

Memoria PECL2

Estructuras de Datos

Intercambiador



Universidad
de Alcalá

Pedro Barquín Ayuso
Wasim El Hallak Díez
3º GIC Curso 15/16

ÍNDICE

<i>Descripción TAD's implementados</i>	3
<i>Lista (Tren):</i>	3
<i>Nodo (Vagón o locomotora):</i>	4
<i>Definición de operaciones de los TAD's</i>	4
<i>Lista (Tren):</i>	4
<i>Nodo (Vagón o locomotora):</i>	6
<i>Explicación funcionamiento del programa y métodos más importantes</i>	6
<i>Método main:</i>	6
<i>Capturas de pantalla:</i>	7
<i>Método intercambiar:</i>	12
<i>Problemas encontrados y solución adoptada</i>	12

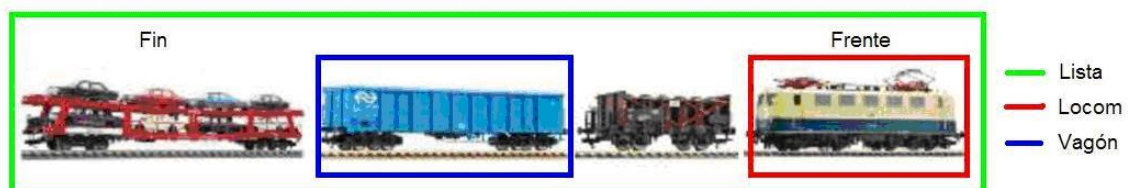
Descripción TAD's implementados

Lista (Tren):

```
class Lista
{
public:
    Lista() : frente(NULL), fin(NULL), actual(NULL), tamaño(0) {} //constructor con fin y
    frente a NULL ya que empieza con 0 nodos y el numero de cola que es
    ~Lista(); //destructor
    void insertarNodoVagon(int v, char *cV, char *o, char * d, bool valor);
    //funciones insercion del vagon
    void insertarNodoTren(int v, char *cV, int nT, char *o, char * d, bool valor);
    //insercion de la locomotora
    void insertarFinal(pnodo auxiliar); //inserta un nodo concreto al final de un tren
    pnodo borrarNodo(int v); //elimina un nodo concreto (numero de vagon)
    bool listaVacía(); //comprueba si esta vacía
    void mostrarLista(); //muestra todo el tren (llama a mostrar
    contenido)
    void esSiguiente();
    void esPrimero();
    void esUltimo();
    bool esActual();
    int valorActual(); //numero del vagon en el que
    estamos
    int numTren(); //numero del tren
    char *origenTren(); //info origen tren
    char *destinoTren(); //info destino tren
    char *destinoVagon(); //info destino vagon
    void mostrarContenido(pnodo x); //muestra los valores del nodo
    int tamañoTren(); //da el tamaño total del tren
    bool comprobarVagones(); //comprueba que no hay ningun vagon erroneo si
    es asi false si no true
    int vagonErroneoUltimo(); //da el numero del vagon erroneo ultimo
    para su busqueda
    char *destinoVagonPosicion(int x); //devuelve el destino del vagon pasado como
    parametro
    int numeroCargas(char *y); //devuelve y muestra la cantidad e informacion de los
    vagones con esa carga

private:
    pnodo frente, fin, actual; //definimos tres pnodo frente, fin y
    actual
    int tamaño; //variable para el tamaño
};
```

La clase Lista se creó con tres punteros (frente, fin y actual) para un más fácil manejo y el atributo tamaño, y una serie de métodos que harán posible la gestión del intercambio de vagones entre otras cosas.



