

Desarrollo del método de ingeniería.

1. Identificación del problema.

- La empresa necesita un sistema de software que de momento debe ser capaz de soportar un par de requerimientos relacionados con su nuevo servicio en desarrollo.
- El sistema debe permitir al usuario representante de la empresa registrar una nueva ciudad a medida que la empresa va ganando clientes en nuevas ciudades.
- Se requiere que al momento de enviar un equipo a hacer un trabajo en otra ciudad, el sistema muestre la ruta con la distancia mínima y la cantidad de kilómetros que se necesitan para llegar al destino objetivo.
- Se requiere que el sistema pueda registrar los equipos de hardware que se deben conectar cuando el usuario está registrando el detalle de un servicio.
- El sistema debe poder calcular las conexiones más adecuada entre los dispositivos de hardware de modo que todos estén conectados y se gaste la menor cantidad de cable posible.

Definición del problema

La empresa de técnicos necesita un sistema de software capaz de gestionar de forma adecuada un par de funciones relacionadas con un nuevo servicio que desean ofrecer a compañías de todo el país. Dichas funciones serán: el envío de un equipo de trabajo de forma que gasten menos kilómetros y el cálculo y diseño de la ramificación que deberá haber entre los equipos de hardware de la compañía que esté recibiendo el servicio en un momento dado.

2. Recopilación de información

Definiciones y fuentes:

Red de computadoras:

Una red de computadoras, red de ordenadores o red informática es un conjunto de equipos nodos y software conectados entre sí por medio de dispositivos físicos que envían y reciben impulsos eléctricos, ondas electromagnéticas o cualquier otro medio para el transporte de datos, con la finalidad de compartir información, recursos y ofrecer servicios.

https://es.wikipedia.org/wiki/Red_de_computadoras

Árboles de expansión mínima:

El árbol de expansión de peso mínimo es aquel que comienza desde un vértice y encuentra todos sus nodos accesibles y las relaciones en conjunto que permiten que se conectan dichos nodos con el menor peso posible.

<https://www.grapheverywhere.com/arbol-de-expansion-de-peso-minimo/#:~:text=El%20%C3%A1rbol%20de%20expansi%C3%B3n%20de,con%20el%20menor%20peso%20posible.>

3. Búsqueda de soluciones creativas

- a. Podría hacer que todas las ciudades se guarden en un hashMap donde la clave sería el nombre de la ciudad y debería estar asociada a la distancia que hay desde Cali hasta la correspondiente ciudad.
De esta forma, al momento de calcular la distancia entre Cali y otra ciudad, me ahorraría las complejidades mayores a uno. Puesto que, solo debería retornar la distancia directa entre Cali y la ciudad objetivo.
- b. Podría hacer que todas y cada una de las ciudades estén contenidas dentro de un grafo como vértices del mismo. De esta forma, podría guardar las distancias entre una ciudad y otras de ellas como aristas.
Siguiendo lo anterior, podría hacer rutas más complejas y elaboradas y evitar ir siempre directamente desde Cal hasta otra ciudad.
- c. Al momento de registrar un proyecto, cada equipo de hardware será almacenado en una matriz donde cada uno de sus espacios también sería un metro cuadrado de la zona donde se va a construir el sistema interconectado. Así, el sistema internamente podrá modelar el terreno donde van a trabajar los empleados y hacer cálculos teniendo en cuenta la distancia que hay entre cada uno de los equipos de hardware y su ubicación exacta en el terreno para obtener la distribución mínima de cable que se necesita a partir de sus posiciones en el terreno.
- d. Al momento de registrar un proyecto, cada uno de los equipos de cómputo será guardado como el vértice de un grafo y de aquí se podrá guardar también la distancia en cable que hay entre uno y varios de los computadores. En el momento en el que estén todos los equipos de hardware registrados, se usará el algoritmo de árbol de expansión mínima propio de la estructura de datos que se está usando para obtener los pesos y las conexiones idóneas para que todo quede interconectado de forma que se gaste el mínimo cable posible.
- e. La interfaz de usuario podría ser mediante la consola. Dicha interfaz tendría los menús necesarios para llegar a todas las funcionalidades y podría imprimir los resultados esperados.
- f. La interfaz de usuario podría ser modelada mediante java fx de modo que se trate de emular gráficamente en un mapa como se verían los nodos y sus conexiones con lo demás, tanto para las ciudades como para uno de los servicios.

