

# Listas

Explicación P7 - Parte 1

# Listas - Agregar adelante

## Ejercicio 1

Escriba un programa que lea y almacene información de jugadores de básquet. De cada jugador se lee: dni, apellido y nombre, y altura en cm. La lectura finaliza cuando se lee el jugador con dni 0, el cual no debe procesarse.

¿Cómo se representa la información de un jugador?

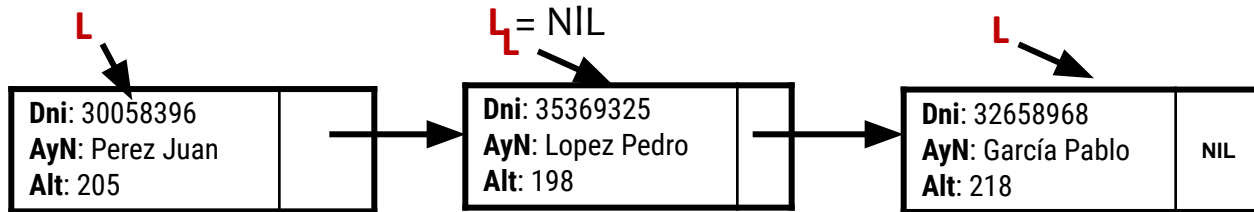
¿En qué estructura de datos se puede almacenar la información de todos los jugadores?



# Listas - Agregar adelante

## Ejercicio 1

 <b>Dni:</b> 32658968 <b>ApyNom:</b> García Pablo <b>Altura:</b> 218	 <b>Dni:</b> 35369325 <b>ApyNom:</b> Lopez Pedro <b>Altura:</b> 198	 <b>Dni:</b> 30058396 <b>ApyNom:</b> Perez Juan <b>Altura:</b> 205	<b>Dni:</b> 0 <b>ApyNom:</b> <b>Altura:</b>
--	---	--	---



*Notar que quedaron almacenados en el orden inverso al leído*

# Listas - Agregar adelante

```
program ejercicio;
type
  jugador = record
    dni: integer;
    nomyAp: string[30];
    altura: integer;
  end;

  lista = ^nodo;

  nodo = record
    dato: jugador;
    sig : lista;
  end;
```

```
var {PROGRAMA PRINCIPAL}
  L: lista;
begin
  L:= nil;
  cargarLista(L);
end.
```

```
procedure agregarAdelante(var L:lista; j:jugador);
var
  nue: lista;
begin
  new (nue);           {Creo un nodo}
  nue^.dato := j;      {Cargo el dato}
  nue^.sig := L;        {Realizo el enlace}
  L:= nue;              {Actualizo el primero}
end;

procedure cargarLista(var L:lista);
var
  j: jugador;
Begin
  leerJugador(j); {Lee un registro de jugador}
  while(j.dni <> 0) do begin
    agregarAdelante(L, j);
    leerJugador(j);
  end;
end;
```

# Listas - Recorrido

## Ejercicio 2

A partir de la lista generada en el Ejercicio 1, informar la cantidad de jugadores con dni par.

