

# Investigacion de Lenguajes - Smalltalk

Jimmy Banchón - René Balda

23 de octubre de 2013

## 1. Introducción

Smalltalk es un lenguaje de programación orientada a objeto puro, se basa en la comunicación entre objetos mediante envío de mensajes, está considerado como el primer lenguaje de este paradigma, ya que en **Smalltalk** todo es un objeto; éste influyó drásticamente en el surgimiento de lenguajes como JAVA, PHP, Python, Ruby y muchos de los actuales lenguajes orientados a objetos. **Smalltalk** es mucho más que un lenguaje de programación, es un ambiente completo de desarrollo de programas, que integra muchas características, que lo hacen un sistema de ventanas y manejador de código fuente.[1]

## 2. Características

Las principales características de **Smalltalk** son:

### 2.1. Orientado a Objetos

**Smalltalk** por ser un lenguaje de programación orientado a objeto puro, en él todo tipo de dato es una instancia de una clase, es por eso, que la comunicación entre objetos se hace por envío de mensajes.

### 2.2. Tipado Dinámico

En **Smalltalk** una variable puede tomar valores de distinto tipo en distintos momentos.

### 2.3. Compilacion JIT

**Smalltalk** es un lenguaje interpretado, es decir, el código fuente es traducido a byte-code para que luego éste sea traducido a lenguaje de máquina, ésta característica hace que el programa compilado sea multiplataforma y compatible con cualquier arquitectura.

## 2.4. **Garbage Collection:**

**Smalltalk** utiliza este mecanismo de gestión de memoria, para que el usuario no deba hacer una subrutina para liberar memoria cuando no hayan objetos referenciados a alguna variable.[2]

## 3. **Historia**

EL comienzo de lo que se conocería como programación orientada a objeto tuvo sus orígenes en Simula-67 pero luego el desarrollo fue completado con la evolución del lenguaje **Smalltalk**.

Para este paradigma la implementación y la definición estructural de los procesos se ocultó de las unidades del programa que las usan, este paradigma fue popular en los 1980s.

El principal responsable del desarrollo fue Alan Kay, entre sus principales logros fue darle mayor interactividad y uso de gráficos en los programas para que el usuario obtenga una interfaz mucho más amigable de lo que se mostraba en ese entonces.

El lenguaje de programación FLEX todavía estaba muy orientado a los especialistas, por lo que Kay decidió tomar algunas ideas de LOGO para hacerlo más fácil de usar. Así nació Smalltalk [3].

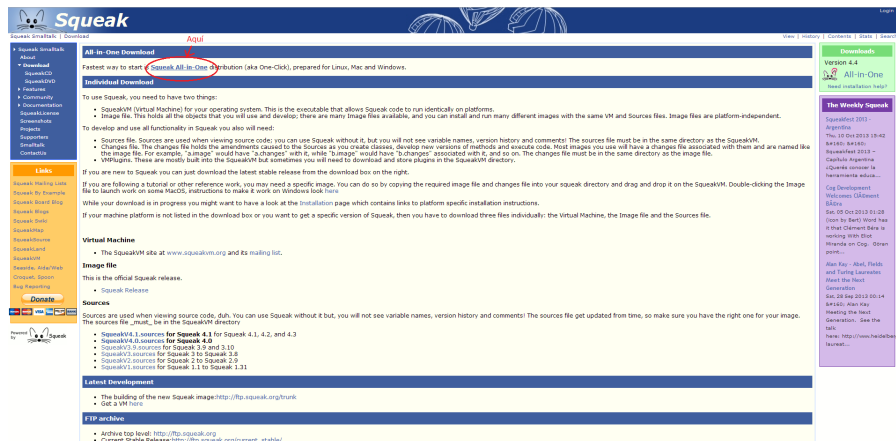
Este a su vez fue implementado en el modelo Alto salido de Xerox PARC. La primera versión fue smalltalk-80, anterior a esta fue creada smalltalk-71 desarrollada por Dan Ingalls que dio a conocer un lenguaje de programación que se podía reducir a una página de código. Pero esta versión fue bastante diferente de lo que sería la que conocemos.

Los sistemas operativos orientados a objetos de Apple Computers fueron inspirados por **Smalltalk**[4].

