



**Universidad Simón Bolívar**

**Departamento de Computación y Tecnología de la Información**

**Ingeniería de la Computación**

**Laboratorio de Algoritmos y Estructuras 1 CI-2691**

# **La Vieja Tridimensional**

**Profesor de Laboratorio:**

Prof. Carolina Chang

**Integrantes:**

Adelina Figueira 15-10484

Leonardo Duarte 17-10169

11-06-2019

## Introducción

La vieja tridimensional consiste en un juego de 3 tableros iguales a los que se juegan en una vieja convencional donde los jugadores colocarán alternativamente sus fichas y gana quien logre realizar la mayor cantidad de líneas horizontales, verticales o diagonales. Las líneas pueden realizarse al igual que una vieja tradicional; vertical, horizontal y diagonal en un tablero y vertical, horizontal y diagonal entre tableros.

En este proyecto se realiza la implementación de este juego para una N cantidad de tableros, es decir, se permite al usuario jugar desde el tablero de 1x1 hasta un tablero de 11x11 en el cual se generarán 11 tableros de dimensión 11x11, a su vez en esta versión de la vieja tridimensional no se toman en cuenta las líneas diagonales entre tableros para el puntaje final de cada jugador.

Este proyecto se incentivó debido a la necesidad de implementar las herramientas aprendidas en el curso del laboratorio de Algoritmos y Estructuras 1. Además, al realizar el proyecto se optó por utilizar un estilo sencillo pero bien detallado y funcional que permitiese hacer uso de los conocimientos aprendidos en el curso e introducir el desarrollo de interfaces gráficas.

Se tiene como objetivo principal la elaboración de un programa en Python que permite jugar la vieja tridimensional con N tableros de NxN, conformando un super-tablero cúbico. El programa utilizará el módulo Pygame que permitirá la realización de la interfaz por medio de Python. Este juego debe contener las siguientes funcionalidades:

- Permitir el ingreso del nombre de los jugadores
- Indicar el jugador a quien le toca hacer la jugada
- Permitir varias partidas, luego de cada partida los jugadores deciden si continúan jugando o no
- Iniciar la primera partida al azar y las siguientes deben alternarse
- Realización de un tablero gráfico con el cual pueda interactuar el usuario
- Resaltar en el tablero la línea que ha sido realizada

Gracias a la utilización del módulo Pygame se logrará realizar una interfaz apropiada para la implementación de la lógica desarrollada para el programa en cuestión. La lógica de este programa hace uso de diversas estructuras de datos que permiten una sencilla implementación del juego de la vieja tridimensional a su vez que la interfaz es de fácil manejo para el usuario.

El presente informe presentará todos los detalles relacionados al diseño del programa, como lo son el análisis descendente, los procedimientos, funciones y la demostración de los mismos, a su vez la implementación del juego, los tipos de datos utilizados, la estructuración del código y el estado actual del programa, sus fallos y modo de operación.

## Diseño

### Resultado del análisis descendente del problema

El juego de la vieja tridimensional ha sido implementado en Python a través del uso de análisis descendente, donde el problema de generar un programa que implemente dicho juego ha sido llevado a una serie de subproblemas por medio de funciones que resuelven cada uno de estos subproblemas. Además se utiliza una clase llamada Jugador en la cual se almacenan los datos de cada uno de los jugadores; como nombres, turno, cantidad de líneas y los puntos que llevan acumulados. El programa consiste en un total de 38 funciones, las cuales son explicadas a continuación:

Función	Subproblema
Añadirlínea	Modifica la variable almacenada en la clase jugador añadiendo las mismas en la variable del jugador correspondiente
Añadirpuntos	Modifica la variable almacenada en la clase Jugador para los puntos, añade los puntos correspondientes en la variable jugador y se ha establecido un valor de 1000 puntos por cada línea.
cambiarjugador	Modifica el jugador para alternar entre ellos y además cambia el aspecto del cursor dependiendo del jugador que esté jugando la partida.
click	Invierte los colores de un rectángulo al presionarlo.
líneaDiagonalHecha	Verifica si se ha realiza una línea diagonal en un tablero, se añaden los elementos de la diagonal a una lista llamada líneas que luego será verificada por la función verificarlista
lineaHecha	Determina cuántas líneas en total ha realizado un jugador y retorna la suma de estas

lineaHorizontalHecha	Determina si se ha realizado una línea horizontal por medio de la función verificarlista
lineaVerticalHecha	Determina si se ha realiza una línea vertical añadiendo en dos listas los valores del super-tablero que luego se verifican con la función verificarListacompleja
verificarLista	Verifica si todos los elementos de una lista son iguales
verificarListacompleja	Verifica si hay una línea por medio de iteraciones utilizando la función verificarLista
dibujarCasillas	Se utiliza para dibujar las casillas de un tablero, es exclusiva de la interfaz
dibujarCositos	Se utiliza para dibujar cada tablero jugable y devuelve cada casilla en ellos.
dibujarFicha	Dibuja la ficha del jugador de turno en el espacio donde da click el usuario y si se hizo una línea, la dibuja.
dibujarMenu	Esta función dibuja todo el menú principal que es el primero que observa el usuario al abrir el juego
dibujarRectInteractivos	Genera rectángulos especiales para el menú de inicio
dibujarRectanguloWin95	Genera rectángulos en la posición especificada y con el grosor deseado
dibujarTablero	Dibuja rectángulos con esquinas redondeadas que corresponden a cada uno de los N tableros
dibujarTexto	Introduce un texto en la pantalla, en la posición y colores deseados
dimension	Obtiene la dimensión a jugar que es dada por el usuario y genera un error si es mayor a 11 permitiéndole al usuario ingresar de nuevo la dimensión deseada
elegirjugador	Elige aleatoriamente un jugador por medio

