

Enunciado

Dado el arreglo `MuchasLetras` de `MaxLetras` caracteres que está totalmente completo con caracteres alfabéticos sin blancos y un número pedido por teclado en `CantLetras`, eliminar todas las letras que se encuentran en `MuchasLetras` exactamente `CantLetras` veces y dejar todos los lugares que ocupaban al principio del arreglo con blancos sin modificar el orden de las restantes letras.

Enunciado

Dado el arreglo **MuchasLetras** de **MaxLetras** caracteres que está totalmente completo con caracteres alfabéticos sin blancos y un número pedido por teclado en **CantLetras**, eliminar todas las letras que se encuentran en **MuchasLetras** exactamente **CantLetras** veces y dejar todos los lugares que ocupaban al principio del arreglo con blancos sin modificar el orden de las restantes letras.

Enunciado

Dado el arreglo MuchasLetras de MaxLetras caracteres que está totalmente completo con caracteres alfabéticos sin blancos y un número pedido por teclado en CantLetras, eliminar todas las letras que se encuentran en MuchasLetras exactamente CantLetras veces y dejar todos los lugares que ocupaban al principio del arreglo con blancos sin modificar el orden de las restantes letras.

Ejemplo: (CantLetras = 3)

a b c d e f g h i j a c e a f f x r w x m x o r

- - - - - b c d e g h i j c e x r w x m x o r

Enunciado

Dado el arreglo MuchasLetras de MaxLetras caracteres que está totalmente completo con caracteres alfabéticos sin blancos y un número pedido por teclado en CantLetras, eliminar todas las letras que se encuentran en MuchasLetras exactamente CantLetras veces y dejar todos los lugares que ocupaban al principio del arreglo con blancos sin modificar el orden de las restantes letras.

Ejemplo: (CantLetras = 3)

a b c d e **f** g h i j **a** c e **a** **f** **f** x r w x m x o r

- - - - - b c d e g h i j c e x r w x m x o r

Si la cantidad es 1 ?

Si la cantidad es 1000 ?

Si no hay ninguna letra con esa cantidad?

Componentes

Dado el arreglo **MuchasLetras** de **MaxLetras** caracteres que está totalmente completo con caracteres alfabéticos sin blancos y un número pedido por teclado en **CantLetras**, eliminar todas las letras que se encuentran en **MuchasLetras** exactamente **CantLetras** veces y dejar todos los lugares que ocupaban al principio del arreglo con blancos sin modificar el orden de las restantes letras.

Ejemplo: (CantLetras = 3)

a b c d e f g h i j a c e a f f x r w x m x o r

- - - - - b c d e g h i j c e x r w x m x o r

Variables: CantLetras, MuchasLetras

Constantes?

Tipos?

Constantes, Variables, Tipos¿?

MuchasLetras

| | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|--|

1

MaxLetras

CantLetras

| |
|--|
| |
|--|

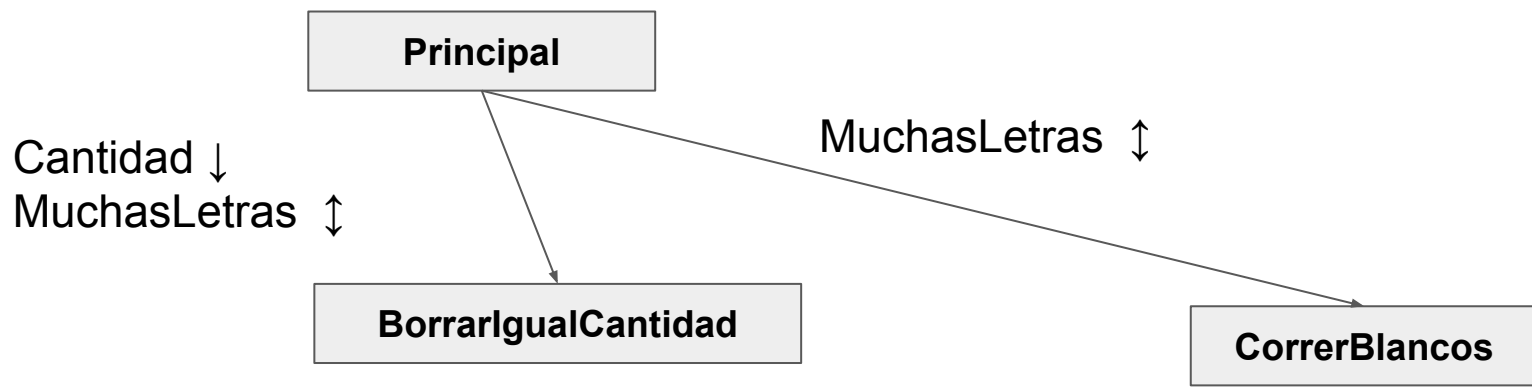
Componentes

```
program .....;  
const MaxLetras = 20;  
type TipoArregloLetras = array [1..MaxLetras] of Char;  
... (Funciones y Procedimientos)  
var CantLetras      : integer;  
    MuchasLetras : TipoArregloLetras;  
begin  
    ... (Cuerpo del programa principal)  
end.
```

Una estrategia / división posible.....