Nodo (Vagón o locomotora):

```
class Nodo //Declaracion del nodo que
forman las colas
{
private: //Variables y punteros a informacion
    int nVagon;
    char *carga;
    int nTren;
    bool locomotora;
    char *origen;
    char *destino;
    Nodo *siguiente; //puntero al siguiente
    friend class Lista; //clase amiga

public:
    Nodo(int v, char *cV, int nT, char *o, char * d, bool valor, Nodo *sig = NULL)
    //constructor con la informacion y *sig se pasa como inicial NULL y luego va cambiando
    {
        nVagon = v; //variables con la informacion
        del tren
        carga = cV;
        nTren = nT;
        locomotora = valor;
        origen = o;
        destino = d;
        siguiente = sig;
    }
};
typedef Nodo *pnodo; //un *Pnodo es un tipo definido de nodo
```

El nodo se decidió crearlo con varios atributos y el puntero al siguiente nodo en una sola estructura de tal manera que nVagon y carga no se utilizan si locomotora es True.

Definición de operaciones de los TAD's

Lista (Tren):

-Creación y destrucción: Se crea la lista vacía (frente a NULL, fin a NULL, actual a NULL y tamaño a 0) y más tarde se rellena con la locomotora y después con los vagones, en el caso del destructor elimina la lista.

-insertarNodoVagon: Añade un nuevo nodo en este caso vagón el cual recibe el número del tren de aquél al que se engancha, se añaden a partir del tren.

-insertarNodoTren: Añadimos el nodo tren que es el primero en añadirse, metiendo los parámetros necesarios.

-insertarFinal: Añade un nodo concreto al final.

- borrarNodo: Elimina uno de los nodos.
- listaVacía: Comprueba si está vacía la lista.
- mostrarLista: Muestra todo el contenido por medio del método mostrarContenido, que muestra la información de un nodo.
- esSiguierte: Avanza al siguiente el nodo actual.
- esPrimero: Coloca actual en el primer vagón.
- esUltimo: Coloca actual en el último vagón.
- esActual: Devuelve True si es el actual.
- valorActual: Devuelve el número de vagón del nodo actual.
- numTren: Devuelve el número de la locomotora.
- origenTren: Devuelve el origen de la locomotora.
- destinoTren: Devuelve el destino de la locomotora.
- destinoVagon: Devuelve el destino del vagón.
- mostrarContenido: Muestra los datos en función de si es vagón o locomotora.
- tamañoTren: Devuelve el tamaño del tren.
- comprobarVagones: Devuelve True si hay algún vagón con diferente destino que la locomotora.
- vagonErroneoUltimo: Devuelve el número del último vagón erróneo.
- destinoVagonPosicion: Dado el número de vagón como parámetro, devuelve el destino del primer vagón que coincida con dicho número.
- numeroCargas: Dada la carga como parámetro, devuelve la cantidad de vagones que la contienen y muestra dichos vagones.

Nodo (Vagón o locomotora):

-Creación y destrucción: Se crea el nodo con todos sus parámetros y siguiente a NULL, modificándose más tarde, si el atributo locomotora es True, será locomotora, si no será un vagón. La destrucción de un nodo se lleva a cabo con el método borrarNodo.

Explicación funcionamiento del programa y métodos más importantes

Se trata de una simulación del intercambio de vagones entre trenes de mercancías, cada uno de los vagones, además del tipo de carga, consta de un número de identificación individual, un número de identificación del tren que lo transporta, una estación de origen y una estación de destino. Mientras el tren de mercancías está en ruta, en el centro de intercambio se analizan estos datos para cada vagón y se decide a qué tren de los disponibles debe trasvasarse cuando llegue a la central de intercambio. Cada tren será simulado con una lista dinámica cuyos nodos serán vagones o locomotoras.

