

## ANEXOS

LISTA DE TABLAS.....	II
LISTA DE FIGURAS .....	III
ANEXO A: Manual del usuario del Multi transmisor Virtual.....	1
ANEXO B: Guía de prácticas .....	10
ANEXO C: Técnicas de procesamiento de imágenes alternativos.....	36
ANEXO D: Diagrama de Moody.....	38
ANEXO E. Manual del usuario de la Instalación del Sistema Software de soporte del Multi transmisor Virtual .....	39
ANEXO F. Código Fuente del Multi transmisor Virtual de alturas piezométricas	53
REFERENCIAS BIBLIOGRÁFICAS.....	81

## LISTA DE TABLAS

Tabla 1. Características de aditamentos.....	27
Tabla 2. Consignación de datos de alturas piezométricas. ....	29
Tabla 3. Cálculo de coeficientes de pérdidas por fricción. ....	30
Tabla 4. Cálculo de coeficientes de pérdidas locales.....	32
Tabla 5. Detalle de los lanzadores. ....	42

## LISTA DE FIGURAS

Figura 1. Calibrar Multi transmisor Virtual .....	1
Figura 2. GUI de configuración para el Multi transmisor Virtual. ....	2
Figura 3. Captura de la escena con guía del puntero.....	3
Figura 4. Elección de puntos de perspectiva.....	4
Figura 5. Resultado de la corrección de perspectiva. ....	5
Figura 6. Estructura de la matriz de perspectiva. ....	6
Figura 7. Estructura de los parámetros de configuración. ....	6
Figura 8. Diagrama en Scicoslab del Multi transmisor virtual.....	7
Figura 9. SuperBlock sistema de control de caudal en Scicoslab. ....	8
Figura 10. Generación de código RTAI con éxito.....	8
Figura 11. Conexión Multi transmisor y drivers DAQ.....	9
Figura 12. Planta de pérdidas de carga en conductos del LRPCC. Fuente: [4] ..	11
Figura 13. Panel de control. Fuente: [4] .....	12
Figura 14. Detalle de válvulas .....	14
Figura 15. Botones y puertos de la bandeja retráctil. Fuente: [4].....	14
Figura 16. Pantalla de inicio de Ubuntu solicitando nombre de usuario. ....	15
Figura 17. Pantalla de inicio de Ubuntu solicitando la contraseña de usuario.....	15
Figura 18. Acceso directo a Cargar Módulos RTAI. ....	16
Figura 19. Acceso directo para descargar los módulos RTAI. ....	16
Figura 20. Acceso directo a la Interfaz de usuario del laboratorio de pérdidas de carga en conductos a presión. ....	17
Figura 21. Ventana de entrada a la Interfaz de usuario. ....	18
Figura 22. Acceso directo a Módulo de control de caudal en RTAI.....	19
Figura 23. Terminal del modulo de control. ....	19
Figura 24. Acceso directo a Multi transmisor Virtual de alturas piezométricas....	20
Figura 25. Interfaz de calibración de imagen. ....	20
Figura 26. Mensaje en terminal del Multi Transmisor Virtual.....	21

Figura 27. Terminal de instrucciones. ....	22
Figura 28. Ventana de interacción con el usuario. ....	23
Figura 29. Visualización de las alturas piezométricas. ....	23
Figura 30. Cálculo de pérdidas por fricción. ....	24
Figura 31. Acceso directo a Servidor LRPCC. ....	25
Figura 32. Autenticación de usuario desde el Browser ....	26
Figura 33. Página inicial del Servidor Web.....	26
Figura 34. Página web de proceso.....	27
Figura 35. Página web para visualización de las medidas de las alturas piezométricas. ....	28
Figura 36. Página web para cálculo de pérdidas por fricción. ....	29
Figura 37. Página web para cálculo de coeficientes de pérdidas locales.....	31
Figura 38. Página web para despliegue de Tendencias Vs Caudal. ....	33
Figura 39. Página web para despliegue de Tendencias Vs Reynolds ....	33
Figura 40. (a) Imagen original, (b) Resultado de búsqueda por plantilla (Template Matching). ....	36
Figura 41. Diagrama de Moody ampliado. Tomado de de [41] ....	38
Figura 42. Estructura de directorios ....	39
Figura 43. Distribución de menú. ....	40
Figura 44. Crear Menú nuevo. ....	41
Figura 45. Propiedades del lanzador.....	42
Figura 46. Estructura de archivos. ....	53

## ANEXO A: MANUAL DEL USUARIO DEL MULTI TRANSMISOR VIRTUAL

El Multi transmisor virtual de alturas piezométricas diseñado e implementado en el presente trabajo de grado, presenta una interfaz grafica de configuración y calibración para algunos aspectos relevantes al funcionamiento de este.

### Ejecución del Multi transmisor Virtual

Situado en el escritorio de Ubuntu, en la barra superior debe acceder al menú “Aplicaciones”, allí encontrará una carpeta de accesos directos llamada RTAI, ingrese en ella y luego acceda a “Calibrar Multi Transmisor Virtual”, ver Figura 1.

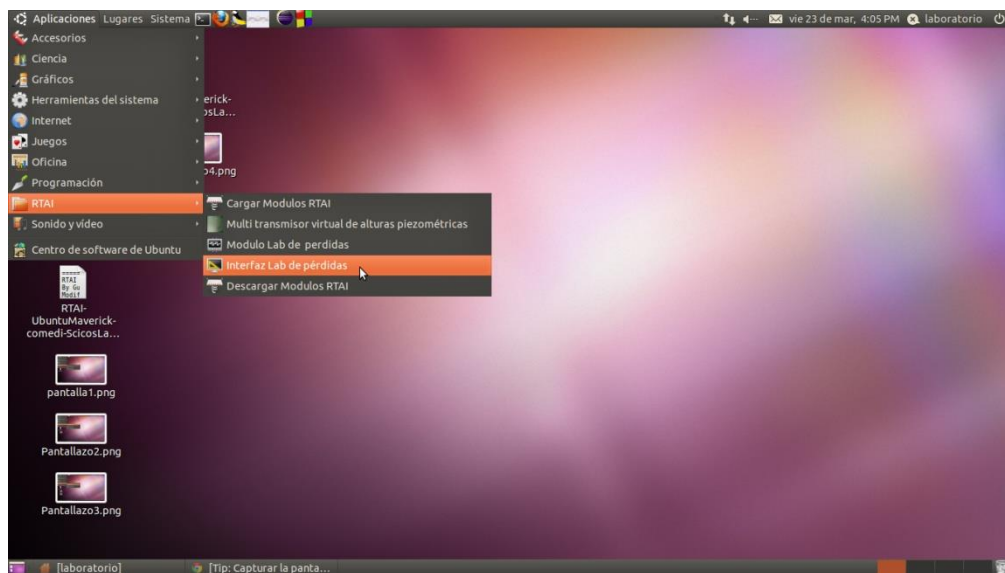


Figura 1. Calibrar Multi transmisor Virtual

No debe olvidare, que para la ejecución del transmisor se necesita el nombre de un mailbox donde el componente de comunicaciones de Multi transmisor depositará la información correspondiente a las 16 alturas piezométricas. Este mailbox puede haber sido creado y registrado previamente por la tarea de control en tiempo real que recibe la información proveniente del Multi transmisor Virtual,

en este caso debe colocar el nombre de mailbox tal como se identifica en la tarea de control.

### Interfaz Gráfica de configuración

Se presenta en la Figura 2 el aspecto de la interfaz grafica diseñada con ayuda de las librerías QT [1], ésta cuenta con:

- Un botón de corrección de perspectiva.
- Una caja de incrementos para los marcos de aprendizaje.
- Una caja de incrementos para el contraste de la imagen.
- Una caja de incrementos para el brillo de la imagen.
- Una caja de texto para el nombre del mailbox.
- Dos cajas de texto para definir el rango del Multi transmisor Virtual.
- Una caja de texto para el cero.
- Una caja de texto para el span.

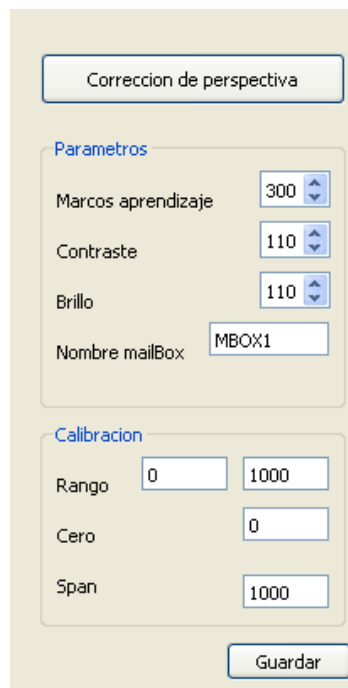


Figura 2. GUI de configuración para el Multi transmisor Virtual.

Los parámetros que se configuran con la interfaz grafica son guardados en un archivo de tipo YAML .yaml (del inglés “YAML Ain't Another Markup Language”) con el fin de que la calibración se realice únicamente cuando las características del Multi transmisor requieran cambiarse, como por ejemplo cuando se mueva el tablero piezométrico o la cámara digital.

Cuando se presiona en botón de corrección de perspectiva, la interfaz grafica despliega una nueva ventana como la mostrada en la Figura 3, donde se indica mediante dos guías con líneas perpendiculares el lugar en el espacio de la escena donde se encuentra el puntero, se debe elegir 4 puntos de corrección de perspectiva para realizar esta corrección, para ver cómo funciona esta corrección ver en el capítulo dos el apartado 2.5. Donde se expone el funcionamiento de la corrección de perspectiva.

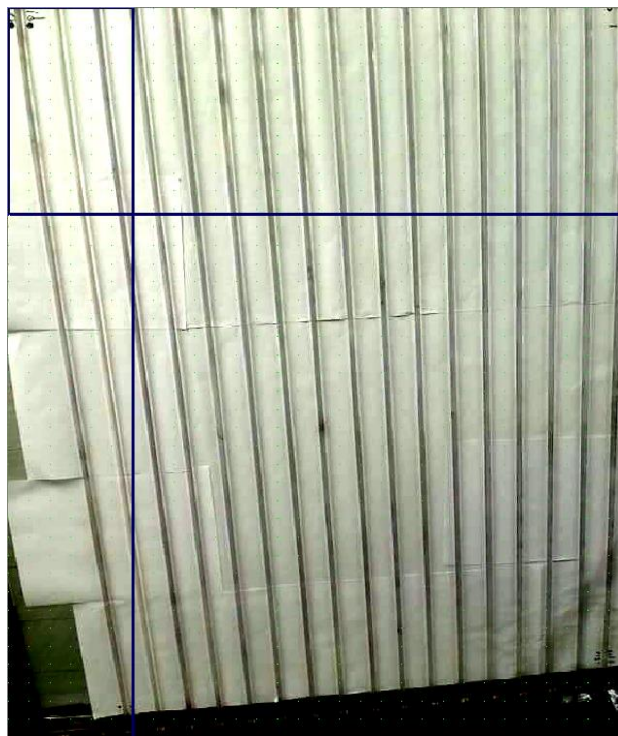


Figura 3. Captura de la escena con guía del puntero.

Los puntos p1, p2, p3 y p4 deben ubicarse respetando la secuencia uno tras otro, como se muestra en la Figura 4

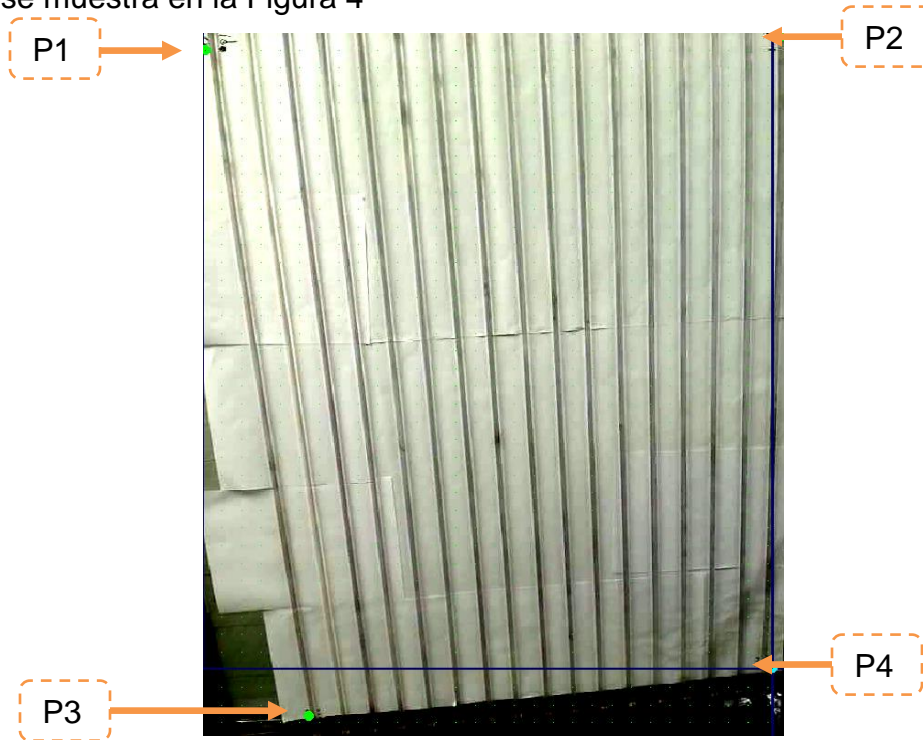


Figura 4. Elección de puntos de perspectiva.

Cuando se termina de elegir los puntos de perspectiva p1, p2, p3 y p4 y se presiona “enter”, aparece una nueva ventana con la corrección de perspectiva aplicada como se puede ver en la Figura 5, en esta corrección se puede ver que los tubos piezométricos se encuentran organizados de manera paralela.





Figura 5. Resultado de la corrección de perspectiva.

En la caja de agrupación se encuentran los parámetros de marcos de aprendizaje cuyo valor numérico de marcos es utilizado para el aprendizaje de fondo en la construcción de fondo y primer plano expuesto en aparte 2.5.5 del capítulo 2, contraste y brillo pertenece a las características de la imagen que pueden elegirse para una buena representación de información de la captura de la imagen, por ultimo nombre del mailbox una cadena de texto sin espacios que consigne el nombre del mailbox por el cual se realiza la comunicación entre el Multi transmisor y el sistema de control o interfaz grafica donde se utilice.

Con el fin de realizar la configuración una sola vez y no cada vez que se utilice el Multi transmisor Virtual de alturas piezométricas, los parámetros de configuración son guardados en archivos de texto de tipo YAML un formato que intenta remplazar los archivos XML [2], la matriz de perspectiva es guardada como una matriz de la utilizada en OpenCV como se puede ver en la Figura 6, donde los parámetros “rows” y “cols” representas las filas y columnas de la matriz respectivamente, en este caso una matriz de 3x3, y en el campo data se almacenan uno a uno los datos de esta matriz.

```

%YAML:1.0
MatPerspective: !!opencv-matrix
  rows: 3
  cols: 3
  dt: d
  data: [ -9.2915369264016567e-001, -1.2842723276196921e-001,
          7.7569222756261724e-001, 2.5177408678539154e-002,
          8.3601364281701429e-001, -1.5801196432132173e+001,
          -1.2136460056261239e-004, -2.5136279773094357e-004, -1. ]

```

Figura 6. Estructura de la matriz de perspectiva.

El resto de parámetros de configuración son guardados en una estructura como se observa en la Figura 7 donde se primero se encuentra el nombre del parámetro seguido de el valor que se le da a este, como primer parámetro se encuentra los marcos de aprendizaje con un valor de 200, en otra línea se encuentra el brillo con un valor de 100, por último se encuentra un parámetro de modo prueba que se utiliza para realizar pruebas de ejecución de código en la implementación del Multi transmisor Virtual.

```

%YAML:1.0
frames_learning: 200
brightness: 100
contrast: 160
create_perspective: 0
calculate_pixilm: 0
height_panel_mm: 1070
min_area_blobs: 20
max_area_blobs: 10000
test_mode: 0

```

Figura 7. Estructura de los parámetros de configuración.

## Bloques de RTAI-Lab en Scicoslab del Multi transmisor Virtual.

Para la construcción de bloques de Scicoslab del Multi transmisor Virtual de alturas piezométricas, basta simplemente con incluir un bloque “Mbx rcv blk”

como se muestra en la Figura 8 y dar el nombre del mailbox el cual se definió en la interfaz grafica de la Figura 2, este bloque envía constantemente el valor de las 16 alturas piezométricas y pueden ser utilizadas en cualquier parte de un sistema de control construido en Scicoslab con RTAI-Lab.

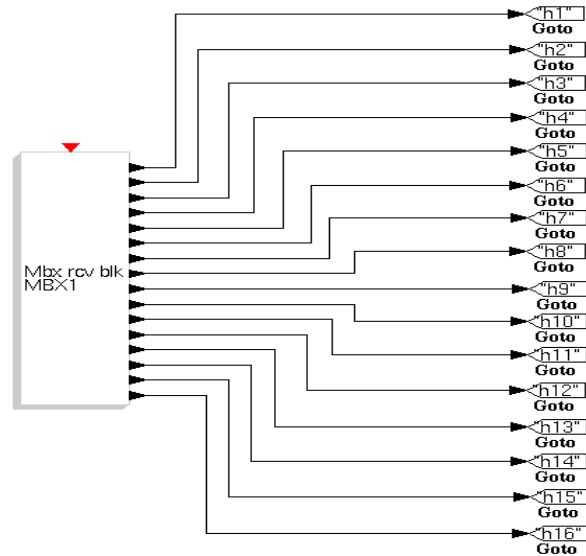


Figura 8. Diagrama en Scicoslab del Multi transmisor virtual.

### Generación de Código para RTAI.

Una vez se construya el diagrama en bloques en Scicoslab del sistema de control, se debe proceder a generar y compilar el código para las tareas de tiempo real asociadas al diagrama, se selecciona RTAI Set Target en el menú RTAI, al seleccionar el súper bloque aparecerá una ventana como se muestra en la Figura 9 donde se pueden configurar las propiedades del ejecutable, como su nombre y directorio donde se guardará el código generado.

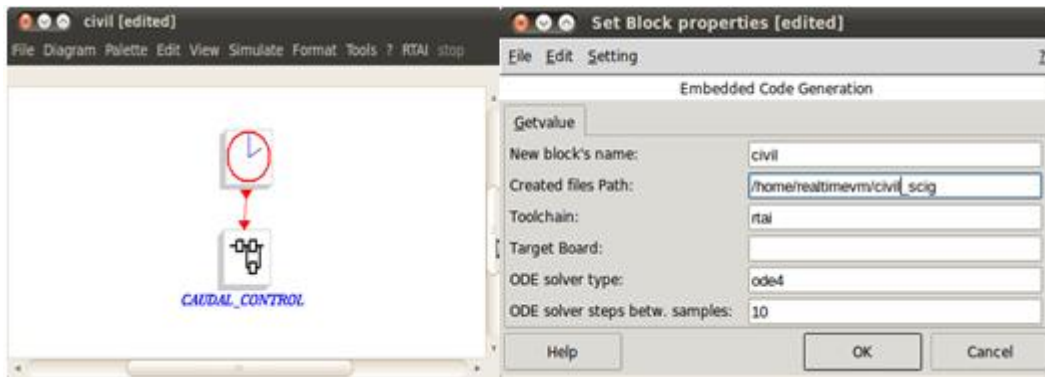


Figura 9. SuperBlock sistema de control de caudal en Scicoslab.

