

---

## PROYECTO 1, OPTIMIZACION DEL GASTO DE MODIFICACIÓN DE PATRONES DE PISOS, IMPLEMENTANDO METODOS TDA Y GRAFICACION POR MEDIO DEL SOFTWARE GRAPHVIZ

---

202006629 – Wilber Steven Zúñiga Ruano

### Resumen

Se desarrollo un programa apto para leer archivos con estructura XML los cuales vienen compuestos de “N” número de pisos y cada piso a su vez contiene etiquetas tales como su dimensión que está determinado por una cantidad “R” de filas y la cantidad “C” de columnas, un precio “F” de volteo y un precio “S” de intercambio. Estos pisos a su vez contienen una cantidad “N” de patrones, cada patrón tiene un color específico en cada posición. El programa es capaz de obtener el precio óptimo para el cambio de un patrón a otro y representarlo de manera gráfica en cualquier momento. Para la solución de este proyecto se utilizó datos de tipo (TDA), como la implementación de una lista enlazada simple y la librería Graphviz para mostrar los patrones de forma gráfica, todo el proyecto se trabajó bajo el concepto de POO (programación orientada a objetos).

### Palabras clave

Lista  
Patrón  
TDA  
XML  
Óptimo

### Abstract

*A program was developed to read XML structure files which are composed of "N" number of floors and each floor in turn contains labels such as its dimension which is determined by an "R" number of rows and a "C" number of columns, a "F" flip price and an "S" exchange price. These apartments in turn contain an "N" number of patterns, each pattern having a specific color at each position. The program is able to obtain the most optimal price for the change from one pattern to another and represent it graphically at any time. For the solution of this project we used data type (TDA), as the implementation of a simple linked list and the Graphviz library to display the patterns graphically, the whole project was worked under the concept of OOP (object oriented programming).*

### Keywords

List  
Patterns  
TDA  
XML  
Optimum

## Introducción

La estructura de datos es una forma especial de organización de los datos en la computadora para que puedan conseguir un uso eficaz. Diferentes tipos de estructuras de datos son adecuadas para diferentes tipos de aplicaciones, algunas son altamente dedicados a tareas específicas.

La estructura de datos es una forma de procesamiento de grandes cantidades de datos de forma eficaz, usos como grandes bases de datos y servicios de Internet. Generalmente hablando, la estructura de datos es la clave del diseño de algoritmos eficientes. Algunos métodos de diseños formales y lenguaje de programación destacan la estructura de datos en lugar del algoritmo como factor principal de la aplicación y del diseño de software. En programación, una estructura de datos se puede declarar inicialmente ingresando una palabra reservada, luego ingrese un identificador y nombre de la estructura, luego un identificador para cada miembro de ella.

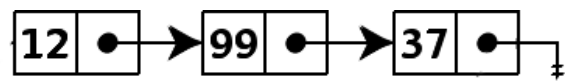
## Desarrollo del tema

Una lista enlazada es una estructura de datos dinámica. La cantidad de nodos en una lista no es fija y puede crecer y contraerse a demanda. Cualquier aplicación que tenga que tratar con un número desconocido de objetos necesitará usar una lista enlazada. Una desventaja de una lista vinculada frente a una matriz es que no permite el acceso directo a los elementos individuales. Si desea acceder a un artículo en particular, debe comenzar por la cabecera y seguir las referencias hasta que llegue a ese artículo.

Puede ser dividido en secciones estructurales que doten de coherencia al discurso.

### a. Lista enlazada simple:

Es una lista enlazada de nodos, donde cada nodo tiene un único campo de enlace. Una variable de referencia contiene una referencia al primer nodo, cada nodo (excepto el último) enlaza con el nodo siguiente, y el enlace del último nodo contiene None para indicar el final de la lista. Aunque normalmente a la variable de referencia se la suele llamar Cabecera.



