteración, estas son:

1. Realizar Movimiento
2. Deshacer Movimiento
3. Ver puntos originales → Le permite al usuario visualizar el tablero original generado, es decir, en el estado inicial de creación aleatoria.
4. Salir

Si la elección corresponde a una de las 3 primeras acciones, el sistema ejecutará el método respectivo y se válida si existe una victoria para determinar una nueva iteración del usuario, por el contrario, si la opción seleccionada es la #4, el sistema da por finalizado el juego y termina rápidamente el programa.

4.2. Realizar Movimiento

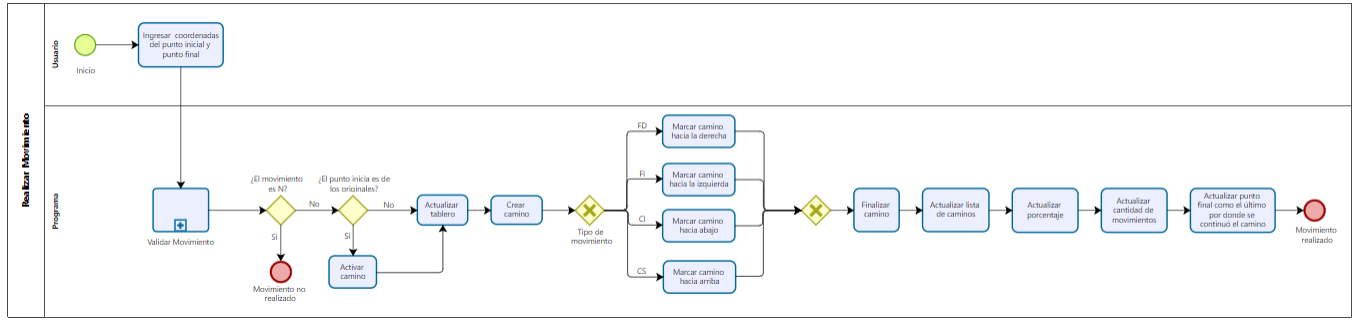


Figura 5: Flujo de realizar movimiento

Cuando la opción seleccionada es la #1, el usuario debe ingresar las coordenadas (fila y columna) del punto inicial y punto final del movimiento que desea marcar, estas deben ser válidas, es decir, encontrarse en un rango permitido (tanto filas como columnas indicadas no deben ser menores a 0 o mayores a la dimensión de la matriz).

Una vez las dimensiones son adecuadas, el sistema ejecuta el subproceso *ValidarMovimiento* que dado un punto inicial y final se clasifica el movimiento en los diferentes tipos (*N*, *FD*, *FI*, *CI*, *CS*) y se retorna el tipo asignado. Posteriormente, se verifica un primer condicional que descarta los movimientos clasificados en "N" y no los realiza, el camino alterno conduce a un nuevo condicional que evalúa si el punto inicial corresponde a un punto original, de no ser así se salta el paso de activar camino (cuando si es original) y se pasa directamente a actualizar tablero y crear un nuevo camino, allí dependiendo de los demás tipos se marca el movimiento en la dirección correspondiente:

- FD → Hacia la derecha
- FI → Hacia la izquierda
- CI → Hacia abajo
- CS → Hacia arriba

Se finaliza el camino, se actualiza la lista de caminos y el porcentaje, se incrementa la cantidad de movimientos y se determina el punto final como el último por donde se continuó el camino, finalmente el método retorna un valor booleano *True* indicando que se realizó el movimiento.

