

# EDIPO: Un entorno para el desarrollo de aplicaciones en entornos distribuidos<sup>1</sup>

F. Almeida, D. González, L. M. Moreno, C. A. De Pablos

Dpto. de Estadística, Investigación Operativa y Computación

Universidad de La Laguna

38271, La Laguna, Tenerife

e-mail: {falmeida, dgonmor, lmmoreno}@ull.es

## Resumen

Se presenta una herramienta informática que facilita el desarrollo de aplicaciones paralelas en entornos de redes heterogéneas. El usuario obtiene una versión unificada del conjunto de máquinas sobre el que ejecutará su aplicación. Para garantizar su portabilidad, EDIPO ha sido desarrollado con herramientas estándar en los sistemas Unix/Linux. Facilita la labor de edición, compilación y ejecución de aplicaciones desde una única consola de trabajo.

## 1. Introducción

En los actuales planes de estudios de los títulos de Ingeniero en Informática es habitual encontrar asignaturas con contenidos relacionados con la Programación Distribuida o la Programación en Paralelo [7]. En ellas el alumno debe desarrollar prácticas sobre máquinas de laboratorios que pueden estar geográficamente dispersos, gestionados por distintos servidores y con diferentes sistemas de archivos conectados a través de una red.

En el desarrollo de las prácticas de estas asignaturas el alumno debe mantener abiertas sesiones de trabajo, con distintos nombres de usuario, sobre varios servidores, en las que combina la edición local o remota, la transferencia de archivos, hacia y desde los distintos sistemas, y la compilación y ejecución distribuida de las aplicaciones.

Se denomina máquina virtual al conjunto de ordenadores interconectados sobre el que el alumno va a ejecutar su aplicación. Cada uno de estos ordenadores (llamados nodos o *hosts*) puede estar situado en una localización distinta y pertenecer a redes independientes. Esta máquina virtual puede ser homogénea si todos los procesadores son idénticos y comparten el mismo sistema operativo o bien puede ser heterogénea si existen procesadores de más de una arquitectura o de más de un sistema de archivos. En este último caso los comandos necesarios para compilar y ejecutar la aplicación pueden variar de un sistema a otro. La herramienta que se presenta, EDIPO (*Environment for Distributed programming*) [7], constituye un entorno de desarrollo que permite realizar esta tarea. Proporciona todos elementos necesarios para editar, configurar, compilar y ejecutar una aplicación en una máquina virtual que puede ser heterogénea. EDIPO permite construir aplicaciones que hacen uso simultáneamente de varios procesadores desde una única máquina central (*host* de desarrollo) sin tener en cuenta cómo y dónde están distribuidas el resto de las máquinas que conforman la máquina virtual, y ejecutarlas utilizando diferentes herramientas de programación paralela.

La herramienta EDIPO está formada por una GUI implementada en Tcl/Tk [10] y un conjunto de *scripts* que se han escrito en `tclsh`, que utilizan los mecanismos de comunicación remota `rsh` y `rsh`. Tanto el lenguaje de implementación Tcl/Tk como los mecanismos de comunicación remota están habitualmente

---

<sup>1</sup> El trabajo descrito en este artículo ha sido parcialmente financiado por el Ministerio Español de Ciencia y Tecnología (CICYT). TIC1999-0754-C03

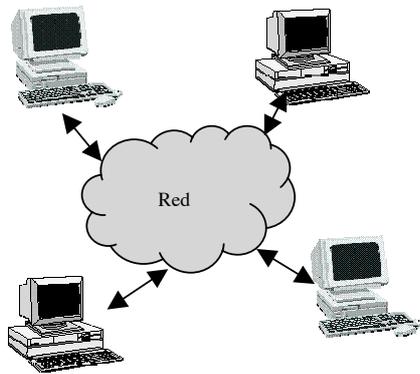


Figura 1. Máquina Virtual

instalados en la mayoría de las versiones de Unix/Linux [5, 8], lo que hace que este entorno sea fácilmente portable a la mayor parte de estos sistemas.

Dos entornos que proporcionan funcionalidades similares a EDIPO son PADE (*Parallel Application Development Environment*) [3] y WAMM (*Wide Area Metacomputing Manager*) [1].

