

CARTA AL ESTUDIANTE

Carrera: Ingeniería en Sistemas de Información con grado de Bachillerato y salida lateral de Diplomado en Programación de Aplicaciones Informáticas.

Código de Carrera: ISIN (Ingeniería Sistemas Información)

Nombre del curso	Fundamentos de Informática				
Tipo de curso	Regular				
Código de curso	EIF200				
Nivel y grado	I Nivel Grado: Bachillerato y salida lateral de Diplomado en Programación de Aplicaciones Informáticas				
Periodo lectivo	II Ciclo del 2025				
Modalidad	Presencial				
Naturaleza	Teórico – Práctico				
Créditos:	3				
Horas totales semanales	8				
Horas del Curso*	4 horas contacto (2 teoría / 2 práctica) 6 horas de estudio Independiente				
Horas docentes	4				
Horario del curso	Grupo	NRC	Profesor	Días	Horas
	01	51011	Carolina Gómez Fernández	L-J	10:00 - 11:40
	02	51025	Karol Leitón Arrieta	L-J	13:00 - 14:40
	03	51027	Marcial Hernández Villalobos	M-V	18:00 - 19:40
	04	51028	Carolina Gómez Fernández	M-V	08:00 - 09:40
	05	51029	Marjorie Chavarría Nerio	L-J	18:00 - 19:40
	06	51031	Oscar Benavides Arguello	M-V	10:00 - 11:40
	07	53741	Luis Raúl Rodríguez Oconitrillo	L-J	10:00-11:40
	08	53742	Oscar Chaves Barrantes	M-V	18:00 - 19:40
Horario de atención al estudiantado	Grupos	Profesor	Horas Consulta	Correo	
	01	Carolina Gómez Fernández	L 09:00 a 10:00	carolina.gomez.fernandez@una.cr	
	02	Karol Leitón Arrieta	L 16:00 a 17:00	karol.leiton@gmail.com	

	03	Marcial Hernández Villalobos	L 16:30 a 17: 30	Mihv78@gmail.com
	04	Carolina Gómez Fernández	K 09:00 a 10:00	carolina.gomez.fernandez@una.cr
	05	Marjorie Chavarría Nerio	L y J 19:40 a 20:30	fundamentosprograuna@gmail.com
	06	Oscar Benavides Arguello	K y V 12:00 a 13:00	oscar.benavides.arguello@una.cr
	07	Luis Raúl Rodríguez Oconitrillo	J 13:20pm a 14:30pm	una.luisraul@gmail.com
	08	Oscar Chaves Barrantes	K 20:00	oscar.chaves.barrantes@una.cr
Requisitos	Admisión a carrera			
Correquisitos	Ninguno			
Área disciplinaria	Ingeniería de Software			
Persona Docente	Ver docente por grupo en la sección Horario del Curso .			

* **Horas Presenciales:** tiempo en el que la persona estudiante trabaja en el aula, de manera virtual o mediada por la tecnología, supervisada por el profesorado.

** **Horas estudio independiente:** tiempo en el que la persona estudiante, de manera independiente, se dedica a repasar el material de estudio, hacer tareas, ejercicios extras, laboratorios, etc.

“En esta Universidad nos comprometemos a prevenir, investigar y sancionar el hostigamiento sexual entendido como toda conducta o comportamiento físico, verbal, no verbal, escrito, expreso o implícito, de connotación sexual, no deseado o molesto para quien o quienes lo reciben, reiterado o aislado. Si usted está siendo víctima de hostigamiento diríjase a la Fiscalía de Hostigamiento Sexual de la UNA o llame al teléfono: 2277-3961.”

“La UNA se compromete a construir ambientes universitarios libres de discriminación por razones de edad, etnia, sexo, religión, raza, orientación sexual, estado civil, opinión política, ascendencia nacional, origen social, filiación, discapacidad, afiliación sindical, situación económica o cualquier otra forma análoga de discriminación.”

