

# Informática

## Programación estructurada

Franco Di Pietro

Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

[dipietro@fceia.unr.edu.ar](mailto:dipietro@fceia.unr.edu.ar)

abril 2018

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

Secuencia

Selección

Simple

Múltiple

Iteración

Para

Mientras

## 1 Introducción

## 2 Programación Estructurada

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

Secuencia

Selección

Simple

Múltiple

Iteración

Para

Mientras

## 1 Introducción

## 2 Programación Estructurada

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

Secuencia

Selección

Simple

Múltiple

Iteración

Para

Mientras

Hasta ahora vimos, entre otras cosas:

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

Secuencia

Selección

Simple

Múltiple

Iteración

Para

Mientras

Hasta ahora vimos, entre otras cosas:

- algoritmos

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

Secuencia

Selección

Simple

Múltiple

Iteración

Para

Mientras

Hasta ahora vimos, entre otras cosas:

- algoritmos
- acciones primitivas

Hasta ahora vimos, entre otras cosas:

- algoritmos
- acciones primitivas
- expresiones

Hasta ahora vimos, entre otras cosas:

- algoritmos
- acciones primitivas
- expresiones

¿Cómo utilizar todas estas herramientas en conjunto?

Hasta ahora vimos, entre otras cosas:

- algoritmos
- acciones primitivas
- expresiones

¿Cómo utilizar todas estas herramientas en conjunto?



¿Cómo construir un algoritmo?

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

Secuencia

Selección

Simple

Múltiple

Iteración

Para

Mientras

1 Introducción

2 Programación Estructurada

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

Secuencia

Selección

Simple

Múltiple

Iteración

Para

Mientras

## ¿Qué es la programación estructurada?

Informática

Franco Di Pietro

Introducción

Programación Estructurada

Secuencia

Selección

Simple

Múltiple

Iteración

Para

Mientras

## ¿Qué es la programación estructurada?

Es una forma de desarrollar programas (algoritmos).

## ¿Qué es la programación estructurada?

Es una forma de desarrollar programas (algoritmos).

En ella el algoritmo podrá ser escrito utilizando únicamente tres tipos de estructuras u organización de las acciones:

## ¿Qué es la programación estructurada?

Es una forma de desarrollar programas (algoritmos).

En ella el algoritmo podrá ser escrito utilizando únicamente tres tipos de estructuras u organización de las acciones:

- secuencias

## ¿Qué es la programación estructurada?

Es una forma de desarrollar programas (algoritmos).

En ella el algoritmo podrá ser escrito utilizando únicamente tres tipos de estructuras u organización de las acciones:

- secuencias
- selecciones

## ¿Qué es la programación estructurada?

Es una forma de desarrollar programas (algoritmos).

En ella el algoritmo podrá ser escrito utilizando únicamente tres tipos de estructuras u organización de las acciones:

- secuencias
- selecciones
- iteraciones

## Secuencia

Es un conjunto de acciones ordenadas de tal manera que cada una determina cual es la siguiente.

## Secuencia

Es un conjunto de acciones ordenadas de tal manera que cada una determina cual es la siguiente.

```
acción 1  
acción 2  
.  
.  
.  
acción n
```

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

**Secuencia**

Selección

Simple

Múltiple

Iteración

Para

Mientras

## Ejemplo

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

**Secuencia**

Selección

Simple

Múltiple

Iteración

Para

Mientras

## Ejemplo

**Algoritmo** suma  
**variables**

**inicio**

**fin**

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

**Secuencia**

Selección

Simple

Múltiple

Iteración

Para

Mientras

## Ejemplo

```

Algoritmo suma
  variables
    real: n1, n2, suma
  inicio

  fin
  
```

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

Secuencia

Selección

Simple

Múltiple

Iteración

Para

Mientras

## Ejemplo

```
Algoritmo suma  
  variables  
    real: n1, n2, suma  
inicio  
  Leer(n1, n2)  
  
fin
```

## Ejemplo

```

Algoritmo suma
  variables
    real: n1, n2, suma
  inicio
    Leer(n1, n2)
    suma ← n1 + n2

  fin
  
```

## Ejemplo

```

Algoritmo suma
  variables
    real: n1, n2, suma
  inicio
    Leer(n1, n2)
    suma ← n1 + n2
    Escribir("suma = ", suma)
  fin
  
```

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

Secuencia

Selección

Simple

Múltiple

Iteración

Para

Mientras

## Selección

Estructura que organiza la ejecución de las acciones después de la evaluación de una expresión.

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

Secuencia

Selección

Simple

Múltiple

Iteración

Para

Mientras

## Selección

Estructura que organiza la ejecución de las acciones después de la evaluación de una expresión.

Tipos:

## Selección

Estructura que organiza la ejecución de las acciones después de la evaluación de una expresión.

Tipos:

- simple

## Selección

Estructura que organiza la ejecución de las acciones después de la evaluación de una expresión.

