



# *UNIVERSIDAD AUTÓNOMA DE ZACATECAS*

*“Francisco García Salinas”*

*ÁREA DE INGENIERÍAS Y TECNOLÓGICAS*

*UNIDAD ACADÉMICA DE INGENIERÍA I*

**PROGRAMA DE INGENIERÍA EN MANUFACTURA**

## Lógica de programación

---

*Unidad Didáctica*

|               |                              |
|---------------|------------------------------|
| SEMESTRE      | SEGUNDO                      |
| HORAS TOTALES | 80 horas                     |
| CRÉDITOS      | 5                            |
| ANTECEDENTE   |                              |
| CONSECUENTE   | PRINCIPIOS DE<br>ELECTRÓNICA |

|                         |                              |
|-------------------------|------------------------------|
| EJE FORMATIVO           | CIENCIAS DE LA<br>INGENIERÍA |
| ACADEMIA                |                              |
| FECHA DE<br>ELABORACIÓN | ENERO 2017                   |
| PRÓXIMA REVISIÓN        | JUNIO 2022                   |

Programa de la asignatura LÓGICA DE PROGRAMACIÓN con las competencias genéricas y disciplinares para el aprendizaje.

---

### COMPETENCIAS GENÉRICAS

**Desarrollar soluciones de automatización de procesos productivos y servicios mediante la incorporación sinérgica de elementos mecánicos, eléctricos, electrónicos, control y sistemas robóticos para mejorar la productividad y calidad del proceso y producto.**

---

---

---

---

---

### COMPETENCIAS DISCIPLINARES DEL EJE FORMATIVO DE CIENCIAS BÁSICAS

### UNIDAD DE APRENDIZAJE

**Elaborar, analizar y simular los circuitos de control utilizados para la automatización.**

| I | II | III | IV | V | VI |
|---|----|-----|----|---|----|
| X | X  |     |    |   |    |

**Elaborar entornos visuales utilizados en los sistemas de automatización.**

X

**Elaborar entornos visuales de diseño en la programación que controlan los sistemas automatizados.**

X

**Elaborar y desarrollar las diferentes etapas de la programación en el lenguaje ensamblador**

X

---

---

| <b>Unidad de Aprendizaje</b> | <b>Nombre de la Unidad de Aprendizaje</b>    | <b>Tiempo Requerido</b> |
|------------------------------|--|-------------------------|
| <b>I</b>                     | <b>Algoritmos y diagramas de flujo</b>       |                         |
|                              | 1. Introducción a leguajes de programación   | 5                       |
|                              | 2. Algoritmos y diseño de diagramas de flujo | 5                       |
|                              | 3. Pseudocódigo                              | 5                       |

### **Desempeño de estudiante al concluir la unidad**

Identificar conceptos de procesos de programación, las fases de la compilación de programas. Distinguir los niveles de programación y sus aplicaciones. Definir conceptos, describir la estructura de los diagramas de flujo. Relacionar el pseudocódigo con lenguajes de programación.

### **Objetivos de aprendizaje**

Desarrollar algoritmos para la solución de problemas de ingeniería

### **Competencias a desarrollar**

Seleccionar lenguajes de programación, desarrollar diagramas de flujo y pseudocódigos

### **Actividad de enseñanza**

Mapa Conceptuales  
Aprendizaje basado en problemas  
Discusión en grupo

### **Actividad de aprendizaje**

Integra un portafolio de evidencias que incluya:

- Mapa conceptual sobre lenguajes de programación, diagramas de flujo, pseudocódigo.
- Cuadro sinóptico de la clasificación de los lenguajes de programación y sus características.
- Diagrama de flujo del proceso de compilación de un

### **Instrumento de evaluación**

Portafolio de evidencias  
Rubrica

programa  
 d) Ejercicios de solución de problemas de ingeniería que incluyan:  
 - Resultados del análisis del problema  
 - Diagrama de flujo  
 - Pseudocódigo  
 - Compilación

## **Rol del docente**

## **Material didáctico**

Pizarrón, Equipo de cómputo, Software Especializado, TIC´s, Cañón.

## **Material bibliográfico de consulta**

1. Como programar en C++. Harvey M. Deitel, Paul J. Deitel. Ed. Pearson Education, México. 2008.
2. Fundamentos de Programación, Algoritmos, estructura de datos y objetos. Luis Joyanes Aguilar. Ed. McGraw Hill, España. 2008.
3. Diseño de algoritmos y su codificación en lenguaje C. María Adriana Corona. Ed. McGraw Hill, México. 2011.
4. C Programming for Arduino. Julien Bayle. Ed. Pack Publishing, Reino Unido. 2013.

