

# 1. Programación y bases de datos

## INTRODUCCIÓN

En este módulo de 228 horas pedagógicas, se promueve el pensamiento algorítmico para que los y las estudiantes desarrollen la estructura lógica que les permita lograr los objetivos propuestos para la especialidad. Para ello, se sugiere establecer relaciones constantes con el escenario del sector productivo, ya que los elementos de la programación se usan como base para generar soluciones de *software* que son aplicados en contextos laborales. El objetivo principal de este módulo es que representen situaciones y problemas mediante la programación de algoritmos, y reconozcan patrones lógicos secuenciales. Los modelos a implementar, en este contexto, son NetBeans y MySQL.

Este módulo propone como metodología de enseñanza la resolución de casos, que potencia la capacidad para tomar decisiones frente a problemas asociados a la disciplina y el contexto laboral. En las actividades de aprendizaje sugeridas, se proponen casos para trabajar semana a semana.

Se sugiere que las actividades incorporen metodologías activas, centradas en los y las estudiantes y en el desarrollo de los Objetivos de Aprendizaje Genéricos, como el trabajo colaborativo y el cumplimiento de estándares y de los protocolos asociados.

## APRENDIZAJES ESPERADOS Y CRITERIOS DE EVALUACIÓN

MÓDULO 1 · PROGRAMACIÓN Y BASES DE DATOS		228 HORAS	TERCERO MEDIO
OBJETIVOS DE APRENDIZAJE DE LA ESPECIALIDAD			
<b>OA 1</b>			
Desarrollar programas y rutinas de baja y mediana complejidad que involucren estructuras y bases de datos de acuerdo a los requerimientos de la empresa.			
APRENDIZAJES ESPERADOS	CRITERIOS DE EVALUACIÓN	OBJETIVOS DE APRENDIZAJE GENÉRICOS	
1. Construye algoritmos y diagramas de flujo, en forma estructurada, empleando sentencias necesarias para solucionar requerimientos simples de programación, según estándares de la industria.	<b>1.1</b> Entrevista y registra los requerimientos planteados por un usuario en relación a un problema que requiere de programación simple.	A	
	<b>1.2</b> Registra los pasos necesarios que requieren de datos y acciones tales como: control del flujo, cálculos, decisiones e iteraciones para resolver el problema planteado, según protocolo de registro.	C	
	<b>1.3</b> Determina las relaciones, e interacciones secuenciales y operaciones de cálculo con datos necesarios para resolver un problema simple.	C	
	<b>1.4</b> Ordena los pasos que constituyen el algoritmo secuencial, selectivo y/o repetitivos, con las instrucciones y operaciones necesarias para dar solución efectiva a un problema en un mínimo de pasos, respetando reglas generales de los algoritmos.	C	

APRENDIZAJES ESPERADOS	CRITERIOS DE EVALUACIÓN	OBJETIVOS DE APRENDIZAJE GENÉRICOS
	<p><b>1.5</b> Crea rutinas algorítmicas para resolver un problema o procedimiento particular que se repite frecuentemente en otros algoritmos mayores, de acuerdo a reglas de representación.</p>	C
	<p><b>1.6</b> Diseña diagrama de flujo que represente la estructura y cada sentencia que compone un algoritmo, utilizando cajas y conectores correspondientes, según normas de diagramación.</p>	C
<p><b>2.</b> Desarrolla programas simples empleando la programación estructurada y/o modular, para resolver sistemas de información a requerimiento de la empresa, según estándares de programación.</p>	<p><b>2.1</b> Selecciona el lenguaje de programación apropiado en función del área de aplicación y recursos disponibles, considerando normas de licenciamiento.</p>	C
	<p><b>2.2</b> Codifica un programa, según el orden y estructura de un algoritmo y/o diagrama de flujo simple, utilizando los comandos y sentencias del lenguaje, respetando la sintaxis según guías de referencia.</p>	H
	<p><b>2.3</b> Ejecuta un programa simple, utilizando los procedimientos y recursos de <i>hardware</i> y <i>software</i> que aseguren su correcto funcionamiento.</p>	C
	<p><b>2.4</b> Soluciona problemas de error en programación, haciendo uso de herramientas de <i>software</i>, validando resultados y buscando alternativas de reprogramación para su corrección.</p>	C
	<p><b>2.5</b> Desarrolla un programa básico en forma modular, de acuerdo a protocolo, en un lenguaje apropiado al área de aplicación del problema a resolver, de acuerdo a estándares de la industrial.</p>	C
	<p><b>2.6</b> Desarrolla un programa básico en forma estructurada de acuerdo a especificaciones haciendo uso de lenguaje apropiado al área de aplicación del problema a resolver, de acuerdo a estándares.</p>	C
	<p><b>2.7</b> Grafica un diagrama, seleccionando los componentes propios de la estructura de un programa orientado a objetos, haciendo uso de <i>software</i> específico y de acuerdo a protocolos establecidos.</p>	C

