 <b>Proyecto CUPi2</b>	<b>ISIS-1205 Algorítmica y Programación Manual del Entrenador</b>
Ejercicio:	Entrenador de Sockets
Autor:	Mario Sánchez
Fecha:	

## Objetivo

El objetivo de este entrenador es ayudar a:

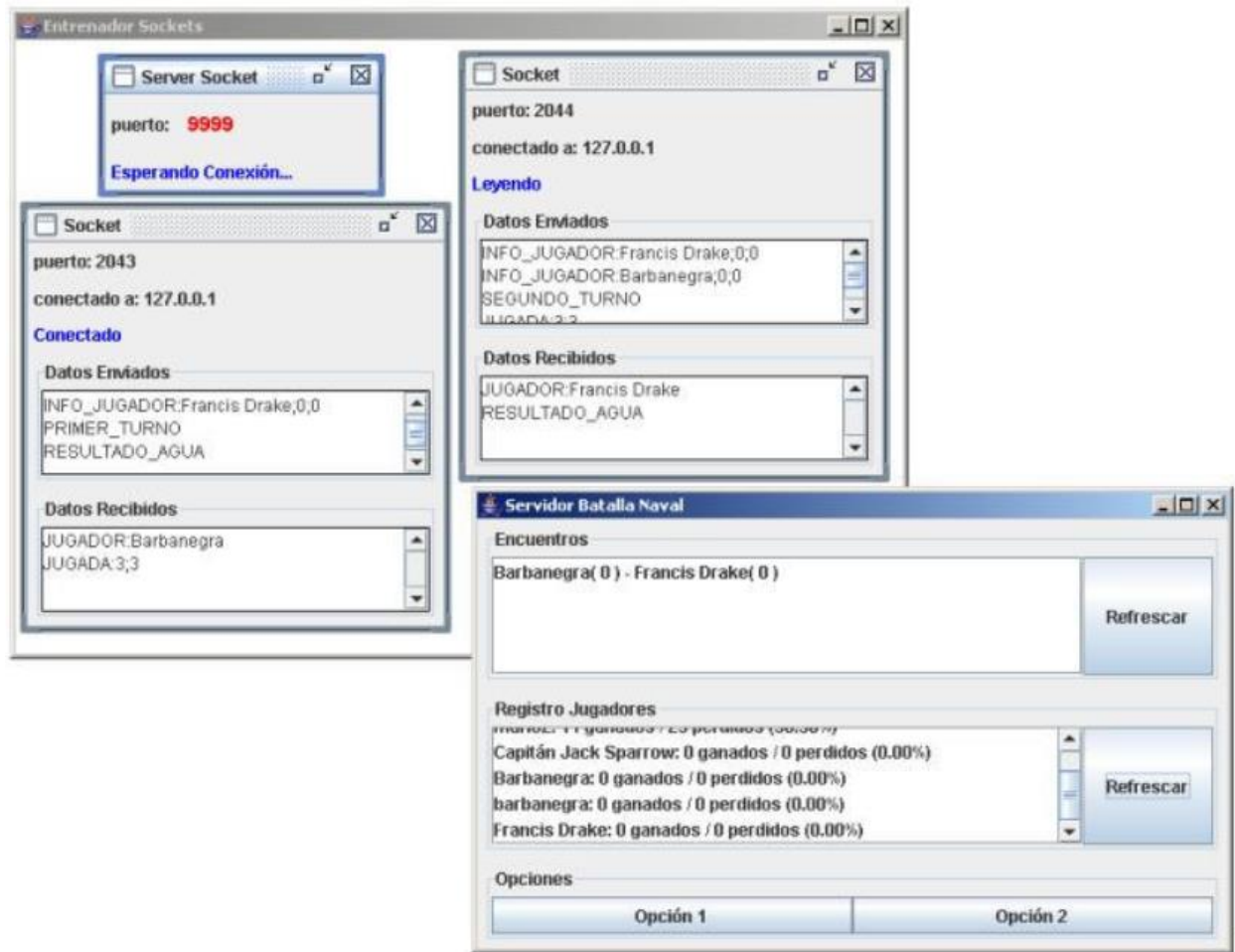
- Practicar las tareas básicas de conexión de sockets
- Practicar los algoritmos para enviar y recibir datos a través de sockets

Este entrenador puede utilizarse dentro de aplicaciones ya existentes o que estén siendo desarrolladas, usando una estrategia en la cual se usen las clases ofrecidas por el entrenador en lugar de las clases del framework de java (`java.net.Socket` y `java.net.ServerSocket`). De esta forma el usuario tendrá un comportamiento similar al de las clases de java que se verá enriquecido con una representación gráfica del estado de los sockets.

