ANALISIS DEL PROYECTO

"GTR – *MERCEDES AMG"*

***AUTORES*:**
Erick Johan Perales Quispe, N00204989@upn.pe
Alvaro Sebastian Neciosup Rivera, N00115103@upn.pe
Samir Renzo Huamani Quispe, N00166838@upn.pe
Carlos Alberto Alarcon Quijayte, N00251907@upn.pe
Diego Alonso Diestra Vega, N00211342@upn.pe
Jailene Milagros García Candela, N00202965@upn.pe

ASESOR:
Ing. Juan Ricardo Tapia Carbajal, juan.tapia@upn.edu.pe

UNIVERSIDAD PRIVADA DEL NORTE

Facultad de Ingeniería
Av. El Sol 461, San Juan de Lurigancho 15434
Teléfono (+76) 602525

Alvaro Sebastian Neciosup Rivera

```
srand(time(NULL));  //2
int contA = 1; //1
int auxcodigo_Car;
int auxcod_ChE;                          //2+1 = 3
string auxcod_CarE;
string auxcontenido_Car, auxdestino_Car, auxestado_Car, auxhorario_Car;
double auxPeso,auxCosto;                                       // 3 + 146+ 5n logn +5logn = 149+5n logn+5logn

for(int i = 0; i < ALEA ;i++){ //2n+2                 => (73 + 5log n)*(2n+2)=146+5n log n+5log n
    string contenido[8] = {"Carnes","Material de construccion","Telas","Frutas","Muebles","Agua","Autos","Animales"}; //1
    string destinos[10] ={"Trujillo","Lima","Cuzco","Cuzco","Arequipa","Madre de Dios","Loreto","Puno","Tumbes","Ica"}; //1
    string horas[24] ={"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20","21","22","23","00"};//1
    string minutos[60]={"00","01","02","03","04","05","06","07","08","09","10","11","12","13","14","15","16","17","18","19","20","21","22","23","24","25","26".
     //1

    auxcodigo_Car =10000 + rand()%(99999); //4
    auxcontenido_Car = contenido[0+rand()%(8)]; //5
    auxdestino_Car = destinos[0+rand()%(10)]; //5
    auxestado_Car = "No se entrego";//1
    auxhorario_Car = horas[0+rand()%(23)] + ":"+minutos[1+rand()%(60)];//11
    double decimal = 0+rand()%(58); // 4
    auxPeso = 110+rand()%(1000) + (0.10*decimal); //6
    auxCosto = auxPeso*4.5; //2
    carga[i] = auxcodigo_Car;//2
                                   //1+1+1+1+4+5+5+1+11+4+6+2+2+3+4+1+1=53
    if(contA % 2 != 0){ //3                                      //
    int codigo_Entrega = 10000+ rand()%(99999); //4            //
                                                             //
    auxcod_ChE = 0;//1
    do{//1
     auxcod_ChE = choferes[0+rand()%(10)]; //5       logn+1+1 => longn+2(5) = 5long n +5        |  53 + 5log n + 5 +15 = 73 + 5long n
    }while(auxcod_ChE == 0); //Log n +1
                                                   //
    ostringstream o;                            //
    o << auxcodigo_Car;//1
    auxcod_CarE = o.str();//2          ------------1+2+2+1+2+3+1+1+1 = 15

    string *valores = aleatorioListaEntrega(listaTrabajador, auxcod_ChE);//2
    if/*1*/(insertarListaEntrega(listaEntrega, codigo_Entrega,*(valores+0), *(valores+1), auxcod_CarE, auxcontenido_Car, auxCosto,"Se entrego")){//3
        auxestado_Car = "Se entrego";   //1
    }
    }

    insertarListaCarga(listaCarga, auxcodigo_Car, auxcontenido_Car, auxdestino_Car, auxestado_Car, auxhorario_Car,auxPeso,auxCosto);//1
    contA++;//1
}
```

F(n) = 149 +5n logn + 5logn

O (F(n)) = O(149 + 5nlogn + 5logn) ➔ el O distribuir

O(F(n)) = O(149) + O(5nlogn) + O(5logn) ➔ Constante

O(F(n)) = O(1) +O(logn)+O(logn) ➔ O mayor

O(F(n)) = O(logn) ➔ la complejidad algorítmica es en base a logn.

