

## PRÁCTICA CURSO 2001/02

**Todos** los alumnos matriculados en esta asignatura en el curso académico 2001/02 serán examinados y evaluados según sus conocimientos y habilidades para desarrollar y modificar esta práctica individualmente, lo que incluye a los alumnos de "plan antiguo". **Ningún alumno será evaluado con ninguna práctica desarrollada en años anteriores.**

La práctica consiste en resolver la problemática que se corresponde con el entorno geográfico de las Islas Canarias (*figura 1*), en el que un empresario armador especializado en transporte marítimo, tanto de pasajeros como de mercancías, tiene un capital importante invertido en estas islas. Dentro de esta Comunidad Autónoma, el armador necesita controlar el desplazamiento de sus barcos de unas islas a otras, suponiendo siempre que en cada isla hay un único puerto. Se trata de tener controlados los puertos y las rutas, sabiendo dónde están los barcos, con sus pasajeros o mercancías. Es decir, en cada puerto se deben controlar las salidas y llegadas de los barcos de pasajeros y de mercancías. Por otro lado, entre un puerto y otro hay trazadas unas rutas marítimas y hay establecido un tiempo determinado en realizar un recorrido en circunstancias normales. Pero, ante circunstancias excepcionales, estos tiempos pueden variar (por ejemplo, una tempestad entre dos islas). Además se debe tener en cuenta que no existe una ruta marítima entre todo par de islas de forma directa, incluso puede haber una rutas en una determinada dirección pero no en la opuesta.

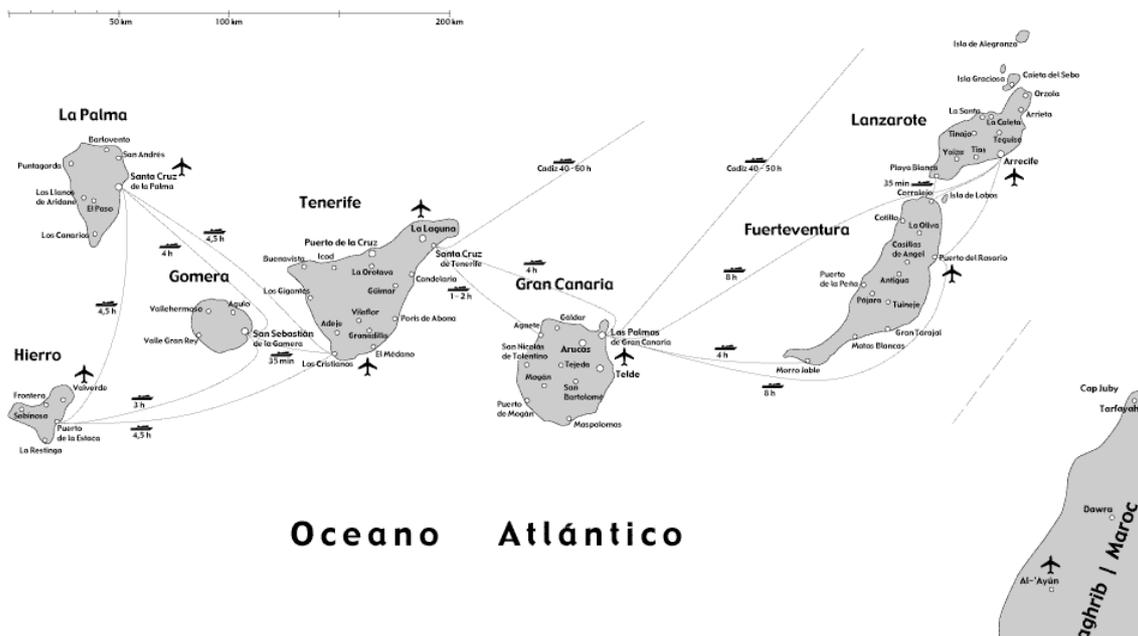


Fig.1.- Mapa real de las rutas marítimas de las Islas Canarias

Tras escuchar el planteamiento del problema que nos hace nuestro cliente, tal y como se ha expuesto anteriormente, nos paramos a pensar en las posibles formas de organizar toda la información que desea mantener, acerca de sus barcos, las mercancías, los pasajeros, las rutas, etc...

