

# PROGRAMACION DE VIDEOJUEGOS III



## Instalando y configurando el entorno de desarrollo

Durante éste tutorial se detallarán los pasos del proceso de instalación del toolkit haXe, así como también de la instalación y configuración de las bibliotecas y herramientas necesarias para desarrollar videojuegos con HaxeFlixel.

### Instalación de haXe toolkit

El primer paso es descargar e instalar el toolkit haXe. Instaladores para las distintas plataformas puede descargarse desde: <http://haxe.org/download>. En la Fig. 1 pueden observarse algunas capturas de pantalla del proceso de instalación en Windows. Para instrucciones detalladas sobre el proceso de instalación en ésta y otras plataformas, puede consultarse la página: <http://haxe.org/doc/install/>.

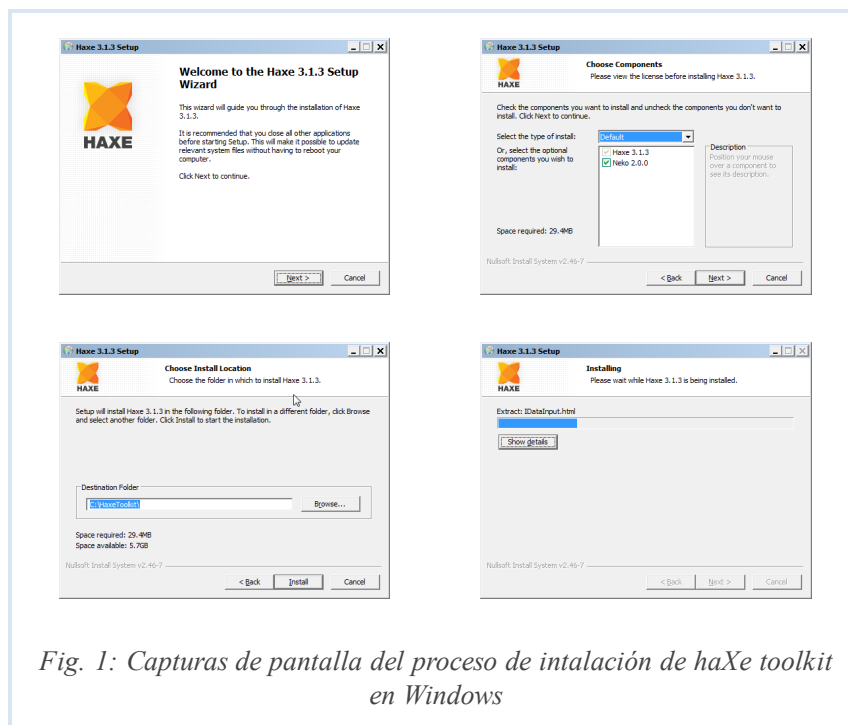
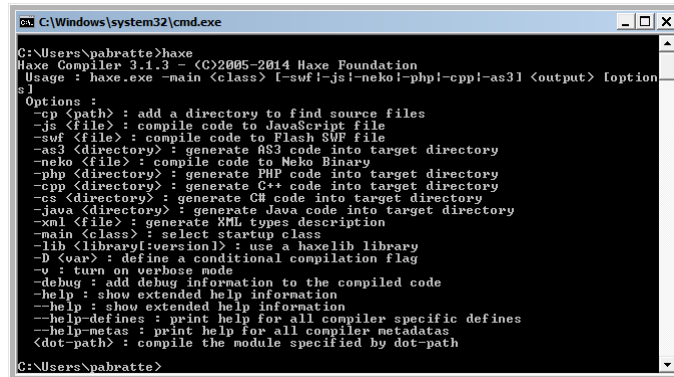


Fig. 1: Capturas de pantalla del proceso de intalación de haXe toolkit en Windows

