

## 1 Capítulo I EL PROYECTO

### 1.1 Presentación Del Proyecto

#### 1.1.1 Título

ENTORNO SIMULADO PARA UN CIRCUITO EN EL CUAL SE REALICE ENTRADA Y SALIDA DE DATOS POR PUERTO USB DE ARQUITECTURA ABIERTA.

#### 1.1.2 Carrera

Ingeniería Informática.

#### 1.1.3 Duración del proyecto

Doce Meses.

#### 1.1.4 Área\línea de investigación priorizada

- Investigación Robótica\Electrónica.
- Investigación en Programación y desarrollo.

### 1.2 Personal vinculado al proyecto

#### 1.2.1 Director(a) del proyecto

Apellido Paterno	Apellido Materno	Nombres	C.I.
	Castrillo	Magali Antonia	7169790
Teléfono	Celular	Correo	Firma
	75136089	<a href="mailto:magui_16_13@hotmail.com">magui_16_13@hotmail.com</a> <a href="mailto:magui.06.13@gmail.com">magui.06.13@gmail.com</a>	

**Tabla 1 Información personal del Director del Proyecto.**

### 1.2.2 Equipo de trabajo Del Laboratorio de Robótica

<b>Nombre:</b> Laboratorio de Robótica, Carrera de Informática, UAJMS			
<b>Dirección:</b> Campus Universitario, Zona el Tejar			
<b>Nombre y Apellidos</b>	<b>Cargo</b>	<b>C.I.</b>	<b>Firma</b>
Ing. Marcelo Céspedes Machicao	Asesor		

**Tabla 2 Información del equipo de trabajo del laboratorio de Robótica**

### 1.2.3 Actividades previstas para los integrantes del equipo de investigación

<b>Responsable*</b>	<b>Actividades</b>
Director  Univ. Magali Antonia Castrillo	El director debe ser responsable de la producción de los componentes, hacer el seguimiento de los supuestos, recomendar acciones correctivas cuando estas sean necesarias, informar a los Docentes tutores de la materia Taller III como a al Laboratorio de Robótica sobre los problemas que pueden ocasionar que el programa se concluya con normalidad.

**Tabla 3 Actividades previstas para el director del proyecto**

### 1.3 Descripción del proyecto

### 1.4 Resumen ejecutivo del proyecto

El uso del puerto USB está bastante difundido a través de diferentes equipos y dispositivos en una variedad de áreas como en el académico, de comunicación, industriales y otros; y es una tecnología creciente, cada vez de empleo más común debido a sus características de transmisión de datos y a su facilidad de manejo.

En la actualidad el puerto serie y el puerto paralelo ya quedaron atrás. Con la súper población de ordenadores portátiles, el espacio de acción de los enlaces por puerto serie se redujeron, lo mismo ocurrió con las conexiones por puerto paralelo. Ya ni siquiera los ordenadores personales traen un puerto de este tipo utilizado antiguamente para las impresoras hoy todo trae conectividad USB. El mouse, el teclado, la impresora multifunción, los discos duros portátiles, el modem, etc., todo se conecta a través de un puerto USB.

El laboratorio de Robótica de la Carrera de Informática de la UAJMS, tiene planificado el desarrollo de sistemas físicos contralados o monitorizados por computador; para este propósito requiere una interface de software y hardware de arquitectura abierta, para poder conectar y transferir datos desde cualquier dispositivo físico que cuente con un puerto USB específicamente. Ante esta coyuntura, el objetivo de este proyecto consiste en desarrollar una interface de hardware y software de arquitectura abierta para poder controlar un puerto mediante un microcontrolador de manera que podamos comunicar un ordenador y un microcontrolador ( $\mu\text{C}$ ) utilizando este tipo de tecnología por puerto USB, para determinar los límites de transferencia de datos, implementar una comunicación serial entre una computadora y un  $\mu\text{C}$  mediante puerto USB.

La Tecnología del puerto USB llego a ser imprescindible en la vida cotidiana del ser humano porque con el uso constante de diferentes dispositivos y el avance de la tecnología, el puerto llego a convertirse en una alternativa de habito general en la conexión envío y recepción de datos de distintos dispositivos.

El presente trabajo está dirigido a la investigación, documentar el mismo de acuerdo a las normas establecidas, para el desarrollo de software y simulación del proyecto.

Para ello, se implementara diferentes dispositivos montados en un SIMULADOR para ser controlados mediante el puerto USB para sus diferentes funcionamientos y un funcionamiento en conjunto en la transmisión y recepción de datos.

### 1.4.1 Descripción, Fundamentación y Justificación del Proyecto

En la actualidad el desarrollo de nuevos dispositivos conlleva una gran responsabilidad por los desarrolladores puesto que la gran demanda de los usuarios de estos mismos tienen cada vez exigencias más grandes respecto a las conexiones de varios tipos de dispositivos, el puerto USB (Bus de Serie Universal en inglés) siendo una alternativa para las distintas empresas desarrolladoras. Convirtiéndose en un estándar universal para las distintas conexiones de nuevos dispositivos. La metodología del empleo de estos puertos es muy sencilla y fácil de manejar por el usuario ya que no necesita saber nada en específico para su completo uso. Convirtiéndose en un puerto eficiente. El Departamento de desarrollo reconoce que existe gran ímpetu en el momento de desarrollo de nuevas aplicaciones para el uso de este puerto que ya es un estándar para todo ordenador. Con la coordinación y la adecuación correcta se puede decir que el puerto USB puede contener cualquier tipo de envío y recepción de datos dando también una conexión de corriente eléctrica lo cual es perfecto para el implemento de diferentes estructuras de dispositivos, más sencillo para una arquitectura abierta la cual su único requisito sea el de tener un puerto USB para su desarrollo.

Actualmente en nuestro medio el desarrollo de sistemas está enfocado a sistemas informáticos tradicionales ya sea de ofimática o de base de datos, es decir, la construcción de sistemas que no interacciona comúnmente con un entorno físico.

Debido a esto, nuestra tendencia es construir sistemas deterministas que son abstracciones del mundo real, por lo que nos lleva a tener problemas cuando deseamos diseñar o programar sistemas de control para el mundo real.

El presente proyecto está dividido en cuatro modelos, los mismos que se justifican por las siguientes razones:

**Análisis y diseño:** Todo proyecto comienza por un análisis del entorno, que tiene que ver con el conocimiento que se debe tener del trabajo de los diferentes dispositivos. Sin este previo análisis es imposible seguir con el diseño ya que el mismo se basa en los requerimientos establecidos en el análisis.

**Construcción del sistema:** La construcción del sistema está justificada ya que permite tener un control completo de los dispositivos y también se puede decir que es el único medio disponible para adquirir información en cuestión de segundos y que ayuda de forma directa a dichos dispositivos.

**Capacitación:** No puede haber un correcto funcionamiento del PRODUCTO si no se capacita al público para que pueda operarlo. Las ventajas que ofrece el PROYECTO están íntimamente relacionadas con el correcto uso que realice el usuario del sistema.

**Alcances:**

- Se realizara la simulación de la placa base.
- Se utilizara un canal común entre la plaqueta (de simulación) y el computador.
- Se utilizara los algoritmos propuestos, estudiados dentro de la malla curricular de la carrera.
- Se definirán los compiladores necesarios para la implementación de la aplicación.

**Limitaciones:**

- El Proyecto no será montado en una placa real.
- El Pic solo accede a la grabación de un programa en .hex.
- El proyecto no se realizara en la plataforma de Windows 7 de 64 bits.
- Poca accesibilidad a los componentes o dispositivos para la integración al microcontrolador.

**1.4.2 Objetivos****1.4.2.1 Objetivo General**

Crear un entorno simulado para un circuito en el cual se realice entrada y salida de datos por puerto PC/Microcontrolador mediante el puerto USB.

**1.4.2.2 Objetivos Específicos**

- Desarrollar el software, para la simulación de un dispositivo USB (circuito) de arquitectura abierta.
- Presentar el trabajo de grado a los estudiantes de Ingeniería informática
- Simular la conexión del circuito.

**1.4.3 Metodología de Investigación**

Este apartado realiza una descripción de algunas de las herramientas, conceptos de metodologías del modelado usados en este proyecto.

Este apartado empieza con la descripción de los conceptos básicos del lenguaje UML para el desarrollo de aplicaciones informáticas.

### 1.4.3.1 Lenguaje de Modelado Unificado UML

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar configurar, mantener y controlar la información sobre tales sistemas.

Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje del modelado pretende unificar la experiencia basada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser usado en herramientas iterativas de modelado visual que tengan modeladores de código.

### 1.4.3.2 Compilador PIC C (PCWHD)

Se decide utilizar PCWHD de la casa CSS porque nos permitir la programación desde un nivel cercano al programador (alto nivel), también brinda la posibilidad de controlar aspectos más cercanos al hardware (bajo nivel), como la manipulación directa de bits y bytes, por lo que se considera que es un lenguaje de nivel medio, más que de alto nivel, está ya sería una razón para decidirse por compiladores de C y no de otros lenguajes de alto nivel como el Basic o Pascal.

Aunque el desarrollo de programas para microcontroladores es posible realizarlo totalmente en Ensamblador, la utilización del lenguaje C supone una alternativa muy interesante por su rapidez, facilidad, y portabilidad, sin que esto signifique que este nuevo enfoque venga a sustituir definitivamente al Ensamblador, en casos en los que se necesita crear partes de código sujetas a determinadas restricciones ( reducido número de instrucciones, alta velocidad de ejecución, etc.) se puede implementar esa parte de código directamente en Ensamblador dentro de un programa en C, consiguiendo de este modo un código mucho más eficiente. Esta ventaja no la tienen otros compiladores de Basic o Pascal.

#### ➤ Ventajas:

- Es un lenguaje de alto nivel más cercano a la máquina.
- Puedes construir rutinas matemáticas fácilmente.
- Puede ser de ayuda al combinarlo con Ensamblador sobre todo en la gama alta.
- Se pueden crear macros con este lenguaje, para después simplificar el código en diferentes desarrollos.
- Es aceptado por la empresa fabricante Microchip. [w23]
-

➤ **Desventajas:**

- Los programas al compilarlos pueden resultar un poco extensos y pesados por ello debe tenerse en cuenta la capacidad de memoria de programa del PIC a utilizar.
- Con este lenguaje tampoco se puede controlar del todo los tiempos y los registros bit a bit. [w23]

### 1.4.3.3 Simulador Proteus7 Professional

Se decide utilizar Proteus7 Professional ya que es un software utilizado para la educación en miles de escuelas y universidades del mundo, Proteus7 Professional es un paquete de software de diseño asistido por ordenador de circuitos electrónicos. El paquete es un sistema de simulación de circuitos, con base en modelos de componentes electrónicos fabricados en PSpice.

Una característica distintiva del *paquete* de Proteus Professional es la posibilidad de simular el funcionamiento de los dispositivos programables: microcontroladores, microprocesadores, DSP y otros. Además, el paquete de Proteus Professional es un diseño de sistema de placas de circuitos impresos. Proteus Professional puede simular el funcionamiento de los microcontroladores siguientes: PIC, Basic Stamp, etc.

En el presente proyecto se contempla la realización de dos componentes: Simular un Dispositivo USB de Arquitectura abierta y Capacitar el trabajo de grado a los estudiantes de la carrera de Ingeniería informática para el uso del mismo. Las Metodologías a utilizar se describen a continuación:

### 1.4.3.4 Simulación del Dispositivo USB

Para el desarrollo de la simulación del Dispositivo USB se utilizará la metodología de investigación y un lenguaje gráfico que maneja diagramas ya definidos para especificar o describir métodos o procesos, y definir un sistema.

Como también distintos algoritmos para el control del tráfico de datos y comandos.

**Fase de Inicio:** En esta fase se establece los requisitos del software y del hardware que cubrirá al proyecto por completo, se obtendrá la especificación de requerimientos.

**Fase de Elaboración:** En esta segunda fase el problema se analiza y comprende desde varios puntos de vista. Al final de la fase se tiene definida la arquitectura y el modelo de requisitos del sistema y del hardware empleando los diagramas y librerías correspondientes especificados en lenguaje C. Para su posterior compilación en .hex.

**Fase de Construcción:** En esta tercera fase se profundiza en el diseño de los componentes del sistema y de los dispositivos a emplearse y de manera iterativa se van añadiendo las funcionalidades al software a medida que se construyen y prueban,

permitiendo a la vez que se puedan ir incorporando cambios. Al final de esta fase se obtiene un sistema, un hardware (un entorno de circuito simulado) completamente operativo y la documentación (manual de usuario).

**Fase de Transición:** Se ocupa del traslado del software desde los entornos de desarrollo a los entornos del hardware (entorno del circuito simulado), en los que los usuarios finales (público en general) harán uso del sistema.

Para la Capacitación, la metodología a utilizar se fundamenta en los siguientes factores:

**Estructura:** El modelo básico de estructura de capacitación que se toma en cuenta es el modelo de capacitación simultánea, en el que se instruirá a todo el público simultáneamente, en un día.

#### 1.4.3.5 Capacitación del Sistema

**Diseño del programa de capacitación:** Se enfoca en el objetivo, conseguir que el público (estudiantes) maneje el sistema implementado sin dificultades, la disposición del público (estudiantes), los principios pedagógicos de aprendizaje que se tomarán en cuenta para esta capacitación son los de participación, repetición y retroalimentación.

**Grado de conocimientos sobre los distintos componentes por parte de los usuarios:** Se toma en cuenta el nivel de conocimiento de los usuarios sobre los componentes para impartir la instrucción básica y la capacitación del sistema.

**Implementación del programa de capacitación:** Se utiliza el método de uso más general; la capacitación en el trabajo, que proporciona la ventaja de trabajar directamente con el software logrando una ventaja directa sobre el sistema y la mayor optimización del hardware (entorno de un circuito simulado) al momento de su empleo por parte de los estudiantes.

**Estilo para impartir la capacitación:** Se realiza el estilo participativo, con un cronograma accesible para el público (estudiantes) de la universidad Juan Misael Saracho.

Por otro parte, la tecnología a utilizar se fundamenta en que el software que se desarrolla es una interacción para con la simulación, las herramientas a usar son las siguientes:

- Se desarrolla la interfaz usuario en el lenguaje de programación Java, porque es una de las tecnologías seguras para el desarrollo de programas, es multiplataforma, de fuente abierta y nos proporciona la mejor solución al permitir crear programas modulares, visuales de fácil manejo para el usuario y facilitar el mantenimiento del software.

- El software utilizado dentro del dispositivo controlador de los dispositivos es en el lenguaje de programación C. Ya que por su flexibilidad al momento de crear funciones esto permite velocidad y flexibilidad para su posterior compilación en .hex.

#### **1.4.4 Resultados esperados**

Se espera la aprobación del laboratorio de Robótica de la carrera de ingeniería Informática según se vaya desarrollando el proyecto **“Entorno simulado para un circuito en el cual se realice entrada y salida de datos por puerto usb de arquitectura abierta.”** Hasta la culminación del mismo en la gestión 2013 mediante el Desarrollo de un sistema de comunicación bidireccional, la implementación del hardware de una interface USB de arquitectura abierta en un simulador.

#### **1.4.5 Transferencia de resultados**

##### **1.4.5.1 Medios y estrategias para la transferencia de resultados**

La transferencia de resultados se realizara mediante:

- Convenios con el Laboratorio de Robótica de la carrera de ingeniería informática donde existirá la supervisión del jefe de laboratorio para un mejor control mediante se desarrolla el sistema de comunicación bidireccional, configurable, la implementación del hardware de una interface USB de arquitectura abierta en un simulador de circuitos.

##### **1.4.5.2 Grupo de beneficiarios de los resultados.**

El grupo de beneficiarios del presente proyecto son:

- La dirección de investigación científica y tecnológica.
- La carrera de Ingeniería Informática.

### 1.4.6 Cronograma de actividades

Nº	Actividad	Nº días	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
1	Investigación sobre la comunicación PC/Microcontrolador.	120			✓	✓	✓	✓						
2	Presentación del Perfil	30			✓	✓								
3	Presentación de las modificaciones del perfil.	15				✓								
4	Documentar Proyecto.	120					✓	✓	✓	✓				
5	Desarrollo del Sistema.	120								✓	✓	✓	✓	
6	Desarrollo de la presentación y manual de usuario.	30									✓	✓		
7	Capacitación del proyecto	1											✓	
8	Presentación del proyecto	1												✓
9														
10														
11														

Tabla 2 Cronograma de actividades del proyecto

## 1.4.7 Marco Lógico de Proyecto

### 1.4.7.1 Árbol de Problemas

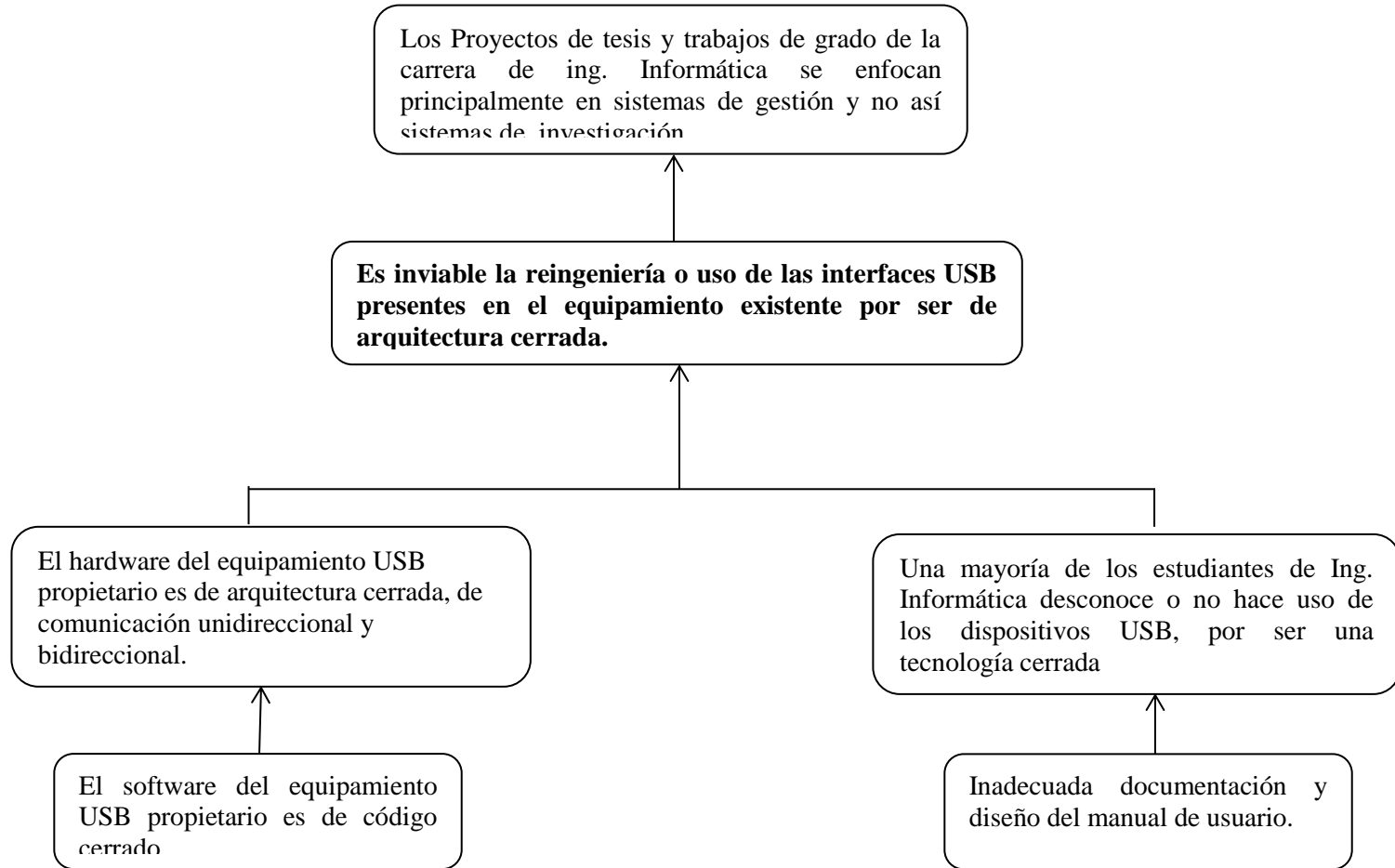


Tabla 5 Árbol de problemas

### 1.4.7.2 Árbol de Objetivos

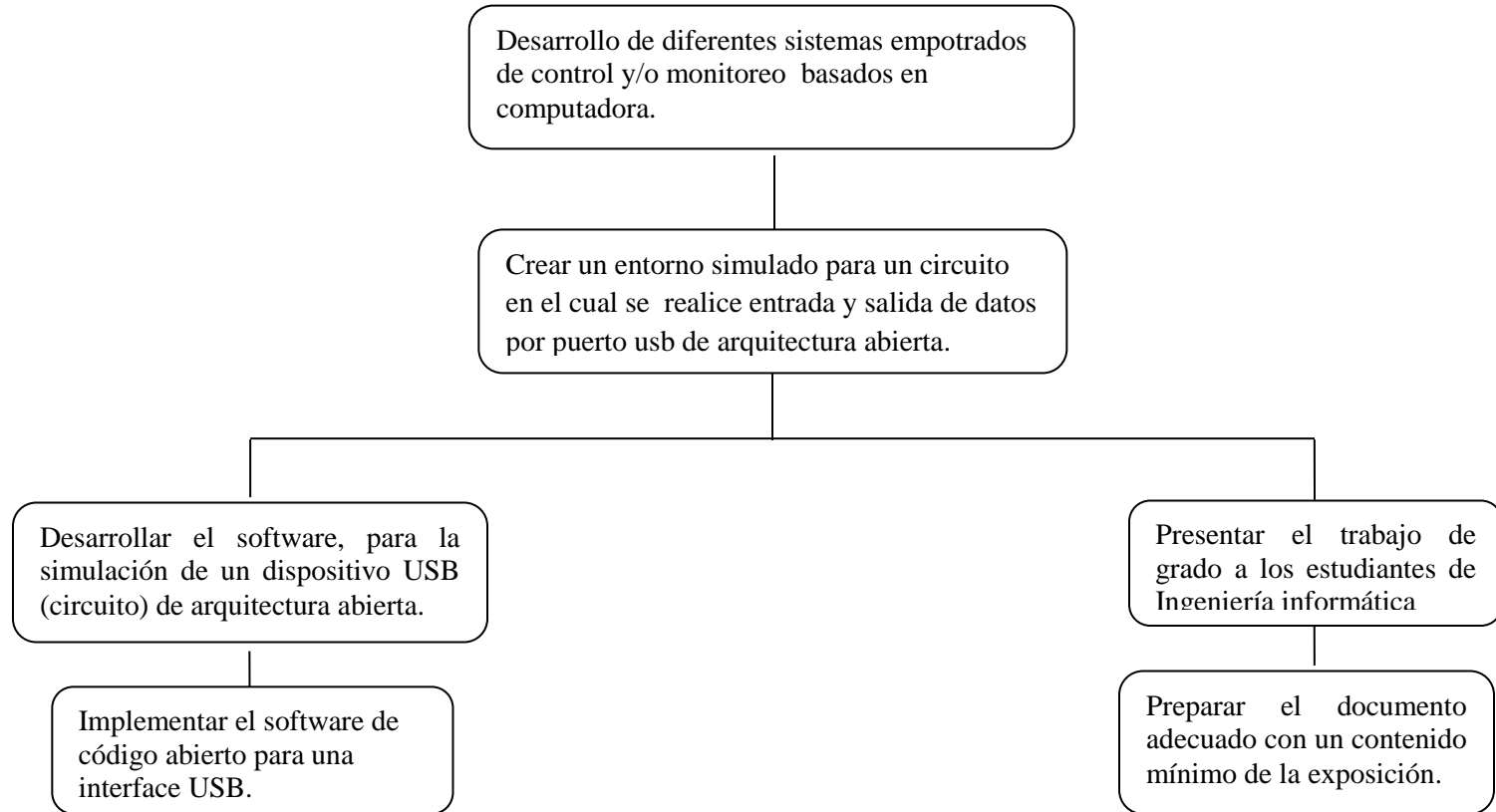


Tabla 6 Árbol de objetivos

### 1.4.7.3 Matriz de Marco Lógico

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
<p><b>Fin</b> Desarrollo de diferentes sistemas empotrados de control y/o monitoreo basados en computadora.</p>	<p>➤ Futuras investigaciones de los trabajos de grado o tesis en aplicaciones de la Robótica.</p>	<p>➤ Informe del encargado del laboratorio de Robótica.</p>	<p>➤ Se brindan las condiciones para el desarrollo de sistemas empleando la interface USB.</p>
<p><b>Objetivo General (Propósito)</b> Crear un entorno simulado para un circuito en el cual se realice entrada y salida de datos por puerto usb de arquitectura abierta.</p>	<p>➤ Al finalizar la gestión 2013 el 100% de este software estará implementado, capacitado y entregado. CALIDAD <math>F = \frac{\text{Núm. de componentes} * 100}{\text{Total de componentes}}</math></p>	<p>➤ Informe del encargado del laboratorio de Robótica.</p>	<p>➤ Se brindaran los ambientes e instrumentos necesarios para su desarrollo.</p>

<p><b>Objetivos Específicos (Componentes)</b></p> <ul style="list-style-type: none"> <li>✓ Desarrollar el software, para la simulación de un dispositivo USB (circuito) de arquitectura abierta.</li> <li>✓ Presentar el trabajo de grado a los estudiantes de Ingeniería informática</li> </ul>	<ul style="list-style-type: none"> <li>➤ Al finalizar noviembre del 2013 el software para un dispositivo USB estará desarrollado en un 100%. CALIDAD <math>F = \frac{\text{Núm. De elementos} * 100}{\text{Total de elementos}}</math></li> <li>➤ El 05/11/13 se ha capacitado al 53% de los estudiantes de la materia de Arquitectura de computadoras grupo 1. CALIDAD <math>F = \frac{\text{Núm. De estudiantes} * 100}{\text{Total de estudiantes}}</math></li> </ul>	<ul style="list-style-type: none"> <li>➤ Informe de prueba del encargado del laboratorio de Robótica, documentación de análisis y diseño aprobado por el mismo.</li> <li>➤ Listado de los participantes en la exposición del sistema.</li> <li>➤ Certificados de asistencia a la exposición del proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Se brindara los ambientes e instrumentos necesarios y suficientes para su desarrollo.</li> <li>➤ Interés y disposición de tiempo de los estudiantes de Ing. Informática.</li> </ul>
<p><b>Actividades</b></p> <ul style="list-style-type: none"> <li>✓ Implementar el software de código abierto para una interface USB.</li> <li>✓ Preparar el documento adecuado con un contenido mínimo de la exposición</li> </ul>	<ul style="list-style-type: none"> <li>➤ Fecha 15/06/13 al 30/11/12. Presupuesto 260 Bs.</li> <li>➤ Fecha 15/10/13 al 15/11/15. Presupuesto 1295 Bs.</li> </ul> <p><b>TOTAL: 1555Bs.</b></p>	<ul style="list-style-type: none"> <li>➤ Sistema Documentado y desarrollado.</li> <li>➤ Documento y pruebas del sistema</li> </ul>	<ul style="list-style-type: none"> <li>➤ Recursos suficientes para la elaboración del sistema de comunicación bidireccional.</li> <li>➤ Se brindara los ambientes e instrumentos necesarios para su desarrollo.</li> </ul>

