

---

## PROYECTO 1

---

**201602834 – Anderson Danilo García Alvizures**

### Resumen

Los TDA asocian un conjunto de datos u objetos, lo cual permite de manera dinámica manipularlos asociándolos a diferentes operaciones, como insertar, buscar, mostrar, etc. A esto le añadimos que por medio de POO permite el encapsulamiento de los datos, permitiendo manipularlos únicamente por la interfaz provista.

El TDA implementado es una Matriz Dispersa que consta de dos listas doblemente enlazadas unidas por un nodo raíz, estos a su vez se conectan con los nodos internos que no estén vacíos, esta matriz está representada por un tablero gráfico, que va adquiriendo valores con forme se ingresen las coordenadas. Las partidas también se pueden guardar en un archivo XML, donde están datos de la partida como las piezas colocadas y color de cada jugador, para posterior continuar con la partida si así se desea

### Abstract

*ADTs associate a set of data or objects, which allows dynamically manipulating them associating them to different operations, such as insert, search, show, etc. To this we add that by means of OOP it allows the encapsulation of the data, allowing it to be manipulated only through the interface provided.*

*The implemented ADT is a Sparse Matrix that consists of two doubly linked lists joined by a root node, these in turn relate to the internal nodes that are not empty, this matrix is represented by a graphic board, which acquires values with the form coordinates are entered. The games can also be saved in an XML file, where there are game data such as the pieces placed and the color of each player, to later continue with the game if desired.*

### Palabras clave

Lista

. Nodo

Matriz dispersa

### Keywords

List

Node

Sparse Matrices

## Introducción

Se implementó un juego en el lenguaje Python para dos jugadores que va por turnos, simulando las piezas de Tetris, se escoge un color y el tamaño del tablero donde se va a jugar, cuando inicia la partida se tiene un tiempo límite por turnos, en cada turno los jugadores pueden ingresar las coordenadas iniciales para las piezas que se muestran a un costado del tablero. Posterior a que finalice la partida se puedan visualizar los resultados por cada jugador.

## Desarrollo del tema

Una matriz dispersa permite trabajar con infinidad de filas y columnas, esto a su vez permite la optimización de memoria, ya que solo guarda las posiciones distintas a vacío o cero. El acceso a cada nodo interno se da por medio de las coordenadas de la lista cabecera.

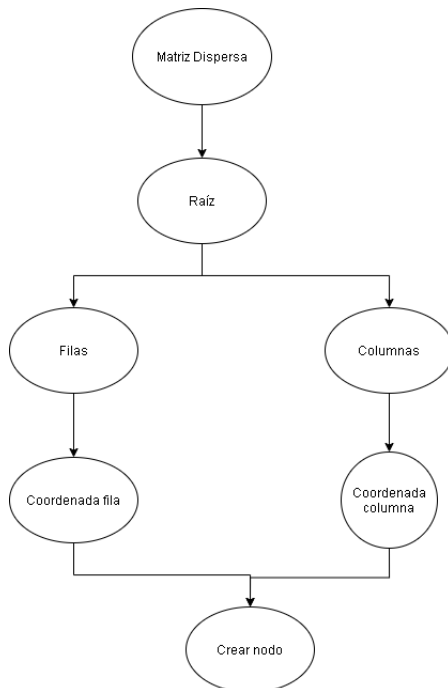


Figura 1. Bosquejo de la matriz dispersa

Fuente: elaboración propia.

La interfaz gráfica fue desarrollada por medio de la librería Tkinter, que por medio de POO muestra diferentes operaciones dentro del juego, las cuales son: Abrir una partida guardada con anterioridad, guardar la partida actual, iniciar una partida y por último ver los reportes de partidas anteriores.

### a. Abrir una partida

Se abre un archivo XML guardado con anterioridad

### b. Guardar partida

Si se desea terminar la partida posteriormente, esta guardaría, las piezas colocadas por cada jugador en el tablero

### c. Iniciar una partida

Al momento de iniciar la partida se selecciona el color por cada jugador, la cantidad de piezas y el tamaño del tablero a jugar para posterior ingresar las coordenadas por los turnos correspondientes

### d. Reportes

Se muestra un grafico de la matriz dispersa por medio de la librería Graphviz.

Se comparten imágenes sobre la interfaz de usuario

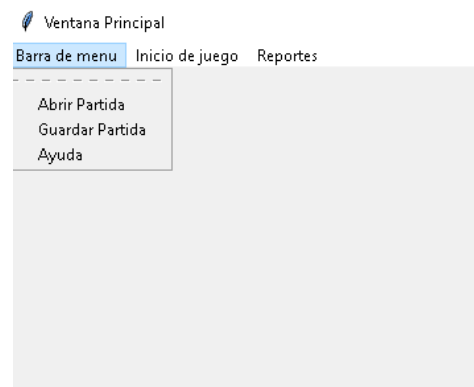


Figura 2. Pantalla inicial

Fuente: elaboración propia.

