

Parámetros

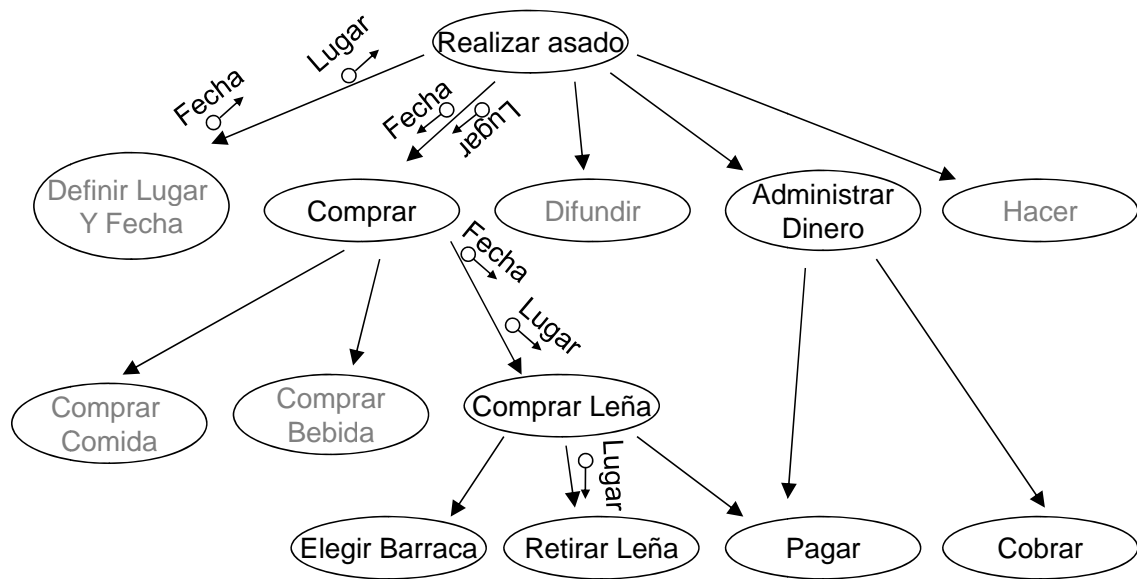


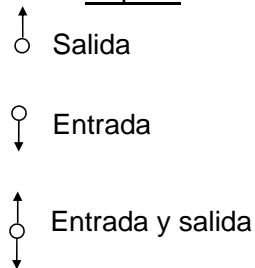
Diagrama de Estructura con cuplas

Es una herramienta que permite desagregar un problema en tareas (**módulos**) y dejar documentada la forma en que se pensó su solución.

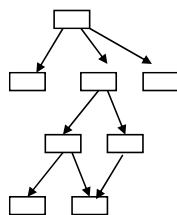
Componentes:



Cuplas:



Ejemplo:





```

program PasaDatosOrdenados;
{Dada la pila Origen que puede poseer números
 en forma desordenada, los pasa a la pila
 Destino quedando todos ordenados de igual
 forma a la que lo estaban en Origen. }
uses estructu;
var Origen, Destino, Aux : Pila;
begin
    InicPila(Destino,"");
    InicPila(Aux,"");
    ReadPila(Origen);
    while Not PilaVacía(Origen) do
        Apilar(Aux, Desapilar(Origen));
    while Not PilaVacía(Aux) do
        Apilar(Destino, Desapilar(Aux));
    WritePila(Origen);
    WritePila(Destino)
end.

```

```

program PasaDatosOrdenados;
{Dada la pila Origen que puede poseer números
 en forma desordenada, los pasa a la pila
 Destino quedando todos ordenados de igual
 forma a la que lo estaban en Origen. }
uses estructu;
var Origen, Destino, Aux : Pila;
procedure PasaOrigenAux;
begin
    while Not PilaVacía(Origen) do
        Apilar(Aux, Desapilar(Origen));
end;
procedure PasaAuxDestino;
begin
    while Not PilaVacía(Aux) do
        Apilar(Destino, Desapilar(Aux));
end;
begin
    InicPila(Destino,"");
    InicPila(Aux,"");
    ReadPila(Origen);
    PasaOrigenAux;
    PasaAuxDestino;
    WritePila(Origen);
    WritePila(Destino)
end.

