

GOI ESKOLA
POLITEKNIKO

ESCUELA
POLITÉCNICA
SUPERIOR



CLÍNICA DE REHABILITACIÓN

2.CURSO DE GRADO DE INGENIERÍA INFORMÁTICA
PROYECTO PBL



AUTORES: Aitor Landa, Ane Sajeras, Josu Garralda, Ibai Rodriguez, Oihane
Lameirinhas

Tutor: Juan Carlos Mugarza

En Arrasate, a 18 de enero de 2019

LABURPENA

Dokumentu honek “REmotion” proiektua nola egin den azaltzen du. Eskuetako errehabilitazioetan lan egiten duten profesionalentzako bideratuta. Hau garatzeko eman dizkiguten baldintzak JAVA, VHDL, Packet Tracer eta Sistema Eragile ezberdinak erabiltzea izan da. Hasiera batean proiektuaren ideia eta planteamendua adostu da. Ondoren beharrezko atazak definitu eta zereginak banatu dira proiektuaren helburura iristeko.

RESUMEN

Este documento explica cómo se ha realizado el proyecto “REmotion”. Enfocado para el uso de expertos que trabajan con la rehabilitación de manos. La condición que se ha impuesto para el desarrollo de dicho proyecto ha sido la de utilizar Java, VHDL, Packet Tracer y Sistemas Operativos varios como Windows y Linux. En un principio se ha acordado el planteamiento y la idea del proyecto. Después se han definido y repartido las tareas necesarias para llevarlo a cabo.

SUMMARY

This document explains how the "REmotion" project has been carried out. Focused for the use of experts who work with hand rehabilitation. The condition that has been imposed for the development of this project has been the use of Java, VHDL, Packet Tracer and various Operating Systems such as Windows and Linux. Initially, he has agreed on the approach and the idea of the project. Then they defined and distributed the necessary tasks to carry it out.

ÍNDICE

1. INTRODUCCION	1
2. PLANTEAMIENTO DEL PROBLEMA	2
2.1. EXAMEN DEL PROBLEMA	2
2.2. ESPECIFICACIÓN DE LOS OBJETIVOS	2
2.3. PLANIFICACIÓN.....	2
3. DISEÑO DE LA SOLUCIÓN	3
4. DESARROLLO Y RESULTADO	5
4.1. PROGRAMACION AVANZADA	5
DIAGRAMA PROGRAMACIÓN AVANZADA [1]	5
HERENCIA JAVA [2].....	6
4.1.1 PACIENTE.....	6
EJERCICIOS [3].....	7
EJERCICIOS LEAP MOTION [4]	7
RESULTADO DE EJERCICIOS [5].....	8
4.2.3 ADMINISTRADOR.....	8
ADMINISTRADOR [6]	9
4.1.3 FISIOTERAPEUTA.....	9
FISIOTERAPEUTA- TABLA [7].....	10
FISIOTERAPEUTA- GRÁFICO [8].....	10
4.1.4 LEAP MOTION	10
LEAP MOTION [9].....	11
4.2. SISTEMAS LÓGICOS PROGRAMABLES	11
SISTEMAS LÓGICOS PROGRAMABLES [10].....	12
BASYS3 [11]	12

4.2.3 COMPONENTE REHABILITACIÓN	12
4.2.2 COMPONENTE UART	13
UART [12]	13
4.2.3 COMPONENTE PRINCIPAL MAIN	13
COMPONENTE PRINCIPAL MAIN [13].....	14
4.3. REDES DE COMUNICACIONES.....	14
REDES [14].....	15
4.3.1 RED CLÍNICA.....	15
RED CLÍNICA [15]	16
4.3.2 RED CLÍNICA.....	16
RED CLÍNICA [16]	16
4.3.3 RED ISP	17
RED ISP [17].....	17
4.4.- INFRAESTRUCTURAS Y SISTEMAS	17
TOPOLOGÍA SERVIDORES [18].....	17
4.4.1.- Windows Server	18
4.4.2.- Ubuntu Server	19
5. CONCLUSIONES Y LÍNEAS FUTURAS.....	21
6. BIBLIOGRAFÍA	22
7. APÉNDICE.....	23
Copias de Seguridad.....	23
Diagrama de Gantt.....	28

TABLA DE LAS IMÁGENES

DIAGRAMA PROGRAMACIÓN AVANZADA [1]	5
HERENCIA JAVA [2]	6
EJERCICIOS [3]	7
EJERCICIOS LEAP MOTION [4]	7
RESULTADO DE EJERCICIOS [5]	8
ADMINISTRADOR [6]	9
FISIOTERAPEUTA- TABLA [7]	10
FISIOTERAPEUTA- GRÁFICO [8]	10
LEAP MOTION [9]	11
SISTEMAS LÓGICOS PROGRAMABLES [10]	12
BASYS3 [11]	12
UART [12]	13
COMPONENTE PRINCIPAL MAIN [13]	14
REDES [14]	15
RED CLÍNICA [15]	16

1. INTRODUCCIÓN

El objetivo de este proyecto es crear una aplicación que permita gestionar un proceso de rehabilitación y mejorar la calidad del servicio que los fisioterapeutas/médicos ofrecen a los pacientes con diferentes lesiones en las manos. Para ello se ha hecho uso de una Leap Motion para monitorizar los movimientos de la mano y registrarlos en una aplicación desarrollada en Java. También se ha utilizado una placa Basys3 con código VHDL para gestionar un temporizador de ejercicios. Además mediante Packet Tracer se ha realizado una simulación de redes de las conexiones de comunicación entre la sede informática y la clínica. Finalmente en la sede informática se han configurado servidores Windows Server y Ubuntu Server para realizar copias de seguridad de los ficheros de la aplicación Java entre otros.

Esta memoria está dividida en cinco partes. Tras la introducción, se explica el objetivo de este proyecto. En el segundo apartado, se plantea el problema a resolver y el diseño de la solución. En el tercer apartado en cambio, se diseña la solución y en el cuarto, el desarrollo de las diferentes partes como Java, Servidores o VHDL. A continuación, aparecen las conclusiones y las líneas futuras. Al final, están la bibliografía y el apéndice, en el que se encuentra un diagrama de gantt con información detallada de las partes del proyecto desarrolladas y también el código implementado de "REmotion".

A continuación, se expone el procedimiento que se ha seguido desde el planteamiento del problema hasta el diseño de la solución para su posterior desarrollo y resultado.

2. PLANTEAMIENTO DEL PROBLEMA

En este apartado se detalla el problema y se mencionan los puntos necesarios para llegar al objetivo.

2.1. EXAMEN DEL PROBLEMA

La mayoría de las personas sufren roturas, torceduras, esguinces, luxaciones... en manos, muñecas o dedos en algún momento de su vida. Algunos se solucionan con reposo pero muchos de ellos necesitan de rehabilitación y a veces de cirugía para su completa recuperación.

En la sanidad pública la lista de espera para rehabilitación es de semanas y por ello se ha querido dar solución a los casos de lesiones en la mano con el objetivo de ayudar por un lado a los pacientes con un tratamiento eficiente y semiautomático, y por otro al médico/fisioterapeuta ayudándole a reducir el tiempo de trabajo y poder atender a otros pacientes..

2.2. ESPECIFICACIÓN DE LOS OBJETIVOS

El objetivo es crear una aplicación que sea capaz de gestionar de manera eficaz la lesión del paciente. Para ello:

Se debe de poder almacenar toda la información del paciente, como son:

- Datos personales
- Lesión
- Estadísticas
- Resultados

El fisioterapeuta deberá elegir ejercicios específicos para el paciente.

El fisioterapeuta añadirá tiempo en cada ejercicio.

El fisioterapeuta visualizará estadísticas de ejercicios para ver el progreso del paciente.

2.3. PLANIFICACIÓN

Para poder llevar a cabo este proyecto se han dividido las tareas por personas y orden ya que algunas tareas dependen de otras.

Se ha comenzado con la base de cada apartado:

Topología de la red, realizar un ejercicio y registrar correctamente los resultados, realizar un temporizador que funcione independientemente con la Basys3, creación de un dominio en el servidor principal y activación de servicios básicos.

