

**Astrid Edith Hernández González, Henry Adolfo Gálvez
César Dionicio Sazo Mayen, Oscar Alejandro Rodríguez Calderón
Edwar Everaldo Zacarías, Javier Estuardo Lima Abrego
Monica Raquel Calderón Muñoz, Marco Antonio López Grajeda**

- 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 Agencia Guatemalteca de Investigación Espacial (AGIE) ha diseñado un nuevo robot de exploración, llamado r2e2, que tiene la habilidad de explorar nuevos terrenos, este nuevo robot puede moverse en todo tipo de terrenos, sólo necesita más combustible para moverse en terrenos accidentados, y menos combustible en terrenos planos. El único problema con r2e2 es que solo puede moverse ortogonalmente, es decir, únicamente puede moverse en dirección Norte, Este, Sur y Oeste de su posición.

R2e2 ha sido diseñado para comunicarse con el satélite Quetzal01 para tener una imagen del terreno que va a explorar, así que debe seleccionar la mejor ruta para llegar al punto de destino, r2e2 siempre debe seleccionar el camino que necesita el mínimo uso de combustible para completar su exploración, sin embargo, los científicos que están experimentando con r2e2 necesitan un programa que calcule el camino que utilice la mínima cantidad de combustible. La máxima cantidad de combustible que r2e2 puede almacenar son 9,999 unidades.

El terreno que r2e2 recibe desde el satélite Quetzal01 está dividido en una cuadrícula, donde cada celda de dicha cuadrícula tiene asignada la cantidad de combustible que r2e2 necesitará para pasar a través de esa celda, r2e2 también recibe las coordenadas de inicio y final del terreno de exploración, la imagen 1 muestra un ejemplo de la información recibida desde el satélite Quetzal01.

Fila / Columna	1	2	3	4	5
1	1	1	5	3	2
2	4	1	4	2	6
3	3	1	1	3	3
4	5	2	3	1	2
5	2	1	1	1	1
Coordenada inicial (1,1) – Coordenada final (5,5)					

Figura 1 – Terreno de exploración recibido desde el satélite Quetzal01

La Universidad de San Carlos de Guatemala cooperará desarrollando el algoritmo que r2e2 necesita para identificar el camino con menor consumo de combustible para moverse por el área de exploración y usted ha sido seleccionado para desarrollar esta tarea. La figura 2 ejemplifica el efecto del algoritmo sobre la información recibida por el satélite Quetzal01.

Fila / Columna	1	2	3	4	5
1	1	1	5	3	2
2	4	1	4	2	6
3	3	1	1	3	3
4	5	2	3	1	2
5	2	1	1	1	1
Coordenada inicial (1,1) – Coordenada final (5,5) Combustible necesario: 10 unidades					

Figura 2 – Mejor ruta para explorar el terreno recibido desde el satélite Quetzal01

ENTRADA

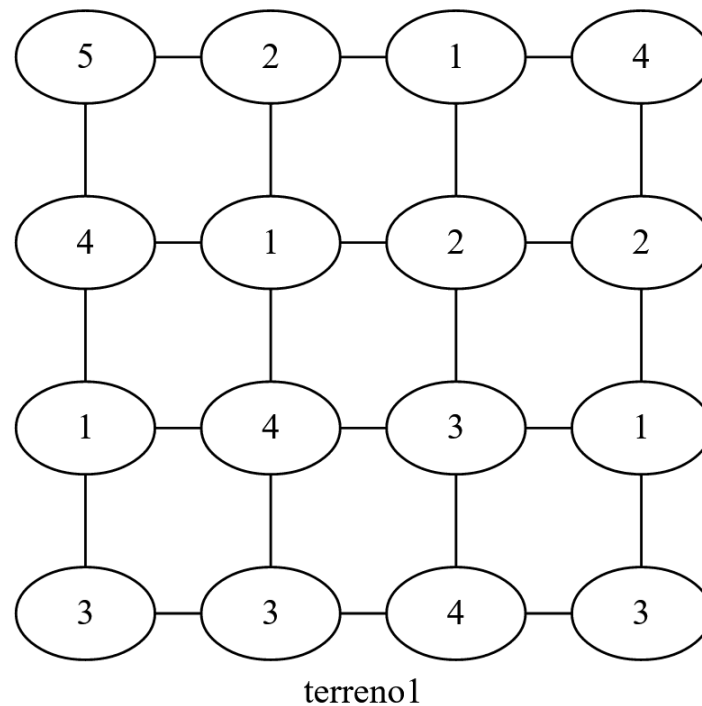
La entrada va a consistir en una serie de terrenos de exploración a evaluar que estarán contenidos en una estructura XML, cada terreno de exploración tendrá un nombre. Esta estructura XML contendrá el número de filas y columnas que la cuadrícula con el terreno de exploración contendrá. Se debe considerar que $0 < \text{filas} \leq 100$, y $0 < \text{columnas} \leq 100$. A continuación, la estructura XML contendrá la cuadrícula con las unidades de combustible necesarias para cada celda y finalmente, presentará la coordenada de inicio y la coordenada final. Esta estructura se procesará en la aplicación por medio de un archivo XML con la información.

