
Proyecto 3: Tecnologías chapinas SA

3023673810101 – David Abraham Noriega Zamora

Resumen

El Objetivo de este proyecto es ayudar al laboratorio de investigación epidemiológica de Guatemala a investigar la forma en que las enfermedades infectan las células del cuerpo y se expanden produciendo enfermedades graves e incluso mortales.

Es por este motivo que se ha desarrollado un software de simulación que recopila los datos de un grupo de pacientes y crea una línea del tiempo mostrando el avance teórico de la enfermedad. Al terminar, reporta los resultados indicando si el paciente desarrollara una versión grave, ligera o mortal de la enfermedad y en qué momento sucederá.

Palabras clave

Lista enlazada, simulación, graphviz, estructuras de datos, python

Summary

This Project's objective is helping Guatemala's epidemic investigation laboratory to research the way in which diseases infect body cells and expand producing severe or even deathly illnesses.

Is because of this that a simulation software which gathers data from a group of patients and creates a timeline that shows disease advancement has been developed. When finished, it reports results telling if the patient will develop a severe, light, or deathly version of the disease and when it will happen.

Keywords

Linked list, simulation, graphviz, data structures, python

Introducción

El software de simulación fue desarrollado utilizando el lenguaje de programación Python, utilizando estructuras de datos y listas enlazadas para almacenar la información con el objetivo de procesarla al terminar el proceso. Para la vista grafica de cada paso de la simulación se utilizo la herramienta graphviz, teniendo esta la capacidad de renderizar grandes cantidades de datos de manera optima y en alta calidad. El proyecto se trabajó utilizando un enfoque de programación orientado a objetos.

Desarrollo del tema

Aplicación

En el siguiente apartado se exhiben las pantallas de la aplicación junto con su funcionalidad.

La aplicación fue desarrollada utilizando el framework de django y consta de 5 pantallas con las cuales el usuario podrá interactuar:

- Menú principal.
- Mensaje de configuración.
- Mensaje de consumo.
- Opciones.
- Ayuda.

Menú principal



Figura 1. Menú principal.

Fuente: elaboración propia.

Esta pantalla es la que se muestra al iniciar la aplicación, posee 4 opciones:

1. Enviar mensaje de configuración:

Pasa a la pantalla Mensaje de configuración

2. Enviar mensaje de consumo:

Pasa a la pantalla Mensaje de consumo.

3. Operaciones del sistema:

Pasa a la pantalla Opciones.

4. Ayuda:

Pasa a la pantalla Opciones

Pantalla Mensaje de configuración



Figura 2. Pantalla Mensaje de configuración.

Fuente: elaboración propia.

Pide la dirección de un archivo xml de entrada conteniendo todos los datos de los clientes y las configuraciones y lo carga a la aplicación.

Pantalla Mensaje de consumo

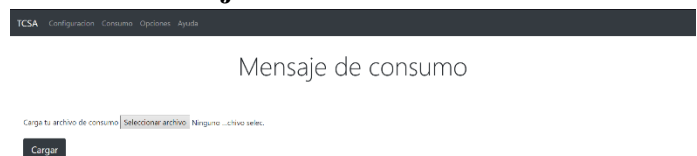


Figura 3. Mensaje de consumo.

Fuente: elaboración propia.

Pide la dirección de un archivo xml de entrada
conteniendo todos los datos de los consumos de los
clientes.

Pantalla Opciones



Figura 4. Opciones.

Fuente: elaboración propia.

Pantalla Ayuda



Figura 5. Ayuda.

Fuente: elaboración propia.

Funcionamiento Backend

Tabla I.

Librerías importadas.

Liberia	Variable importada
minidom	-
flask	-
re	-
datetime	-
Clases	-

Tabla II.

Variables globales.

variable	tipo
global_categorias	Categoria[]
global_recursos	Recurso[]
global_configs	Configuración[]
global_clientes	Cliente[]
global_consumos	Consumo[]
global_facturas	Factura[]
global_instancias	Instancia[]

Endpoints

consultar_datos

Esta función es utilizada para crear un estado inicial de prueba con parámetros aleatorizados con motivo de comprobar la funcionalidad de la aplicación.

crear_recurso

paciente_actual se refiere al Paciente que se está procesando, est_actual es una lista de tuplas en donde se encuentran las células infectadas (x,y), s es la grafica en donde queremos escribir nuestro tablero e inicio es una variable indicando la fila en la que va a iniciar la renderización ya que cada fila tiene una capacidad máxima de 16384 bytes.