Método main:

```
int main() {
    //Random
    srand(time(NULL));

    int valorMenu = 9;

    //Creacion de las locomotoras
    tren1.insertarNodoTren(-1, NULL, 124, o1, d1, true);
    tren2.insertarNodoTren(-1, NULL, 456, o2, d2, true);
    tren3.insertarNodoTren(-1, NULL, 224, o3, d3, true);
    tren4.insertarNodoTren(-1, NULL, 801, o4, d4, true);
    tren5.insertarNodoTren(-1, NULL, 652, o5, d5, true);

    //creacion y asignacion de los contenedores automaticamente
    crearContenedores();

    while (valorMenu!=0)
    {
        cout << "MENU RENFE" << endl;
        cout << endl;
        cout << "1: Intercambiar trenes" << endl;
        cout << "2: Informacion trenes" << endl;
        cout << "0: Salir" << endl;
    }
```

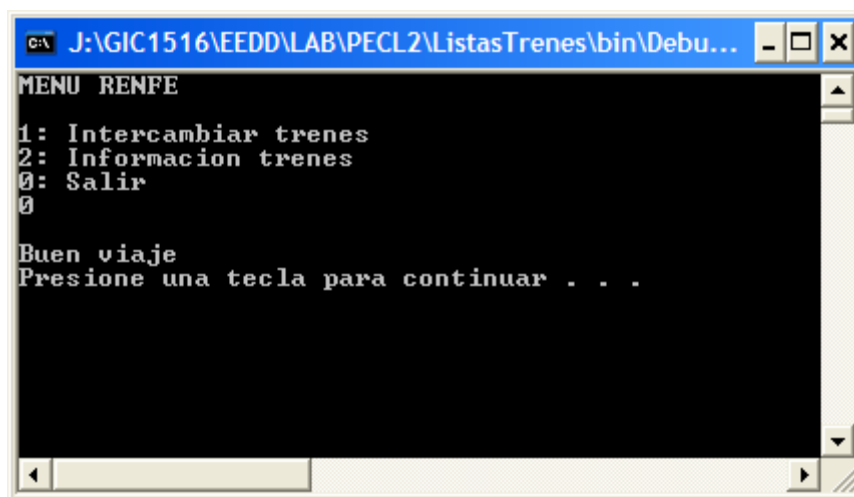
```

        cin >> valorMenu;
        cout << endl;
        switch (valorMenu)
        {
        case 1: intercambiar(); break;
        case 2: cout << "MENU Informacion" << endl;
                cout << endl;
                cout << "1: Mostrar trenes" << endl;
                cout << "2: Numero de trenes con coches" << endl;
                cout << "3: Numero de trenes con comida" << endl;
                cout << "4: Numero de trenes con medicamentos" << endl;
                cout << "5: Numero de trenes con combustible" << endl;
                cout << "6: Numero de trenes con dinero" << endl;
                cin >> valorMenu;
                cout << endl;
                switch (valorMenu)
                {
                case 1: tren1.mostrarLista();
                        tren2.mostrarLista();
                        tren3.mostrarLista();
                        tren4.mostrarLista();
                        tren5.mostrarLista();
                        break;
                case 2: cantidadCarga(c1);break;
                case 3: cantidadCarga(c2);break;
                case 4: cantidadCarga(c5);break;
                case 5: cantidadCarga(c3);break;
                case 6: cantidadCarga(c4);break;
                default: break;
                }
                break;
        default: cout << "Buen viaje" << endl;break;
        }
    }

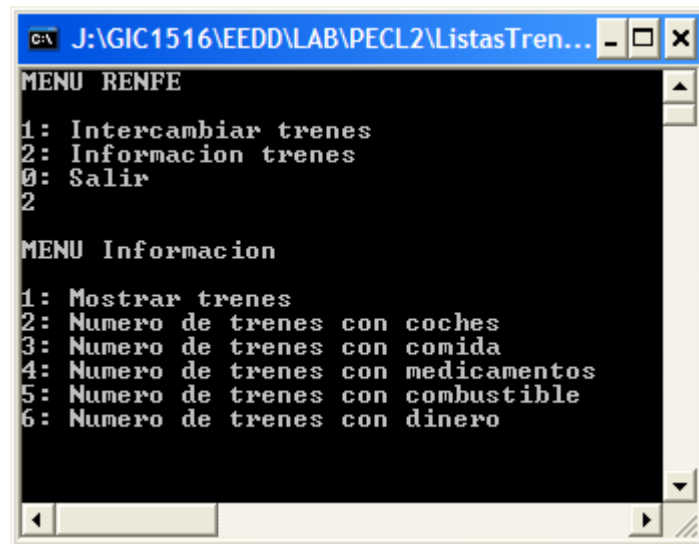
    system("Pause");
    return 0;
}