Desde esta base se ha expandido el trabajo con la intención de agregar nuevas funcionalidades necesarias para poder unir al final todos los componentes:

Ejercicios ligados a pacientes supervisados por fisioterapeutas, visualización de los resultados en gráficas y tablas, unión del temporizador con el módulo UART para su posterior conexión con la aplicación de Java mediante puerto serie, adición de un servidor de soporte para algunos servicios y para la realización de copias de seguridad de los ficheros de la clínica.

Finalmente se han unido los diferente bloques y se han adaptado para su correcto funcionamiento.

Para ver el diagrama de Gantt que muestra la realización de las tareas cronológicamente, ver el apéndice.

3. DISEÑO DE LA SOLUCIÓN

Cada paciente tendrá asignados una serie de ejercicios por su fisioterapeuta. Estos están pensados para que se realicen en el día asignado, en el caso de no realizarlos quedarán sin resultado. Con esta medida se piensa que el paciente estará más motivado a seguir adelante con la rehabilitación.

Los ejercicios se basan en repetir un movimiento preciso tantas veces como el fisioterapeuta decida dentro de un tiempo estipulado. Cada ejercicio está compuesto por 3 registros: movimiento bien hecho, movimiento casi hecho y movimiento no hecho.

Los resultados se pueden visualizar con dos criterios: diario y mensual. En ambos, se puede observar la cantidad de ejercicios bien y casi hechos ante un objetivo definido por el fisioterapeuta.

Las copias de seguridad de todos estos datos se realiza de lunes a viernes a las 23:00 de la noche y se guardan en un servidor de la sede informática. Así en caso de pérdida de datos se podrán recuperar de forma fácil y rápida.

4. DESARROLLO Y RESULTADO

En este apartado se podrán ver las diferentes tareas relacionadas con sus respectivas asignaturas.

4.1. PROGRAMACION AVANZADA

En la parte del proyecto referida a Java se ha utilizado Eclipse como entorno de desarrollo. Se ha llevado a cabo una aplicación con el fin de establecer una interfaz mediante la cual tanto el paciente como los fisioterapeutas pueden interactuar con ella. La aplicación cuenta con un login principal donde el usuario inicia sesión. Hay tres tipos de usuarios: pacientes, fisioterapeutas y administradores y según el tipo de usuario se abrirá una ventana u otra.

Para llevar a cabo esta aplicación en Java ha sido indispensable el uso de una Leap Motion, de la que se hablará en los siguientes puntos en profundidad.

En el diagrama de casos que se encuentra bajo estas palabras se muestra las diferentes acciones que puede hacer cada tipo de usuario.

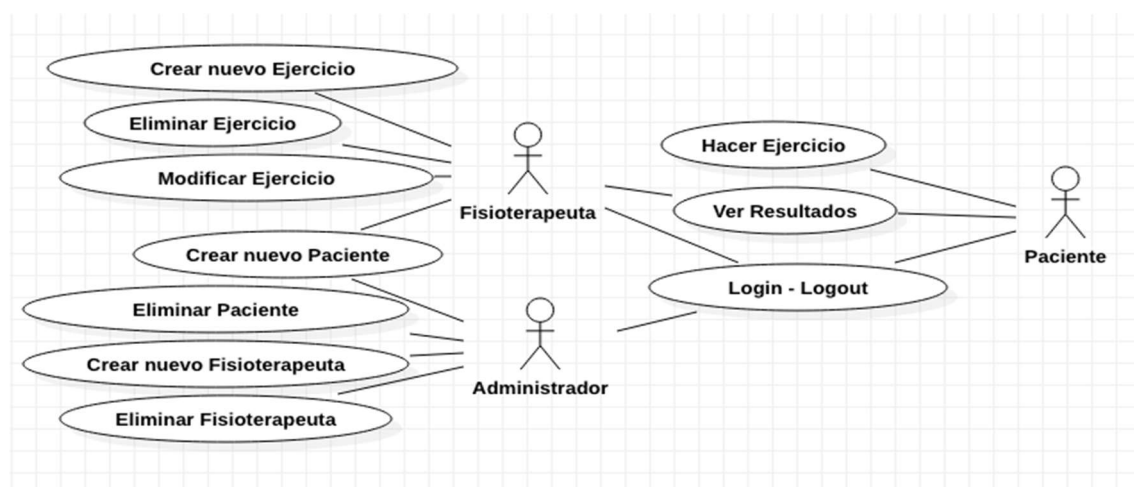
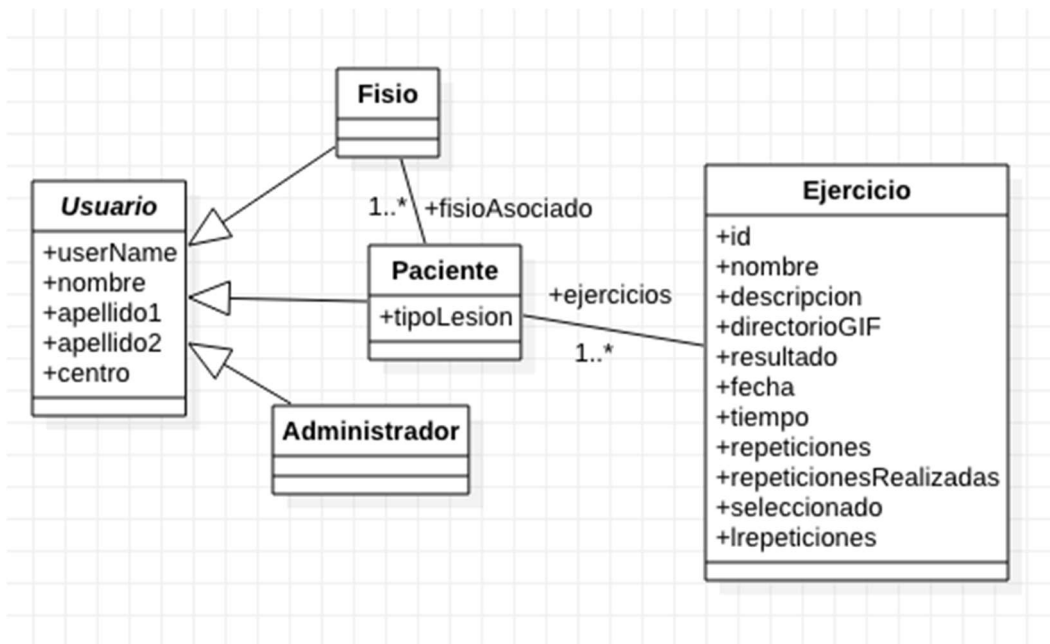


DIAGRAMA PROGRAMACIÓN AVANZADA [1]

La clase Usuario generaliza las clases Fisio, Paciente y Administrador. En consecuencia, estas tres clases heredan de Usuario todos sus atributos y métodos.



HERENCIA JAVA [2]

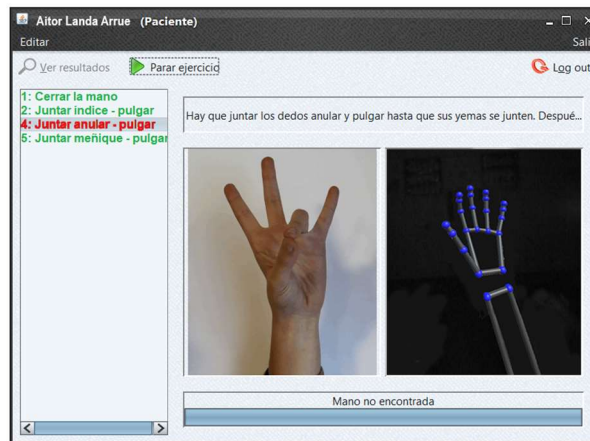
4.1.1 PACIENTE

Cuando el paciente inicia sesión, se abre la pantalla para realizar diferentes ejercicios (). La ventana hereda de *JFrame* las características, y se le aplica un *BorderLayout*. En el oeste del *BorderLayout* se visualiza una *JList* de *Ejercicios* donde se pueden observar los nombres de cada ejercicio que tiene disponibles el paciente. Mediante un renderer a los ejercicios completados se les cambia el color de la fuente a verde.

