

## 1. Objetivos

- 1.1 Comprender el concepto de códigos binarios aplicados en el diseño de sistemas digitales  
Realizar un programa en C++ que realice la conversión de números y obtenga la codificación Hamming de un número de 12 bits de datos
- 1.2 Comprender el concepto de paridad en la detección de errores para la transmisión de datos binarios  
Manipular correctamente información binaria en distintos códigos.

## 2. Descripción

Los medios de transmisión por lo general introducen errores en los datos binarios; en el ambiente industrial principalmente por la cantidad de cargas inductivas: motores, transformadores, entre otros, se producen errores en los datos provenientes desde un sensor hacia una computadora o dispositivo de control programable (PLC). El desarrollo de sistemas confiables y estándares de comunicación en ambientes industriales como la interfase RS-485 obligan a utilizar códigos de detección y corrección de errores.

El Código Hamming permite detectar y corregir errores en la transmisión de datos binarios entre dos sistemas distantes. Este código utiliza varios bits de paridad insertados dentro de la trama de datos en lugares definidos y detectando paridad en el campo de datos también definida.

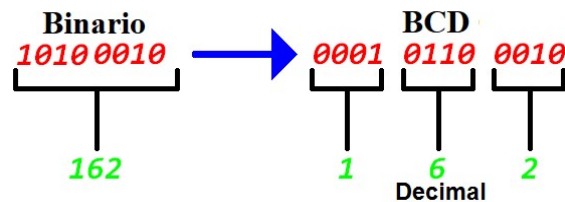
## 3. Metodología

El trabajo debe ser realizado en grupos de **4 estudiantes**, cada grupo debe traer su propia portátil para presentar el programa funcionando el día de la entrega del proyecto.

Investigar inicialmente cuál es la forma de codificación Hamming para luego plantear el algoritmo de solución.

## 4. Requerimientos

- Realizar un programa en C++ ejecutable (.exe) que le permita al usuario ingresar un número binario de 12 bits, si se ingresa un número no binario, el programa debe indicar el error y pedir otro número.
- Internamente el programa convierte ese número binario en 3 dígitos hexadecimales que se debe mostrar como una de las salidas del programa
- convertir el dato binario en uno decimal y mostrarlo en pantalla también
- seguidamente debe mostrarse el dato BCD correspondiente al decimal anterior como se muestra en el siguiente ejemplo con 8 bits:



- Permitir al usuario definir la paridad (par/impar)
- Codificar en Hamming la hilera introduciendo en los lugares definidos los bits de paridad  $P_1, P_2 \dots$
- Generar las tablas 1 y 2 con la información allí indicada ajustada para los 12 bits de datos.
- Presentar en pantalla la hilera final en binario aplicada la codificación correspondiente en la forma mostrada en la tabla No. 1
- Permitir al usuario cambiar uno de los bits de la hilera original para obligar a una falla en cualquier bit
- Detectar y presentar en pantalla el número de bit donde ocurre la falla (no haga comparación de hileras, el código Hamming permite calcular la posición del bit erróneo). El error debe mostrarse como en la tabla No. 2

- El programa debe permitir el ingreso de un nuevo dato o terminar la aplicación

**tabla No 1 Cálculo de los bits de paridad en el código Hamming**

	p <sub>1</sub>	p <sub>2</sub>	d <sub>1</sub>	p <sub>3</sub>	d <sub>2</sub>	d <sub>3</sub>	d <sub>4</sub>	p <sub>4</sub>	d <sub>5</sub>	d <sub>6</sub>	d <sub>7</sub>
<b>Palabra de datos (sin paridad):</b>			<b>0</b>		<b>1</b>	<b>1</b>	<b>0</b>		<b>1</b>	<b>0</b>	<b>1</b>
<b>p<sub>1</sub></b>	<b>1</b>		0		1		0		1		1
<b>p<sub>2</sub></b>		<b>0</b>	0			1	0			0	1
<b>p<sub>3</sub></b>				<b>0</b>	1	1	0				
<b>p<sub>4</sub></b>								<b>0</b>	1	0	1
<b>Palabra de datos (con paridad):</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>

**tabla No. 2 Comprobación de los bits de paridad (con primer bit de la derecha cambiado)**

	p <sub>1</sub>	p <sub>2</sub>	d <sub>1</sub>	p <sub>3</sub>	d <sub>2</sub>	d <sub>3</sub>	d <sub>4</sub>	p <sub>4</sub>	d <sub>5</sub>	d <sub>6</sub>	d <sub>7</sub>	Prueba de paridad	Bit de paridad
<b>Palabra de datos recibida:</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	
<b>p<sub>1</sub></b>	<b>1</b>		0		1		0		1		0	Error	<b>1</b>
<b>p<sub>2</sub></b>		<b>0</b>	0			1	0			0	0	Error	<b>1</b>
<b>p<sub>3</sub></b>				<b>0</b>	1	1	0					Correcto	0
<b>p<sub>4</sub></b>								<b>0</b>	1	0	0	Error	<b>1</b>

## 5. Evaluación

interfase de usuario amigable, tablas, etc.	10%
conversión binario HEXADECIMAL	15%
conversión binario DECIMAL	15%
conversión DECIMAL BCD	10%
código Hamming	50%

## 6. fuente de información

<https://es.wikipedia.org/wiki/Wikipedia:Portada>

## 7. Fecha de Entrega

Martes 19 de marzo 9:30 am