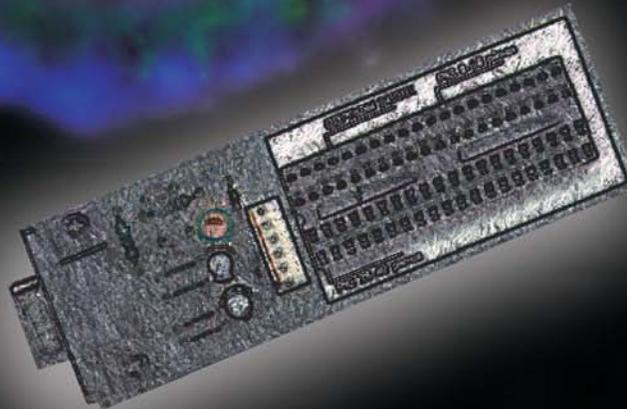
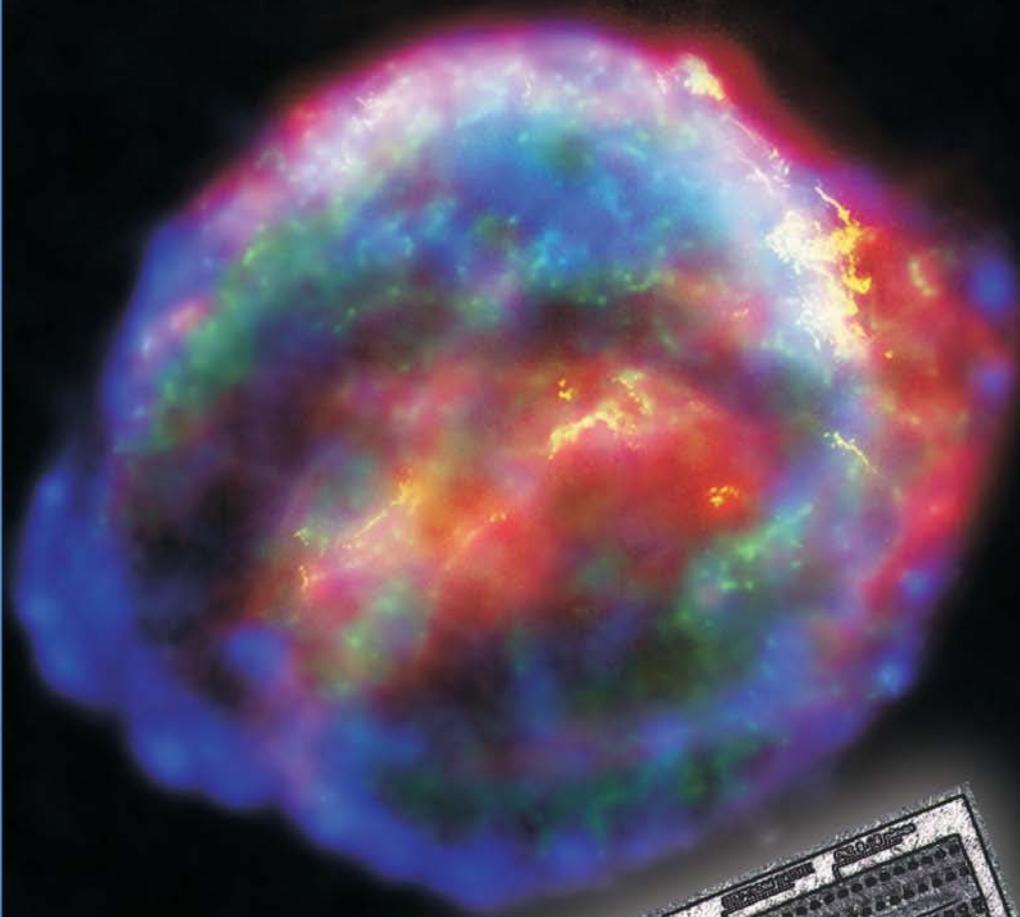




Entorno de desarrollo para programación de microcontroladores PIC



Serie: Recursos didácticos

Tapa:
Imagen combinada de la Supernova Remnant captada
por el telescopio Hubble - NASA.

a u t o r i d a d e s

PRESIDENTE DE LA NACIÓN

Dr. Néstor Kirchner

MINISTRO DE EDUCACIÓN, CIENCIA Y TECNOLOGÍA

Lic. Daniel Filmus

SECRETARIO DE EDUCACIÓN, CIENCIA Y TECNOLOGÍA

Prof. Alberto E. Sileoni

DIRECTORA EJECUTIVA DEL INSTITUTO NACIONAL DE
EDUCACIÓN TECNOLÓGICA

Lic. María Rosa Almandoz

DIRECTOR NACIONAL DEL CENTRO NACIONAL DE
EDUCACIÓN TECNOLÓGICA

Lic. Juan Manuel Kirschenbaum

Entorno de desarrollo para programación de microcontroladores PIC -circuitos integrados programables-

Jorge Eduardo Cano

Colección Serie "Recursos didácticos".
Coordinadora general: Haydeé Noceti.

Distribución de carácter gratuito.

Queda hecho el depósito que previene la ley n° 11.723. © Todos los derechos reservados por el Ministerio de Educación, Ciencia y Tecnología - Instituto Nacional de Educación Tecnológica.

La reproducción total o parcial, en forma idéntica o modificada por cualquier medio mecánico o electrónico incluyendo fotocopia, grabación o cualquier sistema de almacenamiento y recuperación de información no autorizada en forma expresa por el editor, viola derechos reservados.

Industria Argentina.

ISBN 950-00-0521-2

Cano, Jorge
Entorno de desarrollo para microcontroladores PIC / Jorge Cano; coordinado por Juan Manuel Kirschenbaum.
- 1a ed. - Buenos Aires: Ministerio de Educación, Ciencia y Tecnología de la Nación. Instituto Nacional de Educación Tecnológica, 2005.
104 p. + CD-Rom; 22x17 cm. (Recursos Didácticos; 13)

ISBN 950-00-0521-2

1. Electrónica-Circuitos. 2. Microcontroladores-Programación.
I. Kirschenbaum, Juan Manuel, coord. II. Título

CDD 621.381 5

Fecha de catalogación: 3/11/2005

Impreso en Gráfica Pinter S. A., México 1352 (C1097ABB), Buenos Aires,
en noviembre 2005

Tirada de esta edición: 3.000 ejemplares

Serie: “**Recursos didácticos**”

- 1 Invernadero automatizado
- 2 Probador de inyectores y motores paso a paso
- 3 Quemador de biomasa
- 4 Intercomunicador por fibra óptica
- 5 Transmisor de datos bidireccional por fibra óptica, entre computadoras
- 6 Planta potabilizadora
- 7 Medidor de distancia y de velocidad por ultrasonido
- 8 Estufa de laboratorio
- 9 Equipamiento EMA -Características físicas de los materiales de construcción-
- 10 Dispositivo para evaluar parámetros de líneas
- 11 Biodigestor
- 12 Entrenador en lógica programada
- 13 Entorno de desarrollo para programación de microcontroladores PIC
- 14 Relevador de las características de componentes semiconductores
- 15 Instalación sanitaria de una vivienda
- 16 Equipamiento para el análisis de estructuras de edificios
- 17 Cargador semiautomático para máquinas a CNC de accionamiento electroneumático
- 18 Biorreactor para la producción de alimentos
- 19 Ascensor
- 20 Pila de combustible

LAS METAS, LOS PROGRAMAS Y LAS LÍNEAS DE ACCIÓN DEL INSTITUTO NACIONAL DE EDUCACIÓN TECNOLÓGICA

El Instituto Nacional de Educación Tecnológica -INET- enmarca sus líneas de acción, programas y proyectos, en las metas de:

- Coordinar y promover programas nacionales y federales orientados a fortalecer la educación técnico-profesional, articulados con los distintos niveles y ciclos del sistema educativo nacional.
- Implementar estrategias y acciones de cooperación entre distintas entidades, instituciones y organismos –gubernamentales y no gubernamentales-, que permitan el consenso en torno a las políticas, los lineamientos y el desarrollo de las ofertas educativas, cuyos resultados sean considerados en el Consejo Nacional de Educación-Trabajo –CoNE-T– y en el Consejo Federal de Cultura y Educación.
- Desarrollar estrategias y acciones destinadas a vincular y a articular las áreas de educación técnico-profesional con los sectores del trabajo y la producción, a escala local, regional e interregional.
- Diseñar y ejecutar un plan de asistencia técnica a las jurisdicciones en los aspectos institucionales, pedagógicos, organizativos y de gestión, relativos a la educación téc-

nico-profesional, en el marco de los acuerdos y resoluciones establecidos por el Consejo Federal de Cultura y Educación.

- Diseñar y desarrollar un plan anual de capacitación, con modalidades presenciales, semipresenciales y a distancia, con sede en el Centro Nacional de Educación Tecnológica, y con nodos en los Centros Regionales de Educación Tecnológica y las Unidades de Cultura Tecnológica.
- Coordinar y promover programas de asistencia económica e incentivos fiscales destinados a la actualización y el desarrollo de la educación técnico-profesional; en particular, ejecutar las acciones relativas a la adjudicación y el control de la asignación del Crédito Fiscal –Ley N° 22.317–.
- Desarrollar mecanismos de cooperación internacional y acciones relativas a diferentes procesos de integración educativa; en particular, los relacionados con los países del MERCOSUR, en lo referente a la educación técnico-profesional.

Estas metas se despliegan en distintos programas y líneas de acción de responsabilidad de nuestra institución, para el periodo 2003-2007:

Programa 1. Formación técnica, media y superior no universitaria:

- 1.1. Homologación y validez nacional de títulos.
- 1.2. Registro nacional de instituciones de formación técnica.
- 1.3. Espacios de concertación.
- 1.4. Perfiles profesionales y ofertas formativas.
- 1.5. Fortalecimiento de la gestión institucional; equipamiento de talleres y laboratorios.
- 1.6. Prácticas productivas profesionalizantes: Aprender emprendiendo.

Programa 2. Crédito fiscal:

- 2.1. Difusión y asistencia técnica.
- 2.2. Aplicación del régimen.
- 2.3. Evaluación y auditoría.

Programa 3. Formación profesional para el desarrollo local:

- 3.1. Articulación con las provincias.
- 3.2. Diseño curricular e institucional.
- 3.3. Información, evaluación y certificación.

Programa 4. Educación para el trabajo y la integración social.

Programa 5. Mejoramiento de la enseñanza y del aprendizaje de la Tecnología y de la Ciencia:

- 5.1. Formación continua.
- 5.2. Desarrollo de recursos didácticos.

Programa 6. Desarrollo de sistemas de información y comunicaciones:

- 6.1. Desarrollo de sistemas y redes.
- 6.2. Interactividad de centros.

Programa 7. Secretaría ejecutiva del Consejo Nacional de Educación Trabajo –CoNE-T–.

Programa 8. Cooperación internacional.

Los materiales de capacitación que, en esta ocasión, estamos acercando a la comunidad educativa a través de la serie “Recursos didácticos”, se enmarcan en el Programa 5 del INET, focalizado en el mejoramiento de la enseñanza y del aprendizaje de la Tecnología y de la Ciencia, uno de cuyos propósitos es el de:

- Desarrollar materiales de capacitación destinados, por una parte, a la actualización de los docentes de la educación técnico-profesional, en lo que hace a conocimientos tecnológicos y científicos; y, por otra, a la integración de los recursos didácticos generados a través de ellos, en las aulas y talleres, como equipamiento de apoyo para los procesos de enseñanza y de aprendizaje en el área técnica.

Estos materiales didácticos han sido elaborados por especialistas del Centro Nacional de Educación Tecnológica del INET y por especialistas convocados a través del Programa de las Naciones Unidas para el Desarrollo –PNUD– desde su línea “Conocimientos científico-tecnológicos para el desarrollo de equipos e instrumentos”, a quienes esta Dirección expresa su profundo reconocimiento por la tarea encarada.

María Rosa Almandoz

Directora Ejecutiva del Instituto Nacional de Educación Tecnológica.
Ministerio de Educación, Ciencia y Tecnología

LAS ACCIONES DEL CENTRO NACIONAL DE EDUCACIÓN TECNOLÓGICA

Desde el Centro Nacional de Educación Tecnológica –CeNET– encaramos el diseño, el desarrollo y la implementación de proyectos innovadores para la enseñanza y el aprendizaje en educación técnico-profesional.

El CeNET, así:

- Es un ámbito de desarrollo y evaluación de metodología didáctica, y de actualización de contenidos de la tecnología y de sus sustentos científicos.
- Capacita en el uso de tecnología a docentes, profesionales, técnicos, estudiantes y otras personas de la comunidad.
- Brinda asistencia técnica a autoridades educativas jurisdiccionales y a educadores.
- Articula recursos asociativos, integrando a los actores sociales involucrados con la Educación Tecnológica.

Desde el CeNET venimos trabajando en distintas líneas de acción que convergen en el objetivo de reunir a profesores, a especialistas en Educación Tecnológica y a representantes de la industria y de la empresa, en acciones compartidas que permitan que la educación técnico-profesional se desarrolle en la escuela de un modo sistemático, enriquecedor, profundo... auténticamente formativo, tanto para los alumnos como para los docentes.

Una de nuestras líneas de acción es la de diseñar y llevar adelante un sistema de capaci-

tación continua para profesores de educación técnico-profesional, implementando trayectos de actualización. En el CeNET contamos con quince unidades de gestión de aprendizaje en las que se desarrollan cursos, talleres, pasantías, conferencias, encuentros, destinados a cada educador que desee integrarse en ellos presencialmente o a distancia.

Otra de nuestras líneas de trabajo asume la responsabilidad de generar y participar en redes que vinculan al Centro con organismos e instituciones educativas ocupados en la educación técnico-profesional, y con organismos, instituciones y empresas dedicados a la tecnología en general. Entre estas redes, se encuentra la Red Huitral, que conecta a CeNET con los Centros Regionales de Educación Tecnológica -CeRET- y con las Unidades de Cultura Tecnológica –UCT– instalados en todo el país.

También nos ocupa la tarea de producir materiales de capacitación docente. Desde CeNET hemos desarrollado distintas series de publicaciones –todas ellas disponibles en el espacio web www.inet.edu.ar–:

- *Educación Tecnológica*, que abarca materiales que posibilitan una definición curricular del área de la Tecnología en el ámbito escolar y que incluye marcos teóricos generales, de referencia, acerca del área en su conjunto y de sus contenidos, enfoques, procedimientos y estrategias didácticas más generales.

- *Desarrollo de contenidos*, nuestra segunda serie de publicaciones, que nuclea fascículos de capacitación en los que se profundiza en los campos de problemas y de contenidos de las distintas áreas del conocimiento tecnológico, y que recopila, también, experiencias de capacitación docente desarrolladas en cada una de estas áreas.
- *Educación con tecnologías*, que propicia el uso de tecnologías de la información y de la comunicación como recursos didácticos, en las clases de todas las áreas y espacios curriculares.
- *Educadores en Tecnología*, serie de publicaciones que focaliza el análisis y las propuestas en uno de los constituyentes del proceso didáctico: el profesional que enseña Tecnología, ahondando en los rasgos de su formación, de sus prácticas, de sus procesos de capacitación, de su vinculación con los lineamientos curriculares y con las políticas educativas, de interactividad con sus alumnos, y con sus propios saberes y modos de hacer.
- *Documentos de la escuela técnica*, que difunde los marcos normativos y curriculares que desde el CONET –Consejo Nacional de Educación Técnica– delinearon la educación técnica de nuestro país, entre 1959 y 1995.
- *Ciencias para la Educación Tecnológica*, que presenta contenidos científicos asociados con los distintos campos de la tecnología, los que aportan marcos conceptuales que permiten explicar y fundamentar los problemas de nuestra área.
- *Recursos didácticos*, que presenta contenidos tecnológicos y científicos,

estrategias –curriculares, didácticas y referidas a procedimientos de construcción– que permiten al profesor de la educación técnico-profesional desarrollar, con sus alumnos, un equipamiento específico para integrar en sus clases.

Desde esta última serie de materiales de capacitación, nos proponemos brindar herramientas que permitan a los docentes no sólo integrar y transferir sus saberes y capacidades, sino también, y fundamentalmente, acompañarlos en su búsqueda de soluciones creativas e innovadoras a las problemáticas con las que puedan enfrentarse en el proceso de enseñanza en el área técnica.

En todos los casos, se trata de propuestas de enseñanza basadas en la resolución de problemas, que integran ciencias básicas y tecnología, y que incluyen recursos didácticos apropiados para la educación técnico-profesional.

Los espacios de problemas tecnológicos, las consignas de trabajo, las estrategias de enseñanza, los contenidos involucrados y, finalmente, los recursos didácticos están planteados en la serie de publicaciones que aquí presentamos, como un testimonio de realidad que da cuenta de la potencialidad educativa del modelo de problematización en el campo de la enseñanza y del aprendizaje de la tecnología, que esperamos que resulte de utilidad para los profesores de la educación técnico-profesional de nuestro país.

Juan Manuel Kirschenbaum

Director Nacional del Centro Nacional de Educación Tecnológica.
Instituto Nacional de Educación Tecnológica

LA SERIE “RECURSOS DIDÁCTICOS”

Desde esta serie de publicaciones del Centro Nacional de Educación Tecnológica, nos proponemos:

- Poner a consideración de los educadores un equipamiento didáctico a integrar en los procesos de enseñanza y de aprendizaje del área técnica que coordinan.
- Contribuir a la actualización de los docentes de la educación técnico-profesional, en lo que hace a conocimientos tecnológicos y científicos.

Inicialmente, hemos previsto el desarrollo de veinte publicaciones con las que intentamos abarcar diferentes contenidos de este campo curricular vastísimo que es el de la educación técnico-profesional.

En cada una de estas publicaciones es posible reconocer una estructura didáctica común:

1 Problemas tecnológicos en el aula. En esta primera parte del material se describen situaciones de enseñanza y de aprendizaje del campo de la educación técnico-profesional centradas en la resolución de problemas tecnológicos, y se presenta una propuesta de equipamiento didáctico, pertinente como recurso para resolver esas situaciones tecnológicas y didácticas planteadas.

2 Encuadre teórico para los problemas. En vinculación con los problemas didácticos y tecnológicos que constituyen el punto de partida, se presentan conceptos

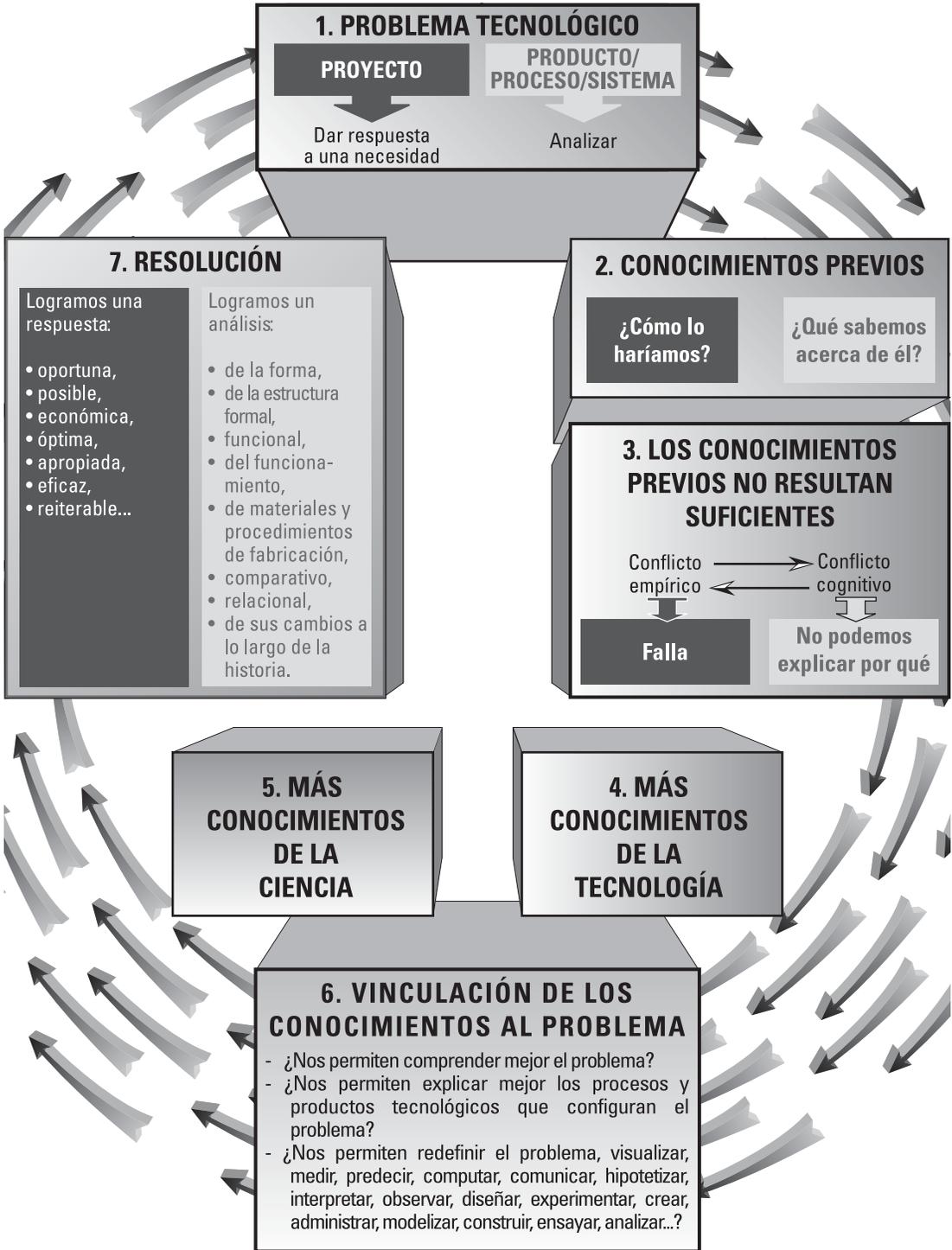
tecnológicos y conceptos científicos asociados.

3 Hacia una resolución técnica. Manual de procedimientos para la construcción y el funcionamiento del equipo. Aquí se describe el equipo terminado y se muestra su esquema de funcionamiento; se presentan todas sus partes, y los materiales, herramientas e instrumentos necesarios para su desarrollo; asimismo, se pauta el “paso a paso” de su construcción, armado, ensayo y control.

4 El equipo en el aula. En esta parte del material escrito, se retoman las situaciones problemáticas iniciales, aportando sugerencias para la inclusión del recurso didáctico construido en las tareas que docente y alumnos concretan en el aula.

5 La puesta en práctica. Este tramo de la publicación plantea la evaluación del material didáctico y de la experiencia de puesta en práctica de las estrategias didácticas sugeridas. Implica una retroalimentación –de resolución voluntaria– de los profesores destinatarios hacia el Centro Nacional de Educación Tecnológica, así como el punto de partida para el diseño de nuevos equipos.

Esta secuencia de cuestiones y de momentos didácticos no es azarosa. Intenta replicar –en una producción escrita– las mismas instancias de trabajo que los profesores de Tecnología ponemos en práctica en nuestras clases:



Es a través de este circuito de trabajo (problema-respuestas iniciales-inclusión teórica-respuestas más eficaces) como enseñamos y como aprenden nuestros alumnos en el área:

- La tarea comienza cuando el profesor presenta a sus alumnos una **situación codificada en la que es posible reconocer un problema tecnológico**; para configurar y resolver este problema, es necesario que el grupo ponga en marcha un proyecto tecnológico, y que encare análisis de productos o de procesos desarrollados por distintos grupos sociales para resolver algún problema análogo. Indudablemente, no se trata de cualquier problema sino de uno que ocasiona obstáculos cognitivos a los alumnos respecto de un aspecto del mundo artificial que el profesor –en su marco curricular de decisiones– ha definido como relevante.
- El proceso de enseñanza y de aprendizaje comienza con el planteamiento de esa situación tecnológica seleccionada por el profesor y con la construcción del espacio-problema por parte de los alumnos, y continúa con la búsqueda de **respuestas**.
- Esta detección y construcción de respuestas no se sustenta sólo en los conocimientos que el grupo dispone sino en la **integración de nuevos contenidos**.
- El enriquecimiento de los modos de “ver” y de encarar la resolución de un problema tecnológico –por la adquisición de nuevos conceptos y de nuevas formas técnicas de intervención en la situación

desencadenante– suele estar **distribuida materialmente** –en equipamiento, en materiales, en herramientas–.

No es lo mismo contar con este equipamiento que prescindir de él.

Por esto, lo que intentamos desde nuestra serie de publicaciones es acercar al profesor distintos recursos didácticos que ayuden a sus alumnos en esta tarea de problematización y de intervención –sustentada teórica y técnicamente– en el mundo tecnológico.

Caracterizamos como **recurso didáctico** a todo material o componente informático seleccionado por un educador, quien ha evaluado en aqué posibilidades ciertas para actuar como mediador entre un problema de la realidad, un contenido a enseñar y un grupo de alumnos, facilitando procesos de comprensión, análisis, profundización, integración, síntesis, transferencia, producción o evaluación.

Al seleccionar los recursos didácticos que forman parte de nuestra serie de publicaciones, hemos considerado, en primer término, su potencialidad para posibilitar, a los alumnos de la educación técnico-profesional, configurar y resolver distintos problemas tecnológicos.

Y, en segundo término, nos preocupó que cumplieran con determinados rasgos que les permitieran constituirse en medios eficaces del conocimiento y en buenos estructurantes cognitivos, al ser incluidos en un aula por un profesor que los ha evaluado como perti-

nentes. Las cualidades que consideramos fundamentales en cada equipo que promovemos desde nuestra serie de publicaciones "Recursos didácticos", son:

- Modularidad (puede adaptarse a diversos usos).
- Resistencia (puede ser utilizado por los alumnos, sin peligro de romperse con facilidad).
- Seguridad y durabilidad (integrado por materiales no tóxicos ni peligrosos, y durables).
- Adaptabilidad (puede ser utilizado en el taller, aula o laboratorio).
- Acoplabilidad (puede ser unido o combinado con otros recursos didácticos).
- Compatibilidad (todos los componentes, bloques y sistemas permiten ser integrados entre sí).
- Facilidad de armado y desarmado (posibilita pruebas, correcciones e incorporación de nuevas funciones).
- Pertinencia (los componentes, bloques funcionales y sistemas son adecuados para el trabajo con los contenidos curriculares de la educación técnico-profesional).
- Fiabilidad (se pueden realizar las tareas preestablecidas, de la manera esperada).
- Coherencia (en todos los componentes, bloques funcionales o sistemas se siguen las mismas normas y criterios para el armado y utilización).
- Escalabilidad (es posible utilizarlo en proyectos de diferente nivel de com-

plejidad).

- Reutilización (los diversos componentes, bloques o sistemas pueden ser desmontados para volver al estado original).
- Incrementabilidad (posibilidad de ir agregando piezas o completando el equipo en forma progresiva).

Haydeé Noceti

Coordinadora de la acción "Conocimientos científico-tecnológicos para el desarrollo de equipos e instrumentos".
Centro Nacional de Educación Tecnológica



13. Entorno de desarrollo para programación de microcontroladores PIC -circuitos integrados programables-

Este material de capacitación fue desarrollado por:

Jorge Eduardo Cano

Coordinación general:

Haydeé Noceti

Diseño didáctico:

Ana Rúa

Administración:

Adriana Perrone

Monitoreo y evaluación:

Laura Irurzun

Diseño gráfico:

Tomás Ahumada

Karina Lacava

Alejandro Carlos Mertel

Diseño de tapa:

Laura Lopresti

Juan Manuel Kirschenbaum

Con la colaboración
del equipo de profesionales
del Centro Nacional
de Educación Tecnológica



Índice

Las metas, los programas y las líneas de acción del Instituto Nacional de Educación Tecnológica	IV
Las acciones del Centro Nacional de Educación Tecnológica	VI
La serie “Recursos didácticos”	VII

1 Problemas tecnológicos en el aula 4

- El recurso didáctico que proponemos

2 Encuadre teórico para los problemas 11

- ¿Qué es un microcontrolador?
- Microcontroladores PIC
 - Procesador o CPU
 - Memoria para el programa tipo ROM
 - Memoria RAM para contener los datos
 - Líneas de E/S para comunicarse con el exterior
- Utilizando un microcontrolador
- El microcontrolador PIC 16F84
 - Estructura del PIC 16F84
 - Los puertos del PIC 16F84

3 El equipo en el aula 22

- Baliza electrónica
- Baliza electrónica programable
- Contador digital

4 La puesta en práctica 64

Anexo: Disco compacto conteniendo el entorno de desarrollo para programación de microcontroladores PIC

1. PROBLEMAS TECNOLÓGICOS EN EL AULA

En la educación técnico-profesional, diferentes espacios curriculares se ocupan de formar alumnos con competencia en electrónica analógica y digital, en instrumentos y dispositivos de control, y en el diseño de productos, procesos y métodos de control.

