

SC-202 Introducción a la programación
Universidad Fidélitas
Facultad de Ingeniería
Escuela de Ingeniería en Sistemas de Computación
Bachillerato en Ingeniería en Sistemas de Computación

Código del curso	SC-202
Nombre del curso	INTRODUCCIÓN A LA PROGRAMACIÓN
Sede	CENTRAL Y HEREDIA
Créditos	4
Horas por semana	12 horas distribuidas en: 2 horas de Teoría, 1 hora trabajo práctico y 9 horas trabajo independiente
Duración	14 SEMANAS
Requisitos	SC-115 PROGRAMACION BASICA
Correquisitos	NO TIENE
Modalidad	CUATRIMESTRAL
Nivel	II CUATRIMESTRE
Naturaleza	TEÓRICO-PRÁCTICO
Asistencia	Obligatoria
Profesor	POR DEFINIR

DESCRIPCION GENERAL DEL CURSO

La programación se encarga de sistematizar instrucciones dentro de un dispositivo computacional para que realice tareas específicas, es decir la programación automatiza tareas dentro de diferentes sistemas de cómputo. El propósito de este curso es desarrollar las habilidades para crear aplicaciones de software¹ base con el paradigma Orientada a Objetos, considerando los requisitos de cómputo adecuados, las necesidades de los usuarios y los sistemas de información existentes, para la mejora de las empresas.

La importancia del curso radica en que el estudiante podrá aplicar soluciones de tecnologías de información coherentes con los requerimientos de los usuarios y el entorno de estos, para la solución de un problema empresarial y analizar la nomenclatura de la programación imperativa y orientada a objetos, para la mejora de las empresas.

En concordancia con el perfil profesional de la carrera, las principales habilidades que desarrolla el curso son: aplicar el lenguaje de programación para la solución de diferentes problemas, analizar la nomenclatura imperativa y orientada a objetos, aplicar estándares de programación para la mejora de las empresas y aplicar soluciones de tecnologías de información coherentes con los requerimientos de los usuarios y el entorno de estos analizando las necesidades de computación.

¹ Programa informático

Asimismo, se promueve el desarrollo de habilidades blandas como el liderazgo y el trabajo colaborativo, así como el pensamiento crítico, mediante el estudio de las buenas prácticas del manejo de lenguajes de programación.

El curso es de naturaleza tanto teórica como práctica.

El curso responde a la siguiente pregunta generativa: **¿Cómo desarrollar una aplicación de software base que considera requisitos de cómputo adecuados, necesidades de los usuarios y los sistemas de información existentes, para la mejora de las empresas?**

Durante el curso se pretende lograr que los estudiantes contesten la pregunta anterior mediante la aplicación del ciclo de vida de un programa, uso de variables, estructuras de decisión y de repetición, análisis y diseño Orientada a objetos, así como el uso de arreglos de una o dos dimensiones.

Competencias, criterios de desempeño y evidencias

Para la Universidad Fidélitas las competencias son integraciones de valores éticos y conductuales (saber ser) con saberes cognitivos (saber saber) y con habilidades y aptitudes (saber hacer), enfocadas a la solución de problemas profesionales y sociales, en un contexto determinado, en una perspectiva de mejora continua y en un marco conceptual.

A continuación, se presentan tanto las competencias disciplinares como las genéricas, relacionadas a los criterios de desempeño y evidencias para este curso.

Competencias específicas de la carrera	Criterios de desempeño	Resultados de Aprendizaje
Desarrolla aplicaciones de software base Orientada a Objetos, para la mejora de las empresas, considerando los requisitos de cómputo adecuados, las necesidades de los usuarios y los sistemas de información existentes.	Discute las bases necesarias del lenguaje de programación definido, considerando el estudio de este en la solución a diversos problemas.	Discutir las bases necesarias del lenguaje de programación definido. Evidencia: Informes de ejercicios prácticos.
	Aplica soluciones de tecnologías de información coherentes con los requerimientos de los usuarios y el entorno de estos, considerando el análisis de las necesidades de computación.	Aplica soluciones de tecnologías de información coherentes con los requerimientos de los usuarios y el entorno de estos. Evidencia: Informe de Proyecto Final
	Analiza la nomenclatura de la programación imperativa y orientada a objetos, considerando la aplicación de estándares de trabajo.	Analizar la nomenclatura de la programación imperativa y orientada a objetos. Evidencia: Informe de investigación
Competencias generales	Criterios de desempeño	Resultados de Aprendizaje
CG-1 Integra los conocimientos, habilidades y actitudes necesarias para el aprendizaje de las técnicas de trabajo en equipo y liderazgo considerando la colaboración, la cooperación y el acuerdo.	Participa y colabora activamente en las tareas del equipo y fomenta la confianza, la cordialidad y la orientación a la tarea conjunta. Contribuye con la consolidación del equipo, favoreciendo la comunicación, el reparto equilibrado de tareas, el clima interno y la cohesión	Aplicar las técnicas de trabajo en equipo y liderazgo considerando la colaboración, la cooperación y el acuerdo.

	Dirige grupos de trabajo, asegurando la integración y colaboración de los miembros mostrando capacidad de resolución de problemas comunes. Desarrolla capacidad para diseñar y ofrecer soluciones en diálogo y colaboración, estableciendo y cumpliendo compromisos	
CG-2 Integra los conocimientos, habilidades y actitudes necesarias para la utilización de las tecnologías digitales y tratamiento de la información como elemento esencial para informarse, aprender y comunicarse.	Desarrolla habilidades en el uso de las tecnologías de la información y de la comunicación utilizando herramientas tecnológicas. Organiza la información, la relaciona, analiza, sintetiza, haciendo inferencias y deducciones de distinto nivel de complejidad utilizando las herramientas tecnológicas.	Aplicar las tecnologías de la información y de la comunicación considerando la transformación digital de las organizaciones educativas.