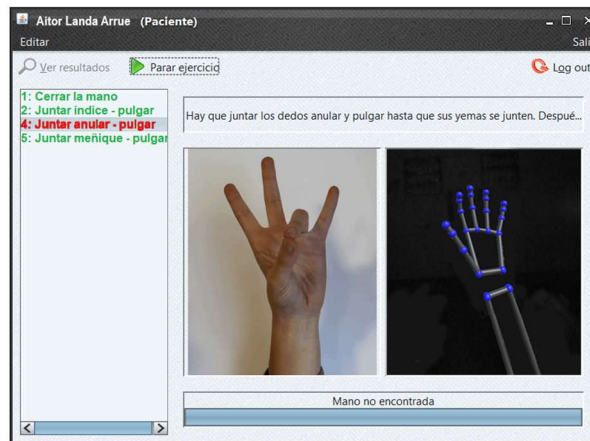
Por otra parte, en el centro del panel se encuentran añadidos en un *GridLayout* la imagen que se utiliza como guía para el ejercicio y el *universo* donde se mostrará la mano que detecta la Leap Motion. Justo encima de este *GridLayout* se encuentra el *JLabel* en que se muestra la descripción de cada ejercicio.

Se ha de comentar el *JMenuBar* que implementa esta ventana, haciendo click en él se despliegan las opciones de *Ver Resultados*, *Hacer ejercicio* y *Log out*, las mismas funciones que ofrece el *JToolBar* que se encuentra justo debajo. Si se hace click en *Hacer Ejercicio* la Leap Motion empezaría a enviar los datos de sus cámaras infrarrojas().

Desde la opción de *Ver resultados* se mostrará los resultados de los ejercicios en una tabla o un gráfico. El gráfico y la tabla están separadas por pestañas creadas con *TabPane*. Gracias a *JCalendar* se pueden elegir el día del que quieres obtener resultados, así como del progreso del mes ().



EJERCICIOS [3]



EJERCICIOS LEAP MOTION [4]



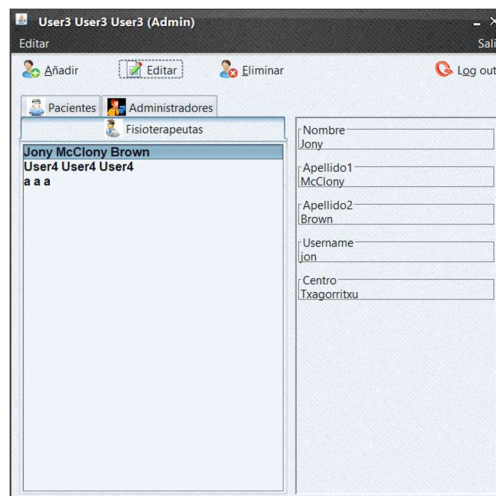
RESULTADO DE EJERCICIOS [5]

4.2.3 ADMINISTRADOR

Cuando se inicia sesión como administrador, se ofrece la opción de añadir pacientes, fisioterapeutas o administradores, además de poder editarlos o incluso borrarlos.

Esta ventana está compuesta por un *JMenuBar* en la parte superior con un *JMenu* incluido en su interior, seguida de un *toolbar* que ofrece las opciones anteriormente citadas. En la parte izquierda de la ventana hay un *JTabbedPane* compuesto por tres pestañas diferentes como se puede observar en la imagen inferior. Además, hay tres *JList* en las que aparecen los pacientes, fisioterapeutas o administradores creados. Por otro lado, en la parte derecha de la ventana se podrá ver toda la información del usuario seleccionado.

Por último, en la parte superior derecha existe la posibilidad de *salir* de la ventana o simplemente de *Log out*.



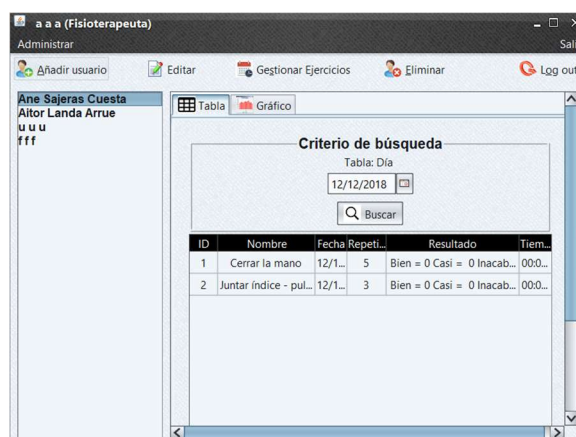
ADMINISTRADOR [6]

4.1.3 FISIOTERAPEUTA

Cuando se inicia sesión como fisioterapeuta, se ofrece la opción de añadir o modificar los usuarios, gestionar los diferentes ejercicios que se le van a asignar a cada usuario y la opción de eliminar el paciente.

Esta ventana está compuesta por un *JMenuBar* en la parte superior que contiene un *JMenu* en su interior, seguida de un *toolbar* en la que se ofrecen las opciones anteriormente citadas. En la parte izquierda de la ventana se visualizarán todos los pacientes del fisioterapeuta con el que se ha iniciado sesión y en la parte central hay un *JTabbedPane* compuesto por una tabla y un gráfico. En la pestaña “tabla” se ofrece la opción de elegir una fecha mediante un *JDateChooser* y así en la parte inferior de esta ventana se pueden ver los diferentes ejercicios que están programados para ese paciente en la fecha seleccionada. Para visualizar la tabla, será necesario darle al botón de buscar. Respecto a la pestaña de “gráficos”, se ofrecen dos opciones diferentes para la visualización de los gráficos. Para ello se emplea un *ButtonGroup* con dos *RadioButton* uno de día y otro de mes. Se debe seleccionar uno de ellos y darle al botón de buscar para ver el gráfico de la opción seleccionada. Para ello se ha utilizado la librería *JFreeChart*.

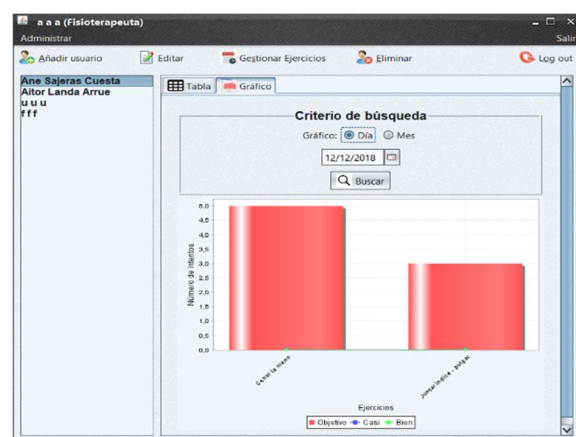
Respecto a la asignación de los ejercicios, se asignan por paciente y por día ofreciendo la opción de modificar los ejercicios programados con anterioridad e incluso de borrarlos. De esta manera se puede seguir de cerca la evolución y el progreso del paciente. En la opción de gestionar los ejercicios se deberá poner la fecha, el ejercicio, las repeticiones y el tiempo estimado para llevar a cabo el ejercicio. Por último, en la parte superior derecha existe la posibilidad de salir de la ventana o únicamente de cerrar la sesión.



The screenshot shows the 'Fisioterapeuta' application interface. On the left, there is a sidebar with the user's name 'Ane Sajeras Cuesta' and some statistics. The main area has a 'Criterio de búsqueda' (Search criteria) section with a date selector set to '12/12/2018' and a 'Buscar' button. Below this is a table with the following data:

ID	Nombre	Fecha	Repeti...	Resultado	Tiem...
1	Cerrar la mano	12/1...	5	Bien = 0 Casi = 0 Inacab...	00:0...
2	Juntar índice - pul...	12/1...	3	Bien = 0 Casi = 0 Inacab...	00:0...

FISIOTERAPEUTA- TABLA [7]



FISIOTERAPEUTA- GRÁFICO [8]

4.1.4 LEAP MOTION

La Leap Motion es un sensor que cuenta con tres cámaras infrarrojas con las que se detecta el movimiento de la mano. Para ello, se crea un objeto llamado SimpleUniverse en Java, es decir, crea un espacio 3D apto para la visualización de diferentes objetos.

La Leap Motion proporciona diferentes funciones para así poder crear una mano en 3D y poder transmitir los datos. Para esto, se usan los métodos de las clases Bond y Puntos.

Para la realización de ejercicios se usan las funciones de la clase Hand.