1. DESCRIPCIÓN DEL CURSO

Este curso introduce al estudiantado en los distintos métodos y técnicas para la resolución computacional de problemas, dando énfasis a su solución algorítmica, dentro de varios paradigmas de programación que aportan al proceso de aprendizaje. Se inicia con una introducción a la sintaxis de un lenguaje de programación, particularmente C++, y a las principales estructuras de la programación algorítmica: secuenciales, condicionales e iterativas. Seguidamente se introduce el Paradigma de Orientación a Objetos (POO), con una profundidad o alcance básico, ya que se le dará continuidad en los siguientes

cursos de la carrera. De manera particular, el curso se enfoca en la resolución de problemas desde la perspectiva algorítmica, por lo que la práctica y el ejercicio constante dan la pauta para el aprendizaje. El tiempo de estudio independiente debe ser utilizado por el estudiantado de la mejor manera posible, para lo cual se propone la realización continuada de laboratorios e ejercicios y se ofrecen espacios de consulta y tutorías.

2. OBJETIVOS

Objetivo general:

Desarrollar soluciones a problemas de programación computacional, desde las perspectivas algorítmicas y de Programación Orientada a Objetos, utilizando estructuras de programación y estructuras de datos básicas; propiciando, además, espacios para que el estudiantado vaya formando una actitud investigativa, reflexiva y autodidacta, así como habilidades para el trabajo colaborativo.

Objetivos específicos: Que el estudiantado sea capaz de:

1. Identificar y explicar los conceptos básicos que soportan la programación de computadores: organización y arquitectura de un computador, elementos de software, lenguajes de programación, compiladores e intérpretes, código fuente y ejecutable, estructuras de programación y de datos básicas.
2. Identificar y aplicar a la solución de problemas el proceso básico de la programación de computadores: diseño (conceptualización del problema y representación en un modelo que lo resuelve), diagramación, codificación, validación, prueba y depuración, así como la documentación.
3. Investigar acerca de problemas de la vida real y sus posibles soluciones computacionales.
4. Resolver problemas de programación, utilizando estructuras de programación y estructuras de datos básicas, e implementar las soluciones en el lenguaje C++, en un entorno integrado de programación (IDE).
5. Analizar situaciones problema que permitan identificar los elementos básicos del POO para diseñar e implementar una clase de objetos que sea, al menos parcialmente, la solución adecuada al problema (no se incluye en el alcance del curso la implementación de relaciones entre clases).
6. Reflexionar sobre su proceso de aprendizaje y su participación efectiva en equipos de trabajo.

Habilidades y destrezas:

El estudiantado al terminar el curso debe ser capaz de:

1. Aplicar sus habilidades de pensamiento computacional y sus capacidades algorítmicas.
2. Resolver problemas simples de programación algorítmica con estructuras de programación secuenciales, condicionales e iterativas, utilizando un IDE como ambiente para escribir, ejecutar, probar y depurar el código.
3. Identificar las clases de objetos del entorno o contexto de un problema.
4. Definir e implementar una clase de objetos, incluyendo sus atributos y comportamientos.

5. Resolver problemas simples de programación con estructuras de datos: arreglos unidimensionales, implementados como colecciones en el POO.
6. Desempeñarse en equipos de trabajo, asumiendo sus roles y responsabilidades, haciendo aportes individuales que contribuyan con el logro de los resultados del equipo y resolviendo los conflictos.

3. CONTENIDOS

Tema I: Conceptos básicos de computación.

- a. Definición y características de paradigmas de programación.
- b. Creación de código, depuración y prueba.

Tema II: Conceptos básicos de programación estructurada.

1. Estructuras básicas para la solución de problemas.
 - a. Identificadores, variables, constantes, operadores de asignación, operadores aritméticos, operadores relacionales, operaciones lógicas, precedencia de los operadores.
 - b. Tipos de datos: int, float, bool, char, string, etc.
 - c. Entrada y salida de datos.
 - d. Función main(), como medio para probar las soluciones.
 - e. Orden de operadores elementales (lógico-matemáticos).
 - f. Repaso conocimientos básicos: operadores aritméticos, prioridad de los operadores, cambios de base numérica, cálculos con figuras geométricas, porcentajes, etc.
 - g. Introducción al uso del IDE: variables, cálculos sencillos, salidas por consola, main(), tipos básicos, cin/cout, uso del debugger.
 - h. Estructuras secuenciales.
 - i. Estructuras condicionales y de control de flujo (if, else, switch).
 - j. Estructuras iterativas (ciclos: while, do-while, for).
 - k. Implementación de funciones y procedimientos. Tipo de datos de retorno, Parámetros de entrada y de salida (parámetros formales) y argumentos (parámetros reales). Paso de parámetros por valor y referencia.

