

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE: INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE: Telecomunicaciones

MODALIDAD PRESENCIAL

SÍLABO

FUNDAMENTOS DE PROGRAMACION

NIVEL

SEGUNDO

AMBATO - ECUADOR
2026

I. INFORMACIÓN GENERAL

<table>
<tr><td colspan="3">Nombre de la asignatura
FUNDAMENTOS DE PROGRAMACION
Carrera
Telecomunicaciones</td></tr>
<tr><td>Código UTA-FISEI-T-MP-UB-02-03</td><td colspan="2">PRERREQUISITOS</td></tr>
<tr><td rowspan="2">Modalidad
PRESENCIAL</td><td>Asignatura</td><td>Código</td></tr>
<tr><td>TECNOLOGIAS DE LA INFORMACION (TIC)</td><td>UTA-FISEI-T-MP-UB-01-05</td></tr>
<tr><td colspan="3">Unidad organizacional curricular
BÁSICA</td></tr>
<tr><td rowspan="3">Créditos/Horas
3
Nivel
SEGUNDO</td><td colspan="2">CORREQUISITOS</td></tr>
<tr><td>Asignatura</td><td>Código</td></tr>
<tr><td>BASE DE DATOS</td><td>UTA-FISEI-T-MP-UB-02-04</td></tr>
<tr><td colspan="3">CARGA HORARIA</td></tr>
<tr><td>Componente de aprendizaje en contacto con el docente por semana
3</td><td>Componente de aprendizaje en contacto con el docente por ciclo académico
48</td><td>Componente de aprendizaje práctico experimental, componente de aprendizaje autónomo
96</td></tr>
<tr><td>Horas de tutoría académica semanales
1</td><td>Horas de tutoría presenciales por ciclo académico
16</td><td>Horas de tutorías virtuales por ciclo académico
0</td></tr>
</table>

TOTAL DE HORAS DE APRENDIZAJE EN EL CICLO DE ESTUDIOS

Número de horas del componente de aprendizaje en contacto con el docente semanal	3
Número de horas del componente de aprendizaje en contacto con el docente semestral	48
Número de horas del componente de aprendizaje práctico experimental y componente de aprendizaje autónomo	96
TOTAL DE HORAS AL SEMESTRE (Número de horas del componente de aprendizaje en contacto con el docente semestral + Número de horas del componente de aprendizaje práctico experimental y componente de aprendizaje autónomo.)	144

III DESCRIPCIÓN Y OBJETIVOS DE LA ASIGNATURA

Propósito La asignatura Fundamentos de Programación desarrolla en el estudiante la capacidad de analizar, diseñar e implementar aplicaciones informáticas eficientes, precisas y adaptadas a la solución de problemas relacionados con el manejo y procesamiento de la información. Contribuye al perfil de egreso del Ingeniero en Telecomunicaciones, fortaleciendo sus competencias para el desarrollo de software bajo estándares internacionales de ingeniería, fomentando la innovación, la calidad y la mejora continua.

Descripción de la asignatura Fundamentos de Programación combina teoría y práctica para el diseño, implementación y validación de algoritmos y programas, siguiendo criterios de eficiencia, precisión y buenas prácticas de programación. Se emplean herramientas y entornos de desarrollo actuales para crear soluciones informáticas que optimicen el manejo de datos y la información. La asignatura se estructura en cuatro unidades: fundamentos de programación, estructuras de datos, programación orientada a objetos (POO) y POO avanzada con herencia y polimorfismo. Todo el desarrollo se sustenta en buenas prácticas de ingeniería de software, empleando metodologías consolidadas y criterios de calidad que favorecen la eficiencia, mantenibilidad y escalabilidad de las aplicaciones.

Objetivo general de la asignatura Desarrollar aplicaciones informáticas orientadas a la solución de problemas de manejo de información, aplicando principios de programación, estructuras de datos y metodologías de desarrollo de software con criterios de precisión, eficiencia y calidad, alineados a estándares internacionales.

Objetivos específicos de la asignatura

1.-Diseñar y representar algoritmos de programación lineal usando pseudocódigo y diagramas de flujo para resolver problemas de forma lógica.

2.-Diferenciar y utilizar métodos con y sin parámetros, así como con y sin valor devuelto, para estructurar programas de forma modular y mejorar la organización del código.

3.-Implementar la reutilización de código mediante el uso de paquetes y clases predefinidas del lenguaje de programación, promoviendo la eficiencia y mantenibilidad de las aplicaciones.

