



PROYECTO LÓGICA PARA CIENCIAS DE LA COMPUTACIÓN

NONOGRAMA

Comisión 18

Integrante 1: [Giacomodonato Giulia - 142049]

Integrante 2: [Kreczmer Tomás - 140009]

Proyecto 1: Nonograma

Implementación en Prolog

A partir del código molde de la cátedra, la estrategia general que usamos para resolver el problema fue que cada vez que el jugador pinta un celda, corroborar si la fila y/o la columna de la celda satisfacen sus respectivas pistas.

Primero obtenemos las pistas de la fila o columna de la celda pintada por el jugador, y con la lista de elementos de la grilla resultante, recorremos ambas para verificar que coincidan el número y la disposición de las celdas pintadas en la grilla con la de las pistas de la lista.

Para obtener la columna de la grilla, utilizamos un predicado incorporado en Prolog *nth0/1*, que toma un índice y devuelve el elemento en esa posición de la lista. En este caso, se usa *nth0/1* para extraer el elemento en la posición de la columna buscada de cada lista en Grilla.

Luego, *maplist/3* se utiliza para aplicar un predicado a cada elemento de una lista o a elementos correspondientes de varias listas. Le aplicamos *nth0/1* a cada lista de la Grilla, utilizando el mismo índice de la columna buscada, y los resultados se unifican en Columna.

Para verificar si una fila o una columna está satisfecha, utilizamos *nth0/3* para acceder a un elemento en una lista dada su posición. En este caso, para obtener las pistas específicas para una fila o columna de la grilla, según el índice proporcionado. El resultado también se unifica con *PistasAVerificar*.

Tanto como para filas y columnas utilizamos el mismo predicado *verificarPistasEnLista* que recorre la lista de pistas y la fila/columna de la celda. Luego utilizando un predicado auxiliar *verificarPistasConsecutivas* comprobamos que se cumplan las pistas con un número mayor a 1 (celdas que deben ir pintadas consecutivamente).

En resumen, el código realiza la colocación de un contenido en la grilla, y luego verifica si se cumplen las pistas específicas en las filas y columnas donde se colocó el contenido, asegurando así que la colocación sea válida según las reglas del Nonograma.

Uso de React

- Para indicar visualmente si las pistas de las filas y columnas están satisfechas, realizamos con CSS una pequeña animación que colorea la pista satisfecha.
- Inicialmente, recorremos todas las filas y las columnas de la grilla y consultamos si ya se estaba cumpliendo alguna pista.
- Para permitir el modo de marcado, utilizamos dos botones que se colorean según cuál es el seleccionado.
- Para verificar que se resolvió el Nonograma, recorremos las pistas de las filas y de las columnas y corroboramos que todas estén satisfechas.
- Al concluir el juego, se presenta una animación que ilustra la resolución del Nonograma.

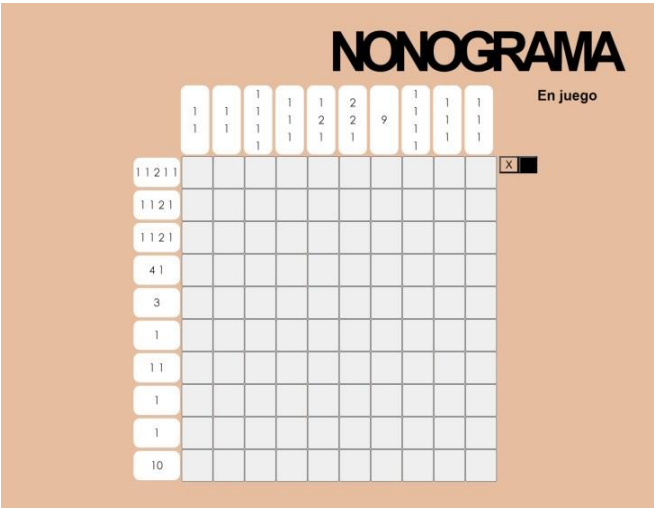
Casos de prueba

Para los siguientes casos de prueba usamos este init:

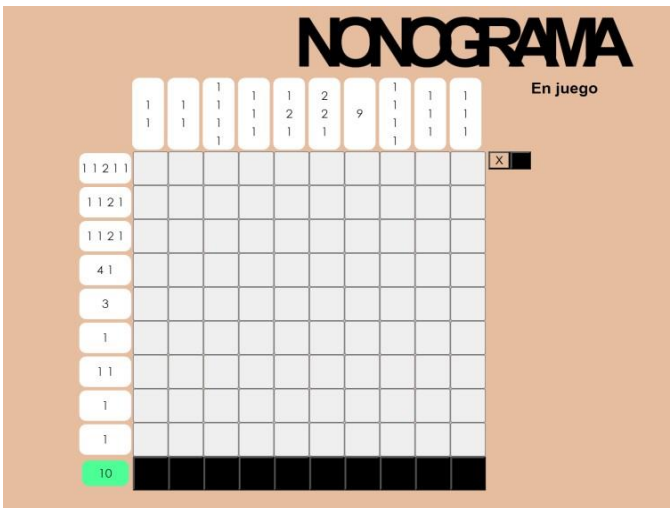
```
:- module(init, [ init/3 ]).  
init(  
    [ [1,1,2,1,1] , [1,1,2,1], [1,1,2,1], [4,1], [3], [1], [1,1], [1], [1], [10] ],      % PistasFilas  
  
    [ [1,1], [1,1], [1,1,1,1], [1,1,1], [1,2,1], [2,2,1], [9], [1,1,1,1], [1,1,1], [1,1,1]],  % PistasColumnas  
  
    [[ _ , _ , _ , _ , _ , _ , _ , _ , _ , _ ],  
     [ _ , _ , _ , _ , _ , _ , _ , _ , _ , _ ],  
     [ _ , _ , _ , _ , _ , _ , _ , _ , _ , _ ],  
     [ _ , _ , _ , _ , _ , _ , _ , _ , _ , _ ],  
     [ _ , _ , _ , _ , _ , _ , _ , _ , _ , _ ],  
     [ _ , _ , _ , _ , _ , _ , _ , _ , _ , _ ],  
     [ _ , _ , _ , _ , _ , _ , _ , _ , _ , _ ],  
     [ _ , _ , _ , _ , _ , _ , _ , _ , _ , _ ],  
     [ _ , _ , _ , _ , _ , _ , _ , _ , _ , _ ],  
     [ _ , _ , _ , _ , _ , _ , _ , _ , _ , _ ] ]  
).
```

Se verifica una fila

Antes:

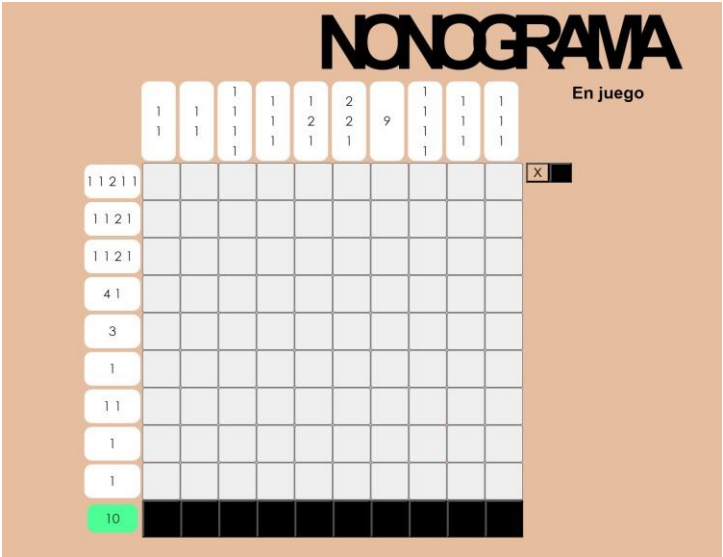


Después:

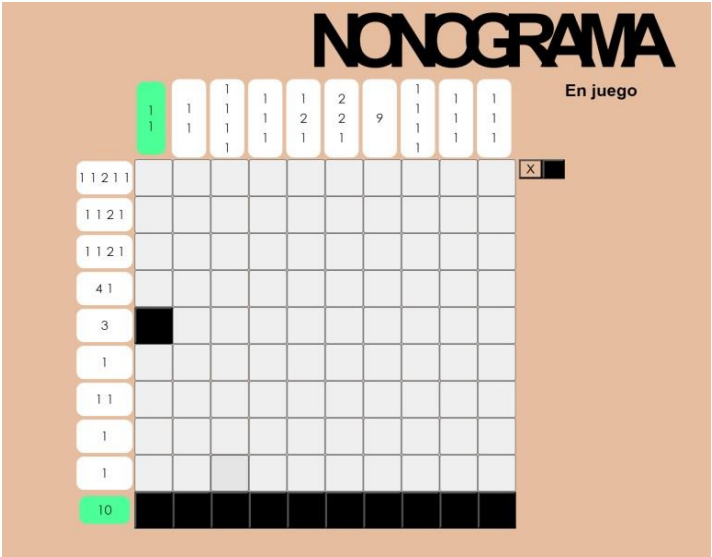


Se verifica una columna

Antes:

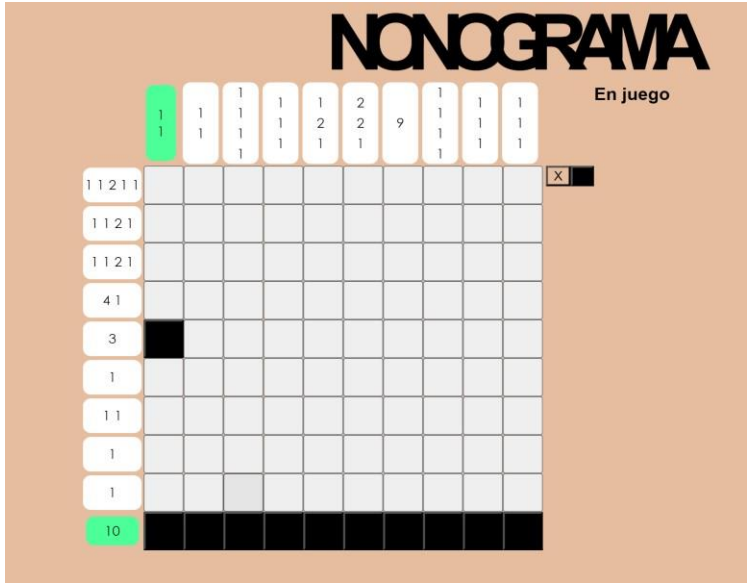


Después:

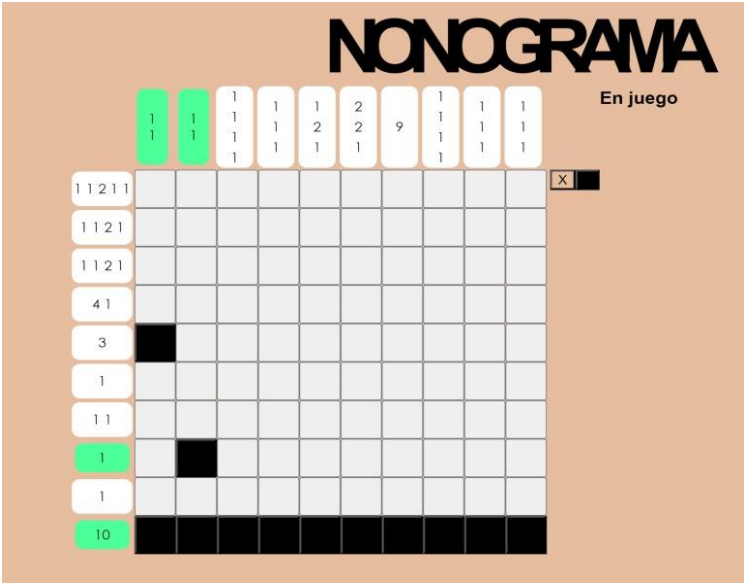


Se verifican una fila y una columna a la vez

Antes:



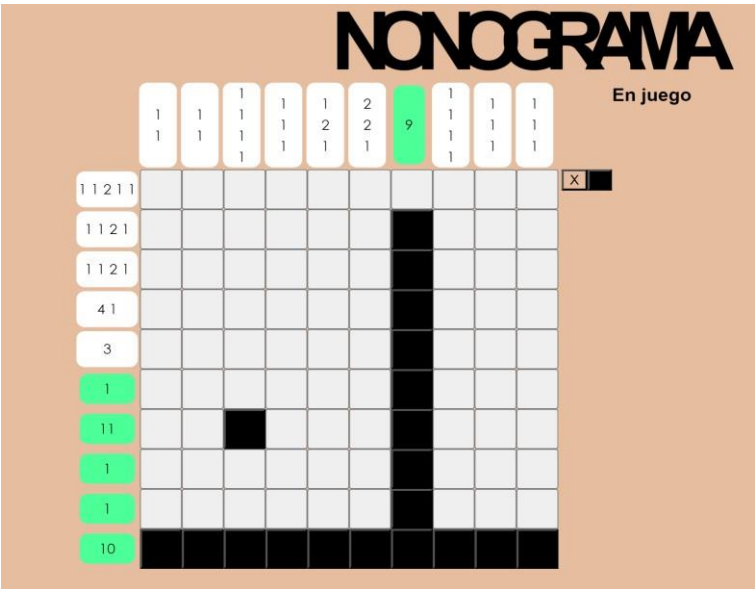
Después:



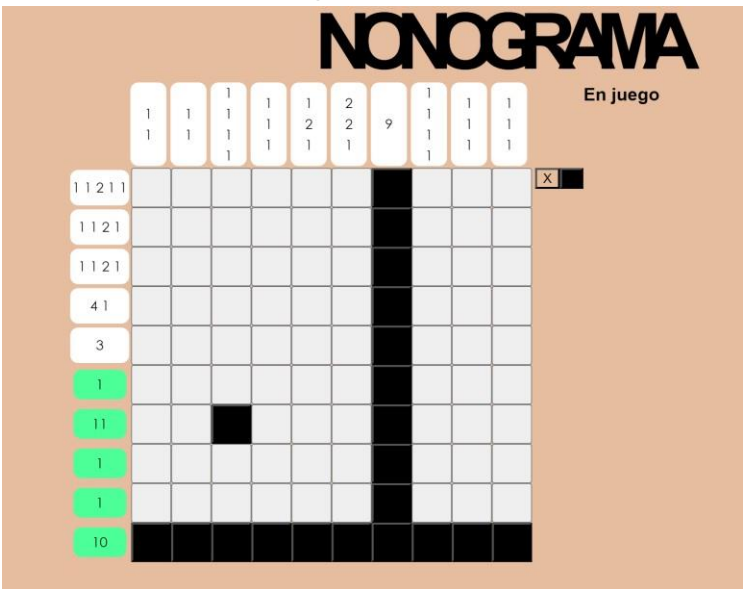
Se deja de verificar una pista

Si la pista se cumplía y pintamos otra celda.

Antes:



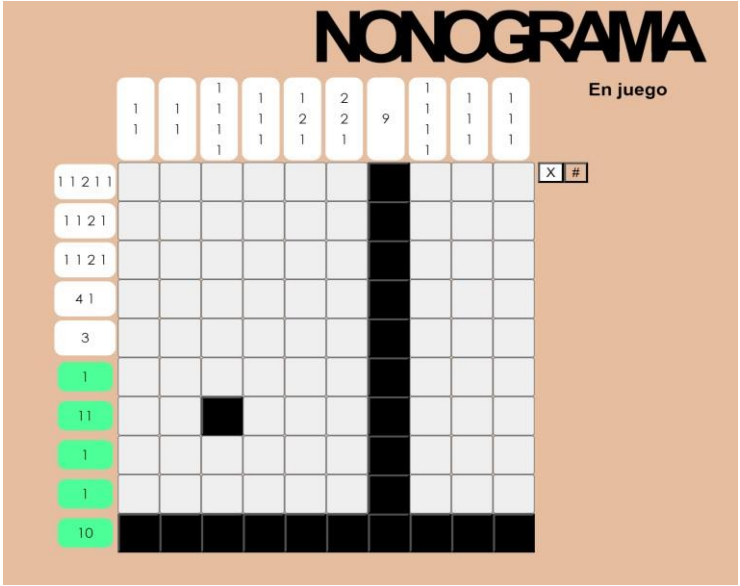
Después:



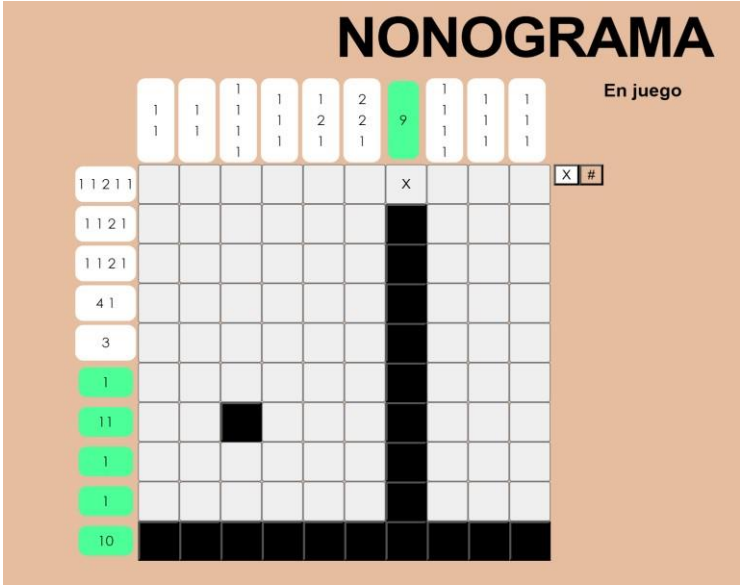
Marcado de cruces

Si la pista no se cumplía y marcamos con una cruz/despintamos la celda y pasa a verificarse.

Antes:

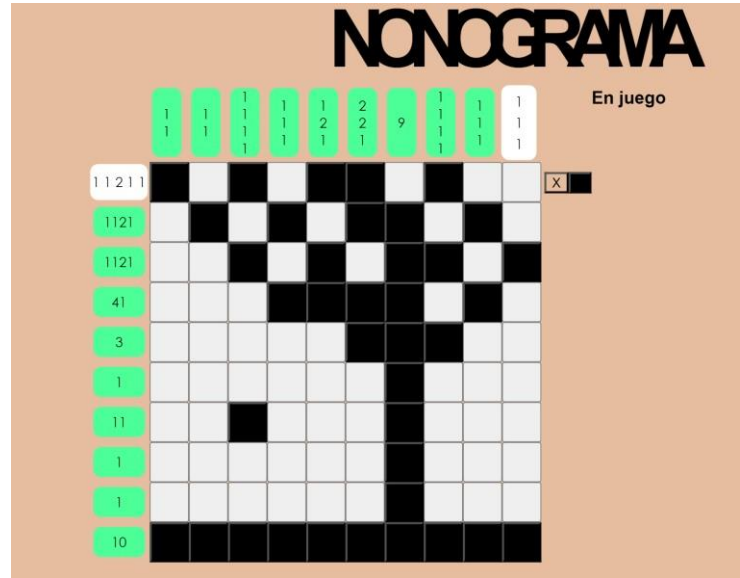


Después:

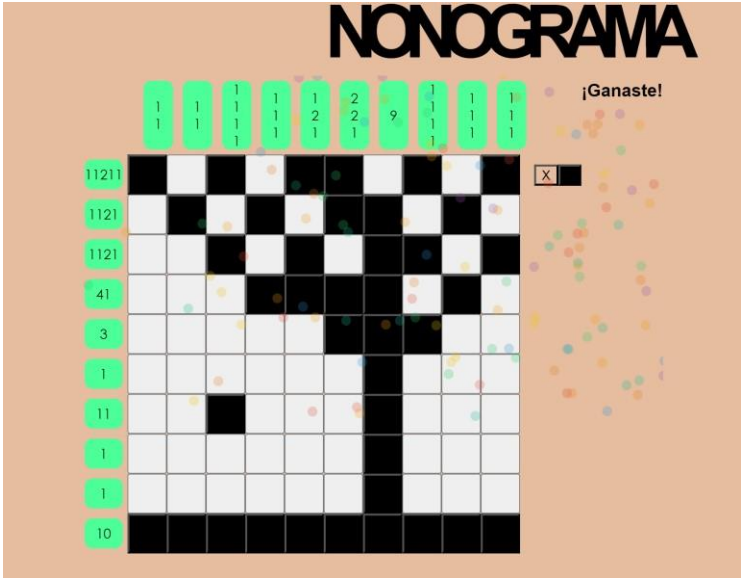


Se resuelve el Nonograma

Antes:



Después:



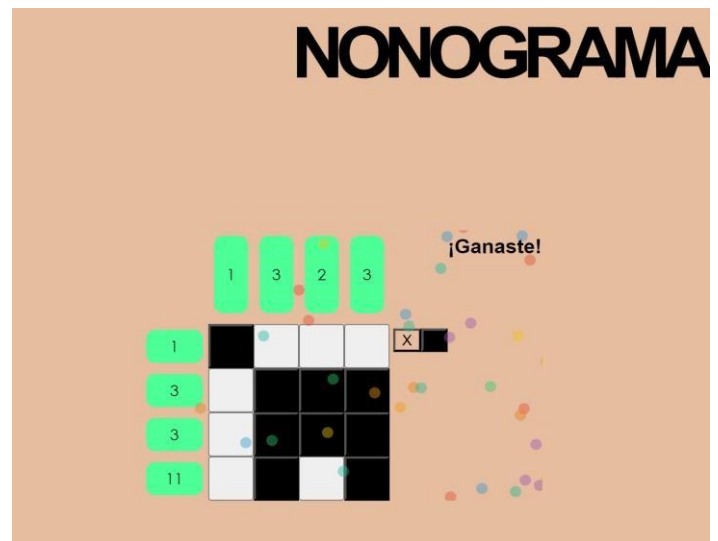
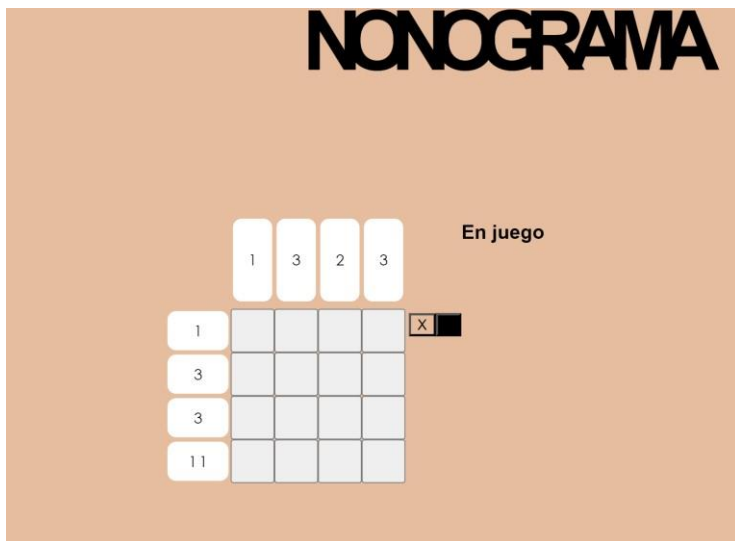
Cambia el tamaño de la grilla

