**REQUERIMIENTO GESTIÓN DE ANTEPROYECTOS CORBA**

**MANUAL DE INSTALACIÓN**

****

**INTEGRANTES**

Albeiro Silva Muñoz

Santiago Ramirez Chavez

**PROFESOR**

Ing. Daniel Eduardo Paz Perafán

**UNIVERSIDAD DEL CAUCA**

**FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES**

**PROGRAMA INGENIERÍA DE SISTEMAS**

**CURSO: SISTEMAS DISTRIBUIDOS**

**POPAYÁN, AGOSTO 2019**

Tabla de contenido

[1. INTRODUCCIÓN 3](#_Toc15594681)

[2. REQUERIMIENTOS DEL SISTEMA 4](#_Toc15594682)

[3. DESCRIPCIÓN DE LA DOCUMENTACIÓN ENTREGADA 5](#_Toc15594683)

[4. DESCRIPCIÓN DE LA ESTRUCTURA DE DIRECTORIOS, DESCRIPCIÓN DE LOS ARCHIVOS DE SOPORTE. 6](#_Toc15594684)

[4.1. DIRECTORIO SRC 6](#_Toc15594685)

[5. DESCRIPCIÓN DE LOS ARCHIVOS FUENTE (INTERFACES, ARCHIVOS FUENTE CLIENTE, ARCHIVOS FUENTE SERVIDOR) 10](#_Toc15594686)

[5.1. INTERFACES 10](#_Toc15594687)

[5.2. ARCHIVOS FUENTE CLIENTE 12](#_Toc15594688)

[5.3. ARCHIVOS FUENTE SERVIDOR 13](#_Toc15594689)

[6. DESCRICIÓN DEL PROCESO DE COMPILACIÓN 14](#_Toc15594690)

[7. DESCRIPCIÓN DEL PROCESO DE EJECUCIÓN 15](#_Toc15594691)

[7.1. EJECUCIÓN DEL SERVIDOR 16](#_Toc15594692)

[7.2. EJECUCIÓN DEL CLIENTE 17](#_Toc15594693)

[8. DESCRIPCIÓN DEL MANEJO DEL INSTALADOR 18](#_Toc15594694)

# INTRODUCCIÓN

Este documento está dirigido a los usuarios que deseen instalar el programa para gestionar anteproyectos. A lo largo del documento se describirán los aspectos necesarios para el correcto funcionamiento del software. En la apliacción se podrá encontrar el código fuente listo para compilar y ejecutar. Ya que está diseñado para funcionar como una aplicación cliente/servidor podrá ejecutarlo en una misma computadora o en dos máquinas distintas.

# REQUERIMIENTOS DEL SISTEMA

Sistema operativo

Para poder ejecutar la carpeta comprimida con códigos fuente que se le ha proporcionado se debe tener en su equipo instalado los siguientes elementos:

Para descargar e instalar java lo puedes hacer desde la página oficial de Oracle. Al instalar java se instala:

* + - **JRE** (Entorno de ejecución de Java): El JRE es la implementación de la Máquina virtual de Java que realmente ejecuta los programas de Java
    - **JDK** (Java Development Kit): El JDK es un kit de desarrollo de software que se utiliza para escribir aplicaciones con el lenguaje de programación Java.

El desarrollo y ejecución de aplicaciones en Java exige que las herramientas para compilar (javac.exe) y ejecutar (java.exe).

* + - **Javac:** Es una herramienta para leer clases y definiciones de interface escritas en java, y compilarlas al lenguaje de la máquina virtual Java (archivos bytecode).
    - **Java:** Es una herramienta para ejecutar los programas en lenguaje de máquina virtual java.

No existen requerimientos en cuanto al sistema operativo que se esté utilizando, aprovechando las ventajas que ofrece Java en cuanto a versatilidad y portabilidad de los códigos fuente.

IDE de desarrollo

* IDLE Java [Preferiblemente NetBeans versión 8.0 o superior]

Servidor local

* Servidor local Xampp para que nos permita instalar de forma sencilla Apache y además el servidor de bases de datos MySql.

Librerías

* Mysql-connector-java-5.1,46.jar: Utilizada para la conexión a la base de datos MySql.
* Jcalendar-1.4.jar: Utilizada para manejar las fechas al momento de realizar un determinado registro.

