

1. Datos Generales de la asignatura

Nombre de la asignatura:	Programación Orientada a Objetos
Clave de la asignatura:	IBC-1023
SATCA¹:	2-2-4
Carrera:	Ingeniería Biomédica

2. Presentación

Caracterización de la asignatura
<p>Esta asignatura aporta al perfil del Ingeniero Biomédico, la capacidad de desarrollar programas mediante la programación orientado a objetos.</p> <p>Se analiza, diseña y desarrolla aplicaciones enfocadas al área biomédica, en un lenguaje de programación orientada a objetos,</p> <p>Consiste en conocer y trabajar con un lenguaje de programación orientado a objetos para resolver problemas de la ingeniería biomédica.</p> <p>Esta materia da soporte a la materia de tecnologías de base de datos ya que proporciona conocimientos para trabajar con cualquier lenguaje orientado a objetos, metodología de análisis y diseño orientado a objetos, de los sistemas gestores de bases de datos.</p>
Intención didáctica
<p>El alumno aprenderá a programar usando el paradigma orientado a objetos para generar soluciones computacionales en el área biomédica.</p> <p>La primera y segunda unidad introducen los conceptos fundamentales del modelo de objetos (clases y métodos) y su aplicación al desarrollo de programas.</p> <p>La tercera unidad proporciona herramientas para la generación de código reusable (herencia y polimorfismo).</p> <p>La cuarta unidad permite al alumno usar bibliotecas predefinidas, para la resolución de problemas que requieren la aplicación de estructuras de datos.</p> <p>La quinta unidad aborda diferentes algoritmos de búsqueda y ordenamiento internos y externos.</p>

¹ Sistema de Asignación y Transferencia de Créditos Académicos

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Evento
Instituto Tecnológico de Mérida del 29 de septiembre al 1 de octubre de 2010.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Apizaco, Ensenada, La Paz, Mérida, Mexicali, Saltillo, San Luis Potosí, Tijuana, Pachuca y Veracruz.	Reunión Nacional de Diseño e Innovación Curricular para el Desarrollo y Formación de Competencias Profesionales de la Carrera de Ingeniería Biomédica.
Instituto Tecnológico de Mérida del 1 al 3 de diciembre de 2010.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Apizaco, Ensenada, La Paz, Mérida, Mexicali, Pachuca, Saltillo, San Luis Potosí, Tijuana y Veracruz.	Reunión Nacional de Consolidación de los Programas en Competencias Profesionales de la Carrera de Ingeniería Biomédica.
Instituto Tecnológico de Mérida del 26 y 27 de octubre de 2011.	Representantes de los Institutos Tecnológicos de: Apizaco, Mérida, Pachuca y San Luis Potosí.	
Instituto Tecnológico de Hermosillo del 26 al 29 de noviembre de 2013.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Boca del Río, Hermosillo, Mérida, Orizaba, Purhepecha, Saltillo, Tijuana.	Reunión de Seguimiento Curricular de la Carrera de Ingeniería Biomédica.
Instituto Tecnológico de Toluca, del 10 al 13 de febrero de 2014.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Boca del Río, Celaya, Mérida, Orizaba, Puerto Vallarta y Veracruz.	Reunión de Seguimiento Curricular de los Programas Educativos de Ingenierías, Licenciaturas y Asignaturas Comunes del SNIT.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Analiza, diseña, implementa y aplica software utilizando la programación orientada a objetos, para resolver problemas biomédicos.

5. Competencias previas

Analiza, diseña y desarrolla soluciones a problemas conceptuales en la ingeniería biomédica utilizando algoritmos computacionales e implementarlos con un lenguaje de programación estructurado.
--

6. Temario

No.	Temas	Subtemas
1	Clases y objetos	<ul style="list-style-type: none"> 1.1. Introducción al modelado de objetos con UML (diagrama de casos de uso) 1.2. Definición de una clase. 1.3. Representación UML 1.4. Declaración de clases. 1.5. Miembros de una clase (encapsulamiento). 1.6. Especificadores de acceso. 1.7. Creación de objetos. 1.8. Puntero this. 1.9. Constructores y destructores. 1.10. Software para Análisis y Diseño
2	Métodos	<ul style="list-style-type: none"> 2.1 Definición de un método. 2.2 Estructura de un método. 2.3 Valor de retorno. 2.4 Declaración de un método. 2.5 Ámbito y tiempo de vida de variables. 2.6 Argumentos y paso de parámetros. 2.7 Sobrecarga de métodos. 2.8 Manejo de excepciones.
3	Herencia y polimorfismo	<ul style="list-style-type: none"> 3.1 Concepto de herencia y polimorfismo. 3.2 Definición de una clase base. 3.3 Definición de una clase derivada. 3.4 Clases abstractas e interfaces. 3.5 Bibliotecas de clases

4	Estructuras de datos	<p>4.1 Conceptos de Pilas, Colas, Listas, Árboles</p> <p>4.2 Uso de bibliotecas de clases en problemas de estructura de datos.</p>
5	Ordenamiento y búsqueda	<p>5.1 Algoritmos de ordenamiento</p> <p>5.1.1 Burbuja</p> <p>5.1.2 Mezcla</p> <p>5.1.3 Inserción directa</p> <p>5.1.4 Inserción Indirecta</p> <p>5.1.5 QuickSort</p> <p>5.1.6 ShellSort, entre otros.</p> <p>5.2 Algoritmos de búsqueda</p> <p>5.2.1 Búsqueda Secuencial</p> <p>5.2.2 Búsqueda binaria</p> <p>5.3 Búsqueda indexada</p>

7. Actividades de aprendizaje de los temas

Clases y Objetos	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Desarrolla clases y objetos en un lenguaje de programación orientado a objetos para la solución de problemas con herramientas UML.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Capacidad de análisis y síntesis. • Uso de las Tic's. • Capacidad crítica y autocrítica. • Trabajo en equipo. • Capacidad de traducir el contexto 	<ul style="list-style-type: none"> • Elaborar modelo de casos de uso y de clases con UML • Realizar ejercicios para la creación de clases y objetos. • Investigar la inicialización de un objeto. • Utilizar en ejercicios los modificadores de acceso. • Utilizar las clases definidas en el lenguaje orientado a objetos

<p>del ambiente al contexto lógico computacional.</p> <ul style="list-style-type: none"> • Capacidad de aplicar los conocimientos en la práctica. • Búsqueda del logro 	
Métodos	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Implementa métodos para definir el comportamiento de los objetos en una aplicación</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Habilidades necesarias de manejo de la computadora. • Capacidad crítica y autocrítica. • Trabajo en equipo. • Compromiso. • Capacidad de traducir el contexto del ambiente al contexto lógico computacional. • Capacidad de relacionarse con otras áreas interdisciplinarias • Capacidad de aplicar los conocimientos en la práctica. • Capacidad de autoaprender. • Capacidad de generar nuevas ideas. • Búsqueda del logro 	<ul style="list-style-type: none"> • Desarrollar ejercicios en donde se definan los comportamientos de los objetos. • Realizar ejercicios haciendo uso de excepciones como try, catch y finally.

Herencia y Polimorfismo	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Aplicar la herencia y polimorfismo para elevar la reusabilidad del código en aplicaciones biomédicas</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Habilidades necesarias de manejo de la computadora. • Capacidad crítica y autocrítica. • Trabajo en equipo. • Compromiso. • Capacidad de traducir el contexto del ambiente al contexto lógico computacional. • Capacidad de relacionarse con otras áreas interdisciplinarias • Capacidad de aplicar los conocimientos en la práctica. • Capacidad de autoaprender. • Capacidad de generar nuevas ideas. • Búsqueda del logro 	<ul style="list-style-type: none"> • Desarrollar y analizar las semejanzas y diferencias entre herencia y polimorfismo para determinar las posibles relaciones entre ellos. • Crear aplicaciones en donde se pueda demostrar el polimorfismo y razonar su utilidad. • Reutilizar código en el desarrollo de aplicaciones. • Aplicar el concepto de clases abstractas e interfaces. • Crear clases y jerarquías de clases para aplicarlas a problemas reales. • Plantear y resolver problemas informáticos mediante la utilización del polimorfismo apoyados en UML.
Estructuras de datos	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Identifica e interpreta bibliotecas externas de clases para su implementación de estructuras de datos en aplicaciones</p>	<ul style="list-style-type: none"> • Analizar por grupos el funcionamiento de pilas, colas, listas y árboles para presentar en plenaria las conclusiones de cada grupo. • Desarrollar aplicaciones que hagan uso de diferentes estructuras de

<p>biomédicas.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Habilidades necesarias de manejo de la computadora. • Capacidad crítica y autocrítica. • Trabajo en equipo. • Compromiso. • Capacidad de traducir el contexto del ambiente al contexto lógico computacional. • Capacidad de relacionarse con otras áreas interdisciplinarias • Capacidad de aplicar los conocimientos en la práctica. • Capacidad de autoaprender. • Capacidad de generar nuevas ideas. • Búsqueda del logro 	<p>datos rehusando las clases contenedoras predefinidas en el lenguaje de programación</p>
<p>Ordenamiento y búsqueda</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <p>Aplica algoritmos de ordenamiento y búsqueda de acuerdo a los requerimientos de las aplicaciones para necesidades biomédicas.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. 	<ul style="list-style-type: none"> • Investigar en grupos el funcionamiento de diferentes algoritmos de ordenamiento y búsqueda y discutir de manera grupal el funcionamiento. • Implementar diferentes algoritmos de ordenamiento y búsqueda en un lenguaje orientado a objetos