CONTENIDOS

1. Tema: Introducción a la tecnología Java

- 1.1. Reseña Histórica de Java, desglose de sus principales características
- 1.2. Declaración, inicialización y utilización de variables. Estándar para nombrar identificadores.
- 1.3. Diferentes tipos de datos en Java
- 1.4. Uso de operadores aritméticos.
- 1.5. Elementos gráficos para la presentación y captura de datos.
- 1.6. La clase "String".

2. Tema: Uso de estructuras de decisión

- 2.1. Utilidad de las estructuras de decisiones / condicionales.
- 2.2. Tablas de verdad.
- 2.3. El condicional IF y sus variantes.
- 2.4. El condicional SWITCH / CASE.

3. Tema: Uso de estructuras de repetición

- 3.1. Estructura de repetición WHILE.
- 3.2. Estructura de repetición DO / WHILE.
- 3.3. Estructura de repetición FOR.

4. Tema: Primeros pasos en la programación modular.

- 4.1. Utilización de métodos.
- 4.2. Paso de parámetros a los métodos.
- 4.3. Retorno de valores desde los métodos.

5. Tema: Programación Orientada a Objetos

- 5.1. Análisis y perspectiva del paradigma de Programación Orientada a Objetos.
- 5.2. Clases, objetos y sus principales componentes.

6. Tema: Arreglos de datos

- 6.1. Arreglos unidimensionales.
- 6.2. Arreglos de múltiples dimensiones.
- 6.3. Arreglos de objetos.

METODOLOGIA

Este es un curso de interacción entre los estudiantes y el profesor de tipo participativo, donde las actividades planificadas incentivan el aprendizaje basado en la resolución de problemas (Aprendizaje Basado en Problemas), proyectos, investigaciones y ejercicios prácticos.

La metodología del aprendizaje basado en problemas (ABP) está centrada en el estudiante, para que éste adquiera las competencias a través de situaciones de contexto, que le permitan identificar e interpretar el papel de las ciencias de datos en una perspectiva de cambio ético y social y de mejora continua de la calidad de vida de las personas.

Su finalidad es la formación de estudiantes capaces de desarrollar aplicaciones de software base Orientada a Objetos, para la mejora de las empresas, considerando los requisitos de cómputo adecuados, las necesidades de los usuarios y los sistemas de información existentes. El ABP pretende que los estudiantes analicen los principios de la ética en la disciplina de ingeniería en Sistemas de Computación, para la resolución de problemas sociales y organizacionales, y asuman una mayor responsabilidad de su propio aprendizaje, su intención es encaminarlos a situaciones que los lleven a rescatar, comprender y aplicar lo que aprenden como una herramienta para resolver problemas y realizar tareas.

Es un aprendizaje orientado a la acción, no se trata solo de aprender acerca de algo sino de hacer algo, por lo que las personas estudiantes podrán discutir las bases necesarias del lenguaje de programación definido, considerando el estudio de este en la solución a diversos problemas, aplicar soluciones de tecnologías de información coherentes con los requerimientos de los usuarios y el entorno de estos, considerando el análisis de las necesidades de computación, y analizar la nomenclatura de la programación imperativa y orientada a objetos, considerando la aplicación de estándares de trabajo.

El trabajo en equipo y colaborativo se estimula mediante, trabajos de proyectos y técnicas expositivas, donde se pone en práctica la capacidad de análisis, reflexión y síntesis. El papel del docente será en todo momento de mediador y facilitador del proceso educativo, permitiendo la construcción y autorregulación de los aprendizajes por parte de los estudiantes.

El método de enseñanza-aprendizaje utilizado en el desarrollo de este curso se concibe como un proceso de análisis y síntesis, lo cual a través de cada actividad servirá para la internalización de la importancia del pensamiento crítico y de una visión sistémica de la ingeniería en Sistemas de Computación.

El Aprendizaje Basado en problemas aplicado en las actividades de aprendizaje: Proyecto, investigación y ejercicios prácticos, abre un abanico de posibilidades a la innovación didáctica, desarrolla capacidades tales como el aprendizaje significativo, autónomo, además de la visión integral de los problemas en forma interdisciplinaria.

El curso se desarrolla con una metodología constructivista, donde los estudiantes analizan los recursos disponibles, y se dedican a realizar actividades prácticas, lo que los lleva a una comprensión más profunda de los contenidos, mediante una metodología de aprendizaje basada en problemas, la cual supone una manera concreta de aprender críticamente tomando elementos y problemas del contexto.

