

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Introducción a la Programación y Computación 2

Inga. Claudia Liceth Rojas Morales
Ing. Marlon Antonio Pérez Türk
Ing. Byron Rodolfo Zepeta Arevalo
Ing. Jose Manuel Ruiz Juarez
Ing. Edwin Estuardo Zapeta Gómez

Tutores de curso:
Alexander Raymundo Ixvalan Pacheco
Jackeline Benitez
Javier Lima
Viany Paola Juárez
José Carlos Estrada



PROYECTO 1

OBJETIVO GENERAL

Se busca que el estudiante sea capaz de dar una solución al problema que se le plantea, mediante la lógica y conocimientos que se le han impartido durante la clase y que han sido aplicados en el laboratorio.

OBJETIVOS ESPECÍFICOS

- Que el estudiante sea capaz de aplicar abstracción a un problema dado.
- Implementar una solución utilizando el lenguaje de programación Python.
- Utilizar estructuras de programación secuenciales, cíclicas y condicionales
- Que el estudiante aprenda a generar reportes con la herramienta Graphviz.
- Que el estudiante sea capaz de manipular archivos XML.
- Que el estudiante utilice los conceptos de TDA y aplicarlos a memoria dinámica.

DESCRIPCIÓN DEL PROBLEMA

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.

La empresa ha comprado un robot especializado capaz de colocar los pisos de dimensiones $R \times C$ con cualquier patrón, combinando azulejos del lado blanco con azulejos del lado negro. Además, la empresa garantiza que cada piso colocado podrá cambiar el patrón, a cualquier patrón deseado siguiendo las siguientes reglas:

Para cambiar el patrón original del piso colocado por “Pisos Artesanales, S.A.”, se debe cumplir que el nuevo patrón corresponda a las dimensiones $R \times C$ 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.

A la empresa “Pisos Artesanales, S.A.” le 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.

Debido a que la empresa “Pisos Artesanales, S.A.” posee una cuota fija pactada por contrato con los clientes que han colocado los pisos que ofrecen, se le ha solicitado realizar un programa que garantice 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.

ENTRADA

La entrada consiste en un archivo XML con los patrones de los pisos que la empresa vende.

```
<?xml version="1.0"?>
<pisosArtesanales>
  <piso nombre="nombrePiso">
    <R> valorEnteroPositivoMayorOIgualQueUno </R>
    <C> valorEnteroPositivoMayorOIgualQueUno </C>
    <F> valorEnteroPositivoMayorOIgualQueUno </F>
    <S> valorEnteroPositivoMayorOIgualQueUno </S>
    <patrones>
      <patron codigo="codigoPatron">
        RxC_Letras_WparaBlanco_BparaNegro
      </patron>
      ...
    </patrones>
```

```
</piso>
...
</pisosArtesanales>
```

Donde las etiquetas “R”, “C”, “F” y “S” corresponden a valores enteros positivos mayores o iguales a 1 y representan el número de filas y columnas del piso, y el costo en Quetzales para voltear o intercambiar los pisos respectivamente. La etiqueta “patrones” corresponde a una serie de “n” patrones que la máquina puede crear y cada etiqueta “patron” contendrá un string de R x C caracteres “W” y “B” en cualquier orden y distribución, siendo “W” la representación de un azulejo blanco y “B” la representación de un azulejo negro.

PROGRAMA A DESARROLLAR

Deberá ser capaz de cargar el archivo de entrada con el formato XML descrito en la sección “Entrada” y permitir las siguientes operaciones al seleccionar un piso y patrón específico:

1. Mostrar gráficamente el patrón.
2. Seleccionar un nuevo código de patrón y mostrar lo siguiente:
 - a. El costo mínimo para realizar el cambio hacia el nuevo patrón
 - b. Las instrucciones paso a paso para que el robot construya el nuevo patrón al costo mínimo calculado. En este punto el programa deberá permitir seleccionar si las instrucciones se generan en consola o en un archivo.
 - c. Mostrar gráficamente el nuevo patrón

Además, el programa debe tener una opción para mostrar todos los pisos cargados al sistema en orden alfabético según el nombre asignado al piso y los códigos de patrones disponibles en orden alfabético del código asignado al patrón.