## Descripción

Este entrenador se utiliza desde adentro de una aplicación que está siendo implementada. Cuando esta se ejecute una ventana mostrará el estado de los Sockets y ServerSockets, al igual que la información que sea enviada o recibida por cada uno de los sockets.

En la imagen siguiente puede verse cómo, para el caso del servidor de Batalla Naval (`n12_batallaNaval`), se puede ver el estado del `ServerSocket`, cuales sockets han sido creados, su estado y cual información ha sido enviada y recibida por cada uno. Note que la ventana que está arriba a la derecha indica que en este momento se están esperando datos que lleguen a través de ese socket (se llamó un `readLine()` sobre el stream de entrada del socket).



## Instrucciones de Uso

1. Descargue el archivo `entrenadorSockets.jar` y cópielo a la carpeta `lib` de su proyecto.
2. Modifique las propiedades de su proyecto para que incluya el archivo `entrenadorSockets.jar` dentro de las librerías del proyecto. Para esto haga clic derecho sobre su proyecto y seleccione la opción *Propiedades* o *Properties*. En la opción llamada *Java Build Path* seleccione la pestaña que dice *Libraries*. Haga clic en el botón *Add Jar* y seleccione al archivo `entrenadorSockets.jar` que está dentro de su proyecto. Haga clic en *Ok* y cierre la ventana con las propiedades del proyecto.
3. Ubique en qué clases de su aplicación está usando objetos de las clases `Socket` o `ServerSocket`: modifique las líneas que se encuentran en la parte superior de esas clases y que se encargan de importar las clases usadas en ese archivo.
  - a. En lugar de `import java.net.Socket;` escriba `import uniandes.cupi2.entrenadorSockets.mundo.Socket;`
  - b. En lugar de `import java.net.ServerSocket;` escriba `import uniandes.cupi2.entrenadorSockets.mundo.ServerSocket;`
4. Salve los archivos modificados y ejecute su aplicación. Cuando ahora cargue o salve un archivo debería