Las conclusiones a las que llegamos son las siguientes: parece que la estructura más adecuada para mantener la información de las rutas y los caminos entre cada puerto será una estructura de grafo.

Dado que en cada puerto se debe tener siempre actualizada la información de los barcos que hay atracados, así como los que van a salir, y si son de pasajeros o de mercancías, y pensando que, además, va a haber bastante movimiento de barcos que entran y que salen, cada puerto organizará los barcos que van llegando en una estructura que permita un rápido acceso a la información a fin de realizar búsquedas, inserciones y extracciones de forma rápida y eficiente.

Cada barco de pasajeros debe llevar una lista con las personas que viajan en él. Se trata de una exigencia del gobierno de la comunidad autónoma, que, en caso de accidente, debe conocer los nombres de los pasajeros que viajaban en cada barco. De igual forma cada barco de mercancías debe llevar información acerca de las mercancías que transporta y qué contenedores lleva. Esto último es una exigencia del cliente, puesto que desea saber en qué puerto o en qué barco se encuentran sus contenedores.

## 1. Descripción del funcionamiento

Para controlar las distintas rutas marítimas con nuestro sistema es necesario:

### 1.1. Inicialización del sistema:

En esta primera fase de la aplicación, las tareas que se han de llevar a cabo se resumen en los siguientes pasos:

**A.-** Inicializar la estructura de los puertos a considerar. Sólo consideramos un puerto por cada isla. Así mismo se deben almacenar todas las rutas posibles y el tiempo de tránsito entre islas. Para realizar esta operación se utilizará el fichero **rutas.txt** que se explicará más adelante.

**B.-** Situar los distintos barcos (los de pasajeros y los de mercancías), que sean propiedad de nuestro cliente, en los puertos en los que se encuentren inicialmente. Para esta operación utilizaremos el fichero **barcos.txt** que se detalla más adelante.

**C.-** Situar los contenedores de mercancías en los puertos que nos indica el cliente. Para realizar esta operación utilizaremos el fichero **contenedores.txt** que se explica más adelante.

### 1.2. Funcionamiento habitual del sistema:

Esta sería la fase de ejecución habitual de la aplicación. Simulamos los diferentes eventos que puedan tener lugar en la realidad, mediante una cola de sucesos. Esta cola se forma a partir de un fichero de texto (**eventos.txt**, explicado más adelante) en el que cada

línea refleja un suceso que debe ser atendido por nuestra aplicación. Todos estos eventos se orientan a tratar los siguientes aspectos del sistema.

- D.- Situar pasajeros o contenedores en los barcos que vayan a zarpar del puerto.
- E.- Mantener la situación de los barcos en las distintas rutas marítimas.
- F.- Mantener la información de los barcos que van llegando a los distintos puertos.
- G.- Mostrar el estado de los puertos.
- H.- Mostrar el estado de las diferentes rutas.

### 1.3. Finalización del sistema:

Esta sería la última fase de ejecución, en la que se deben destruir de forma adecuada y ordenadamente todas las estructuras de datos, de modo que se devuelvan todos los recursos asignados.

## 2. Descripción de las operaciones

### 2.1. Inicialización del sistema:

Nuestro sistema es necesario que realice las tareas anteriormente comentadas y que se detallan a continuación. Para esto se utilizarán cuatro ficheros de texto, que deben especificarse como parámetros de entrada al programa principal.

#### A.- Inicializar la estructura de los puertos a considerar

La primera vez que empiece a funcionar nuestro sistema de control, lo primero que hará será inicializar la información de todos los puertos a tener en cuenta. La estructura que inicializamos en esta fase debe proporcionarnos facilidad para mantener y representar la información correspondiente a los tiempos de viaje para las diferentes rutas.

Tal y como se indicó antes, parece que la estructura más adecuada en este caso es la correspondiente a un grafo. Es fácil observar en el mapa que se tratará de un grafo dirigido y valuado. Por tanto debe mantenerse información acerca del conjunto de vértices y arcos del mismo.

