

# UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

COORDINACIÓN GENERAL DE FORMACIÓN BÁSICA  
COORDINACIÓN GENERAL DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA  
PROGRAMA DE UNIDAD DE APRENDIZAJE

## I. DATOS DE IDENTIFICACIÓN

- 1. Unidad Académica:** Facultad de Ingeniería, Mexicali; Facultad de Ingeniería, Arquitectura y Diseño, Ensenada y Facultad de Ciencias Químicas e Ingeniería, Tijuana.
- 2. Programa Educativo:** Ingeniero en Computación
- 3. Plan de Estudios:** 2020-1
- 4. Nombre de la Unidad de Aprendizaje:** Programación Estructurada
- 5. Clave:** 36276
- 6. HC:** 00 **HL:** 02 **HT:** 02 **HPC:** 00 **HCL:** 00 **HE:** 00 **CR:** 04
- 7. Etapa de Formación a la que Pertenece:** Básica
- 8. Carácter de la Unidad de Aprendizaje:** Obligatoria
- 9. Requisitos para Cursar la Unidad de Aprendizaje:** Ninguno



### Equipo de diseño de PUA

Luis Guillermo Martínez Méndez  
Carelia Guadalupe Gaxiola Pacheco  
Luz Evelia López Chico  
Pablo Martín Navarro Álvarez

**Fecha:** 17 de octubre de 2019

### Vo.Bo. de Subdirectores de Unidades Académicas

Alejandro Mungaray Moctezuma  
Humberto Cervantes De Ávila  
Rocío Alejandra Chávez Santoscoy

## **II. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE**

Esta unidad de aprendizaje requiere que el alumno haya cursado Metodología de la Programación, por otra parte, los conceptos adquiridos en esta unidad de aprendizaje son necesarios para otras como Programación Orientada a Objetos y Algoritmos y Estructuras de Datos, es recomendable que el alumno tenga bases sobre un lenguaje de programación estructurado de alto nivel. Es de carácter obligatorio, pertenece a la etapa básica, y corresponde al área de conocimiento de Ciencias de la Ingeniería, contribuye a la formación del estudiante permitiéndole adquirir destreza en el manejo del paradigma de la programación estructurada a fin elaborar programas eficientes que resuelvan problemas específicos de diversas áreas.

## **III. COMPETENCIA DE LA UNIDAD DE APRENDIZAJE**

Diseñar programas correctamente estructurados, aplicando los principios de abstracción procedimental del diseño descendente, para construir programas eficaces, eficientes y fáciles de actualizar, con una actitud analítica y creativa.

## **IV. EVIDENCIA(S) DE DESEMPEÑO**

Elabora prototipo funcional de un sistema donde se aplique el diseño modular.

## V. DESARROLLO DE CONTENIDO

1. Tipos de datos
2. Expresiones aritméticas, lógicas y relacionales
3. Estructuras de control de selección
4. Estructuras de control de iteración
5. Abstracción procedimental
  - 5.1 Diseño descendente, tareas y subtareas.
  - 5.2 Módulos
  - 5.3 Parámetros y argumentos
  - 5.4 Resultados (valores de retorno)
6. Recursión
7. Programas multiarchivo y compilación separada
8. Biblioteca de funciones
9. Archivos

## VI. ESTRUCTURA DE LAS PRÁCTICAS DE TALLER

No. de Práctica	Competencia	Descripción	Material de Apoyo	Duración
1	Utilizar los tipos de datos adecuados, mediante el análisis del dominio de las variables involucradas en la solución de problemas, para optimizar el uso de recursos de memoria, con actitud analítica y de síntesis.	<p>El docente: Explica los tipos de datos básicos y estructurados, así como el proceso de declararlos en un lenguaje de programación estructurado de alto nivel y almacenarlos en constantes y variables.</p> <p>El alumno: Determinará el tipo de dato adecuado para representar el dominio de variables de entrada y salida presentes en los problemas propuestos. Entregará el reporte con los tipos de datos propuestos y una justificación de su decisión.</p>	Material audiovisual sobre el tema a exponer, videoprojector, bibliografía de apoyo, computadora, editor de texto, lápiz y papel.	2 horas
2	Desarrollar expresiones aritméticas, lógicas y relacionales, aplicando la jerarquía de operadores de un lenguaje de programación estructurado de alto nivel, para una toma de decisiones que favorezca la resolución de problemas mediante el uso de computadora, de forma ordenada y propositiva.	<p>El docente: Expone el tema de operadores y cómo se representan en algoritmos, diagramas de flujo, pseudocódigo y como código en un lenguaje de programación estructurado de alto nivel. Posteriormente desarrolla expresiones aritméticas, lógicas y relacionales, a partir de planteamientos propuestos como ejemplo.</p> <p>El alumno: Construirá las expresiones aritméticas, lógicas y relacionales a partir de planteamientos propuestos por el docente.</p>	Material audiovisual sobre el tema a exponer, videoprojector, bibliografía de apoyo, computadora, editor de texto, lápiz y papel.	2 horas