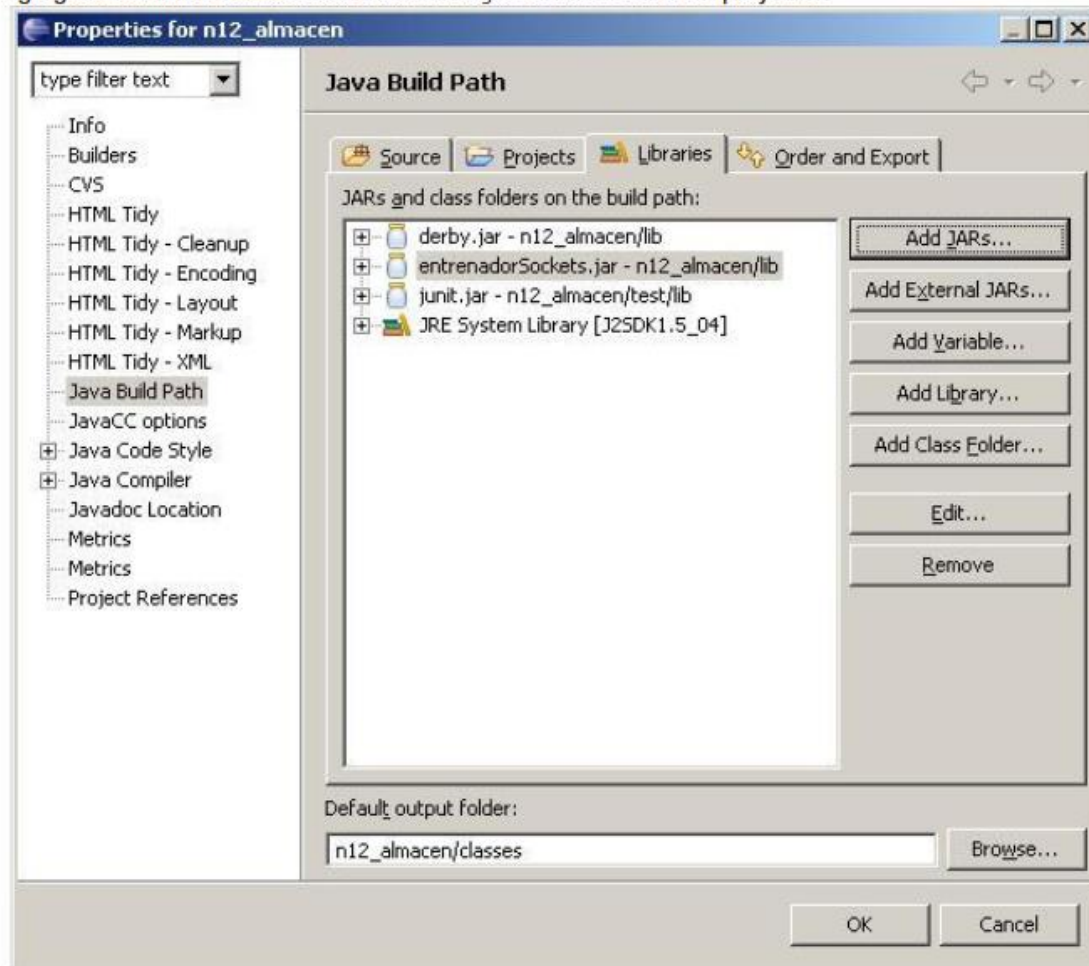
aparecer una ventana donde podrá ver el estado de los sockets.

5. Cuando ya no necesite la ventana de ayuda del entrenador, deberá modificar nuevamente los *imports* y devolverlos a su forma original.

### Ejemplo: Almacén

A continuación se describe el proceso completo para agregar el entrenador de sockets al ejemplo `n12_almacen`.

- En primer lugar descargue el ejemplo `n12_almacen` y el archivo `entrenadorSockets.jar` de la página de `cupi2` y copie este último archivo a la carpeta `lib` del ejemplo.
- Cargue el proyecto en Eclipse.
- Agregue el archivo `entrenadorSockets.jar` a las librerías del proyecto.



