



PROYECTO LÓGICA PARA CIENCIAS DE LA COMPUTACIÓN

NONOGRAMA

Comisión 18

Integrante 1: [Giacomodonato Giulia - 142049]

Integrante 2: [Kreczmer Tomás - 140009]

Proyecto 2: Nonograma

Implementación en Prolog

Para resolver el problema, inicialmente realizamos un primer recorrido para resolver filas y columnas triviales, luego realizamos recorridos adicionales hasta que no queden celdas vacías.

Se intenta resolver de manera trivial las filas y columnas que tienen soluciones únicas basadas en las pistas dadas. Esto se hace con el predicado `esListaConUnicaSol/2` comprobando si las pistas para una fila o columna coinciden con una configuración específica de celdas.

Si una fila o columna tiene una solución única, se completa directamente en la grilla en el predicado `resolverFilaTrivial/4`. Definimos un predicado `trasponer/3` que sirve para transformar una grilla de filas en una de columnas y poder utilizar la misma idea para completar las filas y las columnas.

Después de la resolución trivial inicial, se inicia un ciclo iterativo para resolver la grilla. Durante cada iteración, se verifica cada celda de la grilla que aún no está resuelta. Si una celda aún no tiene un valor asignado (ya sea vacía o pintada), se intenta determinar su valor mediante el uso de las pistas y las celdas vecinas ya resueltas.

Por ejemplo, si una celda tiene una celda llena a su izquierda y una celda vacía a su derecha, entonces la celda en cuestión debe ser vacía según las pistas. Esto se resuelve en los predicados `solucionUnicaFila/4` y `verificarGrillaFinal/3`, utilizando el predicado `controlarPistas/2` para analizar cada celda individualmente en función de las pistas y las celdas vecinas ya resueltas, utilizando el predicado `calcularConsecutivos/3` para obtener las posibles soluciones.

Luego en `solucionGrillaAux/6`, obteniendo todas las posibles soluciones mediante las pistas y la fila/columna dada, implementamos la intersección de soluciones en el predicado `interseccion/3`. Este predicado busca identificar qué celdas deben estar llenas o vacías en una fila o columna al considerar las soluciones posibles para esa fila o columna y cómo se intersectan con otras filas y columnas.

Esta estrategia ayuda a mejorar la resolución del nonograma al aprovechar la información disponible en la grilla.

Una vez completadas todas las iteraciones y resueltas todas las celdas posibles, realizamos una pasada final sobre la grilla para asegurar que no queden celdas indeterminadas. Esto se hace mediante el predicado `solucionFinal/5`, que busca resolver cualquier celda vacía restante utilizando las pistas restantes.

En resumen, utilizamos una combinación de estrategias de resolución inicial, iteración y resolución final para resolver el nonograma y completar la grilla de manera lógica y eficiente.

La solución planteada es eficiente para nonogramas pequeños a medianos (hasta 30x30). Sin embargo, su rendimiento puede decrecer con el tamaño debido a la complejidad combinatoria de las posibles configuraciones de celdas.

Por otro lado, desarrollamos el predicado `verificarPistasIniciales/5` con el objetivo de determinar cuáles filas y columnas están inicialmente satisfechas en el marcado inicial. Este predicado reemplaza el chequeo inicial que anteriormente realizábamos desde React, permitiendo así aprovechar las capacidades de Prolog. De este modo, eliminamos la necesidad de realizar consultas individuales para cada fila y cada columna.