4.-Aplicar principios de herencia y polimorfismo junto con el uso de sentencias de control y lenguaje algorítmico, para desarrollar soluciones flexibles y estructuradas orientadas a problemas de ingeniería.

NOTA: Las horas de tutoría académica no se suman, por lo tanto no se incluyen dentro de las horas totales del semestre

IV PROGRAMA DE ESTUDIOS DE LA ASIGNATURA

U.1	Diseñar y representar algoritmos de programación lineal usando pseudocódigo y diagramas de flujo para resolver problemas de forma lógica.			
Unidades temáticas	**Horas Clase / Componente de aprendizaje en contacto con el docente**	**Componente de aprendizaje práctico experimental.**	**Componente de aprendizaje autónomo incluidas las actividades de investigación y vinculación con la sociedad**	**Instrumentos de evaluación**
1.1.- Concepto y características de los algoritmos de programación lineal	2	1	3	Pruebas de Diagnostico EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
1.2.- Variables, constantes y tipos de datos	2	2	2	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
1.3.- Operadores y expresiones	2	2	2	Trabajos o tareas realizadas fuera de clases Trabajos y ejercicios en clases Evaluaciones escritas u orales en actividades practicas
1.4.- Prueba de escritorio aplicada a algoritmos de programación lineal	4	1	7	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
1.5.- Desarrollo de problemas con algoritmos secuenciales	1	2	0	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Trabajos y ejercicios en clases Evaluaciones escritas u orales en actividades practicas
SUBTOTAL HORAS	11	8	14	**TOTAL HORAS:** 33

Resultados de aprendizaje: Aplica algoritmos de programación lineal para resolver problemas básicos mediante el uso de estructuras secuenciales, garantizando soluciones correctas y ordenadas.

Metodología de enseñanza : Aprendizaje Basado en Problemas, Método expositivo
Estrategias educativas : Simulaciones, Actividades académicas desarrolladas en escenarios experimentales o en laboratorios, Resolución de problemas, Talleres, Trabajos y exposiciones
Recursos didácticos : Marcadores, Proyector, Audiovisuales, Internet, Inteligencia Artificial, Tecnología del Aprendizaje y el Conocimiento
Habilidades Blandas : Liderazgo, Trabajo en equipo, Pensamiento crítico, Responsabilidad

IV PROGRAMA DE ESTUDIOS DE LA ASIGNATURA

U.2	Diferenciar y utilizar métodos con y sin parámetros, así como con y sin valor devuelto, para estructurar programas de forma modular y mejorar la organización del código.			
Unidades temáticas	**Horas Clase / Componente de aprendizaje en contacto con el docente**	**Componente de aprendizaje práctico experimental.**	**Componente de aprendizaje autónomo incluidas las actividades de investigación y vinculación con la sociedad**	**Instrumentos de evaluación**
2.1.- Concepto y finalidad de los métodos en programación	2	1	3	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
2.2.- Encapsulamiento y visibilidad de métodos	1	1	1	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
2.3.- Métodos constructores y su relación con y sin parámetros de valor devuelto	1	2	0	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
2.4.- Uso de métodos en la interacción entre clases	1	2	0	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
2.5.- Ejercicios de aplicación	4	2	6	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
SUBTOTAL HORAS	9	8	10	**TOTAL HORAS:** 27

Resultados de aprendizaje: Diseña e implementa métodos con y sin parámetros y con valor devuelto para resolver problemas.

Metodología de enseñanza : Aprendizaje Basado en Problemas, Aprendizaje Basado en Proyectos, Método expositivo
Estrategias educativas : Simulaciones, Actividades académicas desarrolladas en escenarios experimentales o en laboratorios, Talleres, Lectura, análisis y comprensión de materiales bibliográficos y documentales, Trabajos y exposiciones
Recursos didácticos : Marcadores, Proyector, Audiovisuales, Internet, Inteligencia Artificial, Tecnología del Aprendizaje y el Conocimiento
Habilidades Blandas : Liderazgo, Trabajo en equipo, Pensamiento crítico, Responsabilidad

