

# TP 1 MAPLE - INTRODUCCION

MASTER T.E.C.I.

## 1 OPERACIONES DE BASE

```
> 1+3
> 4-2
> 3*5
> 6/8
> 3^ 3
> 10^ (-6)
> sqrt(5)
> Pi
> exp(0)
> exp(2)
> log(1)
> cos(Pi)
> sin(0)
> tan((1/2)*pi)
> abs(-5)
> factorial(10)
> round(5.5)
> ceil(5.4)
> floor(5.4)
```

## 2 FORMATO DE ESCRITURA

```
> 2+5 :
> 2+5, 2+6
> 2+5 ; 2+6 ; 2+7
> Pi
> evalf(Pi)
> cos(.9898*Pi)
> evalf(cos(.9898*Pi))
> evalf[3](cos(.9898*Pi))
```

### 3 ASIGNACION DE VARIABLES

```
> X := 5
> X+5
> unassign('X')
> X+5
> restart :
```

### 4 MATRICES Y VECTORES

```
> Y := Vector([1, 2, 3, 4])
> Y+Y
> with(linalg)
> MA := Matrix([[1, 2], [3, 4]])
> MA2 := Matrix(2, 3)
> MA3 := Matrix(1 .. 4, 1 .. 5, 5)
> VD := [1, 2, 3]
> MD := Matrix(1 .. 3, 1.3, VD, shape = diagonal)
> det(MA)
> trace(MA)
> Transpose(MA)
> with(LinearAlgebra) :
> MA := Matrix([[1, 2], [3, 4]])
> Eigenvalues(MA)
> MA := Matrix([[1, 0], [0, 4]])
> MatrixInverse(MA)
> Y := Vector([1, 2])
> MA.Y
```

### 5 COORDENADAS

```
> MA(1, 2)
> MA(1, 2) := 3
> MA(1 .. 2, 2) := 0
```

### 6 POLINOMIOS

```
> P := x^3 - x^2 + 3
> subs(x = 3, P)
> fsolve(P = 0, x, complex)
```

### 7 FUNCIONES

```
> f1 := x -> (x^2 + 1)
> f1(3)
```

```
> f2 := (x, y) -> (x^2 + y^2)
> f(1, 1)
```

## 8 NUMEROS ALEATORIOS

```
> rand()
> rand()
```

## 9 SALIDA TEXTUAL

```
> print(hola soy yo)
> x := 10
> print(el valor de 'x' es, x)
```

## 10 SALIDA GRAFICA

```
> func := x -> cos(x) + sin(x)
> plot(func)
> plot(func(x), x = -5 .. 5)
> func2d := (x, y) -> (exp(x) + log(y))
> plot3d(func2d(x, y), x = -10 .. 10, y = 1 .. 5)
```

## 11 SALIDA EN FICHERO

```
> MA3 := matrix([1, 2, 3], [2, 4, 6], [1.8, 3.1, 6.7])
> writedata(sol, MA3)
```

## 12 TESTS LOGICOS

```
> x := 5
> evalb(x > 4)
> evalb(x = 4)
> evalb(x <= 4)
> evalb(x <> 5)
> y := 10
> evalb(x = 5 and y = 10)
> evalb(x = 5 or y = 11)
> evalb(x = 5 xor y = 10)
```

## 13 PROGRAMACION

```
> x := 99
> if x > 99 then print('x' es demasiado grande) else print(valor de 'x' correcto) end if
```

```
> for i by 2 to 10 do print(la raiz de, i, es, evalf(sqrt(i))) end do  
> for n to 20 while n < 10 do print(el cuadrado de, n, es, evalf(n ^ 2)); sol_n := n^ 2  
end do
```

## 14 EJERCICIOS

- 1) Escribir un 'script' que permite calcular el factorial de un numero entero
- 2) Dibujar las funciones  $f(x) = x^n$  con  $n=1\dots 10$  y  $x$  en  $[-5,5]$