## Enunciado

Algoritmia es un mar gobernado por piratas. En la Isla Capital, manda el Führer, el más feroz y poderoso de ellos. Él nos ha encomendado hacer un bot de navegación que permita a cualquier marinero orientarse por estas aguas malditas y así traer algo de turismo a la zona.

El bot debe ser capaz de:

* Utilizar la base de datos existente con las coordenadas de las islas para mapearlas y determinar las distancias entre los puertos a través de las rutas establecidas.
* Determinar cuál es la ruta más corta de una isla a otra.
* Indicar por cuáles islas pasar y cuánta distancia hay entre la ubicación actual y la de destino.
* Dar información al usuario acerca de la ubicación en la que se encuentra y la ubicación de destino, para lo cual se provee de los nombres y las descripciones de cada isla.
* Permitir un poco de aventura: darle la opción de que se le envíe a un destino al azar.

Para ello debemos valernos del poder de los algoritmos, las matrices y confiar en la ambición de nuestros corazones.

## Pseudocódigo

Procedimiento cargar\_mapa(  
 S:  
 mapa[n][n]: Booleano,  
 mapa\_lugares[n][n]: Entero,  
 E:  
 n: Entero  
);  
Inicio  
 var i, j: Entero;  
  
 // El mapa de booleanos representa caminos y obstaculos  
 // El mapa de enteros representa zonas  
  
 Para (i = 1 Hasta i = n) Hacer;  
 Para (j = 1 Hasta j = n) Hacer;  
 mapa[i][j] = 0;  
 mapa\_lugares[i][j] = 1;  
 Fin\_Si  
 Fin\_Si  
  
 // Aca hay que cargar ambos mapas, ya que por ahora son mapas planos  
Fin\_Procedimiento  
  
  
Procedimiento cargar\_rutas(E/S: rutas[20][2]: Entero);  
Inicio  
 // ...  
Fin\_Procedimiento  
  
Funcion obtener\_distancia(coordenada\_1[2], coordenada\_2[2]: Entero): Entero;  
Inicio  
 var x1, x2, y1, y2: Entero;  
 var distancia\_x, distancia\_y, distancia: Entero;  
  
 x1 = coordenada\_1[1];  
 x2 = coordenada\_2[1];  
 y1 = coordenada\_1[2];  
 y2 = coordenada\_2[2];  
  
 distancia\_x = x1 - x2;  
 distancia\_y = y1 - y2;  
  
 Si (distancia\_x < 0) Entonces;  
 distancia\_x = distancia\_x \* -1;  
 Fin\_Si  
 Si (distancia\_y < 0) Entonces;  
 distancia\_y = distancia\_y \* -1;  
 Fin\_Si  
  
 distancia = distancia\_x + distancia\_y;  
  
 Devolver (distancia);  
Fin\_Funcion  
  
Procedimiento pantalla(S: opc: Entero);  
Inicio  
 Mostrar << "Bienvenido a GaboMaps. Elija una opcion: ";  
 Mostrar << "1. Actualizar mi ubicacion. ";  
 Mostrar << "2. Ver lista de lugares.";  
 Mostrar << "3. Ver como llegar desde tu ubicacion.";  
 Mostrar << "4. Como llegar a un lugar aleatorio desde tu ubicacion.";  
 Mostrar << "5. Ver informacion de tu ubicacion actual. ";  
 Mostrar << "6. Salir";  
 Leer >> opc;  
  
 Mientras (opc < 1 Or opc > 6) Hacer;  
 Mostrar << opc, "No esta en las opciones. Intente nuevamente. ";  
 Leer >> opc;  
 Fin\_Mientras  
Fin\_Procedimiento  
  
Procedimiento ubicacion(  
 E:  
 coordenadas[n][2]: Entero,  
 nombres[n],  
 n: Entero,  
 S:  
 ubicacion[2]  
 );  
Inicio  
 var i, opc\_isla: Entero;  
  
 Mostrar << "¡Hola! Dime en cuál de estas islas estas: ";  
 Para (i = 1 Hasta i = n) Hacer;  
 // Mostramos la i y el nombre de la isla  
 Mostrar << i, ". ", nombres[i]; // "1. Isla Del Muerto" por ejemplo  
 Fin\_Si  
  
 Mientras (opc\_isla < 1 Or opc\_isla > n) Hacer;  
 Mostrar << "Esa isla no existe. ";  
 Leer >> opc\_isla;  
 Fin\_Mientras  
  
 // opc\_isla sera una de las n islas guardadas  
Fin\_Procedimiento  
  
Algoritmo GaboMaps  
Inicio  
 // Mapa  
 var n = 500: Entero;  
 var mapa[n][n]: Entero;  
  
 // Informacion de las islas  
 var coordenadas[n][2]: Entero;  
 var nombres[n], descripciones[n]: Cadena;  
  
 // Rutas y distancias  
 var rutas[20][2], distancias[20]: Entero;  
  
 // Interaccion del usuario con el menu  
 var opc: Entero;  
 var ubicacion[2], destino[2]: Entero;  
  
 cargar\_islas(coordenadas[n][2], nombres[n], descripciones[n], n);  
 cargar\_rutas(rutas[20][2]);  
  
 Mientras (opc != 6) Hacer;  
 ubicacion(coordenadas[n][2], nombres[n], ubicacion[2], n);  
 Mostrar << ubicacion;  
  
 pantalla(opc);  
 Mostrar << pantalla;  
  
 En\_Caso (opc > 0) Sea;  
 Caso(opc = 1);  
 Mostrar << "Igresando nueva ubicacion...";  
 ubicacion(ubicacion[2]);  
 Mostrar << ubicacion;  
 Caso(opc = 2);  
 // Listar los lugares  
 lista\_lugares();  
 Caso(opc = 3);  
 // Encontrar un camino desde tu ubicacion  
 // 1. Obtener el destino  
 // 2. Utilizar el algoritmo de Dijkstra para encontrar el camino  
 // 3. Almacenar las instrucciones  
 // 4. Entregar las instrucciones en formato de texto  
 Caso(opc = 4);  
 // Ir a un lugar aleatorio  
 // 1. Obtener dos numeros aleatorios  
 // 2. Repetir el procedimiento anterior con esos numeros  
 Caso(opc = 5);  
 // Ver informacion de tu ubicacion actual  
 ver\_info(mapa\_lugares[n][n], n, ubicacion[2]);  
 Mostrar << ver\_info;  
 Otro\_Caso  
 Mostrar << "Que tenga buen viaje! Gracias por usar GaboMaps!";  
 Fin\_Caso  
 Fin\_Mientras  
Fin