Las técnicas propias del curso se describen a continuación:

Ejercicios prácticos: El método de Aprendizaje Basado en Problemas (ABP) aplicado a la resolución de ejercicios prácticos promueve en los estudiantes nuevos conocimientos, la práctica reflexiva y la adaptación a los cambios. El propio estudiante identifica sus necesidades y pone en marcha los medios y las estrategias a su alcance para dar respuesta al problema. El docente ejerce como impulsor de ese primer reto y como apoyo en el camino hacia su solución, a modo de guía, supervisor y facilitador. Se trata, además, de un proceso continuo, en el que los nuevos conocimientos no solo consiguen resolver el reto propuesto, sino que además plantean nuevos problemas y nuevas necesidades que siguen impulsando el aprendizaje, por lo que la actividad es individual y las personas estudiantes podrán discutir las bases necesarias del lenguaje de programación definido, considerando el estudio de este en la solución a

diversos problemas. La práctica reflexiva permite razonar sobre la problemática planteada y posibilita la construcción del conocimiento mediante procesos de diálogo y discusión que ayudan a los estudiantes a desarrollar habilidades transversales de comunicación y expresión oral, al mismo tiempo que desarrollan el pensamiento crítico y la argumentación lógica, para la exploración de sus valores y de sus propios puntos de vista. Estas competencias les permiten afrontar una práctica profesional más reflexiva y crítica. La actividad se vincula a la competencia y evaluación del curso. Las indicaciones para realizar la actividad son las siguientes:

1. La solución del problema debe estar bien desarrollada. Se deben presentar nuevas ideas exploradas. La estrategia descrita debe apoyar los resultados.
2. El problema resuelto evidencia mayor nivel de pensamiento.
3. La explicación dada para el concepto se desarrolla fácilmente y es efectivo.
4. Los gráficos y textos desarrollados exhiben un grado de creatividad y ayudan al entendimiento de los procedimientos.
5. Todas las estrategias desarrolladas deben estar relacionadas con el tema. Se deben indicar la fuente de donde fueron tomados
6. Todos los textos y gráficos deben estar relacionados con el tema y hacen que sea más fácil de entender.
7. Desarrolla aportaciones de calidad y demuestra ser el resultado de un proceso de pensamiento propio y crítico
8. En la presentación existe dominio de los conceptos y estos son comunicados de manera efectiva

Proyecto: El proyecto final consiste en situar a un educando en un contexto que imite algún aspecto de la realidad y en establecer en ese ambiente situaciones, problemáticas o reproductivas, similares a las que él deberá enfrentar en su desempeño profesional contribuyendo a elevar la calidad del proceso de aprendizaje, por lo que la actividad se vincula a las competencias del curso. Con las temáticas correspondientes del curso de Introducción a la Programación de manera colaborativa entre los estudiantes y con el acompañamiento del docente. Se desarrollarán los temas de Condicionales, Ciclos, Orientación a Objetos, Métodos, Constructores, Encapsuladores, Arreglos y la aplicación práctica de todos estos elementos para el curso de Introducción a la Programación. El proyecto es un trabajo colaborativo en grupos de estudiantes no mayores a los 4 integrantes, en donde se desarrollará la solución, con la participación periódica del docente que asesorará el avance de este. Se desarrollan las habilidades de trabajo colaborativo, cuyos marcos teóricos se verán aplicados en contextos reales y estos se vinculan a la evaluación, por lo que el propio estudiante identifica sus necesidades y pone en marcha los medios y las estrategias a su alcance para dar respuesta al problema. El docente ejerce como impulsor de ese primer reto y como apoyo en el camino hacia su solución, a modo de guía, supervisor y facilitador. Por lo que la persona estudiante podrá aplicar soluciones de tecnologías de información coherentes con los requerimientos de los usuarios y el entorno de estos, considerando el análisis de las necesidades de computación.

A continuación se presenta un ejemplo para el planteamiento del proyecto:

Las instrucciones para desarrollar el proyecto son las siguientes:

Descripción general del requerimiento – Parqueo “El descanso”:

La cadena de Parqueos “El descanso”, como forma de adaptarse a la nueva realidad, quiere contratar un sistema de información que le permita ser flexible con las medidas sanitarias y que le ayude a cumplir con porcentajes de aforo variables tanto en ingreso de vehículos, pago y demás.

Le han solicitado a su equipo de trabajo que les desarrollen la solución de acuerdo con una serie de requerimientos que se expondrán a continuación, pero ellos están de acuerdo con aceptar propuestas adicionales de los estudiantes, siempre y cuando estas no afecten los requerimientos originales.

Condiciones técnicas del requerimiento

Registro de la cantidad de espacios de parqueo: Solicitar la ubicación y cantidad de posibles vehículos a recibir en el parqueo

Módulo de registro de vehículos: Solicitar número de placa, forma de pago (efectivo, transferencia, tarjeta de crédito). Solicitar cuánto tiempo estimado de parqueo

Módulo de reservaciones de espacios: Solicitar los días del espacio, verificar la disponibilidad de acuerdo con las fechas y las reservaciones, verificar las tarifas de entre semana y fin de semana por persona por noche, calcular el monto a pagar, el IVA y el monto total.

Módulo de reservaciones de mensuales: Solicitar días, tiempos y horas de ocupación para reservar de acuerdo con la capacidad del parqueo.