Primero borro todas las letras que cumplen la condición, borrar es convertirlas a blanco ' '.

Luego hago el corrimiento al inicio de todos los blancos.



programa principal

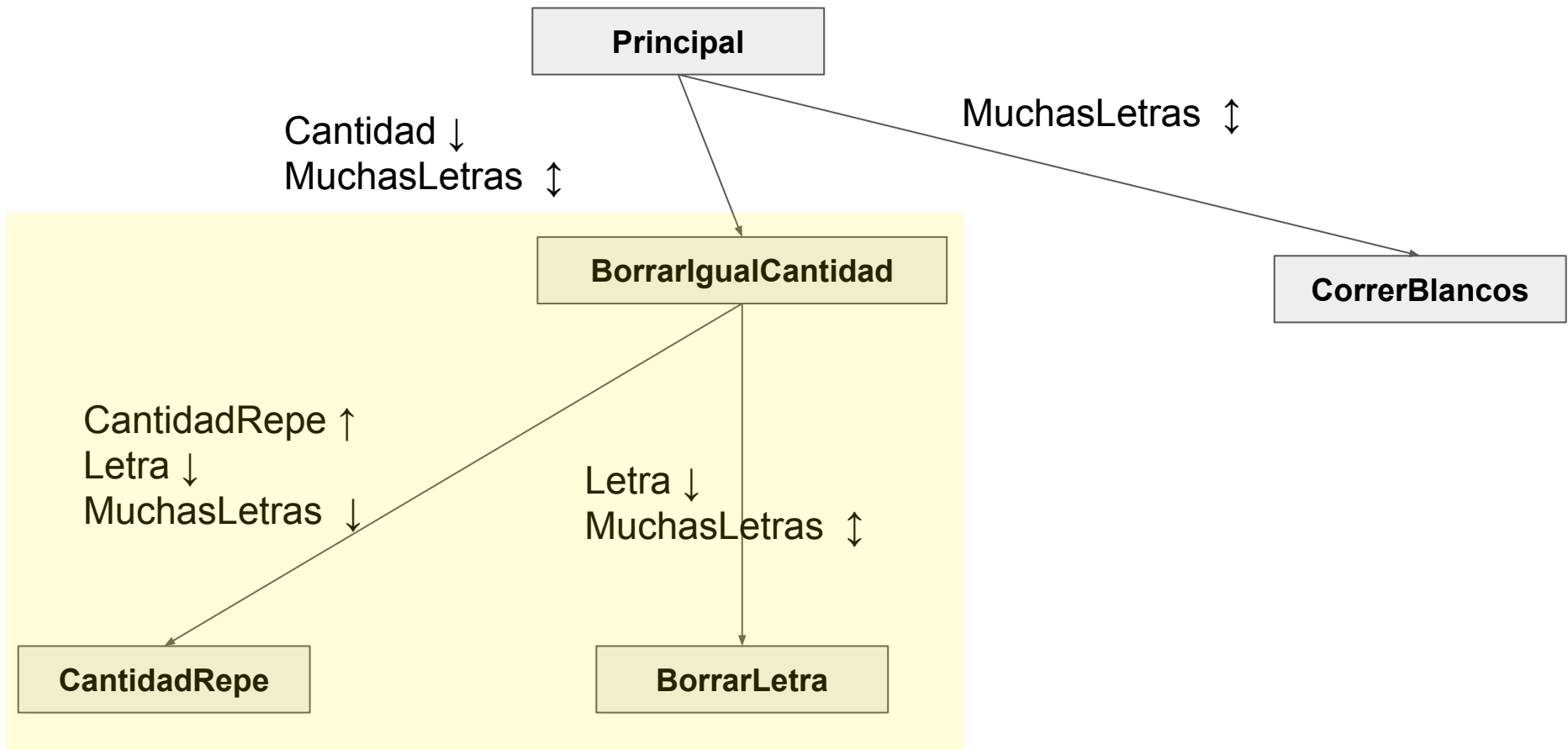
```
program eliminaletras;  
const MaxLetras = 3;  
type TipoArregloLetras = array [1..MaxLetras] of Char;  
(PROCEDIMIENTOS Y FUNCIONES)  
var CantLetras    : integer;  
    MuchasLetras  : TipoArregloLetras;  
    i              : integer;  
begin  
    MuchasLetras[1] := 'a'; MuchasLetras[2] := 'b'; MuchasLetras[3] := 'a';  
    readln(CantLetras);  
    BorrarIgualCantidad(MuchasLetras,CantLetras);  
    CorrerBlancos(MuchasLetras);  
    for i := 1 to MaxLetras do begin  
        writeln(MuchasLetras[i]);  
    end;  
    writeln(CantLetras);  
end.
```