Tipos:

- simple
- múltiple

```

si (expresión) entonces
    acción/es caso verdadero
sino // opcional
    otra/s acción/es caso falso
fin si
    
```

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

Secuencia

Selección

**Simple**

Múltiple

Iteración

Para

Mientras

## Ejemplo

Mostrar el mayor de dos números reales distintos ingresados.

## Ejemplo

Mostrar el mayor de dos números reales distintos ingresados.

**Algoritmo** mayor\_2nros  
**variables**

**inicio**

**fin**

## Ejemplo

Mostrar el mayor de dos números reales distintos ingresados.

**Algoritmo** mayor\_2nros

**variables**

**real:** n1, n2

**inicio**

**fin**

## Ejemplo

Mostrar el mayor de dos números reales distintos ingresados.

**Algoritmo** mayor\_2nros

**variables**

**real:** n1, n2

**inicio**

**Leer**(n1, n2)

**fin**

## Ejemplo

Mostrar el mayor de dos números reales distintos ingresados.

```

Algoritmo mayor_2nros
  variables
    real: n1, n2
  inicio
    Leer(n1, n2)
    si(n1 > n2) entonces

        fin si
  fin
  
```

## Ejemplo

Mostrar el mayor de dos números reales distintos ingresados.

```

Algoritmo mayor_2nros
  variables
    real: n1, n2
  inicio
    Leer(n1, n2)
    si(n1 > n2) entonces
      Escribir(n1)

    fin si
  fin
  
```

## Ejemplo

Mostrar el mayor de dos números reales distintos ingresados.

```

Algoritmo mayor_2nros
  variables
    real: n1, n2
  inicio
    Leer(n1, n2)
    si(n1 > n2) entonces
      Escribir(n1)
    sino

    fin si
  fin
  
```

## Ejemplo

Mostrar el mayor de dos números reales distintos ingresados.

```

Algoritmo mayor_2nros
  variables
    real: n1, n2
  inicio
    Leer(n1, n2)
    si(n1 > n2) entonces
      Escribir(n1)
    sino
      Escribir(n2)
    fin si
  fin
  
```

```

según sea(variable)
  caso cte/s_1:
    acción/es caso cte/s_1
  caso cte/s_2:
    acción/es caso cte/s_2
  :
  caso cte/s_n:
    acción/es caso cte/s_n
sino // opcional
  otra/s acción/es
fin según
    
```

Informática

Franco Di Pietro

Introducción

Programación  
Estructurada

Secuencia

Selección

Simple

**Múltiple**

Iteración

Para

Mientras

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

**Algoritmo** `cant_dias_mes`  
**variables**

**inicio**

**fin**

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

```

Algoritmo cant_dias_mes
variables
    entero: mes
inicio
    
```

**fin**

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

```

Algoritmo cant_dias_mes
  variables
    entero: mes
inicio
  Leer(mes)
  
```

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

```

Algoritmo cant_dias_mes
variables
    entero: mes
inicio
    Leer(mes)
    según sea(mes)

```

```

        fin según
fin

```

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

```

Algoritmo cant_dias_mes
variables
    entero: mes
inicio
    Leer(mes)
    según sea(mes)
        caso 1,3,5,7,8,10,12:

```

```

        fin según
    fin

```

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

```

Algoritmo cant_dias_mes
variables
    entero: mes
inicio
    Leer(mes)
    según sea(mes)
        caso 1,3,5,7,8,10,12:
            Escribir(" 31 días")

```

```

        fin según
fin

```

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

```

Algoritmo cant_dias_mes
variables
    entero: mes
inicio
    Leer(mes)
    según sea(mes)
        caso 1,3,5,7,8,10,12:
            Escribir(" 31 días")
        caso 4,6,9,11:

```

```

        fin según
fin

```

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

```

Algoritmo cant_dias_mes
variables
    entero: mes
inicio
    Leer(mes)
    según sea(mes)
        caso 1,3,5,7,8,10,12:
            Escribir(" 31 días")
        caso 4,6,9,11:
            Escribir(" 30 días")

    fin según
fin
    
```

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

```

Algoritmo cant_dias_mes
variables
    entero: mes
inicio
    Leer(mes)
    según sea(mes)
        caso 1,3,5,7,8,10,12:
            Escribir(" 31 días")
        caso 4,6,9,11:
            Escribir(" 30 días")
        caso 2:

    fin según
fin
    
```

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

```

Algoritmo cant_dias_mes
variables
entero: mes
inicio
  Leer(mes)
  según sea(mes)
    caso 1,3,5,7,8,10,12:
      Escribir(" 31 días")
    caso 4,6,9,11:
      Escribir(" 30 días")
    caso 2:
      Escribir(" 28 días")

  fin según
fin
  
```

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

```

Algoritmo cant_dias_mes
  variables
    entero: mes
  inicio
    Leer(mes)
    según sea(mes)
      caso 1,3,5,7,8,10,12:
        Escribir(" 31 días")
      caso 4,6,9,11:
        Escribir(" 30 días")
      caso 2:
        Escribir(" 28 días")
    sino

  fin según
fin
  
```

## Ejemplo

Mostrar la cantidad de días que tiene un mes ingresado.

```

Algoritmo cant_dias_mes
  variables
    entero: mes
  inicio
    Leer(mes)
    según sea(mes)
      caso 1,3,5,7,8,10,12:
        Escribir(" 31 días")
      caso 4,6,9,11:
        Escribir(" 30 días")
      caso 2:
        Escribir(" 28 días")
      sino
        Escribir(" Mes equivocado")
    fin según
  fin

```

## Iteración

Estas estructuras permiten controlar la repetición de las acciones. Las mismas se pueden repetir una cantidad conocida de veces o una cantidad desconocida de veces.