```

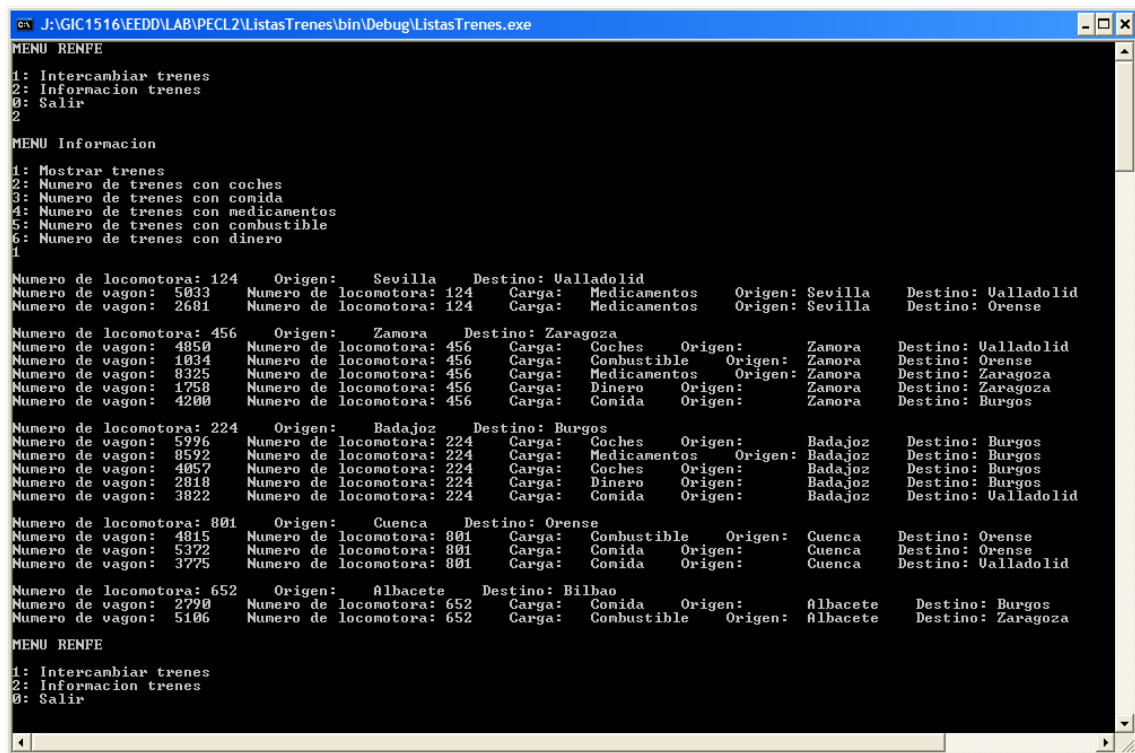
Capturas de pantalla:



Al ejecutar el programa aparece un menú dando las opciones que aparecen.



