

# **PRÁCTICA 2**

## **EJERCICIO 5**

### **Definición e implantación de un protocolo de aplicación**

## **ÍNDICE:**

### **1. DESCRIPCIÓN DE LA APLICACIÓN**

#### **1.1 FUNCIONALIDAD**

#### **1.2 ACTORES QUE INTERVIENEN**

### **2. DIAGRAMA DE ESTADOS DEL SERVIDOR**

### **3. MENSAJES QUE INTERVIENEN**

### **4. IMPLEMENTACIÓN**

# 1. DESCRIPCIÓN DE LA APLICACIÓN

Se trata de una aplicación cliente-servidor escrita en Java, utilizando sockets TCP. La finalidad de la misma es permitir a las empresas dar cumplimiento a la nueva normativa de control horario.

*El artículo 34.9 del Estatuto de los Trabajadores (modificado por el Real Decreto Ley 8/2019, de 8 de marzo, de medidas urgentes de protección social y de lucha contra la precariedad laboral en la jornada de trabajo) establece que la empresa garantizará el registro diario de la jornada de los trabajadores. Dicho registro deberá incluir el horario concreto de inicio y finalización de la jornada de trabajo de cada trabajador, sin perjuicio de la flexibilidad horaria que pudiera estar prevista.*

Así como permitir ‘trackear’ los recursos (medidos en tiempo de trabajo) que la empresa destina a distintos a los distintos proyectos de los distintos departamentos que tiene.

Tener conocimiento del tiempo que se destina a cada proyecto es de vital importancia para las empresas que podrán llevar una contabilidad correcta de los ingresos y gastos de cada proyecto, mejorando así la capacidad de respuesta ante situaciones indeseadas.

Este tipo de **programas** son **muy demandados hoy en día**, y se adaptan muy bien al **patrón Cliente-Servidor**. Por estos dos motivos hemos **decidido diseñar** este protocolo de **aplicación**.

## 1.1 FUNCIONALIDAD

**Desde el punto de vista del cliente, el protocolo de aplicación permitirá:**

- 1.) Identificarse como trabajador de la empresa o como administrador.
- 2.) Una vez identificado como trabajador de la empresa se podrá:
  - Consultar horas **propias** trabajadas desde que hay registro.
  - Seleccionar departamento en el que va a empezar a trabajar
  - Identificarse como administrador
- 2.BIS) Una vez identificado como administrador, además de las funcionalidades descritas en el punto 2.) (salvo identificarse como administrador) podrá:
  - Consultar horas **totales (de todos los trabajadores de la empresa)** trabajadas desde comienzo de jornada laboral actual, desde fecha, por intervalo de tiempo, por departamento, por proyecto.
  - Consultar horas **de un trabajador específico** trabajadas desde comienzo de jornada laboral actual, desde fecha y por intervalo de tiempo.
- 3.) Una vez que se ha seleccionado el departamento se puede seleccionar el proyecto en el que se va a trabajar.
- 4.) Una vez que se ha seleccionado tanto el departamento como el proyecto puede indicar al sistema que comienza a trabajar.
- 5.) Desde el estado trabajando puede:
  - Consultar horas **propias** trabajadas desde comienzo de jornada laboral actual y consultar horas desde que hay registros.
  - Elegir otro departamento
  - Elegir otro proyecto
  - Dejar de trabajar

## 1.2 ACTORES QUE INTERVIENEN

### **Servidor**

Consta únicamente de la función main. En ella, define el puerto (8989) y crea un ServerSocket.

En un bucle infinito crea un Socket y espera a que un cliente acepte la conexión.

Una vez un cliente la acepta, crea un objeto de tipo Procesador y le pasa al constructor el socket de cliente.

### **Procesador**

En el constructor recibe el socket del servicio que le pasa el servidor cuando lo crea. En el constructor también se inicializan las variables que simulan la base de datos del sistema (usuarios, contraseñas, departamentos, proyectos, horas trabajadas...). También se inicializan las variables que mantienen el estado del sistema (departamento actual, proyecto actual, usuario actual...).