PADE es un entorno construido por el Instituto Nacional de Estándares y Tecnología (NIST) del Gobierno Federal de Estados Unidos en 1995. Proporciona una interfaz gráfica que facilita al usuario el desarrollo de aplicaciones paralelas basándose en la librería de paso de mensajes PVM (*Parallel Virtual Machine*) [9]. Este entorno permite editar archivos, transferir un conjunto de archivos a cada nodo de la máquina virtual y compilar y ejecutar en cada uno de sus nodos.

WAMM es una interfaz gráfica basada en OSF/Motif y PVM y que ha sido desarrollada por el Grupo de Investigación en Procesamiento Paralelo del CNUCE, Pisa, en 1995. Permiten al usuario definir y gestionar una máquina virtual, tener una visión geográfica del sistema, ejecutar comandos remotos y realizar tareas de control y monitorización.

PADE y WAMM obligan a instalar y ejecutar PVM 3.0 en cada uno de los nodos, por lo que se produce un alto consumo de recursos en las máquinas y, además, la aplicación debe estar escrita en PVM. La versión actual de EDIPO soporta la ejecución sobre la librería de paso de

mensajes MPI [6] y puede ser fácilmente extendida para la ejecución con herramientas como PVM, Corba o Java RMI. Lo que proporciona una flexibilidad y generalidad superior a la de los entornos similares.

Entornos más complejos como Condor [2] o Globus [4], abordan también el problema desde una perspectiva aún más amplia. Sin embargo, se trata de entornos que implican un esfuerzo de instalación, aprendizaje y un consumo de recursos considerable para poder hacer un uso efectivo de ellos a corto plazo en docencia.

Este trabajo está organizado de la siguiente manera: En la sección 2 se describe lo que denominamos una máquina virtual. En la sección 3 se muestra el entorno de desarrollo que se presenta, cómo configurar la máquina virtual y las operaciones que pueden realizarse a través de esta herramienta. En la sección 4 se explican los pasos necesarios para su instalación. Se finaliza el trabajo en la sección 5 con algunas conclusiones y trabajos futuros.

## 2. La Máquina Virtual y las Aplicaciones

En el contexto que nos ocupa, una máquina virtual está formada por un conjunto de ordenadores o nodos que están unidos mediante una red de interconexión (Figura 1). Cada ordenador ejecuta su propio programa, accede a su memoria local y se comunica a través de la red con el resto de los nodos. Habitualmente el desarrollo de una aplicación en este entorno implica algún mecanismo de intercambio de mensajes.

La aplicación tendrá componentes que residen en sistemas de archivos de diferentes nodos con distintos sistemas operativos. Esas componentes tendrán que compilarse en cada nodo, normalmente con diferentes directivas y opciones de compilación. Las versiones compiladas se ejecutarán en cada nodo, y los flujos de entrada-salida se gestionarán en cada uno de ellos.

Dado que las fases de compilación y ejecución pueden tratarse de forma genérica, cualquiera de las plataformas software de computación paralela puede beneficiarse de EDIPO, puesto que proporciona al desarrollador una interface de alto nivel que simplifica las etapas de desarrollo de la aplicación desde una única consola de trabajo.

### 3. El Entorno de Desarrollo EDIPO

EDIPO es un entorno reconfigurable para el desarrollo de aplicaciones sobre una máquina virtual heterogénea compuesta de varias arquitecturas y/o varios sistemas de archivos. Utiliza los mecanismos de comunicación remota `rsh` y `rcp`. Proporciona un marco de trabajo para todas las fases de desarrollo de una aplicación paralela: edición, configuración, compilación y ejecución. El paquete está compuesto por una interface gráfica de usuario y un conjunto de *scripts* escritos en `Tcl/Tk`.

El entorno dispone de una consola de desarrollo para la máquina virtual, de forma que todas las tareas del desarrollo pueden realizarse desde la máquina de desarrollo (Figura 2). La consola permite:

- Definir la máquina virtual, identificando las máquinas que van a ejecutar los distintos componentes de la aplicación paralela.
- Organizar los archivos fuente, un conjunto de archivos que puede ser diferente para cada nodo de la máquina virtual.
- Editar el código fuente apropiado para cada máquina mientras se mantienen todos los archivos en una localización común
- Transferir el conjunto de archivos entre los nodos, cuando sea necesario.

