

```
#include "stdafx.h"
#include <stdio.h>
#include <windows.h>
```

PUNTEROS

Algoritmos/vgarro

```
int main(int argc, char* argv[])
```

```
{
```

```
//*****
```

```
int Numero;
```

```
int *PtrNumero;
```

```
Numero = 7;
```

```
PtrNumero = &Numero;
```

```
*PtrNumero = 33;
```

```
printf("Numero Resultante %d\n", Numero);
```

```
printf("Numero Resultante %d\n", *PtrNumero);
```

```
//*****
```

```
PtrNumero = new(int); // Se solicita al sistema operativo la memoria para la variable anonima
```

```
*PtrNumero = 99;
```

```
printf("Numero Resultante %d\n", *PtrNumero);
```

```
//*****
```

```
int **Ptr_Ptr_Numero;
```

```
Ptr_Ptr_Numero = &PtrNumero;
```

```
**Ptr_Ptr_Numero = 15;
```

```
printf("Numero Resultante %d\n", **Ptr_Ptr_Numero);
```

```
GlobalFree(PtrNumero); // Se libera la memoria reservada x el new para la variable anonima
```

```
//PtrNumero = null;
```

```
//printf("Numero Resultante %d\n", *PtrNumero);
```

```
//*****
```

```
typedef struct TPersona
```

```
{
```

```
int Cedula;
```

```
int Carnet;
```

```
float Salario;
```

```
TPersona *Siguiete;
```

```
}x;
```

```
TPersona Funcionario;
```

```
Funcionario.Cedula=1234567;
```

```
Funcionario.Carnet=20052;
```

```
Funcionario.Siguiete=new(TPersona);
```

```
Funcionario.Siguiete->Cedula=7654321;
```

```
Funcionario.Siguiete->Carnet=20002;
```

```
Funcionario.Siguiete->Siguiete=new(TPersona);
```

```
printf("Cedula1 %d\n", Funcionario.Cedula);
```

```
printf("Carnet1 %d\n", Funcionario.Carnet);
```

```
printf("Cedula2 %d\n", Funcionario.Siguiete->Cedula);
```

```
printf("Carnet2 %d\n", Funcionario.Siguiete->Carnet);
```

```
GlobalFree(Funcionario.Siguiete->Siguiete);
```

```
//Se libera la memoria reservada x el new para la variable anonima Funcionario.Siguiete
```

```
GlobalFree(Funcionario.Siguiete);
```

```
//Se libera la memoria reservada x el new para la variable anonima Funcionario.Siguiete
```

```
return 0;
```

```
}
```

Declaración de un puntero a Entero

Asignación de una dirección o referencia a un puntero

Asignación de contenido a una variable anonima a través

Declaración de un puntero "Siguiete" a una estructura con la forma de Tpersona

Asignación de una dirección o referencia al puntero "Siguiete"

```
#include "stdafx.h"
#include <stdio.h>
#include <windows.h>
```

LISTA ENLAZADA Algoritmos/vgarro

```
//*****
```

```
typedef struct TArticulo
```

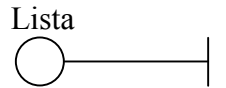
```
{
    int Codigo;
    char *Nombre;
    int Disponible;
    double Precio;
    TArticulo *Siguiete;
};
```

Declaración de un puntero "Siguiete" a una estructura con la forma de Tpersona

```
typedef TArticulo * PtrTArticulo;
```

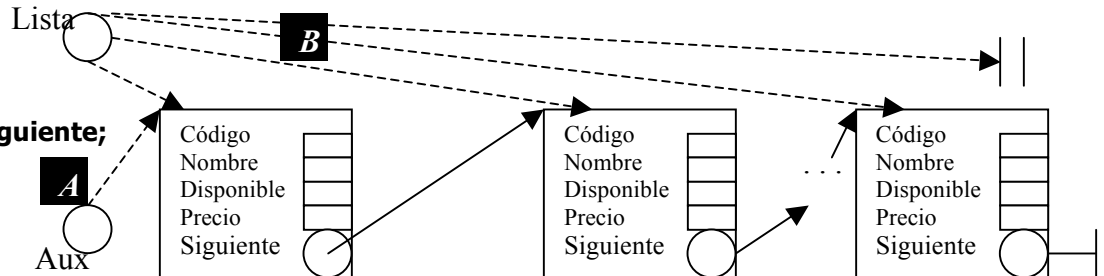
```
//*****
```

```
void InicializarInventario(PtrTArticulo &Lista)
{
    Lista = NULL;
}
```



```
void DestruirInventario(PtrTArticulo &Lista)
```

```
{
    PtrTArticulo Aux;
    Aux = Lista;
    while (Aux!=NULL)
    {
        Lista = Lista->Siguiete;
        GlobalFree(Aux);
        Aux = Lista;
    }
}
```