A los profesores a cargo de estos espacios les resulta necesario contar con un recurso didáctico que permita a sus alumnos analizar situaciones, elaborar propuestas de

solución a problemas tecnológicos, diseñar, construir, analizar, y modelizar equipos y sistemas a partir del control de variables, mediante la programación de microcontroladores.

Consideremos por ejemplo, estas situaciones problemáticas que distintos profesores presentan a sus alumnos y en las que resultaría oportuno integrar el recurso didáctico que vamos a proponerle:

Accidentes de tránsito

En la ruta nacional 14 ocurren accidentes automovilísticos, recurrentemente.

La *Secretaría de Medios de Transporte* ha analizado detenidamente las situaciones que provocan estos accidentes y ha llegado a la conclusión de que las tres causas más frecuentes están relacionadas con:

- Imprudencia de los conductores al sobrepasar a otros vehículos en la ruta (causas humanas).
- Pérdida de *grip* -apretón- de los neumáticos en días muy húmedos o lluviosos, hecho que se ve agravado por las características de la capa asfáltica del corredor, que deriva en que los conductores pierdan el control sobre el vehículo (causas estructurales).
- Ausente o deficiente señalización cuando los automovilistas se detienen en la banquina por algún desperfecto. Se comprueba que, en los días de viento o lluviosos, las balizas reglamentarias resultan inadecuadas porque, cuando los vehículos que circulan pasan a su lado, éstas salen despedidas por efecto de la succión que producen (causas fortuitas).

Entre las acciones llevadas a cabo por la Secretaría para reducir los accidentes de tránsito en la ruta está la de convocarnos para que imaginemos y desarrollemos una solución tecnológica que tenga en cuenta las siguientes condiciones:

- Bajo costo (para que los automovilistas puedan implementarla).
- Confiable (debe funcionar en condiciones desfavorables: viento, lluvia, etc.).
- Fácil accionamiento (cualquier persona tiene que poder ponerla en funcionamiento sin dificultad).
- Porte pequeño (para que no represente una carga excesiva en el baúl).
- Bajo consumo energético.

Básquet en apuros

El presidente de una comisión vecinal de la ciudad de San Rafael se ha acercado al Centro de Formación Profesional para plantear un problema que se les ha presentado a raíz del acondicionamiento de las instalaciones de la cancha de básquet del club del barrio, para que se puedan jugar partidos con los otros clubes barriales.

Como necesitan cronometrar el tiempo del partido y los tableros oficiales son muy costosos, solicitan al instructor y a los aprendices del Centro la construcción de un reloj de juego de bajo costo.

Este reloj debe cumplir las características de:

- Contar 8 o 10 minutos de juego en cuenta regresiva (minutos y segundos).
- Tener un dispositivo que detenga y reanude la cuenta del reloj en los cortes del juego (tiempos muertos, sanción de faltas, etc.)
- Encender una luz detrás de los tableros de juego y, simultáneamente, activar una bocina cuando el tiempo expire.

Playa de estacionamiento inteligente

En la playa de estacionamiento *Auto-matic* trabaja una sola persona por turno.

El dueño nos ha solicitado que elaboremos un sistema que permita un control estricto de los automóviles que entran y salen de la playa.

Un *display* electrónico debe registrar la cantidad exacta de automóviles en la playa (por lo tanto, debe ir sumando los ingresos y restando los egresos en forma automática), para que el encargado vea, en la pantalla de su computadora, el detalle de la cantidad de automóviles que ingresaron, la cantidad que egresó y cuántos automóviles están en la playa.

Agua que quieras beber... primero tendrás que subir al tanque

Cuando llega el período estival, en la zona del puerto de Gualeguaychú baja tanto la presión del agua corriente que el viejo edificio de departamentos *Aguadas* se queda sin agua.

Sus propietarios han decidido ponerle fin a este problema; construyeron un tanque cisterna en el sótano del edificio; una bomba aspirante enviará agua hacia los tanques ubicados en la terraza.

El consorcio del *Aguadas* nos solicita que diseñemos un dispositivo que controle todo el sistema y que cumpla con las siguientes condiciones de funcionamiento:

- Cuando el tanque superior haya vaciado dos tercios de su capacidad, la bomba debe encenderse; debe detenerse cuando esté lleno.
- Si no hubiera agua en el tanque cisterna, la bomba debe apagarse (o no encenderse); porque, si la hiciéramos funcionar en vacío, se quemaría.

¡Basta de flores marchitas!

El señor García y su esposa tienen un vivero en Escobar. Ellos hacen todo el trabajo: siembran los almácigos, transplantan los plantines, preparan diferentes tipos de tierra, podan las plantas, atienden al público, mantienen el predio arreglado, riegan las plantas de los viveros cerrados y las que están a la intemperie, etc.

Como quieren garantizar que las plantas de su vivero estén en el medio correcto, nos han solicitado que diseñemos un sistema automático que encienda los aspersores, cuando la humedad de las macetas no sea la adecuada, y que los apague después de un determinado periodo de tiempo.

En cada uno de los casos, las tareas básicas a realizar por los grupos de estudiantes son:

- Analizar la información y las condiciones que debe cumplir el dispositivo.
- Analizar sus condiciones de seguridad y de confiabilidad.
- Identificar dos o más alternativas para solucionar el problema.
- Determinar energías necesarias para que funcionen las diferentes propuestas elaboradas.
- Optar por una de ellas, justificar la elección y armar el prototipo.
- Realizar un croquis del modelo.
- Construir el sistema.
- Ensayar su funcionamiento.
- Elaborar un informe técnico.

El recurso didáctico que proponemos

La industria moderna utiliza sistemas de automatización que garantizan condiciones óptimas de trabajo y que logran productos que cumplen estándares de calidad con la mayor seguridad para el operario y el medio ambiente, con la más alta rentabilidad y el menor costo de producción.

Los sistemas de automatización más generalizados hoy en día son controlados por sistemas eléctricos o electrónicos, desde sencillos microrruptores mecánicos hasta comple-

jos controladores lógicos programables -PLC-.

La escuela técnico-profesional no desconoce que esta realidad le exige, cada vez más, actualizar los contenidos y la metodología de enseñanza, si no quiere quedar a la zaga del desarrollo actual. Para ello, enfrenta el reto y acerca a sus alumnos al conocimiento de nuevas tecnologías, alentando y favoreciendo el trabajo en equipos como el camino para la construcción del aprendizaje.

El recurso didáctico que le proponemos, nuestro **Entorno de desarrollo para programación de microcontroladores PIC -circuitos integrados programables-¹**, permite encarar esta tarea a través del diseño de un diagrama de flujo, de manera gráfica y con alto nivel de programación, sin necesidad de escribir con lenguaje *assembler* -ensamblador- o escribiendo la menor cantidad de código, constituyéndose así

en una herramienta rápida, sencilla y productiva para desarrollar proyectos con microcontroladores PIC.

En este programador visual, un simple diagrama de flujo va configurando el programa, ya que el software traduce al lenguaje *assembler* cada uno de los bloques estructurados.



¹ El nombre comercial de este equipo es *Niple*®, registrado por Jorge Eduardo Cano y Silvio Ernesto Galán en la Dirección Nacional de Derechos de Autor, número de expediente 244692. www.niplesoft.net.
En este material de capacitación nos referimos sólo a algunas de sus funciones.

ENTRE LAS
PRINCIPALES
PRESTACIONES
QUE EL
SOFTWARE
OFRECE,
PODEMOS
DESTACAR:

Entorno visual de alto nivel de programación que plantea un procedimiento totalmente visual y en lenguaje humano

- Incluye bloques estandarizados que facilitan la interpretación del proyecto.
- Controla errores lógicos, automáticamente.
- Supervisa el ingreso de datos.

Control automático de registros y bits

- Detecta la existencia de todos los registros y bits necesarios para el correcto funcionamiento del proyecto, en el momento de generar el código *assembler*.
- Controla que no existan nombres de registros y bits duplicados.
- Valida los nombres de los registros y los bits en el momento de ser declarados.

Corrección automática de errores lógicos

- Controla la correcta configuración de los puertos.
- Controla la configuración y activación de las interrupciones.
- Supervisa la creación de vínculos entre bloques.

Generación del código *assembler*

- Controla la sintaxis del código generado.
- Controla el cambio de página.
- Genera etiquetas.

LAS FUNCIONES DISPONIBLES DEL EQUIPO QUE PROPONEMOS SON:

- Asignación de valores a registros y bits.
- Configuración y manejo de interrupciones.
- Manejo de rutinas de usuario.
- Lectura / escritura de puertos.
- Lectura / escritura de EEPROM.
- Funciones de comparación de registros y bits (condiciones =, >, <, >=, <=, <>).
- Manejo automático de tablas.
- Temporizadores por bucle.
- Visualización de datos en dígitos 7 segmentos (ánodo común y cátodo común)
- Ingreso de datos por teclado matricial (3 x 3 y 3 x 4).
- Configuración y manejo de temporizador interno TMR0.
- Comunicaciones RS232.
- Cálculos matemáticos a 8 bits: (+, -, /, *), conversión a BCD.
- Escalado de valores de un rango a otro (a 8 y 16 bits).
- Manejo de registros de desplazamiento

COMO HERRAMIENTAS ADICIONALES:

- Interfaz de comunicación RS232.
- Comunicaciones a 8 o 16 bits con visualización en distintos formatos (decimal, hexadecimal, binario, texto, o modo gráfico).

La inclusión del **entorno de desarrollo para programación de microcontroladores PIC-circuitos integrados programables**- permite dar respuestas a situaciones problemáticas de la educación técnico-profesional -como las que le hemos presentado y muchas otras- que, seguramente, podrían ser solucionadas utilizando la electrónica tradicional pero que, ahora, cobran otra dimensión a través de este *kit* versátil y combinado con un dispositivo confiable de estructura de montaje pequeña.

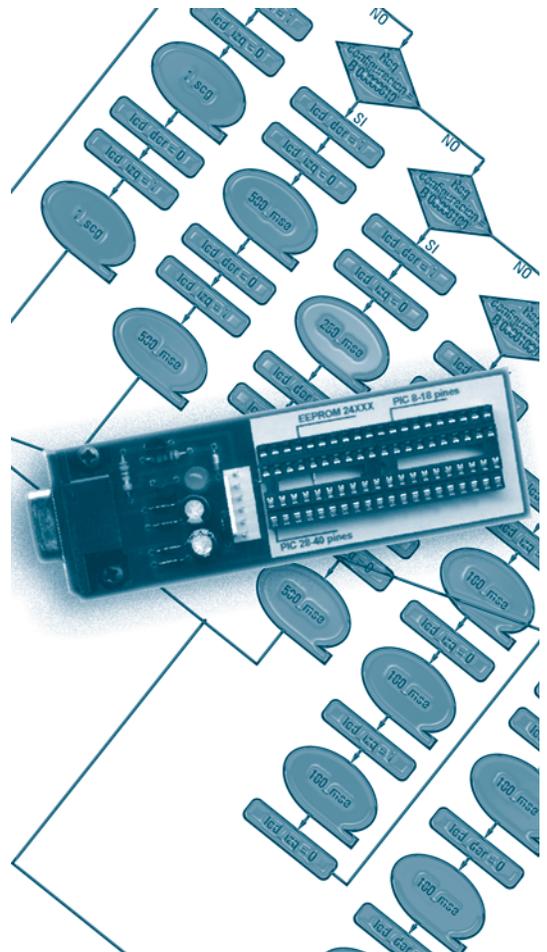
Este recurso didáctico hace posible, así, que los laboratorios tecnológicos cuenten con un equipo que facilita los aprendizajes de la electrónica y su aplicación práctica.

Porque, sabido es que el aprendizaje del lenguaje *assembler* que hace posible la programación de los microcontroladores, presenta muchos escollos a superar: requiere el acceso a diferentes manuales, resulta complejo de aprender para los alumnos -ya que exige complejos procesos de abstracción con que no siempre cuentan estudiantes de los primeros años- y requiere la escritura de muchas líneas de programa, incluso para diseños muy simples.

El programador visual intenta constituirse en una herramienta de trabajo útil para profesores y alumnos, en la tarea de programar microcontroladores PIC 16F84, uno de los modelos más populares en el mercado por sus características estructurales y funcionales que, como es reescribible, puede ser reusado sin dificultad, para:

- concretar proyectos tecnológicos exigentes, haciendo posible que los alumnos trabajen con autonomía.

- integrar contenidos teóricos de electrónica,
- resolver problemas, diseñar y construir circuitos, e identificar funciones básicas,
- diseñar técnicas de control, incorporando una visión comparativa de los procesos,
- desarrollar proyectos de automatización que apunten a mejoras en la eficiencia de procesos productivos.



2. ENCUADRE TEÓRICO PARA LOS PROBLEMAS

¿Qué es un microcontrolador?

Los microcontroladores hacen su aparición a principios de los '80; se trata de circuitos integrados programables que contienen toda la estructura (arquitectura) de una microcomputadora.

Dentro de un microcontrolador, podemos encontrar:

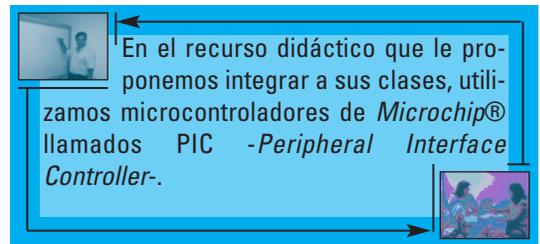
- CPU -*Central Processing Unit*; unidad central de proceso- o procesador.
- Memoria RAM para contener los datos.
- Memoria ROM.
- Memoria EEPROM -memoria de lectura y escritura no volátil; los datos no se pierden cuando el circuito es desconectado-.
- Puertos de entrada y de salida -pines de E/S-.
- Distintos módulos periféricos: convertidores analógico/digital -A/D-, módulos PWM -control por ancho de pulso-, módulos de comunicaciones seriales o en paralelo, y más.

Por supuesto, todo contenido dentro del mismo circuito integrado.

Cada vez existen más productos que incorporan microcontroladores con el fin de

aumentar sustancialmente sus prestaciones, reducir su tamaño y costo, mejorar su confiabilidad y disminuir el consumo de energía.

Existen varios fabricantes de microcontroladores, entre los cuales podemos nombrar a las empresas *Motorola*, *Microchip*, *Atmel*, *Parallax*, etc.



En el recurso didáctico que le proponemos integrar a sus clases, utilizamos microcontroladores de *Microchip®* llamados PIC -*Peripheral Interface Controller*-.

The box contains a light blue background with a white border. On the left, there is a small inset image of a person standing in a classroom. On the right, there is another small inset image showing a group of people working together. The text is centered in the middle of the box.

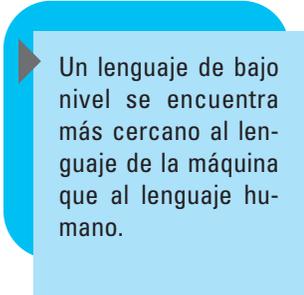
Microcontroladores PIC

Existe una gran cantidad de modelos de microcontroladores cuyas características y prestaciones varían de un modelo a otro. De esta manera, los desarrolladores pueden seleccionar el modelo que mejor se ajuste a sus necesidades.

Los distintos modelos de microcontroladores se agrupan por familias. Una familia está formada por un conjunto de modelos de microcontroladores cuyas caracterís-

ticas y prestaciones son bastante similares.

Cuando compramos un microcontrolador, su memoria se encuentra "en blanco": El microcontrolador no sabe hacer nada. Para que un microcontrolador funcione, es necesario que sea programado, lo que se realiza mediante un lenguaje de programación llamado *Assembler* cuya principal característica es su alta complejidad, ya que se trata de un lenguaje de bajo nivel.



Un lenguaje de bajo nivel se encuentra más cercano al lenguaje de la máquina que al lenguaje humano.

Este rasgo implica que sólo las personas altamente capacitadas están en condiciones de realizar desarrollos electrónicos que incluyan microcontroladores e, incluso a estos especialistas, les requiere un gran esfuerzo intelectual y mucho tiempo de desarrollo.

Vamos a ir refiriéndonos a:

- Procesador o CPU.
- Memoria para el programa tipo ROM.
- Memoria RAM para contener los datos.
- Líneas de E/S para comunicarse con el exterior.
- Módulos para el control de periféricos (temporizadores, puertos serie y paralelo, CAD -convertidores analógico/digital-, CDA -convertidores digital/análogo, etc.).

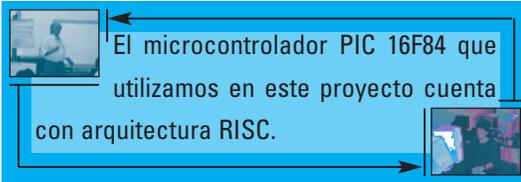
Procesador o CPU

Es el elemento más importante del microcontrolador y el que determina sus principales características, tanto de hardware como de software.

La CPU se encarga de la decodificación y de la ejecución del programa.

Actualmente, existen tres tipos de arquitectura de procesadores:

- **CISC** -computadoras de juego de instrucciones complejo-. Disponen de más de 80 instrucciones en su repertorio, algunas de las cuales son muy sofisticadas y potentes, y que requieren muchos ciclos para su ejecución. Una ventaja de los procesadores CISC es que ofrecen instrucciones complejas que actúan como macros.
- **RISC** -computadoras de juego de instrucciones reducido-. En estos procesadores, el repertorio de instrucciones es muy reducido; las instrucciones son simples y, generalmente, se ejecutan en un ciclo. La ventaja de éstos es que la sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.
- **SISC** -computadoras de juego de instrucciones específico-. En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es específico -las instrucciones se adaptan a las necesidades de la aplicación prevista-.



Memoria ROM

La memoria ROM es una memoria no volátil -no se pierden los datos al desconectar el equipo- que se destina a contener el programa de instrucciones que gobierna la aplicación. Los microcontroladores disponen de capacidades de ROM comprendidas entre 512 bytes y 8 kbytes.

Existen distintos tipos de memorias ROM, que determinan la aplicación del microcontrolador:

- **ROM con máscara.** Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado costo del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.
- **OTP -One Time Programmable-.** El microcontrolador contiene una memoria no volátil de sólo lectura "programable una sola vez" por el usuario; es éste quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde una PC. La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto; o bien, en la construcción de prototipos y series muy pequeñas. Tanto en este tipo de memoria como en
- **EPROM -Erasable Programmable Read Only Memory-.** Los microcontroladores que disponen de memoria EPROM pueden borrarse y grabarse muchas veces. La grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde una PC. Si, posteriormente, se desea borrar el contenido, se dispone de una ventana de cristal en su superficie, por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico, por lo que el producto es más caro que un microcontrolador con memoria OTP -que está hecho con material plástico-.
- **EEPROM -Electrical Erasable Programmable Read Only Memory-.** Se trata de memorias de sólo lectura que se pueden escribir y borrar eléctricamente. Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de una PC, por lo que resultan operaciones cómodas y rápidas. No disponen de ventana de cristal en la superficie. Los microcontroladores dotados de memoria EEPROM, una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera, sin ser retirados de dicho circuito; para esto se usan grabadores en circuito que confieren una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo. El número de veces que puede grabarse y borrarse una memoria EEPROM es finito, por lo que no es

recomendable una reprogramación continua. Son muy idóneos para la enseñanza y la ingeniería de diseño. La inclusión de una pequeña zona de memoria EEPROM en los circuitos programables para guardar y modificar cómodamente una serie de parámetros que adecuan el dispositivo a las condiciones del entorno, se va extendiendo en los fabricantes. Esta memoria es relativamente lenta.

- **Flash.** Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM; pero, consume menos y es más pequeña. A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM. La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura y borrado.

Memoria RAM

La memoria RAM es una memoria volátil: Los datos se pierden al desconectar el equipo. Se destina a guardar las variables y los datos.

Los microcontroladores disponen de capacidades de RAM comprendidas entre 20 y 512 bytes.

Registros y bits

La memoria está dividida en pequeñas

"partes", los registros. Un registro es una posición de memoria en la que se puede almacenar un dato.

Dentro de la memoria, cada registro es identificado mediante un número, llamado dirección de memoria, generalmente expresado en formato hexadecimal. El primer registro de una memoria corresponde a la dirección 00H.

Dado que identificar a cada registro mediante un número hexadecimal resulta muy complejo para el programador, existe la posibilidad de asignar un nombre para cada registro. En general, este nombre está directamente relacionado con la función que el registro cumple dentro del sistema.

Los registros no sólo son utilizados por el programador (usuario) para almacenar los datos que la aplicación debe procesar sino que, además, sirven para controlar todo el funcionamiento del microcontrolador en su conjunto. Esta función es cumplida por un conjunto de registros predefinidos desde la fábrica: el fabricante asigna las funciones de configuración y control del microcontrolador en un grupo de registros, y el usuario no puede modificar la función que cumple cada uno.

Cada registro está dividido en 8 "casilleros", los bits. Entonces, podemos decir que un registro está formado por un conjunto de 8 bits.

La palabra **bit** proviene de la abreviatura de *binary digit*.

El bit es la menor unidad de información

que un sistema digital puede procesar y sólo puede contener los valores lógicos 0 y 1.

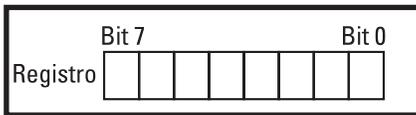
Los sistemas digitales representan la información en forma de bits, porque sus circuitos sólo pueden tener dos estados: encendido o apagado.

En general, podemos decir que:

1 = Encendido = Verdadero = Sí = +5 V

0 = Apagado = Falso = No = 0 V

Cada bit se identifica por la posición que ocupa dentro del registro, siendo el primer bit el número 0 -el que se encuentra en el extremo derecho del registro-.



Al igual que los registros, se puede asignar un nombre a cada bit para facilitar su identificación.

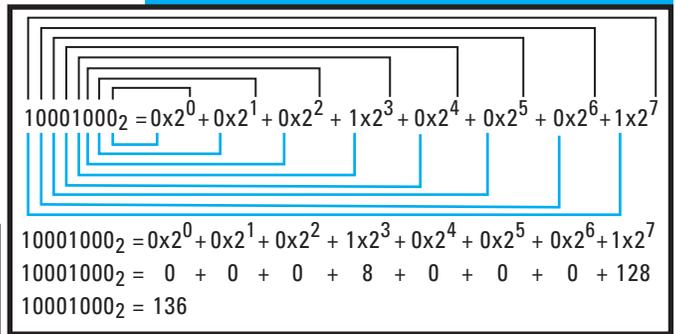
En un registro, podemos almacenar una combinación de 8 ceros y unos, lo que nos da una cantidad de 28 combinaciones; es decir, 256 posibles combinaciones de ceros y unos.

Esto significa que un registro puede procesar valores entre 0 y 255.

Esta característica de procesar números, dígitos, es lo que da el nombre a la **electrónica digital**. La electrónica digital procesa números formados por combinaciones de ceros y unos. Toda señal analógica debe ser digitalizada -es decir, convertida a números-, para poder ser procesada en un sistema digital.

Podemos decir que el conjunto de ceros y unos almacenado en un registro es un número entre 0 y 255, expresado en sistema binario -esto es, en base 2-. Para interpretar este número, debemos realizar el cálculo de base y expresarlo en sistema decimal.

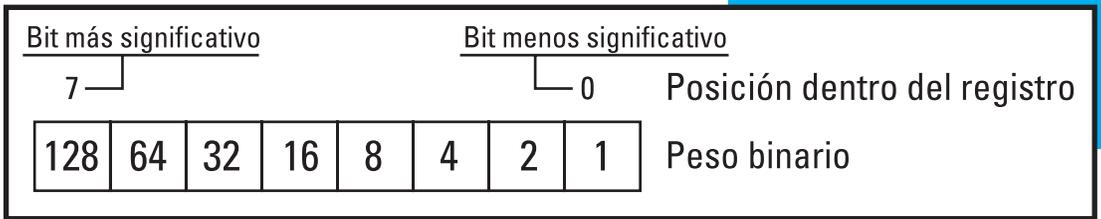
El siguiente ejemplo muestra el desarrollo de un cálculo de conversión de sistema binario (base 2) a sistema decimal (base 10):



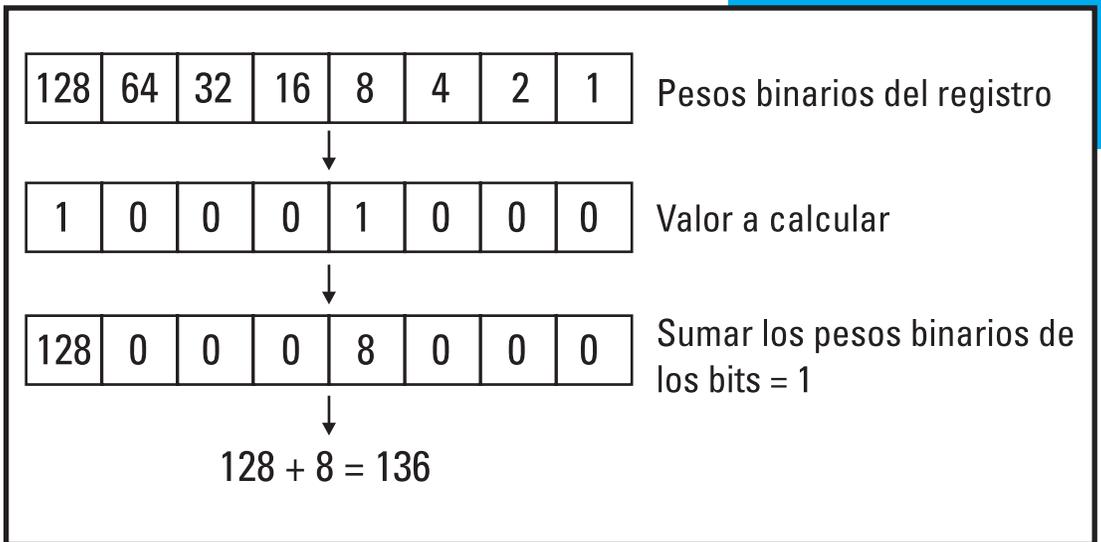
Se llama **peso binario** al valor que representa un bit según la posición que ocupa dentro del registro.

- El bit que está ubicado más a la derecha es el bit menos significativo -LSB; *Least Significant Bit*-; tiene un peso de $2^0=1$.
- El bit del extremo izquierdo del registro el bit más significativo -MSB; *Most Significant Bit*- y tiene un peso de $2^7=128$.

Los pesos binarios crecen de derecha a izquierda en potencias de 2.



Una manera de simplificar el cálculo de conversión de binario a decimal es, directamente, sumar los valores de los pesos binarios de los bits cuyo valor sea 1.



El sistema hexadecimal es un sistema en base 16 y consta de 16 dígitos diferentes que son: del 0 al 9 y, luego, de la letra A a la F -es decir, 10 dígitos numéricos y seis caracteres alfabéticos-.

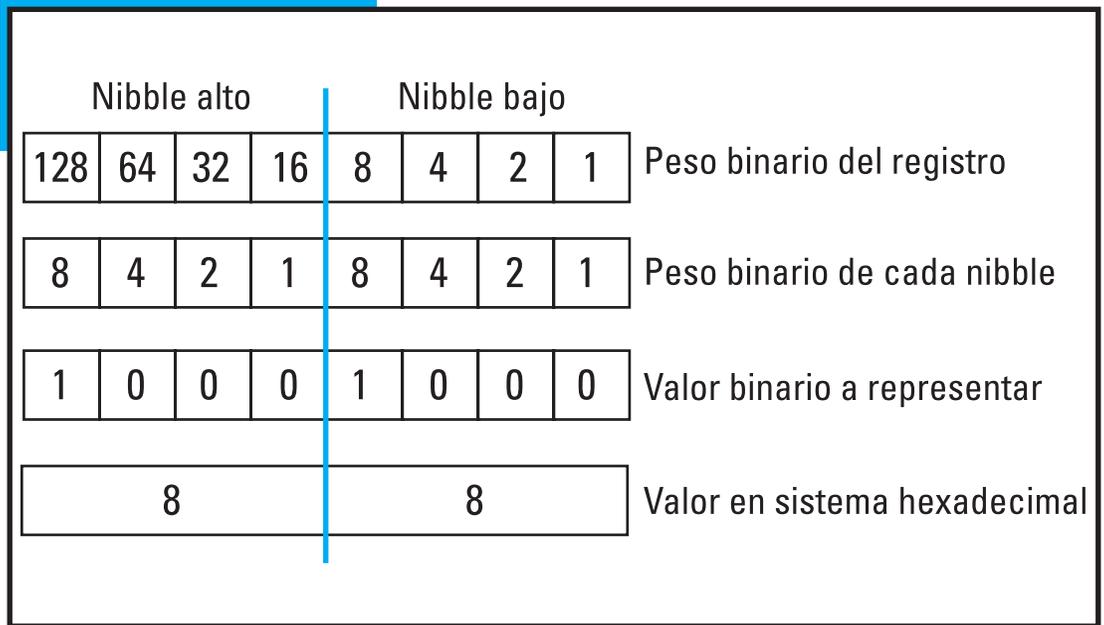
El sistema hexadecimal se usa como forma simplificada de representación de números binarios. Debido a que 16 es una potencia de 2 ($2^4=16$), resulta muy sencilla la conversión de los números del sistema binario al hexadecimal y viceversa.