Uso de React

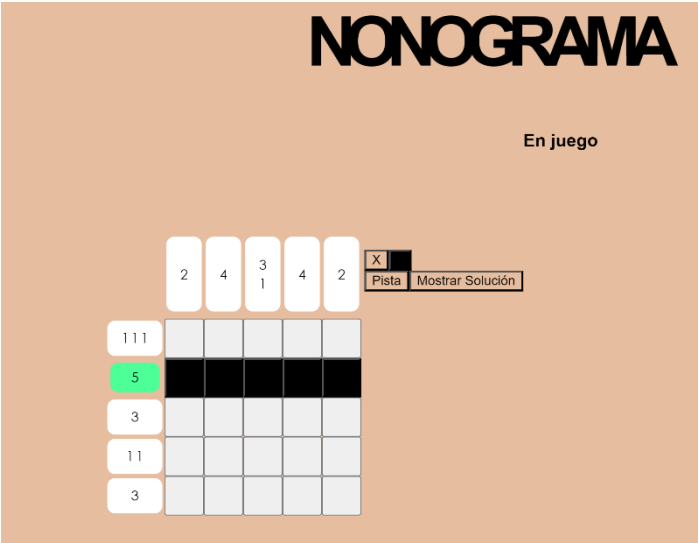
- Creamos un botón Pista que se pinta cuando el modo utilizado es el de pista, y que al clicar una celda revela el valor que esta debe tener en la solución final.
- Creamos un botón Mostrar Solución que permite ver la solución del Nonograma para luego poder seguir jugando.
- Creamos un botón Deshacer que permite borrar la última jugada.
- Creamos un botón Reiniciar que permite volver al marcado inicial de la grilla y comenzar a jugar de nuevo.

Casos de test

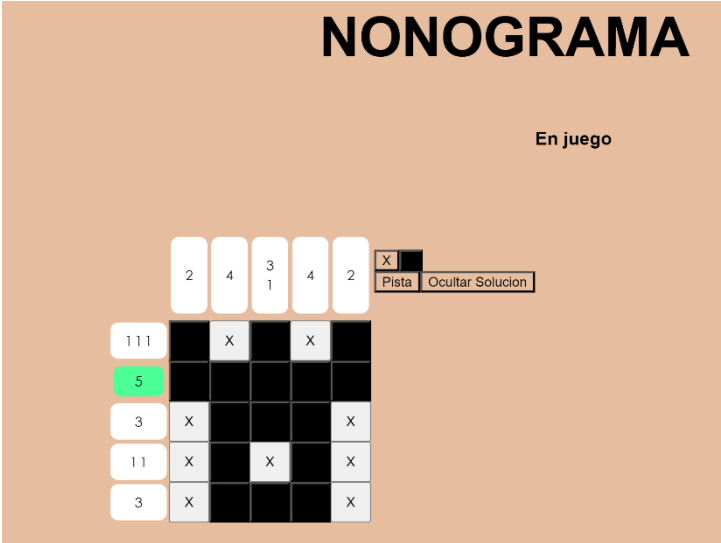
La grilla comienza con una fila completa.

```
:- module(init, [ init/3 ]).  
  
init(  
  [ [1,1,1], [5], [3], [1,1], [3]], % PistasFilas  
  [ [2], [4], [3,1], [4],[2] ], % PistasColumnas  
  [ [_ , _ , _ , _ ],  
    [ "#", "#", "#", "#", "#"],  
    [ _ , _ , _ , _ ],  
    [ _ , _ , _ , _ ],  
    [ _ , _ , _ , _ ]  
  ).
```

Antes:



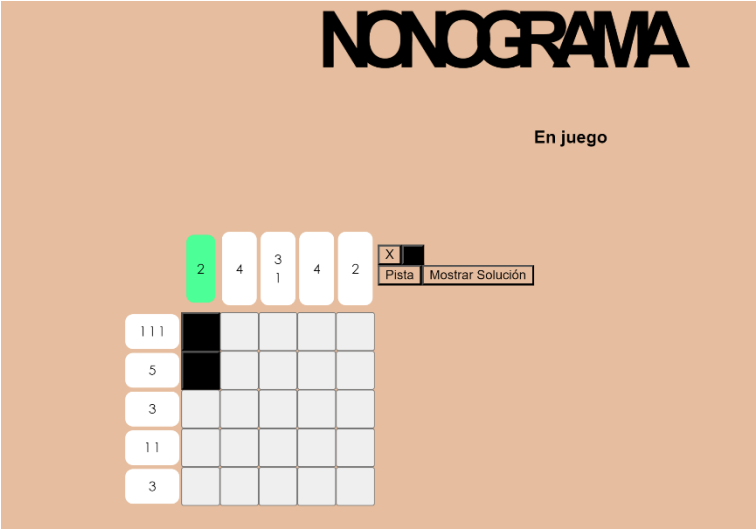
Después:



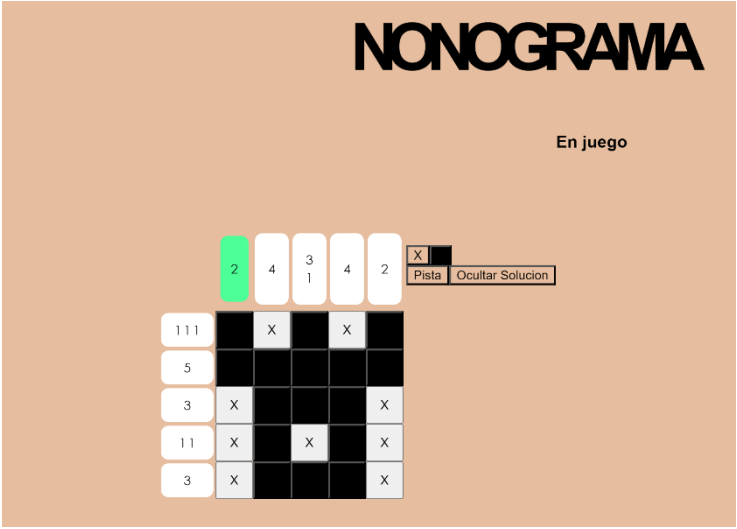
La grilla comienza con una columna completa

```
:- module(init, [ init/3 ]).  
  
init(  
  
  [ [1,1,1], [5], [3], [1,1] , [3]], % PistasFilas  
  
  [ [2], [4], [3,1], [4],[2] ], % PistasColumnas  
  
  [ "#",_,_,_,_],  
  
  [ "#",_,_,_,_],  
  
  [_,_,_,_],  
  
  [_,_,_,_],  
  
  [_,_,_,_] ]  
  
).
```

Antes:



Después:



La grilla comienza con todas las pistas completas.

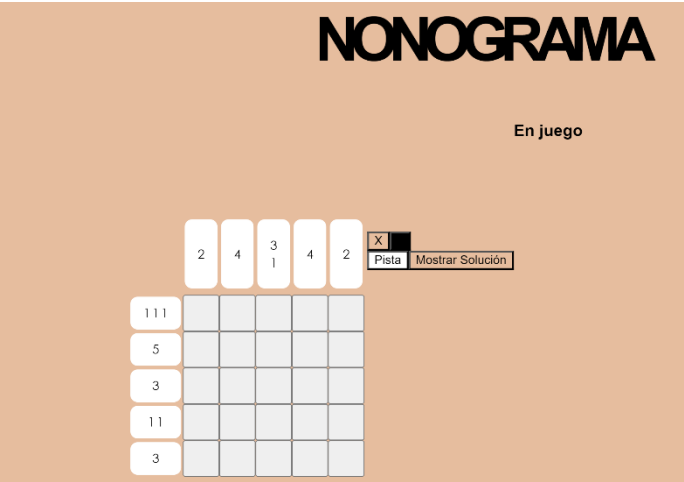
```
:- module(init, [ init/3 ]).  
  
init(  
  [ [1,1,1], [5], [3], [1,1], [3]], % PistasFilas  
  [ [2], [4], [3,1], [4],[2] ], % PistasColumnas  
  [ ["#", "_", "#", "_", "#"],  
    ["#", "#", "#", "#", "#"],  
    [ "_", "#", "#", "#", "_"],  
    [ "_", "#", "_", "#", "_"],  
    [ "_", "#", "#", "#", "_"]  
  ).
```