El programa deberá ser fácil de utilizar (amigable) y debe mostrar de forma precisa los pasos para realizar el cambio de patrón.

Se deberá utilizar la herramienta Graphviz para mostrar gráficamente los patrones en las opciones que se destinen para esta finalidad.

LÍMITES

T, R, C, F y S son enteros positivos mayores o iguales a 1

El programa debe ser capaz de manejar “n” pisos diferentes con “m” patrones diferentes, siendo m y n enteros positivos mayores o iguales a 1.

EJEMPLO DE ARCHIVO DE ENTRADA

```
<?xml version="1.0"?>
<pisosArtesanales>
  <piso nombre="ejemplo01">
    <R> 2 </R>
    <C> 4 </C>
    <F> 1 </F>
    <S> 1 </S>
    <patrones>
      <patron codigo="cod11">
        WBWBWWWB
```

```

    </patron>
    <patron codigo="cod12">
      BWBWWWWW
    </patron>
  </patrones>
</piso>
<piso nombre="ejemplo02">
  <R> 3 </R>
  <C> 3 </C>
  <F> 1 </F>
  <S> 1 </S>
  <patrones>
    <patron codigo="cod21">
      WBBBWBWWW
    </patron>
    <patron codigo="cod22">
      WWWWBWWWB
    </patron>
  </patrones>
</piso>
<piso nombre="ejemplo03">
  <R> 1 </R>
  <C> 5 </C>
  <F> 1000 </F>
  <S> 1 </S>
  <patrones>
    <patron codigo="cod31">
      WBBBB
    </patron>
    <patron codigo="cod32">
      BBBWW
    </patron>
  </patrones>
</piso>
</pisosArtesanales>

```

EJEMPLO DE OPERACIONES

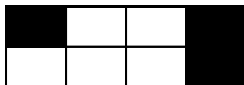
Ejemplo # 1: Para el piso “ejemplo01” convertir el patrón origen código “cod11” al patrón destino código “cod12”.

Debe considerar que hay 5 azulejos que tienen un color diferente entre el patrón original y el patrón deseado para el piso. Dado que cada operación puede cambiar como máximo 2 azulejos, se necesitan al menos 3 operaciones, que cuestan 3 Quetzales. Una forma de hacerlo utilizando exactamente 3 Quetzales es la siguiente:

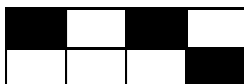
Patrón inicial:



1. Intercambie los dos azulejos más a la izquierda de la fila de arriba. Costo Q. 1.00



2. Intercambie los dos azulejos más a la derecha en la fila de arriba. Costo Q. 1.00



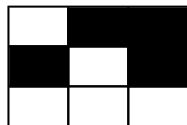
3. Voltee el azulejo de la esquina derecha de la fila de abajo. Costo 1.00. Esto nos lleva al patrón final.



Por lo tanto, se necesitan 3 Quetzales como mínimo para transformar el patrón inicial al patrón final.

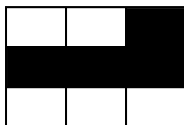
Ejemplo # 2: Para el piso “ejemplo02” convertir el patrón origen código “cod21” al patrón destino código “cod22”.

En este caso debe considerar que hay 6 azulejos que necesitan cambiar. Sin embargo, debido a que solamente los intercambios pueden cambiar dos azulejos a la vez, se puede resolver con 3 operaciones de intercambio, pero no hay manera de obtener el patrón final de los 6 azulejos utilizando intercambios, así que necesitamos por lo menos 4 operaciones. Una forma de hacerlo utilizando exactamente 4 Quetzales es la siguiente:

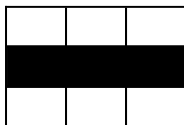


Patrón inicial:

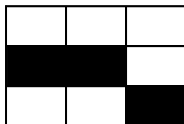
1. Intercambie los dos azulejos superiores en la columna central. Costo Q. 1.00.