		<p>Evaluará las expresiones dando valores a las variables. Entregará un reporte con las expresiones construidas, incluyendo el resultado de las evaluaciones de las mismas.</p>		
3	<p>Diseñar programas de cómputo, aplicando las estructuras de control de selección, para proporcionar soluciones óptimas a problemas del área de ingeniería, de manera innovadora y ordenada.</p>	<p>El docente: Expone el tema de estructuras de control de selección y cómo se representan en algoritmos, diagramas de flujo, pseudocódigo y como código en un lenguaje de programación estructurado de alto nivel; posteriormente muestra ejemplos de problemas dónde para su solución es necesario la aplicación de estructuras de control de selección para la toma de decisiones. El alumno: Resuelve problemas propuestos por el docente dónde se utilizan las estructuras de control de selección en su solución. Entregará un reporte con el pseudocódigo de las soluciones a los problemas planteados.</p>	<p>Material audiovisual sobre el tema a exponer, videoprojector, bibliografía de apoyo, computadora, editor de texto, lápiz y papel.</p>	2 horas
4	<p>Diseñar programas de cómputo, aplicando las estructuras de control de iteración, para proporcionar soluciones óptimas a problemas del área de ingeniería, de manera innovadora y ordenada.</p>	<p>El docente: Expone el tema de estructuras de control de iteración y cómo se representan en algoritmos, diagramas de flujo, pseudocódigo y como código en un lenguaje de programación estructurado de alto nivel; posteriormente muestra ejemplos de problemas dónde para su solución es necesario la aplicación de estructuras de</p>	<p>Material audiovisual sobre el tema a exponer, videoprojector, bibliografía de apoyo, computadora, editor de texto, lápiz y papel.</p>	2 horas

		<p>control de selección para la toma de decisiones.</p> <p>El alumno: Resuelve problemas propuestos por el docente dónde se utilizan las estructuras de control de selección en su solución. Entregará un reporte con el pseudocódigo de las soluciones a los problemas planteados.</p>		
5	<p>Aplicar la abstracción procedimental, al analizar las generalidades en las estrategias de solución de problemas complejos, para simplificar el proceso de resolución, con actitud creativa y organizada.</p>	<p>El docente: Expone los conceptos de diseño descendente, tareas y subtareas; mostrando ejemplos generales para demostrar como la abstracción procedimental simplifica la resolución de problemas complejos.</p> <p>El alumno: Resolverá problemas propuestos por el docente. Entregará el reporte con los algoritmos de soluciones a los problemas propuestos.</p>	<p>Material audiovisual sobre el tema a exponer, videoprojector, bibliografía de apoyo, computadora, editor de texto, lápiz y papel.</p>	2 horas
6		<p>El docente: Expone el concepto de módulo y cómo se representan en algoritmos, diagramas de flujo, pseudocódigo y código en un lenguaje de programación estructurado de alto nivel; mostrando ejemplos generales donde se aplique la abstracción procedimental en soluciones de problemas complejos mediante el uso de procedimientos.</p> <p>El alumno: Resolverá problemas propuestos</p>	<p>Material audiovisual sobre el tema a exponer, videoprojector, bibliografía de apoyo, computadora, editor de texto, lápiz y papel.</p>	2 horas