Tema III: Conceptos básicos de Programación Orientada a Objetos (POO).

1. Características fundamentales de un lenguaje orientado a objetos.
2. Definición de una clase como abstracción de un objeto. Proceso de clasificación de objetos: Conceptualización, especificación.
3. Caracterización de objetos: identidad, estado, comportamiento, interfaz de los comportamientos e interfaz de los objetos.
 - a. Representación de clases por medio de un subconjunto de UML. Representación de clases, atributos, métodos, visibilidad.
 - b. Conceptos de abstracción y encapsulamiento.
 - c. Atributos: tipos básicos de datos (con variables de memoria estática, NO punteros).
 - d. Estándares de programación.
 - e. Estado de un objeto.
 - f. Identidad de un objeto.
4. Métodos
 - a. Visibilidad y acceso.
 - b. Constructor:
 - i. Por defecto.
 - ii. Sobrecargado.

- c. Finalización (destructor).
- d. Operaciones para recuperar y modificar el estado de un objeto (sets y gets).
- e. Invocación de las conductas y del cambio de estado de los atributos de un objeto creando instancias en el main().
- f. Métodos de cálculo.

Tema IV: Estructuras de datos (4 semanas).

Arreglos unidimensionales de datos básicos implementados mediante Colecciones

- a. Algoritmos básicos de manipulación de vectores.
- b. Algoritmos semi numéricos (ordenamiento y búsquedas básicos).
- c. Colecciones de objetos.
- d. Manipulación de Colecciones y paso de Colecciones como parámetros por referencia.

4. ESTRATEGIA METODOLÓGICA

Este curso está diseñado para fomentar el aprendizaje activo y el trabajo en equipo como pilares fundamentales para el desarrollo de competencias en programación. A través de actividades prácticas y colaborativas, el estudiantado enfrentará problemas y retos que estimulan el desarrollo de la lógica. Las clases combinan teoría y práctica, incentivando a los estudiantes a investigar, proponer soluciones y compartir conocimientos en grupos de trabajo, fortaleciendo no solo las habilidades técnicas, sino también las capacidades de comunicación, colaboración y autonomía.

El profesorado acompaña a los estudiantes en la resolución de ejercicios tanto en papel como en computadora, garantizando un equilibrio que favorezca la comprensión profunda de conceptos, el desarrollo de algoritmos y la capacidad de análisis y abstracción.

El trabajo en el aula se complementa con prácticas de laboratorio que permiten al estudiantado investigar y proponer soluciones a los problemas planteados. El profesorado promueve en el estudiantado la investigación y la práctica constante.

Se espera que el profesor destine parte de su clase para acompañar al estudiante en la resolución de ejercicios, promover el trabajo en equipo, considerando, además, que la resolución de problemas de programación en papel tiene una fuerte demanda sobre los procesos cognitivos de análisis, descomposición, abstracción, identificación de patrones y desarrollo de algoritmos implicados en la solución computacional de problemas.

El profesorado facilitará el proceso de aprendizaje y:

1. Abordará algunos temas a través de presentaciones/videos. Con estos recursos se desarrolla la problemática desde un enfoque integral, atendiendo a la resolución de problemas, el lenguaje de programación C++ y el paradigma de programación orientado a objetos (POO).
2. Desarrollará prácticas de laboratorio para facilitar en el estudiantado una mejor comprensión de los temas del curso y el desarrollo de las habilidades esperadas.
3. Realizará diversas actividades (trabajo en clase, dinámicas, retos, desafíos o laboratorios) para promover el aprendizaje activo y la asimilación más natural de los conceptos.
4. Promoverá en el estudiantado la investigación constructiva, de tal manera que contribuyan activamente con su propio aprendizaje y el de sus compañeros/as. Se promoverá que el estudiantado construya y comparta sus propias soluciones a los problemas planteados.