APRENDIZAJES ESPERADOS	CRITERIOS DE EVALUACIÓN	OBJETIVOS DE APRENDIZAJE GENÉRICOS
<p><b>3.</b> Desarrolla programas de mediana complejidad, utilizando la orientación a objetos que involucre estructuras y bases de datos, aplicando “buenas prácticas” definidas por la industria.</p>	<p><b>3.1</b> Selecciona el lenguaje de programación apropiado a un problema por resolver, en función del área de aplicación y recursos disponibles.</p>	<p>C</p>
	<p><b>3.2</b> Desarrolla un programa de computación de mediana complejidad sobre la base de lenguajes orientados a objetos; considerando las características del problema a resolver, ventajas, inconvenientes y grado de implementación del mismo aplicando estándares de la industria.</p>	<p>C</p>
	<p><b>3.3</b> Diagrama programación orientada a objetos identificando: clases y relación entre ellas, objeto y su relación con otro objeto, paso de mensajes, responsabilidades, generalización, polimorfismo y enlace dinámico, interfaz e implementación; respetando protocolo de programación.</p>	<p>C</p>
	<p><b>3.4</b> Desarrolla aplicaciones básicas, que permiten realizar operaciones de gestión sobre bases de datos orientada a objetos, aplicando técnicas para mantener la persistencia de la información.</p>	<p>C</p>
	<p><b>3.5</b> Desarrolla programas que gestionan información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos, usando un entorno integrado de desarrollo.</p>	<p>C</p>
	<p><b>3.6</b> Aplica técnicas de reutilización del <i>software</i> (composición, herencia, polimorfismo) más adecuadas a la solución de un problema dado, basándose en la tecnología de objeto.</p>	<p>C</p>

## EJEMPLO DE ACTIVIDAD DE APRENDIZAJE

NOMBRE DEL MÓDULO	<b>Programación y bases de datos</b>
NOMBRE DE LA ACTIVIDAD DE APRENDIZAJE	Construcción de algoritmos
DURACIÓN DE LA ACTIVIDAD	6 horas
<b>APRENDIZAJES ESPERADOS</b>	<b>CRITERIOS DE EVALUACIÓN QUE INCLUYE</b>
<p><b>1.</b> Construye algoritmos y diagramas de flujo, en forma estructurada, empleando sentencias necesarias para solucionar requerimientos simples de programación, según estándares de la industria.</p>	<p>1.4 Ordena los pasos que constituyen el algoritmo secuencial, selectivo y/o repetitivos, con las instrucciones y operaciones necesarias para dar solución efectiva a un problema en un mínimo de pasos, respetando reglas generales de los algoritmos.</p>
METODOLOGÍAS SELECCIONADAS	Estudio guiado
<b>DESCRIPCIÓN DE LAS TAREAS QUE REALIZAN DOCENTES Y ESTUDIANTES, Y LOS RECURSOS QUE SE UTILIZAN EN CADA UNA DE LAS SIGUIENTES ETAPAS</b>	
<p><b>PREPARACIÓN DE LA ACTIVIDAD</b></p>	<p><b>Docente:</b></p> <ul style="list-style-type: none"> <li>› Diseña una guía de trabajo sobre construcción de algoritmos para el trabajo en el taller.</li> <li>› Crea una guía de aprendizaje sobre algoritmos).</li> </ul> <p><b>Recursos:</b></p> <ul style="list-style-type: none"> <li>› Laboratorio de computación con conexión a internet.</li> <li>› Guía de aprendizaje.</li> <li>› Equipo para proyección.</li> </ul>

