
IMPLEMENTACIÓN DE UN SISTEMA DE SIMULACIÓN DE REACCIONES DE PROTEÍNAS PARA EL TRATAMIENTO DEL CÁNCER MEDIANTE TDA Y GRAPHVIZ

202300689 – Josue Daniel Herrera Cottom

Resumen

Este proyecto implementa una solución computacional para evaluar la efectividad de un tratamiento contra el cáncer basado en reacciones de proteínas. Utilizando programación orientada a objetos (POO), tipos de datos abstractos (TDA) y la herramienta Graphviz, se simula el comportamiento de una rejilla de tejido afectado por células cancerígenas. El sistema procesa archivos XML para cargar experimentos, identifica parejas de proteínas reactivas y calcula el porcentaje de células inertes tras iteraciones de reacciones adyacentes. Los resultados determinan si el medicamento es exitoso, no efectivo o fatal. La solución integra estructuras de datos personalizadas y visualización gráfica, cumpliendo con los requisitos académicos y técnicos del curso.

Palabras clave

Cáncer, proteínas, TDA, Graphviz, XML

Abstract

This project implements a computational solution to evaluate the effectiveness of a cancer treatment based on protein reactions. Using object-oriented programming (OOP), abstract data types (ADT), and the Graphviz tool, the system simulates the behavior of a grid of tissue affected by cancer cells. The system processes XML files to load experiments, identifies reactive protein pairs, and calculates the percentage of inert cells after iterations of adjacent reactions. The results determine whether the drug is successful, ineffective, or fatal. The solution integrates custom data structures and graphical visualization, meeting the academic and technical requirements of the course.

Keywords

Cancer, proteins, ADT, Graphviz, XML

Introducción

El cáncer es una de las principales causas de mortalidad global, por lo que la investigación de tratamientos innovadores es crucial. Este proyecto aborda un método propuesto por la Facultad de Medicina de la USAC, que utiliza reacciones entre proteínas para eliminar células cancerígenas. La solución desarrollada automatiza la simulación de este proceso mediante técnicas de programación avanzadas. El objetivo es proveer una herramienta que determine la eficacia del tratamiento basándose en el porcentaje de células inertes generadas. Este ensayo detalla el diseño, implementación y resultados del sistema, destacando el uso de TDA, POO y visualización de datos.

Desarrollo del tema

a. Diseño del sistema

El sistema se modeló mediante un diagrama de clases que incluye:

Experimento: Gestiona la rejilla de proteínas y las parejas reactivas.

Rejilla: Representa la matriz de tejido como una estructura de datos personalizada (lista doblemente enlazada para filas y columnas).

ProcesadorXML: Parsea archivos XML para cargar experimentos.

Visualizador: Genera gráficos de estados iniciales y finales usando Graphviz.

b. Implementación de estructuras de datos

Se desarrollaron listas doblemente enlazadas para almacenar la rejilla, permitiendo acceso eficiente a celdas adyacentes. Cada nodo contiene una proteína (cadena de aminoácidos) y su estado (activo/inerte).

c. Algoritmo de reacción

El algoritmo sigue estos pasos:

Identificar todas las parejas de proteínas reactivas cargadas desde XML.

Recorrer la rejilla para detectar celdas adyacentes (horizontal/vertical) que coincidan con las parejas.

Marcar celdas como inertes y repetir el proceso hasta que no haya más reacciones.

Calcular el porcentaje de inertes y clasificar el medicamento.

d. Integración con Graphviz

Se utilizó la librería graphviz para generar representaciones visuales en formato DOT. Cada celda se muestra como un nodo con color según su estado (verde: activo, rojo: inerte). La Figura 1 muestra un ejemplo.

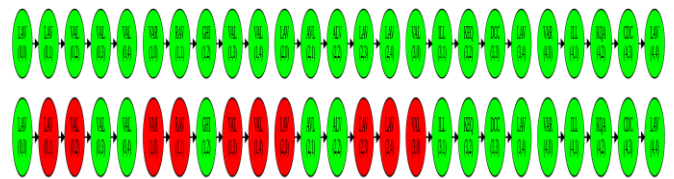


Figura 1. Visualización de una rejilla antes y después de las reacciones.

Fuente: Elaboración propia.

e. Manejo de archivos XML

Se implementó un parser manual sin bibliotecas externas. El archivo XML se procesa línea por línea, extrayendo datos de experimentos, rejillas y parejas de proteínas.

f. Validación y pruebas

Se probaron casos límite, como rejillas vacías o sin parejas reactivas. Un caso de éxito fue una rejilla de 5x5 donde el 45% de celdas se volvieron inertes, clasificando el medicamento como exitoso.

Conclusiones

- El uso de TDA y POO permitió un diseño modular y escalable.

- Graphviz fue esencial para depurar y visualizar iteraciones complejas.
- Las restricciones de no usar estructuras nativas de Python fomentaron la comprensión profunda de listas enlazadas.
- El proyecto demuestra cómo la computación puede apoyar investigaciones médicas innovadoras.

Enfatizando, lo importante es destacar las principales posturas fundamentadas del autor, que desea transmitir a los lectores.

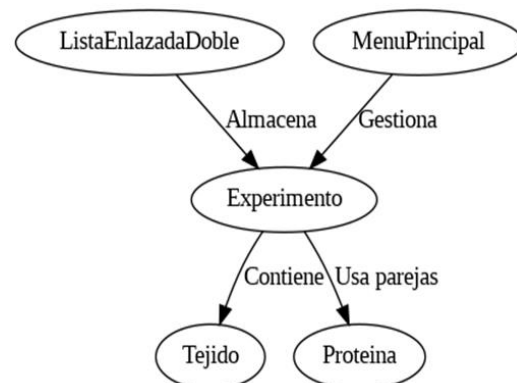
Adicionalmente, pueden incluirse preguntas abiertas a la reflexión y debate, temas concatenados con el tema expuesto o recomendaciones para profundizar en la temática expuesta.

Referencias bibliográficas

1. USAC. (2025). Proyecto 1 - IPC2. Documento interno.
2. Graphviz. (2023). Documentación oficial. <https://graphviz.org>
3. Python Software Foundation. (2023). Python 3.11 documentation.
4. Universidad San Carlos. (2025). Guía de ensayos académicos.
5. Date, C. J. (1991). *An Introduction to Database Systems*. Addison-Wesley.

Extensión:

- **Apéndice A:** Diagrama de clases del sistema.



- **Apéndice B:** Ejemplo de archivo XML de entrada.

Ejemplo de XML

```
<?xml version="1.0" encoding="UTF-8"?>
<experimentos>
  <experimento nombre="paciente01">
    <tejido filas="3" columnas="3">
      <rejilla>
        AGV STY QWE
        RTY UIO PAS
        ZXC VBN MLK
      </rejilla>
    </tejido>
  </experimento>
</experimentos>
```

```
<proteinas>
  <pareja>AGV STY</pareja>
</proteinas>
</experimento>
</experimentos>
```

- **Apéndice C:** Capturas de pantalla de visualizaciones en Graphviz.

LAV	LAV	VAL	VAL	VAL
VAR	RAV	GHI	VAL	VAL
LAV	AVL	ALV	LAV	LAV
VAL	ILL	KEQ	DCC	LAV
VAR	ILL	KQA	CDC	LAV

LAV	INERTE	INERTE	VAL	VAL
INERTE	INERTE	GHI	INERTE	INERTE
INERTE	AVL	ALV	INERTE	INERTE
INERTE	ILL	KEQ	DCC	LAV
VAR	ILL	KQA	CDC	LAV