

# *Inspección de Tuberías Industriales*

## *Ejercicio N° 1*

<b>Objetivos</b>	<ul style="list-style-type: none"><li>• Buenas prácticas en programación de Tipos de Datos Abstractos (TDAs)</li><li>• Modularización de sistemas</li><li>• Correcto uso de recursos (memoria dinámica y archivos)</li></ul>
<b>Instancias de Entrega</b>	<b>Entrega 1:</b> clase 4 (08/09/2015). <b>Entrega 2:</b> clase 6 (22/09/2015).
<b>Temas de Repaso</b>	<ul style="list-style-type: none"><li>• Uso de structs y typedef</li><li>• Uso de macros y archivos de cabecera</li><li>• Funciones para el manejo de Strings</li></ul>
<b>Criterios de Evaluación</b>	<ul style="list-style-type: none"><li>• Criterios de ejercicios anteriores</li><li>• Cumplimiento de la totalidad del enunciado del ejercicio</li><li>• Ausencia de variables globales</li><li>• Correcta encapsulación en TDAs y separación en archivos</li><li>• Uso de interfaces para acceder a datos contenidos en TDAs</li><li>• Empleo de memoria dinámica de forma ordenada y moderada</li><li>• Acceso a información de archivos de forma ordenada y moderada</li></ul>

## Índice

[Introducción](#)

[Ultrasonic Testing](#)

[Magnetic Flow Leakage](#)

[Descripción](#)

[Formato de Línea de Comandos](#)

[Códigos de Retorno](#)

[Archivos de Entrada](#)

[Mediciones](#)

[Tuberías](#)

[Recorrido](#)

[Entrada y Salida Estándar](#)

[Salida Estándar de Error](#)

[Ejemplos de Ejecución](#)

[Un Sensor](#)

[Restricciones](#)

[Referencias](#)

## Introducción

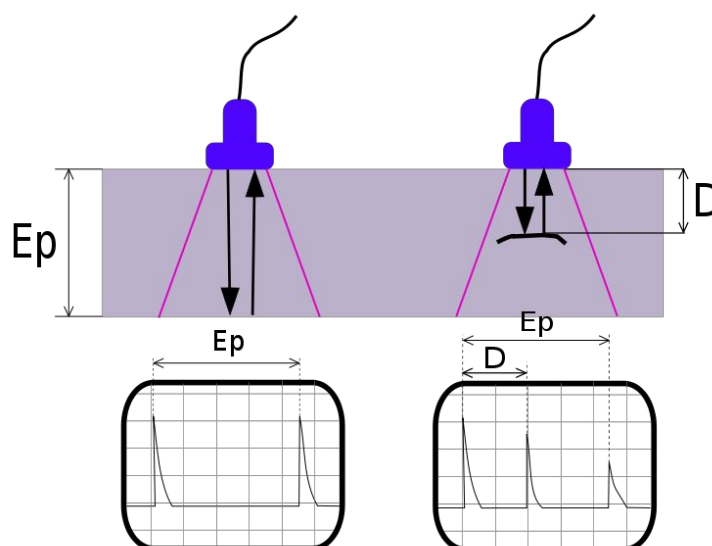
El estudio de tuberías para el ámbito industrial presenta gran importancia debido a la dependencia de los procesos fabriles respecto del transporte de fluidos. En planificación urbana, las líneas de abastecimiento de agua y gas son indispensables y abarcan complicadas redes subterráneas y plantas reductoras. De la misma manera, en el circuito energético de un país se requiere la construcción de grandes tuberías para el transporte de gas y derivados del petróleo tanto subterráneos como al aire libre.

Uno de los mayores problemas que presentan las tuberías son las pérdidas por desgaste mecánico, corrosión y defectos de fabricación. En algunas situaciones estas pequeñas rupturas son evidentes a la vista y pueden ser controladas pero en la mayoría de los casos es conveniente realizar un trabajo preventivo de detección antes que se detecten fugas de los fluidos transportados.

Existen distintos mecanismos para la detección de corrosión, defectos de fabricación y rupturas (hoyos o *pits*) en los tubos metálicos entre los que se destacan las pruebas por ultrasonido (*Ultrasonic Testing* o UT) y la detección de fugas de flujo magnético (*Magnetic Flux Leakage* o MFL).

