

Estructura de Datos

[Fecha]

UPN.EDU.PE

Actividades



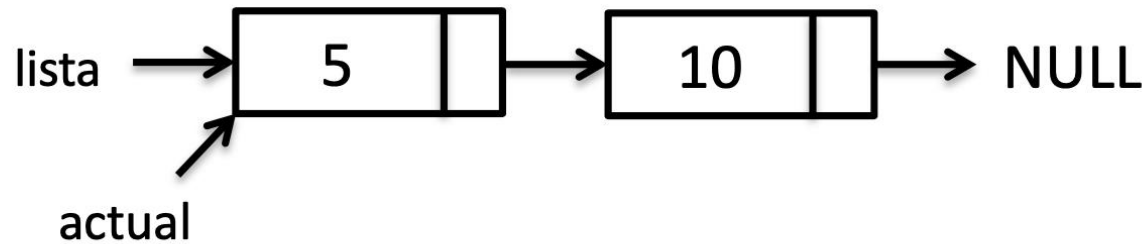
Actividades semana 01

1. CREAR UN NUEVO NODO(ACTUAL).



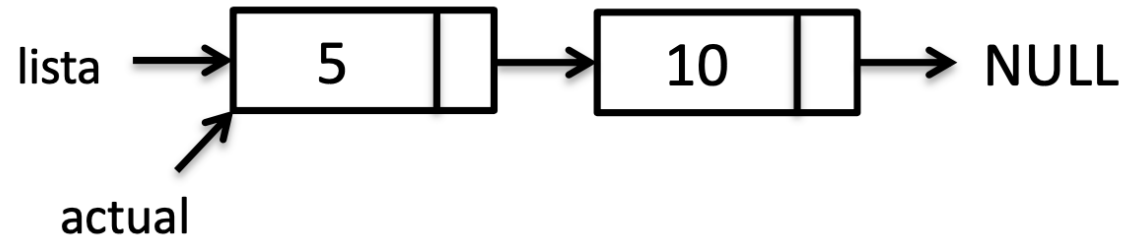
```
void buscarLista(Nodo *lista, int n){  
    Nodo *actual = new Nodo();  
}
```

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.

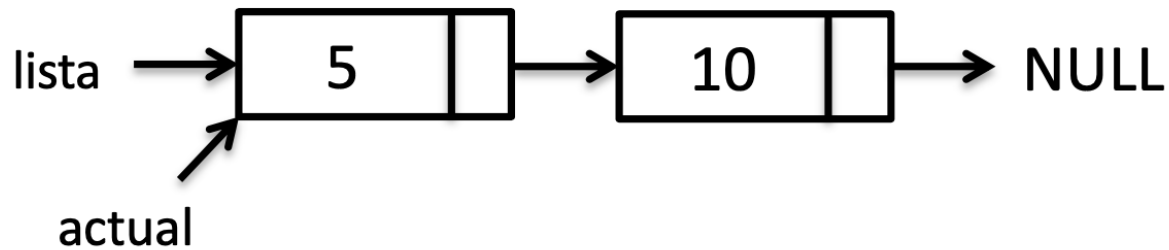


`n = 10;`

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

```
}
```

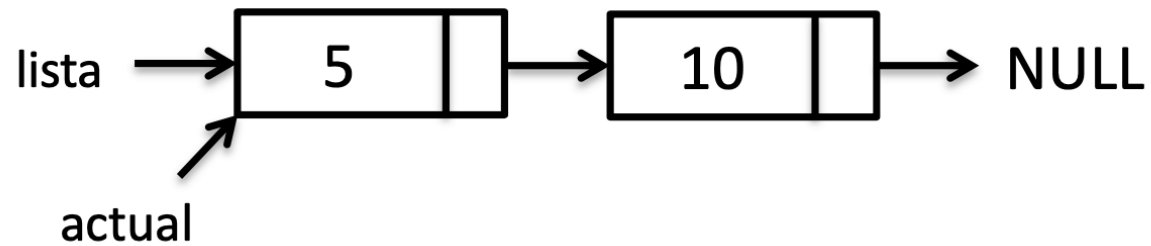
4. DETERMINAR SI EL ELEMENTO EXISTE O NO EN LA LISTA.