Al realizar la generación de código, aparece en el prompt de ScicosLab una serie de comandos como se ve en la Figura 10 donde se puede ver que la generación y compilación de código se llevo a cabo con éxito “Target generation terminated”, en este momento se ha creado un ejecutable y una directorio con el código generado.

```

ScicosLab
File Control Demos Help Editor toolboxes Graphic Window 1000
l -DMODELN=civil.c -c -o rtmain44.o rtmain44.c
gcc -O -DNDEBUG -Dlinux -DNARROWPROTO -D_GNU_SOURCE -O2 -I/usr/lib/scicoslab-gtk-4.4/routines -I/usr/lib/scicoslab-gtk-4.4/routines/scicos -I. -I/usr/realtime/i
nclude -O2 -I/usr/src/linux/include -Wall -Wstrict-prototypes -pipe -DMODEL=civi
l -DMODELN=civil.c -c -o common.o common.c
gcc -O -DNDEBUG -Dlinux -DNARROWPROTO -D_GNU_SOURCE -O2 -I/usr/lib/scicoslab-gtk-4.4/routines -I/usr/lib/scicoslab-gtk-4.4/routines/scicos -I. -I/usr/realtime/i
nclude -O2 -I/usr/src/linux/include -Wall -Wstrict-prototypes -pipe -DMODEL=civi
l -DMODELN=civil.c -c -o civil.o civil.c
gcc -O -DNDEBUG -Dlinux -DNARROWPROTO -D_GNU_SOURCE -O2 -I/usr/lib/scicoslab-gtk-4.4/routines -I/usr/lib/scicoslab-gtk-4.4/routines/scicos -I. -I/usr/realtime/i
nclude -O2 -I/usr/src/linux/include -Wall -Wstrict-prototypes -pipe -DMODEL=civi
l -DMODELN=civil.c -c -o civil_Cblocks.o civil_Cblocks.c
gcc -static -o ../civil rtmain44.o common.o civil.o civil_Cblocks.o /usr/lib/sc
icoslab-gtk-4.4/libs/scicos.a /usr/lib/scicoslab-gtk-4.4/libs/poly.a /usr/lib/sc
icoslab-gtk-4.4/libs/caelm.a /usr/lib/scicoslab-gtk-4.4/libs/blas.a /usr/lib/sc
icoslab-gtk-4.4/libs/lapack.a /usr/lib/scicoslab-gtk-4.4/libs/os_specific.a /usr
/realtime/lib/libsciblk.a /usr/realtime/lib/liblxrt.a -pthread -lcomedi -lm
### Created executable: civil ###

----> Target generation terminated!

```

Figura 10. Generación de código RTAI con éxito.

Para ejecutar una tarea de tiempo real que se ha generado mediante los bloques de ScicosLab se debe iniciar un terminal de comandos, donde se digita el nombre del ejecutable generado, para el caso general es “*sudo*

`./nombre_ejecutable -v`” la opción `-v` permite mostrar las características de la tarea de tiempo real en el terminal.

## Obtención de la salida Analógica

Para obtener una salida analógica en alguno de los canales de la DAQ, se utilizan los drivers Comedi, ver [3], para dispositivos de adquisición y envío de datos. Éstos drivers están implementados en bloques dentro de la paleta RTAI ó RTAI-Lib de ScicosLab.

El proceso consiste en seleccionar una ó todas las señal de salida Multi transmisor, según la disponibilidad de salidas en su(s) tarjeta(s) DAQ. Y conecta al bloque de salida analógica de Comedi con una ganancia de 65.535. Para más información sobre la configuración de bloque Comedi remítase al RTAI Tutorial en [4].

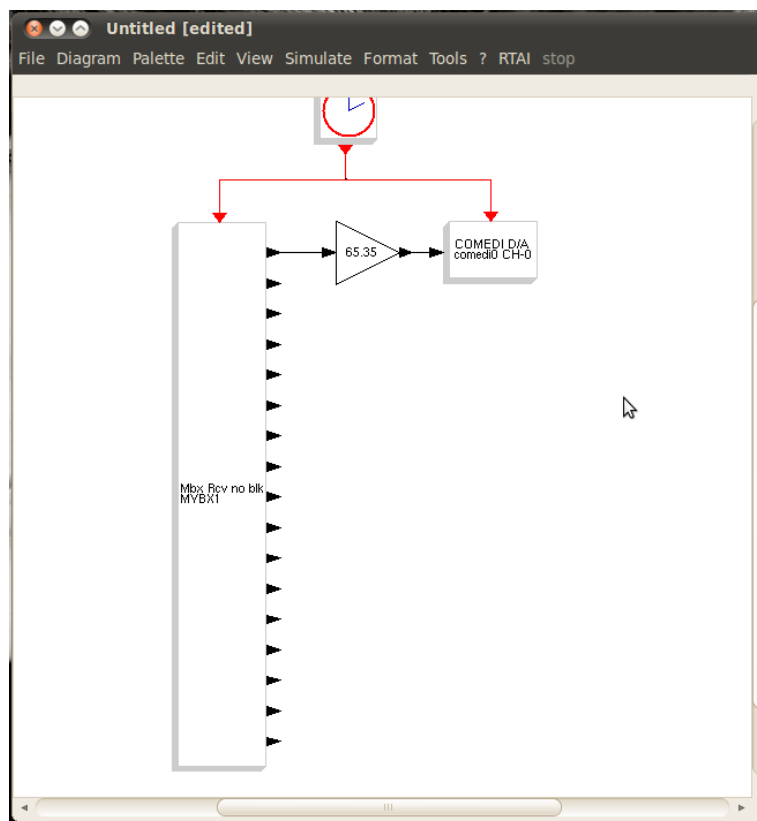


Figura 11. Conexión Multi transmisor y drivers DAQ.

## **ANEXO B: GUÍA DE PRÁCTICAS**

### **LRPCC**

#### **Laboratorio Remoto de Pérdidas de Carga en Conductos a Presión PRÁCTICA DE PÉRDIDAS DE CARGA EN CONDUCTOS A PRESIÓN**

### **OBJETIVOS**

1. Examinar y comprender el comportamiento fluido-dinámico del agua en un circuito hidráulico compuesto por aditamentos, y tubos circulares de diferentes diámetros.
2. Determinar las diferentes formas de pérdidas de energía hidráulica en un circuito hidráulico como el mencionado en el primer objetivo.
3. Caracterizar los diferentes elementos que componen la planta del LRPCC basándose en la Figura 12 la cual representa la planta real.
4. Determinar las pérdidas de energía hidráulica.
5. Medir los parámetros necesarios para el cálculo de los coeficientes concernientes a las pérdidas de carga en conductos a presión.
6. Determinar el comportamiento hidráulico del conducto.

### **TRABAJO DE LABORATORIO**

La Figura 12 ilustra la planta de pérdidas de carga en conductos a presión. Esta planta básicamente es un circuito hidráulico en tubería de PVC que inicia con un suministro de agua dado por una motobomba o por el servicio de acueducto y que finaliza en un tanque de almacenamiento. En el circuito hidráulico se encuentran una serie de aditamentos y dos tramos de tubería, en los cuales se deben medir las pérdidas de carga de presión. Para ello, a lado y lado de cada aditamento o tramo, se han instalado tomas piezométricas que conducen la medida de presión, por medio de mangueras de polipropileno transparente, a un panel piezométrico dispuesto a un lado de la planta.

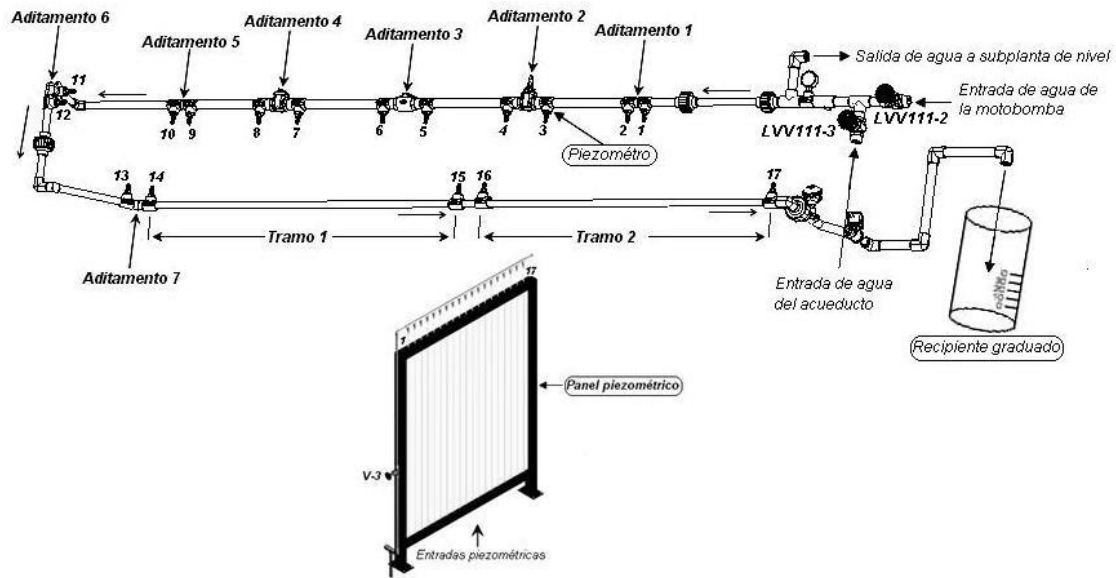


Figura 12. Planta de pérdidas de carga en conductos del LRPCC. Fuente: [5]

La planta anteriormente descrita tiene una instrumentación y un panel de control. La planta tiene instrumentos para medir temperatura y caudal en línea. Un armario de control donde por medio de una perilla manual se puede fijar el caudal de trabajo (0 % – 100 %) para el circuito hidráulico, un totalizador empleado para energizar el tablero de control, un medidor analógico de voltaje total del sistema y uno de corriente para medir el consumo de la motobomba, una fila de tres indicadores luminosos amarillos que nos indican si las electroválvulas están activas (iluminado = abiertas), una fila de cuatro indicadores luminosos para mostrar si las fases eléctricas y la puesta a tierra están activas (iluminado = activas) donde tres son de color verde para las fases y uno es amarillo para la puesta a tierra, una llave selectora para escoger ente control manual o automático, dos botones: de color rojo y verde para parar y encender la motobomba respectivamente y un PC donde se indica y visualiza las medidas de los instrumentos. Ver Figura 13 .

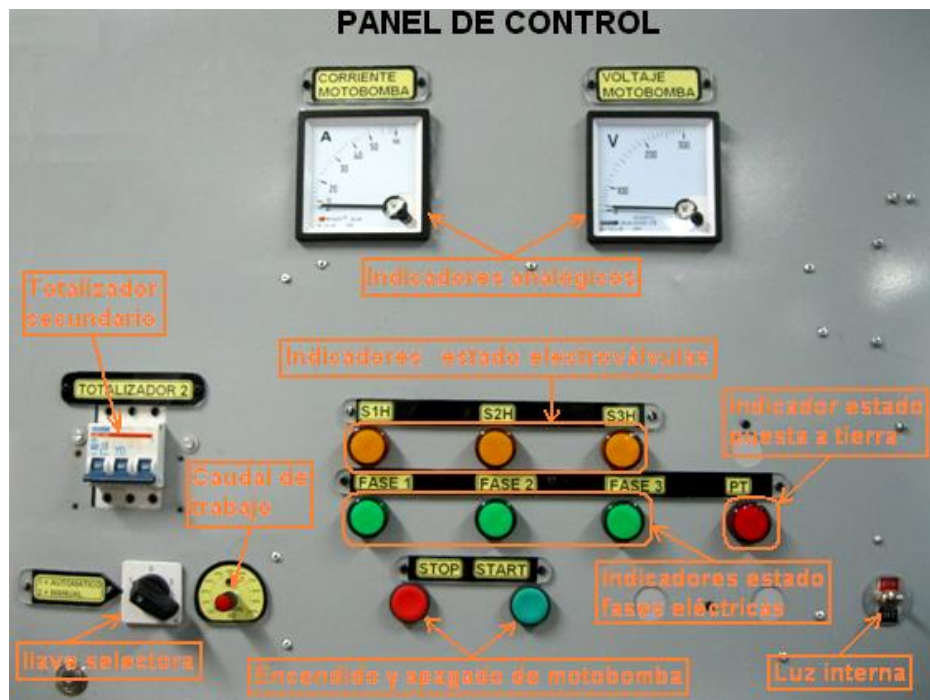


Figura 13. Panel de control. Fuente: [5]

### Establecer nivel de referencia.

Antes de iniciar la práctica de laboratorio, el laboratorista o un estudiante debe ubicar los piezómetros en un nivel de referencia, para ello debe seguir los pasos que se describen a continuación.

1. Ubicar el suiche en la posición 2 = manual.
2. Encender la bomba con el botón "Start".
3. Abrir la válvula ubicada a la derecha del tablero piezométrico.
4. Girar la perilla de control de caudal totalmente en el sentido de las manecillas del reloj.
5. Dejar circular agua por el circuito hidráulico hasta que dentro de los piezómetros ubicados en el tablero, se observe la menor cantidad de burbujas, apenas perceptibles al ojo.
6. Cerrar la válvula ubicada a la derecha del tablero piezométrico.



7. Girar la perilla de control de caudal totalmente en el sentido contrario a las manecillas del reloj.
8. Encender la bomba con el botón “Stop”.
9. Ubicar el suiche en la posición 0.
10. Ajustar los piezómetros a un nivel de referencia apropiado superior o igual a 40cm por encima de la base del tablero piezométrico.

**Advertencia:**

Por ningún motivo debe cruzarse entre la cámara y el tablero piezométrico durante el desarrollo de la práctica, esto conllevará a una obtención de resultados erróneos.

**Pasos a realizar por el laboratorista o el estudiante directamente en la planta del LRPCC:**

1. Debe haberse establecido con anterioridad un horario para utilizar la planta en prácticas remotas, y es al inicio de éste horario que el laboratorista u operador remoto, desarrollará los pasos del 1 al 21, y al finalizar éste horario, los pasos 32 y 33.
2. El laboratorista u operador debe encargarse de ubicar todos los piezómetros a un mismo nivel de referencia, el cual debe estar a la mitad del tablero piezométrico. Esto con el fin de observar los cambios en los niveles de los piezómetros dentro de los límites del tablero, puesto que niveles por debajo del tablero generan imprecisiones y dificultades en la medida.
3. Inicialmente se debe hacer una identificación visual de los accesorios hidráulicos que componen la planta del LRPCC siguiendo la trayectoria del fluido a lo largo del circuito hidráulico, iniciando desde la succión de la motobomba hasta el desagüe al tanque 1 (*Tank 1*).
4. Identifique claramente los elementos de control manual y de visualización del estado de las fases y puesta a tierra del armario control, de las electroválvulas, del voltaje y corriente de la motobomba, cotejando la Figura 13 con el tablero de control real.
5. Cierre completamente la válvula LV111-3 (*sentido horario*) y abra completamente la válvula LV111-2 (*sentido anti horario*) lo que permite disponer el sistema para ser alimentado únicamente por la motobomba e

inhabilitar la entrada de agua del servicio de acueducto. Ver Figura 14 para recordar la ubicación de las válvulas mencionadas.

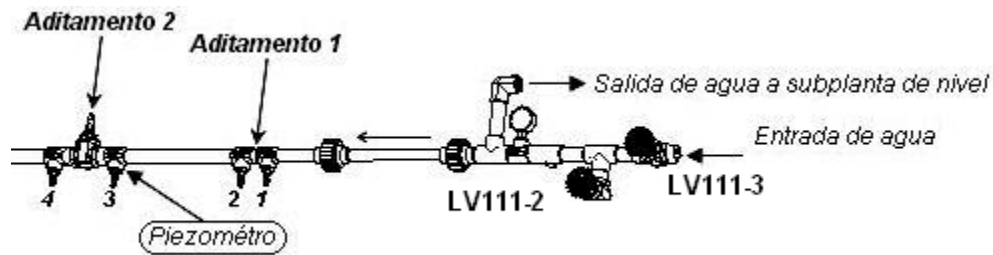


Figura 14. Detalle de válvulas

6. Energice el armario de control. Para esto se debe abrir la puerta del armario de control y accionar manualmente el totalizador 1 hacia arriba (*estado On*) para energizar la primera etapa. Posteriormente se debe cerrar la puerta del armario y continuar con el resto de la guía.
7. Encienda el PC, colocando en estado On el botón azul de “UPS” y posteriormente presionando el botón verde de “RESET”, ubíquese en la parte derecha de la bandeja retráctil, del armario de control, que soporta el teclado y el ratón del PC (Figura 15), y seleccione la opción RTAI rápidamente “*nuevo kernel compilado*” que se presenta en el primer menú en pantalla, ubicándose con la flecha de desplazamiento vertical del teclado, dar ENTER.

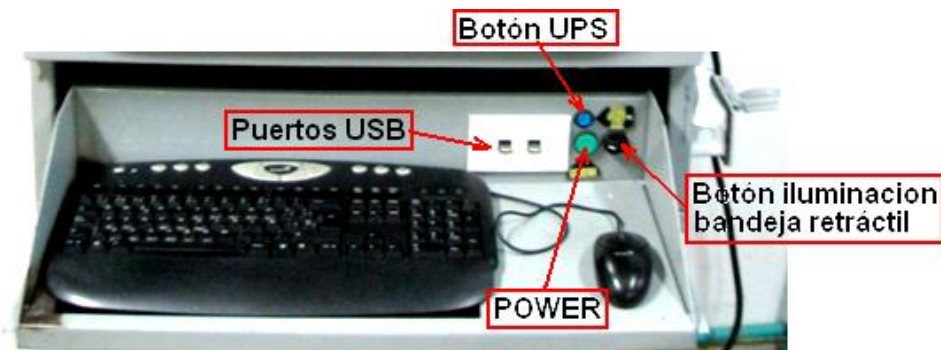


Figura 15. Botones y puertos de la bandeja retráctil. Fuente: [5]

8. Espere aproximadamente 40 segundos hasta que carguen la ventana de acceso que se muestra en la Figura 16, aquí debe oprimir el botón “Other”, de manera que se despliegue una casilla donde debe ingresar el nombre de usuario, escriba entonces “root”. Posteriormente se desplegará una ventana solicitando el la contraseña, escriba “unicauca”. Ver Figura 17.



Figura 16. Pantalla de inicio de Ubuntu solicitando nombre de usuario.

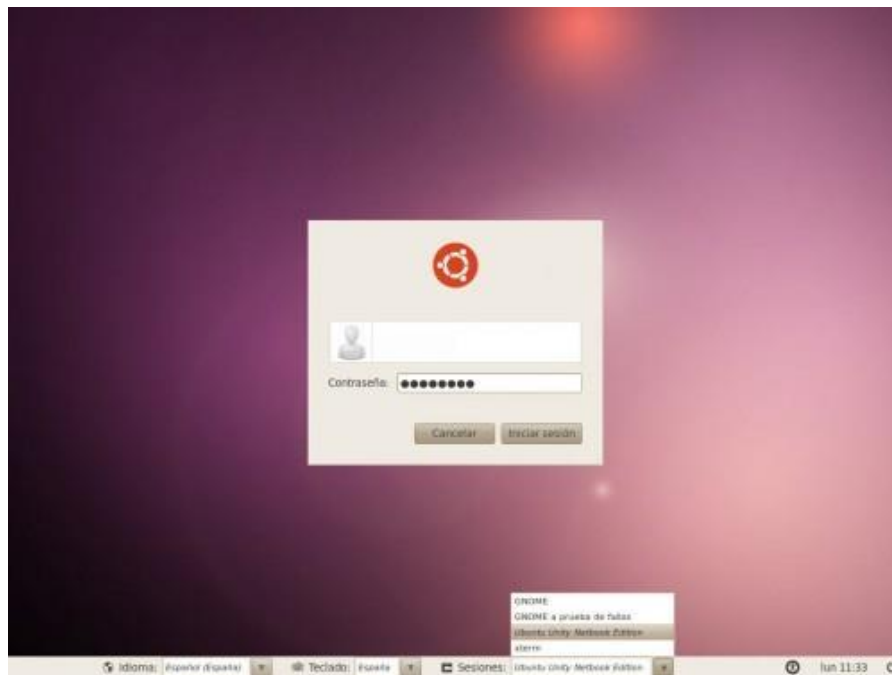


Figura 17. Pantalla de inicio de Ubuntu solicitando la contraseña de usuario.

9. Debemos tener en cuenta que para ejecutar la práctica se usan tareas de tiempo real que deben ser cargadas por módulos del RTAI. Para hacer esto, se debe situar en el escritorio de Ubuntu, en la barra superior debe acceder al menú “Aplicaciones”, allí encontrará una carpeta de accesos directos llamada RTAI, ingrese en ella y luego acceda a “Cargar Módulos RTAI”, ver Figura 18.

