

Proyecto

Orquestador de Redes de conectividad

1. Motivación

Una de las funciones más importantes de un Orquestador de red es encontrar un camino entre un par de Routers o dispositivos de red, tales como camino entre “Router A” y “Router B” o “Dslam Edge” y “BRAS”. Este proyecto requiere que implemente una simplificación de esta herramienta de manera que se gestionen datos asociados a Dispositivos de red, su direccionamiento, rutas existentes y tipo de conexión Fibra o 5G.

2. Requisitos de la aplicación

El programa debe ofrecer las siguientes funcionalidades:

- Cargar un archivo con datos asociados a Dispositivos y rutas.
- Encontrar la ruta más corta entre dos Dispositivos dado sus Hostnames, sus Dirección IP o combinaciones de estas.
- Eventualmente informar que no hay una ruta que conecte un par de Dispositivos, ni directa, ni indirecta.
- Indicar la distancia en saltos entre dos puntos, para una o todas las rutas que los conectan, clasificando por tipo de conexión (fibra óptica, 5G).
- Agregar, editar y eliminar un dispositivo.
- Agregar, editar y eliminar una conexión entre Dispositivos.

La forma en la cual será provista cada una de las funcionalidades será descrita en las siguientes secciones.

3. Requisitos funcionales

El programa debe ser, ante todo, eficiente. Para tal fin, se deben seleccionar, proponer y utilizar las estructuras de datos más eficientes para cumplir los requisitos de la aplicación. Es estrictamente obligatorio el uso de las siguientes estructuras, las cuales pueden ser utilizadas incorporándolas en la definición de Tipos de Datos Abstractos (TDA) de Clases (POO):

- | | | |
|---------------------------------|----------------------------------|------------------|
| • Arreglo (estático y dinámico) | • Registro (estático y variante) | • Lista enlazada |
| • Cola | • Grafos | • Árbol |
| | | • Archivo |

Ejemplos del uso de las estructuras:

- **Grafo con lista de adyacencia**¹: puesto que un grafo de Dispositivos puede ser bastante disperso², es necesario que se optimice el uso de memoria, por lo tanto, se debe guardar por cada dispositivo sólo la lista de sus conexiones y no todas las conexiones posibles. En pocas palabras, se debe evitar una implementación de grafo basada en matriz de adyacencia³. La lista de adyacencia puede ser implementada bien, como una lista doblemente enlazada o como un arreglo dinámico.

¹ http://es.wikipedia.org/wiki/Lista_de_adyacencia https://es.frwiki.wiki/wiki/Liste_d%27adjacence

² http://es.wikipedia.org/wiki/Grafo_denso

³ http://es.wikipedia.org/wiki/Matriz_de_adyacencia

- **Árbol binario de búsqueda:** para mejorar el desempeño de la búsqueda de Dispositivos y la información asociada, se requiere que implemente un árbol de búsqueda binario y algunas funciones asociadas, en especial, la eliminación perezosa ⁴ (Joyanes, 2006, p. 748).
- **Cola:** para la búsqueda de la ruta más corta entre dos Dispositivos, se debe implementar el **Algoritmo de Búsqueda Primero en Anchura**⁵ o BFS por sus siglas en inglés. Para ello, se debe implementar la estructura de datos **Cola** (Joyanes, 2006, p. 670).

Si selecciona programar la solución mediante Programación Orientada a Objeto, se sugiere que se haga uso de herencia para implementar la **clase Cola**; específicamente que herede de la **clase Lista**.

4. Formato del archivo de Dispositivos

El archivo que contiene la información de los Dispositivos y las relaciones entre ellos, contendrá la siguiente información:

- Una lista de Dispositivos, indicando su hostname y su dirección IP expresada en el [formato correspondiente](#). El hostname de cada dispositivo será único y estará conformado por caracteres alfabéticos en minúscula y mayúsculas, sin incluir espacios y signos.
- Una lista de relaciones entre Dispositivos y la distancia, en saltos, que existe entre los Dispositivos.