**Tabla 7 Matriz de marco lógico**

### 1.5 Presupuesto / Justificación

ITEM	RUBROS	Aporte Universidad	Otro Aporte	TOTAL (Bs.)
<b>10000</b>	<b>SERVICIOS PERSONALES</b>			
	<b>12000 Empleados no Permanentes</b>			
	<b>1.5.1 Sub total rubro</b>			
<b>20000</b>	<b>SERVICIOS NO PERSONALES</b>			
	<b>21000. Servicios Básicos</b>		80	80
	<b>22000. Servicios de transporte</b>			
	<b>23000. Alquileres</b>			
	<b>24000. Mantenimiento y reparación</b>			
	<b>25000. Servicios Profesionales y Comerciales</b>			
	<b>1.5.2 Sub total rubro</b>			80
<b>30000</b>	<b>MATERIALES Y SUMINISTROS</b>			
	<b>31000. Alimentos y Productos Forestales</b>			
	<b>32000. Productos de Papel, Cartón e Impresos</b>		1195	1195
	<b>33000. Textiles y Vestuario.</b>			
	<b>34000. Productos Químicos, Combustibles y Lubricantes</b>			
	<b>39000. Productos Varios.</b>		100	100
	<b>1.5.3 Sub total rubro</b>			1295
<b>40000</b>	<b>ACTIVOS REALES</b>			
	<b>43000. Maquinaria y Equipo.</b>		180	180
	<b>46000. Descripción de estudios y proyectos para inversión</b>			
	<b>49000. Otros Activos</b>			
	<b>1.5.4 Sub total rubro</b>			180
	<b>1.5.5 TOTAL</b>			1555
	<b>TOTAL + 40% Incentivo</b>			2177

**Tabla 8 Resumen del presupuesto del proyecto**

**1) GRUPO 10000. SERVICIOS PERSONALES**

**a) SUB GRUPO 12000. Empleados no Permanentes**

Partida	Personal	Remuneración	Tiempo/meses	Total
12100	Personal Eventual			
<b>Total</b>				

**Tabla 9** Tabla de empleados no permanentes

**2) GRUPO 20000. SERVICIOS NO PERSONALES**

**a) SUB GRUPO 21000. Descripción de los gastos de servicios básicos**

Partida	Tipo de servicio básico *	Costo	Tiempo mes	Costo Total
21100	Comunicación			
21200	Energía Eléctrica			
21300	Agua			
21400	Servicios Telefónicos	10	8	80
<b>Total</b>				80

**Tabla 10** Tabla de servicios no personales

**b) SUB GRUPO 22000. Descripción de los gastos de viajes y transporte de personal**

Partida	Personal	Lugar	Nº de viajes	Costo unitario*	Costo total
22100	Pasajes				
<b>Total</b>					

**Tabla 11** Tabla de Descripción de viajes y transporte de personal

<b>Partida</b>	<b>Personal</b>	<b>Lugar</b>	<b>Duración (días)</b>	<b>Costo unitario*</b>	<b>Costo total</b>
22200	Viáticos				
22300	Fletes y Almacenamientos				
22600	Transporte de Personal				
<b>Total</b>					
<b>Total sub grupo 22000</b>					

**Tabla 12** Tabla de gastos de viáticos y transporte de personal

**c) SUB GRUPO 23000. Descripción de los gastos por concepto de alquileres de equipos y maquinarias**

<b>Partida</b>	<b>Alquiler de equipo y maquinaria</b>	<b>Costo unitario</b>	<b>Tiempo mes</b>	<b>Costo total</b>
23100	Alquiler de Edificios			
23200	Alquiler de Equipos y Maquinaria			
23300	Alquiler de Tierras y Terrenos			
<b>Total</b>				

**Tabla 13** Tabla de gastos de alquileres de equipo y maquinaria

**d) SUB GRUPO 24000. Descripción mantenimiento y reparación**

<b>Partida</b>	<b>Mantenimiento y reparación de equipo y maquinaria</b>	<b>Costo unitario</b>	<b>Tiempo mes</b>	<b>Costo total</b>
24100	Mantenimiento y Reparación de Edificios y Equipos			
24300	Otros Gastos por Mantenimiento y Reparación			
<b>Total</b>				

**Tabla 14** Tabla de gastos de mantenimiento y reparación

**e) SUB GRUPO 25000. Descripción de los gastos en servicios profesionales y comerciales**

<b>Partida</b>	<b>Tipo de servicio profesional y comercial *</b>	<b>Cantidad</b>	<b>Costo unitario</b>	<b>Tiempo Mes</b>	<b>Costo total</b>
25200	Estudios e Investigaciones				
25500	Publicidad				
25600	Imprenta (fotocopias, impresiones)				
25700	Capacitación de Personal				
25800	Estudios e Investigaciones Para Proyectos de Inversión				
<b>Total</b>					

**Tabla 15** Tabla de gastos de servicios profesionales y comerciales

**3) GRUPO 30000. MATERIALES Y SUMINISTROS**

**f) SUB GRUPO 31000. Descripción de los gastos Alimentos y Productos Agroforestales**

<b>Partida</b>	<b>Tipo de material *</b>	<b>Cantidad</b>	<b>Costo/Unitario</b>	<b>Total</b>
31110	Refrigerios y Gastos Administrativos			
31200	Alimento para Animales			
31300	Productos Agroforestales y Pecuarios			
<b>Total</b>				

**Tabla 16** Tabla de gastos en alimentos y productos agroforestales

**g) SUB GRUPO 32000. Descripción del gasto de Productos de Papel, Cartón e Impresos**

<b>Partida</b>	<b>Tipo de material *</b>	<b>Cantidad</b>	<b>Costo/Unitario</b>	<b>Total</b>
32100	Papel de Escritorio	2	35	70
32200	Productos de Artes Gráficas, Papel y Cartón(impresiones)	1500	0.25	1125
32300	Libros y Revistas			
32400	Textos de Enseñanza			
32500	Periódicos			
<b>Total</b>				1195

**Tabla 17** Tabla de gastos en papel, cartón e impresos

#### h) SUB GRUPO 33000. Descripción del gasto en textiles y vestuario

Partida	Productos textiles y vestuarios	Cantidad	Costo/Unitario	Total
33100	Hilados y Telas			
33200	Confecciones Textiles			
33300	Prendas de vestir			
33400	Calzados			
<b>Total</b>				

**Tabla 18** Tabla gastos en textiles y vestuario

#### i) SUB GRUPO 34000. Combustibles, Productos Químicos, Farmacéuticos y Otros

Partida	Combustibles, Productos Químicos, Farmacéuticos y Otros	Cantidad	Costo/Unitario	Total
34110	Combustibles y Lubricantes para Consumo			
34200	Productos químicos y Farmacéuticos			
34400	Productos de Cuero y Caucho			
34500	Productos de Minerales no Metálicos y Plásticos			
34600	Productos Metálicos			
34700	Minerales			
34800	Herramientas Menores			
<b>Total</b>				

**Tabla 19** Tabla de gastos de productos químicos

#### k) SUB GRUPO 39000. Descripción del gasto en productos varios

Partida	Productos de cuero y caucho	Cantidad	Costo/Unitario	Total
39100	Material de Limpieza			
39400	Instrumental Menor Médico – Quirúrgico			
39500	Útiles de Escritorio y de Oficina			
39700	Útiles y Materiales Eléctricos	2	50	100
39800	Otros Repuestos y Accesorios			
<b>Total</b>				100

**Tabla 20** Tabla de gastos de productos varios.

**4) GRUPO 40000. ACTIVOS REALES**

**l) SUB GRUPO 43000. Descripción del gasto de Maquinaria y Equipo**

<b>Partida</b>	<b>Tipos de productos</b>	<b>Cantidad</b>	<b>Costo/Unitario</b>	<b>Total</b>
43100	Equipo de Oficina y Muebles			
43200	Maquinaria y Equipo de Producción			
43300	Equipos de Transporte, Tracción y Elevación			
43400	Equipo Médico y de Laboratorio			
43600	Equipo Educativo y Recreativo			
43700	Otra Maquinaria y Equipo(microcontroladores)	2	90	180
<b>Total</b>				180

**Tabla 21 Tabla de gastos de maquinaria y equipo**

**m) SUB GRUPO 46000. Descripción de estudios y proyectos para inversión**

<b>Partida</b>	<b>Productos textiles y vestuarios</b>	<b>Cantidad</b>	<b>Costo/Unitario</b>	<b>Total</b>
46100	Para Construcción de Bienes de Dominio Privado			
<b>Total</b>				

**Tabla 32 Tabla de gastos de estudios y proyectos para la investigación**

**n) SUB GRUPO 49000. Descripción del gasto de Otros Activos**

<b>Partida</b>	<b>Tipos de productos *</b>	<b>Cantidad</b>	<b>Costo/Unitario</b>	<b>Total</b>
49100	Activos Intangibles			
49200	Compra de Bienes Muebles Existentes (Usados)			
49300	Semovientes y otros Animales			
49900	Otros Activos			
<b>Total</b>				

**Tabla 23 Tabla de gastos de otros servicios**

## 1.6 Curriculum Vitae

### 1.6.1 Antecedentes personales

<b>Apellido Paterno</b>	Castrillo <b>Apellido Materno</b>	Magali Antonia <b>Nombre</b>	7169790 <b>CI.</b>
13/06/1989 <b>Fecha de nacimiento:</b>	Femenino <b>Sexo</b>	La victoria- Provincia Méndez <b>Dirección</b>	
Tarija <b>Ciudad</b>	66-62385 <b>Teléfono Domicilio</b>	75136089 <b>Celular</b>	magui_16_13@hotmail.com <b>Correo electrónico</b>

**Tabla 24 Tabla de información personal**

### 1.6.2 Antecedentes académicos

<b>Título obtenido</b>	<b>Universidad</b>	<b>País</b>	<b>Año</b>
Cursos de Autodesk Maya	Universidad Autónoma Juan Misael Saracho.	Bolivia	2011
XIX Congreso nacional de Ciencias de la Comunicación (CCBOL).	Universidad Autónoma Gabriel René Moreno.	Bolivia	2012

**Tabla 25 Tabla de antecedentes académicos**

### 1.6.3 Participación en proyectos de investigación

Título proyecto	Institución	Cargo	Año

**Tabla 26** Tabla de proyectos en los que se participó

### 1.6.4 Publicaciones realizadas (libros, revistas, compendios y otros)

Autor	Tipo de publicación, Año, título, volumen, páginas, editorial

**Tabla 27** Tabla de publicaciones realizadas

### 1.6.5 Presentaciones realizadas

Título	Nombre del evento, lugar, fecha, año

**Tabla 4** Tabla de presentaciones realizadas

### 1.6.6 Antecedentes en docencia

#### a) Experiencia Docente

Experiencia Docente		
Institución	Dedicación	Años
Laboratorio de química de la U.A.J.M.S.	20 horas Académicas	1

**Tabla 29** Tabla de experiencia docencia

#### j) Jerarquía Docente (en la presente gestión, y de gestiones anteriores)

Categoría docente	Carrera/Departamento	Tiempo dedicación	Gestión
Instructor del curso de capacitación EN Microsoft Excel.	Departamento de Química	20 horas académicas.	2009
Ayudante de Arquitectura de Computadoras	Departamento de Ingeniería informática	Un semestre	2012

**Tabla 30** Tabla de jerarquía docente

## **1.7 MARCO TEÓRICO**

### **1.7.1 CAPÍTULO 1.-**

#### **1.7.1.1 Puerto USB.**

##### **1.7.1.1.1 ¿Qué es un puerto USB?**

Un puerto USB es una entrada o acceso para que el usuario pueda compartir información almacenada en diferentes dispositivos como una cámara de fotos, un pen drive, entre otros, con un computador. Las siglas USB quieren decir Bus de Serie Universal que es un estándar que define un bus utilizado para conectar periféricos al Pc. [\[w2\]](#)

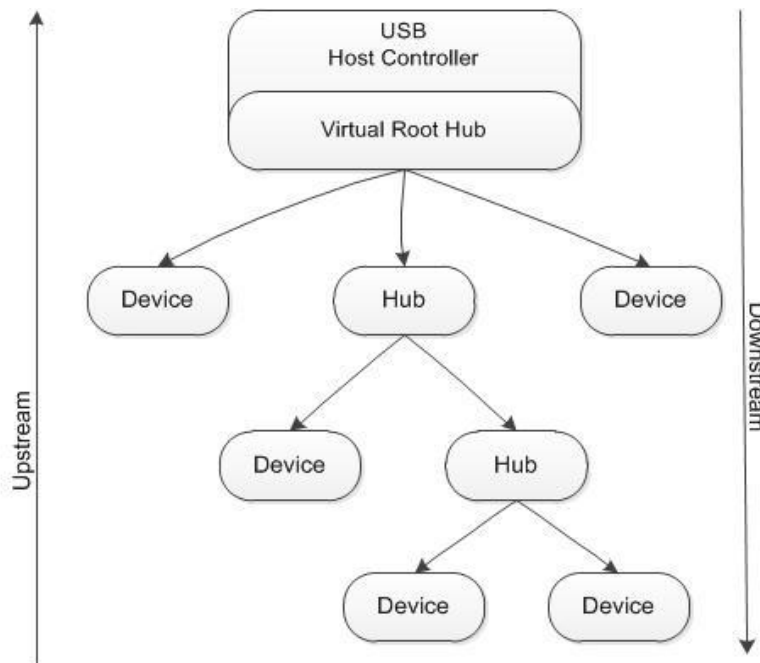
##### **1.7.1.1.2 Características de puerto USB.**

La principal característica que tiene es que la conexión es muy sencilla, ya que maneja un único conector para conectar a través de un bus serie todos los dispositivos.

Un puerto USB permite conectar en una topología especial de tipo estrella hasta 127 dispositivos y protocolos que permiten la detección y configuración automática de los dispositivos conectados ya es un estándar en los ordenadores de última generación, que incluyen al menos cuatro puertos USB 2.0 en los más modernos, y algún USB 1.1 en los más anticuados. Cabe asimismo destacar que USB 2.0 nunca llegará a reemplazar completamente a USB 1.0, ya que coexisten algunos tipos de dispositivos, como los dispositivos de interfaz humana (human interface devices) (HID) (teclados, ratones y joysticks), que no requieren las altas velocidades que consigue esta nueva versión y que únicamente encarecerían el dispositivo [5].

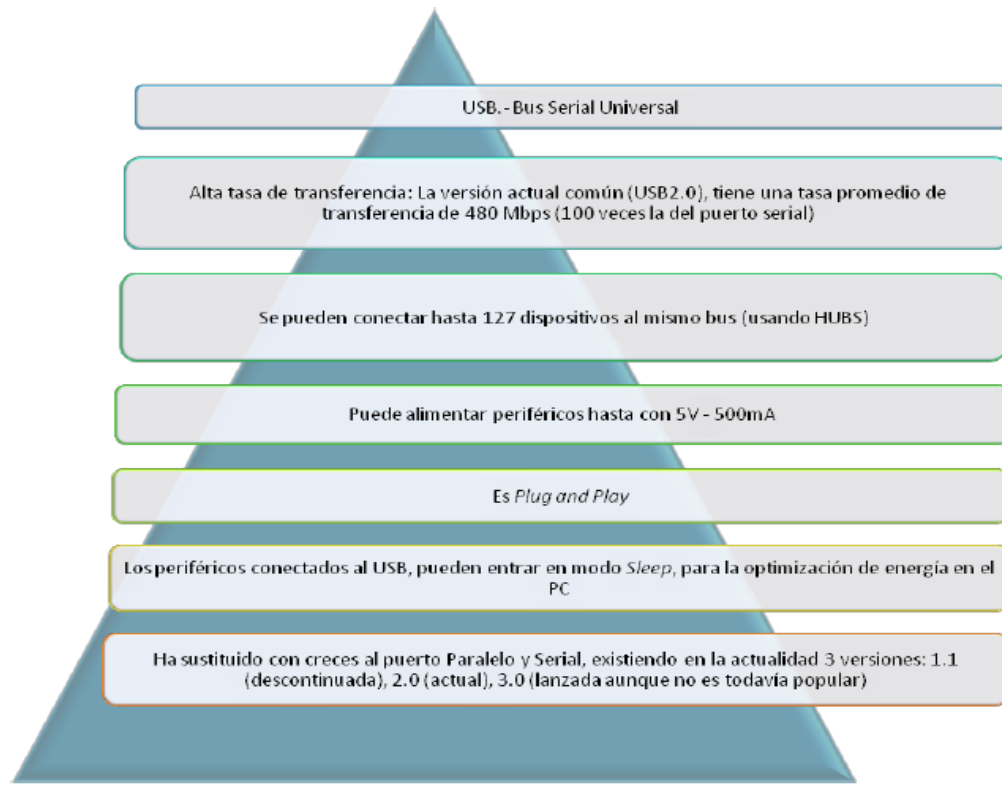
- ✓ El USB1.0 soporta dos tasas de transferencia diferentes, una baja de 1,5 Mbps para la conexión de dispositivos lentos de bajo coste (joysticks, ratones) y otra alta de hasta 12 Mbps para la conexión de dispositivos que requieren un mayor ancho de banda (discos o CD-ROM).
- ✓ La versión 2.0, que multiplica la velocidad del bus por un factor de 30 o 40, llegando a alcanzar una velocidad de 480 Mbps, con una diferencia de costo casi inapreciable.
- ✓ Transferencia isócrona y asíncrona en el mismo bus.
- ✓ El modo isócrono consume un ancho de banda significativo. Por ello el PC impide este tipo de transferencia cuando el ancho de banda consumido supera el 90% del ancho de banda disponible.

- ✓ Una arquitectura fácilmente escalable para permitir la existencia de varios Controladores USB en un sistema.
- ✓ El Controlador USB configura el hub y controla y monitoriza los puertos Downstream.
- ✓ Posibilidad de conectar hasta 127 periféricos.
- ✓ Tiempo de respuesta garantizado (para audio y vídeo).
- ✓ Flexibilidad a nivel de banda de paso.
- ✓ Fiabilidad, control de errores.
- ✓ Perfectamente integrado en el PC, plug& play (conectar y usar)
- ✓ Coste reducido en la versión de baja velocidad (1,5 Mbits/s).
- ✓ Posible expansión del bus.



**Fig.1: topología especial de tipo estrella**

### 1.7.1.1.3 Características Generales del Puerto USB.



**Fig.2: Explicación del USB.**

### 1.7.1.1.4 Protocolo USB.

El protocolo de nivel físico se basa en tokens (testigos). El controlador USB transmite tokens que incluyen la dirección del dispositivo destino, y el dispositivo que detecta su dirección en el token responde y lleva a cabo la transferencia de datos con el controlador. De esta manera, el Controlador USB maneja la parte más compleja del protocolo, generando los tokens de transferencias de datos a 12 Mbps o a 1,5 Mbps, y controlando la conexión lógica entre el sistema y las funciones internas de cada dispositivo. El controlador USB asimismo maneja el consumo en el bus a través de las funciones Suspend/Continuar, por medio de las cuales controla los modos Reposo/Activo de los dispositivos. Esta arquitectura permite el diseño de dispositivos extremadamente simples y de bajo coste.

USB divide el tiempo en espacios de 1 ms denominados Tramas, durante las cuales se llevan a cabo las comunicaciones a través de Transacciones, las cuales se integran a su vez de Paquetes. Las Transacciones se componen de 3 fases: Token, Dato y Validación (Handshake): [24]

- ✓ La fase de Token se compone de un paquete de token enviado por el Controlador USB, y siempre está presente en toda transacción. El paquete contiene los campos:
  - PID (identifica el tipo de paquete). Todos los PIDs van protegidos por bits redundantes.
  - Dirección del elemento destino (7 bits de dispositivo + 4 bits de elemento interno al dispositivo)
  - CRC5.
- ✓ La fase de Datos (opcional) se compone de los paquetes de datos que se transfieren entre el Controlador USB y el dispositivo. Cada paquete se compone de los campos PID, Datos, y CRC16.
- ✓ La fase de Validación (opcional) se usa para indicar el resultado de la Transacción. Se compone sólo de un campo PID.

Adicionalmente, el Controlador USB indica el principio de cada Trama y la transmisión hacia dispositivos LS mediante tokens especiales. [5]

#### **1.7.1.1.5 Descripción del sistema USB.**

El USB es un bus punto a punto: Dado que el lugar de partida es el host (PC o hub), el destino es un periférico u otro hub. No hay más que un único host (PC) en una arquitectura USB.

Los PC estándar tienen dos tomas USB, lo que implica que, para permitir más de dos periférico simultáneamente, es necesario un hub. Algunos periféricos incluyen un hub integrado, por ejemplo, el teclado USB, al que se le puede conectar un Mouse USB.

Los periféricos comparten la banda de paso del USB. El protocolo se basa en el llamado paso de testigo (token). El ordenador proporciona el testigo al periférico seleccionado y seguidamente, éste le devuelve el testigo en su respuesta.

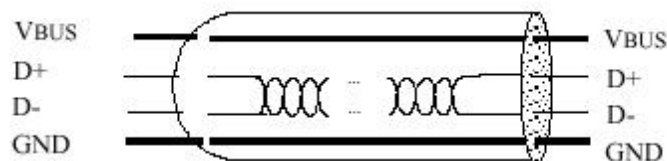
Este bus permite la conexión y la des-conexión en cualquier momento sin necesidad de apagar el equipo.

A continuación se describen los primeros aspectos de este protocolo.

##### **Interfaz física**

##### **Aspecto eléctrico**

**A nivel eléctrico**, el cable USB transfiere la señal y la alimentación **sobre 4 hilos**.



**Fig.3: Diagrama del circuito de un puerto USB.**

- ✓ **A nivel de alimentación:** El cable proporciona la tensión nominal de 5 V. Es necesario definir correctamente el diámetro del hilo con el fin de que no se produzca una caída de tensión demasiado importante en el cable. Una resistencia de terminación instalada en la línea de datos permite detectar el puerto y conocer su configuración (1,5 o 12 Mbits/s).
- ✓ **A nivel de señal:** Se trata de un par trenzado con una impedancia característica de  $90 \Omega$ . La velocidad puede ser tanto de 12 Mbits/s como de 1,5 Mbits/s. La sensibilidad del receptor puede ser de, al menos, 200mV y debe poder admitir un buen factor de rechazo de tensión en modo común. El reloj se transmite en el flow de datos, la codificación es de tipo NRZI, existiendo un dispositivo que genera un bit de relleno (bit stuffing) que garantiza que la frecuencia de reloj permanezca constante. Cada paquete va precedido por un campo de sincronismo.
- ✓ **Consumo:** Cada sección puede proporcionar una determinada potencia máxima siendo el PC el encargado de suministrar la energía. Además, el periférico puede estar autoalimentado (self powered).
- ✓ **Control de consumo:** El ordenador gestiona el consumo, teniendo capacidad de poner en reposo (suspend) o en marcha a un periférico USB. En reposo, este reduce su consumo (si puede), quedándose la parte USB funcional. Esta gestión está orientada especialmente a los equipos portátiles.

#### 1.7.1.1.6 Terminología USB.

**Host:** Dispositivo maestro que inicia la comunicación (Generalmente la computadora). Fig. [4]

**Hub:** Dispositivo que contiene uno o más conectores o conexiones internas hacia otros dispositivos usb, el cual habilita la comunicación entre el host y con diversos dispositivos. Cada conector representa un puerto USB.

**Dispositivo compuesto:** Es aquel dispositivo con múltiples interfaces independientes. Cada una tiene una dirección sobre el bus para cada interface puede tener un diferente driver device en el host.

**Puerto USB:** Cada host soporta solo un bus, cada conector en el bus representa un puerto USB por lo tanto sobre el bus puede haber varios conectores, pero solo existe una ruta y solo un dispositivo puede transmitir información a un tiempo.

**Driver:** es un programa que habilita aplicaciones para poderse comunicar con el dispositivo. Cada dispositivo sobre el bus debe tener un driver, algunos periféricos utilizan los drivers que trae Windows.

**Puntos terminales (Endpoints):** Es una localidad específica dentro del dispositivo. El **Endpoint** es un buffer que almacena múltiples bytes, típicamente es un bloque de la memoria de datos o un registro dentro del microcontrolador. Todos los dispositivos deben soportar el punto terminal 0. Este punto terminal es el que recibe todo el control y la peticiones de estado durante la enumeración cuando el dispositivo esta sobre el bus. Fig. [4]

**Tuberías (Pipes):** Es un enlace virtual entre el host (la PC) y el dispositivo USB, este enlace configura los parámetros asociados con el ancho de banda que tipo de transferencia se va a utilizar. Fig. [4]

**(Control, Bulk, Isócrona o Interrupt)** dirección del flujo de datos y el máximo y/o mínimo tamaño de los paquetes/buffers.

Cada enlace está caracterizado por su banda de paso (Token), su tipo de servicio, el número de punto terminal (End Point) y el tamaño de los paquetes.

Estos enlaces se definen y crean durante la inicialización del USB.

Siempre existe un enlace virtual 0 que permite tener acceso a la información de configuración del periférico USB (estado, control e información).

La norma USB define 2 tipos de enlaces virtuales (pipe); stream y message.

**Stream Pipes:** se trata de un flujo sin formato USB definido, esto significa que se puede enviar cualquier tipo de dato. Este tipo de pipe soporta las transferencias **bulk, isócronas, e interrupt**. Además tanto el host como el dispositivo USB pueden controlar.

**Message Pipes:** este tipo de enlace virtual si tiene un formato USB definido y solo puede soportar la transferencia **Control**.

#### 1.7.1.1.7 Tipos de Transferencia.

USB está diseñado para manejar distintos tipos de periféricos con una gran variedad de requerimientos como lo son la frecuencia de transferencia, tiempo de respuesta y corrección de errores. El estándar de cuatro tipos de transferencias las que manejan diferentes necesidades.

De esta manera los dispositivos pueden utilizar los tipos de transferencias que mejor se adecuen para sus propósitos. [24]

- ✓ **Control: Modo utilizado para realizar configuraciones:** existe siempre sobre el Punto terminal 0 (EndPoint 0). Todos los dispositivos USB deben soportar este tipo de transferencia.  
 Los datos de control sirven para configurar el periférico en el momento de conectarse al USB. Algunos drivers específicos pueden utilizar este enlace para transmitir su propia información de control.  
 Este enlace no tiene pérdida de datos, puesto que los dispositivos de detección de recuperación de errores están activos a nivel USB.
- ✓ **Bulk:** Este modo se utiliza para la transmisión de importantes cantidades de información. Como el tipo control, este enlace no tiene pérdida de datos. Este tipo de transferencia es útil cuando la razón de transferencia no es crítica como por ejemplo, el envío de un archivo a imprimir o la recepción de datos desde un escáner.  
 En estas aplicaciones, la transferencia es rápida, pero puede esperar si fuera necesario.  
 Solo los dispositivos de media y alta velocidad utilizan este tipo de transferencia.
- ✓ **Interrupt,** modo utilizado para transmisiones de pequeños paquetes, rápidos, orientados a percepciones humanas (ratón, punteros).  
 Este tipo de transferencia es para dispositivos que deben recibir atención periódicamente y lo utilizan los dispositivos de baja velocidad  
 Este tipo de transmisión garantiza la transferencia de pequeñas cantidades de datos. El tiempo de respuesta no puede ser inferior al valor especificado por la interfaz. El ratón o cualquier otro dispositivo apuntador es una aplicación típica de este modo de transmisión.
- ✓ **Isochronous o Flujo en tiempo real:** modo utilizado para la transmisión de audio o video comprimido. Este tipo de transmisión funciona en tiempo real. Este es el modo de mayor prioridad.  
 La transmisión de la voz es un ejemplo de esta aplicación. Si ésta no se transmite correctamente, pueden llegar a oírse parásitos (glitch) y la aplicación puede detectar ciertos errores de los llamados underruns.