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

init(
    [ [1], [3], [2], [3] ],    % PistasFilas

    [ [1], [3], [3], [1,1] ], % PistasColumnas

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

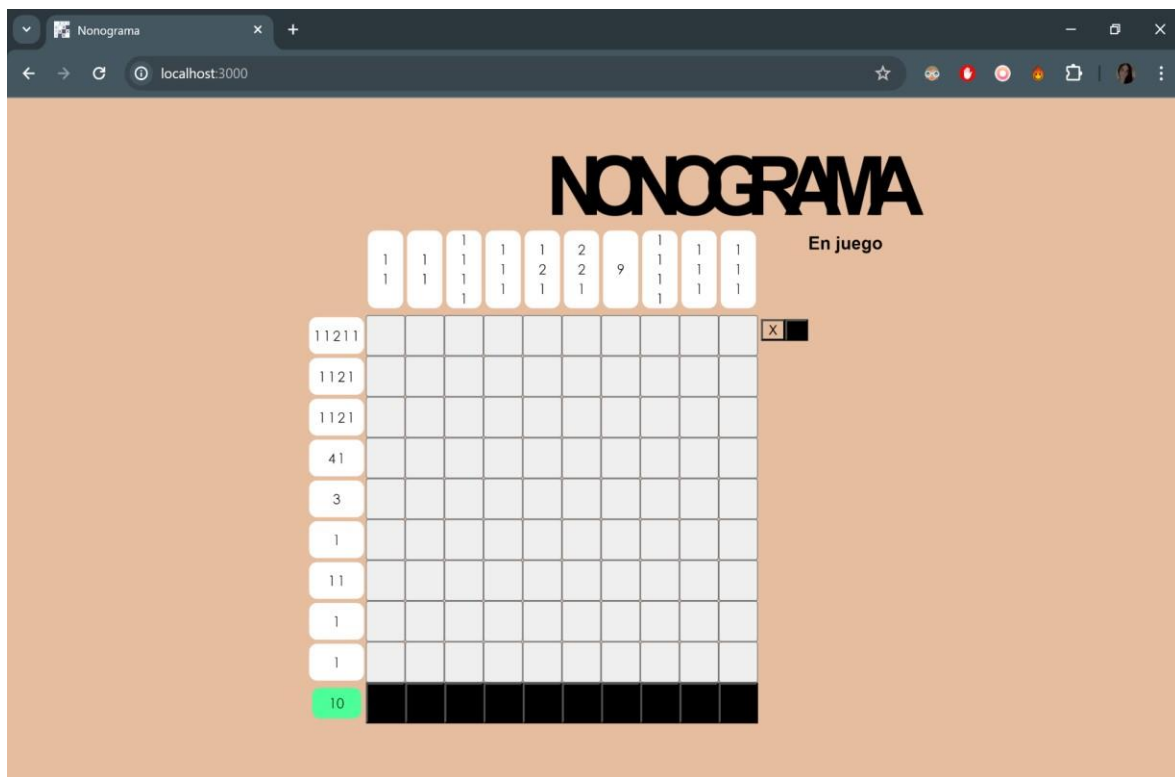


Comienza la grilla con una fila satisfecha

```
:- module(init, [ init/3 ]).
init(
    [ [1,1,2,1,1], [1,1,2,1], [1,1,2,1], [4,1], [3], [1], [1,1], [1], [1], [10] ],      % PistasFilas

    [ [1,1], [1,1], [1,1,1,1], [1,1,1], [1,2,1], [2,2,1], [9], [1,1,1,1], [1,1,1], [1,1,1]],  % PistasColumnas

    [[_,_,_,_,_,_,_,_,_,_,_],
     [_,_,_,_,_,_,_,_,_,_,_],
     [_,_,_,_,_,_,_,_,_,_,_],
     [_,_,_,_,_,_,_,_,_,_,_],
     [_,_,_,_,_,_,_,_,_,_,_],
     [_,_,_,_,_,_,_,_,_,_,_],
     [_,_,_,_,_,_,_,_,_,_,_],
     [_,_,_,_,_,_,_,_,_,_,_],
     [_,_,_,_,_,_,_,_,_,_,_],
     [_,_,_,_,_,_,_,_,_,_,_],
     ["#", "#", "#", "#", "#", "#", "#", "#", "#", "#"] ]
).
```