La segunda opción lleva a un submenú que ofrece las opciones que se muestran.




```
J:\GIC1516\EEDD\LAB\PECL2\ListasTrenes\bin\Debug\ListasTrenes.exe
MENU RENFE
1: Intercambiar trenes
2: Informacion trenes
0: Salir
2

MENU Informacion
1: Mostrar trenes
2: Numero de trenes con coches
3: Numero de trenes con comida
4: Numero de trenes con medicamentos
5: Numero de trenes con combustible
6: Numero de trenes con dinero
2

Numero de vagon: 2332    Numero de locomotora: 652    Carga: Coches    Origen: Albacete    Destino: Orense
Numero de vagon: 6746    Numero de locomotora: 652    Carga: Coches    Origen: Albacete    Destino: Bilbao
Numero de vagon: 9487    Numero de locomotora: 652    Carga: Coches    Origen: Albacete    Destino: Orense
El numero de vagones con la carga Coches es de 3
MENU RENFE
1: Intercambiar trenes
2: Informacion trenes
0: Salir
2

MENU Informacion
1: Mostrar trenes
2: Numero de trenes con coches
3: Numero de trenes con comida
4: Numero de trenes con medicamentos
5: Numero de trenes con combustible
6: Numero de trenes con dinero
3

Numero de vagon: 9047    Numero de locomotora: 124    Carga: Comida    Origen: Sevilla    Destino: Bilbao
Numero de vagon: 6727    Numero de locomotora: 124    Carga: Comida    Origen: Sevilla    Destino: Orense
Numero de vagon: 5101    Numero de locomotora: 456    Carga: Comida    Origen: Zamora    Destino: Orense
Numero de vagon: 2513    Numero de locomotora: 456    Carga: Comida    Origen: Zamora    Destino: Bilbao
El numero de vagones con la carga Comida es de 4
MENU RENFE
1: Intercambiar trenes
2: Informacion trenes
0: Salir
-
```

Las demás opciones del submenú ofrecen una información añadida al programa y se trata de mostrar por pantalla la cantidad de vagones con la carga elegida, mostrando además las características de dichos vagones.

```
C:\J:\GIC1516\EEDD\LAB\PECL2>ListasTrenes\bin\Debug>ListasTrenes.exe
1: Intercambiar trenes
2: Informacion trenes
0: Salir
1
Examinamos el tren 1
Error en el tren 1
Numero de vagon erroneo: 9191 Con destino a Orense
Colocado en el tren 4 con destino a Orense
Error en el tren 1
Numero de vagon erroneo: 3845 Con destino a Burgos
Colocado en el tren 3 con destino a Burgos
Error en el tren 1
Numero de vagon erroneo: 9654 Con destino a Orense
Colocado en el tren 4 con destino a Orense
Tren 1 examinado
Examinamos el tren 2
Error en el tren 1
Numero de vagon erroneo: 5046 Con destino a Valladolid
Colocado en el tren 1 con destino a Valladolid
Error en el tren 1
Numero de vagon erroneo: 3737 Con destino a Bilbao
Colocado en el tren 5 con destino a Bilbao
Error en el tren 1
Numero de vagon erroneo: 3219 Con destino a Burgos
Colocado en el tren 3 con destino a Burgos
Error en el tren 1
Numero de vagon erroneo: 4947 Con destino a Bilbao
Colocado en el tren 5 con destino a Bilbao
Tren 2 examinado
Examinamos el tren 3
Error en el tren 3
Numero de vagon erroneo: 1299 Con destino a Orense
Colocado en el tren 4 con destino a Orense
Error en el tren 3
Numero de vagon erroneo: 7321 Con destino a Orense
Colocado en el tren 4 con destino a Orense
Error en el tren 3
Numero de vagon erroneo: 3803 Con destino a Bilbao
Colocado en el tren 5 con destino a Bilbao
Error en el tren 3
Numero de vagon erroneo: 5239 Con destino a Zaragoza
Colocado en el tren 2 con destino a Zaragoza
Tren 3 examinado
Examinamos el tren 4
Error en el tren 4
Numero de vagon erroneo: 7373 Con destino a Bilbao
Colocado en el tren 5 con destino a Bilbao
Error en el tren 4
Numero de vagon erroneo: 5377 Con destino a Bilbao
Colocado en el tren 5 con destino a Bilbao
Tren 4 examinado
Examinamos el tren 5
Error en el tren 5
Numero de vagon erroneo: 1849 Con destino a Valladolid
Colocado en el tren 1 con destino a Valladolid
Error en el tren 5
Numero de vagon erroneo: 4027 Con destino a Burgos
Colocado en el tren 3 con destino a Burgos
Error en el tren 5
Numero de vagon erroneo: 7187 Con destino a Orense
Colocado en el tren 4 con destino a Orense
Error en el tren 5
Numero de vagon erroneo: 3459 Con destino a Zaragoza
Colocado en el tren 2 con destino a Zaragoza
Tren 5 examinado
FIN DEL INTERCAMBIO
```

