

MANUAL DE INSTALACIÓN



Universidad
del Cauca

Integrantes:

Leyder Alexander Inagan
Jeison José Ortiz

Profesor:

Ing. Daniel Eduardo Paz

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Programa Ingeniería de Sistemas
Curso: Laboratorio de Sistemas Distribuidos
Popayán, noviembre 2020

Tabla de contenido

1. Introducción.....	3
2. Requerimientos del sistema.....	4
2.1. Sistema Operativo:.....	4
2.2. Tecnología Utilizada:	4
2.3. Lenguaje de Programación:.....	4
3. Descripción de la Documentación Entregada.....	5
4. Descripción de la estructura de directorios y archivos de soporte.....	5
4.1. Estructura de los directorios.	5
4.2. Archivos de Soporte.....	9
5. Descripción de los archivos fuente	10
6. Descripción del proceso de Compilación.....	11
7. Descripción del Proceso de Ejecución	11
8. Descripción del manejo del instalador	14

Universidad
del Cauca

1. Introducción

En este proyecto se implementa de forma distribuida utilizando el modelo RMI de JAVA, siguiendo la arquitectura de cliente/servidor, se presenta un sistema para la empresa Chubby Burger, que permitirá a los clientes realizar sus pedidos desde las mesas sin la intervención de meseros u otro tipo de personal. La aplicación se restringe a los siguientes roles: un administrador que puede consultar las facturas de los pedidos, un cliente que realiza los pedidos, un servidor que atiende peticiones y un servidor que notifica a los cocineros que tienen que preparar una nueva orden.

El presente documento tiene como objetivo servir como manual para la instalación de la aplicación, para que garantice el correcto funcionamiento de esta. Para ello, se describen algunos requerimientos de software y los pasos a seguir para la compilación y ejecución de la aplicación. Además, se describen las funcionalidades principales de algunos de los archivos. Por otra parte, se resalta que el manual está orientado a personas que tengan conocimientos básicos en el sistema operativo Windows, en el modelo cliente/servidor y funcionamiento del modelo RMI de JAVA.

La estructura del documento se define en las siguientes partes, requerimientos del sistema, descripción de la documentación entregada, descripción de la estructura de directorios y archivos de soporte, descripción de los archivos fuente, descripción de los procesos de compilación, descripción del proceso de ejecución y descripción del manejo del instalador.

1. Requerimientos del sistema

Antes de instalar la aplicación, se deben tener en cuenta los requerimientos que se describen a continuación:

1.1. Sistema Operativo:

Esta aplicación no está ligada a un sistema operativo específico, ya que puede ser ejecutada en Windows, Linux y Mac OS. Es recomendable que el sistema operativo usado, este en su versión estable más reciente para reducir los riesgos a fallos.

1.2. Tecnología Utilizada:

Para el desarrollo de la aplicación se utilizó RMI de JAVA, el cual fue diseñado para implementarse con la arquitectura de cliente-servidor. Las aplicaciones desarrolladas con esta tecnología solo podrán ser desarrolladas con JAVA ya que RMI es nativo de Java, es decir, es una extensión al núcleo del lenguaje. RMI depende totalmente del núcleo de la Serialización de Objetos de Java, en este tipo de aplicaciones el servidor exporta un objeto, una vez hecho esto el objeto puede ser accedido desde la red, el programa siempre va a estar a la espera de una petición por medio de un puerto TCP. De esta forma un cliente puede conectarse e invocar los métodos proporcionados por el objeto previamente exportado.

1.3. Lenguaje de Programación:

Se utilizó el lenguaje de programación JAVA en el entorno de programación NetBeans para el desarrollo de la aplicación.

2. Descripción de la Documentación Entregada

La documentación de la aplicación se realizó en los siguientes manuales:

Manual de Instalación:

Contiene las instrucciones para la instalación de la aplicación. Lo anterior lo encontrara en este documento, además también se describe los requerimientos del sistema, estructura de directorios, archivos fuente y procesos de compilación y ejecución de la aplicación.

Manual Técnico:

Contiene el análisis y diseño, donde se realiza una detallada explicación de los requerimientos, historias de usuario, escenarios de usuario, diagramas de secuencia presentados para desarrollar la aplicación y arquitectura del sistema.

Manual de Usuario:

Contiene una guía clara y detallada de las instrucciones de como el usuario puede realizar un manejo correcto de la aplicación.