		<p>por el docente. Entregará el reporte con el pseudocódigo de soluciones a los problemas propuestos</p>		
7		<p>El docente: Expone el uso de los parámetros y argumentos en los módulos para representar la abstracción de valores y cómo se representan en algoritmos, diagramas de flujo, pseudocódigo y código en C; mostrando ejemplos generales donde se aplique la abstracción procedimental en soluciones de problemas complejos mediante el uso de procedimientos con paso de parámetros. El alumno: Resolverá problemas propuestos por el docente. Entregará el reporte con el pseudocódigo de soluciones a los problemas propuestos</p>	<p>Material audiovisual sobre el tema a exponer, videoprojector, bibliografía de apoyo, computadora, editor de texto, lápiz y papel.</p>	4 horas
8		<p>El docente: Expone el uso y utilidad de los resultados (valores de retorno) de los módulos y cómo se representan en algoritmos, diagramas de flujo, pseudocódigo y como código en un lenguaje de programación estructurado de alto nivel; mostrando ejemplos generales donde se aplique la abstracción procedimental en soluciones de problemas complejos mediante el uso de procedimientos con paso de parámetros y los cuales regresen</p>	<p>Material audiovisual sobre el tema a exponer, videoprojector, bibliografía de apoyo, computadora, editor de texto, lápiz y papel.</p>	4 horas

		<p>un valor como resultado de su ejecución.</p> <p>El alumno:          Resolverá problemas propuestos por el docente.          Entregará el reporte con el pseudocódigo de soluciones a los problemas propuestos</p>		
9	<p>Utilizar la recursión, aplicando la abstracción procedimental en la estrategia de solución de problemas complejos, para ofrecer una perspectiva basada en sus propias definiciones al plantear una solución, con actitud analítica y organizada.</p>	<p>El docente:          Expone el tema de recursión en la invocación de módulos y cómo se representan en algoritmos, diagramas de flujo, pseudocódigo y como código en un lenguaje de programación estructurado de alto nivel; explicando problemáticas clásicas donde se aplica la recursión en su solución.</p> <p>El alumno:          Resolverá problemas propuestos por el docente aplicando la recursión.          Entregará el reporte con el pseudocódigo de soluciones a los problemas propuestos, aplicando la recursión, dónde se especifique el estado basé y cómo se resolvería el mismo problema de manera iterativa.</p>	<p>Material audiovisual sobre el tema a exponer, videoprojector, bibliografía de apoyo, computadora, editor de texto, lápiz y papel.</p>	4 horas
10	<p>Construir programas multiarchivo, aplicando la abstracción procedimental, al analizar las generalidades en las estrategias de solución de problemas complejos, para favorecer el reúso de software en la solución de programas, con actitud creativa y organizada.</p>	<p>El docente:          Expone el tema de programas multiarchivo y compilación separada, sus ventajas y su utilidad, así mismo cómo se representan en algoritmos, diagramas de flujo, pseudocódigo y como código en un lenguaje de programación estructurado de</p>	<p>Material audiovisual sobre el tema a exponer, videoprojector, bibliografía de apoyo, computadora, editor de texto, lápiz y papel.</p>	2 horas

		<p>alto nivel.</p> <p>El alumno: Determinará como se puede dividir la solución de problemas propuestos por el docente en diferentes archivos. Entregará el reporte con el pseudocódigo de soluciones a los problemas propuestos.</p>		
11	<p>Construir bibliotecas de funciones, aplicando la abstracción procedimental, al analizar las generalidades en las estrategias de solución de problemas complejos, para proporcionar servicios a programas independientes, con actitud creativa y organizada.</p>	<p>El docente: Expone el tema de bibliotecas de funciones, sus ventajas y su utilidad, así mismo cómo se representan en algoritmos, diagramas de flujo, pseudocódigo y como código en un lenguaje de programación estructurado de alto nivel. El alumno: Diseñará la estructura de una biblioteca de funciones, siguiendo las instrucciones dadas por el docente. Entregará el reporte con el pseudocódigo de la biblioteca de funciones.</p>	<p>Material audiovisual sobre el tema a exponer, videoprojector, bibliografía de apoyo, computadora, editor de texto, lápiz y papel.</p>	2 horas
12	<p>Diseñar archivos binarios, mediante el manejo de memoria secundaria, para la solución de problemas que requieran el almacenamiento de información de forma permanente, con disciplina y actitud propositiva.</p>	<p>El docente: Expone el tema de archivos binarios, así mismo cómo se representan en algoritmos, diagramas de flujo, pseudocódigo y como código en un lenguaje de programación estructurado de alto nivel. Proporciona ejemplos de su aplicación. El alumno: Diseñará la estructura de archivos que permitan la</p>	<p>Material audiovisual sobre el tema a exponer, videoprojector, bibliografía de apoyo, computadora, editor de texto, lápiz y papel.</p>	4 horas

		organización y almacenamiento de la información requerida para la solución de los problemas planteados por el docente. Entregará el reporte con el diseño de la estructura de los archivos.		
--	--	--	--	--

