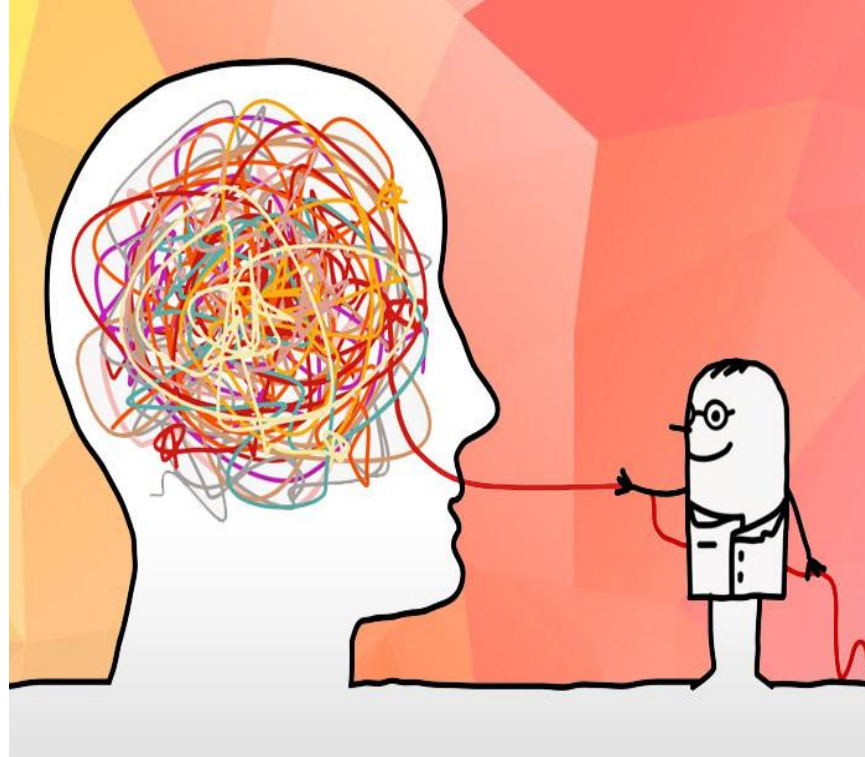


# PROGRAMACIÓN I

TEORÍA – CECILIA SANZ

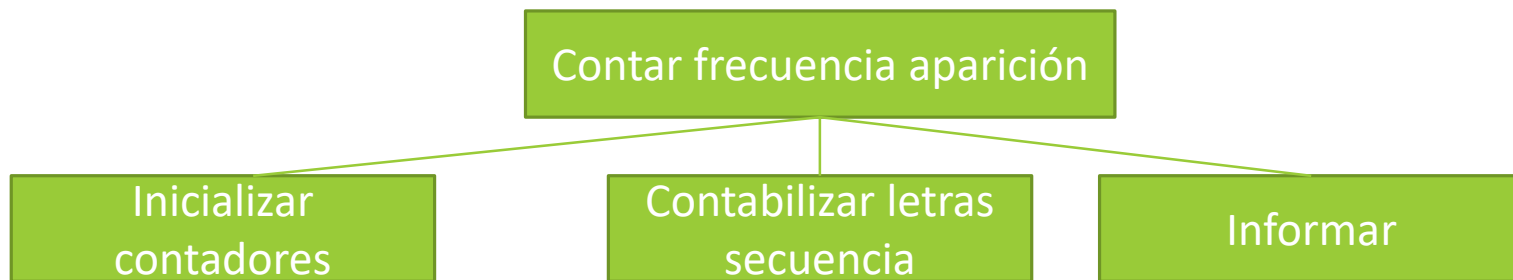
# Ejemplos de ejercicios



# Estructura de datos EJERCICIOS

---

1. Se lee una secuencia de letras minúsculas que termina en punto. Informar la cantidad de veces que aparece cada letra.



# Estructura de datos-ARREGLO – EJERCICIO 1

---

**Program uno;**

**CONST** FIN='.'; inicio='a'; termina='z';

**type** rango= inicio..termina;

cantidades = **array** [rango] **of** integer;

{módulos del programa}

**Var** num: cantidades;

Begin

    inicializar(num);

    contabilizar(num);

    informar(num);

End.

