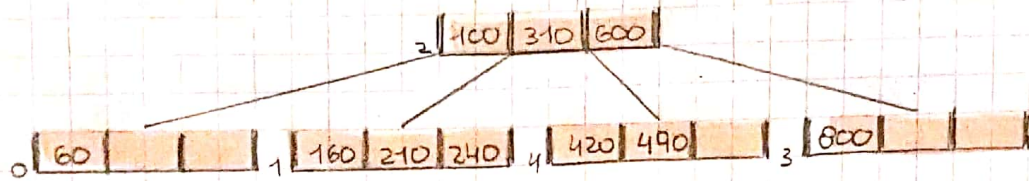
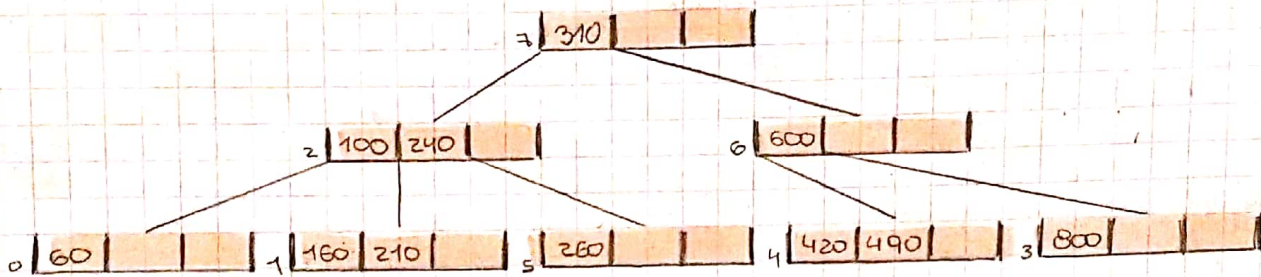


② ARBOLES

Arbol B - Orden 4 - política derecha

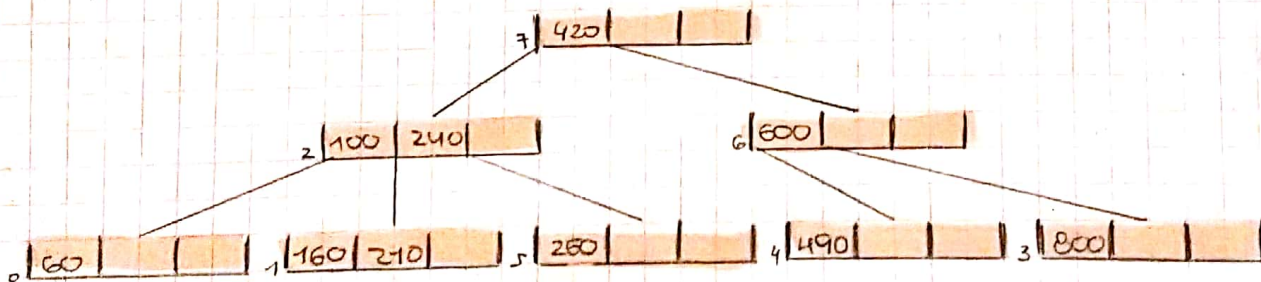


+260 Overflow en nodo 1 160 210 240 260 promoción de clave 240 se genera nodo 5. se propaga el overflow al nodo 2. 100 240 310 600 Promoción de clave 310 y nuevo nodo se genera nodo 6 y 7 y se incrementa el nivel del árbol



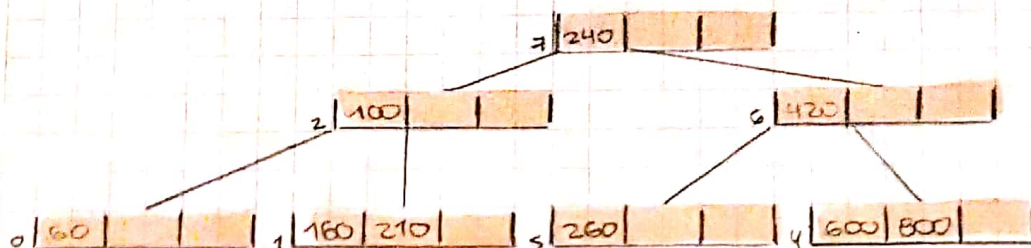
+260 L/E = L2 L1 E1 E5 E2 E6 E7

-310 Como la clave a eliminar no esta en un nodo hoja, cambiar con la mayor clave ubicada en subarbol derecho (310 por 420) y luego eliminar. No tengo deudas porque la cantidad de claves restantes en el nodo hoja no es menor al minimo necesario.



