



PROGRAMA DE LA ASIGNATURA

1. INFORMACIÓN GENERAL DE UBICACIÓN DE LA ASIGNATURA

Área: Ciencias Tecnológicas y Agrarias	Docente: Peñaranda Villavicencio Hernan Gonzalo
Facultad: CIENCIA Y TECNOLOGIA	
Sistema: Normal Semestralizado	Asignatura: PROGRAMACIÓN INTERMEDIA
Gestión: 1/2025	Sigla: SIS101
Fecha: 14-02-2025	Curso: 2

2. DESCRIPCIÓN Y JUSTIFICACIÓN DE LA ASIGNATURA

La asignatura forma parte de las asignaturas del área de las ciencias de la computación, mismas que son cruciales en la formación de las habilidades con las que debe contar el futuro profesional.

En esta asignatura se refuerzan y complementan los conocimientos y habilidades adquiridas en la materia prerrequisito: SIS101, al mismo tiempo que se dan a conocer y se aplican los principios de la programación orientada a objetos.

Al ser una asignatura de nivel básico, en la misma se resuelven problemas simples que, sin embargo, son de utilidad real en los problemas que se presentan en el ejercicio real de la profesión.

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.

3. RELACIONES DE LA ASIGNATURA

La asignatura es parte de las asignaturas troncales de la Carrera, conjuntamente la materia precedente (programación básica) y las subsiguientes (programación avanzada, programación web, etc.), proporcionan las habilidades y los conocimientos que los estudiantes requieren para la elaboración científica y metodológica de programas.

La asignatura se relaciona también con materias básicas, principalmente aquellas del área de las matemáticas (álgebra, geometría y cálculo): los fundamentos y las relaciones matemáticas se aplican en la resolución de problemas. En este sentido se emplean también los fundamentos y conocimientos de las asignaturas correspondientes a la física básica (física básica 1 y 2).

4. OBJETIVO GENERAL

El estudiante, al concluir la asignatura, será capaz de elaborar programas de computadora, aplicando correcta y eficientemente los fundamentos tanto de la programación orientada a objetos, como de la programación estructurada.

5. OBJETIVOS ESPECÍFICOS

- Construir algoritmos, siguiendo los lineamientos del razonamiento lógico.
- Resolver problemas complejos dividiéndolos en problemas más simples (principio de modularidad).
- Elaborar métodos que eviten la repetición de código y reduzcan las posibles fuentes de error.
- Elaborar métodos y/o funciones que validen la información recibida, de manera que se eviten resultados ilógicos.
- Resolver problemas iterativos, empleando las estructuras y/o procedimientos que mejor se adapten al problema que se está resolviendo.
- Resolver problemas creando una comunidad de objetos que colaboren entre sí.
- Elaborar métodos y/o funciones que operen sobre conjuntos de datos (matrices y objetos literales).
- Elaborar programas que promuevan la reutilización del código, aprovechando los principios de la herencia y composición.

6. CONTENIDOS MÍNIMOS

Tema 1: Fundamentos de la programación orientada a objetos		
Objetivo particular: Familiarizarse con el lenguaje de programación, el ambiente de desarrollo y los fundamentos básicos de la programación orientada a objetos.		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores



PROGRAMA DE LA ASIGNATURA

<p>Conoce las características más relevantes del lenguaje de programación a emplear. Conoce las funciones básicas del o de los ambientes de desarrollo a emplear. Conoce los fundamentos básicos de la programación orientada a objetos. Sabe cómo escribir y evaluar expresiones matemáticas simples. Sabe cómo escribir expresiones matemáticas complejas, empleando él o los objetos más adecuados.</p>	<p>Emplea correctamente él o los ambientes de desarrollo elegidos. Escribe correctamente, expresiones que involucran objetos. Programa y evalúa expresiones matemáticas simples. Programa y evalúa expresiones matemáticas complejas, empleando objetos predefinidos para ese fin.</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>
<p>Contenidos mínimos: Lenguajes de programación a emplear en la asignatura. Software y herramientas para la elaboración y ejecución de programas. Fundamentos básicos de la programación orientada a objetos. Escritura de expresiones matemáticas simples en él o en los lenguajes de programación a emplear. Escritura de expresiones algebraicas complejas, empleando objetos predefinidos. Ejemplos y ejercicios.</p>		