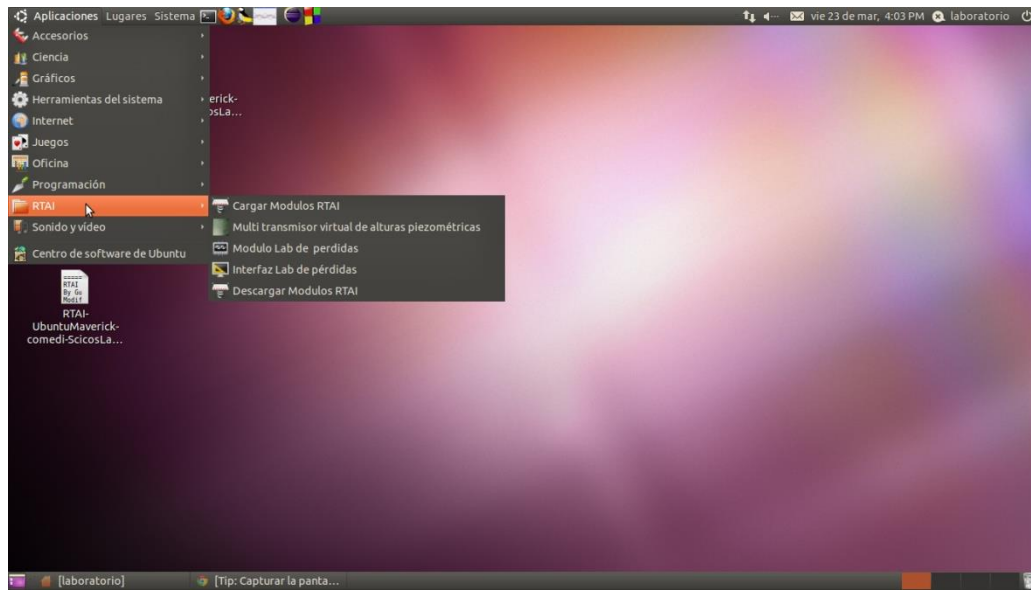


Figura 18. Acceso directo a Cargar Módulos RTAI.

10. Para descargar los módulos, en la barra superior y dentro de la carpeta RTAI acceder a “Descargar Módulos RTAI”, Ver Figura 19.

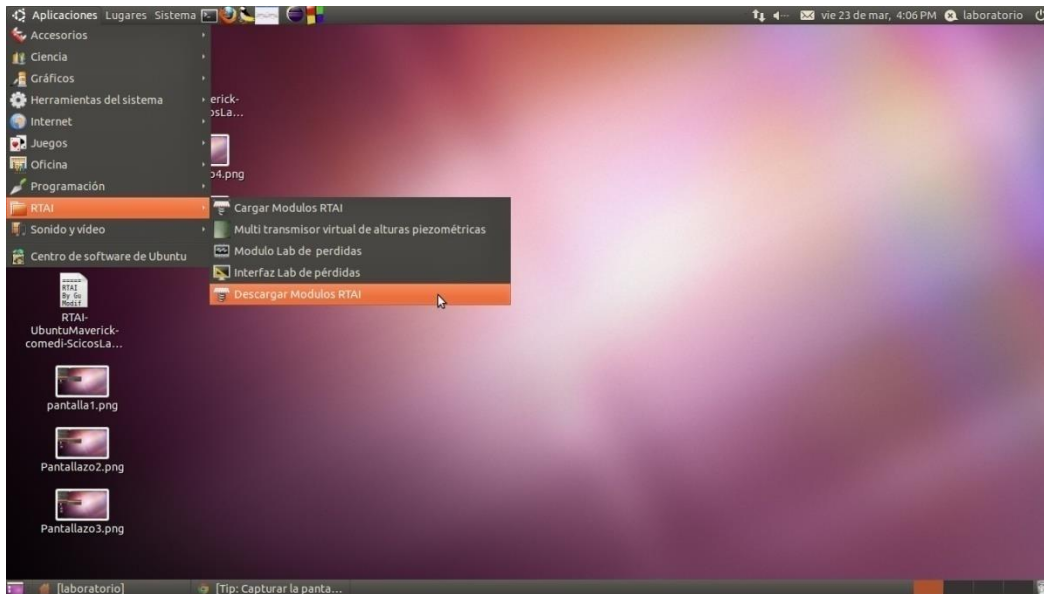


Figura 19. Acceso directo para descargar los módulos RTAI.

11. Ahora se debe ir nuevamente al tablero de control real (Figura 13) y realizar las siguientes acciones en estricto orden:

- Poner en estado On (*palanca hacia arriba*) el totalizador 2 para energizar el tablero de control.
- Ubicar la llave selectora en la posición 1 (*Automático*). Para disponer al sistema a funcionar en control automático.
- Verificar visualmente que los pilotos que indican los estados de las 3 líneas trifásicas *Fase 1*, *Fase 2* y *Fase 3* y del la puesta a tierra *PT* estén encendidos (*encendidos = activas*), como también que el indicador análogo de voltaje muestre entre 215 y 230 voltios AC, lo que corroboraría el buen funcionamiento del sistema, en caso contrario solicitar asistencia técnica.
- Verificar que los pilotos de estado de las tres electroválvulas estén apagados *S1H*, *S2H* y *S3H* lo que indica que dichas electroválvulas se encuentran cerradas (*desactivadas*), de lo contrario llamar al laboratorista para pedir asistencia técnica.
- Regresar a la interfaz del PC.

12. Nuevamente en el escritorio del Computador, en la barra superior, acceda a la carpeta RTAI y luego al acceso directo “Interfaz Lab de pérdidas”. Ver Figura 20.

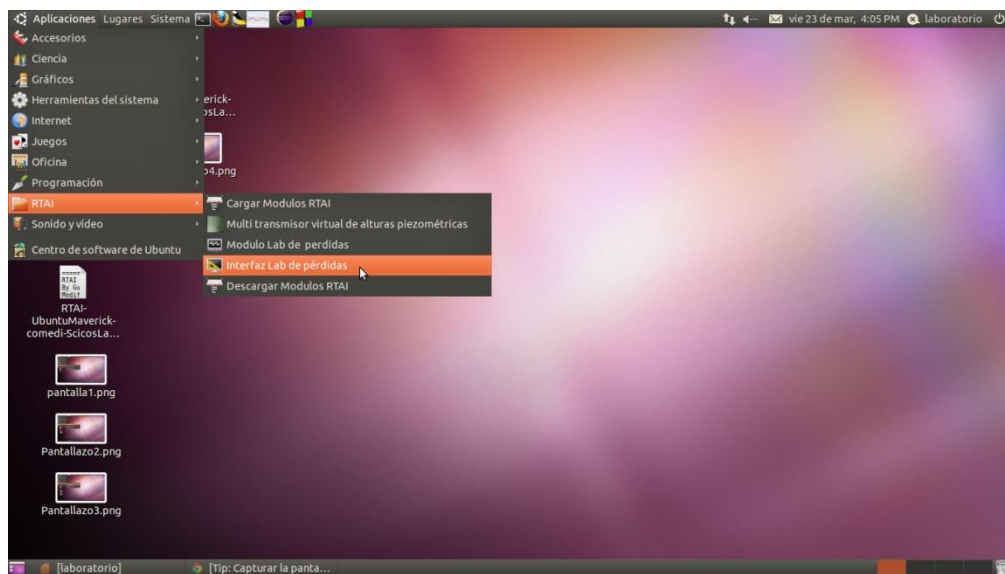


Figura 20. Acceso directo a la Interfaz de usuario del laboratorio de pérdidas de carga en conductos a presión.

13. como la que se muestra en la siguiente imagen, en la cual se debe dar clic en “ENTRAR” (*recuadro rojo*). Esta interfaz se debe dejar abierta hasta terminar la práctica. Ver Figura 21.



Figura 21. Ventana de entrada a la Interfaz de usuario.

14. En la ventana desplegada oprima el botón “Start” para iniciar la bomba, observará que la altura de los piezómetros se eleva, ingrese entonces en la ventana “caudal deseado” un caudal de 7.67 litros/minuto y oprima “Enviar”. Espere hasta que todos los piezómetros se estabilicen y oprima “Stop” para detener la bomba. No minimice ó cierre la interfaz.
15. Ahora nuevamente desde la carpeta RTAI, acceda a “Modulo de control de caudal en RTAI”. Ver Figura 22.

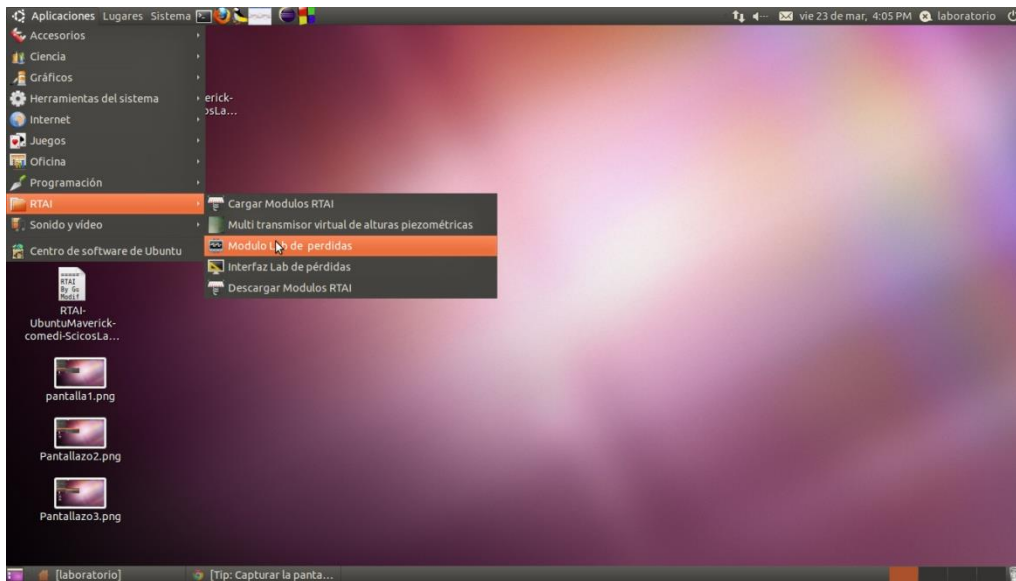


Figura 22. Acceso directo a Módulo de control de caudal en RTAI.

Se visualiza una terminal como en la Figura 23, en caso contrario repita éste paso. Minimice la terminal.

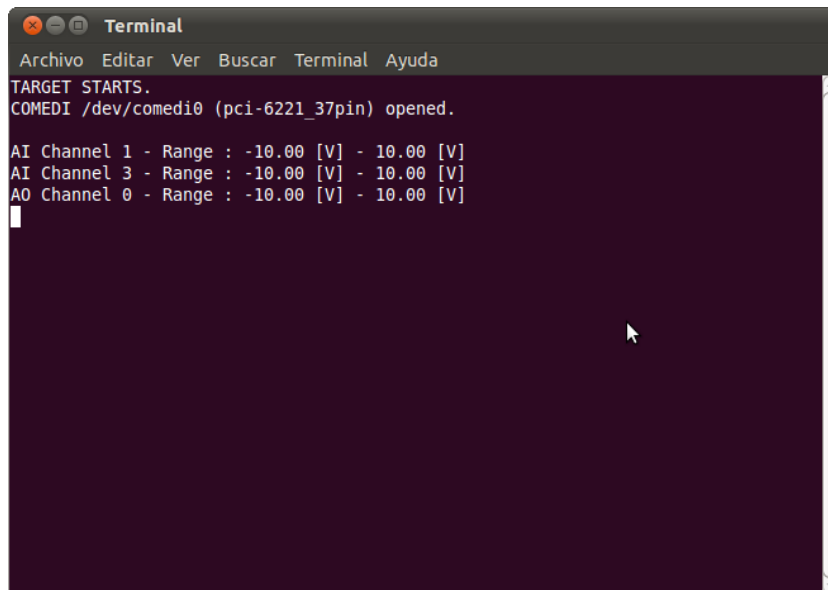


Figura 23. Terminal del modulo de control.

16. Nuevamente, en la barra superior, acceda a la carpeta RTAI y luego al acceso directo “Multi transmisor Virtual de alturas piezométricas”, ver Figura 24.



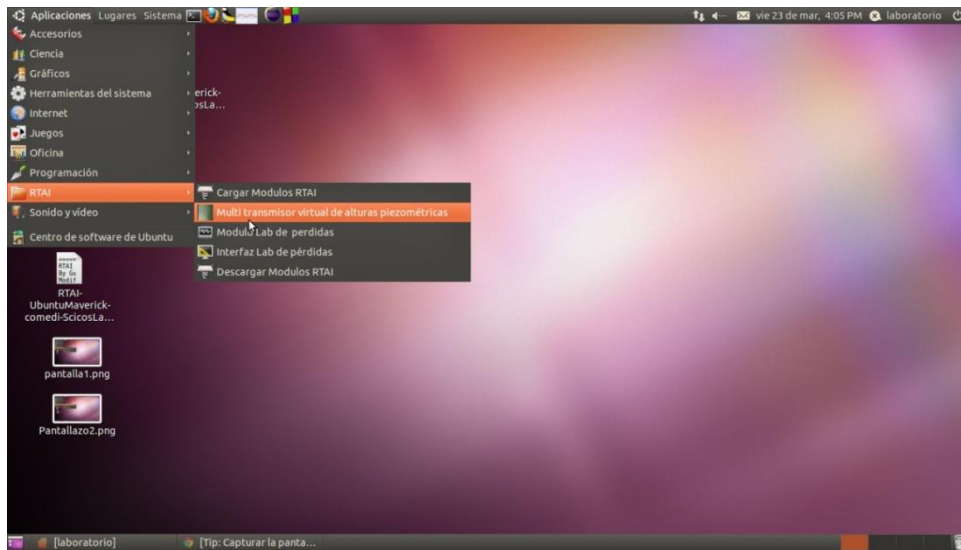


Figura 24. Acceso directo a Multi transmisor Virtual de alturas piezométricas.

Se abrirán dos terminales, una interfaz gráfica como en la y una imagen del tablero piezométrico. En ésta pueden configurarse los valores de brillo y contraste, no los modifique. Cierre ésta interfaz última presionando el botón "Salir". En caso de que no se despliegue la interfaz, omita éste párrafo.

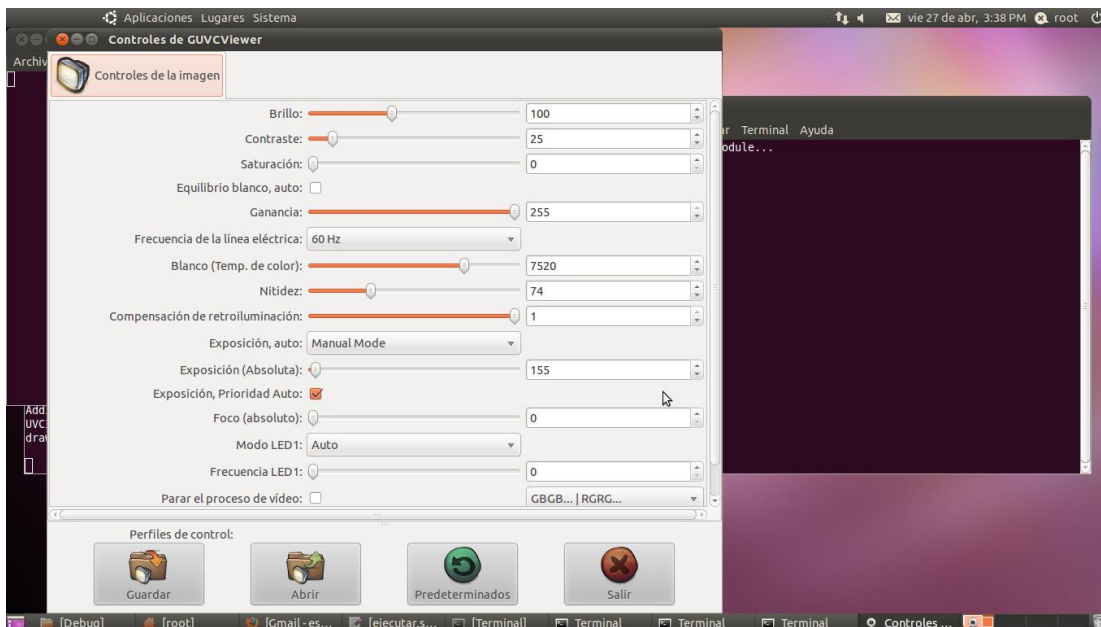
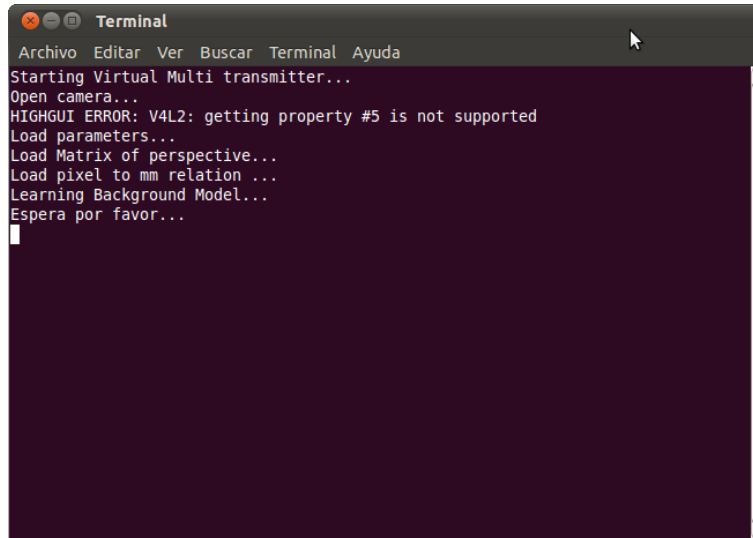


Figura 25. Interfaz de calibración de imagen.

En una de las terminales abiertas se despliega el mensaje "Espere por favor...", ver Figura 26Figura 28Figura 28, indicando que el Multi transmisor está



preparándose para iniciar la lectura de las alturas en el tablero piezométrico. Además se despliega otra terminal con una serie de instrucciones, ver Figura 27. Minimice ésta última.



```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
Starting Virtual Multi transmitter...
Open camera...
HIGHGUI ERROR: V4L2: getting property #5 is not supported
Load parameters...
Load Matrix of perspective...
Load pixel to mm relation ...
Learning Background Model...
Espera por favor...
```

Figura 26. Mensaje en terminal del Multi Transmisor Virtual.

**Nota:**

En el caso de que no se despliegue en la terminal el mensaje “Espere por favor...”. Cierre las dos últimas terminales abiertas y solicite comedidamente al laboratorista que verifique la conexión de la cámara en el PC, conectando y desconectando el terminal USB, de la cámara digita, ubicado en la parte posterior del PC en el armario de control.

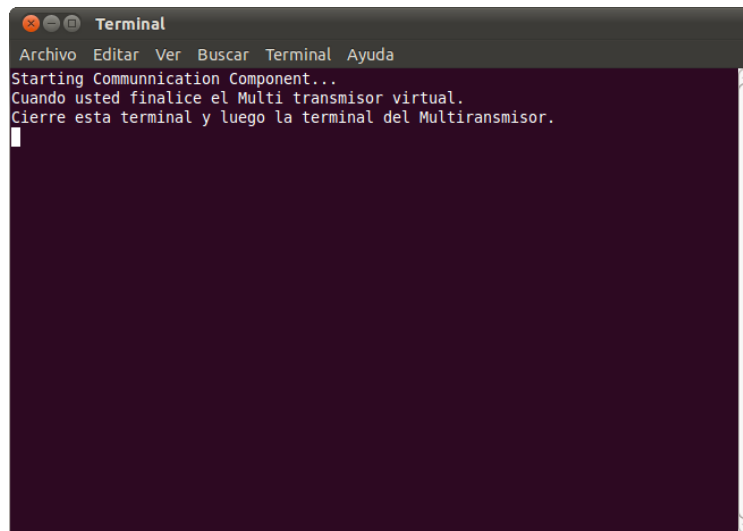


Figura 27. Terminal de instrucciones.

**Advertencia:**

Por ningún motivo debe cruzarse entre la cámara y el tablero piezométrico, esto conllevará a una obtención de resultados erróneos.

Cuando la terminal mostrada en la Figura 26 despliegue el mensaje "Finished Learning Background Model..." el Multi transmisor estará listo para operar.

