



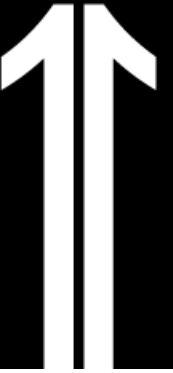
Estructura de Datos

Mg. Cinthia J. Calderon Aquino

Semana 02

UPN.EDU.PE

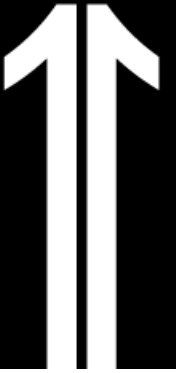
SEMANA 02



**Listas enlazadas Simples. Operaciones:
ordenamiento, búsqueda y mezcla**

PRESENTACIÓN DE LA SESIÓN

Logro de la Sesión y Temario



Logro:

Al término de la sesión, el estudiante aprende algoritmos de listas enlazadas, buscar, mezclar y ordenar, usándolos con eficacia.

Temario:

1. Listas enlazadas Simples.
2. Operaciones: ordenamiento, búsqueda y mezcla

BUSCAR UN ELEMENTO EN UNA LISTA



Para buscar un elemento en una lista, sólo hay que seguir 4 pasos:

1. Crear un nuevo nodo(actual).
2. Igualar ese nuevo nodo(actual) a la lista.
3. Recorrer la lista.
4. Determinar si el elemento existe o no en la lista.

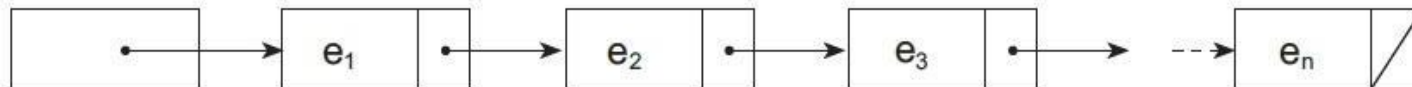
DECLARACIÓN DE UN NODO

Una lista enlazada se compone de una serie de nodos enlazados mediante referencias,

```
class Nodo
{
    int dato;
    Nodo enlace;
    public Nodo(int t)
    {
        dato = t;
        enlace = null;
    }
}
```

```
class Elemento
{
    //...;
}

class Nodo
{
    Elemento dato;
    Nodo enlace;
}
```



e_1, e_2, \dots, e_n son valores del tipo TipoElemento



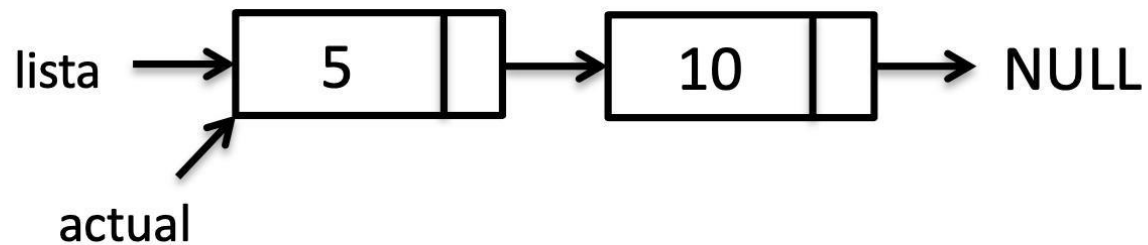
1. CREAR UN NUEVO NODO(ACTUAL).



```
void buscarLista(Nodo *lista, int n){  
    Nodo *actual = new Nodo();  
}
```