- Compilar el conjunto de archivos en cada nodo, el compilador para cada arquitectura, o para cada nodo, puede variar.
- Ejecutar el programa.
- Comprobar la salida de la compilación y ejecución.

#### 3.1. Configuración de la máquina

La figura 3 muestra la pantalla principal de EDIPO. En el título de la ventana se observa el proyecto que se encuentra abierto en cada momento. Bajo el título se presenta un menú que permite realizar la gestión y el mantenimiento de estos proyectos. En el menú *Edit* se encuentran las opciones para gestionar los proyectos, los archivos de máquinas, activar las preferencias del usuario y salir de EDIPO. *Setup* es el menú que permite introducir los datos de la sesión (la lista de máquinas, los archivos a transferir, los comandos a ejecutar y los directorios de trabajo en las máquinas). Con las opciones que se encuentran en *Make* se administra el archivo de configuración, se realiza la transferencia de archivos y se ejecutan los comandos remotos. En el menú *Run* están las opciones para obtener información de las máquinas y ejecutar las aplicaciones.

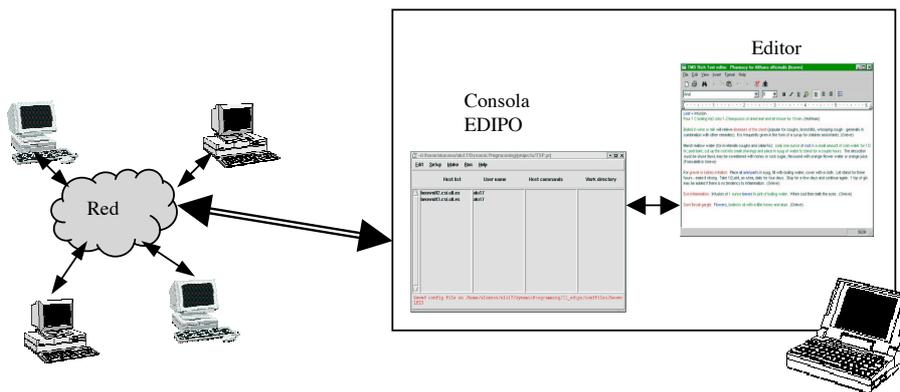


Figura 2. Entorno de desarrollo EDIPO

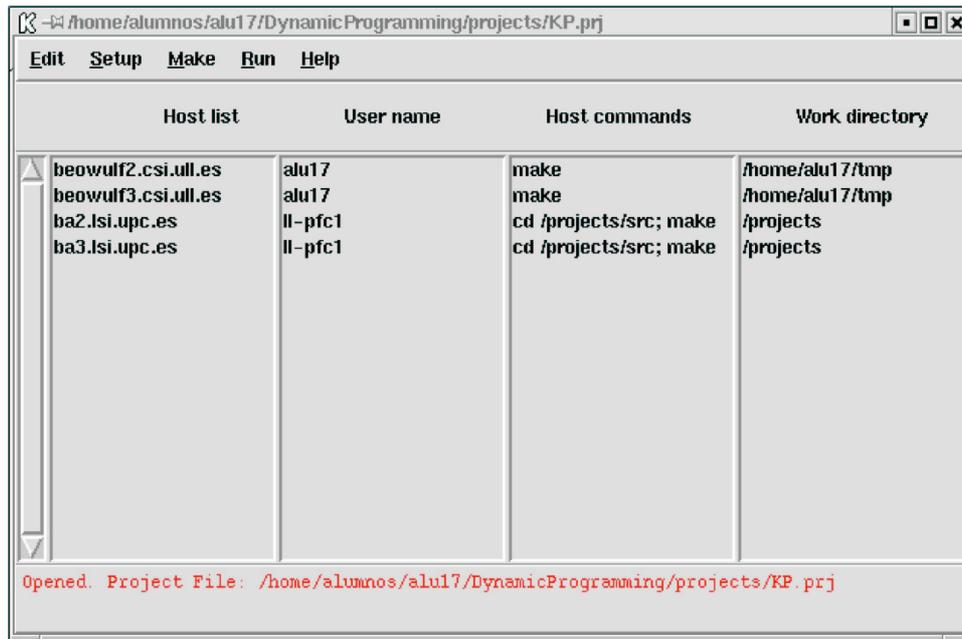


Figura 3. Pantalla principal de EDIPO

La parte principal de la ventana se divide en cuatro columnas: en la primera columna se presentan los nombres de todos los nodos que forman la máquina virtual sobre la que se está trabajando. El ejemplo que se muestra en la figura 3 es un caso en el que la máquina virtual está compuesta por cuatro nodos: las dos primeras máquinas (beowulf2.csi.ull.es y beowulf3.csi.ull.es) se encuentran localizadas geográficamente en Universidad de La Laguna, mientras que las dos últimas (ba2.lsi.upc.es y ba3.lsi.upc.es) se encuentran en la Universidad Politécnica de Cataluña.