5. Usará un entorno de desarrollo integrado (Zinjai) libre y gratuito para programar en C/C++, como una herramienta que permita estandarizar el uso de un lenguaje y reforzar la necesidad de “probar” lo que se programa y hacer uso de las herramientas disponibles para hacerlo. Zinjai ofrece como particular ventaja que es un IDE liviano y exige pocos requerimientos del computador.
6. Utilizará dinámicas de trabajo en grupo para fomentar en el estudiantado el desarrollo de competencias para el trabajo en equipo.
7. Promoverá la capacidad del estudiantado para valorar su propio aprendizaje, ser reflexivo y crítico.
8. Además, hará uso de las siguientes plataformas digitales:
 - **MS TEAMS:** plataforma de comunicación y colaboración que facilita la presencia remota.
 - **Aula Virtual Institucional**, donde se compartirán materiales de apoyo para el curso, tales como presentaciones, tareas, ejercicios, soluciones de referencia, plantillas, guías para el curso, entre otros. Es responsabilidad del estudiantado la revisión periódica del **aula virtual**; así como el mantener actualizado su perfil en dicha plataforma, incluida una fotografía reciente. La dirección del aula virtual institucional es: <https://aulavirtual.una.ac.cr/>

Papel del estudiantado:

Se espera que el estudiante adopte un rol activo y comprometido durante todo el curso. Su participación constante en las actividades y discusiones será fundamental para su aprendizaje. Además, se requiere que mantenga hábitos de estudio responsables, dedicando tiempo regular para la revisión y práctica de los contenidos, con el fin de fortalecer sus conocimientos y habilidades.

El éxito en el curso dependerá en gran medida de su iniciativa para practicar, resolver problemas, investigar y aplicar lo aprendido de manera constante. Durante todas las semanas del ciclo, el estudiante deberá dedicar de forma constante y regular, tiempo de estudio independiente y autónomo para desarrollar practicas asignadas por el profesor. Todas las semanas el profesor guiara el estudio independiente con la asignación de prácticas a realizar en casa, para lo cual se espera que, como mínimo, el estudiante dedique la cantidad de horas semanales de estudio independiente propuestas para este curso.

El estudiantado deberá mantener en todas sus acciones académicas una conducta ética y responsable, de acuerdo con las normas establecidas por la Universidad Nacional.

Dada la naturaleza del curso y sus objetivos, así como su importancia en sentar las bases fundamentales para el resto de la carrera y la vida universitaria, la asistencia a las clases se considera **necesaria** para el logro de los objetivos de aprendizaje y el buen desempeño del estudiante. Por lo tanto, se espera que el estudiante asista puntualmente a clases y, en caso de ausentarse por alguno motivo de fuerza mayor, justifique su ausencia y se actualice en el estudio de los contenidos en el cumplimiento de las actividades asignadas durante la clase.

Cada profesor define la manera en que se llevara el registro y control de la asistencia a, por ejemplo: pasar lista de manera oral, firmar en manuscrito una lista de asistencia, registrar la asistencia por medio del Aula Virtual, entre otros.

Papel de la cátedra:

Los profesores trabajaran de manera colegiada y coordinada esto con el objetivo de estandarizar los objetivos y contenidos del curso, por tanto la carta al estudiante será la misma para todos los grupos así como los exámenes aplicados.

Cada docente utilizará el aula virtual como medio de comunicación, apoyo y consulta. Los materiales de apoyo producidos por la cátedra, la información sobre exámenes, asignación de aulas o cualquier comunicación importante se hará por este medio. Por lo tanto, la coordinación de la carrera recomienda que el Aula Virtual se utilice en todos los cursos.

A nivel de catedra se realizarán algunas actividades extracurriculares que complementan la formación del estudiantado y le acerca a la realidad de la disciplina y la profesión.

5. ESTRATEGIA EVALUATIVA

La evaluación del curso incluye los siguientes rubros, con su correspondiente ponderación para el cálculo de la calificación final:

Rubro de evaluación	Ponderación
Actividades de trabajo en clase	15%
Tareas y/o pruebas cortas (<i>mínimo 4</i>)	20%
Examen #1	20%
Examen #2	20%
Examen #3	25%
Total	100%

Actividades de trabajo en clase

Cada profesor evalúa el trabajo en clase con base en actividades específicas realizadas durante las lecciones, las cuales buscan promover el trabajo en equipo y poner en práctica las habilidades que se espera logren las personas estudiantes.