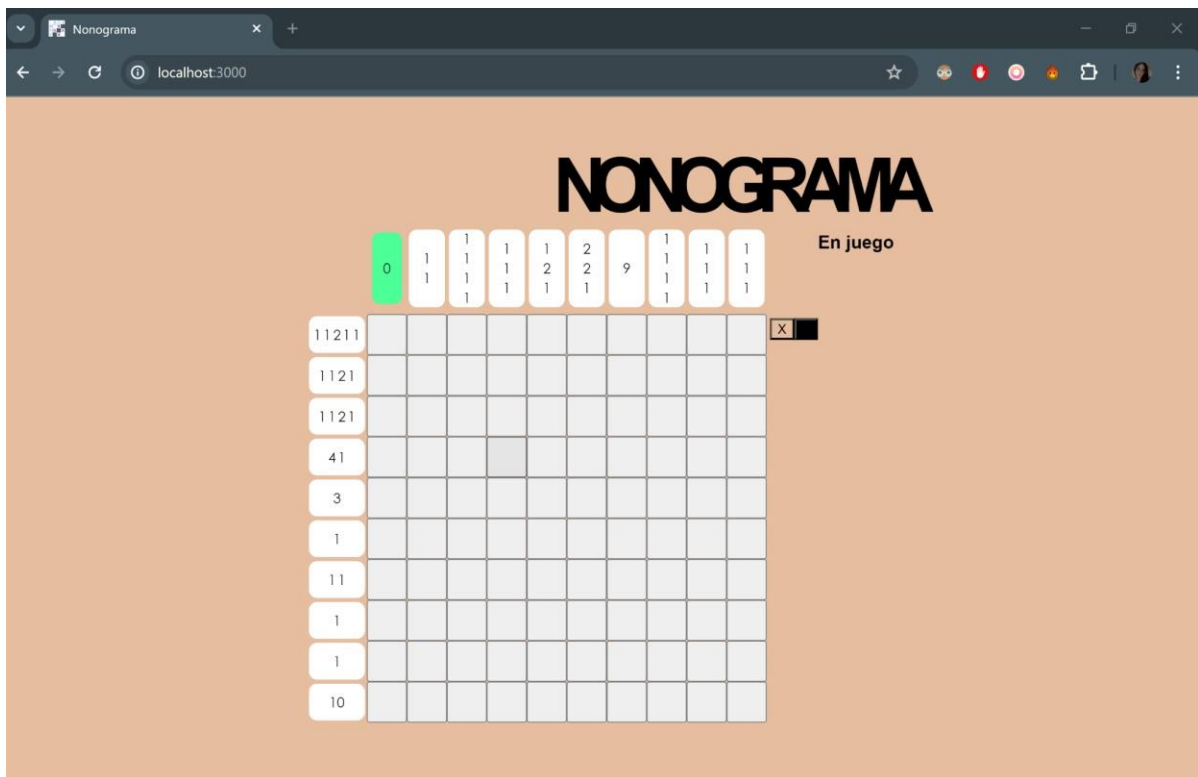


Comienza la grilla con una columna satisfecha

```
:- module(init, [ init/3 ]).
init(
  [ [1,1,2,1,1] , [1,1,2,1], [1,1,2,1], [4,1], [3], [1], [1,1], [1], [1], [10] ],      % PistasFilas

  [ [0], [1,1], [1,1,1,1], [1,1,1], [1,2,1], [2,2,1], [9], [1,1,1,1], [1,1,1], [1,1,1]],      % PistasColumnas

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



Se puede resolver el Nonograma de diferentes formas

Si init es ambiguo y el Nonograma tiene dos posibles resoluciones

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

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