Mediante el sistema hexadecimal podemos

representar un valor numérico de 8 bits, utilizando sólo 2 dígitos.

De esta manera, estamos dividiendo el registro de 8 bits en 2 partes de 4 bits cada una, llamada *nibble*:

- el *nibble* correspondiente a los 4 bits menos significativos, se denomina *nibble* bajo,
- el *nibble* correspondiente a los 4 bits más significativos, se denomina *nibble* alto.



$$10001000_2 = 136 = 88_{16}$$

El sistema hexadecimal es utilizado para identificar las direcciones de registros de las memorias en sistemas digitales porque nos permite representar el valor de un *nibble* con sólo un dígito, ya que:

$$1111_2 \rightarrow (8+4+2+1) = 15 = F_{16}$$

Esto nos permite representar números grandes utilizando unos pocos dígitos.

Por ejemplo:

$$FF_{16} = 11111111_2 = 65535$$

En la programación de microcontroladores, es habitual utilizar los tres sistemas de numeración -binario, decimal y hexadecimal-, dependiendo del proceso que deseamos realizar.

Líneas de entrada/salida -E/S-

Los microcontroladores cuentan con una serie de pines destinados a entrada y salida de datos o señales digitales. A estos pines se los denomina **puerto**. Un microcontrolador puede contener varios puertos, según su modelo.

Todo el funcionamiento del microcontrolador está controlado a través de los registros, y los puertos no son la excepción; también están controlados por los registros. Por esto, un puerto no puede estar formado por más de 8 pines: 1 pin por cada bit de un registro. Un puerto sí puede estar formado por menos de 8 pines.

Cada puerto se identifica con una letra: "Puerto A", "Puerto B", etc.

Para poder utilizar un puerto, éste debe ser

configurado. Cada pin de un puerto puede ser configurado como entrada o salida, independientemente del resto de los pines del mismo puerto.

Utilizando un microcontrolador

Decíamos que el microcontrolador tiene una memoria de programa de tipo Flash, en la que grabamos las instrucciones necesarias para que el micro realice el trabajo que necesitamos. Cuando compramos un microcontrolador, la memoria flash viene vacía; el microcontrolador "no sabe hacer nada".

Para que un microcontrolador funcione, es necesario programarlo.

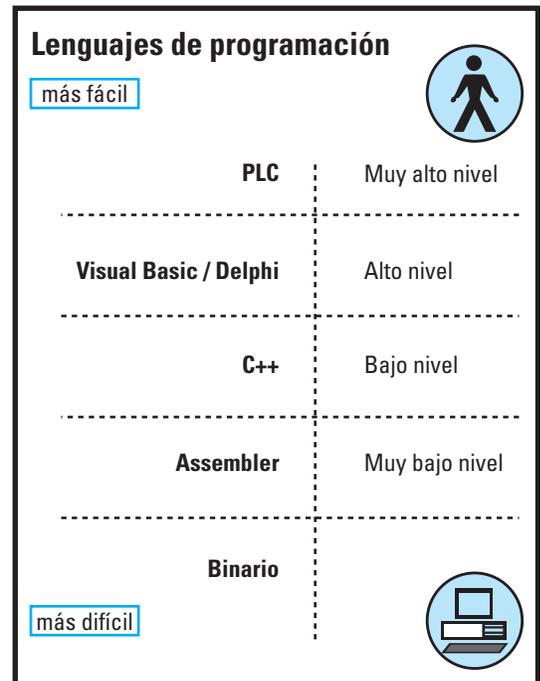
Los microcontroladores se programan en un lenguaje de programación llamado **ensamblador** -*assembler*-, cuya principal característica es su altísima complejidad.

Los lenguajes de programación se clasifican según el nivel de programación en:

- **Lenguaje de alto nivel.** Permite que los algoritmos se expresen en un nivel y estilo de escritura fácilmente legible y comprensible por el hombre. En la actualidad, se trata de lenguajes de tipo visual.
- **Lenguaje de bajo nivel.** Con ellos, el usuario se acerca un poco más al lenguaje de máquina. Permiten un acceso más amplio al control físico de aquella -hardware-.

- **Lenguaje *assembler*.** Podríamos considerarlo el lenguaje de más bajo nivel; el usuario escribe código en el mismo idioma del procesador, cuenta con un control total del sistema e implica un conocimiento de la arquitectura mecánica del procesador para realizar una programación efectiva. El lenguaje de programación es muy específico para cada modelo de procesador; incluso, puede variar de un modelo a otro de procesador dentro de un mismo fabricante.

Podemos decir que los lenguajes de alto nivel se asemejan más al lenguaje humano y que los lenguajes de bajo nivel se asemejan más al lenguaje de máquina, y que, en el lenguaje ensamblador, el usuario debe programar en el propio idioma del procesador.



El microcontrolador sólo entiende de números, por lo que el código *assembler* (un texto) no puede ser procesado directamente por el PIC. Para poder grabar el programa en el micro, primero debemos convertir el texto del código *assembler* a números -en general, en formato hexadecimal-. Este proceso se llama **compilación**. Una vez desarrollado el código *assembler*, es necesario compilarlo.

La compilación se realiza con un soft que se puede obtener de manera gratuita desde la página web del fabricante del microcontrolador en www.microchip.com. El archivo compilado tiene una extensión *.hex*.

Por último, después de compilado, el programa está listo para ser grabado al microcontrolador. Esto se realiza mediante una placa programadora o programador. Existen distintos tipos de programadores que permiten conectar a la PC, mediante el puerto serie (COM) o paralelo (LPT).

Cada programador trabaja en conjunto con un software, mediante el cual se realiza el proceso de lectura y escritura de los microcontroladores.

El proceso de desarrollo del código y grabación de un microcontrolador implica, entonces:

1. Escribir el código *assembler*. Se genera un archivo con extensión *.asm*.
2. Compilar el código *assembler*. Se genera un archivo con extensión *.hex*.
3. Grabar (transferir) el programa desde la PC al microcontrolador, mediante un programador.

De estos tres pasos, el primero insume el 95 % de tiempo y esfuerzo.

El microcontrolador PIC 16F84

Estructura del PIC 16F84

Inicialmente, todos los microcontroladores incorporaron la **arquitectura de Von Neumann** que se caracteriza por disponer de una sola memoria principal donde se almacenan los datos y las instrucciones. Se accede a esta memoria a través de un sistema de buses único (direcciones, datos y control).

El PIC 16F84 -al igual que el resto de los microcontroladores de la actualidad- está diseñado con la **arquitectura Harvard** que dispone de dos memorias independientes: una que contiene sólo instrucciones y otra donde se almacenan los datos. Ambas memorias cuentan con sus respectivos sistemas de buses de acceso; es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias.

Las características técnicas del PIC 16F84 son:

- Voltaje de alimentación: 5 VCC.
- CPU RISC.
- Circuito integrado de 18 pines.
- 13 pines de configurables como entrada o salida.

- Frecuencia operación máxima: 10 Mhz.
- Memoria Flash: 1 kbytes.
- Memoria de datos: 68 bytes.
- Memoria de datos EEPROM: 64 bytes.
- 1 temporizador interno.
- 4 interrupciones.

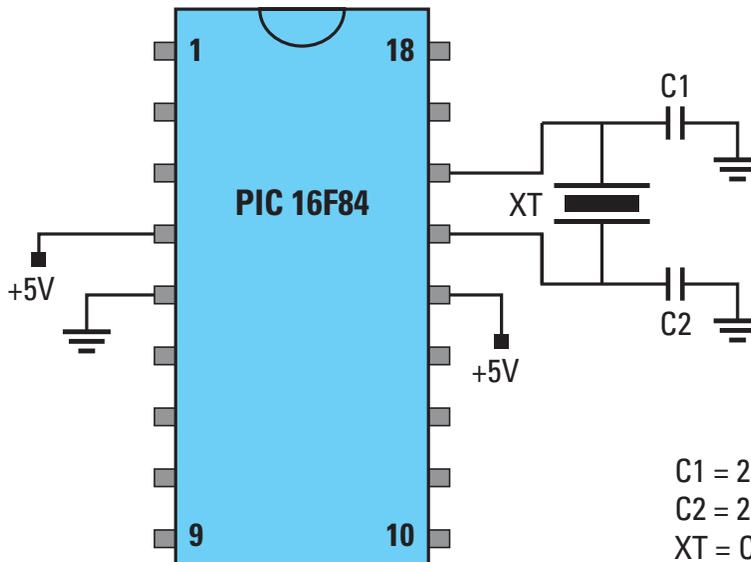
funcionamiento de todo el sistema.

Para que un microcontrolador funcione, es necesario grabar un programa en su interior. Este programa es leído y ejecutado línea a línea por la CPU. Para que la CPU funcione, debe contar con un generador de impulsos de reloj externo que sincroniza el



En nuestro recurso didáctico, para el origen de la señal de reloj externa utilizamos un cristal de 4 MHz.

Dado que el microcontrolador ejecuta cada instrucción -es decir, cada línea de código- a $\frac{1}{4}$ de la velocidad del cristal, el resultado es que el microcontrolador está ejecutando el código a una velocidad de 1 MHz. Es decir que ejecuta 1.000.000 de líneas de código del programa por segundo. Por lo tanto, cada línea de código se ejecutará en 1 microsegundo (1/1.000.000 s).

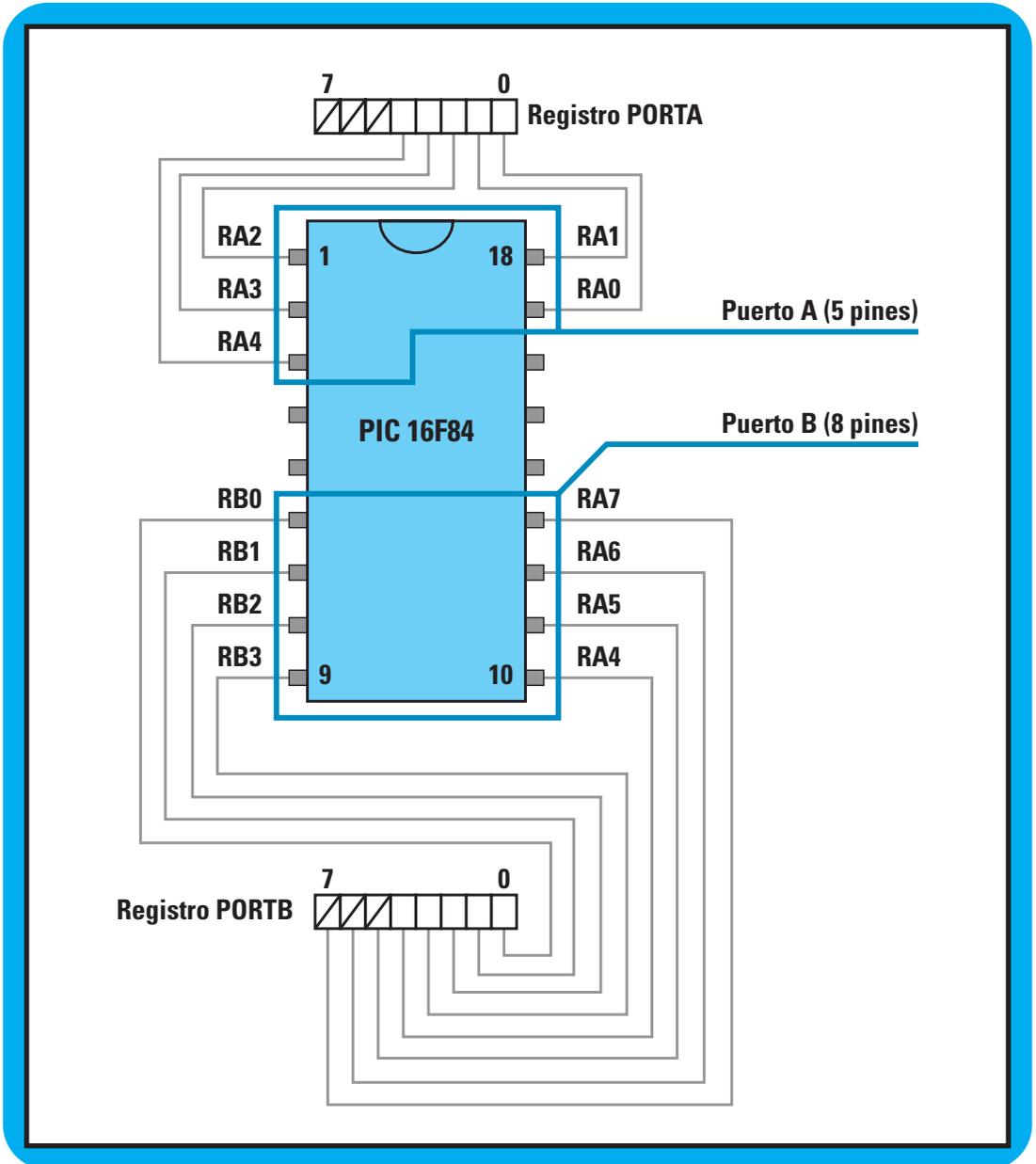


Conexión del PIC 16F84

Los puertos del PIC 16F84

El PIC 16F84 dispone de dos puertos denominados "Puerto A" y "Puerto B":

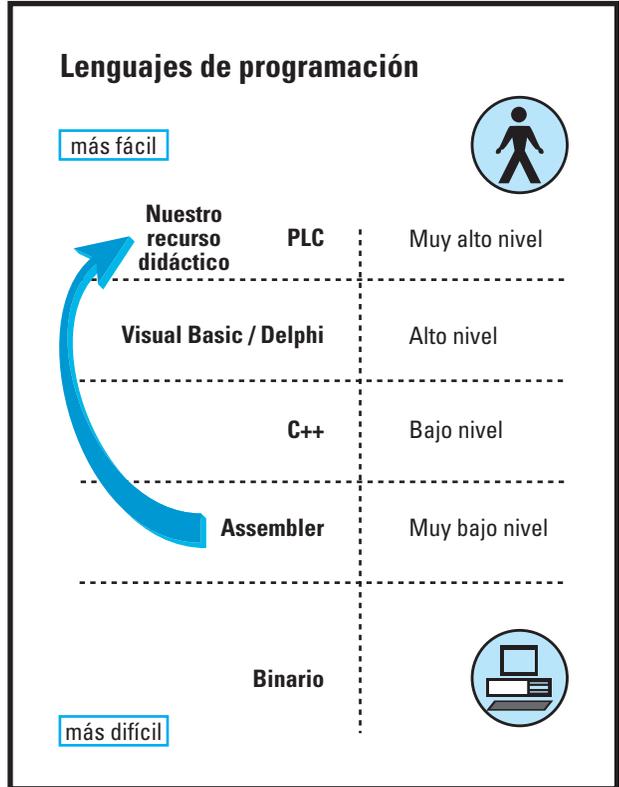
- El puerto A cuenta con 5 pines de entrada/salida.
- El puerto B, con 8 pines de entrada/salida.



3. EL EQUIPO EN EL AULA

La programación de los microcontroladores suele resultar una tarea muy compleja, dado que el usuario debe conocer la arquitectura del procesador y, además, dominar el lenguaje ensamblador *-assembler-* de programación. Porque, si bien existen distintas versiones del lenguaje C o Basic adaptadas a la programación de microcontroladores, la programación todavía resulta muy compleja para un alumno de escuela técnica o profesional, por el alto grado de abstracción que implica (Actualmente, la tecnología de microcontroladores está siendo implementada sólo en las carreras de ingeniería en electrónica).

Nuestro Entorno de desarrollo para programación de microcontroladores PIC *-circuitos integrados programables-* propone una herramienta de software para programación que va a permitir que usted y sus alumnos sólo diseñen un diagrama de flujo de manera totalmente visual y en alto nivel de programación, en lenguaje humano, y que el software genere el código *assembler* de manera automática. Además, el software controla y supervisa todo el proceso de diseño, advirtiendo a los usuarios sobre posibles errores en la lógica del programa que están diseñando.



De esta manera, el recurso didáctico que le proponemos integrar en sus clases ofrece la posibilidad de realizar desarrollos con microcontroladores y resulta la herramienta ideal para el aprendizaje de esta tecnología de última generación.

Para esto, el software ofrece un entorno totalmente visual que se adapta a las características del modelo de microcontrolador

seleccionado:

- carga el mapa de memoria con sus respectivos nombres de registros y bits,
- asigna las funciones correspondientes a cada registro y bit dentro del sistema,
- ajusta el tamaño de la memoria EEPROM,
- activa los módulos correspondientes, etc.

Así, usted y sus alumnos tienen disponibles sólo las características que corresponden al modelo de micro en uso.

El concepto que sirve de fundamento a este recurso es que profesores y alumnos se concentren en "qué es lo que quieren hacer para concretar sus proyectos tecnológicos" y el entorno "sabe cómo hacerlo"; la tarea del aula se dedica a diseñar la estructura lógica del proceso que los estudiantes van a realizar y el software se encarga de resolver todo el desarrollo del código *assembler*, controlando hasta el más mínimo detalle.

¿Recuerda las consignas de trabajo que nuestros colegas presentaban a sus alumnos y que sintetizamos al comienzo de este material? Vamos a volver a ellas, ahora integrando el Entorno de desarrollo para programación de microcontroladores PIC -circuitos integrados programables-.



Baliza electrónica

Como respuesta a la situación problemática generada por accidentes de tránsito constantes, los alumnos generan una baliza -de bajo costo, confiable, de fácil accionamiento y porte pequeño-.

Para concretarla, prevén:

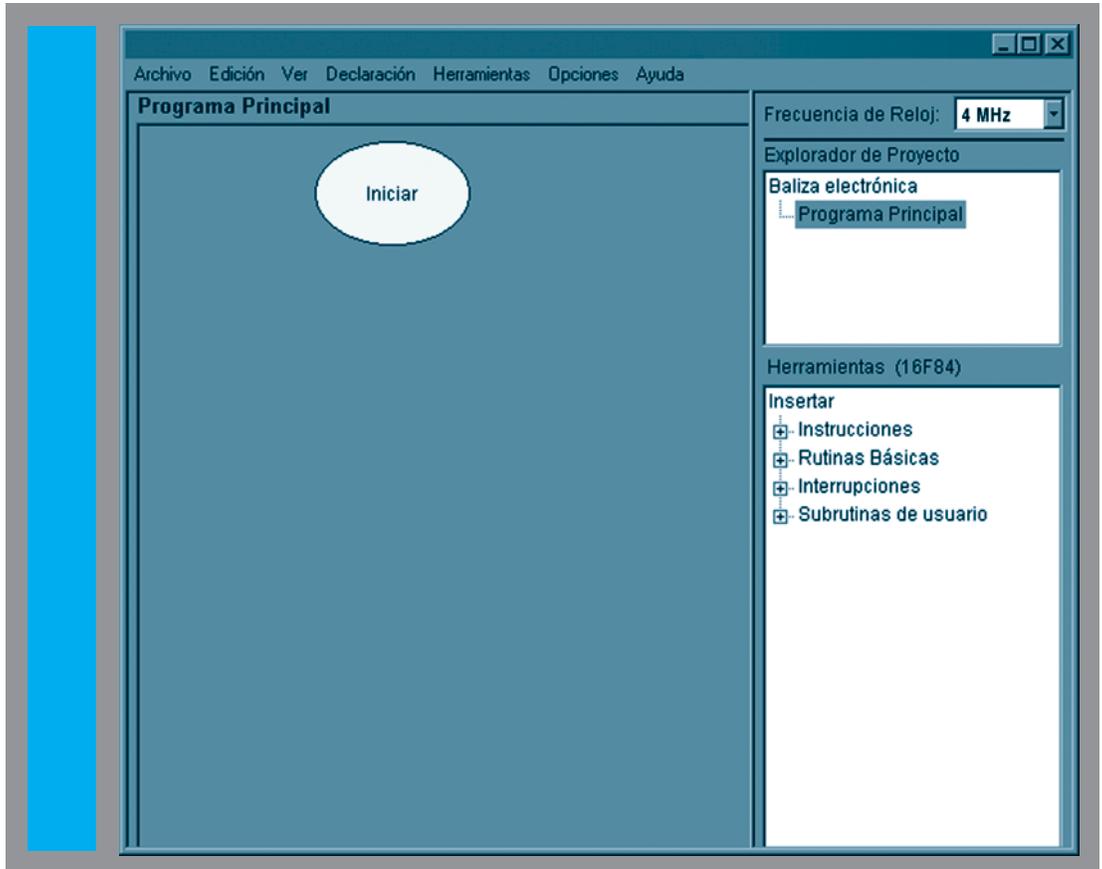
Desarrollar un programa para que el PIC controle 2 diodos led de manera intermitente, encendiendo y apagando cada led durante 1 segundo, de manera alternada. Los led serán conectados a los pines 0 y 1 de un puerto del puerto A del microcontrolador.

La experiencia permite a los alumnos:

- Conocer los puertos de comunicación de los microcontroladores, su configuración y uso.
- Realizar temporizaciones con un microcontrolador.

Veamos cómo ayuda el equipo en esta tarea...

Al crear un nuevo proyecto, el equipo inserta el bloque de inicio del diagrama de flujo, de manera automática. A partir de este bloque, se va desarrollando todo el diagrama de flujo.



Para desarrollar la baliza electrónica es necesario:

- Asignar un nombre a los pines del puerto.
- Configurar los puertos del PIC.
- Utilizar el puerto.
- Realizar temporizaciones.
- Cambiar el estado de los diodos led.
- Generar el código *assembler*.
- Compilar el programa.
- Grabar el programa al micro.



Asignar un nombre a los pines del puerto

Para identificar más fácilmente a los bits 0 y 1 del puerto A, vamos a cambiar el nombre de cada uno de ellos:

- Al bit A (0) lo llamamos "Led izquierdo".
- Al bit A (1) lo llamamos "Led derecho".

Para esto, ingresamos al menú "Declaración/Bits", seleccionamos en el mapa de memoria, el Bit RA0. Luego, presionamos la tecla "Enter".

Declaración de Bits

Mapa de Memoria

Banco 0 | Banco 1

Banco 0									
Registro		Bits							
Dir.	Nombre	7	6	5	4	3	2	1	0
00h	INDF								
01h	TMR0								
02h	PCL								
03h	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C
04h	FSR								
05h	PORTA				RA4_T0CKI	RA3	RA2	RA1	RA0
06h	PORTB	RB7	RB6	RB5	RB4	RB3	control	RB1	RB0_INT
07h									
08h	EEDATA								
09h	EEADR								
0Ah	PCLATH								
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

Referencias

- Registros Predefinidos por el PIC.
- Registros de Usuario.
- Registros deshabilitados.

Información Adicional

Nombre del Bit: ---
Función de Bit: ---

Aceptar

Al presionar la tecla "Enter", la pantalla presenta un mensaje donde le informa que se cambiará el nombre del bit seleccionado. Usted debe responder afirmativamente para continuar con el proceso de renombrar el bit: Presione el botón "Sí" del cuadro de diálogo.

A continuación, se presenta la pantalla "Renombrar bit".

Ingrese el nombre del bit, en este caso "led_der" y, como comentario, "Led derecho".

Renombrar un Bit.

Renombrar el bit RA0.

Bit número 0 del registro 'PORTA'.

Nombre del Bit: led_der

Función del Bit: Led Derecho

Renombrar

Presione le botón "Renombrar".

Si el bit al cual estamos cambiando el nombre (en este caso, RA0) ya está siendo utilizado por algún bloque o módulo dentro del diagrama de flujo, el soft le ofrece la posibilidad de actualizar el diagrama de flujo, actualizando el nombre del bit de manera automática también en los textos de bloques del diagrama.

Renombrar bit

¿Desea que Niple actualice el diagrama de flujo con el nuevo nombre de bit?

SI NO

Como en este caso aún no se ha utilizado el Bit RA0 en ningún bloque, no es necesario realizar la actualización del diagrama de flujo; por esto, presionamos el botón "NO".

Declaración de Bits
Mapa de Memoria

Banco 0 | Banco 1

Registro		Banco 0							
Dir.	Nombre	7	6	5	4	3	2	1	0
00h	INDF								
01h	TMR0								
02h	PCL								
03h	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C
04h	FSR								
05h	PORTA				RA4_TOCKI	RA3	RA2	RA1	led_der
06h	PORTB	RB7	RB6	RB5	RB4	RB3	control	RB1	RB0_INT
07h									
08h	EEDATA								
09h	EEADR								
0Ah	PCLATH								
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

Referencias:
 Registros Predefinidos por el PIC.
 Registros de Usuario.
 Registros deshabilitados.

Información Adicional:
 Bit número 0 del registro 'PORTA':
 Nombre del Bit: led_der
 Función de Bit: Led Derecho

Aceptar

De esta manera hemos, cambiado el nombre del bit RA0 a "led_der", con lo cual nos resulta más fácil identificarlo y recordar la función que le hemos asignado.

De la misma manera, cambiamos el nombre del Bit "RA1" por "led_izq".

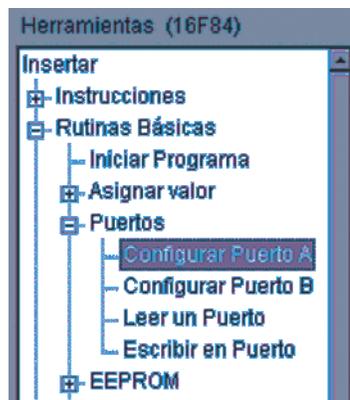
Configurar los puertos del PIC

Para poder utilizar los puertos del microcontrolador, es necesario configurarlos: Indicamos al micro cuáles son los pines que queremos utilizar como entrada y cuáles son los pines que queremos utilizar como salida.

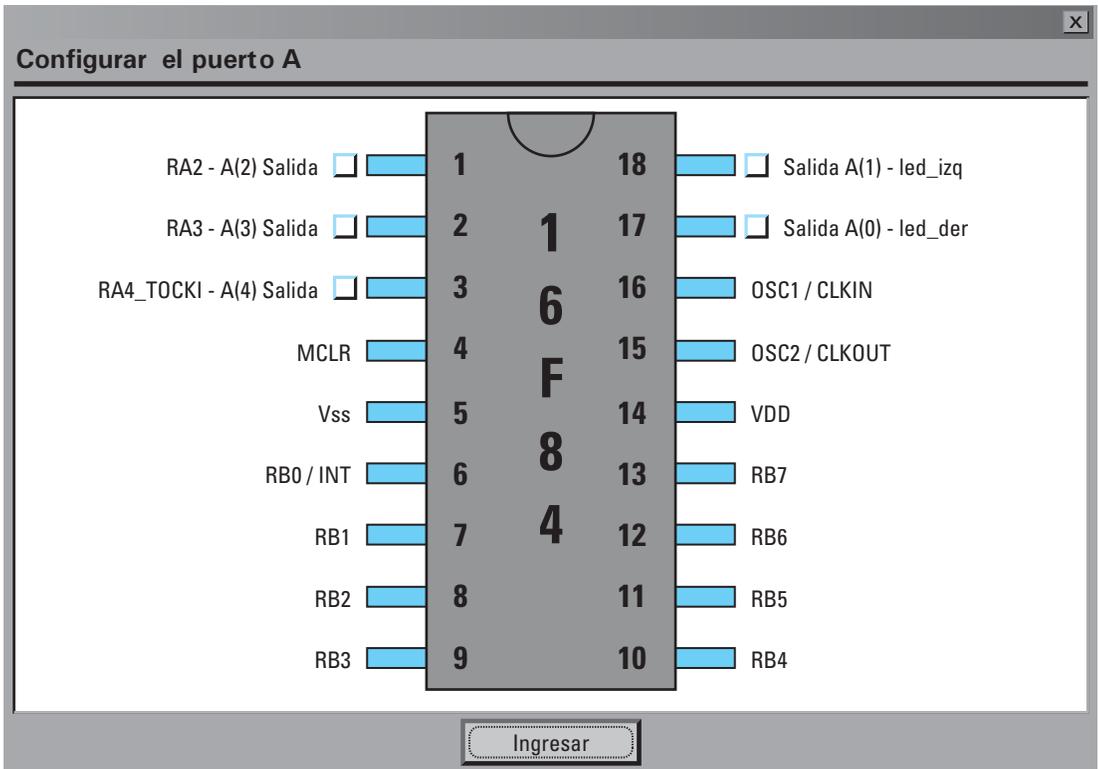
Como mencionamos anteriormente, los puertos del micro -así como el resto del sistema-, se controlan mediante registros en la

memoria cuyas funciones vienen grabadas de fábrica (registros TRISA y TRISB).

