

Nonograma documentación

Descripción general.

El objetivo principal del proyecto es implementar una aplicación web del juego Nonograma. Durante el transcurso del proyecto aprendimos nuevas tecnologías y herramientas para llevar a cabo la aplicación web. Pudiendo utilizar el lenguaje Prolog para lo que es la parte lógica, como también React para la gráfica de la aplicación. Además, vimos un poco de HTML, CSS y JavaScript, de estos últimos tres lenguajes podemos destacar que:

- HTML se utiliza para el desarrollo y creación de páginas web, el cual es un lenguaje de marcado que nos permite indicar la estructura de nuestro documento mediante etiquetas.
- CSS es un lenguaje que maneja el diseño y presentación de las páginas web, es decir, cómo lucen cuando un visitante las visita.
- JavaScript su principal función es incluir contenido “dinámico” a las páginas web.

Parte I: Desarrollo en Prolog.

En prolog, en el predicado **put**, está definido de la siguiente manera:

```
put(+Contenido, +Pos, +PistasFilas, +PistasColumnas, +Grilla, -GrillaRes, -FilaSat, -ColSat).
```

- **Entradas:**
 - **Contenido:** Es el valor que toma el square una vez que el visitante hace click sobre el mismo, los valores que puede tomar son: “#”, “X” y “_”.
 - **Pos:** Es la posición [FilaN, ColN] de la grilla que fue clickeada. Siendo FilaN, la fila enésima de la grilla y ColN la columna ésima de la grilla.
 - **PistasFilas:** Es una lista de listas que indica las pistas de cada fila.
 - **PistasColumnas:** Es una lista de listas que indica las pistas de cada columna.
 - **Grilla:** Es una lista de listas con los elementos de cada posición de la grilla.
- **Salidas:**
 - **GrillaRes:** Es la grilla resultante al reemplazar el contenido en la posición [FilaN, ColN].
 - **FilaSat:** Retorna 1 en caso de que la fila satisfaga las pistas y 0 en caso contrario.
 - **ColSat:** Retorna 1 en caso de que la columna satisfaga las pistas y 0 en caso contrario.

Lo que realiza el predicado **put** es lo siguiente:

Toma una **Grilla** y reemplaza en la posición **Pos** un **Contenido**, si este es igual a lo que había antes, en su lugar queda un “_”, caso contrario se reemplaza lo que había por el **Contenido**.

Luego, se verifica si la fila en la que se reemplazó, cumple con su respectiva **pistasFilas**, en caso de que se verifique, **FilaSat** tendrá el valor de 1, caso contrario será 0.

Esto último sucede también con **pistasColumnas** y **ColSat**.

Toma una **Grilla** y mete el **Contenido** en la posición **Pos**. Una vez que reemplaza ese **Contenido** por el que ya había chequea si se satisface la FilaN y la ColN en base a las **PistasFilas** y **PistasColumnas** recibidas para luego asignarle los valores correspondientes a **FilaSat** y **ColSat**. Además, retorna la **GrillaRes**.

```
replace(Fila, FilaN, NewFila, Grilla, NewGrilla)
```

1. Obtenemos una nueva Grilla, que es el resultado de reemplazar la Fila en la posición FilaN de Grilla, por una fila nueva NewFila.

```
replace(Cell, ColN, _, Fila, NewFila),  
  Cell == Contenido;  
  replace(_Cell, ColN, Contenido, Fila, NewFila)),
```

2. Si lo que se ingresó (Contenido) es igual a lo que ya estaba en la posición, obtenemos una nueva Fila que es el resultado de reemplazar lo que haya en la posición por “_”. **Caso contrario** (que el contenido no sea igual a lo que ya estaba), obtenemos una nueva fila que es el resultado de reemplazar en la posición lo que se haya por contenido.

```
( getActual(FilaN,PistasFilas,PistaF),  
  FilaSat is 1, controlarSat(PistaF,NewFila); FilaSat is 0 ),
```

3. Obtenemos la pista de la fila (utilizando el predicado getActual) en la cual se realizó una modificación, y controlamos que satisfaga la pista alcanzada. En caso de que la nueva fila cumpla con la pista, FilaSat será 1, caso contrario, FilaSat es 0.

```
( getCol(ColN,NewGrilla,ColRet) , getActual(ColN,PistasColumnas,PistaC) ,  
  ColSat is 1, controlarSat(PistaC,ColRet) ; ColSat is 0 ).
```

4. Obtenemos la pista de la columna (utilizando el predicado getActual) en la cual se realizó una modificación, y controlamos que satisfaga la pista alcanzada. En caso de que la columna cumpla con la pista, ColSat será 1, caso contrario, ColSat es 0.