	de la variable turno
entradatexto	Permite el ingreso del texto en el submenú donde se ingresan los nombres y la dimensión, genera los rectángulos interactivos donde hace click el usuario
error	Determina si una jugada es válida y da un mensaje en la pantalla si se hace click en una casilla que ya ha sido tomada
jugarDeNuevo	Devuelve las variables iniciales a sus valores por defecto y así se comienza una nueva partida
esValida	Determina si una jugada es válida por medio de la lista super-tablero
guardarpuntajes	Guarda los puntajes obtenidos en la partida en un archivo txt que luego es usado en la sección de rankings
imprimirTablerosRestantes	Dibuja los tableros que sobran al cambiar de tablero
jugadores	Inicializa las variables relacionadas a los jugadores que se encuentran contenidas en la clase Jugador
letrajuego	Dibuja los textos secundarios de ayuda para el usuario y los tableros secundarios de juego, que no están siendo jugados por quien se encuentra en la partida
rancc	Esta función dibuja el submenú de los rankings donde se encuentran los jugadores con las mayores puntuaciones
ratonSobre	Determina si el ratón está sobre un rectángulo del menú y lo resalta
ratonSobreCasillas	Determina si el mouse está sobre una casilla
reflejarEnTablero	Reemplaza la casilla escogida en la lista super-tablero con el número que identifica al jugador
resultado	Dibuja el submenú que muestra el resultado en la pantalla a los jugadores
salir	Dibuja una pantalla final con un mensaje

	para los jugadores al finalizar el juego
variables	Inicializa la lista super-tablero en 0
variablesPygame	Contiene las variables utilizadas por Pygame como colores y tamaños de la pantalla
main	Esta es la función principal donde ocurre el while principal y es donde se llaman a todas las funciones y se estructura el programa, igualmente se desarrollan los eventos, se inicializa y finaliza pygame dentro de ella

### **Especificación y desarrollo de los programas, procedimientos y funciones, demostraciones de correctitud**

Las especificaciones y pruebas de las funciones para hallar las líneas y determinar si una jugada es válida se encuentran en el anexo de este informe.

## **Detalles de la implementación**

### **Tipos de datos**

Para facilitar la comunicación entre la parte lógica y gráfica del programa, se utilizaron dos esquemas de tableros: una lista tridimensional que contiene los tableros jugables en forma de listas de enteros, las filas y las casillas (representadas con 0,1 y 2) denominada super-tablero. Y una lista de superficies que contienen las casillas de cada tablero, las superficies son objetos especiales de Pygame que permiten reflejarse en la pantalla.

Las funciones de verificar si hay una línea trabajan alrededor del tablero de enteros. La función que verifica si hay línea horizontal toma la fila de la posición jugada. La función que verifica si hay línea vertical toma la línea vertical o columna, tanto la del tablero como la que está entre tableros, y coloca los elementos de cada línea en una lista correspondiente, luego se verifican esas listas. Igualmente, la función que verifica las diagonales coloca los elementos de cada una en una lista y luego verifica esa lista.

Las imágenes utilizadas se dibujan sobre una superficie y luego esa superficie se dibuja en la pantalla.

### **Estructuración del código**

El código consiste en un ciclo que revisa los eventos realizados en cada vuelta y un ciclo utilizado cuando se inicia el juego para registrar las jugadas hechas en cada tablero. En cada ciclo se llaman a los procedimientos y funciones descritos anteriormente.

Los eventos principales se detectan cuando:

- El usuario presiona el botón izquierdo del ratón en algunos rectángulos que sirven de botones y de casillas.
- Se presiona una tecla del teclado, se utiliza para registrar el nombre de los jugadores y la dimensión de los tableros. Además, se utilizan las flechas para cambiar el tablero jugable actual en el juego.
- Se presiona el botón de salir en la ventana del programa, para salir del ciclo principal.

Los ciclos y las funciones son llamadas por la función main, que es la función principal donde ocurre el ciclo fundamental del programa y es donde se llaman a todas las funciones y se estructura el programa, igualmente se desarrollan los eventos, se inicializa y finaliza pygame dentro de ella.

## **Estado actual**

### **Operatividad del programa**

El programa no presenta anomalías en cuanto a la jugabilidad. Sin embargo, en la sección de rankings al momento de guardar los puntajes de los jugadores, no muestra el mayor puntaje obtenido, solo muestra los puntajes obtenidos en partidas anteriores.

### **Manual de operación: Nombre del archivo a ejecutar y modo de operar el programa**

El archivo a ejecutar es vieja3D.py, está en la raíz de la carpeta que contiene los archivos. Al iniciar el programa se observa una ventana que sirve de menú, allí se pueden clicar los botones correspondientes a las funcionalidades que tiene el programa. Estos botones son:

- Jugar: Muestra la pantalla donde se obtienen los nombres de los jugadores y la dimensión de los tableros. Luego inicia el juego.
- Rankings: Muestra la pantalla de los puntajes obtenidos en las partidas anteriores por cada jugador.
- Salir: Muestra la pantalla final del programa y cierra el mismo.
- Extras: Muestra la pantalla de extras, una sección especial del programa donde se pueden reiniciar los puntajes y activar la opción de jugar con mayor facilidad en tableros de altas dimensiones.

Al hacer click en el botón jugar se despliega un submenú donde se colocan los nombres de los jugadores y la dimensión con la que se jugará, al presionar la tecla enter en el rectángulo de dimensión inicia el juego y los jugadores deberán moverse entre tableros por medio de las teclas de dirección (arriba y abajo) y podrán realizar sus jugadas por medio del

mouse al hacer click en la casilla deseada, una vez que todas las casillas han sido llenadas finaliza la partida.

Al finalizar la partida se despliega un submenú que indica los puntajes obtenidos y si los jugadores desean volver a jugar pueden realizarlo y se dirige al submenú inicial donde se colocan los nombres y la dimensión en el cual deberán colocar con qué dimensión desean jugar. En caso contrario se despliega otra pantalla con un mensaje de salida para los usuarios.



## **Conclusiones**

Durante la realización de este proyecto se lograron cumplir todos los objetivos planteados. La parte lógica funciona tal cual fue planeada y el resultado de la parte gráfica fue satisfactorio. Además se logró profundizar en el aprendizaje del lenguaje Python y se pueden observar las diferencias de organización, optimización y utilidad del código entre el primer programa realizado en el laboratorio y este proyecto. Sin embargo, este es sólo el comienzo para seguir en el camino de ser maestros en el lenguaje.