#### 1.7.1.1.8 Enumeración.

Cuando se conecta un dispositivo USB a la PC se produce el **Proceso de Enumeración**, el cual consiste en que el host le pregunta al dispositivo que se presente y le diga cuáles son sus parámetros, tales como:

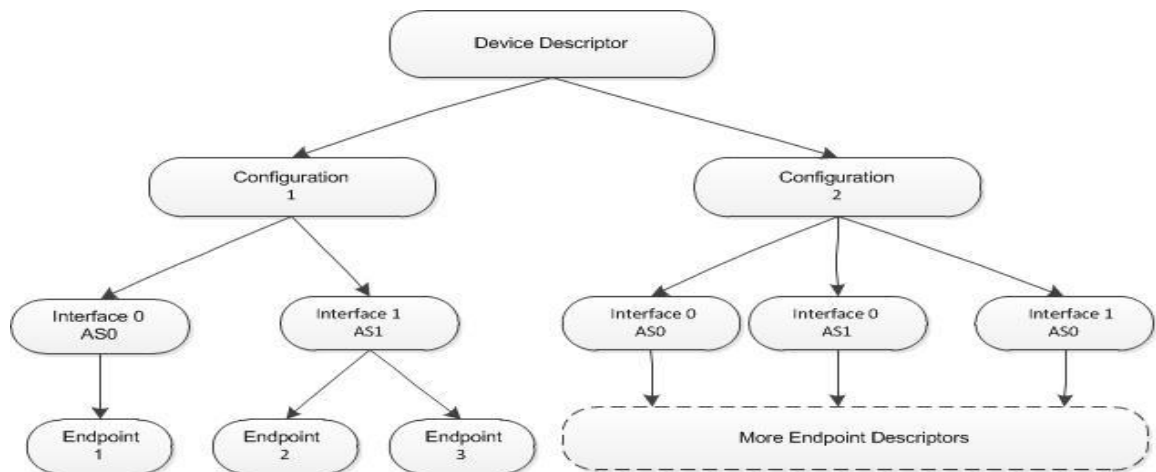
- ✓ Consumo de energía expresada en unidades de Carga
- ✓ Número y tipos de Puntos terminales
- ✓ Clase del producto.
- ✓ Tipo de transferencia
- ✓ Razón de escrutinio, etc.

El proceso de enumeración es inicializado por el host cuando detecta que un nuevo dispositivo que ha sido adjuntado al Bus. El host le asigna una dirección al dispositivo adjuntado al bus y habilita su configuración permitiendo la transferencia de datos sobre el bus.

#### 1.7.1.1.9 Descriptores.

Los descriptores USB son estructuras de datos, o bloques de información con formato, que le permiten al host aprender acerca de un dispositivo. Cada descriptor contiene información acerca del dispositivo como un todo o un elemento del dispositivo como puede verse en la figura [4]. Todos los dispositivos USB deben responder a pedidos para los descriptores USB estándar.

El dispositivo debe guardar información de los descriptores y responder a sus pedidos. [24]

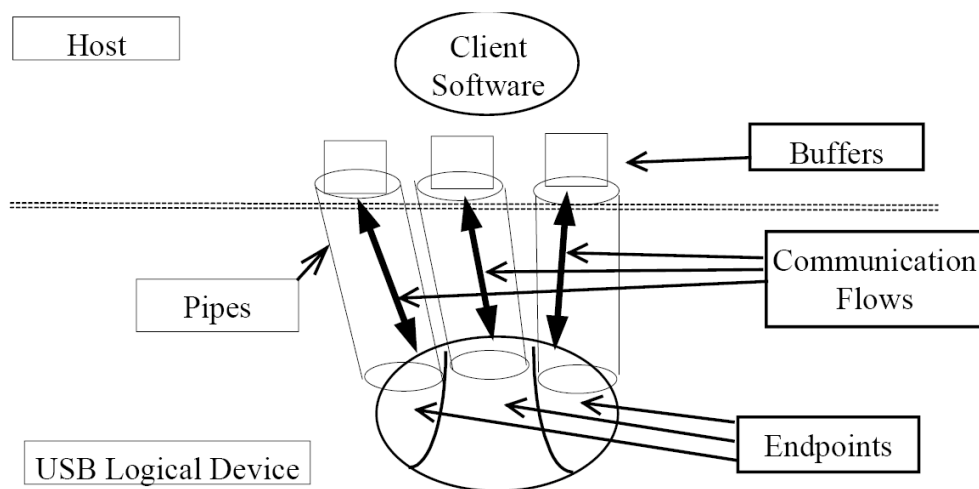


**Fig.4: Jerarquía de Descriptores**

### 1.7.1.1.10 Funcionamiento.

Los dispositivos tienen asociados canales lógicos unidireccionales, llamados pipes, que conectan al host controlador con una entidad lógica en el dispositivo llamada endpoint. Los datos son enviados en paquetes de largo variable. Típicamente estos paquetes son de 64, 128 o más bytes.

Los endpoints y sus respectivos pipes, son numerados del 0 al 15 en cada dirección, por lo cual un dispositivo puede tener hasta 32 endpoints (16 de entrada y 16 de salida). La dirección se considera siempre desde el punto de vista del host controlador. Así un endpoint de salida será un canal que transmite datos desde el host controlador al dispositivo. Un endpoint solo puede tener una única dirección. El endpoint 0 (en ambas direcciones) está reservado para el control del bus. [24]



**Fig.5: Comunicación virtual entre un dispositivo y el host**

Cuando un dispositivo es conectado al bus USB, el host controlador le asigna una dirección única de siete bits (mediante el proceso de enumeración) que es utilizada luego en la comunicación para identificar el dispositivo. [24] Luego, el host controlador consulta continuamente a los dispositivos para ver si tienen algo para enviar, de manera que ningún dispositivo puede enviar datos sin la solicitud previa explícita del host controlador. Para acceder a un endpoint se utiliza una configuración jerárquica de la siguiente manera: un dispositivo (llamado función) conectado al bus tiene un único descriptor de dispositivo, quien a su vez tiene uno (o varios) descriptors de configuración. Estos últimos guardan generalmente el estado del dispositivo (ej. activo, suspendida, ahorro de energía, etc.). Cada descriptor de configuración tiene uno (o más) descriptors de interfaz. Y éstos últimos finalmente

son los que contienen los endpoint, que a su vez pueden ser reutilizados entre varias interfaces (y distintas configuraciones) como muestra la Fig. [4].

#### 1.7.1.1.11 Opciones De Conectividad.

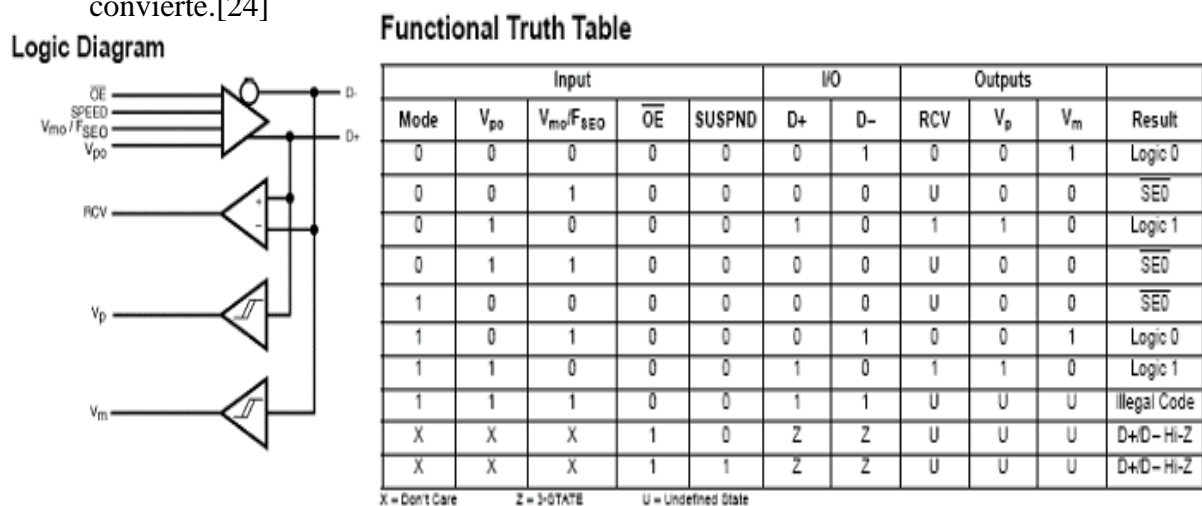
Debido a que los objetivos de este proyecto incluyen el desarrollo de un hardware, se debieron relevar las soluciones de conectividad USB entre el hardware a construir y la PC.

Si bien hay muchos tipos de soluciones realizadas por un varios productores de semiconductores, luego de hacer un relevamiento se pudo realizar una categorización de ellas. Dentro de estas categorías, sólo se dejaron aquellas que no eran kits de desarrollo, y se muestran a continuación:

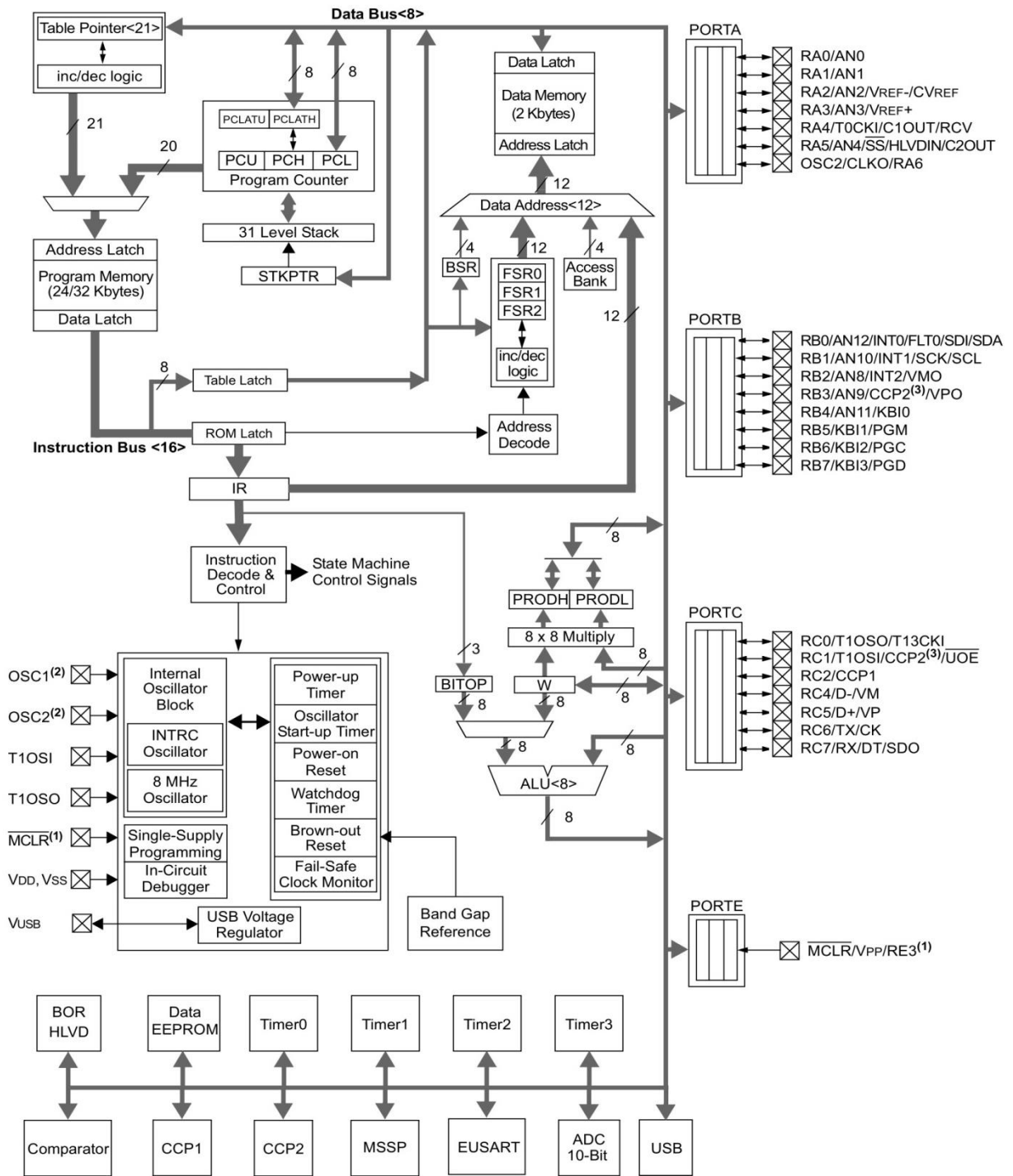
✓ **Transceivers USB:** La principal responsabilidad de los transceptores (transceivers) USB es encapsular la capa física y realizar una interfaz con otros dispositivos programables. Esto sólo incluye una traducción de los voltajes que codifican la transmisión de información en dos señales D+ y D-, a un conjunto de señales para su posterior procesamiento de capas superiores realizadas por otros dispositivos.

En las capas superiores se debe realizar un manejo de transacciones y endpoints, entre otros.

De esta forma estos son dispositivos muy simples, que a los sumo incorporan reguladores de voltaje, y detectores de conexión, lo que los hace muy baratos. Como representantes de esta categoría se seleccionaron el USB1T20 de Fairchild, y el Philips ISP110x. En la Fig. [5] se muestra un diagrama lógico de las señales que traduce y la tabla de verdad de los valores lógicos que convierte.[24]



**Fig. 6: Diagrama lógico y tabla de verdad del Transceiver USB**



**Fig. 7: Diagrama en bloques del PIC18f4550**

Luego el componente controlador UART FIFO, que con ayuda de los buffers de memoria compartida realizan las transferencias de datos con el Host USB. Este componente también sirve para setear la configuración del UART (velocidad, paridad, etc). Finalmente a la derecha de la figura se encuentra el UART que es donde se conecta el hardware vía una interfaz serial.

Otras características que tiene este conversor son:

- Cumple con el estándar USB 2.0 (Full speed).
- Velocidad de 300 a 3M bauds.
- Manejo de handshaking y señales de modem.
- Bit Bang mode: Transforma las señales de control en puerto de E/S de 8 bits.
- Interfaz con EEPROM para caracterizar VID, PID, etc.
- Drivers de puerto COM virtual para Windows.

- ✓ **Controladores de periféricos:** Estos dispositivos incorporan un transceiver USB y la lógica para el manejo del protocolo. En estos es configurable además un número variable de endpoints y tipos de transferencias, así como también descriptores de dispositivo, interfaz, VID, PID, etc. Pueden encontrarse en dos modalidades: externos o embebidos en un microcontrolador. Teniendo en cuenta los requerimientos que se tenían para el proyecto de grado, resultaba más atractivo el uso de estas soluciones

- **Controladores de periféricos externos.**

Estos dispositivos, como ya dijimos, manejan las comunicaciones USB al nivel de transacciones y endpoints y además es visto como otro periférico más para los microcontroladores o microprocesadores con los cuales interactúan. Vale la pena aclarar que estos dispositivos no son autónomos sino que deben interactuar con microcontroladores o microprocesadores para realizar parte de sus tareas. Una de las ventajas más importantes que tiene, es el poco impacto que causa su aplicación en sistemas ya existentes, es decir, si se quiere agregar la funcionalidad a un dispositivo ya existente en un microcontrolador, sólo hay que agregar firmware necesario para el manejo del USB y no migrar todo a otra solución que utilice un controlador de periféricos embebido. En esta sección se tomó como representante el ISP1581 de Philips.

A continuación se presentan las principales características relacionadas con la conectividad USB:

- ❖ Cumplen con el estándar USB 2.0 y soporta velocidades high y full.
- ❖ 7 Endpoints de entrada, 7 de salida.
- ❖ Soporta todos los tipos de transferencias.
- ❖ 8Kb de memoria FIFO integrada.

- ❖ Interfaces con un microcontrolador:
  - Interfaz de bus independiente para la mayoría de los microcontroladores/ microprocesadores (12.5 Mbyte/s).
  - Interfaz DMA de alta velocidad (12.8 Mbytes/s).
  - Interfaz directa con periféricos ATA/ATAPI
- ❖ Conexión al bus USB controlada por software (SoftConnect tm).
- ❖ Data transceiver y regulador de voltaje de 3.3 V integrados
- **Controladores de periféricos embebidos:** En este tipo de soluciones, se incorpora dentro del mismo microcontrolador el hardware necesario para conectarse directamente al Host USB. Brinda las mismas funcionales que el controlador de periféricos externo pero con algunas diferencias. Normalmente es utilizado como un periférico más y utiliza registros dedicados y un tipo de memoria especial, a veces llamada RAM de doble puerto (Dual Port RAM) para intercambiar información con el microcontrolador, además de un poseer una rama completa de interrupciones asociadas a los eventos USB. La comunicación en los casos relevados se maneja a nivel de endpoints y el manejo de transferencias es manejado por rmware con el soporte de hardware de este periférico especial, comúnmente conocido como SIE. Una desventaja que genera el hecho de que se utiliza un recurso embebido en un microcontrolador, es que se genera una dependencia con la arquitectura éste.
 

Esta opción ameritó tener un relevamiento más completo que los anteriores, debido a que fue evaluado como la opción más adecuada para los requerimientos del proyecto y que una decisión en favor de una u otra opción podían tener consecuencias importantes. Los dispositivos relevados fueron:

  - ❖ TSUB3210 de Texas Instruments.
  - ❖ PIC18F4550 de Microchip.
  - ❖ AT90USB1287 de Atmel.

En este caso se presenta el cuadro 2.6 que muestra un comparativo con los aspectos considerados relevantes a la hora de realizar una selección del dispositivo a utilizar para el proyecto.

	<b>TUSB3210</b>	<b>PIC18F4550</b>	<b>AT90USB1287</b>
Arquitectura	CISC (8052)	Harvard RISC 75+8 inst	Harvard RISC 135 inst
Velocidad	12 Mhz	48 Mhz	16 Mhz
Package	TQFP 64	TQFP 44, QFN 44, DIP 40	TQFP 64, QFN 64
Memoria de programa	*6K ROM, 8K RAM (Firmware)	32Kb Flash autoprogramable por software	128Kb Flash autoprogramable por software
Memoria datos	768 bytes	2 Kb	8 Kb (hasta 64 KB externos)
USB 2.0 (full y low speed)	512 Bytes compartida, 3 endp IN, 3 OUT. transferencias interrupt y bulk	1024 Bytes compartida, hasta 32 endp con ping pong buffering, soporta todas las transferencias	832 bytes compartida, 6 endpoints con ping pong buffering, soporta todas las transferencias
Eeprom	No	256 bytes	4 Kbytes
Modo Bajo Consumo	Si	NanoPower, 3 modos Sleep	Si, 6 Modos Sleep
Pines de E/S	Hasta 36	Hasta 35	Hasta 48
Timers	3 de 16 bits	1 de 8 bits 3 de 16 bits	2 de 8 bits 2 de 16 bits
I2C	Master	Master/Slave	TWI* Master/Slave
SPI	No	Master/Slave	Master/Slave
USART	No	Si	Si
Canales PWM	No	Hasta 2 de 10 bits de resolución	Hasta 6 de 2-16 bits de resolución
A/D	No	13 canales 10 bits	8 canales 10 bits
Otros	Bootloader I2C o USB, niveles de prioridad en interrupciones, soporte multiproducto	Soporte bootloader, prioridad de interrupciones programables, multiplicador por hardware, 2 Comparadores analógicos, Streaming Paralel Port. ICSP e ICD Bloqueo de secciones de memoria.	Soporte bootloader, vector de interrupciones con prioridad fija, multiplicación por hardware, modos bajo consumo, USB OTG, Bloqueo de secciones de memoria JTAG.
Documentación	Poca, algunas notas de aplicación.	Mucha, recursos en la web, muchas notas de aplicación, framework USB	Poca, Framework USB, algunas notas de aplicación.
Entornos de desarrollo y compiladores	En general los de 8052, de 3eras partes, algunos gratuitos	MPLAB, 3ras partes, varios compiladores	AVR Studio 4, 3ras partes

**Fig.8: Cuadro comparativo de los microcontroladores**

## 1.7.2 CAPÍTULO 2.-

### 1.7.2.1 Los Microcontroladores.-

#### 1.7.2.1.1 ¿Qué es un microcontrolador?

Un microcontrolador (abreviado  $\mu\text{C}$ , UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

Algunos microcontroladores pueden utilizar palabras de cuatro bits y, funcionan a velocidad de reloj con frecuencias tan bajas como 4 kHz, con un consumo de baja potencia (mW o microvatios). Por lo general, tendrá la capacidad para mantener la funcionalidad a la espera de un evento como pulsar un botón o de otra interrupción, el consumo de energía durante el sueño (reloj de la CPU y los periférico de la mayoría) puede ser sólo nanovatios, lo que hace que muchos de ellos muy adecuados para aplicaciones con batería de larga duración. Otros microcontroladores ( $\mu\text{C}$ ) pueden servir para roles de rendimiento crítico, donde sea necesario actuar más como un procesador digital de señal (DSP), con velocidades de reloj y consumo de energía más altos.

Al ser fabricados, la memoria ROM del  $\mu\text{C}$  no posee datos. Para que pueda controlar algún proceso es necesario generar o crear y luego grabar en la EEPROM o equivalente del microcontrolador algún programa, el cual puede ser escrito en lenguaje ensamblador u otro lenguaje para microcontroladores; sin embargo, para que el programa pueda ser grabado en la memoria del  $\mu\text{C}$ , debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que hace trabajar al microcontrolador cuando éste es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento.

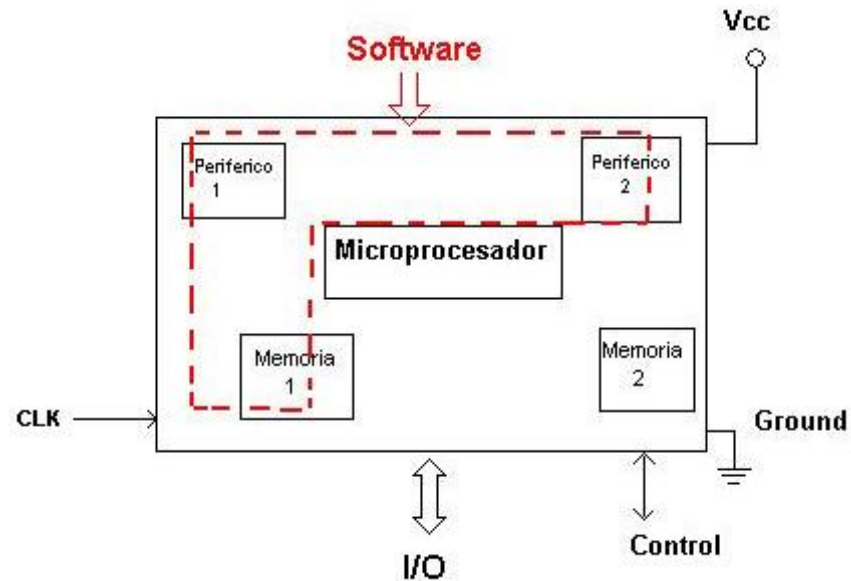


**Fig. 9: Microcontrolador.**

### 1.7.2.1.2 Características.

- ✓ Los microcontroladores son diseñados para reducir el costo económico y el consumo de energía de un sistema en particular. Por eso el tamaño de la unidad central de procesamiento, la cantidad de memoria y los periféricos incluidos dependerán de la aplicación. El control de un electrodoméstico sencillo como una batidora utilizará un procesador muy pequeño (4 u 8 bits) porque sustituirá a un autómatas finitos. En cambio, un reproductor de música y/o vídeo digital (MP3 o MP4) requerirá de un procesador de 32 bits o de 64 bits y de uno o más códecs de señal digital (audio y/o vídeo). El control de un sistema de frenos ABS (Antilock Brake System) se basa normalmente en un microcontrolador de 16 bits, al igual que el sistema de control electrónico del motor en un automóvil.
- ✓ Los microcontroladores representan la inmensa mayoría de los chips de computadoras vendidos, sobre un 50% son controladores "simples" y el restante corresponde a DSPs más especializados. Mientras se pueden tener uno o dos microprocesadores de propósito general en casa, distribuidos seguramente entre los electrodomésticos de su hogar una o dos docenas de microcontroladores. Pueden encontrarse en casi cualquier dispositivo electrónico como automóviles, lavadoras, hornos microondas, teléfonos, etc.
- ✓ Un microcontrolador ( $\mu$ C) difiere de una unidad central de procesamiento normal, debido a que es más fácil convertirla en una computadora en funcionamiento, con un mínimo de circuitos integrados externos de apoyo. La idea es que el circuito integrado se coloque en el dispositivo, enganchado a la fuente de energía y de información que necesite, y eso es todo.
- ✓ Un microprocesador tradicional no le permitirá hacer esto, ya que espera que todas estas tareas sean manejadas por otros chips. Hay que agregarle los módulos de entrada y salida (puertos) y la memoria para almacenamiento de información.
- ✓ Un  $\mu$ C típico tendrá un generador de reloj integrado y una pequeña cantidad de memoria de acceso aleatorio y/o ROM/EPROM/EEPROM/flash, con lo que para hacerlo funcionar todo lo que se necesita son unos pocos programas de control y un cristal de sincronización. Los microcontroladores disponen generalmente también de una gran variedad de dispositivos de entrada/salida, como convertidor analógico digital, temporizadores, UARTs y buses de interfaz serie especializados, como I2C y CAN. Frecuentemente, estos dispositivos integrados pueden ser controlados por instrucciones de procesadores especializados.
- ✓ Los modernos microcontroladores frecuentemente incluyen un lenguaje de programación integrado, como el lenguaje de programación BASIC que se utiliza bastante con este propósito.

- ✓ Los microcontroladores negocian la velocidad y la flexibilidad para facilitar su uso. Debido a que se utiliza bastante sitio en el chip para incluir funcionalidad, como los dispositivos de entrada/salida o la memoria que incluye el microcontrolador, se ha de prescindir de cualquier otra circuitería.



**Fig. 10: Microcontrolador dentro de un encapsulado de circuito integrado**

En la Fig.[9], vemos al microcontrolador dentro de un encapsulado de circuito integrado, con su procesador (CPU), buses, memoria, periféricos y puertos de entrada/salida. Fuera del encapsulado se ubican otros circuitos para completar periféricos internos y dispositivos que pueden conectarse a los pines de entrada/salida. También se conectarán a los pines del encapsulado la alimentación, masa, circuito de completamiento del oscilador y otros circuitos necesarios para que el microcontrolador pueda trabajar.

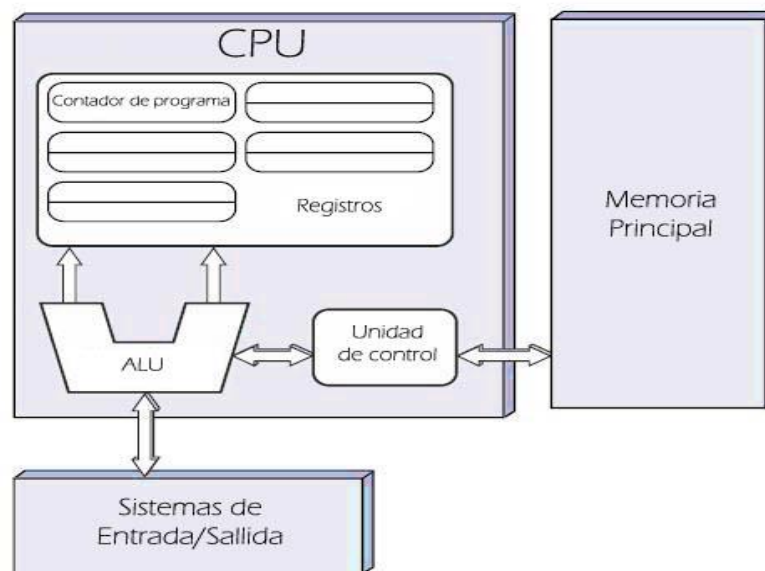
### 1.7.2.1.3 Arquitectura de computadoras.