3. Descripción de la estructura de directorios y archivos de soporte.

3.1. Estructura de los directorios.

La aplicación cuenta con la siguiente estructura de directorios.

Los archivos fuente se encuentran distribuidos en la carpeta principal llamada:

lsd_rpc_archivos_fuente_InaganA_OrtizJ

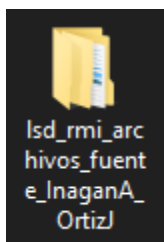


Figura 1: Directorio Principal

Dentro de esta carpeta se encuentra la carpeta del proyecto:

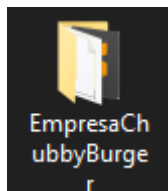


Figura 2: Carpeta del proyecto

Dentro de la carpeta del proyecto se encontraran las carpetas donde se almacenan los archivos necesarios para el funcionamiento adecuado de la aplicación:

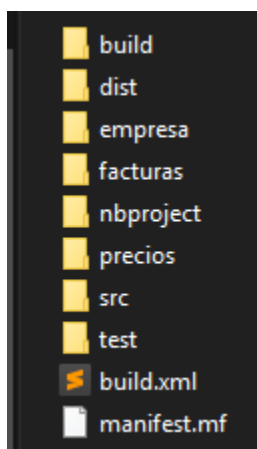


Figura 3: Subcarpetas Principales

Para este documento solo vamos a centrarnos en la carpeta src que es la que contiene las cuatro subcarpetas principales del código fuente y una carpeta adicional para almacenar los iconos usados en la aplicación:

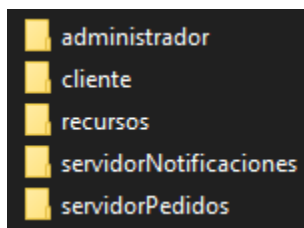


Figura 4: Subcarpetas Principales

Estas cuatro carpetas contienen los archivos necesarios para el funcionamiento de la aplicación, a continuación, mostraremos como están conformadas cada carpeta mostrada anteriormente:

- a. **Administrador:** Contiene los archivos que se encargan de manejar la lógica necesaria para el funcionamiento del administrador, el cual se encarga de consultar las facturas que han sido generadas en el servidor de pedidos.

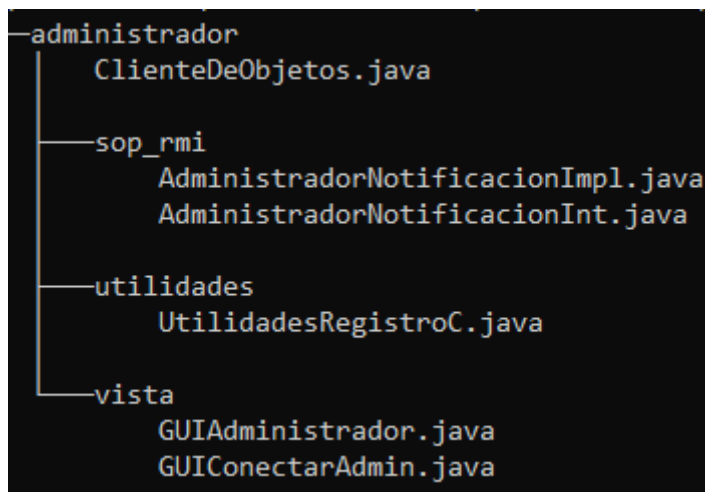


Figura 5: Archivos de la carpeta Administrador

- b. **Cliente:** Contiene los archivos que se encargan de la lógica del cliente, la cual es, que tiene a su disposición métodos u opciones como ver la información de la empresa y precios, comprar una hamburguesa, pagar su pedido, entre otras. Además, es el que realiza las peticiones al servidor de pedidos junto al Administrador.