Estas dos librerías se adjuntan en el proyecto

Tegnología utilizada

* La tecnología utilizada fue java CORBA
* Motor de bases de datos MySql

Lenguaje de programación

El lenguaje de programación utilizado fue Java.

Cumpliendo con las características descritas anteriormente se asegura que la aplicación se ejecutará con total normalidad, seguridad, eficacia deseada.

# DESCRIPCIÓN DE LA DOCUMENTACIÓN ENTREGADA

Este software se entrega junto con la siguiente documentación:

**Manual de instalación** (Actual): Documento que tiene como objetivo orientar a los usuarios en la instalación del software. Describirá los requisitos hardware y software para el funcionamiento de la aplicación para la gestión de anteproyectos.

**Manual de usuario**: Documento que tiene como objetivo presentar una descripción detallada de los servicios habilitados para el usuario.

**Manual técnico**: Documento que proporciona información técnica relacionada con el desarrollo del software.

# DESCRIPCIÓN DE LA ESTRUCTURA DE DIRECTORIOS, DESCRIPCIÓN DE LOS ARCHIVOS DE SOPORTE.

Dentro de la estructura de directorio que se envió y que fue nombrada como lsd\_corba\_archivos\_fuente\_silvaMuñozA\_RamirezS, se encuentran los siguientes directorios, ver ilustración 1.

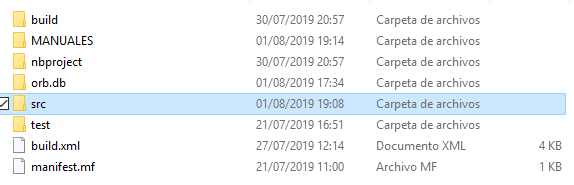


Ilustración . Estructura de directorios

En los siguientes puntos únicamente se explicará el contenido de la carpeta “src”, la cual es la que contiene los códigos fuente de la aplicación, todas las demás carpetas y archivos corresponden a ficheros generados por el entorno de desarrollo, que, aunque no son de importancia para el fin de este documento, si lo son para tener una correcta visualización del proyecto en el entorno de desarrollo, dentro de la carpeta “src” se tiene la siguiente estructura de directorios:

## DIRECTORIO SRC

Dentro del directorio **src** se encuentran los sigueintes directorios: Ver Ilustración 2.

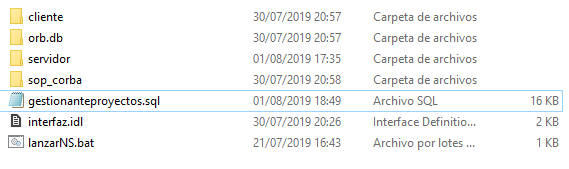


Ilustración . Directorios del directorio src

SRC/SERVIDOR

Dentro de **servidor** se encuentran los siguientes directorios: Ver Ilustración 3.

Cada uno de estos directorios tiene archivos necesarios para la ejecución del programa

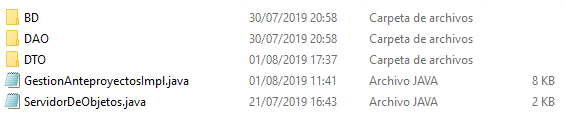


Ilustración . Directorios del directorio servidor

Src/servidor/BD

Dentro del directorio accesoDatos se encuentran los siguientes archivos: Ver Ilustración 4.

Estas clases se utilizan para gestionar el acceso a la base de datos.



Ilustración . Archivos del directorio BD

Src/servidor/DAO

Dentro del directorio DAO se encuentran los siguientes archivos. Ver Ilustración 5.

Estos archivos se utilizan para desarrollar el CRUD a la base de datos

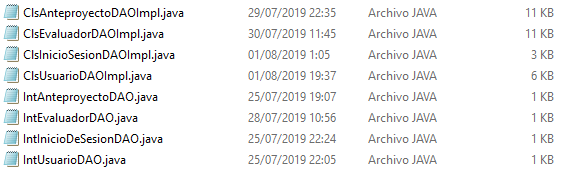


Ilustración . Archivos del directorio DAO

Src/servidor/DAO

Dentro del directorio DTO se encuentran los siguientes archivos: Ver Ilustración 6.

En este directorio se encuentran los archivos necesarios para intercambiar infomación cliente-servidor. Permite tener objetos por medio de los cuales se transporta datos entre procesos.

