



# Tecnológico de Monterrey

Campus Monterrey

Diseño de Compiladores

Ing. Elda Quiroga González

Ing. Héctor Ceballos Cancino

## **Manual de Usuario Patito++**

---

Rafael Serna Salazar  
A01282298

---

Héctor de Luna Pámanes  
A01282272

# Instalación

La única librería adicional que se debe instalar para poder correr Patito++ es numpy, se puede descargar desde la terminal con el siguiente comando:

```
$ pip3 install numpy
```

Una vez descargado todos los archivos y teniendo instalado numpy, lo primero que se debe hacer es navegar dentro de la terminal al directorio donde existen las descargas, después solo se necesita correr el siguiente comando:

```
$ python3 analizador_lark.py ./test/x.ppp
```

en este ejemplo un el archivo a ejecutar es x.ppp, que se encuentra en un folder llamado *test* en el mismo directorio que analizador\_lark.py, eso puede variar dependiendo de cada usuario.

## Reglas del lenguaje

---

### Estructura general

```
Programa nombre_programa;  
<declaracion variables globales>  
  
<definicion de funciones>  
  
principal () {  
    <Estatutos>  
}
```

---

### Declaración de variables

```
var <tipo> <id>;  
  
var int x;
```

En Patito++ las variables pueden ser locales o globales.

Existen cuatro posibles tipos:

- Int

- Float
- Char
- Bool

Además de la posibilidad de crear variables dimensionadas

```
var <tipo> <id>[x][y];  
var int matriz[2][4];
```

Donde por ejemplo, se creo una matriz de enteros con una dimensión de 2x4

---

## Declaración de funciones

En Patito++ se pueden declarar funciones de la siguiente manera

```
funcion <tipo-retorno> <id>(<parametros>);  
<variables locales> {  
    <estatutos>  
}  
  
funcion int operacion (int x);  
var int y {  
    y = 5;  
    regresa(x + y);  
}
```

El ejemplo anterior solo es para ilustrar el posible uso de las funciones, pueden ser de los mismo tipos que las variables, además de un tipo void, el cual no requiere del estatuto regresa.

Patito++ soporta llamadas recursivas.

---

## Lectura

Patito++ soporta la lectura de identificadores del usuario.

```
lee(id, id2, id3);  
lee(arr[1], x);
```

Además Patito++ permite la lectura la mismo tiempo, siempre y cuando los ids sean separados por comas.

---

## Escritura

Patito++ puede escribir letreros en pantalla, usando la misma sintaxis que Lectura

```
escribe(id, id2, id3);  
escribe(arr[1], x, "Hola");
```

---

## Estatuto de decisión

Patito++ cuenta con un estatuto de decisión IF

```
si (<expresion>) entonces {  
    <estatutos>  
}  
sino {  
    <estatutos>  
}  
  
si (x > y) entonces {  
    escribe(x);  
}  
sino {  
    escribe("No");  
}
```

El sino es opcional.

---

## Estatuto de repetición condicional

Patito++ cuenta con un ciclo tipo WHILE.

```
mientras (<expresion>) haz {  
    <estatutos>  
}  
  
mientras (x > y) haz {  
    escribe(x);  
    x = x - 1;  
}
```

---

## Estatuto de repetición no-condicional

Patito++ cuenta con un ciclo de repetición tipo FOR

```
desde <id> = <exp> hasta <exp> haz {  
    <estatutos>  
}  
  
desde (x = 0) hasta (x > 5) haz {  
    escribe(x);  
}
```

---

## Expresiones

Patito++ cuenta con todas las expresiones tradicionales:

+, -, \*, /, &, |, <, >, ==, !=, <=, >=.

---

## Característica especial

Patito++ cuenta con unas operaciones de matrices incluidas con estas 3 expresiones especiales:

- j : genera la transpuesta de una matriz
- ? : genera la inversa de una matriz
- \$ : calcula el determinante de una matriz

```
detC = C$  
invA = A?  
transB = Bi
```

Además, Patito++ cuenta con operaciones de matrices tipo +, -, \*, /, etc.

---

## Comentarios

Los comentarios en Patito++ se escriben como

```
%% Esto es un comentario
```