

---

## PROYECTO NO. 1

---

201807154 – Denny Alexander Chalí Miza

### Resumen

Para la realización del proyecto No. 1 se realizó una aplicación cuyas funciones son las de analizar un archivo de texto plano con formato XML, el cual contiene los datos de distintos pacientes. La función principal del programa es la de crear una malla con la información cargada en el archivo para luego analizarla y obtener una un diagnostico de la gravedad de la enfermedad del paciente.

Para la solución del problema se hizo uso de TDA, usando listas simples, listas doblemente enlazadas y listas ortogonales, siendo la lista ortogonal la que contiene los datos de las mallas de las células de cada paciente.

Para mostrar gráficamente la malla de las células sanas y enfermas se hizo uso del software Graphviz, la cual muestra la matriz que contiene la lista ortogonal y la lista de celdas que componen la malla.

### Palabras clave

Pila, Lista, Funciones, Métodos y Clases.

### Abstract

*For the realization of project No. 1, an application was developed whose functions are to analyze a plain text file in XML format, which contains the data of different patients. The main function of the program is to create a mesh with the information loaded in the file and then analyze it to obtain a diagnosis of the severity of the patient's disease.*

*For the solution of the problem we made use of TDA, using simple lists, double linked lists and orthogonal lists, being the orthogonal list the one that contains the data of the meshes of the cells of each patient.*

*To graphically display the mesh of healthy and diseased cells, use was made of Graphviz software, which shows the matrix containing the orthogonal list and the list of cells that make up the mesh.*

### Keywords

*Stack, List, Functions, Methods and Clases.*

## Introducción

Se desarrollo una aplicación que permite la lectura y análisis de archivos con formato XML, el archivo contiene los nombres de pacientes, el tamaño de la rejilla y la representación de la rejilla con el patrón inicial.

La aplicación permite elegir un paciente para luego ejecutar los periodos uno para ver gráficamente el desarrollo de la enfermedad, también se dispone de una opción que permite de forma automática la ejecución de los periodos necesarios para diagnosticar la gravedad de la enfermedad. Otra función del programa es la de generar un archivo en formato XML que muestra los datos de cada paciente y la gravedad de la enfermedad.

## Desarrollo del tema

### 1. Preparando el entorno de trabajo

Para la resolución del problema se utilizó el lenguaje de programación Python, el cual deberá de descargar e instalar en su ordenador, para el correcto funcionamiento del programa se recomienda instalar una versión superior a la 3.6.

#### 1.1 Graphviz

Graphviz es un software de visualización de gráficos de código abierto que nos servirá para graficar los patrones de los pisos de forma gráfica.

Para su instalación debemos de tener previamente instalado Python.

Abrimos el símbolo del sistema (CMD) e ingresamos:

```
$ pip install graphviz
```

Para renderizar el código fuente DOT generado se puede usar el software Grahviz descargándola desde su sitio oficial.

## 2. TDA implementados

Para almacenar los datos de los archivos y manejar las rutas en cada malla fueron necesarios implementar distintos tipos de TDA.

### 2.1 Listas simples enlazadas

Una lista simple o lista simplemente ligada esta constituida por un conjunto de nodos alineados de manera lineal (uno después de otro) y unidos entre sí por una referencia (apuntador).

A diferencia de un arreglo, el cual también es un conjunto de nodos alineados de manera lineal, el orden está determinado por una referencia, no por un índice, y el tamaño no es fijo.

La unidad básica de un alista simple es un elemento nodo, cada elemento de la lista es un objeto que contiene la información que se desea almacenar, así como una referencia (NEXT) al siguiente elemento (sucesor).

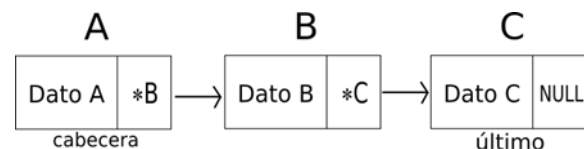


Figura 1. Lista enlazada simple

Fuente: Alberto Castillo G, obtenido de:  
[www.betoissues.com](http://www.betoissues.com).

## 2.2 Listas Doblemente enlazadas

Es una estructura de datos dinámica que se compone de un conjunto de nodos en secuencia enlazados mediante dos apuntadores (uno hacia adelante y otro hacia atrás).

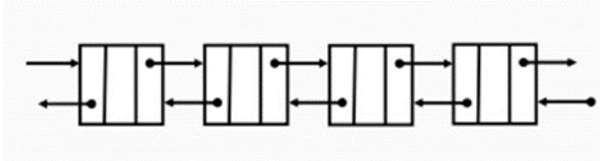


Figura 2. Lista doblemente enlazada.