En este caso, el conjunto de vértices lo forman los puertos, que deben tener la información correspondiente a los barcos atracados en ellos. El conjunto de arcos lo podemos representar mediante una matriz de adyacencia, tal y como se explica en la asignatura “Estructuras de Datos y Algoritmos”.

Se asociará un identificador con cada puerto (que serán los *vértices* del grafo a considerar). Sería muy interesante en este caso la utilización de un *tipo enumerado*, ya que así tendremos acceso directo a la información de los puertos:

```
typedef enum { LP, HI, GO, TE, GC, FU, LA } islas_canarias;
```

Inicialmente los tiempos que tardan los barcos en desplazarse de un puerto a otro queda reflejado en la *figura 2*.

A fin de facilitar la interpretación de los datos que refleja la figura, disponemos de un fichero de texto, **rutas.txt**, cuyo formato será el que se indica en el ejemplo que hay a continuación:

```
LP#HI#GO#TE#GC#FU#LA
0#7#4.5#inf#inf#inf#inf
inf#0#inf#5#inf#inf#inf
inf#4#0#1.5#inf#inf#inf
6#inf#2#0#3#inf#inf
inf#inf#inf#2#0#3.5#inf
inf#inf#inf#inf#inf#0#4.5
inf#inf#inf#inf#5#inf#0
```

Este fichero se corresponde con la siguiente matriz de adyacencia:

	LP	HI	GO	TE	GC	FU	LA
LP	0	7	4.5	inf	inf	inf	inf
HI	inf	0	inf	5	inf	inf	inf
GO	inf	4	0	1.5	inf	inf	inf
TE	6	inf	2	0	3	inf	inf
GC	inf	inf	inf	2	0	3.5	inf
FU	inf	inf	inf	inf	inf	0	4.5
LA	inf	inf	inf	inf	5	inf	0

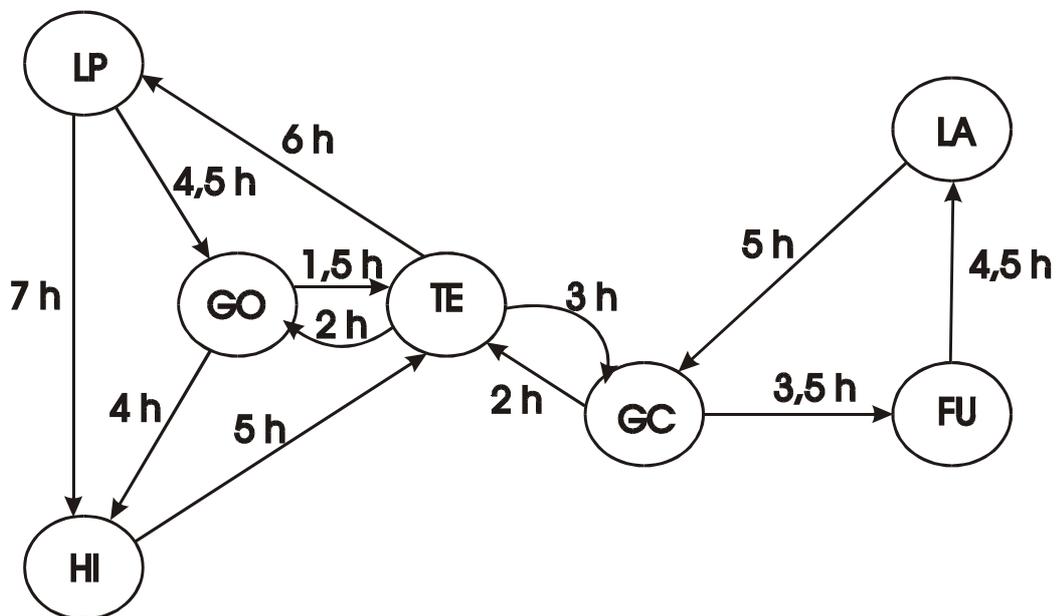


Fig.2.- Grafo correspondiente al desarrollo de la práctica

Además, nuestro sistema de control debe mantener un mecanismo que nos permita conocer los mejores caminos entre todas las islas (*algoritmos de Floyd*). Esta información debe mantenerse actualizada en todo momento, puesto que los tiempos de acceso entre las distintas islas pueden variar debido a condiciones climatológicas u otros factores, y esto genera la necesidad de recalcular las rutas de los barcos en un momento dado.