Las actividades de este rubro **son NO colegiadas**, por lo cual cada profesor diseñara estas actividades de acuerdo a las necesidades específicas del grupo, contemplando así estrategias como: laboratorios, practicas, dinámicas, retos, problemas, ejercicios entre otros.

Las actividades de trabajo en clase propuestas por cada profesor podrán resolverse en una sola sesión o desarrollarse a lo largo de varias lecciones, abarcando diversos temas mediante la expansión gradual de un problema específico.

En la ejecución de estas actividades el estudiante podrá consultar al profesor y a los/las compañeros/as, realizar consultas en notas de clase, aula virtual y sitios de internet, haciendo el máximo esfuerzo por realizar el trabajo asignado. Las actividades de trabajo en clase serán evaluadas por el profesor considerando aspectos como: trabajo en equipo, avance logrado, aprovechamiento de fuentes de información y recursos disponibles, compromiso y esfuerzo demostrados en el trabajo asignado.

El profesor podrá (si lo considera conveniente) solicitar a las personas estudiantes que entreguen en el Aula Virtual los resultados logrados en las actividades de trabajo en clase, sin embargo, estas actividades solamente se evaluarán a estudiantes que asistan a la clase. No se evaluarán entregas de actividades de trabajo en clase de estudiantes que no hayan asistido a la lección correspondiente.

Dado que este rubro evalúa específicamente el **trabajo en clase**, la asistencia a la clase es indispensable para obtener la calificación asignada a estas actividades. Dichas actividades no requerirán ser anunciadas con anticipación por el profesor, dado que es responsabilidad del estudiante asistir a clases y comprometerse con el trabajo que se realice durante las lecciones. En casos de ausencia, por la naturaleza misma del trabajo en clase, la actividad no se podrá reponer.

Tareas y/o pruebas cortas

Las tareas o las pruebas cortas NO son colegiadas, por lo tanto, cada profesor diseñará las actividades que aplicará, al menos 4 durante el ciclo lectivo. Deben ser asignadas o anunciadas al menos con una semana de anticipación a la fecha de entrega o aplicación.

En caso de realizarse más de 4 actividades en este rubro, quedará a criterio de cada profesor la posibilidad de eliminar la actividad con la calificación menor.

Exámenes

Los exámenes se realizan de manera presencial y colegiada, es decir, en todos los grupos del curso se aplican los mismos exámenes, en las mismas fechas y a la misma hora, de acuerdo con la programación incluida en el cronograma del curso, esto con el objetivo de coordinar el trabajo del profesorado de la cátedra y mantener a los grupos lo más alineados posible con el desarrollo de los contenidos y el nivel del curso.

En los exámenes se evalúan los contenidos estudiados y las habilidades esperadas practicadas en las lecciones, por lo tanto, es fundamental que el estudiantado asista y participe activamente en cada una de las lecciones. Por la naturaleza del curso y su temática, los contenidos evaluados en los exámenes son acumulativos.

Los exámenes se resuelven y entregan manuscritos en papel. Por lo tanto, se recomienda que las personas estudiantes resuelvan ejercicios de programación en papel como parte de sus actividades de estudio independiente. Los estudiantes deben llevar a los exámenes cuaderno de examen y lapicero. La respuesta a los exámenes se debe escribir con lapicero de tinta, en caso de entregar la respuesta a los exámenes escrita en lápiz no se aceptarán reclamos posteriores.

Plagio

En todas las actividades de evaluación se deben respetar las normas éticas establecidas en la Universidad. Por lo tanto, si se detecta plagio (la copia parcial o total de texto escrito por parte de otro autor/a, sin hacer referencia explícita del mismo por medio de una cita bibliográfica) o copia se asignará una calificación de cero en la actividad de evaluación correspondiente y se procederá según lo establece el Reglamento General del Proceso de Enseñanza y Aprendizaje de la Universidad Nacional:

“ARTÍCULO 24. PLAGIO.