Tema 2: Métodos y propiedades de los objetos

Objetivo particular: Emplear, reemplazar y añadir propiedades y métodos simples, a objetos nuevos y predefinidos.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>Conoce las propiedades de la programación orientada a objetos. Conoce los métodos de la programación orientada a objetos. Sabe como declarar las propiedades estáticas y dinámicas de los objetos. Sabe como declarar métodos estáticos y dinámicos de los objetos. Sabe como declarar métodos getter y setter.</p> <p>Contenido mínimo: Propiedades de los objetos. Métodos de los objetos. Declaración de propiedades estáticas. Declaración de propiedades dinámicas. Declaración de métodos estáticos. Declaración de métodos dinámicos. Métodos getter y setter. Ejemplos y ejercicios.</p>	<p>Crea propiedades estáticas y dinámicas tanto en objetos predefinidos como en objetos nuevos. Crea métodos estáticos y dinámicos tanto en objetos predefinidos como en objetos nuevos. Crea métodos getter y setter. Emplea correctamente los métodos y propiedades creados.</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: Elaboración de métodos que implican una lógica condicional.

Objetivo particular: Elaborar métodos, que resuelven problemas que implican una lógica condicional, empleando los operadores y/o estructuras más adecuadas para el problema que está siendo resuelto.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
--------------------------	------------------------	--------------------



PROGRAMA DE LA ASIGNATURA

<p>Conoce y comprende el uso de los operadores relacionales. Conoce y comprende el uso de los operadores lógicos. Comprende la simbología y lógica de los diagramas de flujo. Comprende la lógica de la estructura if, else-if y else-if-else. Comprende la lógica y uso del comando throw. Comprende la lógica y uso del operador ternario ?. Comprende la lógica y uso de la estructura switch/case. Elabora métodos que resuelven problemas que involucran una lógica condicional.</p> <p>Contenido mínimo: Elaboración de métodos empleando operadores lógicos. Evaluación en corto circuito. Elaboración de métodos empleando la estructura if, if-else y if-else-if Elaboración de métodos empleando el operador ternario ?. Uso del comando Throw. Elaboración de métodos empleando la estructura switch/case. Ejemplos y ejercicios.</p>	<p>Escribe correctamente expresiones relacionales y predice su resultado. Escribe correctamente expresiones lógicas y predice su resultado. Maneja con solvencia las estructuras selectivas: if, if - else, if - else if, el operador "?" y switch. Maneja con solvencia el comando throw. Resuelve correctamente problemas condicionales, elaborando métodos que hacen uso de los operadores y las estructuras más adecuadas.</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: Elaboración de métodos que implican una lógica iterativa.

Objetivo particular:Elaborar métodos, que resuelven problemas iterativos, empleando las estructuras más adecuadas para el problema que está siendo resuelto.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>Conoce y comprende el funcionamiento de las estructuras while, do-while y for. Conoce y comprende el uso de los operadores compuestos. Comprende el uso de los comandos break y continue. Conoce y sabe como emplear/evitar ciclos infinitos. Sabe como tratar los errores recurriendo a la estructura try - catch - finally.</p> <p>Contenido mínimo: Elaboración de métodos empleando las estructuras while, do-while y for. Elaboración de métodos empleando los comandos break/continue. Identificación y construcción de ciclos infinitos. Elaboración de métodos empleando ciclos infinitos. Tratamiento de errores: la estructura try - catch - finally. Ejemplos y ejercicios.</p>	<p>Emplea con solvencia las estructuras iterativas: while, do-while y for. Maneja con solvencia los comandos break y continue. Evita la creación accidental de ciclos infinitos. Crea ciclos infinitos cuando con ello logra soluciones más claras y eficientes. Genera y atrapa errores. Elabora métodos que resuelven correctamente problemas repetitivos.</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: Métodos que involucran una lógica recursiva

Objetivo particular:Resolver eficientemente problemas repetitivos/iterativos, mediante la elaboración de métodos recursivos.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
--------------------------	------------------------	--------------------