Por ejemplo, para un ejercicio de juntar dos dedos primero determinamos el dedo y le añadimos el tipo (meñique, pulgar...) mediante la función `type()` de `Finger`. Gracias a esto se puede saber la posición donde se encuentra cada dedo. Sabiendo la posición de ambos dedos, se puede determinar que se han tocado cuando la posición de cada punta del dedo coincide. Con la cercanía de estos dos puntos también se registra las veces que casi se consigue realizar el movimiento.

En el caso del ejercicio de cerrar mano, al crear la mano determinamos la radio de la mano, y así se percibe cuando se cierra la mano. Todo esto es posible gracias a la clase `Hand`.

En el caso del giro de mano, se usa `Hand` ya que las funciones calculan constantemente la posición de la mano en los ejes tridimensionales X Y Z respecto al sensor. Cuando la palma de la mano está mirando a la leap, el vector z tendrá un valor entre 0 y -1. Cuando se gira la mano 180 grados, el vector z también cambia a tener un valor acotado entre 0 y 1. Cuanto más cerca esté del 1 o -1 sabremos cuando está la palma girada del todo en horizontal.

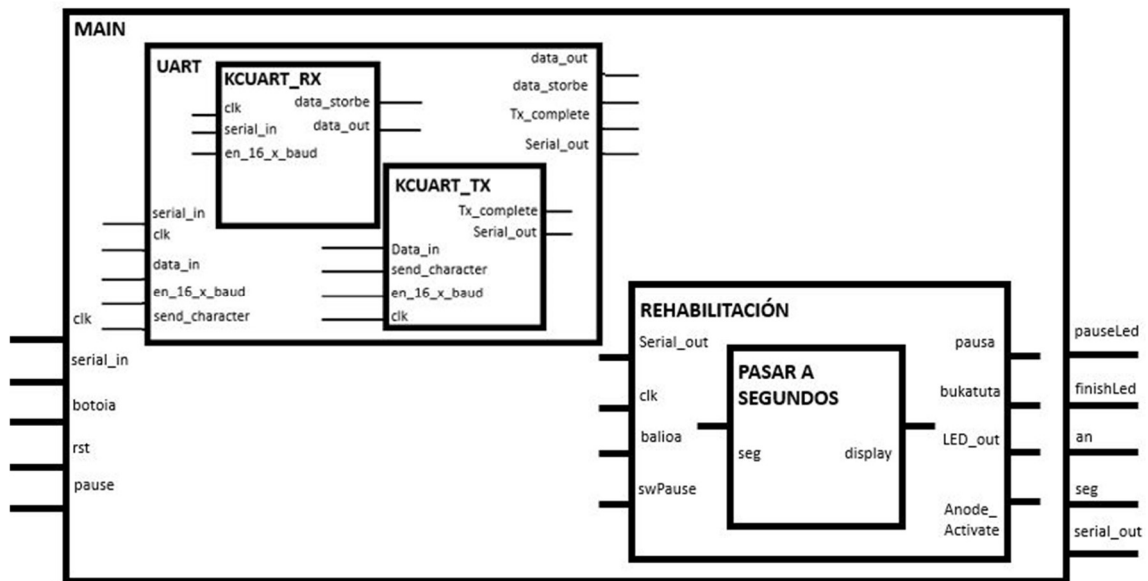


LEAP MOTION [9]

4.2. SISTEMAS LÓGICOS PROGRAMABLES

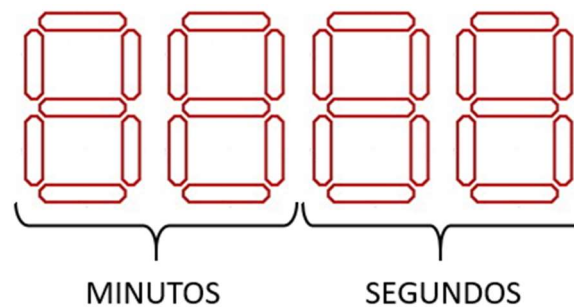
Se utiliza una placa Basys3 para hacer la cuenta atrás en cada ejercicio. Para ello se ha utilizado el programa Vivado 2017.4.

El programa tiene un programa principal, `main.vhdl` que contiene dos componentes internos, que a la vez tienen otros componentes:



SISTEMAS LÓGICOS PROGRAMABLES [10]

El programa se encarga de recibir los segundos y visualizarlos en la placa Basys3 en el siguiente orden:



BASYS3 [11]

Como mucho se visualizará 59:59, una hora, con lo cual, sólo se podrá pasar el número 3559. $3559/60 = 59,59$ Además cuando el contador llegue a 0, se iluminará un LED de aviso.

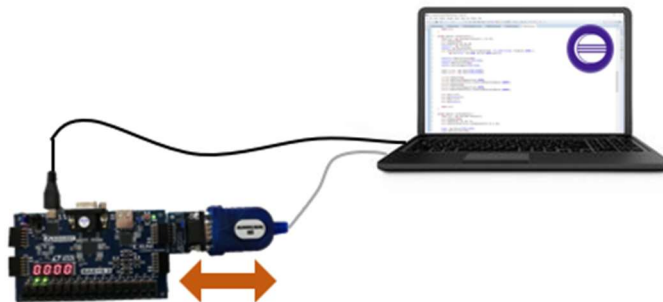
4.2.3 COMPONENTE REHABILITACIÓN

El componente “**Rehabilitación**” contiene otro componente que hace el cambio de los segundos a minutos y segundos y los sitúa en su respectivo lugar. Por ejemplo:

Si pasamos 70 segundos este componente **“Pasar a segundos”** lo dividirá entre 60 sacando así los minutos $70/60 = 1$ y los situará en el tercer ánodo (en caso de tener 2 dígitos los situará en el cuarto y tercer ánodo). Después calcula los segundos sacando el módulo de éste, $70\%60 = 10$ y los situará en el segundo y primer ánodo respectivamente.

4.2.2 COMPONENTE UART

El componente **“UART”** que contiene otros dos componentes **“KUART_RX”** y **“KUART_TX”**, receptor y transmisor sucesivamente, se encarga de recibir y enviar la señal entre la placa Basys3 y el programa de Java.



UART [12]

4.2.2.1 COMPONENTES KUART_RX Y KUART_TX

El componente KUART_RX se encarga de recibir la información y el componente KUART_TX de transmitirlo. Para ello la Basys3 deberá estar conectada al pc mediante USB y un USB serial converter.

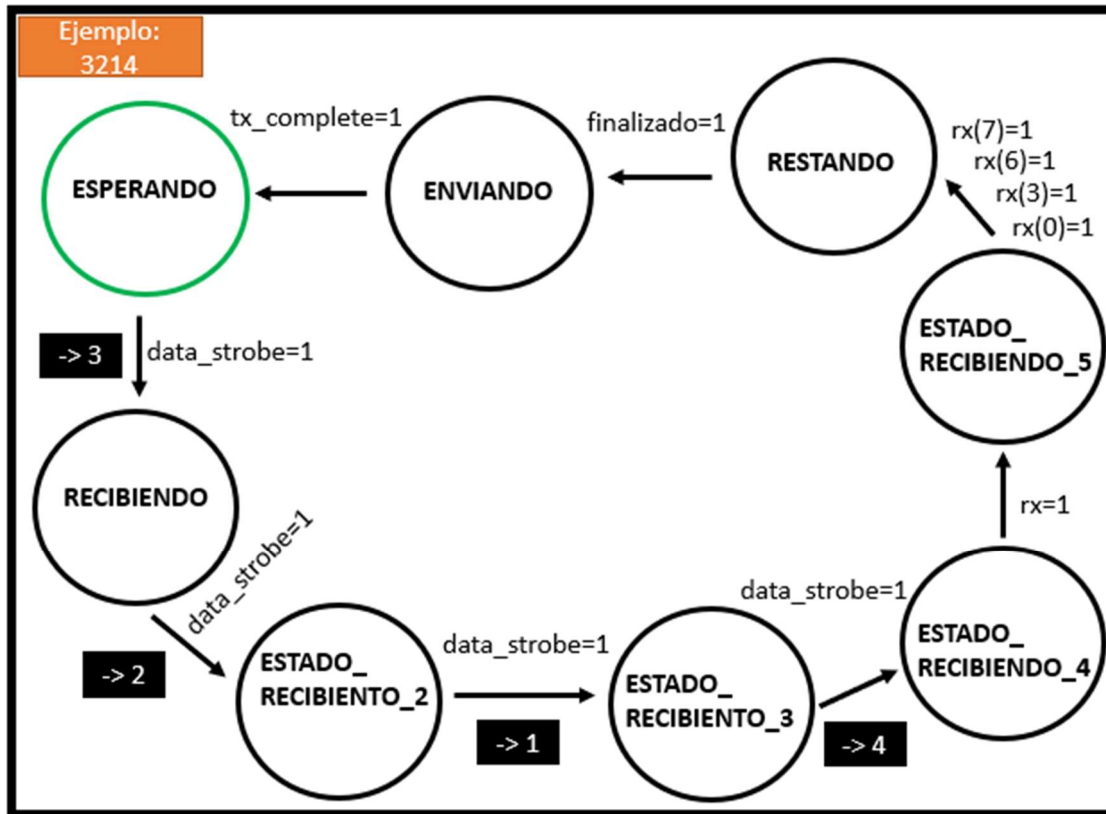
Para poder conseguir esta comunicación, se han hecho *“Diagramas de Estado”* donde el programa espera a recibir “algo” en nuestro caso un número, para después responder con otro.

