Camilo vivas
Jaime Cardona
Felipe Garcia

A. Identificación del problema

a. Contexto del problema

La empresa Pródigos App, surge como un emprendimiento hace dos años, la cual ha tenido un crecimiento acelerado. Se ha podido constituir en el mercado Colombiano operando en las principales cuatro ciudades del País: Cali, Bogotá, Medellín y Barranquilla. Pródigos es una plataforma digital la cual consta de dos servicios:

1. Cuidado y acompañamiento: Pródigos gestiona servicios de cuidado y acompañamiento a adultos y adultos mayores a través de su plataforma en la cual conecta familias que requieren servicios de cuidado y acompañamiento para sus seres queridos con personal experto de manera segura, fácil y confiable.

2 Trámites médicos: Como una línea complementaria al servicio de cuidado y acompañamiento surge la línea de trámites médicos, en el cual mensajeros propios especializados realizan todo tipo de diligencia médicas cómo autorización de citas, procedimientos, autorización de medicamentos en las EPS y recolección de medicamentos en farmacias autorizadas. Este es un excelente servicio pues le ahorra a los clientes en promedio de 3 a 4 horas por trámite. Este servicio se ofrece en formato B2C y B2B. Durante el periodo de la pandemia este servicio ha tenido un crecimiento del 1000%, logrando cerrar varias alianzas comerciales con grandes aseguradoras y fondos de empleados del país.

b. Situación problema

Pródigos, teniendo como una de las principales líneas de negocio los trámites médicos y la entrega última milla de medicamentos de entidades a sus pacientes. Se enfrenta al problema de poder organizar y ejecutar las rutas de manera eficaz y efectiva respecto al tiempo para cada mensajero, cuyo objetivo es poder dar con el cumplimiento a todos los pedidos diarios en las distintas ciudades del país.

c. Caracterización del proceso del proceso de trámites médicos:

i. **B2C:** El proceso empieza con la recepción de solicitud del servicio de trámite médico por parte del área de ventas, la cual recepciona toda la información del trámite a realizar, esta información se consigna en el software administrativo. Las jefes de operación con corte cada día 5:30 PM de cada día de lunes a sábado hacen enrutamiento manual y le asignan ciertos trámites a los mensajeros, dependiendo de: Lugar de vivienda, tipo de trámites y entidades donde se realizan los trámites. Los mensajeros a través de la aplicación móvil recepcionan la información y al día siguiente empiezan la ruta desde las 7:00 de la mañana. La ruta de los mensajeros contempla:

1. Por cada usuario los mensajeros deben recoger la documentación impresa en un solo punto de origen (empresa pródigos de cada ciudad).

Camilo vivas
Jaime Cardona
Felipe Garcia

2. Los mensajeros empiezan la ruta, hacen los trámites médicos y recogen los medicamentos necesarios.
3. Al finalizar todos los trámites, van a cada domicilio a entregar la documentación impresa y el trámite médico realizado.
4. El mensajero finaliza el servicio cuando ha entregado todos los trámites médicos.

ii. **B2B:** Las jefes de operación B2B reciben pedidos a través de una plantilla de excel de las entidades dispensadoras de medicamentos hasta las 5:30 PM de cada día de lunes a sábados. Luego hacen enrutamiento manual y le asignan las rutas a los mensajeros, dependiendo de: Lugar de vivienda del mensajero,dispensador de medicamentos de la entidad y sector de vivienda de los pacientes a entregar medicina. Los mensajeros a través de la aplicación móvil recepcionan la información y al día siguiente empiezan la ruta desde las 7:00 de la mañana. La ruta de los mensajeros contempla:

1. El mensajero asignado a cada entidad va a un dispensador de medicinas para sus pacientes, donde recoge todas las medicinas a entregar.
2. Los mensajeros van punto por punto entregando la medicina a cada paciente.
3. El mensajero finaliza el servicio cuando ha entregado todas las medicinas a los pacientes asignados.

d. Identificación requerimientos funcionales:

I. **R1:** el programa debe permitir el registro de las casas de los clientes con su respectivo nombre e identificacion
II. **R2:** el programa debe permitir el registro de las farmacias
III. **R3:** el programa debe mostras cual es la ruta a hacer
IV. **R4:** el programa debe permitir el registro de los mensajeros
V. **R5:** dar dos opciones para resolver el problema
VI. **R6:** asignar una ruta a cada mensajero
VII. **R7:** el programa debe permitir registrar servicios B2B
VIII. **R8:** el programa debe permitir registrar servicios B2C

B. Recopilación de información

Grafo

Es una composición de un conjunto de objetos conocidos como vértices que se relacionan con otros vértices a través de un conjunto de conexiones conocidas como aristas.

