

Universidad Fidélitas
Facultad de Ingeniería
Escuela de Ingeniería en Sistemas de Computación
Bachillerato en Ingeniería en Sistemas de Computación

I. INFORMACIÓN GENERAL DEL CURSO

Código del curso SC-202

| | |
|------------------------|--|
| Nombre del curso | INTRODUCCIÓN A LA PROGRAMACIÓN |
| Créditos | 4 |
| Horas por semana | 2 horas de teoría, 1 hora de práctica y 9 horas de trabajo independiente |
| Duración | 14 SEMANAS |
| Requisitos | SC-115 PROGRAMACIÓN BÁSICA |
| Correquisitos | NO TIENE |
| Modalidad | CUATRIMESTRAL |
| Nivel | II CUATRIMESTRE |
| Naturaleza | TEÓRICO-PRÁCTICO |
| Examen por suficiencia | SI |
| Asistencia | OBLIGATORIA |

II. DESCRIPCIÓN 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 se 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 ESPECÍFICOS

- 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
 - Declaración, inicialización y utilización de variables. Estándar para nombrar identificadores.
 - Diferentes tipos de datos en Java
 - Uso de operadores aritméticos.
 - Elementos gráficos para la presentación y captura de datos.
 - La clase "String".
- 2) Uso de estructuras de decisión
 - Utilidad de las estructuras de decisiones / condicionales.
 - Tablas de verdad.
 - El condicional IF y sus variantes.
 - El condicional SWITCH / CASE.
- 3) Uso de estructuras de repetición.
 - Estructura de repetición WHILE.
 - Estructura de repetición DO / WHILE.
 - Estructura de repetición FOR.
- 4) Primeros pasos en la programación modular.
 - Utilización de métodos.
 - Paso de parámetros a los métodos.
 - Retorno de valores desde los métodos.
- 5) Programación Orientada a Objetos
 - Análisis y perspectiva del paradigma de Programación Orientada a Objetos.
 - Clases, objetos y sus principales componentes.
- 6) Arreglos de datos.
 - Arreglos unidimensionales.
 - Arreglos de múltiples dimensiones.
 - Arreglos de objetos.

VI. METODOLOGÍA

En 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 que aporte prácticas que desarrollen los estudiantes.

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, dos pruebas cortas colegiadas y dos 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"> Presentación de JAVA y el entorno de desarrollo NetBeans. Variables y operadores <ul style="list-style-type: none"> Declaración, inicialización y utilización de variables. Diferentes tipos de datos primitivos en java Uso de operadores aritméticos Operadores de incremento y decremento. Precedencia de operadores y conversión de tipos. Estándar para nombrar identificadores. Elementos gráficos prediseñados. | <ul style="list-style-type: none"> Presentación del profesor y los estudiantes Lectura y discusión del programa del curso Introducción a la programación Java, se presentan sus principales características. Presentación de la dinámica del curso. Asignación de lecturas de la semana. Desarrollo de una demostración y acompañamiento en una práctica general donde se crea un programa básico con los elementos aprendidos. Definición del proyecto del curso |
| 2 | Uso de estructuras de decisión <ul style="list-style-type: none"> If, if/else, if else if Switch / case La clase "String" | <ul style="list-style-type: none"> Repaso de los temas vistos en la clase anterior, aclaración de dudas y revisión de ejercicios. Utilización de estructuras condicionales en el lenguaje. Avance proyecto #1. Conformación de grupos y selección de temática. |
| 3 | Uso de estructuras de repetición while, do/while, for | <ul style="list-style-type: none"> Repaso de los temas vistos en la clase, aclaración de dudas y revisión de ejercicios. Introducción a las estructuras cíclicas. Desarrollo de ejemplos y acompañamiento en ejercicios prácticos. |
| 4 | Programación Orientada a Objetos <ul style="list-style-type: none"> ¿Qué es? Ventajas de la Programación Orientada a Objetos. | <ul style="list-style-type: none"> El docente abarcará la temática desde la necesidad de solventar un requerimiento de la vida real, pero enfocando el mismo en los componentes de la Programación Orientada a Objetos. |

| Semana | Temas | Actividades de enseñanza |
|-----------|--|--|
| | <ul style="list-style-type: none"> Definición de un requerimiento enfocado desde la perspectiva de Orientación a Objetos. Representación de una solución orientada a objetos (diagramas) | <ul style="list-style-type: none"> Avance proyecto #2. Conforme a la definición del proyecto, se crea la diagramación necesaria para representar la solución a desarrollar. |
| 5 | Programación Orientada a Objetos <ul style="list-style-type: none"> Clases e Instancias de Clases. Métodos y Atributos. Parámetros y retorno de objetos. Modificadores. | <ul style="list-style-type: none"> El docente utiliza las estructuras de programación con el propósito de resolver un requerimiento a partir de un análisis y de los elementos encontrados. Avance proyecto #3. Se implementan los elementos aprendidos en la estructura del proyecto en desarrollo. |
| 6 | Programación Orientada a Objetos <ul style="list-style-type: none"> Constructores. Encapsuladores. Clases como atributos. Clases estáticas y enumeraciones. | <ul style="list-style-type: none"> El docente utiliza las estructuras de programación con el propósito de resolver un requerimiento a partir de un análisis y de los elementos encontrados. Avance proyecto #4. Se implementan los elementos aprendidos en la estructura del proyecto en desarrollo. Quiz colegiado #1. |
| 7 | Integración de conocimientos | <ul style="list-style-type: none"> Caso de simulación programado |
| 8 | Arreglos <ul style="list-style-type: none"> Arreglos unidimensionales. Arreglos multidimensionales. | <ul style="list-style-type: none"> Se demuestra la utilización de arreglos a los arreglos unidimensionales, su implementación y utilización. Desarrollo de ejercicios y acompañamiento en prácticas sobre arreglos. |
| 9 | Arreglos de objetos. <ul style="list-style-type: none"> Utilización de estructuras matriciales para almacenar objetos. Implementación de arreglos de objetos dentro de soluciones de Programación Orientada a Objetos. | <ul style="list-style-type: none"> El docente expone los elementos de la semana y ejemplifica mediante ejercicios de programación la implementación y el valor de estas estructuras. Avance proyecto #5. Se implementan los elementos aprendidos en la estructura del proyecto en desarrollo. |
| 10 | Ejercicio integrador / Cierre de componente técnicos del proyecto. | <ul style="list-style-type: none"> El docente desarrolla un ejercicio partiendo desde un caso de la vida real hasta la ejecución del programa que modela la solución a ese caso. Incluye todos los aspectos vistos en el curso. Se crean los espacios para que los estudiantes presenten dudas puntuales y finales sobre los temas desarrollados en el proyecto final. |
| 11 | Práctica para examen final del curso | <ul style="list-style-type: none"> Realización de prácticas sobre los temas de las semanas anteriores como preparación para examen. Quiz colegiado #2. |
| 12 | Integración de conocimientos | <ul style="list-style-type: none"> Caso de Simulación Programado |

| Semana | Temas | Actividades de enseñanza |
|-----------|---------------------------------------|---|
| 13 | Defensa final del proyecto del curso | <ul style="list-style-type: none"> Presentación final del proyecto, en cumplimiento con todas las características y funcionalidades acordadas. |
| 14 | Defensa final del proyecto del curso. | <ul style="list-style-type: none"> Presentación final del proyecto, en cumplimiento con todas las características y funcionalidades acordadas. |

VIII. RECURSOS DIDÁCTICOS

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:

- Software de desarrollo Netbeans como IDE, utilizando Java SE como lenguaje de programación.

IX. EVALUACIÓN

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 |
|-------------------------------|-------------|
| Caso de Simulación Programado | 15% |
| Caso de Simulación Programado | 15% |
| 2 quices colegiados (5% c/u). | 10% |
| Proyecto final | 60% |
| Total | 100% |

Caso de Simulación Programado 15%: Esta prueba es sumativa y se evaluarán los siguientes temas: Introducción a Java, declaración, inicialización y utilización de variables, estructuras condicionales, estructuras cíclicas y los primeros temas en el desarrollo de la Programación Orientada a Objetos.

Caso de Simulación Programado 15%: Esta prueba es sumativa y se evaluará los siguientes temas: Orientación a Objetos, Arreglos Unidimensional y Multidimensionales, Arreglos de Objetos. Además de los temas anteriores que sean necesarios para el desarrollo de las soluciones requeridas.

Quices colegiados 10%: Son actividades prácticas desarrolladas de manera colegiada por medio de la plataforma Campus Virtual.

Proyecto Final: 60% Esta actividad propiciará la puesta en práctica de los conocimientos y habilidades adquiridos en el curso de manera incremental. 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 repeticiones, así como utilizar estructuras de almacenamiento temporal en memoria como los arreglos, este trabajo es grupal, su presentación y evaluación es incremental como se indica en la rúbrica del mismo.

X. RÚBRICAS

Rúbrica para la evaluación del proyecto final

| Semana | Detalle | Porcentaje |
|---------|--|------------|
| 2 | Conformación de grupos y selección de temática. | 5% |
| 4 | Conforme a la definición del proyecto, se crea la diagramación necesaria para representar la solución a desarrollar. | 10% |
| 5 | Se implementan los elementos aprendidos en la estructura del proyecto en desarrollo. | 5% |
| 6 | Se implementan los elementos aprendidos en la estructura del proyecto en desarrollo. | 5% |
| 9 | Se implementan los elementos aprendidos en la estructura del proyecto en desarrollo. | 5% |
| 14 / 15 | Presentación final y defensa del proyecto del curso | 30% |

XI. BIBLIOGRAFÍA

- 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 y con los reglamentos específicos del curso.

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.