Básicamente existen dos arquitecturas de computadoras, y por supuesto, están presentes en el mundo de los microcontroladores: Von Neumann y Harvard. Ambas se diferencian en la forma de conexión de la memoria al procesador y en los buses que cada una necesita.

- ✓ **Arquitectura Von Neumann:** La arquitectura Von Neumann utiliza el mismo dispositivo de almacenamiento tanto para las instrucciones como para los datos, siendo la que se utiliza en un ordenador personal porque permite ahorrar una

buena cantidad de líneas de E/S, que son bastante costosas, sobre todo para aquellos sistemas donde el procesador se monta en algún tipo de zócalo alojado en una placa madre. También esta organización les ahorra a los diseñadores de placas madre una buena cantidad de problemas y reduce el costo de este tipo de sistemas.

En un ordenador personal, cuando se carga un programa en memoria, a éste se le asigna un espacio de direcciones de la memoria que se divide en segmentos, de los cuales típicamente tendremos los siguientes: código (programa), datos y pila. Es por ello que podemos hablar de la memoria como un todo, aunque existan distintos dispositivos físicos en el sistema (disco duro, memoria RAM, memoria flash, unidad de disco óptico). [w25]



**Fig. 11: Diagrama de la arquitectura Von Neumann.**

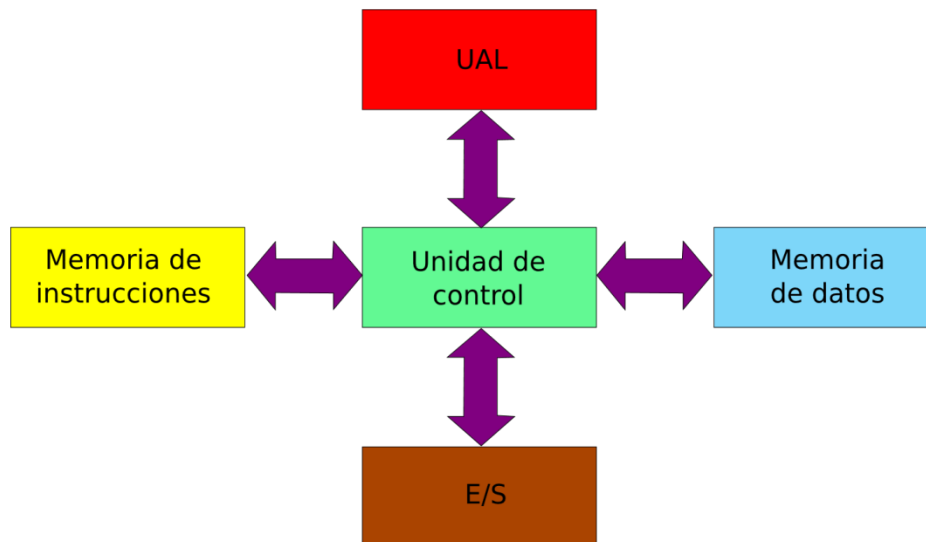
En el caso de los microcontroladores, existen dos tipos de memoria bien definidas: memoria de datos (típicamente algún tipo de SRAM) y memoria de programas (ROM, PROM, EEPROM, flash u de otro tipo no volátil). En este caso la organización es distinta a las del ordenador personal, porque hay circuitos distintos para cada memoria y normalmente no se utilizan los registros de segmentos, sino que la memoria está segregada y el acceso a cada tipo de memoria depende de las instrucciones del procesador.

A pesar de que en los sistemas integrados con arquitectura Von Neumann la memoria esté segregada, y existan diferencias con respecto a la definición tradicional de esta arquitectura; los buses para acceder a ambos tipos de memoria

son los mismos, del procesador solamente salen el bus de datos, el de direcciones, y el de control. Como conclusión, la arquitectura no ha sido alterada, porque la forma en que se conecta la memoria al procesador sigue el mismo principio definido en la arquitectura básica.

Algunas familias de microcontroladores como la Intel 8051 y la Z80 implementan este tipo de arquitectura, fundamentalmente porque era la utilizada cuando aparecieron los primeros microcontroladores.

- ✓ **Arquitectura Harvard:** La arquitectura Harvard es otra variante, y por excelencia la utilizada en supercomputadoras, en los microcontroladores, y sistemas integrados en general. En este caso, además de la memoria, el procesador tiene los buses segregados, de modo que cada tipo de memoria tiene un bus de datos, uno de direcciones y uno de control.



**Fig. 12: Diagrama de Harvard**

La ventaja fundamental de esta arquitectura es que permite adecuar el tamaño de los buses a las características de cada tipo de memoria; además, el procesador puede acceder a cada una de ellas de forma simultánea, lo que se traduce en un aumento significativo de la velocidad de procesamiento. Típicamente los sistemas con esta arquitectura pueden ser dos veces más rápidos que sistemas similares con arquitectura Von Neumann.

La desventaja está en que consume muchas líneas de E/S del procesador; por lo que en sistemas donde el procesador está ubicado en su propio encapsulado, solo se utiliza en supercomputadoras. Sin embargo, en los microcontroladores y otros

sistemas integrados, donde usualmente la memoria de datos y programas comparten el mismo encapsulado que el procesador, este inconveniente deja de ser un problema serio y es por ello que encontramos la arquitectura Harvard en la mayoría de los microcontroladores.

Por eso es importante recordar que un microcontrolador se puede configurar de diferentes maneras, siempre y cuando se respete el tamaño de memoria que este requiera para su correcto funcionamiento. [w27]

#### **1.7.2.1.4 Historia de lo microcontroladores.**

En el año de 1969 un equipo de ingenieros japoneses de la compañía BUSICOM llego al país de los Estados Unidos debido a una petición para construir algunos circuitos integrados para calculadoras. Estos nuevos circuitos integrados debían construirse en base a sus proyectos desarrollados anteriormente. La petición fue hecha por la compañía INTEL, y Marcian Hoff era el responsable del proyecto. Él era la persona con más experiencia trabajando con computadoras así que sugiere crear un circuito integrado en donde la función era determinada por un programa almacenado dentro del mismo. Esto significo que la configuración fuera más sencilla, pero esto requería de una mayor memoria que el proyecto propuesto por los ingenieros japoneses. Después de un profundo análisis del proyecto, los ingenieros japoneses trataron de encontrar una solución más simple al problema pero la idea de Marcían gano, es así cuando las primeras ideas de un microprocesador son llevadas a cabo para su creación.

Federico Faggin fue la mayor ayuda que tubo INTEL al transformar y mejorar como construir el nuevo proyecto. En solo 9 meses terminaron y construyeron el primer concepto del producto. Inmediatamente compran la licencia de la compañía BUSICOM la cual no tenía idea del gran tesoro que poseían. Intel obtiene los derechos del circuito integrado 1971. Durante ese mismo año, aparece en el Mercado el primer microprocesador llamado 4004 con la capacidad de ejecutar 6000 operaciones por Segundo. No pasa mucho tiempo cuando una compañía Americana solicita a INTEL y Texas Instruments la creación de un microprocesador de 8 bits. En abril de 1972 el primer microprocesador de 8 bits aparece en el Mercado bajo el nombre de 8008. Este tenía 45 instrucciones de operación, capacidad de direccionar hasta 16kb de memoria y trabajar con una velocidad de 300,000 operaciones por segundo. Este fue el precursor de todos los microprocesadores de hoy. Intel continua con sus desarrollos y en abril de 1974 pone en el Mercado el microprocesador bajo el nombre 8080 con capacidad de direccionar 64kb de memoria, con 75 instrucciones y un precio de inicio de \$360 dólares.

En otra compañía Americana “Motorola” rápidamente analizaron que pasaba, y pusieron en el Mercado un microprocesador de 8 bits, llamado 6800. Motorola fue la primera compañía en construir otros periféricos como el 6820 y el 6850.

En estas fechas muchas compañías reconocieron el gran impacto del microprocesador e iniciaron sus propios desarrollos.

En 1975 ocurrió un evento crítico para la historia de los microprocesadores, la MOS Technology anunciaba que los microprocesadores 6501 y 6502 se vendían a solo \$25 dólares cada uno, mientras el precio de los 8080 y 6800 eran de \$179 dólares. Como una respuesta a esto Intel y Motorola bajaron sus precios al primer día del anuncio del competidor a \$69.95 por microprocesador.

MOS Technology detuvo su producción del 6501 pero mantuvo su producción del 6502, el cual fue un microprocesador de 8 bits con 56 instrucciones y una capacidad de direccionar 64 kb de memoria. Debido al bajo costo, el 6502 se volvió muy popular, de tal forma que instalado en computadoras, tales como: KIM-1, Apple I, Apple II, Atari, Comodore, Acorn, Oric, Galeb, Orao, Ultra y muchos otros. Pronto aparecieron muchos otros fabricantes de 6502 (Rockwell, Sznertek, GTE, NCR, Ricoh, y Comodore toman como base a MOS Technology) la cual estaba en sus mejores ventas de hasta 15 millones de procesadores por año.

Algunas compañías no mejoraban y sin embargo Federico Faggien deja Intel y comienza su propia compañía “Zilog Inc”. En 1976 Zilog anuncia su nuevo microprocesador con el nombre de Z80. Durante el desarrollo de este microprocesador, en donde Faffin toma una importante decisión, dado que sabía bien que muchos de los programas estaban desarrollados para trabajar en el 8080. Mucha gente se mantenía fiel al a pesar de precio tan caro en el Mercado, pero con excelentes características. Decide que el nuevo procesador sea capaz de funcionar con todos los programas que habían sido creados para el 8080. Con todas estas características, muchos nuevos usuarios agregaron el Z80 sus proyectos haciendo uno de los microprocesadores más poderosos en ese tiempo.

El Z80 tenía 176 instrucciones, un gran número de registros, una opción para refrescar la memoria dinámica RAM, solo necesitaba una sola fuente de alimentación, gran velocidad de trabajo, era capaz de direccionar 64k de memoria, entre otras cosas más. Gran parte de los programadores cambiaron de 8080 a Z80. Este microprocesador se volvió el más utilizado en su tiempo. Junto con Zilog, otras nuevas compañías aparecieron como Mostek, NEC, SHARP y SGS. Z80 fue el corazón de muchas computadoras como Spectrum, Partner, TRS703 y Z-3.

En 1976, Intel mejora su versión de microprocesador de 8 bits llamado 8085 pero sin embargo, Z80 fue rápidamente mejorado y la compañía Intel perdió rápidamente la batalla. Siguieron apareciendo más microprocesadores en el mercado (6809, 2650, SC/MP) pero todo ya estaba decidido. No había mejoras importantes que convirtieran

el producto como algo nuevo, así que el 6502 y Z80 eran los representantes de los microprocesadores de 8 bits de esos tiempos. el producto como algo nuevo, así que el 6502 y Z80 eran los representantes de los microprocesadores de 8 bits de esos tiempos.

#### **1.7.2.1.5 Registros**

Son un espacio de memoria muy reducido pero necesario para cualquier microprocesador, de aquí se toman los datos para varias operaciones que debe realizar el resto de los circuitos del procesador. Los registros sirven para almacenar los resultados de la ejecución de instrucciones, cargar datos desde la memoria externa o almacenarlos en ella.

Aunque la importancia de los registros parezca trivial, no lo es en absoluto. De hecho una parte de los registros, la destinada a los datos, es la que determina uno de los parámetros más importantes de cualquier microprocesador. Cuando escuchamos que un procesador es de 4, 8, 16, 32 o 64 bits, nos estamos refiriendo a procesadores que realizan sus operaciones con registros de datos de ese tamaño, y por supuesto, esto determina muchas de las potencialidades de estas máquinas.

Mientras mayor sea el número de bits de los registros de datos del procesador, mayores serán sus prestaciones, en cuanto a poder de cómputo y velocidad de ejecución, ya que este parámetro determina la potencia que se puede incorporar al resto de los componentes del sistema, por ejemplo, no tiene sentido tener una ALU de 16 bits en un procesador de 8 bits.

Por otro lado un procesador de 16 bits, puede que haga una suma de 16 bits en un solo ciclo de máquina, mientras que uno de 8 bits deberá ejecutar varias instrucciones antes de tener el resultado, aun cuando ambos procesadores tengan la misma velocidad de ejecución para sus instrucciones. El procesador de 16 bits será más rápido porque puede hacer el mismo tipo de tareas que uno de 8 bits, en menos tiempo. [w25]

#### **1.7.2.1.6 Unidad de control.**

Esta unidad es de las más importantes en el procesador, en ella recae la lógica necesaria para la decodificación y ejecución de las instrucciones, el control de los registros, la ALU, los buses y cuanta cosa más se quiera meter en el procesador.

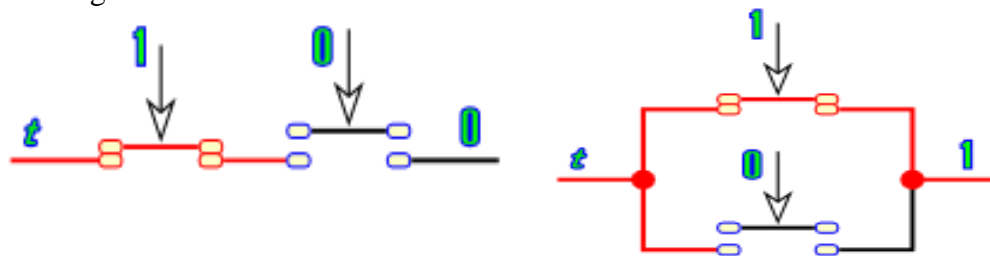
La unidad de control es uno de los elementos fundamentales que determinan las prestaciones del procesador, ya que su tipo y estructura, determina parámetros tales como el tipo de conjunto de instrucciones, velocidad de ejecución, tiempo del ciclo de

máquina, tipo de buses que puede tener el sistema, manejo de interrupciones y un buen número de cosas más que en cualquier procesador van a parar a este bloque.

Por supuesto, las unidades de control, son el elemento más complejo de un procesador y normalmente están divididas en unidades más pequeñas trabajando de conjunto. La unidad de control agrupa componentes tales como la unidad de decodificación, unidad de ejecución, controladores de memoria cache, controladores de buses, controlador de interrupciones, pipelines, entre otros elementos, dependiendo siempre del tipo de procesador. [w27]

### 1.7.2.1.7 Unidad aritmética - lógica.

Como los procesadores son circuitos que hacen básicamente operaciones lógicas y matemáticas, se le dedica a este proceso una unidad completa, con cierta independencia. Aquí es donde se realizan las sumas, restas, y operaciones lógicas típicas del álgebra de Boole.



**Fig.13: Interruptores lógicos**

Actualmente este tipo de unidades ha evolucionado mucho y los procesadores más modernos tienen varias ALU, especializadas en la realización de operaciones complejas como las operaciones en coma flotante. De hecho en muchos casos le han cambiado su nombre por el de “coprocesador matemático”, aunque este es un término que surgió para dar nombre a un tipo especial de procesador que se conecta directamente al procesador más tradicional.

Su impacto en las prestaciones del procesador es también importante porque, dependiendo de su potencia, tareas más o menos complejas, pueden hacerse en tiempos muy cortos, como por ejemplo, los cálculos en coma flotante.

Esto no siempre funciona, puesto que se necesita de un termoresistor para compensar su registro de memoria, por lo que este termoresistor se puede sustituir por un transductor de presencia, para que su memoria de registro no sea complaciente para su velocidad de registro.

### 1.7.2.1.8 Buses.

Son el medio de comunicación que utilizan los diferentes componentes del procesador para intercambiar información entre sí, eventualmente los buses o una parte de ellos estarán reflejados en los pines del encapsulado del procesador.

En el caso de los microcontroladores, no es común que los buses estén reflejados en el encapsulado del circuito, ya que estos se destinan básicamente a las E/S de propósito general y periféricos del sistema. [w27]

Existen tres tipos de buses:

- ✓ **Dirección:** Se utiliza para seleccionar al dispositivo con el cual se quiere trabajar o en el caso de las memorias, seleccionar el dato que se desea leer o escribir.
- ✓ **Datos.**
- ✓ **Control:** Se utiliza para gestionar los distintos procesos de escritura lectura y controlar la operación de los dispositivos del sistema.

El bus funciona ordenando la información que es transmitida desde distintas unidades y periféricos a la unidad central, haciendo las veces de semáforo o regulador de prioridades y operaciones a ejecutar.

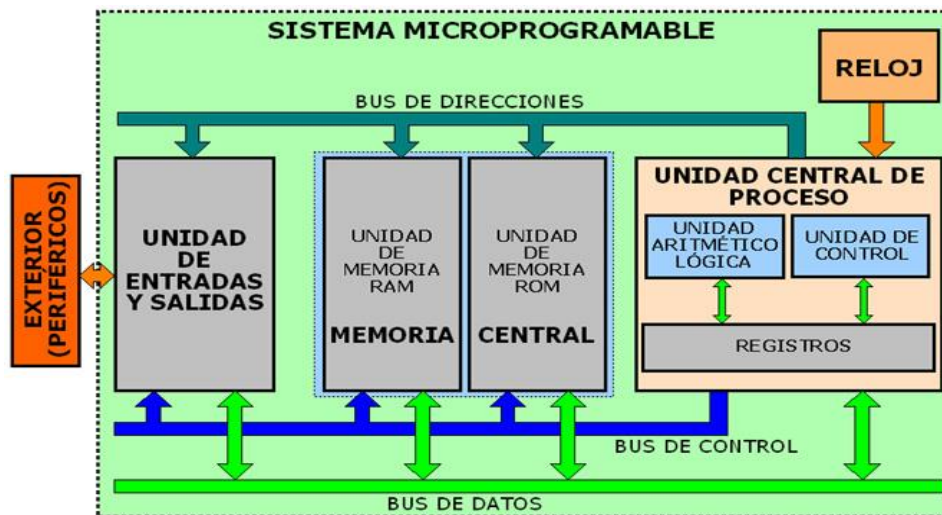


Fig. 14: Diagrama de sistema de buses

### 1.7.2.1.9 Conjunto de instrucciones.

Aunque no aparezca en el esquema, no podíamos dejar al conjunto o repertorio de instrucciones fuera de esta fiesta, porque este elemento determina lo que puede hacer el procesador.

Define las operaciones básicas que puede realizar el procesador, que conjugadas y organizadas forman lo que conocemos como software. El conjunto de instrucciones vienen siendo como las letras del alfabeto, el elemento básico del lenguaje, que organizadas adecuadamente permiten escribir palabras, oraciones y cuanto programa se le ocurra.

Existen dos tipos básicos de repertorios de instrucciones, que determinan la arquitectura del procesador: CISC y RISC.

CISC, del inglés Complex instruction set computing, Computadora de Conjunto de Instrucciones Complejo. Los microprocesadores CISC tienen un conjunto de instrucciones que se caracteriza por ser muy amplio y que permiten realizar operaciones complejas entre operandos situados en la memoria o en los registros internos. Este tipo de repertorio dificulta el paralelismo entre instrucciones, por lo que en la actualidad, la mayoría de los sistemas CISC de alto rendimiento, convierten las instrucciones complejas en varias instrucciones simples del tipo RISC, llamadas generalmente microinstrucciones. [w27]

Dentro de los microcontroladores CISC podemos encontrar a la popular familia Intel 8051 y la Z80, aunque actualmente existen versiones CISC-RISC de estos microcontroladores, que pretenden aprovechar las ventajas de los procesadores RISC a la vez que se mantiene la compatibilidad hacia atrás con las instrucciones de tipo CISC.

RISC, del inglés Reduced Instruction Set Computer, Computadora con Conjunto de Instrucciones Reducido. Se centra en la obtención de procesadores con las siguientes características fundamentales:

Instrucciones de tamaño fijo.

Pocas instrucciones.

Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.

Número relativamente elevado de registros de propósito general.

Una de las características más destacables de este tipo de procesadores es que posibilitan el paralelismo en la ejecución, y reducen los accesos a memoria. Es por eso que los procesadores más modernos, tradicionalmente basados en arquitecturas CISC implementan mecanismos de traducción de instrucciones CISC a RISC, para aprovechar las ventajas de este tipo de procesadores.

Los procesadores de los microcontroladores PIC son de tipo RISC.

### 1.7.2.1.10 Memoria.

Anteriormente habíamos visto que la memoria en los microcontroladores debe estar ubicada dentro del mismo encapsulado, esto es así la mayoría de las veces, porque la idea fundamental es mantener el grueso de los circuitos del sistema dentro de un solo integrado.

En los microcontroladores la memoria no es abundante, aquí no encontrará Gigabytes de memoria como en las computadoras personales. Típicamente la memoria de programas no excederá de 16 K-localizaciones de memoria no volátil (flash o eeprom) para contener los programas. [w27]

La memoria RAM está destinada al almacenamiento de información temporal que será utilizada por el procesador para realizar cálculos u otro tipo de operaciones lógicas. En el espacio de direcciones de memoria RAM se ubican además los registros de trabajo del procesador y los de configuración y trabajo de los distintos periféricos del microcontrolador. Es por ello que en la mayoría de los casos, aunque se tenga un espacio de direcciones de un tamaño determinado, la cantidad de memoria RAM de que dispone el programador para almacenar sus datos es menor que la que puede direccionar el procesador.

El tipo de memoria utilizada en las memorias RAM de los microcontroladores es SRAM, lo que evita tener que implementar sistemas de refrescamiento como en el caso de las computadoras personales, que utilizan gran cantidad de memoria, típicamente alguna tecnología DRAM. A pesar de que la memoria SRAM es más costosa que la DRAM, es el tipo adecuado para los microcontroladores porque éstos poseen pequeñas cantidades de memoria RAM. [w27]

En el caso de la memoria de programas se utilizan diferentes tecnologías, y el uso de una u otra depende de las características de la aplicación a desarrollar, a continuación se describen las cinco tecnologías existentes, que mayor utilización tienen o han tenido:

- ✓ **Máscara ROM:** En este caso no se “graba” el programa en memoria sino que el microcontrolador se fabrica con el programa, es un proceso similar al de producción de los CD comercial mediante masterización. El costo inicial de producir un circuito de este tipo es alto, porque el diseño y producción de la máscara es un proceso costoso, sin embargo, cuando se necesitan varios miles o incluso cientos de miles de microcontroladores para una aplicación determinada, como por ejemplo, algún electrodoméstico, el costo inicial de producción de la máscara y el de fabricación del circuito se distribuye entre todos los circuitos de

la serie y, el costo final de ésta, es bastante menor que el de sus semejantes con otro tipo de memoria.

- ✓ **Memoria PROM:** (Programmable Read-Only Memory) también conocida como OTP (One Time Programmable). Este tipo de memoria, también es conocida como PROM o simplemente ROM. Los microcontroladores con memoria OTP se pueden programar una sola vez, con algún tipo de programador. Se utilizan en sistemas donde el programa no requiera futuras actualizaciones y para series relativamente pequeñas, donde la variante de máscara sea muy costosa, también para sistemas que requieren serialización de datos, almacenados como constantes en la memoria de programas.
- ✓ **Memoria EPROM:** (Erasable Programmable Read Only Memory). Los microcontroladores con este tipo de memoria son muy fáciles de identificar porque su encapsulado es de cerámica y llevan encima una ventanita de vidrio desde la cual puede verse la oblea de silicio del microcontrolador ( $\mu\text{C}$ ). Se fabrican así porque la memoria EPROM es reprogramable, pero antes debe borrarse, y para ello hay que exponerla a una fuente de luz ultravioleta, el proceso de grabación es similar al empleado para las memorias OTP. Al aparecer tecnologías menos costosas y más flexibles, como las memorias EEPROM y FLASH, este tipo de memoria han caído en desuso, se utilizaban en sistemas que requieren actualizaciones del programa y para los procesos de desarrollo y puesta a punto. EEPROM (Electrical Erasable Programmable Read Only Memory). Fueron el sustituto natural de las memorias EPROM, la diferencia fundamental es que pueden ser borradas eléctricamente, por lo que la ventanilla de cristal de cuarzo y los encapsulados cerámicos no son necesarios. Al disminuir los costos de los encapsulados, los microcontroladores con este tipo de memoria se hicieron más baratos y cómodos para trabajar que sus equivalentes con memoria EPROM. Otra característica destacable de este tipo de  $\mu\text{C}$  es que fue en ellos donde comenzaron a utilizarse los sistemas de programación en el sistema que evitan tener que sacar el microcontrolador de la tarjeta que lo aloja para hacer actualizaciones al programa.
- ✓ **Memoria flash:** En el campo de las memorias reprogramables para microcontroladores, son el último avance tecnológico en uso a gran escala, y han sustituido a los microcontroladores con memoria EEPROM. A las ventajas de las memorias flash se le adicionan su gran densidad respecto a sus predecesoras lo que permite incrementar la cantidad de memoria de programas a un costo muy bajo. Pueden además ser programadas con las mismas tensiones de alimentación del microcontrolador, el acceso en lectura y la velocidad de programación es superior, disminución de los costos de producción, entre otras.

Lo más habitual es encontrar que la memoria de programas y datos está ubicada toda dentro del microcontrolador, de hecho, actualmente son pocos los microcontroladores que permiten conectar memoria de programas en el exterior del encapsulado. Las razones para estas “limitaciones” están dadas porque el objetivo fundamental es obtener la mayor integración posible y conectar memorias externas consume líneas de E/S que son uno de los recursos más preciados de los microcontroladores.

A pesar de lo anterior existen familias como la Intel 8051 cuyos microcontroladores tienen la capacidad de ser expandidos en una variada gama de configuraciones para el uso de memoria de programas externa. En el caso de los PIC, estas posibilidades están limitadas sólo a algunos microcontroladores de la gama alta, la Figura 5 muestra algunas de las configuraciones para memoria de programa que podemos encontrar en los microcontroladores. La configuración (a) es la típica y podemos encontrarla casi en el 100% de los microcontroladores. La configuración (b) es poco frecuente y generalmente se logra configurando al microcontrolador para sacrificar la memoria de programas interna, sin embargo el Intel 8031 es un microcontrolador sin memoria de programas interna. La configuración (c) es la que se encuentra habitualmente en los microcontroladores que tienen posibilidades de expandir su memoria de programas como algunos PIC de gama alta.

Cuando se requiere aumentar la cantidad de memoria de datos, lo más frecuente es colocar dispositivos de memoria externa en forma de periféricos, de esta forma se pueden utilizar memorias RAM, FLASH o incluso discos duros como los de los ordenadores personales, mientras que para los cálculos y demás operaciones que requieran almacenamiento temporal de datos se utiliza la memoria RAM interna del microcontrolador. Esta forma de expandir la memoria de datos está determinada, en la mayoría de los casos, por el tipo de repertorio de instrucciones del procesador y porque permite un elevado número de configuraciones distintas, además del consiguiente ahorro de líneas de E/S que se logra con el uso de memorias con buses de comunicación serie.

#### **1.7.2.1.11 Periféricos.**

##### **❖ Temporizadores y contadores.**

Son circuitos sincrónicos para el conteo de los pulsos que llegan a su poder para conseguir la entrada de reloj. Si la fuente de un gran conteo es el oscilador interno del microcontrolador es común que no tengan un pin asociado, y en este caso trabajan como temporizadores. Por otra parte, cuando

la fuente de conteo es externa, entonces tienen asociado un pin configurado como entrada, este es el modo contador.