4. Transición de formulación de ideas a diseños preliminares

Tras el análisis de las ideas planteadas puedo hablar de dos modelos bastante factibles que podrían dar una solución efectiva al problema planteado.

-El primer modelo sería un sistema con interfaz de usuario mediante la consola. Dicho sistema, almacenará las ciudades donde la empresa en cuestión ofrece el servicio, como vértices de un grafo. De esta forma, se podrán guardar las distancias entre las ciudades como el peso de las aristas y se podrá calcular la ruta y la distancia mínima entre la ciudad de origen, Cali, y la ciudad donde se enviará el equipo de trabajo. Además, este sistema también guardaría los equipos de cómputo de un servicio dentro de grafos con el objetivo de poder calcular la mejor distribución posible mediante un árbol de expansión mínima.

-El segundo modelo, sugiere un sistema con una interfaz hecha usando Java FX. Dicho sistema, guardará todas las ciudades donde la empresa ofrece el servicio mediante un hashMap, donde la clave será la ciudad y el elemento guardado será la distancia entre dicha ciudad y Cali. Además, en este sistema cada servicio se modelaría en una matriz donde cada espacio sería un metro cuadrado del terreno donde se está trabajando. Esto para que se pueda modelar correctamente el espacio y así generar los cálculos pertinentes para buscar la distancia mínima entre un equipo de hardware y otro.

5. Evaluación y selección de la mejor solución

Criterios

- Criterio A. Exactitud. La solución responde de forma correcta a todos los eventos a los que puede ser sometido.

[2] La solución responde de forma precisa (Se prefiere una solución precisa)
[1] La solución responde de forma imprecisa
- Criterio B. Espacio en memoria. La solución debe usar la menor cantidad posible de espacio en memoria.

[3] La solución usa la menor cantidad posible de atributos y de estructuras (Se prefiere que la solución use la menor cantidad de estructuras y atributos posibles)

[2] La solución usa más atributos y estructuras que los mínimamente necesarios para dar solución a los eventos planteados.

[1] La solución usa muchos más atributos y estructuras de los estrictamente necesarios.
- Criterio C. Facilidad en implementación algorítmica

[3] Su implementación se torna intuitiva y se puede emular de “forma visual” fácilmente. (Se prefiere que su implementación algorítmica sea intuitiva para que el equipo de desarrollo lleve el proyecto a calidad de forma mas rapida)

[2] Su implementación se torna manejable pero no es fácil de emular de “forma visual”.

[1] Su implementación se torna muy compleja de implementar y es muy difícil de emular de “forma visual”.

- Criterio D. Emulación correcta del modelo de servicio ofrecido por la empresa.

[2] El sistema simula correctamente el modelo de servicio ofrecido por la empresa.

(Es preferible que sea de esta forma para garantizar el funcionamiento correcto del sistema se software)

[1] El sistema simula no simula de forma adecuada el modelo de servicio ofrecido por la empresa.

Evaluación

| | Criterio A | Criterio B | Criterio C | Criterio D | Puntos totales |
|----------------|------------|------------|------------|------------|----------------|
| Primer modelo | 2 | 3 | 3 | 2 | 10 |
| Segundo modelo | 1 | 3 | 2 | 1 | 7 |

Selección

Siguiendo los resultados obtenidos en el análisis anterior, se debe llevar a producción la primera solución. Puesto que, tiene más puntos y esto significa que atiende mejor a los criterios de calidad planteados.

6. Preparación de informes y especificaciones

Hemos decidido plantear en este paso las tablas de análisis de requerimientos para cada una de las funcionalidades que debe tener el software y el diseño de la La solución será presentado como un diagrama de clases. Para ver estos elementos, ver los archivos **ClassDiagram** y **RequirementAnalysis** en la carpeta doc del proyecto donde también se encuentra este documento.

7. Implementación del diseño

Lista de tareas a implementar:

- a. Construcción de todas las estructuras de datos necesarias para solucionar los eventos planteados.
- b. Inserción de todas las clases planteadas en el modelo del sistema.
- c. Implementación de todos los menues en la interfaz de usuario
- d. Implementación de las funcionalidades del modelo
- e. Conexión del modelo del sistema y la interfaz de usuario mediante la controladora