

# PROYECTO FINAL INTELIGENCIA ARTIFICIAL I

Minería de reglas de asociación espacial  
sobre Minecraft

Martín Cogo Belver

# MOTIVACIÓN



- El objetivo es resolver algún problema utilizando **algoritmos de Inteligencia Artificial**.
- Dentro de la industria de creación de videojuegos existe un área centrada en la generación procedural.
- *¿Sería posible, con un algoritmo de inteligencia artificial y un entorno (artificial o natural), poder inferir parámetros para que un algoritmo de generación procedural genere entornos similares?*

- La obtención de datos de un entorno real, resulta complejo.
- Se utiliza una porción de mapa del videojuego **Minecraft** que utiliza la generación procedural para la creación de terrenos del juego utilizando cubos.
- Los datos del videojuego son: de **fácil acceso y extracción**, **fácil generación** y la **disposición de los objetos** dentro del juego son bloques con **coordenadas rectangulares**.

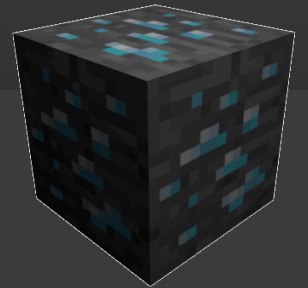
# ENFOQUE



- La **Minería de Datos Espaciales** o **SDM**:
  - Es un proceso de descubrimiento
  - Extracción de conocimiento
  - Enfocada sobre bases de datos espaciales o geográficas.
- Durante el trabajo buscaremos descubrir el patrón de posicionamiento de los bloques del juego.

- Para la *búsqueda de patrones* en datos espaciales se tomó el enfoque basado en **reglas de asociación**.
- La parte más importante es la creación de transacciones booleanas sobre el espacio de tal manera que se pueda usar el algoritmo **Apriori**.
- También se generarán **gráficos para el análisis de resultados**.

# MÉTRICAS



Para evaluar las reglas utilizaremos las siguientes métricas:

- Soporte: Se trata de la probabilidad de que aparezca X e Y en las transacciones.

$$\text{Support}(X \rightarrow Y) = \frac{\text{Transaction containing X and Y}}{\text{Total number of transactions}}$$

- Confianza: Mide la probabilidad de que aparezca el Y dado que en una transacción aparece X.

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Transaction containing X and Y}}{\text{Transactions containing X}}$$



# RESULTADOS

Experimentos:

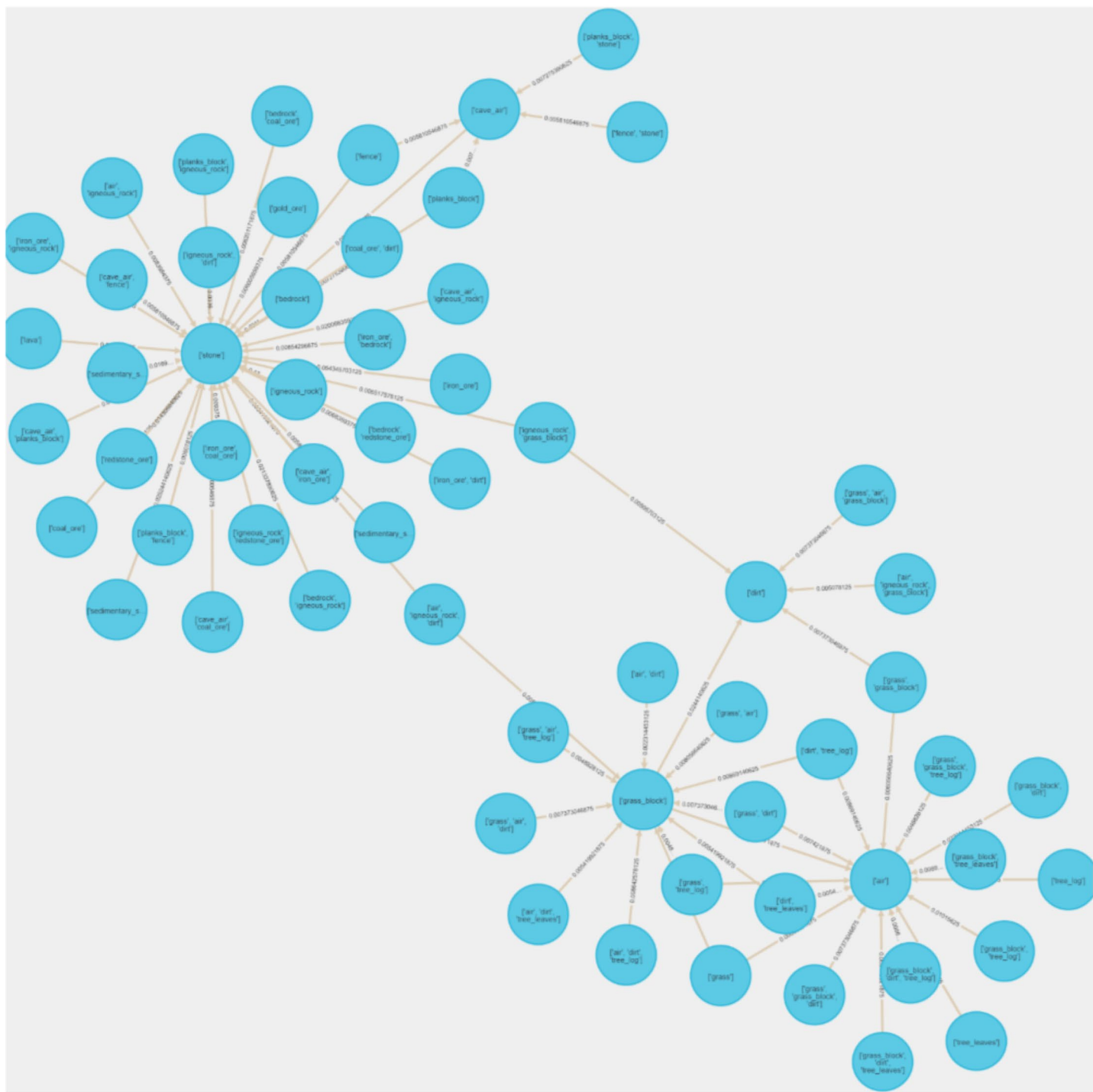
1. Windows centric model
  1. Transacciones con ventanas de 4x4x4.
2. Reference feature centric model
  1. Transacciones centradas en el atributo **diamond**. (Diamante)
  2. Transacciones centrada en el atributo **grass\_block**. (bloques de pasto)
  3. Transacciones centradas en el atributo **tree\_log** y **tree\_leaves**. (troncos y hojas)
  4. Transacciones centradas en todos los atributos de un **chunk**. (todos los posibles bloques)