## Ultrasonic Testing

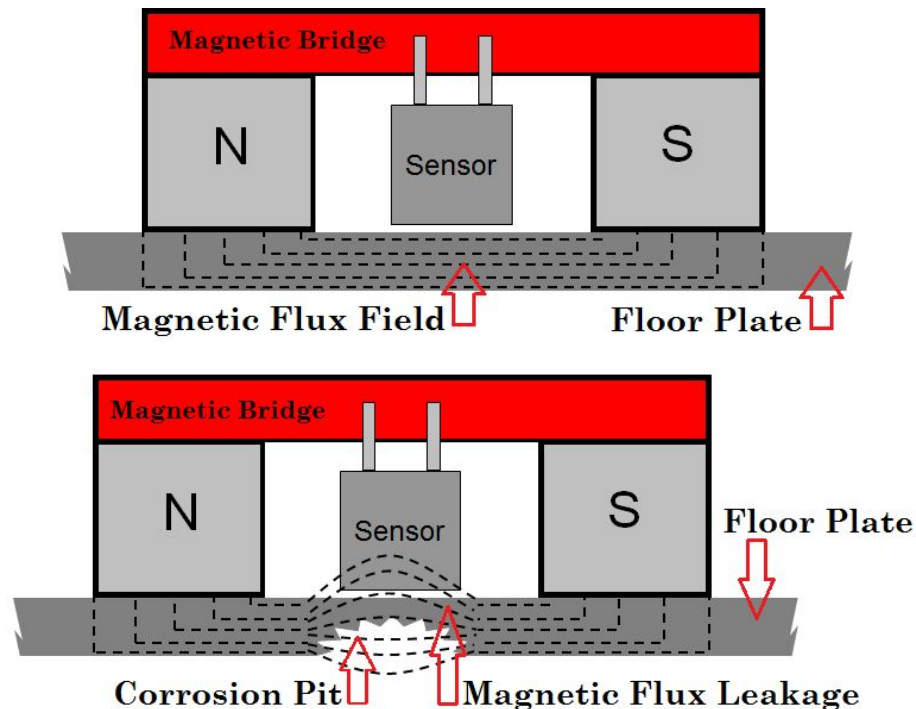
En el caso de UT, se utiliza un cabezal que emite cierta señal sobre un área muy pequeña de la pared del tubo. La recepción del rebote de dicha señal permite determinar el grosor de la pared siendo que cualquier imperfección interna provoca anomalías en el rebote.



Izq: Pared de un tubo sin anomalías. Der: Pared de un tubo con anomalías a una distancia  $D$  de la superficie. Fuente: Wikimedia.org

## Magnetic Flow Leakage

El método MFL se puede utilizar con metales ferrosos pero demostró ser más preciso que UT [1]. En este caso se introduce un campo magnético entre el material y un potente imán. En condiciones normales, el metal conduce el campo magnético sin pérdidas entre los polos N y S del imán. Por el contrario, la aparición de pérdidas en dicho flujo implican impurezas en el material, corrosión o rupturas que impiden un flujo magnético homogéneo.



Sup: el puente magnético no posee interrupciones y no se detectan pérdidas de flujo. Inf: frente a una ruptura (*pit*) se detecta cierto patrón de pérdida en el flujo magnético.

Fuente: <http://www.mfeenterprises.com/what-is-mfl>

Esta técnica se puede utilizar tanto en herramientas manuales para inspección de superficies [2] como para la medición automática con herramientas transportadas por el propio flujo [3]. Los sensores utilizados para este último método se conocen como medidores para inspección de tuberías (*Pipeline Inspection Gauges*, PIG o simplemente *pigs*) y se encuentran en gran variedad en para la industria del petróleo y gas [4, 5].

## Descripción

El presente trabajo trata sobre algunos mecanismos de detección de rupturas y corrosión en tuberías metálicas enfocado al software necesario para reportar el análisis de resultados.

Se nos pide un sistema de software que analice las señales obtenidas por una máquinas MFL en cierto trayecto, detecte sectores con mayor probabilidad de corrosión o fugas en las tuberías y reporte los resultados indicando la ubicación.

El circuito real de la tubería consta de distintos nodos de distribución y tramos de tubos que los interconectan. Cada nodo puede tener más de una entrada y más de una salida, permitiendo elegir cuales estarán activas y, por lo tanto, controlar el caudal de fluido de acuerdo a la demanda. Se cuenta con los

planos de cada tubería incluyendo los nombres de los nodos, conexiones y distancia de dichas conexiones.

El instrumento de medición utilizado es un *pig* con cierta cantidad de sensores que es ingresado en la tubería para realizar un recorrido definido. El recorrido del *pig* se define de antemano y todos los nodos distribuidores son preparados para garantizar dicho flujo. Una vez que el *pig* termina su recorrido, es extraído para obtener el conjunto de mediciones y computar el análisis de fugas con nuestro software.

