 **Método de la Ingeniería**

**Aplicación para la Solución de un Problema**

**Alejandra Gonzalez, Angie Cordoba, Miguel Rosas**

Contents

[Identificación del problema: 1](#_Toc528255969)

[Requerimientos: 2](#_Toc528255970)

[TADS 4](#_Toc528255971)

[Recopilación de Información: 8](#_Toc528255972)

[Alternativas Creativas: 10](#_Toc528255973)

[Elementos relacionados entre las Alternativas creativas escogidas: 12](#_Toc528255974)

[Búsqueda de Soluciones Creativas: 12](#_Toc528255975)

[Alternativas de solución creativas: 13](#_Toc528255976)

[Ventajas y desventajas: 15](#_Toc528255977)

[Fase 6. En este paso procedemos a 17](#_Toc528255978)

[Fase 7. Evaluación y selección de la mejor solución 19](#_Toc528255979)

[Fase 8. Diseño preliminar de cada idea no descartada: (Modelos de Simulación) 20](#_Toc528255980)

[Diagramas de clases: 21](#_Toc528255981)

[Fuentes: 21](#_Toc528255982)

# Identificación del problema:

Definición del problema:

//general…….

//zona wifi

//turistas

//auge

La corporación “MAA” se ve en la necesidad de plantear una manera con la cual cualquier persona pueda acceder a una locación especificada dependiendo de donde se encuentre y que le mostrar en donde se encuentra la ubicación de cada\_\_\_\_\_\_\_ para satisfacer sus necesidades. Cada usuario ingresa en un sistema el cual le permite realizar diferentes consultas dependiendo del mapa que este desea ingresar y las locaciones que le gustaría visitar.

Causas y Síntomas:

Debido a la avalancha de personas que requieren ir a diferentes locaciones de\_\_\_\_\_\_\_\_\_\_dentro de un mapa, cada lugar representa un dato mínimamente útil. la Federación Internacional de Locaciones “MAA” ha considerado optimo consolidar los datos de mayor relevancia de cada uno de los lugares a los que un usuario le gustaría ir. Con la falta de disposición de un aplicativo que permita satisfacer esta necesidad, se ven en disposición de contratar un ente externo que pueda elaborar dicho software. El cual le permitirá al usuario encontrar los puntos de\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

# Requerimientos:

Requerimientos:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** |  | **#** | 1 |
| **Descripción** |  | | |
| **Entrada** |  | | |
| **Salida** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **R2** | **#** | 2 |
| **Descripción** |  | | |
| **Entrada** |  | | |
| **Salida** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **R3** | **#** | 3 |
| **Descripción** |  | | |
| **Entrada** |  | | |
| **Salida** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **R4** | **#** | 4 |
| **Descripción** |  | | |
| **Entrada** |  | | |
| **Salida** | - Ninguna - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **R5** | **#** | 5 |
| **Descripción** |  | | |
| **Entrada** |  | | |
| **Salida** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** |  | **#** | 1 |
| **Descripción** |  | | |
| **Entrada** |  | | |
| **Salida** |  | | |

# TADS

|  |
| --- |
| **Nombre: HASH TABLE** |
| **Dibujo:**  Image result for hash table |
| **Invariante:**  { Inv : Para cada key, la función hash h(k) devuelve la posición en que se encuentra ubicado en la tabla hash} |
| **Operaciones Primitivas:**   * **Hast Table** - **HashTable** * **Put Key x Value HashTable** * **getKey** Value * **Remove** key**value** * **isEmpty HashTable booelan** |

|  |
| --- |
| **HashTable 🡪 HashTable** |
| “Construye una tabla Hash Vacia” |
| Precondiciones: |
| Postcondiciones: |

|  |
| --- |
| **Put Key x Value 🡪 HashTable** |
| “Inserta en la tabla una clave asociada a un valor” |
| Precondiciones: key es una llave valida |
| Postcondiciones:  La pareja insertada en la tabla hash |

