

## PROYECTO 1

# DETECCIÓN DE ENFERMEDADES POR MEDIO DE ANÁLISIS DE CÉLULAS

---

**Carnet: 202101149 – Mariano Roberto Rac Noguera**

### Resumen

Tras una exhaustiva investigación el Laboratorio de epidemiología consulta el desarrollo de un software que permita realizar diagnósticos sobre ciertas enfermedades que infectan las células, basado en los patrones que ellos han encontrado, verificando como se infectan y sana las células con el paso del tiempo y determinar la severidad de la enfermedad.

Para el desarrollo del software se utilizó principalmente la estructura de datos de listas enlazadas y sus nodos así también se auxilio de distintas librerías y lectura a los pacientes por medio de un archivo XML. El usuario ingresa los pacientes y células infectadas en el archivo y el programa lo reconoce, guarda los datos, genera un tablero, aplica el algoritmo y busca los parámetros para determinar la propagación de la enfermedad en base a los estatutos para determinar el estado del paciente y finalmente generar un documento con el diagnóstico.

### Palabras clave

Listas , enlazada, nodo, algoritmo, búsqueda.

### Abstract

*After a deep research The epidemiological Lab consults a software developing that allows to perform diagnoses about some diseases that infect cells, based on the patterns the have found, verifying how are they infected and healed within the passage of time and determining the severity from the disease.*

*For the software developing the data structure linked list and its nodes were mainly used with some libraries aid and lecturing the data with an XML document. The user indexes the patients and infected cells in the document and the program gives read to it, saves the data, generates a table, applies the algorithm and searches the parameters to determine the diseases spreading based on the bylaws to determine the patient status and finally generating a document with de diagnosis.*

### Keywords

*Lists, linked, node, algorithm, search*

## Introducción

Uno de los principales problemas que afronta la población mundial es la propagación de diversos virus, en el proyecto se desarrolló un programa enfocado en analizar el comportamiento de células infectadas basado en los requerimientos del Laboratorio de Investigación Epidemiológica de Guatemala . El software desarrollado consiste en una serie de algoritmos que permite analizar la propagación de estas células infectadas y como se comportan y cambian con el paso del tiempo, utilizando a su vez listas enlazadas simples y dobles para el algoritmo además de realizar una visualización gráficas por medio de Graphviz para poder ver el cambio de las células. Posterior al análisis por medio de un documento de salida se visualizan los resultados obtenidos por paciente.

## Desarrollo del tema

Se empleo el lenguaje multiparadigma de Python para la realización del software requerido creando una aplicación del tipo consola y aprovechando las distintas librerías que ofrece el lenguaje para realizar un código eficiente además de utilizar principalmente las estructuras de datos denominadas listas

tanto simples y dobles para la realización del programa.

En caso de incluir referencias documentales o información recopilada, ésta debe referenciarse de acuerdo con las normas APA, identificando con claridad las citas textuales para distinguirlas de las redacciones propias.

