

**Enunciado**

Han solo se ha contactado con la compañía StatGraph para que le ayude a encontrar la manera mas rápida para huir del imperio intergaláctico hacia un planeta fuera del radar de estos. Para esto hay que tener en cuenta que la nave de Han solo necesita ser recargada despues de cada salto entre los diferentes planetas por los que puede pasar para llegar a su destino( vuelo entre planetas ) y obtener suministros,una correcta administracion del tiempo y recarga del combustible van a ser cruviales para el exito de la mision,por lo que el programa debe de mostrarle a han solo cual es el camino que menos costo le va a generar haciendo que las probabilidades de exito sean muchisimo mas altas,sin embargo puede que en algunos de estos diferentes planetas el imperio intergaláctico tenga preparada una emboscada esperando que Han y su equipo pasen por alguno de los planetas en los que estan estas emboscados,por lo que si Han solo aterriza en alguno de esos planetas para recargar combustible y obtener suministros no podra continuar su escape,haciendo que su misión fracase rotundamente, es por eso que antes de la mision la resistencia envio a miembros de su facción por todos los posibles planetas que pueden ser parte de la ruta de Han para que antes de iniciar el escape se enviaran los datos de las emboscadas que prepare el imperio para que las reciba el programa, y en caso de que han se desvie por alguna razon de la ruta de menor costo pueda saber si hay alguna emboscada por este camino secundario que esta tomando.

**(no ditrigido , ponderado)**

**posible agragacion para el ultimo tema:**

ademas de estos una vez finalizada la mision un miembro de la resistencia pasara por todos los planetas para llevar a casa los miembros de la resistencia y los suministros que no se llevo han solo para que este pueda continuar con su siguiente mision,por lo que el programa debe de ser capaz de mostrarle a este miembro cual seria el camino minimo pasando por todos los planetas por los que pudo pasar han solo anteriormente.

**Especificacion de requerimientos**

**Requerimientos funcionales:**

**R1:**El programa debe de ser capaz de identificar la ruta mas corta a seguir entre los planetas para poder llegar del punto “A” al punto “B”.

**R2**:El programa debe de poder identificar los planetas emboscados(sin salida) e mostrarlos al usuario.

**R3:El programa debe de ser capaz de identificar la ruta mínima pasando por todos los planetas.**

**Requerimientos de proceso**

**RP1:** El software debe de desarrollarse en grupos de maximo 3 personas

**Requerimientos no funcionales:**

**RNF1:** El sistema debe de ser facil de usar para han solo o sus ayudantes.

**2. Recopilacion de informacion:**

**Hiperespacio:**

El hiperespacio en ciencia ficción es una especie de región conectada con nuestro universo gracias a los agujeros de gusano, y a menudo sirve como atajo en los viajes interestelares para viajar más rápido que la luz.

**FUENTE:https://es.wikipedia.org/wiki/Hiperespacio**

**Tamaño de la galaxia star wars:**

La galaxia, como se la conocía comúnmente, era una de las mil millones de galaxias en el universo. Era un vasto compuesto de más de 400 mil millones de estrellas estimadas[1] y más de 3.2 millones de sistemas habitables[1] que orbitaba alrededor de un supermasivo agujero negro en el corazón de la galaxia..[6] Los brazos galácticos giraban alrededor de este agujero negro a través de un diámetro de más de 100,000 años luz.[4] Fue el hogar de innumerables especies inteligentes y sistemas estelares.

**FUENTE:https://starwars.fandom.com/es/wiki/La\_galaxia**

**Imperio galactico:**El Imperio Galáctico es una de las principales facciones del universo ficticio de la saga de películas de La guerra de las galaxias. Se trata de un régimen autoritario imperial que abarca toda la Galaxia.

**FUENTE:https://starwars.fandom.com/es/wiki/Imperio\_Gal%C3%A1ctico**

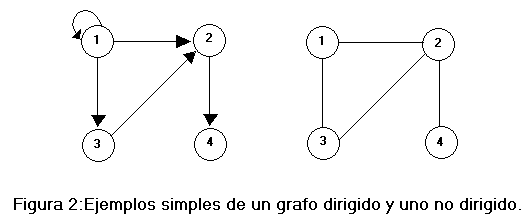
**Resistencia:**

En el universo de Star Wars, la Resistencia es una fuerza militar privada creada por la República para enfrentar a la Primera Orden, una junta militar nacida de los restos del Imperi**o**

**FUENTE:https://es.wikipedia.org/wiki/Resistencia\_(Star\_Wars)**

**3.búsqueda de soluciones creativas:**

**Grafos:**

****

Un grafo es una composición de un conjunto de objetos conocidos como nodos que se relacionan con otros nodos a través de un conjunto de conexiones conocidas como aristas, estos permiten estudiar las relaciones que existen entre unidades que interactúan con otras, pueden ser dirigidos o no dirigidos

**Grafo dirigido:**

Un grafo dirigido conocido también como dígrafo consta de un conjunto de vértices y aristas donde cada arista se asocia de forma unidireccional a través de una flecha con otro.