Grafo dirigido

Camilo vivas
Jaime Cardona
Felipe Garcia

Un grafo dirigido consta de un conjunto de vértices y aristas donde cada arista se asocia de forma unidireccional a través de una flecha. En este grafo se manejan dos tipos de arsitas que son las saliente o entrante.

Grafo no dirigido

Los grafos no dirigidos son aquellos que constan de un conjunto de vértices que están conectados a un conjunto de aristas de forma no direccional. Esto significa que una arista puede recorrerse desde cualquiera de los puntos.

(https://www.grapheverywhere.com/que-son-los-grafos/)

Algoritmo de Kruskal

El algoritmo de Kruskal, dado un grafo conexo, no dirigido y ponderado, encuentra un árbol de expansión mínima. En palabras más simples es capaz de encontrar un subconjunto de las aristas que forman un árbol que incluya todos los vértices del grafo inicial, donde el peso total de las aristas del árbol es el mínimo posible.

Funcionamiento

1. Se selecciona, de entre todas las aristas restantes, la de menor peso siempre que no cree ningún ciclo.
2. Se repite el paso 1 hasta que se haya seleccionado |Vertices| - 1 aristas.

(https://sites.google.com/site/complejidadalgoritmicaes/kruskal)

Algoritmo de Prim

El algoritmo de Kruskal, dado un grafo conexo, no dirigido y ponderado, encuentra un árbol de expansión mínima. En palabras más simples es capaz de encontrar un subconjunto de las aristas que forman un árbol que incluya todos los vértices del grafo inicial, donde el peso total de las aristas del árbol es el mínimo posible.

Funcionamiento

1. Se marca un vértice cualquiera, este sera el vértice de partida.
2. Se selecciona la arista de menor peso incidente en el vértice seleccionado anteriormente y se selecciona el otro vértice en el que incide dicha arista.
3. Repetir el paso 2 siempre que la arista elegida no cree ningún ciclo.
4. El árbol de expansión mínima será encontrado cuando hayan sido seleccionados todos lo vértices del grafo.

(https://sites.google.com/site/complejidadalgoritmicaes/prim)

Camilo vivas
Jaime Cardona
Felipe Garcia

C. Soluciones creativas

La primicia para solucionar el problema es que se tiene que hacer mediante grafos, ya que estamos hablando de rutas y de lograr los mejores tiempos, podríamos utilizar un multigrafo que nos calcule la mejor ruta solo tomando en cuenta las vías principales, por ejemplo la autopista, y que las vías pequeñas, una vía de barrio por ejemplo, ya sean de la elección del domiciliario, ya que estas vías no tienen doble sentido (la mayoría) y al estar trabajando con multigrafo normal no le podemos dar dirección, por esa razón esas vías ya son elección del domiciliario, la ventaja de esta solución es que la creación del programa se facilitará más, la desventaja es que no abordaremos todo la ruta es decir puede que el domiciliario tome una mala decisión y el tiempo se afecte. Otra opción que manejamos es implementar un multigrafo dirigido, con este ya podríamos abordar todo el problema ya que podríamos tener en cuenta las vías pequeñas, la ventaja de esto es abordar todo el problemas, es decir podríamos generar toda la ruta para el domiciliario y que este no tenga que tomar decisiones, la desventaja es que la solución del problema sería más denso, ya que tendríamos que implementar más vértices y aristas y además de eso darles dirección.

D. Transición de ideas a diseños preliminares

idea #1 : Se tomarán en cuenta sólo las vías principales sin dirección, se usarán archivos planos para guardar la información de las vías (el grafo), el domiciliario introducirá el punto de partida y el punto de llegada, el programa le mostrará la mejor vía principal que pueda tomar.

idea #2 : Se tendrán en consideración todas las vías es decir tanto principales como secundarias, también se usarán archivos planos para guardar la información , el domiciliario introducirá el punto de partida y el punto de llegada y el programa le mostrara todas las vías que debe tomar (tanto principales como secundarias).

idea#3 : Se crearán los domiciliarios previamente, se crearán un mapa propio con un plano cartesiano, para asi tener mas manejo y poder crear las rutas de forma más fácil, se tendrá en cuenta solo el norte de la ciudad de Cali, se ingresara un csv, donde se especifique el conductor, la placa y donde tiene que llevar el pedido, el programa devolverá la ruta que debe realizar el domiciliario para entregar el pedido.