B.- Situar los distintos barcos (de pasajeros y de mercancías) en los puertos.

Llegados a este punto es necesario situar los barcos que posee nuestro empresario en los puertos en los que se encuentren, necesitando acceder a ellos de forma rápida (*árbol binario ordenado polimórfico*), lo que vendrá dado por el fichero **barcos.txt**. El formato de las líneas de este fichero será el siguiente:

*CreaBarcoP#id\_puerto#id\_barco#capac\_vip#capac\_turista*

*CreaBarcoM#id\_puerto#id\_barco#tonelaje\_max*

Como ya se ha comentado, existen dos tipos de **barcos**, los de *pasajeros* y los de *mercancías*. Todos ellos tienen un identificador (*id\_barco*) y la posibilidad de guardar información acerca del puerto de *origen* y del *destino* cuando zarpe el barco.

Los barcos de pasajeros (*CreaBarcoP*), deben contemplar, además de los datos anteriormente mencionados, información específica para controlar la *capacidad máxima* de pasajeros (tanto de clase *VIP* como de clase *turista*). Siempre se debe conocer el *número de pasajeros* que se encuentran en la actualidad en el barco, para las dos clases anteriormente mencionadas (*VIP* y *turista*). Por motivos de seguridad, este tipo de barcos también debe conservar una *lista* con la información relativa a los pasajeros que se encuentran en el barco.

Los barcos de mercancías (*CreaBarcoM*), por su parte, deben almacenar los datos relativos a su *tonelaje máximo*, así como al *tonelaje actual* si van a salir del puerto. También se necesita conocer el *número de contenedores* que transporta en ese momento. Evidentemente, también es necesario guardar información relativa a los contenedores que se encuentran *apilados* en las bodegas.

C.- Situar los contenedores de mercancías en puertos.

Por otro lado, los contenedores de mercancías se encuentran apilados en los puertos. La información correspondiente a los contenedores que tiene nuestro cliente vendrá dada por el fichero **contenedores.txt**. El formato de estas líneas será:

*CreaContenedor#id\_puerto#id\_contenedor#contenido#peso*

## **2.2. Utilización diaria del sistema:**

Para la utilización del sistema debemos basarnos en una cola de sucesos. Cada nodo de dicha cola se forma a partir de las líneas del fichero **eventos.txt** que se explica más adelante. No obstante debe quedar claro que las operaciones a realizar se resumen de la siguiente manera:

D.- Situar pasajeros o contenedores en los barcos que vayan a zarpar del puerto.

Estas operaciones reflejan el hecho de cargar un barco, bien sea de pasajeros o de contenedores, en base a los eventos *VentaBilleteT*, *VentaBilleteVIP* y *CargaContenedor*.

La información referente a los *pasajeros* debe registrar su código *identificador* y el *precio* del billete, así como la *clase* de pasajero, que puede ser únicamente *VIP* o *turista*. En la clase *VIP* es necesario especificar el número del *camarote* que tendrá asignado, así como la *empresa* a la que pertenece. Para la clase *turista*, la *edad* del pasajero es importante, puesto que puede influir en la distribución interna de los pasajeros dentro del barco.

Las mercancías se almacenan en **contenedores**, para los cuales se necesita registrar un código *identificador*, el *contenido* del mismo, así como su *peso*.

#### E.- Mantener la situación de los barcos en las distintas rutas marítimas.

Un barco inicia su camino al zarpar de un puerto. Esta información se comunica mediante el evento *SaleBarco*. El que un barco se desplace de un puerto a otro, significa que estará en una ruta marítima determinada (una *arista* del grafo, en cada momento) situándose los barcos que pasan por una misma ruta (arista) en una cola. Los barcos irán avanzando en su camino hacia el puerto de destino a través de diferentes rutas. Con el fin de reflejar el avance de los barcos dispondremos del evento *Tiempo*. La llegada al puerto de destino dependerá del tiempo que el barco emplea en su camino.

