

Listas

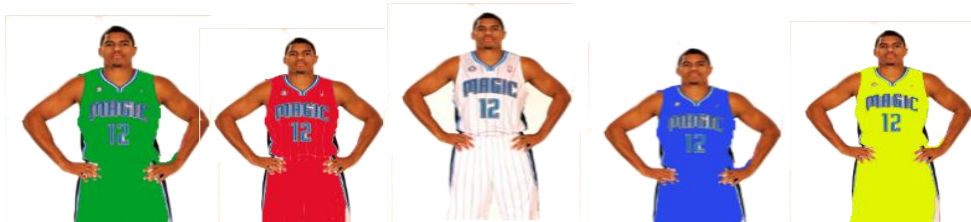
Explicación P7 - Parte 3

Listas - Insertar ordenado

Ejercicio 2

Modifique la solución del ejercicio 1 para que la información de los jugadores quede ordenada **por altura** de manera ascendente.


¿Cómo debería realizarse la carga?



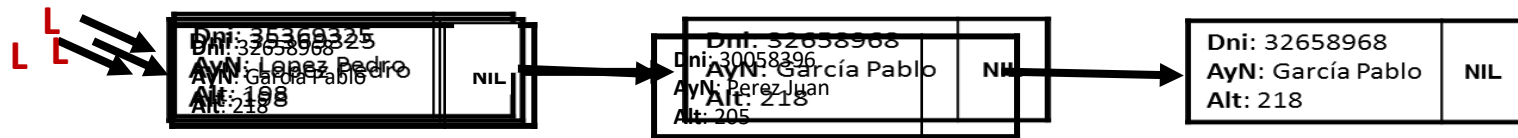
Listas - Insertar ordenado

La forma de realizar los enlaces varia de acuerdo al lugar donde corresponde insertar el nodo

Ejercicio 2

			
Dni: 32658968 ApyNom: García Pablo Altura: 218	Dni: 35369325 ApyNom: Lopez Pedro Altura: 198	Dni: 30058396 ApyNom: Perez Juan Altura: 205	Dni: 0 ApyNom: Altura:

L = NIL

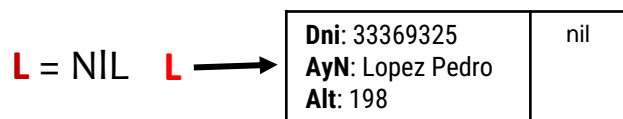


Notar que quedaron almacenados ordenados por altura de forma ascendente

Listas - Insertar ordenado

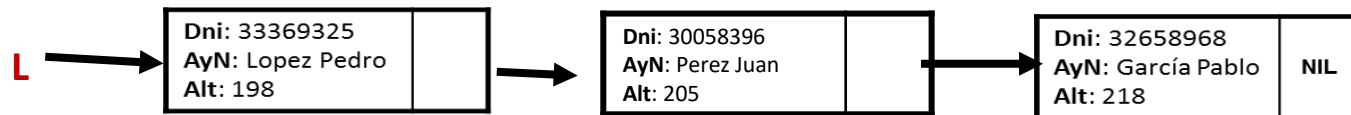
Casos:

1- Insertar nodo en una lista vacía



Se asigna el nodo como primer elemento de la lista

2- Insertar en lista no vacía



Pueden darse 3 casos: insertar adelante - insertar en el medio - insertar atrás

Al inicio

Dni: 34769325 AyN: Fernández Juan Alt: 185	nil
---	-----

Al medio

Dni: 29769325 AyN: Martínez Carlos Alt: 210	nil
--	-----

Al final

Dni: 32769325 AyN: González Mario Alt: 219	nil
---	-----

Listas - Insertar ordenado

Pasos a seguir:

1.Crear el nodo a insertar

2.Buscar la posición correspondiente para insertar el nodo creado

- *Es necesario ubicarse al inicio de la lista y recorrer hasta encontrar la posición.*
- *Vamos a utilizar 2 punteros auxiliares para realizar el recorrido. ¿por qué?*

3.Realizar los enlaces

- *Una vez encontrada la posición, deberán actualizarse los enlaces de acuerdo al caso. Es importante entonces determinar dicho caso*

Listas - Insertar ordenado

```
procedure cargarLista(var L:lista);  
var  
  j: jugador;  
Begin  
  leerJugador(j);  
  while(j.dni <> 0) do begin  
    insertarOrdenado(L, j);  
    leerJugador(j);  
  end;  
end;
```

1. Crear el nodo a
insertar

2. Buscar la posición para
insertar el nodo creado

3. Realizar los enlaces

```
procedure insertarOrdenado(var L:lista; j:jugador);  
var  
  nue: lista;  
  act, ant: lista; {punteros auxiliares para recorrido}  
begin  
  new (nue);  
  nue^.dato := j;  
  act := L; {ubico act y ant al inicio de la lista}  
  ant := L;  
  while( act <> nil)and(j.altura > act^.dato.altura)do  
  begin  
    ant := act;  
    act:= act^.sig;  
  end;  
  if (act = ant) then {al inicio o lista vacía}  
    L:= nue;  
  else {al medio o al final}  
    ant^.sig:= nue;  
    nue^.sig:= act;  
  end;
```