E. Evaluación y selección de la mejor opción

Criterio A: La velocidad del programa.

[3] La solución es la más rápida

[2] La solución es relativamente rápida

[1] La solución es extremadamente lenta

Criterio B: Confiabilidad de información

[3]: El programa es confiable al no perder la información.

[2]: El programa pierde cierta información

Camilo vivas
Jaime Cardona
Felipe Garcia

[1]: El programa pierde mucha información

Criterio C: Dificultad de implementación

[3]: El programa es fácil de entender e implementar

[2]: el programa es moderadamente difícil de entender e implementar

[1]: el programa es muy difícil de entender e implementa

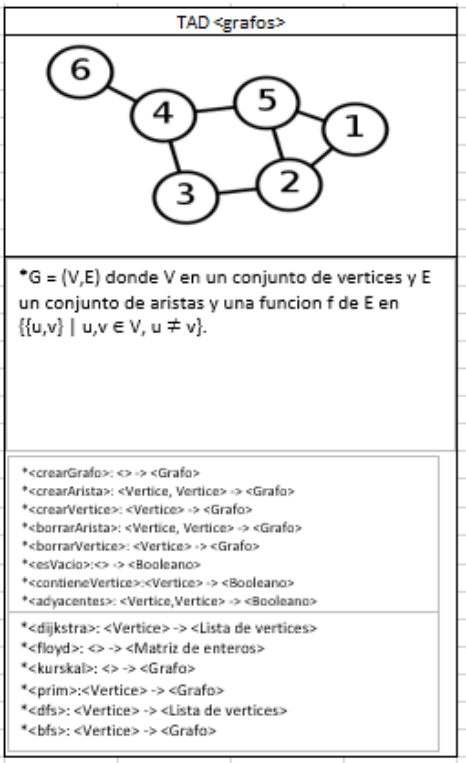
Alternativa (programa)	Criterio A	Criterio B	Criterio C	Total
Idea #1	3	1	2	6
Idea #2	3	2	1	6
Idea #3	2	3	2	7

F. Entrega de informes

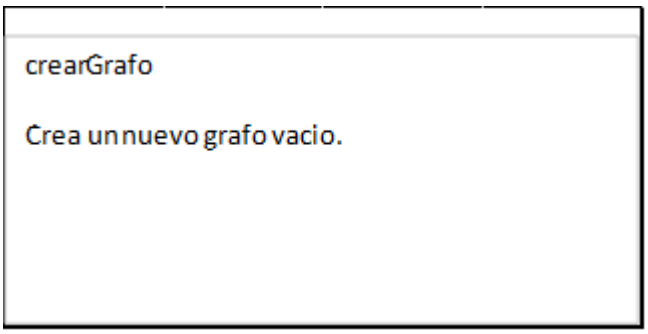
1. diagrama de fllujo

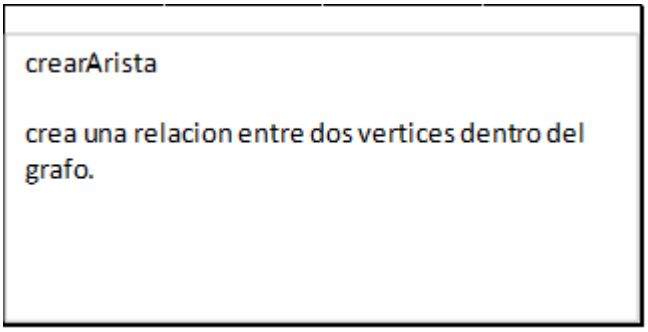
Camilo vivas
Jaime Cardona
Felipe Garcia