17. Si realiza la práctica desde el Panel de control de la planta, en la interfaz de usuario del laboratorio ingrese ahora un caudal entre 22 litros/minuto y 7.67 litros/minuto, y oprima "enviar". Realice éste proceso para 5 caudales y repita los pasos del 17 al 19.
18. Desarrolle el paso 25 de ésta guía. A continuación acceda dentro de la interfaz gráfica, en la pestaña "Procesos", establezca un caudal y en la pestaña "Lecturas Piezométricas", observará los valores de las diferentes alturas de los piezómetros incluyendo la del piezómetro de referencia. Ver Figura 28. Registre éstos datos en la Tabla 2, la que se muestra más adelante



Figura 28. Ventana de interacción con el usuario.

19. Si se oprime el botón “Alturas piezométricas”, se desplegará una ventana donde podrán visualizarse las alturas piezométricas, según los cambios realizados en el caudal. Ver Figura 29.

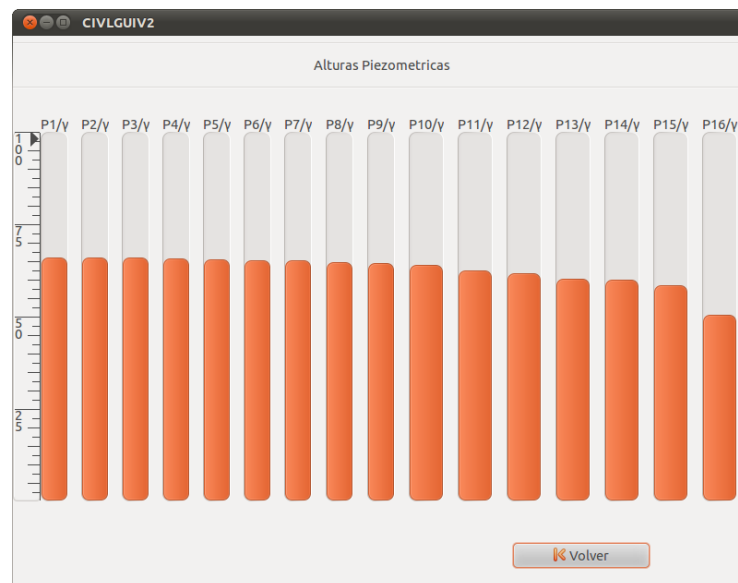


Figura 29. Visualización de las alturas piezométricas.

20. Oprima “Calcular pérdidas por fricción” se desplegará una ventana, ver Figura 30 , que permite realizar los cálculos de pérdidas por fricción por tramos.

Ingrese los valores de diámetro y longitud de los dos tramos largos del circuito hidráulico. Para hacer el calcular velocidad presione el botón “Calcular velocidad” y luego proceda de igual modo para realizar los demás cálculos de interés y consigne los datos en la Tabla 3 desplegados por la interfaz, para cada uno de los 5 caudales.

The screenshot shows a web application window titled "PERDIDAS POR FRICCION". The main heading is "PERDIDAS POR FRICCION - TRAMOS". Below the heading is a navigation menu with five tabs: "VELOCIDAD" (highlighted in orange), "PERDIDAS ENERGIA", "GRADIENTE PIEZOMETRICO", "GRADIENTE HIDRAULICO", and "NUMERO REYNOLDS". The main content area contains input fields for two pipe sections. For each section, there are fields for "DIAMETRO" (with units "m.m") and "LONGITUD" (with units "cms"). The current values in all fields are "0,00". Below the input fields, there are labels for "VELOCIDAD 1" and "VELOCIDAD 2" with units "cms/seg". To the right of the input fields is a large button labeled "CALCULAR VELOCIDAD". At the bottom right of the application window is a button labeled "VOLVER".

Figura 30. Cálculo de pérdidas por fricción.

**Los pasos del 20 al 34 deben desarrollarse si se desea hacer la práctica vía Web**

21. En el menú acceder a la carpeta RTAI y luego al acceso directo a “*Enlace hacia Servidor LRPCC*” para abrir la carpeta “*civil*”, como se indica en Figura 31.

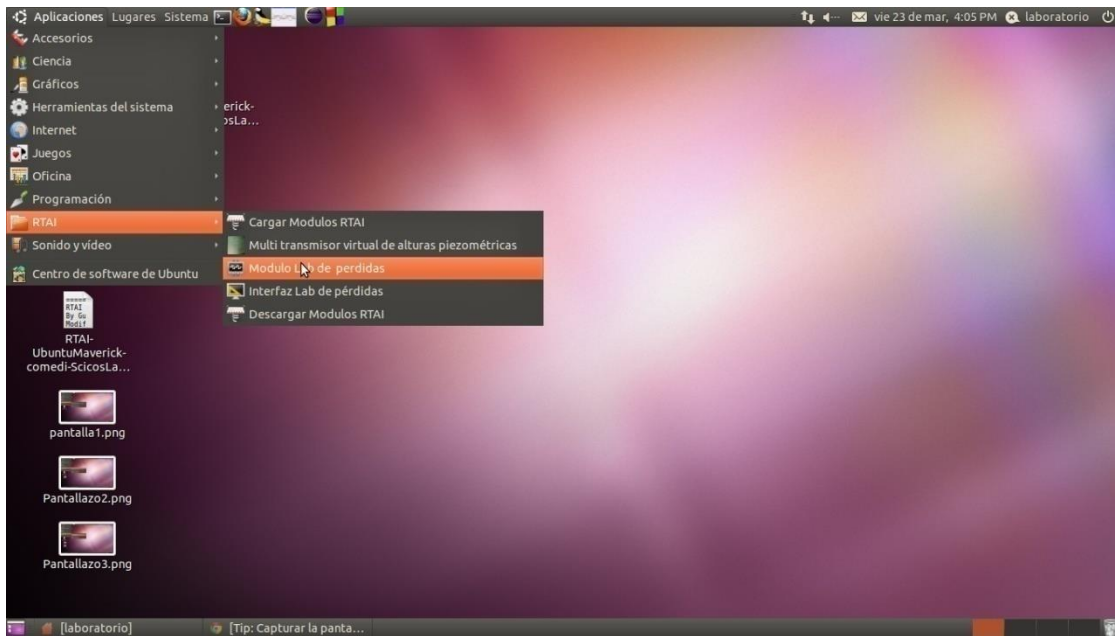


Figura 31. Acceso directo a Servidor LRPCC.

22. En este momento las personas quienes están a la espera para realizar la práctica remota, ya tienen vía libre para iniciar la conexión remotamente desde el PC cliente para que activen el navegador Web **Mozilla Fire Fox** e introducir el URL <http://192.168.190.24:8082/CIVIL.xhtml>, para enlazar al LRPCC, en cuyo caso si todo resulta bien entonces se debe obtener una página Web como la que se indica en la siguiente imagen, en la cual se debe introducir el nombre de usuario (“civil”), el password (“tesis2010”) y dar clic en “pass”. Si se desea realizar la práctica desde el mismo PC de control del LRPCC entonces basta con sustituir el URL anterior por <http://localhost:8082/CIVIL.xhtml> y escribir el mismo nombre de usuario y password, Ver Figura 32



## INGRESE SUS DATOS

usuario  Password:

Figura 32. Autenticación de usuario desde el Brrrowser

23. ahora y después de dar clic en el link “pass” de la página anterior se obtiene la siguiente página en la cual se puede seleccionar alguna según se desee de las 4 opciones (*recuadro rojo*) y visualizar la temperatura ( $^{\circ}\text{C}$ ), el caudal circulante por el circuito hidráulico ( $\text{cm}^3/\text{seg}$  y  $\text{Lit}/\text{min}$ ) y la viscosidad cinemática del agua ( $\text{cm}^2/\text{seg}$ ) en los displays (*recuadro verde*). Ver Figura 33.

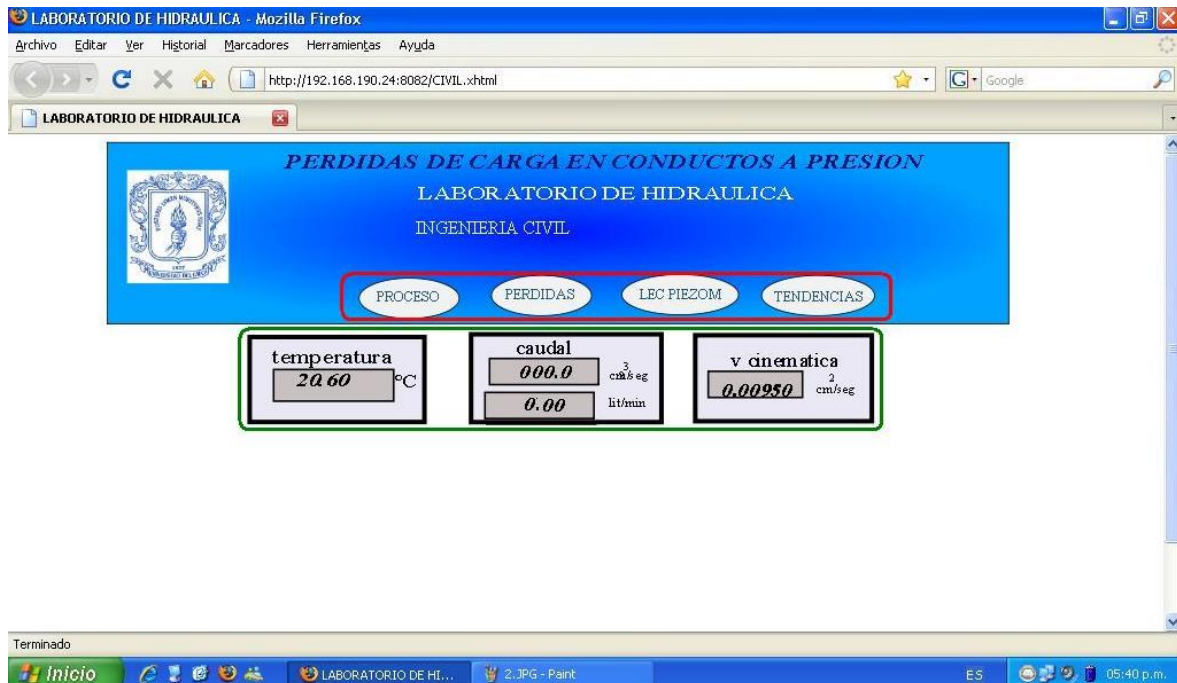


Figura 33. Página inicial del Servidor Web.

24. Se debe hacer clic en “PROCESO” y se obtiene una interfaz como la que se muestra en la siguiente imagen en la cual se debe establecer un caudal de trabajo (*recuadro rojo*) y dar clic en “Enviar”. Luego se debe poner en marcha

la motobomba dando clic en el botón verde de “start” (recuadro verde). Ver Figura 34.

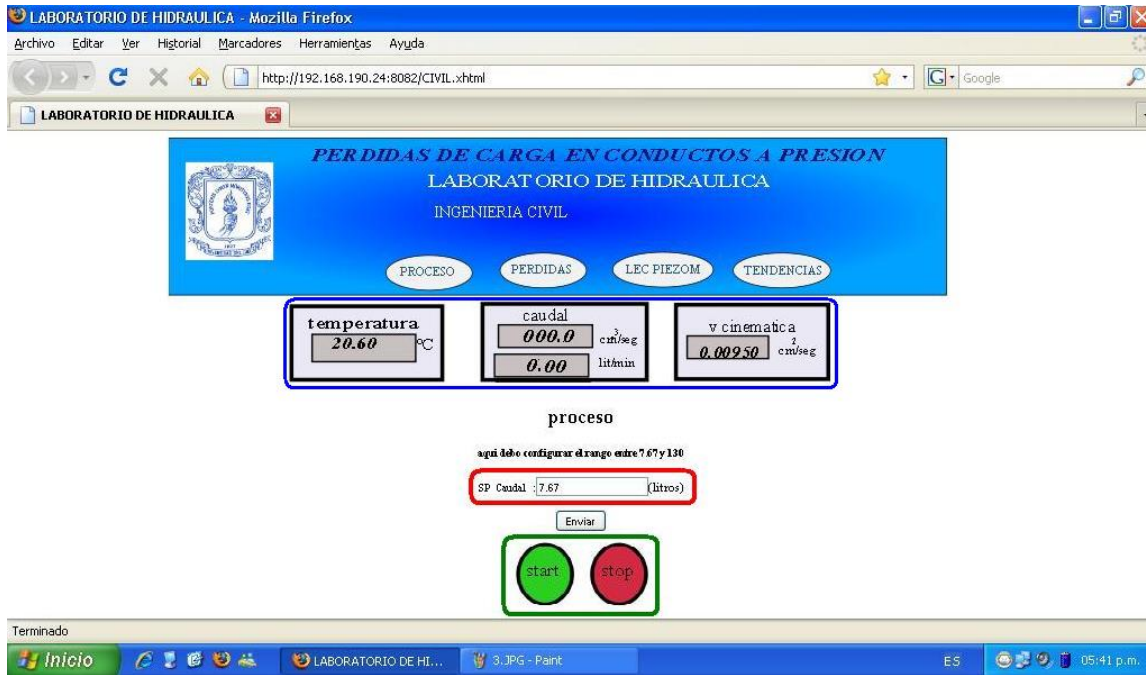


Figura 34. Página web de proceso.

25. De igual modo se deben anotar los valores de los diámetros internos, las longitudes de las tuberías de los tramos 1 y 2, los valores de los diámetros internos de los aditamentos previamente medidos, de la temperatura y la viscosidad cinemática visualizadas en los despliegues (recuadro azul de la Figura 34) en la Tabla 1 para la realización del respectivo informe.

Tabla 1. Características de aditamentos.

Datos básicos				
Temperatura $T$ (°C):				
Viscosidad cinemática $\nu$ (cm <sup>2</sup> /s):				
En aditamento ( <i>Perdida singular</i> )		En tramo recto ( <i>Perdida por fricción</i> )		
<p style="text-align: center;"><b>Ensamblamiento brusco</b></p>		<p style="text-align: center;"><b>Tramo recto</b></p>		
Tramo	Diámetro	Diámetro	Longitud	Tipo de

De piezómetro No.	A piezómetro No.	Interno de entrada (cm)	Interno de salida (cm)	d (cm)	Elemento en prueba
1	2	2.3	3.1	---	Ensanchamiento brusco
16	17	2.3	2.3	125	Tramo recto en PVC

26. Después de establecer un caudal deseado y poner en marcha la motobomba, las medidas de los niveles piezométricos observados en el panel de piezómetros se pueden observar en las casillas de la interfaz “*LEC PIEZOM*”, posteriormente se presiona el link “*Ver alturas*” para que se observen las alturas. Ver Figura 35.

### Lecturas piezométricas

H1 51.00 cms	H9 77.00 cms
H2 51.00 cms	H10 73.00 cms
H3 51.00 cms	H11 79.00 cms
H4 56.00 cms	H12 79.00 cms
H5 72.00 cms	H13 81.00 cms
H6 62.00 cms	H14 81.00 cms
H7 68.00 cms	H15 83.00 cms
H8 74.00 cms	H16 83.00 cms
H17 82.00 cms	

Figura 35. Página web para visualización de las medidas de las alturas piezométricas.

27. Se deben registrar los valores de las alturas piezométricas en la Tabla 2 que se muestra a continuación como también el caudal de trabajo establecido en el paso 24.



Tabla 2. Consignación de datos de alturas piezométricas.

Lecturas Piezométricas																		
Q (cm <sup>3</sup> /s )	$\frac{P_1}{\gamma}$ (cm )	$\frac{P_2}{\gamma}$ (cm )	$\frac{P_3}{\gamma}$ (cm )	$\frac{P_4}{\gamma}$ (cm )	$\frac{P_5}{\gamma}$ (cm )	$\frac{P_6}{\gamma}$ (cm )	$\frac{P_7}{\gamma}$ (cm )	$\frac{P_8}{\gamma}$ (cm )	$\frac{P_9}{\gamma}$ (cm )	$\frac{P_{10}}{\gamma}$ (cm )	$\frac{P_{11}}{\gamma}$ (cm )	$\frac{P_{12}}{\gamma}$ (cm )	$\frac{P_{13}}{\gamma}$ (cm )	$\frac{P_{14}}{\gamma}$ (cm )	$\frac{P_{15}}{\gamma}$ (cm )	$\frac{P_{16}}{\gamma}$ (cm )	$\frac{P_{17}}{\gamma}$ (cm )	

28. Después del paso anterior se debe ir a la opción “*PERDIDAS/perdidas por fricción*” e ingresar los valores de los diámetros internos de las tuberías y su longitud (*recuadro rojo*) de acuerdo a la Tabla 1, para luego presionar “*Calcular*” (*recuadro verde*) para obtener el resultado de las diferentes variables que serán calculadas por el software y que se pueden visualizar dando clic sobre cada link (*recuadro azul*). Ver Figura 36. Cada una de estas variables deben ser registradas en la Tabla 3y las que no, deben ser calculadas manualmente.

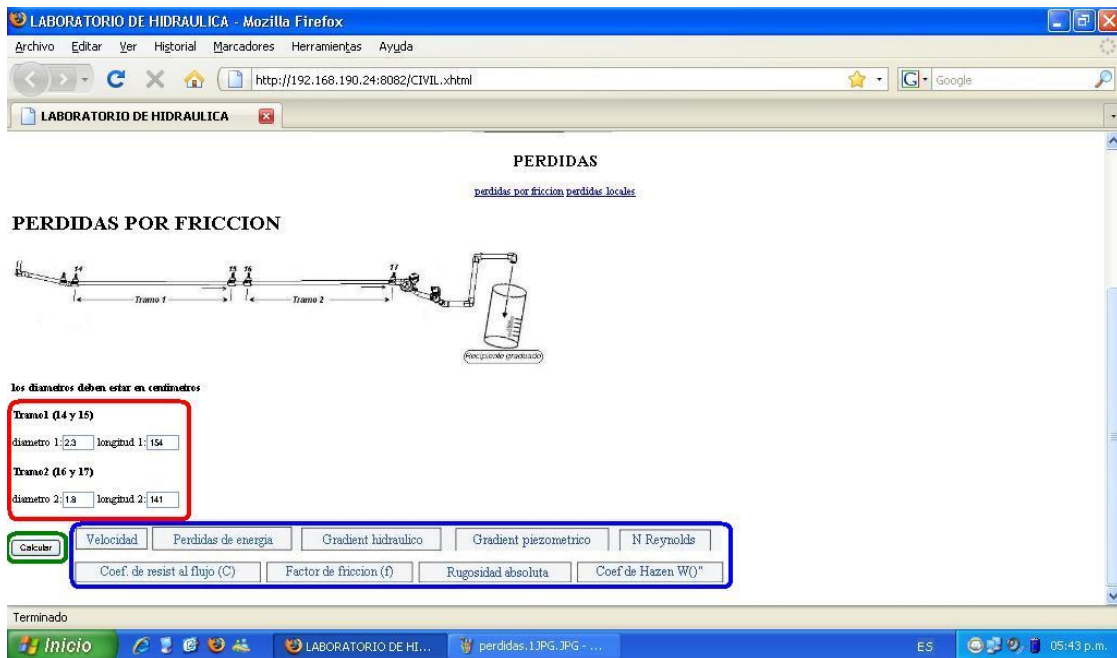


Figura 36. Página web para cálculo de pérdidas por fricción.

Tabla 3. Cálculo de coeficientes de pérdidas por fricción.