PROGRAMA DE LA ASIGNATURA

<p>Conoce y comprende el concepto recursivo. Comprende la lógica recursiva. Sabe como resolver problemas repetitivos, mediante la elaboración de métodos recursivos.. Sabe como validar las soluciones recursivas. Comprende la lógica de las funciones doblemente recursivas. Conoce y comprende las clausuras. Reconoce y programa fórmulas matemáticas recursivas.</p> <p>Contenido mínimo: Concepto de recursividad. La lógica recursiva. La condición de finalización. Validación de soluciones recursiva. Elaboración de métodos recursivos. Procesos doblemente recursivos. Uso de clausuras en la elaboración de métodos recursivos. Fórmulas matemáticas recursivas. Ejemplos y ejercicios.</p>	<p>Explica claramente el concepto y proceso recursivo. Implementa correctamente métodos recursivos. Valida correctamente soluciones recursivas. Elabora métodos doblemente recursivos. Programa métodos recursivos haciendo uso de clausuras. Implementa métodos recursivos que resuelven problemas repetitivos/iterativos.</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: Manejo de listas y matrices dinámicas con clases y objetos.

Objetivo particular:Elaborar métodos, objetos y clases, para resolver problemas que involucran conjuntos de datos ordenados en forma de listas y/o matrices.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>Conoce las listas y matrices dinámicas. Sabe cómo crear y emplear listas y matrices dinámicas. Conoce las propiedades y métodos predefinidos, para el trabajo con listas y matrices. Conoce y comprende la forma en que funciona la estructura for-of. Sabe como resolver problemas que crear métodos que operan sobre matrices.</p> <p>Contenidos Mínimos: Datos estructurados. Declaración y manejo de matrices (Array). Métodos estándar del objeto Array. La estructura for-of. Los métodos y el trabajo con matrices. Ejemplos y ejercicios</p>	<p>Identifica correctamente los datos que corresponden a una lista y/o matriz. Crea, manipula y emplea correctamente listas y matrices. Emplea correctamente los métodos predefinidos para el trabajo con matrices. Emplea correctamente la estructura for-of en la solución de problemas que involucran listas y matrices. Elabora clases, objetos y métodos, que resuelven problemas, que involucran el trabajo con listas y matrices.</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: Métodos, propiedades, clases y objetos para el trabajo con cadenas de caracteres.

Objetivo particular:Resolver problemas que involucran el trabajo con cadenas de texto, empleando clases, objetos y métodos, tanto propios como predefinidos.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
--------------------------	------------------------	--------------------



PROGRAMA DE LA ASIGNATURA

<p>Sabe como crear, recuperar, modificar y emplear cadenas de caracteres. Conoce y comprende las secuencias de escape y plantillas literales. Conoce y sabe crear clases, objetos, propiedades y métodos para el trabajo con cadenas. Conoce las expresiones regulares y su uso en el manejo de cadenas. Conoce los caracteres especiales de límite, clase, grupo, cuantificadores y modificadores de las expresiones regulares.</p> <p>Contenidos Mínimos: Declaración y uso de cadenas. Secuencias de escape. Plantillas literales. Propiedades métodos predefinidos para el trabajo con cadenas. Elaboración de clases, objetos, propiedades y métodos para el trabajo con cadenas. Expresiones regulares. Caracteres especiales de límite, clase, grupo, cuantificadores y banderas de las expresiones regulares. Ejemplos y ejercicios.</p>	<p>Crea, manipula y emplea correctamente cadenas de caracteres. Emplea correctamente los métodos predefinidos para el trabajo con cadenas de caracteres. Elabora clases, objetos, métodos y propiedades para el trabajo con cadenas de caracteres. Emplea expresiones regulares en la búsqueda, reemplazo, selección, validación y manipulación de cadenas de caracteres.</p>	<p>Disciplina y puntualidad. Autosuperación. Autocrítica. Responsabilidad profesional. Ética. Perseverancia.</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.				
1	Fundamentos de la programación orientada a objetos	3	0	2	0	4	10	1	10
2	Métodos y propiedades de los objetos	3	0	4	0	4	12	1	12
3	Elaboración de métodos que implican una lógica condicional.	3	0	4	0	4	12	1	12
4	Elaboración de métodos que implican una lógica iterativa.	3	0	4	0	4	12	1	12
5	Métodos que involucran una lógica recursiva	3	0	4	0	4	12	1	12
6	Manejo de listas y matrices dinámicas con clases y objetos.	3	0	4	0	4	12	1	12
7	Métodos, propiedades, clases y objetos para el trabajo con cadenas de caracteres.	3	0	4	0	4	12	1	12
Total horas semestre									82
Total horas extracurriculares semestre									82

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	21	
1	Fundamentos de la programación orientada a objetos																						



PROGRAMA DE LA ASIGNATURA

2	Métodos y propiedades de los objetos																				
3	Elaboración de métodos que implican una lógica condicional.																				
4	Elaboración de métodos que implican una lógica iterativa.																				
5	Métodos que involucran una lógica recursiva																				
6	Manejo de listas y matrices dinámicas con clases y objetos.																				
7	Métodos, propiedades, clases y objetos para el trabajo con cadenas de caracteres.																				

Actividades

No hay actividades

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

Evaluacion	Grupo	Fecha	Temas
Primer Parcial	1	11-04-2025	- Fundamentos de la programación orientada a objetos - Métodos y propiedades de los objetos - Elaboración de métodos que implican una lógica condicional.
	2	11-04-2025	
	3	11-04-2025	
	4	11-04-2025	
Segundo Parcial	1	30-05-2025	- Elaboración de métodos que implican una lógica iterativa. - Métodos que involucran una lógica recursiva - Manejo de listas y matrices dinámicas con clases y objetos.
	2	30-05-2025	
	3	30-05-2025	
	4	30-05-2025	
Final	1	27-06-2025	- Manejo de listas y matrices dinámicas con clases y objetos. - Métodos, propiedades, clases y objetos para el trabajo con cadenas de caracteres.
	2	27-06-2025	
	3	27-06-2025	
	4	27-06-2025	
Segunda Instancia	1	04-07-2025	- Manejo de listas y matrices dinámicas con clases y objetos. - Métodos, propiedades, clases y objetos para el trabajo con cadenas de caracteres.
	2	04-07-2025	
	3	04-07-2025	
	4	04-07-2025	

9. INDICACIONES METODOLÓGICAS Y DE ORGANIZACIÓN

Como ocurre con la mayoría de las asignaturas en el campo de la ingeniería, **Programación Intermedia**, 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 asimilarla 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, actualmente, las clases se organizan de manera que sean 100% prácticas, pero sin necesidad de llevarlas a cabo 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 se aplican los procesos de *evaluación continua* y *aprendizaje a ritmo individual*.

El material digital se pone a disposición de los estudiantes a través de Internet (en la página [SIS101 - Programación Intermedia \(inghpv.com\)](http://inghpv.com)). Dicho material contiene la teoría, tutoriales, ejemplos (interactivos) y ejercicios (interactivos).

Bajo el denominativo de “ejercicios” se engloban las tareas, cuestionarios y ejercicios propiamente.

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.



PROGRAMA DE LA ASIGNATURA

Al concluir un capítulo (al terminar de resolver satisfactoriamente los ejercicios) el estudiante queda habilitado para rendir una prueba del capítulo: una defensa. La prueba es generada automáticamente por el sistema, con ejercicios y opciones aleatorias y está basada en los ejercicios resueltos en el capítulo, puede ser rendida en cualquiera de los horarios habilitados por el docente y en cualquier dispositivo programable. La defensa es presencial y debe ser autorizada por el docente. La nota mínima de aprobación de la defensa es 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 impedir el copiado digital). En casos excepcionales, cuando la defensa no puede ser tomada de forma presencial, la misma es controlada y autorizada mediante Zoom, Google Meet, Teams, Skype u otra herramienta de video conferencia.

Los estudiantes pueden estudiar el material, resolver los ejercicios y rendir las defensas a su propio ritmo, no existe límite de repeticiones, pero una repetición implica el rehacer el capítulo (para adquirir el dominio requerido en el tema: Mastery Learning). Debido a que la asignatura debe ajustarse al calendario académico, cada capítulo debe ser resuelto, defendido y validado en el plazo (aproximado) de una semana.

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, guía al estudiante en su proceso de formación (Flipped Classroom) y valida que sea el estudiante quien efectivamente está resolviendo los capítulos. En situaciones excepcionales (cuando no son posibles las clases presenciales) el docente responde a las consultas y guía al estudiante empleando un medio de comunicación digital, como WhatsApp, Telegram, Zoom, Google Meet, etc.