-310 L/E = L7 L6 L4 E4 E7

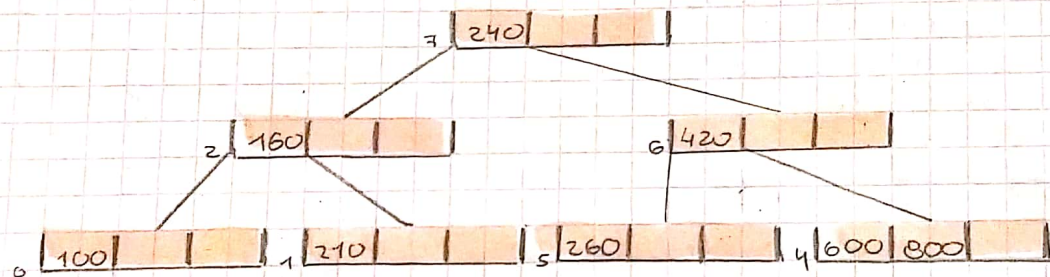
-490 Underflow en nodo 4, intento redistribuir con hermano derecho pero no es posible, por lo tanto fusiono con nodo 3 y libero nodo 3 luego se propaga overflow a nodo 6, como no tiene hermano derecho redistribuyo con el nodo hermano izquierdo [2].



-490 L/E = L7 L6 L4 L3 E4 L6 L2 E2 E6 E7

-60

Underflow en modo 0. Redistribución con hermano derecho
100 160 210 Promoción de clave 160



-60 L/E = L7 L2 L0 L1 E0 E1 E2


```
program examen;
```

```
type
```

```
empR = record
```

```
  dni : integer;
```

```
  nombre : string[15];
```

```
  apell : string[15];
```

```
  edad : integer;
```

```
  domic : string[20];
```

```
  nac : string[8];
```

```
end;
```

```
empleado = file of empR;
```

```
procedure cargar_empleados (var a : empleado);
```

```
var
```

```
  e, cab : empR;
```

```
begin
```

```
  leerEmp(e);
```

```
  {leo los datos de emp por teclado y guardo en reg}
```

```
  if (not existe(e.dni, a)) then begin
```

```
    reset(a);
```

```
    read(a, cab);
```

```
    if (cab.dni = 0) then begin { no tengo espacio libre }
```

```
      seek(a, filesize(a));
```

```
      write(a, e);
```

```
    end
```

```
  else begin
```

```
    seek(a, Abs(cab.dni));
```

```
    read(a, cab);
```

```
    seek(a, filePos(a)-1);
```

```
    write(a, e);
```

```
    seek(a, 0);
```

```
    write(a, cab);
```

```
  end;
```

```
  close(a);
```

```
end else
```

```
  writeln('El empleado ya existe');
```

```
end;
```

```
procedure quitar_empleados (var a : empleado);
```

```
var
```

```
  dni : integer;
```

```
  e, cab : empR;
```

```
begin
```

```
  writeln('Ingrese dni a eliminar');
```

```
  read(dni);
```

```
  if (not existe(dni, a)) then
```

```
    writeln('El empleado no existe');
```

```
  else begin
```

```
    reset(a)
```

```
    read(a, cab);
```

```
    read(a, e);
```

```
    while (e.dni <= dni)
```

```
      read(a, e);
```

```
    e.dni := cab.dni;
```

```
    seek(a, filePos(a)-1);
```

```
    cab.dni := filePos(a)*-1;
```

```
    write(a, e);
```

sigu. acci

```
    seek(a, 0);
```

```
    write(a, cab);
```

```
    close(a);
```

```
  end;
```

```
end;
```