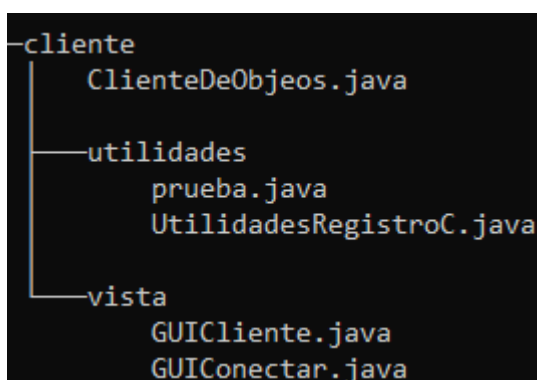


Figura 6: Archivos de la carpeta Usuario

- c. **ServidorPedidos:** Contiene los archivos que se encargan de la lógica del servidor que atiende las peticiones como: realizar pedido, mostrar información de la empresa, precios, notificar al administrador cuando se genere la factura de un pedido y notifica al servidor de notificaciones un nuevo pedido.

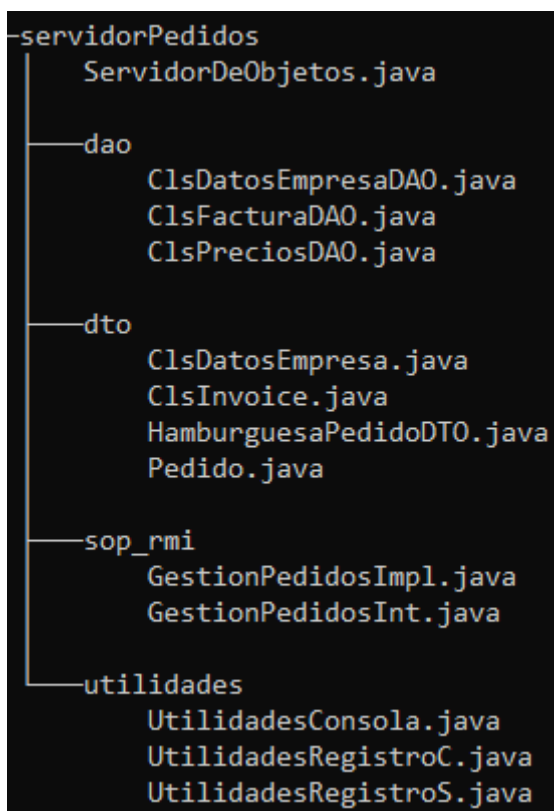


Figura 7: Archivos de la carpeta ServidorPedidos

- d. **ServidorNotificaciones:** Contiene los archivos que se encargan de la lógica del servidor de notificaciones que consiste en notificar a los cocineros cuando un pedido se realiza.

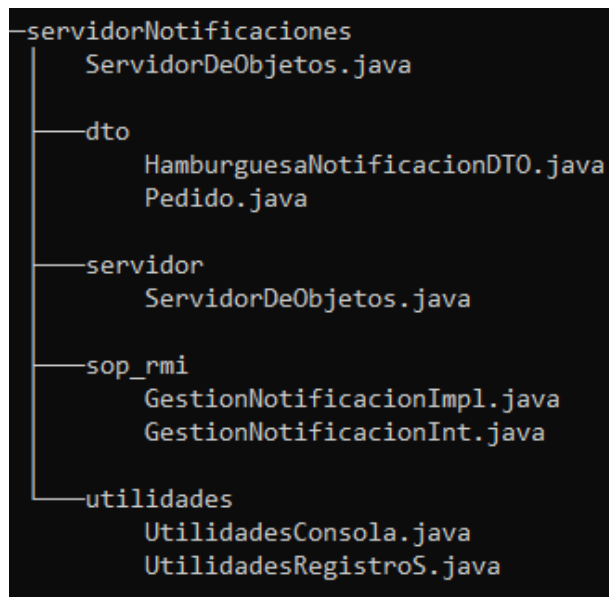


Figura 8: Archivos de la carpeta ServidorNotificaciones

Como se puede observar se cuenta con cuatro carpetas principales y una carpeta adicional: Usuario, ServidorPedidos, ServidorNotificaciones y Administrador. Además, se hace uso de tres interfaces que son las clases donde se definen los métodos que serán implementados para luego ser consumidos por el cliente o servidor, según sea el caso.

3.2. Archivos de Soporte

Dentro de la carpeta de ServidorPedidos se cuenta con dos archivos de soporte .txt, datosEmpresa.txt que contiene los datos de la empresa y datosHamburguesa.txt que contiene los datos de los precios que se encuentran en las carpetas llamadas empres y precios respectivamente.