$n = 10;$

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

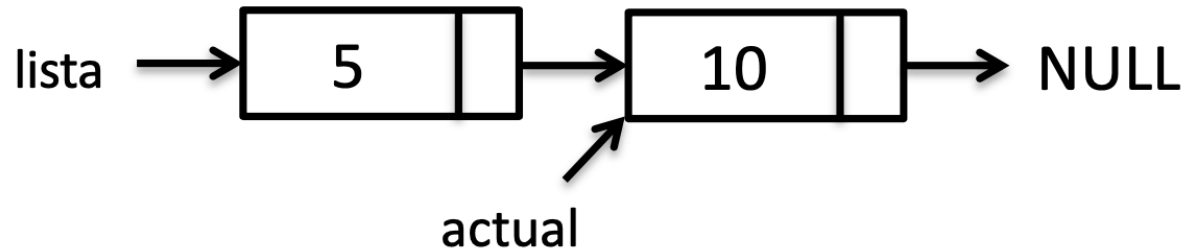
4. DETERMINAR SI EL ELEMENTO EXISTE O NO EN LA LISTA.



$n = 12;$

```
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;  
}
```


ACTIVIDAD



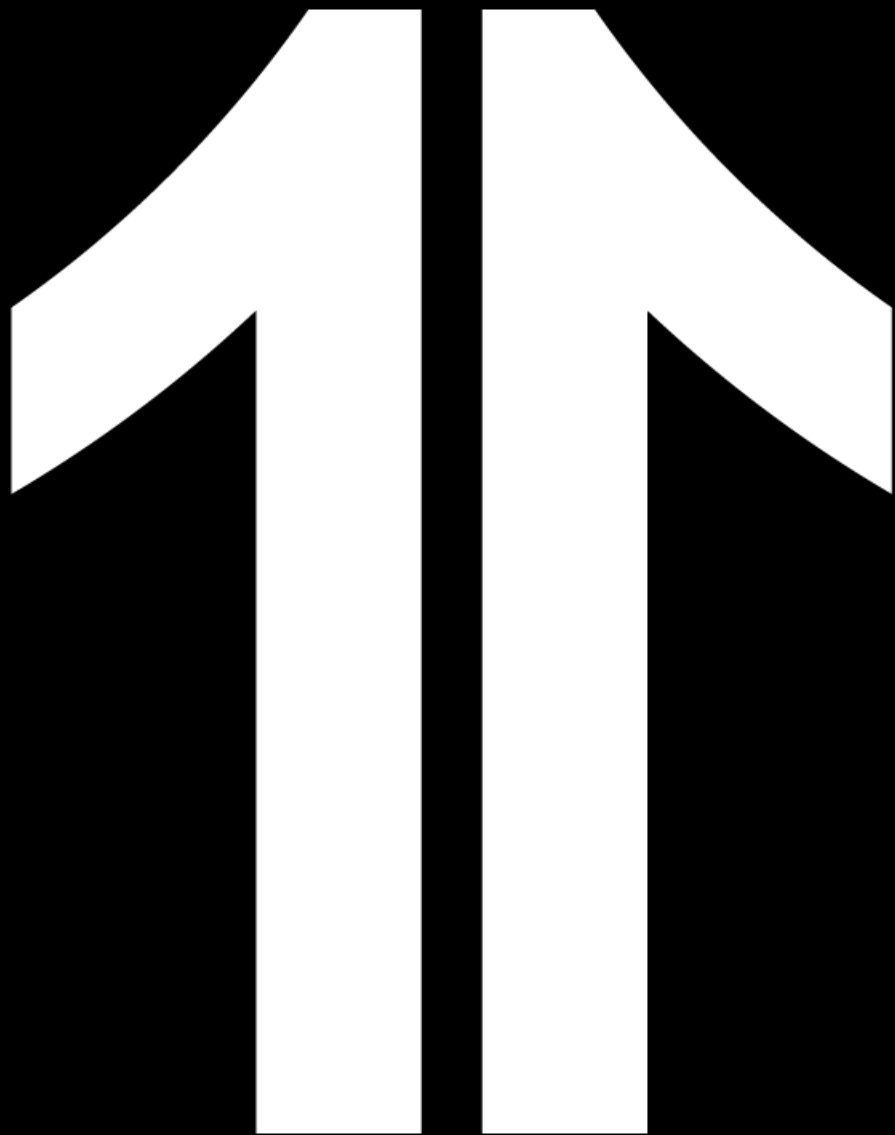
- Desarrollar y analizar el siguiente código:

```
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";
    }
}
```



GRACIAS