```

```

program PasaDatosOrdenados;
{Dada la pila Origen que puede poseer números
 en forma desordenada, los pasa a la pila
 Destino quedando todos ordenados de igual
 forma a la que lo estaban en Origen. }
uses estructu;
var Origen, Destino, Aux : Pila;
procedure PasaPila(var Inicial:Pila; var Final:
 Pila);
begin
    while Not PilaVacía(Inicial) do
        Apilar(Final, Desapilar(Inicial));
end;
begin
    InicPila(Destino,"");
    InicPila(Aux,"");
    ReadPila(Origen);
    PasaPila(Origen, Aux);
    PasaPila(Aux, Destino);
    WritePila(Origen);
    WritePila(Destino)
end.

```

```

program PasaDatosOrdenados;
{Dada la pila Origen que puede poseer números
 en forma desordenada, los pasa a la pila
 Destino quedando todos ordenados de igual
 forma a la que lo estaban en Origen. }
uses estructu;
var Origen, Destino, Aux : Pila;
procedure PasaOrigenAux;
begin
    while Not PilaVacía(Origen) do
        Apilar(Aux, Desapilar(Origen));
end;
procedure PasaAuxDestino;
begin
    while Not PilaVacía(Aux) do
        Apilar(Destino, Desapilar(Aux));
end;
begin
    InicPila(Destino,"");
    InicPila(Aux,"");
    ReadPila(Origen);
    PasaOrigenAux;
    PasaAuxDestino;
    WritePila(Origen);
    WritePila(Destino)
end.

```


Tipo de Pasaje de Parámetro

VAR o no var, esa es la cuestión

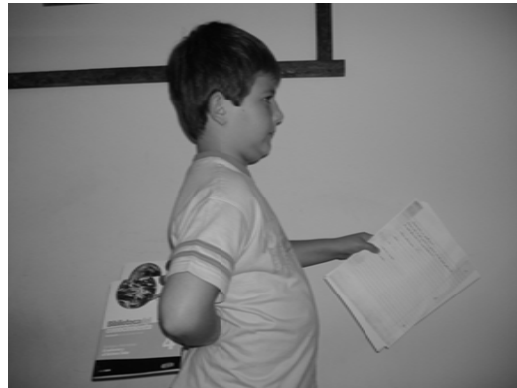
VAR

(Pasaje por Referencia)
(Se pasa el objeto real)
(Vuelve modificado)



~~VAR~~

(Pasaje por copia o valor)
(Se pasa una copia)
(No afectan las modificaciones)



Tipo de Pasaje de Parámetro

VAR

(Pasaje por Referencia)
(Se pasa el objeto real)
(Vuelve modificado)

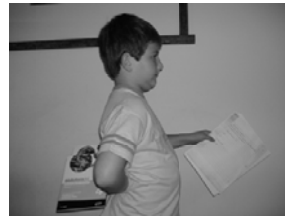


Salida

Entrada y salida

~~VAR~~

(Pasaje por copia o valor)
(Se pasa una copia)
(No afectan las modificaciones)



Entrada


```

program Ejemplazo;
{Ejemplo de un programa complejo de entender. }
uses estructu;

var
    Origen : Pila;
    Destino : Fila;

procedure Procedimiento(var Destino:Pila;
Origen:Fila);
begin
    WriteFila(Origen);
    WritePila(Destino);
    InicFila(Origen,"");
    InicPila(Destino,"")
end;

begin
    InicPila(Origen,'1');
    InicFila(Destino,'2');
    Procedimiento(Origen, Destino);
    WritePila(Origen);
    WriteFila(Destino)
end.

```

Resultados posibles:

1) Error de compilación.

2) Error de ejecución.

3)

1

2

1

Vacíá

4)

2

1

Vacíá

2