4.2.3 COMPONENTE PRINCIPAL MAIN

El programa principal “Main” une todos los componentes para que la transmisión de datos y el cronómetro funcionen correctamente y en conjunto. Antes de probarlo con la aplicación Java, se han realizado pruebas con Tera Term para que se asegure una buena conexión emisor/receptor de los datos.

En el programa final, cuando Java envíe los segundos a la basys3, ésta la escuchará, inicializará la cuenta atrás de los segundos asignados y al llegar a

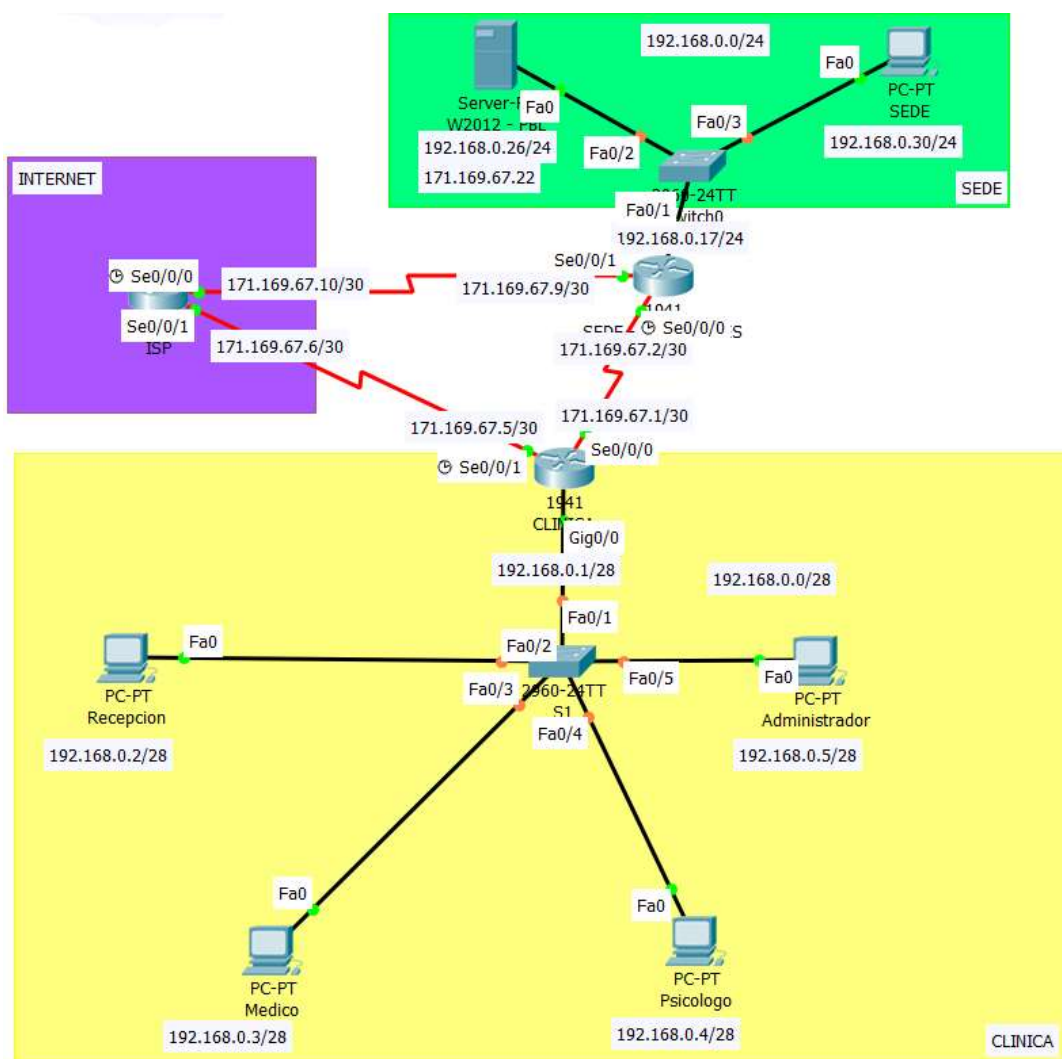
00:00 enviará una señal de vuelta a Java para transmitirle la finalización del tiempo y por ello el final del ejercicio.



COMPONENTE PRINCIPAL MAIN [13]

4.3. REDES DE COMUNICACIONES

La red se divide en tres redes: Clínica, Sede e ISP que están interconectadas mediante cables seriales. La red Clínica es la red principal, mediante ella tanto los pacientes como el personal de la Clínica de Rehabilitación pueden conectarse a Internet mediante la conexión con el ISP (Internet Service Provider). El personal de la clínica podría guardar los datos en la red Sede, ya que, en ella se aloja un servidor, el cual cuenta con el servicio TFTP que sería válido tanto para guardar las copias de seguridad de los routers como para guardar los datos de los pacientes. A continuación, se pasará a describir en profundidad las redes.



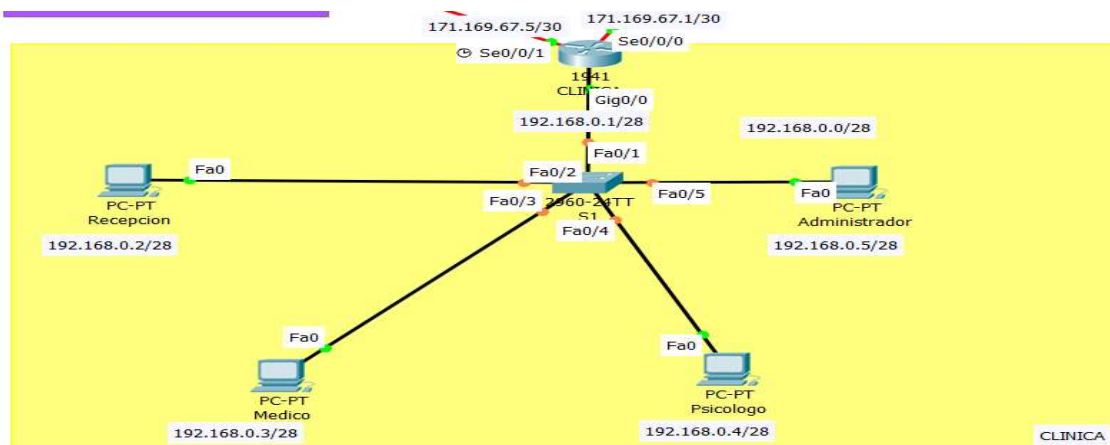
REDES [14]

4.3.1 RED CLÍNICA

La red Clínica es la red a la que se conectan los usuarios habituales de la Clínica de Rehabilitación. La red consta de la IP privada 192.168.0.0/28 y admite hasta un máximo de catorce dispositivos, ya que, tiene una máscara de subred "/28", es decir, 255.255.255.240. Teniendo en cuenta el personal de clínica que podría aproximarse a siete personas de forma simultánea, por lo que quedarían otras siete direcciones IP disponibles.

Los dispositivos de esta red están configurados con direcciones IP estáticas, por lo que tienen mayor estabilidad, son más fáciles de acceder ya que no hay que preocuparse de los cambios en la dirección IP, y en ningún tiempo está el dispositivo desconectado ya que no hay que renovar la dirección.

