

Universidad Fidélitas

Facultad de Ingeniería

Escuela de Ingeniería en Sistemas de Computación

**Bachillerato en Ingeniería en Sistemas de Computación**

### **I. INFORMACION GENERAL DEL CURSO**

Código del curso SC-202

Nombre del curso	<b>INTRODUCCIÓN A LA PROGRAMACIÓN</b>
Créditos	<b>4</b>
Horas por semana	2 horas de teoría, 1 hora de práctica y 9 horas de trabajo independiente
Duración	<b>15 SEMANAS</b>
Requisitos	<b>SC-115 PROGRAMACIÓN BÁSICA</b>
Correquisitos	<b>NO TIENE</b>
Modalidad	<b>CUATRIMESTRAL</b>
Nivel	<b>II CUATRIMESTRE</b>
Naturaleza	<b>TEÓRICO-PRÁCTICO</b>
Examen por suficiencia	<b>SI</b>
Asistencia	<b>OBLIGATORIA</b>

### **II. DESCRIPCION GENERAL DEL CURSO**

La programación de sistemas comprende quizá la mayor área de desarrollo de un profesional en Ingeniería en Sistemas de Computación. Este primer curso de programación le presenta al estudiante la forma de cómo se desarrollan los sistemas computacionales y por tanto le involucra en etapas tempranas de su carrera en esta basta área de desarrollo, lo cual le permite identificar su aprehensión por esta área de la disciplina. Durante el curso el estudiante es capaz de observar situaciones del mundo real y plantear una solución tecnológica mediante un lenguaje de programación, por ello el curso de “Introducción a la Programación” provee las bases para la determinación de requerimientos y el desarrollo de sistemas computacionales, utilizando el paradigma de programación orientada a objetos. Durante la ejecución del curso se desarrollan tareas y prácticas de clase que irán incrementando su complejidad y alcance y que serán el insumo para el proyecto final donde crea un sistema elemental utilizando estructuras de datos, estructuras de decisión, de repetición, y datos complejos, como matrices (arreglos) de una o más dimensiones.

### **III. OBJETIVOS GENERALES**

Desarrollar una aplicación de software base considerando los requisitos de cómputo adecuados, las necesidades de los usuarios y los sistemas de información existentes.

#### **IV. OBJETIVOS ESPECIFICOS**

- Comprender las bases necesarias para solucionar cualquier tipo de problema con la ayuda del lenguaje de programación definido, para con ello dar solución a diversos problemas.
- Comprender la nomenclatura de la programación imperativa y orientada a objetos, mediante la aplicación de estándares de trabajo.
- Implementar soluciones de Tecnologías de Información coherentes con los requerimientos de los usuarios y el entorno de estos.
- Determinar los requerimientos de computación necesarios para la solución de un problema de Tecnologías de Información específicos.
- Implementar soluciones de sistemas de información a partir de un diseño

#### **V. CONTENIDOS**

- 1) Introducción a la tecnología Java
  - Reseña Histórica de Java, desglose de sus principales características
- 2) Declarando, inicializando y usando variables
  - Creación de clases con métodos
  - Declaración, inicialización y utilización variables
  - Diferentes tipos de datos primitivos en Java
  - Uso de operadores aritméticos
  - Estándar para nombrar identificadores, revisa las funciones de incremento y decremento así como precedencia de operadores y conversión de tipos.
  - Elementos gráficos prediseñados: ShowInputDialog y ShowMessageDialog
- 3) Creación y uso de objetos y los métodos
  - Creación de instancias de una clase (objeto)
  - Inicialización de variables instanciadas.
  - La clase "String"
  - Ventajas de usar métodos, parámetros y retorno de valores
- 4) Uso de estructuras de decisión
  - If, if/else, if else if
  - Switch / case
- 5) Uso de estructuras de repetición
  - While, do/while, for
- 6) Análisis de un problema y Diseño de su solución
  - Análisis Orientado a Objetos
  - Ámbito de un problema
  - Identificación de objetos
  - Criterios adicionales para el reconocimiento de objetos
  - Identifican atributos y operaciones
- 7) Implementando encapsulamiento y constructores
  - Private, Protected
  - Constructores
  - Static: variable de clase, métodos código estático
- 8) Creación y uso arreglos