IV PROGRAMA DE ESTUDIOS DE LA ASIGNATURA

U.3	Implementar la reutilización de código mediante el uso de paquetes y clases predefinidas del lenguaje de programación, promoviendo la eficiencia y mantenibilidad de las aplicaciones.			
Unidades temáticas	**Horas Clase / Componente de aprendizaje en contacto con el docente**	**Componente de aprendizaje práctico experimental.**	**Componente de aprendizaje autónomo incluidas las actividades de investigación y vinculación con la sociedad**	**Instrumentos de evaluación**
3.1.- IDE de programación	3	1	5	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
3.2.- Paquetes y librerías: organización del código	1	1	1	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
3.3.- Diseño de soluciones aprovechando reutilización de código	1	2	0	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
3.4.- Uso de clases predefinidas del lenguaje de programación	1	2	0	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
3.5.- Ejercicios de aplicación enfocada a la programación orientada a objetos	4	0	8	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
SUBTOTAL HORAS	10	6	14	**TOTAL HORAS:** 30

Resultados de aprendizaje: Utiliza paquetes y clases predefinidas en la construcción de programas, demostrando reutilización de código y organización adecuada.

Metodología de enseñanza : Aprendizaje Basado en Problemas, Aprendizaje Basado en Proyectos, Método expositivo
Estrategias educativas : Simulaciones, Actividades académicas desarrolladas en escenarios experimentales o en laboratorios, Lectura, análisis y comprensión de materiales bibliográficos y documentales, Trabajos y exposiciones
Recursos didácticos : Marcadores, Proyector, Audiovisuales, Internet, Inteligencia Artificial, Tecnología del Aprendizaje y el Conocimiento
Habilidades Blandas : Liderazgo, Trabajo en equipo, Pensamiento crítico, Responsabilidad

IV PROGRAMA DE ESTUDIOS DE LA ASIGNATURA

U.4	Aplicar principios de herencia y polimorfismo junto con el uso de sentencias de control y lenguaje algorítmico, para desarrollar soluciones flexibles y estructuradas orientadas a problemas de ingeniería.			
Unidades temáticas	**Horas Clase / Componente de aprendizaje en contacto con el docente**	**Componente de aprendizaje práctico experimental.**	**Componente de aprendizaje autónomo incluidas las actividades de investigación y vinculación con la sociedad**	**Instrumentos de evaluación**
4.1.- Lenguaje algorítmico y su relación con la programación orientada a objetos	2	2	2	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
4.2.- Sentencias de control en algoritmos y programas	4	2	6	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
4.3.- Herencia en la programación orientada a objetos	2	2	2	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
4.4.- Polimorfismo y su aplicación en la solución de problemas	10	2	18	EvaluaciÃ³n del desempeÃ±o en escenarios reales o simulados Trabajos o tareas realizadas fuera de clases Evaluaciones escritas u orales en actividades practicas
SUBTOTAL HORAS	18	8	28	**TOTAL HORAS:** 54

Resultados de aprendizaje: Fortalece la P.O.O. utilizando entornos de desarrollo integrados (IDE) que permitan optimizar el proceso de codificación, pruebas y mantenimiento de software.
Metodología de enseñanza : Aprendizaje Basado en Problemas, Aprendizaje Basado en Proyectos, Método expositivo
Estrategias educativas : Simulaciones, Actividades académicas desarrolladas en escenarios experimentales o en laboratorios, Lectura, análisis y comprensión de materiales bibliográficos y documentales, Trabajos y exposiciones
Recursos didácticos : Marcadores, Proyector, Audiovisuales, Internet, Inteligencia Artificial, Tecnología del Aprendizaje y el Conocimiento
Habilidades Blandas : Liderazgo, Trabajo en equipo, Comunicación asertiva, Pensamiento crítico, Flexibilidad, Responsabilidad

V ESCENARIOS DE APRENDIZAJE (REAL, VIRTUAL, ÁULICO)

En concordancia con el modelo educativo de la UTA, la asignatura se desarrolla mediante la aplicaciÃ³n de diversos mÃ©todos, tÃ©cnicas y estrategias que promueven el aprendizaje activo, significativo y colaborativo. El proceso formativo combina trabajo individual y grupal, tomando como referencia enfoques como el Aprendizaje Basado en Problemas (ABP), Aprendizaje Basado en Proyectos (ABPro), talleres prÃ¡cticos, simulaciones y mÃ©todo de casos, todo ello articulado con un sistema de evaluaciÃ³n continua y formativa orientado al logro de los resultados de aprendizaje. Para garantizar un aprendizaje integral y contextualizado, se implementan los siguientes escenarios: Escenario Real: desarrollo de proyectos prÃ¡cticos vinculados a problemÃ¡ticas reales del entorno profesional, permitiendo constatar la aplicaciÃ³n de los conocimientos adquiridos y fomentando la transferencia de competencias a contextos laborales. Escenario Ã¡ulico (Presencial): actividades de enseÃ±anza-aprendizaje desarrolladas en aula, combinando la exposiciÃ³n de contenidos, ejercicios guiados, prÃ¡cticas en laboratorio y dinÃ¡micas grupales que estimulan la participaciÃ³n activa. Escenario Virtual: utilizaciÃ³n de aulas virtuales, simuladores, entornos de programaciÃ³n en lÃnea y recursos TAC (TecnologÃas para el Aprendizaje y el Conocimiento) para facilitar el aprendizaje autÃ³nomo, la interacciÃ³n remota y el trabajo colaborativo.