Descripción de predicados **principales**:

- **verificarPistas:**

Toma la lista y la cantidad de # consecutivos y devuelve **true** si la lista contiene esa cantidad y lo que sigue al último consecutivo no es un #. Luego retorna por parámetro la lista restante.

El predicado toma como parámetros de entrada :

- Cantidad N de “#” consecutivos que debería aparecer en la lista.
- Una lista que representa una fila o columna del tablero.

Y el parámetro de salida

- La lista restante luego de eliminar los N “#” consecutivos.

- **controlarSat:**

El predicado toma como parámetros de entrada:

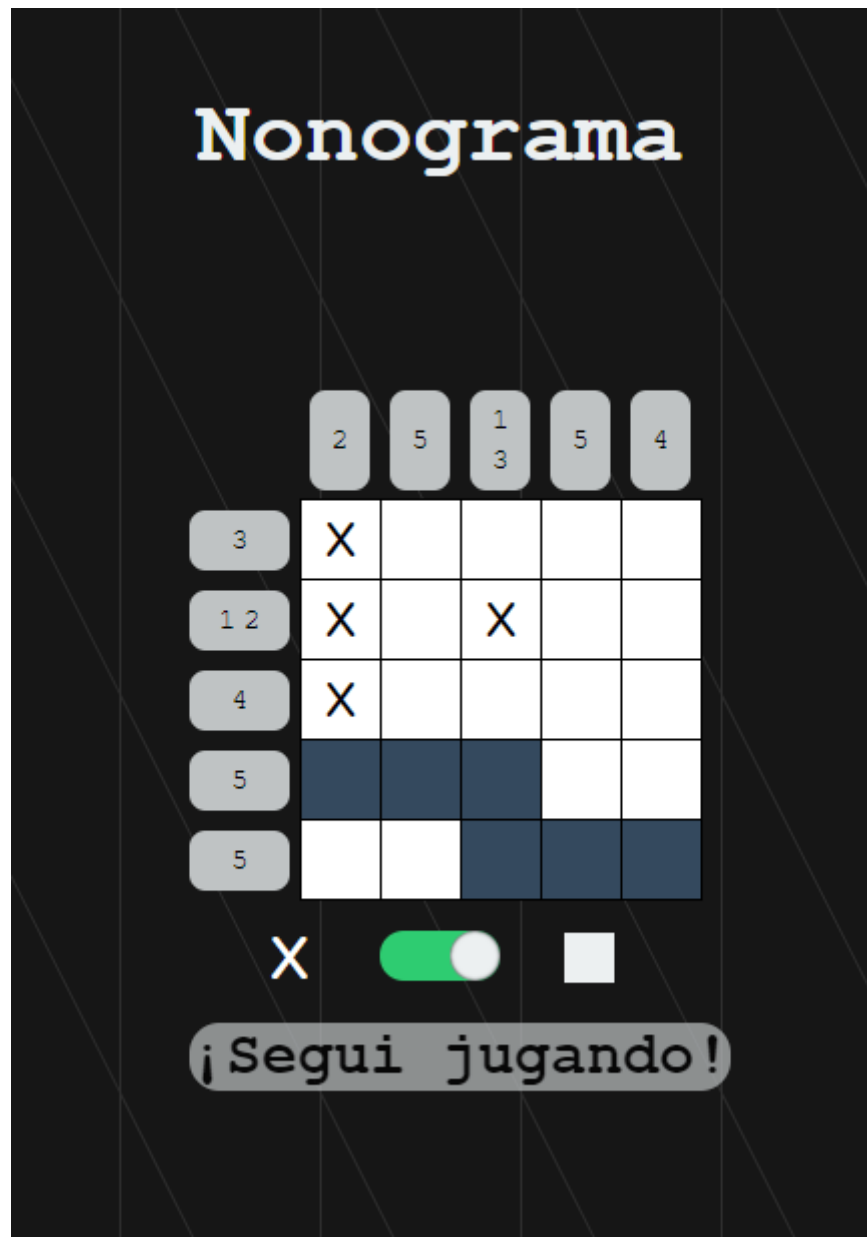
- Una lista que representa a una pista de una linea en el tablero y otra que representa una fila o columna en el tablero.

Toma una lista (de fila o columna) y devuelve **true** en caso de que la misma cumpla con la lista de pistas, **false** caso contrario.