Los temporizadores son uno de los periféricos más habituales en los microcontroladores y se utilizan para muchas tareas, como por ejemplo, la medición de frecuencia, implementación de relojes, para el trabajo de conjunto con otros periféricos que requieren una base estable de tiempo entre otras funcionalidades. Es frecuente que un microcontrolador típico incorpore más de un temporizador/contador e incluso algunos tienen arreglos de contadores. Como veremos más adelante este periférico es un elemento casi imprescindible y es habitual que tengan asociada alguna interrupción. Los tamaños típicos de los registros de conteo son 8 y 16 bits, pudiendo encontrar dispositivos que solo tienen temporizadores de un tamaño o con más frecuencia con ambos tipos de registro de conteo.

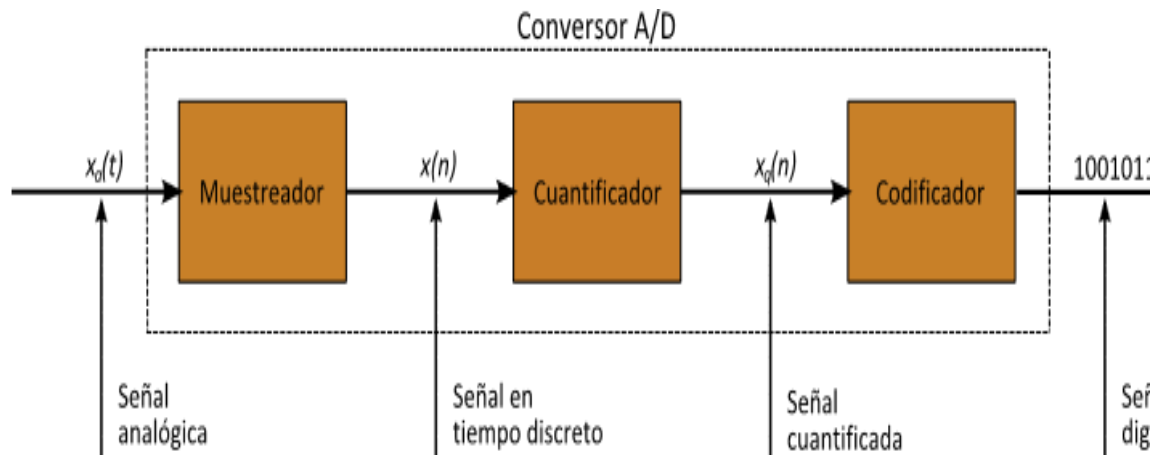
- ❖ **Entradas y salidas de propósito general:** También conocidos como puertos de E/S, generalmente agrupadas en puertos de 8 bits de longitud, permiten leer datos del exterior o escribir en ellos desde el interior del microcontrolador, el destino habitual es el trabajo con dispositivos simples como relés, LED, o cualquier otra cosa que se le ocurra al programador.

Algunos puertos de E/S tienen características especiales que le permiten manejar salidas con determinados requerimientos de corriente, o incorporan mecanismos especiales de interrupción para el procesador.

Típicamente cualquier pin de E/S puede ser considerada E/S de propósito general, pero como los microcontroladores no pueden tener infinitos pines, ni siquiera todos los pines que queramos, las E/S de propósito general comparten los pines con otros periféricos. Para usar un pin con cualquiera de las características a él asignadas debemos configurarlo mediante los registros destinados a ellos.

- ❖ **Convertor analógico\digital:** Como es muy frecuente el trabajo con señales analógicas, éstas deben ser convertidas a digital y por ello muchos microcontroladores incorporan un convertor analógico-digital, el cual se utiliza para tomar datos de varias entradas diferentes que se seleccionan mediante un multiplexor.

Las resoluciones más frecuentes son 8 y 10 bits, que son suficientes para aplicaciones sencillas. Para aplicaciones en control e instrumentación están disponibles resoluciones de 12bit, 16bit y 24bit.6 También es posible conectar un convertidor externo, en caso de necesidad. [w27]



**Fig. 15: Proceso de conversión Analógico\Digital**

#### 1.7.2.1.12 Interrupciones.

Las interrupciones son esencialmente llamadas a subrutina generadas por los dispositivos físicos, al contrario de las subrutinas normales de un programa en ejecución. Como el salto de subrutina no es parte del hilo o secuencia de ejecución programada, el controlador guarda el estado del procesador en la pila de memoria y entra a ejecutar un código especial llamado "manejador de interrupciones" que atiende al periférico específico que generó la interrupción. Al terminar la rutina, una instrucción especial le indica al procesador el fin de la atención de la interrupción. En ese momento el controlador restablece el estado anterior, y el programa que se estaba ejecutando antes de la interrupción sigue como si nada hubiese pasado. Las rutinas de atención de interrupciones deben ser lo más breves posibles para que el rendimiento del sistema sea satisfactorio, porque normalmente cuando una interrupción es atendida, todas las demás interrupciones están en espera.

Imagine que está esperando la visita de un amigo, al que llamaremos Juan. Usted y Juan han acordado que cuando él llegue a su casa esperará pacientemente a que le abra la puerta. Juan no debe tocar a la puerta porque alguien en la casa duerme y no quiere que le despierten.

Ahora usted ha decidido leer un libro mientras espera a que Juan llegue a la casa, y para comprobar si ha llegado, cada cierto tiempo detiene la lectura, marca la página donde se quedó, se levanta y va hasta la puerta, abre y comprueba si Juan ha llegado, si éste todavía no está en la puerta, esperará unos minutos, cerrará la puerta y regresará a su lectura durante algún tiempo.

Como verá este es un método poco eficiente para esperar a Juan porque requiere que deje la lectura cada cierto tiempo y vaya hasta la puerta a comprobar si él ha llegado, además debe esperar un rato si todavía no llega. Y por si fuera poco, imagine que Juan no llega nunca porque se le presentó un problema, tuvo que cancelar la cita y no pudo avisarle a tiempo, o peor, que Juan ha llegado a la puerta un instante después que usted la cerraba. Juan, respetando lo acordado, espera un tiempo, pero se cansa de esperar a que le abran y decide marcharse porque cree que ya usted no está en la casa o no puede atenderlo. A este método de atender la llegada de Juan lo llamaremos encuesta.

Veamos ahora otro método. En esta ocasión simplemente se recuesta en el sofá de la sala y comienza a leer su libro, cuando Juan llegue debe tocar el timbre de la puerta y esperar unos momentos a que le atiendan. Cuando usted oye sonar el timbre, interrumpe la lectura, marca la página donde se quedó y va hasta la puerta para atender a la persona que toca el timbre. Una vez que Juan o la persona que ha tocado el timbre, se marcha, usted regresa a su asiento y retoma la lectura justo donde la dejó. Este último es un método más eficiente que el anterior porque le deja más tiempo para leer y elimina algunos inconvenientes como el de que Juan nunca llegue o se marche antes de que usted abra la puerta. Es, en principio, un método simple pero muy eficaz y eficiente, lo llamaremos atención por interrupción.

El primero de ellos, la encuesta, es un método eficaz, pero poco eficiente porque requiere realizar lecturas constantes y muchas veces innecesarias del estado del proceso que queremos atender. Sin embargo, es muy utilizado en la programación de microcontroladores porque resulta fácil de aprender, la implementación de código con este método es menos compleja y no requiere de hardware especial para llevarla adelante. Por otra parte, la encuesta, tiene muchas deficiencias que con frecuencia obligan al diseñador a moverse hacia otros horizontes.

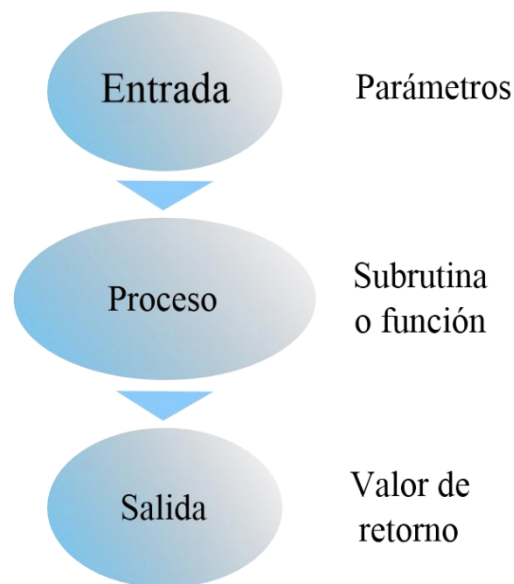
El mundo está lleno de situaciones; de las cuales no podemos determinar ni cuando, ni cómo ni por qué se producen, en la mayoría de los casos lo único que podemos hacer es enterarnos de que determinada situación, asociada a un proceso, ha ocurrido. Para ello seleccionamos alguna condición o grupo de condiciones que nos indican que el proceso que nos interesa debe ser atendido, a este fenómeno, en el cual se dan las condiciones que nos interesa conocer, lo llamaremos evento. En el segundo ejemplo vemos que para atender a Juan, éste debe tocar el timbre, por tanto, la llegada de Juan es el proceso que debemos atender y el sonido del timbre es el evento que nos indica que Juan ha llegado. [w27]

El método de atención a procesos por interrupción, visto desde la óptica del ejemplo que utilicé para mostrarlo, es más simple que el de la encuesta, pero no es cierto, el método se complica porque requiere que el microprocesador incorpore circuitos

adicionales para registrar los eventos que le indican que debe atender al proceso asociado y comprender estos circuitos y su dinámica no es una tarea sencilla.

Los circuitos para la atención a las interrupciones y todas las tareas que debe realizar el procesador para atender al proceso que lo interrumpe son bastante complejos y requieren una visión diferente de la que estamos acostumbrados a tener de nuestro mundo.

Los seres humanos no estamos conscientes de las interrupciones, en nuestro organismo existen mecanismos que nos interrumpen constantemente, para ello tenemos a nuestro sistema sensorial, pero no somos conscientes del proceso de interrupción, aunque sí de la atención a las interrupciones. Eso es porque incorporamos mecanismos que nos sacan rápidamente de la tarea que estamos haciendo para atender una situación que no puede o no debe esperar mucho tiempo. Bien, esa misma es la idea que se incorpora en los microprocesadores para atender procesos que no pueden esperar o que no sabemos cuándo deben ser atendidos porque ello depende de determinadas condiciones.



**Fig. 16: Diagrama de funcionamiento de una subrutina o interrupción.**

La cosa se complica en la secuencia de acciones a realizar desde el momento en que se desencadena el proceso de interrupción, hasta que se ejecuta el programa que lo atiende, y en la secuencia de acciones posteriores a la atención. Piense en cuantas cosas debe hacer su organismo ante una interrupción, utilicemos el segundo ejemplo para atender la llegada de Juan. Piense en cuantas cosas su cerebro hace a espaldas de su conciencia, desde el momento en que suena el timbre hasta que usted se encuentra

listo (consciente de que es probable que Juan ha llegado) para abrir la puerta, y todo lo que su cerebro debe trabajar para retomar la lectura después que Juan se ha marchado. Todo eso, excepto abrir la puerta y atender a Juan, lo hacemos de forma “inconsciente” porque para ello tenemos sistemas dedicados en nuestro organismo, pero en el mundo de los microcontroladores debemos conocer todos esos detalles para poder utilizar los mecanismos de interrupción.

Los procesos de atención a interrupciones tienen la ventaja de que se implementan por hardware ubicado en el procesador, así que es un método rápido de hacer que el procesador se dedique a ejecutar un programa especial para atender eventos que no pueden esperar por mecanismos lentos como el de encuesta.

En términos generales, un proceso de interrupción y su atención por parte del procesador, tiene la siguiente secuencia de acciones:

- ✓ En el mundo real se produce el evento para el cual queremos que el procesador ejecute un programa especial, este proceso tiene la característica de que no puede esperar mucho tiempo antes de ser atendido o no sabemos en qué momento debe ser atendido.
- ✓ El circuito encargado de detectar la ocurrencia del evento se activa, y como consecuencia, activa la entrada de interrupción del procesador.
- ✓ La unidad de control detecta que se ha producido una interrupción y “levanta” una bandera para registrar esta situación; de esta forma si las condiciones que provocaron el evento desaparecen y el circuito encargado de detectarlo desactiva la entrada de interrupción del procesador, ésta se producirá de cualquier modo, porque ha sido registrada.
- ✓ La unidad de ejecución termina con la instrucción en curso y justo antes de comenzar a ejecutar la siguiente comprueba que se ha registrado una interrupción.
- ✓ Se desencadena un proceso que permite guardar el estado actual del programa en ejecución y saltar a una dirección especial de memoria de programas, donde está la primera instrucción de la subrutina de atención a interrupción.
- ✓ Se ejecuta el código de atención a interrupción, esta es la parte “consciente” de todo el proceso porque es donde se realizan las acciones propias de la atención a la interrupción y el programador juega su papel.
- ✓ Cuando en la subrutina de atención a interrupción se ejecuta la instrucción de retorno, se desencadena el proceso de restauración del procesador al estado en que estaba antes de la atención a la interrupción.

Como podemos observar, el mecanismo de interrupción es bastante complicado, sin embargo tiene dos ventajas que obligan a su implementación: la velocidad y su

capacidad de ser asíncrono. Ambas de conjunto permiten que aprovechemos al máximo las capacidades de trabajo de nuestro procesador.

Los mecanismos de interrupción no solo se utilizan para atender eventos ligados a procesos que requieren atención inmediata sino que se utilizan además para atender eventos de procesos asíncronos.

Las interrupciones son tan eficaces que permiten que el procesador actúe como si estuviese haciendo varias cosas a la vez cuando en realidad se dedica a la misma rutina de siempre, ejecutar instrucciones una detrás de la otra.

#### **1.7.2.1.13 Comparadores**

Son circuitos analógicos basados en amplificadores operacionales que tienen la característica de comparar dos señales analógicas y dar como salida los niveles lógicos '0' o '1' en dependencia del resultado de la comparación. Es un periférico muy útil para detectar cambios en señales de entrada de las que solamente nos interesa conocer cuando está en un rango determinado de webetas.

#### **1.7.2.1.14 Modulador de ancho de pulsos**

Los PWM (Pulse Width Modulator) son periféricos muy útiles sobre todo para el control de motores, sin embargo hay un grupo de aplicaciones que pueden realizarse con este periférico, dentro de las cuales podemos citar: inversión DC/AC para UPS, conversión digital analógica D/A, control regulado de luz (dimming) entre otras.

#### **1.7.2.1.15 Memoria de datos no volátil**

Muchos microcontroladores han incorporado este tipo de memoria como un periférico más, para el almacenamiento de datos de configuración o de los procesos que se controlan. Esta memoria es independiente de la memoria de datos tipo RAM o la memoria de programas, en la que se almacena el código del programa a ejecutar por el procesador del microcontrolador.

Muchos de los microcontroladores PIC, incluyen este tipo de memoria, típicamente en forma de memoria EEPROM, incluso algunos de ellos permiten utilizar parte de la memoria de programas como memoria de datos no volátil, por lo que el procesador tiene la capacidad de escribir en la memoria de programas como si ésta fuese un periférico más.

### **1.7.3 CAPÍTULO 3.-**

#### **1.7.3.1 Los PICs de Microchip.**

La evolución de la electrónica desde la aparición del circuito integrado ha sido constante. Actualmente podemos encontrar dispositivos cada vez más complejos ubicados en encapsulados cada vez más pequeños, un ejemplo de esto son los microcontroladores.

Un microcontrolador, es un dispositivo electrónico encapsulado en un chip, capaz de ejecutar un programa. El microcontrolador reúne en un solo integrado: microprocesador, memoria de programa, memoria de datos y puertos de entrada/salida. Además, también suelen disponer de otras características especiales como: puertos serie, comparadores, convertidores analógico-digitales, etc. En el mercado existen gran variedad de microcontroladores de marcas y características distintas que podremos utilizar dependiendo de la aplicación a realizar.

Un microcontrolador ejecuta instrucciones. El conjunto de instrucciones es lo que llamamos programa. Las instrucciones son leídas de la memoria de programa para ejecutarlas una detrás de otra. La memoria de programa contiene las instrucciones que queremos que el microcontrolador ejecute.

Programar un microcontrolador consiste en introducir el programa en la memoria del microcontrolador. Las instrucciones son operaciones simples como sumar, restar, escribir en un puerto, activar un bit de un dato, etc.

Mediante estas instrucciones básicas podemos realizar operaciones más complejas y así llegar al objetivo de la aplicación.

En este Proyecto de taller III nos vamos a centrar en los microprocesadores de la casa Microchip Technology, es decir los PICs. Este tipo de microprocesadores están muy extendidos actualmente en el mercado gracias a su gran variedad y bajo coste. Otra razón del éxito de los PICs es su utilización, ya que una vez se ha aprendido a utilizar uno, conociendo su arquitectura y juego de instrucciones, es muy fácil emplear otro modelo diferente.

##### **1.7.3.1.1 Historia de los PICs.**

En 1965 GI formó una división de microelectrónica destinada a generar las primeras arquitecturas viables de memoria EPROM y EEPROM. A principios de 1970 se creó el CP1600 que era un microprocesador bastante bueno pero no manejaba muy bien

los puertos de E/S, por esta razón en 1975 se creó el PIC que era un chip que funcionaba en combinación con este microprocesador para controlar las E/S. [w26]

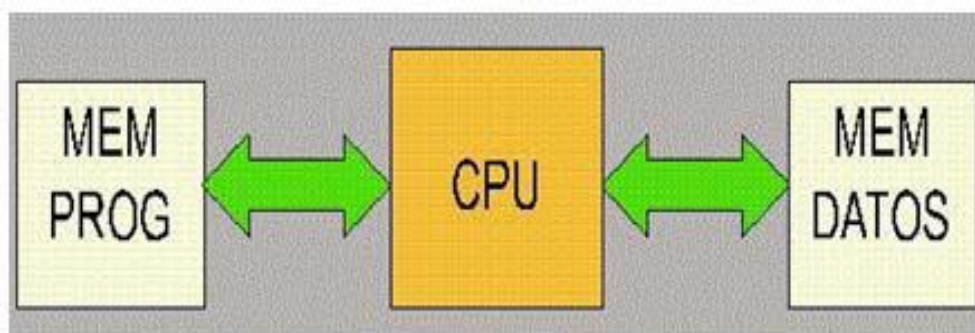
Alrededor de 1980 GI reestructuró la empresa y creó la GI Microelectronics que finalmente fue vendida a un grupo de inversores de capital de riesgo que crearon la actual Arizona Microchip Technology y centraron su negocio en la fabricación de PICs, memorias EEROM y EPROM. Actualmente Microchip ha realizado un gran número de mejoras a la arquitectura original de los PICs, adaptándola a las actuales tecnologías y al bajo costo de los semiconductores. [w26]

### 1.7.3.1.2 Características de los PICs.

Las características más destacadas de los PICs las enumeramos en los siguientes puntos:

- ✓ La arquitectura del procesador sigue el modelo Harvard.  
Tradicionalmente, las computadoras y microprocesadores siguen el modelo propuesto por John Von Neumann, en el cual la unidad central de proceso, o CPU, está conectada a una memoria única que contiene las instrucciones del programa y los datos. El tamaño de la unidad de datos o instrucciones está fijado por el ancho del bus de la memoria. Esto limita la velocidad de operación del microprocesador, ya que no se puede buscar en la memoria una nueva instrucción, antes de que finalicen las transferencias de datos que pudieran resultar de la instrucción anterior.

En los microprocesadores PIC se utiliza el modelo Harvard. Este tipo de arquitectura conecta de forma independiente y con dos buses distintos la memoria de instrucciones y la de datos:

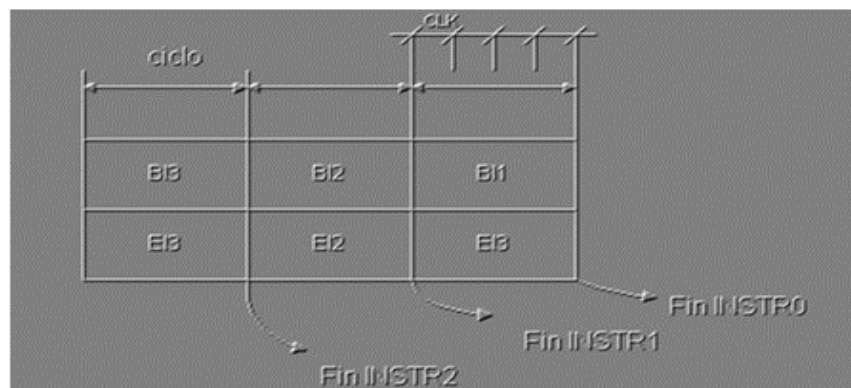


**Fig.17. Arquitectura Harvard.**

La arquitectura Harvard permite al CPU acceder simultáneamente a las dos memorias. Esto proporciona mayor velocidad además de numerosas ventajas al funcionamiento del sistema.

- ✓ Se aplica la técnica de segmentación ("pipe-line") en la ejecución de las instrucciones.

La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de la siguiente. De esta forma se puede ejecutar cada instrucción en un ciclo (un ciclo de instrucción equivale a cuatro ciclos de reloj).



**Fig.18. Técnica de segmentación.**

La segmentación permite al procesador ejecutar cada instrucción en un ciclo de instrucción equivalente a cuatro ciclos de reloj. En cada ciclo se realiza la búsqueda de una instrucción y la ejecución de la anterior.

Las instrucciones de salto ocupan dos ciclos al no conocer la dirección de la siguiente instrucción hasta que no se haya completado la de bifurcación.

- ✓ El formato de todas las instrucciones tiene la misma longitud.  
Todas las instrucciones de los microcontroladores de la gama baja tienen una longitud de 12 bits. Las de la gama media tienen 14 bits y más las de la gama alta. Esta característica es muy ventajosa en la optimización de la memoria de instrucciones y facilita enormemente la construcción de ensambladores y compiladores.
- ✓ Procesador RISC (Computador de Juego de Instrucciones Reducido).  
Dependiendo de la gama del procesador (baja, media o alta) tienen más o menos número de instrucciones. Los modelos de la gama baja disponen de un repertorio de 33 instrucciones, 35 los de la gama media y unas 76 los de la alta.
- ✓ Todas las instrucciones son ortogonales.

Cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.

- ✓ Arquitectura basada en un banco de registros.  
Esto significa que todos los objetos del sistema (puertos de E/S, temporizadores, posiciones de memoria, etc.) están implementados físicamente como registros.
- ✓ Diversidad de modelos de microcontroladores con prestaciones y recursos diferentes.  
La gran variedad de modelos de microcontroladores PIC permite que el usuario pueda seleccionar el más conveniente para su proyecto.
- ✓ Herramientas de soporte potentes y económicas.  
En nuestro caso utilizaremos las herramientas de software que nos facilita la casa Microchip. Estas herramientas las podemos descargar fácilmente de la Web de microchip ([www.microchip.com](http://www.microchip.com)).

### 1.7.3.1.3 Gamas de PICs.

Existen gran cantidad de aplicaciones que podemos realizar con Pics, aplicaciones sencillas en las cuales no necesitamos muchos recursos y aplicaciones más complejas en las cuales necesitamos microcontroladores muy potentes, por ello y siguiendo esta filosofía, la empresa Microchip fabrica tres tipos de gamas de microcontroladores Pic para atender todas las aplicaciones, microcontroladores de gama baja, gama media y gama alta. Así, hay disponibles microcontroladores sencillos y baratos para atender las aplicaciones simples y otros complejos y más costosos para las de mucha envergadura.

Con las tres gamas de PIC se dispone de gran diversidad de modelos y encapsulados, pudiendo seleccionar el que mejor se acople a las necesidades de acuerdo con el tipo y capacidad de las memorias, el número de líneas de E/S y las funciones auxiliares precisas. Sin embargo, todas las versiones están construidas alrededor de una arquitectura común, un repertorio mínimo de instrucciones y un conjunto de opciones muy apreciadas, como el bajo consumo y el amplio margen del voltaje de alimentación.

Existen dos arquitecturas utilizadas en la fabricación de microcontroladores:

- ✓ **Microcontroladores de arquitectura cerrada**  
En este tipo de arquitectura el microcontrolador tiene unos recursos específicos los cuales no permiten ningún tipo de modificación, es decir, no admiten ningún tipo de variaciones ni de ampliaciones. La aplicación a la que se destina debe encontrar en su estructura todo lo que precisa y, en caso contrario, hay que desecharlo. Microchip ha elegido principalmente este modelo de arquitectura.

✓ **Microcontroladores de arquitectura abierta**

Este tipo de microcontroladores a parte de tener una estructura interna determinada, permiten ampliación emplear sus líneas de E/S para sacar al exterior los buses de datos, direcciones y control, con lo que se posibilita la ampliación de la memoria y las E/S con circuitos integrados externos. Microchip dispone de modelos PIC con arquitectura abierta, sin embargo, esta alternativa se escapa de la idea de un microcontrolador incrustado y se asemeja a la solución que emplean los clásicos microprocesadores.

**1.7.3.1.4 La memoria.**

Los PICs, al estar contruidos con arquitectura Harvard, poseen dos bloques de memoria distintos, una para la memoria de programa y otra para la de datos. Estas dos memorias son independientes entre ellas teniendo tamaño y longitudes de palabra distintas. Cada bloque posee su propio bus, de tal forma que el acceso a cada uno puede producirse durante el mismo ciclo del oscilador.

❖ **Memoria de datos.**

La memoria de datos puede dividirse en la RAM de fines generales y los Registros de Funciones Especiales (SFR).

La memoria de datos contiene también los datos de la memoria EEPROM. Esta memoria no está directamente introducida en la memoria de datos, si no es registrada en forma indirecta. Esto significa que un puntero indirecto de direcciones especifica la dirección de la memoria de datos EEPROM para escribir y leer.

La memoria de datos se organiza en “bancos” y la cantidad de estos depende de la gama de PIC que estemos utilizando. Dependiendo del área de memoria a la cual queremos referir a la hora de programar la tenemos que definir mediante el registro FSR.

Esta memoria de datos funciona de forma similar al "banco de registros" de un procesador por lo cual sus posiciones implementan registros de propósito especial y propósito general.

Las primeras posiciones de la memoria se destinan a registros específicos.

### ❖ **Memoria de programa.**

Esta memoria también se le denomina memoria de instrucciones. Aquí es donde nosotros escribimos nuestro programa.

Existen diferentes tipos de memorias de programa dependiendo de las necesidades, en nuestro caso nos dedicaremos a los PICs con memoria tipo flash. Este tipo de memoria se puede programar y borrar eléctricamente alrededor de 1000 veces, lo cual la hace muy útil para el aprendizaje ya que con una pequeña inversión en un chip lo podemos programar tantas veces como queramos.

La cantidad de memoria también depende de la gama de PIC que estemos utilizando ya que para gamas bajas y programas simples disponemos de PICs con menos memoria y menos costosos y para proyectos de mayor envergadura disponemos de PICs de más memoria.

### ❖ **Registros.**

Los PICs utilizan una arquitectura basada en registros. Esto significa que todos los objetos del sistema (puertos de E/S, temporizadores, posiciones de memoria, etc.) están implementados físicamente como registros.

Existen diferentes tipos de registros:

- ✓ Registro de propósito general.
- ✓ Puertos E/S, Contadores, etc.
- ✓ Registros especiales.
- ✓ Registros de funcionamiento y configuración.

Todos los registros están ubicados en una posición específica de la memoria.