BorrarIgualCantidad: Una estrategia / división posible.....

Hago una función a la que le paso una letra y el arreglo y me devuelve cuántas veces está esa letra en el arreglo.

Hago un procedimiento que le paso el arreglo y una letra y reemplaza esa letra por blanco en todos los lugares en la que la encuentra.

Con los dos módulos anteriores recorro el arreglo del principio al fin y consulto la cantidad de veces que está la letra de esa posición, si está la cantidad deseada invoco al procedimiento.



```
function CantidadRepe(ML:TipoArregloLetras;L:char):integer;  
var i, resultado : integer;  
begin  
    resultado := 0;  
    for i :=1 to MaxLetras do begin  
        if ML[i] = L then begin  
            resultado := resultado + 1;  
        end;  
    end;  
    CantidadRepe := resultado;  
end;
```

BorrarLetra

```
procedure BorrarLetra(var ML:TipoArregloLetras;L:char);  
var i : integer;  
begin  
  for i := 1 to MaxLetras do begin  
    if ML[i] = L then begin  
      ML[i] := ' ';  
    end;  
  end;  
end;
```

BorrarIgualCantidad

```
procedure BorrarIgualCantidad(var ML:TipoArregloLetras;CL:integer);  
(FUNCIONES Y PROCEDIMIENTOS INTERNOS)
```

```
var i    : integer;
```

```
begin
```

```
  for i := 1 to MaxLetras do begin
```

```
    if ML[i] <> ' ' then begin
```

```
      if CL = CantidadRepe(ML, ML[i]) then begin
```

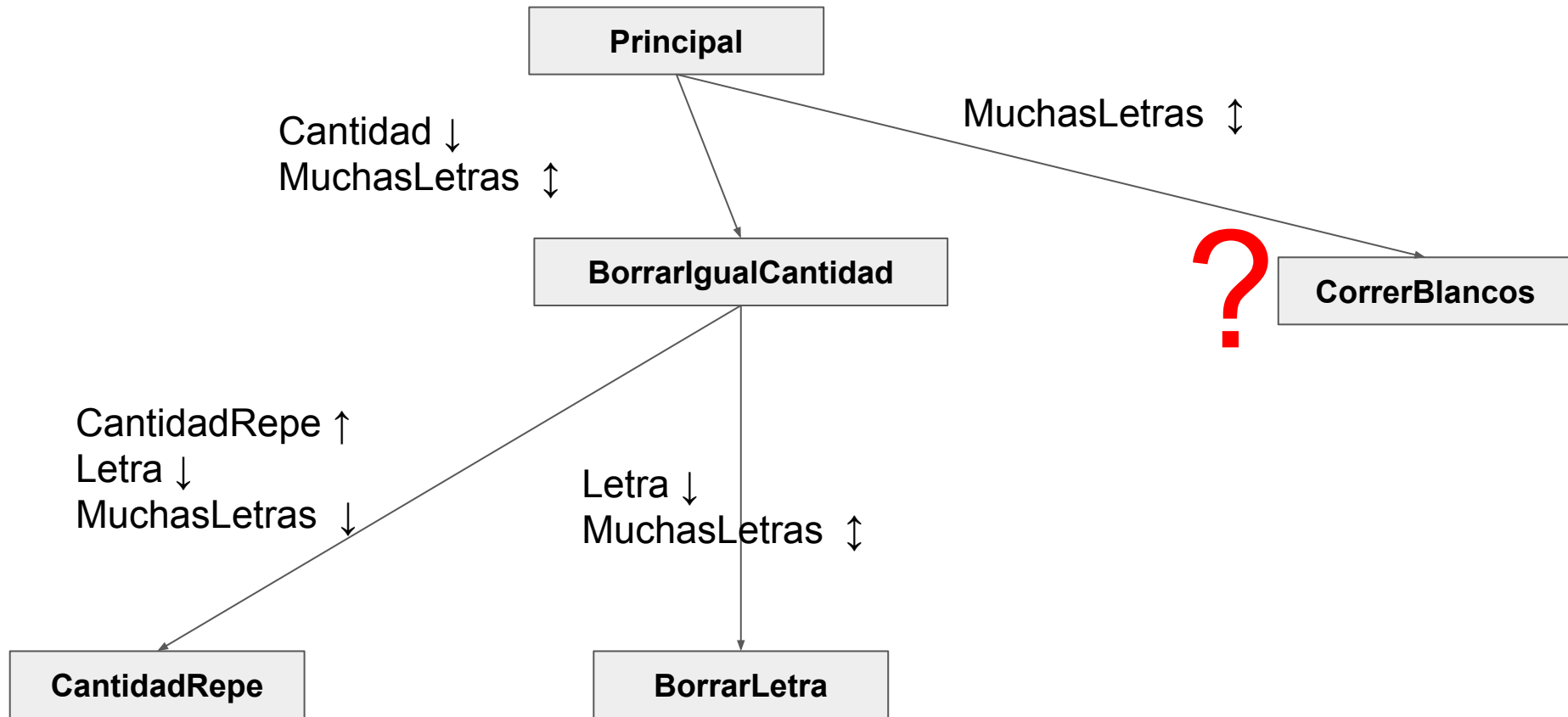
```
        BorrarLetra(ML,ML[i]);
```

```
      end;
```

```
    end;
```

```
  end;
```

```
end;
```