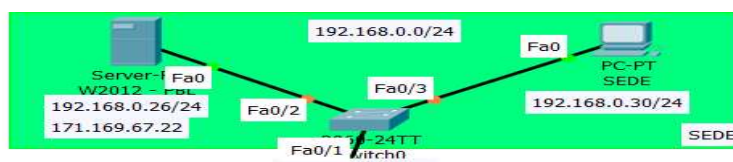
Para la comunicación con redes remotas, esta red cuenta con el protocolo PAT o NAT por sobrecarga, gracias a este protocolo para la traducción de las direcciones IP de la red las direcciones IP privadas se traducen a la IP pública de la interfaz Serial 0/0/0 diferenciándose mediante el puerto (IP publica:puerto).



RED CLÍNICA [15]

4.3.2 RED SEDE

Esta red aloja el servidor de la red, que proporciona los servicios de SYSLOG para monitorear los cambios en la configuración de la red, TFTP (Trivial File Transfer Protocol) para la transferencia de archivos y el servicio de NTP (Network Time Protocol) para la sincronización de los relojes de los dispositivos de red. El router de Sede, conoce las demás redes de la red gracias a EIGRP. El servidor tiene una dirección IPv4 estática privada asignada para que pueda ser accedido desde el exterior. Esta traducción de dirección privada a pública y viceversa se realiza utilizando NAT estático.



RED CLÍNICA [16]

4.3.3 RED ISP

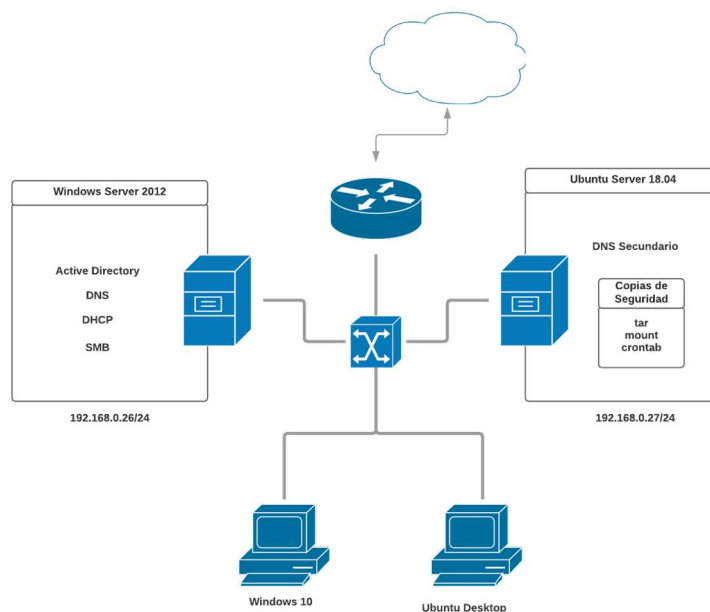
La red ISP proporciona conexión a Internet y también un segundo camino en forma de ruta estática para llegar a la sede informática en caso de que el enlace directo entre la clínica y la sede falle. A la hora de configurar EIGRP también se tiene en cuenta ya que debe conocer las demás redes de Sede y Clínica.



RED ISP [17]

4.4.- INFRAESTRUCTURAS Y SISTEMAS

Se utilizan un total de dos servidores, uno equipado con Windows Server 2012 R2 y el otro con Ubuntu Server 18.04. El Servidor Windows se utiliza como el servidor principal con los servicios de Active Directory, DHCP, DNS y SMB. Por otro lado, el servidor Ubuntu dará soporte al DNS secundario y a la realización de las copias de seguridad de los ficheros de la aplicación de Java que se harán mediante un script y la utilización de crontab.



TOPOLOGÍA SERVIDORES [18]

4.4.1.- Windows Server

Para tener bien localizado el servidor, se asigna una IPv4 estática, en este caso es el 192.168.0.26 . También se procede a cambiar el nombre del equipo por uno más sencillo y fácil de recordar.

4.4.1.1.- Dominio Active Directory

Se crea el dominio ESKOLA y desde Active Directory se crean una serie de usuarios, grupos y unidades organizativas. Las unidades organizativas de fisios, recepción y médicos contienen grupos de fisios, recepción y médicos respectivamente. También hay un grupo creado que engloba todos los usuarios de pacientes.

4.4.1.2.- DNS

Se añaden diferentes registros de tipo HOST(A) y CNAME o alias. Se añade el registro con nombre “eskola” y se asigna la dirección IPv4 192.168.0.26. Se realiza lo mismo para un registro con nombre “www”. Para que en caso de que un usuario se confunda y escriba “ww” o “www”, se crean otros dos registros de tipo Alias(CNAME) que apuntan al registro HOST(A) www.

4.4.1.3.- DHCP

En un nuevo ámbito con descripción rehabilitación y dirección IPv4 192.168.0.0 se reparten las direcciones correspondientes al rango 192.168.0.1 - 192.168.0.14 con la máscara 255.255.255.240 o /28. La duración de la concesión de las direcciones será de un día. También se especifican otras opciones del ámbito como la dirección del router (192.168.0.1) y el nombre del dominio DNS (eskola.edu)

4.4.1.4.- Recursos Compartidos / SMB

Para poder acceder a los ficheros de la aplicación Java desde el exterior, se procede a habilitar la posibilidad de acceder a la carpeta que contiene estos ficheros de la aplicación Java. Al usuario Administrador se le otorga control total sobre este directorio y se utilizarán sus credenciales cuando se quiera acceder al recurso compartido.

4.4.2.- Ubuntu Server

Para tener bien localizado el servidor, se asigna una IPv4 estática, en este caso es el 192.168.0.27.

4.4.2.1.- DNS esclavo

Se configura el DNS secundario o esclavo, como soporte al DNS principal en Windows Server para repartir la carga de trabajo, y/o poder funcionar en caso de que el servidor principal caiga.

4.4.2.2.- Copias de seguridad

Para realizar las copias de seguridad. Primero vamos a crear una carpeta en Windows Server que posteriormente lo compartiremos como recurso compartido mediante SMB.

Combinando las herramientas de crontab, tar, mount y el uso de un script vamos a automatizar el proceso. Como la clínica abre los días laborables, hemos decidido hacer una copia de seguridad cada día de lunes a viernes después del cierre de la clínica, a las 23:00.

El script que se ejecutará creará una carpeta temporal donde poder montar el recurso compartido que contiene todos los ficheros. Una vez hecho el montaje, tar creará una copia de seguridad con la fecha que se guardará localmente. Una vez que esté hecho la copia de seguridad, desmontamos la carpeta temporal y la borramos.

Este script contiene un comando mount que el teoría debería funcionar pero que en la práctica no lo hemos podido llevar a cabo. Todo el proceso de pruebas y verificaciones se explican en el apéndice.

5. CONCLUSIONES Y LÍNEAS FUTURAS

En conclusión, se ha conseguido que nuestro producto funcione como lo esperado. Teniendo en cuenta de que siempre se puede mejorar se añaden las siguientes líneas futuras:

- **Líneas Futuras**

Un aspecto a mejorar sería cambiar la Leap Motion por un Hardware más preciso que proporcione datos más detallados y sea más sensible a la lectura de los movimientos.

Trabajar en consenso con un equipo de fisioterapeutas para acercar el programa a la realidad además de añadir más ejercicios, mejorar el sistema y la experiencia del usuario.

Crear un servicio de streaming de los movimientos que el paciente realiza en un lugar independiente a la clínica y que el fisioterapeuta los vea a tiempo real desde su equipo.