<b>Cálculo de coeficientes de pérdidas por fricción</b>																
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Tramo:				Material:						Diámetro int. :						
$Q$ $cm^3/s$	$\frac{P_i}{\gamma}$ $cm$	$\frac{P_j}{\gamma}$ $cm$	$V$ $cm/s$	$H_i$ $cm$	$H_j$ $cm$	$h_f$ $cm$	$S_f$	$Re$	$C$ Chezy $cm^{1/2}/s$	$f$ D- W	$\epsilon$ C-W ( $cm$ )	$\delta_o$ ( $cm$ )	$a$ ( $cm$ )	$\epsilon$ Ec. Log ( $cm$ )	$C_{H-W}$	

$C$ -Chezy promedio:		Análisis:
$f$ promedio:		Análisis:
$\epsilon$ Ec. Log promedio:		Análisis:
$C_{H-W}$ promedio:		Análisis:

Tramo:				Material:						Diámetro int. :						
$Q$ $cm^3/s$	$\frac{P_i}{\gamma}$ $cm$	$\frac{P_j}{\gamma}$ $cm$	$V$ $cm/s$	$H_i$ $cm$	$H_j$ $cm$	$h_f$ $cm$	$S_f$	$Re$	$C$ Chezy $cm^{1/2}/s$	$f$ D- W	$\epsilon$ C-W ( $cm$ )	$\delta_o$ ( $cm$ )	$a$ ( $cm$ )	$\epsilon$ Ec. Log ( $cm$ )	$C_{H-W}$	

$C$ -Chezy promedio:		Análisis:
$f$ promedio:		Análisis:

Tramo:		Material:										Diámetro int. :			
$Q$ $cm^3/s$	$\frac{P_i}{\gamma}$ $cm$	$\frac{P_j}{\gamma}$ $cm$	$V$ $cm/s$	$H_i$ $cm$	$H_j$ $cm$	$h_f$ $cm$	$S_f$	$Re$	$C$ Chezy $cm^{1/2}/s$	$f$ $D-W$	$\epsilon$ $C-W$ $(cm)$	$\delta_o$ $(cm)$	$a$ $(cm)$	$\epsilon$ $Ec. Log$ $(cm)$	$C_{H-W}$
$\epsilon$ Ec. Log promedio:							Análisis:								
$C_{H-W}$ promedio:							Análisis:								

28. Ahora ingrese en la opción “*PERDIDAS/perdidas locales*”, ver Figura 37, y registre los datos (previamente anotados en la Tabla 3) que se piden en la parte izquierda (recuadro rojo) para luego dar clic en “*calcular*” y así obtener los resultados en la parte derecha (recuadro verde) como se muestra en la siguiente imagen. Cada una de estos resultados deben ser registradas en la

- 29.
- 30.
- 31.

32. Tabla 4y los que no, deben ser calculados manualmente.

DIÁMETRO (Centímetros)	LONGITUDES IGUALMENTE (METROS)	Coeficiente local
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

Figura 37. Página web para cálculo de coeficientes de pérdidas locales.

Tabla 4. Cálculo de coeficientes de pérdidas locales.

Cálculo de coeficientes de pérdidas locales										
Accesorio:					Material:					
Q cm <sup>3</sup> /s	$\frac{P_i}{\gamma}$ cm	$\frac{P_j}{\gamma}$ cm	V <sub>i</sub> cm/s	V <sub>j</sub> cm/s	H <sub>i</sub> cm	H <sub>j</sub> cm	h <sub>l</sub> cm	K	Le Hierro Galvanizado cm	Le PVC cm
<b>K promedio:</b>					<b>Análisis:</b>					
<b>Le HG promedia:</b>					<b>Análisis:</b>					
<b>Le PVC promedia:</b>					<b>Análisis:</b>					

Nota: Es necesario adicionar en el informe 6 tablas más como la Tabla 3 para completar los 7 accesorios de la planta.

33. Por último se debe entrar en la opción “TENDENCIAS” donde se puede observar dos graficas: en la primera, ver Figura 38, “TENDENCIAS/caudal” se puede observar el comportamiento del caudal Q (cm<sup>3</sup>/seg) circulante en la tubería, mientras que en “TENDENCIAS/f vs Reynolds”, ver Figura 39, se pueden apreciar dos graficas, la de la parte superior para el coeficiente de fricción  $f_1$  contra el numero de Reynolds y la de la parte inferior para el  $f_2$  contra el numero de Reynolds correspondientes a cada uno de los tramos con las que se puede hacer un análisis rápido del tipo de tubería evaluada. Ver las siguientes figuras.

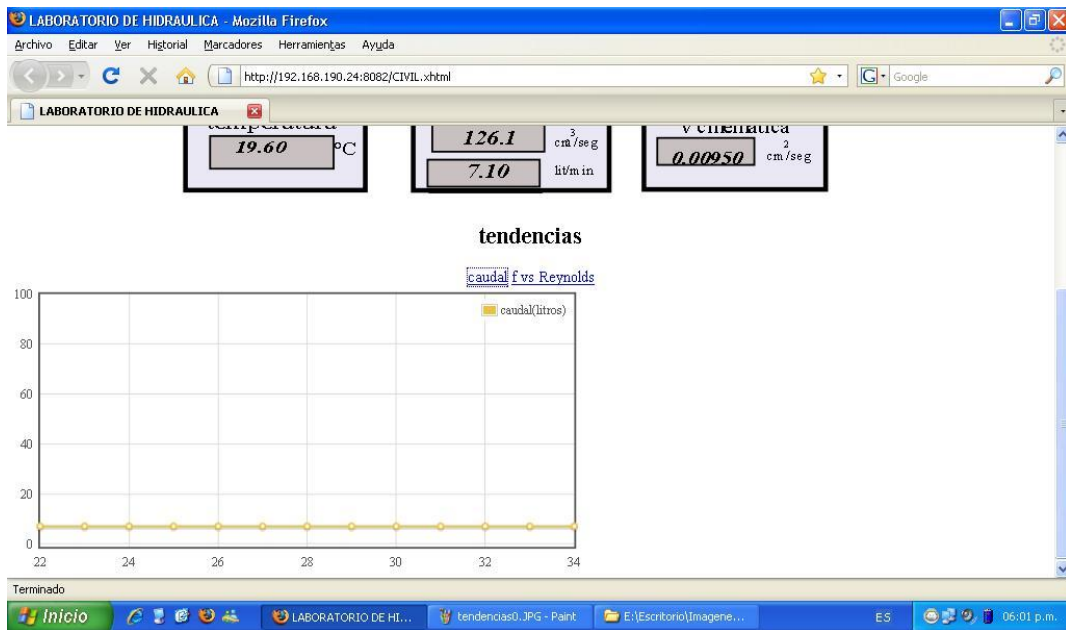


Figura 38. Página web para despliegue de Tendencias Vs Caudal.

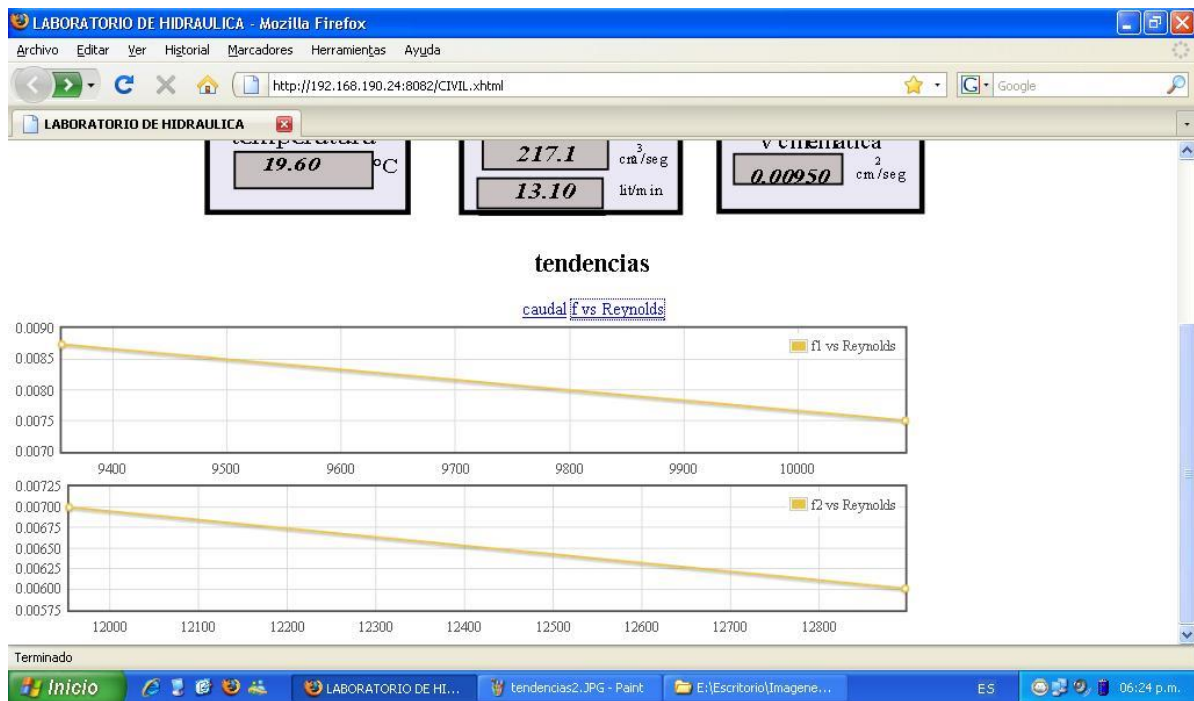


Figura 39. Página web para despliegue de Tendencias Vs Reynolds

34. El mismo procedimiento a partir del paso 21 se debe repetir para 5 diferentes caudales de prueba y así poder realizar una práctica de laboratorio idónea.

35. Al terminar la práctica y después de haber registrado los datos para realizar el correspondiente informe se debe abrir la opción “PROCESO” y hacer clic en el botón rojo de “stop” para detener la. Igualmente se deben cerrar el navegador Web si así se desea. Y se deben cerrar todas las ventanas.
36. Terminado el horario establecido para la realización de prácticas remotas, el laboratorista puede finalizar las tareas de control, esto debe hacerlo cerrando cada uno de los “*TERMINALES*” abiertos anteriormente y tecleando la combinación de teclas Ctrl+c en cada uno de estos para finalizarlos correctamente. Posteriormente debe cerrar cada una de las ventanas abiertas (*x en la esquina superior izquierda*).
37. Finalmente, el laboratorista debe apagar el PC con el uso del *mouse* como se muestra en la siguiente imagen o teniendo presionado el botón “POWER” de la bandeja retráctil. Por último se debe retornar el totalizador 2 del tablero de control al estado “Off” como también el totalizador 1 del armario de control.
38. Con los datos registrados en esta guía, se debe proceder a elaborar el informe final en el plazo establecido por el profesor.

### **Elaboración del Informe:**

1. Calcule para cada tramo recto y para cada caudal de ensayo.
  - a) El coeficiente  $a$  de comportamiento hidráulico de la ecuación logarítmica.
  - b) El espesor de la subcapa laminar viscosa  $\delta_o$ .
  - c) La rugosidad absoluta  $\varepsilon$  del conducto a partir de la ecuación logarítmica.
  - d) Analice los valores de rugosidad absoluta obtenidos por la ecuación de Colebrook y White (C-W) y por la ecuación logarítmica.
2. Con base en la pérdida local calculada para cada caudal, determine para cada aditamento la longitud equivalente  $L_e$ , especificando diámetro y material.
3. Con los resultados anteriores, dibuje en papel semilogarítmico la curva  $f$  vs  $Re$  para cada diámetro y material. (*Re en la escala logarítmica*). Puede usar el diagrama de *MOODY* y sobre él ubicar los resultados para facilitar el análisis.

4. Resuma los resultados en las Tablas XIII A. 3 y XIII A. 4, según corresponda.
5. Dibuje el tramo de conducción usado y la línea de alturas piezométricas correspondiente para el caudal mayor.
6. Observaciones.
7. Conclusiones.

## ANEXO C: TÉCNICAS DE PROCESAMIENTO DE IMÁGENES ALTERNATIVOS

A continuación se hace una descripción de algoritmos opcionales que fueron probados pero los resultados obtenidos no fueron satisfactorios para dar solución al problema planteado.

### Búsqueda de patrones (Template Matching)

La búsqueda por patrones Template Matching se basa en algoritmos de correspondencia pixel a pixel, encuentra la correlación entre las imágenes que es comparada con un umbral definido, es así que se puede buscar un objeto dentro de una escena sin depender de la orientación del mismo.

La técnica del Template matching ó búsqueda de patrones aunque dio resultados positivos en el reconocimiento de gran parte de los niveles en las columnas piezométricas, precisa de un número de plantillas considerable, de acuerdo a las condiciones de luminosidad presentes en el lugar, y que varían durante todo el día, debido a la alta incidencia de la luz solar.



Figura 40. (a) Imagen original, (b) Resultado de búsqueda por plantilla (Template Matching).



En la Figura 40.b, se observan las 16 columnas piezométricas horizontalmente, donde la columna 1 está en la parte inferior y la 16 en la parte superior. Se advierte que en las columnas piezométricas 2 a la 8, la coincidencia es bastante cercana al lugar donde se presenta la interfaz aire – agua, sin embargo en las columnas 9 y 10 se presentan dos valores, de los cuales uno es falso, lo que se evidencia observando la foto del tablero a la izquierda, y teniendo en cuenta la configuración del circuito hidráulico donde están ubicados los piezómetro, de donde se hace necesario plantear una serie de condiciones ó ecuación para validar los datos y detectar los errores en el Template Matching.

Respecto a las columnas 11 a la 16, debido a las condiciones de luz en el lugar donde se encuentra dispuesto el tablero piezométrico. No es posible utilizar la misma plantilla (Template) utilizada con las columnas 1 – 10, dado que ésta zona presenta una baja luminosidad, por tanto la plantilla debe ser consecuente con esto.

## ANEXO D: DIAGRAMA DE MOODY.

En la Figura 41 se muestra una imagen ampliada del diagrama de Moody mencionado en el capítulo 1.

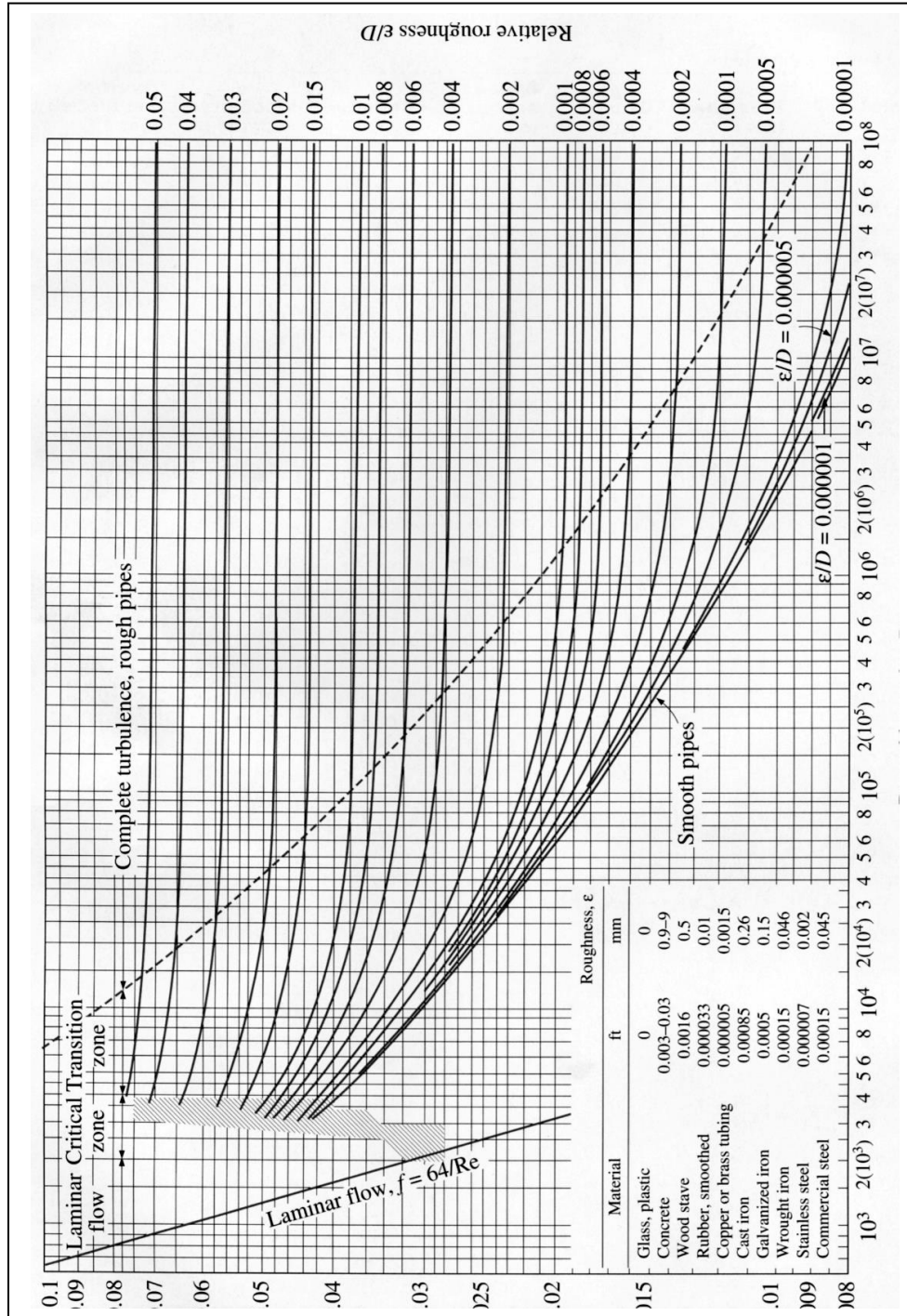


Figura 41. Diagrama de Moody ampliado. Tomado de de [6]

## ANEXO E. MANUAL DEL USUARIO DE LA INSTALACIÓN DEL SISTEMA SOFTWARE DE SOPORTE DEL MULTI TRANSMISOR VIRTUAL

Para que el Multi transmisor Virtual basado en visión por computador pueda ser ejecutado, se requiere un sistema operativo Linux en su distribución Ubuntu, y la instalación, de una serie de herramientas software mencionadas en los Capítulos 1 y 2, como se describirá a continuación:

### Prerrequisitos:

15 G de espacio en Disco Duro.

Ubuntu 10.10

OpeCV 2.2 o superior

### Instalación del Multi transmisor Virtual

Dentro de CD suministrado con la monografía del proyecto de grado, se encuentra un archivo comprimido nombrado: “MultiTxVirtual.tar.gz”, en él se encuentran todos y cada uno de los archivos necesarios para la ejecución del Multi transmisor Virtual, así como para la realización de la práctica del laboratorio de pérdidas de carga en conductos a presión.

El archivo debe descomprimirse dentro del directorio /home. De modo que se obtenga una estructura de directorios como se muestra a continuación en la Figura 42:

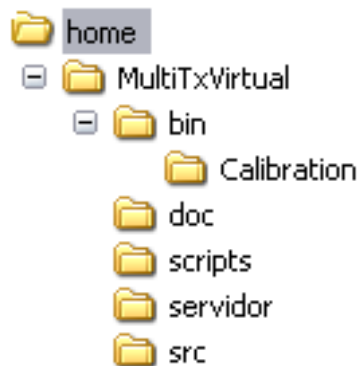


Figura 42. Estructura de directorios

La carpeta “scripts” contiene los scripts para la carga y descarga de los módulos de tiempo real. Dentro de “bin” se encuentran los archivos binarios para ejecutar la interfaz de usuario del laboratorio LRPCC, la interfaz de usuario de

configuración del Multi transmisor virtual, el módulo de control de tiempo real de la planta LRPC y el Multi transmisor virtual.

Adicionalmente hay una carpeta “scr”, donde pueden hallarse los códigos fuente desarrollados en C/C++ para producir cada de los archivos binarios encontrados dentro de la carpeta “bin”. A excepción del código fuente del módulo de tiempo real, todos tienen la estructura de archivos de un proyecto desarrollado con el IDE CodeBlocks; además de las fuentes en lenguaje C para el modulo de tiempo real se tiene el diagrama civilbox.cos implementado con el programa Scicos.

### Creación de lanzadores:

Además deben crearse los enlaces directos ó lanzadores, para acceder a las aplicaciones ubicadas en “bin”. Para crear un acceso directo en los menús, se ingresa en en el barra superior del menú aplicaciones: Sistema → Preferencias → Menú principal. Se abrirá la ventana mostrada en la Figura 45.



Figura 43. Distribución de menú.

Primero se creará una carpeta que contendrá todos los accesos directos. Para esto pulse el botón “Menú nuevo”, ver Figura 44. En la nueva ventana de

propiedades del directorio añada la información referente a la carpeta que quiere crear. En Nombre escriba “LRPCC” y en comentarios: “Laboratorio de pérdidas de carga en conductos a presión”. Para terminar presione “Aceptar”.