# Estructura de datos-ARREGLO – EJERCICIO 1

---

**Procedure inicializar (var a: cantidades);**

**Var**

**i:rango;**

**Begin**

**for i:= inicio to termina do**

**a[i]:= 0;**

**End;**

# Estructura de datos-ARREGLO – EJERCICIO 1

**Procedure contabilizar (var a: cantidades);**

**Procedure validar (var valor: char);**

**Begin**

---

**while ((valor < inicio) or (valor > termina)) and (valor <> FIN) do**

**readln(valor);**

**End;**

**Var valor: char;**

**Begin**

**readln(valor);**

**validar(valor);**

**while (valor <> FIN) do**

**begin**

**a[valor]:= a[valor] + 1;**

**readln(valor);**

**validar(valor);**

**end;**

**End;**

# Estructura de datos-ARREGLO – EJERCICIO 1

---

**Procedure informar (a: cantidades);**

**Var**

**i:rango;**

**Begin**

**for i:= inicia to termina do**

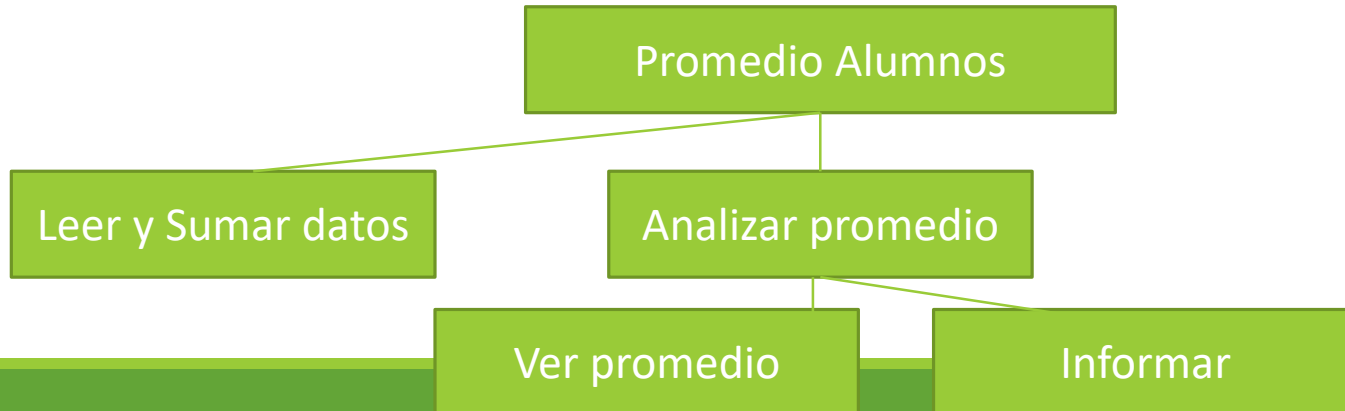
**write (a[i]);**

**End;**

# Estructura de datos-ARREGLO – EJERCICIO 2

---

2. Se lee una sucesión de datos de a lo sumo 100 alumnos que se inscribieron a una materia. De cada alumno se conoce nombre y nota. Informar los nombres de aquellos alumnos que superen el promedio del grupo. La lectura termina con el alumno de nombre ZZZ.





# Estructura de datos-ARREGLO – EJERCICIO 2

---

**Program Ej2;**

**Const** maxdatos = 100; FIN='zzz';

**type** rango=0..maxdatos; rango2= 1..maxdatos;

str20 = string [20];

alumnos = **record**

nombre : str20;

nota : real;

**end;**

Inscriptos = record

**datos:array** [rango2] **of** alumnos;

DimL: rango;

**end;**

**{módulos}**

**var**

alus: Inscriptos;

---

suma, promedio : Real;

**begin**    {Principal}

Leo\_y\_Sumo\_Datos (alus, suma);

if (alus.dimL >0) then begin

          promedio := suma / alus.dimL;

          informar (alus, promedio);

        end

    else writeln ('No se cargaron alumnos');

**End.**

# Estructura de datos-ARREGLO – EJERCICIO 2

---

```
procedure Leo_y_Sumo_Datos (var alus: inscriptos;var sum:Real);  
Var unAlu: alumnos;  
begin  
  sum := 0; alus.dimL:= 0;  
  leerAlumno (unAlu);  
  while (unAlu.nombre <> FIN) and (alus.dimL < maxdatos)do  
    begin  
      alus.dimL:= alus.dimL +1;  
      alus.datos[alus.dimL]:= unAlu;  
      sum := sum + alus. datos[alus.dimL].nota;  
      leerAlumno (unAlu);  
    end  
  end;  
end;
```