VI CRITERIOS NORMATIVOS PARA LA EVALUACIÓN

Objetivos específicos	Evaluación diagnóstica (Conocimientos previos)	Evaluación formativa (Grado de logro de destrezas)	Evaluación sumativa (valorar los objetivos generales alcanzados y el logro de destrezas)
1.- Diseñar y representar algoritmos de programación lineal usando pseudocódigo y diagramas de flujo para resolver problemas de forma lógica.	Talleres de Razonamiento	RÚBRICA	Evaluación del desempeño en escenarios reales o simulados Trabajos escritos
2.- Diferenciar y utilizar métodos con y sin parámetros, así como con y sin valor devuelto, para estructurar programas de forma modular y mejorar la organización del código.	Talleres de Razonamiento	RÚBRICA	Evaluación del desempeño en escenarios reales o simulados Trabajos escritos
3.- Implementar la reutilización de código mediante el uso de paquetes y clases predefinidas del lenguaje de programación, promoviendo la eficiencia y mantenibilidad de las aplicaciones.	Talleres de Razonamiento	RÚBRICA	Evaluación del desempeño en escenarios reales o simulados Trabajos escritos
4.- Aplicar principios de herencia y polimorfismo junto con el uso de sentencias de control y lenguaje algorítmico, para desarrollar soluciones flexibles y estructuradas orientadas a problemas de ingeniería.	Talleres de Razonamiento	RÚBRICA	Evaluación del desempeño en escenarios reales o simulados Trabajos escritos

VII BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

Autores	Año	Título	Nro Edición	Editorial	Pais o Ciudad	Nro Ejemplares	Nro Páginas	Tipo
Aristizábal Martínez, Delio A. Quiceno Metaute, Sandra M.	2023	Lógica de programación básica orientada a objetos con ejercicios resueltos	1	Instituto Tecnológico Metropolitano		1	530	Virtual
Código/Ubicación base de datos:								
Comentario:	Conocimiento profundo de la lógica de la programación orientada a objetos con ejercicios prácticos							
URL:	https://0x102pxvp-y-https-elibro-net.itmsp.museknowledge.com/es/lc/uta/titulos/253798							

Autores	Año	Título	Nro Edición	Editorial	Pais o Ciudad	Nro Ejemplares	Nro Páginas	Tipo
Hinojosa Gutiérrez; Ángel Pablo	2016	Python paso a paso	1a edición	Ediciones de la U	Bogotá	2	229 páginas	Físico
Código/Ubicación base de datos:	BFISEI1764a							
Comentario:	El libro Python paso a paso constituye una guía didáctica orientada a quienes se inician en la programación.							
URL:								

Autores	Año	Título	Nro Edición	Editorial	Pais o Ciudad	Nro Ejemplares	Nro Páginas	Tipo
Ramírez Marín, Jairo Hernando	2019	Fundamentos iniciales de lógica de programación I. Algoritmos en PseInt y Python	1	D - Institución Universitaria de Envigado		5	276	Virtual
Código/Ubicación base de datos:								
Comentario:	Fundamental en el desarrollo de la lógica de programación							
URL:	https://0x102pphf-y-https-elibro-net.itmsp.museknowledge.com/es/lc/uta/titulos/226488							

BIBLIOGRAFÍA COMPLEMENTARIA

Autores	Año	Título	Nro Edición	Editorial	Pais/Ciudad	Nro Ejemplares	Nro Páginas	Tipo
Joyanes Aguilar; Luis		Fundamentos de programación:	5a edición			2	713 páginas	Físico
Código/Ubicación base de datos:	BFISEI3179a							
Comentario:	Su enfoque estructurado permite al estudiante comprender los principios lógicos que sustentan la programación, comenzando desde conceptos básicos como algoritmos, diagramas de flujo y seudocódigo, hasta la implementación en un lenguaje formal.							
URL:								