- Arreglos de enteros, Strings y objetos
- 9) Creación y uso arreglos multidimensionales
- Matrices de enteros, Strings y objetos

## VI. METODOLOGIA

El curso se desarrolla una metodología teórico-práctica en el laboratorio con sesiones presenciales de tres horas por semana. Se pretende que el profesor actúe como facilitador o guía de la materia y prácticas que desarrolle el estudiante.

El profesor desarrollará los temas mediante exposiciones dinámicas y proactivas y los alumnos aplicarán los conceptos desarrollados por el profesor en pequeñas aplicaciones. El profesor será un guía y facilitador para la búsqueda de soluciones a los problemas planteados.

Para medir el aprovechamiento del curso se pone en práctica el desarrollo de un proyecto que abarca los conceptos expuestos en el temario, así como tres exámenes que abarcarán los temas vistos según cronograma.

## VII. ACTIVIDADES DE APRENDIZAJE (CRONOGRAMA)

Semana	Temas	Actividades de enseñanza
1	Introducción a Java <ul style="list-style-type: none"> <li>• Presentación de JAVA y el entorno de desarrollo NetBeans.</li> </ul>	Introducción a la programación Java, se presentan sus principales características. El docente consensua con los estudiantes los requerimientos de forma y fondo necesarios para la realización del proyecto final. Asignación de lectura sobre la historia del lenguaje de programación Java.
2	Declaración, inicialización y utilización de variables <ul style="list-style-type: none"> <li>• Declaración, inicialización y utilización de variables.</li> <li>• Diferentes tipos de datos primitivos en java</li> <li>• Uso de operadores aritméticos</li> <li>• Estándar para nombrar identificadores, revisa las funciones de incremento y decremento, así como precedencia de operadores y conversión de tipos.</li> <li>• Ventajas de usar métodos, parámetros y retorno de valores</li> <li>• Inicialización de variables instadas (uso de constructores).</li> </ul>	Desarrollo de una demostración y acompañamiento en una práctica general donde se crean clases con métodos que declaran, inicializan y usan variables. Revisan los diferentes tipos de datos primitivos en Java y se discute la conveniencia de tener un estándar para nombrar identificadores, y utilizan las funciones de incremento y decremento, así como precedencia de operadores y conversión de tipos.
3	Desarrollo de Laboratorio Dirigido Creación y uso de objetos y los métodos <ul style="list-style-type: none"> <li>• Creación de instancias de una clase (objeto)</li> </ul>	Desarrollo de una demostración y acompañamiento en una práctica de laboratorio sobre la creación de objetos, inicialización de variables se crean instancias de una clase (objeto),

