



PROGRAMA DE LA ASIGNATURA

1. INFORMACIÓN GENERAL DE UBICACIÓN DE LA ASIGNATURA

Área: Ciencias Tecnológicas y Agrarias

Docente: Camacho Calderon Francisco Javier

Facultad: CIENCIA Y TECNOLOGIA

Carrera: ING.QUIMICA

Sistema: Normal Semestralizado

Asignatura: INFORMÁTICA I

Gestión: 1/2026

Sigla: MAT204

Fecha: 18-02-2026

Curso: 3

2. DESCRIPCIÓN Y JUSTIFICACIÓN DE LA ASIGNATURA

La solución de problemas en Ciencias e Ingeniería, implica realizar cientos y hasta miles cálculos y tomas de decisiones. Llevar a cabo estas operaciones en forma manual, requiere desde unas cuantas horas hasta meses, además los cálculos manuales son propensos a errores, por lo que no es posible garantizar la confiabilidad de los resultados obtenidos.

Por esa razón, en las carreras del área de ciencias y tecnología, es imprescindible contar con una herramienta que permita automatizar dichos cálculos, así como garantizar la confiabilidad de los resultados obtenidos, siendo la herramienta por excelencia para ese fin la informática.

Informática I es una asignatura básica, que forma parte de las materias del área de matemática, orientada a la adquisición de conocimientos y habilidades para desarrollar programas de computadora, una habilidad actualmente imprescindible en todas las áreas de la ingeniería, pues permite resolver de manera eficiente y confiable, los problemas complejos que se presentan tanto en la etapa de formación profesional, como en el ejercicio profesional.

Al ser una asignatura de nivel básico, en la misma se resuelven problemas reales simples, pero principalmente problemas modelo que ayudan a afianzar los conocimientos y adquirir práctica, tanto con el lenguaje de programación como con la lógica involucrada. A pesar de no ser problemas reales concretos, los problemas modelo, o variantes de los mismos, se presentan con frecuencia en muchas situaciones prácticas reales, por lo que los conocimientos y habilidades adquiridos son de utilidad real tanto en la formación como en el desempeño laboral de los futuros profesionales.

Estos conocimientos y habilidades son también la base sobre la cual se construyen los conocimientos y habilidades en asignaturas posteriores del área y en asignaturas de la profesión en general.

A la conclusión del curso, el estudiante será capaz de: Analizar un problema, plantear una solución en forma de algoritmo, representarla en un diagrama de actividades y codificarla en un lenguaje de programación.

3. RELACIONES DE LA ASIGNATURA

La asignatura tiene las siguientes relaciones horizontales y verticales con otras asignaturas de la carrera. Horizontalmente, se relaciona con las siguientes asignaturas:

- Ecuaciones diferenciales (**MAT207**), pues permite programar los métodos numéricos que se estudian en la misma.
- Físico química (**QMC206**), ya que permite elaborar programas que simplifican el cálculo de propiedades y la resolución de ecuaciones de estado.
- Física básica III (**FIS 200**) y Química Orgánica II (**QMC 204**), donde permite programar y simular modelos físicos y químicos que se estudian en las mismas.

Verticalmente, se relaciona con las siguientes asignaturas:

- Álgebra I (**MAT 100**) que brinda la base teórica de lógica que se requiere para la asignatura.
- Cálculo I (**MAT 101**) se constituye también en un prerrequisito de la asignatura pues en dicha asignatura se estudian algunos métodos numéricos básicos y algunas aplicaciones del cálculo que son útiles en la búsqueda de soluciones numéricas a los problemas de ingeniería.
- Álgebra II o Álgebra Lineal y Teoría Matricial (**MAT 103**) que se constituye en un prerrequisito de la asignatura pues brinda la base teórica del álgebra matricial muy útil en las soluciones numéricas de los problemas de ingeniería y asegura haber aprobado también Álgebra I.
- Física Básica I (**FIS 100**), Física Básica II (**FIS 102**) y Química General (**QMC 100**), porque debido al nivel en el que se imparte la asignatura se la aplicará directamente a la solución de problemas relacionados con estas asignaturas, por lo que su dominio es requerido.
- Análisis Numérico, Métodos Numéricos o Informática II (**MAT 205**), asignatura a la que sirve de base, y en la que se aplican los conocimientos y habilidades adquiridos en Informática I, y se los complementa estudiando los Métodos Numéricos que permitirán al futuro profesional plantear soluciones numéricas a problemas matemáticos, así como su aplicación a la resolución de problemas prácticos de ingeniería. Es importante notar que **son las soluciones numéricas y no las analíticas las que se adecúan mejor al uso de las computadoras como herramienta para la solución de problemas profesionales.**



PROGRAMA DE LA ASIGNATURA

4. OBJETIVO GENERAL

Desarrollar programas informáticos empleando la lógica, estructuras algorítmicas y de datos, y comandos más adecuados para resolver problemas profesionales propios de ciencias e ingeniería, velando que los resultados obtenidos sean confiables.

5. OBJETIVOS ESPECÍFICOS

Objetivos Instructivos

- Diseñar algoritmos, siguiendo los lineamientos del razonamiento lógico, para resolver problemas de ciencias y tecnología.
- Codificar los algoritmos diseñados en un lenguaje de programación.
- Aplicar las estructuras, comandos y funciones disponibles en el lenguaje de programación, para implementar los algoritmos diseñados, evitando la repetición de código, con el fin de lograr programas claros y eficientes.
- Aplicar el principio de modularidad, por medio de la elaboración de funciones, para resolver problemas complejos dividiéndolos en problemas más simples.
- Optimizar la resolución de problemas, mediante la aplicación de las estructuras selectivas, iterativas, y el principio de recursividad, a fin de lograr soluciones eficientes, claras y sencillas a los problemas de ciencias y tecnología.
- Aplicar algoritmos y procesos a un conjunto de datos, empleando datos estructurados, para resolver problemas complejos, de manera más eficiente.
- Probar las soluciones implementadas, ejecutando los programas con valores estándar y valores límite, con la finalidad de garantizar la validez de los resultados.

Objetivos Educativos

- Desarrollar la imaginación como instrumento para la aplicación de los conceptos teóricos a la solución de problemas cotidianos.
- Desarrollar las formas del pensamiento lógico y la capacidad de razonamiento y abstracción mediante la formación de un sistema de habilidades que le permitan plantear soluciones generalizadas de un problema, de manera que pueda plantear las mismas en forma de algoritmos que puedan ser implementados como un programa de computadora.
- Comprende que no solo es necesario saber codificar, sino que para resolver un problema se debe tener una comprensión completa del problema y su entorno, apropiándose del concepto de que Programar es Enseñar, y nadie enseña lo que no sabe y comprende, bajo el supuesto de que la computadora nada sabe, sino solo aquello que le enseñamos a través de un programa.
- Comprender que el orden y la disciplina son normas de vida que todo ingeniero debe tener, que la ética y la honestidad son valores y atributos que deben caracterizarlo, y que dichos valores deben reflejarse en actitudes tanto de su vida profesional como de su vida social y privada.
- Valorar con rigor científico-técnico las ventajas del método termodinámico para resolver problemas en el campo del equilibrio físico y químico.
- Adquirir sensibilidad y responsabilidad frente a los impactos negativos que generan las actividades humanas sobre el medio ambiente.
- Comprender que el hombre puede decidir la manera como interviene en el medio ambiente, valorando las consecuencias ambientales, económicas y sociales de sus actividades.
- Comprender que toda acción que contribuya a reducir los impactos negativos sobre el medio ambiente favorece también a mejorar la calidad de vida del hombre y a preservar el entorno natural en el que vivimos.
- Asumir que el autocontrol es la base de una vida mesurada y armoniosa con la sociedad y que la práctica constituye la base del conocimiento, para crear una personalidad cualitativamente nueva.
- Asumir que la auto - preparación es fundamental para una buena formación y una constante actualización en el campo de la ingeniería, debido a que es una ciencia que se desarrolla permanentemente.

6. CONTENIDOS MÍNIMOS

Tema 1: Introducción. (14 % del avance)		
Objetivo particular: Conocer los fundamentos básicos de programación, los tipos datos básicos y resolver problemas matemáticos simples, en el lenguaje de programación, empleando operadores, variables y funciones propias del lenguaje.		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores



PROGRAMA DE LA ASIGNATURA

<p>* Conoce y comprende el concepto de programación * Conoce el léxico, la sintaxis y la semántica de los lenguajes de programación. * Conoce el lenguaje de programación y sus características. * Conoce el entorno de programación y su funcionamiento. * Conoce y comprende lo que es una variable en programación. * Conoce los tipos de datos numéricos. * Conoce y comprende la forma en que se programan expresiones matemáticas simples. * Conoce los operadores aritméticos, el orden de prioridades de los operadores y el uso de paréntesis. * Conoce la sentencia de asignación y su relación con las variables. * Conoce las cadenas de caracteres y las operaciones básicas que se pueden realizar con ellas. * Conoce los tipos de datos lógicos y las operaciones básicas que se pueden realizar con ellos.</p>	<p>* Emplea correctamente el léxico, la sintaxis y la semántica del lenguaje de programación elegido. * Emplea con solvencia el entorno de programación. * Sabe declarar y emplear variables. * Emplea correctamente los tipos de datos numéricos. * Programa correctamente expresiones matemáticas simples. * Emplea correctamente el orden de prioridad de los operadores aritméticos y los paréntesis, al programar expresiones matemáticas. * Guarda los resultados obtenidos, al evaluar expresiones matemáticas, en variables. * Crea y emplea correctamente cadenas de caracteres simples. * Crea y emplea correctamente datos de tipo booleano.</p>	<p>* Es honesto. * Es ordenado. * Es disciplinado. * Es responsable, cumplido en la presentación de trabajos. * Es autodidacta, estudia en forma independiente. * Es crítico, no acepta un resultado sin analizar su coherencia. * Es autocrítico, y plantea sus dudas para que le sean absueltas demostrando así madurez.</p>
---	---	--

Tema 2: Algoritmos. (10 % del avance)

Objetivo particular:Elaborar algoritmos para la solución, paso a paso, de problemas simples, verificando que la lógica de la solución propuesta sea correcta.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>* Conoce y comprende el propósito de los algoritmos. * Describe e identifica los componentes de un algoritmo. * Conoce la forma de elaborar algoritmos empleando el lenguaje natural. * Conoce la forma de elaborar algoritmos empleando diagramas de flujo. * Conoce la forma de elaborar algoritmos empleando pseudocódigo. * Conoce la forma de incorporar algoritmos en el código que lo implementa.</p>	<p>* Elabora algoritmos para resolver problemas simples, empleando los componentes estándar de un algoritmo. * Elabora algoritmos, lógicamente correctos, empleando el lenguaje natural. * Elabora algoritmos, lógicamente correctos, empleando diagramas de flujo. * Elabora algoritmos, lógicamente correctos, empleando pseudocódigo. * Elabora algoritmos, lógicamente correctos, directamente en el código donde luego se implementa el mismo.</p>	<p>* Es honesto. * Es ordenado. * Es disciplinado. * Es responsable, cumplido en la presentación de trabajos. * Es autodidacta, estudia en forma independiente. * Es crítico, no acepta un resultado sin analizar su coherencia. * Es autocrítico, y plantea sus dudas para que le sean absueltas demostrando así madurez.</p>

Tema 3: Funciones. (12 % del avance)

Objetivo particular:Elaborar funciones simples, en el lenguaje de programación elegido, para resolver problemas matemáticos y físicos simples, verificando que las soluciones obtenidas sean correctas.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>* Comprende la relación que existe entre una función matemática y una función de programación. * Conoce la forma en que se declara una función estándar y las partes de las que consta. * Conoce la forma en que se invoca (llama) a una función. * Conoce los tipos de parámetros que puede recibir una función. * Conoce los tipos de funciones que pueden ser declaradas. * Conoce los principios de la programación modular. * Sabe cómo programar funciones matemáticas y físicas simples.</p>	<p>* Programa funciones estándar para resolver problemas simples. * Recibe los datos, en los tipos de parámetros más adecuados, para el problema que se está resolviendo. * Invoca a funciones propias y del sistema, para resolver problemas matemáticos simples. * Emplea el tipo de función más adecuado para el problema que se está resolviendo.</p>	<p>* Es honesto. * Es ordenado. * Es disciplinado. * Es responsable, cumplido en la presentación de trabajos. * Es autodidacta, estudia en forma independiente. * Es crítico, no acepta un resultado sin analizar su coherencia. * Es autocrítico, y plantea sus dudas para que le sean absueltas demostrando así madurez.</p>

Tema 4: Estructuras Selectivas. (16 % del avance)

Objetivo particular:Elaborar programas, empleando estructuras selectivas, para resolver problemas que involucran la toma de decisiones.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores



PROGRAMA DE LA ASIGNATURA

<p>* Conoce los operadores relacionales y comprende su uso. * Conoce los operadores lógicos comprende su uso. * Comprende la evaluación de expresiones lógicas en corto circuito. * Conoce la estructura "if" y la lógica que implica. * Conoce la estructura "if-else" y la lógica que implica. * Conoce el operador lógico "?" y comprende su uso. * Conoce la estructura "switch" (o su equivalente) y comprende la lógica involucrada.</p>	<p>* Escribe expresiones booleanas empleando expresiones relacionales. * Escribe expresiones booleanas empleando expresiones lógicas. * Resuelve problemas de selección empleando las estructuras "if" e "if-else". * Resuelve problemas de selección empleando el operador "?". * Resuelve problemas de selección empleando la estructura "switch". * Elige correctamente la estructura más adecuada para resolver problemas de selección.</p>	<p>* Es honesto. * Es ordenado. * Es disciplinado. * Es responsable, cumplido en la presentación de trabajos. * Es autodidacta, estudia en forma independiente. * Es crítico, no acepta un resultado sin analizar su coherencia. * Es autocrítico, y plantea sus dudas para que le sean absueltas demostrando así madurez.</p>
--	---	--

Tema 5: Estructuras Iterativas. (18 % del avance)

Objetivo particular:Elaborar programas, empleando las estructuras iterativas, para resolver problemas de carácter repetitivo.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>* Conoce y comprende la lógica de la estructura "while". * Conoce y comprende la lógica de la estructura "do-While". * Conoce y comprende la lógica de las estructuras "for". * Conoce los ciclos iterativos infinitos y sabe cómo identificarlos y/o evitarlos. * Sabe cómo romper un ciclo iterativo y salir del mismo. * Sabe cómo saltar a la siguiente iteración de un ciclo. * Sabe cómo capturar y tratar errores generados en el código.</p>	<p>* Elabora programas, que resuelven problemas iterativos, empleando correctamente las estructuras "while", "do-while" y "for". * Elabora programas que rompen los ciclos y/o saltan a las siguientes iteraciones, para conseguir soluciones más claras y eficientes. * Atrapa los errores que genera el código y responde a los mismos.</p>	<p>* Es honesto. * Es ordenado. * Es disciplinado. * Es responsable, cumplido en la presentación de trabajos. * Es autodidacta, estudia en forma independiente. * Es crítico, no acepta un resultado sin analizar su coherencia. * Es autocrítico, y plantea sus dudas para que le sean absueltas demostrando así madurez.</p>

Tema 6: Recursividad. (10 % del avance)

Objetivo particular:Elaborar programas para resolver problemas iterativos, empleando funciones recursivas, cuando con ello se logra una solución más sencilla.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>* Comprende el concepto de Recursividad. * Comprende la lógica que involucran los procesos recursivos. * Sabe cómo crear funciones recursivas y comprende la forma en que operan. * Sabe cómo plantear y probar algoritmos recursivos. * Conoce la forma en que se programan las fórmulas matemáticas recursivas.</p>	<p>* Elabora programas que resuelven, de forma sencilla, algunos problemas iterativos. * Plantea y valida algoritmos recursivos simples. * Elabora programas para resolver fórmulas matemáticas recursivas.</p>	<p>* Es honesto. * Es ordenado. * Es disciplinado. * Es responsable, cumplido en la presentación de trabajos. * Es autodidacta, estudia en forma independiente. * Es crítico, no acepta un resultado sin analizar su coherencia. * Es autocrítico, y plantea sus dudas para que le sean absueltas demostrando así madurez.</p>

Tema 7: Arreglos. (20 % del avance)

Objetivo particular:Elaborar programas, para resolver problemas que involucran conjuntos de datos, haciendo uso datos estructurados de tipo arreglo.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>* Conoce la forma de declarar arreglos. * Sabe cómo acceder y modificar los elementos de un arreglo. * Conoce las funciones y/o métodos para el manejo de arreglos. * Comprende el uso de las estructuras "for" en la manipulación de los elementos de un arreglo. * Conoce la forma de elaborar programas para el trabajo con vectores. * Conoce la forma de elaborar programas para el trabajo con matrices.</p>	<p>* Crea y emplea variables de tipo arreglo. * Emplea con solvencia la estructura "for" para manipular los elementos de un arreglo. * Elabora programas que reciben, modifican y/o utilizan los elementos de un vector. * Crea programas que reciben, modifican y/o utilizan los elementos de una matriz.</p>	<p>* Es honesto. * Es ordenado. * Es disciplinado. * Es responsable, cumplido en la presentación de trabajos. * Es autodidacta, estudia en forma independiente. * Es crítico, no acepta un resultado sin analizar su coherencia. * Es autocrítico, y plantea sus dudas para que le sean absueltas demostrando así madurez.</p>

7. DISTRIBUCIÓN DEL FONDO DEL TIEMPO

Nro	Tema	Horas Teo. (P/V)	Horas Prácticas (P/V)			Horas Lab.	Horas Extrac.	Horas Eval.	Total Horas
			Taller	Invest.	Exten.				



PROGRAMA DE LA ASIGNATURA

1	Introducción. (14 % del avance)	6	0	6	0	6	10	2	20
2	Algoritmos. (10 % del avance)	2	0	2	0	2	2	1	7
3	Funciones. (12 % del avance)	2	0	2	0	4	4	1	9
4	Estructuras Selectivas. (16 % del avance)	6	0	6	0	7	10	2	21
5	Estructuras Iterativas. (18 % del avance)	6	0	6	0	8	10	2	22
6	Recursividad. (10 % del avance)	2	0	6	0	4	6	1	13
7	Arreglos. (20 % del avance)	6	0	8	0	8	10	2	24
Total horas semestre									116
Total horas extracurriculares semestre									52

P = Presencial; V = Virtual

8. CRONOGRAMA

8.1 Cronograma de plan temático y actividades

PLAN TEMÁTICO

Nro	Tema	Semanas																					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
1	Introducción. (14 % del avance)	■	■	■																■	■	■	
2	Algoritmos. (10 % del avance)			■	■																■	■	■
3	Funciones. (12 % del avance)					■	■	■													■	■	■
4	Estructuras Selectivas. (16 % del avance)							■	■	■	■										■	■	■
5	Estructuras Iterativas. (18 % del avance)									■	■	■	■								■	■	■
6	Recursividad. (10 % del avance)															■	■				■	■	■
7	Arreglos. (20 % del avance)																■	■	■		■	■	■

Actividades

No hay actividades

8.2 Cronograma de evaluaciones (parciales, final y segunda instancia)

Evaluación	Grupo	Fecha	Temas
Primer Parcial	2	10-04-2026	- Introducción. (14 % del avance) - Algoritmos. (10 % del avance) - Funciones. (12 % del avance) - Estructuras Selectivas. (16 % del avance)
Segundo Parcial	2	22-05-2026	- Introducción. (14 % del avance) - Algoritmos. (10 % del avance) - Funciones. (12 % del avance) - Estructuras Selectivas. (16 % del avance) - Estructuras Iterativas. (18 % del avance) - Recursividad. (10 % del avance)
Final	2	12-06-2026	- Introducción. (14 % del avance) - Algoritmos. (10 % del avance) - Funciones. (12 % del avance) - Estructuras Selectivas. (16 % del avance) - Estructuras Iterativas. (18 % del avance) - Recursividad. (10 % del avance) - Arreglos. (20 % del avance)



PROGRAMA DE LA ASIGNATURA

Segunda Instancia	2	23-06-2026	<ul style="list-style-type: none">- Introducción. (14 % del avance)- Algoritmos. (10 % del avance)- Funciones. (12 % del avance)- Estructuras Selectivas. (16 % del avance)- Estructuras Iterativas. (18 % del avance)- Recursividad. (10 % del avance)- Arreglos. (20 % del avance)
-------------------	---	------------	--

9. INDICACIONES METODOLÓGICAS Y DE ORGANIZACIÓN

Como ocurre con la mayoría de las asignaturas en el campo de la ingeniería, Informática I, es una asignatura eminentemente práctica, donde no es posible, ni recomendable, separar la teoría de la práctica, razón por la cual es imprescindible que el alumno pruebe, diseñe e implemente las soluciones en un dispositivo programable (computadora de escritorio, portátil, tableta, teléfono móvil o consola de juegos) a la par que estudia los conceptos y ejemplos, pues no es posible asimilar a cabalidad un concepto o principio, si no es probado y puesto en práctica.

Por lo anteriormente expuesto, la forma de organización, empleada hasta finales del anterior siglo (debido al limitado acceso que se tenía a los dispositivos programables): las clases de laboratorio, han quedado obsoletas, pues actualmente los programas pueden ser elaborados en prácticamente cualquier dispositivo programable y la gran mayoría de los estudiantes cuentan con al menos un dispositivo programable: el celular.

Por ello, la asignatura está estructurada de manera que sea 100% práctica, pero sin necesidad de realizarlas necesariamente en los laboratorios de informática, debido al acceso prácticamente universal a los dispositivos programables.

La metodología de enseñanza - aprendizaje, se ajusta a los modelos la clase invertida (Flipped Classroom) y aprendizaje para el dominio (Mastery Learning) y en concordancia con estos modelos los procesos de evaluación continua y aprendizaje a ritmo individual.

Si bien la Asignatura esta formalmente organizada en 7 temas, por razones practicas, se ha estructurado en 12 capítulos, en los cuales se pone a disposición de los estudiantes todo el material de forma digital a través de Internet y de la Plataforma Virtual de Aprendizaje de la Asignatura (en la página www.inghpu.com/materias/mat204). Dicho material contiene la teoría, tutoriales, ejemplos (interactivos) y ejercicios (interactivos).

Bajo el denominativo de "ejercicios" se engloban las tareas, cuestionarios y ejercicios propiamente dichos.

La teoría debe ser estudiada, los ejemplos analizados y probados y los ejercicios resueltos (los ejercicios son corregidos automáticamente por el sistema). Los ejercicios deben ser resueltos en su totalidad y aprobados con una nota mínima de 70 puntos. No existe límite de repeticiones para resolver los ejercicios, de manera que el estudiante puede resolverlos varias veces hasta aprobar los mismos, y aun si los tiene aprobados para perfeccionarlos y aprobarlos con la nota máxima.

Al concluir un capítulo (terminar de resolver satisfactoriamente los ejercicios planteados), el estudiante queda habilitado para rendir una prueba del capítulo, una defensa (evaluación continua). La prueba se genera automáticamente con ejercicios y opciones aleatorias y está basada en los ejercicios resueltos en el capítulo, puede ser rendida en cualquier momento, debe ser aprobada con una calificación mínima de 70 puntos y durante la misma la ventana de la prueba se pone en pantalla completa, no se permite cambiar de ventana y tampoco copiar texto u otro material en la misma (para minimizar la posibilidad de fraude). En casos excepcionales, cuando la defensa no puede ser tomada de forma presencial, la misma es controlada y autorizada mediante Zoom, con apoyo de otras herramientas para garantizar la privacidad y la identificación inequívoca de los participantes de una videoconferencia.

Los estudiantes pueden estudiar el material, resolver los ejercicios y rendir las defensas a su propio ritmo y desde el lugar donde se sientan mas cómodos, solo las defensas deben realizarlas de forma presencial en el aula y en los horarios de clases asignados. No existe límite de repeticiones, pero una repetición implica el rehacer el capítulo (Mastery Learning), de manera que al igual que con los ejercicios, el estudiante puede repetir un capítulo cuantas veces requiera hasta aprobar el mismo y aun si lo aprobó, puede repetirlo para mejorar su nota y aprobarlo con la nota máxima. Los únicos límites de tiempo son los impuestos por el calendario académico para las evaluaciones parciales y finales.

Es importante notar que los 12 capítulos en los que la asignatura esta estructurada, estan diseñados (pensados, medidos, graduados y divididos) para que un estudiante pueda estudiar y defender un capítulo por semana, dedicándole un tiempo equivalente al de las horas de clases asignadas a la asignatura.

Sin embargo, para defender un capítulo, el estudiante debera haber defendido y aprobado todos los capítulos anteriores, ya que no esta permitido omitir o saltar un capítulo a fin de garantizar que el estudiante, adquiera todos los conocimientos y habilidades que se busca desarrollar en el, con la asignatura.

En los horarios de clases (durante un semestre normal) el docente responde a las consultas individuales de los estudiantes, expone, de ser necesario, conceptos y/o puntos conflictivos y guía al estudiante en su proceso de formación (Flipped Classroom). En situaciones excepcionales (cuando las clases presenciales son interrumpidas) el docente responde a las consultas y guía al estudiante empleando un medio de comunicación social como WhatsApp o Zoom.

En cuanto a la fuente de adquisición de conocimientos, la metodología empleada se ajusta a los métodos de enseñanza práctica,



PROGRAMA DE LA ASIGNATURA

en lo referente a la relación de la actividad del profesor y los estudiantes, se ajusta al método de trabajo independiente y en lo relativo al carácter de la actividad cognoscitiva, se ajusta al método de solución de problemas.

En lo concerniente a los medios de enseñanza, se han elaborado tutoriales para los procesos más usuales y se ha elaborado el sistema informático que se encarga de hacer el seguimiento y evaluación de los estudiantes.

Como ya se explicó, en lo concerniente a los medios didácticos virtuales, en los medios expositivos se emplea el material digital elaborado con los contenidos de la materia y los ejemplos interactivos, para la interacción asíncrona se emplean las herramientas y la página de la materia, creadas y desarrolladas por el Ing. Hernán Gonzalo Peñaranda Villavicencio, disponible en: www.inghvpv.com/materias/mat204, mientras que la interacción síncrona se realiza por WhatsApp.

10. RECURSOS DIDÁCTICOS

En el desarrollo de la materia se emplean los siguientes recursos:

- Plataforma Virtual de la Asignatura, con herramientas creadas y desarrolladas por el Ing. Hernán Gonzalo Peñaranda Villavicencio. En ella esta disponible el material de de estudio y de consulta, así como con los enunciados de los ejercicios y problemas planteados. También a través de ella se ejecutan todas las actividades de aprendizaje descritas en las Indicaciones Metodológicas y de organización.
- Computadora y/o dispositivos móviles programables, a través de los cuales el docente pone a disposición del estudiante el material de aprendizaje en la Plataforma Virtual de la Asignatura y los estudiantes interactúan con la mencionada plataforma.
- Conexión a Internet, imprescindible para acceder a la Plataforma Virtual de la Asignatura
- Archivos con el material de estudio y de consulta, así como con los enunciados de los ejercicios y problemas planteados.
- Visual Studio Code.
- Audacity.
- Demo Builder.
- Adobe Premiere.
- Katex.
- Navegadores Internet.
- www.w3schools.com/
- www.mycompiler.io/#
- Intranet del Laboratorio
- Router wi-fi.
- Pizarra de acrílico.
- Marcador.
- Almohadilla.

11. ACTIVIDADES DE INVESTIGACIÓN Y/O INTERACCIÓN

Las actividades de investigación en la materia se llevan a cabo, mediante de la realización de los trabajos que se proporcionan a los estudiantes en los diferentes temas. Se trata sobre todo de trabajos de investigación teóricos, con la implementación práctica correspondiente mediante la elaboración de programas de computadora.

Al ser una materia del área básica, los estudiantes no cuentan aún con el cúmulo de conocimientos y habilidades necesarias como para proponer soluciones completas a problemas reales. Por ello las actividades de interacción se reducen al acopio de información en el medio y a la solución de problemas básicos, pensados sobre todo en automatizar algunas tareas tediosas.

12. SISTEMA DE EVALUACIÓN DE APRENDIZAJE

En la asignatura en concordancia a los modelos educativos adoptados, los conocimientos están organizados de forma incremental acumulativa, de manera que no es posible adquirir las capacidades de un tema sin haber adquirido previamente los conocimientos, habilidades y actitudes de los temas anteriores. Ello implica que las evaluaciones diagnósticas no son necesarias y la **Evaluación** debe ser **Continua**, como se explicó ya en las Indicaciones Metodológicas y de Organización, pues los conocimientos previos son imprescindibles para adquirir, comprender y aplicar los nuevos conocimientos, lo que además permite implícitamente evaluar en cada capítulo las capacidades adquiridas en capítulos previos. Adicionalmente, considerando que se adopta el aprendizaje para el dominio (Mastery Learning), se exige al estudiante un aprendizaje a nivel productivo y creativo.

A pesar de lo anterior, cada tema es evaluado y/o autoevaluado individualmente, para que tanto docente como estudiantes puedan juzgar objetivamente si se han alcanzado o no los objetivos propuestos.

Las evaluaciones son prácticas (en computadora, tableta, celular o consola de juego) pues la aplicación de la materia en la vida real es también práctica.

Al estar consignado en el programa de la materia horas de laboratorio, el sistema de evaluación que se adopta es el sistema I, de los aprobados por nuestra Universidad, sin embargo y como se explicó previamente, actualmente no existe separación entre la teoría y la práctica (pues dicha separación sería contraproducente para el proceso formativo). El sistema de ponderación se detalla a continuación:



PROGRAMA DE LA ASIGNATURA

- | | | |
|----------------------|------|--|
| • Exámenes Parciales | 30 % | Representan la Evaluación Parcial. |
| • Practicas | 10 % | Representa la evaluación Frecuente. |
| • Laboratorio | 25 % | Expresa también la evaluación frecuente. |
| • Examen final | 35 % | Representa la Evaluación Final. |

Considerando que, como ya se indico, la evaluacion es continua no se requiere que el estudiante rinda exámenes parciales los clasicos exámenes parciales y final, por lo que la cuantificación de las evaluaciones se realiza como se describe a continuación:

- **La nota del primer parcial es el promedio de las notas de defensa de los capítulos 1 al 5**, que deberá defenderlos y aprobarlos hasta la primera semana del periodo de exámenes de primer parcial de acuerdo al cronograma establecido por nuestra universidad. Los capítulos que no fueran defendidos y aprobados en este periodo, ya no tendrán valor para la nota del primer parcial, pero si para la nota de prácticas y laboratorio.
- **La nota del segundo parcial es el promedio de las notas de defensa de los capítulos 6 al 10**, que deberá defenderlos y aprobarlos hasta la primera semana del periodo de exámenes de segundo parcial de acuerdo al cronograma establecido por nuestra universidad. Los capítulos que no fueran defendidos y aprobados en este periodo, ya no tendrán valor para la nota del segundo parcial, pero si para la nota de prácticas y laboratorio.
- **La nota de prácticas y la nota de laboratorio se calcula de la siguiente manera: El 50 % de la nota es el promedio de las notas de defensa de los capítulos 1 al 11, el otro 50 % de la nota es la nota de defensa del capítulo 12.** Esta forma de cálculo está pensada para que el estudiante estudie, resuelva los ejercicios, defienda y apruebe los 12 capítulos, de otro modo se corre el riesgo de que los últimos capítulos no los estudie, ni los apruebe al alcanzar una nota de aprobación de la asignatura. El estudiante deberá defender estos capítulos hasta la última semana de clases de acuerdo al cronograma establecido por nuestra universidad. Los capítulos que no fueran defendidos y aprobados en este periodo, ya no tendrán valor para la nota de prácticas y laboratorio. Sin embargo, si el estudiante tiene una nota semifinial mayor o igual a 16, como se establece en las normas universitarias, podrá defenderlos y aprobarlos en la fecha fijada para el examen final de la Asignatura.
- **La nota de examen final será la nota de defensa del capítulo 12**, si este es defendido y aprobado hasta la última semana de clases de acuerdo al cronograma establecido por nuestra universidad o en la fecha fijada para el examen final de la Asignatura. Los capítulos que no fueran defendidos hasta esta fecha, y siempre que el estudiante tuviera una nota semifinial mayor o igual a 16, como se establece en las normas universitarias, podrá defenderlos y aprobarlos en la fecha fijada para el examen de segunda instancia de la Asignatura.
- **La nota de examen de segunda instancia será la nota de defensa del capítulo 12**, si este es defendido y aprobado o en la fecha fijada para el examen de segunda instancia de la Asignatura.

En relación a los cursos de temporada, el estudiante deberá defender y aprobar los capítulos que no pudo defender en el semestre regular en la fecha fijada para el examen de dichos cursos.

La nota del curso de temporada se calcula de la siguiente manera: El 50 % de la nota es el promedio de las notas de defensa de los capítulos 1 al 11 y el otro 50 % de la nota es la nota de defensa del capítulo 12.

13. BIBLIOGRAFÍA

BIBLIOGRAFIA PRINCIPAL

1. Lopez, J. (2011). **Domine JavaScript** (3ª Ed.). México: AlfaOmega.
2. Valdivia, D. (2001). **Programación en JavaScript**. Lima, Perú: Macro E.I.R.L.
3. Afergant, M. (1997). **Programación en Web 6 en 1**. México: Prentice Hall Hispanoamericana.
4. Lemay, L (1997). **Aprendiendo HTML 4 para WEB en una semana** (3ª Ed.). México: Prentice Hall Hispanoamericana.

BIBLIOGRAFIA COMPLEMENTARIA O DE APOYO

1. Trostler, M. (2013). **Testable Javascript**. USA: O'Reilly Media.
2. Liang, E. (2010). **JavaScript Testing – Beginner's Guide**. Birmingham, UK: Packt Publishing.
3. Goodman, D. (2001). **JavaScript Bible - Gold Edition**. USA: Wiley.
4. Mullins, C. (2020). **The Great Debate**. Visitado 2020-06.23, <http://www.mullinsconsulting.com/The%20Great%20Debate.pdf>.
5. Deitel, H., P. DEITEL (2013). **Como programar en C/C++** (2ª Ed.). México: Prentice Hall.
6. Stroustrup, B. (2013). **The C++ Programming Language** (4ª Ed.). New York, USA.: Addison-Wesley.
7. Josuttis, N. (2012). **The C++ Standard Library: A tutorial and Reference** (2ª Ed.). New York, USA.: Addison-Wesley.
8. Jacobs, B. (2009). **Object Oriented Programming Oversold!**. Visitado: 2020-06-23, Programming Wisdom Center: <https://www.oocities.org/tablizer/>



PROGRAMA DE LA ASIGNATURA

9. Edmonds, J. (2008). **How to Think About Algorithms**. New York, USA: Cambridge University Press.
10. Joyanes, L. (2008). **Fundamentos de Programación**. Oviedo, España: Mc Graw Hill.
11. Skiena, S. (2008). **The Algorithm Design Manual**. New York, USA: Springer-Verlag.
12. Vine, M. (2008). **C Programming for Absolute Beginner (2ª Ed.)**. EEUU: Thomson.
13. Pressman, R. (2007). **Software Engineering: A Practitioner's Approach (6ª Ed)**. New York, USA: McGraw-Hill.
14. Blanchette, J., M. Summerfield (2006). **C++ Gui Programming with QT4 (3ª Ed.)**. EEUU: Prentice Hall.
15. Teran, J. (2006). **Fundamentos de la Programación**. La Paz, Bolivia: Universidad Mayor de San Andrés, Facultad de Ciencias Puras y Naturales, Carrera de Informática.
16. Arndt, J. (2005). **Algorithms for Programmers** [documento pdf]. Descargado: 2020-06-23. Disponible en: <https://sciarium.com/file/9/>.
17. Cairo, O. (2005). **Metodología de la Programación (3ª Ed.)**. Mexico: Alfaomega.
18. Meyer, S. (2005). **Effective C++: 55 Specific Ways to Improve Your Programs and Designs (3ª Ed.)**. EEUU: Addison-Wesley.
19. Randy, S. (2004). **C++ for Dummies (5ª Ed.)**. Indianapolis, USA: Wiley Publishing Inc.
20. Franek, F. (2004). **Memory as a Programming Concept in C and C++**. Cambridge, UK: Cambridge University Press.
21. Schildt, H. (2003). **C/ C++ Programmers Reference (3ª Ed.)**. Osborne, USA: McGraw-Hill.
22. Gormen, T. et al (2002). **Introduction to Algorithms (2ª Ed.)**. New York, USA: MacGraw-Hill.
23. Pratta, S. (2002). **C++ Primer Plus (4ª Ed.)**. EEUU: Sams Publishing.
24. Smiley, J. (2002). **Learn to Program with C++**. Osborne, USA: McGraw-Hill.
25. Shtern, V. (2000). **Core C++ A Software Engineering Approach**. EEUU: Prentice Hall.
26. Kirch-Prinz, U., P. Prinz (2002). **A Complete Guide to Programming in C++**. Boston, USA: Janes and Bartlett Publishers.
27. Schild, Herbert (2002). **C++ The Complete Reference (4ª Ed.)**. New York, USA: McGraw-Hill.
28. Levi, L. (2001). **Programación Estructurada**. Visitado: 2020-06-23, www.alipso.com: https://www.alipso.com/monografias/programacion_estructurada/.
29. Venugopal, K.; R. Buyya, T. Ravi (1999). **Mastering C++**. New Delhi, India: McGraw-Hill.
30. Finch, L. (1998). **So Much 00, So Little Reuse**. Visitado: 2010-03-01, Dr. Dobb's Web Site: <http://www.ddj.com/oped/1998/finc.htm>.
31. Knuth, D. (1997). **The Art of Computing Programming (3ª Ed.)**, Vol.1: Fundamental Algorithms. Massachusetts, USA: Addison Wesley.
32. Lecca, R. (1997). **Algoritmos en Borland Pascal**. Lima, Perú: Mundigraf.
33. Meyer, B. (1997). **Object-Oriented Software Construction (2ª Ed.)**. Santa Barbará, CA, USA: ESE Inc.
34. Martin, J., J. Odell (1994). **Análisis y Diseño Orientado a Objetos**. Mexico: Prentice Hall.
35. Joyanes, L. (1993). **Programación en Turbo Pascal: versiones 5.5, 6.0 y 7.0**. Madrid, España: Mc Graw Hill.
36. Joyanes, L. (1993). **Turbo Pascal 6.0. A su alcance**. Madrid, España: Mc Graw Hill.
37. Joyanes, L. (1992). **Introducción a la Programación**. Madrid, España: Mc Graw Hill.
38. Schneider, D. (1991). **Microsoft Q Basic: An Introduction to Structured Programming (2ª Ed.)**. New York, USA: Dellen Pub Co.
39. Joyanes, L. (1990). **C++ a su Alcance**. Madrid, España: McGraw-Hill.
40. Alcalde, E., M. Garcia (1992). **Metodología de la Programación**. España: Mc Graw Hill.
41. Knuth, D. (1974). **Structured Programming with go to Statements**. ACM Journals, VI (December 1974 N° 4).
42. Dijkstra, E. (1968). **Go To Statement Considered Harmful**. Communications of the ACM, XI, (March 1968 N° 3):147-148.
43. Cabrera, H (2020). **Programación estructurada**. Visitado: 2020-06-23, www.monografias.com: <https://www.monografias.com/trabajos/progestructu/progestructu.shtml>.

Datos Complementarios

Programa elaborado por primera vez: 07/02/2013

Programa modificado por última vez: 18/02/2026

Apartados actualizados:

Bibliografía:

Contenido Mínimo:

Indicaciones metodológicas y de investigación:

Actividades de investigación y/o interacción:

Firma del docente

Firma del(a) Director(a) de Carrera