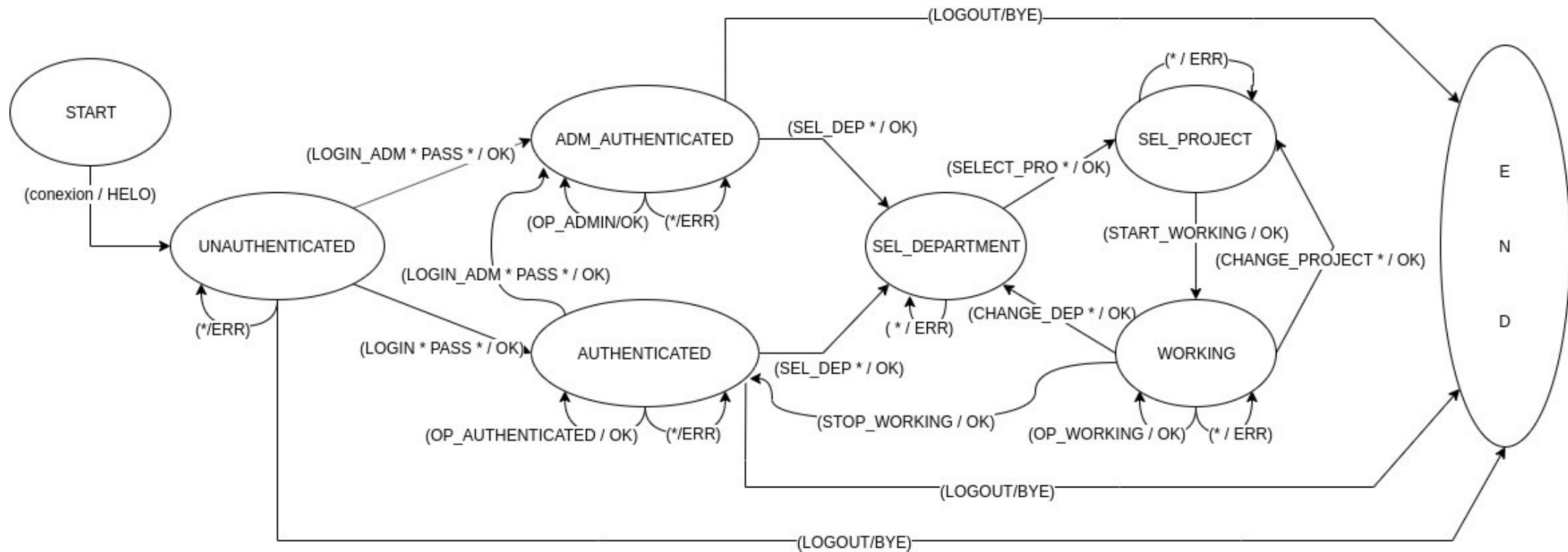
En la función procesa se comunica con el cliente, recibiendo y enviando paquetes dentro de un bucle while hasta que se recibe la petición de LOGOUT (102). El resto de peticiones están detalladas junto con sus códigos en la sección 3 Mensajes que intervienen.

### **Cliente**

Es el programa que utilizaran los trabajadores de la empresa, podrán ser administradores o no. En función de ello tendrán acceso a unas funciones extra o no.

Al iniciar se envía una petición al servidor para obtener un socket. Luego a través de este socket se comunica con el procesador del servidor dentro de un bucle do while hasta que el cliente envía la petición LOGOUT (102) que se desloguea y cierra el socket.

## 2. DIAGRAMA DE ESTADOS DEL SERVIDOR



### 3. MENSAJES QUE INTERVIENEN

C/S	CÓDIGO	CUERPO	DESCRIPCIÓN
Cliente	100	LOGIN + user + PASS + password	Este mensaje será enviado por el cliente para autenticarse con el servidor.
Servidor	200	OK	Mensaje de confirmación para la petición de autenticación.
Servidor	400	ERROR + "El usuario o password no existen"	Mensaje de error cuando el usuario no existe o su password no es correcta.
Servidor	401	ERROR + "Ya había un usuario logueado"	Mensaje de error cuando se intenta loguear habiéndose logueado previamente.
Cliente	101	LOGIN_ADM+ user + PASS + password	Este mensaje será enviado por el cliente para autenticarse con el servidor como administrador.
Servidor	201	OK	Mensaje de confirmación para la petición de autenticación como administrador.
Servidor	402	ERROR + "El usuario o password no existen"	Mensaje de error cuando el usuario no existe o su password no es correcta.
Servidor	403	ERROR + "El usuario no es admin"	Mensaje de error cuando el usuario no es admin
Cliente	102	LOGOUT	Este mensaje será enviado por el cliente para cortar la conexión con el servidor.
Servidor	202	BYE	Mensaje de confirmación para la petición de desconexión.
Cliente	103	CONSULT_TIME_WORKED	Este mensaje será enviado por el cliente para ver el tiempo que se ha trabajado desde que hay registro.
Servidor	203	TIME WORKED	Salida a la petición de tiempo trabajado.
Cliente	104	CONSULT_TIM_WOR + fecha_ini + fecha_fin	Este mensaje será enviado por el cliente para ver el tiempo que se ha trabajado desde y hasta una fecha concreta por todos los trabajadores.