Al llegar al intercambiador con la opción 1 del menú, se rastrean todos los vagones de cada tren colocándolos en su lugar.

```

J:\GIC1516\EEDD\LAB\PECL2>ListasTrenes\bin\Debug>ListasTrenes.exe
Numero de vagon erroneo: 5377 Con destino a Bilbao
Colocado en el tren 5 con destino a Bilbao
Tren 4 examinado

Examinamos el tren 5
Error en el tren 5
Numero de vagon erroneo: 1849 Con destino a Valladolid
Colocado en el tren 1 con destino a Valladolid
Error en el tren 5
Numero de vagon erroneo: 4027 Con destino a Burgos
Colocado en el tren 3 con destino a Burgos
Error en el tren 5
Numero de vagon erroneo: 7187 Con destino a Orense
Colocado en el tren 4 con destino a Orense
Error en el tren 5
Numero de vagon erroneo: 3459 Con destino a Zaragoza
Colocado en el tren 2 con destino a Zaragoza

Tren 5 examinado
FIN DEL INTERCAMBIO
MENU RENFE

1: Intercambiar trenes
2: Informacion trenes
0: Salir
2

MENU Informacion
1: Mostrar trenes
2: Numero de trenes con coches
3: Numero de trenes con comida
4: Numero de trenes con medicamentos
5: Numero de trenes con combustible
6: Numero de trenes con dinero
1

Numero de locomotora: 124 Origen: Sevilla Destino: Valladolid
Numero de vagon: 8835 Numero de locomotora: 124 Carga: Dinero Origen: Sevilla Destino: Valladolid
Numero de vagon: 5119 Numero de locomotora: 124 Carga: Coches Origen: Sevilla Destino: Valladolid
Numero de vagon: 5046 Numero de locomotora: 124 Carga: Comida Origen: Zamora Destino: Valladolid
Numero de vagon: 1849 Numero de locomotora: 124 Carga: Dinero Origen: Albacete Destino: Valladolid

Numero de locomotora: 456 Origen: Zamora Destino: Zaragoza
Numero de vagon: 5239 Numero de locomotora: 456 Carga: Medicamentos Origen: Badajoz Destino: Zaragoza
Numero de vagon: 3459 Numero de locomotora: 456 Carga: Dinero Origen: Albacete Destino: Zaragoza

Numero de locomotora: 224 Origen: Badajoz Destino: Burgos
Numero de vagon: 6278 Numero de locomotora: 224 Carga: Dinero Origen: Badajoz Destino: Burgos
Numero de vagon: 3845 Numero de locomotora: 224 Carga: Combustible Origen: Sevilla Destino: Burgos
Numero de vagon: 3219 Numero de locomotora: 224 Carga: Combustible Origen: Zamora Destino: Burgos
Numero de vagon: 4027 Numero de locomotora: 224 Carga: Coches Origen: Albacete Destino: Burgos

Numero de locomotora: 801 Origen: Cuenca Destino: Orense
Numero de vagon: 8514 Numero de locomotora: 801 Carga: Comida Origen: Cuenca Destino: Orense
Numero de vagon: 4426 Numero de locomotora: 801 Carga: Combustible Origen: Cuenca Destino: Orense
Numero de vagon: 7213 Numero de locomotora: 801 Carga: Medicamentos Origen: Cuenca Destino: Orense
Numero de vagon: 7191 Numero de locomotora: 801 Carga: Dinero Origen: Sevilla Destino: Orense
Numero de vagon: 9654 Numero de locomotora: 801 Carga: Coches Origen: Sevilla Destino: Orense
Numero de vagon: 1299 Numero de locomotora: 801 Carga: Combustible Origen: Badajoz Destino: Orense
Numero de vagon: 7321 Numero de locomotora: 801 Carga: Medicamentos Origen: Badajoz Destino: Orense
Numero de vagon: 7187 Numero de locomotora: 801 Carga: Medicamentos Origen: Albacete Destino: Orense

Numero de locomotora: 652 Origen: Albacete Destino: Bilbao
Numero de vagon: 3737 Numero de locomotora: 652 Carga: Medicamentos Origen: Zamora Destino: Bilbao
Numero de vagon: 4947 Numero de locomotora: 652 Carga: Dinero Origen: Zamora Destino: Bilbao
Numero de vagon: 3803 Numero de locomotora: 652 Carga: Dinero Origen: Badajoz Destino: Bilbao
Numero de vagon: 7373 Numero de locomotora: 652 Carga: Medicamentos Origen: Cuenca Destino: Bilbao
Numero de vagon: 5377 Numero de locomotora: 652 Carga: Medicamentos Origen: Cuenca Destino: Bilbao

MENU RENFE
1: Intercambiar trenes
2: Informacion trenes
0: Salir
-