De igual manera, se descubrieron nuevos horizontes con Pygame. Este proyecto se inició sin saber nada acerca de la librería, y ahora se tiene el conocimiento necesario para crear nuevos y mejores proyectos utilizando Pygame siendo este proyecto una introducción al desarrollo de interfaces. Por otra parte, se utilizó la plataforma GitHub y la aplicación Git para el control de versiones del proyecto en la última parte del desarrollo del mismo.

En otro orden de ideas se presentaron algunos inconvenientes en cuanto a la comunicación entre el grupo involucrado en el proyecto. Debido a la falta de dispositivos de comunicación o mensajería instantánea, fallas en el servicio de electricidad e internet o problemas generales en la universidad, se vió en la necesidad de utilizar fuentes alternativas como encuentros en la universidad, correo electrónico, Google Hangouts, etc. Sin embargo, el proyecto se terminó con la colaboración de todos los integrantes del grupo.

Finalmente planteamos una serie de recomendaciones para este proyecto, como la optimización del código ya que algunas funciones pueden escribirse nuevamente, mejorar legibilidad y simplificarse, además se propone el uso de una librería distinta a Pygame ya que, esta librería posee herramientas limitadas y también se pueden agregar sonidos o animaciones en el diseño del juego. También es posible realizar una interfaz en tres dimensiones la cual sería más agradable a la vista para el usuario pero en este caso se necesitaría replantear la lógica del juego, ya que al ser una interfaz tridimensional no es posible utilizar la lógica desarrollada para una interfaz bidimensional.

## **Bibliografia**

Pygame Developers: Pygame Documentation Release 1.9.2.

<https://python101.readthedocs.io/pl/latest/downloads/pygame192.pdf>. Marzo 2014.

Pygame Developers: Pygame Documentation Release 1.9.5. <https://www.pygame.org/docs/>.  
Julio 2016

Python Developers: Python 3.7.3 Documentation. <https://docs.python.org/3/> . Marzo 2019

## ANEXOS

## Demostraciones

A continuación se presentan las pruebas de las funciones que se implementan en el juego de la Vieja tridimensional. Se demostrarán las funciones que cuentan las líneas y las que determinan si una partida es válida o no.

### Procedimiento que cuenta las líneas diagonales

```

proc lineadiagonalHecha(in supertablero: array [0..N)x[0..M)x[0..S] of
int; in tablero: int; in fila: int; in casilla: int)
  { Precondicion  $N \geq 0 \wedge M > 0 \wedge S > 0$  }
  { Postcondicion  $(\forall i : 0 \leq i < M : (\forall j : 0 \leq j < N \wedge i = j : \text{diagonalenTablero}[i] = \text{supertablero}[i][i][i])) \wedge (\forall i : 0 \leq i < M : (\forall j : 0 \leq j < N \wedge i \neq j : \text{True}) \wedge j \geq i \wedge (\forall i : 0 \leq i < N : (\forall j : 0 \leq j < M \wedge (j = N - 1 - i) : \text{diagonalsecundaria}[i] = \text{supertablero}[i][i][i])))$  }
  [vari, j : int
  vardiagonalenTablero : array[0..M) of int
  vardiagonalsecundaria : array[0..M) of int
  i, j := 0, 0
  { Invariante  $(\forall j : 0 \leq j < i : \text{diagonalenTablero}[j] = \text{supertablero}[j][j][j]) \wedge 0 \leq i \leq M$  }
    doi  $\neq M \wedge (fila = casilla) \rightarrow \{ \text{Invariante } j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N \}$ 
      doj  $\neq N \rightarrow \text{if } i = j \rightarrow \text{diagonalenTablero}[i] = \text{supertablero}[i][i][i]$ 
        []  $i \neq j \rightarrow \text{Skip}$ 
        fi
        j := j + 1
      od
      i, j := i + 1, 0
    od
  { Invariante  $(\forall j : 0 \leq j < i : \text{diagonalsecundaria}[j] = \text{supertablero}[j][j][j]) \wedge 0 \leq i \leq M$  }
    doi  $\neq M \wedge (casilla = M - fila) \rightarrow \{ \text{Invariante } j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalsecundaria}[p] = \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N \}$ 
      doj  $\neq N \rightarrow \text{if } (j = N - 1 - i) \rightarrow \text{diagonalsecundaria}[i] = \text{supertablero}[i][i][i]$ 
        fi
        j := j + 1
      od
      i, j := i + 1, 0
    od
  []

```

*Prueba 1*

$$N; 0 \wedge M > 0 \wedge S > 0 \Rightarrow (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge 0 \leq i \leq M) \wedge j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N)[i, j := 0, 0]$$

$$\begin{aligned} & (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge 0 \leq i \leq M) \wedge j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N)[i, j := 0, 0] \\ \equiv & \quad \{SustitucionTextual\} \\ & (((\forall q : 0 \leq q < 0 : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge 0 \leq 0 \leq M) \wedge 0 \geq 0 \wedge (\forall p : 0 \leq p < 0 : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge 0 \leq 0 \leq N) \\ \equiv & \quad \{Rangovacio\} \\ & True \wedge 0 \leq 0 \leq M) \wedge 0 \geq 0 \wedge True \wedge 0 \leq 0 \leq N) \\ \equiv & \quad \{p \wedge True \equiv p, a \geq a \equiv True\} \\ & 0 \leq 0 \leq M \wedge 0 \leq 0 \leq N) \\ \equiv & \quad \{a \geq b \equiv a > b \vee a = b, Transitividad\} \\ & (M > 0 \vee M = 0) \wedge (N > 0 \vee N = 0) \\ \Leftarrow & \quad \{p \Rightarrow p \vee q\} \\ & (M > 0) \wedge (N > 0) \\ \Leftarrow & \quad \{p \wedge q \Rightarrow p\} \\ & (M > 0) \wedge (N > 0) \wedge (S > 0) \end{aligned}$$