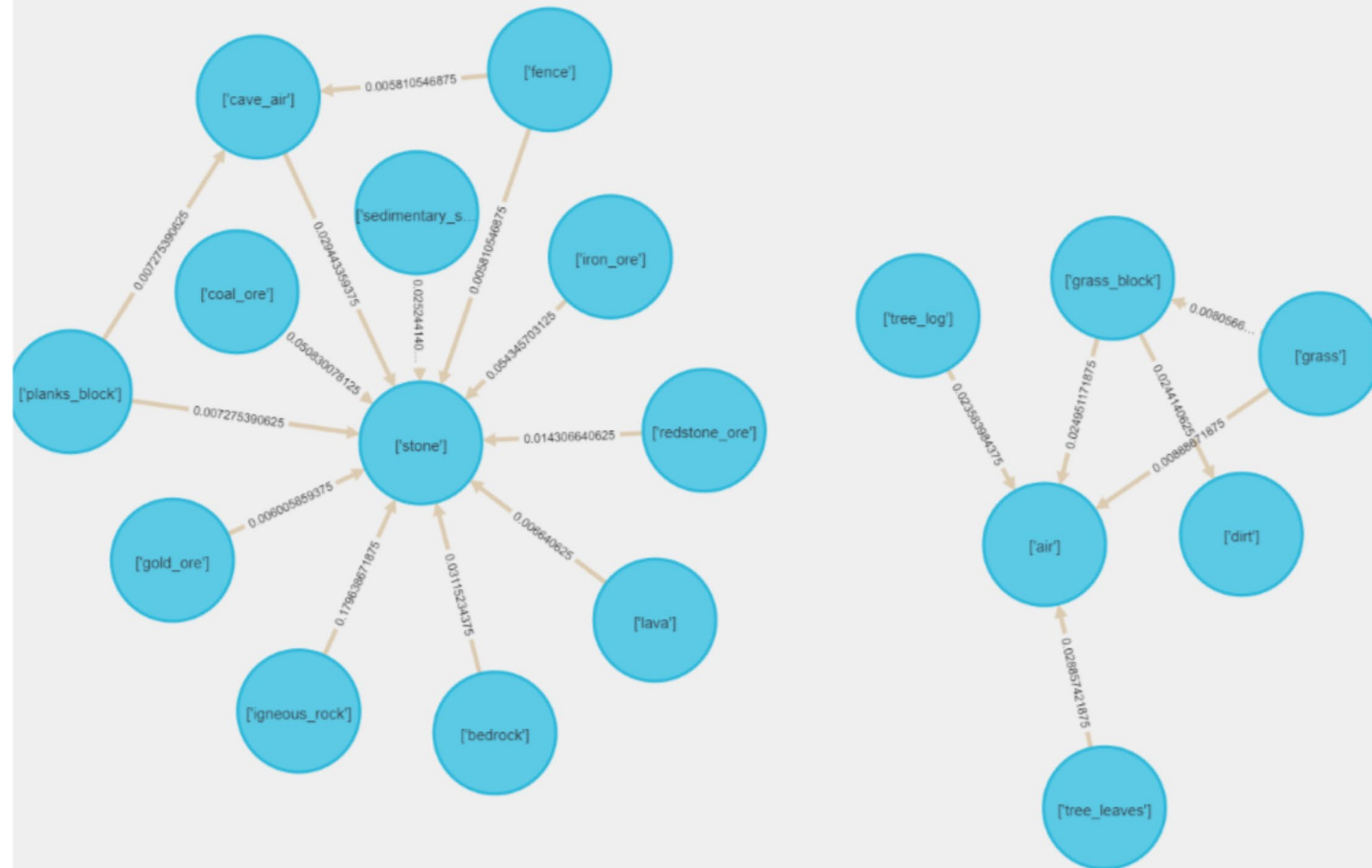
# Transacciones de ventanas de 4x4x4 sin solapamiento

```
`tiempo de preprocesamiento` = 9 minutos
`total de transacciones` = 20480
`soporte mínimo` =  $100/20480 = 0.0049$ 
`confianza mínima` = 90%
```

- Las transacciones implementan un `Window centirc model` y en este caso son ventanas de  $4 \times 4 \times 4$  es decir cubos de 64 bloques.
- En el Grafo cada nodo representa un antecedente o consecuente de una regla.
- Al tratarse de un grafo muy grande resulta muy difícil analizarlo.



- El grafo ilustra el conjunto de reglas podadas y clasificadas como interesantes.
- Las demás reglas no podadas y clasificadas como no interesantes no tienen más información que la que se puede interpretar desde este grafico.
- El análisis resulta sencillo en este grafo.





- Estas son las reglas interesantes graficadas en el grafo anterior.

- El **soporte es bajo** y la **confianza es alta** en todas.

- El soporte es tan bajo debido a:

- La enorme cantidad de bloques.

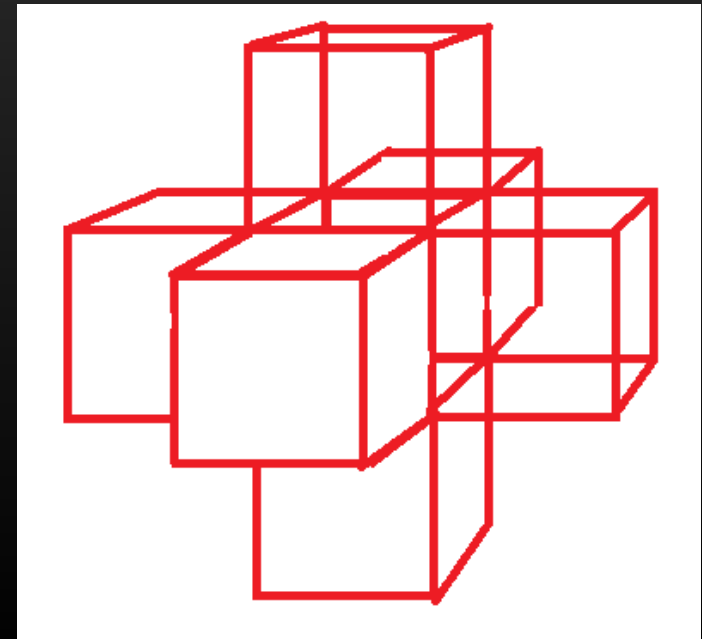
- Lo desbalanceado que se encuentra el conjunto de bloques de entrada.