Debe tenerse presente la restricción de realizar movimientos únicamente de forma horizontal o vertical.

```

Ingrese una opcion: 1
Ingrese la fila del punto inicial: 4
Ingrese la columna del punto inicial: 1
Ingrese la fila del punto final: 4
Ingrese la columna del punto final: 3
Movimiento realizado exitosamente

Dimension: 6 x 6
Cantidad de movimientos: 1
Porcentaje: 33.33 %
Tablero actual:
 0 1 2 3 4 5
0      M
1
2 V      M V
3      N      R
4  A A A A
5      N      R
Opciones:
1. Realizar movimiento
2. Deshacer movimiento
3. Ver puntos originales
4. Salir
Ingrese una opcion: _

```

Figura 6: Interfaz de usuario- Movimiento realizado

4.2.1. Validar Movimiento

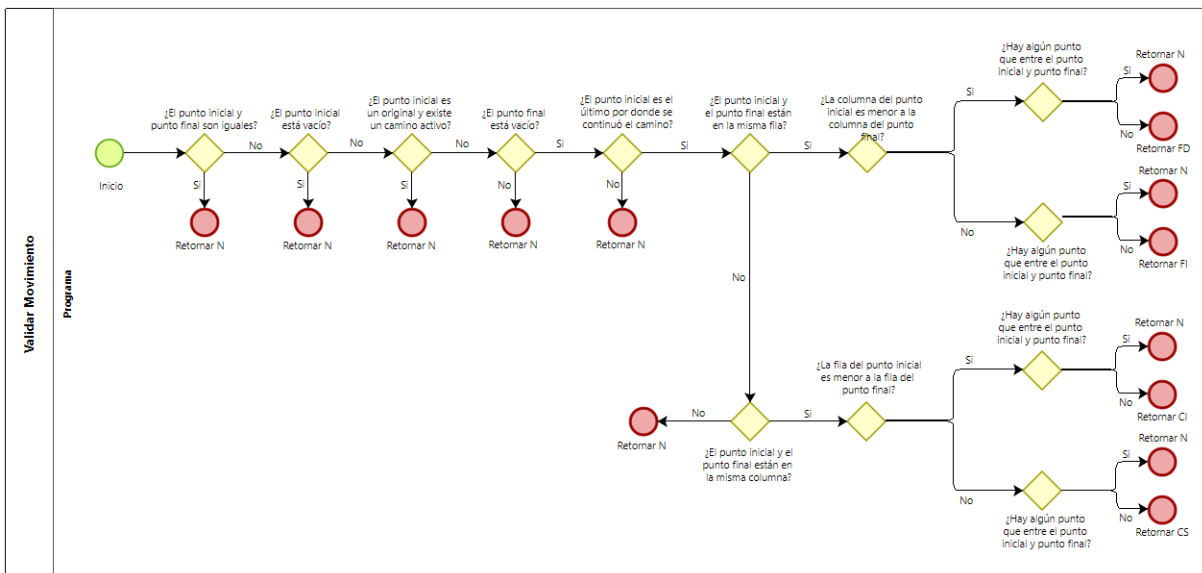


Figura 7: Flujo de validar movimiento

Este subproceso, dado un punto inicial y un punto final, le asigna un tipo (*N*, *FD*, *FI*, *CI*, *CS*) a un movimiento, teniendo en cuenta si los puntos cumplen o no los siguientes condicionales:

- $N \rightarrow$ No válido
 1. Si el punto inicial y final son el mismo (iguales)
 2. Si el punto inicial es un punto vacío
 3. Si el punto inicial es un punto original y existe un camino activo con el color
 4. Si el punto final NO es un punto vacío

5. Si el punto inicial NO es el último por donde se continuó el camino
 6. Si existen puntos entre el punto inicial y el punto final
- FD → Filas Derecha
 1. Si la fila del punto inicial es la misma del punto final y la columna del punto inicial es menor a la del punto final
 - FI → Filas Izquierda
 1. Si la fila del punto inicial es la misma del punto final y la columna del punto inicial NO es menor a la del punto final
 - CI → Columnas Inferior
 1. Si la columna del punto inicial es la misma del punto final y la fila del punto inicial es menor a la del punto final
 - CS → Columnas Superior
 1. Si la columna del punto inicial es la misma del punto final y la fila del punto inicial NO es menor a la del punto final

```

Dimension: 6 x 6
Cantidad de movimientos: 1
Porcentaje: 33.33 %
Tablero actual:
 0 1 2 3 4 5
0   M
1
2 V   M V
3   N   R
4 A A A A
5   N R
Opciones:
1. Realizar movimiento
2. Deshacer movimiento
3. Ver puntos originales
4. Salir
Ingrese una opcion: 1
Ingrese la fila del punto inicial: 2
Ingrese la columna del punto inicial: 1
Ingrese la fila del punto final: 2
Ingrese la columna del punto final: 3
ERROR: El punto inicial está vacío