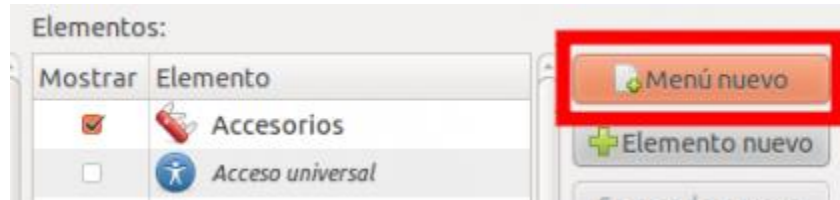


Figura 44. Crear Menú nuevo.

Ahora nuevamente ubicado en la ventana de distribución de menús, Figura 43 . Haga doble click izquierdo de ratón en el menú recién creado: “LRPCC”. Para crear cada uno de los lanzadores ó accesos directos de las aplicaciones en la carpeta “bin”. Proceda como se indica a continuación.

Estando ubicado en el menú “LRPCC” en la ventana distribución de menús. Oprima “Elemento nuevo”, aparecerá un pequeño cuadro de diálogo, ver Figura 45 , con un pequeño formulario:

- Tipo: Se indica si es una aplicación gráfica, aplicación de consola, o un archivo y ya se encargará el sistema de abrirlo con el programa que proceda.
- Nombre: Es el nombre que tendrá el lanzador.
- Comando/Lugar: Comando si es una aplicación, se le dice el nombre del ejecutable. Si es un ejecutable de sistema no se tiene que indicar nada más que el nombre. En caso contrario se tendrá que escribir la ruta completa, ayúdese del botón “Examinar...”. En el caso de un archivo se le indicará el lugar del archivo.
- Comentario: Una pequeña descripción del lanzador si se quiere.
- Sin icono: Al oprimir éste botón se podrá seleccionar una imagen para asignarla como icono del lanzador.

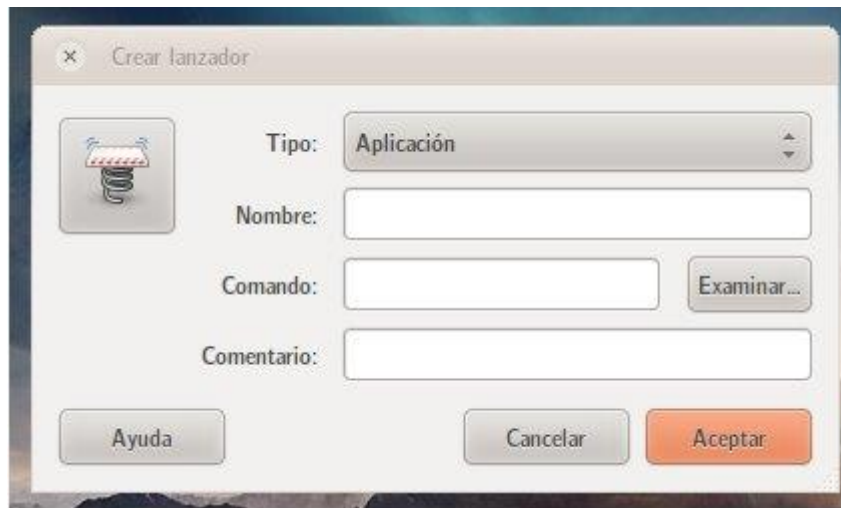


Figura 45. Propiedades del lanzador

Ingrese la información de la Tabla 5, para crear cada uno de los lanzadores correspondientes a los archivos binarios de la carpeta “bin”. Tal y como se indicó anteriormente.

Tabla 5. Detalle de los lanzadores.

Nombre	Tipo	Comando
<b>Carga modulos RTAI</b>	<i>en terminal</i>	<i>sudo /home/MultiTxVirtual/scripts/rtailab start -H</i>
<b>Modulo Lab de perdidas</b>	<i>aplicacion</i>	<i>/home/MultiTxVirtual/bin/ejecutar_tarea</i>
<b>Interfaz Lab de perdidas</b>	<i>aplicacion</i>	<i>/home/MultiTxVirtual/bin/CIVLGUIV2</i>
<b>Servidor LRPCC</b>	<i>aplicacion</i>	<i>/home/MultiTxVirtual/bin/ejecutar_servidor</i>
<b>Setup Multitransmisor Virtual</b>	<i>aplicacion</i>	<i>sh /home/MultiTxVirtual/bin/ejecutar3.sh</i>
<b>Multitransmisor Virtual</b>	<i>terminal</i>	<i>sh /home/MultiTxVirtual/bin/ejecutar.sh</i>
<b>Finalizar Servidor LRPCC</b>	<i>terminal</i>	<i>sudo sh /home/servidor/civil/finalizar.sh</i>
<b>Descargar Modulos RTAI</b>	<i>en terminal</i>	<i>sudo /home/MultiTxVirtual/scripts/rtailab stop</i>

---

---

Para terminar, cierre la ventana de distribución de menùs, oprimiendo "cerrar", ver Figura 43, en la esquina inferior derecha.

## Instalación de la plataforma de RTAI

### 1. Compilado e instalación del Kernel

- **instalar los paquetes:** Se deben instalar los paquetes libncurses5-dev, cvs, build-essential, kernel-package. En una terminal de comandos se teclea:

```
$ sudo apt-get install libncurses-5-dev, cvs, build-essential, kernel-package
```

- **Descargar del kernel de Linux:** la versión 2.6.35.7 desde <http://kernel.org/pub/linux/kernel/v2.6/linux-2.6.35.7.tar.bz2>, copie los archivos descargados en /usr/src

```
cd /usr/src
wget -P /tmp http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.34.5.tar.bz2
tar xjf /tmp/linux-2.6.34.5.tar.bz2
mv linux-2.6.34.5 linux-2.6.34.5-rtai
ln -s linux-2.6.34.5-rtai linux
```

- **Descargar las fuentes RTAI:** Ubicado en `cd /usr/src`, descargue la versión en desarrollo.

```
cd /usr/src
$cvs -d:pserver:anonymous@cvs.gna.org:/cvs/rtai co magma
mv magma rtai-3.8.1
ln -s rtai-3.8.1 rtai
```

- **Aplicación del parche:** Ubíquese en /usr/src/Linux, Para aplicar el parche ejecute el comando patch

```
$cd /usr/src/linux
$sudo patch -p1 < /usr/src/rtai/base/arch/x86/patches/hal-linux-2.6.35.7-x86-2.8-01.patch
```

- **Configuración del kernel:** Para esto se hará la copia del archivo de configuración de kernel actual, y luego ejecute la interfaz de configuración con menuconfig.

```
$sudo cp /boot/config-`uname -r` .config
```



\$ sudo make menuconfig

- **Configurar Módulos:** Luego de copiar el archivo de configuración anterior, se deja tal cual, excepto:

*Enabled loadable module support*

→Module unloading→enabled

→Source checksum for all modules→enabled

→ Module versioning support → disabled

*Processor type and features*

→ Preemption Model →Preemptible Kernel (Low-Latency Desktop)

→Interrupt pipeline → enabled

→Timer frequency → 1000 Hz

*Para sistemas de un solo procesador (Un único procesador, maquinas con P3 o P4,etc ), desabilite el soporte, si tiene como es el caso de ésta guía, un Core 2 Duo, Quad, u otro, entonces dejelea habilitada.*

→Symmetric multiprocessing support→enabled

*Si la habilita asegurese de coloca en la opcion maximo numero de CPUs: 2*

→Maximum number of CPUs→2

*En familia de procesador seleccione la correspondiente a su hardware, aunque para la mayoría de los casos puede seleccionar 586/K5/5x86/6x86/6x86MX y así crear un kernel genérico.*

→Processor family→Core 2Duo (Xeon)

*Power management and ACPI options*

→ CPU Frequency scaling→CPU Frequency scaling→disabled

→ APM (Advanced Power Management) BIOS support →disabled

- **Compilar el kernel:** Ubicado en `/usr/src/linux` ejecute `make`.

`cd /usr/src/linux`

`sudo make-kpkg clean`

- **Instalación del kernel:**

`make-kpkg --rootcmd fakeroot --append-to-version -rtai38 --revision r1 --initrd kernel_source kernel_image kernel_headers`

```
sudo dpkg -i ../linux-source-2.6.34.5-rtai38_r1_i386.deb ../linux-headers-2.6.34.5-rtai38_r1_i386.deb ../linux-image-2.6.34.5-rtai38_r1_i386.deb
```

## 2. Instalar RTAI

- **Compilado e instalación de fuentes:** Compilar e instalar RTAI (De momento sin soporte a Comedi):

```
cd /usr/local/src/rtai
sudo make menuconfig
```

- **Configuración y ajuste:**

→General → (/usr/src/linux) Linux source tree →linux  
→Menu Machine (x86): adjust Number of CPUs (default = 2)

- **Compilar e instalar:** luego de ejecutar make y make install, Adicione '/usr/realtime/bin' a las variables de entorno en '/etc/environment':

```
sudo make
sudo make install
sudo sed -i 's/(PATH=\\)/\\1\\usr\\realtime\\bin:/' /etc/environment
```

## 3. Instalar Comedi

- Ubicado en `cd /usr/local/src`, descargue y descomprima el paquete comedi:

```
$sudo wget http://www.comedi.org/download/comedi-0.7.76.tar.gz
$sudo tar xvzf comedi-0.7.76.tar.gz
```

```
$cd /usr/local/src
$sudo wget http://www.comedi.org/download/comedilib-0.8.1.tar.gz
$sudo tar xvzf comedilib-0.8.1.tar.gz
```

- Ubicado en `cd /usr/local/src`, descargue y descomprima el paquete comedi\_calibrate:

```
$cd /usr/local/src
```

```
$sudo wget http://www.comedi.org/download/comedi_calibrate-1.tar.gz
$sudo tar xvzf comedi_calibrate-1.tar.gz
$cd comedi_calibrate-1
```

- Se instalan las dependencias necesarias para ejecutar Comedi, antes de instalar.

```
sudo apt-get install bison flex
```

- **Compilar e instalar Comedi:** Ubicado en /usr/src/local

```
cd /usr/src/local
$sudo apt-get install libboost-dev libboost-program-options-dev libgsl0-dev
$sudo apt-get install libtool automake
$cd /usr/local/src/comedi-0.7.76
```

```
$ sh autogen.sh
$ ./configure
$ make
$sudo make
$sudo make install
$sudo make dev
```

- Si tiene tarjetas DAQ pcmcia, ejecute en otro terminal:

```
$sudo su
$echo 'options comedi comedi_num_legacy_minors=4' > etc/modprobe.d/comedi
$exit
```

- **Compilado e instalación de comedilib:**

```
cd /usr/local/src/comedilib-0.8.1
sudo sh autogen.sh
sudo ./configure
sudo make
sudo make install
sudo make dev
sudo mkdir /usr/local/include/linux
```

- **Compilar e instalar Comedi-calibrate:** Instale las dependencias necesarias:

```
$sudo apt-get install libboost-dev libboost-program-options-dev libgsl0-dev
$cd /usr/local/src/comedi_calibrate-1
$sudo autoreconf -i
$sudo ./configure
```

- Dentro de `/usr/local/src/comedilib_calibrate-1`: Al configurar puede fallar y dar errores en `libboost-program-options`. En `/usr/lib`, liste todas las librerías `libboost`: Probablemente tenga `libboost_program_options-mt` en lugar de `libboost-program_options`. Se necesita entonces cambiar algunas referencias a ésta librería con el nombre correcto. En el archivo `comedi_calibrate/configure.ac`, busque la línea:

```
AC_CHECK_LIB([boost_program_options],[main],,AC_MSG_ERROR([Failed to
find libboost_program_options.]))
```

```
$ ls | grep libboost
$sudo gedit configure.ac
```

- Cambie las opciones del `boost_program_options` en el primer argumento `boost_program_options-mt`. También necesita editar `/comedi/comedi_soft_calibrate/Makefile.am`. Al comienzo en la línea `comedi_soft_calibrate_LDADD`, cambie `lboost_program_options` por `lboost_program_options-mt`. Repita todos los pasos para compilar Comedi-Calibrate y Ejecute.

```
$sudo gedit comedi_soft_calibrate/Makefile.am
$cd /usr/local/src/comedi_calibrate-1
$sudo autoreconf -i
$sudo ./configure
$sudo make
$sudo make install
mkdir -p /usr/local/var/lib/comedi/calibrations/
```

#### 4. Instalar RTAI + Comedi:

- **Configurar RTAI con COMEDI:**

```
$cd /usr/local/src
```

```
$sudo cp /usr/local/src/comedi-0.7.76/include/linux/comedi.h /usr/local/include/linux/  
$sudo cp /usr/local/src/comedi-0.7.76/include/linux/comedilib.h /usr/local/include/linux/  
$cd /usr/local/src/rtai  
$make menuconfig (o xconfig or gconfig)
```

- En Menu Add-ons, seleccione “Real Time COMEDI support in user space”, y ejecute:

```
$make  
$sudo checkinstall --fstrans=no  
$sudo make install  
$sudo make dev
```

## 5. ScicosLab + RTAI-Lab Add-ons

- Se deben instalar inicialmente las dependencias:

```
$sudo apt-get install tcl8.5-dev  
$sudo apt-get install gfortran sablotron tcl8.5-dev tk8.5-dev xaw3dg-dev libpvm3 pvm-  
dev libgtkhtml2-dev libvte-dev ocaml-native-compilers  
$sudo apt-get install libgfortran3 tk8.5
```

- Descargue e instale ScicosLab desde el sitio web, luego instale:

```
$wget -P /tmp http://cermics.enpc.fr/~jpc/scilab-gtk-tiddly/files/scicoslab-gtk_4.4-  
2_i386.maverick.deb  
$sudo dpkg -i /tmp/scicoslab-gtk_4.4-2_i386.maverick.deb  
$sudo dpkg -i /tmp/tkdnd1_1.0-1_i386.maverick.deb
```

- Instale RTAI-Lab Add-ons. como root:

```
$cd /usr/local/src/rtai/rtai-lab/scilab/  
$sudo make install  
$cd ../scicoslab/macros
```

- Modificar la versión del ScicosLab en */usr/local/src/rtai/rtai-lab/scicoslab/macros*, En el archivo Makefile está por defecto *SCILAB\_VERSION = 4.4b7*, coloque entonces la versión de SicosLab instalada, para este caso es la 4.4, e instale.

```
$sudo gedit Makefile  
$sudo su -c "make install"  
$make user
```

- Para poder compilar el código generado por rtai-lab se debe hacer lo siguiente:

```
$sudo cp /usr/src/rtai/rtai-lab/scicoslab/rtmain44.c /usr/realtime/share/rtai/Scicos
```

## **6. Instalación de QRtaiLab**

- Se debe instalar las dependencias:

```
$sudo apt-get install libqt4-dev libqwt5-qt4-dev
```

- Descargue las fuentes QRtaiLab desde <http://qrtailab.sourceforge.net> y extráigalas en /usr/local/src:

```
$cd /usr/local/src  
$wget -P /tmp http://downloads.sourceforge.net/qrtailab/QRtaiLab-0.1.12.tar.gz  
$tar xzf QRtaiLab-0.1.12.tar.gz  
$sudo apt-get install qt4-qmake libqt4-dev libqwt5-qt4-dev
```

- Compilar e instalar:

```
$cd qrtailab-0.1.12  
$qmake-qt4  
$make  
$sudo make install
```

## **7. Post instalación:**

- Copie las siguientes líneas al script *rlinsmod*, que usted podrá usar para una rápida carga de los módulos de RTAI-Lab y Comedi en el kernel y configurar los drivers.

```
$sudo nano /usr/realtime/bin/rlinsmod
```

==== Inicie a copier desde la línea de abajo ==== (No incluya ésta línea!)

```

# RLINSMOD - RTAI-Lab Insmod.
# Inserts RTAI-Lab and Comedi modules in kernel and configures the drivers.

insmod /usr/realtime/modules/rtai_smi.ko # only if you have an INTEL chipset
insmod /usr/realtime/modules/rtai_hal.ko
insmod /usr/realtime/modules/rtai_lxrt.ko
insmod /usr/realtime/modules/rtai_fifos.ko
insmod /usr/realtime/modules/rtai_sem.ko
insmod /usr/realtime/modules/rtai_mbx.ko
insmod /usr/realtime/modules/rtai_msg.ko
insmod /usr/realtime/modules/rtai_netrpc.ko
insmod /usr/realtime/modules/rtai_shm.ko
modprobe kcomedilib
insmod /usr/realtime/modules/rtai_comedi.ko

# modprobe -r uhci-hcd # trick to avoid shared interrupts with uhci_hcd devices
# Hardware dependent lines below. This example is for a NI PCI 6221 DAQ Card.
modprobe ni_pcimio
# modprobe uhci-hcd # for the trick to avoid shared interrupts
==== Pare de copiar en la línea de arriba ==== (No incluya ésta línea)

```

- Copie las siguientes líneas en el script *rlrmmo* que puede usar para remover rápidamente los módulos de RTAI-Lab y Comedi del kernel.

```
$sudo nano /usr/realtime/bin/rlrmmmod
```

```
==== Inicie a copiar desde la línea de abajo ==== (No incluya ésta línea!)
```

```
# RLRMMOD - RTAI-Lab Rmmmod.
# Removes RTAI-Lab and Comedi modules from kernel.
```

```
modprobe -r ni_pcimio
```

```

rmmmod rtai_comedi
modprobe -r kcomedilib
rmmmod rtai_shm
rmmmod rtai_netrpc
rmmmod rtai_msg
rmmmod rtai_mbx
rmmmod rtai_sem
rmmmod rtai_fifos

```

```
rmmmod rtai_lxrt
```

```
rmmmod rtai_hal
```

```
rmmmod rtai_smi
```

==== Pare de copiar en la línea de arriba ==== *(No incluya ésta línea!)*

Ejecute los scripts:

```
$sudo chmod +x /usr/realtime/bin/rirmmod
```

```
$sudo chmod +x /usr/realtime/bin/rlnsmmod
```

- Si desea arrancar rlnsmmod después de cada arranque del sistema, adicione la línea *'/usr/realtime/bin/rlnsmmod'* (sin comillas), a su archivo *"rc.local"*. Reemplace la línea antes de *"exit 0"*

```
$sudo nano /etc/rc.local
```

Para más información se sugiere leer el tutorial de RTAI-Lab [4]. Particularmente, puede buscar una descripción de la paleta de RTAI-Lab, una guía para hacer una onda seno en tiempo real, unos pocos ejemplos que muestran cómo usar COLAS y semáforos, y como convertir una planta de tiempo continuo en un modelo de tiempo discreto.



## ANEXO F. CÓDIGO FUENTE DEL MULTI TRANSMISOR VIRTUAL DE ALTURAS PIEZOMÉTRICAS

El Multi transmisor Virtual de alturas piezométricas fue implementado utilizando el lenguaje de programación C/C++, con ayuda de las librerías de visión por computador OpenCV, su diseño modular permitió implementarlo mediante funciones que se encuentran agrupadas en archivos que las clasifican según su especialidad, en la Figura 46, se observa la estructura de archivos que se utilizó para su implementación.

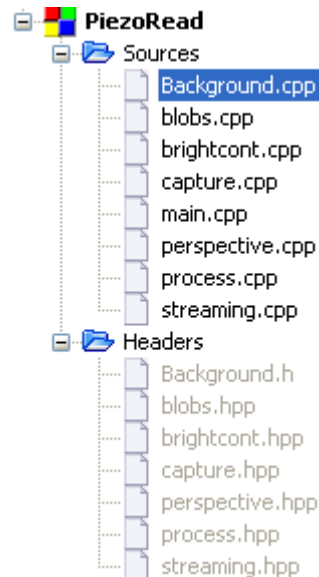


Figura 46. Estructura de archivos.

Cada archivo contiene el código que permite al Multi transmisor ejecutar su funcionalidad, a continuación se detalla el código dentro de cada uno de estos archivos.