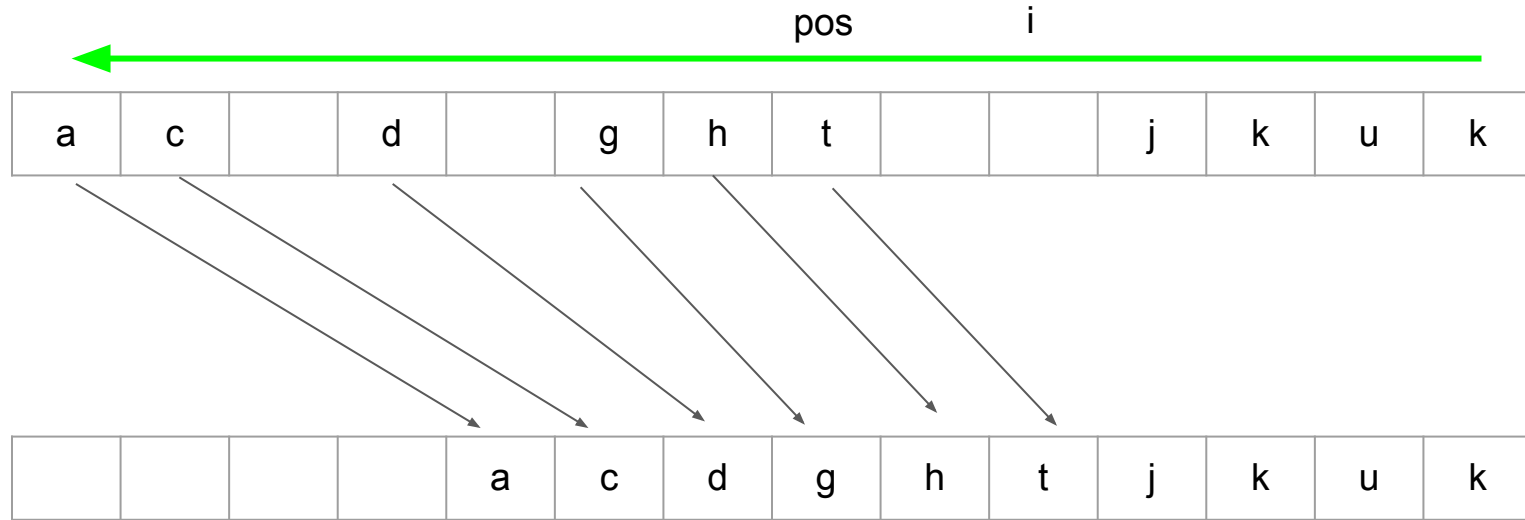


CorrerBlancos: Una estrategia / división posible.....