6. **BIBLIOGRAFÍA**

[1] **“CRÓNOMETRO CÓDIGO”** [Accedido en Enero del 2019] Accessible desde <http://guareza.blogspot.com/2013/12/cronometro-codigo-vhdl.html>

[2] **Java Chart ,fecha 19 de diciembre** www.java2s.com, Recuperado de <http://www.java2s.com/Code/Java/Chart/CatalogChart.htm>

[3] **Rootwyrn, (2016) fecha 4 de diciembre, Using BIND (Linux) as a backup DNS server to a Windows DNS server?**, reddit.com, Recuperado de www.reddit.com/r/homelab/comments/3zqg2y/using_bind_linux_as_a_backup_dns_server_to_a/

7. APÉNDICE

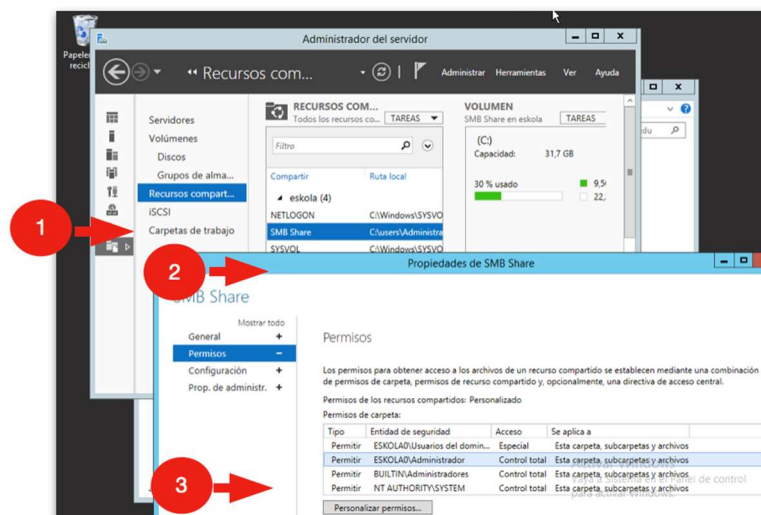
Copias de Seguridad

Tenemos una aplicación Java en la cual se gestionan los datos de una clínica. Esta aplicación guarda todos los datos en una serie de ficheros. Para prevenir la pérdida de estos datos, decidimos guardar una copia de seguridad diaria en un servidor equipado con Ubuntu Server.

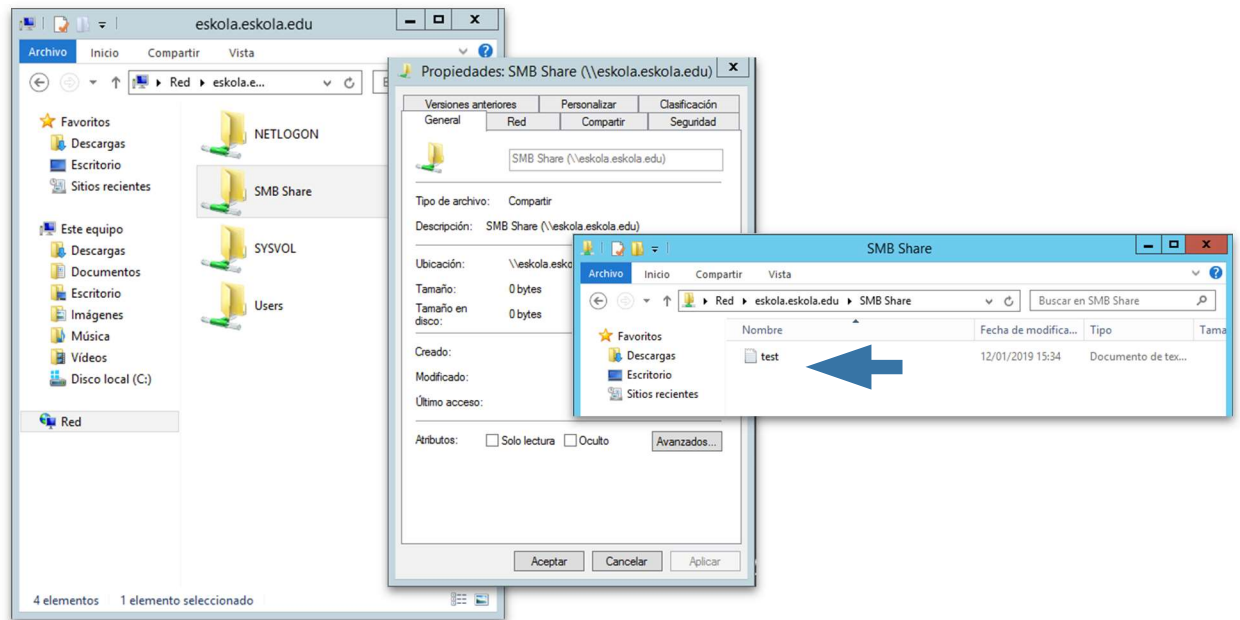
Para poder llevar a cabo todo este proceso vamos a seguir los siguientes pasos:

- 1.- Desde Windows Server compartir por SMB la carpeta que contiene los ficheros de la aplicación.
- 2.- Montar la carpeta en Ubuntu Server.
- 3.- Utilizando la herramienta tar, hacer una copia de seguridad de la carpeta montada que guardaremos en un directorio de Ubuntu Server.
- 4.- Para poder automatizar el proceso vamos a utilizar crontab, pondremos los pasos 2 y 3 en un script y crontab ejecutará este de lunes a viernes a las 00:01.

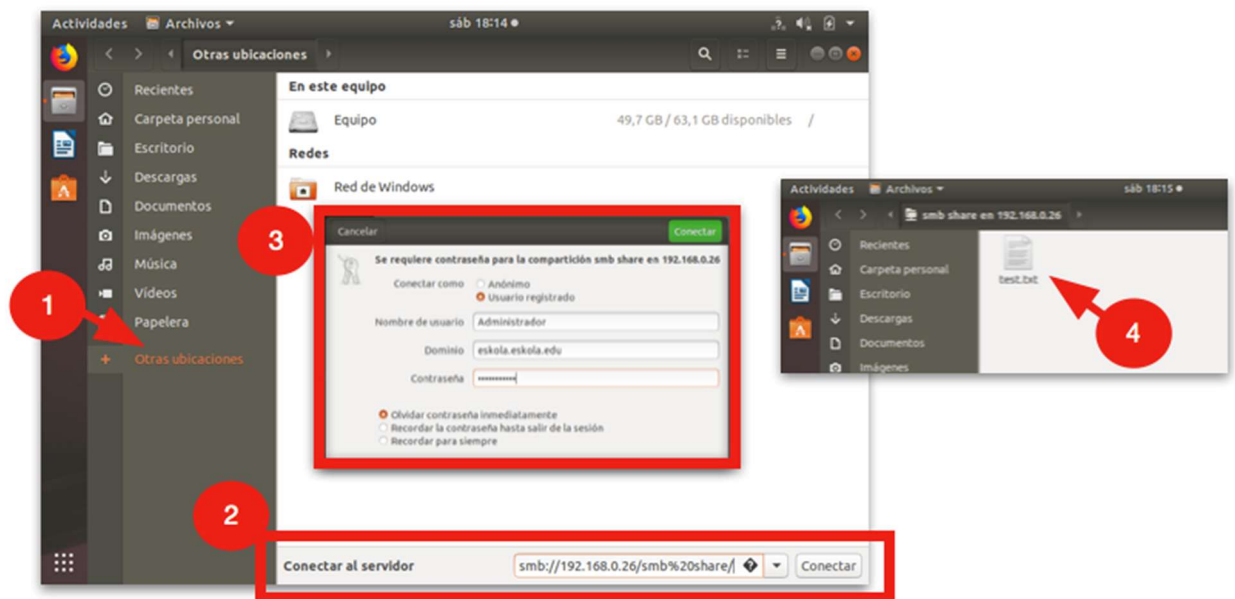
Los ficheros de la aplicación están alojados en una carpeta llamada SMB Share y está configurada como recurso compartido. El usuario Administrador tiene control total sobre este directorio.



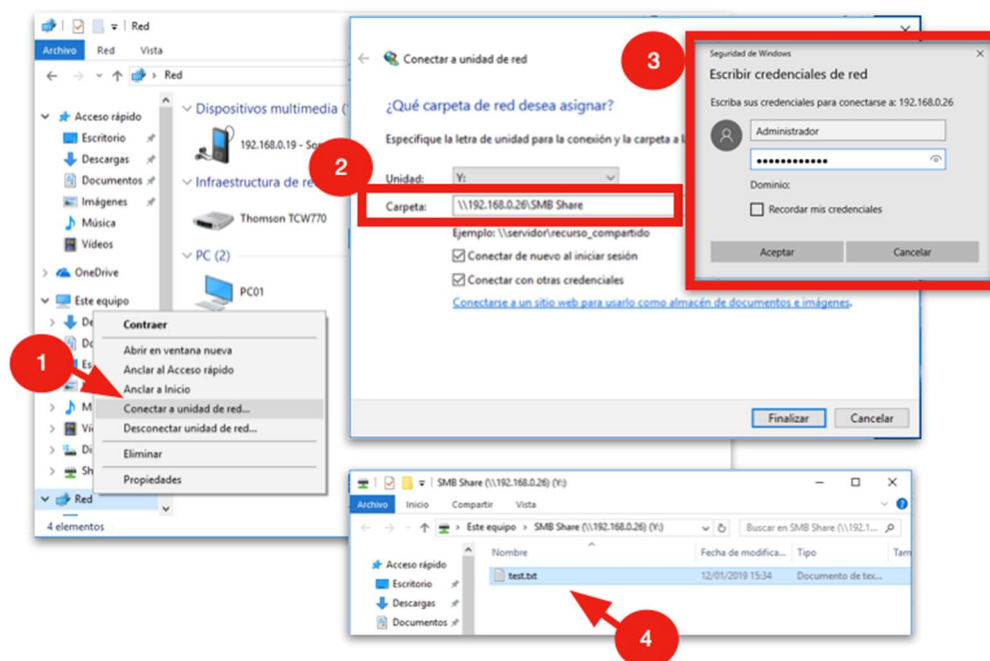
Dentro de el directorio hemos creado un fichero de texto llamado test.txt



Para asegurarnos de que se puede acceder al directorio, vamos a probar utilizando el explorador de archivos Nautilus que trae la version de sobremesa de Ubuntu. Nautilus nos pide un nombre de usuario, un nombre del dominio y una contraseña. Introduciendo los datos correctos, logramos acceder a los contenidos del directorio.

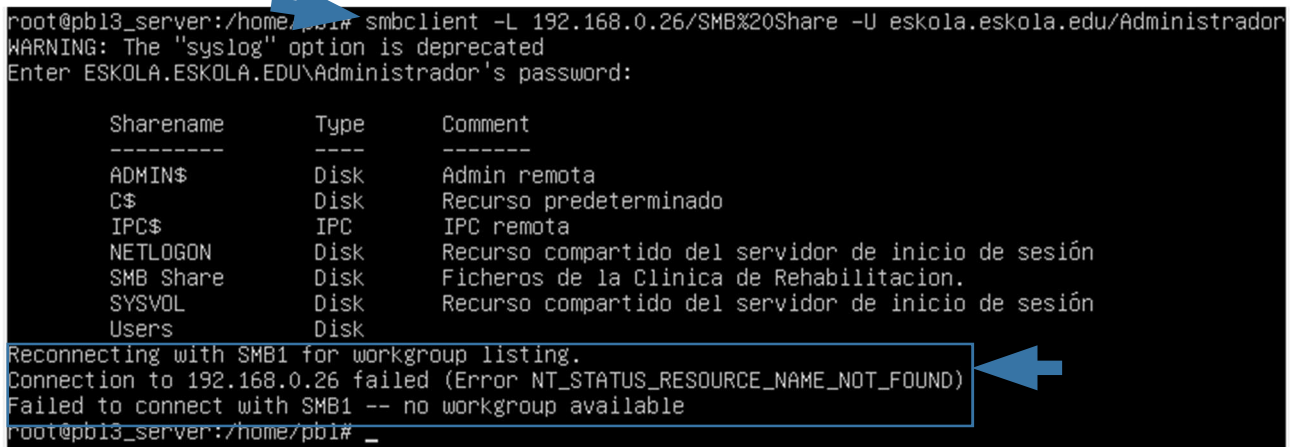


Para asegurar el correcto funcionamiento, haremos lo mismo desde la versión de sobremesa de Windows.



Desde Ubuntu Server debemos utilizar los comandos de terminal. Para poder ver el directorio hemos instalado el paquete smbclient y para poder montar el directorio en la máquina hemos instalado cifs-utils.

Utilizando smbclient -L 192.168.0.26/SMB%20Share -U Administrador -W eskola.eskola.edu, muestra todos los recursos compartidos que tiene el servidor, pero seguido muestra un mensaje de error



```

root@pbl3_server:/home/pbl# smbclient -L 192.168.0.26/SMB%20Share -U eskola.eskola.edu/Administrador
WARNING: The "syslog" option is deprecated
Enter ESKOLA.ESKOLA.EDU\Administrador's password:

      Sharename      Type      Comment
      -----
      ADMIN$         Disk      Admin remota
      C$              Disk      Recurso predeterminado
      IPC$            IPC       IPC remota
      NETLOGON        Disk      Recurso compartido del servidor de inicio de sesión
      SMB Share       Disk      Ficheros de la Clinica de Rehabilitacion.
      SYSVOL          Disk      Recurso compartido del servidor de inicio de sesión
      Users           Disk
Reconnecting with SMB1 for workgroup listing.
Connection to 192.168.0.26 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Failed to connect with SMB1 -- no workgroup available
root@pbl3_server:/home/pbl# _

```

NT_STATUS_RESOURCE_NAME_NOT_FOUND Y NO muestra nada más.

Hemos conseguido el mismo resultado con los siguientes comandos:

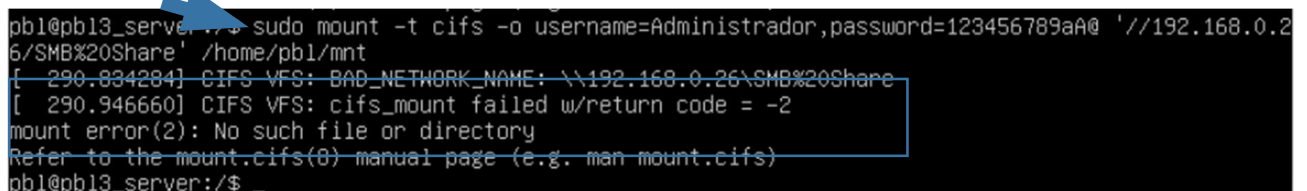
```

smbclient -L 192.168.0.26/SMB%20Share -U Administrador -W
eskola.eskola.edu
smbclient -L 192.168.0.26/SMB%20Share -U Administrador -W
eskola.eskola.edu
smbclient -L 192.168.0.26/SMB%20Share -U eskola.eskola.edu/Administrador -
W eskola.eskola.edu

```

- poniendo //192.168.0.26/SMB%20Share en todos

No hemos podido acceder al directorio utilizando smbclient. Sin embargo, probamos si se puede montar el recurso compartido en nuestro sistema. Primero creamos un directorio (home/pbl/mnt) donde poder montar y utilizando el comando mount -t cifs -o username=Administrador,password=123456789aA@ '//192.168.0.26/SMB%20Share' /home/pbl/mnt , pero lanza el error BAD_NETWORK_NAME y notifica que el montaje ha fallado con el error -2.



```

pbl@pbl3_server:~$ sudo mount -t cifs -o username=Administrador,password=123456789aA@ '//192.168.0.26/SMB%20Share' /home/pbl/mnt
[ 290.834284] CIFS VFS: BAD_NETWORK_NAME: \\192.168.0.26\SMB%20Share
[ 290.946660] CIFS VFS: cifs_mount failed w/return code = -2
mount error(2): No such file or directory
Refer to the mount.cifs(8) manual page (e.g. man mount.cifs)
pbl@pbl3_server:/$ _

```

Para asegurar que el error no se debe a los permisos que tiene el directorio mnt, le damos todos los privilegios a este.

Hemos conseguido el mismo resultado con los siguientes comandos:

```
mount -t cifs -o username=Administrador,password=123456789aA@  
//192.168.0.26/SMB%20Share home/pbl/mnt  
mount -t cifs //192.168.0.26/SMB%20Share /mnt -o  
username=Administrador,password=12345678aA@,domain=eskola.eskola.edu  
,vers=1.0  
mount -t cifs //192.168.0.26/SMB%20Share /mnt -o  
username=Administrador,password=12345678aA@,domain=eskola.eskola.edu  
,vers=2.0  
mount -t cifs //192.168.0.26/SMB%20Share /mnt -o  
username=Administrador,password=12345678aA@,domain=eskola.eskola.edu  
,vers=2.1  
mount -t cifs //192.168.0.26/SMB%20Share /mnt -o  
username=Administrador,password=12345678aA@,domain=eskola.eskola.edu  
,vers=3.0
```

Diagrama de Gantt