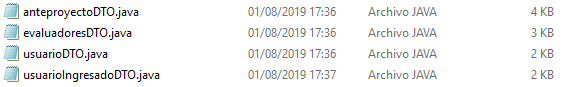


Ilustración . Archivos del directorio DTO

Src/servidor/

Dentro del directorio servidor se encuentran los siguientes archivos: Ver Ilustración 7.

En este directorio se encuentra la clase que implementa la interfaz que tiene definido los métodos remotos, también se encentra el servidor de objetos, el cual se utiliza para recibir las peticiones.



Ilustración . Archivos del directorio sop\_rmi

SRC/CLIENTE

Dentro del directorio cliente se encuentran los siguientes directorios: Ver Ilustración 8.

Dentro de cada directorio se encuentran archivos específicos para la buena ejecución del programa.



Ilustración . Directorios del directorio cliente

Src/cliente/img

Dentro del directorio img se encuentran las imágenes utilizadas para el enrriquecimiento visual de la interfaz gráfica. Ver Ilustración 9.

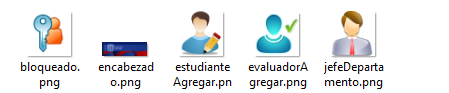


Ilustración . Imágenes utilizadas para la interfaz gráfica

Src/cliente/vistas

Dentro del directorio vistas se encuentran los archivos necesarios para generar la interfaz con la que el cliente va a interactuar, el menú del JefeDeDepartamento, menú del Evaluador, menú del EstudianteDirector...etc. Ver Ilustración 10.

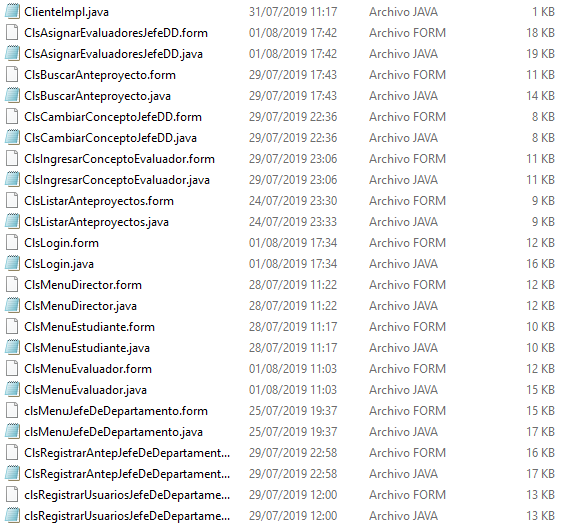


Ilustración 10. Archivos del directorio vistas.

Además dentro de este directorio se encuentra la clase ClienteImpl la cual implementa los métodos de la interfaz ClienteInt utilizados para realizar el proceso callback.

Src/ sop\_corba

Dentro de este directorio se encuentran archivos importantes para la ejecución del programa como, las interfaces InterfaceOpretaions las cuales tienen definidos los métodos remotos, también se encuentran los stub y los POA

# DESCRIPCIÓN DE LOS ARCHIVOS FUENTE (INTERFACES, ARCHIVOS FUENTE CLIENTE, ARCHIVOS FUENTE SERVIDOR)

## INTERFACES

La interfaz principal es la interfaz.idl la cual se ejecuta con el comando para generar los archivos base para la ejecución del programa.

Dentro de la interfaz idl se crearon 2 interfaces, GestionAnteproyectos y ClienteInt.

En el patrón DAO se crearon 4 interfaces, cada una con la responsabilidad asociada a su nombre

En el patrón **DAO** se utilizaron las siguientes interfaces: Ver Ilustración 11

En cada interfaz entán declarados los métodos relacionados a cada nombre de interfaz.

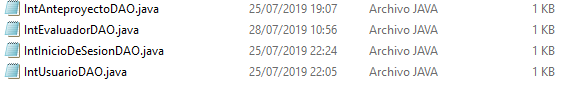


Ilustración 11. Interfaces utilizadas para el patrón DAO

La interfaz **intAnteproyectoDAO** tiene los siguientes métodos: Ver Ilustración 12. Todos los métodos están relacionados. Se utiliza esta interfaz para gestionar todo lo que tiene que ver con un anteproyecto.