Cada medición (o muestra) consta de 2 bytes por cada sensor indicando la cantidad de flujo magnético detectado. Valores entre 0x0000 y 0x0099 son considerados normales. Valores superiores pueden indicar alguna anomalía.

Analizando las muestras consecutivas de un sensor determinado es posible determinar el estado de un sector de la pared del tubo. Por otra parte, la suma de los sensores cubren la totalidad de las paredes del tubo pudiendo detectar 2 tipos de anomalías:

- Corrosión: valores iguales o superiores a 0x00A0 durante 0,5m o más de trayecto.
- Fugas: valores superiores a 0x00FF

Si bien el *pig* suele rotar dentro de la tubería confundiendo la posición de los sensores, vamos a asumir que los efectos de rotación fueron compensados por el hardware y no afectan la posición de las muestras.

Cabe aclarar que los archivos de mediciones pueden ser extremadamente grandes dependiendo de la cantidad de kilómetros recorridos, la velocidad de sensado y la velocidad del flujo. A fines de simplificar, vamos a asumir una velocidad de flujo constante que será indicada junto con los nodos del recorrido y la velocidad de sensado como cabecera del archivo de mediciones.

Como resultado del análisis, se pretende indicar la ubicación de cualquier anomalía respecto de los nodos distribuidores de la tubería más cercanos. Esto es, indicar el nodo de partida, nodo hacia donde es necesario avanzar y la distancia a recorrer para llegar hasta la anomalía.

## Formato de Línea de Comandos

Para invocar el programa es necesario contar con un archivo de binario de mediciones, un archivo de tuberías y un archivo de recorrido que serán indicados como argumentos de entrada en ese orden:

```
./tp <arch-mediciones> <arch-tuberias> <arch-recorrido>
```

## Códigos de Retorno

El programa debe retornar 0 en caso de ejecución exitosa y 1 en caso de encontrar algún error.

## Archivos de Entrada

### Mediciones

El archivo de mediciones es un archivo binario con formato bigendian [6,7] y la siguiente secuencia:

velocidad-flujo	velocidad-sensado	cantidad-sensores	muestras
-----------------	-------------------	-------------------	----------

[m/h] (4 bytes)	[muestras por minuto] (4 bytes)	(4 bytes)	(N bytes)
--------------------	------------------------------------	-----------	-----------

Donde N es necesariamente un múltiplo de cantidad-sensores y podrá utilizarse junto con velocidad-sensado para calcular la distancia recorrida por el *pig* de la siguiente forma:

$$\text{distancia-total} = (\text{N} * \text{velocidad-flujo}) / (\text{cantidad-sensores} * \text{velocidad-sensado} * 60)$$

El bloque de muestras se puede representar de la siguiente forma:

sensor 1 (2 bytes)	sensor 2 (2 bytes)	...	sensor X (2 bytes)	sensor 1 (2 bytes)	sensor 2 (2 bytes)	...	sensor X (2 bytes)	...
muestras instante 1				muestras instante 2				...

## Tuberías

El archivo de tuberías es de texto plano indicando los distintos nodos y las distancias que hay entre uno y otro. Cada línea del archivo indica un par de nodos y la distancia que los separa de la siguiente forma:

```
<nombre-nodo-inicio>,<distancia>,<nombre-nodo-fin>\n
```

Donde `<distancia>` está dada en metros y los nombres de los nodos son únicos; esto es, cada aparición de un nombre repetido refiere al mismo nodo de distribución. Los nombres de nodos tienen como máximo 20 caracteres.

Se puede asumir que los archivos de tuberías están siempre bien formados y que los circuitos que conforman no tienen errores.

## Recorrido

Este archivo de texto plano es el más simple de interpretar. Posee un nombre de nodo por cada línea y de esa forma define el recorrido que hace el *pig*. El formato de cada línea es muy simple:

```
<nombre-nodo>\n
```

Se puede asumir que los archivos de recorrido están siempre bien formados e indican nombres válidos de nodos.

## Entrada y Salida Estándar

El programa utiliza la salida estándar para imprimir el resultado del análisis. El formato de la salida es en modo texto, con una línea por cada anomalía encontrada con la siguiente forma:

```
{anomalia} {nodo-cercano}->{nodo-siguiente} ({distancia-en-metros}m)\n
```

Donde los nodos son indicados mediante su nombre y `{anomalia}` puede ser únicamente:

- CORROSION
- RUPTURA

En la descripción, {nodo-cercano} indica el primer nodo del tramo que está recorriendo el *pig* y {nodo-siguiente} el siguiente nodo del tramo. Por otro lado {distancia-en-metros} será un valor flotante redondeado a 2 decimales que corresponde a la distancia entre el punto de anomalía y el {nodo-cercano}. Los decimales deben ser impresos siempre aunque sean cero (".00").