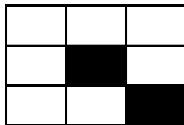
2. Voltee el azulejo de la esquina superior derecha. Costo Q. 1.00



3. Intercambie los dos azulejos inferiores de la 3ª. Columna. Costo Q. 1.00



4. Voltee el 1er. Azulejo de la fila central. Esto nos lleva al patrón final. Costo Q. 1.00



Por lo tanto, se necesitan 4 Quetzales como mínimo para transformar el patrón inicial al patrón final.

Ejemplo #3: Para el piso “ejemplo03” convertir el patrón origen código “cod31” al patrón destino código “cod32”.

En este caso hay 3 azulejos que necesitan cambiar. Sin embargo, debido al alto costo de voltear azulejos, se debe considerar evitar su uso. Necesitamos por lo menos uno debido a que nuestro patrón final tiene más blanco que el patrón original, por lo que la forma óptima de convertir el patrón sería la siguiente:

Patrón inicial:

1. Intercambiar los azulejos más a la izquierda. Costo Q. 1.00



2. Voltear el azulejo más a la derecha. Costo Q. 1000.00



3. Intercambiar el 2do y el 3er azulejos desde la izquierda. Costo Q. 1.00



4. Intercambiar el 3er y el 4to azulejo desde la izquierda. Costo Q. 1.00. Esto nos lleva al patrón final



Por lo tanto, se necesitan 1003 Quetzales como mínimo para transformar el patrón inicial al patrón final.

CONSIDERACIONES

Se deberán implementar listas enlazadas para resolver el problema, creadas por el estudiante, creando una clase Nodo y una o varias clases lista enlazada, de tal manera que al recorrer la lista se pueda validar la existencia de los pisos y sus respectivos patrones.

El cómo se utilicen las estructuras anteriormente descritas para guardar los datos del archivo de entrada queda a discreción del alumno. Tomar en cuenta que al ingresar el archivo de entrada todos los pisos dentro del archivo XML deben de ser cargados a las listas para luego tener la habilidad de elegir cuál piso se quiere procesar.

Debe utilizarse versionamiento para el desarrollo del proyecto. Se utilizará la plataforma **Github** en la cual se debe crear un repositorio en el que se gestionará el proyecto. Se deben realizar 4 releases o versiones del proyecto (se recomienda realizar una por semana del tiempo disponible). Se deberá agregar a sus respectivos auxiliares como colaboradores del repositorio. El último release será el release final y se deberá de realizar antes de entregar el proyecto en la fecha estipulada.

DOCUMENTACIÓN

Para que el proyecto sea calificado, el estudiante deberá entregar la documentación utilizando el formato de ensayo definido para el curso. En el caso del proyecto, el ensayo puede tener un mínimo de 4 y un máximo de 7 páginas de contenido, este máximo no incluye los apéndices o anexos donde se pueden mostrar modelos y diseños utilizados para construir la solución. Es obligatorio incluir el diagrama de clases que modela la solución de software presentada por el estudiante.

RESTRICCIONES

- Solo se permitirá la utilización de los IDEs discutidos en el laboratorio.
- Uso obligatorio de programación orientada a objetos (POO) desarrollada por completo por el estudiante. De no cumplir con la restricción, no se tendrá derecho a calificación.
- El nombre del repositorio debe de ser **IPC2_Proyecto1_#Carnet**.
- El estudiante debe entregar la documentación solicitada para poder optar a la calificación.
- Los archivos de entrada no podrán modificarse.
- Los archivos de salida deben llevar la estructura mostrada en el enunciado obligatoriamente.
- Deben existir 4 releases uno por cada semana, de esta manera se corrobora el avance continuo del proyecto.
- Se calificará de los cambios realizados en el cuarto release. Los cambios realizados después de ese release no se tomarán en cuenta.
- Cualquier caso de copia parcial o total tendrá una nota de 0 y será reportada a escuela.
- Para dudas concernientes al proyecto se utilizarán los foros en UEDI de manera que todos los estudiantes puedan ver las preguntas y las posteriores respuestas.
- **NO HABRÁ PRÓRROGA.**

ENTREGA

- La entrega será el **04 de marzo** a más tardar a las 11:59 pm.
- La entrega será por medio de la UEDI.
- La documentación debe estar subida en el repositorio en una carpeta separada.
- Para entregar el proyecto en UEDI se deberá subir un archivo de texto con el link del repositorio.