

1er. Semestre 2023

**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 Zepeda Arevalo**  
**Ing. Jose Manuel Ruiz Juarez**  
**Ing. Edwin Estuardo Zapeta Gómez**

**Tutores de curso:**  
**Diego Pérez Alvarez**  
**Marvin Daniel Rodríguez**  
**Oscar Velásquez Leon**  
**Daniel Arturo Alfaro**  
**Sergio Felipe Zapeta**



## **PROYECTO 1**

### **OBJETIVO GENERAL**

Solucionar problemas planteados mediante el uso de lógica, herramientas y conocimientos que se han impartido durante la clase y que han sido aplicados en el laboratorio.

### **OBJETIVOS ESPECÍFICOS**

- Aplicar abstracción a un problema dado.
- Implementar una solución utilizando el lenguaje de programación Python.
- Utilizar estructuras de programación secuenciales, cíclicas y condicionales.
- Aprender a desarrollar reportes con la herramienta Graphviz.
- Utilizar XML para manipulación de datos.
- Aplicar conceptos de memoria dinámica por medio de la implementación de TDAs.

## A. Descripción del problema

La “Agencia Espacial de Guatemala” (AEG) ha logrado enviar naves robóticas a Marte y ha logrado traer muestras de distintos lugares del suelo marciano. Estas muestras han sido enviadas al “Laboratorio Nacional de Guatemala” (LNG) con el objetivo de identificar algún tipo de vida en las muestras.

LNG identifica distintos organismos vivos en las muestras obtenidas por AEG, estos organismos son denominados “Organismo A”, “Organismo B”, etc., además, establece que estos organismos siguen una única regla para prosperar en las muestras recolectadas. La regla básica que siguen los organismos marcianos es la siguiente:

- Un organismo “X” solo puede extenderse si existe otro Organismo “Y” que sirva para alimentar la expansión, de lo contrario, el organismo “X” no podrá extenderse por las muestras y no sobrevivirá.

El “Instituto Tecnológico de Palín” ha logrado crear un dispositivo capaz de analizar las muestras marcianas y generar una cuadrícula de “M” filas por “N” columnas que representa la composición orgánica actual de la muestra. A Continuación, se muestra un ejemplo de estas cuadrículas:

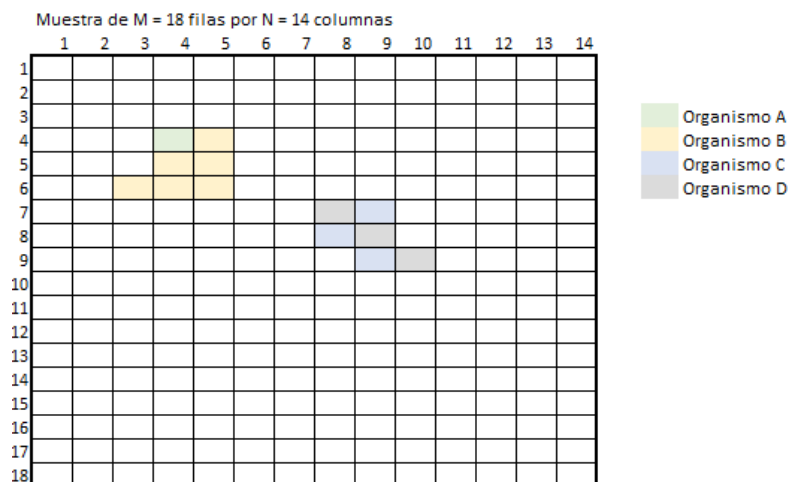


Fig. 1 – Muestra de 18 filas por 14 columnas con 4 Organismos marcianos identificados

La figura No. 1 presenta un ejemplo de una muestra analizada por el dispositivo creado por el “Instituto Tecnológico de Palín” donde se puede identificar que la cuadrícula es de 18 filas por 14 columnas y que cuenta con los organismos marcianos A, B, C y D representados en distintos colores, cuando hay ausencia de organismos en la muestra, las celdas se representan en color blanco.