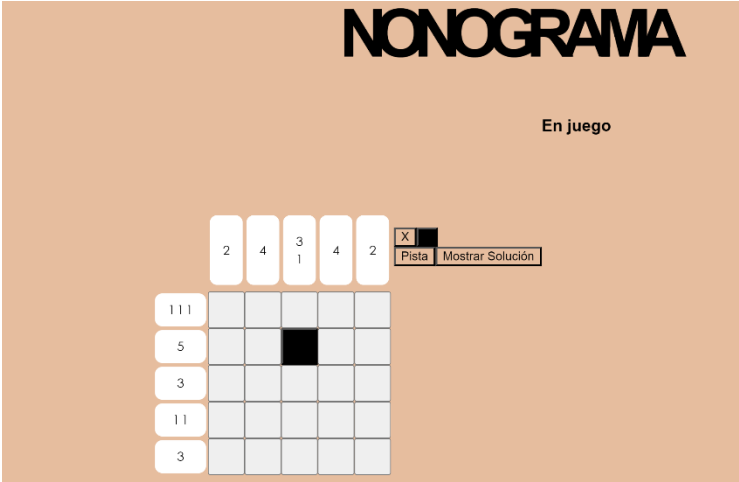
La pista es una celda pintada

```
:- module(init, [ init/3 ]).  
  
init(  
  [ [1,1,1], [5], [3], [1,1], [3]], % PistasFilas  
  [ [2], [4], [3,1], [4],[2] ], % PistasColumnas  
  [ [_,_,_,_,_],  
    [_,_,_,_,_],  
    [_,_,_,_,_],  
    [_,_,_,_,_],  
    [_,_,_,_,_] ]  
).
```

Antes:



Después:



La pista es una celda vacía
Antes:

NONOGRAMA

En juego

2

4

3
1

4

2

X

Pista

Mostrar Solución

111

5

3

11

3

Después:

NONOGRAMA

En juego

2

4

3
1

4

2

X

Pista

Mostrar Solución

111

5

3

11

3

	X				

Tamaños de grillas más grandes

15x15

```
:- module(init, [ init/3 ]).
```

```
init(
```

```
[ [], [2], [5,1],[6],[1],[1],[5,1,3],[13],[3,9],[2,9],[13],[12],[11],[9],[6] ], % PistasFilas
```

```
[ [2], [6], [3,3], [2,4] , [3,2,4], [3,9],[2,8],[2,10],[3,8],[1,8],[1,9],[8], [7], [4], []], % PistasColumnas
```

 $[[_ , _ , _ , _ , _ , _ , _ , _ , _ , _] ,$ $[_ , _ , _ , _ , _ , _ , _ , _ , _ , _ , _ , _] ,$ $[_ , _ , _ , _ , _ , _ , _ , _ , _ , _ , _ , _],$ $[_ , _ , _ , _ , _ , _ , _ , _ , _ , _ , _ , _],$ $[_ , _ , _ , _ , _ , _ , _ , _ , _ , _],$ $[_ , _ , _ , _ , _ , _ , _ , _ , _ , _] ,$ $[_ , _ , _ , _ , _ , _ , _ , _ , _ , _],$ $[_ , _ , _ , _ , _ , _ , _ , _ , _ , _] ,$ $[_ , _ , _ , _ , _ , _ , _ , _ , _ , _] ,$

[_, _, _, _],

 $[_, _, _],$

[_, _, _],

[_, _, _, _, _, _],

[_,_,_,_,_,_,_,_,_,_,_,_,_,_],

[_,_,_,_,_,_,_,_,_,_]