Parte II: UI en React.

- Manual de usuario:
 - Interfaz e interacción:

Imagen de la interfaz.



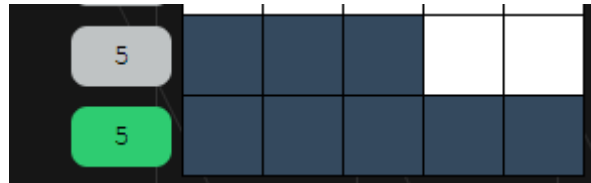
- El juego comienza automáticamente, en la interfaz de la aplicación podemos ver una grilla la cual desde un principio puede estar parcialmente completa con celdas en las cuales sus valores son X, celdas vacías (color blanco) o celdas pintadas (color azul oscuro). En la imagen de arriba se puede apreciar lo mencionado.
- Al inciar el juego, la opción por default para empezar a jugar es el “pintado”. Esto se puede observar en el botón toggle de abajo de la grilla, el cual está del lado del cuadrado blanco. Este símbolo representa el modo “pintar”. En la siguiente imagen representamos lo mencionado anteriormente:



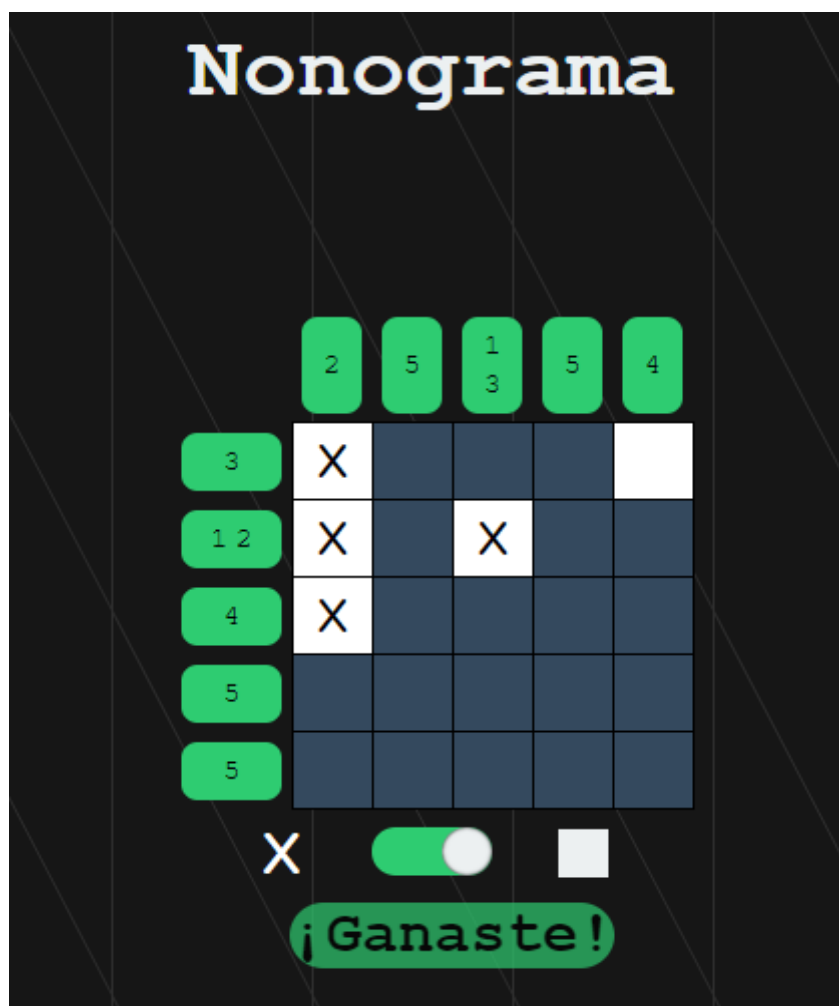
- Para seleccionar el modo “X”, apretar el botón toggle y quedará del lado izquierdo (cerca del símbolo “X”). Este modo sirve para indicar que celdas no deben ir pintadas (no afectan a la solución, es una forma de facilitar la solución del juego). En la imagen siguiente podemos observar el botón en el modo “X”.