En cuanto a la *fuerza de adquisición de los conocimientos*, la metodología empleada se ajusta a los métodos de enseñanza práctica, 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 automatizada de los estudiantes. No obstante, las calificaciones obtenidas en las defensas no pasan a formar parte, automáticamente, de las calificaciones, sino que deben ser validadas previamente, por el docente, mediante una prueba oral/práctica, donde el estudiante debe demostrar que ha sido él quien ha resuelto los capítulos y rendido las defensas.

Como ya se explicó, en lo referente 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. En cuanto a los medios interactivos, se emplea la plataforma de la materia para la interacción asíncrona [SIS101 - Programación Intermedia \(inghqv.com\)](#) y [WhatsApp](#) y/o Telegram, así como Google Meet y/o Zoom para la interacción síncrona.

10. RECURSOS DIDÁCTICOS

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

- Internet.
- Computadoras y/o dispositivos móviles programables.
- Plataforma propia: [SIS101 - Programación Intermedia \(inghqv.com\)](#)
- Visual Studio Code.
- Audacity.
- Camtasia.
- Katex.
- Navegadores Internet (Google Chrome, Microsoft Edge).
- CodeSandbox
- StackBlitz
- Replit
- e-Campus

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 prácticos, en los diferentes temas. Son, sobre todo, trabajos de investigación teóricos, con la implementación práctica correspondiente en 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. De esa manera, al evaluar un capítulo, se evalúan también (de manera implícita) los capítulos previos.



PROGRAMA DE LA ASIGNATURA

Por esa razón, las evaluaciones diagnósticas no son necesarias, pues los conocimientos previos son imprescindibles para adquirir, comprender y aplicar los nuevos conocimientos.

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

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, 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). Además, dado que el proceso de evaluación es continuo, las calificaciones son el promedio ponderado de las calificaciones obtenidas en dichas evaluaciones.

Exámenes Parciales	30% (dos evaluaciones).
Prácticas	10% (trabajos presentados).
Laboratorio	25% (trabajos de investigación y prácticas de laboratorio).
Examen final	35% (una evaluación).