Una vez finalizado el proceso de instalación, podemos abrir una

consola de comando (hacer click en "Inicio"->"Ejecutar" luego escribir el comando "cmd" y presionar la tecla *ENTER*) y lanzar el comando "haxe". Si la instalación finalizó correctamente, el compilador de haXe nos responderá mostrándonos las opciones disponibles como lo ilustra la *Fig. 2*.



```

C:\Windows\system32\cmd.exe
C:\Users\pabratte>haxe
Haxe Compiler 3.1.3 - (C)2005-2014 Haxe Foundation
Usage : haxe.exe -main <class> [-swf <file>|-js <file>|-neko <file>|-php <directory>|-cpp <directory>|-cs <directory>|-java <directory>|-xml <file>|-main <class>|-lib <library[:version]>|-D <var>|-v|-debug|-help|--help-extended|--help-defines|--help-metas] [-dot-path <dot-path>]
Options :
-cp <path> : add a directory to find source files
-js <file> : compile code to JavaScript file
-swf <file> : compile code to Flash SWF file
-as3 <directory> : generate AS3 code into target directory
-neko <file> : compile code to Neko Binary
-php <directory> : generate PHP code into target directory
-cpp <directory> : generate C++ code into target directory
-cs <directory> : generate C# code into target directory
-java <directory> : generate Java code into target directory
-xml <file> : generate XML types description
-main <class> : select startup class
-lib <library[:version]> : use a haxelib library
-D <var> : define a conditional compilation flag
-v : turn on verbose mode
-debug : add debug information to the compiled code
-help : show extended help information
--help : show extended help information
--help-defines : print help for all compiler specific defines
--help-metas : print help for all compiler metadata
<dot-path> : compile the module specified by dot-path
C:\Users\pabratte>

```

*Fig. 2: Invocación del compilador de haXe desde la línea de comando de Windows*

## Instalación de OpenFL, HaxeFlixel y otras bibliotecas necesarias

El toolkit haXe no consiste solamente en un compilador sino que también incluye diversas herramientas. Una de ellas es el comando "haxelib", que permite buscar, descargar e instalar bibliotecas de manera sencilla y automática.

Para instalar las bibliotecas OpenFL y HaxeFlixel, así como también sus dependencias, procederemos a ingresar en la consola de comandos cada una de las siguientes líneas:

```

haxelib install hxcp
haxelib install lime
haxelib install lime-tools
haxelib install openfl
haxelib install openfl-native
haxelib install openfl-tools
haxelib install flixel
haxelib install flixel-addons
haxelib install flixel-demos
haxelib install flixel-templates
haxelib install flixel-tools
haxelib install flixel-ui

```

Cada línea descargará e instalará una biblioteca. En caso de no contar con una conexión a internet o encontrarse detrás de un proxy, las bibliotecas pueden descargarse en formato zip desde la dirección: <http://lib.haxe.org/> e instalarse localmente luego mediante el comando:

```
haxelib test "ruta_y_nombre_del_archivo_zip"
```

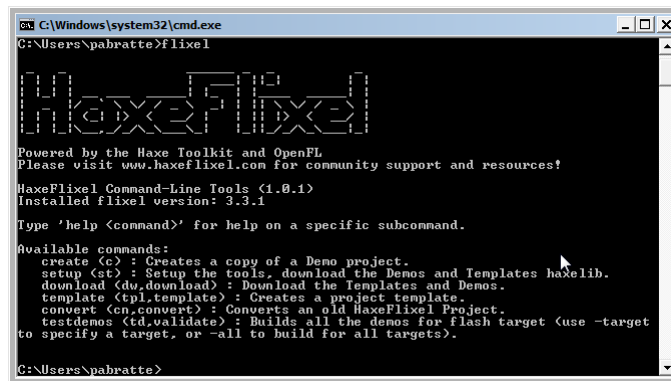
Finalmente, una vez que se hayan terminado de instalar todas las

bibliotecas, ejecutaremos el siguiente comando de consola para configurar la biblioteca HaxeFlixel:

```
haxelib run flixel-tools setup
```

Al ejecutar el comando se nos preguntará si deseamos crear un alias para el comando *"flixel"* (debemos responder que sí), y nos pedirá nuestro nombre de usuario (para incluir en la cabecera del código fuente de los proyectos que generemos), además de que seleccionemos el IDE que utilizaremos por defecto, en nuestro caso FlashDevelop.

Luego de hacer ésto, dispondremos de un nuevo comando que podremos utilizar desde la consola: el comando *"flixel"*, que podremos utilizar, como veremos más adelante, para crear nuevos proyectos. En la Fig. 3 la ayuda que despliega el comando *"flixel"* al ser invocado sin ningún parámetro.



```
C:\Windows\system32\cmd.exe
C:\Users\pabratte>flixel

HAXEFLIXEL

Powered by the Haxe Toolkit and OpenFL
Please visit www.haxeeflixel.com for community support and resources!

HaxeFlixel Command-Line Tools (1.0.1)
Installed flixel version: 3.3.1

Type 'help <command>' for help on a specific subcommand.

Available commands:
create <c> : Creates a copy of a Demo project.
setup <st> : Setup the tools, download the Demos and Templates haxelib.
download <du,download> : Download the Templates and Demos.
template <tpl,template> : Creates a project template.
convert <cn,convert> : Converts an old HaxeFlixel Project.
testdemos <td,validate> : Builds all the demos for flash target (use -target
to specify a target, or -all to build for all targets).

C:\Users\pabratte>
```