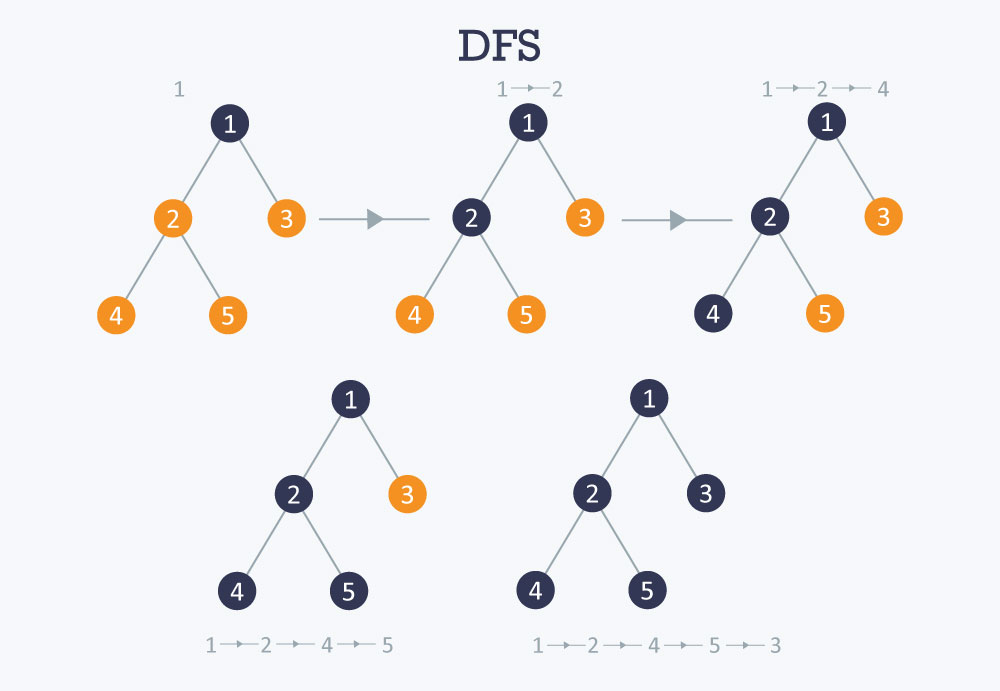
Las aristas dependiendo de su salida o ingreso reciben la calificación de entrante o saliente, la condición común, es que siempre tienen un destino hacia un nodo.

**Grafo no dirigido:**

Los grafos no dirigidos son aquellos que constan un conjunto de vértices que están conectados a un conjunto de aristas de forma no direccional**.**

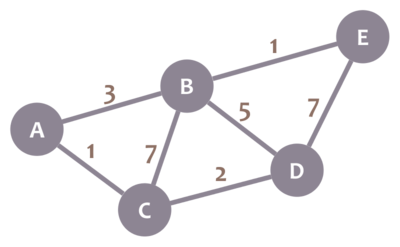
Esto significa que una arista puede indistintamente recorrerse desde cualquiera de sus puntos y en cualquier dirección.