Dispositivos: La segunda línea de este archivo contendrá un entero L indicando la cantidad de Dispositivos existentes. Las siguientes L líneas tendrán el siguiente formato:

<hostname>, <dirección IP>

Ejemplos:

HUA_LABC_910_14, 172.19.88.15

HUA_CATF_910_1, 172.23.244.144

HUA_HECH_910_1, 172.23.242.187

HUA_COGB_910_1, 172.23.240.244

HUA_AROB_910_1, 172.18.88.242

HUA_PJOE_910_1, 172.18.88.247

Relaciones: seguidamente, la línea L+2 contendrá un entero R indicando la cantidad de relaciones existentes entre los Dispositivos. Cada una de las siguientes R+1 líneas tendrá el siguiente formato con información de las relaciones:

<hostname1>, <hostname2>, <cantidad de saltos>, <ping>, <tipo_conexión>

Ejemplos:

HUA_LABC_910_14, HUA_CATF_910_1, 30, 38, Fibra óptica

HUA_COGB_910_1, HUA_LABC_910_14, 10, 913; Aéreo

HUA_CATF_910_1, HUA_COGB_910_1, 15, 23, Fibra óptica

HUA_AROB_910_1, HUA_PJOE_910_1, 30, 36, Fibra óptica

Se garantiza que no habrá relaciones duplicadas.

⁴ Descrito en (Sedgewick, 1995, p. 233), consiste en marcar el nodo del árbol como eliminado, sin eliminarlo físicamente.

⁵ http://es.wikipedia.org/wiki/B%C3%BAsqueda_en_anchura

El archivo se llamará “*Dispositivos.dat*” y deberá ser cargado SIEMPRE al iniciar el programa. El **Anexo 1** contiene un ejemplo del archivo “*Dispositivos.dat*”, el cual será suministrado con datos de prueba posteriormente.

4.1. Límites

$$1 < L < 1000$$

$$1 < R < 500$$

Por cada dispositivo, se tendrá a lo sumo 12 conexiones con otros Dispositivos. El hostname de cada dispositivo no excederá los 48 caracteres y siempre será mayor o igual a 3 caracteres.

5. Interacción con el usuario

5.1. Entrada y Funcionalidades

Luego de haber cargado el archivo con la información de los Dispositivos y las relaciones entre ellas, se deben mostrar menús con las opciones observadas en la Tabla 1:

Tabla 1. Funcionalidades del programa

PRINCIPAL	Para AGREGAR información	Para ELIMINAR información
<i>Identificación de la Universidad</i> <i>Facultad</i> <i>Escuela</i> <i>Identificación del proyecto</i> Menú de opciones: 1. Agregar información. 2. Eliminar información. 3. Buscar y listar. 4. Mostrar respaldos. 5. Créditos. 6. Salir de la aplicación.	<i>Identificación de la Universidad</i> <i>Facultad</i> <i>Escuela</i> <i>Identificación del proyecto</i> Menú de opciones: 1. Agregar dispositivo. 2. Agregar ruta. 3. Volver a Principal. 4. Salir de la aplicación.	<i>Identificación de la Universidad</i> <i>Facultad</i> <i>Escuela</i> <i>Identificación del proyecto</i> Menú de opciones: 1. Eliminar dispositivo. 2. Eliminar ruta. 3. Volver a Principal. 4. Salir de la aplicación.
Para BUSCAR Y LISTAR información		Para RESPALDAR información
<i>Identificación de la Universidad</i> <i>Facultad</i> <i>Escuela</i> <i>Identificación del proyecto</i> Menú de opciones: 1. Consultar dispositivo. 2. Listado de Dispositivos. 3. Buscar ruta (1: 5G, 2: fibra óptica, 3: ambas). 4. Dispositivos adyacentes. 5. Volver a Principal. 6. Salir de la aplicación.		<i>Identificación de la Universidad</i> <i>Facultad</i> <i>Escuela</i> <i>Identificación del proyecto</i> Menú de opciones: 1. Listado Dispositivos existentes. 2. Listado Dispositivos eliminados. 3. Listado rutas eliminadas. 4. Volver a Principal. 5. Salir de la aplicación.

A continuación se detallará cómo debe ser la entrada por cada una de las opciones.

- **Agregar dispositivo**

Sigue el mismo formato que el archivo *“Dispositivos.dat”*. No se aceptan Dispositivos con nombre repetidos en el sistema. Se debe validar que la información suministrada corresponda al formato correcto, es decir:

<hostname>, <dirección IP>

Ejemplo:

HUA_PJOE_910_1, 172.23.244.144

Respuesta del sistema:

El dispositivo HUA_PJOE_910_1 ha sido agregado.

