

COMPETENCIA DE VIDA ARTIFICIAL

Instalación



1 INSTALACIÓN: PASOS GENERALES

A continuación se lista una serie de pasos que deben seguir todos los usuarios del sistema sin importar el sistema operativo y el lenguaje de programación que utilicen.

1. Descomprimir el archivo **alifecontest-java-1.0.2.zip**.
2. Instalar la máquina virtual y el entorno de desarrollo java.
3. Instalar el compilador de C/C++ (gcc y g++).
4. Iniciar el entorno.

1.1. Usuarios de Windows

1. Bajar e instalar [JDK](http://java.sun.com/javase/downloads/index.jsp) versión $\geq 1,6$ (<http://java.sun.com/javase/downloads/index.jsp>). Descargar la versión correspondiente al sistema operativo.
2. Bajar e instalar [MINGW](http://www.mingw.org) (<http://www.mingw.org>) del siguiente [link](http://sourceforge.net/projects/mingw/files/) (<http://sourceforge.net/projects/mingw/files/>).

El archivo que se debe descargar se llama mingw-get-inst-20110316.exe y se encuentra en la sección "Automated MinGW Installer/mingw-get-inst". Durante la instalación, al elegir los componentes a instalar, se deben tildar: C++ compiler y MinGW Make.

3. Configurar variable de entorno PATH de Windows: para modificar el PATH se debe acceder a las propiedades del Sistema utilizando alguna de las alternativas:
 - a) Vista Clásica: Inicio → Panel de Control → Sistema.
 - b) Vista por Categorías: Rendimiento y Mantenimiento → Sistema

Luego en el tab de Opciones Avanzadas, hacer click en Variables de Entorno, buscar PATH en la sección Variables del Sistema y hacer click en modificar para editarlo.

Se debe agregar al inicio el directorio donde se encuentran instalados los archivos binarios de java JDK, normalmente se encuentran en C:\Archivos de Programa\java\jdk1.6.XXX\bin. A MINGW, se lo encuentra generalmente en C:\Mingw\bin, también agregarlo al PATH. Por ejemplo:

Directorio de java: C:\Archivos de programa\Java\jdk1.6.0_16\bin

Directorio de MINGW: C:\MinGW\bin

Variable PATH: C:\msys\1.0\bin;

Luego de modificar el PATH queda: C:\Mingw\bin;C:\Archivos de programa\Java\jdk1.6.0_16\bin;C:\msys\1.0\bin;

Importante: el orden del path debe ser, en primer lugar el jdk, después MinGW y por último lo demás.

4. Ejecutar en consola o hacer doble click en run.bat.

1.2. Usuarios de GNU/Linux

- Bajar e instalar Java Development Kit ([JDK](#))
- Instalar g++
- Ejecutar en consola sh run.sh

1.2.1. Descripción detallada para GNU/Linux Ubuntu

- Instalar el paquete de java para desarrolladores (Java Development Kit):

```
sudo apt-get install sun-java-jdk
```
- Instalar el compilador de C++:

```
sudo apt-get install g++
```
- Iniciar el entorno:

```
sh run.sh
```



2 ACLARACIONES

2.1. Programación

1. Cuando se crea un MO en C/C++ no está permitido poner comentarios o símbolos en la línea donde se extiende de la clase Microorganism ya que el entorno utiliza esta línea para detectar el nombre de la clase, es decir, la línea debería quedar así:

```
class MiMO: public Microorganism {  
/*codigo*/  
};
```

2.2. Soporte

1. No hay soporte para arquitecturas Windows de 64 bits en el lenguaje C/C++ (solo Java).

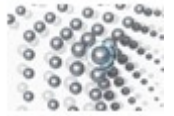
3 PROBLEMAS COMUNES

3.1. GNU/Linux Ubuntu

1. java.lang.NoClassDefFoundError: javax.tools.ToolProvider: si se instaló correctamente el compilador de java de Sun probablemente no este definido como el predeterminado. Primero realicemos la siguiente prueba en consola: `java -version` y la salida debería ser algo similar a lo que muestro abajo:

```
java version 1.6.0_16  
Java(TM) SE Runtime Environment (build 1.6.0_16-b01)  
Java HotSpot(TM) 64-Bit Server VM (build 14.2-b01, mixed mode)
```

Es importante que la segunda línea diga Java SE Runtime Environment y que la versión de java sea mayor a 1.5 (1.6 en mi caso). Si esto no se cumple, significa que tenemos la versión del compilador por defecto equivocado y tenemos que configurarlo utilizando el siguiente comando: **sudo update-alternatives -config java**. Cuando lo ejecutamos el sistema va a presentar una serie de alternativas, debemos elegir la que corresponda con la máquina virtual de java de Sun, en mi caso: `/usr/lib/jvm/java-6-sun/jre/bin/java`. Si tenemos duda podemos seleccionar alguna alternativa y luego teclear por consola `java -version` para asegurarnos que sea la alternativa correcta.



4 UTILIDADES

4.1. Util.jar

El objetivo de esta utilidad es permitir una mejor utilización del entorno.

1. Modo de uso: en forma genérica se puede invocar a Util.jar como lo indica el ejemplo:

java -jar Util.jar OPCION [PARAMETROS]

donde las opciones pueden ser MO, compile o clean y los parámetros varían de acuerdo a la opción.

2. Compilar la aplicación: el competidor puede utilizar el comando, como se muestra a continuación, para compilar y chequear la sintaxis de su código fuente (puede estar implementado tanto en el lenguaje Java como en C/C++).

java -jar Util.jar MO [Directorio de los MOs]/nombreMO.[java/h]

por ejemplo:

java -jar Util.jar MO Contest_02/MOs/miMO.java java -jar Util.jar MO Contest_02/MOs/miMO.h

3. Eliminar archivos innecesarios: utilizando el parámetro clean se puede eliminar archivos temporales y/o de compilación que estén en el proyecto.

java -jar Util.jar clean

4. Compilar el entorno: el parámetro compile permite compilar nuevamente toda la aplicación.

java -jar Util.jar compile