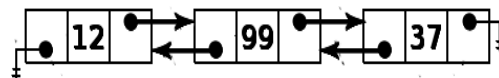
Una lista simplemente enlazada que contiene tres valores enteros

Figura 1. Lista simplemente enlazada.

Fuente: Xochipily, 2020, Wikimedia Commons.

### b. Lista doblemente enlazada:

Un tipo de lista enlazada más sofisticado es la lista doblemente enlazada o lista enlazada de dos vías. Cada nodo tiene dos enlaces: uno apunta al nodo anterior, o apunta al valor None si es el primer nodo; y otro que apunta al nodo siguiente, o apunta al valor None si es el último nodo. En algún lenguaje de muy bajo nivel, XOR-Linking ofrece una vía para implementar listas doblemente enlazadas, usando una sola palabra para ambos enlaces, aunque esta técnica no se suele utilizar.



Una lista doblemente enlazada que contiene tres valores enteros

Figura 2. Lista doblemente enlazada.

Fuente: Xochipily, 2020, Wikimedia Commons.

El problema que nos presenta es el siguiente:

La empresa “Pisos Artesanales, S.A.” ha construido un azulejo especial con el que puede crear pisos con distintos patrones. Cada piso consiste en una matriz de R filas y C columnas de azulejos cuadrados. Cada azulejo es reversible, un lado es blanco y el otro es negro, para poder crear patrones diversos. Además, la empresa garantiza que, para los pisos ya instalados, podrá cambiar el patrón original por un nuevo patrón que el cliente desee sin necesidad de comprar nuevos azulejos por el precio mínimo.

Para cambiar el patrón original del piso colocado por “Pisos Artesanales, S.A.”, se debe cumplir que el nuevo patrón corresponda a las dimensiones R x C del piso colocado y el robot será capaz de realizar una de las siguientes operaciones con cada uno de los azulejos que componen el piso: 1. Voltear un azulejo, cambiando el color visible de blanco a negro o viceversa, y 2. Intercambiar dos azulejos adyacentes (horizontal o verticalmente, pero no en diagonal), sin voltear ninguno.

Para la solución de este problema se usó el lenguaje Python, el cual importamos las librerías para leer el archivo, usamos la librería de “mini Dom” para la manipulación de archivos XML y poder pasarlas listas enlazadas que es el tema principal del proyecto. Para todo este proceso usamos programación orientado a objetos, el cual creamos varias clases.

Esto nos ayudó a crear una mejor estructura de datos, las clases fueron las siguientes:

- main
- Cuadrato
- Patron
- Piso
- nodoCuadrato
- nodoPatron
- nodoPiso
- listaCuadritos
- listaPatrones
- listaPisos

La clase principal es main, donde se manejan todos los métodos, se inicia manteniendo un menú en ejecución con las opciones que el programa ofrece, alelegir alguna entra a un submenú, a método de ejemplo, la opción número 1 que es cargar el archivo, pide la ruta del archivo XML posteriormente la pasa a una variable y usando la librería miniDom lo convierte en un archivo miniDom XML que es manipulable y de fácil acceso a todos sus atributos. Primero se obtiene todos los pisos que vienen en el archivo junto con sus patrones seguidamente se almacenan creando un objeto de tipo piso el cual contiene una lista simple enlazada de objetos tipo patrón que a su vez cada patrón contiene una lista enlazada simple de objetos cuadritos, luego se procede a correr el menú de pisos cargados selecciona el piso, luego el patrón a elegir y por último su nuevo patrón. Procede a recorrerse y hacer los intercambios y volteos necesarios para devolver el precio en una variable juntos con las instrucciones.

Los métodos más importantes dentro de la clase main:

1. Menú: El cual contiene todo el texto del menú y lo mantiene en ejecución, tiene las opciones de carga de archivo, mostrar pisos cargados y salir.
2. cargaArchivo: Este método anteriormente explicado es el encargado de recibir el archivoXML y obtener todos los atributos necesarios, extraerlos y pasarlos a las variables requeridas, y lleva todo el manejo de las listas enlazadas.

Tabla I.