Reportes de asignaciones: Emite un análisis de la ocupación mensual que lleva el sistema
Salir

Estudios de caso: El método de estudio de casos promueve en los estudiantes tres aspectos básicos: la gestión del conocimiento, la práctica reflexiva y la adaptación a los cambios. Con la gestión del conocimiento se busca que el estudiante adquiera las estrategias y las técnicas que le permitan aprender por sí mismo; esto implica la toma de conciencia de la asimilación, la reflexión y la interiorización del conocimiento para que finalmente pueda valorar y profundizar a partir de una opción personal. El docente fomenta la capacidad de resolver problemas mediante pensamiento crítico, potencia el diálogo entre profesionales y estudiantes, se comparten formas de comprensión y se permite reconstruir significados. La actividad es individual, y se vincula a la evaluación del curso por lo que la persona estudiante podrá aplicar los elementos de programación básica utilizados en el campo de la computación, para la representación de soluciones a los problemas contemporáneos. Las personas estudiantes podrán hacer uso de las bases de datos digitales de la universidad y la biblioteca para localizar información y realizar la investigación para los estudios de caso, así como uso de la bibliografía del curso. El paso a paso para realizar los estudios de caso son los siguientes:

1. Se debe de presentar el estudio de caso con una portada con los elementos solicitados por el profesor.
2. Se deben de analizar tres puntos claves del caso y su entorno
3. El problema central debe ser analizado y se debe plantear en forma de pregunta. Se deben de analizar las oportunidades detectadas.
4. Se deben de presentar dos estrategias analizadas, dando una justificación para resolver el problema u oportunidad.
5. Se debe analizar la alternativa viable y se debe explicar por qué, en al menos tres párrafos.
6. El marco teórico debe sustentar de muy buena manera, las alternativas de soluciones al caso
7. Se deben de presentar al menos tres conclusiones referidas al tema.
8. Se deben presentar las referencias utilizadas en formato de referencia estándar (IEE de APA) vigente.
9. Se debe presentar al menos cuatro fuentes bibliográficas con no menos de 5 años
10. Se debe presentar un comentario personal sobre sus puntos de vista con respecto al estudio de caso.
11. Se deben de cumplir con los lineamientos de redacción y ortografía vigentes.

Estrategias de aprendizaje

A través de diferentes técnicas y actividades, se busca la adquisición de conocimientos, destrezas, actitudes y habilidades de reflexión, relacionándolos con la práctica en el campo profesional y dentro de un contexto educativo, de forma que sea tan real como sea posible.

Para tal fin, el docente se encargará de establecer y esclarecer conceptos clave que permitan integrar el conocimiento previo con la información más actualizada, propiciando la evacuación de dudas, las discusiones, los aportes, experiencias e ideas de los estudiantes.

La integración de los saberes del curso se evidencia mediante el desarrollo de aplicaciones de software base Orientada a Objetos, para la mejora de las empresas, considerando los requisitos de cómputo adecuados, las necesidades de los usuarios y los sistemas de información existentes, estos rasgos se sustentan en el enfoque curricular constructivista de la carrera y en la concepción epistemológica por competencias y tópicos generativos, relacionados a Introducción a la tecnología Java y al ciclo de vida de un programa, Declarando, inicializando y usando variables; Uso de estructuras de decisión; Creación y uso de objetos y los métodos; Análisis de un problema y diseño de su solución; Uso de estructuras de repetición; Creación y uso arreglos; y Creación y uso arreglos multidimensionales

Todo lo anterior se instrumenta a través de estrategias de aprendizaje activo, como son los ejercicios prácticos, investigación y el proyecto. Estas estrategias permiten la delimitación indicada en el desempeño competencial específico planteado en el curso, así como en los tópicos generativos específicos.

La integración de los rasgos de este curso está pensada en función de las competencias específicas, las estrategias de aprendizaje, los tópicos generativos y el enfoque curricular, tal como se puede evidenciar en la coherencia relacionada al componente curricular de competencias, criterios de desempeño, evidencias, metodología, estrategias de aprendizaje y evaluación. Cómo se puede evidenciar todos los elementos están unidos por un hilo conductor que da coherencia al curso.

Las técnicas propias del curso son:

Ejercicios prácticos: El método de Aprendizaje Basado en Problemas (ABP) aplicado a la resolución de ejercicios prácticos promueve en los estudiantes nuevos conocimientos, la práctica reflexiva y la adaptación a los cambios. El propio estudiante identifica sus necesidades y pone en marcha los medios y las estrategias a su alcance para dar respuesta al problema. El docente ejerce como impulsor de ese primer reto y como apoyo en el camino hacia su solución, a modo de guía, supervisor y facilitador. Se trata, además, de un proceso continuo, en el que los nuevos conocimientos no solo consiguen resolver el reto propuesto, sino que además plantean nuevos problemas y nuevas necesidades que siguen impulsando el aprendizaje, por lo que la actividad es individual y las personas estudiantes podrán discutir las bases necesarias del lenguaje de programación definido, considerando el estudio de este en la solución a diversos problemas. La práctica reflexiva permite razonar sobre la problemática planteada y posibilita la construcción del conocimiento mediante procesos de diálogo y discusión que ayudan a los estudiantes a desarrollar habilidades transversales de comunicación y expresión oral, al mismo tiempo que desarrollan el pensamiento crítico y la argumentación lógica, para la exploración de sus valores y de sus propios puntos de vista. Estas competencias les permiten afrontar una práctica profesional más reflexiva y crítica. La actividad se vincula a la competencia y evaluación del curso.