|  |
| --- |
| **get Key 🡪 Value** |
| “recupera el valor asociado a una clave de una tabla hash” |
| Precondiciones: pareja(key, value) que esta en la tabla con algún valor |
| Postcondiciones:  El valor retornado correctamente |

|  |
| --- |
| **Remove Key 🡪 Value** |
| “Recupera el valor del elemento en la parte de enfrente de la cola” |
| Precondiciones: key es una llave valida |
| Postcondiciones:  Remueve el elemento al que le corresponde esa llave |

|  |
| --- |
| **IsEmpty HashTable 🡪 Boolean** |
| “Determina si la tabla hash está vacía o no” |
| Precondiciones: tablahash h |
| Postcondiciones:  Verdadero si q = Falso si q es diferente de |

# Recopilación de Información:

GPS:

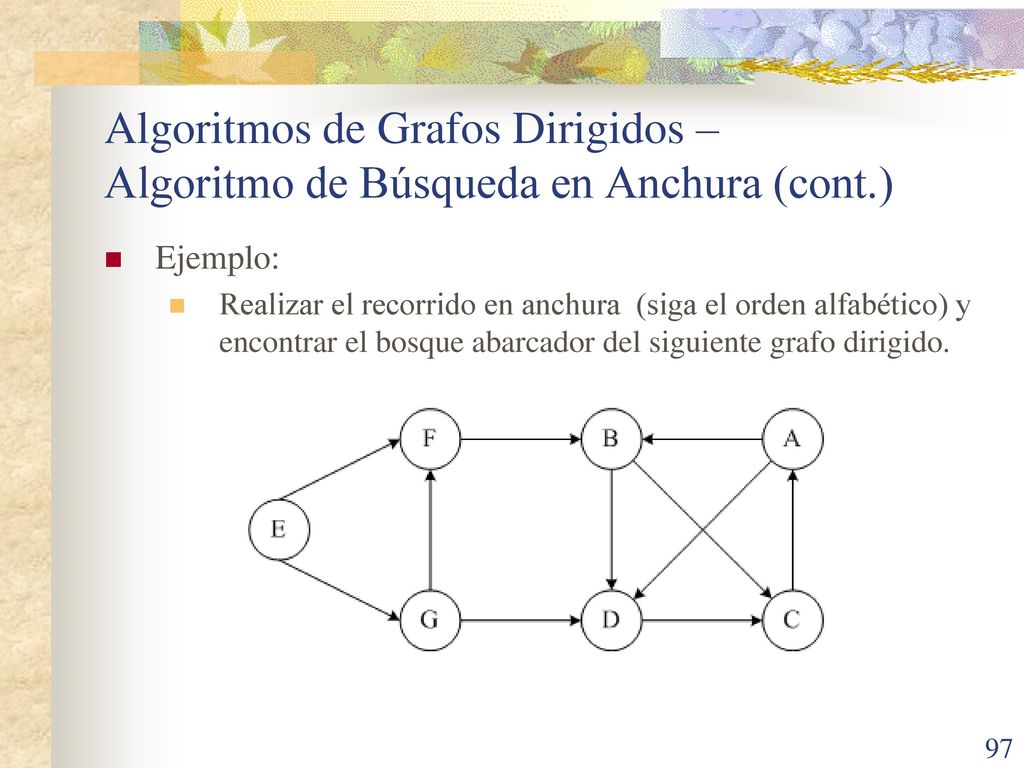
El Sistema de posicionamiento global (GPS), originalmente Navstar GPS , es un sistema de radionavegación basado en satélites , propiedad del gobierno de los Estados Unidos y operado por la Fuerza Aérea de los Estados Unidos. Es un sistema global de navegación por satélite que proporciona información de geolocalización y tiempo a un receptor de GPS en cualquier lugar o cerca de la Tierra donde hay una línea de visión sin obstáculos para cuatro o más satélites GPS. Obstáculos como montañas y edificios bloquean las señales GPS relativamente débiles.