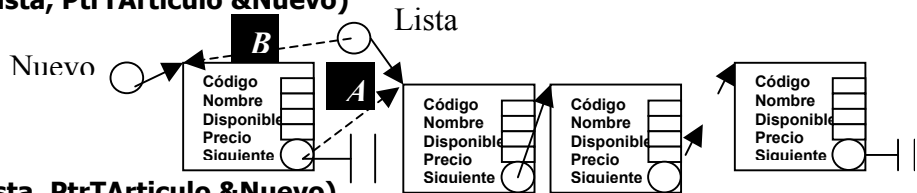


```
PtrTArticulo CrearArticulo(int NCodigo, int NDisponible, double NPrecio)
```

```
{
    PtrTArticulo Pieza = new(TArticulo);
    Pieza->Codigo = NCodigo;
    Pieza->Disponible = NDisponible;
    Pieza->Precio=NPrecio;
    Pieza->Nombre="Nada";
    Pieza->Siguiete=NULL;
    return Pieza;
}
```

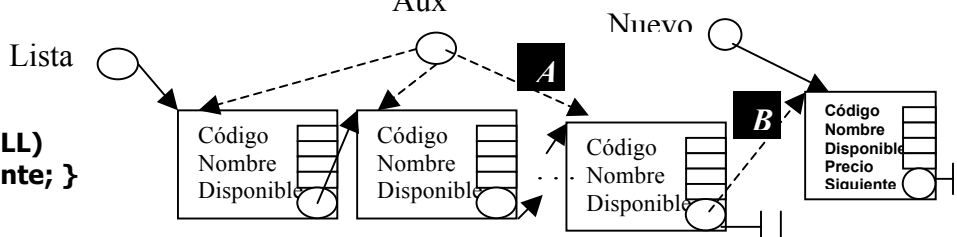
```
void AgregarInicioInventario(PtrTArticulo &Lista, PtrTArticulo &Nuevo)
```

```
{
    Nuevo->Siguiete=Lista;
    Lista = Nuevo;
}
```



```
void AgregarFinalInventario(PtrTArticulo &Lista, PtrTArticulo &Nuevo)
```

```
{
    PtrTArticulo Aux;
    Aux = Lista;
    if (Aux != NULL)
    {
        while (Aux->Siguiete!=NULL)
        {
            Aux = Aux->Siguiete;
        }
        Aux->Siguiete = Nuevo;
    }
}
```



```

else
{
    Lista = Nuevo;
}

}

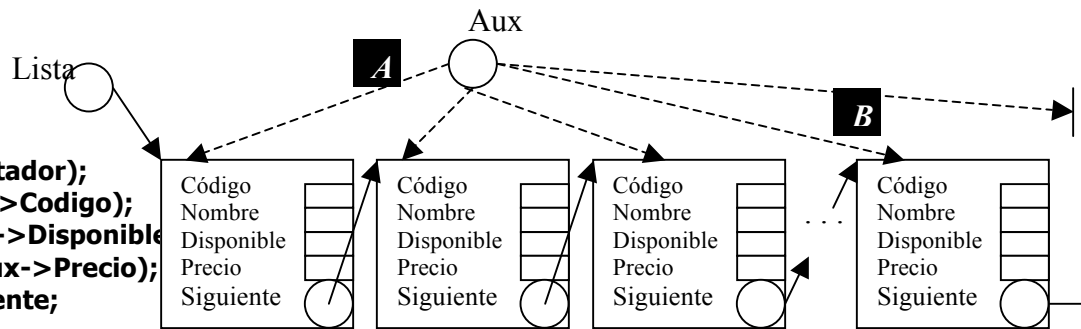
void ListarInventario(PtrTArticulo &Lista)
{

```

```

    int Contador = 1;
    PtrTArticulo Aux;
    A Aux = Lista;
    while (Aux!=NULL) B
    {
        printf(" %d ", Contador);
        printf("%d ", Aux->Codigo);
        printf(" %d ", Aux->Disponibilidad);
        printf(" %f\n ", Aux->Precio);
        Aux = Aux->Siguiente;
        Contador ++;
    }
}

```



```

}

void main(int argc, char* argv[])
{
    //*****
    PtrTArticulo Llantas;
    PtrTArticulo Abarrotes;
    PtrTArticulo Nuevo;
    //*****
    InicializarInventario(Abarrotes);
    for (int i = 100; i<=200; i++)
    {
        Nuevo = CrearArticulo(i,i,i);
        AgregarInicioInventario(Abarrotes,Nuevo);
    }
    ListarInventario(Abarrotes);
    DestruirInventario(Abarrotes);
    //*****
    InicializarInventario(Abarrotes);
    for (i = 200; i<=300; i++)
    {
        Nuevo = CrearArticulo(i,i,i);
        AgregarFinalInventario(Abarrotes,Nuevo);
    }
    ListarInventario(Abarrotes);
    DestruirInventario(Abarrotes);
    //*****
    InicializarInventario(Llantas);
    for (i = 100; i<=200; i++)
    {
        Nuevo = CrearArticulo(i,i,i);
        AgregarInicioInventario(Llantas,Nuevo);
    }
    ListarInventario(Llantas);
    DestruirInventario(Llantas);
    //*****
    InicializarInventario(Llantas);
    for (i = 200; i<=300; i++)
    {
        Nuevo = CrearArticulo(i,i,i);
        AgregarFinalInventario(Llantas,Nuevo);
    }
    ListarInventario(Llantas);
    DestruirInventario(Llantas);
}

```