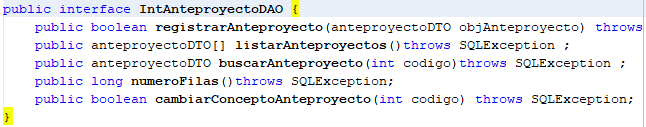


Ilustración 12. Métodos de la interfaz intAnteproyectoDAO

La interfaz **intEvaluadorDAO** tiene los siguientes métodos. Ver Ilustración 13. Es ta interfaz se utiliza para gestionar todo lo que tiene que ver con un evaluador.

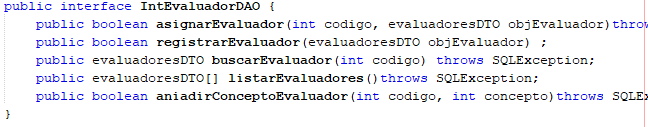


Ilustración 13. Métodos de la interfaz intEvaluadorDAO

La interfaz **intInicioSesionDAO** solo tiene un método, el método que se utiliza para verificar los credenciales de un usuario. Ver Ilustración 14.

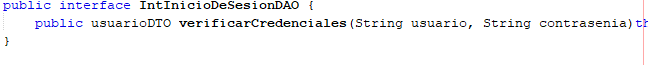


Ilustración 14. Método de la interfaz intInicioSesion

La interfaz **intUsuarioDAO** se utiliza para gestionar todo lo que tiene que ver con un usuario, tiene los siguientes métodos. Ver Ilustración 15

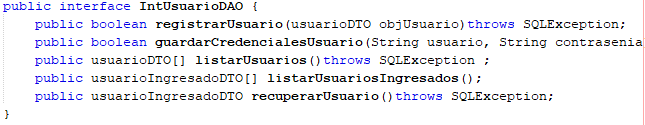


Ilustración 15. Métodos de la interfaz intUsuarioDAO

En el directorio **sop\_corba** seutilizaron las siguientes interfaces:

**GestionAnteproyectosOperations**

La interfaz GestionAnteproyectosOperations la cual tiene definidos los métodos remotos que serán implementados por la clase GestionAnteproyectosImpl y a su vez serán invocados por el cliente mediante el objeto remoto. Ver Ilustración 16

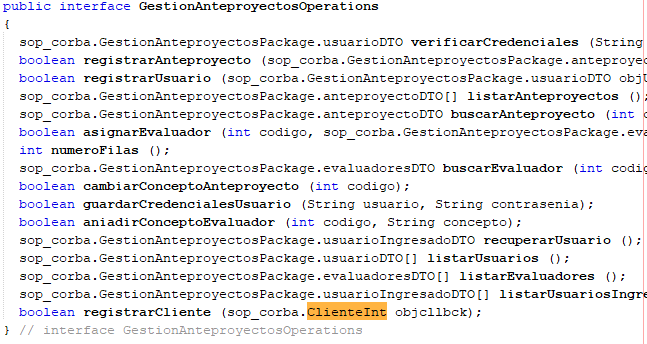


Ilustración 16. Métodos de la interfaz GestionAnteproyectosOperations

**ClienteIntOperations**

La interfaz **ClienteIntOperations** tiene definido el método que será utilizado para realizar el proceso callback Ver Ilustración 17.

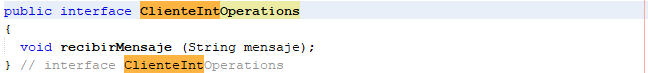


Ilustración 17. Métodos de la interfaz ClienteIntOperations

## ARCHIVOS FUENTE CLIENTE

CLIENTE DE OBJETOS

Hace referencia al cliente de objetos, el cuale tiene la responsabilidad de establecer la referencia remota hacia el servidor y servir como puente entre las interfaces gráficas y el servidor remoto, invocando los procedimientos remotos y entregando los valores retornados de vuelta a la interfaz gráfica para ser procesados visualmente.

VISTAS

Los archivos llamados <nombre>.form Son los archivos que contienen la especificación de los formularios gráficos con los cuales interactúa el usuario y la aplicación, existe un archivo de este tipo por cada interfaz gráfica definida para cada tipo de usuario (una interfaz por cada funcionalidad del usuario).