Ejemplo 2:

JUN-STGO-BRAS-2, 172.23.255.66

Respuesta del sistema:

El dispositivo JUN-STGO-BRAS-2 ha sido agregado.

- **Eliminar dispositivo**

Como cada dispositivo es único, sólo se necesita indicar el nombre del dispositivo a eliminar. Formato:

<hostname>

Ejemplo: 2

JUN-STGO-BRAS-2

Respuesta del sistema:

El dispositivo JUN-STGO-BRAS-2 fue eliminado del sistema.

NOTA: Previo a la eliminación, se respaldan los datos del dispositivo en el archivo *Dispositivos_resp.dat*

- **Agregar relación o ruta**

Se indica los nombres de las ciudades, la distancia en kilómetros, el tiempo estimado del viaje (hh:mm) y el tipo de recorrido (fibra óptica, 5G) de la ruta. Se mantiene el formato del archivo *“Dispositivos.dat”*:

<hostname1>, <hostname2>, <Saltos>, <Ping>, <tipo_recorrido>

Respuesta del sistema:

Se agregó una ruta entre <hostname1> y <hostname2>.

- **Eliminar relación**

Se suministra los nombres o las direcciones IP de los Dispositivos cuya ruta (directa o indirecta) será eliminada. Se eliminan TODAS las rutas existentes entre el par de Dispositivos indicados. Se mantiene

el formato del archivo “*Dispositivos.dat*”:

<hostname1>, *<hostname2>*
<Dirección IP 1>, *<Dirección IP 2>*

Respuesta del sistema:

Entre los dispositivos *<hostname1>* y *<hostname2>* se eliminaron las siguientes rutas:

// Listado de todas las relaciones existentes entre las ciudades indicadas, sean directas
// o indirectas, fibra ópticas o 5Gs.

Total de rutas eliminadas: *<cantidad_rutas_entre_dispositivo1_y_dispositivo2>*

NOTA: Previo a la eliminación, se respaldan los datos de la ruta o las rutas existentes entre los Dispositivos en el archivo *rutas_resp.dat*

- **Consultar dispositivo:** se busca un dispositivo dado su nombre (N) o sus coordenadas (C), y se muestran todos sus datos según el formato acordado:

<hostname>, *<Dirección IP>*

- **Listado de Dispositivos:** se muestran los Dispositivos en orden alfabético ascendente

<hostname1>, *<Dirección IP>*

<hostname2>, *<Dirección IP>*

...

<hostnameL>, *<Dirección IP>*

- **Buscar ruta o rutas**

Se indicarán los nombres o las coordenadas de los Dispositivos, además del tipo de recorrido (1: 5G, 2: fibra óptica, 3: ambas), se busca y se lista la ruta o rutas existentes.

Formato:

Indique las ciudades:

// opciones:

<hostname1>, *<hostname2>*

<Dirección IP 1>, *<Dirección IP 2>*

< Dirección IP 1>, *<hostname2>*

<hostname1>, *< Dirección IP 2>*

Indique el tipo de ruta:

// opciones:

$op \in \{ "1", "2", "3" \}$ // 1: 5G, 2: fibra óptica, 3: ambas.

Respuesta del sistema:

Entre los dispositivos **<hostname1>** y **<hostname2>** existen estas rutas de tipo **<tipo_recorrido>**:

// Listado de todas las relaciones o rutas existentes entre los dispositivos indicados,
// sean directas o indirectas, según el tipo de recorrido indicado por el usuario.

Total de rutas encontradas: *<cantidad_rutas_entre_dispositivo1_y_dispositivo2>*

NOTA: Eventualmente se debe indicar al usuario que no hay ninguna ruta entre las ciudades.

- **Ciudades adyacentes**

Dado el nombre de una ciudad indicar todas las ciudades que tienen una ruta (relación) directa con ella.

Formato:

<hostname>

< Dirección IP>

Respuesta del sistema:

Los dispositivos adyacentes a <hostname> son:

// Listado de todas las ciudades con relación directa, distancia, duración y tipo de recorrido.

<hostname21>, *<Saltos>*, *<Ping>*, *<tipo_recorrido>*

<hostname22>, *<Saltos>*, *<Ping>*, *<tipo_recorrido>*

...