Carlos Alberto Alarcon quijayte

```
void ordenarListaCamion(NodoCamion *&listaCamion)
{
    NodoCamion *actual = new NodoCamion();//------>2
    actual = listaCamion;//------>3
    NodoCamion *siguiente = new NodoCamion(); //------>2
     int C;
     string P;
     while(actual->sgte != NULL)//------>log (n)+2
     {//-------------------------------log (n)----------------------------------------------------------------------------------------------------
          siguiente = actual->sgte;//------>3log (n)

          while(siguiente!=NULL)//------>log (n)+2
          {//-------------------------------log (n).log (n)-------------------------------------------------------------------
               if(actual->Codigo_camion > siguiente->Codigo_camion)//------>4log (n).log (n)
               {
                    C = siguiente->Codigo_camion;//------>2log (n).log (n)
                    siguiente->Codigo_camion = actual->Codigo_camion;//------>3log (n).log (n)
                    actual->Codigo_camion = C;//------>3log (n).log (n)
                    P = siguiente->Placa_camion ;//------>2log (n).log (n)
                    siguiente->Placa_camion = actual->Placa_camion ;//------>3log (n).log (n)
                    actual->Placa_camion  = P;//------>3log (n).log (n)
               }
               siguiente = siguiente->sgte;//------>3log (n).log (n)
          }//-------------------------------log (n).log (n)-------------------------------------------------------------------
          actual = actual->sgte;//------>3log (n)
          siguiente = actual->sgte;//------>3log (n)
      //-------------------------------log (n)----------------------------------------------------------------------------------------------------
     }
}
```

F(n) = $24(\log(n))^2 + 12\log(n)+9$

O (F(n)) = O ($24(\log(n))^2 + 12\log(n)+9$) ➔ el O distribuir

O(F(n)) = O ($24(\log(n))^2$) + O ($12\log(n)$) + O (9) ➔ Constante

O(F(n)) = O ($(\log(n))^2$) + O ($\log(n)$)+O (1) ➔ O mayor

O(F(n)) = O ($(\log(n))^2$)

Samir Renzo Huamani Quispe

```
void pagarCarga(double precio)
{
    const int TAM=14;
    float Monedas [TAM]={200,100,50,20,10,5,2,1,0.5,0.2,0.1,0.05,0.02,0.01};
    int Acumulado[TAM]={0,0,0,0,0,0,0,0,0,0,0,0,0,0};
    float saldo;
    int indice=0;
    double pago_actual;
    float vuelto;
    cout<<"El precio a pagar es de "<<precio<<" desearia pagar con cuanto?"<<endl;
    cin>>pago_actual;

    vuelto = pago_actual-precio;
    if(vuelto>0){
    //se imprime los datos del precio por paquete y el monto que pago el usuario
    cout<<"\nEl pago del usuario fue de: S/. "<<pago_actual<<endl;
    cout<<"Precio de la Carga: S/. "<<precio<<endl;
    cout<<"\n Vuelto "<<vuelto<<endl;

        while(vuelto>0){
                if(vuelto>=Monedas[indice]){
                    Acumulado[indice]++;
                    vuelto=round((vuelto-Monedas[indice])*100)/100;
                } else{ indice++; }
        }
        for(int i=0; i<TAM;i++){
            if(Acumulado[i]>0){
                if(Monedas [i]==10 || Monedas [i]==20 || Monedas [i]==50 || Monedas [i]==100 || Monedas [i]==200){
                    cout<<"Billete de "<<Monedas [i]<<"\t => "<<Acumulado[i]<<endl;
                }
                else{
                    cout<<"Monedas de "<<Monedas [i]<<"\t => "<<Acumulado[i]<<endl;
                }

            }
        }
    }//fin del if
    else{
        cout<<"\nERROR";
    }
        getch();
}
```

F(n)=11+11+13logn+2+33n+2+2

O(F(n))= O(28+13logn + 33n) ->dist

O(F(n))= O(28)+O(13logn)+O(33n) ->cons

O(F(n))= O(1)+O(logn)+O(n) ->mayor

O(F(n)=O(n)