El programa no utiliza la entrada estándar.

## Salida Estándar de Error

El sistema debe abortar la ejecución y reportar un mensaje por la salida estándar de error frente a las siguientes situaciones:

- Archivo inexistente:
  - Motivo: las rutas de archivos indicadas al inicio del programa no coinciden con archivos accesibles.
  - Mensaje de error  
■ Archivo inválido\n
- Cantidad de valores sensados no es múltiplo de la cantidad de sensores.
  - Motivo: se ha perdido la medición de uno o más sensores.
  - Mensaje de error  
■ Cantidad de muestras incorrectas\n

En cualquiera de los casos, el código de retorno es 1. La detección sobre la cantidad de valores sensados debe demorarse el mayor tiempo posible, es decir, el reporte del error debe ocurrir sólo al llegar al final del archivo.

## Ejemplos de Ejecución

### Un Sensor

Recibimos los siguientes archivos de entrada:

mediciones.dat	tuberias.txt	recorrido.txt
00 00 00 32 00 00 00 06 00 00 00 01 00 28 00 36 00 4A 00 62 00 7B 00 93 <b>00 A7 00 B6 00 BE 00 BE 00 B6 00 A7</b> 00 93 00 7B <b>01 90</b> 00 4A 00 36 00 28 00 20 00 20 00 27 00 36 00 4A 00 62 00 7B 00 93 <b>00 A7 00 B6</b>	A, 2, B B, 1, C C, 3, D B, 2, D D, 5, E	A B D

Análisis de los datos:

- Según el archivo mediciones.dat podemos observar:
  - Velocidad del fluido: 50 m/h
  - Velocidad del sensado: 6 muestras por minuto
  - Cant. sensores: 1
  - Umbral muestras corrosión:  $0,5 \text{ m} * 60 \text{ min/h} * 6 \text{ muestras/min} * (50 \text{ m/h})^{-1} = 3,6 \text{ muestras} \sim 4$

muestras (para garantizar la medición de los 0,5 m)

- Según tuberías.txt y recorrido.txt determinamos:
  - Distancia total del recorrido:  $2\text{m} + 2\text{m} = 4\text{m}$
  - Cant. total de muestras esperadas:  $4 * 60 * 6 * 1/50 = 28,8$  muestras ~ 28 muestras (cantidad total antes antes del finalizar el recorrido)
- Anomalías encontradas:
  - Entre las muestras 7 y 12 (6 muestras) se observan valores superiores a 0x00A0 con lo cual se debe informar la ocurrencia de una CORROSION.
    - La muestra 7 ocurre en la distancia:  $7 \text{ muestras} * (6 \text{ muestras/min})^{-1} * (60 \text{ min/h})^{-1} * 50 \text{ m/h} \sim 0,97 \text{ m}$
    - La distancia 0,97m está ubicada entre los nodos A y B, a 0,97m de A.
  - En la muestra 15 se observa un valor mayor a 0x00FF con lo cual se debe informar una RUPTURA
    - La muestra 15 ocurre en la distancia:  $15 * 6^{-1} * 60^{-1} * 50 \text{ m} \sim 2,08 \text{ m}$
    - La distancia 2,08m está ubicada entre los nodos B y D, a 0,08m de B.
  - Entre las muestras 27 y 28 (2 muestras) se observan valores superiores a 0x00A0 pero no alcanza la cantidad exigida por el umbral.

Si ejecutamos el sistema con la siguiente línea de comandos:

```
./tp mediciones.dat tuberías.txt recorrido.txt
```

obtenemos la siguiente salida:

STDOUT	STDERR
CORROSION A->B (0.97m) RUPTURA B->D (0.08m)	

## Dos Sensores

Recibimos los siguientes archivos de entrada:

mediciones.dat	tuberías.txt	recorrido.txt
00 00 00 46 00 00 00 0A 00 00 00 02 00 29 00 19 00 37 00 27 00 4B 00 3B 00 63 00 53 00 7C 00 6C 00 94 00 84 <b>00 A8 00 98 00 B7 00 A7 00 BF 00 AF</b> <b>00 BF 00 AF 00 B7 00 A7 00 A8 00 A7</b> 00 94 00 84 00 7C 00 6C 00 63 00 53 00 4B 00 3B 00 37 00 27 00 29 00 19 00 21 00 11 00 21 00 11 00 28 00 18 00 37 00 27 00 4B 00 3B 00 63 00 53 00 7C 00 6C 00 94 00 84 <b>00 A8 00 98</b> <b>00 B7 00 A7 00 BF 00 AF 00 BF 00 AF</b>	A, 2, B B, 1, C C, 3, D B, 2, D D, 5, E	A B D