```

Figura 8: Interfaz de usuario- Validar Movimiento

4.3. Deshacer Movimiento

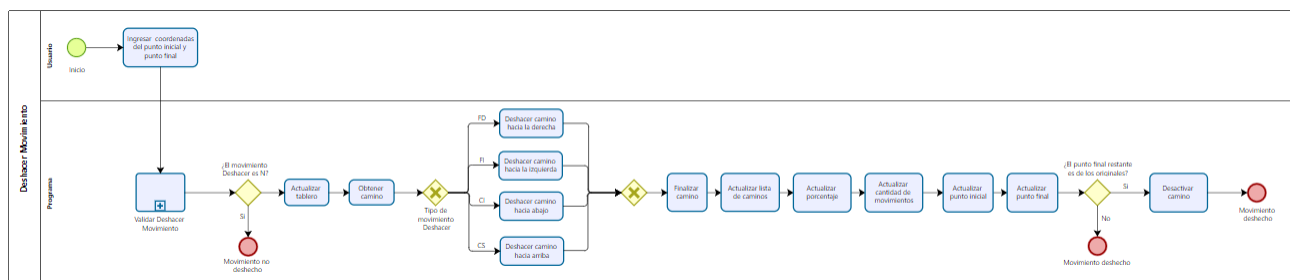


Figura 9: Flujo de deshacer movimiento

Cuando la opción seleccionada es la #2, el usuario debe ingresar las coordenadas (fila y columna) del punto inicial y punto final del movimiento que desea deshacer, estas deben ser válidas, es decir, encontrarse en un rango permitido (tanto filas como columnas indicadas no deben ser menores a 0 o mayores a la dimensión de la matriz).

Una vez las dimensiones son adecuadas, el sistema ejecuta el subproceso *ValidarDeshacerMovimiento* que dado un punto inicial y final se clasifica el movimiento deshacer en los diferentes tipos (*N, FD, FI, CI, CS*) y se retorna el tipo asignado. Posteriormente, se verifica un primer condicional que descarta los movimientos clasificados en "N" y no los deshace, el camino alternativo conduce a la actualización del tablero y obtención del camino correspondiente a los puntos, allí dependiendo de los demás tipos se deshace el movimiento en la dirección correspondiente:

- FD → Hacia la derecha
- FI → Hacia la izquierda
- CI → Hacia abajo
- CS → Hacia arriba

Se da por finalizado el camino, se actualiza la lista de caminos y el porcentaje, se incrementa la cantidad de movimientos y se actualizan los puntos inicial y final, conduciendo a un nuevo condicional que evalúa si el punto final restante corresponde a un punto original, de ser así, se desactiva el camino y se deshace el movimiento, de lo contrario se salta directamente al deshecho del movimiento.

Finalmente el método retorna un valor booleano *True* indicando que se deshizo el movimiento correctamente.

```

Dimension: 9 x 9
Cantidad de movimientos: 1
Porcentaje: 24.69 %
Tablero actual:
  0 1 2 3 4 5 6 7 8
0      R
1      R
2    G V  B N
3    M    B G
4      V  B N
5      M  B
6      C A B
7      C A B
8      C A

Opciones:
1. Realizar movimiento
2. Deshacer movimiento
3. Ver puntos originales
4. Salir
Ingrese una opcion: 2
Ingrese la fila del punto inicial a remover: 6
Ingrese la columna del punto inicial a remover: 6
Ingrese la fila del punto final a remover: 3
Ingrese la columna del punto final a remover: 6
Movimiento deshecho exitosamente

Dimension: 9 x 9
Cantidad de movimientos: 0
Porcentaje: 19.75 %
Tablero actual:
  0 1 2 3 4 5 6 7 8
0      R
1      R
2    G V  B N
3    M    B G
4      V  B N
5      M  B
6      C A B
7      C A
8      C A