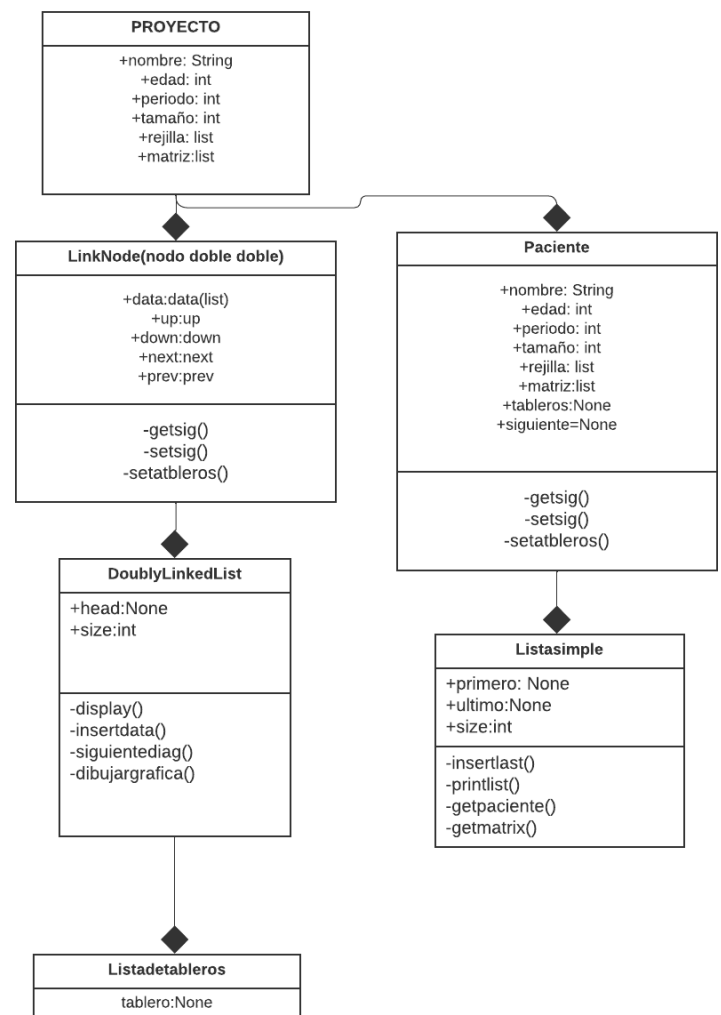
El lenguaje de Python tiene demasiadas herramientas entre ellas las listas nativas de Python, ahora bien ¿Por qué utilizar listas enlazadas en vez de listas simples? Primero se debe conceptualizar correctamente las listas que son básicamente una colección de elementos sin embargo en muchos lenguajes estas listas o arreglos nativos tienen las dimensiones ya definidas a diferencia de las listas enlazadas que puede modificarse, se define a una lista enlazada como una colección lineal de estructuras que se auto referencian esto se denominan nodos, y estas están conectadas por medio de enlaces apuntadores.

En el desarrollo de la aplicación se utilizó lista simplemente enlazada para poder llenarla con los datos e información de los pacientes obteniendo los datos del XML y por medio de métodos ir llenando la lista simple donde cada paciente apunta al siguiente y esto facilita la manipulación y almacenamiento de los datos. Para acceder al XML se utilizó la librería

El enfoque principal del proyecto es analizar al paciente y propiamente analizar el tejido de células infectadas de este en un tablero N x N, para ello se reciben los datos y se llena una lista doblemente enlazada ( en realidad usa 4 apuntadores up, down, next y previous) con un algoritmo que va formando una matriz básicamente, donde cada espacio es un nodo del tablero y puede analizarse, el algoritmo busca las células infectadas y determina en cada período los cambios según los parámetros establecidos y modifica directamente en el nodo utilizando los valores de 1 para infectada y 0 para sana, además después de cada período se almacena en una lista simplemente enlazada un historial por paciente que permite comparar el actual con el siguiente y basado en el comportamiento y repeticiones determinar el estado del paciente y si tiene repetición de patrones, al final se acceden a estos datos que fueron comparados y de existir las condiciones necesarias se establece el estado del paciente y se crea el documento de salida con los datos de este.

ARCHIVOS	CLASES
Proyecto.py	Main
Nodo.py	Paciente, LinkNode, DoublyLinkedList, Listadetableros
Listasimple.py	listasimple

Fuente: elaboración Propia



Fuente: elaboración Propia

En el uso de las listas simplemente enlazadas y dobles la estructura es la que ya ha sido definida es decir están compuestos por una clase nodo con los métodos inicializable, get que retorna los datos, set que pone datos y además un set tablero que se relaciona al tablero de células infectadas por cada paciente la estructura para el nodo de la lista doble es bastante similar.