00 B7 00 98 00 A8 00 98 00 94 00 84 00 7C		
--	--	--

#### Análisis de los datos:

- Según el archivo mediciones.dat podemos observar:
  - Velocidad del fluido: 70 m/h
  - Velocidad del sensado: 10 muestras por minuto
  - Cant. sensores: 2
  - Umbral muestras corrosión:  $0,5 \text{ m} * 60 \text{ min/h} * 10 \text{ muestras/min} * (70 \text{ m/h})^{-1} = 4,29 \text{ muestras} \sim 5 \text{ muestras}$
- Según tuberias.txt y recorrido.txt determinamos:
  - Distancia total del recorrido:  $2\text{m} + 2\text{m} = 4 \text{ m}$
  - Cant. total de muestras esperadas:  $4 * 60 * 10 * 1 / 70 = 34,3 \text{ muestras} \sim 34 \text{ muestras}$
  - El archivo presenta 34 muestras para el sensor 1 y 33 muestras para el sensor 2 con lo cual debe reportar un error por "Cantidad de muestras incorrectas"
- Anomalías encontradas:
  - Sensor 1
    - Entre las muestras 7 y 12 (6 muestras) se observan valores superiores a 0x00A0 con lo cual se debe informar la ocurrencia de una CORROSION.
      - La muestra 7 ocurre en la distancia:  $7 * 1/10 * 1/60 * 70 \text{ m} \sim 0,82 \text{ m}$
      - La distancia 0,82m está ubicada entre los nodos A y B, a 0,82m de A.
    - Entre las muestras 27 y 32 (6 muestras) se observan valores superiores a 0x00A0 con lo cual se debe informar la ocurrencia de una CORROSION.
      - La muestra 27 ocurre en la distancia:  $27 * 1/10 * 1/60 * 70 \text{ m} \sim 3,15 \text{ m}$
      - La distancia 3,15m está ubicada entre los nodos B y D, a 1,15m de A.
  - Sensor 2
    - Entre las muestras 8 y 12 (5 muestras) se observan valores superiores a 0x00A0 con lo cual se debe informar la ocurrencia de una CORROSION.
      - La muestra 8 ocurre en la distancia:  $8 * 1/10 * 1/60 * 70 \text{ m} \sim 0,93 \text{ m}$
      - La distancia 0,93m está ubicada entre los nodos A y B, a 0,93m de A.
    - Entre las muestras 28 y 30 (3 muestras) se observan valores superiores a 0x00A0 pero no alcanza la cantidad exigida por el umbral.

Si ejecutamos el sistema con la siguiente línea de comandos:

```
./tp mediciones.dat tuberias.txt recorrido.txt
```

obtenemos la siguiente salida:

STDOUT	STDERR
CORROSION A->B (0.82m) CORROSION A->B (0.93m) CORROSION B->D (1.15m)	Cantidad de muestras incorrectas



# Restricciones

La siguiente es una lista de restricciones técnicas exigidas por el cliente:

1. El sistema debe desarrollarse en ISO C (C99).
2. Está prohibido el uso de variables globales.
3. Está prohibido cargar el archivo de mediciones enteramente en memoria.

# Referencias

[1] Comparación métodos UT y MFL para detección de fugas:

<http://www.ndt.net/article/wcndt00/papers/idn701/idn701.htm>

[3] Equipo de control por MFL guiado por el fluido: <https://www.youtube.com/watch?v=hk-nq8LMR6I>

[2] Equipo de control por flujo magnético guiado por humanos:

<https://www.youtube.com/watch?v=J0GzshNLXyl>

[4] Ejemplos de tuberías y herramientas para inspección de tubos:

<https://www.nde-ed.org/AboutNDT/SelectedApplications/PipelineInspection/PipelineInspection.htm>

[5] Ejemplo de productos para inspección de tubos:

<https://www.puretechltd.com/technologies-brands/pure-mfl>

[6] Instructivo para detección y programación tomando en cuenta endianness:

<http://www.ibm.com/developerworks/aix/library/au-endianc/>

[7] Explicación simple sobre endianness:

<http://betterexplained.com/articles/understanding-big-and-little-endian-byte-order/>