13. BIBLIOGRAFÍA

1. AGARWALD, Sundeep. JavaScript RegExp - An example based guide. www.dbooks.org, 2023, Creative Commons Attribution-NonCommercial-ShareAlike 4.0, International License.
2. LARS SVEKIS, Laurence, PERCIBAL, Rob, VAN PUTEN, Maaïke. JavaScript from Beginner to Professional. Packt Publishing Ltd., Birmingham Mumbai, 2021. ISBN 978-1-80056-252-3.
3. RAUSHMAYER, Axel. JavaScript for impatient Programmers. Exploring.com, 2020, ECMAScript 2020 edition. ISBN: 978-1-09-121009-7.
4. COPEL, Flavio. JavaScript from ES5 to ESNext. www.dbooks.org, 2019, Open publication license.
5. BRAITHWAITE. JavaScript Allongé, the six Edition - Programming from Functions to Classes in ECMAScript 2015, LeanPub, 2019. Creative Commons Attribution-ShareAlike 3.0 license.
6. HARVERBEKE, Marijn. Eloquent JavaScript - A Modern Introduction to Programming. Marijn Haverbeke publisher, 3rd Edition, 2018, Creative Commons Licencia.
7. MEYER, Jeanine. The Essential Guide to HTML5: Using Games to Learn HTML5 and JavaScript. Apress Media, 2018, New York. ISBN-13 (electronic): 978-1-4842-4155-4.
8. GAUCHAT, J. D. El gran libro de HTML5, CSS3 Y JavaScript. Marcombo, 3ra Edición, 2017, Barcelona, España. ISBN: 978-84-267-2463-2.
9. ZAKAS, Nicholas C. Understanding ECMAScript 6 - The Definitive Guide for JavaScript Developers. No Starch Press, Inc., 2016, San Francisco CA. ISBN-10: 1-59327-757-1.
10. SEYDNEJAD, Sasan. Modular Programming with JavaScript. Packt Publishing, 2016, Birmingham, UK. ISBN: 978-1-78588-065-0.
11. DALE, kyran. Data Visualization with Python & JavaScript. O'Reilly Media, Inc. 2016, Sebastopol, California. ISBN: 978-1-491-95643-4.
12. SCOTT, Emit A. Designing scalable JavaScript Applications. Manning Publications, 2016, New York. ISBN: 9781617294174.
13. PRUSTY, Narayan. Modern JavaScript Applications. Packt Publishing, 2016, Birmingham, UK. ISBN: 978-1-78588-144-2.
14. PATRICK, James. JavaScript - A Beginner's Guide to Learning the basics of JavaScript Programming. pdfroom.com/books, 2016, Open publication license.
15. KUNGTA, Krishna. Learn NodeJS in 1 Day. Krisna Kungta publisher, 2016, all right reserved.
16. SIMPSON, Kyle. You Don't Know JS: Async & Performance. O'Reilly Media, Inc., 2015, Sebastopol, CA. ISBN: 9781491904220.
17. CAMERON, Dane. HTML5, JavaScript, and JQuery. John Wiley & Sons, Inc., 2015, Indianapolis. ISBN: 978-1-119-00116-4
18. TIMMS, Simon. Mastering Javascript Desing Patterns. Packt Publishing, 2014, Brimingham, UK. ISBN: 978-1-78398-798-6.
19. NIXON, Robin. Learning PHP, MySQL, JavaScript, CSS & HTML5. O'Reilly Media, Inc., Third Edition, 2014, Sebastopol, CA. ISBN: 9781491949450.
20. TROSTLER, Mark Ethan. Testable JavaScript. O'Rielly, United States of America, 2013. ISBN: 978-1-449-32339-4.
21. OSMANI, ADDY. JavaScript Design Patterns. O'Reilly Media, Inc., 2012. ISBN: 9781449334871.
22. HARRIS, Andy. JavaScript & AJAX for Dummies. Wiley Publishing, Inc., 2010, Indianapois, Indiana. ISBN: 978-0-470-41799-7.
23. GOODMAN, Danny. JavaScript Bible. Hungry Minds, Inc., 7th Edition, 2010, New York. ISBN: 978-0-470-52691-0.
24. EUGENE, Liang Yuxian. JavaScript Testing – Beginner's Guide, U.K.: Packt Publishing Ltd. 2010. 272 p. ISBN: 978-1-849510-00-4.
25. COURINGTON, Rebecca. *Structured Programming* [en línea]. Disponible en: <http://acweb.colum.edu/users/rcourington/Progclass/progclass.html> [consulta: 1 marzo 2010]
26. DIJKSTRA, E. *Go-to statement considered harmful* [en línea]. Communications of the ACM, 1968 XI, 3:147-148. Disponible



PROGRAMA DE LA ASIGNATURA

en: <http://www.cs.utexas.edu/users/EWD> [consulta: 1 marzo 2010]

27. CABRERA, H. *Programación estructurada* [en línea]. Disponible en: <http://www.monografias.com>[consulta: 1 marzo 2010]

28. JACOBS, B. *Object Oriented Programming Oversold* [en línea]. Disponible en <http://www.geocities.com> [consulta: 1 marzo 2010]

29. MEYER, B. *Critique of Object Oriented Software Construction* [en línea]. Disponible en <http://www.geocities.com> [consulta: 1 marzo 2010]

30. FINCH L. *So Much OO, So Little Reuse* [en línea]. Disponible en: <http://www.ddj.com> [consulta: 1 marzo 2010]

31. MULLINS, G. *The Great Debate* [en línea]. Disponible en: <http://www.byte.com> [consulta: 1 marzo 2010]

32. Instituto Técnico de La Paz – México. *Tutorial de desarrollo de algoritmos estructurados* [en línea]. Disponible en: <http://www.itlp.edu.mx> [consulta: 1 marzo 2010]

33. EDMONDS, Jeff. *How to Think About Algorithms*. New York: Cambridge University Press, 2008. ISBN: 978-0-521-615410-8.

34. VINE, Michael. *C Programming for Absolute Beginner*, Second Edition. EEUU: Thomson, 2008. 334 p. ISBN: 978-1-59863-480-8.

35. VEER, Emily Vander. *JavaScript for Dummies*. Wiley Publishing, Inc., 4th Edition, 2005, Indianapolis, Indiana. ISBN: 0-7645-7659-3.

36. SANCHEZ, Lopez Martín. *Programación Estructurada en C* [en línea]. 2002. Disponible en: <http://elmartin.org> [consulta: 1 marzo 2010].

Datos Complementarios

Programa elaborado por primera vez: 30/01/2013

Programa modificado por última vez: 13/02/2025

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