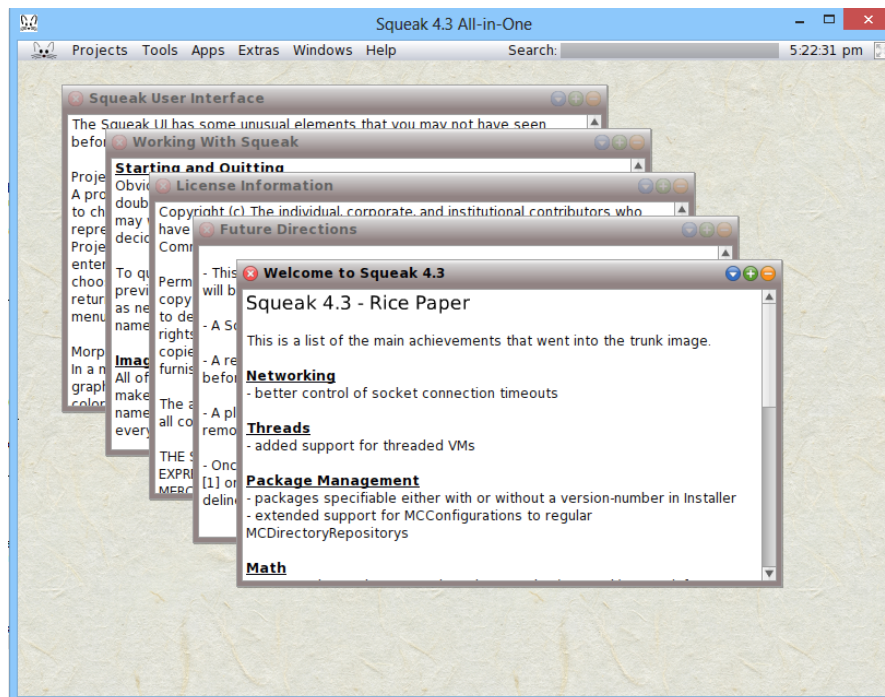
## 4. **Tutorial de Instalación**

### 4.1. **Instalación en Windows**

Para comenzar a programar en smalltalk nos descargaremos Squeak el cual es una implementación de **Smalltalk** este es el link Squeak All in one Al abrir la página nuestro link de descarga se encontrará en donde nos dice la figura siguiente.



Una vez descargado lo descomprimos en una carpeta y dentro encontraremos el ejecutable de Squeak para de aquí realizar nuestro primer programa en este lenguaje.



## 4.2. Instalacion en Linux/Debian

Para instalar **Smalltalk** en sistemas Linux/Debian se debe primero agregar el repositorio que contiene el paquete Squeak, para hacerlo agregaremos las siguientes líneas al archivo `/etc/apt/source.list`

```
deb http://ftp.squeak.org/debian/ stable main
deb-src http://ftp.squeak.org/debian/ stable main
```

Este repositorio sólo esta disponible para sistemas Debian y sus derivados.

Una vez agregadas las líneas del repositorio que contiene squeak, abriremos el terminal y ejecutaremos los siguientes comandos

```
$ sudo apt-get update
$ sudo apt-get install squeak squeak-plugin
```

Lo que actualizara la lista de repositorios e instalar Squeak, para ejecutar squeak debemos abrir la terminal y ejecutar

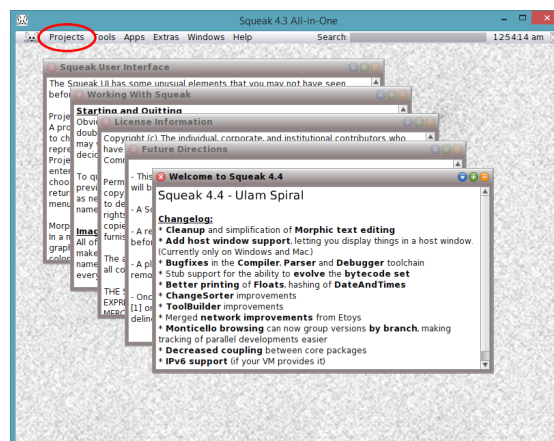
```
$ squeak
```

## 5. Hola Mundo y otros Programas Introductorios

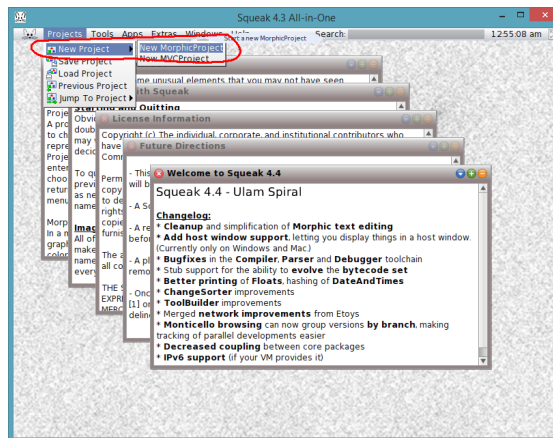
### 5.1. ¡Hola Mundo!