DESCRIPCIÓN DE LAS TAREAS QUE REALIZAN DOCENTES Y ESTUDIANTES, Y LOS RECURSOS QUE SE UTILIZAN EN CADA UNA DE LAS SIGUIENTES ETAPAS

EJECUCIÓN	<p><b>Docente:</b></p> <ul style="list-style-type: none"> <li>› Organiza a sus estudiantes en grupos de cuatro.</li> <li>› Presenta una introducción con respecto a los algoritmos.</li> <li>› Explica el trabajo a realizar en cada grupo.</li> <li>› Entrega la guía de aprendizaje, en la que se señala el tema y los procedimientos de investigación relacionados con el trabajo a realizar.</li> </ul> <p><b>Estudiantes:</b></p> <ul style="list-style-type: none"> <li>› Observan la presentación y explicación dada antes de la actividad.</li> <li>› Organizan las tareas para desarrollar el trabajo.</li> <li>› Analizan la guía y formulan consultas.</li> <li>› Investigan en internet acerca de los algoritmos, su estructura y aplicación.</li> <li>› Debaten sobre procedimientos cotidianos.</li> <li>› Conducen en el planteamiento de un problema determinado para construir su algoritmo de solución.</li> <li>› Construyen el algoritmo siguiendo pautas investigadas.</li> <li>› Debaten sobre la validez del algoritmo con respecto a la solución esperada y efectúan correcciones, si es necesario.</li> <li>› Debaten conclusiones en el grupo y las presentan en una disertación, en la que incluyen el algoritmo y los antecedentes investigados.</li> </ul>
CIERRE	<p><b>Estudiantes:</b></p> <ul style="list-style-type: none"> <li>› Comentan y debaten los problemas derivados de la investigación y concluyen resultados con respecto a la creación de algoritmos.</li> </ul> <p><b>Docente:</b></p> <ul style="list-style-type: none"> <li>› Retroalimenta los trabajos presentados y aclara algunas ideas expuestas.</li> <li>› Complementa con información acerca de la estructura, componentes y tipos de algoritmo, y agrega ejercicios de aplicación para ilustrar.</li> </ul>

## EJEMPLO DE ACTIVIDAD DE APRENDIZAJE

NOMBRE DEL MÓDULO	<b>Programación y bases de datos</b>
NOMBRE DE LA ACTIVIDAD DE APRENDIZAJE	Diagrama de flujo para representar un algoritmo que soluciona un problema determinado
DURACIÓN DE LA ACTIVIDAD	6 horas
<b>APRENDIZAJES ESPERADOS</b>	<b>CRITERIOS DE EVALUACIÓN QUE INCLUYE</b>
<b>1.</b> Construye algoritmos y diagramas de flujo, en forma estructurada, empleando sentencias necesarias para solucionar requerimientos simples de programación, según estándares de la industria.	1.6 Diseña diagrama de flujo que represente la estructura y cada sentencia que compone un algoritmo, utilizando cajas y conectores correspondientes, según normas de diagramación.
METODOLOGÍAS SELECCIONADAS	Estudio de caso

### DESCRIPCIÓN DE LAS TAREAS QUE REALIZAN DOCENTES Y ESTUDIANTES, Y LOS RECURSOS QUE SE UTILIZAN EN CADA UNA DE LAS SIGUIENTES ETAPAS

#### PREPARACIÓN DE LA ACTIVIDAD

##### **Docente:**

- › Prepara el laboratorio con puestos de trabajo y energía disponible.
- › Prepara una clase expositiva acerca de las cajas empleadas en el diagrama de flujo.
- › Elabora una guía de trabajo y procedimientos, que incluye el algoritmo a representar.
- › Selecciona manuales de uso y referencia de *software* de apoyo.
- › Dispone de recursos.

##### **Recursos:**

- › Laboratorio con puestos de trabajo y energía disponible.
- › Computador.
- › NetBeans o DIA.
- › Presentación o video tutorial.
- › Manuales de uso y referencia de *software* de apoyo.