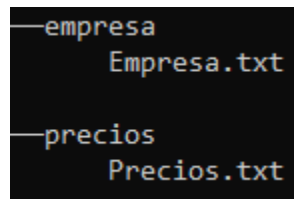


Figura 9. Archivos de Soporte.

Adicional se cuenta con una carpeta llamada “facturas” donde por cada pedido se almacenan las facturas en archivos .txt, que contienen la información del costo, el número de hamburguesas y el número de ingredientes extra usados en el pedido.



Figura 10: subcarpeta facturas.

4. Descripción de los archivos fuente

Como se puede ver en el ítem anterior, se tiene cinco carpetas, de las cuales una es el administrador, otra es el cliente (Usuario), otra es la carpeta de recursos y las otras dos son servidores (ServidorPedidos y ServidorNotificaciones). En el proyecto se hizo uso de tres interfaces remotas `GestionPedidosInt`, `GestionNotificacionInt` y `AdministradorNotificacionInt` donde se declararon los métodos remotos que van a ser implementados en las clases `GestionPedidosImpl`, `GestionNotificacionImpl` y `AdministradorNotificacionImpl` para luego ser invocados por el administrador, el cliente y el servidor de pedidos respectivamente.

- **Interface `GestionPedidosInt`:**

- Contiene el método `obtenerDatosEmpresa` que se encarga de obtener la información de la empresa del archivo correspondiente y retornar un objeto que contiene esa información.
- Contiene el método `obtenerPrecios` que se encarga de obtener la información de los precios del archivo correspondiente y retornar un objeto que contiene esa información.
- Contiene el método `registrarPedidoSistema` que se encarga de generar la factura con la información del pedido enviado desde el cliente, notificar al servidor de notificaciones y al administrador de la nueva factura generada.
- Contiene el método `consultarFacturas` que se encarga de consultar una factura de la carpeta “facturas” con el número de la factura que el administrador escoja y retorna un objeto con la información de la factura consultada.
- Contiene el método `registrarObjetoRemotoAdministradorNotificacion` que se encarga de agregar objetos remotos de tipo administrador en un `ArrayList`.

- **Interface `GestionNotificacionInt`:**

- Contiene el método `notificarRegistro` que se encarga de mostrar la notificación luego de que el servidor de pedidos haya generado la factura con la información del pedido que se envió desde el cliente.

- **Interface `AdministradorNotificacionInt`:**

- Contiene el método `mostrarNotificacion` que se encarga de mostrar la notificación luego de que el servidor de pedidos haya creado una nueva factura.

5. Descripción del proceso de Compilación

El proceso de compilación de la aplicación lo hace internamente el entorno de programación NetBeans cuando ejecutamos las diferentes partes del proyecto.

6. Descripción del Proceso de Ejecución

Para la ejecución de la aplicación es necesario primero ejecutar los servidores y por último el cliente, se recomienda seguir el orden siguiente.

El primer paso es abrir el entorno de programación, en este caso como ejemplo usaremos NetBeans, una vez tenemos abierta la aplicación:

1. **Abrir el proyecto:** Seleccionamos la opción “Open Project”.

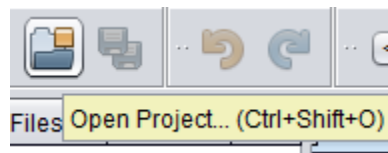


Figura 11: Abrir el proyecto.

2. **Seleccionamos el proyecto:** Con ayuda del explorador de archivos que genera NetBeans seleccionamos la carpeta que contiene el proyecto completo:

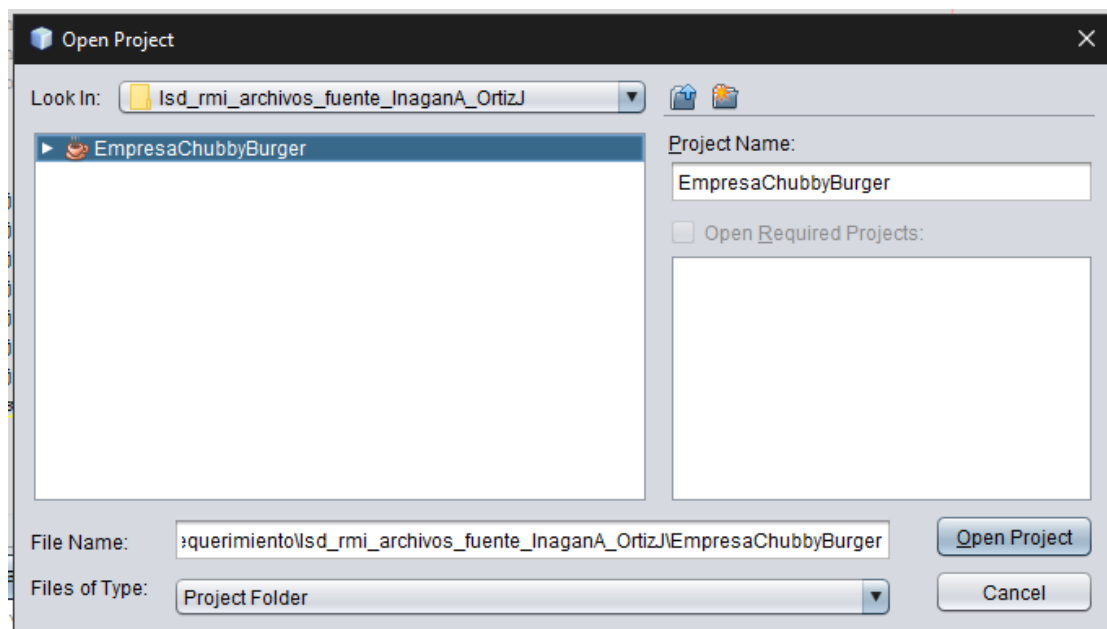


Figura 11: Seleccionamos la carpeta del proyecto.

3. **Ejecutar el servidor de notificaciones:** Para ejecutar el servidor nos debemos ubicar en el paquete “servidorNotificaciones” y hacer click derecho en el “ServidorDeObjetos.java” y seleccionar la opción “Run file”:

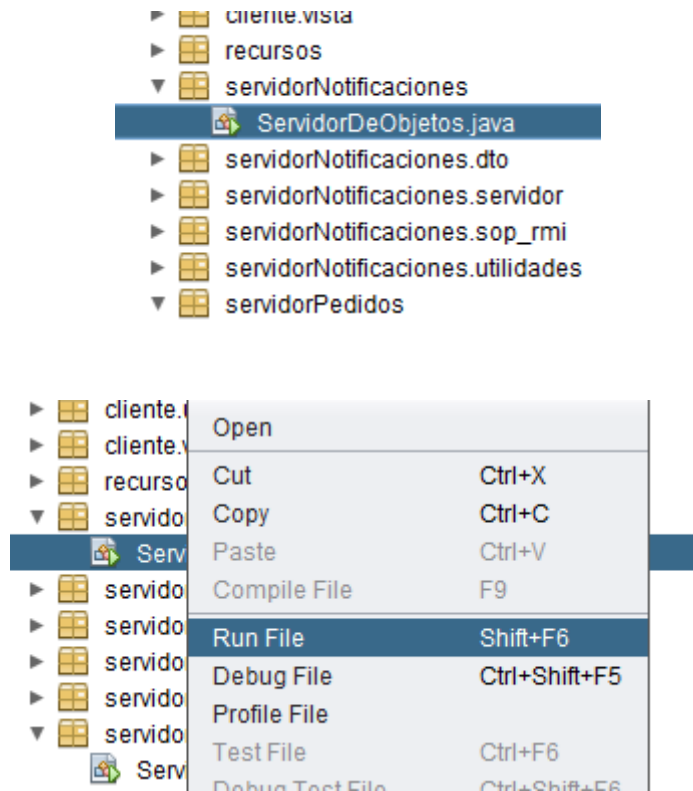
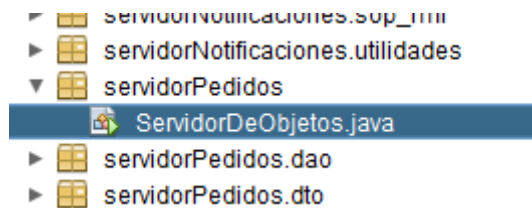


Figura 12: Selección y ejecución del servidor de notificaciones.