id	antecedants	=>	consequents	support	confidence
0	['coal_ore']	['=>']	['stone']	0.050830	1.000000
1	['lava']	['=>']	['stone']	0.006641	1.000000
2	['tree_leaves']	['=>']	['air']	0.028857	1.000000
3	['grass']	['=>']	['grass_block']	0.008057	0.906593
4	['grass_block']	['=>']	['dirt']	0.024414	0.902527
5	['iron_ore']	['=>']	['stone']	0.054346	1.000000
6	['sedimentary_stone']	['=>']	['stone']	0.025244	0.966355
7	['tree_log']	['=>']	['air']	0.023584	0.997934
8	['cave_air']	['=>']	['stone']	0.029443	0.982085
9	['grass_block']	['=>']	['air']	0.024951	0.922383
10	['planks_block']	['=>']	['stone']	0.007275	1.000000
11	['gold_ore']	['=>']	['stone']	0.006006	1.000000
12	['fence']	['=>']	['stone']	0.005811	1.000000
13	['fence']	['=>']	['cave_air']	0.005811	1.000000
14	['redstone_ore']	['=>']	['stone']	0.014307	1.000000
15	['igneous_rock']	['=>']	['stone']	0.179639	0.997019
16	['grass']	['=>']	['air']	0.008887	1.000000
17	['planks_block']	['=>']	['cave_air']	0.007275	1.000000
18	['bedrock']	['=>']	['stone']	0.031152	1.000000



# OTRO MODELO

- Modelar las transacciones con ventanas de 4x4x4 no es interesante.
- Simplemente nos muestra que bloque existe en una ventada dado que otro existe en ella.
- No podemos saber claramente como es el patrón entre bloques.
- **Windows centric model** no nos da las reglas que buscamos.
- Los próximos experimentos se realizan con un **Reference feature centric model**.
- Las transacciones con ventanas de la forma cruz con 7 bloques, donde el bloque que hay en el centro es de un atributo determinado.