Estrategia de aprendizaje basada en estudios de caso: El método de estudio de casos promueve en los estudiantes tres aspectos básicos: la gestión del conocimiento, la práctica reflexiva y la adaptación a los cambios. Con la gestión del conocimiento se busca que el estudiante adquiera las estrategias y las técnicas que le permitan aprender por sí mismo; esto implica la toma de conciencia de la asimilación, la reflexión y la interiorización del conocimiento para que finalmente pueda valorar y profundizar a partir de una opción personal. El docente fomenta la capacidad de resolver problemas mediante pensamiento crítico, potencia el diálogo entre profesionales y estudiantes, se comparten formas de comprensión y se permite reconstruir significados. La actividad es individual, y se vincula a la evaluación del curso por lo que la persona estudiante podrá aplicar los elementos de programación básica utilizados en el campo de la computación, para la representación de soluciones a los problemas contemporáneos. Las personas estudiantes podrán hacer uso de las bases de datos digitales de la universidad y la biblioteca para localizar información y realizar la investigación para los estudios de caso, así como uso de la bibliografía del curso

Proyecto: El proyecto final consiste en situar a un educando en un contexto que imite algún aspecto de la realidad y en establecer en ese ambiente situaciones, problemáticas o reproductivas, similares a las que él deberá enfrentar en su desempeño profesional contribuyendo a elevar la calidad del proceso de aprendizaje, por lo que la actividad se vincula a las competencias del curso. Con las temáticas correspondientes del curso de Introducción a la Programación de manera colaborativa entre los

estudiantes y con el acompañamiento del docente. Se desarrollarán los temas de Condicionales, Ciclos, Orientación a Objetos, Métodos, Constructores, Encapsuladores, Arreglos y la aplicación práctica de todos estos elementos para el curso de Introducción a la Programación. El proyecto es un trabajo colaborativo en grupos de estudiantes no mayores a los 4 integrantes, en donde se desarrollará la solución, con la participación periódica del docente que asesorará el avance de este. Se desarrollan las habilidades de trabajo colaborativo, cuyos marcos teóricos se verán aplicados en contextos reales y estos se vinculan a la evaluación, por lo que el propio estudiante identifica sus necesidades y pone en marcha los medios y las estrategias a su alcance para dar respuesta al problema. El docente ejerce como impulsor de ese primer reto y como apoyo en el camino hacia su solución, a modo de guía, supervisor y facilitador. Por lo que la persona estudiante podrá aplicar soluciones de tecnologías de información coherentes con los requerimientos de los usuarios y el entorno de estos, considerando el análisis de las necesidades de computación.

RECURSOS DIDÁCTICOS

Laboratorio informático: equipado con equipo de cómputo, video proyector, pizarra acrílica, servicio de Internet. Programas de cursos, guías de evaluación y recursos bibliográficos impresos y digitales de **Introducción a la programación**.

Biblioteca: Es un recurso que cuenta con gran cantidad de ejemplares, y que están a disposición tanto de estudiantes como de profesores, lo cual permite enriquecer el proceso de enseñanza-aprendizaje del curso de **Introducción a la programación**

Bases de datos de la universidad: Es un poderoso sistema de referencia en línea con búsquedas configurables de fácil manejo, que permiten una destacada recuperación de la información, este sistema ofrece una variedad de bases de datos de texto completo patentadas y bases de datos populares de proveedores de información líderes. Hoy en día la Biblioteca de la Universidad Fidélitas cuenta con acceso a nueve bases de datos que abarcan distintos temas sobre **Introducción a la programación**

Office 365 Educativo: Para el desarrollo del curso tanto el docente como los estudiantes tienen el acceso a Microsoft 365 (también conocido como Office 365) en su versión educativa, el cual es una herramienta que permite crear, acceder y compartir documentos de Word, Excel, PowerPoint y OneNote, entre otros; en la nube para el curso de **Introducción a la programación**

ACTIVIDADES DE EVALUACIÓN

La metodología de evaluación debe ser coherente con las competencias 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. Evaluación formativa: Proceso de evaluación continua, que ocurre durante la enseñanza aprendizaje, basado en la búsqueda e interpretación de evidencia, acerca del logro del estudiante respecto a un objetivo. Evaluación sumativa: Establece instrumentos. Se realiza al término de una etapa del proceso de una etapa del proceso de enseñanza aprendizaje, para verificar los resultados y objetivos. La evaluación de este curso refleja la evaluación formativa y sumativa.

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

Rubro	Porcentaje
Ejercicios prácticos (11 de 2% cada una)	22%
Casos programados (2 de 15% cada uno)	30%
Proyecto	48%

Total	100%
--------------	-------------

Rúbrica para ejercicios prácticos 22%, 2% cada una:

El método de Aprendizaje Basado en Problemas (ABP) aplicado a la resolución de ejercicios prácticos promueve en los estudiantes nuevos conocimientos, la práctica reflexiva y la adaptación a los cambios. El propio estudiante identifica sus necesidades y pone en marcha los medios y las estrategias a su alcance para dar respuesta al problema. El docente ejerce como impulsor de ese primer reto y como apoyo en el camino hacia su solución, a modo de guía, supervisor y facilitador. Se trata, además, de un proceso continuo, en el que los nuevos conocimientos no solo consiguen resolver el reto propuesto, sino que además plantean nuevos problemas y nuevas necesidades que siguen impulsando el aprendizaje, por lo que la actividad es individual y las personas estudiantes podrán discutir las bases necesarias del lenguaje de programación definido, considerando el estudio de este en la solución a diversos problemas. La práctica reflexiva permite razonar sobre la problemática planteada y posibilita la construcción del conocimiento mediante procesos de diálogo y discusión que ayudan a los estudiantes a desarrollar habilidades transversales de comunicación y expresión oral, al mismo tiempo que desarrollan el pensamiento crítico y la argumentación lógica, para la exploración de sus valores y de sus propios puntos de vista. Estas competencias les permiten afrontar una práctica profesional más reflexiva y crítica. La actividad se vincula a la competencia y evaluación del curso. Las indicaciones para realizar la actividad son las siguientes:

1. La solución del problema debe estar bien desarrollada. Se deben presentar nuevas ideas exploradas. La estrategia descrita debe apoyar los resultados.
2. El problema resuelto evidencia mayor nivel de pensamiento.
3. La explicación dada para el concepto se desarrolla fácilmente y es efectivo.
4. Los gráficos y textos desarrollados exhiben un grado de creatividad y ayudan al entendimiento de los procedimientos.
5. Todas las estrategias desarrolladas deben estar relacionadas con el tema. Se deben indicar la fuente de donde fueron tomados
6. Todos los textos y gráficos deben estar relacionados con el tema y hacen que sea más fácil de entender.
7. Desarrolla aportaciones de calidad y demuestra ser el resultado de un proceso de pensamiento propio y crítico
8. En la presentación existe dominio de los conceptos y estos son comunicados de manera efectiva

Nombre: Rubrica para evaluar ejercicios prácticos				
Producto:				
Ejercicio práctico 1: Introducción a la tecnología Java Ejercicio práctico 2: Uso de estructuras de decisión Ejercicio práctico 3: Uso de estructuras de repetición Ejercicio práctico 4: Programación orientada a objetos. Ejercicio práctico 5: Programación orientada a objetos. Ejercicio práctico 6: Arreglos Ejercicio práctico 7: Arreglos de objetos Ejercicio práctico 8: Ejercicio integrador Ejercicio práctico 9: Ejercicio integrador Ejercicio práctico Colegiado 1: Programación Orientada a Objetos Ejercicio práctico Colegiado 2: Ejercicio integrador				
Criterios por evaluar	Cumple con lo solicitado 2 puntos	Cumple parcialmente 1 punto	No cumple lo solicitado 0 puntos	Observaciones al estudiante
La solución del problema está bien desarrollada. Nuevas ideas han				

sido exploradas. La estrategia descrita apoya los resultados.				
El problema resuelto evidencia mayor nivel de pensamiento.				
Explicación del concepto se desarrolla fácilmente y es efectivo.				
Los gráficos y textos desarrollados exhiben un grado de creatividad y ayudan al entendimiento de los procedimientos.				
Todas las estrategias desarrolladas están relacionadas con el tema. Indican la fuente de donde fueron tomados				
Todos los textos y gráficos están relacionados con el tema y hacen que sea más fácil de entender.				
La resolución del problema demuestra creatividad/innovación y búsqueda de nuevas ideas y estrategias.				
Desarrolla aportaciones de calidad y demuestra ser el resultado de un proceso de pensamiento propio y crítico				
En la presentación existe dominio de los conceptos y estos son comunicados de manera efectiva				
La resolución y entrega del trabajo incluye todos los elementos requeridos a nivel de tiempo y forma.				
Utiliza correctamente la sintaxis y la ortografía en la redacción. Selecciona cuidadosamente las palabras. No utiliza el copiar y pegar				
Valor 22 puntos. Para obtener la nota se utiliza regla 3.				

Rúbrica para el Proyecto 48%:

El proyecto final consiste en situar a un educando en un contexto que imite algún aspecto de la realidad y en establecer en ese ambiente situaciones, problémicas o reproductivas, similares a las que él deberá enfrentar en su desempeño profesional contribuyendo a elevar la calidad del proceso de aprendizaje, por lo que la actividad se vincula a las competencias del curso. Con las temáticas correspondientes del curso de Introducción a la Programación de manera colaborativa entre los estudiantes y con el acompañamiento del docente. Se desarrollarán los temas de Condicionales, Ciclos, Orientación a Objetos, Métodos, Constructores, Encapsuladores, Arreglos y la aplicación práctica de todos estos elementos para el curso de Introducción a la Programación. El proyecto es un trabajo colaborativo en grupos de estudiantes no mayores a los 4 integrantes, en donde se desarrollará la solución, con la participación periódica del docente que asesorará el avance de este. Se desarrollan las habilidades de trabajo colaborativo, cuyos marcos teóricos se verán aplicados en contextos reales y estos se vinculan a la evaluación, por lo que el propio estudiante identifica sus necesidades y pone en marcha los medios y las estrategias a su alcance para dar respuesta al problema. El docente ejerce como impulsor de ese primer reto y como apoyo en el camino hacia su solución, a modo de guía, supervisor y facilitador. Por lo que la persona estudiante podrá aplicar

soluciones de tecnologías de información coherentes con los requerimientos de los usuarios y el entorno de estos, considerando el análisis de las necesidades de computación.
A continuación, se presenta un ejemplo para el enunciado del proyecto.

Las instrucciones para desarrollar el proyecto son las siguientes:

Descripción general del requerimiento – Parqueo “El descanso”:

La cadena de Parques “El descanso”, como forma de adaptarse a la nueva realidad, quiere contratar un sistema de información que le permita ser flexible con las medidas sanitarias y que le ayude a cumplir con porcentajes de aforo variables tanto en ingreso de vehículos, pago y demás.

Le han solicitado a su equipo de trabajo que les desarrollen la solución de acuerdo con una serie de requerimientos que se expondrán a continuación, pero ellos están de acuerdo con aceptar propuestas adicionales de los estudiantes, siempre y cuando estas no afecten los requerimientos originales.