*Prueba 2*

$$(((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge 0 \leq i \leq M) \wedge j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N) \wedge i \neq M \wedge j \neq N \wedge i = j \Rightarrow (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge 0 \leq i \leq M) \wedge j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N)[\text{diagonalenTablero}[i] := \text{supertablero}[i][i][i]$$

$$\begin{aligned} & (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge 0 \leq i \leq M) \wedge j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N)[\text{diagonalenTablero}[i] := \text{supertablero}[i][i][i] \\ \equiv & \quad \{SustitucionTextual\} \\ & (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge 0 \leq i \leq M) \wedge j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N) \\ \Leftarrow & \quad \{p \wedge q \Rightarrow p\} \\ & (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge 0 \leq i \leq M) \wedge j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N) \wedge i \neq M \wedge j \neq N \wedge i = j \end{aligned}$$

*Prueba 3*

$$\begin{aligned}
& (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge \\
& 0 \leq i \leq M) \wedge j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \\
& \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N) \wedge i \neq M \wedge j \neq N \wedge i! = j \Rightarrow \\
& (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge \\
& 0 \leq i \leq M) \wedge j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \\
& \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N)
\end{aligned}$$

$$\begin{aligned}
& (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge \\
& 0 \leq i \leq M) \wedge j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \\
& \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N) \\
& \Leftarrow \{p \wedge q \Rightarrow p\} \\
& (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge \\
& 0 \leq i \leq M) \wedge j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \\
& \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N) \wedge i \neq M \wedge j \neq N \wedge i! = j
\end{aligned}$$

*Prueba 4*

$$\begin{aligned}
& (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge \\
& 0 \leq i \leq M) \wedge j \geq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \\
& \text{supertablero}[p][p][p]) \wedge 0 \leq i \leq N) \wedge i = M \wedge j = N \Rightarrow (\forall i : 0 \leq i < M : \\
& (\forall j : 0 \leq j < N \wedge i = j : \text{diagonalenTablero}[i] = \text{supertablero}[i][i][i])) \wedge \\
& (\forall i : 0 \leq i < M : (\forall j : 0 \leq j < N \wedge i! = j : \text{True}))
\end{aligned}$$

*Suponiendo el antecedente*

$$\begin{aligned}
& H0 : (\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) == \\
& \text{True}
\end{aligned}$$

$$H1 : 0 \leq i \leq M == \text{True}$$

$$H2 : j \geq i == \text{True}$$

$$\begin{aligned}
& H3 : (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) == \\
& \text{True}
\end{aligned}$$

$$H4 : 0 \leq i \leq N == \text{True}$$

$$H5 : i = M == \text{True}$$

$$H6 : j = N == \text{True}$$

*True*

$$\begin{aligned}
& \equiv \{H0 \wedge H3\} \\
& (\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge (\forall p : \\
& 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \\
& \equiv \{H5 \wedge H6\} \\
& (\forall q : 0 \leq q < M : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge (\forall p : \\
& 0 \leq p < N : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \\
& \equiv \{\text{Renaming}\} \\
& (\forall i : 0 \leq i < M : \text{diagonalenTablero}[i] = \text{supertablero}[i][i][i]) \wedge (\forall j : \\
& 0 \leq j < N : \text{diagonalenTablero}[j] = \text{supertablero}[j][j][j])
\end{aligned}$$

*Prueba 5*

$$\begin{aligned}
& (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge \\
& 0 \leq i \leq M) \wedge i \neq M \Rightarrow M - i > 0 \\
& (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge 0 \leq \\
& i \leq M) \wedge i \neq M \\
& \Rightarrow \{p \wedge q \Rightarrow p\} \\
& (0 \leq i \leq M) \\
& \equiv \{a \leq b \leq c \equiv a \leq b \wedge b \leq c\} \\
& (0 \leq i \wedge i \leq M) \\
& \Rightarrow \{p \wedge q \Rightarrow p\} \\
& (i \leq M) \\
& \equiv \{a \leq b \equiv 0 \leq b - a\} \\
& (0 \leq M - i)
\end{aligned}$$

*Prueba 6*

$$\begin{aligned}
& j \leq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge \\
& 0 \leq i \leq N \wedge i \neq N \Rightarrow N - i > 0 \\
& j > i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge \\
& 0 \leq i \leq N \wedge i \neq N \\
& \Rightarrow \{p \wedge q \Rightarrow p\} \\
& (0 \leq i \leq N) \\
& \equiv \{a \leq b \leq c \equiv a \leq b \wedge b \leq c\} \\
& (0 \leq i \wedge i \leq N) \\
& \Rightarrow \{p \wedge q \Rightarrow p\} \\
& (i \leq N) \\
& \equiv \{a \leq b \equiv 0 \leq b - a\} \\
& (0 \leq N - i)
\end{aligned}$$

*Prueba 7*

$$\begin{aligned}
& (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge \\
& 0 \leq i \leq M) \wedge i \neq M \wedge M - i = c \Rightarrow (M - i \leq c)[\text{diagonalenTablero}[i] = \\
& \text{supertablero}[i][i][i]] \\
& \equiv \{SustitucionTextual\} \\
& (((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge 0 \leq \\
& i \leq M) \wedge i \neq M \wedge M - i = c \Rightarrow (M - i \leq c) \\
& \quad \text{Suponiendo el antecedente} \\
& M - i \leq c \\
& \equiv \{M - i = c\} \\
& c \leq c \\
& \equiv \{a \leq a \equiv \text{True}\} \\
& \text{True}
\end{aligned}$$

*Prueba 8*

$$((\forall q : 0 \leq q < i : \text{diagonalenTablero}[q] = \text{supertablero}[q][q][q]) \wedge$$

$$0 \leq i \leq M) \wedge i \neq M \wedge M - i = c \Rightarrow (M - i \leq c)$$

*Suponiendo el antecedente*

$$\begin{aligned} & M - i \leq c \\ \equiv & \{M - i = c\} \\ & c \leq c \\ \equiv & \{a \leq a \equiv \text{True}\} \\ & \text{True} \end{aligned}$$

*Prueba 9*

$$\begin{aligned} & j \leq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge \\ & 0 \leq i \leq N \wedge i \neq N \wedge N - i = c \Rightarrow (N - i \leq c) [\text{diagonalenTablero}[i] = \\ & \text{supertablero}[i][i][i]] \\ \equiv & \{SustitucionTextual\} \\ & j > i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge \\ & 0 \leq i \leq N \wedge i \neq N \wedge N - i = c \Rightarrow (N - i \leq c) \\ & \text{Suponiendo el antecedente} \\ & N - i \leq c \\ \equiv & \{N - i = c\} \\ & c \leq c \\ \equiv & \{a \leq a \equiv \text{True}\} \\ & \text{True} \end{aligned}$$