En la segunda columna se encuentran los nombres de usuario en cada uno de los nodos. En el ejemplo el alumno tiene un nombre en las máquinas de La Laguna y otro nombre de usuario distinto en las máquinas de Cataluña. En la tercera columna se pueden ver los comandos que se quieren ejecutar en los nodos. Y en la última columna están definidos los directorios de trabajo.

Bajo las columnas se muestra el último comando ejecutado.

Definir una máquina virtual consiste simplemente en añadir cada una de las máquinas a la lista de nodos que aparece en la ventana principal. Esto se realiza mediante la entrada *Add Host* del menú *Setup*.

Otra forma de hacerlo es crear un archivo de máquinas con la opción *New HostFile* del menú *Edit*. Se escribe una máquina por línea, y después se cargan todas las máquinas del archivo a la lista mediante la entrada *Add from HostFile* del menú *Setup*.

Toda la información de la sesión se debe almacenar en un archivo de configuración de la máquina de desarrollo. Este archivo es necesario para realizar la transferencia genérica de archivos y para la compilación remota. Para almacenar el archivo de configuración se utiliza la entrada *Save Config File* del menú *Make*.

Durante la configuración de la máquina es posible incluir la información relativa a los archivos que se van a transferir, los comandos a ejecutar en esa máquina, y el directorio de trabajo y el nombre del usuario para acceder a cada nodo.

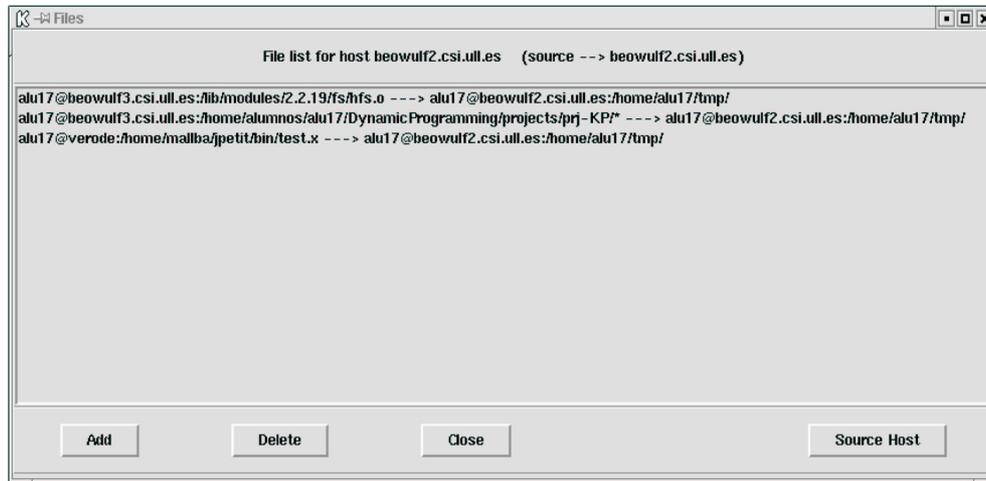


Figura 4. Pantalla de transferencias de archivos entre nodos

En lugar de introducir toda la información cada vez que se inicie una sesión, se puede cargar un archivo de configuración previamente guardado que recupere el estado general de dicha configuración (máquinas, comandos, transferencias, directorios, nombres de usuario, etc.). Para hacerlo hay que pulsar la opción *Load Config File* del menú *Make*.

La entrada *List Config File* del menú *Make* muestra en una nueva ventana el contenido del archivo de configuración. Esto permite detectar errores antes de realizar la compilación remota u otra operación, o simplemente revisar el estado actual de nuestra sesión.