Para configurar los pines del "Puerto A", ingrese al árbol de herramientas en: "Herramientas/Rutinas Básicas/Puertos/Configurar Puerto A".



Al realizar un doble clic, se presenta la pantalla de configuración del puerto A.



Note que se presentan los nombres de los pines "led_der" y "led_izq" en los pines 0 y 1 del puerto A, respectivamente.

Configuramos los pines del puerto A como salida.

Utilizar el puerto

Una vez configurado el puerto A, vamos a iniciar la secuencia de encendido de los led.

Como usted sabe, la electrónica digital consiste en procesar información lógica

Recordemos que a un bit sólo le podemos asignar los valores lógicos "0" y "1".

binaria; en ella, el valor lógico "1" corresponde a un nivel de tensión -que, en el caso de los microcontroladores es 5 V- y el valor lógico "0" corresponde a 0 V.

De esta manera, si queremos que el micro saque 5 V por un pin, debemos asignar el valor lógico "1" al bit correspondiente al pin.

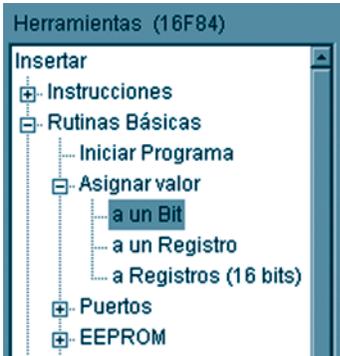
Iniciamos la secuencia de encendido de los led de nuestra baliza, prendiendo el led de la izquierda y apagando el led de la derecha.

Para esto, asignamos:

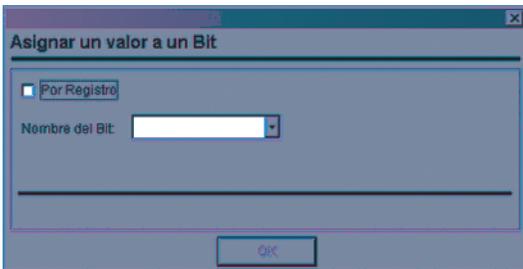
- un "1" al led llamado "led_izq" y
- un "0" al led "llamado "led_der"

que corresponden a los pines A(1) y A(0) respectivamente.

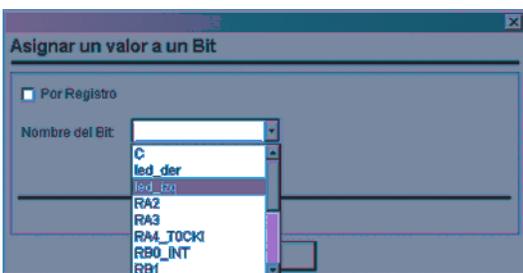
Para asignar un valor a un bit, ingresamos en el árbol de herramientas a:
"Herramientas/Rutinas Básicas/Asignar valor/ a un Bit"



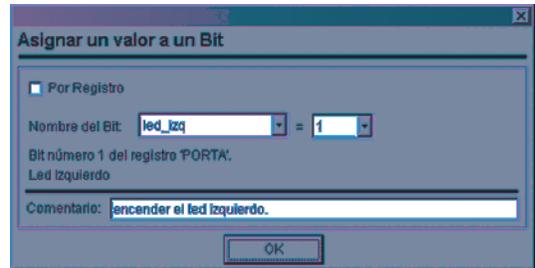
Al realizar un doble clic, se presenta la pantalla "Asignar un valor a un bit".



En la lista, seleccione el nombre del bit al cual quiere asignar un valor; en este caso, seleccione el bit llamado "led_izq".



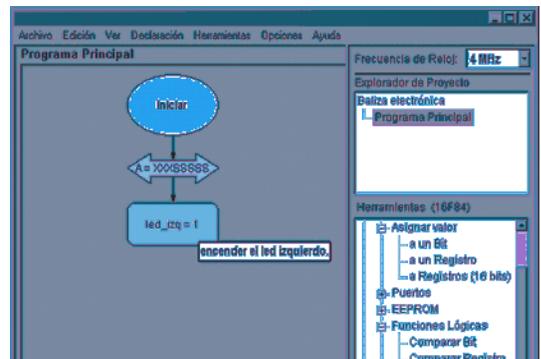
Una vez seleccionado el nombre del bit, especifique el valor que desea asignarle. En este caso, es el valor "1".



Podemos incluir un comentario adicional al bloque. Este comentario resulta de gran utilidad para interpretar el programa, en el momento de revisar o corregir el diagrama de flujo en el futuro. Sobre todo, cuando ha transcurrido un largo período de tiempo desde el desarrollo del proyecto.

Observe que, debajo de la lista de selección del nombre del bit, la pantalla informa a qué registro pertenece y en qué posición se encuentra dentro de aquel. Esto facilita aún más la identificación del bit.

Por último, presione el botón "OK" para insertar el bloque correspondiente en el diagrama de flujo.

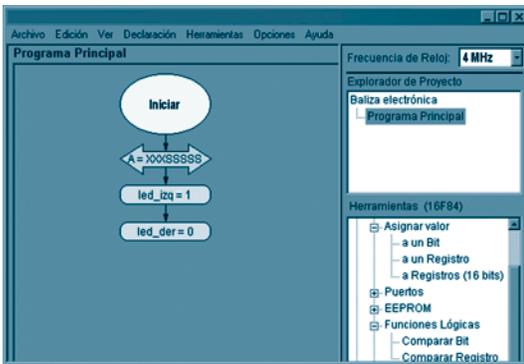


Observe que, al poner el mouse sobre el bloque, se muestra el comentario correspondiente.

De esta manera, usted puede ir recorriendo todo el diagrama de flujo, consultando los comentarios.

Realice el mismo procedimiento para asignar el valor lógico "0" al bit "led_der".

Hasta el momento, el diagrama de flujo debe verse de la siguiente manera:



Una vez que hemos iniciado la secuencia de encendido de los led, el microcontrolador debe "esperar" un tiempo -por ejemplo, 1 segundo-, antes de cambiar el estado de los led (encendido o apagado).

Realizar temporizaciones

Existen, básicamente, dos maneras de realizar temporizaciones en un microcontrolador:

- Mediante un bucle temporizador. Es una temporización por software. El

micro se queda ejecutando un "bucle" (repeticiones cíclicas durante una determinada cantidad de veces) hasta completar una cantidad de ciclos de programa. La temporización se realiza calculando la cantidad de ciclos por el tiempo de ejecución de cada ciclo. El tiempo de ejecución de un ciclo es de $\frac{1}{4}$ de la velocidad de reloj.

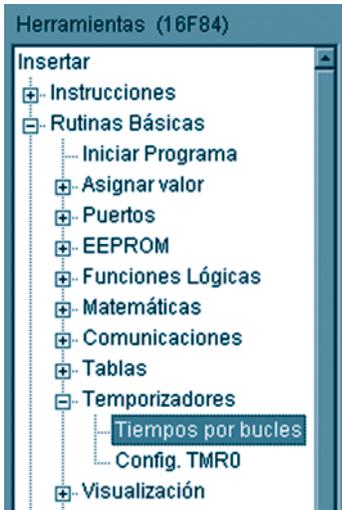
Por ejemplo, en este proyecto de la baliza estamos utilizando un cristal de 4 MHz; la velocidad de ejecución interna del microcontrolador es 1 MHz, es decir, 1.000.000 de ciclos por segundo, con lo cual cada instrucción se ejecuta en un microsegundo (μ s). Como este tipo de temporización se realiza ejecutando una serie de líneas de código, el microcontrolador no puede atender otras tareas mientras temporiza.

- Mediante el temporizador interno del microcontrolador (TMR0). El microcontrolador PIC 16F84 incluye un módulo temporizador, llamado TMR0, que trabaja de manera autónoma y que genera una "interrupción" cuando se completa una temporización. El usuario puede configurar el temporizador para realizar distintas temporizaciones. Como el TMR0 ejecuta la temporización de manera independiente a la concreción del código, el microcontrolador puede "dedicarse" a realizar otras tareas mientras la temporización se realiza.

Para la baliza electrónica utilizamos el método de temporización por bucles, ya

que el microcontrolador no debe atender otras tareas mientras la temporización se efectúa.

Para concretar una temporización por bucles, ingrese a árbol de herramientas en: "Herramientas/Insertar/Rutinas básicas/Temporizadores/Tiempos por bucles"



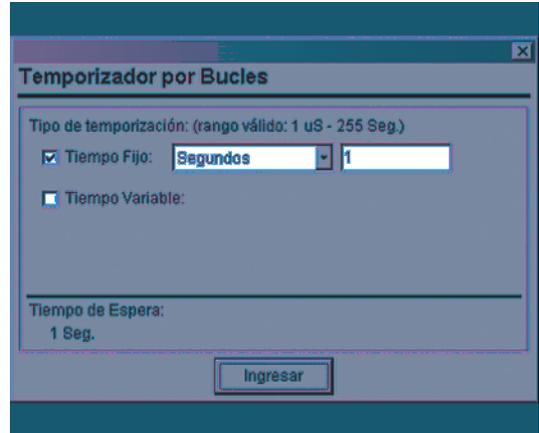
Al hacer doble clic, se presenta la pantalla de "Temporizador por bucles".

El sistema le permite dos maneras de temporizar por bucles:

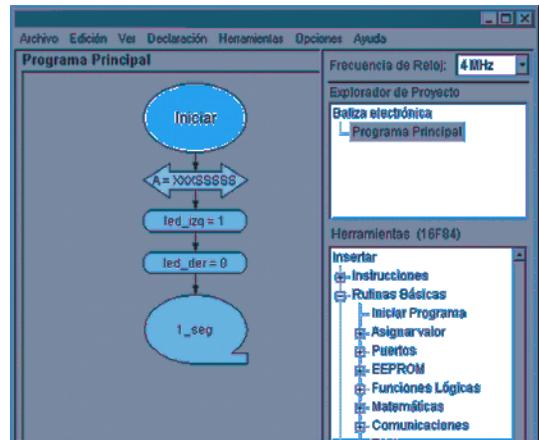
- **Tiempos fijos.** Se especifica el tiempo que se desea temporizar y el microcontrolador realiza siempre la misma temporización.
- **Tiempos variables.** La temporización es controlada por el valor de un registro, el que puede ser modificado en cualquier parte del proceso. De esta manera, puede realizar temporizaciones con distintos períodos de tiempo en la misma parte del proceso.

En este caso, utilizamos un tiempo fijo.

Seleccione la opción "Tipo fijo". La base de tiempo es "segundo". Por último, ingrese el valor 1 en el cuadro de texto.



Presione el botón "Ingresar" para insertar el bloque de temporización en el diagrama de flujo.



Cambiar el estado de los diodos led

Hasta el momento, el programa se inicia encendiendo el led izquierdo y apagando el

led derecho; ambos se mantienen en ese estado durante un segundo mediante el temporizado.

Después de haber transcurrido un segundo, debemos cambiar el estado de los led, apagando el led de la izquierda y encendiendo el led de la derecha.

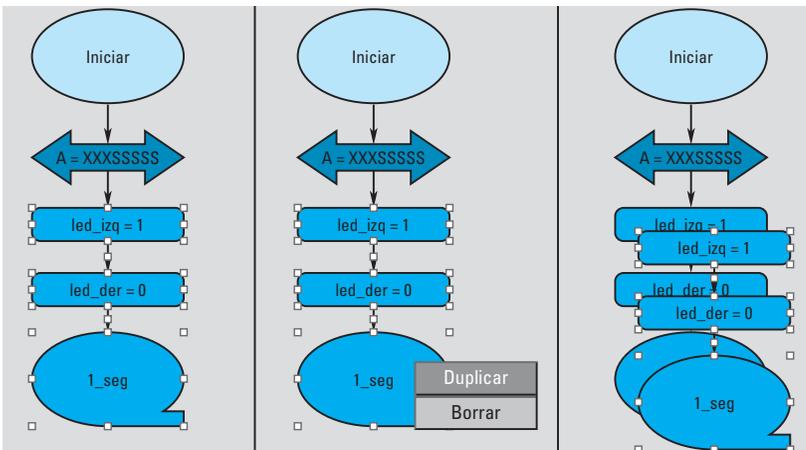
Esto se puede realizar repitiendo los procedimientos de asignación de valores a los displays y temporización, o duplicando los bloques ya insertados y cambiándoles los valores de asignación de bits.

Optamos por este último procedimiento.

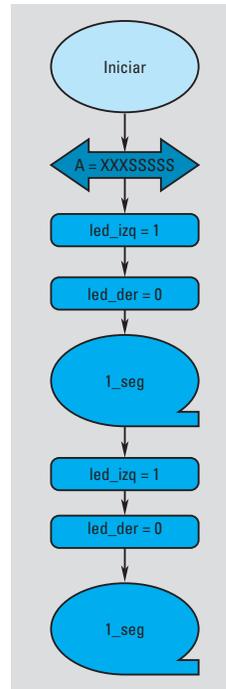
Para concretarlo, seleccione los bloques a duplicar: Los dos bloques de asignación a bit y el bloque de temporización.

Una vez seleccionados, haga un clic con el botón derecho del mouse y seleccione la opción "Duplicar" del menú contextual. De esta manera, se realiza un duplicado de los bloques seleccionados.

La secuencia es ésta: ▼

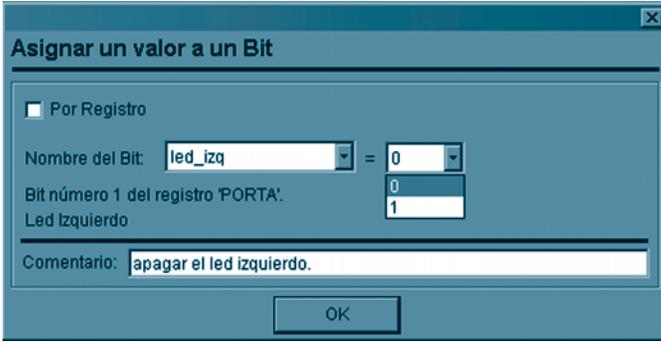


Luego, desplazamos los nuevos bloques hacia abajo, ubicándolos a continuación en el diagrama de flujo. Vinculamos los bloques mediante una flecha (vínculo), desde el bloque de la primera temporización hacia la primera asignación de los bloques recién duplicados.



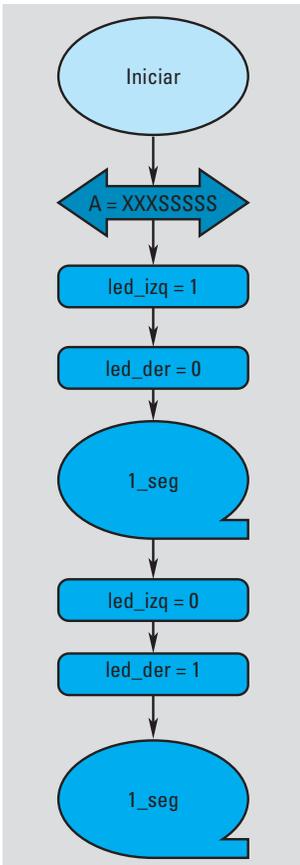
Para cambiar el valor de asignación del bit "led_izq", seleccione el bloque y presione la tecla "Enter". Esto le permite cargar la pantalla del bloque seleccionado para cambiar el valor de asignación -edición de un bloque-. En nuestro equipo, la mayoría de los bloques puede ser editada, permitiendo realizar correcciones o actualizaciones -de manera muy simple, sin necesidad de borrar el bloque-, e insertarlos nuevamente.

Cuando se presenta la pantalla "Asignar un valor a un bit", sólo cambie el parámetro del valor asignado de "1" a "0", corrija el comentario y, luego, presione el botón "Ingresar".



Repita la operación con el bloque de asignación al bit "led_der", cambiando el valor "0" por "1".

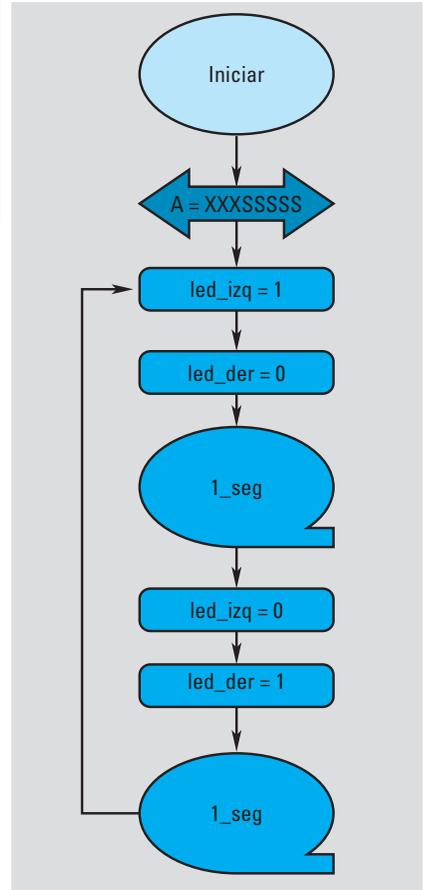
El diagrama de flujo queda así:



Por último, cerramos el ciclo mediante un vínculo entre el último bloque de temporización y el primer bloque de asignación al Bit "led_izq".

Las flechas indican el sentido de ejecución del programa; siempre, comenzando desde el bloque "Iniciar".

Después de realizado el vínculo, el diagrama de flujo se ve así:



Generar el código ensamblar

Una vez diseñado el diagrama de flujo con el proceso que deseamos que el microcontrolador realice, generamos el código *assembler*, el programa escrito en lenguaje ensamblador.

Para esto, ingrese al menú Archivo/Archivo ASM/Generar Código ASM.

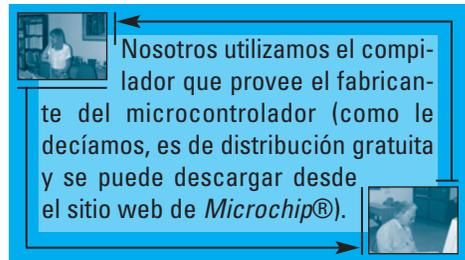


Compilar el programa

Para poder grabar el programa al microcontrolador, debemos compilarlo. La compilación consiste en traducir el programa en lenguaje *assembler* a formato numérico -a cada instrucción le corresponde un código numérico-.

Se presenta la pantalla de opciones de generación del código *assembler*.

La compilación del programa se realiza por medio de un programa "Compilador". Existen distintos programas compiladores.

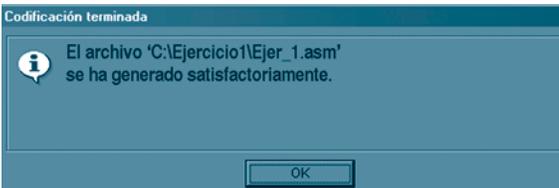


Presione el botón "Codificar ASM".

Grabar el programa al micro

Si la generación del código *assembler* se realiza correctamente, el programa informa la finalización de la codificación mediante el mensaje.

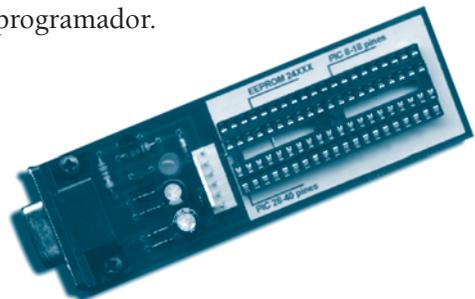
Una vez que el programa ha sido compilado, es necesario grabarlo al microcontrolador. Esto se realiza mediante un programador.



El **programador** es una plaqueta electrónica que se conecta a la PC, que permite transferir el programa al microcontrolador.

A continuación se presenta el código *assembler* generado.

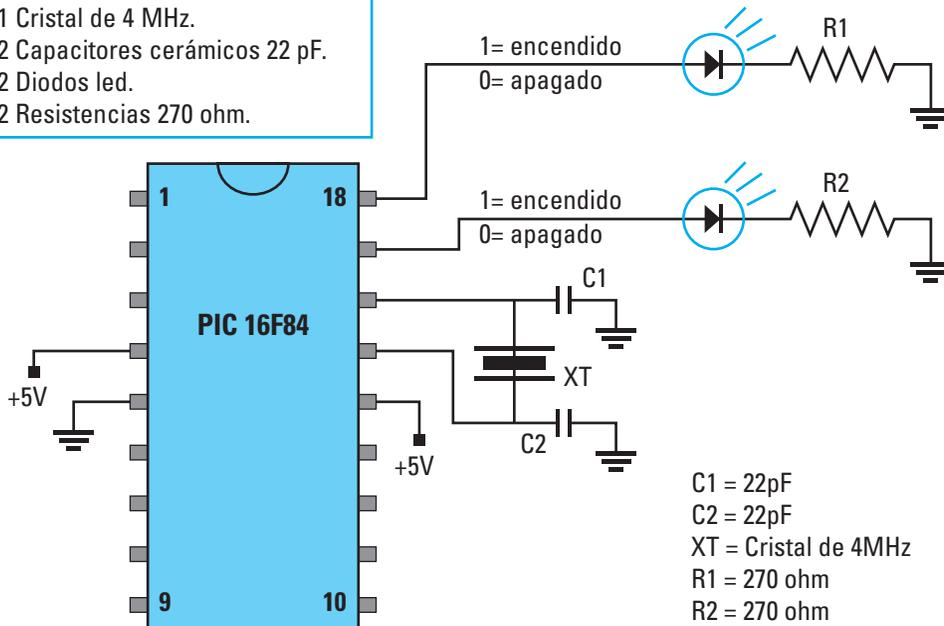
Dado que se trata de un archivo de texto, se muestra en el "Block de notas" de Windows -que es un editor de textos-.



Esquema electrónico de la baliza

Lista de materiales:

- 1 PIC 16F84.
- 1 Cristal de 4 MHz.
- 2 Capacitores cerámicos 22 pF.
- 2 Diodos led.
- 2 Resistencias 270 ohm.



Baliza electrónica programable

Supongamos que un proyecto tecnológico requiere una baliza electrónica programable que muestre diferentes combinaciones de destello según la configuración seleccionada, para llamar la atención o alertar a los conductores de diferente manera, según la situación o el estado en que se encuentre el camino.

Este proyecto tecnológico permite a sus alumnos:

- Comparar un valor y realizar diferentes acciones en función del valor de un dato.
- Diseñar sus propias combinaciones de destello, descubriendo el inmenso potencial de los microcontroladores, y estimulando su creatividad y su imaginación.

Para este proyecto integramos la baliza electrónica ya desarrollada y le agregamos las modificaciones necesarias para realizar otro producto que muestre diferentes combinaciones de destello según una configuración seleccionada mediante un "Dip-switch" de 5 pines.



Para concretar la baliza electrónica programable es necesario:

- Integrar el dispositivo "Dip-Switch".
- Configurar el puerto B.
- Declarar un registro de usuario.
- Leer el puerto B.
- Comparar un registro.



Integrar el dispositivo "Dip-Switch"

En la nueva baliza que estamos diseñando, el usuario puede seleccionar la configuración de destello a visualizar. Esto se realiza mediante un dispositivo llamado *Dip-Switch*.

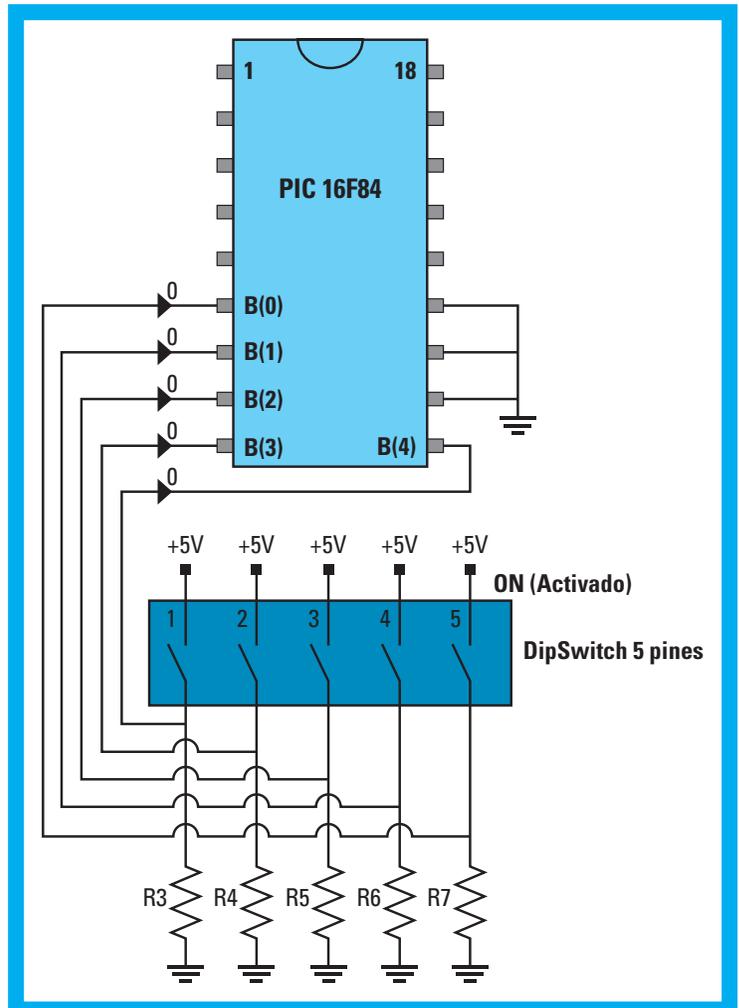
El *Dip-Switch* no es más que un conjunto de llaves de corte en miniatura que se conecta a los pines 0 al 4 del puerto B:

Como podemos ver en el esquema electrónico, mientras las llaves se encuentran en estado "Abierto", los pines del PIC están conectados a masa mediante las resistencias de 10 kΩ. Como en electrónica digital, el valor de 0 V corresponde al valor lógico, vemos que, en este caso, el micro está reci-

biendo el valor binario "00000000" en el puerto B.

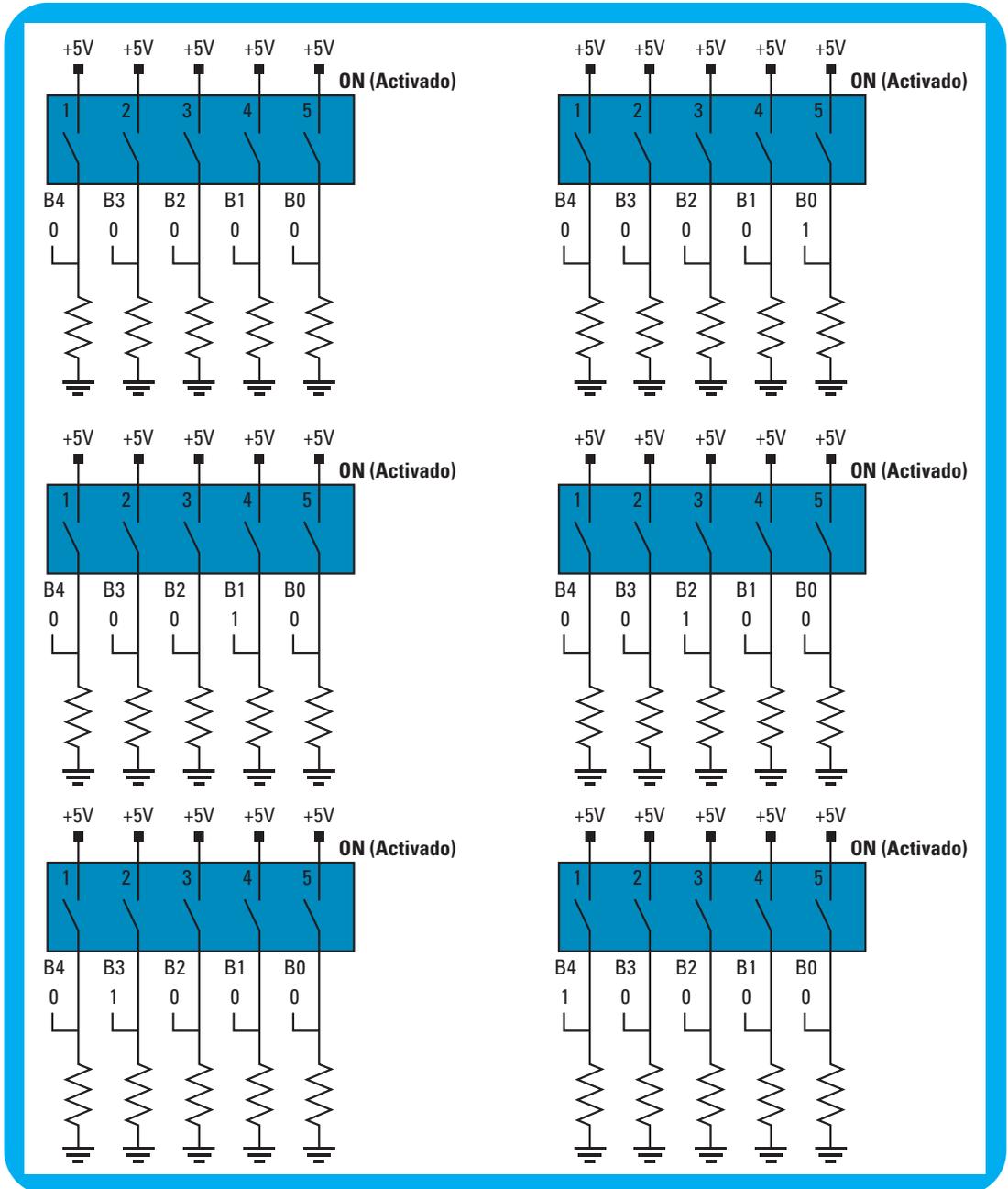
En nuestra baliza, utilizamos un *Dip-Switch* de 5 pines.