# Estructura de datos-ARREGLO – EJERCICIO 2

---

```
procedure Informar (alus: inscriptos;prom:Real);  
Var i:rango2;  
Begin  
  for i := 1 to alus.dimL do  
    if (alus.datos[i].nota > prom) then  
      WriteLn (alus.datos[i].nombre, ' ', alus.datos[i].nota )  
  
end;
```

Ver [código](#) en  
Pascal

# Estructura de datos-ARREGLO – EJERCICIO 3

---

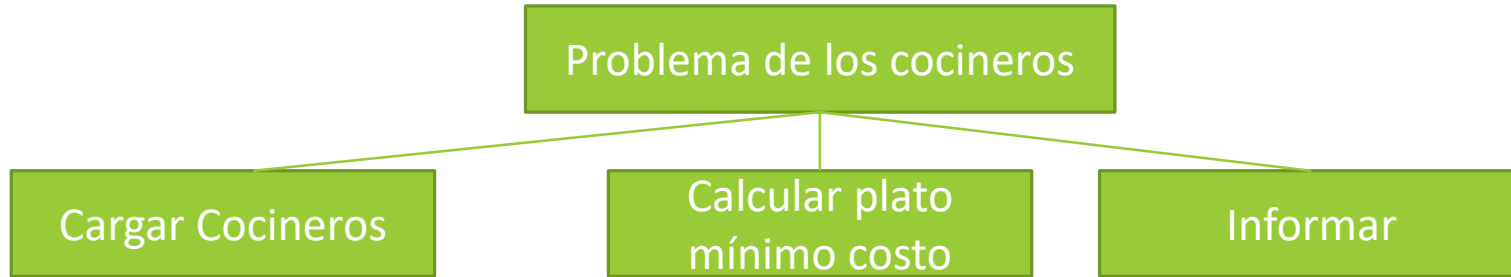
3. Se lee una sucesión de datos de a lo sumo 50 cocineros. De cada cocinero se conoce nombre y código del plato qué es su especialidad. **Simular el proceso de carga de cocineros.**

Además, por cada código de plato (son 100) se cuenta con el costo promedio que tiene la realización de dicho plato.

Informar el nombre del cocinero cuyo plato especial sea el menos costoso.

# Estructura de datos-ARREGLO – EJERCICIO 3

---



# Estructura de datos-ARREGLO – EJERCICIO 3

**Program Ej3;**

---

**Const** maxcoc = 50; FIN='zzz'; maxplatos=100;

**type** rangococ=0..maxcoc; rangopla=1..maxplatos;

str50 = string [50];

cocinero = **record**

nombre : str50;

codiplato: rangopla;

**end;**

Cocineros = **record**

datos:**array** [1.. maxcoc] **of** cocinero;

DimL: rangococ;

**end;**

Platos = **array** [rangopla] **of** **real**;

---

**var**

coc: cocineros; p: platos

min :cocinero;

**begin**    {Principal}

Carga\_Cocineros(coc);

MinimoCosto(coc,p, min);

Writeln('El cocinero con mínimo costo en el plato es: ',min.nombre, 'el plato es:',  
min.codiplato);

**end;**



# Estructura de datos-ARREGLO – EJERCICIO 3

---

```
procedure Carga_Cocineros (var coc: cocineros);  
Var c: cocinero;  
begin  
  coc.dimL:= 0;  
  leerCocinero (c);  
  while (c.nombre <> FIN) and (coc.dimL < maxcoc)do  
    begin  
      coc.dimL:= coc.dimL + 1;  
      coc.datos[coc.dimL]:= c;  
      leerCocinero (c);  
    end;  
end;
```

# Estructura de datos-ARREGLO – EJERCICIO 3

---

```
procedure MinimoCosto (coc:cocineros;p:platos;var min:cocinero);  
Var i:rangococ; costomin:real;  
Begin  
  costomin:=9999;  
  for i:=1 to coc.DimL do  
    if (p[coc.datos[i].codiplato]< costomin )  
    then begin  
      costomin:= p[coc.datos[i].codiplato];  
      min:= coc.datos[i];  
    end;  
end;
```

Ver código en  
Pascal