Fig. 3: Invocación del compilador de flixel desde la linea de comando de Windows

## Instalación de FlashDevelop

*FlashDevelop* es el IDE (Integrated Development Environment o Entorno Integrado de Desarrollo) que recomendamos utilizar para el curso. Se trata de un entorno que soporta, entre otros lenguaje, a haXe. Es rápido y liviano, y posee una interfaz de usuario sencilla e intuitiva y muchas características útiles como autocompletado de código y ayuda rápida.

Es posible descargar *FlashDevelop* desde <http://www.flashdevelop.org/>. En la Fig. 4 puede observarse una captura de pantalla del IDE en funcionamiento.

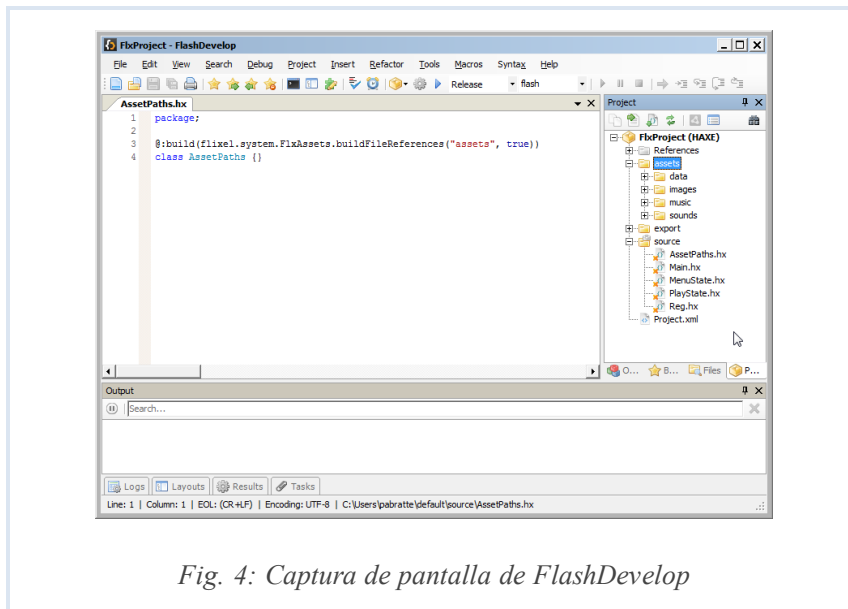


Fig. 4: Captura de pantalla de FlashDevelop

Para poder utilizar *FlashDevelop* será necesario contar con los runtimes de Java instalados, los mismo pueden descargarse desde: <http://www.java.com/es/download/>.

Para una referencia de otros IDEs alternativos que pueden utilizarse con haXe puede consultarse la página: <http://haxe.org/com/ide>.

## Flash Player Debugger

Al desarrollar videojuegos con HaxeFlixel, será recomendable probarlos primero en una plataforma que nos ofrezca un proceso rápido de construcción y testeo, y facilidades para la depuración; para luego, una vez terminado el juego, construir binarios para otras plataformas como teléfonos o tablets. Es por eso que, a pesar de estar cayendo en deshuso, la plataforma *Flash* de *Adobe®* será nuestra principal plataforma de desarrollo.

Una herramienta que nos será particularmente útil, es *Flash Player Proyector Content Debugger*, que puede descargarse desde [http://download.macromedia.com/pub/flashplayer/updaters/11/flashplayer\\_11\\_7\\_sa\\_debug.exe](http://download.macromedia.com/pub/flashplayer/updaters/11/flashplayer_11_7_sa_debug.exe) dicho programa nos permitirá correr los ejecutables de Flash y nos notificará sobre los errores en tiempo de ejecución que ocurran al hacerlo.

## Probando los demos de HaxeFlixel

El comando `"flixel"` configurado anteriormente nos permitirá construir un nuevo proyecto o probar algunos de los demos de la biblioteca. Abriendo nuevamente una consola de comandos y escribiendo el comando que figura debajo se nos exhibirá en pantalla una lista numerada con todos los demos disponibles y se nos preguntará por el número del demo que deseamos crear.

```
flixel create
```

Si, por ejemplo, ingresamos el valor "0" correspondiente al demo "Flappybalt" se nos creará en el directorio actual (por defecto la carpeta de usuario) una carpeta con dicho nombre que contendrá los archivos de proyecto del demo deseado.