*Prueba 10*

$$\begin{aligned} & j \leq i \wedge (\forall p : 0 \leq p < i : \text{diagonalenTablero}[p] = \text{supertablero}[p][p][p]) \wedge \\ & 0 \leq i \leq N \wedge i \neq N \wedge N - i = c \Rightarrow (N - i \leq c) \\ & \text{Suponiendo el antecedente} \\ & N - i \leq c \\ \equiv & \{N - i = c\} \\ & c \leq c \\ \equiv & \{a \leq a \equiv \text{True}\} \\ & \text{True} \end{aligned}$$

### Procedimiento que dice si la jugada es valida

```

proc esvalida(in supertablero: array [0..N)x[0..M)x[0..S) of int; in tablero:
int; in fila: int; in casilla: int; out res: bool; in N: int; in M: int; in S:int)
{Precondicion N > 0 ∧ M > 0 ∧ S > 0 ∧ supertablero[tablero][fila][casilla] >=
0}
{Postcondicion res = supertablero[tablero][fila][casilla] == 0 ∨ res =
supertablero[tablero][fila][casilla] > 0}
[var res : bool
res := False
if supertablero[tablero][fila][casilla] == 0 → res := True
[ supertablero[tablero][fila][casilla] > 0 → Skip
fi

```



$res := res$   
 $||$

*Prueba 1*

$N \dot{!} 0 \wedge M > 0 \wedge S > 0 \wedge supertablero[tablero][fila][casilla] >= 0 \Rightarrow supertablero[tablero][fila][casilla]$   
 $0 \vee supertablero[tablero][fila][casilla] > 0$   
 $N > 0 \wedge M > 0 \wedge S > 0 \wedge supertablero[tablero][fila][casilla] >= 0$   
 $\Rightarrow \{p \wedge q \Rightarrow p\}$   
 $supertablero[tablero][fila][casilla] >= 0$   
 $\equiv \{a >= 0 \equiv a > 0 \vee a = 0\}$   
 $supertablero[tablero][fila][casilla] > 0 \vee supertablero[tablero][fila][casilla] =$   
 $0$

*Prueba 2*

$N \dot{!} 0 \wedge M > 0 \wedge S > 0 \wedge supertablero[tablero][fila][casilla] = 0 \Rightarrow (res =$   
 $supertablero[tablero][fila][casilla] == 0 \vee res = supertablero[tablero][fila][casilla] >$   
 $0)[res := True]$

$(res = supertablero[tablero][fila][casilla] == 0 \vee res = supertablero[tablero][fila][casilla] >$   
 $0)[res := True]$   
 $\equiv \{SustitucionTextual\}$   
 $(True = supertablero[tablero][fila][casilla] == 0 \vee True = supertablero[tablero][fila][casilla] >$   
 $0)$   
 $\equiv \{true \equiv q \equiv q\}$   
 $(supertablero[tablero][fila][casilla] == 0 \vee supertablero[tablero][fila][casilla] >$   
 $0)$   
 $\Leftarrow \{p \Rightarrow p \vee q\}$   
 $(supertablero[tablero][fila][casilla] == 0)$   
 $\Leftarrow \{p \wedge q \Rightarrow p\}$   
 $N > 0 \wedge M > 0 \wedge S > 0 \wedge supertablero[tablero][fila][casilla] = 0$

*Prueba 3*

$N \dot{!} 0 \wedge M > 0 \wedge S > 0 \wedge supertablero[tablero][fila][casilla] > 0 \Rightarrow (res =$   
 $supertablero[tablero][fila][casilla] == 0 \vee res = supertablero[tablero][fila][casilla] >$   
 $0)[res := False]$

$(res = supertablero[tablero][fila][casilla] == 0 \vee res = supertablero[tablero][fila][casilla] >$   
 $0)[res := False]$   
 $\equiv \{SustitucionTextual\}$   
 $(False = supertablero[tablero][fila][casilla] == 0 \vee False = supertablero[tablero][fila][casilla] >$   
 $0)$   
 $\equiv \{\neg p \equiv p \equiv false\}$   
 $(\neg(supertablero[tablero][fila][casilla] == 0) \vee \neg(supertablero[tablero][fila][casilla] >$

$$\begin{aligned}
& 0)) \\
& \Leftarrow \{p \Rightarrow p \vee q\} \\
& \neg(\text{supertablero}[\text{tablero}][\text{fila}][\text{casilla}] == 0) \\
& \equiv \{\neg(a = b) \equiv a \neq b\} \\
& (\text{supertablero}[\text{tablero}][\text{fila}][\text{casilla}] \neq 0) \\
& \equiv \{a \neq b \equiv a < b \vee a > b\} \\
& (\text{supertablero}[\text{tablero}][\text{fila}][\text{casilla}] < 0 \vee \text{supertablero}[\text{tablero}][\text{fila}][\text{casilla}] > 0) \\
& 0) \\
& \Leftarrow \{p \Rightarrow p \vee q\} \\
& (\text{supertablero}[\text{tablero}][\text{fila}][\text{casilla}] > 0) \\
& \Leftarrow \{p \wedge q \Rightarrow p\} \\
& N > 0 \wedge M > 0 \wedge S > 0 \wedge \text{supertablero}[\text{tablero}][\text{fila}][\text{casilla}] > 0
\end{aligned}$$