Los archivos llamados <nombre>.java son archivos de contienen la lógica determinada para cada formulario, la lógica que despliegan los botones, las validaciones de los campos y el punto de entrada mediante el cual se inicializan los formularios (existe un archivo de este tipo por cada .form)

## ARCHIVOS FUENTE SERVIDOR

UTILIDADES

El archivo llamdo utilidadesRegistroS.java hacen referencia al archivo que contiene la lógica para efectuar el registro de servidores en el NS, esta clase es genérica y es usada por los servidores de objetos los cuales la usan de acuerdo a sus necesidades.

SERVIDOR DE OBJETOS

El archivo llamado ServidorDeObjetos se encarga de registrar la referencia remota en el NS para que esta pueda ser instanciada por los clientes, si se tiene más de un servidor esta clase será la encargada de registrarlos todos asignando la ip el puerto y el nombre característico para cada uno de ellos.

DAO

En este directorio se encuentran las interfaces y las clases que implementan als interfaces que componen el patrón DAO, en estos archivos se realiza la implementación de los métodos de acceso a los datos, para las necesidades de cada usuario, se tiene una interfaz para cada grupo de acciones relacionadas.

DTO

En este directorio se encuentran las clases que sirven para tener objetos por medio de los cuales se transporta datos entre procesos.

# DESCRICIÓN DEL PROCESO DE COMPILACIÓN

1. Para realizar el proceso de compilación debe descargar el archivo lsd\_corba\_archivos\_fuente\_silvaMuñozA\_RamirezS.zip y descomprimirlo.
2. Dentro de la carpete src encontará un archivo llamado gestionanteproyectos.sql el cual se utiliza para crear la base de datos.
3. Para la creación de la base de datos siga los siguientes pasos:
   1. Abra el servidor local xampp e inicie el Apache y MySql como se muestra en la Ilustración 18.

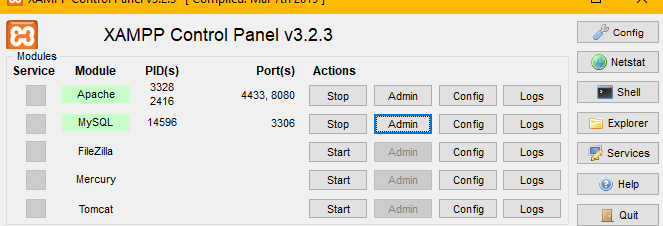


Ilustración 18. Ejecutar Apache y MySql.

* 1. Acceda a phpMyAdmin y diríjase a la parte de importar, en esta ventana aparecerá la opción para subir el archivo.sql y presionar continuar para realizar la importación como se muestra en la Ilustración 19.

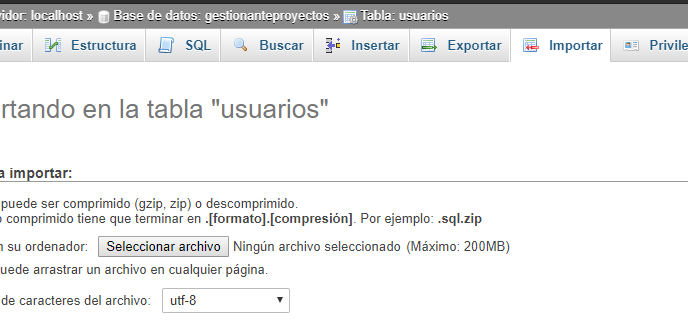


Ilustración 19. Importar archivo.sql

* 1. El nombre de la base de datos y el puerto que se está utilizando para acceder a MySql deben configurarse en la clase clsConexion que se encuentra en el paquete servidor/BD. Como se muestra en la Ilustración 20.

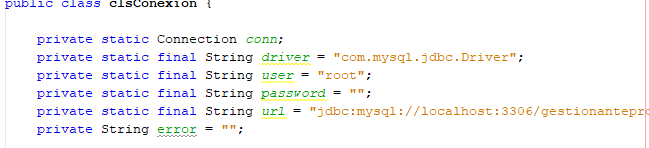


Ilustración 20. Configuración de la base de datos.

* 1. Ahora procedemos a importar el proyecto en el IDE netbeans. Se realiza yendo al botón Open Project que se encuentra en la parte superior izquierda como se muestra en la Ilustración 21.