La función crea una label ocupando el mayor numero de filas posibles y se lo añade a un nodo, en caso de que el string del label supere la cantidad máxima de bytes que permite graphviz, guarda la ultima fila que escribió y se llama a si mismo de manera recursiva creando un nuevo nodo y alojando el resto de filas, hasta terminar el proceso.

crear_categoria

paciente_act se refiere al Paciente que se está procesando, est_actual es una lista de tuplas en donde se encuentran las células infectadas en el formato (x,y), esta función retorna una lista de tuplas en donde cada tupla contiene las células infectadas de la siguiente iteración.

crear_configuracion

paciente_act se refiere al Paciente que se está procesando, cell es una tupla en donde se las coordenadas de una célula infectada en el formato (x,y), posibles es un diccionario siendo las llaves las coordenadas x_y separadas por un guion bajo y los valores son el número de células infectadas alrededor.

La función trabaja con las coordenadas de la célula infectada y suma 1 a todas las células en su vecindad, haciendo que solo se tenga que iterar las células infectadas en vez de toda la matriz, luego devuelve el valor actualizado de todas las células que tienen en su vecindad a una célula infectada.

crear_cliente

paciente se refiere al paciente con el cual se quiere llevar a cabo el proceso de diagnosticar

esta función utiliza la lista de células infectadas guardadas en el paciente, y crea las simulaciones iterando el numero indicado por el atributo “repeticiones” del paciente, guardando las simulaciones anteriores para luego iterar todos los diagnósticos hechos en busca de un patrón o de que el paciente se haya curado y devuelve si es una enfermedad grave, letal o si se curó por completo, y cuando sucedió. Luego renderiza todos los estados del paciente para mostrar de forma grafica el avance de la enfermedad.

crear_instancia

entrada es un string conteniendo la dirección del archivo de entrada xml en donde está la información acerca de cada paciente, la función crea un nodo Paciente y llena sus atributos para luego ponerla en

la lista global todos_pacientes para luego poder ser seleccionada en el menú de Diagnósticos, en caso de que se tope con un error al cargar los datos, la función dejará de ejecutarse e imprimirá en pantalla que hubo un error al intentar cargar los datos, para luego redirigir al usuario al menú principal.

cancelar_instancia

pac se refiere al numero de paciente seleccionado, esta función revisa si el paciente ya ha sido diagnosticado y en caso de que no, utiliza la función procedimiento para conseguir todas las iteraciones del diagnóstico, con los resultados añade al usuario al archivo de salida y cambia el estado en el nodo Paciente a ya reportado para que si lo vuelven a escoger indique que se encuentra listo para ser reportado.

cargar_config

Método que se encarga de imprimir el menú de selección de paciente para diagnosticar, tras hacer la selección llama al método aux_Diagnostico para procesar los datos y al terminar notifica y redirige al menú principal.

cargar_consumo

Escribe la variable global Pronostico_medico, la cual contiene el xml con toda la información de las simulaciones realizadas, en un archivo llamado prueba.xml. en caso de que exista un error en la ejecución, mostrara en pantalla que el reporte no pudo ser realizado para proceder al menú principal

Facturar

función que se encarga de manejar el menú, mostrarlo en pantalla y procesar la selección del usuario.

detalle_factura

Método implementado con el objetivo de añadir a la lista de pacientes global, Pacientes con parámetros aleatorios con el objetivo de comprobar la funcionalidad del código, para utilizarla se debe escribir “rand” en el menú de carga de archivo.

analisis_cat

entrada es un string conteniendo la dirección del archivo de entrada xml en donde está la información acerca de cada paciente, la función crea un nodo Paciente y llena sus atributos para luego ponerla en

la lista global todos_pacientes para luego poder ser seleccionada en el menú de Diagnósticos, en caso de

que se tope con un error al cargar los datos, la función dejará de ejecutarse e imprimirá en pantalla que hubo un error al intentar cargar los datos, para luego redirigir al usuario al menú principal

analisis_rec

Método implementado con el objetivo de añadir a la lista de pacientes global, Pacientes con parámetros aleatorios con el objetivo de comprobar la funcionalidad del código, para utilizarla se debe escribir “rand” en el menú de carga de archivo.