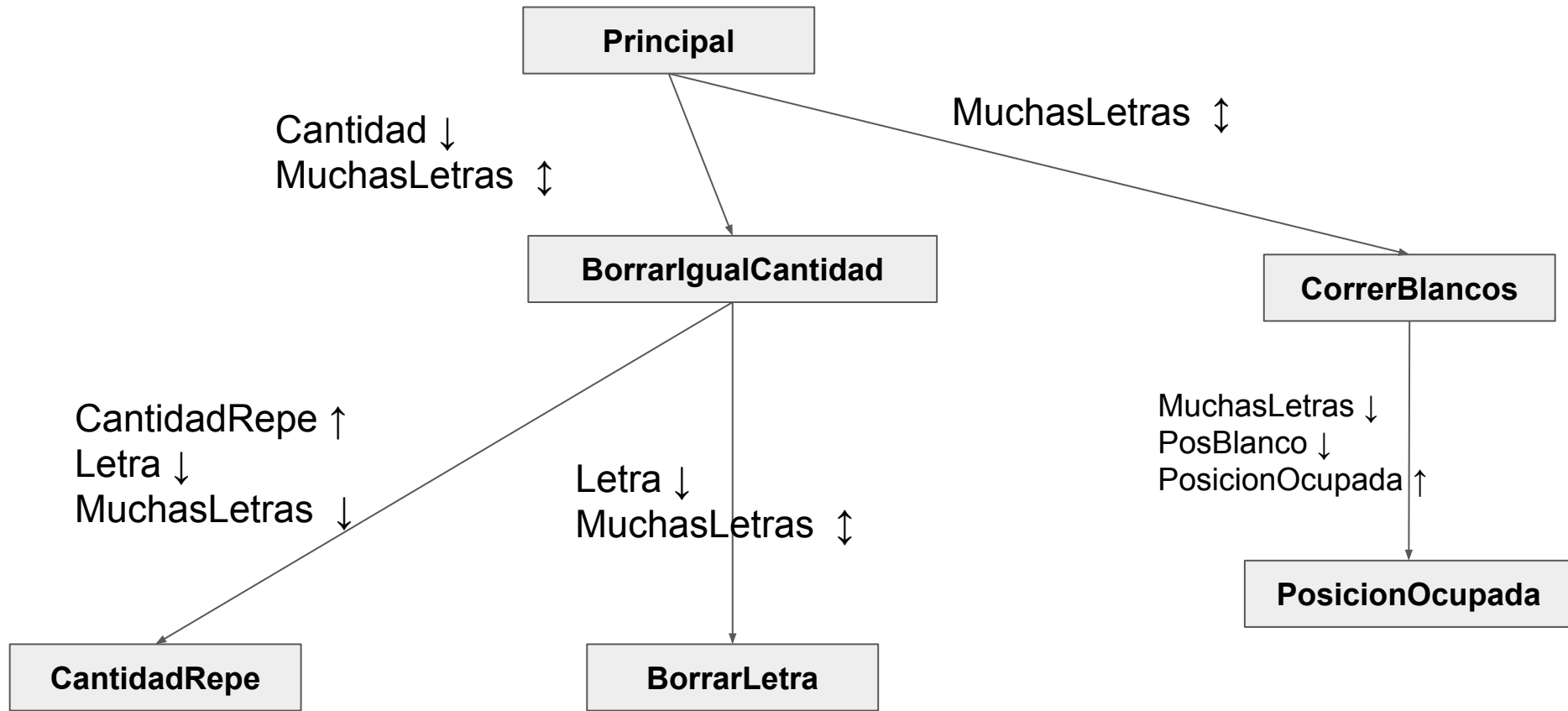
Recorro el arreglo de atrás para adelante.

Si encuentro un blanco busco cuál es la posición anterior a esta que tiene una letra y las intercambio.

Correr Blancos



PosicionOcupada(ML,i):integer;



CorrerBlancos

```
procedure CorrerBlancos(var ML:TipoArregloLetras);  
var i,posicion : integer;  
begin  
  for i := MaxLetras downto 1 do begin  
    if ML[i] = ' ' then begin /  
      posicion := PosicionOcupada(ML,i);  
      ML[i] := ML[posicion];  
      ML[posicion] := ' ';  
    end;  
  end;  
end;
```

PosicionLetraAnterior

```
function PosicionOcupada(ML,posicion):integer;  
var i : integer;  
begin  
    while (posicion >= 1 and ML[posicion] = ' ') do begin  
        posicion := posicion -1;  
    end;  
    PosicionOcupada := posicion;  
end;
```

Temas a ver

- La solución planteada está incompleta.
 - Tiene falta de documentación / comentarios y los nombres de datos y/o módulos son mejorables.
- Es sólo una alternativa, pueden existir soluciones mucho mejores o diferentes.
- Se les ocurre alguna mejora? Por ejemplo: cuando se está recorriendo el arreglo para buscar cantidades y eliminar las repetidas, tiene sentido recorrer cada vez desde el principio del arreglo para ambos módulos? Podrían mejorar eso?