### ❖ **Contador de programa.**

Este registro, normalmente denominado PC (program counter), es totalmente equivalente al de todos los microprocesadores y contiene la dirección de la próxima instrucción a ejecutar. Se incrementa automáticamente al ejecutar cada instrucción, de manera que la secuencia natural de ejecución del programa es lineal, una instrucción después de la otra.

Algunas instrucciones que llamaremos de control, cambian el contenido del PC alterando la secuencia lineal de ejecución. Dentro de estas instrucciones se encuentran el GOTO y el CALL que permiten cargar en forma directa un valor constante en el PC haciendo que el programa salte a cualquier posición

de la memoria. Otras instrucciones de control son los SKIP o “saltos” condicionales, que producen un incremento adicional del PC si se cumple una condición específica, haciendo que el programa salte, sin ejecutar, la instrucción siguiente.

Al resetearse el microprocesador, todos los bits del PC toman valor 1, de manera que la dirección de arranque del programa es siempre la última posición de memoria de programa. En esta posición se deberá poner una instrucción de salto al punto donde verdaderamente se inicia el programa.

A diferencia de la mayoría de los microprocesadores convencionales, el PC es también accesible al programador como registro de memoria interna de datos, en la posición de 02. Es decir que cualquier instrucción común que opere sobre registros puede ser utilizada para alterar el PC y desviar la ejecución del programa. El uso indiscriminado de este tipo de instrucciones complica el programa y puede ser muy peligroso, ya que puede producir comportamientos difíciles de predecir. Sin embargo, algunas de estas instrucciones utilizadas con cierto método, pueden ser muy útiles para implementar poderosas estructuras.

#### ❖ **Stack.**

En los microcontroladores PIC el stack es una memoria interna dedicada, de tamaño limitado, separada de las memorias de datos y de programa, inaccesible al programador, y organizada en forma de pila, que es utilizada solamente, y en forma automática, para guardar las direcciones de retorno de subrutinas e interrupciones. Permite guardar una copia completa del PC. Como en toda memoria tipo pila, los datos son accedidos de manera tal que el primero que entra es el último que sale.

La posición de la pila en un PIC está limitada, lo que no permite hacer uso intensivo del anidamiento de subrutinas. Solo se pueden anidar dos niveles de subrutinas, es decir que una subrutina que es llamada desde el programa principal, puede a su vez llamar a otra subrutina, pero esta última no puede llamar a una tercera, porque se desborda la capacidad del stack, que solo puede almacenar dos direcciones de retorno.

Aunque esto a priori sea un problema es una buena solución de compromiso ya que estos microcontroladores están diseñados para aplicaciones de alta velocidad en tiempo real, en las que el overhead (demoras adicionales) que ocasiona un excesivo anidamiento de subrutinas es inaceptable.

Como ya se mencionó anteriormente, el stack y el puntero interno que lo direcciona, son invisibles para el programador, solo se los accede

automáticamente para guardar o rescatar las direcciones de programa cuando se ejecutan las instrucciones de llamada o retorno de subrutinas, o cuando se produce una interrupción o se ejecuta una instrucción de retorno de ella.

#### ❖ Puertos de Entrada / Salida.

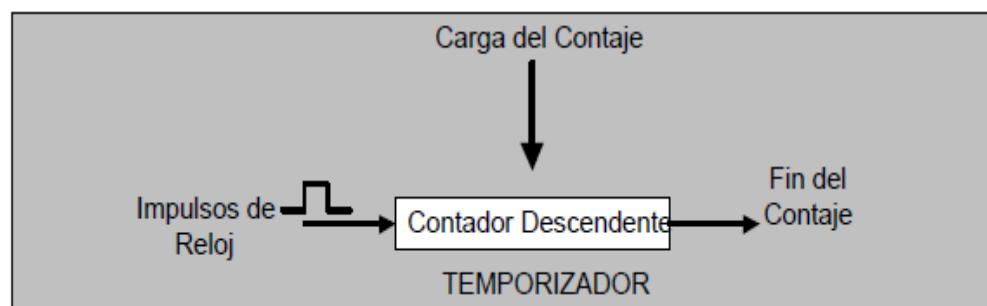
Un recurso imprescindible para los microcontroladores son los puertos de Entradas y Salidas con los cuales se comunica con los periféricos del mundo exterior.

Dependiendo de la gama de PIC que estemos utilizando estos tendrán más o menos puestos de E/S y también dependiendo de la gama utilizada pueden ser digitales, analógicas, multiplexadas, etc. También algunos PICs permiten utilizarlas como comparadores y conversores, pero esas características ya dependen de cada PIC en particular.

Los puertos de entrada y salida utilizan la denominación de Puerto A, Puerto B, etc., dependiendo de la cantidad de puertos que tengan.

#### ❖ Temporizador / Contador.

Una exigencia en las aplicaciones de control es la regulación estricta de los tiempos que duran las diversas acciones que realiza el sistema. El dispositivo típico destinado a gobernar los tiempos recibe el nombre de temporizador o "timer" y, básicamente, consiste en un contador ascendente o descendente que determina un tiempo determinado entre el valor que se le carga y el momento en que se produce su desbordamiento o paso por 0.



**Fig. 19. Temporizador**

La figura 9 es un esquema simplificado de un temporizador. En este caso se trata de un contador descendente, que, una vez cargado con un valor, se decrementa al ritmo de los impulsos de reloj hasta que llega a 0.

La cantidad de temporizadores depende de la gama de PIC que estemos utilizando, la gama baja los microcontroladores PIC sólo disponen de dos temporizadores. Uno de ellos actúa como Principal y sobre él recae el control de tiempos de las operaciones del sistema. El otro recibe el nombre de Perro guardián o "Watchdog". Las gamas más altas de microcontroladores PIC disponen de más temporizadores.

El Perro guardián vigila que el programa no se "cuelgue" y dejen de ejecutarse las instrucciones secuenciales del mismo tal como lo ha previsto el diseñador.

Para realizar esta labor de vigilancia, el Perro guardián da un paseo por la CPU cada cierto tiempo y comprueba si el programa se ejecuta normalmente; en caso contrario, por ejemplo si el control está detenido en un bucle infinito o a la espera de algún acontecimiento que no se produce, el perro ladra y provoca un reset, reiniciando todo el sistema.

Tanto el Temporizador principal, TMR0, como el Perro guardián, WDT, a veces precisan controlar tiempos largos y aumentar la duración de los impulsos de reloj que les incrementan o decrementan. Para cubrir esta necesidad, se dispone de un circuito programable llamado Divisor de frecuencia que divide la frecuencia utilizada por diversos rangos para poder realizar temporizaciones más largas.

#### ❖ **Interrupciones.**

Una interrupción consiste en una detención del programa en curso para realizar una determinada rutina que atienda la causa que ha provocado la interrupción.

Es como una llamada a subrutina, que se origina por otra causa que por una instrucción del tipo CALL. Tras la terminación de la rutina de interrupción, se retorna al programa principal en el punto en que se abandonó.

Las causas que originan una interrupción pueden ser externas, como la activación de una patita con el nivel lógico apropiado, e internas, como las que pueden producirse al desbordarse un temporizador, como el TMR0.

En las aplicaciones industriales, las interrupciones son un producto muy potente para atender los acontecimientos físicos en tiempo real.

Existen diferentes tipo de interrupciones que también dependen del tipo de microcontrolador que utilizamos por ello si queremos saber más sobre las interrupciones de un PIC en concreto debemos consultar del data sheet del PIC utilizado.

### ❖ **Instrucciones.**

Como ya hemos comentado anteriormente, dependiendo de la gama del microcontrolador que estemos utilizando (baja, media o alta) tienen más o menos número de instrucciones. Los modelos de la gama baja disponen de un repertorio de 33 instrucciones, 35 los de la gama media y unas 76 los de la alta.

En este apartado no vamos a entrar en detalle sobre las instrucciones de que dispone cada PIC, para ello tendremos que repasar los data sheet de cada microcontrolador concreto.

### ❖ **Modos de direccionamiento.**

Cuando programamos un PIC, a la hora de especificar los datos y operandos lo podemos hacer de 3 formas distintas que llamamos modos de direccionamiento. Estos 3 modos de direccionamiento son:

- ✓ Inmediato.
- ✓ Directo.
- ✓ Indirecto.

El modo inmediato es cuando utilizamos el valor literal.

El modo directo es cuando utilizamos un valor que apunta a una determinada posición de memoria.

El modo indirecto es cuando utilizamos como operando el registro INDF que accede a la posición que apunta el contenido del registro FSR ubicado en el área de datos.

#### **1.7.3.1.5 Oscilador externo.**

Todo Microcontrolador requiere un circuito externo que le indique la velocidad a la que debe trabajar. Este circuito, que se conoce con el nombre de oscilador o reloj, es muy simple pero de vital importancia para el buen funcionamiento del sistema. Los PIC pueden utilizar cuatro tipos de osciladores diferentes:

- ✓ RC. Oscilador con resistencia y condensador.
- ✓ XT. Cristal de cuarzo.
- ✓ HS. Cristal de alta velocidad.
- ✓ LP. Cristal para baja frecuencia y bajo consumo de potencia.

En el momento de programar o "quemar" el microcontrolador se debe especificar qué tipo de oscilador se usa. Esto se hace a través de unos fusibles llamados "fusibles de configuración".

Generalmente se suele utilizar un cristal de 4Mhz y de esta manera internamente esta frecuencia queda dividida por cuatro, lo que hace que la frecuencia efectiva de trabajo sea de 1 MHz, por lo que cada instrucción se realiza en un microsegundo ( $1 \mu\text{S}$ ). El cristal debe ir acompañado de dos condensadores y se conecta como se muestra en la figura.

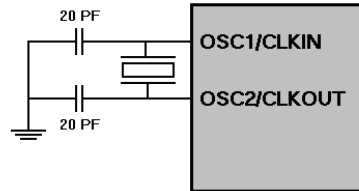


Fig. 20. Oscilador de cristal.

Si no se requiere mucha precisión en el oscilador y se requiere economizar dinero, se puede utilizar una resistencia y un condensador, como se muestra a continuación.

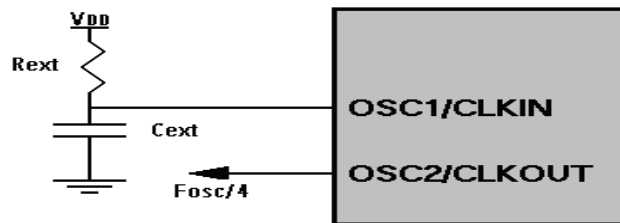


Fig. 21. Oscilador RC.

Los valores recomendados para este tipo de oscilador son:  $5 \text{ K}\Omega \leq R_{\text{ext}} \leq 100 \text{ K}\Omega$  y  $C_{\text{ext}} > 20 \text{ pF}$

## **1.7.4 CAPÍTULO 3.-**

### **1.7.4.1 Código MORSE**

#### **1.7.4.1.1 Introducción**

El Código Morse es un medio de comunicación basado en la transmisión y recepción de mensajes empleando sonidos o rayos de luz y un alfabeto alfanumérico compuesto por puntos y rayas. Aunque este código surgió en el siglo 19, su empleo es perfectamente utilizable hoy en día cuando la existencia de condiciones atmosféricas adversas no permite el empleo de otros medios más desarrollados como, por ejemplo, la transmisión de la voz.

Aun cuando en una transmisión inalámbrica por radiofrecuencia realizada solamente con código Morse aparezcan interferencias producidas por tormentas eléctricas, los sonidos de los puntos y las rayas serán siempre reconocibles para el oído humano aunque se escuchen mezclados con el ruido que produce en esos casos la estática atmosférica.

En sus inicios para transmitir y recibir mensajes en Código Morse se empleaba un primitivo aparato inventado en 1844 por Samuel Morse, creador a su vez del propio código que lleva su nombre. Ese aparato constaba de una llave telegráfica de transmisión, que hacía las veces de interruptor de la corriente eléctrica y un electroimán como receptor de los puntos y las rayas.

Cada vez que la llave se oprimía hacia abajo con los dedos índice y medio se establecía un contacto eléctrico que permitía transmitir los puntos rayas del código Morse. Los impulsos intermitentes que se producían al apretar la llave telegráfica se enviaban a un tendido eléctrico compuesto por dos alambres de cobre. Esos cables, soportados por postes de madera, se extendían muchas veces a cientos de kilómetros de distancia a partir del punto de origen de la transmisión hasta llegar al punto de recepción.



**Fig22: Llave telegráfica Morse**

El primitivo receptor de ese sistema de telegrafía por donde se oía el sonido de los puntos y las rayas estaba formado por un electroimán con una bobina de alambre de cobre enrollada alrededor de un núcleo de hierro. Cuando la bobina recibía los impulsos de corriente eléctrica correspondientes a los puntos y las rayas, el núcleo de hierro se magnetizaba y atraía hacia sí una pieza móvil, también de hierro, que al golpearlo emitía un sonido seco peculiar. Ese sonido era semejante a un “tac” corto cuando se recibía un punto, o un “taaac” más largo si se recibía una raya. Por ejemplo, la letra “a” del código Morse, formada por un punto y una raya ( . – ), se oía aproximadamente así: “tac – taaac”.

Con el invento de Marconi del transmisor elemental de ondas de radio, a partir del año 1901 la transmisión de mensajes por telegrafía se comenzó a realizar también de forma inalámbrica, adaptándolo al mismo sistema inventado por Morse. Esa nueva forma de transmisión tenía la ventaja que no era necesario realizar tendidos de cables a largas distancias, por lo que muy pronto los barcos se adoptaron esa nueva tecnología para comunicarse entre sí y con tierra. El “telegrafista” pasó entonces a llamarse “radiotelegrafista”.

La posterior aparición de la válvula de vacío inventada por Fleming en 1904 y el desarrollo de la válvula tríodo inventada por Lee de Forest tres años después, abrieron la posibilidad de generar ondas de radiofrecuencia por medios electrónicos. Ese avance tecnológico mejoró en gran medida la transmisión de mensajes en código Morse por vía inalámbrica, permitiendo su envío a cualquier confín del mundo.

Con la introducción en el mercado de los transmisores electrónicos por ondas de radiofrecuencia, el electroimán utilizado hasta entonces para recibir las señales del código Morse se sustituyó por un altoparlante o, en su defecto, un par de cascos (audífonos) y el sonido pasó a escucharse como “beeps” cortos o largos, según fuera un punto o una raya lo que se estuviera recibiendo. La llave telegráfica de Morse se sustituyó también por otra llamada "vibroplex bug", inventada en 1903 por Horace G. Martin que posibilitaba enviar los mensajes con mayor rapidez. El pulsador de esta nueva llave funcionaba de forma horizontal y se manipulaba haciendo presión hacia los lados utilizando el dedo índice y el pulgar.

Además de las transmisiones de mensajes que se realizan empleando sistemas eléctricos o electrónicos, el código Morse permite utilizar también otros medios más sencillos. Uno de ellos consiste en utilizar una fuente de luz intermitente, mientras que el otro se basa en producir sonidos empleando cualquier dispositivo que permita reproducir los puntos y las rayas. Un ejemplo del uso práctico de esos diversos métodos lo tenemos principalmente en los barcos, que en determinados casos pueden llegar a emplear cuales quiera de las posibilidades que se han mencionado.

Por ejemplo, para enviar mensajes empleando una fuente de luz los barcos se valen de una especie de reflector llamado “blinker”, dotado de una cortinilla que al abrirse deja pasar los rayos de luz y al cerrarse los interrumpe. Un rayo de luz corto se entiende como un punto, mientras uno más largo es una raya. A la derecha se puede ver un blinker transmitiendo un S.O.S. pidiendo auxilio. La formación de esas siglas en código Morse se realiza con tres puntos que corresponden a la letra (S), tres rayas a la letra (O) y tres puntos más (igualmente para la otra S) ( . . . --- . . . )



**Fig23: Blinker**

En casos de emergencia los barcos suelen utilizar también el “tifón” (silbato accionado por un chorro de vapor o de aire), que llevan comúnmente fijado a su chimenea; gracias al fuerte y grave sonido que emiten los tifones, se pueden utilizar

para propagar los sonidos de mensajes de auxilio en código Morse. Un sonido corto del tífón significa un punto, mientras que uno más largo significa una raya.

Para transmitir las letras del código, cada punto y cada raya se separa haciendo breves pausas. La velocidad de transmisión de las palabras que forman el texto de los mensajes depende en gran medida de la habilidad y experiencia práctica que tenga el telegrafista o el radiotelegrafista, tanto a la hora de transmitir como de recibir los mensajes.

Independientemente de la velocidad y destreza que se pueda llegar a adquirir empleando el código Morse, a la hora de transmitir un mensaje el tiempo de demora de una raya debe superar en tres veces el de un punto.

Cada letra o número del código se compone de uno o más puntos o rayas, o las combinaciones de ambos signos, separados entre sí por una pausa de tiempo equivalente al de la transmisión de un punto. Además, entre la transmisión de una letra y la siguiente, el tiempo de separación debe ser mayor que el necesario para transmitir una raya o tres puntos. El tiempo de separación entre una palabra y la otra debe ser equivalente al que se requiere para transmitir seis puntos. [w21] y [w22]

#### 1.7.4.1.2 Tablas de Código Morse

Letra	Código	Letra/Número	Código
A	. -	R	. - .
B	- . . .	S	. . .
C	- . - .	T	-
Ch	- - - -	U	. . -
D	- . .	V	. . . -
E	.	W	. - -
F	. . - .	X	- . . -
G	- - .	Y	- . - -
H	. . . .	Z	- - . .
I	. .	1	. - - - -
J	. - - -	2	. . - - -
K	- . -	3	. . . - -
L	. - . .	4	. . . . -
M	- -	5	. . . . .

N	-. .	6	- . . . .
Ñ	--- . ---	7	--- . . .
O	---	8	--- . .
P	. --- .	9	--- . . . .
Q	--- . -	0	--- . . . .

**Fig24: Alfanumérica de código Morse**

<b>Signos</b>	<b>Nombre</b>	<b>Código</b>
'	Apóstrofe	. - - - - .
/	Barra diagonal	- . . - .
,	Coma	--- . ---
“ ”	Comillas	. - . . - .
:	Dos puntos	--- . . . .
-	Guión corto	- . . . . -
=	Igualdad	- . . . -
?	Interrogación	. . - - . .
() []	Paréntesis	- . - - . -
;	Punto y coma	- . - . - .
.	Punto y seguido o aparte	. - . - . -

**Fig25: Signos en Código Morse**

<b>Ejecutar:</b>	<b>Código</b>
Acento	. - - - - .
Enterado	. . . - .
Error	. . . . .
Espera	. - . . .
Fin de texto	. . . - . -
Fin del mensaje	. - . - .
Recibido	. - .
Subrayado de texto	--- . . . .

**Fig26: Caracteres especiales en Código Morse**

## **2 Capítulo II COMPONENTES**

### **2.1 Entorno simulado para un circuito en el cual se realice entrada y salida de datos por puerto usb de arquitectura abierta.**

#### **2.1.1 Plan de desarrollo de software y hardware**

##### **2.1.1.1 Introducción**

El presente trabajo se centrara en la simulación, la programación e implementación del circuito, para la comunicación USB con un microcontrolador PIC18F4550, asumiendo que se realizó la previa programación del software en el programa CCS PCWHD en el lenguaje de programación C. y la simulación del hardware con el programa Proteus7 Professional al fin de demostrar la comunicación mediante este puerto, también se llevara a cabo el desarrollo de la interfaz gráfica para la comunicación PC/Microcontrolador en Java haciendo uso de la plataforma Eclipse.

En un comienzo se contaba con muchas ideas muy ambiciosas pero con muy pocos conocimientos sobre la mayoría de las áreas involucradas. Más allá de las limitaciones de conocimientos y la amplitud de las áreas a investigar pensé que era un buen desafío para realizar como proyecto de grado.

Se asume este enfoque debido a que en general los dispositivos electrónicos existentes (sensores, actuadores, displays, ADCs, DACs, etc.) utilizan múltiples interfaces y protocolos de comunicación ajenos al contexto de las computadoras personales, genera que la interacción no sea una tarea trivial y en general implique desarrollar soluciones particulares para lograr su integración.

Esta heterogeneidad motiva la búsqueda de un medio de comunicación existente en un PC, lo suficientemente versátil para satisfacer la mayoría de los requerimientos. Desde hace unos años la tecnología USB se ha convertido en un estándar, lo que ha llevado a una proliferación de dispositivos y un auge en su uso. La facilidad de uso, ancho de banda y funcionalidad Plug&Play son algunas de las características más atractivas para utilizar al USB como medio de comunicación.

El siguiente proyecto utilizara los algoritmos estudiados durante a lo largo de la carrera, aplicando los fundamentos teóricos de [2] y otros. Si bien estos autores implementaron sus algoritmos propuestos de acuerdo a las restricciones que pusieron en sus trabajos, las características del problema propuesto, en este proyecto de grado

son distintas, implicando una adecuación o readecuación de dichos algoritmos a los objetivos y alcances perseguidos por este trabajo.

Esto fue una tarea muy exigente y requirió más tiempo del planificado pero me permitió seleccionar y plasmar de mejor manera todas las ideas generales que se tenían al comienzo del proyecto en un conjunto inicial de requerimientos.

### **2.1.1.2 Propósito**

El propósito del Plan del Desarrollo del Software y del Hardware es proporcionar la información necesaria para controlar la conexión del puerto USB y la interfaz usuario, en él se describe el enfoque de desarrollo del software y del Hardware.

Los usuarios del plan de Desarrollo del software y del Hardware son:

- El jefe del proyecto, que lo utiliza para organizar la agenda y necesidades de recursos, y realizar su seguimiento.
- Los Docentes de la materia de Taller III, para evaluar el cumplimiento del proyecto.

### **2.1.1.3 Alcances**

Con el plan de Desarrollo del Software y hardware se pretende analizar y elaborar el “Proyecto”, abarcando las metodologías de investigación y de desarrollo ya mencionadas.

Para ello se elaborará un cronograma con cada una de las actividades a realizar y se especificará los detalles de construcción del proyecto para los distintos roles que cumplen los desarrolladores.

### **2.1.1.4 Resumen**

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

Vista General del Proyecto.- Proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.

Organización del Proyecto.- Describe la estructura organizacional del equipo de desarrollo.

Gestión del Proceso.- Explica los costos y planificación estimada, define las fases e hitos del proyecto y describe como se realizará su seguimiento.

Planes y Guías de aplicación.- Proporciona una lista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.

## 2.1.2 Vista general del proyecto

### 2.1.2.1 Propósito, alcance y objetivo

La información que a continuación se incluye es el resultado de los diferentes progresos de las investigaciones que se realizó y se realizan para la implementación correcta del software y del hardware para una completa sincronización del usuario con el proyecto.

El desarrollo del siguiente proyecto pretende abrir las puertas para futuras investigaciones sobre la comunicación de diferentes dispositivos a través del puerto USB.

El proceso de administración de proyectos empieza con la identificación de una idea para proyecto. La cual es el desarrollo de un dispositivo de comunicación bidireccional el cual permitirá tener un mejor control del flujo de datos, la misma va dirigida a todo grupo de estudiantes que desee trabajar en nuevos tipos de desarrollo para la interconexión de distintos dispositivos, utilizando la tecnología USB y así poder controlar la fluidez de los datos de los dispositivos el cual brindara una mejor conexión entre si y confiabilidad en el proceso de interacción con el usuario.

El sistema podrá generar distintas funciones donde se podrá visualizar cada resultado tanto en la Pc como en el simulador del circuito en el que estarán conectados los dispositivos.

Podrán tener acceso a las diferentes funciones tanto por la interacción de la Pc como por el simulador del circuito obteniendo una visualización tanto en la interfaz usuario como en el hardware.

### 2.1.2.2 Suposiciones y restricciones

Acontecimientos que deben ocurrir para que el proyecto se ejecute con éxito, pero que están fuera del ámbito de control por parte de la proyectista.

- **Suposiciones.-** En el documento se expresan los requisitos en términos de lo que el sistema debe proporcionar a los usuarios que hagan uso de él para verificar en el hardware (Circuito simulado). No obstante, el sistema ni el hardware (Circuito simulado) no proporcionará nada útil a menos que haya alguien que introduzca los datos. Se asumirá, por tanto, que los usuarios serán los responsables de realizar la introducción de datos, para verificar el envío y recepción de datos mediante el puerto USB.

Se asume que los requisitos descritos en este documento son estables una vez que sea aprobado por los diferentes docentes de taller III.

Tanto el sistema de interacción como el hardware podrá facilitar el manejo de la información de forma eficiente y segura mediante el uso del puerto USB, con la utilización de la misma podrá mejorar el nivel de conocimiento en el uso de las nuevas tecnologías mediante la capacitación al momento de emplearlo.

- **Restricciones.-** Dado que el sistema implementa una conexión mediante un simulador del circuito, se limita la comunicación entre la Pc y el hardware (Circuito simulado) dependiendo del entorno de simulación utilizado, también actualmente se encuentra vigente otro tipo de conexiones inalámbricas de lo que se espera un cambio en el tipo de trabajo de conexiones y de impacto sobre el proyecto.

Otra restricción importante es la naturaleza de la infraestructura del hardware a utilizar, pues siempre está cambiando para un mejor rendimiento.

### 2.1.2.3 Entregable del proyecto

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración del software la perspectiva del hardware, que propondremos para este proyecto.

Es preciso destacar que de acuerdo a la metodología RUP y de investigación se realizaran varias pruebas hasta lograr el resultado esperado, todos los proyectos son objeto de modificaciones a lo largo del proceso de desarrollo con lo cual solo al término del proceso podríamos tener una versión definitiva y completa del proyecto que estamos realizando.

- **Visión**  
Este documento define la visión del proyecto desde la perspectiva del usuario, especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema.
- **Diseño de Interfaces de Usuario**  
Se trata de prototipos de interfaz que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así conseguir una retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Que serán entregados al final de la fase de elaboración. Asimismo, este artefacto, será desechado en la fase de construcción en la

medida que los resultados de las interacción vayan desarrollando el producto final.

- **Modelo de Análisis y Diseño**

Este modelo establece la realización de los circuitos, microcontrolador y otros, pasando desde una representación en términos de análisis (incluye aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

- **Modelo de Implementación**

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: librerías, ejecutables, código fuente, y todo otro tipo de componentes necesarios para la implantación y despliegue del proyecto. (Este modelo es sólo una versión preliminar al final de la fase de Elaboración, posteriormente tiene bastante refinamiento).

- **Manual de Instalación**

Este documento incluye las instrucciones para realizar el manejo de este producto.

- **Material de Apoyo al Usuario Final**

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo; Guías de usuario y otros necesarios.

#### **2.1.2.4 Evolución del plan de desarrollo del software**

El Plan de Desarrollo de Software se revisará semanalmente y se refinará antes del comienzo de cada iteración.

### **2.1.3 Organización del proyecto**

#### **2.1.3.1 Participante del proyecto**

Director del Proyecto: Univ. Magali Antonia Castrillo.

#### **2.1.3.2 Roles y responsabilidades**

A continuación se describen las principales responsabilidades durante las fases de inicio y elaboración, de acuerdo con los roles que desempeñan en RUP.