### Procedimiento que cuenta las líneas verticales

```

proc lineVerticalHecha(in supertablero: array [0..N)x[0..M)x[0..S) of int;
in tablero: int; in fila: int; in casilla: int; out columnasEnTablero:array
[0..M) of int; out columnasEnz: array [0..S) of int; in N: int; in M: int; in
S:int)
  {Precondicion N;0 & M > 0 & S > 0}
  {Postcondicion(∀i : 0 <= i < M : columnasEnTablero[i] = supertablero[tablero][i][casilla]) ∧
(∀i : 0 <= i < S : columnasEnz[i] = supertablero[i][fila][casilla]))}
  [var fil, z : int
  fil, z := 0, 0
  {Invariante(∀j : 0 <= j < i : columnasEnTablero[j] = supertablero[tablero][j][casilla]) ∧
0 <= i <= M}
  do fil ≠ M → columnasEnTablero[fil] := supertablero[tablero][fil][casilla]
    fil := fil + 1
  od
  {Invariante(∀j : 0 <= j < i : columnasEnz[j] = supertablero[j][fila][casilla]) ∧
0 <= i <= S}
  do z ≠ S → columnasEnz[z] := supertablero[z][fila][casilla]
    z := z + 1
  od
  columnasEnTablero := columnasEnTablero
  columnasEnz := columnasEnz
]]

```

#### Prueba 1

$$N;0 \wedge M > 0 \wedge S > 0 \Rightarrow ((\forall j : 0 \leq j < \text{fil} : \text{columnasEnTablero}[j] = \text{supertablero}[\text{tablero}][j][\text{casilla}]) \wedge 0 \leq \text{fil} \leq M)[\text{fil} := 0]$$

$$\begin{aligned}
& ((\forall j : 0 \leq j < \text{fil} : \text{columnasEnTablero}[j] = \text{supertablero}[\text{tablero}][j][\text{casilla}]) \wedge \\
& 0 \leq \text{fil} \leq M)[\text{fil} := 0] \\
\equiv & \quad \{SustitucionTextual\} \\
& N > 0 \wedge M > 0 \wedge S > 0 \Rightarrow ((\forall j : 0 \leq j < 0 : \text{columnasEnTablero}[j] = \\
& \text{supertablero}[\text{tablero}][j][\text{casilla}]) \wedge 0 \leq 0 \leq M) \\
\equiv & \quad \{RangovaciocyTransitividad\} \\
& (\text{True} \wedge 0 \leq M) \\
\equiv & \quad \{a \geq b \equiv a > b \vee a = b, p \wedge \text{True} \equiv p\} \\
& (M > 0 \vee M = 0) \\
\Leftarrow & \quad \{p \Rightarrow p \vee q\} \\
& (M > 0) \\
\Leftarrow & \quad \{p \wedge q \Rightarrow p\} \\
& (M > 0 \wedge N > 0 \wedge S > 0)
\end{aligned}$$

*Prueba 2*

$$\begin{aligned}
& N \neq 0 \wedge M > 0 \wedge S > 0 \Rightarrow ((\forall j : 0 \leq j < z : \text{columnasEnz}[j] = \\
& \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge 0 \leq z \leq S)[z := 0]
\end{aligned}$$

$$\begin{aligned}
& ((\forall j : 0 \leq j < z : \text{columnasEnz}[j] = \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge \\
& 0 \leq z \leq S)[z := 0] \\
\equiv & \quad \{SustitucionTextual\} \\
& ((\forall j : 0 \leq j < 0 : \text{columnasEnz}[j] = \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge \\
& 0 \leq 0 \leq S) \\
\equiv & \quad \{RangovaciocyTransitividad\} \\
& (\text{True} \wedge 0 \leq S) \\
\equiv & \quad \{a \geq b \equiv a > b \vee a = b, p \wedge \text{True} \equiv p\} \\
& (S > 0 \vee S = 0) \\
\Leftarrow & \quad \{p \Rightarrow p \vee q\} \\
& (S > 0) \\
\Leftarrow & \quad \{p \wedge q \Rightarrow p\} \\
& (M > 0 \wedge N > 0 \wedge S > 0)
\end{aligned}$$

*Prueba 3*

$$\begin{aligned}
& (\forall j : 0 \leq j < i : \text{columnasEnTablero}[j] = \text{supertablero}[\text{tablero}][j][\text{casilla}]) \wedge \\
& 0 \leq i \leq M \wedge \text{fil} \neq M \Rightarrow ((\forall j : 0 \leq j < i : \text{columnasEnTablero}[j] = \\
& \text{supertablero}[\text{tablero}][j][\text{casilla}]) \wedge 0 \leq i \leq M)[\text{columnasenTablero}[\text{fil}] := \\
& \text{supertablero}[\text{tablero}][\text{fila}][\text{casilla}]] \\
\equiv & \quad \{SustitucionTextual\} \\
& ((\forall j : 0 \leq j < i : \text{columnasEnTablero}[j] = \text{supertablero}[\text{tablero}][j][\text{casilla}]) \wedge \\
& 0 \leq i \leq M)) \\
\Leftarrow & \quad \{p \wedge q \Rightarrow p\} \\
& ((\forall j : 0 \leq j < i : \text{columnasEnTablero}[j] = \text{supertablero}[\text{tablero}][j][\text{casilla}]) \wedge \\
& 0 \leq i \leq M) \wedge \text{fil} \neq M
\end{aligned}$$

*Prueba 4*

$$\begin{aligned}
& (\forall j : 0 \leq j < i : \text{columnasEnz}[j] = \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge \\
& 0 \leq i \leq S \wedge z \neq S \Rightarrow ((\forall j : 0 \leq j < i : \text{columnasEnz}[j] = \\
& \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge 0 \leq i \leq S) [\text{columnasEnz}[z] := \text{supertablero}[z][\text{fila}][\text{casilla}]) \\
& \equiv \{ \text{SustitucionTextual} \} \\
& (\forall j : 0 \leq j < i : \text{columnasEnz}[j] = \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge \\
& 0 \leq i \leq S \wedge z \neq S \Rightarrow ((\forall j : 0 \leq j < i : \text{columnasEnz}[j] = \\
& \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge 0 \leq i \leq S) \\
& ((\forall j : 0 \leq j < i : \text{columnasEnz}[j] = \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge \\
& 0 \leq i \leq S) \\
& \Leftarrow \{ p \wedge q \Rightarrow p \} \\
& ((\forall j : 0 \leq j < i : \text{columnasEnz}[j] = \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge \\
& 0 \leq i \leq S) \wedge z \neq S
\end{aligned}$$