La regla básica descubierta por el LNG consiste en identificar que si toman una porción de los organismos identificados y los colocan en una celda sin vida, éstos prosperan únicamente si queda algún organismo de otro tipo atrapado horizontal, vertical o en diagonal dentro de la cuadrícula, de lo contrario, los organismos colocados en la celda mueren. A continuación, se muestra un ejemplo de esta regla.

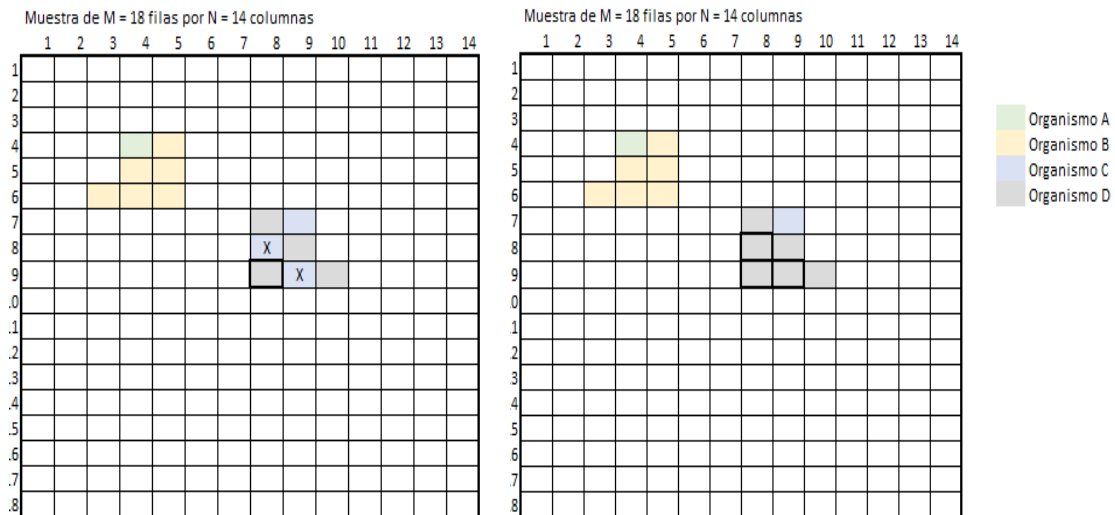


Fig. 2 – Se colocan organismos “D” en la celda sin vida (Fila=9,Col=8)

En la figura 2, se muestra la colocación de “Organismos D” en la celda 9,8; debido a que en las celdas 8,8 y 9,9 hay “Organismos C” que quedan atrapados entre “Organismos D”, entonces el “organismo D” de la celda 9,8 sobrevive, pero además prospera y se alimenta con los organismos de las celdas 8,8 y 9,9, por lo que cambia la composición de la muestra.