get_facturas

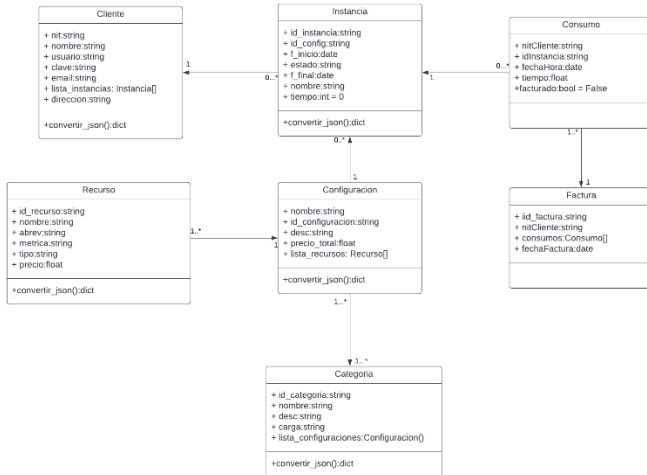
función que se encarga de manejar el menú, mostrarlo en pantalla y procesar la selección del usuario.

Conclusiones

La herramienta graphviz es utilizada para crear graficas de gran fidelidad y precisión, sin embargo, aunque posee limitaciones en cuanto a la libertad al graficar y acomodar la disposición de los objetos en las gráficas, puede dar resultados impresionantes.

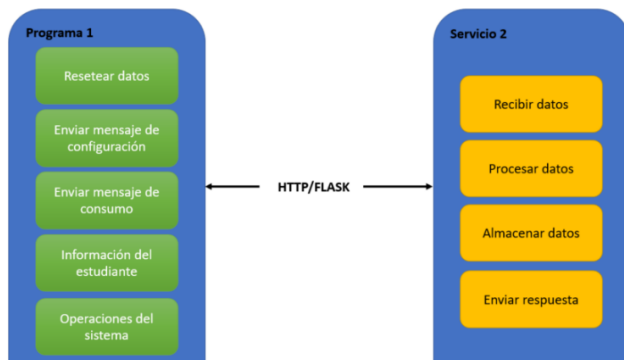
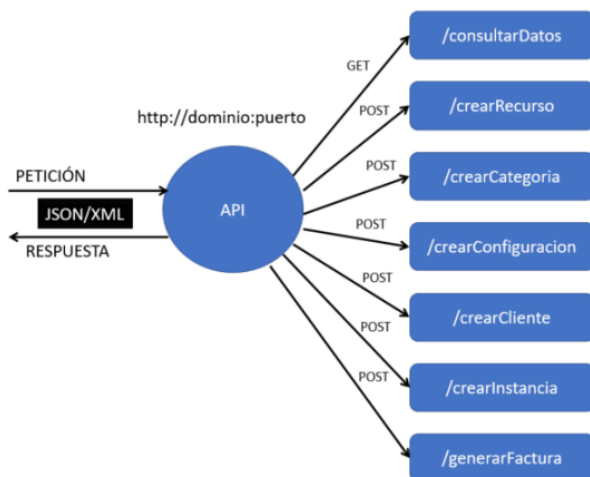
El utilizar listas enlazadas y trabajar con orientación a objetos da control y libertad al programador para poder incorporar sus propias funciones y manipular la información de manera que sea conveniente para trabajar.

Diagramas



Shaffer, Clifford A. *A Practical Introduction to Data Structures and Algorithm Analysis* (1998). NJ: Prentice Hall. ISBN 0-13-660911-2, pp. 77–102.

Wilkes, Maurice Vincent (1964). An Experiment with a Self-compiling Compiler for a Simple List-Processing Language. *Annual Review in Automatic Programming* 4, 1. Published by Pergamon Press.



Referencias bibliográficas

C. J. 31/8/2022, (1991). *An introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.