```

Figura 10: nterfaz de usuario- Deshacer movimiento

4.3.1. Validar Deshacer Movimiento

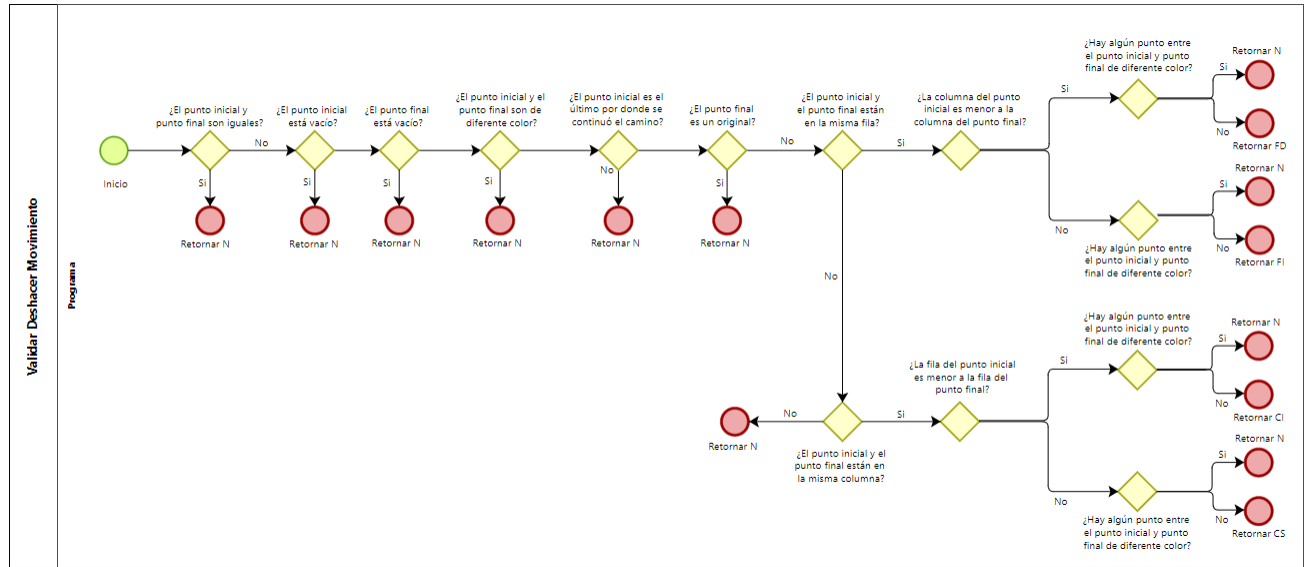


Figura 11: Flujo de validar deshacer movimiento

Este subproceso, dado un punto inicial y un punto final, le asigna un tipo (N, FD, FI, CI, CS) a un movimiento Deshacer, teniendo en cuenta si los puntos cumplen o no los siguientes condicionales:

- $N \rightarrow$ No válido
 1. Si el punto inicial y final son el mismo (iguales)
 2. Si el punto inicial es un punto vacío
 3. Si el punto final es un punto vacío
 4. Si el punto inicial y el punto inicial son de diferente color
 5. Si el punto inicial NO es el último por donde se continuó el camino
 6. Si el punto final es un punto original
 7. Si existen puntos entre el punto inicial y el punto final
- $FD \rightarrow$ Filas Derecha
 1. Si la fila del punto inicial es la misma del punto final y la columna del punto inicial es menor a la del punto final
- $FI \rightarrow$ Filas Izquierda
 1. Si la fila del punto inicial es la misma del punto final y la columna del punto inicial NO es menor a la del punto final
- $CI \rightarrow$ Columnas Inferior
 1. Si la columna del punto inicial es la misma del punto final y la fila del punto inicial es menor a la del punto final
- $CS \rightarrow$ Columnas Superior
 1. Si la columna del punto inicial es la misma del punto final y la fila del punto inicial NO es menor a la del punto final

```

Dimension: 9 x 9
Cantidad de movimientos: 1
Porcentaje: 24.69 %
Tablero actual:
  0 1 2 3 4 5 6 7 8
0      R
1      R
2      G V   B   N
3      M     B   G
4      V     B   N
5      M     B
6      C A   B
7      C A   B
8      C A
Opciones:
1. Realizar movimiento
2. Deshacer movimiento
3. Ver puntos originales
4. Salir
Ingrese una opcion: 2
Ingrese la fila del punto inicial a remover: 3
Ingrese la columna del punto inicial a remover: 6
Ingrese la fila del punto final a remover: 6
Ingrese la columna del punto final a remover: 6
ERROR: El punto inicial DEBE SER el ultimo por donde se continuo el camino