Fuente: Bruno López Takeyas, M.C, obtenido de:  
[www.itnuevolaredo.edu.mx/Takeyas](http://www.itnuevolaredo.edu.mx/Takeyas)

## 2.3 Listas Ortogonales

Este tipo de lista se utiliza para representar matrices. Los nodos contienen cuatro apuntadores. Uno para apuntar al nodo izquierdo, otro para apuntar al derecho, otro al nodo inferior y por último un apuntador al nodo superior.

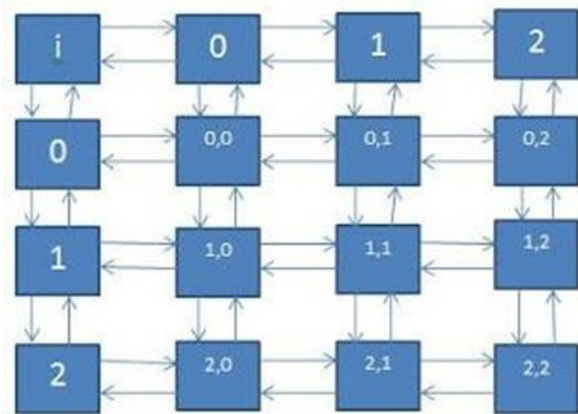


Figura 3. Lista ortogonal.

Fuente: Humberto Herrado, obtenido de: [ciencias-sistemas.blogspot.com/](http://ciencias-sistemas.blogspot.com/)

## 3. Ejecución del programa

### 3.1 Posicionarse en la carpeta (CMD)

Una vez el entorno de trabajo está listo podemos iniciar el programa, para ello abrimos una pestaña en el símbolo del Sistema (CMD) y navegamos hasta estar en la carpeta donde se ubica el ejecutable de nuestro programa:

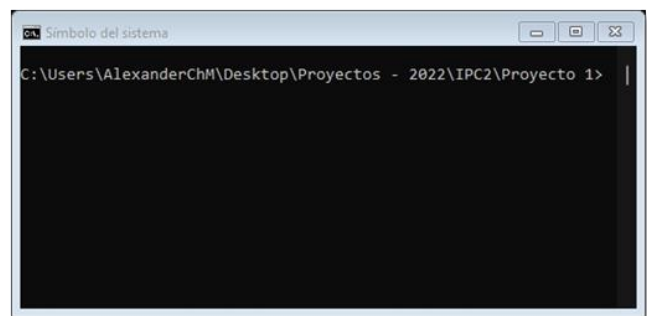


Figura 4. Ejemplo de ubicación en carpeta

Fuente: elaboración propia.

Ingresamos el siguiente comando una vez dentro de la carpeta que contine el ejecutable:

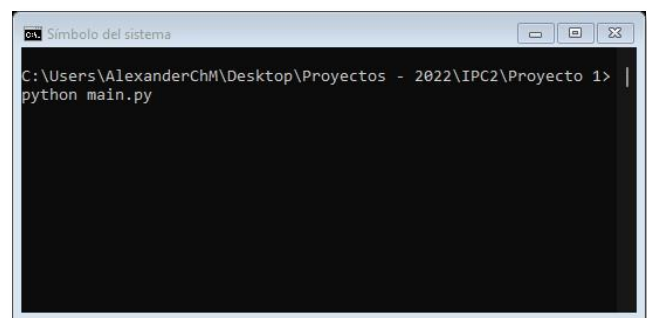


Figura 5. Inicializando el programa

Fuente: elaboración propia.

Una vez realizado los pasos anteriores se desplegará el menú de inicio del programa.

## 3.2 Funciones del programa

### 3.2.1 Menú inicio

Al iniciar el programa se desplegará el menú inicial con las principales funciones como se muestra en el ejemplo:

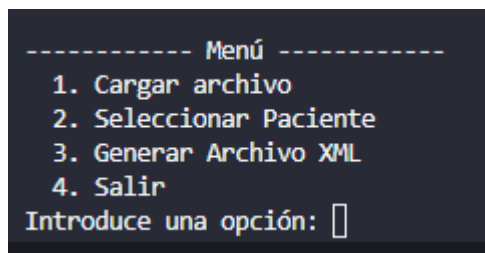


Figura 6. Menú de inicio

Fuente: elaboración propia.

Para seleccionar una función se deberá de ingresar el número de la opción correspondiente como se muestra a continuación:

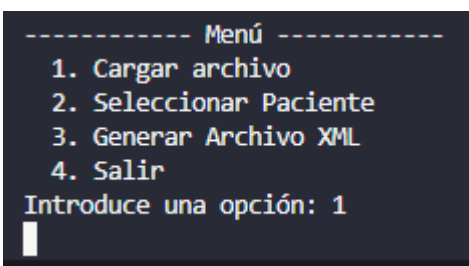


Figura 7. Ingreso de opción

Fuente: elaboración propia.

Para los submenús se utiliza el mismo método para seleccionar opciones y seleccionar elementos.

### 3.2.2 Cargar Archivo XML

Esta función se encarga de desplegar una ventana emergente y seleccionar el archivo de entrada que contine los datos de cada paciente.

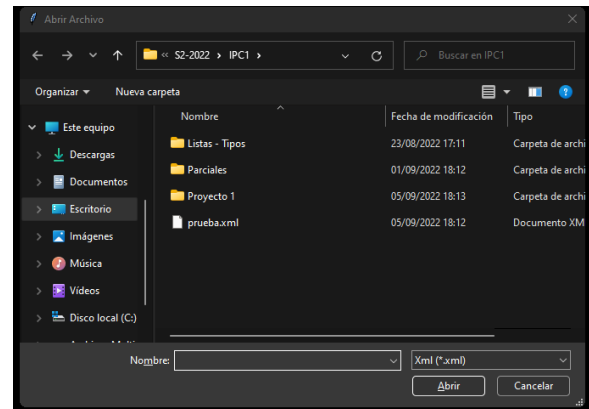


Figura 8. Ventana emergente

Fuente: elaboración propia.

Una vez se localiza el archivo XML con los datos se selecciona, si los datos son correctos se mostrará un mensaje indicando los resultados satisfactorios de lectura, en caso contrario se mostrará un mensaje indicando que los datos no se cargaron.

### 2.2.3 Seleccionar paciente

En esta opción se listan los pacientes cargados con anterioridad al sistema. Se deberá de seleccionar una opción ingresando el número del índice del nombre del paciente.

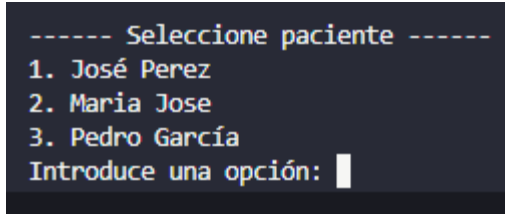


Figura 9. Seleccionar paciente  
Fuente: elaboración propia.

Una vez seleccionando el paciente se desplegará un menú con las siguientes opciones.

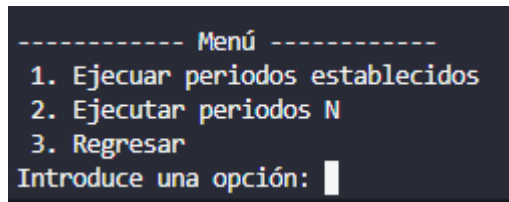


Figura 10. Ingreso de opción  
Fuente: elaboración propia.

### 2.2.3.1 Ejecutar periodos

Esta opción ejecuta los periodos uno a uno para ver gráficamente la forma en que se desarrolla la enfermedad en el tejido proporcionado en la rejilla inicial, además se identifica el momento en que el patrón inicial se repita o en donde el patrón N1 se repite, indicando el periodo en que apareció el patrón y en cuantos periodos empezó a repetirse.

### 5.2.3.2 Ejecutar periodos automáticamente

Ejecuta automáticamente todos los periodos necesarios hasta que se identifique si la enfermedad es mortal, grave o en dado caso se llegó al limite de periodos de la aplicación (10000).

## 2.3.4 Visualización gráfica del desarrollo de la enfermedad

En la carpeta que se encuentra el programa se creará una carpeta que contendrá la información del avance de la enfermedad en el tejido de cada paciente que se analizó, donde se muestran las células sanas y las contagiadas.

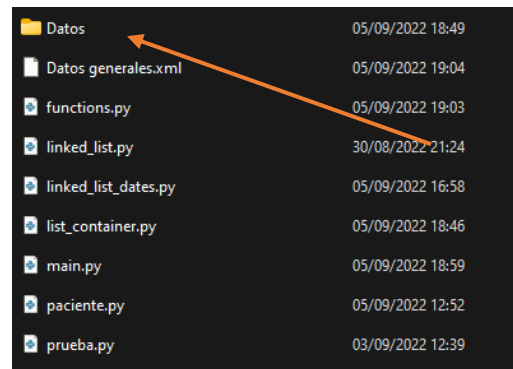


Figura 11. Visualización de carpeta datos.

Fuente: elaboración propia

En la carpeta datos se encontrarán los datos del desarrollo de cada paciente como se muestra a continuación:

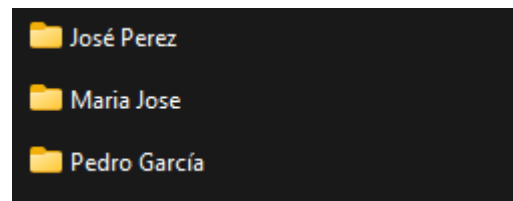


Figura 12. Carpeta de cada paciente

Fuente: elaboración propia

Dicha carpeta contiene una serie de imágenes en las cuales se visualiza el periodo y rejilla que corresponde a cada una de ellas.

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Paciente: José Perez  
Patron inicial  
Periodo: 0  
Rejilla: 1

Figura 13. Representación gráfica de una rejilla

Fuente: elaboración propia

### 2.3.5 Generar archivo XML

El programa genera un archivo de texto plano con formato XML. Este archivo contendrá la información de cada paciente, si desarrollará una versión grave, mortal o leve de la enfermedad y el período en que se obtendrá el patrón inicial, o bien, el período en que se tendrá un patrón que se repetirá, así como el período secundario necesario para repetir dicho patrón.

El archivo Xml se guardará en la carpeta raíz del programa con el nombre “Datos genrales.xml”.

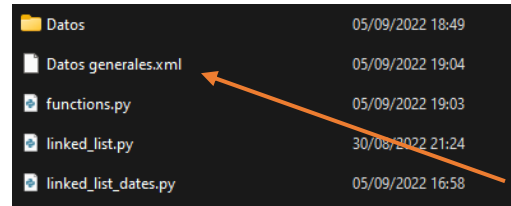


Figura 14. Carpeta de cada paciente

Fuente: elaboración propia

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

### Referencias bibliográficas

Lewis, John; Loftus, William (2009). *Java Software Solutions Foundations of Programming Design* 6th ed. Pearson Education Inc.

Chacón Sartori, Camilo. *Computación y programación funcional: introducción al cálculo lambda y la programación funcional usando Racket y Python*. [Barcelona]: Marcombo.

## Anexos

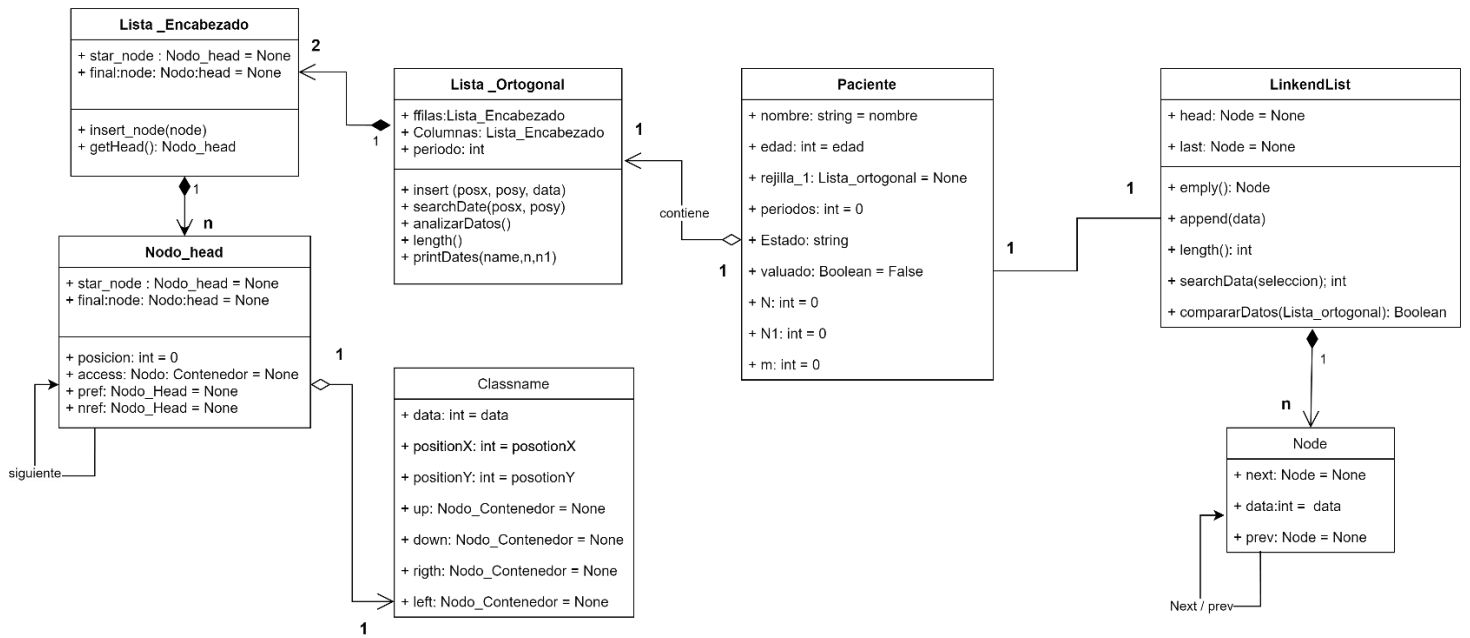


Figura 13. Diagrama de clases para la solución del problema

Fuente: elaboración propia