*Prueba 5*

$$\begin{aligned}
& (\forall j : 0 \leq j < \text{fil} : \text{columnasEnTablero}[j] = \text{supertablero}[\text{tablero}][j][\text{casilla}]) \wedge \\
& 0 \leq \text{fil} \leq M \wedge \text{fil} \equiv M \Rightarrow (\forall \text{fil} : 0 \leq \text{fil} < M : \text{columnasEnTablero}[\text{fil}] = \\
& \text{supertablero}[\text{tablero}][\text{fil}][\text{casilla}]) \\
& (\forall j : 0 \leq j < \text{fil} : \text{columnasEnTablero}[j] = \text{supertablero}[\text{tablero}][j][\text{casilla}]) \wedge \\
& 0 \leq \text{fil} \leq M \wedge \text{fil} \equiv M \\
& \Rightarrow \{ \text{Leibniz} \} \\
& (\forall j : 0 \leq j < M : \text{columnasEnTablero}[j] = \text{supertablero}[\text{tablero}][j][\text{casilla}]) \wedge \\
& 0 \leq M \leq M \\
& \Rightarrow \{ p \wedge q \Rightarrow p \} \\
& (\forall j : 0 \leq j < M : \text{columnasEnTablero}[j] = \text{supertablero}[\text{tablero}][j][\text{casilla}])
\end{aligned}$$

*Prueba 6*

$$\begin{aligned}
& (\forall j : 0 \leq j < z : \text{columnasEnz}[j] = \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge \\
& 0 \leq z \leq S \wedge z \equiv S \Rightarrow (\forall i : 0 \leq i < S : \text{columnasEnz}[i] = \\
& \text{supertablero}[i][\text{fila}][\text{casilla}]) \\
& (\forall j : 0 \leq j < z : \text{columnasEnz}[j] = \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge \\
& 0 \leq z \leq S \wedge z \equiv S \\
& \Rightarrow \{ \text{Leibniz} \} \\
& (\forall j : 0 \leq j < S : \text{columnasEnz}[j] = \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge \\
& 0 \leq S \leq S \\
& \Rightarrow \{ p \wedge q \Rightarrow p \} \\
& (\forall j : 0 \leq j < S : \text{columnasEnz}[j] = \text{supertablero}[j][\text{fila}][\text{casilla}])
\end{aligned}$$

*Prueba 7*

$$\begin{aligned}
& (\forall j : 0 \leq j < \text{fil} : \text{columnasEnTablero}[j] = \text{supertablero}[\text{tablero}][j][\text{casilla}]) \wedge \\
& 0 \leq \text{fil} \leq M \wedge \text{fil} \neq M \Rightarrow M - \text{fil} > 0 \\
& \text{Suponiendo el antecedente} \\
& (\forall j : 0 \leq j < \text{fil} : \text{columnasEnTablero}[j] = \text{supertablero}[\text{tablero}][j][\text{casilla}]) ==
\end{aligned}$$

$$\begin{aligned}
& True \\
& 0 \leq fil \leq M == True \\
& fil \neq M == True \\
& True \\
\equiv & \{0 \leq fil \leq M == True\} \\
& 0 \leq fil \leq M \\
\equiv & \{a \leq b \leq c == a \leq b \wedge b \leq c\} \\
& 0 \leq fil \wedge fil \leq M \\
\Rightarrow & \{p \wedge q \Rightarrow p\} \\
& fil \leq M \\
\equiv & \{a \leq b == 0 \leq b - a\} \\
\Rightarrow & \{p \wedge q \Rightarrow p\} \\
& 0 \leq M - fil
\end{aligned}$$

$$\begin{aligned}
& Prueba 8 \\
& (\forall j : 0 \leq j < z : columnasEnz[j] = supertablero[j][fila][casilla]) \wedge \\
0 \leq z \leq S \wedge z \neq S \Rightarrow S - z >= 0 \\
& Suponiendoelantecedente \\
& (\forall j : 0 \leq j < z : columnasEnz[j] = supertablero[j][fila][casilla]) == \\
True \\
& 0 \leq z \leq S == True \\
& z \neq S == True \\
& True \\
\equiv & \{0 \leq z \leq S == True\} \\
& 0 \leq z \leq S \\
\equiv & \{a \leq b \leq c == a \leq b \wedge b \leq c\} \\
& 0 \leq z \wedge z \leq S \\
\Rightarrow & \{p \wedge q \Rightarrow p\} \\
& z \leq S \\
\equiv & \{a \leq b == 0 \leq b - a\} \\
\Rightarrow & \{p \wedge q \Rightarrow p\} \\
& 0 \leq S - z
\end{aligned}$$

$$\begin{aligned}
& Prueba 9 \\
& (\forall j : 0 \leq j < fil : columnasEnTablero[j] = supertablero[tablero][j][casilla]) \wedge \\
0 \leq fil \leq M \wedge fil \neq M \wedge M - fil = c \Rightarrow (M - fil \leq c)[columnasEnz[j] := \\
supertablero[j][fila][casilla]] \\
& Suponiendoelantecedente \\
& (M - fil \leq c)[columnasEnz[j] := supertablero[j][fila][casilla]] \\
\equiv & \{SustitucionTextual\} \\
& (M - fil \leq c) \\
\equiv & \{M - fil = c\} \\
& c \leq c \\
\equiv & \{a \leq a \equiv True\}
\end{aligned}$$

*True*

*Prueba 10*

$(\forall j : 0 \leq j < z : \text{columnasEnz}[j] = \text{supertablero}[j][\text{fila}][\text{casilla}]) \wedge$   
 $0 \leq z \leq S \wedge z \neq S \wedge S - z = c \Rightarrow (S - z \leq c)[\text{columnasEnz}[j] :=$   
 $\text{supertablero}[j][\text{fila}][\text{casilla}]]$

*Suponiendo el antecedente*

$(S - z \leq c)[\text{columnasEnz}[j] := \text{supertablero}[j][\text{fila}][\text{casilla}]]$   
 $\equiv \{ \text{SustitucionTextual} \}$   
 $(S - z \leq c)$   
 $\equiv \{ S - z = c \}$   
 $c \leq c)$   
 $\equiv \{ a \leq a \equiv \text{True} \}$   
*True*

