



Fecha: febrero 2025

PROGRAMA ACADÉMICO: INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

SEMESTRE: SEGUNDO

ASIGNATURA: PROGRAMACIÓN I

CÓDIGO: 8108255

NÚMERO DE CRÉDITOS: 4

PRESENTACIÓN

El proceso formativo del Ingeniero de Sistemas y Computación requiere del dominio de temáticas particulares de la programación orientada a objetos, por ello se profundiza en los principios básicos de este paradigma (herencia, polimorfismo, abstracción y encapsulamiento...), además de las relaciones que existen entre las clases llevando dichos conceptos a una representación formal que permita contribuir a la solución de problemas reales.

Adicionalmente esta asignatura trabaja en el fortalecimiento de la algoritmia, representación de datos y los elementos que permitan construir un programa para que sea funcional y responda a las necesidades de los usuarios usando una arquitectura multicapa, por ejemplo, algunos tipos de pruebas y el manejo de errores.

Por último, se presentan al estudiante diferentes herramientas usadas en el proceso del ciclo de vida del desarrollo de software.

JUSTIFICACIÓN

En el mundo laboral, se requieren profesionales capaces de afrontar situaciones reales y dinámicas, que estén focalizados en la gestión y desarrollo de las mejoras continuas, por ello esta asignatura contribuye a formar las bases fundamentales, del futuro Ingeniero de Sistemas y Computación, que esté en la capacidad de aplicar los conocimientos adquiridos en la construcción de soluciones a los problemas presentados, haciendo uso de diferentes herramientas computacionales.

Luego de analizar el ámbito internacional es evidente que una de las habilidades más demandadas por las compañías de Tecnología Informática, es la de desarrollar aplicaciones, según la “Encuesta de Salarios IT 2017”, realizada por el Portal Computer World; en el campo nacional el panorama es similar teniendo en cuenta el “Estudio de salarios y profesionales del sector de Software y TI de Colombia 2016”, elaborado por MinTIC y FEDESOFTE, donde se presentan los perfiles más solicitados en la industria nacional, destacándose los relacionados con el área de desarrollo, lo que es corroborado con otros estudios que señalan el déficit de profesionales con estos perfiles, por ello la estructura de esta asignatura contribuye de manera efectiva a responder a las necesidades de esta industria.

Por otra parte, es justo mencionar que las habilidades que se desarrollan en esta asignatura, sirven como base para el abordaje de asignaturas de otras áreas que hacen parte del plan de estudios del Programa.

Por último, es importante resaltar que la lógica de programación actualmente se considera una habilidad transversal, debido a que contribuye al desarrollo del pensamiento lógico y matemático, que se requiere en cualquier campo de la Ingeniería. Existen, además, comunidades académicas que han identificado la necesidad de incluir temas de lógica de programación, desde tempranas edades en los primeros niveles de escolaridad.



COMPETENCIAS

Adquirir una visión global de la Ingeniería de sistemas en el campo laboral y construir modelos de solución algorítmicas focalizadas en las bases de la lógica y la programación, con el fin de construir soluciones reales, viables y de eficaz implementación en un lenguaje de programación.

- Identifica y aplica los fundamentos y principios básicos de la programación orientada a objetos
- Analiza un problema y propone alternativas de solución utilizan diagramas estandarizados
- Diseña y ejecuta pruebas unitarias para generar software de calidad
- Identifica los diferentes tipos de errores y utiliza el manejo de excepciones para obtener aplicaciones robustas
- Genera componentes de código reutilizables aplicando conceptos de herencia y el polimorfismo creando
- Utiliza de forma adecuada las estructuras de datos unidimensionales y bidimensionales para dar solución a problemas
- Conoce y aplica algoritmos de ordenamiento y búsqueda en estructuras de datos unidimensionales y bidimensionales
- Identifica diferentes formas para la organización de la información dentro de estructuras de datos
- Utiliza herramientas de software para soportar los procesos de las etapas del ciclo de vida del desarrollo de software
- Conoce y aplica los principios de única responsabilidad y separación de conceptos
- Trabaja en equipo para dar solución a problemas planteados
- Genera código de calidad aplicando buenas prácticas de programación

RESULTADOS DE APRENDIZAJE

- Conozco los fundamentos y principios de la programación orientada a objetos.
- Utilizo de forma adecuada las pruebas unitarias, arreglos y matrices, así como algoritmos de ordenamiento y búsqueda en estas estructuras, en la construcción de aplicaciones y generación de componentes de código reutilizables, a partir del trabajo en equipo y aplicando buenas prácticas de codificación.
- Empleo herramientas de software para soportar los procesos de las etapas del ciclo de vida del desarrollo de software
- Leo documentos técnicos en idioma inglés y utilizo términos propios del mismo en el proceso de codificación.

METODOLOGÍA

Esta asignatura será guiada en los espacios presenciales por la complementación conceptual del docente al trabajo de preparación previo que los estudiantes han realizado sobre la temática particular a tratar en la sesión; por lo tanto, un tema será abordado en cuatro momentos:

1. Preparación, consulta e investigación conceptual por cuenta del estudiante y su grupo de trabajo.
2. Tratamiento conceptual del tema en sesión del gran grupo junto con el docente.
3. Aplicación de talleres individuales y cooperativos a nivel tutorial.
4. Desarrollo de actividades de refuerzo en sesiones autónomas.
5. Se tendrá en cuenta el aprendizaje basado en proyectos

De lo anterior se verifica que en la actividad 1, el estudiante constituirá conflictos conceptuales de baja complejidad, a solucionar en el transcurso de la actividad 2, entre tanto, la actividad 4 generará conflictos cognitivos orientados a la aplicación, a subsanar con la actividad 3.



Se considera que el estudiante debe adquirir habilidades para el trabajo autónomo, de tal manera que le permitan en todo momento estar a la vanguardia de los adelantos tecnológicos, los cuales será complementado por el trabajo desarrollado en las sesiones de clase.

Las temáticas de la programación orientada a objetos, debe orientarse teniendo como eje principal el análisis y diseño de software, (UML), utilizando buenas y mejores prácticas de programación, proponiendo soluciones óptimas en la construcción de algoritmos.

Para el desarrollo de la asignatura se utilizará el IDE ECLIPSE, y queda a criterio del docente plantear uno de los temas sin IDE.

INVESTIGACIÓN

La asignatura por sus condiciones particulares, promueve en el estudiante la investigación formativa, de modo sea posible obtener los fundamentos con los que podrá posteriormente vincularse en el tratamiento de áreas específicas de la disciplina Informática, por medio de los grupos y semilleros de investigación dinamizados al interior de la Escuela de Ingeniería de Sistemas y Computación.

MEDIOS AUDIOVISUALES

Aula virtual Moodle.

Salas de informática.

La asignatura es eminentemente práctica, de manera que se hace necesario que todas sus clases sean cumplidas en salas de informática.

EVALUACIÓN

EVALUACIÓN COLECTIVA

No aplica.

EVALUACIÓN INDIVIDUAL

La evaluación en la asignatura está orientada a determinar el nivel de desarrollo de los procesos lógicos en el estudiante, junto con su capacidad para abstraer problemas y generar soluciones informáticas.

RANGOS DE VALORACIÓN (%)

El cálculo de la nota final se hará de la siguiente manera:

- 70% Evaluaciones
- 15% Talleres y exposiciones
- 15% Trabajos

CONTENIDOS TEMÁTICOS CENTRALES

1. USO DE PROGRAMACIÓN ORIENTADA A OBJETOS (POO)

- 1.1. Características (encapsulamiento, sobrecarga, visibilidad)
- 1.2. Tipos de métodos y alcance de atributos
- 1.3. Buenas prácticas de programación
- 1.4. Estándar de programación y documentación Java-doc

2. LENGUAJE DE MODELADO UNIFICADO (UML)

- 2.1. Diagrama de clases
- 2.2. Clases (atributos, métodos, modificadores, clases abstractas, interfaces)
- 2.3. Relaciones (dependencia, asociación, herencia, composición, agregación, realización)
- 2.4. Multiplicidad

3. HERENCIA Y POLIMORFISMO

- 3.1. Conceptos



- 3.2. Clases abstractas
- 3.3. Interfaces
- 3.4. Métodos abstractos
- 3.5. Sobreescritura de métodos
- 4. INTERACCION CON EL USUARIO USANDO PATRON MVP Y DIALOGOS DEL SISTEMA**
- 5. API PARA MANEJO DE FECHAS**
- 6. ADMINISTRACIÓN DE EXCEPCIONES**
 - 6.1. Crear una excepción
 - 6.2. Pasar las excepciones entre clases
 - 6.3. Tratar la excepción en el momento
- 7. PRUEBAS UNITARIAS**
 - 7.1. Invariantes de una clase
 - 7.2. Métodos para verificación
 - 7.3. Implementación de pruebas automáticas
 - 7.4. Pruebas automáticas para el manejo de excepciones
- 8. ARREGLOS UNIDIMENSIONALES y MULTIDIMENSIONALES**
 - 8.1. Declaración, creación e inicialización de arreglos
 - 8.2. Arreglos representando multiplicidad
 - 8.3. Problemas con arreglos
 - 8.4. Algoritmos de ordenamiento
- 9. COLECCIONES**
 - 9.1. Interface List
 - 9.2. Tablas Hashing
 - 9.3. Interfaces Map y Set
 - 9.4. Clases HasMap y HasTable

LECTURAS MÍNIMAS

Artículos y apartados bibliográficos de los diferentes temas a tratar, proporcionados por el Docente. Consulta permanente a los foros de las comunidades de desarrolladores.

BIBLIOGRAFÍA

- DEITEL, Harvey M; DEITEL Paul J. Cómo programar en C/C++ y Java. Pearson. 1113 p., 2004.
- Sznajdleder, Pablo. Java a fondo: estudio del lenguaje y desarrollo de aplicaciones. Alfaomega. 2013
- Wu, Introducción a la programación orientada a objetos con Java. McGraw-Hill. 2001
- Torres, Manuel. Desarrollo de aplicaciones con Java. Macro EIRL. 2014
- Joyanes, Luis; Zahonero, Ignacio. Programación en C, C++, Java y UML. McGraw-Hill. 2014.
- Martín, Antonio. Programador certificado Java 2 curso práctico. Alfaomega. 2008.
- Eck David, Programming: Introduction to Programming Using JAVA, Independiente, 2016.
- Martin C. Robert, Código limpio / Clean code: Manual de estilo para el desarrollo ágil de software / A Handbook of Agile Software Craftsmanship (Spanish Edition), Anaya Multimedia-Anaya Interactiva, 2012.
- Sierra Kathy y Bates Bert, Head First Java, 2nd Edition, O'Reilly Media, 2005.
- Eckel Bruce, Piensa en Java, Prentice Hall, 2007.
- Deitel Harvey y Deitel Paul, Java How To Program (Early Objects) (10th Edition), Pearson, 2014.
- Fowler Martin, UML Distilled: A Brief Guide to the Standard Object Modeling Language, Addison-Wesley Professional, 2004.
- Perry Steven, Introducción a la programación Java, parte 1, IBM, 2012. En: <https://www.ibm.com/developerworks/ssa/java/tutorials/j-introjava1/index.html>.
- Reges, Stuart. Building Java programs. Pearson education. 2018. ISN: 9781292161686. <https://www-ebooks7-24-com.biblio.uptc.edu.co/?il=6292&pg=7>



MACROPROCESO: DOCENCIA
PROCESO: GESTIÓN DE PROGRAMAS ACADÉMICOS
PROCEDIMIENTO: FORMULACION O ACTUALIZACION DEL PROYECTO ACADEMICO EDUCATIVO-PAE PARA PROGRAMAS DE
PREGRADO
CONTENIDOS PROGRAMATICOS PROGRAMAS DE PREGRADO

Código: D-GPA-P01-F02

Versión: 02

Página 5 de 3

- Universidad de Antioquia. Tema 2: Programación Orientada a Objetos en Java. [Enlace](#)
- Java documentation Oracle. <https://docs.oracle.com/javase/tutorial/java/TOC.html>
- Java, API especification Oracle. <https://docs.oracle.com/javase/8/docs/api/overview-summary.html>
- Convenciones de Código para el lenguaje de programación. JAVA Scott Hommel
Google Java code guide. <https://google.github.io/styleguide/javaguide.html#s3.3.2-import-line-wrapping>.
- Robert C. Martin. Código limpio. Manual de estilo para el desarrollo ágil de software. Prentice Hall.
2009
- Martin Fowler y Kendall Scot. UML gota a gota. Publisher, Pearson Educación, 1999.