<b>Responsable</b>	<b>Actividades</b>
<b>Director</b> Magali Antonia Castillo	Será el encargado de realizar un seguimiento continuo del desempeño y la evaluación a todas las actividades o tareas que se realizara, para llegar a obtener el producto.
<b>Analista/investigador</b>	Será el encargado de realizar la investigación, analizar y documentar la misma de acuerdo a las normas dadas para la presentación del proyecto.
<b>Diseñador</b>	Será el encargado de diseñar, elaborar el sistema y dispositivo durante el desarrollo del proyecto hasta la culminación del mismo.
<b>Programador</b>	Será el encargado de desarrollar el código fuente que sea necesario en el proyecto tomando en cuenta las normas de programación, según las necesidades, requerimientos del proyecto
<b>Asesor</b>	Será el encargado de guiar, revisar y orientar el desarrollo y elaboración del proyecto.

**Tabla31 Detalle de responsabilidades del personal del proyecto.**

## **2.1.4 Gestión del proyecto**

### **2.1.4.1 Estimaciones del proyecto**

El presupuesto del proyecto y los recursos involucrados se adjuntan en un documento por separado.

### **2.1.4.2 Plan del proyecto**

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

#### **2.1.4.2.1 Plan de las Fases**

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una distribución de tiempo y el número de iteraciones de cada fase de forma preliminar.

Fase	Nro. Iteraciones	Duración (días)
Fase de Inicio	1	80
Fase de Elaboración	2	90
Fase de Construcción	3	115
Fase de Transición	1	45

**Tabla 32 Cuadro de iteraciones de las fases RUP del proyecto.**

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Descripción	Hito
Fase de Inicio	En esta fase se desarrollará los requisitos del proyecto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto visión. Los principales casos de uso serán identificados y se harán un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del Usuario del artefacto visión y el Plan de Desarrollo marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y/o críticas del sistema). Al final de esta fase, todas las funciones de uso correspondientes a requisitos que serán implementados en la primera fase de construcción deben estar analizadas y diseñadas (en el modelo de análisis/diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también como hito. La primera y única iteración tendrá como objetivo la identificación y especificación de los principales casos de función, así como su realización preliminar en el Modelo de Análisis/ Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos.
Fase de Construcción	Durante la fase de construcción se termina de analizar y diseñar todas las funciones de uso, refinando el Modelo de

	Análisis/Diseño. El producto se construye en base a una interacción, con la simulación, en la cual se produce una visualización de datos a la cual se le aplican las pruebas y se valida con el usuario. Se comienza la elaboración de material de apoyo al usuario.
Fase de Transición	En esta fase se tiene todo listo para una distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto.

**Tabla 33 Cuadro de hitos del fin de fases del proyecto.**

#### **2.1.4.2.2 Calendario del Proyecto**

El calendario del proyecto es supervisado por el director del proyecto donde el mismo controla el avance según las fechas de presentación establecidas en el mismo.

#### **2.1.4.3 Seguimiento y Control del Proyecto**

##### **2.1.4.3.1 Gestión de Requisitos**

Esta fase tiene como objetivo la “Definición de los requisitos funcionales del sistema”, desarrollando una descripción narrativa de cada caso de uso donde cada caso de uso es visto como una caja negra considerándose solo características externas. De acuerdo al tiempo de vida de la realización del proyecto se irá analizando los siguientes requisitos.

##### **- Requisitos Organizacionales.-**

El personal debe seguir como estándar la metodología de investigación bajo un modelado con el lenguaje de programación para el software de control y el de interacción, para con el hardware (Circuito simulado).

Se debe contar con personal que conozca sobre la plataforma java, programación, manejo de la pantalla de interacción, y el hardware (Circuito simulado).

- **Requisitos de Usuarios.-**

El usuario debe tener interés de participación en el desarrollo del sistema.

Los usuarios deben contar con un mínimo de conocimientos en lectura de datos para poder realizar aportes de información.

- **Requisitos Físicos y de Funcionamiento.-**

Para el funcionamiento se debe contar con un equipo que soporte el ejecutable y que tenga instalados todos los programas necesarios para la comunicación mediante el puerto USB. El usuario debe contar con un equipo de computación capaz de poder manipular información vía conexión USB.

#### 2.1.4.3.2 Control de Plazos

Fase	Nro. de Iteraciones	Inicio
Inicio	Primera	
Elaboración	Primera	
Construcción	Primera	
Transición	Primera	

**Tabla 34 Control de Plazos**

#### 2.1.4.3.3 Control de Calidad

Nuestro sistema de control de calidad se basa en los puntos siguientes:

- Detectar problemas.
- Delimitar el área problemática.
- Estimar factores que probablemente provoquen el problema.
- Determinar si el efecto tomado como problema es verdadero o no.
- Prevenir errores debido a omisión, rapidez o descuido.
- Confirmar los efectos de mejora.
- Detectar desfases.
- Realizar pruebas en cada versión.

#### Objetivos del control de calidad

- Aumentar la satisfacción del usuario.
- Equilibrar el esfuerzo en múltiples opciones de interfaz.
- Obtener el mejor desarrollo del proyecto.

- Disfrutar de una ventaja competitiva.
- Disponer de métricas objetivas de valoración.
- Ahorrar tiempo y dinero.

## **2.1.5 Modelo de Casos de Uso**

### **2.1.5.1 Introducción**

Iniciando el proceso de análisis en el modelo de casos de uso se presentan a continuación los requisitos funcionales del sistema, representados con diagramas de casos de uso los cuales se han dividido en tres escenarios con sus correspondientes diagramas y documentos de especificaciones.

### **2.1.5.2 Propósito**

- Comprender la estructura y la dinámica del proyecto.
- Comprender problemas actuales e identificar posibles mejoras para el mismo.

### **2.1.5.3 Alcance**

- Definir un caso de uso para cada proceso.

### **2.1.5.4 Descripción Textual de cada uno de los Procesos**

- **Control del Microcontrolador:** Proceso en el que realiza el control del microcontrolador.
- **Envío\Recepción de datos:** Proceso en el que se realiza el envío\recepción de datos al microcontrolador mediante el puerto USB.

### **2.1.5.5 Descripción de Actores**

Al mismo tiempo que se determinan los procesos del negocio, es posible identificar los agentes implicados en el mismo.

Se identificó los siguientes actores:

- **Usuario**  
Persona encargada de la introducción de datos necesarios para su normal y eficiente desarrollo durante su ejecución.

- **Simulador para la conexión del Puerto USB**  
Bus que se utiliza para el envío/recepción de datos del Pc\Microcontrolador.

#### 2.1.5.6 Caso de Uso para el Controlar el Microcontrolador.

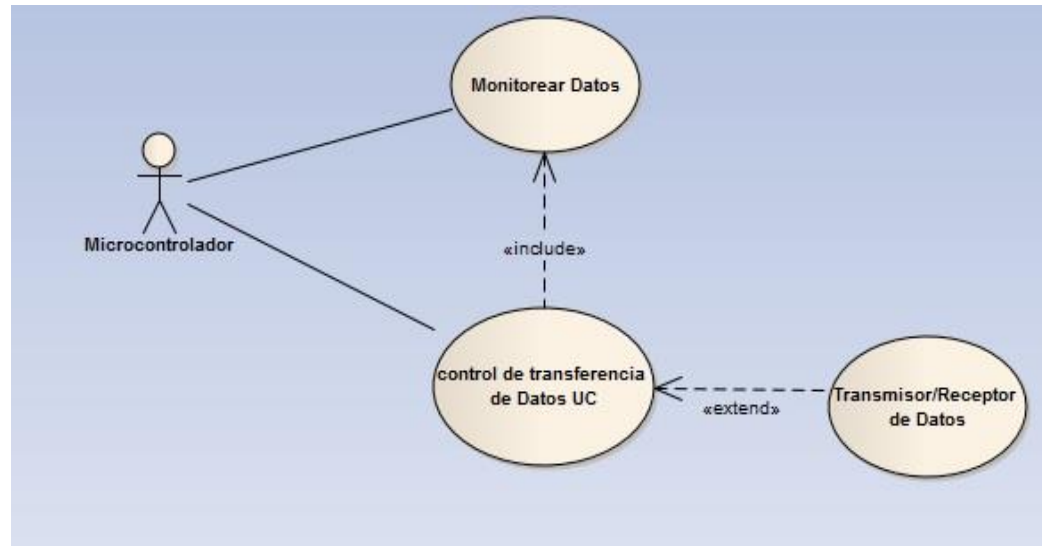


Fig.27: Caso de uso controlar el microcontrolador

<b>Caso de Uso</b>	Monitorear datos.
<b>Actores</b>	Puerto USB.
<b>Resumen</b>	Coordina la lectura de los datos, como así también el procesamiento inicial de estos valores.
<b>Dependencias</b>	Ninguna.
<b>Precondiciones</b>	El microcontrolador debe estar funcionando. El microcontrolador contiene una entrada específica para todas las entradas de datos.
<b>Descripción</b>	El sistema analiza los comandos entregados por transmisor de datos. Si alguno de los transmisores dejara de funcionar se notificara al caso de uso “Controlar el microcontrolador”.
<b>Alternativas</b>	Ninguno.
<b>Poscondiciones</b>	Ninguno.

Tabla 35 Documento del caso de uso

<b>Caso de Uso</b>	Transmisor\receptor de mensajes.
<b>Actores</b>	Transmisor de datos (puerto USB).
<b>Resumen</b>	Coordina automáticamente el envío y recepción de los mensajes del sistema tanto como para el microcontrolador y el computador.
<b>Dependencias</b>	Recibir mensajes y enviar mensajes
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- El sistema en el microcontrolador debe contener el transmisor de datos.</li> <li>- El sistema del computador debe contener el transmisor de datos.</li> </ul>
<b>Descripción</b>	<ol style="list-style-type: none"> <li>1. El sistema recibe los mensajes descomprimiéndolos.</li> <li>2. Los mensajes llegados son descompuestos y luego enviados al actor transmisor de datos.</li> <li>3. La recepción de los mensajes será según la llegada de los mismos.</li> </ol>
<b>Alternativas</b>	Ninguna.
<b>Poscondiciones</b>	<p>El envío y recepción de datos en el microcontrolador está apagado cuando el microcontrolador no se encuentra funcionando.</p> <p>El envío y recepción de datos en el computador está apagado cuando el computador no se encuentra encendido.</p>

Tabla 36 Documento del caso de uso Transmisión\Recepción de mensajes

### 2.1.5.7 Caso de Uso para el Envío y Recepción de Datos

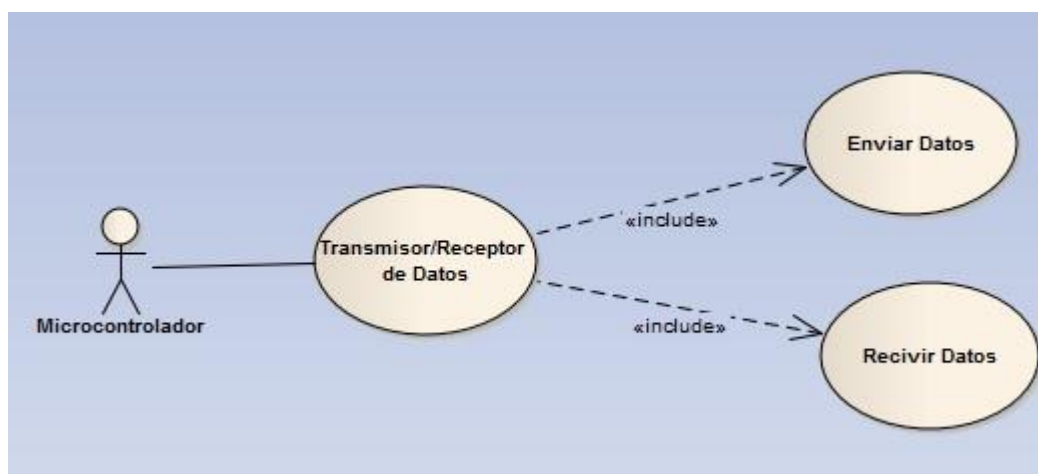


Fig.28: Caso de uso enviar recibir datos

<b>Caso de Uso</b>	Transmisor\receptor de mensajes.
<b>Actores</b>	Transmisor de datos (puerto USB).
<b>Resumen</b>	Coordina automáticamente el envío y recepción de los mensajes del sistema tanto como para el microcontrolador y el computador.
<b>Dependencias</b>	Recibir mensajes y enviar mensajes
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- El sistema en el microcontrolador debe contener el transmisor de datos.</li> <li>- El sistema del computador debe contener el transmisor de datos.</li> </ul>
<b>Descripción</b>	<p>4. El sistema recibe los mensajes descomprimiéndolos.</p> <p>5. Los mensajes llegados son descompuestos y luego enviados al actor transmisor de datos.</p> <p>6. La recepción de los mensajes será según la llegada de los mismos.</p>
<b>Alternativas</b>	Ninguna.
<b>Poscondiciones</b>	<p>El envío y recepción de datos en el microcontrolador está apagado cuando el microcontrolador no se encuentra funcionando.</p> <p>El envío y recepción de datos en el computador está apagado cuando el computador no se encuentra encendido.</p>

**Tabla 38 Documento del caso de uso Transmisión\Recepción de mensajes**

<b>Caso de Uso</b>	Enviar mensajes.
<b>Actores</b>	Ninguno.
<b>Resumen</b>	Coordina el envío de datos.
<b>Dependencias</b>	Ninguno.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- El sistema debe proporcionar una trama de mensajes preestablecida a enviarse.</li> </ul>
<b>Descripción</b>	<p>El sistema recibe el mensaje a enviarse.</p> <p>Al mensaje a ser enviado se le añade un identificador diferente entre ellos.</p>

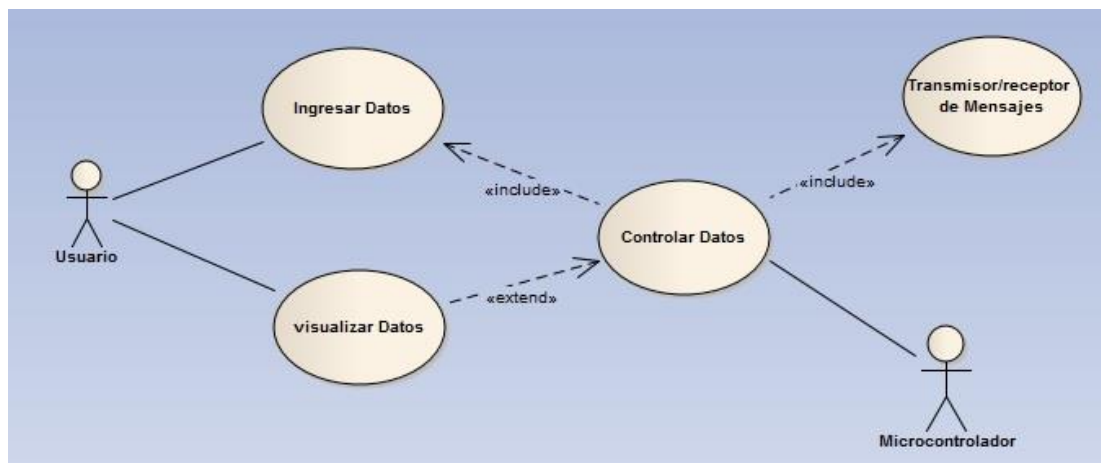
<b>Alternativas</b>	Ninguna.
<b>Poscondiciones</b>	Ninguna.

**Tabla 39 Documento del caso de uso Enviar mensajes**

<b>Caso de Uso</b>	Recibir mensajes.
<b>Actores</b>	Ninguno.
<b>Resumen</b>	Coordina la recepción de los datos.
<b>Dependencias</b>	Ninguno.
<b>Precondiciones</b>	Ninguna.
<b>Descripción</b>	<ul style="list-style-type: none"> <li>- El sistema recibe el mensaje.</li> <li>- Al mensaje recibido se le quita su identificador.</li> <li>- El mensaje recibido será publicado en el sistema para que este pueda descomponerlo.</li> </ul>
<b>Alternativas</b>	Si el mensaje capturado no contuviese el identificador la trama del sistema omitirá el mensaje.
<b>Poscondiciones</b>	Ninguna.

**Tabla 40 Documento del caso de uso Recibir mensajes**

### 2.1.5.8 Caso de Uso para Monitoreo en el Computador



**Fig.29: Caso de uso para Monitoreo en el Computador**

<b>Caso de Uso</b>	Controlar Datos
<b>Actores</b>	Microcontrolador

<b>Resumen</b>	Coordina el envío y la recepción de mensajes enviados por el transmisor de datos. Transforma los mensajes recibidos y enviados en valores o datos de los microcontroladores.
<b>Dependencias</b>	Ingresar Datos, Transmisor/receptor de datos.
<b>Precondiciones</b>	-El computador debe tener disponible la conexión disponible con el transmisor de datos, mediante el caso de uso "Transmisor/receptor de datos". -El usuario debe ingresar los datos al computador para la ejecución de procesos, para la recepción y envío de mensajes
<b>Descripción</b>	<ul style="list-style-type: none"> <li>- El sistema recibe el mensaje.</li> <li>- Al mensaje recibido se le quita su identificador.</li> <li>- El mensaje recibido será publicado en el sistema para que este pueda descomponerlo.</li> </ul>
<b>Alternativas</b>	Si el mensaje capturado no contuviese el identificador la trama del sistema omitirá el mensaje.
<b>Poscondiciones</b>	El envío de los mensajes será de forma automática a un periodo definido.

**Tabla 40 Documento del caso de uso Controlar Datos**

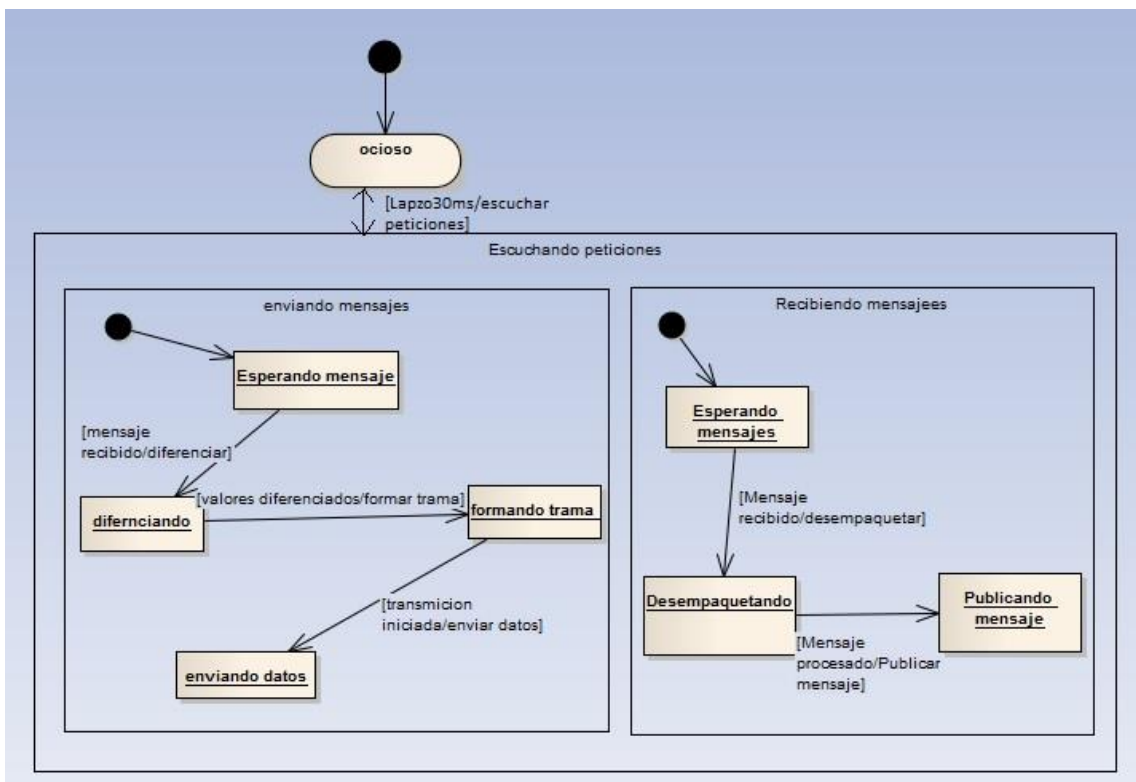
<b>Caso de Uso</b>	Visualizar Datos
<b>Actores</b>	Usuario
<b>Resumen</b>	Muestra automáticamente los mensajes enviados desde el microcontrolador.
<b>Dependencias</b>	Controlar Datos
<b>Precondiciones</b>	-El caso de uso controlar datos debe proporcionar los datos de los mensajes enviados y recibidos
<b>Descripción</b>	<ul style="list-style-type: none"> <li>- El sistema recibe el mensaje del microcontrolador</li> <li>- Al mensaje recibido se le quita su identificador.</li> <li>- El mensaje recibido será publicado en la interfaz del sistema.</li> </ul>
<b>Alternativas</b>	Ninguna.
<b>Poscondiciones</b>	Ninguna.

### 2.1.6 Diagrama de transición de Estados

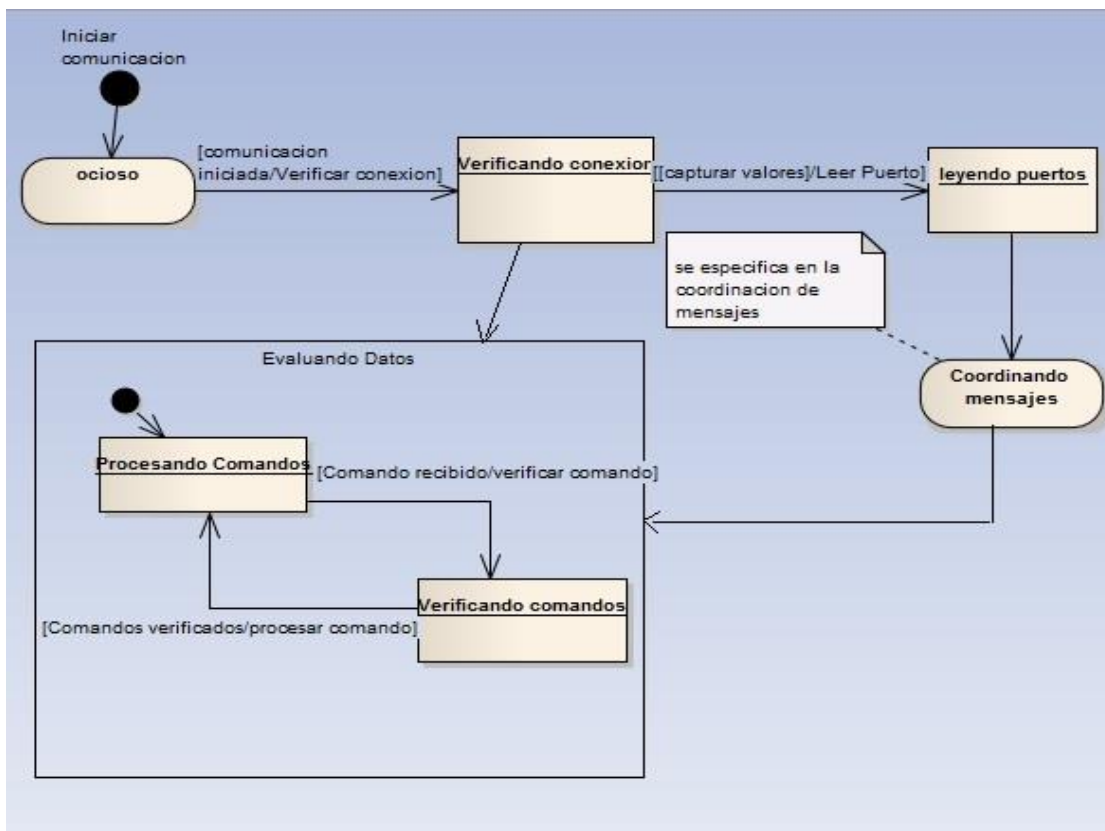
Partiendo de los diagramas de caso de uso se realizaron dos figuras de diagramas de transición de estados, para determinar así las iteraciones de los actores con el sistema.

En la **figura 30** se puede observar las iteraciones a realizarse en el microcontrolador, donde se destaca el procesado de mensajes, que son los que establecen la navegación reactiva del microcontrolador.

En la **figura 31** se detalla un estado en común a ser presentados en el microcontrolador como en el computador, nos referimos al estado de coordinación de mensajes, donde se muestra tanto que la recepción y envío de datos se las realiza de una manera concurrente. Por lo que este diagrama es válido para ambos.



**Fig.30: Diagrama de estado control Microcontrolador.**



**Fig.31: Diagrama de estado Recepción y envío de datos.**

2.1.7 Diagrama del Simulador Proteus7.8

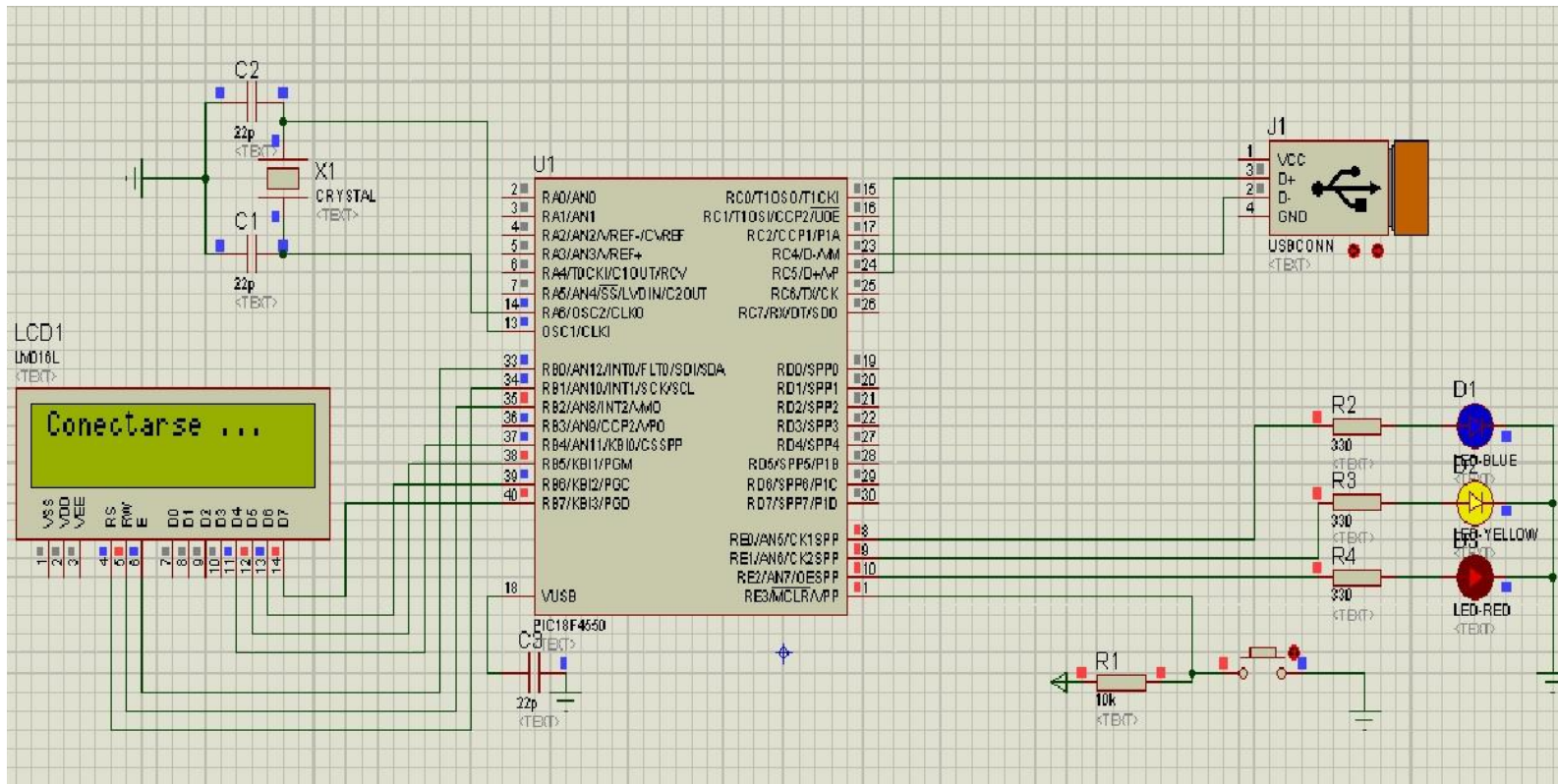


Fig. 22: Diagrama Electrónico

### 2.1.7.1 Adicionales del diagrama eléctrico

Es una característica que en los programas de modelaje eléctrico muchas veces se obvian elementos que necesariamente no se deben detallar, pero en la práctica si no los ponemos en el circuito simplemente no funciona, estos son: La alimentación de energía del Pic, en el pin VDD. La conexión a tierra del Pic, en el pin VSS Cables USB El circuito de oscilación con:

- 1 oscilador de 20 MHz
- 2 capacitores de 22 pF

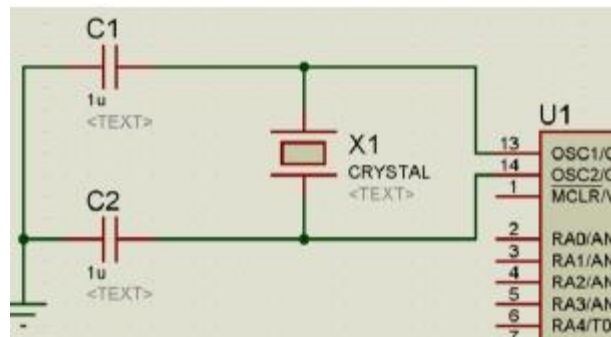


Fig. 23 Conexión Oscilador

### 2.1.7.2 Conexión de los cables USB

La conexión de los cables USB está presentada en el diagrama eléctrico, pero además es necesario construir un cable USB, que sea aplicable con terminales que calcen en nuestro circuito prueba, este lo podemos hacer con un cable USB que no se utilice con uno de sus terminales machos, y la conexión se muestra en los siguientes gráficos.