El archivo de configuración sigue la estructura propuesta por PADE. Contiene cuatro tipos de elementos:

- Comentarios: Líneas que empiezan por #.
- Máquina, Nombre de usuario y Directorio de trabajo:  
+ hostname, username, workdir
- Transferencia de archivos:  
<filename\_sourcehost  
>targethost1  
filename\_targethost1  
...

```
>targethostN
    filename_targethostN
<BLANK LINE>
```

- Comandos:  
@remotehost command1; ... ;  
commandN;

### 3.2. Transferencia de Archivos

La transferencia de archivos entre las distintas máquinas se debe realizar en dos etapas: primero hay que determinar los archivos que se quieren transferir y después se realiza la copia.

Para crear la lista de los archivos a transferir a uno de nodos se elige la máquina en la pantalla principal (Figura 3) y después se selecciona la opción *File Transfer* del menú *Setup*. Este nodo será la máquina destino a la que se quieren enviar los archivos. Se muestra una ventana en la que se pueden ir añadiendo los archivos que se quieren enviar, donde se especifica el nombre del archivo y la máquina donde se encuentra.

La figura 4 muestra la ventana que aparece al elegir esta opción del menú. En el título se puede ver el nodo destino, donde se van a transferir los archivos seleccionados; en este caso de ejemplo se van a transferir tres archivos a la máquina beowulf2.csi.ull.es.

En la parte inferior de la ventana se encuentran los botones que permiten gestionar la transferencia de archivos: se puede añadir un nuevo archivo (botón *Add*) o eliminar una de las transferencias incluidas hasta el momento (botón *Delete*).

El nodo fuente, por defecto, es la máquina de desarrollo, pero también se puede cambiar pulsando sobre el botón *Source Host* que permite elegir un nuevo nodo fuente. En el ejemplo de la figura 4 los archivos que se van a transferir a beowulf2.csi.ull.es tienen dos nodos origen distintos, los dos primeros archivos se copian desde beowulf3.csi.ull.es mientras que el último archivo proviene de verode.

Cuando se ha terminado de insertar todos los archivos que se van a enviar al nodo de destino se pulsa sobre el botón *Close* para cerrar la ventana.

No es necesario almacenar las transferencias sino que al añadirlas ya quedan almacenadas automáticamente.

En la parte central de la ventana se observan las transferencias añadidas. El formato de cada una de las líneas es el siguiente:

```
usuario_nodo_origen@nodo_origen
:/ruta_archivo_origen/nombre_archivo_origen →
```

```
usuario_nodo_destino@nodo_destino
:/ruta_archivo_destino/nombre_archivo_destino
```

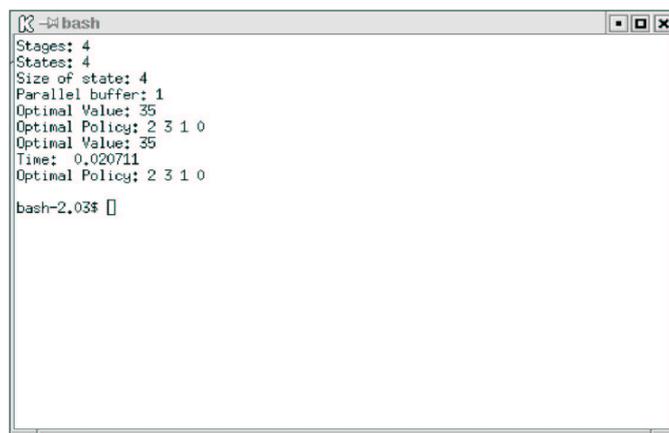
Utilizando la opción *Save Files in Remote Hosts* del menú *Make* se realizan todas las transferencias de archivos que hayan sido guardadas en el archivo de configuración actual.

Se utiliza el comando genérico de copia remota *rcp* para enviar los archivos. La transferencia se realiza de forma paralela.

### 3.3. Ejecución de comandos

Los comandos pueden introducirse mientras se añade la máquina si se utiliza la opción *Add Host* del menú *Setup*. Si se utiliza la opción que carga todas las máquinas desde un archivo, se pueden añadir los comandos seleccionando primero una máquina con el ratón y pulsando la entrada *Modify Host* del menú *Setup* que permite modificar los datos de la máquina (nombre, comandos y directorio de trabajo).