## Iteración

Estas estructuras permiten controlar la repetición de las acciones. Las mismas se pueden repetir una cantidad conocida de veces o una cantidad desconocida de veces.

Tipos:

## Iteración

Estas estructuras permiten controlar la repetición de las acciones. Las mismas se pueden repetir una cantidad conocida de veces o una cantidad desconocida de veces.

Tipos:

- cantidad conocida de repeticiones

## Iteración

Estas estructuras permiten controlar la repetición de las acciones. Las mismas se pueden repetir una cantidad conocida de veces o una cantidad desconocida de veces.

Tipos:

- cantidad conocida de repeticiones
- cantidad **no** conocida de repeticiones

```

Repetir para var_ctrl ← v0, vf, inc
    acción 1
    acción 2
    .
    .
    .
    acción n
fin para
    
```

## Ejemplo

Mostrar los número naturales del 1 al 20.

## Ejemplo

Mostrar los número naturales del 1 al 20.

**Algoritmo** nros\_1a20  
**variables**

**inicio**

**fin**

## Ejemplo

Mostrar los número naturales del 1 al 20.

```

Algoritmo nros_1a20
  variables
    entero: i
  inicio

```

```

fin

```

## Ejemplo

Mostrar los número naturales del 1 al 20.

**Algoritmo** nros\_1a20

**variables**

entero: i

**inicio**

**Escribir**(i)

**fin**

## Ejemplo

Mostrar los número naturales del 1 al 20.

```

Algoritmo nros_1a20
  variables
    entero: i
  inicio
    Repetir para ← , ,
      Escribir(i)
    fin para
  fin
  
```

## Ejemplo

Mostrar los número naturales del 1 al 20.

```

Algoritmo nros_1a20
  variables
    entero: i
  inicio
    Repetir para i ← 1, 20,
      Escribir(i)
    fin para
  fin
  
```

## Ejemplo

Mostrar los número naturales del 1 al 20.

```

Algoritmo nros_1a20
  variables
    entero: i
  inicio
    Repetir para i ← 1, ,
      Escribir(i)
    fin para
  fin
  
```

## Ejemplo

Mostrar los número naturales del 1 al 20.

```

Algoritmo nros_1a20
  variables
    entero: i
  inicio
    Repetir para i ← 1, 20,
      Escribir(i)
    fin para
  fin
  
```

## Ejemplo

Mostrar los número naturales del 1 al 20.

```

Algoritmo nros_1a20
  variables
    entero: i
  inicio
    Repetir para i ← 1, 20, 1
      Escribir(i)
    fin para
  fin
  
```

## Repetir mientras (expresión)

acción 1

acción 2

.

.

.

acción n

**fin mientras**

## Ejemplo

Ingresar números hasta tanto se ingrese uno entre 1 y 10.

## Ejemplo

Ingresar números hasta tanto se ingrese uno entre 1 y 10.

**Algoritmo** ingreso\_num\_entre\_1y10  
**variables**

**inicio**

**fin**

## Ejemplo

Ingresar números hasta tanto se ingrese uno entre 1 y 10.

**Algoritmo** ingreso\_num\_entre\_1y10

**variables**

entero: n

**inicio**

**fin**

## Ejemplo

Ingresar números hasta tanto se ingrese uno entre 1 y 10.

**Algoritmo** ingreso\_num\_entre\_1y10

**variables**

entero: n

**inicio**

**Leer**(n)

**fin**

## Ejemplo

Ingresar números hasta tanto se ingrese uno entre 1 y 10.

```

Algoritmo ingreso_num_entre_1y10
  variables
    entero: n
  inicio

    Repetir mientras(          )
      Leer(n)
    fin mientras
  fin
  
```

## Ejemplo

Ingresar números hasta tanto se ingrese uno entre 1 y 10.

```

Algoritmo ingreso_num_entre_1y10
  variables
    entero: n
  inicio

    Repetir mientras (n<1 o n>10)
      Leer(n)
    fin mientras
  fin
  
```

## Ejemplo

Ingresar números hasta tanto se ingrese uno entre 1 y 10.

```

Algoritmo ingreso_num_entre_1y10
  variables
    entero: n
  inicio
    Leer(n)
    Repetir mientras (n<1 o n>10)
      Leer(n)
    fin mientras
  fin
  
```