

CARRERA: Tecnicatura Superior en Análisis de Sistemas y Desarrollo de Software

PLAN DE ESTUDIOS: 013/2023

AÑO: 2024

CAMPO: Formación específica

CURSO: 1º Año

DIVISIÓN: Única

ASIGNATURA: Fundamentos de Programación

DOCENTE RESPONSABLE: Donato Sandoval

RÉGIMEN DE LA ASIGNATURA: Cuatrimestral (1º)

CANTIDAD DE HORAS-CÁTEDRA: 5 (Cinco)

CONDICIONES PARA REGULARIZAR LA ASIGNATURA.

El alumno accederá a la condición de REGULAR, si cumple con las siguientes condiciones:

- Asistir al 70% de las clases teóricas y prácticas (60% de asistencia con documentación probatoria) dictadas durante el presente cuatrimestre según Resolución del RAM en su artículo N°36.
- Obtener en la única evaluación parcial o en su respectiva recuperación, una calificación mínima de 60/100 puntos posibles.
- Obtener un mínimo del 80% de trabajos prácticos aprobados con un puntaje mínimo de 70/100.

CONDICIONES PARA RENDIR COMO ALUMNO REGULAR.

El estudiante se deberá presentar ante el Tribunal con DNI, libreta y programa de la asignatura.

- Para la acreditación del espacio y en condición de regular el alumno deberá realizar una defensa frente al tribunal, de una unidad o tema seleccionado por el docente. Luego de dicha exposición, el Tribunal podrá realizar preguntas sobre el mismo tema y sobre otros temas del programa.
- Se considera acreditado el espacio si el Tribunal califica al alumno con una nota de al menos 4 (cuatro) puntos sobre 10 (diez) posible.

CONDICIONES PARA RENDIR COMO ALUMNO LIBRE.

El estudiante se deberá presentar ante el Tribunal con DNI, libreta y programa de la asignatura.

Consta de dos instancias:

- En primera instancia, deberá realizar un examen escrito, que consistirá en la resolución de un problema computable, el cual abarcará todas las unidades temáticas planteadas en el presente programa. Para su aprobación deberá obtener una calificación mínima de 4 (cuatro).
- En segunda instancia deberá realizar la defensa de una unidad o tema seleccionado por el docente. Luego de dicha defensa, el Tribunal podrá realizar preguntas sobre el mismo tema y sobre otros temas del programa.

NOTA: Sólo podrá pasar a la segunda instancia si aprueba el examen escrito y la calificación final, en caso de aprobar el escrito será el promedio entre las dos instancias, en caso de desaprobado el escrito o la exposición, la nota del aplazo será la definitiva.

MARCO TEÓRICO

Actualmente, el mundo se encuentra en una profunda transformación impulsada por la emergencia de la cultura digital, en la cual, tanto el pensamiento computacional como la programación y robótica, tienen un rol fundamental, un rol fundamental en el surgimiento de nuevas tecnologías de automatización y de inteligencia artificial. La internet de las cosas, las fábricas inteligentes y los sistemas ciberfísicos, entre otros, dan cuenta de lo que muchos expertos llaman la cuarta revolución industrial.

¿Por qué implementar el Pensamiento Computacional y elementos de Programación?

Desde hace muchos años, el pensamiento lógico-matemático, algorítmico o más conocido como pensamiento computacional ha estado muy nombrado. El pensamiento computacional es el proceso de pensamiento implicado en la formulación y resolución de problemas computables donde dichas soluciones, pueden ser efectivamente ejecutadas por agente de procesamiento de información (humano, computador o una combinación de ambos). Se basa en cuatro pilares básicos: descomposición de problemas, reconocimiento de patrones, abstracción y diseño de algoritmos.

La integración de este campo de conocimiento permite a los estudiantes desarrollar habilidades fundamentales para solucionar diversas problemáticas sociales, crear oportunidades y prepararse para su integración en el mundo del trabajo. Ya que la capacidad de abstracción, de encontrar patrones, de ordenar de manera operativa y de identificar los componentes de un problema son habilidades sobre las que trabaja el pensamiento computacional, no están necesariamente vinculadas con una computadora, y pueden aplicarse a diversas situaciones. Es por eso que la programación resulta una disciplina fundamental en la educación contemporánea.