## Atributo de referencia diamante

`tiempo de preprocesamiento` = 9 segundos

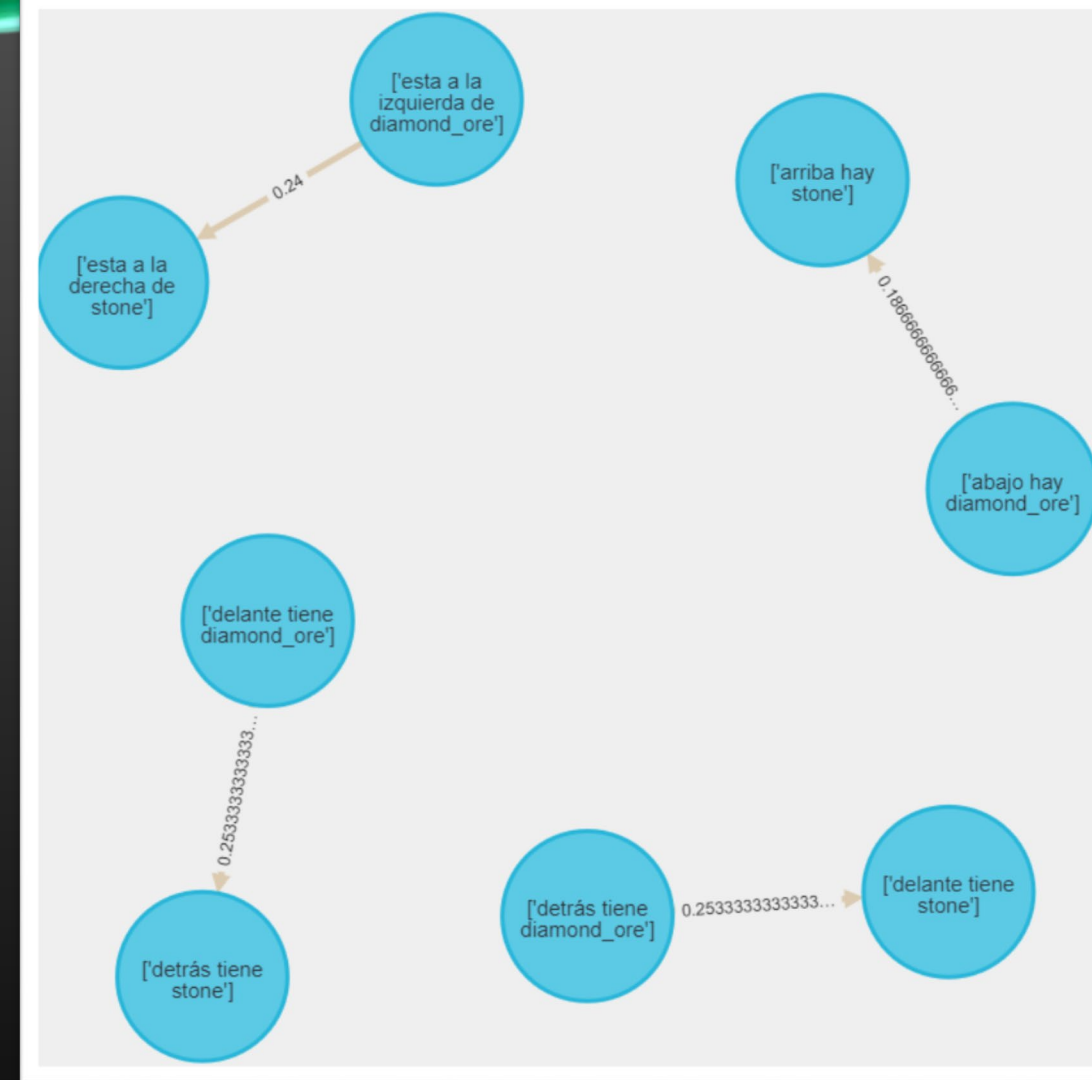
`total de transacciones` = 75

`soporte mínimo` =  $12/75 = 0.16$

`confianza mínima` = 75%

- No tenemos un nodo con ítem [es diamond\_ore].
- Es debido a que este ítem aparece en todas las transacciones generadas en el pre-procesamiento.
- En el conjunto de reglas interesantes solo tenemos 4 reglas que muestran el patrón:
- Los bloques de piedra al lado de un bloque de diamante en el juego están rodeados de bloques de piedra.

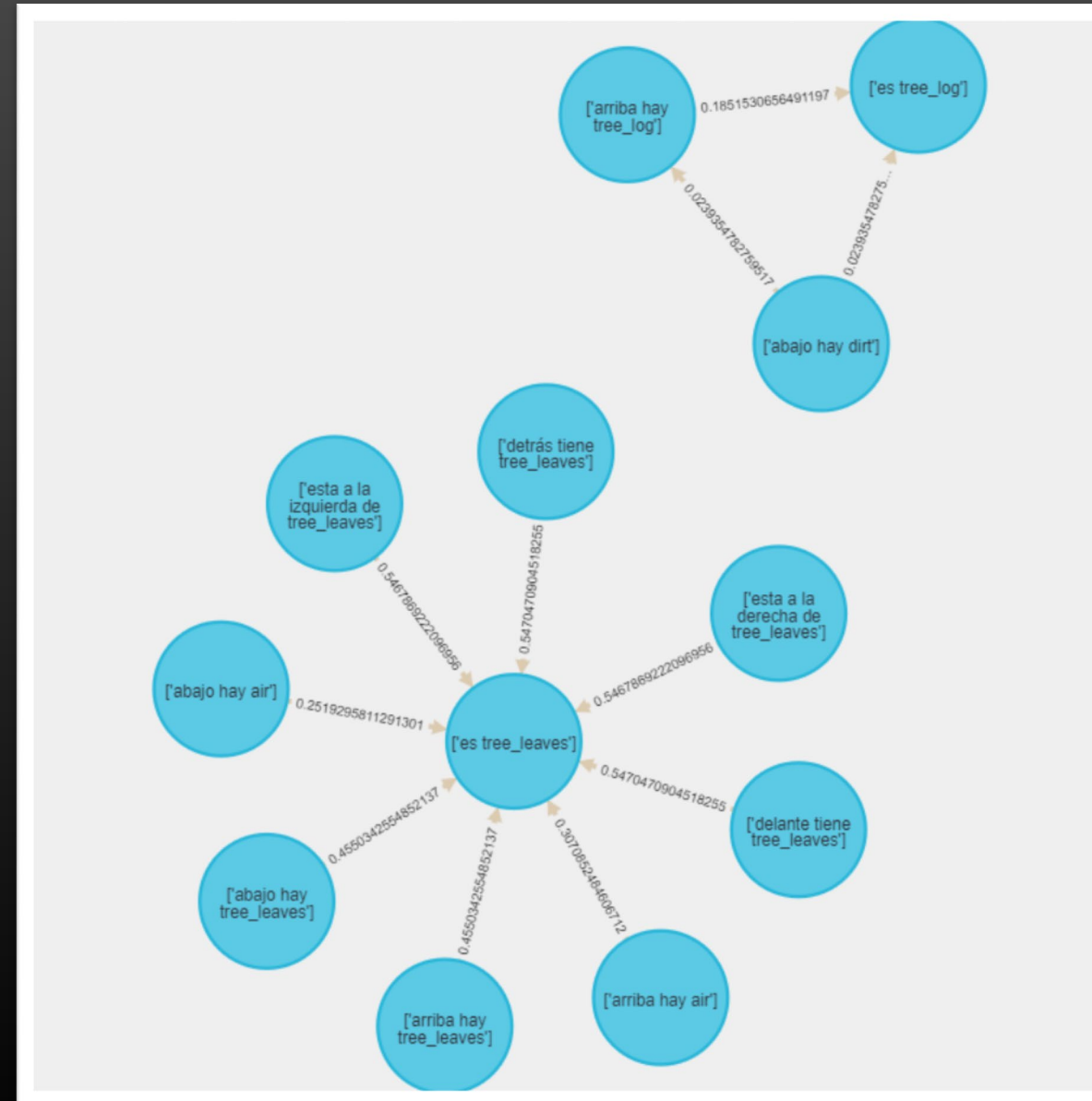
	antecedants	=>	consequents	support	confidence
0	['detrás tiene diamond_ore']	['=>']	['delante tiene stone']	0.253333	0.791667
1	['abajo hay diamond_ore']	['=>']	['arriba hay stone']	0.186667	0.823529
2	['está a la izquierda de diamond_ore']	['=>']	['está a la derecha de stone']	0.240000	0.857143
3	['delante tiene diamond_ore']	['=>']	['detrás tiene stone']	0.253333	0.791667