Grafo Dirigido

Un grafo dirigido (o digrafo) G consiste de un conjunto de vértices y un conjunto de arcos E. A los vértices se les llama también nodos o puntos y a los arcos aristas dirigidas o líneas dirigidas. Un arco es un par ordenado de vértices (v, w) donde v es la cola y w la cabeza del arco. Un arco (v, w) se expresa también como v w.

# Alternativas Creativas:

Grafo Dirigido para las ciudades o localizaciones que introduzca el usuario:

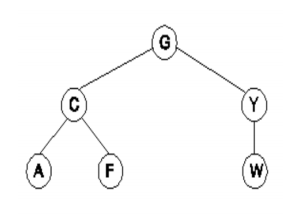


Los grafos dirigidos son grafos en los que las aristas tienen una dirección definida; por ejemplo, se puede dar el caso de poder ir del nodo A al nodo B, pero no al revés. En la mayoría de los casos la dirección de las aristas indica algún tipo de relación de precedencia entre los nodos. Los grafos dirigidos pueden ser usados para:

• Modelar líneas de fabricación, en las que diferentes procesos dependen de otros

• Manejar dependencias en la compilación de archivos, como hace el make

**Alternativa 2: Arboles (Estructuras de Datos)**

A los arboles ordenados de grado dos se les conoce como arboles binarios ya que cada nodo del árbol no tendrá más de dos descendientes directos, los cuales se pueden llamar izquierda o derecha. Las aplicaciones de los arboles binarios son muy variadas ya que se les puede utilizar para representar una estructura en la cual es posible tomar decisiones con dos opciones en distintos puntos. Esta estructura de datos nos va a facilitar un poco a la hora de introducir los datos al Sistema.

# Elementos relacionados entre las Alternativas creativas escogidas:

Una de las relaciones que podemos encontrar entre las soluciones creativas, es que todas tratan de incorporar una base de datos al sistema. Las practicas que se realizan en cada una varían demasiado entre las alternativas. Pero un factor importante con las ideas escogidas es la capacidad de manejo de un gran volumen de datos. Esto nos facilitaría mucho a la hora de la implementación algorítmica. Las alternativas escogidas son compatibles con la solución del problema, lo que hace que el enunciado no sea complejo de tratar.

# Búsqueda de Soluciones Creativas:

método generador de ideas:

Para desarrollar bien las alternativas creativas usamosel método de lluvia de ideas. Primero, dividimos el problema en tres categorías: entradas, procesos, salidas. Luego, en cada categoría, enlistamos ideas abstractas que nos sirven como sustantivos para la siguiente etapa. Categorías como “procesos” tiene ya predefinidas las alternativas investigadas en el paso anterior de recolección de información. Con las categorías de entrada y salida lo que buscamos es encontrar componentes que nos permitan darnos una idea de cómo recibir y mostrar la información. A continuación, cada componente enlistado en cada categoría es indexado. Por ejemplo, las ideas que salieron en la categoría de entrada están enumeradas; las categorías en proceso están ordenados alfabéticamente y en la categoría de salidas están enlistadas con números romanos.

|  |  |  |
| --- | --- | --- |
| 1. Archivos Planos |  | 1. Jlist |
|  | 1. Campo de Texto |
|  | 1. Archivo |
| 2. Campo de texto |  | 1. Ventana Emergente |
|  |  |
| F. |  |

Una vez teniendo llena esta tabla, cada uno procede a escribir combinaciones tomando componentes de cada categoría. Decidimos poner dos condiciones, la primera es solo tomar un componente de la categoría de entrada, y tres, diferentes, de la categoría de procesos. De esta forma, nos aseguramos tener un solo componente de entrada y 3 elementos de la categoría de procesos. Con respecto a la salida, decidimos seleccionar más de uno pues nos daría una idea general, o alternativas, para mostrar el resultado.