Servidor	204	TIME WORKED	Salida a la petición de tiempo trabajado.
Cliente	105	CONSULT_TIM_WOR + fecha_ini + fecha_fin + dep	Este mensaje será enviado por el cliente para ver el tiempo que se ha trabajado desde y hasta una fecha concreta por todos los trabajadores, en un determinado departamento.
Servidor	205	TIME WORKED	Salida a la petición de tiempo trabajado.
Cliente	106	CONSULT_TIM_WOR + fecha_ini + fecha_fin + pro	Este mensaje será enviado por el cliente para ver el tiempo que se ha trabajado desde y hasta una fecha concreta por todos los trabajadores, en un determinado proyecto.
Servidor	206	TIME WORKED	Salida a la petición de tiempo trabajado.
Cliente	107	CONSULT_TIM_WOR + fecha_ini + fecha_fin + dni	Este mensaje será enviado por el cliente para ver el tiempo que se ha trabajado desde y hasta una fecha concreta para un trabajador en concreto.
Servidor	207	TIME WORKED	Salida a la petición de tiempo trabajado.
Servidor	404	ERROR + "El dni no es válido"	Mensaje de error cuando el dni no es correcto o el cliente no pertenece a la empresa.
Cliente	108	SEL_DEP + department	Este mensaje será enviado por el cliente para elegir departamento.
Servidor	208	OK	Mensaje de confirmación para la petición de elección de departamento.
Cliente	109	SEL_PRO + project	Este mensaje será enviado por el cliente para elegir proyecto.
Servidor	209	OK	Mensaje de confirmación para la petición de elección de proyecto.
Cliente	110	START_WORKING	Este mensaje será enviado por el cliente para empezar el contador.
Servidor	210	OK	Mensaje de confirmación para la petición de empezar a trabajar.
Cliente	111	CONSULT_TIME_WORKED _TODAY	Este mensaje será enviado por el cliente para ver el tiempo trabajado en la jornada actual.



Servidor	211	OK	Mensaje de confirmación para la petición de operación.
Cliente	112	CHANGE_PROJECT + project	Este mensaje será enviado por el cliente para cambiar de proyecto.
Servidor	212	OK	Mensaje de confirmación para la petición de cambio de proyecto.
Cliente	113	CHANGE_DEP + department	Este mensaje será enviado por el cliente para cambiar de departamento.
Servidor	213	OK	Mensaje de confirmación para la petición de cambio de departamento.
Cliente	114	STOP_WORKING	Este mensaje será enviado por el cliente para parar el contador de trabajo.
Servidor	214	OK	Mensaje de confirmación para la petición de paro de contador
<b>ERRORES COMUNES A VARIAS OPERACIONES:</b>			
Servidor	405	ERROR + "No se ha empezado a trabajar/Ya se estaba trabajando"	Mensaje de error para cuando no se ha comenzado a trabajar o ya estaba trabajando.
Servidor	406	ERROR + "Error de operación"	Para cualquier estado, mensaje de error para una operación inexistente o no válida para ese estado del cliente.
Servidor	407	ERROR + "operación solo válida para admins"	Mensaje de error cuando la operación es exclusiva para admins, y es un login normal el que realiza la petición.
Servidor	408	ERROR + "La fecha de inicio es posterior a la fecha de fin"	Mensaje de error cuando se han introducido mal las fechas.
Servidor	409	ERROR + "Error de proyecto"	Mensaje de error cuando el proyecto no existe, el cliente ya pertenece a dicho proyecto o el cliente no pertenece a ningún proyecto.
Servidor	410	ERROR + "Error de departamento"	Mensaje de error cuando el departamento no existe, el cliente ya pertenece a dicho departamento o el cliente no pertenece a ninguno.
Servidor	411	ERROR + "No se ha logueado"	Mensaje de error para operaciones que necesitan login previo.