## Atributo de referencia hojas y troncos

`tiempo de preprocesamiento` = 24 minutos  
`total de transacciones` = 11531  
`soporte mínimo` =  $200/11531 = 0.0173$   
`confianza mínima` = 90%

- Observando el Grafo se ven claro los patrones en este conjunto:
- El patrón que indica que si hay un bloque de tierra abajo se trata de un tronco y arriba tiene otro tronco. (Corresponde con el tallo del árbol)
- El patrón que indica que si hay una bloque de tipo hoja en el juego este está rodeada de bloques de hoja y por arriba o debajo puede haber bloques de tipo aire. (Corresponde con la copa del árbol)



## Atributo de referencia pasto

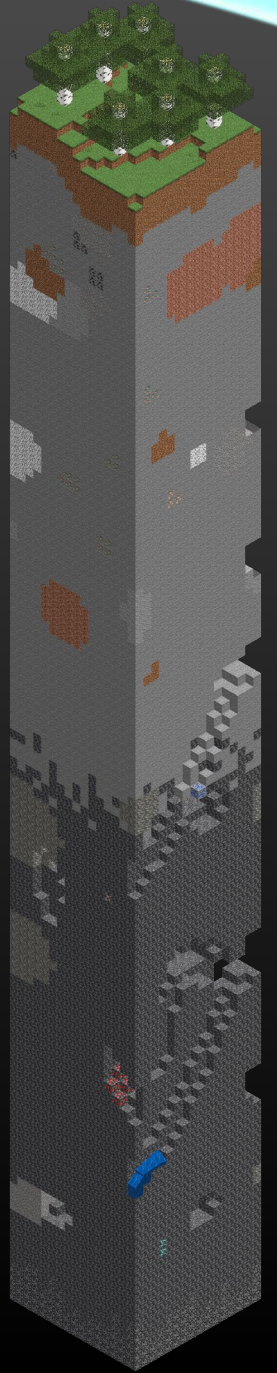
- `tiempo de preprocesamiento` = 6 minutos
- `total de transacciones` = 4108
- `soporte mínimo` =  $200/4108 = 0,0486$
- `confianza mínima` = 90%
- Las reglas del grafo tienden a relacionarse con los ítems:
  - [abajo hay dirt]
  - [arriba hay air]
- Esto muestra 2 patrones:
  - Si el bloque es de tipo pasto, debajo tiene bloques de tierra y por arriba tiene bloques de aire.
  - Si el bloque es de tipo pasto puede estar rodeado de bloques de tierra o más bloques de pasto.