**DFS:**

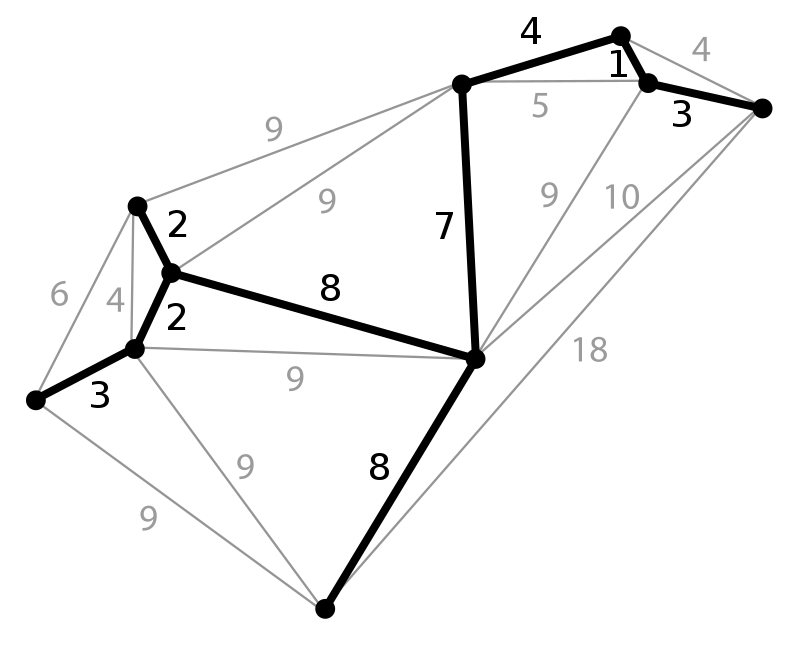
****

es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo. Su funcionamiento consiste en ir expandiendo cada uno de los nodos que va localizando, de forma recurrente (desde el nodo padre hacia el nodo hijo). Cuando ya no quedan más nodos que visitar en dicho camino, regresa al nodo predecesor, de modo que repite el mismo proceso con cada uno de los vecinos del nodo. Cabe resaltar que si se encuentra el nodo antes de recorrer todos los nodos, concluye la búsqueda.

**Dijkstra:**

****

El algoritmo de Dijkstra, también llamado algoritmo de caminos mínimos, es un algoritmo para la determinación del camino más corto dado un vértice origen al resto de vértices en un grafo con pesos en cada arista

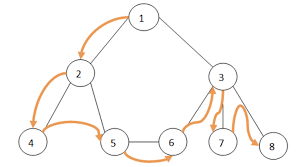
****

El algoritmo de Kruskal es un algoritmo de la teoría de grafos para encontrar un árbol recubridor mínimo en un grafo conexo y ponderado. Es decir, busca un subconjunto de aristas que, formando un árbol, incluyen todos los vértices y donde el valor total de todas las aristas del árbol es el mínimo. Si el grafo no es conexo, entonces busca un bosque expandido mínimo (un árbol expandido mínimo para cada componente conexa). El algoritmo de Kruskal es un ejemplo de algoritmo voraz.

**Alternativa 1. BFS, Floyd:**

En esta alternativa podremos encontrar con el BFS caminos en los que le quieran hacer una emboscada, sin embargo cuando encontremos el primer camino parara asi que no sabremos en que otros planetas hay emboscada

**BFS:**

****

La búsqueda en amplitud (BFS) es un algoritmo para atravesar o buscar estructuras de datos de árboles o grafos. Comienza en la raíz del árbol (o algún nodo arbitrario de un graph, a veces denominado "clave de búsqueda") y explora primero los nodos vecinos antes de pasar a los vecinos del siguiente nivel.

**Alternativa 2. BFS, Floyd y Kruskal**

Con esta alternativa tenemos el mismo problema de BFS, sin embargo kruskal nos ayuda a recorrer todos los planetas de la manera mas rapida para recoger los medicamentos, por lo que nos puede ayudar en este punto

**Alternativa 3. Floyd , DFS, Kruskal**

Esta alternativa utiliza DFS Kruskal, algoritmos que se acomplan bastante a nuestro problema, sin embargo usa floyd y floyd trabaja con grafos dirijisdos y en nuestro problema Han solo puede ir por cualquier camino.

**Alternativa 4. Lapiz y papel**

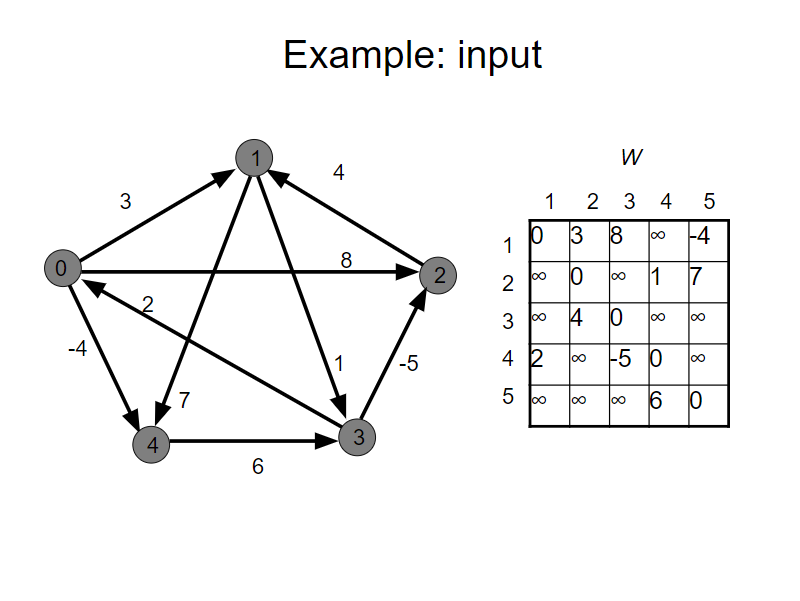
Con lapiz y papel podriamos reunir todos los datos que nos envien de emboscadas para encontrar los planetas a los que Han Solo no debe ir y tambien podriamos calcular el camino mas rapido, sin embargo este metodo es muy poco eficiente

**Alternativa 5. Dfs y dijkstra:**

Aunque esta alternativa podría cumplir los 2 primeros requerimientos le sigue faltando 1 pues no cumple con el camino minimo entre todos los planteas.

