**Manual de Usuario BIXO**

Lenguaje de programación orientado a machine learning y data science, desarrollado con python.

# **Setup**

Bixo se realizó en Python por lo cual es esencial tenerlo para su ejecución.

Se instalaron librerías para el uso de Machine Learning y PLY, se deberá escribir los siguientes comandos en la terminal.

*pip install ply*

*pip install numpy*

*pip install tensorflow*

Se debera clonar el siguiente repositorio:

*git clone* [*https://github.com/JCGranadosV/BIXO*](https://github.com/JCGranadosV/BIXO)

## **Variables**

Bixo maneja las variables de manera local y global. Se pueden declarar sin valor y una vez declaradas se les puede asignar valor.

*var float j;*

*j = 11;*

## **Ciclos**

Bixo maneja únicamente ciclos de tipo while.

*while(c!=3 & m==2 | m<10){*

*m=m+1;*

*e=1+1+2\*4;*

*print(m);*

*};*

## **Funciones**

Bixo maneja funciones de tipo float, int y void, con y sin parámetros (separados por coma). Las funciones de tipo non-void deben tener un return. Las funciones se declaran de esta manera:

*function void prueba(int a){*

*var int c;*

*c=1+2+3\*5\*6;*

*}*

## **Arrays**

Bixo maneja arreglos de 1 dimensiones

*array a[5]=[2,4,6,8,10,12];*

También maneja el find: El cual despliega el índice donde se encuentra el número dentro de un array.

*find(a,6);*

## **Matrices**

Bixo maneja arreglos de 2 dimensiones

*matrix m[3][3]=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16];*

*matrix z[3][3]=[2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32];*

También maneja multiplicaciones matriciales:

mmult(matrix1, matrix2): Multiplica matrix 1 por matrix2.

*mmult(m,z);*

También soporta el sort: El cual realiza y despliega el sort de una matriz en orden ascendente.

*sort(m);*

## **Funciones Especiales**

### Machine Learning

Bixo utiliza las siguientes funciones especiales para hacer predicciones.

* Layers: Creación de capa de tipo Dense, recibe como parámetro las unidades de la capa.

*layers(units=1);*

* Sequential: Inicialización de modelo secuencial para realizar el análisis.

*sequential();*

* Compile: Realiza una compilación de del modelo secuencial creado, utilizando el optimizador “Adam” recibe como parámetro el learning rate.

*compile(0.1);*

* Fit: Entrena nuestro modelo, recibe como parámetro los dos arreglos que utilizaremos para entrenarlo además de los “epochs” las cuales son las iteraciones de entrenamiento.

*fit(a,m, epochs = 1000);*

* Predict: Realiza una predicción a partir del modelo entrenado, recibe como parámetro el valor sobre el que se realizará la predicción, e imprime el resultado de la predicción.

*predict(220.0);*

### Cálculo

#### **Mean**

Bixo calcula el promedio de arreglos y matrices.

*array a[5]=[32,46,59,72,86,100];*

*mean(a);*

#### **Factorial**

Bixo calcula y despliega el factorial del número que recibe como parámetro.

*factorial(5);*

#### **Fibonacci**

Bixo calcula y despliega el fibonacci del número que recibe como parámetro.

*fibonacci(10);*

## **Ejecución**

Para ejecutar Bixo, se debe ejecutar la virtualMachine.py

En bixoParser.py en la sección “"testcases/pruebawhile.bixo"” Se puede modificar la variable “filename” con el archivo que se desee interactuar.