## 4. IMPLEMENTACIÓN

Para ejecutar los códigos, hemos utilizado una extensión de Visual Studio Code llamada “java extension pack” que añade en el código un hipervínculo que ejecuta el programa al hacer click en él.

Adjunto junto a este PDF, se envía el código de la implementación de algunas partes del protocolo escrito en Java.

A continuación, mostraremos una traza completa de la ejecución sin errores en la que el cliente se loguea, elige departamento, elige proyecto, empieza a trabajar, hace un par de consultas, deja de trabajar y se desloguea, cerrando la conexión con el servidor. (Traza del servidor a la izquierda, traza del cliente a la derecha).

Como se puede ver en la captura, los códigos de peticiones y respuestas concuerdan con el protocolo que hemos propuesto (ver tabla anterior).

```
jose@jose-Vostro-5568:~/Dropbox/1cuatrimestre/FR/Practicas/p2/Ejercicio5/codigo$ /usr/lib/jvm/java-11-openjdk-amd64/bin/java -Dfile.encoding=UTF-8 -cp /home/jose/.config/Code/User/workspaceStorage/281bbe7cddb60e568424b414b4277dd0/redhat.java/jdt_ws/Dropbox_b6128385/bin Ejercicio5.ServidorControlHorario
Servidor creado
Servidor recibe petición de cliente desde el puerto 8989
Thread procesador de servido creado
Servidor atiende petición 100
Servidor atiende petición 103
Servidor atiende petición 108
Servidor atiende petición 109
Servidor atiende petición 110
Servidor atiende petición 111
Servidor atiende petición 103
Servidor atiende petición 114
Servidor atiende petición 103
Servidor atiende petición 102

jose@jose-Vostro-5568:~/Dropbox/1cuatrimestre/FR/Practicas/p2/Ejercicio5/codigo$ /usr/lib/jvm/java-11-openjdk-amd64/bin/java -Dfile.encoding=UTF-8 -cp /home/jose/.config/Code/User/workspaceStorage/281bbe7cddb60e568424b414b4277dd0/redhat.java/jdt_ws/Dropbox_b6128385/bin Ejercicio5.codigo
ClienteControlHorario
Cliente creado
Introduzca acción a realizar: login
Introduzca login: 76424266
Introduzca password: 123456
Cliente envía datos
Recibido: Código respuesta: 200 Respuesta: Logueado correctamente, bienvenido 76424266
Introduzca acción a realizar: consult_time_worked
Cliente envía datos
Recibido: Código respuesta: 203 Respuesta: LLeva trabajado desde que hay registros: 12 h y 0 min
Introduzca acción a realizar: sel_dep
Introduzca departamento: Equipo Manuel
Cliente envía datos
Recibido: Código respuesta: 208 Respuesta: AHORA PERTENECE AL DEP Equipo Manuel
Introduzca acción a realizar: sel_pro
Introduzca proyecto: Soporte
Cliente envía datos
Recibido: Código respuesta: 209 Respuesta: AHORA PERTENECE AL PRO Soporte
Introduzca acción a realizar: start_working
Cliente envía datos
Recibido: Código respuesta: 210 Respuesta: EMPIEZA EL CONTADOR DE TRABAJO
Introduzca acción a realizar: consult_time_worked_today
Cliente envía datos
Recibido: Código respuesta: 211 Respuesta: LLeva trabajado desde que empezo: 0 h y 5 min
Introduzca acción a realizar: consult_time_worked
Cliente envía datos
Recibido: Código respuesta: 203 Respuesta: LLeva trabajado desde que hay registros: 12 h y 5 min
Introduzca acción a realizar: stop_working
Cliente envía datos
Recibido: Código respuesta: 214 Respuesta: HAS DEJADO DE TRABAJAR, CONTADOR PARADO
Introduzca acción a realizar: consult_time_worked
Cliente envía datos
Recibido: Código respuesta: 203 Respuesta: LLeva trabajado desde que hay registros: 12 h y 5 min
Introduzca acción a realizar: logout
Cliente envía datos
Recibido: Código respuesta: 202 Respuesta: SE CERRARA LA CONEXION, HASTA MANANA
Cliente finalizado
jose@jose-Vostro-5568:~/Dropbox/1cuatrimestre/FR/Practicas/p2/Ejercicio5/codigo$ |
```