<hostname2n>, *<Saltos>*, *<Ping>*, *<tipo_recorrido>*

Total de ciudades adyacente: *<cantidad_de_adyacencias>*

- **Listado de Dispositivos existentes**

Muestra en orden alfabético los Dispositivos que siguen registradas en el sistema, es decir, aquellos que se encuentran en el archivo *Dispositivos.dat*

- **Listado de Dispositivos eliminados**

Se muestra el contenido del archivo de respaldo *Dispositivos_resp.dat*

- **Listado de rutas eliminadas**

Se muestra el contenido del archivo de respaldo *rutas_resp.dat*

- **Salir:** termina el programa.

5.2. Salida

La salida será efectuada a través de *stdout* y un archivo llamado “*resultados.out*”. La salida efectuada a través de *stdout* puede contener cualquier formato, sin embargo, la salida producida en el archivo “*resultados.out*” debe apegarse al siguiente formato:

- **Agregar dispositivo**

Se debe imprimir una línea indicando el resultado de la operación:

- “1” para operación exitosa.
- “0” en caso contrario. La operación puede resultar infructuosa en caso de que ya exista un dispositivo con el nombre proporcionado.

- **Eliminar dispositivo**

Se debe imprimir una línea indicando el resultado de la operación:

- “1” para operación exitosa.
- “0” en caso contrario. La operación puede resultar infructuosa en caso de que no exista un dispositivo con el nombre proporcionado.

- **Agregar o Eliminar relación o ruta**

Se debe imprimir una línea indicando el resultado de la operación:

- “1” para operación exitosa.
- “-3” si el dispositivo origen no existe.
- “-2” si no existe el dispositivo destino.
- “-1” si no existen ambas Dispositivos.
- “0” si ya existe una relación entre los Dispositivos, en el caso de agregar.
- “0” si no existe una relación entre los Dispositivos, en el caso de eliminar.

- **Buscar una ruta dado el nombre o las coordenadas**

Se debe imprimir una línea indicando la ruta, la cual será una lista de Dispositivos separados por comas, incluyendo el origen y el destino. La sintaxis es la siguiente:

<origen>, <dispositivo_intermedio1>, ... <dispositivo_intermedioN>, <destino>

En caso de no ser posible la operación, se debe imprimir la razón, siguiendo el siguiente formato:

- “-3” si el dispositivo origen no existe.
- “-2” si el dispositivo destino no existe.
- “-1” si no existen ambos Dispositivos.
- “0” si no existe una relación entre los Dispositivos o no existe un camino que los conecte.

En los **Anexos 2, 3 y 4** se muestran ejemplos de los archivos de salida y de respaldo:

- *resultados.out*

- *Dispositivos_resp.dat*
- *rutasp_resp.dat*

7. Requisitos no funcionales

Todo el código fuente debe estar documentado, preferiblemente siguiendo el estilo de documentación de Doxygen. El orden y estilo de programar será tomado en cuenta para la revisión. Consulte referencias de estilo en (Google, s.f.) y (Doxygen, 2022).

8. Aclaratorias y Restricciones del Proyecto

- Usted es libre de seleccionar la(s) estructura(s) de datos estáticas o dinámicas más convenientes para el registro, manipulación y respaldo de la información del proyecto. Sin embargo debe incluir al menos implementaciones de las siguientes:
 - Arreglo (estático y dinámico) • Registro (estático y variante) • Lista enlazada
 - Cola • Grafos • Árbol • Archivo
- Se deben utilizar las estructuras de datos estáticas o dinámicas que sean más eficientes, evitando reservar innecesariamente memoria que no será utilizada.
- Para la implementación del programa puede utilizar cualquier herramienta de programación contemplada en el temario de la asignatura, incluyendo recursividad y *backtracking*. Puede implementarlo con programación estructurada o programación orientada a objetos.
- Puede utilizar interfaz por consola o interfaz gráfica.
- El proyecto deberá ser desarrollado en lenguaje C++ y debe poder ejecutarse en entorno Linux o Windows. Por lo tanto investigue y seleccione un compilador que funcione bien en cualquier de los sistemas operativos.
- El proyecto puede ser desarrollado de manera individual o en equipos de dos estudiantes.
- Se deben realizar las validaciones necesarias para garantizar un comportamiento robusto del programa.
- La aplicación debe poder leer datos desde archivos de entrada y respaldar en archivos de salida.
- La aplicación deberá cargar **OBLIGATORIAMENTE** los archivos de entrada al momento de iniciar el programa. En estos archivos se va a incluir todos los datos iniciales de los atletas, sedes, disciplinas y competencias.
- Los cambios realizados deberán ser reflejados en archivos de salida.
- Archivos con virus tendrá cero (0) puntos de nota.
- En caso de no compilar, el proyecto no será revisado y su calificación será de cero (0) puntos.
- Las copias serán penalizadas con la anulación de los proyectos involucrados, asignando una calificación de cero (0) puntos.
- Asistir al interrogatorio es requisito para la revisión y calificación de la parte 2.
- La entrega de los proyectos, es condición para presentar el examen de Reparación. Se entiende por entrega del proyecto el envío del informe de análisis y del código ejecutable del programa.
- Si tiene dudas sobre el enunciado puede consultar a los profesores de la asignatura por correo electrónico.