| <b>Unidad de Aprendizaje</b> | <b>Nombre de la Unidad de Aprendizaje Programación estructurada</b> | <b>Tiempo Requerido</b> |
|------------------------------|---|-------------------------|
| <b>II</b>                    | 1. Introducción al lenguaje estructurado.                           | 5                       |
|                              | 2. Entrada y salida de datos.                                       | 5                       |
|                              | 3. Instrucciones condicionales.                                     | 10                      |

|                             |    |
|-----------------------------|----|
| 4. Estructuras repetitivas. | 10 |
| 5. Funciones.               | 10 |
| 6. Estructura de datos.     | 5  |

### **Desempeño de estudiante al concluir la unidad**

Identificar la estructura, tipos y operadores de programas en lenguaje estructurado. Describir las características de las entradas y salidas, los tipos de instrucciones y las estructuras de datos. Explicar el funcionamiento y sintaxis de las entradas y salidas, de las instrucciones condicionales y de switcheo y del uso de estructuras de datos.

#### **Objetivos de aprendizaje**

Desarrollar programas en lenguaje estructurado para la solución de problemas en la ingeniería.

#### **Competencias a desarrollar**

Desarrollar programas que involucren instrucciones de entrada y salida, instrucciones condicionales, instrucciones repetitivas, funciones y estructuras de datos

#### **Actividad de enseñanza**

Prácticas en laboratorio  
Aprendizaje Basado en Problemas  
Aprendizaje auxiliado por tecnologías de la información

#### **Actividad de aprendizaje**

Integra un portafolio de evidencias con prácticas de solución de problemas con estructuras de control, que contenga:

- Resultados del análisis del problema
- Diagrama de flujo con:
  - Instrucciones condicionales
  - Estructuras repetitivas
  - Funciones
  - Estructuras de datos
- Pseudocódigo

#### **Instrumento de evaluación**

Ejercicios prácticos  
Rubrica

- Código de programación
- Compilación

## Rol del docente

## Material didáctico

Pizarrón, Equipo de cómputo, Software Especializado, TIC's, Cañón.

## Material bibliográfico de consulta

1. Como programar en C++. Harvey M. Deitel, Paul J. Deitel. Ed. Pearson Education, México. 2008.
2. Fundamentos de Programación, Algoritmos, estructura de datos y objetos. Luis Joyanes Aguilar. Ed. McGraw Hill, España. 2008.
3. Diseño de algoritmos y su codificación en lenguaje C. María Adriana Corona. Ed. McGraw Hill, México. 2011.
4. C Programming for Arduino. Julien Bayle. Ed. Pack Publishing, Reino Unido. 2013.

| Unidad de Aprendizaje | Nombre de la Unidad de Aprendizaje<br>Programación de tarjetas de prototipado rápido | Tiempo Requerido |
|-----------------------|--|------------------|
| III                   | 1. Periféricos de las tarjetas.  | 10               |
|                       | 2. Entrada y salida de señales digitales.  | 10               |
|                       | 3. Entrada y salida de señales análogas.   | 5                |

## Desempeño de estudiante al concluir la unidad

Identificar los periféricos de las tarjetas de prototipado rápido, Describir el funcionamiento y sintaxis de instrucciones de entrada y salida en señales digitales y análogas.

### Objetivos de aprendizaje

Programar aplicaciones en tarjetas de prototipado rápido para implementar sistemas de control de procesos industriales.

### Competencias a desarrollar

Implementar programas de control de señales de entrada y salida digitales y análogas.

**Actividad de enseñanza**

Prácticas en laboratorio  
Aprendizaje Basado en Problemas  
Aprendizaje auxiliado por tecnologías de la información

**Actividad de aprendizaje**

Integra un portafolio de evidencias de prácticas con tarjetas de prototipado rápido que incluya:

- Resultados del análisis del problema
- Diagrama de flujo del proceso de sintaxis de instrucciones
- Pseudocódigo
- Código de programación
- Reporte de la implementación y compilación del programa

**Instrumento de evaluación**

Portafolio de evidencias  
Rubrica

**Rol del docente****Material didáctico**

Pizarrón, Equipo de cómputo, Software Especializado, TIC´s, Cañón.

**Material bibliográfico de consulta**

1. Como programar en C++. Harvey M. Deitel, Paul J. Deitel. Ed. Pearson Education, México. 2008.
2. Fundamentos de Programación, Algoritmos, estructura de datos y objetos. Luis Joyanes Aguilar. Ed. McGraw Hill, España. 2008.
3. Diseño de algoritmos y su codificación en lenguaje C. María Adriana Corona. Ed. McGraw Hill, México. 2011.
4. C Programming for Arduino. Julien Bayle. Ed. Pack Publishing, Reino Unido. 2013.



---

**Políticas del curso**

**Perfil docente**

**Calificación ordinaria**

**De las asistencias**

---

---

**En la realización de esta Unidad Didáctica participaron**

---

**Elaborada por** Luis Ernesto Mendoza Navarro

---

**En la revisión de este programa participaron**

---

**Coordinador de la Academia de:**

---