#### F.- Mantener la información de los barcos que van llegando a su destino.

Mediante esta operación cada puerto debe mantener la información de los barcos que van llegando a él para atracar. Se detecta que un barco ha llegado a su destino mediante el evento *Tiempo*.

#### G.- Mostrar el estado de un puerto

Se realizará esta operación al detectar un evento *MostrarPuerto*. Se debe indicar el número de contenedores almacenados en el puerto y el número de barcos que hay en el mismo. Y para cada barco se mostrará en pantalla toda la información referente a él, tal como el identificador del barco, el tipo de barco, así como el número de pasajeros embarcados, la capacidad del mismo, etc.

#### H.- Mostrar el estado de las rutas

Se debe saber en todo momento en qué situación se encuentra cada ruta. Esta información sirve para conocer la ubicación de los barcos que se encuentran de viaje. Para llevar a cabo esta operación existe el evento *MostrarRuta*. Y se deberá mostrar en pantalla la información de los barcos que se encuentran en la ruta indicada por el evento.

### **3. Descripción del fichero de eventos.**

Tal y como se ha comentado, los eventos quedan organizados en una cola. Dicha cola se construye a partir del fichero **eventos.txt** cuyas líneas siempre tendrán cualquiera de los formatos que se indican a continuación. Al atender cualquier evento de la cola se debe mostrar en pantalla toda la información correspondiente al evento en cuestión. Y una vez llevadas a cabo las tareas correspondientes a cada evento debe mostrarse en pantalla información acerca del desarrollo de éstas.

*VentaBilleteT#id\_puerto#id\_barco#id\_pasajero#edad#precio*

*VentaBilleteVIP#id\_puerto#id\_barco#id\_pasajero#camarote#empresa#precio*

Reflejan la petición de un ticket por parte del pasajero indicado con *id\_pasajero*. Si hay sitio en el barco se considera que el pasajero queda embarcado, se añade a la *lista polimórfica* de pasajeros que hay en el barco, en clase turista o en la clase VIP. Si no hubiera sitio para esa clase de pasajeros se debe comunicar mediante un mensaje en pantalla, informando de la imposibilidad de venta de más billetes en este barco, para esa clase de pasajeros.

Una vez vendido un billete se debe informar de este hecho enviando a la pantalla un mensaje indicando cuántos pasajeros han embarcado hasta el momento en el barco y los identificadores del pasajero que acaba de comprar billete y del barco correspondiente.

*CargaContenedor#id\_puerto#id\_barco#n°\_de\_contenedores*

Si el barco puede cargar el peso de los contenedores especificados, los elimina de la pila del puerto de la isla y los añade a la pila del barco, incrementando el tonelaje actual del mismo. Si no fuera posible la carga de todos los contenedores que se indican en el evento se debe informar de este hecho enviando un mensaje al monitor que indicará que este lote de contenedores no ha podido ser cargado en el barco. Es decir, si se especifica que deben cargarse tres contenedores y el barco sólo puede almacenar dos de ellos, se dejarán los tres contenedores en el puerto, puesto que el lote deseado de tres no puede cargarse en ese barco (no hay que cargar los que quepan, sino el lote entero).

Una vez cargado un lote de contenedores en un barco se debe informar del estado de la carga que lleva el barco, si ya ha alcanzado su límite de capacidad en lo que al peso se refiere...

*SaleBarco#id\_barco#id\_puerto\_origen#id\_puerto\_destino*

Tras la venta de billetes o la carga de contenedores, un barco se encuentra en disposición de zarpar. Este evento sitúa el barco especificado (*id\_barco*) en la cola de la arista que se encuentra entre el puerto de origen y la primera isla por la que va a transcurrir su ruta óptima (de menor coste en tiempo), antes de llegar a la isla de destino.

El evento elimina, en el puerto especificado, el barco del árbol de barcos que está bajo su control y lo añade a la cola correspondiente a los barcos en esa ruta.