Cada vez que activamos una llave del *Dip-Switch*, estamos conectando la línea a +5 V. De esta manera, estamos enviando un "1" lógico al pin correspondiente, formando un código binario según la llave que hemos conectado.



Para interpretar mejor el número que estamos formando con el *Dip-Switch*, conectamos el pin que se encuentra ubicado más a

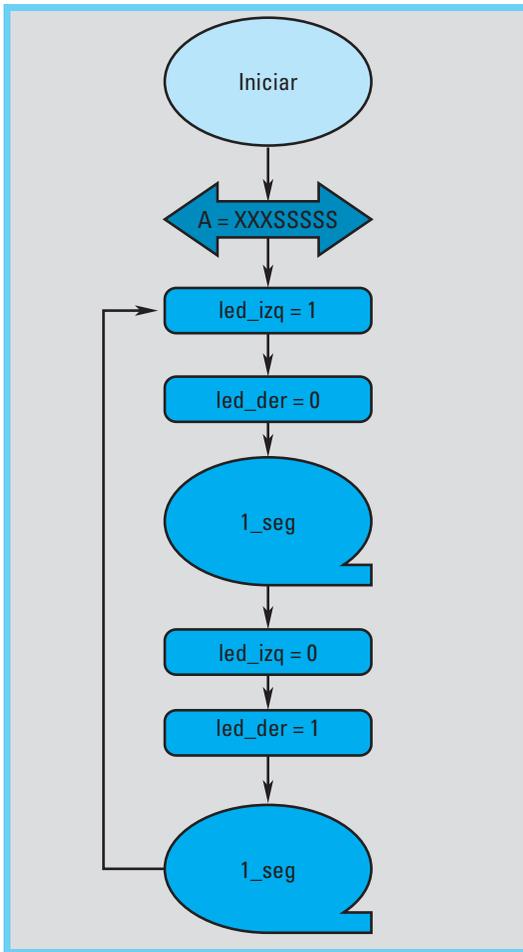
la izquierda del *Dip-Switch* al pin B(0) del microcontrolador. De esta manera, estamos formando los siguientes códigos:



Para poder leer el estado de los pines del puerto B, configuramos el puerto.

Configurar el puerto B

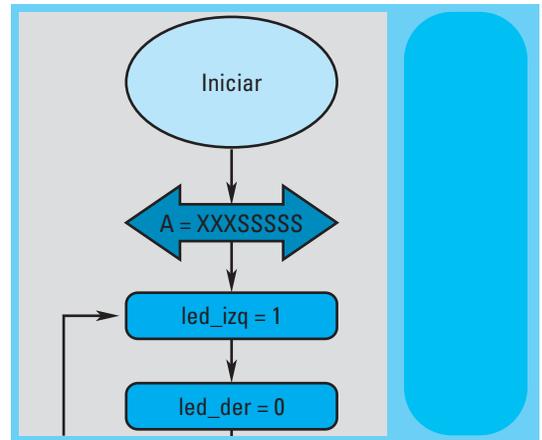
Abrimos el ejemplo 1.



El programa comienza configurando el puerto A como salida.

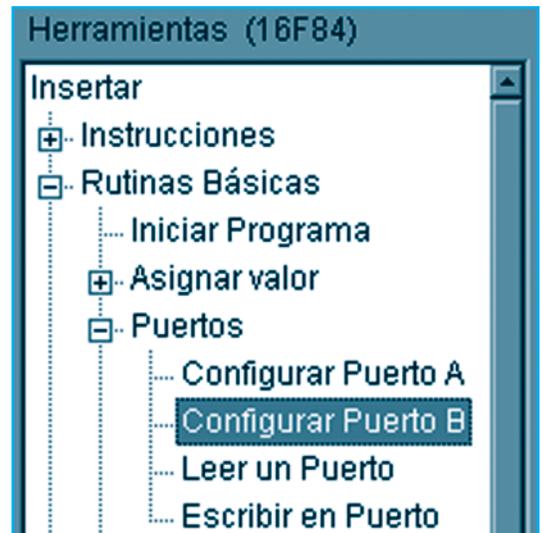
A continuación de este bloque, insertamos un bloque de configuración del puerto B.

Para esto, borramos el vínculo (flecha) entre el bloque de configuración del puerto A y el bloque de asignación de valor al bit "Led_Izq".

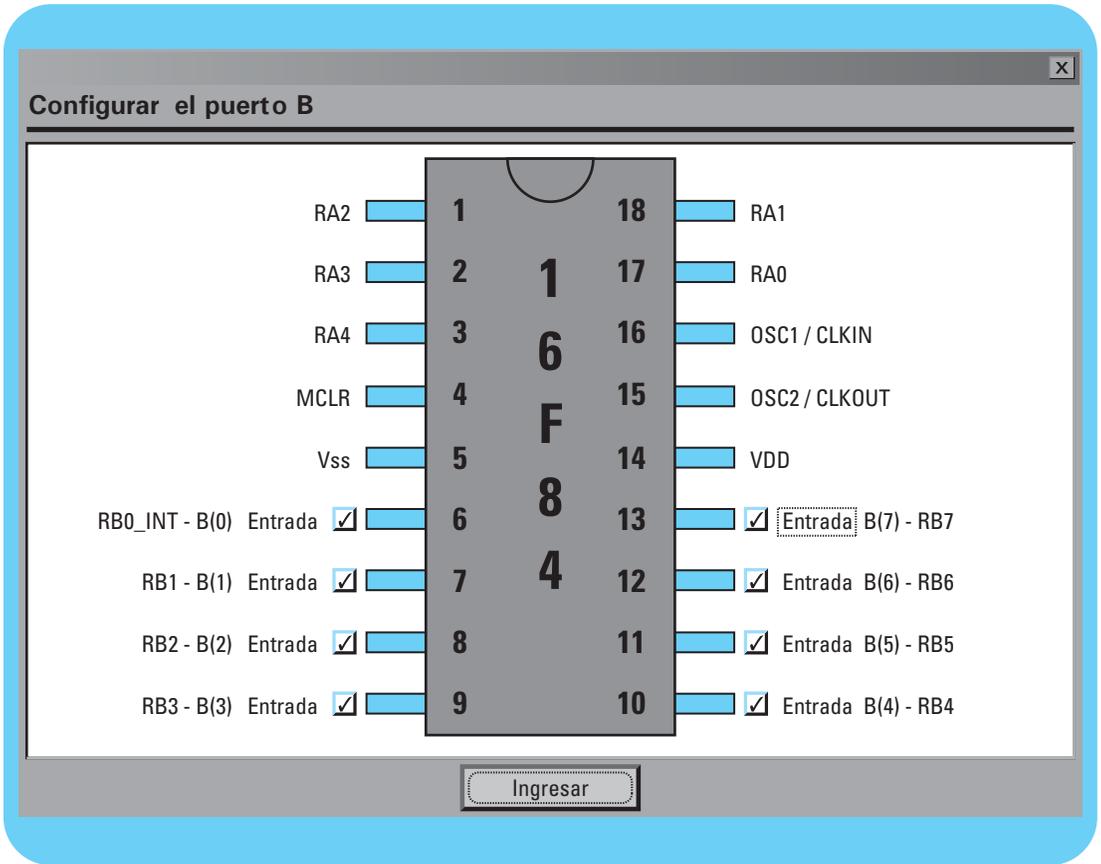


Una vez borrado el vínculo, insertamos el bloque de configuración del puerto B.

Hacemos esto ingresando al bloque de configuración del puerto B en:



Al seleccionar la opción "Configurar el puerto B", se presenta la pantalla de configuración de dicho puerto:



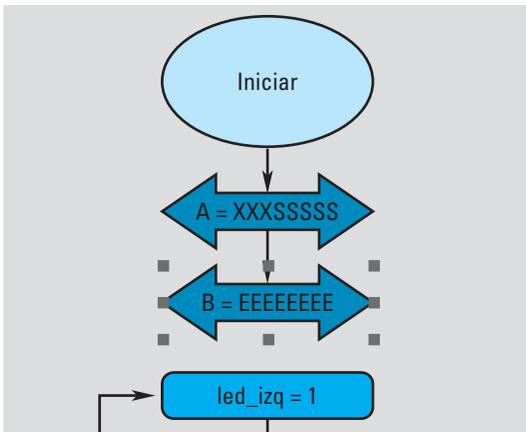
Presione el botón "Ingresar", para insertar el bloque en el diagrama de flujo.

Declarar un registro de usuario

Una vez configurado el puerto, vamos a leer el valor de su estado para determinar la configuración de baliza que el usuario ha seleccionado.

Guardamos el estado del puerto en un registro para, luego, poder compararlo con una serie de códigos y, así, seleccionar la combinación de destello de luces correspondiente.

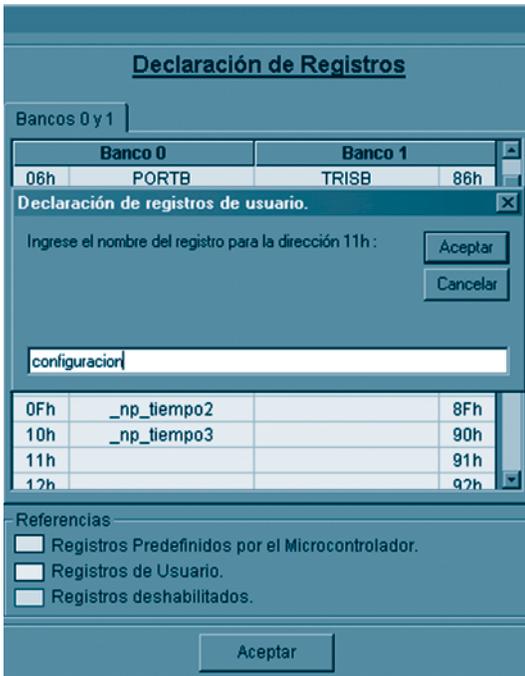
Para esto, declaramos un registro en la memoria del PIC.



Lo hacemos ingresando a:

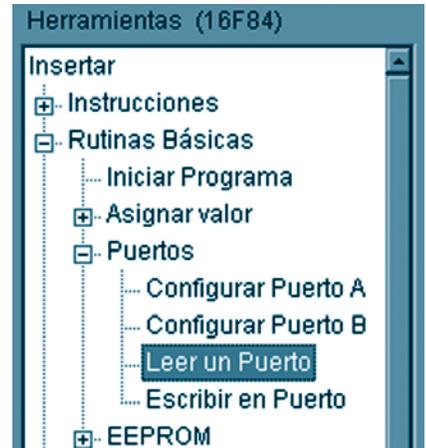


Hacemos un clic en el primer registro de usuario disponible (de color verde) e ingresamos el nombre del registro que queremos declarar. En este caso, le asignamos el nombre "Configuración" -por supuesto, sin tilde-.



Leer el puerto B

El siguiente paso consiste en realizar la lectura del puerto B. Para esto, ingresamos a:



Seleccionamos el puerto B para leer.

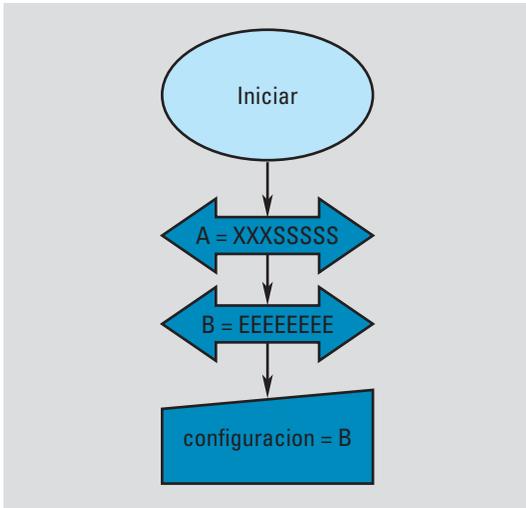


Cuando leemos un puerto, el resultado de la lectura puede ser guardado en el registro W o en un registro de usuario. Recordemos que el registro "W" es un registro de trabajo y el valor que éste contiene será sobrescrito en las siguientes instrucciones -líneas de código- que el microcontrolador ejecute. Dado que nosotros necesitamos almacenar el valor de lectura del puerto para, luego, realizar las comparaciones con los distintos códigos de configuración, guardamos el resultado de la lectura en un registro de usuario.

Para esto, marcamos la opción "En otro registro" y seleccionamos el registro de usuario "Configuración" que declaramos anteriormente.



Presionamos el botón ingresar para insertar el bloque en el diagrama de flujo.

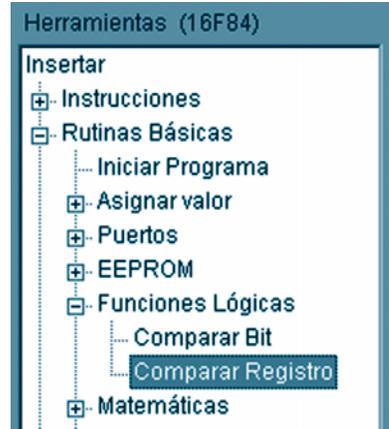


Comparar un registro

Una vez leído el estado del puerto, generamos la combinación de destellos según la configuración de la baliza.

Para esto, comparamos el valor del registro "Configuración".

Ingresamos a:



Primero, seleccionamos el registro que deseamos comparar -en este caso, el registro "Configuración"-.

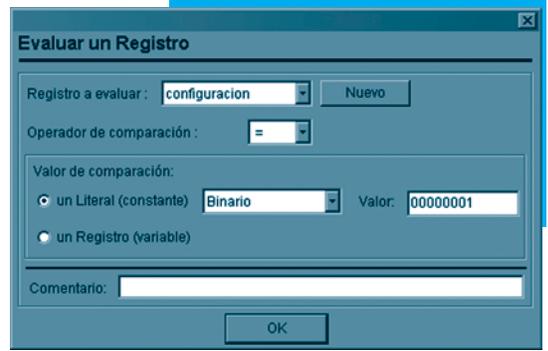
Luego, optamos por el operador de comparación -en este caso, operador "="-.



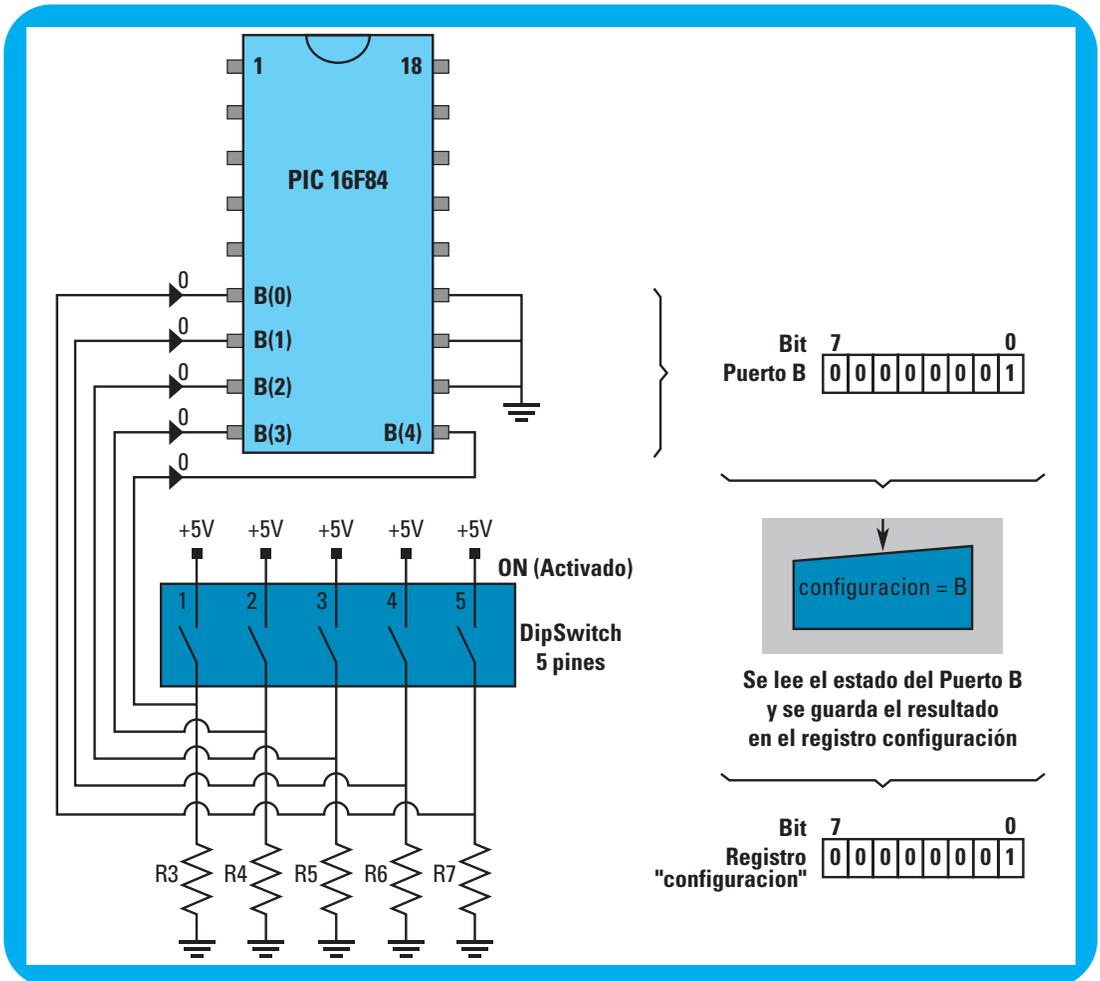
Un registro puede ser comparado con un valor literal -esto es, un valor constante- o con el valor que contiene otro registro -un valor variable-.

Para nuestro proyecto, vamos a comparar el valor que contiene el registro "Configuración" con un valor constante; por esto, seleccionamos la opción "Un literal".

Un valor literal puede ser ingresado en diferentes formatos: decimal, binario o hexadecimal. En este caso, nos resulta más cómodo utilizar el formato binario, ya que cada bit del registro representa, directamente, a cada pin del puerto. Realizamos la primera comparación con el valor Binario '00000001'.



El siguiente esquema ilustra el motivo por el cual realizamos la comparación con el literal '00000001'.



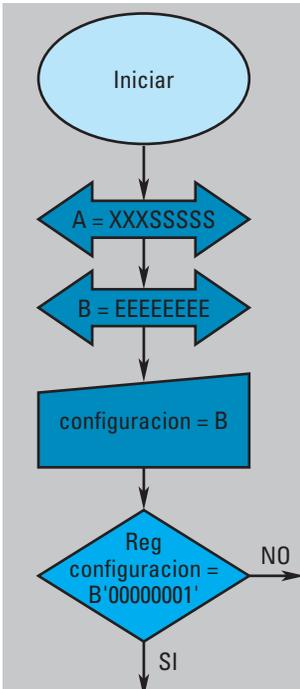
Con esta primera comparación que estamos realizando, evaluamos si la llave que se encuentra más hacia la derecha en la *Dip-switch* se encuentra activada.

Cuando realizamos una comparación, estamos realizando una pregunta cuya respuesta puede ser "Sí" o "No", por lo que el bloque correspondiente a la evaluación debe tener estas dos salidas.

En este caso, estamos preguntando: El registro 'configuracion', ¿es igual al valor binario '00000001'?

La respuesta a esta pregunta puede ser "Sí" o "No" ("verdadero" o "falso", respectivamente).

En el diagrama de flujo, la evaluación se representa con el símbolo de un rombo.



Luego de ingresar el bloque de evaluación, debemos ingresar los bloques correspondientes a sus dos salidas. Por defecto, el primer bloque que se inserta después de una evaluación siempre corresponde a la salida del "Sí" y se ubica debajo del bloque de comparación (rombo); el segundo bloque

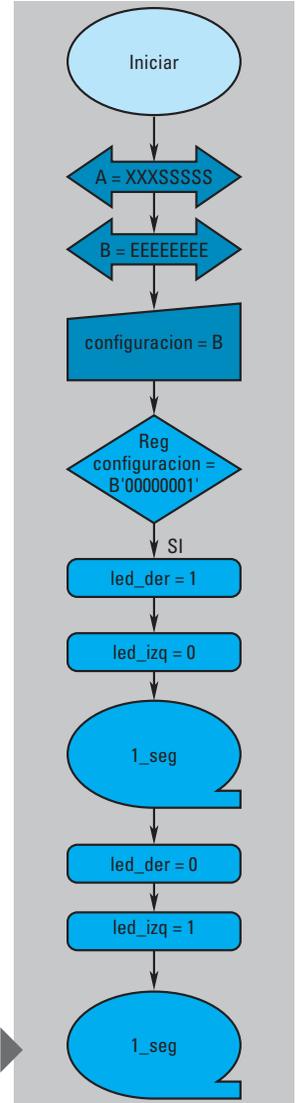
corresponde a la salida del "No" y se ubica hacia la derecha del bloque correspondiente al "Sí".

Para la salida del "Sí", vamos a arrastrar los bloques de asignación y temporización que conforman el destello del proyecto original. Para esto, seleccionamos el bloque de asignación donde encendemos el led derecho y lo arrastramos hasta ubicarlo debajo del bloque de evaluación de registro. Al seleccionar y mover el bloque de asignación, debe moverse el resto de los bloques que se encuentren por debajo de aquel.

Una vez ubicados los bloques, generamos un vínculo desde el bloque de evaluación de registro hacia el bloque de encendido del led derecho.

Note que, de manera automática, la flecha se inserta con el texto "Sí".

El diagrama de flujo queda conformado así:



Entonces, cuando seleccionemos la configuración 1, la baliza muestra un juego de luces intermitentes, alternando el encendido de los led cada segundo.

Ahora, debemos indicarle al microcontrolador qué debe realizar si la configuración seleccionada no es "00000001". Es decir, debemos volver al bloque de evaluación del registro "Configuración" y programar la salida correspondiente al "No".

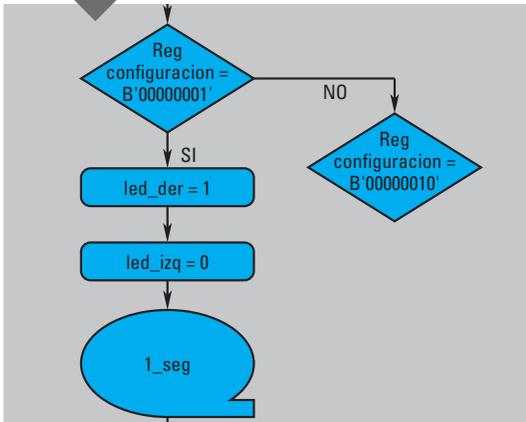
Si la configuración seleccionada no es "00000001", vamos a evaluar si la configuración seleccionada es la segunda opción, es decir, "00000010".

Para esto, insertamos un bloque de evaluación de registro donde comparamos el valor del registro "Configuración" con el valor binario "00000010".

Para insertar un bloque de comparación de registro ingresamos a:

Insertar/Rutinas básicas/Funciones lógicas/Comparar registro

El diagrama de flujo queda conformado de este modo:



Esta estructura lógica evalúa si el registro "Configuración" contiene el valor "00000001"; entonces, realiza la combinación de luces correspondiente. Si no es así, realiza otra evaluación, preguntando si el valor de la configuración seleccionada es "00000010".

De esta manera, podemos ir evaluando todas las combinaciones posibles. Para nuestro ejercicio, sólo vamos a evaluar los siguientes valores de configuración:

Configuración 1 = "00000001"

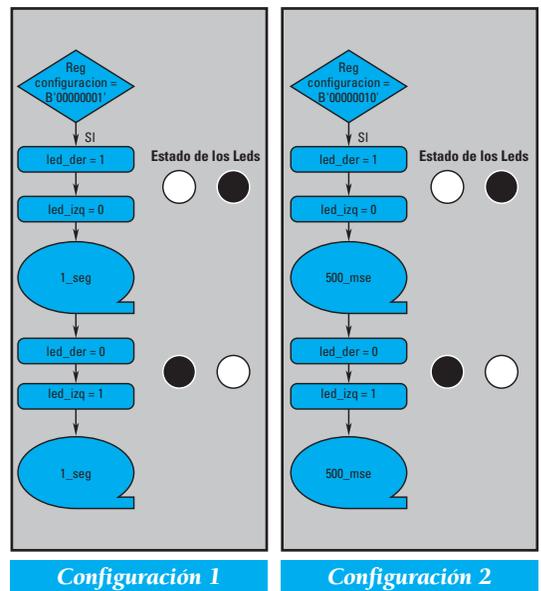
Configuración 2 = "00000010"

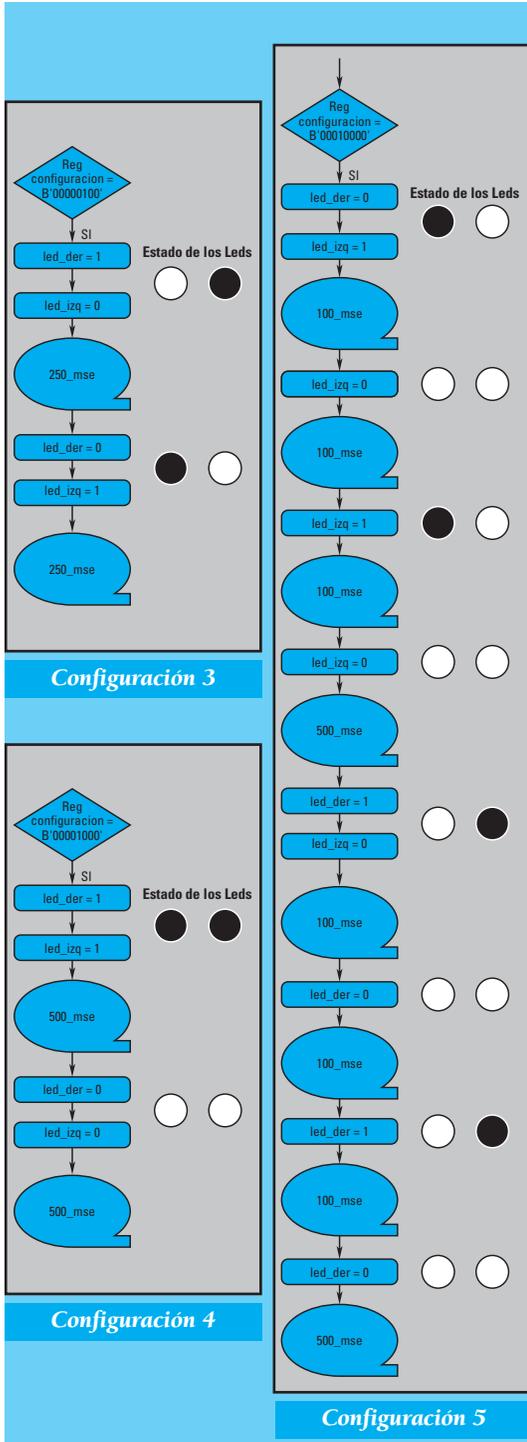
Configuración 3 = "00000100"

Configuración 4 = "00001000"

Configuración 5 = "00010000"

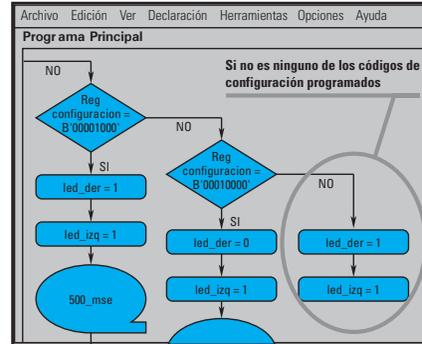
Y, a cada configuración, le asignamos una combinación de destello diferente.