### main.cpp

```
#include <iostream>
#include "opencv2/objdetect/objdetect.hpp"
#include "opencv2/video/tracking.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/video/background_segm.hpp"
#include "opencv2/features2d/features2d.hpp"
#include <stdio.h>
#include "blobs.hpp"
#include "Background.h"
#include "perspective.hpp"
#include "process.hpp"
#include "brightcont.hpp"
#include "streaming.hpp"
```

```

using namespace std;
using namespace cv;

/*Parameters of configuration*/
int frames_learning;
int brightness;
int contrast;
int create_perspective;
int calculate_pixelmm;
int test_mode;
int min_area_blobs;
int max_area_blobs;

CvBGCodeBookModel* model;

int main(int argc, char **argv){

    VideoCapture cap;

    if( argc == 1 ||
        (argc == 2 && strlen(argv[1]) == 1 &&
         isdigit(argv[1][0]))){

        cap.open(0);
        cout << "Open camera..." << endl;
    }
    else if( argc == 2 ){
        cap.open(argv[1]);
        cout << "Open Video..." << endl;
    }

    if( !cap.isOpened() ){
        puts("***Could not initialize capturing...***\n");
        return 0;
    }

    //namedWindow( "Capture ", CV_WINDOW_AUTOSIZE);
    //namedWindow( "Foreground ", CV_WINDOW_AUTOSIZE );
    //namedWindow( "Background ", CV_WINDOW_AUTOSIZE );
    Mat frame,foreground,image;

    int fps=cap.get(CV_CAP_PROP_FPS);
    if(fps<=0)
        fps=10;
    else
        fps=1000/fps;

    // Take a frame
    cap>>frame;
    if( frame.empty() ) return -1;

    // Rotate image
    Mat _timg;
    _timg = frame.t();

```

```

flip(_timg, _timg, 0);
frame = _timg;

// load parameters of configuration
cout<<"Load parameters..."<<endl;
loadParameters(&frames_learning,
               &brightness,
               &contrast,
               &create_perspective,
               &calculate_pixelmm,
               &test_mode);

// Create or load Matrix of perspective
Mat Mperspective;
if(create_perspective){
    cout<<"Create Matrix of perspective..."<<endl;

    Point2f src[4];
    Point2f dst[4];
    // Marcar puntos para transformacion de perspectiva
    setPointsPerspective(frame, &src[0], &dst[0] );

    //Mat mper = getPerspectiveTransform(&srcPer[0], &dstPer[0]);
    Mat mper = getPerspective( &src[0], &dst[0] );
    saveMatPerspective(mper);

    cout<<"Load Matrix of perspective..."<<endl;
    Mperspective = loadMatPerspective();
}
else{
    cout<<"Load Matrix of perspective..."<<endl;
    Mperspective = loadMatPerspective();
}

// Calculate of load relation pixel/mm
double pixel_mm;
if(calculate_pixelmm){
    cout<<"Calculate pixel to mm relation..." <<endl;
    Mat dstf;
    warpPerspective(frame, dstf, Mperspective, frame.size());
    calculateRelationPM(dstf, &pixel_mm);
    cout<<"Load pixel to mm relation ..."<<endl;
    pixel_mm = loadRelationPM();
}
else{
    cout<<"Load pixel to mm relation ..."<<endl;
    pixel_mm = loadRelationPM();
}

// Learning the brackground
model = cvCreateBGCodeBookModel();
initBackground(&model);
cout<<"Learning Background Model..."<<endl;
learnBackground(&model, cap, frames_learning);
cout<<"Finished Learning Background Model..."<<endl;

```

```

// Init main loop
cout << "Init main loop..." << endl;
for(;;)
{
    int64 t = getTickCount();
    frame.setTo(0);

    // Capture a frame
    cap>>frame;

    if( frame.empty() )
        break;

    //Rotate image
    Mat timg;
    timg = frame.t();
    flip(timg,timg, 0);

    //Apply Perspective correction
    Mat dst;
    warpPerspective(timg,dst, Mperspective,timg.size());
    frame = dst;

    // Apply contrast and brightness modification
    Mat image_brcon;
    brightcont(frame, image_brcon, brightness, contrast);

    // Convert to YCrCb
    image.setTo(0);
    cvtColor( image_brcon, image, CV_BGR2YCrCb );

    // Applying Gaussian blur
    GaussianBlur( image, image, Size( 5, 5 ), 0, 0 );

    // Detected background/foreground
    foreground.create(frame.size(),CV_8UC1);
    getForeground(&model, image, foreground);

    // Find Countours of foreground
    vector< vector<Point> > contours;
    Mat foreground_cp = foreground.clone();
    contours = FindContoursBlobs( foreground_cp );
    // Filter countours for area
    FilterContoursBlobs( contours, 15, 9000 );

    // Find BoundingBox of Countours
    vector<Rect> BoundingBox;
    BoundingBoxContoursBlobs( contours, BoundingBox, frame, false );
    //cout<<"Size BoundingBox: "<< BoundingBox.size() <<endl;

    // Merge BoundingBox near
    vector<Rect> BoundingBox_merge;
    BoundingBox_merge.clear();
    Mat mer_img = frame.clone();
    mergeBlobs(BoundingBox, BoundingBox_merge, mer_img);
    //cout<<"size merge blobs: "<< BoundingBox.size() << endl;
}

```

```

// Draw contours merging
//drawBoundingBox(frame, BoundingBox );

if (BoundingBox.size() > 0){

    Mat imgheights(frame.size(),CV_8UC1);
    imgheights.setTo(0);
    drawBoundingBox(imgheights, BoundingBox );

    // Encuentra la altura del tubo piezometrico
    vector<Point2f> vCent;
    vCent = findHeights_debug(imgheights);

    // Check height is complete
    completedPiezo( vCent);
    //cout<<"Num Alturas: "<< vCent.size() <<endl;

    // Draw Heights
    drawPointHeights(frame, vCent);

    // convert pixels to mm output in vector<double> and
double[]
    vector<double> vHeights_mm;
    double heights[17];
    pixel2mm(vCent, vHeights_mm, &heights[0] , pixel_mm );

    // drawing text of heights
    Mat mHeights(frame.size(), CV_8UC3);
    drawTextHeights(mHeights, vHeights_mm);

    namedWindow( "Heights", CV_WINDOW_NORMAL);
    imshow( "Heights", mHeights);
}

namedWindow( "Capture", CV_WINDOW_NORMAL);
namedWindow( "Foreground", CV_WINDOW_NORMAL);

// Show Image result
imshow( "Capture",frame );
imshow( "Foreground",foreground);

t = getTickCount() - t;
cout<<"Time elapsed: "<< t*1000/getTickFrequency() <<endl;

// Generate streaming
//streaming( frame, t )

char c = (char)waitKey(fps);
if( c == 'p')
    waitKey();
if( c == 'w'){
    string filename = "foreground.jpg";
    imwrite(filename, foreground);
    imwrite("capture", frame);
    //imwrite("Heights", mHeights);
}

```

```
    }  
    if(c == '1'){  
        // Learning the background  
        cout<<"Learning Background Model..."<<endl;  
        learnBackground(&model, cap, 20);  
        cout<<"Finished Learning Background Model..."<<endl;  
    }  
    if( c == 27 )    // ESC close  
        break;  
    }  
}
```

## Mejora del brillo y contraste

### Brightcont.hpp

```
#ifndef BRIGHTCONT_HPP_INCLUDED
#define BRIGHTCONT_HPP_INCLUDED

#include <iostream>
#include <cv.h>
#include <highgui.h>

using namespace std;
using namespace cv;

void brightcont( Mat &src_image, Mat &dst_image, int _brightness, int
_contrast );

#endif // BRIGHTCONT_HPP_INCLUDED
```

### Brightcont.cpp

```
#include "brightcont.hpp"

void brightcont( Mat &src_image, Mat &dst_image, int _brightness, int
_contrast ){
    int i;
    Mat lut_mat(1,256,CV_8UC1);

    int brightness = _brightness - 100;
    int contrast = _contrast - 100;

    /*
     * The algorithm is by Werner D. Streidt
     * (http://visca.com/ffactory/archives/5-99/msg00021.html)
     */

    if( contrast > 0 ){
        double delta = 127.*contrast/100;
        double a = 255./(255. - delta*2);
        double b = a*(brightness - delta);
        for( i = 0; i < 256; i++ ){
            int v = cvRound(a*i + b);
            if( v < 0 )
                v = 0;
            if( v > 255 )
                v = 255;
            lut_mat.at<uchar>(0,i)= v;
        }
    }
    else{
        double delta = -128.*contrast/100;
        double a = (256.-delta*2)/255.;
        double b = a*brightness + delta;
        for( i = 0; i < 256; i++ ){
```

```
        int v = cvRound(a*i + b);
        if( v < 0 )
            v = 0;
        if( v > 255 )
            v = 255;
        lut_mat.at<uchar>(0,i)= v;
    }
}

LUT( src_image, lut_mat, dst_image );
}
```



## Corrección de perspectiva

### Perspective.hpp

```
#ifndef PERSPECTIVE_HPP_INCLUDED
#define PERSPECTIVE_HPP_INCLUDED

#include <iostream>
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"

using namespace std;
using namespace cv;

void onMouse( int event, int x, int y, int flags, void* );
void setPointsPerspective(Mat imgP, Point2f* src, Point2f* dst );
void drawPointsPerspective(Mat &img, const Point2f* src, const Point2f*
dst);
Mat getPerspective(const Point2f* src, const Point2f* dst );
void applyPerspective(Mat matPerspective, Mat imgSrc, Mat imgDst );

void saveMatPerspective(Mat &mpcr);
Mat loadMatPerspective(void);

#endif // PERSPECTIVE_HPP_INCLUDED
```

### Perspective.cpp

```
#include "perspective.hpp"

Point ptPointer(-1, -1);
Point pt1(-1, -1);
Point pt2(-1, -1);
Point ptline1(-1, -1);
Point ptline2(-1, -1);

Point2f pt1_src(-1, -1);
Point2f pt2_src(-1, -1);
Point2f pt3_src(-1, -1);
Point2f pt4_src(-1, -1);

Point2f pt1_dst(-1, -1);
Point2f pt2_dst(-1, -1);
Point2f pt3_dst(-1, -1);
Point2f pt4_dst(-1, -1);

int flaglpoint =0;
int flaglines = 0;

void onMouse( int event, int x, int y, int flags, void* ){
```

```

static int countclick;

if( event == CV_EVENT_MOUSEMOVE){
    ptPointer = Point(x,y);
}
if( event == CV_EVENT_LBUTTONDOWN ){
    switch(countclick){
        case 0:
            pt1_src = Point2f(x,y);
            flaglpoint = 1;
            cout<<"Punto 1..."<<endl;
            break;
        case 1:
            pt2_src = Point2f(x,y);
            flaglpoint = 2;
            cout<<"Punto 2..."<<endl;
            break;
        case 2:
            pt3_src = Point2f(x,y);
            flaglpoint = 3;
            cout<<"Punto 3..."<<endl;
            break;
        case 3:
            pt4_src = Point2f(x,y);
            flaglpoint = 4;
            cout<<"Punto 4..."<<endl;
            break;
    }
    countclick++;
}
}

void setPointsPerspective(Mat imgP, Point2f* src, Point2f* dst ){

    Point ptOrigin = Point(0,0);
    Point ptFinal = Point(imgP.cols,imgP.rows);
    //cout<<"size img ("<< imgP.cols <<","<< imgP.rows <<)"<< endl;

    // Ventana para correccion de perspectiva
    namedWindow("Perspective", 0);
    imshow("Perspective",imgP);
    // crea el llamado a control del mouse
    setMouseCallback("Perspective", onMouse, 0 );

    for(;;){
        Mat imgtemp = imgP.clone();

        for(int i=0;i<imgP.size().width;i+=20){
            // para poner grilla para ayuda del usuario de color verde
            for(int j=0;j<imgP.size().height;j+=20){
                imgP.at<cv::Vec3b>(j,i)[0]= 0;
                imgP.at<cv::Vec3b>(j,i)[1]= 255;
                imgP.at<cv::Vec3b>(j,i)[2]= 0;
            }
        }
    }
}

```

```

    }

    rectangle( imgtemp, ptPointer, ptFinal, CV_RGB(0,0,100), 2, 8
,0);
    rectangle( imgtemp, ptOrigin, ptPointer, CV_RGB(0,0,100), 2,
8 ,0);

    if(pt1.x > 0 && pt1.y >0 && pt2.x < 0 && pt2.y < 0)
    rectangle( imgtemp, pt1, ptPointer, CV_RGB(0,255,255), 2, 8
,0);
    else if(pt1.x > 0 && pt1.y >0 && pt2.x > 0 && pt2.y > 0)
    rectangle( imgtemp, pt1, pt2, CV_RGB(0,255,0), 2, 8 ,0);

    switch(flaglpoint){
    case 4:
        // pintar p1
        circle(imgtemp, pt4_src, 3, Scalar(255,255,0),3);
    case 3:
        // pintar p1 y p2
        circle(imgtemp, pt3_src, 3, Scalar(0,255,0),3);
    case 2:
        // pintar p1, p2, p3
        circle(imgtemp, pt2_src, 3, Scalar(0,255,0),3);
    case 1:
        // pintar p1, p2, p3, p4
        circle(imgtemp, pt1_src, 3, Scalar(0,255,0),3);
    break;
    }

    imshow("Perspective",imgtemp);

    if( cvWaitKey(100) >= 0 ){
        cout<<"salio de captura de ROI"<<endl;
        break;
    }
}

pt1_dst = Point2f(1, 1);
pt2_dst = Point2f( imgP.cols, 1);
pt3_dst = Point2f(1, imgP.rows);
pt4_dst = Point2f(imgP.cols,imgP.rows);

cout<<"p1: ("<< pt1_src.x <<","<< pt1_src.y << ")" <<endl;
cout<<"p2: ("<< pt2_src.x <<","<< pt2_src.y << ")" <<endl;
cout<<"p3: ("<< pt3_src.x <<","<< pt3_src.y << ")" <<endl;
cout<<"p4: ("<< pt4_src.x <<","<< pt4_src.y << ")" <<endl;

cout<<"pf1: ("<< pt1_dst.x <<","<< pt1_dst.y << ")" <<endl;
cout<<"pf2: ("<< pt2_dst.x <<","<< pt2_dst.y << ")" <<endl;
cout<<"pf3: ("<< pt3_dst.x <<","<< pt3_dst.y << ")" <<endl;
cout<<"pf4: ("<< pt4_dst.x <<","<< pt4_dst.y << ")" <<endl;

// ya encontro los puntos
*(src) = pt1_src;
*(src+1) = pt2_src;
*(src+2) = pt3_src;

```

```

    *(src+3) = pt4_src;

    *(dst) = pt1_dst;
    *(dst+1) = pt2_dst;
    *(dst+2) = pt3_dst;
    *(dst+3) = pt4_dst;
}

void drawPointsPerspective(Mat &img, const Point2f* src, const Point2f*
dst){
    cout<<"Puntos desde drawing..."<<endl;
    cout<<"p1: ("<< *(src+0)).x <<","<< *(src+0)).y << ")" <<endl;
    cout<<"p2: ("<< *(src+1)).x <<","<< *(src+1)).y << ")" <<endl;
    cout<<"p3: ("<< *(src+2)).x <<","<< *(src+2)).y << ")" <<endl;
    cout<<"p4: ("<< *(src+3)).x <<","<< *(src+3)).y << ")" <<endl;

    cout<<"pf1: ("<< *(dst+0)).x <<","<< *(dst+0)).y << ")" <<endl;
    cout<<"pf2: ("<< *(dst+1)).x <<","<< *(dst+1)).y << ")" <<endl;
    cout<<"pf3: ("<< *(dst+2)).x <<","<< *(dst+2)).y << ")" <<endl;
    cout<<"pf4: ("<< *(dst+3)).x <<","<< *(dst+3)).y << ")" <<endl;

    // Dibujar y mostrar los puntos
    circle(img,*(src+0) , 4, Scalar(0,255,0),2);
    circle(img,*(src+1) , 4, Scalar(0,255,0),2);
    circle(img,*(src+2) , 4, Scalar(0,255,0),2);
    circle(img,*(src+3) , 4, Scalar(0,255,0),2);
    circle(img,*(dst+0) , 3, Scalar(0,0,255),2);
    circle(img,*(dst+1) , 3, Scalar(0,0,255),2);
    circle(img,*(dst+2) , 3, Scalar(0,0,255),2);
    circle(img,*(dst+3) , 3, Scalar(0,0,255),2);
}

Mat getPerspective(const Point2f* src, const Point2f* dst ){
    Mat mper = getPerspectiveTransform(src, dst);
    return mper;
}

void applyPerspective(Mat matPerspective, Mat &imgSrc, Mat &imgDst ){
    warpPerspective(imgSrc,imgDst, matPerspective,imgDst.size());
}

void saveMatPerspective(Mat &mper){
    FileStorage fs("Calibration/MatPerspective.yml",
FileStorage::WRITE);
    fs << "MatPerspective" << mper;
}

Mat loadMatPerspective(void){
    Mat mper;
    FileStorage fs("Calibration/MatPerspective.yml", FileStorage::READ);
    fs["MatPerspective"] >> mper;
    return mper;
}

```

## Deteccion de primer plano y fondo

### Background.hpp

```
#ifndef BACKGROUND_H_INCLUDED
#define BACKGROUND_H_INCLUDED

#include <iostream>
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/video/background_segm.hpp"
#include "opencv2/imgproc/imgproc_c.h"
#include "opencv2/highgui/highgui.hpp"
#include "perspective.hpp"

using namespace cv;
using namespace std;

void initBackground(CvBGCodeBookModel** model);
void updateBackground(CvBGCodeBookModel* model, Mat &frame);
void clearStaleBackground(CvBGCodeBookModel* model);
void learnBackground(CvBGCodeBookModel** model, VideoCapture &cap, int
nframes);
void getForeground(CvBGCodeBookModel** model, Mat frame, Mat
&foreground);

#endif // BACKGROUND_H_INCLUDED
```

### Background.cpp

```
#include "Background.h"
#include "brightcont.hpp"

extern int brightness;
extern int contrast;

void initBackground(CvBGCodeBookModel** model){

    //Set color thresholds to default values
    (*model)->modMin[0] = 3;
    (*model)->modMin[1] = (*model)->modMin[2] = 3;
    (*model)->modMax[0] = 10;
    (*model)->modMax[1] = (*model)->modMax[2] = 10;
    (*model)->cbBounds[0] = (*model)->cbBounds[1] = (*model)-
>cbBounds[2] = 10;
    //model->storage = cvCreateMemStorage();
}

void updateBackground(CvBGCodeBookModel* model, Mat &frame){
    IplImage Image = frame;
    cvBGCodeBookUpdate( model, &Image );
}

void clearStaleBackground(CvBGCodeBookModel* model){
```

```

    cvBGCodeBookClearStale( model, model->t/2 );
}

void learnBackground(CvBGCodeBookModel** model, VideoCapture &cap, int
nframes){
    int i;
    Mat frame, image, dst;

    //cout<<"Load Matrix of perspective..."<<endl;
    Mat Mperspective = loadMatPerspective();

    for(i=0;i<nframes;i++){
        cap>>frame;
        if( frame.empty() ) break;

        //Rotate image
        Mat timg;
        timg = frame.t();
        flip(timg,timg, 0);

        //Apply Perspective correction
        Mat dst_per;
        warpPerspective(timg,dst_per, Mperspective,timg.size());

        // Apply contrast and brightness modification
        Mat image_brcon;
        brightcont(dst_per,image_brcon, brightness, contrast);

        // Convert to YCrCb
        cvtColor( image_brcon, image, CV_BGR2YCrCb );
        // Applying Gaussian blur
        GaussianBlur( image, image, Size( 5, 5 ), 0, 0 );
        image.convertTo(dst, CV_8UC3 );// frame.size(),CV_8UC3);

        //updateBackground(model, dst);
        IplImage Image = dst;
        cvBGCodeBookUpdate( *model, &Image );

        timg.setTo(0);
        frame.setTo(0);
    }
    //clearStaleBackground(model);
    cvBGCodeBookClearStale( *model, (*model)->t/2 );
}

void getForeground(CvBGCodeBookModel** model, Mat frame, Mat
&foreground){
    // Find foreground by codebook method

    //foreground.setTo(Scalar(255));
    IplImage Image = frame;
    IplImage iplforeground = foreground;
    //IplImage iplsegm = foreground;

    //cout<<"w: "<< model->size.width << "r: " << model->size.height