Para comenzar a desarrollar en **Smalltalk** debemos hacer lo siguiente:

1. En la barra de herramientas hacemos click en el menú 'Projects'



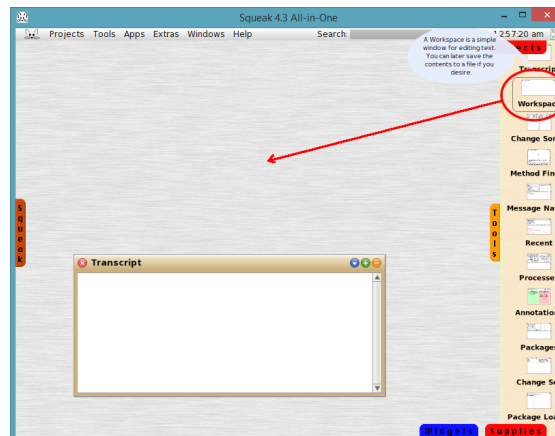
2. Elegimos la opción 'New Project', esta desplegará un submenú en el cual elegiremos la opción 'New Morphic Project'



3. A continuación se desplegará una ventana con etiquetas, seleccionaremos la etiqueta 'Tools ' en el margen derecho de la ventana.



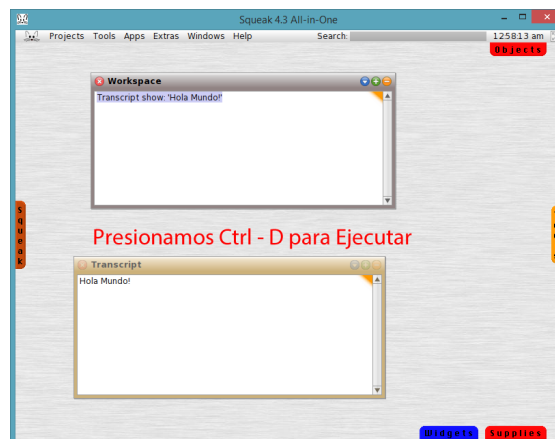
4. Esto hará que se despliegue una barra lateral en la cual aparecen varias ventanas, arrastraremos las ventanas 'Transcript' y 'Workspace' en la parte principal de la ventana.



5. En la subventana 'Workspace' escribiremos nuestro primer programa "¡Hola Mundo!" del siguiente modo:

`Transcript show: 'Hola Mundo'!`

ejecutamos con `ctrl - D`. En la subeventana 'Transcript' podremos visualizar la ejecución de nuestro programa.

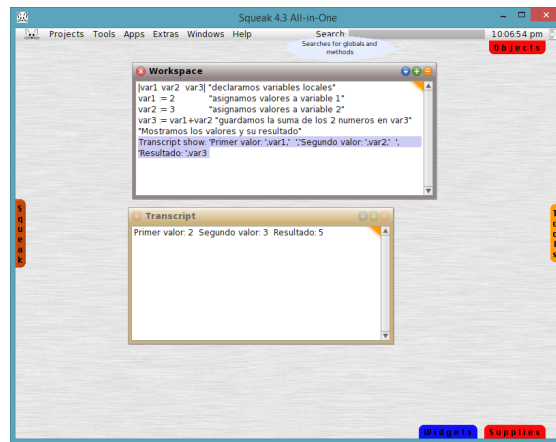


## 5.2. Suma de dos numeros

Para este ejemplo debemos seguir los pasos 1 - 4 del ejemplo anterior y en el 'Workspace' escribimos:

```
|var1 var2 var3| “declaramos variables locales”
var1 := 2          “asignamos valores a variable 1”
var2 := 3          “asignamos valores a variable 2”
var3 := var1+var2 “guardamos la suma de los 2 numeros en var3”
“Mostramos los valores y su resultado”
Transcript show: ‘Primer valor: ’,var1,‘ ’,‘Segundo valor: ’,var2,‘
’, ‘Resultado: ’,var3
```

Presionamos ctrl - D el resultado lo veremos en la venta 'Transcript'.



## Referencias

- [1] Wilf Lalonde. *Discovering Smalltalk*. 1994.
- [2] Christian Gonzales Komiya. Las 5 características de smalltalk. <http://4cuatros.blogspot.com/2009/03/las-5-caracteristicas-de-smalltalk.html>, 2009.
- [3] Cinvestav. <http://computacion.cs.cinvestav.mx/acaceres/courses/itesm/lp/clases/lp12.pdf>.
- [4] Alan C. Kay. The early history of smalltalk. *SIGPLAN Not.*, 28(3):69–95, March 1993.