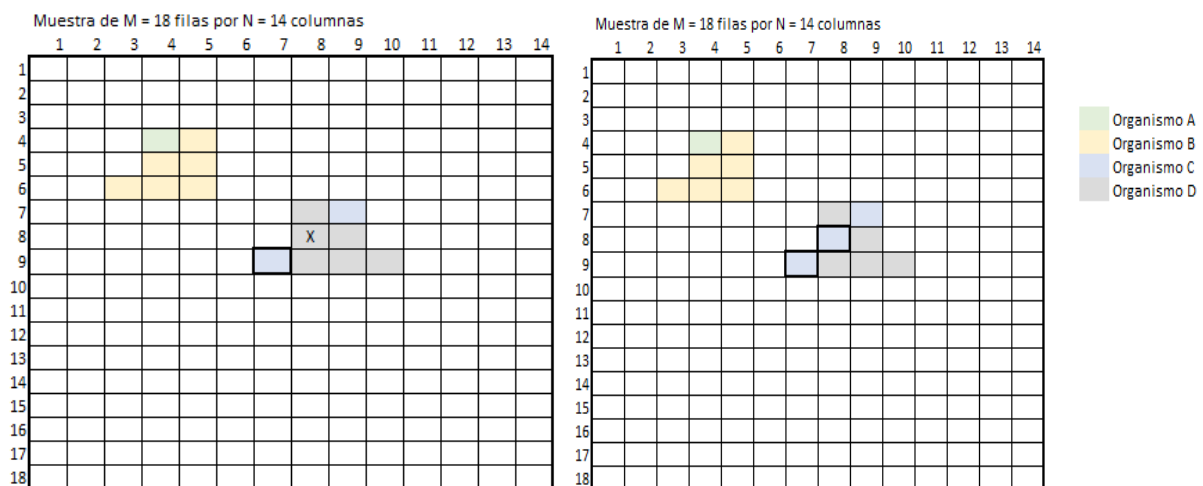


Fig. 3 – Se colocan organismos “C” en la celda sin vida (Fila=9,Col=7)

En la figura 3, se muestra la colocación de “Organismos C” en la celda 9,7; debido a que la celda 8,8 tiene “Organismos D” que quedan atrapados entre “Organismos C”, entonces los “Organismos C” sobreviven y prosperan al alimentarse de los organismos de la celda 8,8, por lo que de nuevo cambia la composición de la muestra.

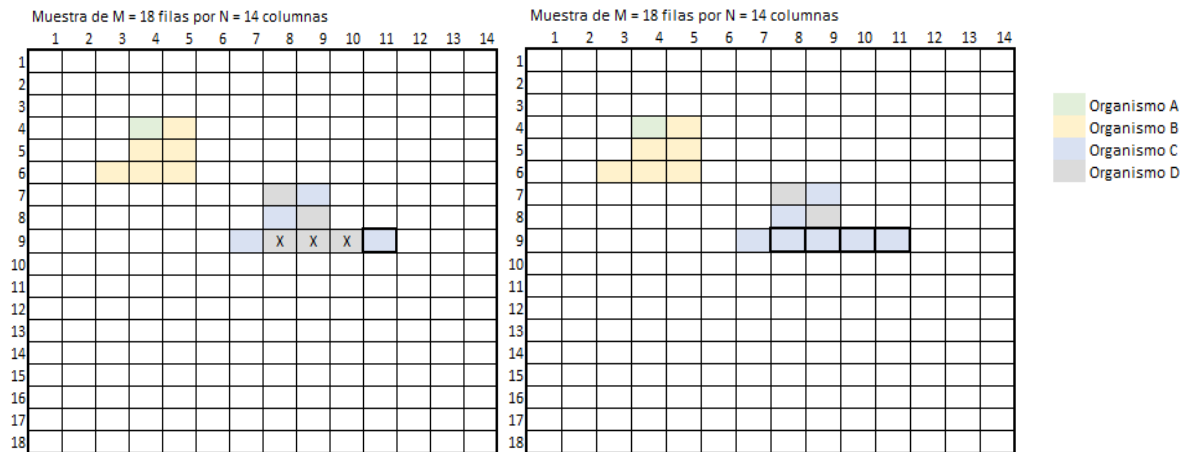


Fig. 4 – Se colocan organismos “C” en la celda sin vida (Fila=9,Col=11)

En la figura 4, se muestra la colocación de “Organismos C” en la celda 9,11; debido a que las celdas 9,8; 9,9 y 9,10 tienen “Organismos D” atrapados, entonces los “Organismos C” sobreviven y prosperan al alimentarse de los “Organismos D” de las celdas 9,8, 9,9 y 9,10, por lo que cambia la composición de la muestra.



Fig. 5 – Se colocan Organismos “C” en la celda sin vida (Fila=10,Col=11)

En la figura 5, se muestra la colocación de Organismos “C” en la celda 10,11; debido a que no hay celdas con organismos distintos a “C” que queden atrapados horizontal, vertical o en diagonal, los organismos “C” no prosperan y mueren, por lo tanto, la muestra mantiene su composición inicial.