## VI. ESTRUCTURA DE LAS PRÁCTICAS DE LABORATORIO

No. de Práctica	Competencia	Descripción	Material de Apoyo	Duración
1	Identificar la estructura de un programa de cómputo en lenguaje C, mediante la declaración de variables del tipo adecuado, el uso de expresiones aritméticas y la utilización de funciones de entrada y salida, para familiarizarse con el entorno de trabajo, de forma ordenada y propositiva.	El alumno desarrollará programas que resuelvan problemas donde para su solución aplique expresiones aritméticas y tipos de datos primarios; así como el uso de funciones de entrada y salida. El alumno individualmente entregará el código de los programas realizados.	Videoprojector, pizarrón, material audiovisual, bibliografía de apoyo, computadora, editor de texto compilador de ANSI C, unidad de almacenamiento.	2 horas
2		El alumno desarrollará programas que resuelvan problemas donde para su solución aplique expresiones aritméticas y tipos de datos estructurados; así como el uso de funciones de entrada y salida. El alumno individualmente entregará el código de los programas realizados.	Videoprojector, pizarrón, material audiovisual, bibliografía de apoyo, computadora, editor de texto compilador de ANSI C, unidad de almacenamiento.	2 horas
3	Elaborar programas de cómputo, aplicando las estructuras de control de selección, para proporcionar soluciones óptimas a problemas del área de ingeniería, de manera innovadora y ordenada.	El alumno desarrollará programas que resuelvan problemas donde para su solución sea necesaria la aplicación de expresiones condicionales para la toma de decisiones. El alumno individualmente entregará el código de los programas realizados.	Videoprojector, pizarrón, material audiovisual, bibliografía de apoyo, computadora, editor de texto compilador de ANSI C, unidad de almacenamiento.	2 horas
4	Elaborar programas de cómputo, aplicando las estructuras de control de iteración, para proporcionar soluciones óptimas a problemas del área de ingeniería, de manera innovadora y ordenada.	El alumno desarrollará programas que resuelvan problemas donde para su solución sea necesaria la aplicación de estructuras de control de iteración. El alumno individualmente entregará	Videoprojector, pizarrón, material audiovisual, bibliografía de apoyo, computadora, editor de texto compilador de ANSI C, unidad de almacenamiento.	2 horas

		el código de los programas realizados.		
5	Diseñar y construir funciones, para utilizar las ventajas de la programación modular en la solución de problemas de procesamiento de información, con actitud propositiva y organizada.	El alumno desarrollará programas que resuelvan problemas donde para simplificar la solución haga uso de funciones que hagan uso de parámetros. El alumno individualmente entregará el código de los programas realizados.	Videoprojector, pizarrón, material audiovisual, bibliografía de apoyo, computadora, editor de texto compilador de ANSI C, unidad de almacenamiento.	4 horas
6	Diseñar y construir funciones las cuáles regresen un valor como resultado de su ejecución, para utilizar las ventajas de la programación modular en la solución de problemas de procesamiento de información, con actitud propositiva y organizada.	El alumno desarrollará programas que resuelvan problemas donde para simplificar la solución haga uso de funciones las cuáles regresen un valor como resultado de su ejecución. El alumno individualmente entregará el código de los programas realizados.	Videoprojector, pizarrón, material audiovisual, bibliografía de apoyo, computadora, editor de texto compilador de ANSI C, unidad de almacenamiento.	6 horas
7	Diseñar y construir funciones recursivas, para utilizar las ventajas de la programación modular en la solución de problemas de procesamiento de información, con actitud propositiva y organizada.	El alumno desarrollará programas que resuelvan problemas donde para simplificar la solución aplique la recursividad de funciones. El alumno individualmente entregará el código de los programas realizados.	Videoprojector, pizarrón, material audiovisual, bibliografía de apoyo, computadora, editor de texto compilador de ANSI C, unidad de almacenamiento.	4 horas
9	Desarrollar aplicaciones de software, utilizando funciones localizadas en diferentes archivos, para permitir el reuso de software en la solución de problemas, de forma analítica, propositiva y organizada.	El alumno desarrollará programas que resuelvan problemas donde para su solución sea necesario el uso de funciones localizadas en diferentes archivos. El alumno individualmente entregará el código de los programas realizados.	Videoprojector, pizarrón, material audiovisual, bibliografía de apoyo, computadora, editor de texto compilador de ANSI C, unidad de almacenamiento..	4 horas
10	Desarrollar bibliotecas de funciones, mediante la definición de archivos de cabecera, para organizar	El alumno desarrollará bibliotecas de funciones y las utilizará en programas que resuelvan problemas	Videoprojector, pizarrón, material audiovisual, bibliografía de apoyo,	2 horas