$$).$$


20x20

```
:- module(init, [ init/3 ]).
```

```
init(
```

```
  [ [3,3], [1,3,3,1],
```

```
  [3,10,2],[3,2,2,2],[5,4],[3,2],[1,1],[1,1],[2,1],[2,1,1,1],[3,3,3,1],[1,1,1,1,1,1,1],[1,1,3,3,1],[1,2,2,1],[1,2,2],[1,2,3,3],
```

```
  [1,2,3,3],[1,2,1,4],[1,1,2,4], [3,1,10] ], % PistasFilas
```

```
  [ [2,1,1,2], [1,1,1,1], [4,7,1], [1,4,4,2,1], [1,5,2,1], [2,2,2],[4,2],[3,4,1],[1,2,2,1],[1,3,1],[1,2,1],[1,3,1], [1,2,1],
```

```
  [1,3,1], [3,2,2,2],[5,4,3],[2,2,4],[1,5,5],[4,4,4],[4]], % PistasColumnas
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

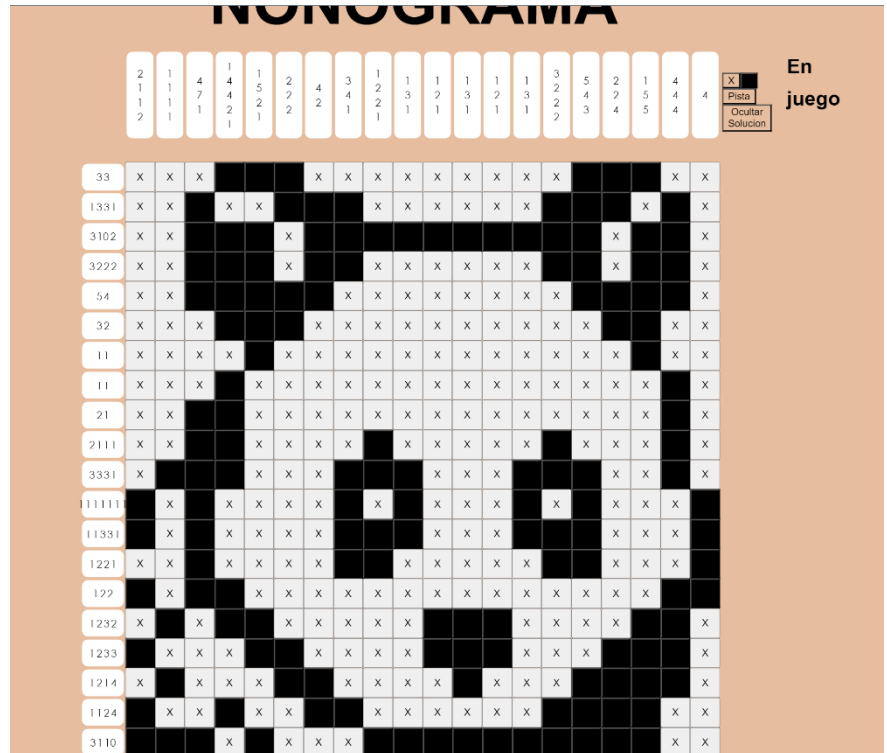
```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_],
```