```

Al mostrar los trenes después del intercambiador, se ve que ahora los orígenes son diferentes pero los destinos iguales.

Método intercambiar:

```
void intercambiar() {
    int nVagon = 0;
    char *auxDestino = NULL;
    pnode auxVagon;

    cout << "Examinamos el tren 1" << endl;
    while (tren1.comprobarVagones())//mientras tenga vagones error repetir
    {
        cout << "Error en el tren 1" << endl;
        nVagon = tren1.vagonErroneoUltimo();    //sacamos cual es el que esta mal
        auxDestino = tren1.destinoVagonPosicion(nVagon);
        cout << "Numero de vagon erroneo: " << nVagon << " Con destino a "<< auxDestino
<< endl;
        auxVagon = tren1.borrarNodo(nVagon);    //eliminamos el vagon del tren (aux) y
recuperamos dicho nodo para su posterior adición a otro tren PROBAR
        cout << "" << endl;
        if (auxDestino == tren2.destinoTren())
        {
            cout << "Colocado en el tren 2 con destino a " << tren2.destinoTren() <<
endl;
            tren2.insertarFinal(auxVagon);
        }
        if (auxDestino == tren3.destinoTren())
        {
            cout << "Colocado en el tren 3 con destino a " << tren3.destinoTren() <<
endl;
            tren3.insertarFinal(auxVagon);
        }
        if (auxDestino == tren4.destinoTren())
        {
            cout << "Colocado en el tren 4 con destino a " << tren4.destinoTren() <<
endl;
            tren4.insertarFinal(auxVagon);
        }
        if (auxDestino == tren5.destinoTren())
        {
            cout << "Colocado en el tren 5 con destino a " << tren5.destinoTren() <<
endl;
            tren5.insertarFinal(auxVagon);
        }
    }
    cout << "Tren 1 examinado" << endl;
    cout << endl;

    cout << "Examinamos el tren 2" << endl;
    while (tren2.comprobarVagones())//mientras tenga vagones error repetir
    ...
    ...
    ...
}
```

Problemas encontrados y solución adoptada

El mayor problema que ha surgido ha sido el del intercambio de vagones, ya que la gestión de cómo hacerlo era un poco laboriosa y enrevesada dependiendo de cómo se planteara. Finalmente se decidió ir tren a tren detectando vagones erróneos y colocándolos en su tren correspondiente.