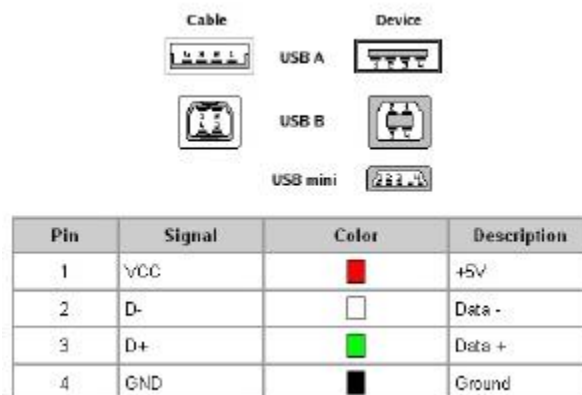
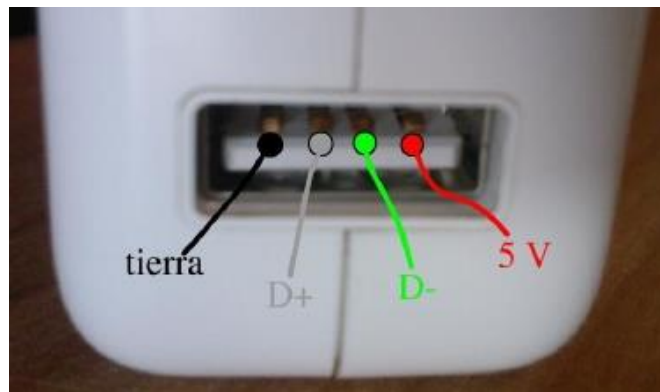


Fig. 24 Conexión de cables USB

La salida de la PC es la siguiente, recordemos que el pin de 5V no significa que debemos alimentar con 5 voltios, sino que esta línea no proporciona 5 V para nuestra aplicación, pero debemos detener a hacer el análisis de la utilidad de nuestra aplicación, y si necesita mucha corriente nos convendría utilizar una fuente adicional de alimentación.



### 2.1.8 Interfaz Usuario

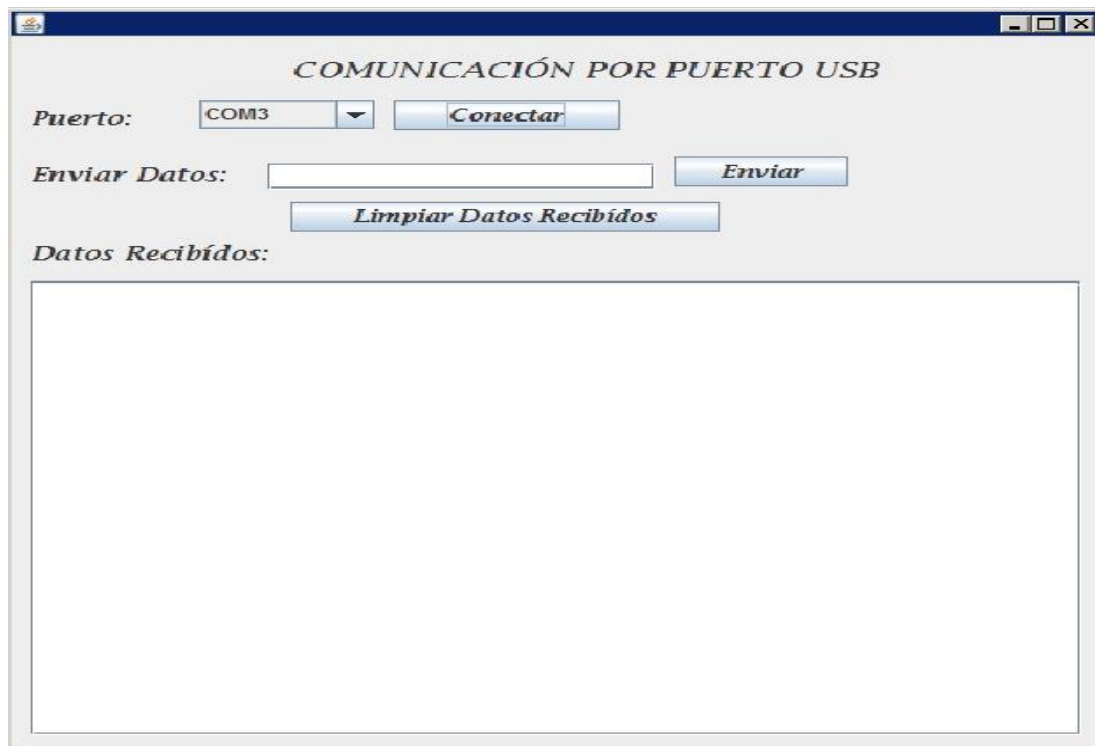


Fig. 25: Interfaz usuario

## **2.2 Capacitación a los Estudiantes de 2° Año (Grupo 1) de Informática.**

### **2.2.1 Introducción**

Los métodos y técnicas de enseñanza constituyen recursos necesarios de la enseñanza; son los vehículos de realización ordenada, metódica y adecuada de la misma. Los métodos y técnicas tienen por objeto hacer más eficiente la dirección del aprendizaje. Gracias a ellos, pueden ser elaborados los conocimientos adquiridos, las habilidades e incorporados con menor esfuerzo los ideales y actitudes que la escuela pretende proporcionar a sus alumnos.

### **2.2.2 Propósito**

Es el de proporcionar a los estudiantes de informática (2° año), una guía importante en el manejo de la aplicación y dispositivo desarrollado de tal forma que los estudiantes de la carrera de Ing. Informática puedan utilizarlo de forma eficiente, con la capacitación se busca que los estudiantes puedan:

- Manipular el sistema agregar datos para su envío al dispositivo.

### **2.2.3 Alcances y Limitaciones**

#### **2.2.3.1 Alcances**

- La capacitación se lo realiza solo a las personas de 2° año (grupo 1) de la carrera de ingeniería informática.
- Planillas de asistencia.
- Certificado de asistencia.

#### **2.2.3.2 Limitaciones**

No se capacitará a toda la carrera de ingeniería informática.

### **2.2.4 Metodología**

Para realizar la capacitación acerca del uso de sistema informático, se hizo uso de las técnicas de capacitación basada en demostraciones especializada aplicada en el sitio de estudio, esta metodología consiste en una explicación de cómo hacer algo.

Permite demostrar visualmente los resultados que se pueden obtener si experimentan con objetos. Esta técnica contempla que los estudiantes vean de forma real todos los

dispositivos utilizados y el programa en sí, la capacitación, se realizó en las instalaciones de la Universidad Autónoma Juan Misael Saracho.

La capacitación basada en demostraciones tiene varias ventajas, entre las cuales podemos hacer mención de que es relativamente económica, no hay necesidad de instalaciones costosas fuera del área de estudio como salones o dispositivos de aprendizaje programados; el método también facilita el aprendizaje de una manera ágil, donde el estudiante puede desenvolverse en nuevas áreas.

#### **2.2.4.1 Materiales utilizados**

Los materiales didácticos que se utilizaron fueron los siguientes:

- Computador Personal.
- Presentaciones en power point
- Data

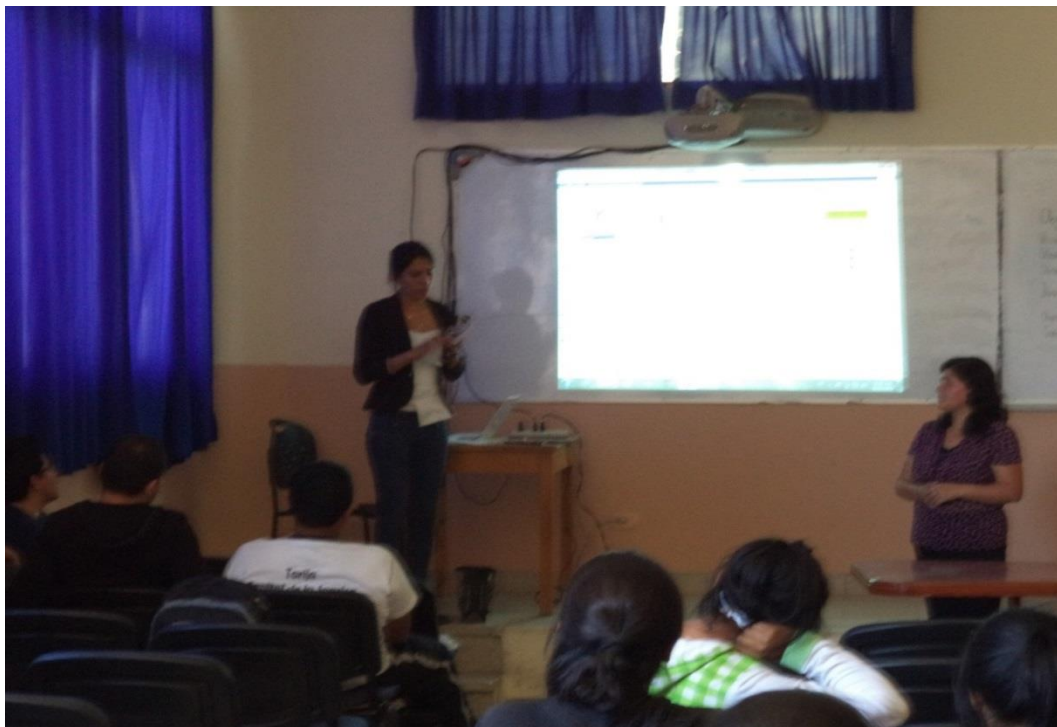
#### **Fotos de la capacitación:**



**Foto 1**



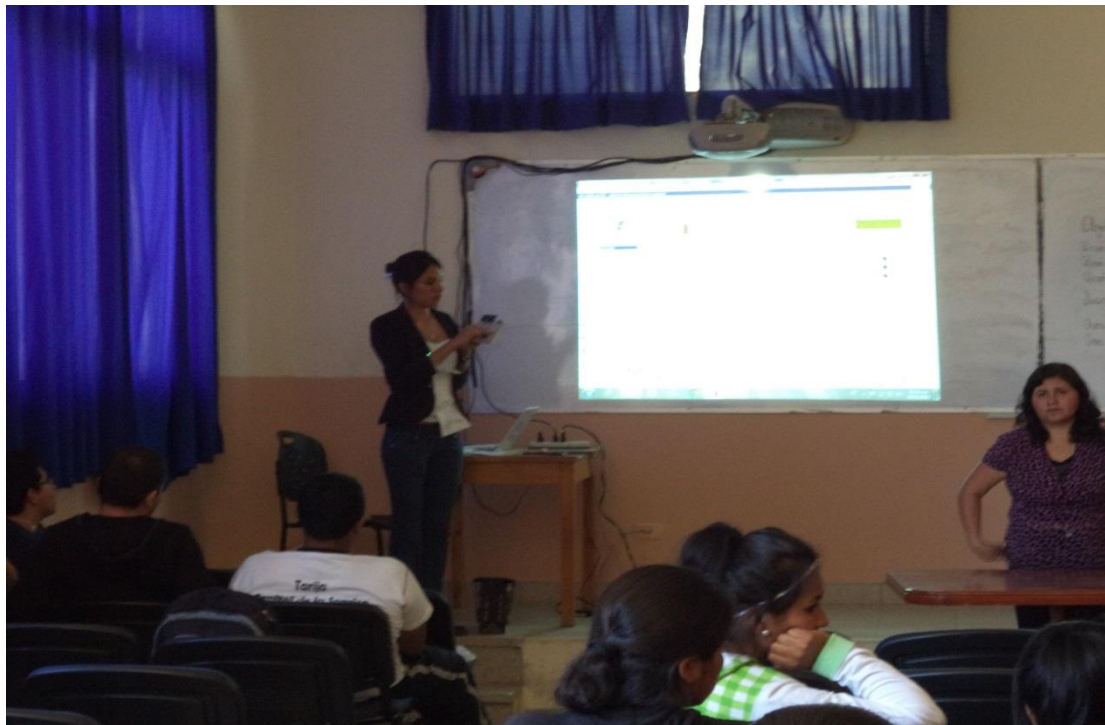
**Foto 2**



**Foto 3**



**Foto 4**



**Foto 5**

### 3 Capítulo III CONCLUSIONES Y RECOMENDACIONES

#### 3.1 Conclusiones

Se pudo realizar la comunicación entre PC-PIC con la librería dinámica **RXTXcomm**. El puerto USB es importante por su popularidad, y lo podemos utilizar en cualquier PC actual, sin preocuparnos de que tan antigua sea, por la portabilidad de su aplicación.

Con la capacitación se pudo ver la gran necesidad de medios, aun de hacer uso de los mismos, terminado el proyecto los resultados esperados fueron de carácter positivo satisfaciendo todas las expectativas de los estudiantes al momento del inicio de la capacitación.

La presente investigación, no es el fin sino el comienzo porque es por donde podemos explotar para sacar el máximo provecho a la combinación pics y Java, podemos desarrollar aplicaciones sencillas y desarrolladas como: Métodos de control industrial por interfaces java. Realizar una red de computadoras, que para comunicarse con dispositivos portables lo hagan por medio de USB. Desarrollar aplicaciones portables (hardware), fácilmente transportables a otra computadora porque el puerto USB es el más popular.

Control de robot por medio de USB, con métodos de envío de señales, recepción de respuesta para tomar decisiones de para manejar su desempeño.

Realizar aplicaciones que no necesiten de una fuente de alimentación extra.

Realizar una aplicación que pueda seguir procesos como, controlar el clima y almacenar los datos en su memoria y al intercambiarlo con la PC y realizar gráficas de representación con JfreeChart o Java Graphics.

#### 3.2 Recomendaciones

Realizar el análisis para determinar que si es necesario que la aplicación que vamos a realizar necesita una fuente de alimentación extra, o puede solventarse con la alimentación del puerto USB.

Recordar que no todos los elementos que se describen en algunos diagramas eléctricos, son todos los necesarios para el circuito físico, por ejemplo (osciladores, fuentes, entre otros).Estar seguros en el momento de alimentar el circuito, por que como sabemos los Pic tiene alta probabilidad de quedar sin funcionamiento, por sobrecargas, y un solo pin mal conectado es basta para terminar su vida útil.

#### 4 Capítulo IV GLOSARIO

**CPU:** Central Processing Unit. (Unidad de procesamiento Central). Este término se refiere específicamente al circuito integrado (contenido dentro de la caja de la computadora), que hace los “verdaderos cálculos”. Sin embargo, a veces el término es usado (aunque incorrectamente), para incluir todo lo que está dentro de la caja, incluyendo el disco rígido, la disquetera, el CD ROM, etc.

**Hub o concentrador:** es un dispositivo que permite centralizar el cableado de una red y poder ampliarla. Esto significa que dicho dispositivo recibe una señal y repite esta señal emitiéndola por sus diferentes puertos. Trabaja en capa 1 del modelo OSI o capa de Acceso en modelo TCP/IP.

**LED:** Light Emitting Diode. (Diodo Emisor de Luz). Es un tipo especial de diodo semiconductor, que cuando es conectado a un circuito electrónico con un resistor limitador de corriente, emite luz visible. Los LED usan muy poca energía y son ideales para ser conectados a dispositivos tales como el Stamp.

**Microcontrolador:** Es un circuito integrado que contiene muchas de las mismas cualidades que una computadora de escritorio, tales como la CPU, la memoria, etc., pero no incluye ningún dispositivo de “comunicación con humanos”, como monitor, teclados o mouse.

Los microcontroladores son diseñados para aplicación de control de máquinas, más que para interactuar con humanos.

**Oscilador:** Es un circuito electrónico que produce una señal electrónica repetitiva, a menudo una onda senoidal o una onda cuadrada.

Un oscilador de baja frecuencia (o LFO) es un oscilador electrónico que engendra una forma de onda de C.A. entre 0,1 Hz y 10 Hz.

**Plug&Play:** Se refiere a la capacidad de un sistema informático de configurar automáticamente los dispositivos al conectarlos. Permite poder enchufar un dispositivo y utilizarlo inmediatamente, sin preocuparte de la configuración.

**PIC:** es un microcontrolador programable, fabricado por microchip.

**PWM:** La modulación por ancho de pulso (pulse width modulation: PWM) es un tipo de control que se puede utilizar para el control de motores eléctricos de corriente continua.

**Programa:** Una secuencia de instrucciones que son ejecutadas por una computadora o un microcontrolador, en una secuencia específica, para realizar una tarea. Los programas son escritos en diferentes tipos de lenguajes, tales como “C”, Fortran o BASIC.

**Programación:** Es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación.

**Robótica:** Es una ciencia o rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano o que requieren del uso de inteligencia.

**STACK:** Es una memoria interna dedicada, de tamaño limitado, separada de las memorias de datos y de programa, inaccesible al programador, y organizada en forma de pila, que es utilizada solamente, y en forma automática, para guardar las direcciones de retorno de subrutinas e interrupciones.

**Token:** también llamado componente léxico es una cadena de caracteres que tiene un significado coherente en cierto lenguaje de programación. Ejemplos de tokens podrían ser palabras clave (if, else, while, int,...), identificadores, números, signos, o un operador de varios caracteres, (por ejemplo, :=).

**USB:** Es un puerto que sirve para conectar periféricos a una computadora.

## INDICE

TÍTULO	Pág.
1 CAPÍTULO I EL PROYECTO.....	<b>¡ERROR! MARCADOR NO DEFINIDO.</b>
1.1 PRESENTACIÓN DEL PROYECTO .....	<b>¡ERROR! MARCADOR NO DEFINIDO.</b>
1.1.1 Título .....	<b>¡Error! Marcador no definido.</b>
1.1.2 Carrera.....	<b>¡Error! Marcador no definido.</b>
1.1.3 Duración del proyecto .....	<b>¡Error! Marcador no definido.</b>
1.1.4 Área\línea de investigación priorizada.....	<b>¡Error! Marcador no definido.</b>
1.2 PERSONAL VINCULADO AL PROYECTO.....	<b>¡ERROR! MARCADOR NO DEFINIDO.</b>
1.2.1 Director(a) del proyecto .....	<b>¡Error! Marcador no definido.</b>
1.2.2 Equipo de trabajo Del Laboratorio de Robótica .....	<b>¡Error! Marcador no definido.</b>
1.2.3 Actividades previstas para los integrantes del equipo de investigación	<b>¡Error! Marcador no definido.</b>
1.3 DESCRIPCIÓN DEL PROYECTO .....	<b>¡ERROR! MARCADOR NO DEFINIDO.</b>
1.4 RESUMEN EJECUTIVO DEL PROYECTO.....	<b>¡ERROR! MARCADOR NO DEFINIDO.</b>
1.4.1 Descripción, Fundamentación y Justificación del Proyecto .....	<b>¡Error! Marcador no definido.</b>
1.4.2 Objetivos .....	<b>¡Error! Marcador no definido.</b>
1.4.2.1 Objetivo General .....	<b>¡Error! Marcador no definido.</b>
1.4.2.2 Objetivos Específicos.....	<b>¡Error! Marcador no definido.</b>
1.4.3 Metodología de Investigación.....	<b>¡Error! Marcador no definido.</b>
1.4.3.1 Lenguaje de Modelado Unificado UML.....	<b>¡Error! Marcador no definido.</b>
1.4.3.2 Compilador PIC C (PCWHD).....	<b>¡Error! Marcador no definido.</b>
1.4.3.3 Simulador Proteus7 Professional .....	<b>¡Error! Marcador no definido.</b>
1.4.3.4 Simulación del Dispositivo USB .....	<b>¡Error! Marcador no definido.</b>
1.4.3.5 Capacitación del Sistema .....	<b>¡Error! Marcador no definido.</b>
1.4.4 Resultados esperados .....	<b>¡Error! Marcador no definido.</b>
1.4.5 Transferencia de resultados.....	<b>¡Error! Marcador no definido.</b>
1.4.5.1 Medios y estrategias para la transferencia de resultados .....	<b>¡Error! Marcador no definido.</b>
1.4.5.2 Grupo de beneficiarios de los resultados. ....	<b>¡Error! Marcador no definido.</b>
1.4.6 Cronograma de actividades .....	<b>¡Error! Marcador no definido.</b>
1.4.7 Marco Lógico de Proyecto .....	<b>¡Error! Marcador no definido.</b>
1.4.7.1 Árbol de Problemas.....	<b>¡Error! Marcador no definido.</b>
1.4.7.2 Árbol de Objetivos .....	<b>¡Error! Marcador no definido.</b>
1.4.7.3 Matriz de Marco Lógico.....	<b>¡Error! Marcador no definido.</b>
1.5 PRESUPUESTO / JUSTIFICACIÓN .....	<b>¡ERROR! MARCADOR NO DEFINIDO.</b>
1.5.1 Sub total rubro.....	<b>¡Error! Marcador no definido.</b>
1.5.2 Sub total rubro.....	<b>¡Error! Marcador no definido.</b>

1.5.3	Sub total rubro.....	;	<b>Error! Marcador no definido.</b>
1.5.4	Sub total rubro.....	;	<b>Error! Marcador no definido.</b>
1.5.5	TOTAL.....	;	<b>Error! Marcador no definido.</b>
1.6	CURRICULUM VITAE .....	;	<b>ERROR! MARCADOR NO DEFINIDO.</b>
1.6.1	Antecedentes personales .....	;	<b>Error! Marcador no definido.</b>
1.6.2	Antecedentes académicos.....	;	<b>Error! Marcador no definido.</b>
1.6.3	Participación en proyectos de investigación .....	;	<b>Error! Marcador no definido.</b>
1.6.4	Publicaciones realizadas (libros, revistas, compendios y otros) ....	;	<b>Error! Marcador no definido.</b>
1.6.5	Presentaciones realizadas .....	;	<b>Error! Marcador no definido.</b>
1.6.6	Antecedentes en docencia .....	;	<b>Error! Marcador no definido.</b>
1.7	MARCO TEÓRICO .....	;	<b>ERROR! MARCADOR NO DEFINIDO.</b>
1.7.1	CAPÍTULO 1.-.....	;	<b>Error! Marcador no definido.</b>
1.7.1.1	Puerto USB. ....	;	<b>Error! Marcador no definido.</b>
1.7.2	CAPÍTULO 2.-.....	;	<b>Error! Marcador no definido.</b>
1.7.2.1	Los Microcontroladores.- .....	;	<b>Error! Marcador no definido.</b>
1.7.3	CAPÍTULO 3.-.....	;	<b>Error! Marcador no definido.</b>
1.7.3.1	Los PICs de Microchip. ....	;	<b>Error! Marcador no definido.</b>
1.7.4	CAPÍTULO 3.-.....	;	<b>Error! Marcador no definido.</b>
1.7.4.1	Código MORSE .....	;	<b>Error! Marcador no definido.</b>
2	CAPÍTULO II COMPONENTES.....		74
2.1	ENTORNO SIMULADO PARA UN CIRCUITO EN EL CUAL SE REALICE ENTRADA Y SALIDA DE DATOS POR PUERTO USB DE ARQUITECTURA ABIERTA.....		74
2.1.1	Plan de desarrollo de software y hardware .....		74
2.1.1.1	Introducción .....		74
2.1.1.2	Propósito .....		75
2.1.1.3	Alcances .....		75
2.1.1.4	Resumen.....		75
2.1.2	Vista general del proyecto.....		76
2.1.2.1	Propósito, alcance y objetivo .....		76
2.1.2.2	Suposiciones y restricciones .....		76
2.1.2.3	Entregable del proyecto.....		77
2.1.2.4	Evolución del plan de desarrollo del software .....		78
2.1.3	Organización del proyecto .....		78
2.1.3.1	Participante del proyecto.....		78
2.1.3.2	Roles y responsabilidades .....		78
2.1.4	Gestión del proyecto .....		79
2.1.4.1	Estimaciones del proyecto.....		79
2.1.4.2	Plan del proyecto.....		79
2.1.4.3	Seguimiento y Control del Proyecto .....		81
2.1.5	Modelo de Casos de Uso.....		83
2.1.5.1	Introducción .....		83
2.1.5.2	Propósito .....		83

2.1.5.3 Alcance.....	83
2.1.5.4 Descripción Textual de cada uno de los Procesos .....	83
2.1.5.5 Descripción de Actores .....	83
2.1.5.6 Caso de Uso para el Controlar el Microcontrolador. ....	84
2.1.5.7 Caso de Uso para el Envío y Recepción de Datos .....	85
2.1.5.8 Caso de Uso para Monitoreo en el Computador .....	87
2.1.6 Diagrama de transición de Estados .....	89
2.1.7 Diagrama del Simulador Proteus7.8 .....	91
2.1.7.1 Adicionales del diagrama eléctrico .....	92
2.1.7.2 Conexión de los cables USB .....	92
2.1.8 Interfaz Usuario.....	93
2.2 CAPACITACIÓN A LOS ESTUDIANTES DE 2° AÑO (GRUPO 1) DE INFORMÁTICA. 94	
2.2.1 Introducción .....	94
2.2.2 Propósito .....	94
2.2.3 Alcances y Limitaciones .....	94
2.2.3.1 Alcances .....	94
2.2.3.2 Limitaciones .....	94
2.2.4 Metodología .....	94
2.2.4.1 Materiales utilizados .....	95
.....	97
Foto 5 .....	97
3 CAPITULO III CONCLUSIONES Y RECOMENDACIONES .....	98
3.1 CONCLUSIONES .....	98
3.2 RECOMENDACIONES.....	98
4 CAPITULO IV GLOSARIO.....	99