```
  [_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_]]).
```



Filas y columnas de distinto tamaño

11x20

`:- module(init, [init/3]).`

`init(`

`[[5], [7], [7],[7],[1,5,1],[2,2],[1,7,1],[2,2],[11],[9],[1],[1],[3],[3,1],[3],[1],[3],[2,1],[4,1] , [7]], % PistasFilas`

`[[3], [2,3], [3,2,2,2], [5,1,2,1,3] , [5,1,2,3,4], [5,1,9,2],[5,1,2,1,1,1,1],[5,1,2,1,1,1],[3,2,2,2],[2,3],[3]], %
PistasColumnas`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

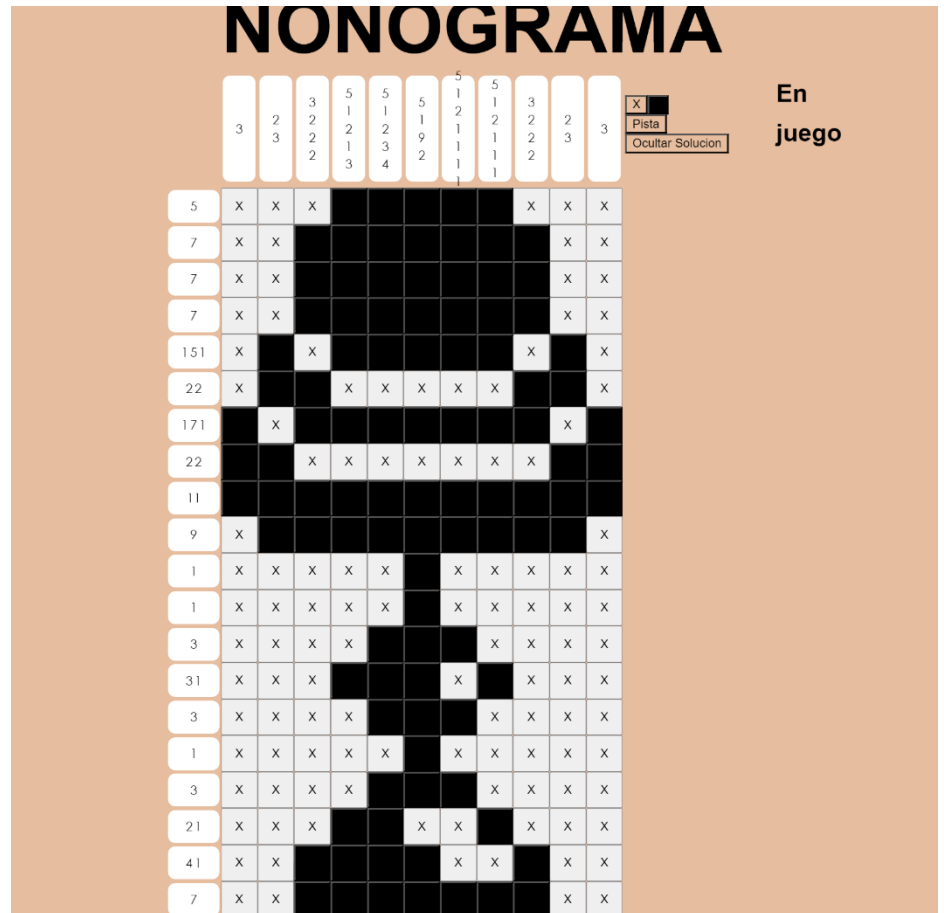
`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_],`

`[[_,_,_,_,_,_,_,_,_,_,_,_]]).`



```
:- module(init, [ init/3 ]). 30x30
```

```
init([[2,1,1,2],[4,3,1,1,2,2],[1,7,1,3,2,1],[1,3,1,2,2,2],[2,1,1,1,2],[2,1,1],[3,7,3],[2,10,2],[2,18,2],[3,2,14,2],[2,1,5,6,2,3,3],[1,7,7,10],[1,8,7,9],[1,1,6,8],[13,4,11],[14,6],[1,16,3],[1,16],[18],[10,7],[3,6,8],[4,6,7],[4,5,1,7],[5,4,2,6],[5,4,3,5],[6,4,4,5],[5,4,4,5],[4,4,4,5,4],[5,5,4,5,2,3],[5,4,4,7]],
%PistasFilas
```

```

[[3,2,4,5],[2,3,4,12],[3,4,1,17],[7,1,18],[1,5,1,9,6,2],[3,1,10,3],[2,1,13],[2,1,19],[2,1,20],[1,20],[22],[2,6,1,2],[5,7,6],[7,7,7],[1,1,8,7,6],[9,8,5],[9,
9,2],[1,1,9,11],[1,4,3,1,11],[1,1,5,2,9],[2,10,7],[1,1,4,6,4],[2,1,10,2],[1,1,9,1],[1,2,5,2],[3,4,2],[1,1,1,4,1,1],[3,1,5,3],[9,1,3],[6,1,3]],
%PistasColumnas

```

[(' ',' ',' ',), (' ',' ',' ',), (' ',' ',' ',)
 ,)]

```
[_,_,_,,,,,,,,,,,,,,,,,,,,,,,,,,,,,),[_,_,_,,,,,,,,,,,,,,,,,,,,,,,,,,,,,),[_,_,_,  
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,],
```

[illegible]

[,],

[_, _, _ ,,,,,,,,,,,,,,,],

[illegible][illegible]

[,] ,

[_, _, _ ,,,,,,,,,,,,,,,],

[illegible][illegible][illegible][illegible][illegible][illegible]

[_____,][_____,]

[_____, _____], [_____, _____],

[_____, _____], [_____, _____],

[illegible]

[_____])