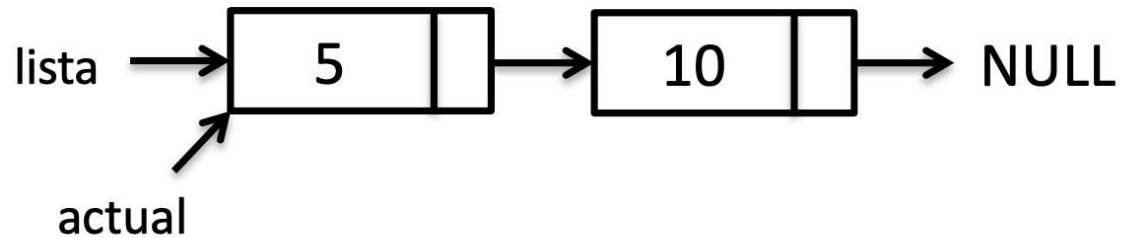
2. IGUALAR ESE NUEVO NODO(ACTUAL) A LA LISTA.



```
void buscarLista(Nodo *lista, int n){  
    Nodo *actual = new Nodo();  
    actual = lista;  
}
```



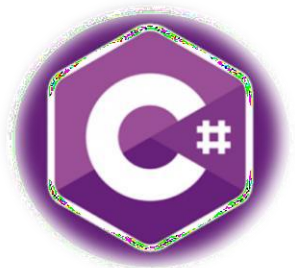
3. RECORRER LA LISTA.



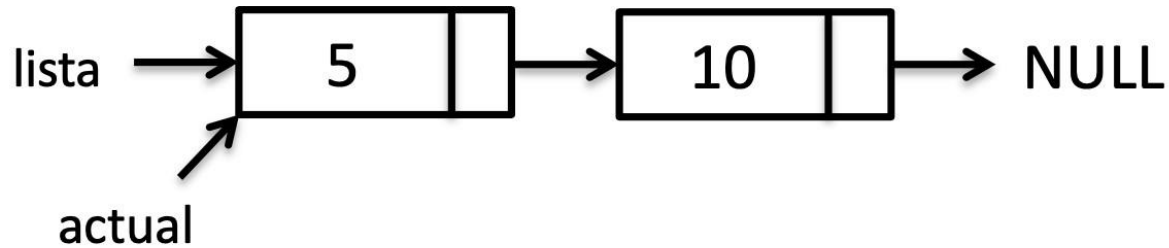
`n = 10;`

```
while((actual != NULL) && (actual->dato <= n)){
```

```
}
```



4. DETERMINAR SI EL ELEMENTO EXISTE O NO EN LA LISTA.

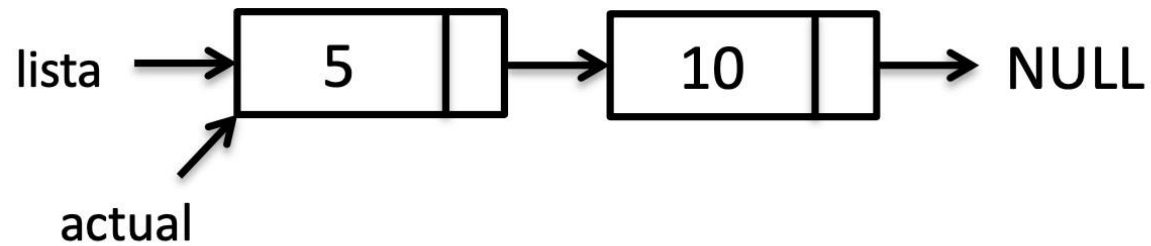


$n = 10;$

```
while((actual != NULL) && (actual->dato <= n)){  
    if(actual->dato == n){  
        band = true;  
    }  
    actual = actual->siguiente;  
}
```



4. DETERMINAR SI EL ELEMENTO EXISTE O NO EN LA LISTA.

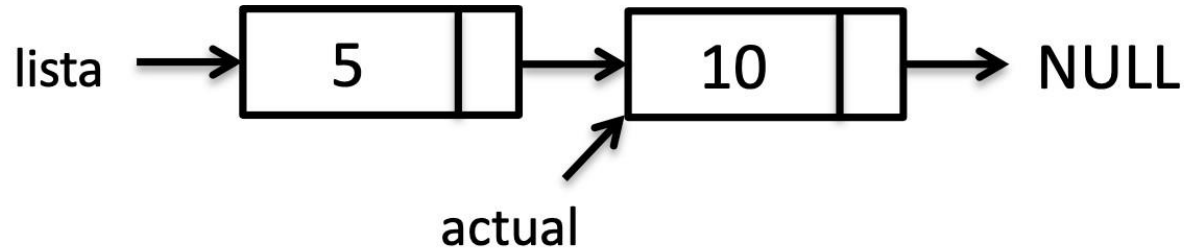


$n = 12;$

```
while((actual != NULL) && (actual->dato <= n)){  
    if(actual->dato == n){  
        band = true;  
    }  
    actual = actual->siguiente;  
}
```