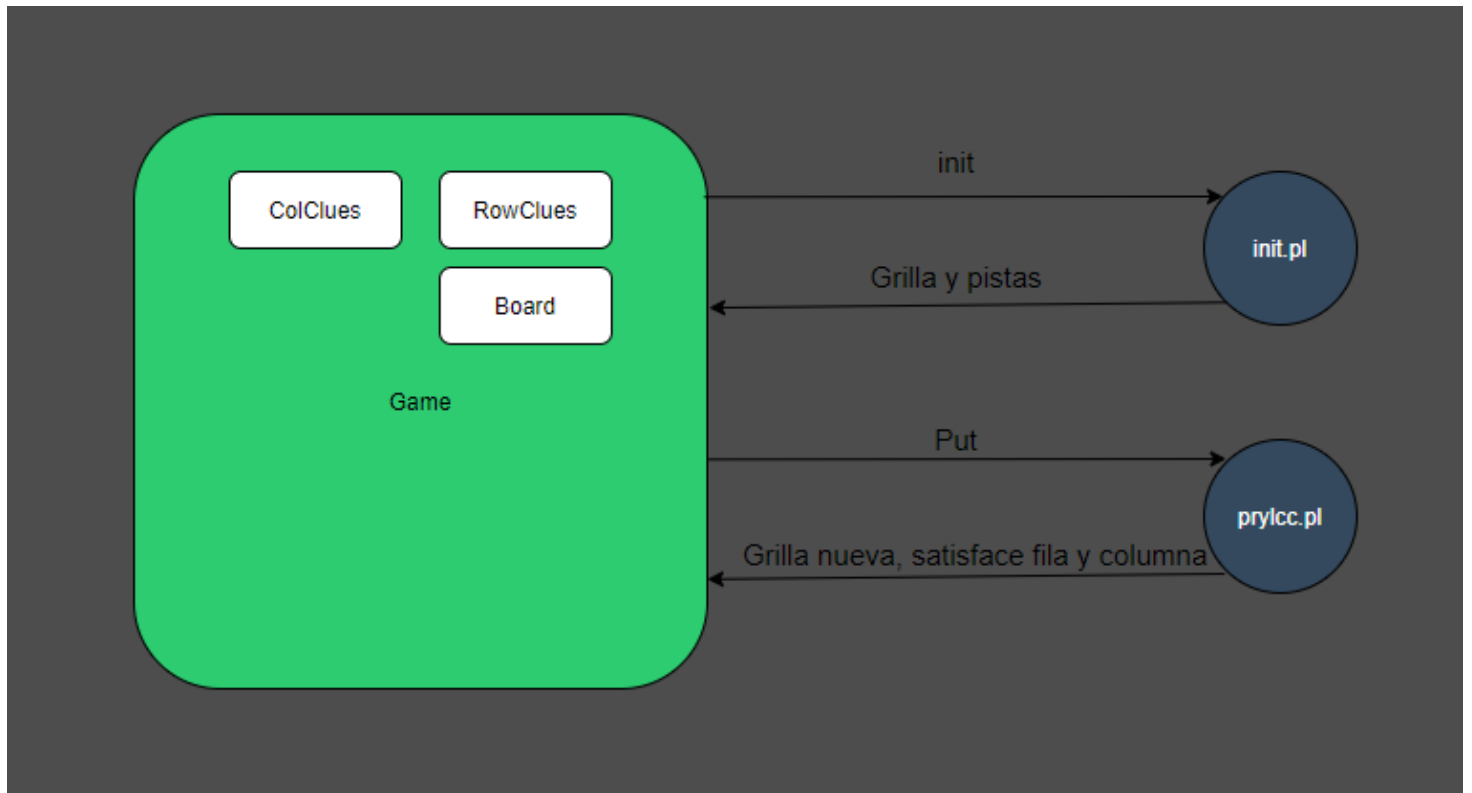
- Las respectivas pistas para cada fila y columna de la grilla se encuentran del lado izquierdo y en la parte superior de la grilla, las cuales aparecen de color gris en caso de no estar satisfechas y de color verde en caso de satisfacerse. Como se muestra en la siguiente imagen:



- Por otro lado se encuentra el estado del juego, ubicado entre la grilla y el botón, que en caso de que el juego no haya terminado dice la siguiente expresión: “Segui jugando!”. Una vez que se satisfacen todas las pistas, el juego finalizará y el estado cambia a: “¡Ganaste!”. La siguiente modela lo mencionado.



- Diagrama de componentes:



A la vuelta del **put** con la grilla nueva, primero actualizamos visualmente la grilla y luego comprobamos si se satisface la fila y/o columna para cambiar el color de las pistas de acuerdo a lo ocurrido. Además, chequeamos si finalizó el juego, en caso de finalizar actualizamos el estado y lo mostramos visualmente.