### Variables principales

CATEGORÍA	CATEGORÍA
documentt	miniDom(XML)
listaPi	Lista enlazada de pisos
nombreFabrica	Nombre de la fábrica trabajada.

Fuente: elaboración propia.

### Conclusiones

- Las estructuras de datos son un medio para manejar grandes cantidades de datos de manera eficiente para usos tales como grandes bases de datos y servicios de indización de Internet.
- Un grafo es una estructura de datos conectada compuesta por nodos. Cada nodo contiene un valor y una o más referencias a otros nodos. Los grafos pueden utilizarse para representar redes, dado que los nodos pueden referenciarse entre ellos.
- Una ventaja del uso de listas enlazadas es la optimización de memoria.

### Referencias bibliográficas

- C. J. Date, (1991). *An introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.
- Narciso Martí, Estructura de datos y métodos algorítmicos, Madrid, McGraw Hill, 2003
- Román Martínez/Elda Quiriga, Estructura de datos: referencias prácticas, México, Thomson Learning, 2001.

### Anexos

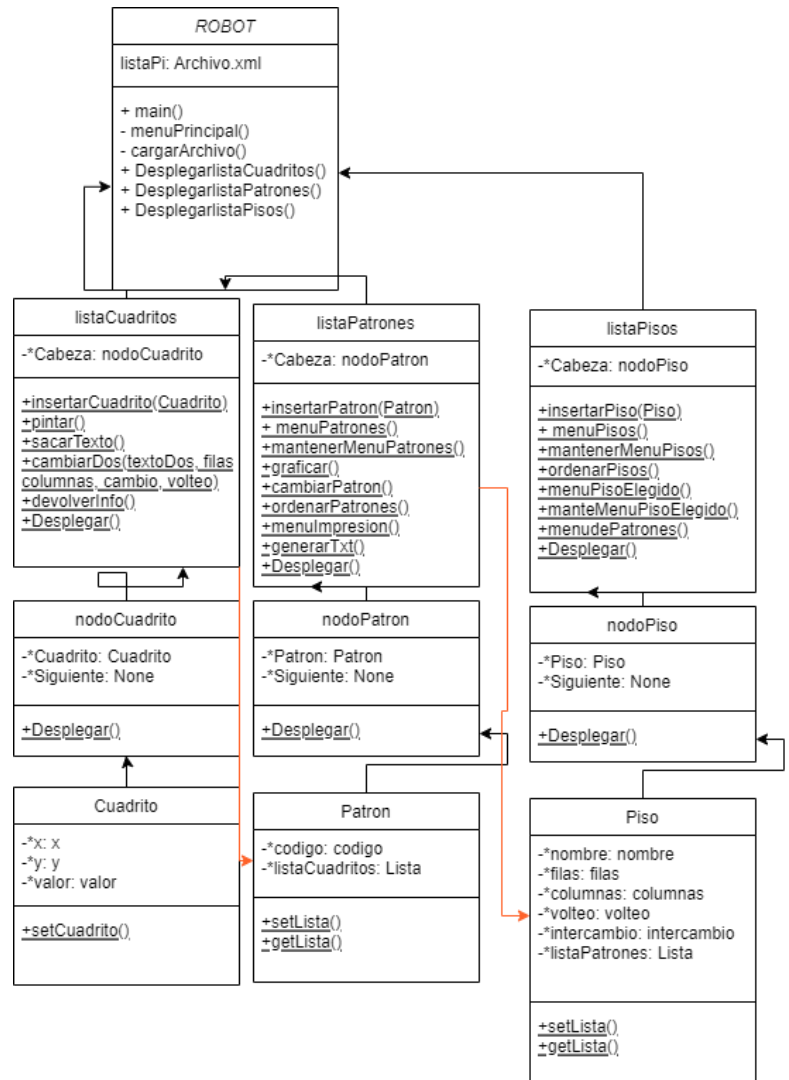


Diagrama de clases del proyecto.

Fuente: Elaboración propia.