	funciones de acuerdo a su propósito, de forma analítica, propositiva y organizada.	donde para su solución sea necesario el uso de funciones localizadas en esas bibliotecas de funciones. El alumno individualmente entregará el código de los programas realizados.	computadora, editor de texto compilador de ANSI C, unidad de almacenamiento.	
12	Desarrollar programas de cómputo, utilizando archivos que permitan el manejo de memoria secundaria, para la solución de problemas, con disciplina y actitud propositiva.	El alumno desarrollará programas que resuelvan problemas donde para su solución sea necesario el uso de memoria secundaria. El alumno individualmente entregará el código de los programas realizados.	Videoprojector, pizarrón, material audiovisual, bibliografía de apoyo, computadora, editor de texto compilador de ANSI C, unidad de almacenamiento.	4 horas

## VII. MÉTODO DE TRABAJO

**Encuadre:** El primer día de clase el docente debe establecer la forma de trabajo, criterios de evaluación, calidad de los trabajos académicos, derechos y obligaciones docente-alumno.

### **Estrategia de enseñanza (docente)**

El docente expone los temas apoyado en material audiovisual, resuelve problemas de ejemplo y apoya a los alumnos en la solución de los problemas propuestos, plantear problemas para ser resueltos en taller, problemas prácticos para ser resueltos en laboratorio.

### **Estrategia de aprendizaje (alumno)**

El estudiante realiza una lectura previa a los temas que se verán en taller, resuelve los problemas propuestos por el docente, reflexiona sobre las soluciones propuestas, realiza las prácticas de laboratorio.

## VIII. CRITERIOS DE EVALUACIÓN

La evaluación será llevada a cabo de forma permanente durante el desarrollo de la unidad de aprendizaje de la siguiente manera:

### **Criterios de acreditación**

- Para tener derecho a examen ordinario y extraordinario, el estudiante debe cumplir con los porcentajes de asistencia que establece el Estatuto Escolar vigente.
- Calificación en escala del 0 al 100, con un mínimo aprobatorio de 60.

### **Criterios de evaluación**

- Actividades de Taller.....40%
  - Prácticas de Laboratorio.....40%
  - Evidencia de desempeño.....20%
- (Prototipo funcional de un sistema donde se aplique diseño modular)
- Total... 100%**

## IX. REFERENCIAS

Básicas	Complementarias
<p>Corona, M.A. y Ancona, M.A. (2011). <i>Diseño de algoritmos y su codificación en lenguaje C</i>. España: McGraw-Hill. ISBN: 9786071505712 [clásica]</p> <p>Deitel, P., y Deitel, H. (2016). <i>How to Program C++</i> (10ª ed.). Estados Unidos: Pearson Educación.</p> <p>Sznajdleder, P. A. (2017). <i>Programación estructurada a fondo: implementación de algoritmos en C</i>. Buenos Aires, Argentina: Alfaomega</p>	<p>Acera, M.A. (2015). <i>Manual Imprescindible C/C++: curso de programación 2015</i>. Anaya Multimedia 2015</p> <p>Gaddis, T. (2018). <i>Starting Out with C++ from Control Structures through Objects</i>. (8ª ed.). Estados Unidos: Pearson.</p> <p>Joyanes, L. y Zahonero, I. (2001). <i>Programación en C. Metodología, estructura de datos y objetos</i>. España: McGraw-Hill. ISBN: 8448130138 [clásica]</p> <p>Joyanes, L. y Zahonero, I. (2014). <i>Programación en C, C++, Java y UML</i> (2ª ed.). Estados Unidos: McGraw-Hill. [clásica]</p>

## X. PERFIL DEL DOCENTE

El docente que imparta la asignatura deberá contar con una formación en ciencias de la computación o área afín, preferentemente con Maestría o Doctorado en área afín. Es deseable contar con experiencia docente mínima de dos años en la enseñanza de programación estructurada en nivel superior, así como haber aprobado cursos de formación docente.