DESCRIPCIÓN DE LAS TAREAS QUE REALIZAN DOCENTES Y ESTUDIANTES, Y LOS RECURSOS QUE SE UTILIZAN EN CADA UNA DE LAS SIGUIENTES ETAPAS

<p><b>EJECUCIÓN</b></p>	<p><b>Docente:</b></p> <ul style="list-style-type: none"> <li>› Realiza una presentación en la que expone las cajas típicas que se utilizan en la representación de cada tipo de proceso constituyente de un algoritmo.</li> <li>› Muestra un ejemplo de algoritmo y, luego, su correspondiente representación en un diagrama de flujo.</li> <li>› Realiza un análisis de dicha representación.</li> <li>› Presenta un caso.</li> <li>› Entrega la guía de procedimiento de trabajo a sus estudiantes, la que incluye el algoritmo a representar.</li> <li>› Entrega los manuales y guías de referencia del <i>software</i> a utilizar para la creación gráfica del diagrama de flujo correspondiente.</li> </ul> <p><b>Estudiantes:</b></p> <ul style="list-style-type: none"> <li>› Escuchan la presentación y explicación dada antes de la actividad.</li> <li>› Leen y revisan guía y manuales entregados.</li> <li>› Analizan el caso y los datos del requerimiento de representación.</li> <li>› Activan los programas de entorno necesarios para trabajar.</li> <li>› Crean el diagrama de flujo correspondiente, usando el <i>software</i> de apoyo.</li> <li>› Autoevalúan sus resultados.</li> <li>› Reconocen errores y corrigen.</li> </ul>
<p><b>CIERRE</b></p>	<p><b>Estudiantes:</b></p> <ul style="list-style-type: none"> <li>› Presentan sus resultados en una puesta en común y las soluciones.</li> </ul> <p><b>Docente:</b></p> <ul style="list-style-type: none"> <li>› Comenta los resultados y otras alternativas de solución, haciendo notar que dicha representación puede hacerse a través medio, no solo digital.</li> </ul>



## EJEMPLO DE ACTIVIDAD DE EVALUACIÓN

NOMBRE DEL MÓDULO	Programación y bases de datos	
APRENDIZAJE ESPERADO	CRITERIOS DE EVALUACIÓN	OBJETIVOS DE APRENDIZAJE GENÉRICOS A EVALUAR
1. Construye algoritmos y diagramas de flujo, en forma estructurada, empleando sentencias necesarias para solucionar requerimientos simples de programación, según estándares de la industria.	1.6 Diseña diagrama de flujo que represente la estructura y cada sentencia que compone un algoritmo, utilizando cajas y conectores correspondientes, según normas de diagramación.	C Realizar las tareas de manera prolija, cumpliendo plazos establecidos y estándares de calidad, y buscando alternativas y soluciones cuando se presentan problemas pertinentes a las funciones desempeñadas.

### Selección de cómo evaluar

DESCRIPCIÓN DE ACTIVIDADES DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN SELECCIONADOS
<p>Diagrama de flujo para representar un algoritmo que soluciona un problema determinado.</p> <p>Se evalúa por medio de una actividad práctica de taller. El o la estudiante realiza la actividad, siguiendo el procedimiento establecido haciendo uso del <i>software</i> gráfico adecuado, con orientación a resultados, según la reglas algorítmicas y de representación en diagramas de flujo.</p>	Escala de apreciación.

### Ejemplo de escala de apreciación

INDICADORES	CONCEPTOS			
	SIEMPRE	GENERALMENTE	OCASIONALMENTE	NUNCA
Aplica la representación establecida.				
Usa herramientas correctamente.				
Presenta orientación a resultados.				
Optimiza los tiempos.				
Analiza los resultados.				
Corrige los errores.				
Presenta la solución del problema.				

## BIBLIOGRAFÍA

**Horstmann, C. y Cornell, G.** (2006). *Core java 2*. Madrid: Pearson Education.

**Luque, I.** (2002). *Bases de datos: Desde Chen hasta Codd con Oracle*. Ciudad de México: Alfaomega.

**Schildt, H.** (2005). *La Biblia de Java 2 v5.0*. Madrid: Anaya Multimedia.

**Sun Microsystems, Inc.** (2008). *Java programming language SL-275-SE6*. California: Sun Microsystems.

**Weiss, M. A.** (2005). *Estructuras de datos y algoritmos*. Delaware: Addison-Wesley Iberoamericana.

## Sitios web recomendados

Java:

<http://www.javaya.com.ar/>

Oracle:

<http://docs.oracle.com/javase/6/docs/>

<http://docs.oracle.com/javase/tutorial/index.html>

(Los sitios web y enlaces sugeridos en este Programa fueron revisados en marzo de 2015).