```

Figura 12: Interfaz de usuario- Validar Deshacer Movimiento

4.4. Verificar Victoria

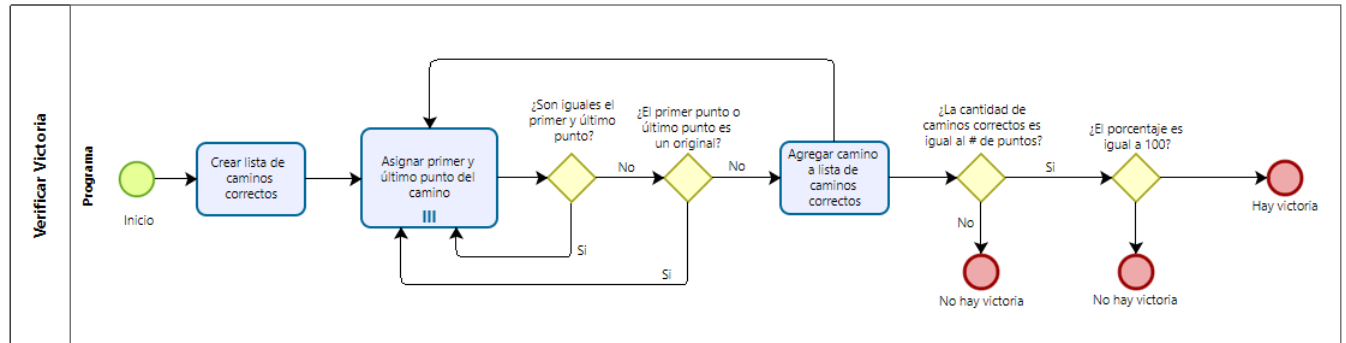


Figura 13: Flujo de verificar victoria

El sistema ejecuta varias instancia de este proceso a lo largo del juego, ya que el método tiene la finalidad de validar si en el tablero existe una victoria, mientras no se gane, se determina una nueva iteración del usuario donde podrá seleccionar nuevamente alguna de las 4 acciones permitidas en el juego:

1. Realizar Movimiento
2. Deshacer Movimiento
3. Ver puntos originales
4. Salir

La victoria se alcanza cuando todos los caminos correctos están completos. Inicialmente se crea una lista de caminos correctos, y se recorre cada uno de los caminos del tablero, guardando su estado (*True* = Correcto, *False* = No Correcto) en la lista creada. Al terminar de recorrer todos los caminos, si el número de instancias *True* de la lista de caminos correctos es igual al número de puntos (colores) definidos en la creación del tablero, se evalúa el porcentaje de llenado del tablero y si es igual al 100 %, el método retorna un valor booleano *True* indicando que hay una victoria.

5. Glosario

- Punto original → Corresponde a un punto de color generado desde la creación del tablero
- Punto vacío → Corresponde a un punto de color que no se encuentra marcado con ningún color
- Camino correcto → Corresponde a un camino que cumple con las siguientes condiciones:
 1. El primer y último punto son diferentes
 2. El primer y último punto son originales

Referencias

- [1] WIKIPEDIA, *Flow Free*, https://en.wikipedia.org/wiki/Flow_Free