Se debe mostrar en pantalla toda la información correspondiente al barco: Identificador, capacidad, situación actual de carga, origen, destino e identificadores de los contenedores que lleva o los pasajeros, según proceda.

*Clima#id\_puerto\_origen#id\_puerto\_destino#factor*

Este evento nos informa de efectos climatológicos que dificultan el paso por la ruta existente entre dos islas. *Factor* (que será un número real) indica el valor por el que se multiplica el tiempo que se tardaría en ir desde el puerto *id\_puerto\_origen* a *id\_puerto\_destino*. Mediante este evento se modifica la *matriz de adyacencia*, con lo que es preciso volver a calcular las *matrices de Floyd*. La matriz de adyacencia debe modificar el

valor indicado en las coordenadas especificadas por *id\_puerto\_origen* e *id\_puerto\_destino*, multiplicándolo por *factor*. Si *id\_puerto\_origen* contiene el carácter comodín “\*” significará que deben actualizarse todas las entradas a la isla *id\_puerto\_destino* (toda la columna de la matriz de adyacencia). Si es *id\_puerto\_destino* el que contiene el carácter comodín “\*”, deben actualizarse todas las salidas a la isla *id\_puerto\_origen* (toda la fila de la matriz de adyacencia). Si aparecen comodines en ambos campos debe modificarse la matriz entera.

*Tiempo#id\_puerto\_origen#id\_puerto\_destino*

Este evento sirve para simular el transcurso de las horas de viaje para cada ruta. El efecto tiene sobre el sistema es el de hacer que un barco avance una posición más en su camino hacia el puerto de destino, es decir, que pase de una arista del grafo a otra.

En realidad simulamos el efecto del tiempo sobre cada arista del grafo (*id\_puerto\_origen*, *id\_puerto\_destino*) haciendo que el primer barco que llegó a esta arista pase a la siguiente fase de su ruta. Para ello utilizará la información que nos proporcionan los algoritmos de Floyd.

Cuando *id\_puerto\_destino* coincide con el puerto de destino del barco que debería avanzar, se hará que dicho barco atraque en su puerto de destino. Esto conlleva que dicho barco se elimina de la ruta, pasando a formar parte del conjunto de barcos que se encuentran atracados en este puerto.

Si se trata de un barco de pasajeros, todos ellos deben desembarcar, informando de este hecho (para cada pasajero que desembarca) mediante mensajes en el monitor. Por supuesto, toda la información del barco, especialmente la referente a pasajeros, debe quedar actualizada.

Si por el contrario se tratara de un barco de mercancías, todos los contenedores deben quedar almacenados en el puerto. Y toda la información del barco referente a la mercancía debe actualizarse.

*MostrarPuerto#id\_puerto*

Este evento será atendido mostrando en la pantalla el identificador del puerto, el número de contenedores y de barcos que se encuentran en ese puerto, así como toda la información correspondiente a cada barco (identificador, tipo de barco, contenido, tanto si se trata de los pasajeros embarcados como de los contenedores cargados, capacidad, etc.).

*MostrarRuta#id\_puerto#id\_puerto*

Este evento será atendido mostrando en la pantalla la información de la ruta (tiempo asociado) así como la información de los barcos que están pasando en ese momento por la ruta indicada entre los dos puertos que señala el evento. Bastaría con mostrar sólo los identificadores de los barcos.

#### 4. Ficheros de ejemplo.

Se incluyen una serie de ficheros de ejemplos comentados.