4. **Ejecutar el servidor de pedidos:** Para ejecutar el servidor nos debemos ubicar en el paquete “servidorPedidos” y hacer click derecho en el “ServidorDeObjetos.java” y seleccionar la opción “Run file”:



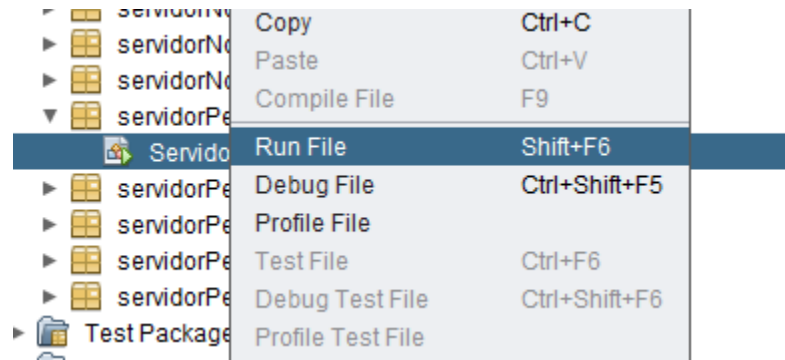


Figura 13: Selección y ejecución del servidor de pedidos.

5. **Ejecutar el administrador:** Para ejecutar el administrador nos debemos ubicar en el paquete de “administrador” y hacer click derecho en el “ClienteDeObjetos.java” y seleccionar la opción “Run file”:

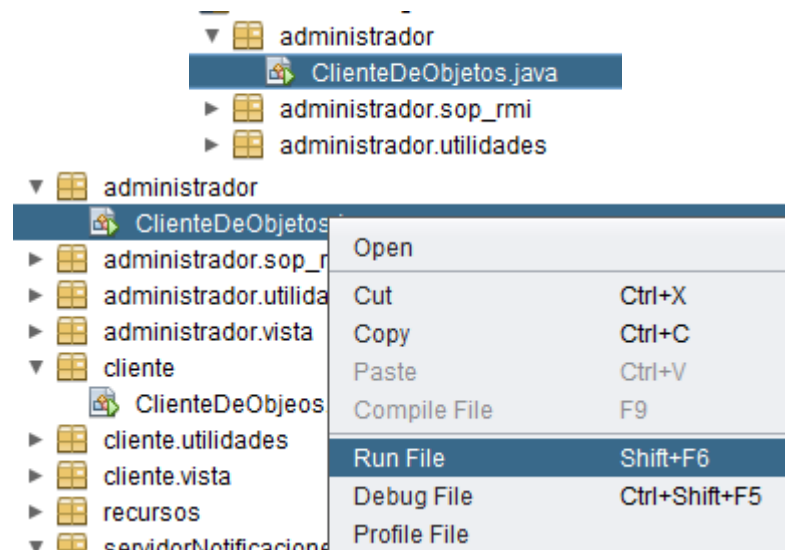
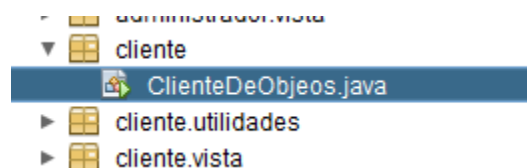


Figura 14: Selección y ejecución del administrador.

6. **Ejecutar el cliente:** Para ejecutar el cliente nos debemos ubicar en el paquete “cliente” y hacer click derecho en el “ClienteDeObjetos.java” y seleccionar la opción “Run file”:



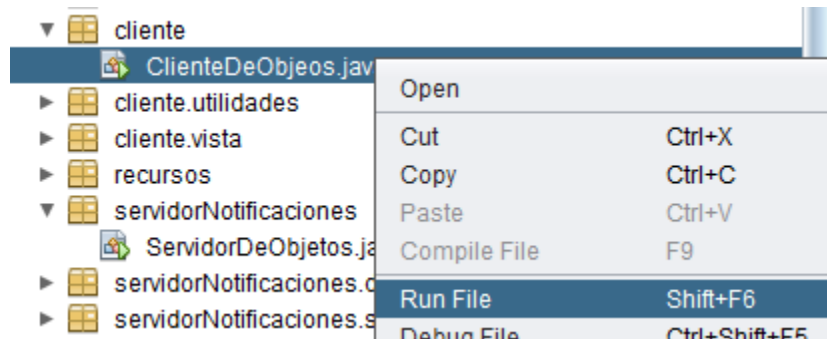


Figura 14: Selección y ejecución del.