## B. Programa a desarrollar

Usted ha sido nombrado por la Facultad de Ingeniería de la USAC para construir un programa en lenguaje Python que permita cargar las muestras y que permita a los científicos de LNG realizar las siguientes operaciones:

- Por cada organismo presente en la muestra, identificar las celdas donde podría prosperar.

- Colocar algún organismo presente en la muestra en una celda sin organismos vivos e identificar el estado final de la muestra, estableciendo claramente si los organismos colocados prosperan o mueren.
- Identificar si una muestra ya no puede ser modificada, es decir, ya no es posible colocar organismos que prosperen en las celdas sin vida disponibles.

### Límites

Tamaño máximo de la cuadrícula: 10,000 filas y 10,000 columnas

Cantidad máxima de organismos: 1,000

### Ingreso de datos iniciales

Los datos iniciales serán cargados en un archivo XML, este archivo contendrá el listado de los organismos que podrán ser incluidos en las muestras. Por cada muestra se indicará la cantidad M (filas) y la cantidad N (columnas), y luego se identificará cada celda que tenga un organismo vivo y el nombre del organismo vivo que contiene.

### Funcionalidad de la aplicación a desarrollar

Se espera que la aplicación permita elegir un archivo XML que genere las muestras iniciales y su composición de vida. Además, la aplicación deberá permitir seleccionar una muestra para que el científico pueda identificar las celdas donde puede colocar alguno de los organismos que contiene la muestra, el investigador podrá colocar organismos en alguna celda y validar la forma en que cambia la muestra, además, la aplicación deberá indicar cuando una muestra ya no pueda ser modificada. El investigador podrá actualizar la información de la muestra, o bien, crear una nueva muestra con los cambios introducidos.

La aplicación también proveerá una opción para generar una salida a un archivo XML con la información de las muestras actualizadas. Este archivo tendrá la misma estructura del archivo de entrada y podrá ser utilizado como un nuevo archivo de entrada.

La aplicación debe brindar una opción de inicialización, de manera que pueda cargarse un nuevo archivo desde 0, si se carga un archivo y existen organismos y muestras previamente existentes, éstas deberán continuar en la aplicación, es decir, la carga de archivos incrementa la cantidad de información en el sistema.

La aplicación debe ser intuitiva y fácil de utilizar por los investigadores de LNG.

### Ejemplo de archivo XML

#### Ejemplo de Entrada:

```
<?xml version="1.0"?>
<datosMarte>
  <listaOrganismos>
    <organismo>
      <codigo>[valorAlfanumérico]</codigo>
      <nombre>[valorAlfanumérico]</nombre>
    </organismo>
    ...
  </listaOrganismos>
  <listadoMuestras>
    <muestra>
      <codigo>[valorAlfanumérico]</codigo>
      <descripcion>[valorAlfanumérico]</descripcion>
      <filas>[valorNumérico]</filas>
      <columnas>[valorNumérico]</columnas>
      <listadoCeldasVivas>
        <celdaViva>
```

```
<fila>[valorNumérico]</fila>
<columna>[valorNumérico]</columna>
<codigoOrganismo>[valorAlfanumérico]<codigoOrganismo>
</celdaViva>
....
</listadoCeldasVivas>
</muestra>
....
</listadoMuestras>
</datosMarte>
```

## C. 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 puedan realizar las operaciones con las muestras.

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 organismos y todas las muestras dentro del archivo XML deben de ser cargados a las listas para luego tener la habilidad de realizar las operaciones requeridas por los investigadores, o bien, generar el archivo de salida.

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 del diseño del software desarrollado y se recomienda incluir el modelo conceptual y los diagramas de actividades que modelan 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 la Escuela de Sistemas.
- 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 **05 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.