Semana	Temas	Actividades de enseñanza
	<ul style="list-style-type: none"> <li>La clase "String"</li> <li>Elementos gráficos prediseñados</li> </ul> <p>Uso de estructuras de decisión</p> <ul style="list-style-type: none"> <li>If, if/else, if else if</li> <li>Switch / case</li> </ul>	<p>inicializan variables instanciadas y se manipulan objetos.</p> <p>La práctica incluye el uso de métodos, llamados y retorno de valores, así como modificadores de alcance.</p> <p>Desarrollo de una demostración y acompañamiento en una práctica de laboratorio sobre las estructuras de decisión que involucran operadores relacionales y condicionales.</p>
4	<p>Uso de estructuras de repetición</p> <p>While, do/while, for</p>	<p>Desarrollo de una demostración y acompañamiento en una práctica de laboratorio sobre las estructuras de repetición que involucran operadores relacionales y condicionales.</p>
5	<p>Uso de estructuras de repetición While, do/while, for</p>	<p>Desarrollo de una demostración y acompañamiento en una práctica de laboratorio sobre las estructuras de repetición que involucran operadores relacionales y condicionales.</p>
6	<p>Desarrollo de Laboratorio Dirigido</p> <p>Creación y uso arreglos unidimensionales</p> <ul style="list-style-type: none"> <li>Arreglos de datos primitivos.</li> </ul>	<p>Desarrollo de una demostración y acompañamiento en una práctica de laboratorio sobre arreglos.</p> <p><b>Se recibe la primera entrega del proyecto, esta corresponde a la propuesta inicial sobre el desarrollo que se entregará en semana 15.</b></p>
7	Primer examen parcial	Primer examen parcial
8	<p>Orientación a Objetos - Análisis de un problema y diseño de su solución</p> <ul style="list-style-type: none"> <li>Análisis Orientado a Objetos</li> <li>Ámbito de un problema</li> <li>Identificación de objetos</li> <li>Criterios adicionales para el reconocimiento de objetos</li> <li>Identifican atributos y operaciones</li> <li>Crea clases con métodos</li> <li>Creación de instancias de una clase</li> </ul>	<p>Desarrollo de una demostración y acompañamiento en una práctica sobre cómo realizar un análisis orientado a objetos a partir de un caso de estudio concreto, se analiza el problema mediante análisis orientado a objetos, se identifica el ámbito del problema, se identifican objetos, se estudian criterios adicionales para el reconocimiento de objetos, se identifican objetos, atributos y operaciones.</p> <p>Los alumnos reconocen los 4 componentes de una clase, desarrollo de las prácticas de laboratorio donde escriben una clase y la ejecutan mediante una clase de prueba.</p>
9	<p>Desarrollo de Laboratorio Dirigido</p> <p>Creación y uso de arreglos multidimensionales.</p> <ul style="list-style-type: none"> <li>Arreglos de datos primitivos.</li> </ul>	<p>Desarrollo de una demostración y acompañamiento en una práctica de laboratorio sobre arreglos multidimensionales.</p>
10	Creación y uso de arreglos de objetos	Desarrollo de una demostración y acompañamiento en una práctica de laboratorio sobre arreglos de objetos.
11	Creación y uso de arreglos de objetos	Desarrollo de una demostración y acompañamiento en una práctica de laboratorio sobre arreglos de objetos.
12	<p>Desarrollo de Laboratorio Dirigido</p> <p>Entrega preliminar del proyecto desarrollado</p>	<p>Se recibe la segunda entrega del proyecto, en esta se presentará el inicio en el desarrollo del proyecto y el cumplimiento del mismo con respecto a los compromisos de la entrega anterior.</p>

Semana	Temas	Actividades de enseñanza
13	Examen final	Examen final
14	Presentación de proyecto	Presentación final del proyecto, en cumplimiento con todas las características y funcionalidades acordadas.

## VIII. RECURSOS DIDACTICOS

Se cuenta con aulas equipadas con diversos medios tecnológicos para fomentar el aprendizaje según las estrategias de enseñanza que se puedan utilizar. Los salones de clase están equipados con computadoras y medios de proyección, como videobeam para fomentar presentaciones y medios modernos de comunicación de la información. Particularmente en el curso se tiene acceso a:

- Curso en línea de Microsoft IT Academy complementario y autogestionado.
- Software de desarrollo Netbeans como IDE, utilizando Java SE como lenguaje de programación.

## IX. EVALUACION

La metodología de evaluación debe ser coherente con los objetivos de aprendizaje, las estrategias didácticas y la metodología del curso. Debe recordarse que las rúbricas de evaluación son las evidencias del logro de los conocimientos, habilidades y actitudes de los estudiantes; de ahí lo importante de que éstas se reflejen en la evaluación del curso, aunque no necesariamente cada criterio debe ser evaluado cuantitativamente. La metodología planteada utiliza exámenes tradicionales como forma de evaluación y utiliza las rúbricas y pruebas de internalización del conocimiento. La evaluación de este curso refleja la evaluación diagnóstica y formativa.