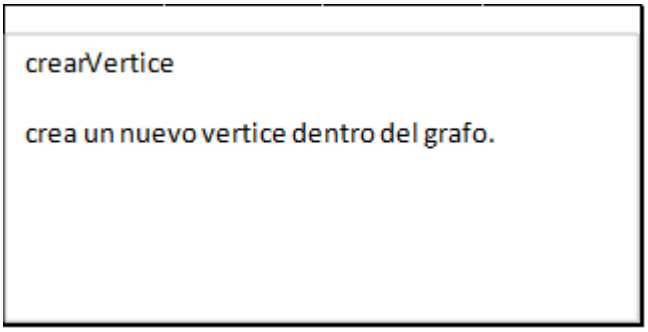
2. TADs



Camilo vivas
Jaime Cardona
Felipe Garcia







Camilo vivas
Jaime Cardona
Felipe Garcia

borrarArista

Elimina una relación entre dos nodos, en caso de que no exista el grafo no sufre cambios.

borrarVertice

Elimina el vertice y todas las aristas o relaciones que tuviese.

esVacio

Si el grafo no tiene ningun vertice devuelve verdadeo, por lo contrario devuelve falso

dijsktra

Nos ayuda a determinar el camino mas corto dado un vertice de origen, hacia el resto de vertices del grafo

floyd

encuentra el camino mas corto entre todos los pares de vertices, el resultado nos lo da en una matriz

Camilo vivas
Jaime Cardona
Felipe Garcia

kurskal

forma un arbol recubridor minimo del todo el grafo

prim

forma un arbol recubridor minimo del todo el grafo

dfs

es un algoritmo de busqueda, recorre los vertices de forma profunda es decir baja hasta donde mas puede y cuando no puede bajar mas se devuele y lo hace en otro vertice del grafo

bfs

es un algoritmo de busqueda, recorre los vertices a lo ancho.

Diseño de test:

Camilo vivas
Jaime Cardona
Felipe Garcia

.

Clases	Metodos	Valores de entrada	Resultado	Escenario
ListAdiacency	public Vetex<V> search(V v)	Un place con guía Cali, con nombre de cliente y producto nulos y valores de latitud y longitud de 0.	son equivalentes, lo que quiere decir que el search(V v) está trabajando de forma correcta y retorna el Vertex<v> que estábamos esperando	Escenario 1

Clases	Metodos	Valores de entrada	Resultado	Escenario
ListAdiacency	public IGraph<VertexConected<V>> prim()	Un place con guía a, con nombre de cliente y productos nulos y valores de longitud y latitud 0.	son equivalentes, lo que quiere decir que el método prim() está cumpliendo con su trabajo y está retornando el grafo esperado	Escenario 1

Clases	Metodos	Valores de entrada	Resultado	Escenario
ListAdiacency	public ArrayList<Vertex<V>> dijsktra(V v)	Un place con guía a, con nombre de cliente y productos nulos y valores de longitud y latitud 0.	son equivalentes, lo que quiere decir que el método dijsktra(V v) está cumpliendo con su trabajo y está retornando la lista de vértices esperada.	Escenario 1

Clases	Metodos	Valores de entrada	Resultado	Escenario
ListAdiacency	public IGraph<VertexConected<V>> kruskal(V v)	Ninguno	son equivalentes, lo que quiere decir que el método kruskal() está cumpliendo con su trabajo y está retornando el grafo esperado	Escenario 1

Camilo vivas
Jaime Cardona
Felipe Garcia

Clases	Metodos	Valores de entrada	Resultado	Escenario
ListAdjacency	public ListAdjacency<V>> bfs(V v)	Un place con guía a, con nombre de cliente y productos nulos y valores de longitud y latitud 0.	son equivalentes, lo que quiere decir que el método bfs(V v) está cumpliendo con su trabajo y está retornando el grado esperado	Escenario 1

Clases	Metodos	Valores de entrada	Resultado	Escenario
ListAdjacency	public ArrayList<V>> dfs(V v)	Un place con guía u, con nombre de cliente y productos nulos y valores de longitud y latitud 0.	son equivalentes, lo que quiere decir que el método dfs(V v) está cumpliendo con su trabajo y está retornando el camino esperado	Escenario 1

Clases	Metodos	Valores de entrada	Resultado	Escenario
MatrixAdjacency	public void addEdge(V u, V v, int w)	Place con guía Cali, con nombre de cliente y producto nulos y valores de latitud y longitud de 0, otro place con guía Popayán, con nombre de cliente y producto nulos y valores de latitud y longitud de 0 y un entero de valor 123	Se agrega el vértice de forma correcta	Escenario 1