Condiciones técnicas del requerimiento

Registro de la cantidad de espacios de parqueo: Solicitar la ubicación y cantidad de posibles vehículos a recibir en el parqueo

Módulo de registro de vehículos: Solicitar número de placa, forma de pago (efectivo, transferencia, tarjeta de crédito). Solicitar cuánto tiempo estimado de parqueo

Módulo de reservaciones de espacios: Solicitar los días del espacio, verificar la disponibilidad de acuerdo con las fechas y las reservaciones, verificar las tarifas de entre semana y fin de semana por persona por noche, calcular el monto a pagar, el IVA y el monto total.

Módulo de reservaciones de mensuales: Solicitar días, tiempos y horas de ocupación para reservar de acuerdo con la capacidad del parqueo.

Reportes de asignaciones: Emite un análisis de la ocupación mensual que lleva el sistema
Salir

Nota: Las estructuras de almacenamiento deben ser las más adecuadas debe adaptar la entrega a los temas vistos en el curso.

Nombre: Rubrica para evaluar el proyecto				
Producto: Informe de proyecto				
Criterios por evaluar	Cumple con lo solicitado 2 puntos	Cumple parcialmente 1 punto	No cumple lo solicitado 0 puntos	Observaciones al estudiante
Analizan el objetivo del proyecto				
Investigan con anticipación la información relacionada con la actividad para saber que necesitan				
Desarrollo de la práctica. El equipo reúne el material o software para simular				
Diseño del modelo de simulación mediante el uso correcto de herramientas tecnológicas				
Construcción del modelo de simulación				
Evidencia la creatividad técnica o artística en el desarrollo de la simulación				

Existe viabilidad en la comunicación que permita la comprensión para el usuario de lo producido o diseñado en la simulación				
El simulador enriquece la relación realidad- modelo				
El simulador es de fácil manipulación y permite rápido acceso a las distintas opciones.				
Alternativas de solución generadas				
Valor 20 puntos. Para obtener la nota se utiliza regla 3.				

Rubrica casos Programados 30%

Título: Rubrica casos programados				
Producto:	Casos Programados			
Valor:	15%			
Población:	Estudiantes de Ingeniería en Sistemas			
Indicadores	Niveles de Dominio			
	Receptivo (1) Saber Saber	Resolutivo (2) Saber Hacer	Autónomo (3) Saber Ser	Estratégico (4) Saber Convivir
DD.1 Desarrolla los requerimientos previamente establecidos en el enunciado o las historias de usuario.	Identifica los requerimientos que se van a desarrollar en el enunciado o las historias de usuario.	Aplica conocimientos técnicos en el desarrollo de los requerimientos establecidos en el enunciado o las historias de usuario.	Analiza la aplicación de buenas prácticas en el proceso de desarrollo de los requerimientos establecidos en el enunciado o las historias de usuario	Crea soluciones que satisfagan los requerimientos previamente establecidos en el enunciado o las historias de usuario.
DD.2 Aplica criterio técnico en la definición de una base de datos optimizada y funcional (en caso que aplique)	Identifica las entidades y atributos para cada tabla necesaria para la base de datos	Aplica conocimientos técnicos en la creación de la base de datos de acuerdo con el motor elegido	Analiza formas de normalización y buenas prácticas necesarias para la creación de la base de datos	Crea una base de datos optimizada y funcional alineada a los requerimientos técnicos del enunciado
DD.3 Vincula el desarrollo de los requerimientos o las historias de usuario a una adecuada funcionalidad.	Identifica los procesos asociados a una adecuada funcionalidad de acuerdo con el enunciado o las historias de usuario.	Aplica criterios de aceptación asociados a los requerimientos establecidos en el enunciado o las historias de usuario.	Analiza las interacciones del usuario final con los requerimientos establecidos en el enunciado o las historias de usuario.	Crea la funcionalidad requerida siguiendo los requerimientos y criterios de aceptación establecidos en el enunciado o las historias de usuario.
DD.4 Presenta el desarrollo de los requerimientos o las historias de usuario en una defensa formal.	Identifica los elementos necesarios para una defensa formal.	Aplica un formato de presentación para la defensa formal.	Analiza preguntas realizadas por el jurado sobre los requerimientos o las historias de usuario desarrollados.	Crea una defensa formal que abarca elementos técnicos y funcionales o de las historias de usuario estipulados en el enunciado.
DD5. Aporta criterio técnico en el desarrollo de la solución, como miembro activo del equipo	Identifica criterio técnico en el desarrollo de la solución, como miembro activo del equipo	Aplica criterio técnico en el desarrollo de la solución, como miembro activo del equipo	Analiza con criterio técnico el desarrollo de la solución, como miembro activo del equipo	Crea con criterio técnico un desarrollo de la solución, como miembro activo del equipo
DD6. Aplica cambios significativos al proyecto utilizando la herramienta	Identifica cambios significativos en el proyecto utilizando la	Aplica cambios significativos en el proyecto utilizando la	Analiza cambios significativos en el proyecto utilizando la	Crea cambios significativos en el proyecto utilizando la