Luego, programamos "qué debe hacer el microcontrolador" cuando seleccionamos una configuración que no corresponde a ninguna de las anteriormente mencionadas. Esto ocurre cuando la última comparación de registro da como resultado "No".

En este caso, dejamos encendidos los dos led de manera permanente.



Por último, cerramos el ciclo para que el micro vuelva a iniciar el

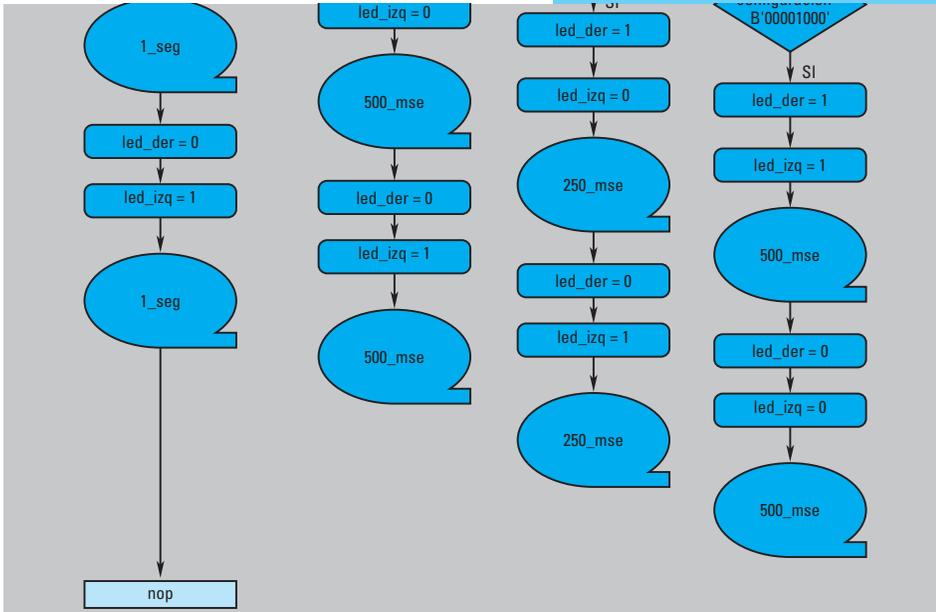
proceso. En nuestro ejercicio, el ciclo comienza con una nueva lectura del puerto B para determinar la configuración seleccionada. De esta manera, el microcontrolador puede detectar un cambio en la configuración de la baliza.

Para cerrar el ciclo, utilizamos un bloque de instrucción "NOP" para reunir los vínculos de cada una de las ramas lógicas. La instrucción "NOP" -no opera- es una instrucción del lenguaje *assembler*: no realiza ninguna operación y consume un ciclo de reloj.

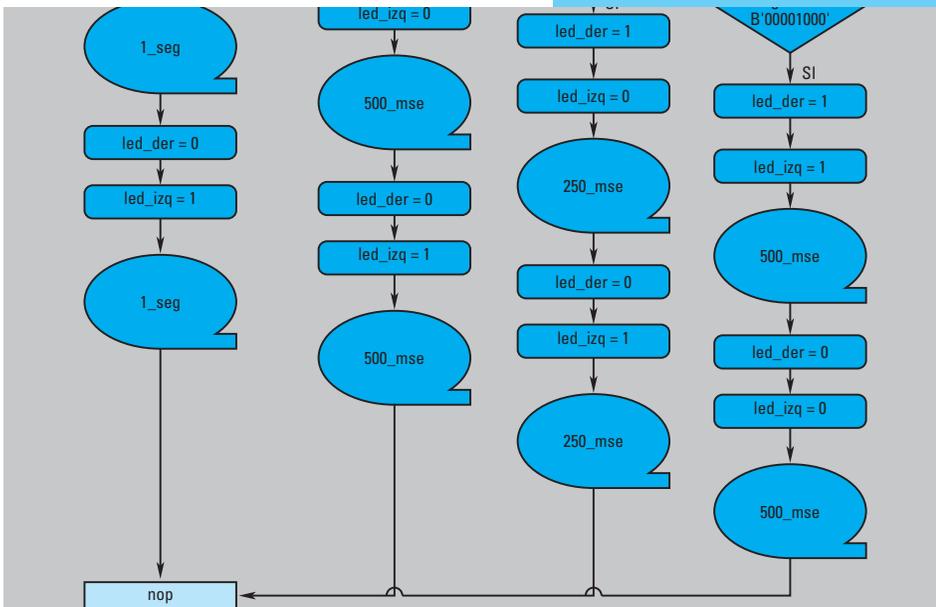
Para insertar un bloque de instrucción NOP ingrese a:



Insertamos un bloque NOP al final de los bloques de la línea lógica de la configuración 1.

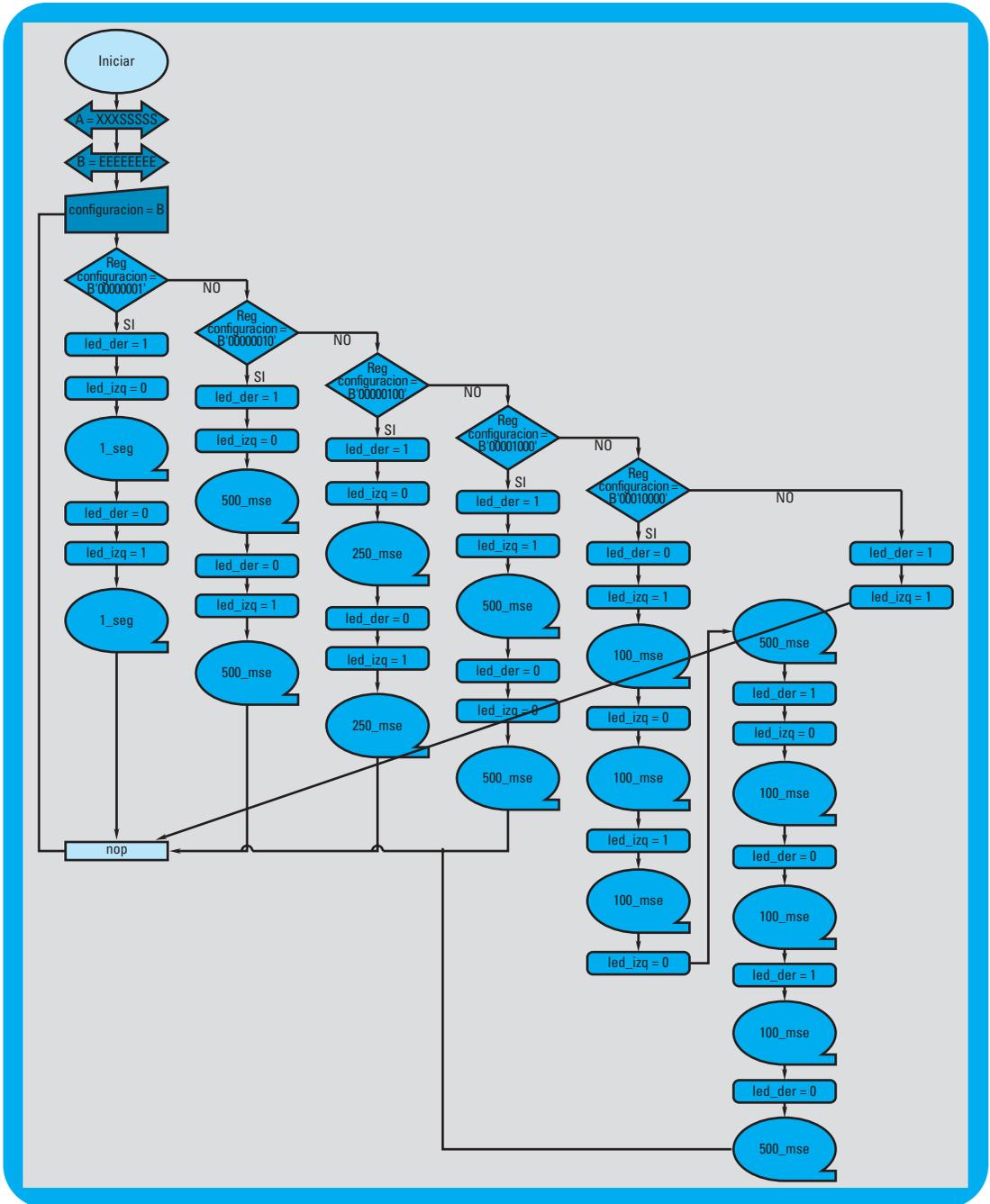


Luego, vinculamos los últimos bloques de cada línea lógica al bloque NOP.

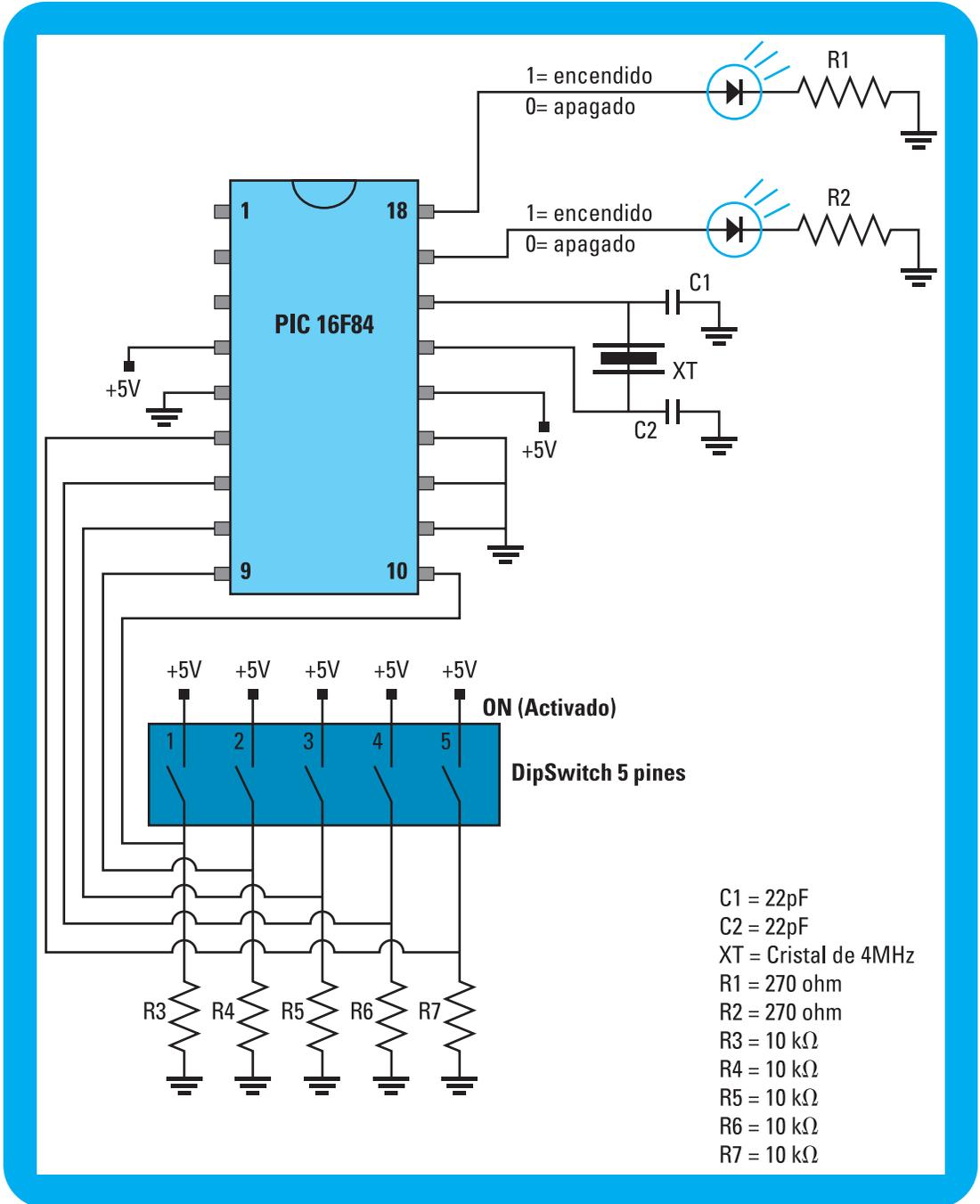


Por último, cerramos el ciclo vinculando el bloque NOP al bloque de lectura del puer-

to, en el inicio del programa. La estructura del programa queda así:



Por último, generamos el archivo ASM, lo compilamos y lo grabamos al micro.



Contador digital

En este ejercicio vamos a desarrollar un contador digital de 00 a 99 que visualiza el estado de la cuenta en 2 displays de 7 segmentos y cuya señal de entrada, para incrementar el estado de la cuenta, proviene de una barrera infrarroja.

Al recibir una señal desde un receptor, el sistema incrementa el estado de la cuenta mientras muestra los datos en 2 display de 7 segmentos tipo cátodo común.

Este proyecto permite a los alumnos:

- Integrar la tecnología de los transmisores infrarrojos.
- Mostrar datos en display de 7 segmentos.



Para desarrollar el contador digital, las tareas con el **entorno de desarrollo para programación de microcontroladores PIC -circuitos integrados programables-** son:

- Utilizar el transmisor-receptor infrarrojo.
- Utilizar el display de 7 segmentos.
- Desarrollar el contador digital.



Utilizar el transmisor-receptor infrarrojo

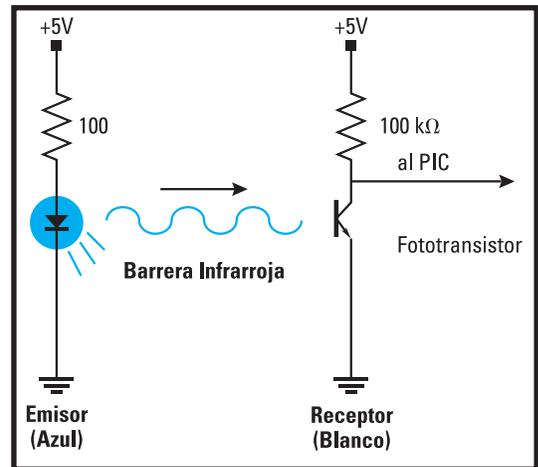
Para este proyecto tecnológico utilizamos un transmisor infrarrojo -TIR- y un receptor infrarrojo -RIR- que creen una barrera infrarroja.

Estos transmisores y receptores infrarrojos son los mismos que podemos encontrar en el control remoto de los televisores.

Tanto el TIR como RIR se parecen, físicamente, a un diodo led; la diferencia es que el TIR es de color azul, y el RIR es de color blanco y un poco más pequeño.

El TIR es un diodo led que emite una luz infrarroja mientras que el RIR es un fototransistor de luz infrarroja.

A continuación, le mostramos el esquema de conexionado:



Además, utilizamos dos display de 7 segmentos -cátodo común- para visualizar el estado de la cuenta.

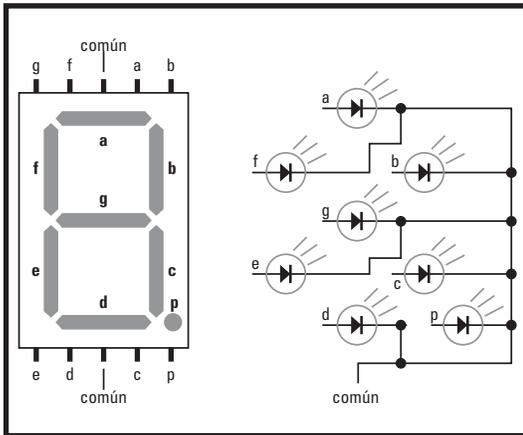
Cada segmento está formado por un led conectado a un pin común:

- Cuando el pin común es el pin negativo de los led, el display se llama cátodo común.
- Cuando el pin común corresponde al pin positivo de los led, se llama ánodo común.

En nuestro proyecto, vamos a integrar dos display de tipo "cátodo común".

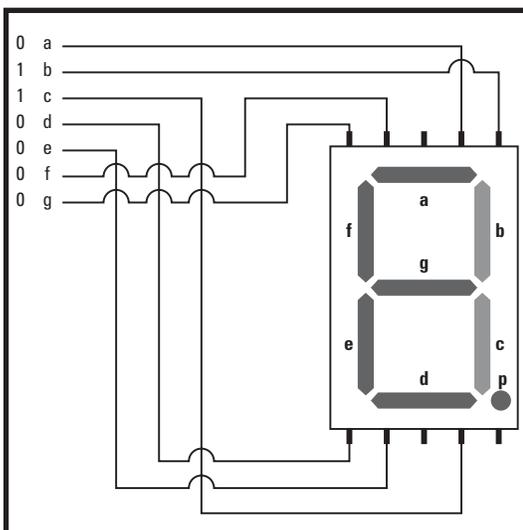
Utilizar el display de 7 segmentos

Éstos son el esquema de conexionado y su esquema electrónico:



Para visualizar un dato en un display, enviamos la combinación de "ceros y unos" correspondiente al dato que queremos representar.

Por ejemplo, para mostrar el valor "1", asignamos la siguiente combinación binaria:

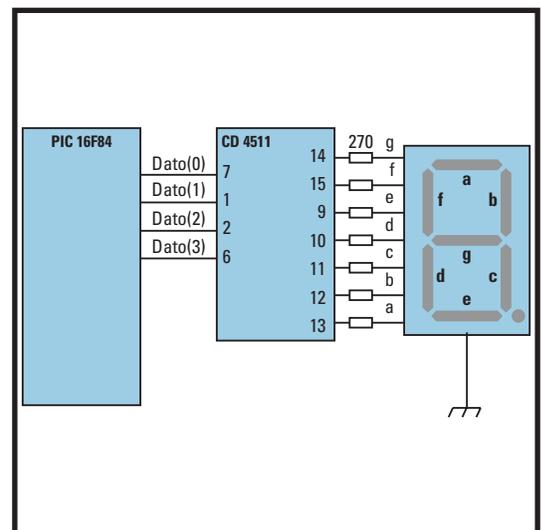


Observe que el código binario para mostrar en el display el valor 1 no corresponde con el valor "1" ("00000001"). Por esto, generamos la combinación binaria de cada valor a mostrar en el display. Este código nos insume 7 pines del microcontrolador.

El codificador CD4511 convierte los valores de formato binario a la combinación binaria correspondiente en el display. Desde el microcontrolador, enviamos al CD4511 el número a mostrar; entonces, el codificador genera el código correspondiente, de manera automática.

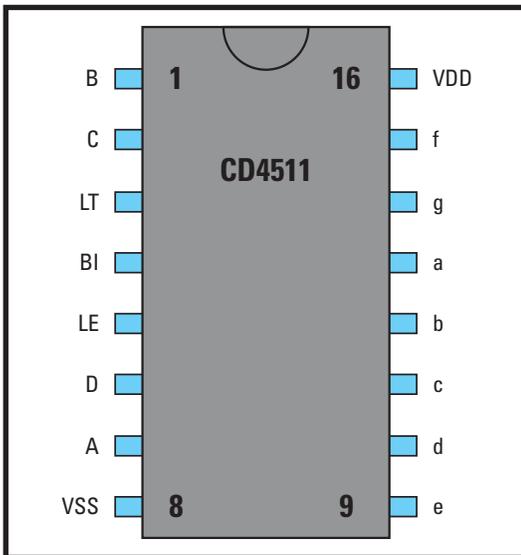
Dado que enviamos al CD4511 los valores a mostrar en formato binario, esto sólo nos insume 4 pines del microcontrolador -con 4 pines podemos formar valores entre 0 y 15-.

De esta manera, al utilizar el CD4511, estamos ahorrando el proceso de codificación del valor a mostrar y, además, ahorramos 3 pines del microcontrolador.



La descripción de los pines del CD4511 es:

Pin	Descripción
VDD	Alimentación positiva, +5 V.
VSS	Masa.
A, B, C, D	Entradas digitales del dato a visualizar. Corresponden a los pesos binarios 1, 2, 4, respectivamente.
LT	Test de segmentos. Si se pone a 0, todos los segmentos se encienden. En uso normal debe estar puesto a 1.
BI	Habilitación de brillo. Si se pone a 0, el display se apaga.
LE	Modo memoria. Si se pone a 0, está en operación normal. Si se pone a 1, trabaja en modo memoria.
a, b, c, d, e, f, g	Conexión a display.



Como vimos anteriormente, un display puede mostrar un solo dígito -un número entre 0 y 9-.

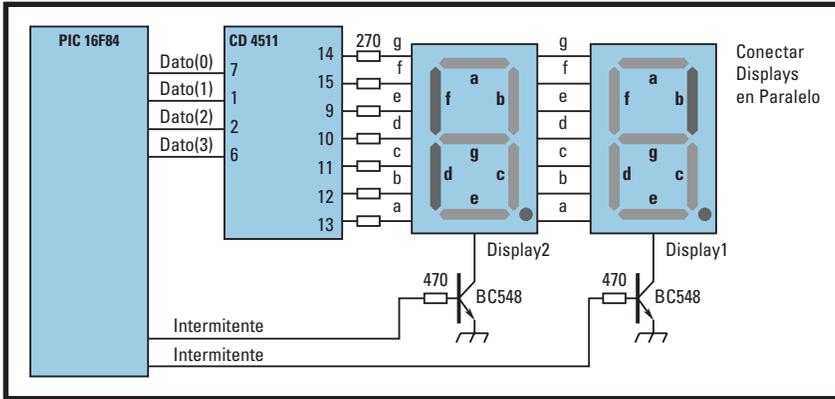
Y, como cada codificador CD4511 insume 4 pines del microcontrolador, esto significa que, si queremos visualizar un valor de 2 cifras, debemos utilizar 2 CD4511, uno por

cada display -con lo cual, necesitamos 8 pines del microcontrolador-. Esto sería muy costoso en términos económicos y, además, utilizaríamos muchos pines del microcontrolador; por esto, cuando necesitamos visualizar datos de más de un dígito, implementamos el **método del multiplexado**.

Supongamos que necesitamos visualizar un dato de dos cifras; por ejemplo, el número 36. Llamamos "Display 1" al display correspondiente a las unidades; al display correspondiente a las decenas lo llamamos "Display 2".

Lo primero que hacemos es descomponer el dato a visualizar en unidades y decenas; esto se realiza mediante un cálculo matemático llamado **conversión a BCD**.

El sistema de numeración BCD -decimal codificado en binario- procesa los valores de las unidades, decenas, centenas... en valores por separado.



Una vez inicializados los puertos, declaramos:

- el registro "contador" -donde vamos a llevar el estado de la cuenta- y
- los registros "uni", "dec" y "cen" -donde vamos a

guardar el resultado de la conversión del registro "contador" de decimal a BCD, para visualizar los datos en los display-.

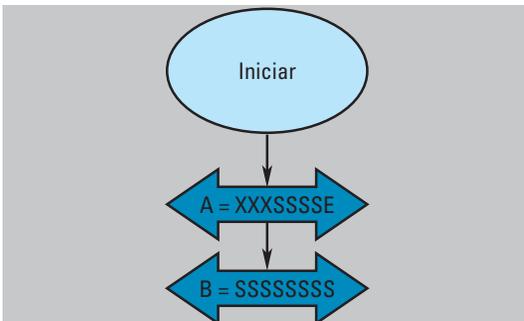
Desarrollar el contador digital

Ingresamos al Entorno de desarrollo para programación de microcontroladores PIC-circuitos integrados programables-.

Como siempre, comenzamos configurando los puertos:

- En el pin 0 del puerto A, vamos a conectar el sensor infrarrojo.
- En el puerto B, vamos a conectar el CD4511 y los pines comunes de los 2 display.

Por esto, configuramos el pin 0 del puerto A como entrada y todo el puerto B como salida.



Declaración de Registros

Bancos 0 y 1

Banco 0		Banco 1	
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h			87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	contador		8Ch
0Dh	uni		8Dh
0Eh	dec		8Eh
0Fh	cen		8Fh
10h			90h

Referencias

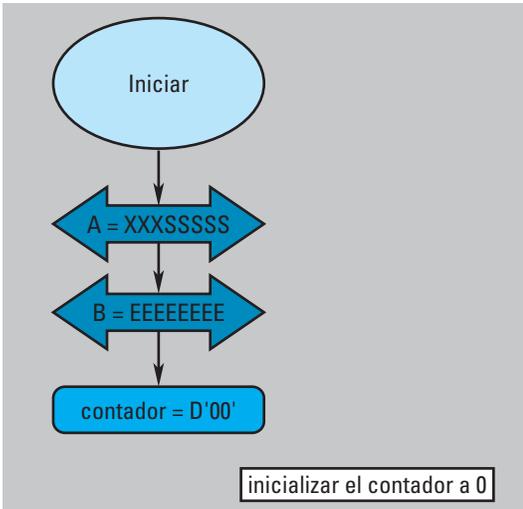
Registros Predefinidos por el Microcontrolador.

Registros de Usuario.

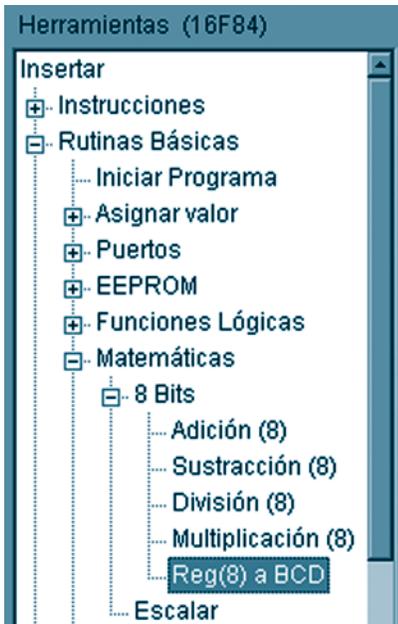
Registros deshabilitados.

Aceptar

Luego, inicializamos el registro contador a 00. Esto es para asegurarnos que el contador siempre inicie la cuenta desde el valor 0.



A continuación, nos centramos en el estado del contador en los displays de 7 segmentos. Para esto, realizamos la conversión a BCD, mediante un cálculo matemático, ingresando a:



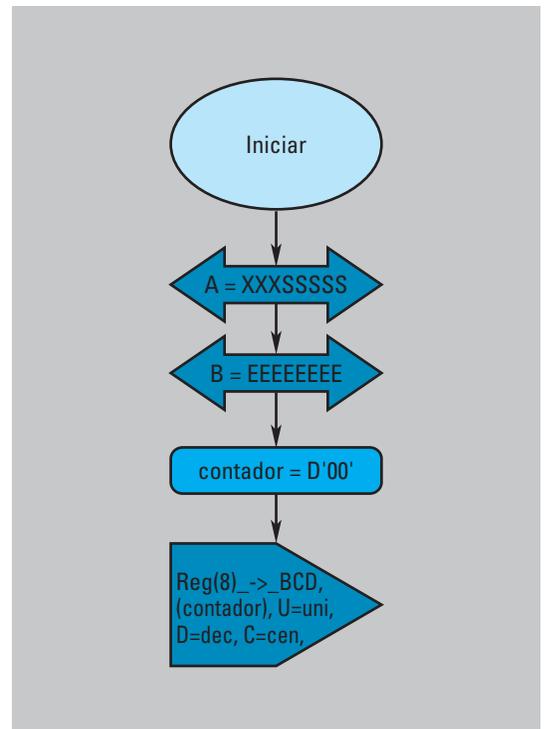
Cargamos la pantalla "Convertir registro (8 Bits) a BCD".

En el campo "Registro a descomponer", seleccionamos el registro "Contador".

Y guardamos el resultado de la conversión en los registros "uni", "dec", y "cen".



Presionamos el botón "Ingresar", para insertar el bloque en el diagrama de flujo.