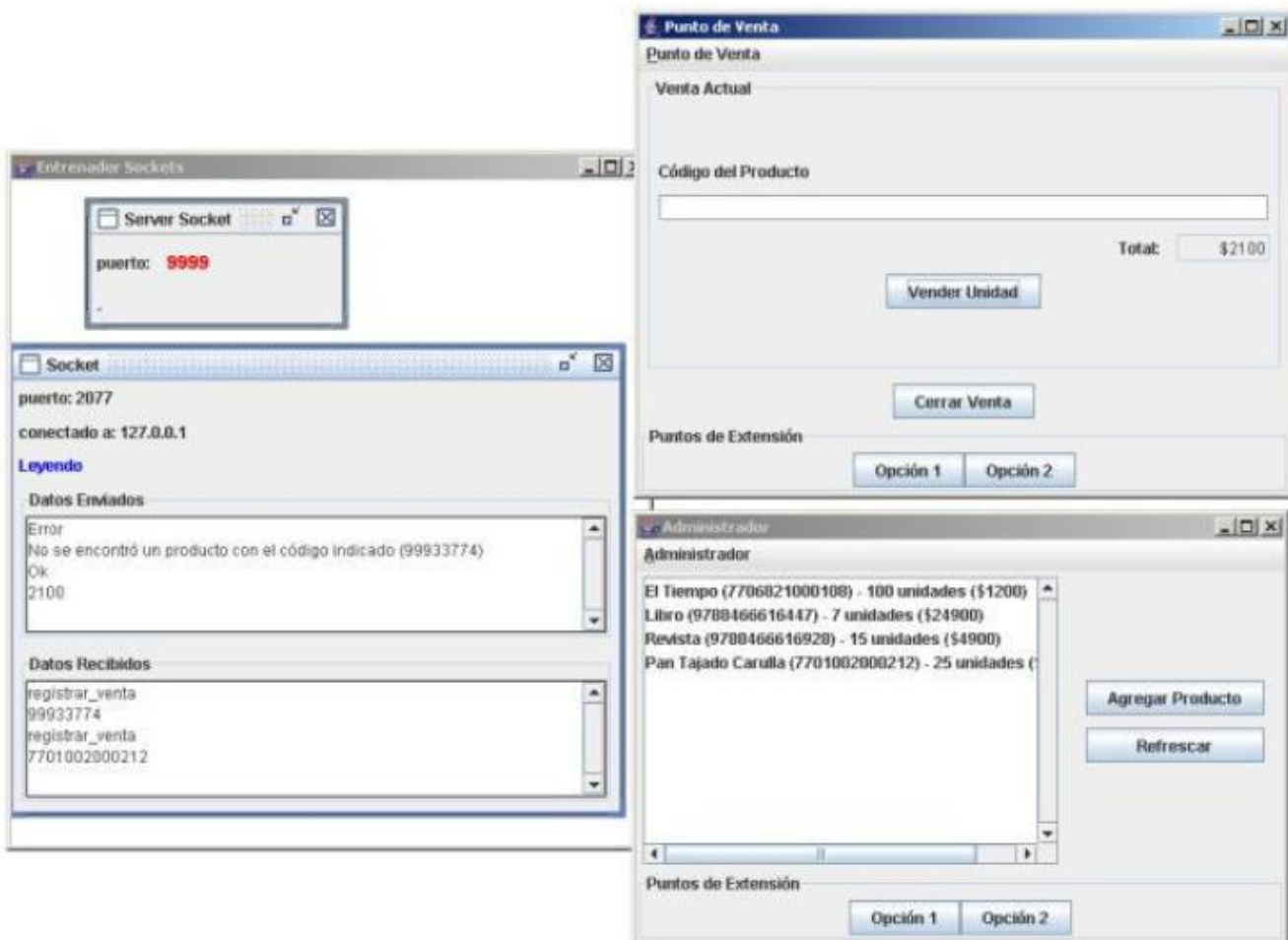
- Busque la clase `uniandes.cupi2.almacen.mundo.ServidorAlmacen`
- Modifique las líneas que importan las clases `Socket` y `ServerSocket`.  
En lugar de:  

```
import java.net.Socket;  
import java.net.ServerSocket;
```

  
Deben decir:

```
import uniandes.cup12.entrenadorSockets.mundo.Socket;  
import uniandes.cup12.entrenadorSockets.mundo.ServerSocket;
```

- Ejecute las aplicaciones del servidor del almacén y del punto de venta y vea cómo ahora puede ver el estado de los sockets que se van creando.



The screenshot displays three overlapping Java Swing windows from a project named 'cup12'.

- Entrenador Sockets:** Contains a sub-window titled 'Server Socket' with the text 'puerto: 9999'.
- Socket:** Shows connection details: 'puerto: 2077', 'conectado a: 127.0.0.1'. It includes a 'Leyendo' section with 'Datos Enviados' (Error, No se encontró un producto con el código indicado (99933774), Ok, 2100) and 'Datos Recibidos' (registrar\_venta, 99933774, registrar\_venta, 7701002000212).
- Punto de Venta:** A sales interface with 'Venta Actual', a 'Código del Producto' input field, a 'Total: \$2100' display, and buttons for 'Vender Unidad', 'Cerrar Venta', and 'Puntos de Extensión' (Opción 1, Opción 2).
- Administrador:** A product management window listing items: 'El Tiempo (7706821000108) - 100 unidades (\$1200)', 'Libro (9708466616447) - 7 unidades (\$24900)', 'Revista (9708466616928) - 15 unidades (\$4900)', and 'Pan Tajado Carulla (7701002000212) - 25 unidades (\$)'. It has 'Agregar Producto' and 'Refrescar' buttons, and 'Puntos de Extensión' (Opción 1, Opción 2) at the bottom.