Diego Alonso Diestra Vega

```
NodoCamion *aux = new NodoCamion();    //2
NodoCamion *aux2 = new NodoCamion();   //2
NodoCamion *devolver = new NodoCamion(); //2
aux = listaCamion;//3
int inferior = 0/*1*/,mitad,mid,sup,inf;
sup = indiceC;//1                                   //=> 2+2+2+3+1+1+1+1+3 = 17
int numitad;
char band='F';//1
inf = 0;//1

if(listaCamion!= NULL){//3
    while(inf<=sup){//log n +1 -----(12logn +47)(logn + 1) = 12log^2n + 59logn + 47----------todo dentro del while
        mid =(inf+sup)/2;//3

        while(aux!= NULL){//log n +1----------5logn +8 + 2 + 2 + 1 + 1 + 1 + logn + 3 + logn + 3 + logn +1 + 3 = 7logn +21//
            if(aux->indice == mid){//3
            numitad = aux->Codigo_camion;//2
            aux2 = aux;//1                          => dentro del if 2+1 = 3 se repirte una vez
            }
           aux = aux->sgte;//2             => el if con el siguiente 3+2 = 5 se repite log n+1 veces => 5log n +5 +3 = 5logn + 8
        }

        aux = listaCamion;//3
        if(numitad == buscar){//2
        band = 'V';//1
        devolver = aux2;//1
        break;//1
        }
    if(numitad>buscar){//2                                  //todo lo de dentro del while es 12 logn + 47
        sup = mid;//1
        mid =(inf+sup)/2;//3
        while(aux != NULL){//Logn + 1
            if(aux->indice == mid){//3
                numitad = aux->Codigo_camion;//2
            }
            aux = aux->sgte;//2               =>logn +1 * 3+2 = 5logn +5 + 2 + 3 + 1 + 2 + 3=5logn + 16 //
        }
        aux=listaCamion;//3
    }
    if(numitad<buscar){//2
        inf = mid;//1
        mid =(inf+sup)/2;//3
        while(aux != NULL){//logn+1
            if(aux->indice == mid){//3
                numitad = aux->Codigo_camion;//2
            }
            aux = aux->sgte;//2                 => logn + 1 * 5 = 5logn +5 + 2 + 3 + 1 + 2 +3 =5logn + 16
        }
        aux=listaCamion;//3
    }

    }

if(band == 'V'){//2
        return devolver;//1
}else{//1                           //=> 2+ 1 +1 +1 = 5                => 17 + 5 + 12log^2n + 59logn + 47 = F(n)=12log^2n + 59 logn + 68//
        return NULL;//1
}

}
```

F(n) = 12log^2n + 59 logn + 68

O(f(n))= O(12log^2n) + O(59 logn) + O(68)

O(f(n)) = O(log^2n) +O(logn) + O (1) O(f(n))=

O(log^2n) => O mayor

Erick Johan Pareles Quispe

```
void ordenarListaCarga(NodoCarga *&listaCarga)
{
    NodoCarga *actual = new NodoCarga();//------>2
    actual = listaCarga;//------>3
    NodoCarga *siguiente = new NodoCarga();//------>2
     int C;
     string Cont;
     string D;
     string E;
     string H;
     while(actual->sgte != NULL)//------>log2(n)+2
     {//-----------------------------log2(n)----------------------------------------------------------------------
          siguiente = actual->sgte;//------>3log2(n)

          while(siguiente!=NULL)//------>log2(n)+2
          {//-----------------------------log2(n).log2(n)----------------------------------------------------------------------
               if(actual->codigo_Car > siguiente->codigo_Car)
               {
                    C = siguiente->codigo_Car ;//------>2log2(n).log2(n)
                    siguiente->codigo_Car = actual->codigo_Car ;//------>3log2(n).log2(n)
                    actual->codigo_Car = C;//------>3log2(n).log2(n)
                    Cont = siguiente->contenido_Car;//------>2log2(n).log2(n)
                    siguiente->contenido_Car = actual->contenido_Car ;//------>3log2(n).log2(n)
                    actual->contenido_Car = Cont;//------>3log2(n).log2(n)
                    D = siguiente->destino_Car;//------>2log2(n).log2(n)
                    siguiente->destino_Car = actual->destino_Car ;//------>3log2(n).log2(n)
                    actual->destino_Car = D;//------>3log2(n).log2(n)
                    E = siguiente->estado_Car;//------>2log2(n).log2(n)
                    siguiente->estado_Car = actual->estado_Car ;//------>3log2(n).log2(n)
                    actual->estado_Car = E;//------>3log2(n).log2(n)
                  H = siguiente->horario_Car;//------>2log2(n).log2(n)
                    siguiente->horario_Car = actual->horario_Car ;//------>3log2(n).log2(n)
                    actual->horario_Car = H;//------>3log2(n).log2(n)
               }
               siguiente = siguiente->sgte;
          }//-----------------------------log2(n).log2(n)----------------------------------------------------------------------
          actual = actual->sgte;//------>3log2(n)
          siguiente = actual->sgte;//------>3log2(n)

     }//-----------------------------log2(n)----------------------------------------------------------------------
}
```