La siguiente captura muestra otra traza pero mostrando errores ya que se hace un uso inadecuado de la aplicación. En este caso veremos los códigos de errores.

```
jose@jose-Vostro-5568:~/Dropbox/1cuatrimestre/FR/Practicas/p2/Ejercicio5/codigo$ /usr/lib/jvm/java-11-openjdk-amd64/bin/java -Dfile.encoding=UTF-8 -cp /home/jose/.config/Code/User/workspaceStorage/281bbe7cddb60e568424b414b4277dd0/redhat.java/jdt_ws/Dropbox_b6128385/bin Ejercicio5.codigo.ServidorControlHorario
Servidor creado
Servidor recibe petición de cliente desde el puerto 8989
Thread procesador de servido creado
Servidor atiende petición 103
Servidor atiende petición 406
Servidor atiende petición 100
Servidor atiende petición 110
Servidor atiende petición 109
Servidor atiende petición 406
Servidor atiende petición 108
Servidor atiende petición 110
Servidor atiende petición 114
Servidor atiende petición 406
Servidor atiende petición 102

jose@jose-Vostro-5568:~/Dropbox/1cuatrimestre/FR/Practicas/p2/Ejercicio5/codigo$ /usr/lib/jvm/java-11-openjdk-amd64/bin/java -Dfile.encoding=UTF-8 -cp /home/jose/.config/Code/User/workspaceStorage/281bbe7cddb60e568424b414b4277dd0/redhat.java/jdt_ws/Dropbox_b6128385/bin Ejercicio5.codigo.Cli
ClienteControlHorario
Cliente creado
Introduzca acción a realizar: consult_time_worked
Cliente envía datos
Recibido: Código respuesta: 411 Respuesta: NO SE HA LOGUEADO
Introduzca acción a realizar: hola
Cliente envía datos
Recibido: Código respuesta: 406 Respuesta: ERROR, OPERACION DESCONOCIDA
Introduzca acción a realizar: login
Introduzca login: 17482989
Introduzca password: 123456
Cliente envía datos
Recibido: Código respuesta: 200 Respuesta: Logueado correctamente, bienvenido 17482989
Introduzca acción a realizar: start_working
Cliente envía datos
Recibido: Código respuesta: 410 Respuesta: ERROR, NO SE PERTENECE A UN DEPARTAMENTO
Introduzca acción a realizar: sel_pro
Introduzca proyecto: Equipo Jose
Cliente envía datos
Recibido: Código respuesta: 410 Respuesta: ERROR, NO SE PERTENECE A UN DEPARTAMENTO
Introduzca acción a realizar: consult_worked_time_day
Cliente envía datos
Recibido: Código respuesta: 406 Respuesta: ERROR, OPERACION DESCONOCIDA
Introduzca acción a realizar: sel_dep
Introduzca departamento: Equipo Jose
Cliente envía datos
Recibido: Código respuesta: 208 Respuesta: AHORA PERTENECE AL DEP Equipo Jose
Introduzca acción a realizar: start_working
Cliente envía datos
Recibido: Código respuesta: 409 Respuesta: ERROR, NO SE PERTENECE A UN PROYECTO
Introduzca acción a realizar: stop_working
Cliente envía datos
Recibido: Código respuesta: 405 Respuesta: ERROR, NO SE ESTABA TRABAJANDO
Introduzca acción a realizar: adios
Cliente envía datos
Recibido: Código respuesta: 406 Respuesta: ERROR, OPERACION DESCONOCIDA
Introduzca acción a realizar: logout
Cliente envía datos
Recibido: Código respuesta: 202 Respuesta: SE CERRARÁ LA CONEXIÓN, HASTA MANANA
Cliente finalizado
jose@jose-Vostro-5568:~/Dropbox/1cuatrimestre/FR/Practicas/p2/Ejercicio5/codigo$ |
```