SALIDA

La salida va a consistir en una serie de caminos de exploración para utilizar la cantidad mínima de combustible y también deberá indicar la cantidad de combustible que dicho camino consumirá. Los datos de salida se mostrarán en un archivo XML que la aplicación debe generar con la estructura descrita posteriormente.

REPORTES

Se deberá utilizar la herramienta Graphviz para crear un grafo que muestre de manera gráfica la estructura del archivo de entrada mostrando los valores de consumo de combustible del mapa. Se debe mostrar de una manera similar a la siguiente. En este ejemplo, “terreno1” será el nombre del mapa mientras en los nodos se mostrarán los datos de cada coordenada. La figura es una sugerencia, queda a discreción del alumno como graficar el terreno siempre cuidando de que los datos sean mostrados correctamente.



Al momento de procesar el terreno seleccionado también se debe de mostrar un reporte, en este caso en consola que muestre un dibujo básico de una representación del archivo de salida. El formato sugerido es el siguiente:

```
| 1 | o | o | o | o |
| 1 | 1 | 1 | o | o |
| o | o | 1 | o | o |
| o | o | 1 | o | o |
| o | o | 1 | 1 | 1 |
```

En este caso el símbolo “1” representa las coordenadas de la ruta seleccionada y el símbolo “o” coordenadas en donde no pasará el robot. Se toma en cuenta un mapa de dimensiones 5x5 en donde la coordenada de inicio es (0,0) y la coordenada final es (5,5).

Además de esto se deberá mostrar en consola el consumo total de combustible que la ruta consumirá, todo esto indicando de manera clara el nombre del terreno de exploración original sobre el cual se realizan los cálculos. La generación del archivo XML se ejecutará de manera separada y ocurrirá al seleccionar la opción del menú principal como se muestra más adelante.

ARCHIVOS DE ENTRADA Y SALIDA

Los archivos de entrada y salida consistirán en archivos con extensión y estructura xml en el cual se limitará a utilizar únicamente las etiquetas:

- **terrenos:** Es la etiqueta raíz del documento. Indica el comienzo de lectura del archivo.
- **terreno:** Esta etiqueta será la que indica que un nuevo terreno será creado para su respectivo análisis y únicamente puede estar dentro de la etiqueta **terrenos**, conteniendo un atributo **nombre** el cual especifica el nombre del terreno. Puede tener las siguientes etiquetas internas:
 - **posicioninicio:** identifica la etiqueta que contiene la información de las coordenadas del inicio de la ruta.
 - **x:** etiqueta interna que describe la coordenada en x.
 - **y:** etiqueta interna que describe la coordenada en y.
 - **posicion fin:** identifica la etiqueta que contiene la información de las coordenadas del final de la ruta.
 - **x:** etiqueta interna que describe la coordenada en x.
 - **y:** etiqueta interna que describe la coordenada en y.
- **posicion:** etiqueta utilizada para definir el valor del gasto de combustible por coordenada del terreno. Contiene dos atributos de etiqueta:
 - **x:** define la posición en x de la coordenada.
 - **y:** define la posición en y de la coordenada.

Un archivo XML puede incluir 'n' terrenos de exploración. Al momento de procesar el archivo de entrada todos los terrenos individuales deben de ser guardados en memoria y poder ser graficados en forma de reporte.