**Alternativa 6.BFS, DIJKSTRA, KRUSKEL:**

en esta alternativa proponemos resolver la problemática planteada haciendo uso de estos 3 algoritmos, cada uno de estos desempeñando una función diferente,dijkstra podria encontrar el cambio mas corto hacia el sitio que necesita desplazarse han solo y su equipo,el BFS se modificara para que han solo pueda introducir un planeta final y uno inicial, asi cada vez que quiera saber si hay emboscada lo podra hacer, y por ultimo Kruskal permitirá recorrer todos los planeas de la forma menos costosa para recolectar los suministros necesarios para han solo.



Floyd-Warshall. Es un algoritmo de análisis sobre grafos que permite encontrar el camino mínimo en grafos dirigidos ponderados. El algoritmo encuentra el camino entre todos los pares de vértices en una única ejecución, constituyendo un ejemplo de programación dinámica.

**4.transición de ideas a diseños preliminares**

**Lapiz y papel:** Esta alternativa puede resolver todos los requerimientos pedidos por el problema para han, sin embargo este metodo no es el mas eficiente, siendo por mucho mas lento que otras opciones y sin contar la probabilidad de error por el que lo esta realizando

**Bfs, Floyd: Esta alternativa aunque intenta resolver el problema que se nos presente no logra cumplir con la ultima condicion que es la de encontrar el camino minimo pasando por todos los planetes**

**Dfs y dijkstra:**Aunque esta alternativa podría cumplir los 2 primeros requerimientos le sigue faltando 1 al igual que la anterior opción, no cumple con el camino minimo entre todos los plantea**s.**

**BFS, Floyd y Kruskal:** Esta alterntiva cumple con los requerimientos necesarios requridos por el problema, pero no es el que mejor se acopla mas que todo por el warshall, debido a que este funciona con grafo dirigidos, cosa que va en contra del contexto del problema en general ya que han solo puede viajar a cualquier planeta sin necesidad de seguir un orden por asi decirlo, ademas de que el algoritmo bfs aunque podria realizar la busqueda de los sitios en los que se les tiene una emboscada existen mejores alternativas para la misma

**Floyd, DFS, Kruskal:** Esta alternativa aunque igualmente cumple con los requerimientos y el dfs funciona de manera mas optima sigue siendo mas ineficiente que los otros, debido al mismo problema de la anterior, warshal funciona con grafos dirigidos cosa que no pide nuestro problema

**Dijkstra, BFS y Kruskal:** Esta alternativa es como las otras, sin embargo estas 3 implementaciones se adaptan de una mejor manera a los requerimientos debido a los requisitos para aplicarlas, es decir para todas 3 sirven los grafos ponderados no dirigidos.

**5.evaluacion y seleccion de la mejor solución**

**Los criterios para evaluar y seleccionar la mejor solución serán:**

**Criterio A Eficiencia:**

Se prefiere una solución con mejor eficiencia que las otras consideradas. La eficiencia puede ser:

[3] Muy eficiente

[2] Eficiente

[1] No muy eficiente

**Criterio B Efectividad:**

Se preferiría lo mas efectivo y que menos errores de al momento de soluicionar el problema. La efectividad puede ser:

[3] Muy efectivo

[2]Efectivo

[1]No muy efectivo

**Criterio C:**

Se preferirá la alternativa que no pueda ayudar a desarrollar todos los requisitos del proyecto sin excepción. El rango será:

[2] Desarrolla todos los requisitos.

[1] Desarrolla la mayoría de los requisitos, sin embargo presenta problemas en algunos de ellos

|  | **Criterio A** | **Criterio B** | **Criterio C** | **Total** |
| --- | --- | --- | --- | --- |
| **Alternativa 1:**  **BFS, Floyd** | **2** | **1** | **1** | **4** |
| **Alternativa 2:**  **BFS, Floyd y Kruskal** | **2** | **2** | **1** | **5** |
| **Alternativa 3:**  **Floyd , DFS, Kruskal** | **1** | **1** | **1** | **3** |
| **Alternativa 4:**  **Lapiz y papel** | **1** | **1** | **2** | **4** |
| **Alternativa 5:**  **Dfs y dijkstra** | **2** | **2** | **1** | **5** |
| **Alternativa 6:**  **Bfs, DIJKSTRA, KRUSKEL** | **3** | **3** | **3** | **9** |

De acuerdo con la evaluación anterior la alternativa que se debe utilizar es la 6ta pues fue la que obtuvo mayor puntaje por ende es la que mejor resuelve nuestros criterios y proyecto.