

PROGRAMACIÓN LÓGICA

CURSO 2025

GRUPO 01

Informe - Laboratorio 2

Autores:

GONZALO APKARIAN

ALONSO

5.494.604-2

DIEGO PISA SÁNCHEZ

5.510.635-2

LEONARDO PESCE LÓPEZ

5.471.535-2

MAURICIO MORALES

GONZALEZ

5.278.642-4

Profesores:

Luis CHIRUZZO

Aiala ROSÁ

Juan Pablo CONDE



FACULTAD DE
INGENIERÍA
UDELAR

9 de junio de 2025

1. Módulos

Para este laboratorio no se usaron módulos extra, se implementaron todos los predicados en el mismo archivo principal. Esta decisión se tomó debido a que se tornó una complicación decidir como modularizar de manera eficiente el programa y no se contaba con mucho tiempo, por lo que el equipo se fue por el camino más rápido aunque sin seguir patrones de un buen diseño en su totalidad.

Sin embargo, para facilitar la lectura y corrección del programa, cada vez que se iniciaba el desarrollo de un predicado principal se dejaban líneas con caracteres comentados de forma que se pueda diferenciar fácilmente en el IDE las distintas funciones.

2. Predicados principales

En esta sección se mencionarán y explicarán, además del funcionamiento de los predicados necesarios para que funcione *ceritolog* (es decir, los que vienen explicados en la letra), aquellos predicados auxiliares que el equipo considera esenciales para que el principal pueda realizar su función.

2.1. Tablero

En este predicado simplemente se pide devolver un tablero de tamaño N vacío, donde N es la cantidad de vértices que tiene el tablero por lado, generando una matriz cuadrada donde en realidad quedan $(N - 1) \cdot (N - 1)$ celdas.

2.1.1. Diseño del tablero

El tablero se crea usando las celdas, la cual es una tupla de 3 elementos (arista superior, arista izquierda, jugador) o (F, C, J). Aquí se tuvo que hacer la primera decisión del diseño del tablero, ya que, si bien el tablero es $N \cdot N$ en vértices, se tomó la decisión de crearlo con $N \cdot N$ celdas para poder tener la información de las aristas de los bordes inferior y derecho. Para esas celdas extra, en la última fila solo se tiene acceso a modificación de la arista superior (que sería la arista inferior del tablero) mientras que en la última columna solo se puede modificar la arista izquierda de esas celdas (que sería la arista derecha del tablero).

2.2. Fin del juego

En este predicado se retorna el final del juego, cuál jugador ganó y con cuantos puntos terminó cada uno. En caso de que no sea el final del juego, el predicado falla.

2.2.1. Diseño de "Fin del juego"

Para el fin del juego se tomó que todas las líneas deben estar marcadas, es decir, una vez que se inicia una partida se debe terminar por completo. Para ello se utiliza un predicado auxiliar que justamente se encarga de verificar que estén todas las líneas marcadas.

2.2.2. Predicado auxiliar 1 - `todas_lineas_marcadas`

El predicado auxiliar *todas_lineas_marcadas* se encarga de verificar que todas las celdas internas (las que están entre 1 y $(N - 1)$) estén en marcadas como $c(1, 1, -)$, mientras que las de la última fila $c(1, 0, 0)$ y las de la última columna $c(0, 1, 0)$. Es importante que las celdas de los bordes derecho e inferior no tengan un jugador que las haya conquistado, de otra manera no estaría funcionando bien el juego.

2.2.3. Predicado auxiliar 2 - `contar_puntos`

El otro predicado auxiliar que usa esta función es el de *contar_puntos* que pasa celda por celda sumando en el contador del jugador correspondiente si está conquistada por él. Por ejemplo, si se tiene la celda $c(-, -, 1)$, se suma un punto al jugador 1. Para que esto funcione se debe verificar antes que el juego está terminado de forma que todas las aristas estén marcadas debido a que $c(1, 0, 1)$ no es una instancia válida de una casilla, así como tampoco lo es $c(0, 1, 1)$ ni $c(0, 0, 1)$ (en todas se usa como ejemplo el jugador 1, pero si en lugar de 1 fuera 2 tampoco serían viables estas instancias).