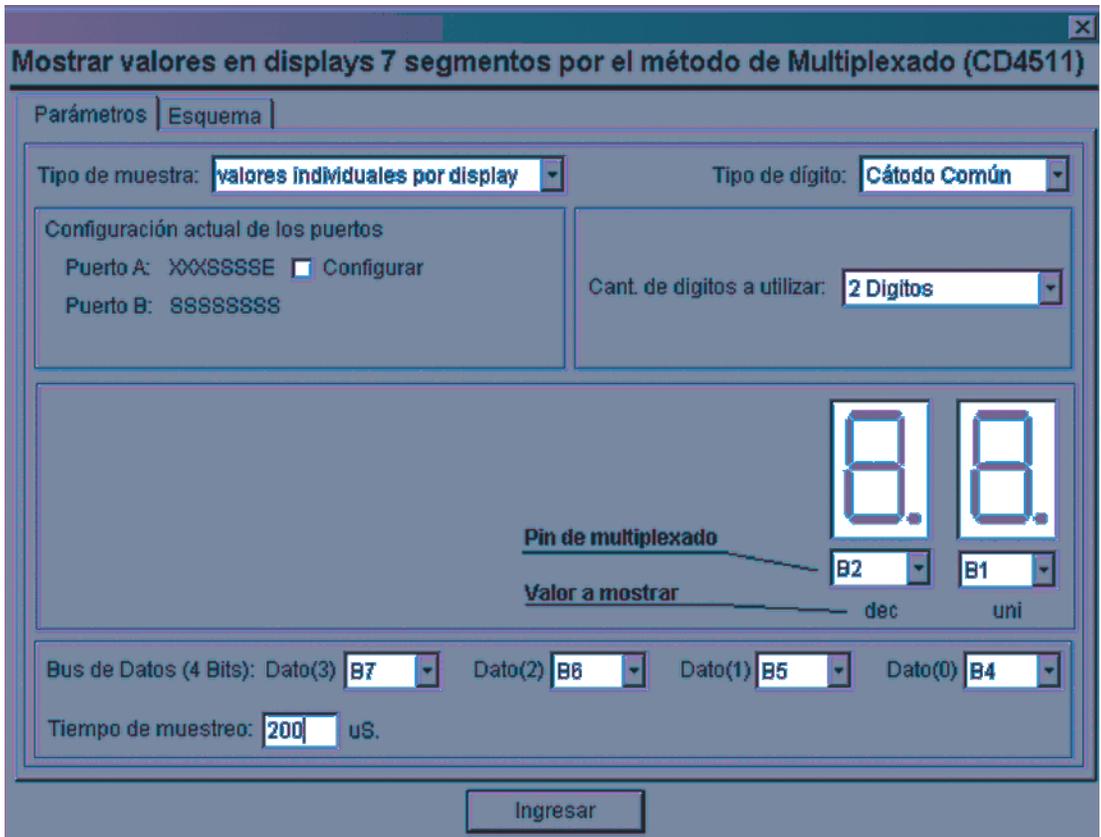
Una vez realizado el cálculo de conversión a BCD, vamos a mostrar los datos en los displays.

Para esto, ingresamos a:

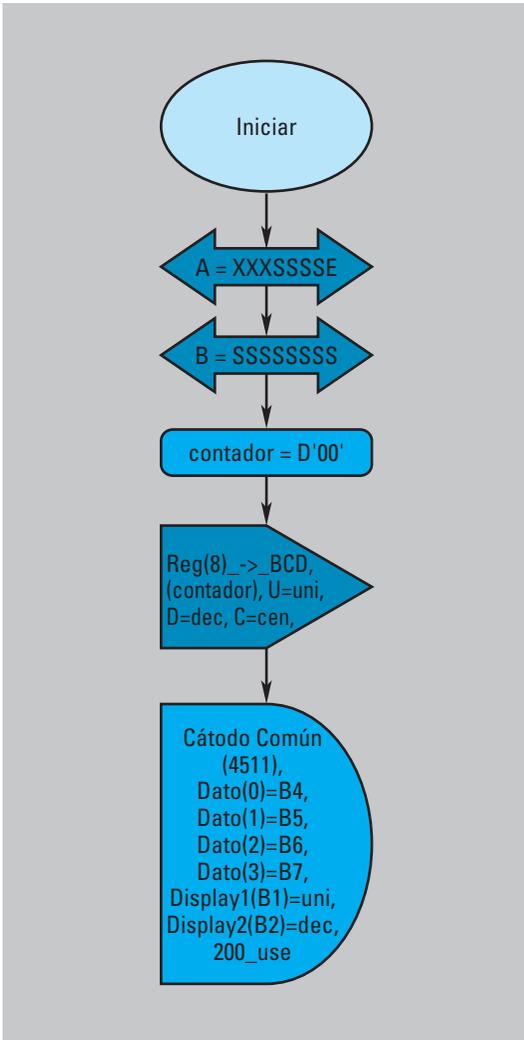


Configuramos la visualización de los displays de la siguiente manera:

Tipo de muestra:	Valores individuales por display.
Tipo de dígito:	Cátodo común.
Cantidad de dígitos a utilizar:	2.
Valor a mostrar en los display:	Registros "dec" y "uni".
Pines de multiplexado:	"B2" y "B1".
Bus de datos:	"B7", "B6", "B5" y "B4".
Tiempo de muestreo:	200 μ s.



Cuando presionamos el botón "Ingresar", el bloque se inserta en el diagrama de flujo.



El bloque de visualización en display realiza la muestra de datos durante un solo ciclo; es decir, muestra el valor de las unidades y de las decenas durante 200 µs para cada valor. Esto no es suficiente para visualizar los valores, ya que el ojo humano no es capaz de percibir la imagen por tan

poco tiempo.

Por esto, necesitamos generar una serie de ciclos para que el microcontrolador se quede repitiendo el proceso de visualización de datos la cantidad de veces que sea necesario para poder visualizar correctamente los valores.

Declaramos dos registros llamados "ciclos" y "ciclos1", los que son utilizados para llevar cuenta de la cantidad de veces que se ha repetido el proceso.

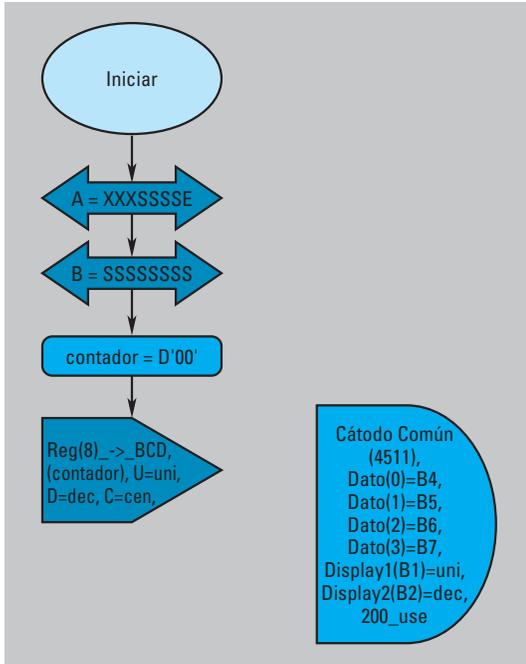
Lo que vamos a hacer es generar dos bucles anidados -un bucle dentro de otro-:

- el primer bucle, de 200 ciclos,
- el segundo bucle de 3 ciclos.

Dado que el segundo bucle está dentro del primero, por cada ciclo del primer bucle se ejecutan los tres ciclos del segundo bucle.

De esta manera, estamos ejecutando 600 veces el bloque de visualización de datos, lo que resulta suficiente para visualizar los datos correctamente. De todas formas, si fuera necesario, podemos variar la cantidad de veces que se repite la ejecución del bloque de visualización, lo que modifica directamente el tiempo en que el microcontrolador permanece mostrando los datos antes de seguir adelante con el resto del programa.

Para esto, es necesario hacer a un lado el bloque de visualización de datos, para insertar los bloques correspondientes a los bucles.

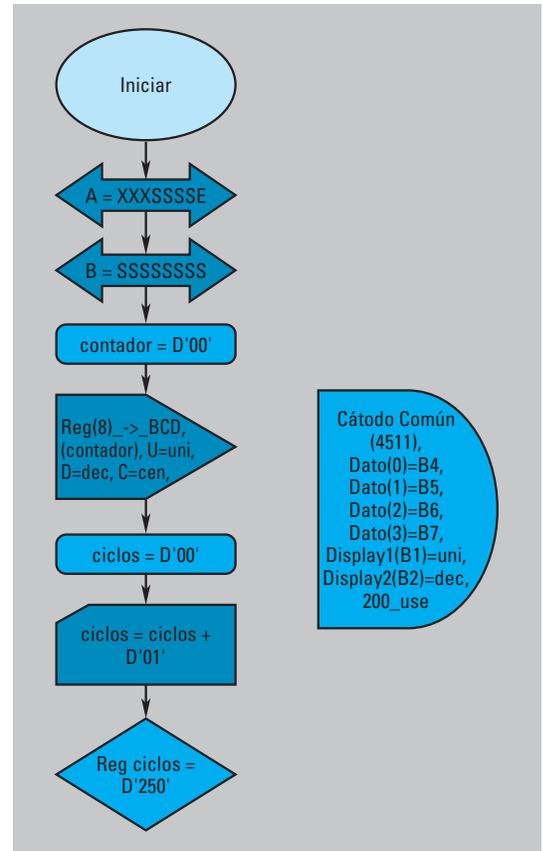


Declaramos los registros "ciclos" y "ciclos" en los que llevamos la cuenta de la cantidad de veces que se repiten los ciclos.



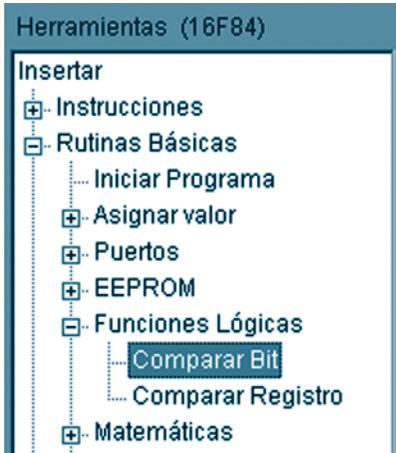
Para generar los bucles, debemos inicializar los registros. El primer bucle realiza un ciclo que se repite 250 veces; por esto, inicializamos el registro "ciclos" a 00 y, luego, lo incrementamos en 1 mediante el cálculo matemático de adición.

Una vez incrementado el registro "ciclos", evaluamos si ya llegó al valor 250.

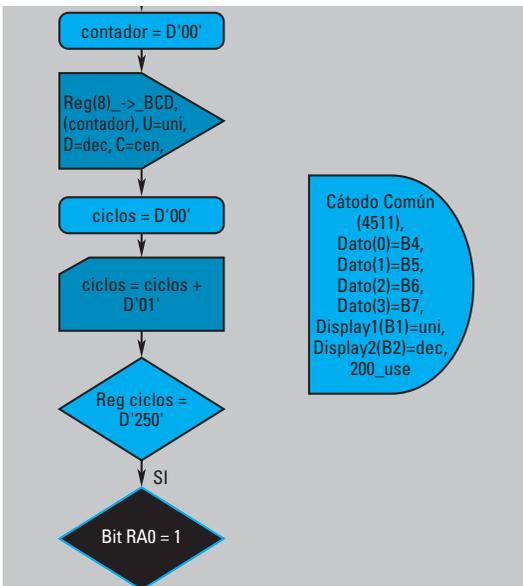
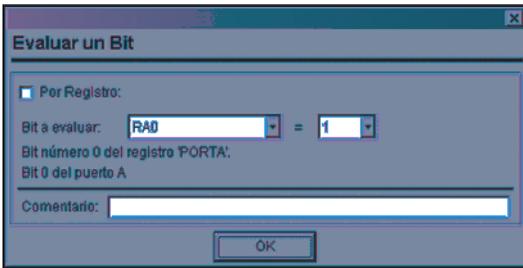


Cuando el registro "ciclos" llega al valor "250", es porque los dos bucles ya terminaron su ejecución. Cuando esto ocurre, el siguiente paso consiste en leer el estado del sensor conectado a la entrada A0. Para esto, evaluamos el estado del BIT A0.

Ingresamos a:



Seleccionamos el pin A0 para ser evaluado y el valor de comparación = "1".

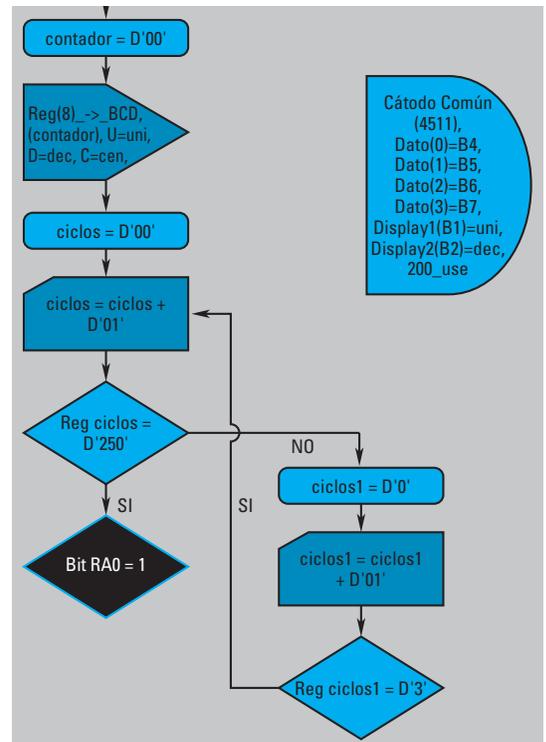


Insertamos el bloque de lectura del pin A0 antes de terminar de diseñar la estructura de los bucles -porque, el primer bloque que insertamos a la salida de una evaluación (símbolo rombo) corresponde al "Sí"-.

Para terminar el segundo bucle, seleccionamos el bloque de evaluación del registro "ciclos" e ingresamos los bloques correspondientes a la salida del "No".

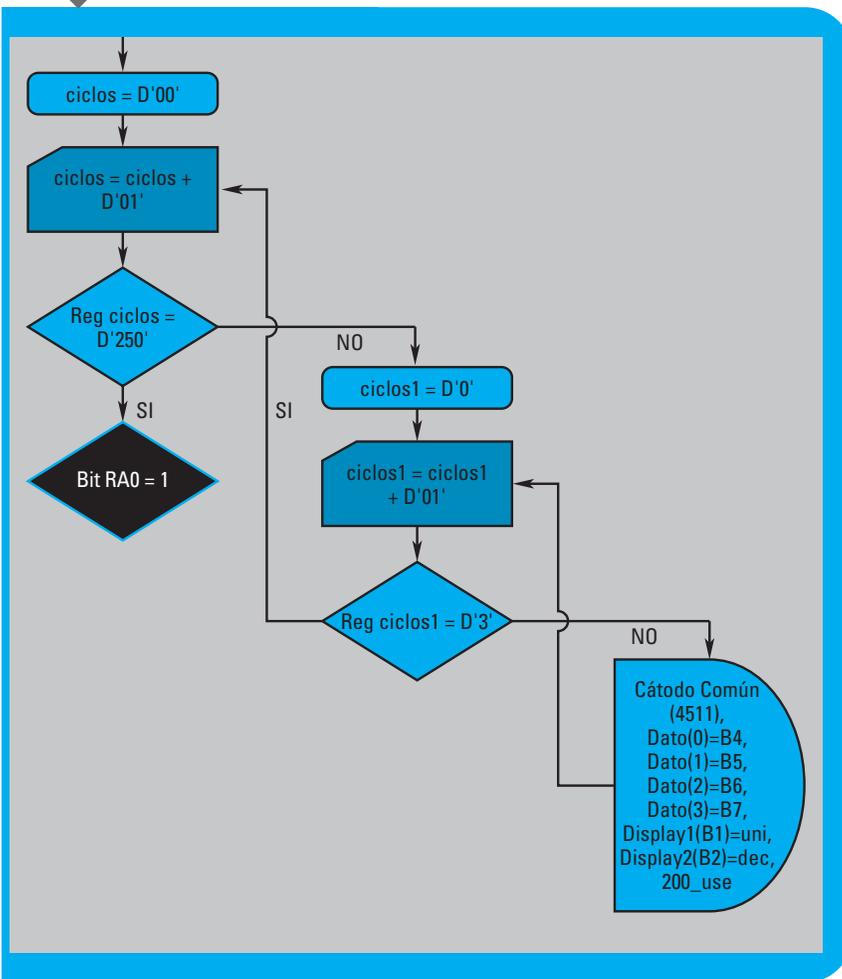
Generamos la misma estructura lógica que para el registro "ciclos". Es decir, inicializamos el registro "ciclos1" a 0; luego, lo incrementamos en 1; y, por último, evaluamos si llegó a 3.

Si el registro llega a 3, entonces se vuelve al bloque de incremento del registro "ciclos".



Ahora, falta insertar la salida correspondiente al "No" de la evaluación del registro "ciclos1". Si el registro no llega a 3, ejecuta el bloque de visualización de datos en los display y, después de mostrar los datos, vuelve a incrementar el valor del registro "ciclos1".

La estructura lógica queda de la siguiente manera:



Así, quedan formados los dos bucles "anidados": el primero de 250 ciclos y el

segundo de 3 ciclos.

El resultado es que el bloque de visualización de datos se ejecuta 750 (250*3) veces, antes de seguir con el resto del programa.

Por último, cuando se terminan de ejecutar los bucles de visualización, evaluamos el estado del sensor conectado al pin A0 mediante el bloque de evaluación de bit

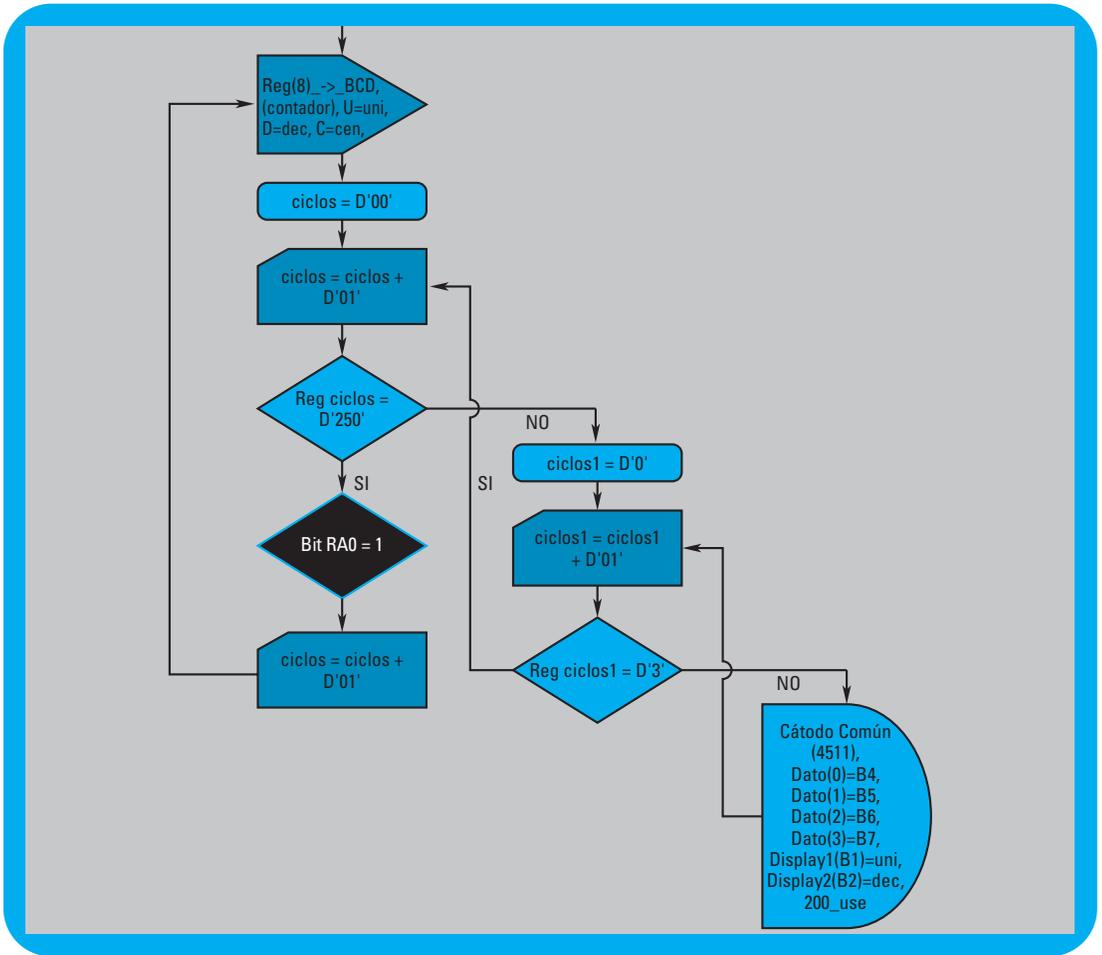
-rombo color magenta en el soft-

Este bloque pregunta: El pin A0, ¿es igual a 1?

Si el resultado de esta evaluación es verdadero, la ejecución del programa toma el camino del "Sí". De lo contrario, toma el camino del "No".

Cuando un objeto interrumpe la barrera infrarroja del sensor, se envían 5 V al pin A0, lo cual es interpretado por el microcontrolador como un "1" lógico.

Si esto ocurre, el microcontrolador debe incrementar en 1 el estado del contador.



Si el estado del sensor es 0, el microcontrolador se queda visualizando los datos, pero sin realizar ningún bucle.

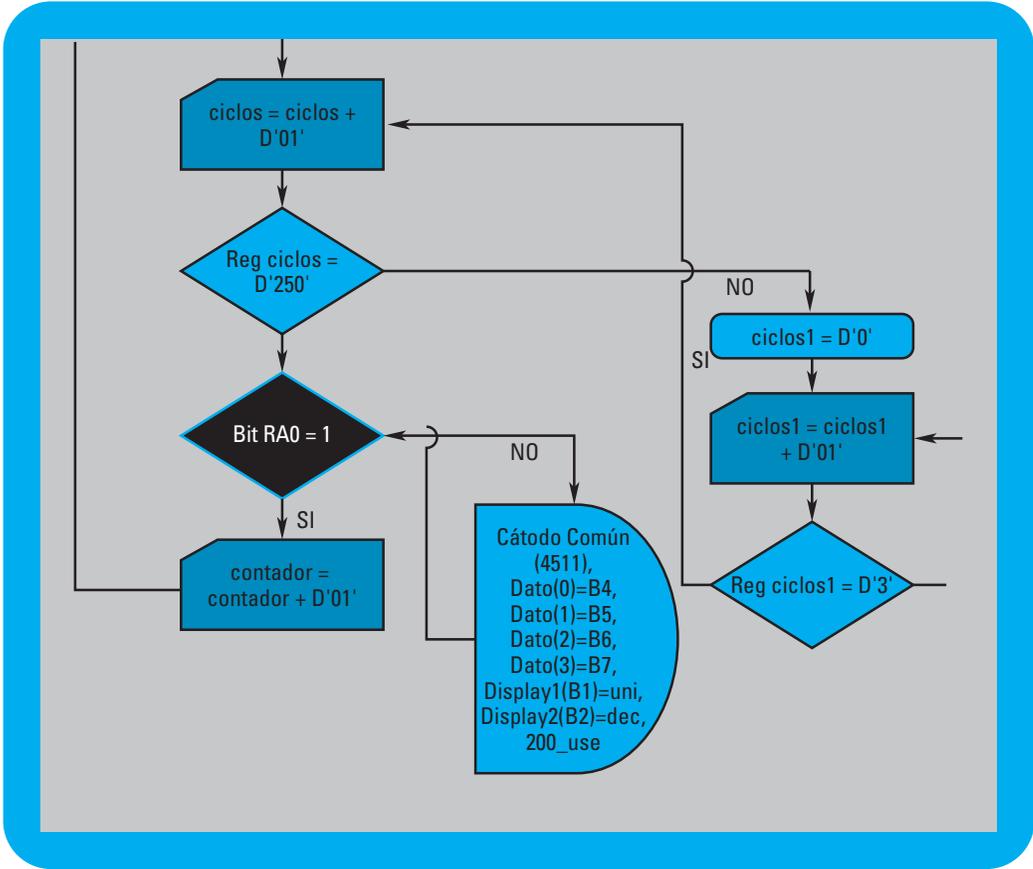
Lo desarrollamos de este modo porque el microcontrolador debe estar supervisando continuamente el estado del sensor: Si el micro realiza los dos bucles, el tiempo en que se encuentra trabajando dentro de los bucles, no supervisa el estado del sensor. En cambio, de esta manera nos aseguramos que el micro detecte cualquier objeto de manera inmediata.

Además, cuando detecta un objeto, el microcontrolador incrementa el estado de la cuenta y, entonces, si ingresa a los bucles para visualizar el dato. Esto se realiza para darle un tiempo al objeto para que salga de la zona de la barrera infrarroja (constituyendo una especie de "antirrebote" por software).

Si no ingresamos a los bucles de visualización cada vez que se detecta un objeto, el microcontrolador vuelve a evaluar el estado del sensor antes de que el objeto salga de la

zona de la barrera infrarroja. De esta manera, incrementa el estado del contador y realiza una cuenta errónea.

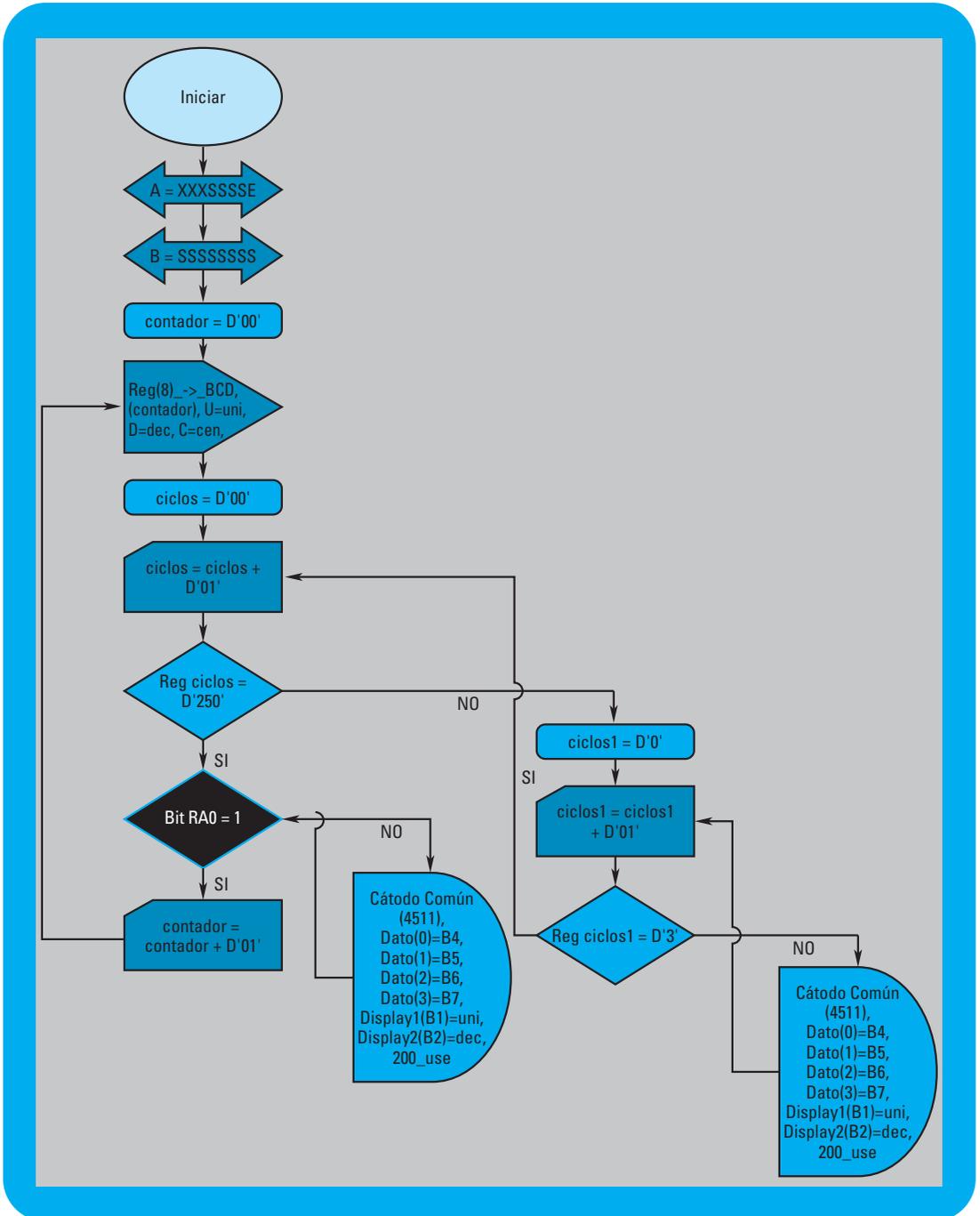
Observe la siguiente estructura lógica para analizar cómo el microcontrolador se queda "esperando" a que se detecte un objeto, mientras se visualiza el estado actual de la cuenta.



