|  |
| --- |
| **DOCUMENTACIÓN** |
| **201700900 – Gueslim Jhoans Fernandez Rivera** |

**Resumen**

Se desarrolló un programa para controlar un robot que realiza operaciones de Voltear un azulejo, cambiando el color visible de blanco a negro o viceversa, e Intercambiar dos azulejos adyacentes (horizontal o verticalmente, pero no en diagonal), sin voltear ninguno. Cada operación resulta en un costo que el robot realice cada una de las operaciones antes mencionadas, de tal manera que, realizar una operación de volteo de azulejo cuesta F Quetzales, mientras que realizar una operación de intercambio de azulejos cuesta S Quetzales.

Por lo que este programa garantiza que, al modificar un patrón en un piso existente, el costo de hacer esta modificación sea el mínimo posible para optimizar el uso del robot especializado adquirido para este fin.

**Palabras clave**

Azulejo, costos, optimización, Python, TDA, POO

***Abstract***

*A program was developed to control a robot that performs operations of Flipping a tile, changing the visible color from white to black or vice versa, and swapping two adjacent tiles (horizontally or vertically, but not diagonally), without flipping any of them. Each operation results in a cost for the robot to perform each of the mentioned operations, such that, executing a tile flip operation costs F Quetzales, while performing a tile swap operation costs S Quetzales.*

*So, this program ensures that when modifying a pattern on an existing floor, the cost of making this modification is the minimum possible to optimize the use of the specialized robot acquired for this purpose.*

***Keywords***

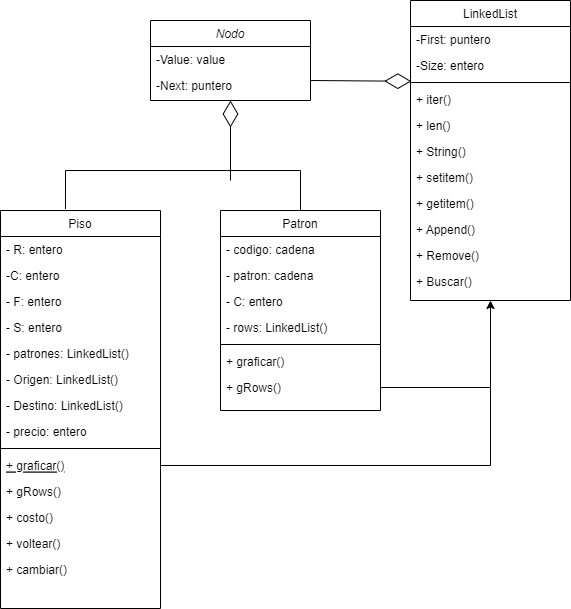
*Tile, cost, cost, optimization, Python, TDA, POO*

**Carga de XML**

Todos los pisos y patrones vienen almacenados en un archivo de formato XML, por lo que el primer paso para que el programa funcione es cargar y almacenar el archivo.

Para esto se utilizó el módulo xml.etree.ElementTree, con el que se iterando para raíz del piso en que se fueron tomando los datos, de nombre, R, C, F, S, así como cada patrón y su respectivo código.

**Lista Enlazada**

Lo más esencial y el núcleo del funcionamiento del programa es la creación de un tipo de dato abstracto para el almacenamiento y el procesamiento de la información obtenido del archivo XML. Para ello se eligió crear una Lista Enlazada Simple con POO. Esta lista enlazada consta de una clase Nodo, y una clase LinkedList como se aprecia en la figura 1. 

*Figura 1.* Diagrama de clases general.

Fuente: elaboración propia, o citar al autor, año y página.

**Piso**

La clase Piso consta de todas las etiquetas del archivo XML como atributos. Esta clase almacena el nombre de piso, el tamaño de filas y columnas del piso, así como el costo de voltear e intercambiar los azulejos. La lista de patrones correspondientes a in piso se van guardando en una lista enlazada, siendo cada nodo un objeto Patrón.

Esta clase es muy importante debido a que almacena toda la información de cada piso, así como grafica la información almacenada para mostrar al usuario. Y sobre todo va almacenando el piso origen y destino elegidos por el usuario y desarrolla el procedimiento para la conversión de patrones mostrando paso a paso el procedimiento y el costo en cada operación.

**Patrón**

La clase Patrón almacena el código y el patrón en cadena, aquí se procesa esa cadena y se convierte en una lista de listas de filas x columnas. Así como graficar los patrones en consola para el usuario.

**Uso**

Al correr el programa aparecerá el menú de las distintas opciones enumeradas, por lo cual debemos elegir una opción presionando uno de los números.

Texto

Descripción generada automáticamente

1. Cargar XML

Al elegir la primera opción, nos pedirá la ruta del archivo XML, luego de introducirla se almacenarán los datos en el sistema y nos regresará al menú principal.

Texto

Descripción generada automáticamente

1. Mostrar Patrón.

Al elegir la segunda opción, si aún no se han elegido el piso y patrón origen, se desplegarán todos los pisos y sus patrones disponibles en el sistema.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Por lo que se deberá introducir el nombre del piso, o el número de piso según es listado, se desplegarán los patrones disponibles de ese piso, y el sistema le pedirá que introduzca del código de patrón o el numero según es listado.

Texto

Descripción generada automáticamente

Luego de seleccionar el Piso y patrón iniciales, regresará al menú principal. Al seleccionar la opción 2 nuevamente luego de elegir el piso y patrón inicial se verá la opción elegida, y el sistema le pedirá que confirme si desea cambiarlos presionando S, dando nuevamente las opciones iniciales.

1. Seleccionar nuevo Patrón.

En esta opción se desplegará el piso actual elegido anteriormente, y el sistema le pedirá que ingrese el código del patrón destino o el numero según listado.

Luego de esto se desplegará el procedimiento paso a paso del procedimiento para transformar del patrón actual al nuevo. Y por último de desplegará una imagen del grafico del patrón original y el nuevo.

1. Ordenar pisos Cargados

Al seleccionar esta opción la lista de pisos y patrones en el sistema de ordenaran alfabéticamente según el nombre del piso y el código del patrón. Desplegando todos los datos ordenados.

**Referencias bibliográficas**

Charles Severance (2015). *Python para Informaticos.* CreateSpace Independent Publishing Platform.

**ANEXOS EN EL FOLDER**