**Instituto Tecnológico y de Estudios Superiores de Monterrey Campus Monterrey**

Diseño de compiladores

Maestra: Elda Quiroga

José Manuel González Castro A01280106

Raúl Axel Suárez Martínez A01195895

Fecha de entrega: miércoles 4 de mayo de 2016

Lenguaje orientado a objetos (Objective Plox)

**Manual de usuario – Objective Plox**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

José Manuel González Castro Raúl Axel Suárez Martínez

Tabla de contenidos

[Sobre Objective Plox 3](#_Toc450040446)

[Tipos de datos 3](#_Toc450040447)

[Declaración de variables 3](#_Toc450040448)

[Declaración de una función 3](#_Toc450040449)

[Llamada de una función 3](#_Toc450040450)

[Estructura de un if, elsif, else 3](#_Toc450040451)

[Estructura de un while 3](#_Toc450040452)

[Estructura de un do while 3](#_Toc450040453)

[Estructura de un arreglo/variable dimensionada 3](#_Toc450040454)

[Declaración de una clase 3](#_Toc450040455)

[Herencia 3](#_Toc450040456)

[Funciones predeterminadas 3](#_Toc450040457)

[Instalación 4](#_Toc450040458)

# Sobre Objective Plox

Objective Plox es un lenguaje orientado a objetos diseñado específicamente para personas que quieren aprender este paradigma de programación de una manera divertida. El lenguaje cuenta con un léxico sumamente particular y más rico que el de cualquier otro lenguaje como Java, C++, etc. Es un lenguaje con funcionalidad básica como herencia, funciones recursivas, arreglos multidimensionales, polimorfismo entre métodos con herencia. A continuación se presentan algunas cosas básicas del lenguaje como los tipos de datos y las formas en las que se declaran diferentes elementos.

## Tipos de datos

* Entero 🡪 number
* Flotante/Double 🡪 decimal
* Caracter 🡪 char
* String/Cadena de caracteres 🡪 string
* Booleano 🡪 logic

## Declaración de variables

* var [tipo de dato] [nombre de la variable];
* var [open/hidden] [tipo de dato] [nombre de la variable];

## Declaración de una función

* funk [open/hidden] [tipo de dato] [nombre de la función] () {}
* funk [tipo de dato] [nombre de la función] () {}

## Llamada de una función

* [nombre de la función] ( [parámetros] );

## Estructura de un if, elsif, else

* If ([condición]) {} elsif ([condición]) {} else {}

## Estructura de un while

* while([condición]) {}

## Estructura de un do while

* do {} while([condición]);

## Estructura de un arreglo/variable dimensionada

* var [tipo de dato] [nombre de la variable] [limíte inferior…límite superior]\*;

## Declaración de una clase

* habemvs species [tipo de dato] [nombre de la clase] {}

## Herencia

* habemvs species [tipo de dato] [nombre de la clase] heirof [nombre de la clase] {}

## Funciones predeterminadas

* say(); – permite imprimir texto o una variable.
* hear(); - permite leer un valor del teclado.

# Instalación

Para poder instalar todo el proyecto primero es necesario bajar un zip con todo el proyecto. Este se puede encontrar en la siguiente liga: <https://www.dropbox.com/s/dt66oq3d0cgjc7a/Objective%20Plox.zip?dl=0>

*Nota: Para los siguientes pasos debes cumplir con el siguiente listado*

* *Tener una versión de Ruby 2.0 o superior.*
* *Tener la última versión de la gema de Rex. Liga:* [*https://github.com/tenderlove/rexical*](https://github.com/tenderlove/rexical)
* *Tener la última versión de la gema de Racc. Liga:* [*https://github.com/tenderlove/racc*](https://github.com/tenderlove/racc)

Después de bajar el zip y de contar con todo lo anterior, solo se descomprime todo en una carpeta.

Seguido de eso se crea un archivo de texto con el código del programa. Para más detalles sobre las palabras reservadas del lenguaje y la estructura sintáctica se puede consultar el documento con toda la documentación del lenguaje.

Una vez que ya se tiene el código se guarda el archivo con la extensión ‘.plox’.

Para poder compilar y ejecutar el programa se tiene que abrir la consola y se tiene que navegar hasta el folder donde está todo el programa descomprimido. Ahora solo se ejecutan 2 comandos simples:

1. rake generate
2. ruby plox.rb [nombre del archivo].plox

El primer comando permite cargar el léxico y la sintaxis de tal forma que se pueda saber por dónde hacer las evaluaciones de reducción.

El segundo comando permite crear dos objetos, uno que hace todo el proceso de compilación y otro que hace todo el proceso de ejecución con la máquina virtual. Después de ejecutar ambos comandos automáticamente se empieza a compilar el archivo y posteriormente se ejecuta exitosamente suponiendo que no haya errores.