7. Descripción del manejo del instalador

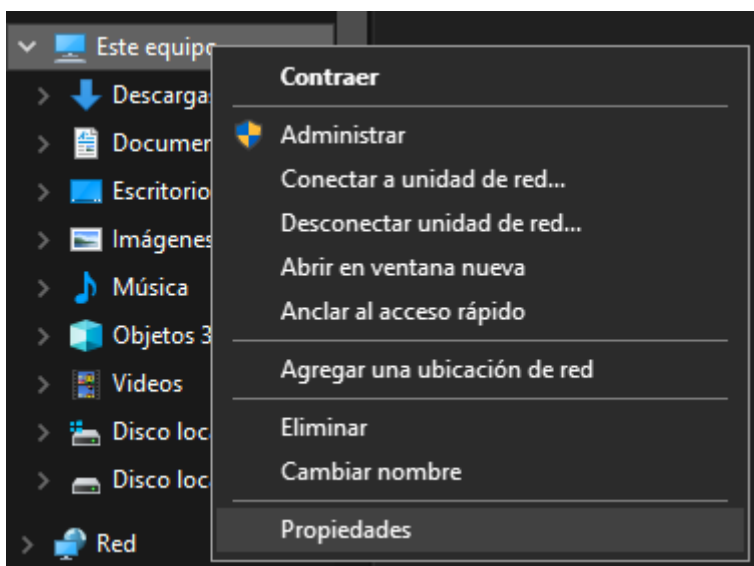
Para la instalación del entorno de programación usado en de ejemplo (NetBeans) debemos descargar primero el JDK desde la página oficial de JDK (<https://www.oracle.com/technetwork/es/java/javase/downloads/index.html>), luego se descarga el archivo correspondiente para el sistema operativo que se requiere, en este caso Windows:



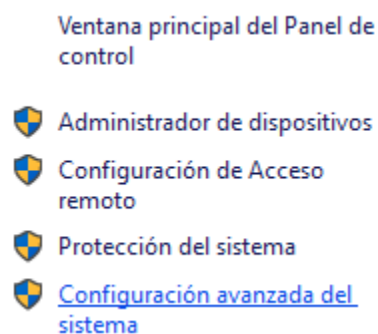
Java Platform (JDK) 8u111 / 8u112

Luego de haber instalado el JDK se debe agregar un nuevo campo en la variable de entorno Path, esto lo podemos hacer siguiendo los siguientes pasos:

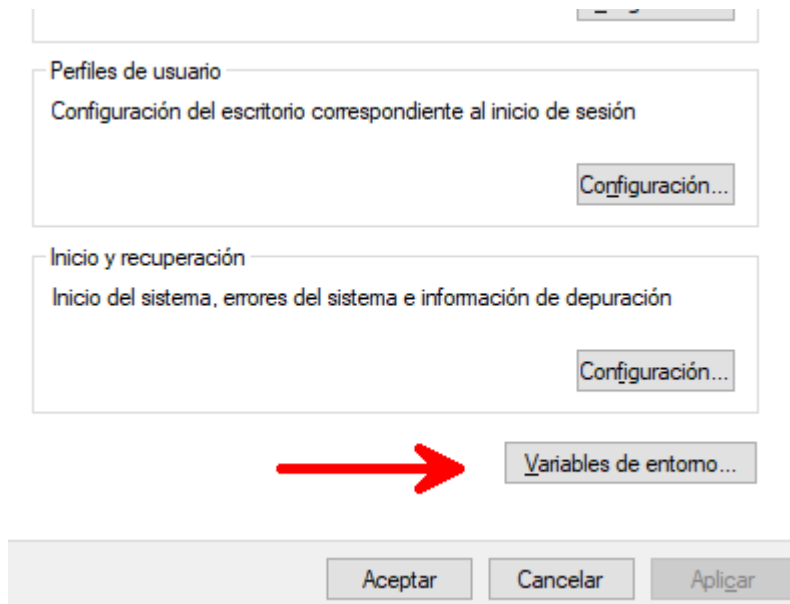
Primero abrimos un explorador de archivos presionando las teclas Win + E, luego en el árbol de archivos que se encuentra en la parte izquierda del explorador de archivos damos click derecho sobre el icono de “Este equipo” y seleccionamos propiedades:



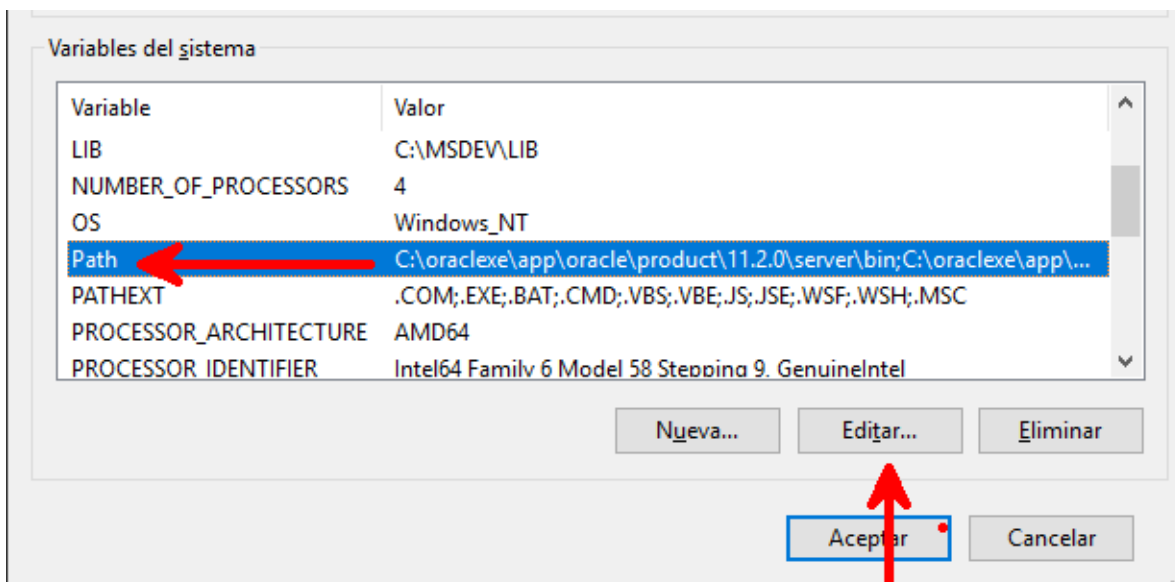
En la nueva ventana que se generó en la parte izquierda de la ventana seleccionamos la opción “Configuración avanzada del sistema”:



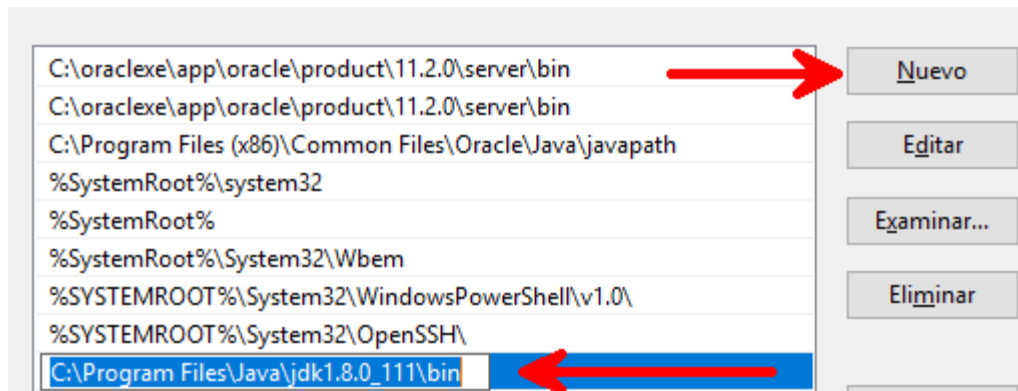
En la nueva ventana generada seleccionamos la opción “Variables de entorno” que se encuentra en la parte inferior izquierda de la nueva ventana:



En la nueva ventana generada vamos a sección de variables del sistema y buscamos la variable que se llama “Path”, y seleccionamos la opción “Editar...”:



En esta nueva ventana generada le damos en la opción “Nuevo” que aparece en la parte derecha de la ventana, y pegamos la url de la carpeta bin que se encuentra dentro de la carpeta donde se instaló el JDK, normalmente es “C:\Program Files\Java\jdk1.8.0_111\bin”



Luego de hacer esto, descargamos el instalador de NetBeans desde la página oficial de NetBeans (<https://netbeans.apache.org/download/index.html>) y descargamos el instalador en la versión 12 Apache, se ejecuta y se instala de forma normal.