Para construirlo y ejecutarlo, desde la consola, podemos ejecutar los siguientes comandos:

```
cd Flappybalt  
haxelib run openfl test flash
```

El comando "cd" (change directory) cambia el directorio de trabajo actual a la carpeta "Flappybalt" antes creada. El segundo comando ordena compilar el programa del directorio actual para la plataforma *Flash* y ejecutarlo. El producto de dicho comando será un ejecutable de *Flash*, si no hay ningún programa asociado para reproducir dicho tipo de archivo, Windows preguntará con qué programa deseamos abrirlo, en cuyo caso deberemos apuntar al ejecutable de *Flash Player Proyector Content Debugger* que descargamos anteriormente.

Para construir y ejecutar un proyecto de HaxeFlixel no es estrictamente necesario utilizar la consola. En la carpeta de proyecto creada antes, se podrá encontrar un archivo con extensión *.hxproj* que puede utilizarse para abrir el proyecto desde *Flash Develop* y realizar la construcción y prueba desde ahí. En la *Fig. 5* se observa una captura de pantalla de *FlashDevelop* en la cual se muestra la barra de herramientas que permite seleccionar la plataforma de destino y construir el ejecutable del proyecto.

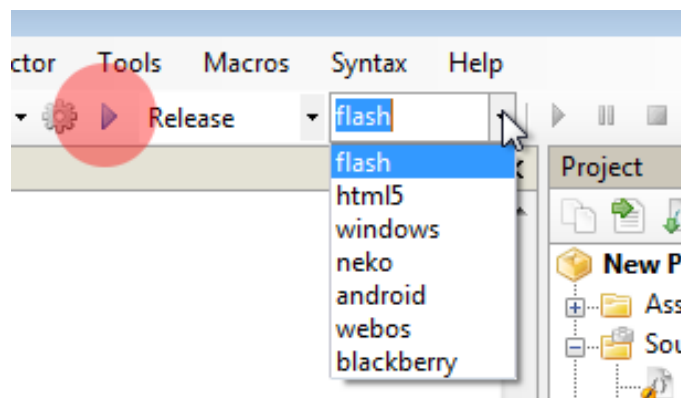


Fig. 5: Barra de herramientas para construir y correr un ejecutable en *FlashDevelop*.

A modo de introducción a las características de HaxeFlixel, se pueden probar algunos de éstos demos desde el navegador, visitando la página <http://www.haxeflixel.com/demos>. Asimismo, en la dirección <http://www.haxeflixel.com/showcase> pueden verse algunos juegos completos realizados utilizando HaxeFlixel.

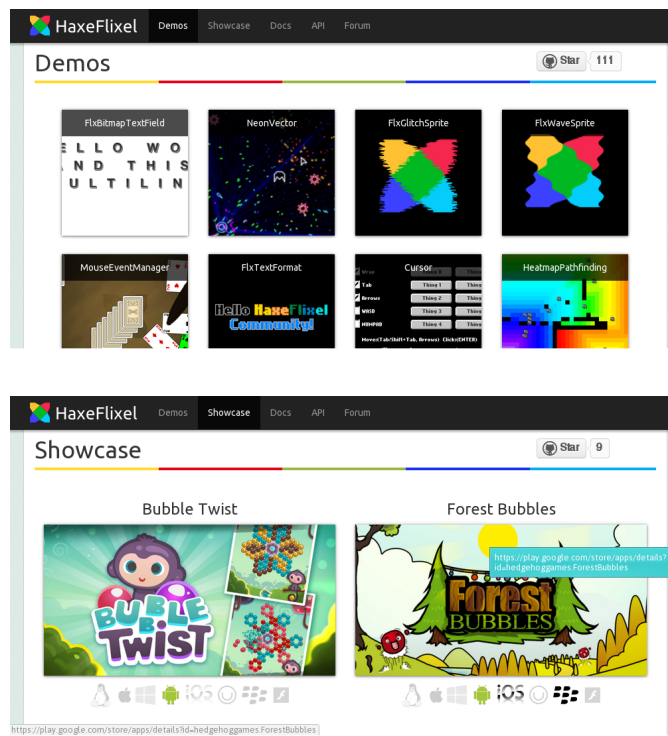


Fig. 6: Capturas de pantalla de la página de demos de HaxeFlixel (arriba) y del showcase (abajo).

[Volver al índice de tutoriales...](#)