Mediante la opción *Remote Make* del menú *Make* se realiza la ejecución de todos los comandos remotos del archivo de configuración. En este caso, cada máquina ejecuta los comandos asociados de forma paralela.



```
bash
Stages: 4
States: 4
Size of state: 4
Parallel buffer: 1
Optimal Value: 35
Optimal Policy: 2 3 1 0
Optimal Value: 35
Time: 0.020711
Optimal Policy: 2 3 1 0

bash-2.03#
```

Figura 5. Pantalla de salida de ejecución por terminal

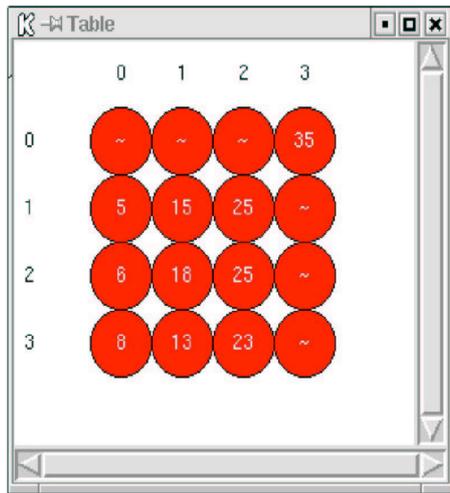


Figura 6. Pantalla de salida de ejecución gráfica

### 3.4. Compilación y ejecución de Aplicaciones

Tal y como se ha indicado en la subsección 3.3 es posible introducir cualquier comando para una máquina particular, aunque lo más común será usar el comando *make* para compilar los archivos fuente de la máquina remota.

Para la ejecución de aplicaciones EDIPO proporciona varios modos de ejecución: ejecución en modo *batch*, ejecución con salida en terminal, con salida gráfica y con salida en archivo. El usuario sólo debe escoger el modo de ejecución que quiere emplear seleccionando la entrada correspondiente del menú *Run*.

En la figura 5 se puede observar la salida que muestra EDIPO al resolver el problema del Viajante de Comercio (*TSP*) mediante programación dinámica paralela si se escoge la opción de ejecutar el proyecto con salida en un terminal.

En la figura 6 se muestra el resultado de la ejecución del problema cuando se selecciona la opción de salida gráfica. Este tipo de salida depende de la forma de resolución del problema, en este caso el método usado es la programación dinámica.

Para realizar la salida en modo gráfico se establece un protocolo entre la aplicación escrita

en cualquier lenguaje de programación tal como C, C++, Java, etc., y el entorno escrito en Tcl/Tk.

La aplicación paralela debe escribir en un archivo temporal los valores necesarios para que Tcl/Tk represente la salida de forma gráfica.

### 3.5 Otras Operaciones Disponibles en EDIPO

En cualquier momento se pueden ver las características físicas de los nodos que forman la máquina virtual. Seleccionando la opción *Info Host* del menú *Run* aparece la ventana con todas las características (Figura 7).

También se tiene la posibilidad de gestionar los proyectos que se quieren resolver. En el menú *Edit* se presentan opciones para crear un proyecto o abrir uno existente (opciones *New Project* y *Open Project*). Al crear un nuevo proyecto crea un fichero de configuración que contiene los nombres y las rutas de cada uno de los archivos fuente que componen ese proyecto.

## 4. Una Instalación Sencilla

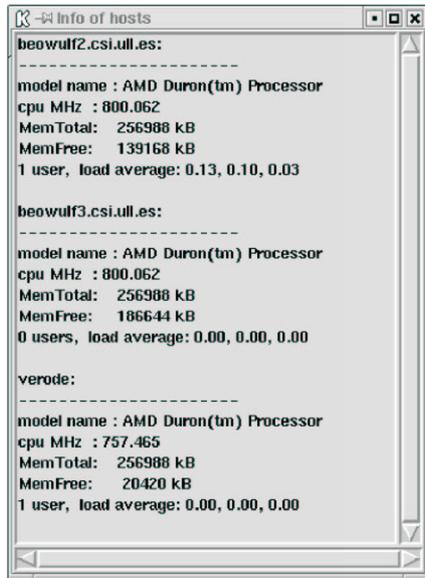
Para su instalación basta con que el alumno descargue un grupo de *scripts* y que defina y modifique algunas variables de entorno. Como se comentó en la sección 1 tanto Tcl/Tk como los mecanismos para la comunicación remota, *rcp* y *rsh*, vienen instalados por defecto en la mayoría de las versiones de Unix/Linux.