Ejemplo de Entrada:

```
<terrenos>
  <terreno nombre="terreno1">
    <posicioninicio>
      <x>1</x>
      <y>1</y>
    </posicioninicio>
    <posicionfin>
      <x>5</x>
      <y>5</y>
    </posicionfin>
    <posicion x="1" y="1">1</posicion>
    <posicion x="1" y="2">1</posicion>
    <posicion x="1" y="3">5</posicion>
    <posicion x="1" y="4">3</posicion>
    <posicion x="1" y="5">2</posicion>
    <posicion x="2" y="1">4</posicion>
    <posicion x="2" y="2">1</posicion>
    <posicion x="2" y="3">4</posicion>
```

```

<posicion x="2" y="4">2</posicion>
<posicion x="2" y="5">6</posicion>
<posicion x="3" y="1">3</posicion>
<posicion x="3" y="2">1</posicion>
<posicion x="3" y="3">1</posicion>
<posicion x="3" y="4">3</posicion>
<posicion x="3" y="5">3</posicion>
<posicion x="4" y="1">5</posicion>
<posicion x="4" y="2">2</posicion>
<posicion x="4" y="3">3</posicion>
<posicion x="4" y="4">1</posicion>
<posicion x="4" y="5">2</posicion>
<posicion x="5" y="1">2</posicion>
<posicion x="5" y="2">1</posicion>
<posicion x="5" y="3">1</posicion>
<posicion x="5" y="4">1</posicion>
<posicion x="5" y="5">1</posicion>
<terreno>
<terreno nombre="terreno2">
  <posicioninicio>
    <x>2</x>
    <y>2</y>
  </posicioninicio>
  <posicionfin>
    <x>6</x>
    <y>1</y>
  </posicionfin>
  <posicion x="1" y="1">1</posicion>
  <posicion x="1" y="2">1</posicion>
  <posicion x="1" y="3">1</posicion>
  <posicion x="1" y="4">3</posicion>
  <posicion x="2" y="1">5</posicion>
  <posicion x="2" y="2">1</posicion>
  <posicion x="2" y="3">2</posicion>
  <posicion x="2" y="4">3</posicion>
  <posicion x="3" y="1">6</posicion>
  <posicion x="3" y="2">9</posicion>
  <posicion x="3" y="3">3</posicion>
  <posicion x="3" y="4">4</posicion>
  <posicion x="4" y="1">8</posicion>
  <posicion x="4" y="2">4</posicion>
  <posicion x="4" y="3">4</posicion>
  <posicion x="4" y="4">1</posicion>
  <posicion x="5" y="1">7</posicion>
  <posicion x="5" y="2">3</posicion>
  <posicion x="5" y="3">1</posicion>
  <posicion x="5" y="4">5</posicion>
  <posicion x="6" y="1">1</posicion>
  <posicion x="6" y="2">2</posicion>
  <posicion x="6" y="3">2</posicion>
  <posicion x="6" y="4">2</posicion>
  <posicion x="7" y="1">3</posicion>
  <posicion x="7" y="2">1</posicion>
  <posicion x="7" y="3">1</posicion>
  <posicion x="7" y="4">1</posicion>
</terreno>
<terreno nombre="terreno3">
<terreno>
.....
</terrenos>

```

Ejemplo de Salida:

```
<terreno nombre="terreno1">
  <posicioninicio>
    <x>1</x>
    <y>1</y>
  </posicioninicio>
  <posicionfin>
    <x>5</x>
    <y>5</y>
  </posicionfin>
  <combustible>7</combustible>
  <posicion x="1" y="1">1</posicion>
  <posicion x="1" y="2">1</posicion>
  <posicion x="3" y="2">1</posicion>
  <posicion x="5" y="2">1</posicion>
  <posicion x="5" y="3">1</posicion>
  <posicion x="5" y="4">1</posicion>
  <posicion x="5" y="5">1</posicion>
</terreno>
```

La aplicación deberá contar con un menú en consola, con las siguientes opciones:

Menú principal:

1. Cargar archivo
2. Procesar archivo
3. Escribir archivo salida
4. Mostrar datos del estudiante
5. Generar gráfica
6. Salida

1. **Cargar Archivo:** Esta Opción solicitará la ruta del archivo a cargar.

```
Opcion Cargar Archivo:
Ingrese la ruta del archivo:
```

2. **Procesar Terreno:** Esta opción será la encargada de procesar un terreno de los que anteriormente fueron cargados en memoria, se debe solicitar el nombre del terreno a procesar y durante el proceso se deben ir mostrando mensajes al usuario para tener el conocimiento de lo que está pasando en el sistema y luego mostrar la ruta elegida por el algoritmo como la más óptima con el formato que se muestra en la sección de Reportes.

- Calculando la mejor ruta
- Calculando cantidad de combustible

3. **Escribir Archivo de salida:** Esta opción será la encargada de escribir el archivo con la salida específica.

```
Escribir una ruta especifica: C:/escritorio/ipc2/final.xml  
Se escribio el archivo satisfactorio
```

4. **Mostrar datos del estudiante:** Mostrar los datos del estudiante, carné, nombre, curso, carrera y semestre.

```
> Juan Pablo Osuna de Leon  
> 201503911  
> Introduccion a la Programación y computación 2 seccion "A"  
> Ingenieria en Ciencias y Sistemas  
> 4to Semestre
```

5. **Generar gráfica:** En esta opción se generará la gráfica de los terrenos del archivo de entrada de la forma descrita anteriormente en la sección de reportes. Debe mostrar una lista de todos los terrenos cargados del archivo de entrada y debe poder dar la opción para elegir cual se necesita graficar en ese momento.

CONSIDERACIONES

Se deberá implementar una lista simplemente enlazada, creada por el estudiante, creando una clase Nodo y una clase lista, de tal manera que al recorrer la lista se pueda validar la existencia de los nombres de los mapas y en base en esta lista poder realizar un reporte gráfico de la misma. El cómo se utilice la estructura anteriormente descrita 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 terrenos dentro del archivo XML deben de ser cargados a las listas para luego tener la habilidad de elegir cuál terreno 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.

Formato del ensayo: <https://drive.google.com/file/d/1jt3wtyTYPD-Gi27W9cJPaoKiLMlwIVVY/view?usp=sharing>

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 **27 de agosto** 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.