- El patrón: **Abajo hay dirt entonces arriba hay flower**. No es una de las reglas obtenidas.
- Se debe a que tiene una frecuencia de aparición menor a 200.
- Aún así se trata de un patrón totalmente válido y no descubierto.
- Esto nos hace replantear si realmente estamos tomando un acercamiento correcto al problema.



- Las reglas obtenidos mediante las transacciones con forma de cruz son más cercanos a los que buscamos.
- Queremos entonces combinar la forma que toman las reglas como resultado de el **reference feature centric model** y aplicarlo en todo el espacio como con el **window centric model**.
- La complejidad temporal del algoritmo de pre-procesamiento no es lineal y depende de la cantidad de bloques de entrada.
- El tiempo necesario para generar las ventanas en cruz para todos los bloques de la porción del mapa es grande.
- Por eso, solo generaremos transacciones para un solo **chunk** que son un total de  $16 \times 16 \times 256 = 65536$  bloques.
- Por lo tanto generamos las transacciones para todos los bloques con ventanas tipo cruz que se superpondrán en todo el espacio.



## Combinación entre los modelos anteriores

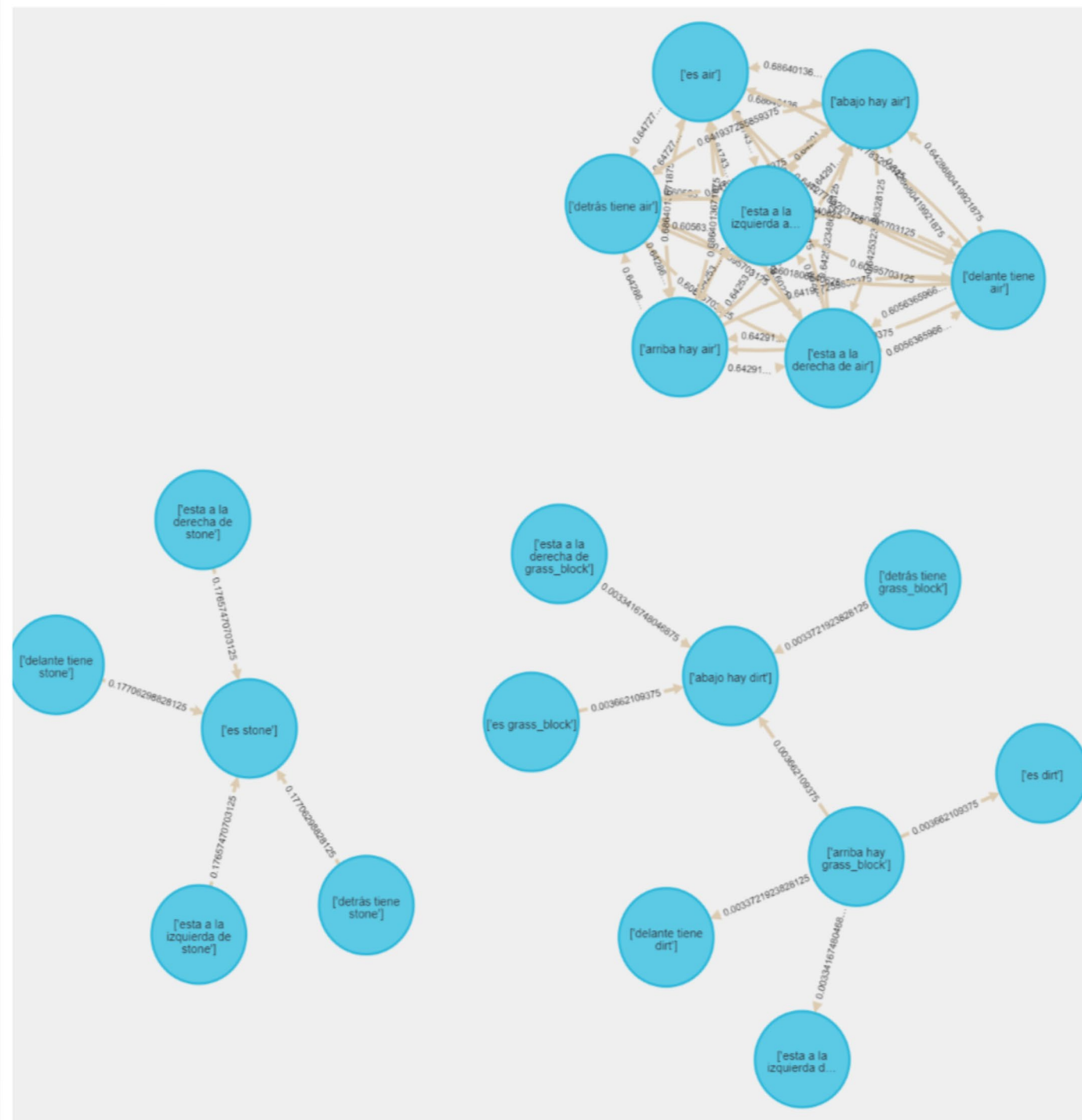
`tiempo de pre procesamiento` = 13 minutos y 30 segundos

`total de transacciones` = 65536

`soporte mínimo` =  $200/65536 = 0.0003$

`confianza mínima` = 90%

- En este grafo solo hay 3 patrones interpretables:
- Si el bloque es de tipo aire, está rodeado de otros bloques de tipo aire.
- Si el bloque es de tipo piedra, está rodeado de bloques de tipo piedra
- Si el bloque es de tierra, arriba hay bloques de pasto y abajo bloques de tierra



# PROBLEMAS QUE SURGIERON

Este proyecto tiene problemas importantes:

- **Tamaño de las ventanas :**

Definir **ventanas** tiene, a veces, un tiempo de procesamiento más alto comparado con el tiempo necesario para correr **Apriori** y tiene **complejidad temporal no lineal**.

- **Soporte mínimo:**

Con respecto al **soporte mínimo**, cómo el conjunto de datos está muy **desbalanceado**, si buscamos conseguir los patrones que son poco frecuentes, la forma de obtenerlos es bajando el *soporte mínimo* lo cual lleva a la generación de una gran cantidad de reglas donde muchas son **redundantes** o **pequeñas variaciones de aquellas reglas con mayor *soporte***.

- **Algoritmo P-DS para la poda de reglas:**

Si bien, el algoritmo devuelve un conjunto de reglas que podrían servir de resumen, tiende a estar formado por todas aquellas reglas con **un solo antecedente y consecuente**.

- **Conjunto de reglas no interesantes:**

No es sencillo **determinar si existen** más patrones que los clasificados como interesantes, en el conjunto de **reglas no interesantes** sobre todo porque su tamaño es grande.



# ALCANCE

Cosas que podrían implementarse en el proyecto:

- Un generador de representaciones gráficas con grafos automático.
- Implementación de algún método o algoritmo que no genere tantas reglas redundantes o que brinden información insignificante.
- Aplicar técnicas para trabajar con conjuntos de datos desbalanceados.
- La implementación de un **cloud model** para que permite expresar datos numéricos en lenguaje coloquial.
  - Por ejemplo, si se aplicara a coordenadas rectangulares de los bloques, podríamos generar reglas como por ejemplo:



[ hay muchos bloques de tipo piedra en lo profundo]

-->

[hay algunos bloques de tipo oro en lo profundo]



# CONCLUSIÓN

- Hay muchos patrones en el mapa de Minecraft, pero **han resultado difíciles de obtener** y el **análisis de los resultados no es sencillo** para la evaluación del rendimiento de las pruebas.
- Los patrones tienen que **ser deducidos por un experto**. Esto indica que es necesario buscar un mecanismo de minería de reglas que expresen mejor los patrones que buscamos.
- Debido a todas las complicaciones que se han presentado durante el proyecto **se concluye que:**
  - Se debe búsqueda de **otro tipo de acercamiento** para solucionar el problema empleando inteligencia artificial.
  - O buscar **otro algoritmo** para la generación de reglas que sea mas enfocado, aunque esto requeriría una exploración bibliográfica de algoritmos más complejos.



The image features a dark grey background with a vibrant, multi-colored wavy border at the top. The colors in the border include yellow, orange, red, green, and blue, creating a dynamic, flowing effect. Centered in the middle of the dark grey area is the word "GRACIAS" in a clean, white, sans-serif font.

GRACIAS