##### RUTAS.TXT

*LP#HI#GO#TE#GC#FU#LA  
0#7#4.5#inf#inf#inf#inf  
inf#0#inf#5#inf#inf#inf  
inf#4#0#1.5#inf#inf#inf  
6#inf#2#0#3#inf#inf  
inf#inf#inf#2#0#3.5#inf  
inf#inf#inf#inf#inf#0#4.5  
inf#inf#inf#inf#5#inf#0*

##### BARCOS.TXT

*CreaBarcoP#Tenerife#B37C36J#20#80  
CreaBarcoP#Tenerife#C83B#25#100  
CreaBarcoP#Tenerife#CT754MJ#100#600  
CreaBarcoP#LaPalma#LJ38JH#10#60  
CreaBarcoP#Gomera#KJ3#15#75  
CreaBarcoP#Hierro#OEI38D#50#180  
CreaBarcoP#GranCanaria#MKD38K#25#100  
CreaBarcoP#GranCanaria#JK83UE#50#200  
CreaBarcoP#Lanzarote#LKDJ83J#80#120  
CreaBarcoP#Fuerteventura#ODI37IUE#32#88  
CreaBarcoM#Tenerife#LJKE89UE#300  
CreaBarcoM#Tenerife#DKI38J#500  
CreaBarcoM#GranCanaria#DKJEI8#400  
CreaBarcoM#GranCanaria#DK3E89#600*

##### CONTENEDORES.TXT

*CreaContenedor#Tenerife#DKEI8JE#Periodicos#150  
CreaContenedor#Tenerife#WICM38#Fruta#100  
CreaContenedor#Tenerife#EIP83JK8#Ropa#120  
CreaContenedor#GranCanaria#FUCNW738#Medicinas#125  
CreaContenedor#GranCanaria#VWI293X8#Platanos#200  
CreaContenedor#Lanzarote#DCIE623#Souvenirs#125*

EVENTOS.TXT

*VentaBilleteT#Tenerife#B37C36J#38739485#30#30*  
*VentaBilleteT#Tenerife#B37C36J#38738872#65#25*  
*VentaBilleteT#Tenerife#B37C36J#83628456#15#20*  
*VentaBilleteT#Tenerife#B37C36J#36483555#35#30*  
*VentaBilleteT#GranCanaria#JK83UE#83784642#22#30*  
*VentaBilleteT# GranCanaria#JK83UE#38562545#25#30*  
*VentaBilleteVIP#Tenerife#B37C36J#386295455#45#Anaya#65*  
*VentaBilleteVIP#Tenerife#B37C36J#93528451#32#Volkswagen#70*  
*VentaBilleteVIP#GranCanaria#JK83UE#93763245#42#Microsoft#75*  
*CargaContenedor#Tenerife#LJKE89UE#2*  
*CargaContenedor#GranCanaria#DKJEI8#1*  
*MostrarPuerto#GranCanaria*  
*SaleBarco#JK83UE#GranCanaria#Lanzarote*  
*MostrarPuerto#GranCanaria*  
*SaleBarco#DKJEI8#GranCanaria#Fuerteventura*  
*MostrarPuerto#GranCanaria*  
*MostrarRuta#GranCanaria#Fuerteventura // Debe haber 2 barcos*  
*Tiempo#GranCanaria#Fuerteventura*  
*MostrarPuerto#Fuerteventura // Ya está el barco de mercancías*  
*MostrarRuta#FuerteVentura#Lanzarote*  
*Tiempo#FuerteVentura#Lanzarote*  
*MostrarPuerto#Lanzarote // Ya está el barco de pasajeros*  
*MostrarPuerto#Tenerife*  
*SaleBarco#B37C36J#Tenerife#Hierro*  
*SaleBarco#LJKE89UE#Tenerife#LaPalma*  
*MostrarPuerto#Tenerife*  
*MostrarRuta#Tenerife#Gomera //Barco de pasajeros*  
*MostrarRuta#Tenerife#LaPalma // Barco de mercancías*  
*Clima#Gomera#Hierro#10*  
*Tiempo#Tenerife#Gomera // Cambia de arco TE →GO a GO →TE*  
*MostrarRuta#Gomera#Tenerife*  
*MostrarRuta#Tenerife#LaPalma // Está el barco de mercancías*  
*Tiempo#Gomera#Tenerife*  
*MostrarRuta#Tenerife#LaPalma // Ahora están los dos barcos*  
*Tiempo#Tenerife#LaPalma*  
*MostrarPuerto#Hierro // Ya está el barco de mercancías*  
*Tiempo#LaPalma#Hierro*  
*MostrarPuerto#Hierro // Ya está el barco de pasajeros*