Algunos beneficios son:

- Desarrolla habilidades lingüísticas y numéricas: Para la solución de un problema (no necesariamente computacional), se necesita interpretar y analizar el problema, establecer un orden, una secuencia, abstraer ideas, reconocer patrones y diseñar soluciones.
- Estimula la creatividad: Para la solución de un problema computacional, existen muchos caminos a una solución y el dejar que los alumnos propongan una solución, motiva su desarrollo creativo.
- Trabaja la resolución de problemas complejos: Mediante la descomposición del problema en partes más pequeñas, haciendo su resolución más sencilla y manejable. No les enseñamos el procedimiento de solución, sino que ellos lo descubren y ese aprendizaje por descubrimiento le da un significado a un nuevo conocimiento.
- Fomenta el aprendizaje colaborativo: Si los alumnos trabajan en grupo para resolver problemas veremos que cada integrante es diferente y aporta un conocimiento distinto, desde su realidad subjetiva. Incentiva el espíritu crítico y facilita la interactividad.

Estos beneficios están relacionados con la teoría de las Inteligencias Múltiples de Howard Garden (1983): La Inteligencia Lingüística, Inteligencia Lógico-Matemática, Inteligencia Espacial, Inteligencia Interpersonal e Inteligencia Intrapersonal.

¿Dónde se aplica el Pensamiento Computacional y elementos de Programación?

El pensamiento computacional puede ser desarrollado y aplicado en distintas disciplinas o actividades de la vida cotidiana, lo cual nos plantea un nuevo desafío educativo para nuestros hijos y nuestra sociedad. Por ello es cada vez más necesario introducir el pensamiento computacional y elementos de Programación en el sistema educativo con el objetivo de preparar a los estudiantes para un mercado laboral cada vez más tecnológico, mejorando las habilidades intelectuales y haciendo uso de abstracciones para resolver problemas complejos.

El pensamiento computacional y elementos de programación, no se instalan en la educación como fin último para atender solamente aprendizajes relacionados con la formación científico-tecnológica, sino también para la promoción del asombro, la curiosidad, la creatividad, el análisis y la experimentación.

Además, es fundamental desarrollar propuestas de aprendizaje en programación con una perspectiva de género, diseñadas específicamente para mujeres, con el fin de integrarlas en campos que tradicionalmente han estado dominados por hombres.

PROPÓSITOS

- Generar espacios para el desarrollo de capacidades de comprensión, análisis de problemas computables mediante el pensamiento computacional (lógico, matemático y algorítmico).
- Propiciar espacios para la resolución de problemas computacionales aplicando algoritmos básicos y diagramas de bloque.
- Fomentar el pensamiento crítico por medio de debates y exposiciones orales de alternativas de resolución a problemas computacionales.
- Incentivar el aprendizaje autónomo a través de problemas complementarios.

METAS DE COMPRESIÓN

- Comprender los conceptos fundamentales de la teoría de álgebra de Boole y aplicarlos para diseñar circuitos lógicos sencillos.
- Comprender los conceptos fundamentales de la teoría de sistemas de numeración binaria, octal y hexadecimal; y aplicarlos en la resolución de problemas como suma, multiplicación y conversión entre sistemas.
- Comprender los conceptos fundamentales de la teoría de lógica proposicional, teoría de algoritmos y diagramación; relacionarlos y aplicarlos en la resolución de problemas computacionales.

CONTENIDOS

UNIDAD I: SISTEMAS DE NUMERACIÓN, LÓGICA PROPOSICIONAL Y ÁLGEBRA DE BOOLE

Sistemas de numeración en distintas bases: sistema binario, octal y hexadecimal. Operaciones aritméticas en cada uno de ellos. Conversión entre sistemas. Lógica proposicional. Álgebra de Boole en el conjunto $\{0,1\}$. Propiedades. Funciones booleanas. Circuitos lógicos.

UNIDAD2: RESOLUCIÓN DE PROBLEMAS COMPUTACIONALES

Introducción a la programación estructurada. Etapas en la resolución: análisis del problema (datos de entrada y salida, modelo matemático), diseño y desarrollo del algoritmo, prueba de escritorio. Algoritmo: concepto y características. Representación de algoritmos: pseudocódigo y diagramas de bloques. Estructuras de control. Acumuladores, contadores. Abstracción y resolución de algoritmos utilizando variables simples. Construcción de pruebas de escritorio. Variables indizadas unidimensionales: concepto y representación en la memoria principal. Taller de resolución de problemas computacionales con diagrama de bloques (diagramación) implementando variables simples e indizadas utilizando el programa Diagramar.

UNIDAD3: ALGORITMOS FUNDAMENTALES

Algoritmos de recorrido, búsqueda, ordenamiento, actualización.

METODOLOGÍA DE TRABAJO

La metodología se diseñará con el objetivo de promover un aprendizaje significativo, participativo y práctico, que permita a los estudiantes adquirir los conocimientos teóricos y desarrollar las habilidades

necesarias para comprender y aplicar los conceptos relacionados con la resolución de problemas computacionales

A continuación, se detallan las estrategias que se utilizarán:

1. Clases expositivas: Para presentar los conceptos teóricos fundamentales de cada unidad, las clases se realizarán utilizando recursos audiovisuales, esquemas y ejemplos prácticos para facilitar la comprensión de los temas.
2. Análisis de problemas: Se plantearán problemas computacionales con contexto reales, para que los estudiantes puedan aplicar los conocimientos adquiridos y practicar en la resolución de los mismos; fomentando de esta manera el pensamiento analítico y crítico.
3. Talleres: Se llevarán a cabo sesiones prácticas en el aula, donde los estudiantes deberán presentar soluciones a problemas específicos relacionados con los contenidos de la asignatura; bajo la supervisión y orientación del docente.
4. Trabajos en equipo: Se fomentará el trabajo colaborativo a través de la realización de trabajos prácticos grupales, promoviendo de esta manera el intercambio de ideas, el debate y la resolución conjunta de problemas.
5. Micro exposiciones: Después de algunas clases expositivas, como cierre, algunos estudiantes deberán pasar al frente a explicar con sus propias palabras algún concepto visto en clases, con el objetivo de incentivar la expresión oral y corroborar la comprensión de los conceptos básicos.
6. Evaluación formativa y continua: Se realizarán evaluaciones periódicas durante el cursado, tanto formativas (micro exposiciones y trabajos grupales) como sumativas (exámenes escritos u orales), que permitan verificar el progreso de los estudiantes y retroalimentar su aprendizaje de manera constante. Se valorará tanto el conocimiento teórico como la capacidad para aplicar los conceptos en situaciones prácticas.

En resumen, la metodología para la asignatura "Fundamentos de Programación" combinará la exposición teórica, análisis de problemas, talleres, trabajos en equipo, micro exposiciones y el uso de recursos tecnológicos, con el objetivo de proporcionar a los estudiantes una formación integral que les permita desenvolverse con éxito en el ámbito de la programación e informática.

EVALUACIÓN PARA EL CURSADO

Criterios de evaluación

Se evaluará según los siguientes criterios:

- Conocimiento de los conceptos fundamentales de la teoría de álgebra de Boole y su aplicación para el diseño de circuitos lógicos sencillos.
- Conocimiento de los conceptos fundamentales de la teoría de lógica proposicional, sistemas de numeración, teoría de algoritmos y diagramación; y su aplicación en la resolución de problemas computacionales.
- Capacidad de resolución de problemas computacionales utilizando diagrama de bloques.
- Participación y pertinencia de las actuaciones e intervenciones.
- Predisposición para la tarea y el aprendizaje individual y grupal.

EVALUACIÓN PARA RENDIR EXÁMENES FINALES

Criterios de evaluación

Se evaluará según los siguientes criterios:

- Experticia en la resolución de problemas computacionales frente al tribunal.
- Demuestra conocimientos sólidos al responder preguntas del tribunal sobre diferentes temas.

BIBLIOGRAFÍA

Bibliografía específica

- Joyanes, Aguilar L. (2008). Fundamentos de Programación. Algoritmos, estructura de datos y objetos. Cuarta edición. McGraw-Hill/Interamericana de España.
- Jiménez Murillo, José A (2015). Matemática para Informática. Segunda edición. Alfaomega Grupo Editor y Marcombo S.A.
- Floyd, Thomas L. (2006). Fundamentos de sistemas digitales. Novena edición. Pearson educación S.A.

Bibliografía general

- Wikipedia (s.f.). Resolución de problemas de programación. https://es.wikipedia.org/wiki/Resoluci%C3%B3n_de_problemas_de_programaci%C3%B3n
- Wikipedia (s.f.). Sistema de numeración. https://es.wikipedia.org/wiki/Sistema_de_numeraci%C3%B3n#Ejemplo_en_el_sistema_decimal
- Wikipedia (s.f.). Sistema binario. https://es.wikipedia.org/wiki/Sistema_binario#Operaciones_con_n%C3%BAmeros_binarios
- Wikipedia (s.f.). Sistema octal. https://es.wikipedia.org/wiki/Sistema_de_numeraci%C3%B3n_octal
- Wikipedia (s.f.). Sistema hexadecimal. https://es.wikipedia.org/wiki/Sistema_hexadecimal
- Wikipedia (s.f.). Algebra de Boole. https://es.wikipedia.org/wiki/%C3%81lgebra_de_Boole#%C3%81lgebra_de_conjuntos