Esta pantalla permite crear la partida desde cero, seleccionando color y el tamaño del tablero, que permite crear tableros tanto nxn y mxn.

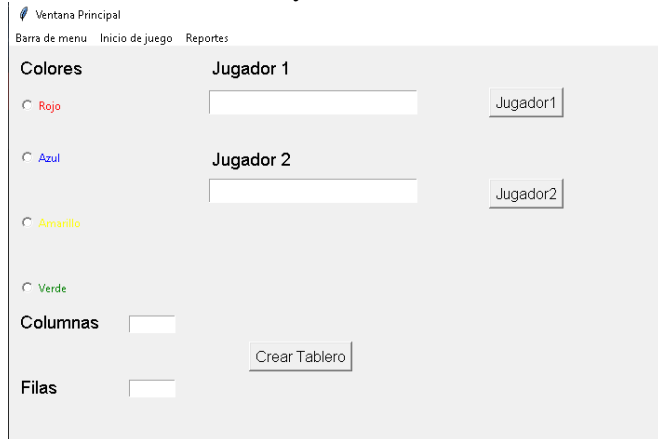


Figura 3. Pantalla de inicio de juego

Fuente: elaboración propia.

La pantalla de juego por medio del ingreso de coordenadas y el tablero se manipula la matriz dispersa, las coordenadas van desde cero hasta n-1, las piezas irán rotando con forme se coloquen en el tablero

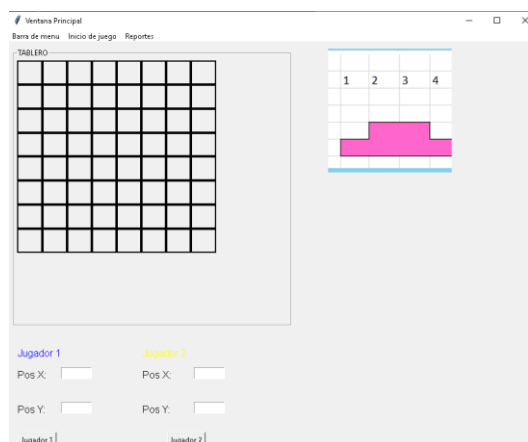
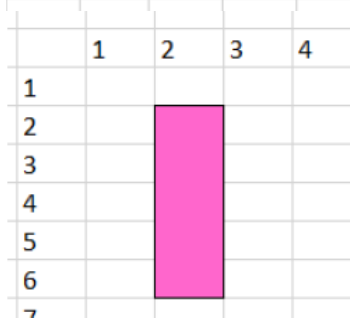
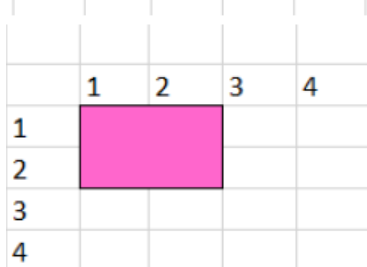
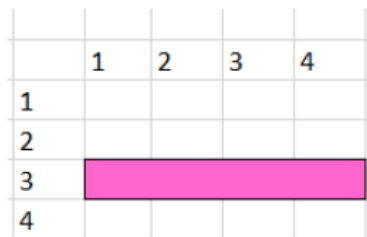
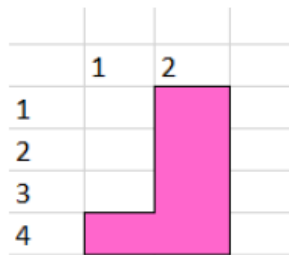
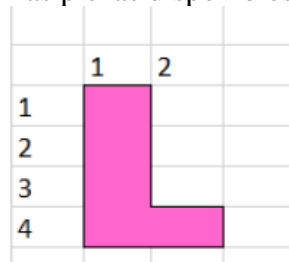


Figura 4. Pantalla de partida

Fuente: elaboración propia.

Las piezas disponibles son las siguientes:



	1	2	3	4
1				
2				
3				
4				

En la optimización de las partidas se tiene un inicio aleatorio de las piezas, luego los turnos de igual manera se van rotando.

### **Conclusiones**

La interfaz gráfica brindó un mejor manejo y visualización de la matriz dispersa, que permitió la optimización de memoria al solo guardar las piezas colocadas sobre el tablero y no el tablero completo con los espacios vacíos incluidos.

Esto permitió un mejor concepto tanto lógico como conceptual de los TDA, en el manejo amplio de datos.

La librería graphviz amplió la perspectiva sobre la matriz, mostrando la conexión de los nodos con sus respectivas cabeceras.

### **Referencias bibliográficas**

Concepto general de un TDA

<https://users.dcc.uchile.cl/~bebustos/apuntes/cc30a/TDA/>.

Fecha 19/06/2021