Un ejemplo del procedimiento es el siguiente:

**1,a,b,c,II,III** esto quiere decir que la idea se arma con el sustantivo 1 de la categoría de entrada, primero, segundo y tercero de la categoría de proceso y el componente nombrado con los números romanos de la categoría de salida.

Con este formato, procedemos a armar oraciones que representen una idea para solucionar nuestro problema.

Continuamos seleccionando las 4 ideas postuladas por cada uno, las unimos para mejorarlas y sacamos las 2 ideas que serán postuladas para el siguiente proceso el cual es; alternativas de soluciones creativas.

# 

# Alternativas de solución creativas:

Alternativa 1:

Alternativa 2:

Alternativa 3:

Alternativa 4:

# Ventajas y desventajas:

Alternativa 1:

Alternativa 2:

Alternativa 3:

Aternativa 4:

Fase 6. En este paso procedemos adefinir los criterios que nos permitirán evaluar las alternativas de las soluciones anteriormente propuestas y con base en este resultado elegir la solución que mejor satisface las necesidades del problema planteado. Los criterios que escogimos en este caso son los que enumeramos a continuación. Al lado de cada uno se ha establecido un valor numérico con el objetivo de establecer un peso que indique cuáles de los valores posibles de cada criterio tienen un mayor puntaje.

**Criterio 1: En el criterio 1 evaluaremos la precisión de la solución que se va a escoger, definiéndola en una escala numérica.**

**A)** La solución es Exacta—Valor: 2

B) La solución es Aproximada –Valor: 1

**Criterio 2: En el criterio 2 evaluaremos la facilidad que tendrá el usuario al momento de ingresar los datos.**

A) Fácil ----) Valor: 3

B) Media ---) Valor: 2

C) Algo Complicada ---) Valor:1

**Criterio 4: En el criterio 4 evaluaremos el tiempo utilizado por cada algoritmo.**

1. A) Constante --) Valor: 1
2. B) Logarítmica –) Valor: 6
3. C) Lineal --) Valor: 5
4. D) Cuadrática --) Valor: 4
5. E) Cubica --) Valor: 3
6. F) Exponencial --) Valor: 2

**Criterio 5: En el criterio 5 evaluaremos el espacio en memoria utilizado por cada algoritmo.**

1. A) Constante --) Valor: 1
2. B) Logarítmica –) Valor: 6
3. C) Lineal --) Valor: 5
4. D) Cuadrática --) Valor: 4
5. E) Cubica --) Valor: 3
6. F) Exponencial --) Valor: 2

**Criterio 6: En el criterio 6 evaluaremos la facilidad con la que se muestran los datos en el programa.**

A) Fácil ----) Valor: 3

B) Media ---) Valor: 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Alternatives** | **Criterio 1** | **Criterio 2** | **Criterio 3** | **Criterio 4** | **Criterio 5** | **Criterio 6** |
|  | **Precisión de la solución** | **Facilidad de ingreso de datos** | **Facilidad de**  **implement ación del mundo** | **Tiempo Utilizado por cada Algoritmo** | **Espacio de**  **Memoria Utilizado por cada Algoritmo** | **Facilidad de**  **implement ación del algoritmo** |
| **Alternative 1** |  |  |  |  |  |  |
| **Alternative 2** |  |  |  |  |  |  |
| **Alternative 3** |  |  |  |  |  |  |
| **Alternative 4** |  |  |  |  |  |  |

# Fase 7. Evaluación y selección de la mejor solución

**Alternativa :**

**Alternativa :**

En esta fase lo que procedimos a hacer fue evaluar la mejor solución, teniendo en cuenda la tabla de criterios anteriormente definida y descartando las ideas que no alcanzaron con el puntaje promedio para pasar a la siguiente fase.

# Fase 8. Diseño preliminar de cada idea no descartada: (Modelos de Simulación)

# Diagramas de clases:

# Fuentes: