Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Estructuras de datos, Vacaciones del 2do. Semestre 2022.

Proyecto 1: Muster cloud

3023673810101 – David Abraham Noriega Zamora

Resumen

El Objetivo de este proyecto es utilizar los conocimientos acerca de estructuras de datos, ordenamientos y javascript, así como su correcta visualización por medio de graphviz con ayuda del desarrollo web.

Es por este motivo que se ha desarrollado un software web que implementa la ejecución de usuarios y su flujo por la aplicación al esta ser un servicio web de música, los usuarios pueden crear playlist, agregar amigos y cargar sus propias canciones.

Palabras clave

Lista enlazada, árbol binario, matriz dispersa, simulación, graphviz, estructuras de datos, javascript.

Summary

The objective of this project is to use the knowledge about data structures, ordering and javascript, as well as its correct visualization through graphviz with the help of web development.

It is for this reason that web software has been developed that implements the execution of users and their flow through the application. Being a music web service, users can create playlists, add friends and upload their own songs.

Keywords

Linked list, binary tree, sparse matrix, simulation, graphviz, data structures, javascript.

Introducción

El software de simulación fue desarrollado utilizando el lenguaje de programación Javascript, 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 en javascript con html y consta de 4 pantallas con las cuales el usuario podrá interactuar:

- Login.
- Registro.
- Usuarios.
- Administradores.

Login

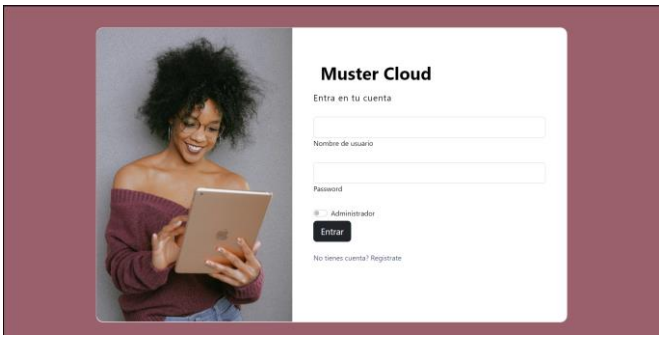

Figura 1. Login.

Fuente: elaboración propia.

Si el usuario está registrado, puede acceder a la aplicación utilizando su nombre de usuario y su contraseña, la cual estará encriptada en el servidor, en caso de ser administrado y querer acceder a esas funcionalidades, puede hacerlo cambiando el slider de administrador.

Registro

Figura 2. registro.

Fuente: elaboración propia.

En caso de que el usuario no se haya registrado aun, se le proporcionará un formulario que podrá llenar para crear su cuenta personal.

Usuarios

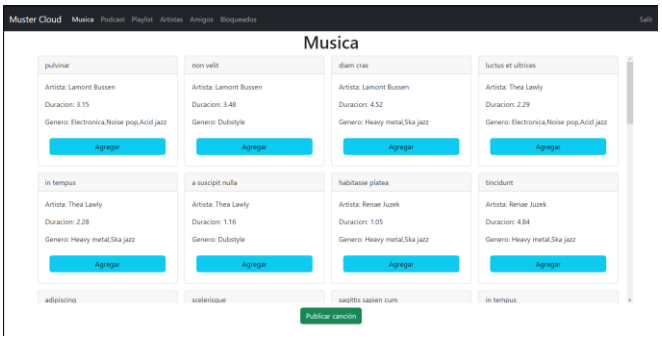

Figura 3. Usuarios.

Fuente: elaboración propia.

Al entrar como un usuario normal, se encontrará con la aplicación, en donde podrá crear una playlist, agregar amigos, bloquear usuarios, publicar canciones propias y crear podcasts.

Administradores

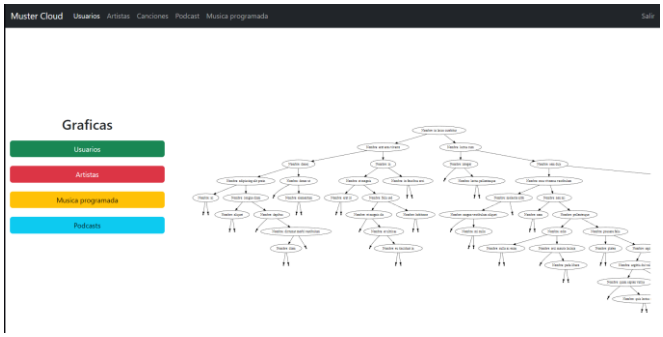

Figura 3. Administradores.

Fuente: elaboración propia.

Al entrar como un administrador, el usuario podrá visualizar la graficas representantes de las estructuras de datos y la información cargada en la aplicación, además, contará con la opción de hacer carga masiva de todos los datos que maneja la aplicación.

Funcionamiento Técnico del código

Estructuras de datos utilizadas

Lista simple enlazada

Utilizada para almacenar a los usuarios al crearlos, utiliza un nodo con un apuntador al siguiente.

Lista circular doblemente enlazada

Utilizada para almacenar las canciones en la playlist de cada usuario en especifico.

Lista de Listas

Utilizada para almacenar a los artistas y sus respectivas canciones, para poder acceder a ellas de manera ordenada, pudiendo ser ordenada de manera Quicksort o burbuja.

Pila

Utilizada para almacenar a los amigos que ha agregado el usuario en sesión.

Cola

Utilizada para almacenar a quienes el usuario en sesión ha bloqueado.

Matriz dispersa

Utilizada para almacenar las canciones programadas, almacena en base al mes, y luego el día, en caso de que una canción se suba el mismo día que otra, esta será reemplazada.

Funciones

Acceso()

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.

sig_estado(paciente_act,est_actual)

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.

posibles_contagios(paciente_act, cell, posibles)

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.

procedimiento(paciente)

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.

registro ()

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.

g_usuarios ()

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_us_artistas ()

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.

Reporte()

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

logout ()

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

cargar_us_musica ()

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.

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.

Referencias bibliográficas

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

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.