4. DETERMINAR SI EL ELEMENTO EXISTE O NO EN LA LISTA.



$n = 12;$

```
while((actual != NULL) && (actual->dato <= n)){  
    if(actual->dato == n){  
        band = true;  
    }  
    actual = actual->siguiente;  
}
```



POR LO TANTO EL CÓDIGO NOS QUEDARÍA ASÍ:

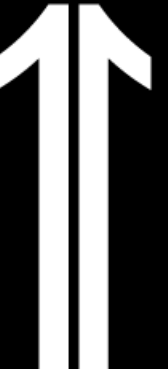


```
void buscarLista(Nodo *lista,int n){
    bool band = false;

    Nodo *actual = new Nodo();
    actual = lista;

    while((actual != NULL) && (actual->dato <= n)){
        if(actual->dato == n){
            band = true;
        }
        actual = actual->siguiente;
    }

    if(band == true){
        cout<<"Elemento "<<n<<" SI a sido encontrado en lista\n";
    }
    else{
        cout<<"Elemento "<<n<<" NO a sido encontrado en lista\n";
    }
}
```



PROBLEMAS

PROBLEMAS

Problema 1



1. En la actualidad el trafico es muy estresante por lo cual varios pasajeros deciden en tomar un tren con diferentes destino y código de vagón. Se ingresa los datos de los pasajeros por cada boleto de viaje(nombre, sexo). Mostrar todo la lista de pasajero de espera. Generar una búsqueda por cada pasajero y una baja de pasajero.

- a) La búsqueda será por cada destino del tren, código de vagón y N° de Boleto.
- b) La baja de pasajero(Destino de tren para bajar, código de vagón y N° de Boleto.



PROBLEMAS

Problema 2

Resolver con LISTA ENLAZADAS SIMPLES en el lenguaje de programación C# con formulario mostrando el resultado en dataGridView con CLASE Nodo, ListaEnlazada y Alumno:

En la UPN se entregará los carnets de medio pasaje a los alumnos que cumplieron con sus requisitos. Por lo cual se necesita sus datos (Cod_Carnet, Nombre y edad).

- a) La búsqueda Avanzada es por el cod_carnet y mostrar el mensaje flotante
- b) Eliminar por cod_carnet de cada Alumno.
- d) Ordenar en forma ascendente por cod_carnet

The screenshot shows a Windows Form titled "Form1" with a standard Windows XP-style title bar. The form contains the following elements:

- Three text boxes for data entry: "CodCarnet", "Nombre", and "Edad".
- A button labeled "ORDENAMIENTO" (Sorting).
- A button labeled "BUSQUEDA" (Search) next to a small text box.
- A button labeled "ELIMINAR" (Delete) next to another small text box.
- A button labeled "AGREGAR" (Add).
- A large, empty rectangular area at the bottom, likely intended for a DataGridView to display the data.

PROBLEMAS

Problema 3

2. Se realizara un evento por el aniversario la empresa starbucks. De lo cual requieren sus datos de los clientes.(N° de Contacto, Nombre, Teléfono, Correo y Fecha de Nacimiento). Se necesita los siguientes datos.

- Insertar al inicio los datos del cliente.
- Generar una búsqueda por N° de contacto.
- Eliminar por primero, ultimo de cada cliente.
- Mostrar la lista de cliente.