F(n) = $44(\log(n))^2 + 12\log(n) + 9$

O (F(n)) = O ($44(\log(n))^2 + 12\log(n) + 9$) ➔ el O distribuir

O(F(n)) = O ($44(\log(n))^2$) + O ($12\log(n)$) + O (9) ➔ Constante

O(F(n)) = O ($(\log(n))^2$) + O ($\log(n)$)+O (1) ➔ O mayor

O(F(n)) = O ($(\log(n))^2$)

Jailene Milagros Garcia Candela

```
void ordenarListaTrabajador(NodoTrabajador *&listaTrabajador)
{
    NodoTrabajador *actual = new NodoTrabajador();//------>2
   actual = listaTrabajador;//------>3
   NodoTrabajador *siguiente = new NodoTrabajador();  //------>2
    int C;
    string NA;
    string usu;
    string contra;
    string cartra;
    while(actual->sgte != NULL)//------>log2(n)+2
    {
        siguiente = actual->sgte;//------>3log2(n)

        while(siguiente!=NULL)//------>(log2(n)+2).log2(n)
        { //--------------------------------log2(n).log2(n)----------------------------------------------------------------------
            if(actual->Codigo_trabajador > siguiente->Codigo_trabajador)
            {
                C = siguiente->Codigo_trabajador;//------>2log2(n).log2(n)
                siguiente->Codigo_trabajador = actual->Codigo_trabajador;//------>3log2(n).log2(n)
                actual->Codigo_trabajador = C; //------>3log2(n).log2(n)
                NA = siguiente->Nombresyapellidos_trabajador;//------>2log2(n).log2(n)
                siguiente->Nombresyapellidos_trabajador = actual->Nombresyapellidos_trabajador ;//------>3log2(n).log2(n)
                actual->Nombresyapellidos_trabajador  = NA;   //------>3log2(n).log2(n)
                usu = siguiente->usuario_Tb;//------>2log2(n).log2(n)
                siguiente->usuario_Tb = actual->usuario_Tb ;//------>3log2(n).log2(n)
                actual->usuario_Tb  = usu;//------>3log2(n).log2(n)
                contra = siguiente->contrasena_Tb;//------>2log2(n).log2(n)
                siguiente->contrasena_Tb = actual->contrasena_Tb ;//------>3log2(n).log2(n)
                actual->contrasena_Tb  = contra;//------>3log2(n).log2(n)
                cartra = siguiente->cargo_trabajador ;//------>2log2(n).log2(n)
                siguiente->cargo_trabajador = actual->cargo_trabajador ;//------>3log2(n).log2(n)
                actual->cargo_trabajador  = cartra;//------>3log2(n).log2(n)

            }
            siguiente = siguiente->sgte;//------>3log2(n).log2(n)
        }//--------------------------------log2(n).log2(n)----------------------------------------------------------------------
         actual = actual->sgte;//------>3log2(n)
        siguiente = actual->sgte;//------>3log2(n)
    }//--------------------------------log2(n)----------------------------------------------------------------------------------------
}
```

F(n) = $44(\log(n))^2 + 12\log(n) + 9$

O (F(n)) = O ($44(\log(n))^2 + 12\log(n) + 9$) ➔ el O distribuir

O(F(n)) = O ($44(\log(n))^2$) + O ($12\log(n)$) + O (9) ➔ Constante

O(F(n)) = O ($(\log(n))^2$) + O ($\log(n)$)+O (1) ➔ O mayor

O(F(n)) = O ($(\log(n))^2$)

Fecha: 10 / 05 / 2022