Ilustración 21. Importar el proyecto en el IDE

* 1. Después de haber importado el proyecto en el IDE procedemos a compilar. Para compilar estas clases entonces se debe hacer click en el botón de clean build ubicado en la parte superior central: como se muestra en la Ilustración 22.



Ilustración 22. Proceso de compilación.

Si todo salió bien procedemos al proceso de ejecución

# DESCRIPCIÓN DEL PROCESO DE EJECUCIÓN

Antes de ejecutar el servidor, tenemos que lanzar el n\_s el cual se hace utilizando el archivo lanzarNS.bat que se encuentra dentro del directorio src, hacemos doble click sobre el archivo para poder lanzarlo y se nos abrirá una consola como la que se muestra en la Ilustración 23.

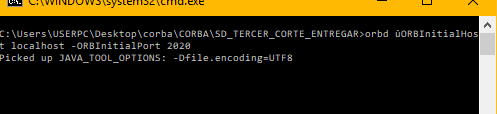


Ilustración 23. Lanzar N\_S

## EJECUCIÓN DEL SERVIDOR

Para la ejecución del servidor nos ubicamos en el directorio servidor y damos click derecho sobre la clase ServidorDeObjetos.java y seleccionamos la opción Run File, de esta manera el servidor de bjetos se ejecutará, como se muestra en la Ilustración 24.

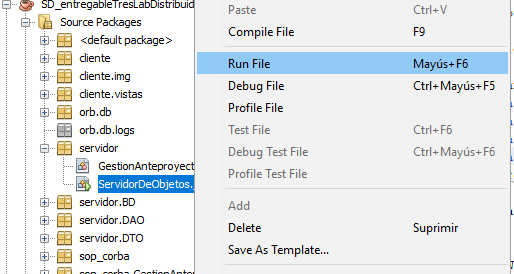


Ilustración 24. Ejecución del servidor

Si todo ha salido bien se mostrará un mensaje en la consola diciendo “servidor listo y esperando” como se muestra en la Ilustración 25.



Ilustración 25. Confirmación de que el servidor se ejecutó con éxito.

## EJECUCIÓN DEL CLIENTE

Para la ejecución del cliente nos ubicamos en el directorio cliente, el cual contiene la clase ClienteDeObjetos.java, sobre la cual deberá dar click derecho y seleccionar la opción Run File como se muestra en la Ilustración 26.

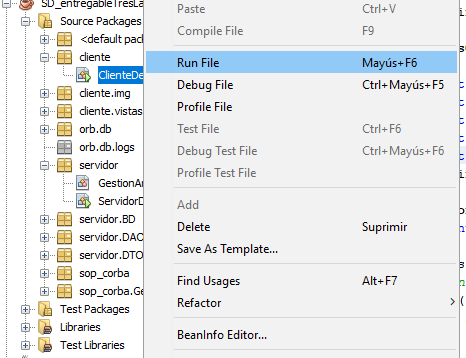


Ilustración 26. Ejecución del cliente

Si todo salió bien se le mostrará una ventana como la que se muestra en la Ilustración 27 en la cual usted podrá ingresar con un usuario y una contraseña para poder realizar la gestión de los anteproyectos.

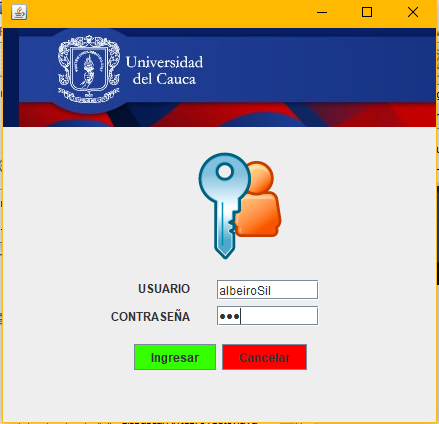


Ilustración 27. Confirmación de que el cliente se ejecutó con éxito

# DESCRIPCIÓN DEL MANEJO DEL INSTALADOR

En este caso, el software no posee un instalador. Viene con el código fuente y la instalación consiste en compilar el código y ejecutarlo, como se describió anteriormente. Queda a decisión del usuario donde guardar el código. En el caso de que quieran ejecutar el cliente y servidor en máquinas diferentes, cada una de estas deberá tener una copia del código para poder hacer el proceso de ejecución.