La clase propia de lista simple tiene su método inicializable, un método para inserción de datos al final de la lista simple enlazada cuya estructura comprende los atributos del paciente y creando un nuevo paciente cada vez verificando por supuesto el tamaño y empleando el apuntador.

Además se realizó un método de impresión que muestra en consola los pacientes añadidos.

Se realizó también un método getpaciente para seleccionar el paciente al cual se desea aplicar el algoritmo y determinar su estado de salud con una estructura donde se auxilia de una variable temporal que va recorriendo la lista enlazada y busca comparar el nombre del paciente que se desea con los existentes así mismo se creó un método getmatrix que retorna el tablero del paciente

La clase doble lista enlazada posee también su método inicializable, un método para imprimir el tablero en forma de doble lista

enlazada, un método que inserta los datos en el cuál añade el primer nodo en la posición de origen y luego recorre y llena los demás nodos simulando una matriz, el método de siguiente diagnóstico es el centro del software pues este contiene el algoritmo que ejecuta los períodos empleando los 4 apuntadores y básicamente buscando los nodos aledaños al cual se esta evaluando donde mayoritariamente serán 8 nodos vecinos salvo excepciones y se verifican las condiciones para alterar los valores del nodo además de utilizar una variable auxiliar columna que emula nuevamente la forma matricial que se desea.

La creación de las Gráficas por medio de Graphviz sucede en esta misma clase donde la forma en la cual se recorren los nodos de la matriz por cada período es similar al algoritmo que recorre los nodos para su impresión, auxiliándose de este código ya planteado se llamo a la librería Graphviz después de haberla establecido como variable del sistema e instalando todos los complementos necesarios y se fue dando formato con los atributos que poseen los “Diagraphs” de esta librería cambiando los caracteres de 1 y 0 por unos más atractivos que emulan la existencia de células sanas y enfermas, al final todos los grafos se crean en

formato .png y se crea una carpeta con el nombre del paciente y ahí estarán almacenados todos los períodos .

Con la lista de tableros se posee una estructura similar en cuanto a los demás nodos y la lista simple y esta lleva un historial por cada paciente que se evalúa al cual posterior se accede en el main que busca los comparativos respectivos para hallar los patrones y generar los diagnósticos.

## Conclusiones

¿Es realmente más efectivo usar listas enlazadas? Si, indudablemente las listas enlazadas tienen demasiadas ventajas destacando primeramente que el orden de los elementos que han sido enlazados puede ser diferente al de almacenamiento, y cada nodo posee un espacio de almacenamiento en otras palabras, los nodos no tienen que estar contiguos en la memoria. Además se sigue manteniendo una secuencia ordenada y el acceso a los datos que se almacenan en los nodos es sumamente rápido y los algoritmos

de búsqueda y ordenamiento son aplicables a estos además de que se pueden generar otro tipo de estructuras como las matrices en base a estas listas enlazadas.

Los algoritmos de búsqueda siguen siendo un pilar de la programación y son la base del software desarrollado auxiliándose de la estructura de datos listas enlazadas y sus apuntadores pueden buscarse datos específicos en los nodos y obtener todos los datos que almacena ese nodo en base a un parámetro. Empleando diferentes apuntadores es decir con distintas referencias otros nodos es de suma utilidad pues la modificación o edición de los datos contenidos en los nodos es bastante efectiva y si fuese necesario añadir o eliminar también es bastante bueno.

Aplicando los conocimientos de este tipo de estructuras y auxiliándose con las librerías que ofrece este lenguaje de programación se desarrolló exitosamente un software que cumple a grandes rasgos con los objetivos planteados y un acceso bastante rápido y simplificado para el usuario.

## Referencias bibliográficas

- García, M. E. (2015). *Estructurad de datos lineales: Lista Doblemente Ligada y doblemente ligada Circular*.
- Rodríguez, R. R. (s.f.). Estrcuturas de Datos.
- Wachenchauzer, R. (2014). *Uniwebsidad*.  
Obtenido de Otras listas enlazadas.