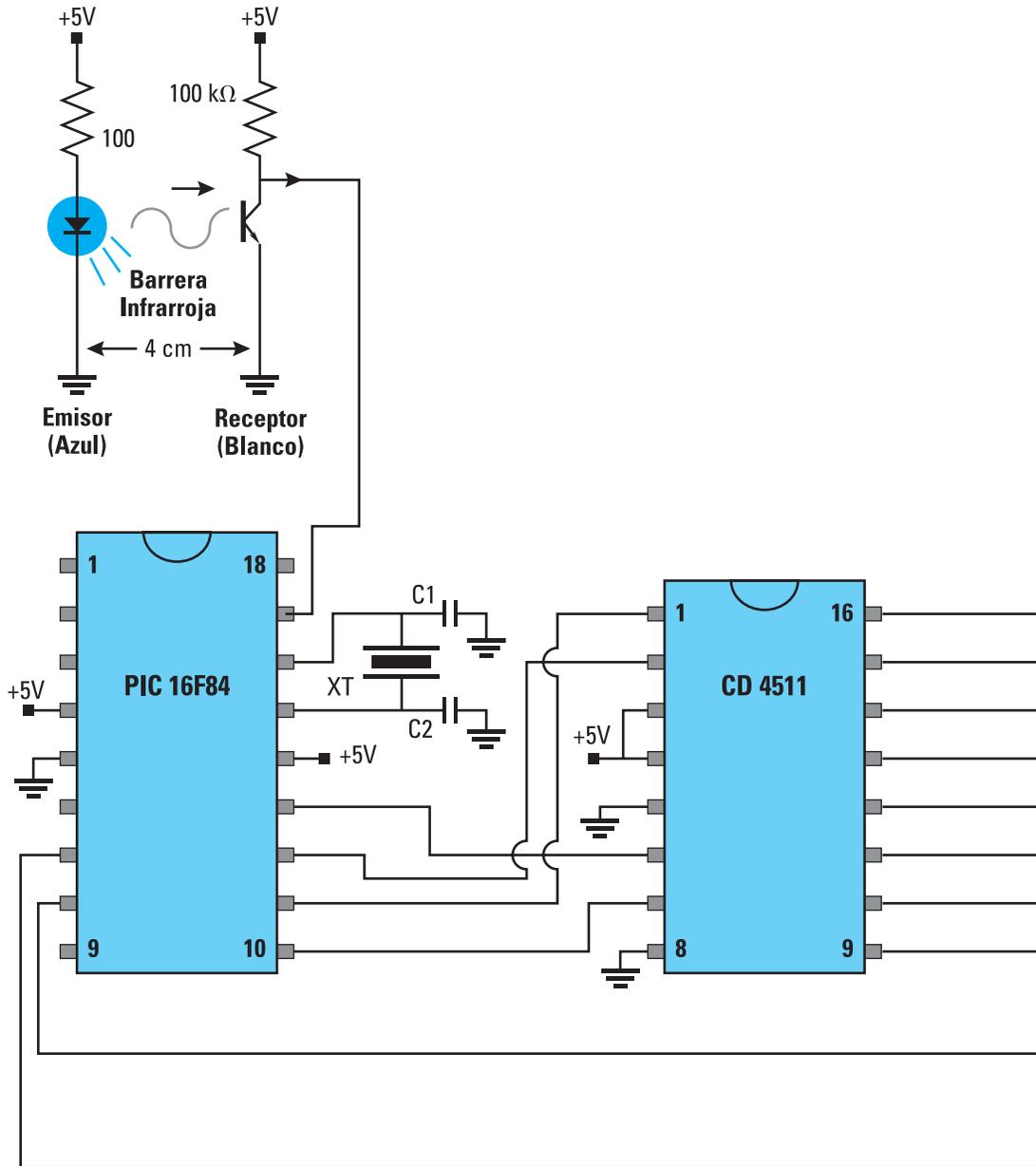
De esta manera, queda terminado el contador digital.

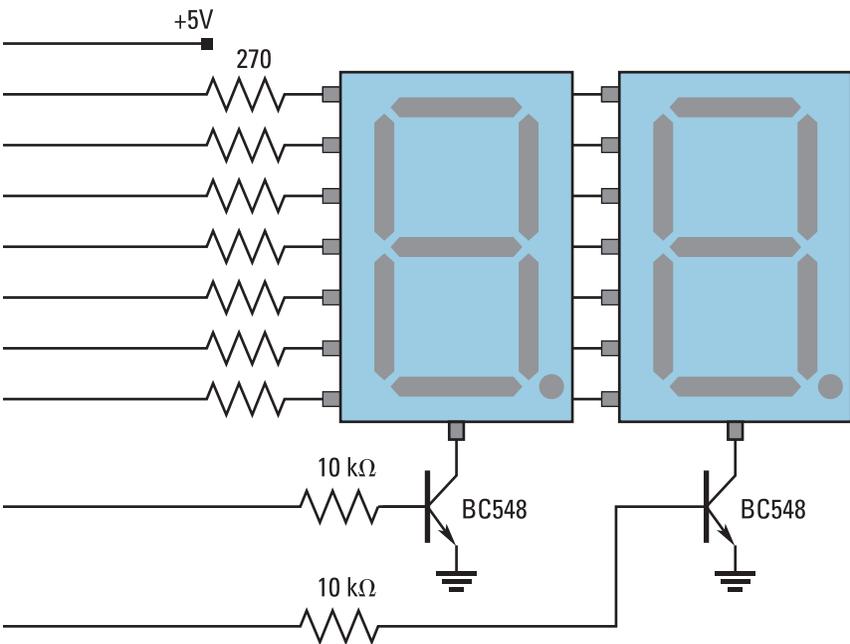
Sólo resta generar el código *assembler*, compilar el programa y grabar el microcontrolador.

El diagrama de flujo completo es éste:



Y, éste, el esquema electrónico:





4. LA PUESTA EN PRÁCTICA

Esta parte final de nuestro módulo de capacitación contiene un cuadernillo para la evaluación del recurso didáctico que le presentamos y, de las experiencias didácticas y contenidos propuestos a partir de él:

Esta evaluación tiene dos finalidades:

- Brindarle a usted, como docente que utiliza este material, la oportunidad de documentar el seguimiento de las actividades que realice con sus alumnos, a partir de nuestras propuestas y, en función de esta memoria de acciones, propiciar una reflexión acerca de los cambios, mejoras o enriquecimiento de su propia tarea de enseñanza.
- Obtener de su parte, como usuario de este material, información sobre todos los aspectos en torno a los cuales gira la propuesta.

Para este relevamiento de información, usted encontrará, a continuación, una serie de cuestionarios organizados básicamente en tablas o matrices para completar. Con los datos que usted exprese en ellos esperamos tener una realimentación que nos permita mejorar todos los componentes de la serie de publicaciones “Recursos didácticos” y enriquecerla con propuestas o documentación complementaria para aquellos docentes que planteen iniciativas, interro-

gantes o dificultades específicas con relación a la construcción del recurso didáctico, a las actividades de aula, a los contenidos científicos y tecnológicos, a la metodología de enseñanza, a los procedimientos incluidos, a la información sobre materiales y a otros aspectos.

Dada la importancia que esta información de retorno tiene para nuestro trabajo de seguimiento, mejora y actualización, le agradecemos que nos remita el cuadernillo con todas las observaciones, comentarios o sugerencias adicionales que nos quiera hacer llegar. Para ello puede remitirnos una copia, a través de correo postal, a

Área de Monitoreo y Evaluación –CeNET–
Oficina 112
Saavedra 789. C1229ACE.
Ciudad Autónoma de Buenos Aires.
República Argentina.

O, si lo prefiere, solicitarnos el archivo electrónico de las páginas que siguen a evcenet@inet.edu.ar, enviándonos la versión digitalizada de sus respuestas a través del mismo correo electrónico.

Desde ya, muchas gracias.

Identificación del material:

Las dimensiones que se consideran para la evaluación del módulo de capacitación y del recurso didáctico son:

- | | |
|--|--|
| 1. Nivel educativo | 5. Documentación |
| 2. Contenidos científicos y tecnológicos | 6. Otras características del recurso didáctico |
| 3. Componentes didácticos | 7. Otras características del material teórico |
| 4. Recurso didáctico | 8. Propuestas o nuevas ideas |

1. Nivel educativo en el que trabajó el material:

Nivel educativo	EGB 2	EGB 3	Polimodal (*)			Escuela técnica (*)						Trayecto técnico- profesional (*)	Formación profesional (*)	Otra (*)	
			1	2	3	1	2	3	4	5	6				
Nivel en el que usted lo utilizó															

Asignatura/espacio curricular en el que usted lo utilizó:.....

(*) Por favor, indique la modalidad, la orientación, la especialidad, etc.

2. Contenidos científicos y tecnológicos trabajados:





3. Componentes didácticos:

3.1. Testimonios (situaciones problemáticas) presentados en el material

	Sí	No	Otro ¹
a. ¿Le resultaron motivadores para iniciar las actividades propuestas?			
b. ¿Le facilitaron el desarrollo de contenidos curriculares que usted tenía previstos?			
c. A su criterio, ¿están vinculados con el recurso didáctico que se le propone desarrollar?			
d. ¿Le facilitan la organización de situaciones didácticas para el trabajo de los contenidos científicos y tecnológicos propuestos?			
e. El nivel de las situaciones problemáticas que se plantean, ¿es el adecuado al nivel educativo para el que está previsto?			
f. En caso negativo, ¿permiten adecuaciones para ser trabajados en el nivel educativo de sus alumnos o en otro nivel educativo?			
g. Los testimonios iniciales, ¿permiten generar diferentes soluciones (soluciones tecnológicas o didácticas)?			

En caso que su respuesta sea negativa (en cualquier ítem), le pedimos que nos indique por qué (señale el número del ítem a que corresponde su comentario).....

.....

.....

.....

.....

.....

.....

.....

Otro (indique el ítem al que corresponde el comentario):

.....

.....

.....

.....

.....

.....

¹ Utilice esta opción para indicar que agregará comentarios al final de este sector de la matriz.

3.2. Estrategias

A partir de la utilización de las propuestas de trabajo en el aula contenidas en el material y del recurso didáctico con el que se asocian, le solicitamos que nos indique (tomando como referencia su forma de trabajo anterior a disponer del material), cómo resolvió las actividades consignadas en la tabla siguiente:

3.2.1. Contextualización de la estrategia didáctica Con respecto a su forma habitual de trabajo, usted logró:	Mejor	Igual	No aplicado ²	Incorporado ³
a. Determinar las capacidades, habilidades, conocimientos previos necesarios para iniciar las actividades propuestas.				
b. Organizar, asociar, relacionar los conocimientos científicos y tecnológicos para resolver un problema tecnológico.				
c. Recortar (identificar) los contenidos científicos y tecnológicos a trabajar con sus alumnos para el desarrollo de un sistema/producto tecnológico como el propuesto por el material.				
d. Vincular estos conocimientos con los saberes previos de los alumnos.				
e. Establecer la secuencia adecuada de los contenidos científicos y tecnológicos, y de los procedimientos para generar una solución tecnológica (la propuesta por el material u otra diferente).				
f. Organizar una experiencia didáctica integrando conocimientos científicos y tecnológicos, metodología de resolución de problemas y procedimientos propios del trabajo tecnológico.				
g. Otras (que haya incorporado o hecho mejor con el recurso).				

² No aplicado: No lo hizo antes ni ahora con este recurso didáctico.

³ Incorporado: Integró la estrategia a sus clases a partir de la utilización del recurso didáctico propuesto.





3.2.2. Desarrollo de la estrategia didáctica	Mejor	Igual	No aplicado	Incorporado
Con respecto a su forma habitual de trabajo, usted logró:				
h. Encuadrar la tarea a partir de la formulación de uno (o varios) problemas.				
i. Explicitar consignas de trabajo que plantean una situación problemática.				
j. Organizar las actividades de aprendizaje atendiendo a las etapas propias de la resolución de problemas.				
k. Utilizar técnicas de trabajo grupal.				
l. Promover el trabajo colaborativo y cooperativo.				
m. Otras (que haya incorporado o hecho mejor con el recurso).				

3.2.3. Aspectos cognitivos (proceso de aprendizaje de sus alumnos)	Mejor	Igual	No aplicado	Incorporado
Con respecto a su forma habitual de trabajo, usted logró:				
n. Estimular a sus alumnos en la búsqueda de información e investigación en torno al problema eje del material.				
o. Promover la consulta a variadas fuentes de información.				
p. Rescatar, incorporar los aportes del grupo para identificar aspectos o variables críticas del problema.				
q. Evaluar los conflictos cognitivos propios del proceso de aprendizaje.				
r. Detectar, evaluar, la comprensión asociativa.				
s. Promover la reflexión sobre las actividades realizadas y las estrategias utilizadas en cada parte del proceso.				
t. Otras (que haya incorporado o hecho mejor con el recurso).				

4.1.2. ¿Realizó todo el proceso de construcción del recurso didáctico con sus alumnos? (Conteste este apartado en caso de que haya construido un equipo igual al propuesto. En caso contrario, pase al apartado 5 “Documentación”)

Sí	No
<input type="checkbox"/>	<input type="checkbox"/>



4.1.3. En caso de que su respuesta sea afirmativa, le pedimos que nos indique:

Sí	No
----	----

a. ¿Pudo seguir sin dificultades los procedimientos indicados en el “Manual de construcción”?	<input type="checkbox"/>	<input type="checkbox"/>
b. La secuencia indicada, ¿fue la adecuada para la construcción?	<input type="checkbox"/>	<input type="checkbox"/>
c. El grado de complejidad, ¿fue el apropiado para el nivel educativo a que se dirige el recurso?	<input type="checkbox"/>	<input type="checkbox"/>
d. Los contenidos científicos asociados, ¿son pertinentes para el desarrollo del recurso propuesto?	<input type="checkbox"/>	<input type="checkbox"/>
e. Los contenidos tecnológicos asociados, ¿son pertinentes para el desarrollo del recurso propuesto?	<input type="checkbox"/>	<input type="checkbox"/>
f. Con sus alumnos, ¿construyó el recurso didáctico siguiendo el proceso y la metodología de resolución de problemas?	<input type="checkbox"/>	<input type="checkbox"/>
g. ¿Siguió todos los procedimientos propuestos para la construcción pero incorporó sus propios contenidos científicos y tecnológicos?	<input type="checkbox"/>	<input type="checkbox"/>
h. Por el contrario, ¿hizo adaptaciones en los procedimientos de construcción pero mantuvo los mismos contenidos?	<input type="checkbox"/>	<input type="checkbox"/>
i. ¿Realizó la construcción siguiendo las actividades de aula propuestas en el material?	<input type="checkbox"/>	<input type="checkbox"/>
j. ¿Diseñó sus propias experiencias en función de su grupo de alumnos?	<input type="checkbox"/>	<input type="checkbox"/>

Sí	No
<input type="checkbox"/>	<input type="checkbox"/>

¿Completó todas las etapas del proceso de construcción propuesta?

En caso negativo, indíquenos a qué fase llegó:

- | | |
|---|--|
| a. <input type="checkbox"/> Planificación. | b. <input type="checkbox"/> Diseño en dos dimensiones. |
| c. <input type="checkbox"/> Construcción, armado. | d. <input type="checkbox"/> Ensayo y control. |
| e. <input type="checkbox"/> Superación de dificultades (evaluación del funcionamiento, siguiendo las indicaciones y la lista de control que brinda el material). | |
| f. <input type="checkbox"/> Construcción de otro equipo que se adapta más a sus necesidades curriculares (Si marcó esta alternativa, lo invitamos a responder, directamente, el apartado 4.1.5.). | |

4.1.4. Complete este ítem sólo si realizó el proceso de construcción del equipo siguiendo los procedimientos indicados en el Manual. Si no fue así, lo invitamos a responder el apartado 4.1.5.

Acerca de los materiales, herramientas e instrumentos:

	Sí	No
a. La especificación de los materiales para la construcción, ¿fue suficiente para conseguirlos?		
b. ¿Utilizó los mismos materiales (en calidad y tipificación) indicados en la documentación?		
c. ¿Reemplazó materiales, instrumentos, componentes, piezas, etc., sin alterar el resultado final previsto en el material?		
d. La especificación de las herramientas a utilizar, ¿le resultó adecuada?		
e. La cantidad de herramientas indicadas, ¿fue la necesaria?		
f. Los instrumentos, ¿estuvieron bien especificados?		
g. El tipo y cantidad de instrumentos, ¿fueron los adecuados para armar el recurso didáctico?		

4.1.5. En caso de que usted haya construido un recurso didáctico diferente al propuesto por el material de capacitación, le pedimos que nos indique si la razón fue:

<p>a. <input type="checkbox"/> El propuesto no se ajustaba a sus necesidades curriculares.</p>	<p>b. <input type="checkbox"/> No pudo conseguir los materiales o instrumentos indicados.</p>
<p>c. <input type="checkbox"/> No pudo interpretar el manual de construcción.</p>	<p>d. <input type="checkbox"/> Otra (Por favor, especifíquela).</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>



4.1.6. ¿Qué características específicas destacaría en este recurso didáctico diferente al propuesto por el material, que sus alumnos han construido. (Marque todas las opciones que considere necesarias):



a. Se ajusta mejor a los contenidos curriculares que necesita trabajar.

b. Es más económico.

c. Permite su reutilización (mediante el desarme y armado, en función de necesidades didácticas).

d. Es más adaptable (a diversos usos).

e. Otra (Por favor, especifique):

.....

.....

.....

.....

f. Descripción del recurso didáctico construido:

.....

.....

.....

.....

.....

g. Indique las principales diferencias con el equipo propuesto (estructurales, funcionales, didácticas):

.....

.....

.....

.....

.....



4.2.2. Ya sea que haya desarrollado el recurso didáctico con sus alumnos según las especificaciones del material, ya sea que haya construido otro diferente o que haya utilizado un equipo ya construido, en relación con las actividades que usted venía realizando, la utilización del recurso didáctico propuesto por el material le permitió (seleccione la opción que coincida con sus experiencias):

Con respecto a su forma habitual de trabajo, este recurso didáctico le permitió a usted, como docente:	Mejor	Igual	No aplicable ⁴	Otro ⁵
a. Integrar contenidos científicos y tecnológicos en la solución de situaciones problemáticas de carácter tecnológico.				
b. Diseñar situaciones de enseñanza y de aprendizaje centradas en la resolución de problemas tecnológicos.				
c. Planificar y promover en sus alumnos la organización del trabajo (planificación y secuenciación de tareas), según el proceso tecnológico.				
d. Favorecer la identificación de aspectos o variables críticas de una situación problemática.				
e. Organizar las actividades de manera que facilite la toma de decisiones por parte de los alumnos (determinación y selección de alternativas, opciones de diseño, materiales, etc.).				
f. Organizar la actividad de sus alumnos en función de soluciones diversas a los problemas planteados.				
g. Agregue otras que usted considere haber logrado de una mejor manera con este recurso didáctico				

⁴ NA: No aplicable; es una actividad que no realizó antes ni ahora.

⁵ Otro: Recuerde utilizar esta opción para indicar que agregará comentarios al final de este sector de la tabla.

Con respecto a su forma habitual de trabajo, este recurso le permitió a los alumnos (habilidades intelectuales):	Mejor	Igual	No aplicable	Otro
Capacidad de planificar				
h. Identificar variables o aspectos fundamentales de un problema tecnológico.				
i. Organizar su trabajo en etapas (identificar y seguir la secuencia de operaciones de un proceso).				
j. Ejecutar las actividades en los plazos o etapas previstas.				
k. Seleccionar materiales, herramientas y piezas, de acuerdo con las necesidades del diseño.				
l. Anticipar y resolver dificultades que podrían surgir en el proceso.				
m. Prever puntos críticos de todo el proceso.				
n. Agregue otras que considere que sus alumnos alcanzaron mejor con este recurso didáctico				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				





Capacidad para tomar decisiones	Mejor	Igual	No aplicable	Otro
o. Analizar alternativas en función de un problema.				
p. Seleccionar alternativas en función de las restricciones planteadas en el problema, o en el contexto de enseñanza y de aprendizaje.				
q. Adecuar la propuesta para la solución del problema planteado.				
r. Agregue otras que considere que sus alumnos alcanzaron mejor con este recurso didáctico				

Capacidad de aplicar y transferir	Mejor	Igual	No aplicable	Otro
s. Interrelacionar los datos, técnicas y procedimientos en el diseño de la solución.				
t. Utilizar técnicas de representación adecuadas al equipo que se construye o en el ya construido que se utiliza.				
u. Integrar los conocimientos científicos y tecnológicos en los momentos pertinentes para el diseño de la solución.				
v. Relacionar, ensamblar componentes en la secuencia adecuada.				
w. Utilizar de manera correcta la simbología y los lenguajes propios de la tecnología (representación gráfica, simbólica, etc.).				
x. Transferir conocimientos científicos y tecnológicos en otras actividades similares.				
y. Agregue otras que considere que sus alumnos alcanzaron mejor con este recurso didáctico				

Otro (Por favor, exprese aquí los comentarios que tenga, identificando el ítem con la letra que corresponda):

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



5. Documentación (Material teórico, manual de procedimientos y propuestas didácticas):



5.1. ¿Cómo calificaría los aportes del material recibido (encuadre y desarrollo teórico, y experiencias propuestas para el aula)?

	MV ⁶	V	PV
a. Por su potencialidad didáctica (sugerencias, propuestas de trabajo en el aula, papel motivador, etc.).			
b. Para sus necesidades curriculares (desarrollo de los contenidos y experiencias previstas en su planificación).			
c. Para organizar, planificar, concretar experiencias didácticas relacionadas con problemas de Educación Tecnológica.			
d. Para renovar, actualizar, ampliar (subraye el que se ajusta más a su experiencia) los contenidos que desarrolla en su área/ disciplina.			
e. Para trabajar conocimientos científicos y tecnológicos de manera asociada a un problema tecnológico.			
f. Para organizar experiencias de aprendizaje en torno a la utilización de recursos didácticos.			
g. Para utilizar un recurso didáctico en el marco de experiencias didácticas organizadas en función de la resolución de problemas.			
h. Para integrar mejor contenidos científicos y tecnológicos en la solución de problemas de carácter tecnológico.			
i. Para estimular la generación creativa de otros recursos didácticos.			

Otras (Especifíquelas, por favor)

.....

.....

.....

.....

.....

.....

.....

.....

⁶ Escala= MV: Muy valioso / V: Valioso / PV: Poco valioso

5.2. Manual de procedimientos para la construcción y el funcionamiento del recurso didáctico

En caso de que haya seguido los procedimientos contenidos en el Manual (ya sea para hacer un equipo igual o uno diferente al propuesto), le pedimos nos indique si:

	Sí	No	Otro
a. ¿Pudo seguir todos los procedimientos descritos, sin dificultad?			
b. ¿La secuencia descrita le resultó la adecuada?			
c. ¿La secuencia establecida le planteó alternativas según algún criterio (disponibilidad de los materiales, trabajo de contenidos específicos, etc.)?			
d. ¿La finalidad (para qué sirve) del equipo está indicada con claridad?			
e. ¿Se establecen cuáles son los contenidos (científicos o tecnológicos) que se asocian al equipo a construir?			
f. ¿Se determina la relación entre conocimientos implicados, procedimientos a seguir, materiales a utilizar y experiencias posibles de realizar?			
g. ¿Considera que la relación anterior es pertinente (es la que corresponde) para la construcción que se propone?			
h. ¿La descripción de los procedimientos le facilitaron la organización de las experiencias de trabajo con sus alumnos?			
i. ¿Pudo seguir las indicaciones para la puesta en funcionamiento?			
j. ¿Todas las indicaciones para el uso son claras?			

Por favor, fundamente sus respuestas negativas o agregue los comentarios que crea pertinentes (identifique el ítem a que se refiere):

.....

.....

.....

Otro (identifique con la letra que corresponda el ítem sobre el que hace observaciones)

.....

.....

.....



6.2. Técnicas (Por favor, complete tanto si construyó el equipo como si utilizó uno ya construido)

	Sí	No
a. Portabilidad. Puede ser utilizado en el taller, aula, laboratorio.		
b. Modularidad. Puede ser adaptado a diversos usos; para trabajar diversos contenidos curriculares o para realizar diferentes experiencias didácticas; para aprendizaje, demostraciones, análisis, etc.		
c. Reutilización. Posee partes, componentes, bloques o subsistemas que pueden ser desmontados para volver a su estado original, y usados en sí mismos o en forma independiente.		
d. Incrementabilidad. Puede complejizarse agregando piezas o completando el sistema para mejorar su funcionalidad, rendimiento, precisión o calidad.		
e. Aplicabilidad múltiple. Como sistema tecnológico, permite que usted seleccione las variables con las que desea trabajar (algunas de las que maneja el sistema, todas las previstas o agregar otras).		

Si su respuesta es negativa en alguna de ellas, indique por qué, identificando su comentario con la letra correspondiente:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



6.3. Didácticas (Por favor, complete tanto si construyó el equipo como si utilizó uno ya construido)



	Sí	No
a. Congruencia. Tiene relación con los testimonios de realidad incluidos en el módulo de capacitación.		
b. Pertinencia. Los componentes, bloques funcionales y sistemas son adecuados para el trabajo con los contenidos curriculares de la educación técnico-profesional.		
c. Integración. Posibilita el tratamiento asociado de los conocimientos científicos y tecnológicos propuestos en el material.		
d. Escalabilidad. Es posible utilizarlo con proyectos o problemas con diferentes niveles de complejidad.		
e. Complejidad creciente. Las soluciones alcanzadas para una parte del problema, sirven de base para las siguientes o permite que, agregando componentes, sea utilizado como solución a problemas más complejos.		
f. Adaptabilidad. Permite su adaptación a soluciones diversas en torno a las problemáticas planteadas.		



Si su respuesta es negativa en alguna de ellas, indique por qué, identificándola con la letra correspondiente:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

7. Otras características del material teórico:

¿Cómo calificaría el diseño del módulo escrito (desarrollo de contenidos científicos y tecnológicos, y propuestas de experiencias didácticas)?

	MB ⁷	B	R	M
a. Formato gráfico del material (distribución del contenido, márgenes, distribución de texto e imágenes, inserción de gráficos, diseño gráfico global, etc.).				
b. Lenguaje utilizado (claridad, adecuación al destinatario).				
c. Organización (secuencia entre cada parte).				
d. Adecuación al destinatario (evidencia que se toma en cuenta que es un material para ser trabajado en un ámbito escolar).				
e. Pertinencia de los conocimientos científicos con las problemáticas planteadas.				
f. Pertinencia de los conocimientos tecnológicos con las problemáticas planteadas.				
g. Vinculación (pertinencia) del recurso didáctico que propone con las situaciones didácticas planteadas.				
h. Congruencia (vinculación) de los contenidos propuestos con el recurso didáctico.				
i. Aporte metodológico para enriquecer sus estrategias didácticas.				
j. Aporte teórico (en general) para su trabajo docente.				
k. Valor motivador para el trabajo con sus alumnos.				
l. Valor orientador para generar sus propios recursos didácticos.				
m. Concepción innovadora para el trabajo didáctico en la educación técnico-profesional.				

Si marcó la opción “Malo”, le pedimos que nos explique por qué:

.....

.....

.....

⁷ Escala= MB: Muy bueno / B: Bueno / R: Regular / M: Malo



8. Propuestas o nuevas ideas:

Tanto para los autores de este material, como para el CeNET como institución responsable de su elaboración y distribución, una de las finalidades más importantes es suscitar en los educadores nuevas ideas, aplicaciones o propuestas creativas a partir de la lectura o el trabajo con el módulo.

En función de ello, le solicitamos que nos indique:

Si a partir del módulo (contenido teórico y recurso didáctico) usted, en su calidad de (marque todas las opciones que correspondan):

a. <input type="checkbox"/> docente a cargo de un grupo de alumnos	b. <input type="checkbox"/> directivo
c. <input type="checkbox"/> responsable de la asignatura:	d. <input type="checkbox"/> lector del material
e. <input type="checkbox"/> otro (especifique):	

ha generado nuevas ideas o propuestas:

Respecto de los contenidos (independientemente del recurso didáctico):

		Sí	No
a. Organización de su asignatura.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
b. Contenidos científicos y tecnológicos (formas de asociarlos, ampliarlos, desarrollarlos, etc.)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
c. Planificación de las experiencias didácticas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
d. Trabajo con resolución de problemas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

En relación con el recurso didáctico. Le pedimos que nos relate (libremente) las nuevas ideas o propuestas que el trabajo con este material le ha suscitado:



A series of horizontal dotted lines providing space for the respondent to write their answers.

Títulos en preparación de la serie “**Recursos didácticos**”.

- Agenda electrónica para personas con disminución visual
- Arquitectura bioclimática
- Auto solar
- Banco de trabajo
- Generador eólico
- Manipulador neumático
- Máquina de vapor
- Matriceria. Moldes y modelos
- Planta de tratamiento de aguas residuales
- Simuladores interconectables basados en lógica digital
- Sismógrafo
- Sistemas SCADA para el control de procesos industriales
- Tren de aterrizaje