de control de versiones (en caso de que se utilice)	herramienta de control de versiones. Menos del 50% de los miembros han realizado "Pull" dentro del proyecto de control de versiones, pull significativo de código.	herramienta de control de versiones. Entre del 50 y menos del 100% de los miembros han realizado "Pull" dentro del proyecto de control de versiones, pull significativo de código.	herramienta de control de versiones. Entre del 50 y menos del 100% de los miembros han realizado "Pull" dentro del proyecto de control de versiones, pull significativo de código.	herramienta de control de versiones. El 100% de los miembros han realizado "Pull" dentro del proyecto de control de versiones, pull significativo de código.
D7. Aplica técnicas de prototipado para el desarrollo de los requerimientos o las historias de usuario (en caso de que se solicite)	Identifica los elementos necesarios para la creación del prototipo	Aplica técnicas de prototipado, pero se presentan esquemas deficientes en cuanto a contenido y completitud de las historias de usuario	Analiza técnicas de prototipado y prototipo muestra algunas de las funcionalidades definidas por las historias de usuario o requerimientos	Crea un prototipo que está bien diseñado, con una herramienta adecuada y cumple a satisfacción lo solicitado en esta etapa
DD6.				
Nivel de dominio logrado				
Tipos de Evaluación	Logros		Aspectos a Mejorar	
Autoevaluación	DD.1: 4			
	DD.2: 4			
	DD.3: 4			
	DD.4: 4			
Sub Total	Sumatoria			
Coevaluación Pares (Compañero)	DD.1:			
	DD.2:			
	DD.3:			
	DD.4:			
Sub Total	Sumatoria			
Heteroevaluación Formativa Coaching del Profesor	DD.1:			
	DD.2:			
	DD.3:			
	DD.4:			
Sub Total	Sumatoria			
Nota Final	Regla de tres para obtener los puntos finales			

CRONOGRAMA

Semana	Contenidos	Actividades de aprendizaje Fechas / Entregable
1	Unidad 1: Introducción a la tecnología Java <ul style="list-style-type: none"> • 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". 	<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 • Ejercicio práctico #1
2	Unidad 2: Uso de estructuras de decisión <ul style="list-style-type: none"> • Utilidad de las estructuras de decisiones / condicionales. • Tablas de verdad. • El condicional IF y sus variantes. • El condicional SWITCH / CASE. 	<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. • Ejercicio práctico #2
3	Unidad 3: Uso de estructuras de repetición <ul style="list-style-type: none"> • Estructura de repetición WHILE. • Estructura de repetición DO / WHILE. • Estructura de repetición 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. • Ejercicio práctico #3
4	Unidad 4: Programación Orientada a Objetos <ul style="list-style-type: none"> • ¿Qué es? • Ventajas de la Programación Orientada a Objetos. • Definición de un requerimiento enfocado desde la perspectiva de Orientación 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. • Ejercicio práctico #4

5	Unidad 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. Ejercicio práctico #5
6	Unidad 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. Ejercicio práctico colegiado #1.
7	Integración de Conocimientos	<ul style="list-style-type: none"> • Caso programado • Entrega del 1er Avance del proyecto (ver rúbrica).
8	Unidad 7: 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. Ejercicio práctico #6
9	Unidad 8: 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. Entrega del 2º Avance del proyecto Ejercicio práctico #7
10	Ejercicio Integrador	<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. Ejercicio práctico #8
11	Cierre de Componentes Técnicos del Proyecto	<ul style="list-style-type: none"> • Se crean los espacios para que cada grupo presente al profesor la implementación de las estructuras, componentes y lógica utilizados. • El docente realizará las recomendaciones para los ajustes necesarios y definirá las fechas para las defensas finales. Ejercicio práctico #9

12	Práctica para la integración de conocimientos.	<ul style="list-style-type: none"> Realización de prácticas sobre los temas de las semanas anteriores como preparación para examen. Entrega del 3er avance del proyecto Ejercicio práctico colegiado #2.
13	Integración de conocimientos	<ul style="list-style-type: none"> Caso Programado.
14	Defensa final del proyecto del curso	<ul style="list-style-type: none"> Entrega y Exposición de proyectos finales.

BIBLIOGRAFIA

Bibliografía obligatoria

Deitel, P. & Deitel, H. (2021). *Cómo programar en Java*. Pearson Educación. <https://elibro.net/es/lc/ufidelitas/titulos/195602>

Salvaggio A. y Testa G. (2019). *JavaScript Guía completa*. Alfaomega, Marcombo. <https://www.alphaeditorialcloud.com/reader/javascript-guia-completa?location=1>

Bibliografía complementaria

Ceballos Sierra, F. (2015). *Java: interfaces gráficas y aplicaciones para internet* (4a. ed.). RA-MA Editorial. <https://elibro.net/en/lc/ufidelitas/titulos/106453> (Clásico última edición)

Joyanes Aguilar, L. (2020). *Fundamentos de programación: algoritmos, estructura de datos y objetos*. McGraw-Hill. <https://www.ebooks7-24.com/stage.aspx?il=10409&pg=&ed=>

López, I. (2019). *Curso avanzado de Java*. Alfaomega, Altaria. <https://www.alphaeditorialcloud.com/reader/curso-avanzado-de-java-manual-practico-2?location=2>

OBSERVACIONES GENERALES

El estudiante debe cumplir con todas las disposiciones de los reglamentos para la educación en modalidad virtual de la Universidad Fidélitas.

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.

Las situaciones anteriormente indicadas se penalizarán según el artículo 31 del reglamento estudiantil vigente, por lo que en una primera ocasión que se detecte y documente una falta el profesor consignará una nota de cero a la actividad evaluativa, y comunicará a vida estudiantil el hecho para su debido registro en el expediente académico del estudiante, si se detecta una segunda incidencia por parte del estudiante automáticamente pierde el curso y en una tercera ocasión documentada (independientemente del curso) provoca la pérdida de todos los cursos matriculados en ese cuatrimestre y la expulsión del programa académico y de la Universidad.