Se considera plagio la reproducción parcial o total de documentos ajenos presentándolos como propios.”

ARTICULO 24 BIS. EFECTOS ACADÉMICOS Y SANCIONES EN CASO DE PLAGIO COMPROBADO.

“ARTÍCULO 25. COPIA.

Se considera copia todo documento o medio no autorizado, utilizado de manera subrepticia por el estudiante durante una prueba evaluativa.

De comprobarse la copia en la realización de una evaluación, esta será calificada con nota de cero y la persona implicada perderá el porcentaje correspondiente a esa evaluación, independientemente de la eventual sanción disciplinaria establecida en la normativa institucional.”

Examen extraordinario

Podrán presentarla aquellos estudiantes cuya nota final registrada en el acta se encuentre entre **6.0 y 6.75**.

Los estudiantes que realicen examen extraordinario deberán presentarse al mismo con el comprobante de pago de los aranceles correspondientes por parte de la universidad.

6. NORMAS PARA EL DESARROLLO DEL CURSO

- El estudiante debe actualizar su perfil en el aula virtual con la información correspondiente y una foto reciente en la cual fácilmente se reconozca su rostro.
- El medio oficial para asignar y enviar cualquier evaluación es el Aula Virtual. El profesor no recibirá entregas de evaluaciones por medio de correo, MS Teams, WhatsApp (entre otros).
- Al realizar cualquier entrega, el estudiante deberá verificar que el archivo que suba o registre en la actividad correspondiente en el Aula Virtual sea el correcto, pues la calificación de la asignación se hará con base al archivo entregado oportunamente. En caso de subir un archivo equivocado no se aceptarán entregas posteriores.
- Toda actividad de evaluación tendrá una fecha y hora máxima de entrega. No se aceptará ninguna entrega posterior. Se recomienda no realizar entregas cerca del tiempo límite. Si el estudiante así lo hace, será bajo su propio riesgo y responsabilidad. Recuerde que enviar evaluaciones a último momento supone riesgos tales como: fallas en el servicio eléctrico, fallos en la conexión o velocidad de internet o problemas en el equipo utilizado por el estudiante, entre otros.
- Cualquier reclamo sobre la calificación de las evaluaciones deberá hacerse dentro del plazo establecido en el Reglamento General del Proceso de Enseñanza Aprendizaje. Posteriormente a este plazo no se recibirán reclamos. Cualquier reclamo sobre las evaluaciones deberá realizarse en forma presencial o a través de correo electrónico, nunca por WhatsApp o algún otro medio no autorizado.
- En caso de corroborarse algún fraude en la aplicación de alguna evaluación escrita o en la documentación, algoritmos o implementación de las tareas o proyectos, la Escuela de Informática aplicará las sanciones establecidas en la normativa correspondiente de la Universidad Nacional. Todo involucrado en un caso de fraude podrán verse perjudicado, independientemente de su nivel de participación en el mismo.

- En caso de ausencia a una evaluación deberá aplicarse lo estipulado en el artículo 26 del Reglamento General del Proceso de Enseñanza y Aprendizaje de la Universidad Nacional.
- Es responsabilidad del estudiante asistir a clases y tomar nota de los contenidos, observaciones y explicaciones realizadas por el profesor. No necesariamente cada material o recurso utilizado en clase se registrará en el Aula Virtual del curso pues el curso se establece 100% presencial.

7. CRONOGRAMA

Este cronograma es una guía de referencia para la calendarización del estudio de los temas del curso. Cada profesor, en ejercicio de la libertad de cátedra, puede desarrollar los contenidos, organizar las lecciones y realizar las actividades de aprendizaje según su criterio y en beneficio de las necesidades particulares de los grupos a su cargo.