### Procedimiento que cuenta las líneas horizontales

```
proc lineaHorizontalHecha(in supertablero: array [0..N)x[0..M)x[0..S] of
int; in tablero: int; in fila; in N: int; in M: int; in S: int; out h: bool)
{Precondicion N > 0 ∧ M > 0 ∧ S > 0}
{Postcondicion}
[var filaActual : int
h : bool
filaActual := supertablero[tablero][fila]
h := verificarlista(filaActual)
h := h

]
```

Como el procedimiento para hallar si una linea horizontal se ha hecho consiste en la llamada de la funcion verificar lista, demostraremos a continuación el procedimiento verificarlista

### Procedimiento que verifica si todos los elementos de una lista son iguales

```
proc verificarlista(in lista: array [0..N] of int; in resultado: bool; out res:
bool) → bool
{Precondicion N > 0 ∧ lista[0]! = 0}
{Postcondicion res = (∀ i : 1 ≤ i < N : lista[0] = lista[i] and lista[0]! =
0) or res = not(∀ i : 1 ≤ i < N : lista[0] = lista[i] and lista[0]! = 0)}
[var res : bool
```

```

res, i := True, 1
Invariantres = ( $\forall j : 1 \leq j < i : lista[0] = lista[j]$ )  $\wedge 1 \leq i \leq N$ 
do i  $\neq N \Rightarrow res := lista[0] = lista[i]$ 
    i := i + 1
od
res := res
]]

```

*Prueba 1*

$N \neq 0 \wedge lista[0]! = 0 \Rightarrow (res = (\forall j : 1 \leq j < i : lista[0] = lista[j]) \wedge 1 \leq i \leq N)[res, i := False, 1]$

$(res = (\forall j : 1 \leq j < i : lista[0] = lista[j]) \wedge 1 \leq i \leq N)[res, i := True, 1]$   
 $\equiv \{SustitucionTextual\}$   
 $(True = (\forall j : 1 \leq j < 1 : lista[0] = lista[j]) \wedge 1 \leq 1 \leq N)$   
 $\equiv \{RangoVacio\}$   
 $(True = True \wedge 1 \leq 1 \leq N)$   
 $\equiv \{Transitividad, p \wedge True \equiv p\}$   
 $(1 \leq N)$   
 $\equiv \{a \geq b \equiv a > b - 1\}$   
 $(N > 0)$   
 $\Leftarrow \{p \wedge q \Rightarrow p\}$   
 $(N > 0 \wedge lista[0]! = 0)$

*Prueba 2*

$res = (\forall j : 1 \leq j < i : lista[0] = lista[j]) \wedge 1 \leq i \leq N \wedge i < N \Rightarrow$   
 $(res = (\forall j : 1 \leq j < i : lista[0] = lista[j]) \wedge 1 \leq i \leq N)[res := lista[0] = lista[i]]$

$(res = (\forall j : 1 \leq j < i : lista[0] = lista[j]) \wedge 1 \leq i \leq N)[res := lista[0] = lista[i]]$   
 $\equiv \{SustitucionTextual\}$   
 $(lista[0] = lista[i] = (\forall j : 1 \leq j < i : lista[0] = lista[j]) \wedge 1 \leq i \leq N)$   
 $\equiv \{Splitoffterm\}$   
 $(lista[0] = lista[i] = (\forall j : 1 \leq j < i : lista[0] = lista[j]) \wedge lista[0] = lista[i - 1] \wedge 1 \leq i \leq N)$   
 $\Leftarrow \{p \wedge q \Rightarrow p, defres\}$   
 $res = (\forall j : 1 \leq j < i : lista[0] = lista[j]) \wedge 1 \leq i \leq N \wedge i < N$

*Prueba 3*

$res = (\forall j : 1 \leq j < i : lista[0] = lista[j]) \wedge 1 \leq i \leq N \wedge i = N \Rightarrow$

$$\begin{aligned}
res &= (\forall i : 1 \leq i < N : lista[0] = lista[i]) \\
res &= (\forall j : 1 \leq j < i : lista[0] = lista[j]) \wedge 1 \leq i \leq N \wedge i = N \\
\Rightarrow & \quad \{Leibniz\} \\
res &= (\forall j : 1 \leq j < N : lista[0] = lista[j]) \wedge 1 \leq N \leq N \\
\Rightarrow & \quad \{p \wedge q \Rightarrow p\} \\
res &= (\forall j : 1 \leq j < N : lista[0] = lista[j])
\end{aligned}$$

*Prueba 4*

$$\begin{aligned}
res &= (\forall j : 1 \leq j < i : lista[0] = lista[j]) \wedge 1 \leq i \leq N \wedge i \neq N \Rightarrow \\
N - i &\geq 0
\end{aligned}$$

*Suponiendo el antecedente*

$$\begin{aligned}
res &= (\forall j : 1 \leq j < i : lista[0] = lista[j]) == True \\
1 \leq i \leq N &== True \\
i \neq N &== True \\
True & \\
\equiv & \quad \{1 \leq i \leq N == True\} \\
1 \leq i \leq N & \\
\equiv & \quad \{a \leq b \leq c \equiv a \leq b \wedge b \leq c\} \\
1 \leq i \wedge i \leq N & \\
\Rightarrow & \quad \{p \wedge q \Rightarrow p\} \\
i \leq N & \\
\equiv & \quad \{a \leq b \equiv 0 \leq b - a\} \\
0 \leq N - i &
\end{aligned}$$

*Prueba 5*

$$\begin{aligned}
res &= (\forall j : 1 \leq j < i : lista[0] = lista[j]) \wedge 1 \leq i \leq N \wedge i \neq \\
N \wedge N - i = c &\Rightarrow (N - i \leq c)[res := lista[0] = lista[i]]
\end{aligned}$$

*Suponiendo el antecedente*

$$\begin{aligned}
res &= (\forall j : 1 \leq j < i : lista[0] = lista[j]) == True \\
1 \leq i \leq N &== True \\
i \neq N &== True \\
N - i = c &== True \\
(N - i \leq c)[res := lista[0] = lista[i]] & \\
\equiv & \quad \{SustitucionTextual\} \\
N - i \leq c & \\
\equiv & \quad \{N - i = c\} \\
c \leq c & \\
\equiv & \quad \{a \leq a \equiv True\} \\
True &
\end{aligned}$$