```

```
<<endl;
    cvBGCodeBookDiff( *model, &Image, &iplforeground );
    // copy iplimage to Mat
    //cvShowImage("image", &iplforeground);
    //cvCopy(&iplforeground,&iplsegm);
    //cvSegmentFGMask( &iplsegm );
    foreground = Mat(&iplforeground,true);
}
```

## Componentes conectadas(blobs)

### blobs.hpp

```
#ifndef BLOBS_HPP_INCLUDED
#define BLOBS_HPP_INCLUDED

#include "opencv2/video/tracking.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/video/background_segm.hpp"
#include "opencv2/features2d/features2d.hpp"
#include <stdio.h>
#include <iostream>

using namespace std;
using namespace cv;

vector< vector<Point> > FindContoursBlobs( Mat img );
void DrawContoursBlobs( Mat img, vector< vector<Point> > contours );
void FilterContoursBlobs( vector< vector<Point> > &contours, int cmin,
int cmax );
void BoundingBoxContoursBlobs( vector< vector<Point> > contours,
vector<Rect> &BoundingBox,Mat img, bool draw );
void CircleContoursBlobs( vector< vector<Point> > &contours,
vector<Point2f> &Center, Mat img, bool draw );
void mergeBlobs(vector<Rect> &BoundingBox, vector<Rect>
&BoundingBoxMerging, Mat &img);
void drawBoundingBox(Mat &img, vector<Rect> &BoundingBox );

#endif // BLOBS_HPP_INCLUDED
```

### blobs.cpp

```
#include "blobs.hpp"

vector< vector<Point> > FindContoursBlobs( Mat img ){
    //Extract the contours so that
    vector< vector<Point> > contours;
    contours.clear();
    findContours( img, contours, RETR_EXTERNAL, CHAIN_APPROX_NONE);

    return contours;
}

void DrawContoursBlobs( Mat img, vector< vector<Point> > contours ){
    if(img.channels() == 3)
        drawContours( img, contours, -3, Scalar(0,255,0),1);
    else
        drawContours( img, contours, -3, Scalar(255,0,0),1);
}

void FilterContoursBlobs( vector< vector<Point> > &contours, int cmin,
int cmax ){
```



```

// Eliminate too short or too long contours
unsigned int _cmin = (unsigned int)cmin;
unsigned int _cmax = (unsigned int)cmax;

vector< vector<Point> >::iterator itc= contours.begin();
while (itc!=contours.end()){
    if (itc->size() < _cmin || itc->size() > _cmax){
        itc= contours.erase(itc);
    }
    else{
        ++itc;
    }
}
}

void BoundingBoxContoursBlobs( vector< vector<Point> > contours,
vector<Rect> &BoundingBox,Mat img, bool draw ){

// Encuentre y dibuje el Bounding Box de los contornos
vector< vector<Point> >::iterator it= contours.begin();
while (it!=contours.end()){
    Rect r0= boundingRect(Mat(*it));
    BoundingBox.push_back(r0);

    if( draw ){
        if(img.channels() == 3)
            rectangle(img,r0,Scalar(0,0,255),1);
        else
            rectangle(img,r0,Scalar(255,0,0),1);
    }
    ++it;
}
}

void CircleContoursBlobs( vector< vector<Point> > &contours,
vector<Point2f> &Center, Mat img, bool draw){

// Encuentra un cilculo que encierre los puntos de los extrmos
superiores de los blobs
vector< vector<Point> >::iterator itc= contours.begin();
while (itc!=contours.end()){
    float radius;
    Point2f center;
    minEnclosingCircle(*itc, center, radius);
    Center.push_back(center);

    if( draw ){
        if(img.channels() == 3)
            circle(img, center, 2, Scalar(0,0,255),2);//, int
thickness=1, int lineType=8, int shift=0)
        else
            circle(img, center, 2, Scalar(255,0,0),2);//, int
thickness=1, int lineType=8, int shift=0)
    }

    ++itc;
}
}

```

```

    }
}

void mergeBlobs(vector<Rect> &BoundingBox, vector<Rect>
&BoundingBoxMerging, Mat &img){
    Mat _img;
    int contblobs = 0;
    // Recorrer cada uno de los contornos en "contours"
    vector<Rect>::iterator it = BoundingBox.end()-1;
    while(it != BoundingBox.begin()-1){

        // Calcular la distancia vertical y horizontal a los blobs
siguientes
        for(int i=1; i<(BoundingBox.size() - contblobs ); i++){

            int distanceVertical = abs( ((*it).y + (*it).height ) -
(*it-i).y );
            int distanceHorizontal = abs( (*it).x - (*it-i).x );

            // Comparar la distancia con el THR, unir en un nuevo blob
"boundingBox" si cumple el criterio
            if (distanceVertical < 300 && distanceHorizontal <13 ){
                Rect nrec;

                nrec.x = min((*it).x, (*it-i).x);
                nrec.y = min((*it).y, (*it-i).y);

                int x1 = (*it).x + (*it).width;
                int x2 = (*it-i).x + (*it-i).width;

                int y1 = (*it).y + (*it).height;
                int y2 = (*it-i).y + (*it-i).height;

                nrec.width = max(x1,x2) - nrec.x ;
                nrec.height = max(y1 , y2) - nrec.y;

                // Delete contour
                BoundingBox.erase(it-i);
                BoundingBox.erase(it-1);
                --it;

                // Inserted new blob...
                //cout<<"Insertando nuevo blob..."<<endl;
                BoundingBox.insert ( it , nrec );
                --i;
            }
        }

        --it;
        contblobs++;
    }
}

void drawBoundingBox(Mat &img, vector<Rect> &BoundingBox ){
    vector<Rect>::iterator it = BoundingBox.begin();
    while(it != BoundingBox.end()){

```

```
        rectangle(img,*it,Scalar(255,255,255),1);  
        ++it;  
    }  
}
```

## Deteccion de regiones

### Process.hpp

```
#ifndef PROCESS_HPP_INCLUDED
#define PROCESS_HPP_INCLUDED

#define HEIGHTS 17
#define HEIGHT_TABLE_MM 1000.0
#define HEIGHT_CAMERA_PIXELS 800
#define DISTANCE_VERTICAL_MM 1080
#define DIV_IMAGE 80

#include <iostream>
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "blobs.hpp"

using namespace std;
using namespace cv;

vector<Point2f> findHeights_debug(Mat img);
void findPointHeights_left2right(Mat &image, vector<Point> &points, Mat
&res);
void findPointHeights_up2below(Mat &image, vector<Point> &points, Mat
&res);
void _findPointHeights_up2below(Mat &image, vector<Point> &points, Mat
&res);
void drawPointHeights(Mat &image, vector<Point2f> vCent);
void completedPiezo( vector<Point2f> &vHeights );
void vector2array( vector<Point2f> &vHeights, double *heights );
void pixel2mm(vector<Point2f> &vHeights_pixel, vector<double>
&vHeights_mm, double *heights , double k );
void drawTextHeights(Mat &img, vector<double> &vHeights_mm);
void calculateRelationPM(Mat &img, double *k);
void onMousec( int event, int x, int y, int flags, void* );
double loadRelationPM(void);
void loadParameters(int *frames_learning,
int *brightness,
int *contrast,
int *create_perspective,
int *calculate_pixilmm,
int *test_mode);

#endif // PROCESS_HPP_INCLUDED
```

### Process.cpp

```
#include "process.hpp"

extern int test_mode;

Point pt(-1, -1);
```

```

Point p1(-1, -1);
Point p2(-1, -1);
int flagpoint = 0;

vector<Point2f> findHeights_debug(Mat img){

    // Encuentra los extrmos superiores de los boundingBox
    vector<Point> points;
    Mat res(img.size(),CV_8UC1);
    res.setTo(Scalar(0));

    findPointHeights_up2below(img, points, res);
    //

    // Extrae los contornos de los extremos superiores
    vector< vector<Point> > contours;
    contours = FindContoursBlobs( res );

    // Filtra por tamaño los contornos de los extremos superiores
    int cmin = 9;
    int cmax = 10000;
    FilterContoursBlobs( contours, cmin, cmax );

    // Encuenrta un cilculo que encierre los puntos de los extrmos
    superiores de los blobs
    vector<Point2f> vCenter;
    CircleContoursBlobs( contours, vCenter, img, false );

    return vCenter;
}

// Encuentra los extremos superiores de los boundingBox de los blobs de
izquierda a derecha
void findPointHeights_left2right(Mat &image, vector<Point> &points, Mat
&res){

    if(test_mode){
        for( int j = 0 ; j<image.rows; ++j ){ // aqui empezar en un
valor mas grande
            for( int i = image.cols-1; i > 1 ; --i){
                if (image.at<uchar>(i,j)>0){
                    res.at<uchar>(i,j) = 255;
                    i = 0;
                }
            }
        }
    }
    else{
        // Iterate over the pixels to obtain all point positions
        for( int j = DIV_IMAGE; j < image.rows; ++j ){
            for( int i = image.cols-1; i > 1; --i){
                if (image.at<uchar>(i,j)>0){
                    res.at<uchar>(i,j) = 255;
                    i = image.rows;
                }
            }
        }
    }
}

```

```

    }
}
// segundo recorrido
for( int j = 0 ; j<DIV_IMAGE; ++j ){ // aqui empezar en un valor
mas grande
    for( int i = 0; i < image.cols; ++i){
        if (image.at<uchar>(i,j)>0) {
            res.at<uchar>(i,j) = 255;
            i = 0;
        }
    }
}
}

// Encuentra los extrmos superiores de los boundingBox de los blobs de
arriba a bajo
void findPointHeights_up2below(Mat &image, vector<Point> &points, Mat
&res){
    if(test_mode){
        for( int j = 0 ; j<image.cols; ++j ){ // aqui empezar en un
valor mas grande
            for( int i = image.rows-1; i > 1 ; --i){
                if (image.at<uchar>(i,j)>0) {
                    res.at<uchar>(i,j) = 255;
                    i = 0;
                }
            }
        }
    }
    else{
        // Iterate over the pixels to obtain all point positions
        for( int j = 0; j < image.cols-DIV_IMAGE; ++j ){ // aqui empezar
en un valor mas grande
            for( int i = 0; i < image.rows; ++i){
                if (image.at<uchar>(i,j)>0) {
                    res.at<uchar>(i,j) = 255;
                    i = image.rows;
                }
            }
        }
    }

    //cout<<"Second Recor..."<<endl;

    for( int j = image.cols-DIV_IMAGE ; j<image.cols; ++j ){ // aqui
empezar en un valor mas grande
        for( int i = image.rows-1; i > 1 ; --i){
            if (image.at<uchar>(i,j)>0) {
                res.at<uchar>(i,j) = 255;
                i = 0;
            }
        }
    }
}
}
}

```

```

// Encuentra los extrmos superiores de los boundingBox de los blobs de
arriba a bajo
void _findPointHeights_up2below(Mat &image, vector<Point> &points, Mat
&res){
    if(test_mode){
        for( int j = 0 ; j<image.cols; ++j ){ // aqui empezar en un
valor mas grande
            for( int i = image.rows-1; i > 1 ; --i){
                if (image.at<uchar>(i,j)>0) {
                    res.at<uchar>(i,j) = 255;
                    i = 0;
                }
            }
        }
    }
    else{
        // Iterate over the pixels to obtain all point positions
        for( int j = DIV_IMAGE; j < image.cols; ++j ){ // aqui empezar
en un valor mas grande
            for( int i = 0; i < image.rows; ++i){

                if (image.at<uchar>(i,j)>0) {
                    res.at<uchar>(i,j) = 255;
                    i = image.rows;
                }
            }
        }

        //cout<<"Second Recor..."<<endl;

        for( int j = 0 ; j<DIV_IMAGE; ++j ){ // aqui empezar en un valor
mas grande
            for( int i = image.rows-1; i > 1 ; --i){
                if (image.at<uchar>(i,j)>0) {
                    res.at<uchar>(i,j) = 255;
                    i = 0;
                }
            }
        }
    }
}

void drawPointHeights(Mat &image, vector<Point2f> vCent){

    vector<Point2f>::iterator itVc = vCent.begin();
    int r=0;
    int g=0;
    int b=0;
    int i = 0;
    while (itVc != vCent.end()){
        Scalar Color;
        Color = Scalar(0,255,0);

        circle(image, *itVc, 2, Color,2);//, int thickness=1, int
lineType=8, int shift=0)

```

```

        ++itVc;
        i++;
    }
}

void completedPiezo( vector<Point2f> &vHeights ){
    int dif;
    //cout<<"Numero de piezometros: "<< vHeights.size() <<endl;

    if(vHeights.size() > HEIGHTS ){
        // se ha detectado alturas de mas
        dif = vHeights.size() - HEIGHTS;
        cout<<"Diferencia de mas: "<< dif <<endl;

        vector<Point2f>::iterator it= vHeights.begin();
        for(int i=0 ; i<dif ; i++){
            it= vHeights.erase(it);
            //++it;
        }
    }
    else if(vHeights.size() < HEIGHTS ){
        // hacen falta alturas
        dif = HEIGHTS - vHeights.size();
        cout<<"Diferencia de menos: "<< dif <<endl;

        vector<Point2f>::iterator it= vHeights.begin();
        for(int i=0 ; i<dif ; i++){
            Point2f p(1,1);
            vHeights.push_back(p);
            ++it;
        }
    }
    else{
        //alturas completas
    }
}

void vector2array( vector<Point2f> &vHeights, double *heights ){
    vector<Point2f>::iterator it= vHeights.begin();
    while (it!=vHeights.end()){
        *heights = (double)(*it).y;
        heights++,
        ++it;
    }
}

void pixel2mm(vector<Point2f> &vHeights_pixel, vector<double>
&vHeights_mm, double *heights , double k ){

    //double k = HEIGHT_TABLE_MM/HEIGHT_CAMERA_PIXELS;

    vector<Point2f>::iterator it= vHeights_pixel.begin();
    while (it!=vHeights_pixel.end()){
        if( (*it).y > 0 ){
            double mm = k*(HEIGHT_CAMERA_PIXELS - (*it).y);
            vHeights_mm.push_back(mm);
        }
    }
}

```



```

        //cout<<"h: " << fixed << mm <<endl;
        *heights = mm;
        heights++;
    }
    ++it;
}
}

void drawTextHeights(Mat &img, vector<double> &vHeights_mm){

    int factor = 1;
    int i = 17;
    int fontFace = CV_FONT_HERSHEY_COMPLEX_SMALL;
    double fontScale = 1;
    int thickness = 2;
    int lineType = 8;

    vector<double>::iterator it= vHeights_mm.begin();
    while (it!=vHeights_mm.end()){

        char strn[20];
        sprintf(strn,"P[%i]: %3.1fmm", i, *it );
        Point org( 10, i*25 );
        putText( img, strn, org, fontFace, fontScale, Scalar(255, 0, 0),
thickness, lineType);

        i--;
        factor++;
        ++it;
    }
}

void calculateRelationPM(Mat &img, double *k){

    // Points Origen and final
    Point ptOrigin = Point(0,0);
    Point ptFinal = Point(img.cols,img.rows);

    // Ventana para correccion de perspectiva
    namedWindow("Perspective", 0);
    // crea el llamado a control del mouse
    setMouseCallback("Perspective", onMousec, 0 );

    cout<<"size img ("<< img.cols <<","<< img.rows <<)"<< endl;

    for(;;){
        Mat imgtemp = img.clone();

        for(int i=0;i<img.size().width;i+=20){
            // para poner grilla para ayuda del usuario de color verde
            for(int j=0;j<img.size().height;j+=20){
                img.at<cv::Vec3b>(j,i)[0]= 0;
                img.at<cv::Vec3b>(j,i)[1]= 255;
                img.at<cv::Vec3b>(j,i)[2]= 0;
            }
        }
    }
}

```

```

rectangle( imgtemp, pt, ptFinal, CV_RGB(0,0,100), 2, 8 ,0);
rectangle( imgtemp, ptOrigin, pt, CV_RGB(0,0,100), 2, 8 ,0);

switch(flagpoint){
    case 2:
        // pintar p1, p2, p3
        circle(imgtemp, p2, 3, Scalar(0,255,0),3);
    case 1:
        // pintar p1, p2, p3, p4
        circle(imgtemp, p1, 3, Scalar(0,255,0),3);
    break;
}

imshow("Perspective",imgtemp);

if( cvWaitKey(1) >= 0 ){
    cout<<"salio de captura de ROI"<<endl;
    break;
}
}

// Puntos 1 y 2 seleccionados
cout<<"p1: ("<< p1.x <<","<< p1.y << ")" <<endl;
cout<<"p2: ("<< p2.x <<","<< p2.y << ")" <<endl;

// calculate distance vertical between pt1 and pt2
double dVerticalPixel = abs( p1.y - p2.y );
cout<<"Vertical Distance: "<< dVerticalPixel <<endl;

*k = DISTANCE_VERTICAL_MM/dVerticalPixel;
cout<<"Relation mm/pixel: "<< *k <<endl;

// save to yaml file
FileStorage fs("Calibration/pixelmm.yml", FileStorage::WRITE);
fs << "pixel2mm" << *k;
}

void onMousec( int event, int x, int y, int flags, void* ){

    static int countclick;

    if( event == CV_EVENT_MOUSEMOVE){
        pt = Point(x,y);
    }
    if( event == CV_EVENT_LBUTTONDOWN ){
        switch(countclick){
            case 0:
                p1 = Point(x,y);
                flagpoint = 1;
                cout<<"Punto 1..."<<endl;
            break;
            case 1:
                p2 = Point(x,y);
                flagpoint = 2;
                cout<<"Punto 2..."<<endl;

```

```

        break;
    }
    countclick++;
}
}

double loadRelationPM(void){
    double k;
    FileStorage fs("Calibration/pixelmm.yml", FileStorage::READ);
    fs["pixel2mm"] >> k;
    return k;
}

void loadParameters(int *frames_learning,
                   int *brightness,
                   int *contrast,
                   int *create_perspective,
                   int *calculate_pixilmm,
                   int *test_mode){

    int _frames_learning;
    int _brightness;
    int _contrast;
    int _create_perspective;
    int _calculate_pixilmm;
    int _test_mode;

    FileStorage fs("Calibration/parameters.yml", FileStorage::READ);
    fs["frames_learning"] >> _frames_learning;
    fs["brightness"] >> _brightness;
    fs["contrast"] >> _contrast;
    fs["create_perspective"] >> _create_perspective;
    fs["_calculate_pixilmm"] >> _calculate_pixilmm;
    fs["test_mode"] >> _test_mode;

    *frames_learning = _frames_learning;
    *brightness = _brightness;
    *contrast = _contrast;
    *create_perspective = _create_perspective;
    *calculate_pixilmm = _calculate_pixilmm;
    *test_mode = _test_mode;
}

```

## Generacion de streaming

### streaming.hpp

```
#ifndef STREAMING_HPP_INCLUDED
#define STREAMING_HPP_INCLUDED

#include <iostream>
#include <cv.h>
#include <highgui.h>

using namespace std;
using namespace cv;

void streaming( Mat img, int64 t );

#endif // STREAMING_HPP_INCLUDED
```

### streaming.hpp

```
#include "streaming.hpp"

void streaming( Mat img, int64 t ){
    static double ms=0;

    ms += t/((double)cvGetTickFrequency() * 1000.0);

    // autosave every x seconds
    if( ceil(ms) >= 1000 ) {
        string filename = "webcam.jpg";
        imwrite(filename, img);
        ms = 0;
    }
}
```

## REFERENCIAS BIBLIOGRÁFICAS.

- [1] Nokia Corporation. (Marzo, 2012) Qt project. [En línea]. <http://www.qt-project.org/>
- [2] Oren Ben-Kiki, "YAML Ain't Markup Language", 2009.
- [3] David Schleeff y Frank Hess, The Control and Measurement Device Interface Handbook, 2003.
- [4] Roberto Bucher, Simone Mannori, y Thomas Netter. (2010, Mayo) [En línea]. <http://www.rtai.org>
- [5] Edgar René Valverde Sanmiguel y Juan Pablo Rivera Plata, "Diseño e Implementación de un Laboratorio remoto: caso de estudio planta de pérdidas de carga en conductos a presión del laboratorio de hidráulica", Universidad del Cauca, Popayán, Tesis de pregrado 2010.
- [6] Irving h. Sshames, *Mecánica de fluidos*, Tercera Edición. ed.: McGraw hill, 1995.