También se debe tener instalada la herramienta elegida para la ejecución de los proyectos (MPI, PVM, Java RMI, Corba).

Las variables de entorno que debe definir el alumno son las siguientes:

- LLEDIPO - Directorio en el que se instaló EDIPO.
- TCL\_LIBRARY - Directorio de la librería TCL.
- TK\_LIBRARY - Directorio de la librería TK.
- EDITOR - Editor por defecto para editar los ficheros fuente (opcional).

Debe modificarse la variable de entorno *PATH* para añadir al path de ejecutables los directorios: `\$LLEDIPO` y `\$LLEDIPO/bin`



Por último, también es necesario generar un archivo `.rhost` en el directorio `home` del usuario donde se especifican las máquinas en las que se tienen permisos para escribir en su sistema de archivos.

Una vez realizada la instalación, solamente hay que escribir EDIPO en la línea de comandos. Primero se leen las variables de entorno y se buscan en la variable de entorno `PATH` del usuario los programas que se deben ejecutar.

Si están disponibles todos los programas que se necesitan se muestra la ventana principal del entorno (figura 3).

## 5. Conclusiones y trabajos futuros

Se ha presentado una herramienta genérica que facilita el desarrollo y ejecución de una aplicación paralela sobre una máquina virtual heterogénea, lo que permite a los alumnos de las asignaturas relacionadas con la Programación Distribuida o la Programación en Paralelo realizar sus prácticas sobre redes geográficamente dispersas.

Una de las mejoras que puede llevarse a cabo en EDIPO está relacionada con la seguridad en las comunicaciones, pueden utilizarse los mecanismos de comunicación segura (como `ssh` y `scp`) en vez de utilizar lugar de los comandos `rsh` y `rcp`.

La metodología utilizada para el desarrollo de EDIPO (*scripts*, mecanismos de comunicación remota, etc.) permite adaptar y extender este entorno con nuevas y distintas funcionalidades y podría ser ampliable para que soporte otras herramientas como Java RMI.

## Referencias

- [1] R.Baraglia, D. Laforenza. *WAMM, A Metacomputer Graphical Interface*. Parallel Processing Group. CNUCE Institute, Pisa. [cnuce-arch.cnuce.cnr.it/wamm/](http://cnuce-arch.cnuce.cnr.it/wamm/)
- [2] Condor: [www.cs.wisc.edu/condor/](http://www.cs.wisc.edu/condor/)
- [3] J.E. Devaney, R. Lipman, M. Lo, W.F.Mitchell, M. Edwards, C.W. Clack. *The Parallel Applications Development Environment (PADE). User's Manual*. National Institute of Standards and Technology. [www.itl.nist.gov/div895/pade/](http://www.itl.nist.gov/div895/pade/)
- [4] Globus: [www.globus.org/](http://www.globus.org/)
- [5] *Linux in a Nutshell*, Second Edition. Ellen Siever and the Staff of O'Reilly & Associates, Inc. 1999.
- [6] *MPI: Message Passing Interface*. [www-unix.mcs.anl.gov/mpi/](http://www-unix.mcs.anl.gov/mpi/)
- [7] C.A. de Pablos Prieto. *Programación dinámica Paralela: Esqueletos y Herramientas de Desarrollo. Proyecto Final de Carrera*. Centro Superior de Informática. Dpto. de Estadística, Investigación Operativa y Computación. Universidad de La Laguna. Enero 2002.
- [8] J. Peek, T. O'reilly, M. Loukides. *Unix Power Tools*. O'Reilly & Associates, Inc. 1994.
- [9] *PVM: Parallel Virtual Machine*. [www.csm.ornl.gov/pvm/pvm\\_home.html](http://www.csm.ornl.gov/pvm/pvm_home.html)
- [10] B. Welch. *Practical Programming Tcl and Tk*. Prentice Hall. 1999