The screenshot shows a Windows application window titled "Form1". It contains several input fields and a list of buttons. On the left, there are five labeled input fields: "No. De Contacto", "Nombre:", "Telefono:", "Correo:", and "Fecha de Nacimiento:". Below these is a large empty rectangular box. On the right, there is a vertical stack of buttons: "Agregar En Inicio", "Agregar", "Buscar", "Eliminar", "Insertar", "Listar", "Eliminar Primero", "Eliminar Ultimo", and "Listar Inverso". To the right of the "Eliminar" button is a label "Posición:" followed by a small input field.



¿¿Preguntas o comentarios?

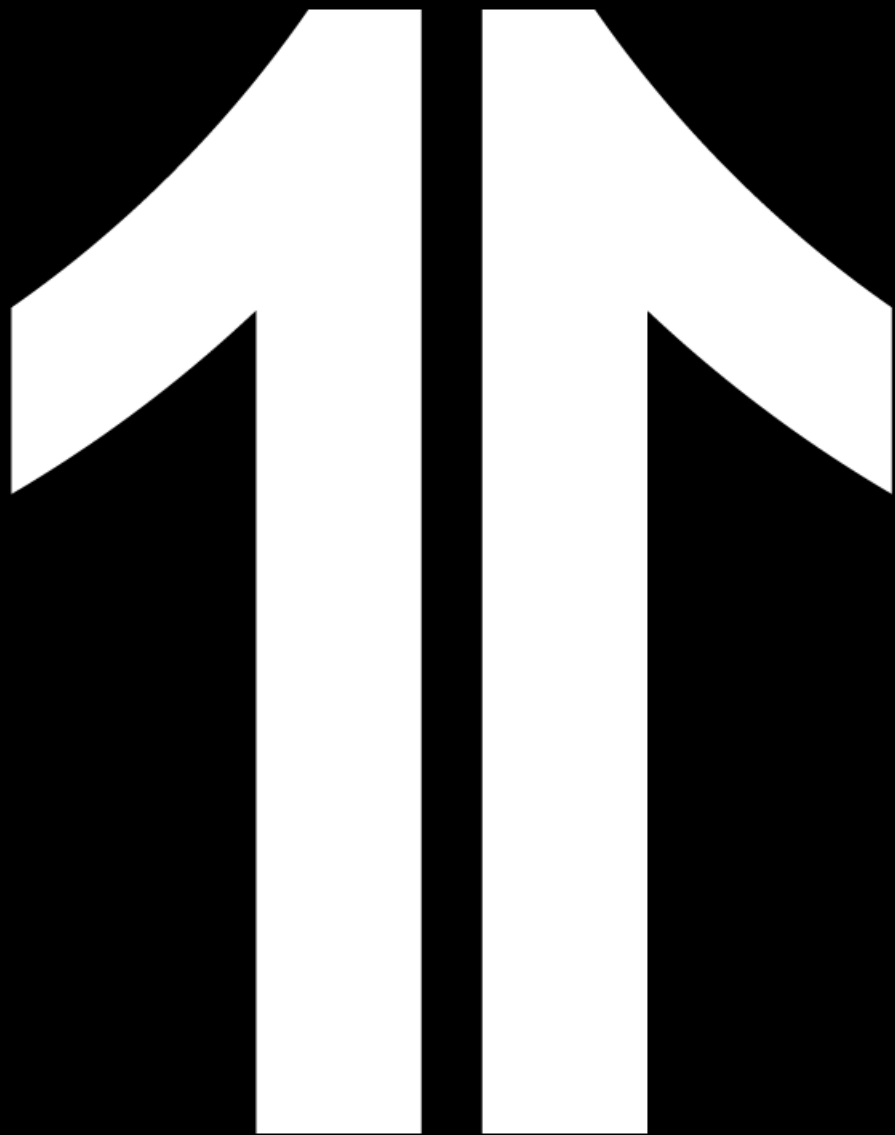
¿Preguntas o comentarios?



BIBLIOGRAFIA REFERENCIAL



- Ceballos Sierra, F. Microsoft C#: Curso de Programación (2a.ed.) 2014
<https://elibronet.eu1.proxy.openathens.net/es/lc/upnorte/titulos/106417>
- Cesar Liza Avila; Estructura de datos con C/C++



GRACIAS