La calificación del curso se distribuye en las siguientes actividades evaluativas:

Rubro	Porcentaje
Primer parcial	25%
Examen final	25%
Laboratorios dirigidos	20%
Proyecto final	30%
Total	100%

**Primera Prueba Parcial 25%:** Esta prueba es sumativa y se evaluarán los siguientes temas: Introducción a Java, Declaración, inicialización y utilización de variables, Creación y uso de objetos, estructuras condicionales, estructuras cíclicas y arreglos unidimensionales.

**Segunda Prueba Parcial 25%:** Esta prueba es sumativa y se evaluará los siguientes temas: Orientación a Objetos, Arreglos Multidimensionales, Arreglos de Objetos, Manejo de Excepciones y Manejo de Archivos Planos.

**Laboratorios Dirigidos 20%:** Son actividades prácticas en la que los estudiantes aplican lo aprendido en clase y resuelven problemas dados por el profesor que están enfocados en soluciones de la vida real.

**Proyecto Final: 30%** Esta actividad propiciará la puesta en práctica de los conocimientos y habilidades adquiridos en el curso. Consiste en la creación de un proyecto programado que tiene como objetivo satisfacer

las necesidades formuladas por los estudiantes con apoyo del docente, dicha solución debe desarrollarse en lenguaje Java y debe contemplar el ingreso de información, instrucciones de toma de decisiones y repetición así como utilizar estructuras de almacenamiento temporal en memoria como los arreglos, este trabajo es grupal, compuesto por 2 o 3 estudiantes, se deben presentar 2 avances parciales (semanas 6, 12) y la defensa pública final en semana 15. El proyecto debe contener al menos 8 clases diferentes, estar encapsuladas y tener sentido el uso de estas clases.

## X. RUBRICAS

Tareas se evalúan con la siguiente rúbrica:

Aspectos a evaluar	Excelente 3 puntos	Bueno 2 puntos	Regular 1 punto	Deficiente 0 puntos	Puntuación
Entendimiento del problema	Lo entiende a cabalidad	Hay algunos aspectos no entendidos	Entendió poco lo que había que hacer	No entendió del todo el problema	
Procedimientos de solución	Los procedimientos están enlistados con pasos claros.	Los procedimientos están enlistados en un orden lógico, pero los pasos no están enumerados	Los procedimientos están enlistados, pero no están en un orden lógico o son difíciles de seguir.	Los procedimientos no enlistan en forma precisa todos los pasos.	
Solución dada	Es correcta totalmente	Hay pequeños errores y pocos	Hay bastantes errores, pero no son críticos	Hay muchos errores y algunos de ellos son críticos	
Ortografía y Gramática	Uno o pocos errores de ortografía	Dos o tres errores de ortografía no relevantes	Cuatro errores de Ortografías algo relevantes	Más de 4 errores de Ortografía y son relevantes	
Puntaje total 12 pts					

## Rúbrica para la evaluación del proyecto final

Entrega de propuesta formal del proyecto (12 pts)	No (0 pts)	Incompleto (1 pt)	Completo (3 pts)
Se encuentran los requerimientos funcionales del proyecto presentado bien identificados.			
Se presenta una documentación que incorpora elementos visuales que presentan una idea general de la interfaz al usuario.			
Se tienen identificadas las entidades y clases necesarias para dar solución al problema planteado.			
Se presenta cronograma de trabajo con roles y responsabilidades del equipo de trabajo.			
Sintaxis y uso del lenguaje de programación JAVA (32 pts)	No (0 pts)	Incompleto (2 pts)	Completo (4 pts)
El programa compila sin errores ni avisos			
Se declaran e inicializan correctamente referencias y objetos			
Las expresiones están bien construidas, se colocan correctamente los paréntesis necesarios.			
Se utilizan correctamente las sentencias condicionales.			
Se utiliza la sentencia repetitiva (FOR, WHILE, DO/WHILE) más adecuada a la funcionalidad presentada.			
Los bucles están bien implementados: se inicializan variables, se realiza correctamente cada paso de iteración.			
Se declaran atributos y métodos con algún modificador a la declaración (private, static, final, etc.)			
Los parámetros de los métodos están correctamente declarados y se emplea correctamente el resultado del método.			
Orientación a Objetos en JAVA (15 pts)	No (0 pts)	Incompleto (1 pt)	Completo (3 pts)
Se ha realizado un análisis de entidades y funcionalidades y se han identificado los módulos del programa.			
Se utilizan clases para representar cada una de las entidades del programa (al menos 8 clases).			
Se evita la repetición de código o la falta de estructura. Por ejemplo, se añaden las clases necesarias, aunque sean pequeñas.			

Se utilizan clases y objetos de librerías. Se evita repetir implementaciones de funcionalidades sin justificación.			
El programa tiene un diseño sencillo y eficiente, pero con el suficiente detalle.			
<b>Desarrollo del programa en JAVA (15 pts)</b>	<b>No (0 pts)</b>	<b>Incompleto (1 pt)</b>	<b>Completo (3 pts)</b>
El programa tiene comentarios o está documentado			
El formato del código fuente es correcto conforme al estándar visto (CAMEL, PASCAL o VERSAL)			
Las estructuras de datos son las adecuadas (vectores, matrices) (al menos 4 diferentes)			
Los algoritmos del programa se han realizado correctamente			
El programa funciona correctamente dentro del alcance y conforme a las funcionalidades esperadas.			
<b>Competencias de Conocimiento (16 pts)</b>	<b>No (0 pts)</b>	<b>Incompleto (2 pts)</b>	<b>Completo (4 pts)</b>
El programa es original en cuanto a su funcionalidad			
El diseño gráfico tiene todos los componentes, es claro y estructurado, además es amigable (no se escribe en consola).			
La funcionalidad del programa es podría ser valiosa.			
El programa tiene el alcance requerido (tamaño del programa, número de funcionalidades, tecnologías empleadas, etc.)			
<b>Defensa del proyecto (10 pts)</b>	<b>No (0 pts)</b>	<b>Incompleto (1 pt)</b>	<b>Completo (2 pts)</b>
La defensa se desarrolla en el tiempo estipulado por el docente y de forma estructurada.			
La exposición se desarrolla con una alta capacidad comunicativa por parte del estudiante.			
El vocabulario común y técnico usado por el estudiante complejo, variado y apropiado.			
La respuesta a las consultas planteadas a la defensa es satisfactorias y convincente.			
El manejo del proyecto demostrado por el estudiante evidencia la profundización de la gestión de la operación efectuada.			

## XI. BIBLIOGRAFIA

- Altadil, P.; Pérez, E. (2017). Java Limpio: Programación Java y buenas prácticas de desarrollo
- Deitel, P.; Deitel, H. (2013). Cómo programar en Java. 10ma Edición. Pearson. USA
- Deitel, P.; Deitel, H. (2017). Java How to Program (English) 11 Edition. Pearson. USA

## XII. OBSERVACIONES GENERALES

El estudiante debe cumplir con todas las disposiciones del Reglamento de Régimen Estudiantil de la Universidad Fidélitas.

## XIII. Directriz sobre Honestidad Académica

Para efectos de este curso, los participantes deben evitar conductas deshonestas tales como el fraude académico o plagio:

- Hacer fraude académico incluye, dentro de otras acciones, falsificar bibliografía, utilizar datos inventados, presentar como propios proyectos elaborados por otras personas, obtener ayuda no autorizada en tareas calificadas o que otra persona desarrolle el trabajo que le corresponde a usted.
- Plagiar incluye copiar textualmente frases, oraciones, párrafos y trozos enteros de material impreso, Internet y otras fuentes, sin realizar la correspondiente cita; incluso parafrasear sin citar las fuentes.

Los casos de fraude académico o plagio implicarán la pérdida automática del curso, y de repetirse la falta, se sancionarán con la expulsión definitiva.