<ul style="list-style-type: none">• Capacidad de análisis y síntesis.• Capacidad de organizar y planificar.• Habilidades necesarias de manejo de la computadora.• Capacidad crítica y autocrítica.• Trabajo en equipo.• Compromiso.• Capacidad de traducir el contexto del ambiente al contexto lógico computacional.• Capacidad de relacionarse con otras áreas interdisciplinarias• Capacidad de aplicar los conocimientos en la práctica.• Capacidad de autoaprender.• Capacidad de generar nuevas ideas.• Búsqueda del logro	
---	--

8. Práctica(s)

<ul style="list-style-type: none">• Elaborar la definición de clases utilizando un lenguaje de programación a partir de problemas proporcionados por el maestro.• Implementar aplicaciones que utilicen clases con comportamientos que impliquen el uso de estructuras secuenciales y expresiones aritméticas y lógicas.• Implementar aplicaciones que utilicen clases con comportamientos que impliquen el uso de estructuras selectivas, haciendo uso de una herramienta de depuración de aplicación.• Implementar aplicaciones que utilicen clases con comportamientos que impliquen el uso de estructuras repetitivas, haciendo uso de una herramienta de depuración de aplicaciones.• Implementar y modelar aplicaciones usando los conceptos orientados a objetos. (Herencia, polimorfismo, clases abstractas e interfases).• Implementar algoritmos de ordenamiento y búsqueda usando bibliotecas de clases predefinidas en el lenguaje de programación.
--

9. Proyecto de asignatura

El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

- **Fundamentación:** marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.
- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

10. Evaluación por competencias

La evaluación debe ser continua y formativa por lo que se debe considerar el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Reportes escritos de las soluciones a problemas desarrollados dentro y fuera de clase.
- Información obtenida durante las investigaciones solicitadas plasmada en documentos escritos.
- Exámenes escritos para comprobar el manejo de aspectos teóricos.
- Desarrollo de programas mediante el uso de un lenguaje orientado a objetos.
- Elaboración de proyectos, donde el estudiante resuelva problemas de su entorno mediante la programación orientada a objetos.
- Descripción de otras experiencias concretas que podrían realizarse adicionalmente (participación, integración, entrega de proyectos en tiempo, etc.).

11. Fuentes de información

1. Booch G., *Análisis y diseño orientado a objetos con aplicaciones*, Addison-Wesley, 1996.
2. Rumbaugh J., Blaha M., Premerlani W., Hedy F., Lorensen W., *Modelado y diseño orientado a objetos*, Prentice-Hall, 1996.
3. Arnold K., Gosling J., Holmes D., *The Java™ Programming Language*, 3rd Edition, Addison Wesley Professional, 2000.
4. Stroustrup B., *El lenguaje de programación C++*, Segunda edición, Addison-Wesley Iberoamericana, 1993.
5. Joyanes AL., *C++ a su alcance, un enfoque orientado a objetos*, McGraw Hill, 1994.
6. Humphrey WS., *Introduction to the Personal Software Process*, Addison-Wesley, 1997.
7. Deitel HM., Deitel PJ., *C++ How to Program*, 5th edition, Prentice-Hall, 2005.
8. Deitel HM., Deitel PJ., *Java How to Program*, 6th edition, Prentice-Hall, 2004.
9. Cairo, Osvaldo; Guardati, Silvia. *Estructura de Datos*, Tercera Edición. Mc Graw Hill, México, 2006.
10. Roman Martinez, Elda Quriga. *Estructura de Datos Referencia practica con orientación a objetos*. Ed. Thomson, Mexico, 2004.
11. Joyanes Aguilar, Luis. *Programación en C++ Algoritmos, Estructuras de Datos y Objetos*. Segunda Edición 2006. McGraw – Hill.
12. Guardati, Silvia. *Estructura de Datos Orientada a Objetos Algoritmos con C++*, Primera Edición. Prentice Hall, México, 2007.
13. Mark Allen Weiss. *Estructura de datos en Java*. Ed. Addison Wesley.
14. C. Thomas Wu. *Introducción a la Programación Orientada a Objetos con Java*. Ed. Pearson Educación.