Semana	Desde	Hasta	Tema/Actividad
1	21/07/2025	27/07/2025	Presentación del curso y Conceptos Básicos de Computación Estructura de un programa en C++ (función main(), instrucciones de entrada y salida) Introducción al IDE Zinjai Viernes 25 de julio feriado (Día de la Anexión del Partido de Nicoya a Costa Rica)
2	28/07/2025	03/08/2025	Variables, identificadores en C++, tipos de datos. Operadores y Expresiones
3	04/08/2025	10/08/2025	Funciones (paso de parámetros por valor y por referencia) Buenas prácticas de programación.
4	11/08/2025	17/08/2025	Estructuras condicionales y de control de flujo (if, if-else, switch). Viernes 15 de agosto feriado (Día de la madre)
5	18/08/2025	24/08/2025	Estructuras iterativas (ciclos: while, do-while, for).
6	25/08/2025	31/08/2025	Estructuras iterativas (ciclos: while, do-while, for).
7	01/09/2025	07/09/2025	Recapitulación y Práctica de los contenidos estudiados. Examen 1: Domingo 07 de setiembre, 13:00 pm
8	08/09/2025	14/09/2025	Conceptos básicos de programación orientada a objetos.
9	15/09/2025	22/09/2025	Conceptos básicos de programación orientada a objetos Lunes 15 de setiembre feriado (Día de la Independencia)
10	22/09/2025	28/09/2025	Conceptos básicos de programación orientada a objetos
11	29/09/2025	05/10/2025	Conceptos básicos de programación orientada a objetos
12	06/10/2025	12/10/2025	Recapitulación y Práctica de los contenidos estudiados. Examen 2: Domingo 12 de octubre, 13:00 pm
13	13/10/2025	19/10/2025	Colecciones como arreglos unidimensionales (vectores) de tipo primitivos
14	20/10/2025	26/10/2025	Colecciones como arreglos unidimensionales (vectores) de objetos
15	27/10/2025	02/11/2025	Ordenamiento y Búsqueda en Arreglos Unidimensionales
16	03/11/2025	09/11/2025	Recapitulación y Práctica de los contenidos estudiados. Examen 3: Domingo 9 de noviembre, 13:00 pm
17	10/11/2025	16/11/2025	Entrega de calificaciones Finales

8. RECURSOS BIBLIOGRÁFICOS

Materiales de Cátedra (aportados por la Cátedra):

1. Caamaño Santiago. Fundamentos de informática. 2021
2. Coto. M, Caamaño. S, Mora. S, Vargas. R. Introducción a la programación por objetos. Guía para estudiantes. 2016.

Bibliografía de Consulta

1. Benjumea, V., Roldán, M. (2017). Fundamentos de Programación con el Lenguaje de Programación C++. Universidad de Málaga. Bajo CC., Málaga. Disponible en: http://www.lcc.uma.es/~vicente/docencia/cppdoc/programacion_cxx.pdf
2. Craig, I.D. (2007). Object-Oriented Programming Languages: Interpretation. UTCS, Springer, London. <https://doi.org/10.1007/978-1-84628-774-9>
<https://www.springer.com/gp/book/9781846287732>
3. Hernández-Ruiz, I., Gómez-Fernández, C., Fallas-Carvajal, L.A. (2022) Compilation of Latin American Initiatives and the Regionalization of a University Project to Reduce the Gender Gap in IT, *XIV Congress of Latin American Women in Computing 2022, October 17–21, 2022*, Quindío, Colombia.
4. Hernández-Yáñez, L. (2013). Fundamentos de la programación. Universidad Complutense. Bajo CC., Madrid. Disponible en: <https://www.fdi.ucm.es/profesorado/luis/fp/fp.pdf>
5. Muñoz Caro, C.; Niño Ramos, A.; Vizcaíno Barceló, A. (2010). Introducción a la programación con orientación a objetos. Pearson Educación, S.A., Madrid. Disponible en: [Introducción a la programación con orientación a objetos / C. Muñoz Caro, A. Niño Ramos, A. Vizcaíno Barceló. \(researchgate.net\)](https://www.researchgate.net/publication/266444444-Introduccion-a-la-programacion-con-orientacion-a-objetos)
6. Sage, K. (2019). Getting into Object Oriented Programming. In: Concise Guide to Object-Oriented Programming. Undergraduate Topics in Computer Science. Springer, Cham. https://doi.org/10.1007/978-3-030-13304-7_3 <https://www.springer.com/gp/book/9783030133030>
7. Wing J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 366(1881), 3717–3725. <https://doi.org/10.1098/rsta.2008.0118>