ENTREGAS

Se envía un archivo comprimido con todos los archivos que conforman la solución en cada etapa al correo electrónico:

- Prof. Marcel Isaac Oviedo <marcelisaac.ucv@gmail.com>

Utilizar para el asunto y el archivo anexo comprimido el formato:

AyED_Proyecto1_NombreApellido_NumeroCedula
AyED_Proyecto1_NombreApellido_NumeroCedula.ext

Por ejemplo:

AyED_Proyecto1_LuisCastillo_23000000 (un integrante)
AyED_Proyecto1_LuisCastillo_23000000_MirnaRivas_24500000 (dos integrantes)
AyED_Proyecto1_LuisCastillo_23000000.rar (anexo con los archivos del programa)

Parte 1: Informe con el análisis y diseño de la solución

Fecha: deberá ser entregado a más tardar el día viernes 21 de abril de de 2023 hasta las 11:59 P. M.

Calificación parte 1 del proyecto: 20% de la nota.

Incluye:

- Un breve análisis del problema y descripción de la solución propuesta.
- La identificación de las estructuras de datos a utilizar, tanto estáticas como dinámicas, explicando para qué serán utilizadas o una breve justificación de su selección.
- La especificación en pseudo-código de las estructuras de datos y su representación gráfica.
- Solución en pseudo-código para los siguientes requerimientos:
 1. Agregar relación o ruta.
 2. Eliminar dispositivo.
 3. Listado de Dispositivos
 4. Ciudades adyacentes.
 5. Listado Dispositivos eliminadas

Parte 2: Programación, prueba e informe final

Fecha: deberá ser entregado a más tardar el día **viernes 2 de Junio de 2023** hasta las 11:59 P. M.

Calificación: 60% Programación, prueba e interrogatorio + 20% Informe final.

Incluye:

Código ejecutable del programa con la solución, el cual será probado mediante la lectura de los archivos de entrada, la verificación de las opciones del programa, la revisión de archivos de salida y de respaldo.

El informe final debe incluir:

- Descripción del proceso de desarrollo del programa.
- Cambios o modificaciones respecto al análisis y diseño de planteados en la primera parte.
- Identificación del compilador utilizado
- Dificultades, si se presentaron, y cómo fueron resueltas.

Elementos del informe:

- Portada, identificando a la universidad, facultad, escuela, asignatura, título, identificación del estudiante(s) con nombre completo y número de cédula de identidad, fecha de entrega.
- Índice

- Contenido, según lo solicitado en cada fase.
- Referencias bibliográficas si se consultaron o usaron fuentes externas.
- Número de página. Configuración: tamaño carta, fuente 10-11 puntos, espacio simple.

Referencias Bibliográficas y Digitales

Doxygen. (2022). *Documenting the code*. Recuperado de <https://www.doxygen.nl/manual/docblocks.html>

Google. (s.f.). *Google C++ Style Guide*. Recuperado de <https://google.github.io/styleguide/cppguide.html>

Joyanes, L. (2006). *Programación en C++ - Algoritmos, estructuras de datos y objetos*. Recuperado de <http://biblioteca.univalle.edu.ni/files/original/ec89128132c7434f1765ceb9cbf1160828ae85f5.pdf>

Sedgewick, R. (1995). *Algoritmos en C++*, 1995.

Prof. Marcel Isaac Oviedo

marcelisaac.ucv@gmail.com

Grupo Docente Algoritmos y Estructuras de Datos

Ciudad Universitaria, 16 de marzo de 2023