

## SÍLABO

### Fundamentos de Programación

<b>Código</b>	ASUC01312		<b>Carácter</b>	Obligatoria
<b>Prerrequisito</b>	30 créditos aprobados			
<b>Créditos</b>	4			
<b>Horas</b>	<b>Teóricas</b>	2	<b>Prácticas</b>	4
<b>Año académico</b>	2021			

#### **I. Introducción**

---

Fundamentos de Programación es una asignatura obligatoria de facultad, ubicada en el tercer periodo académico de las carreras profesionales de la Facultad de Ingeniería. Tiene como prerrequisito haber aprobado 20 créditos y es prerrequisito de las asignaturas de Estructura de Datos y Programación Orientada a Objetos en la carrera profesional de Ingeniería de Sistemas e Informática y prerrequisito de la asignatura de Estructura de Datos en la carrera profesional de Ingeniería Empresarial. Con esta asignatura se desarrolla en un nivel intermedio la competencia transversal: Conocimientos de Ingeniería. La relevancia de la asignatura reside en elaborar programas para computadora, empleando un lenguaje de programación.

**Los contenidos generales que la asignatura desarrolla son:** conceptos básicos de programación: algoritmo, variable, constante, tipos de datos, operadores; estructura general de un programa; estructuras de control de programación: secuencial, selectivas y repetitivas; módulos de programa: conceptos básicos y tipos, función y procedimiento, funciones del lenguaje de programación; recursividad.

---

#### **II. Resultado de aprendizaje de la asignatura**

---

Al finalizar la asignatura, el estudiante será capaz de implementar programas para computadora con la finalidad de resolver problemas de ingeniería.

---

### III. Organización de los aprendizajes

<b>Unidad 1</b> <b>Estructuras de control para la programación:</b> <b>secuencial y selectiva</b>		Duración en horas	<b>24</b>
<b>Resultado de aprendizaje:</b>	Al finalizar la unidad, el estudiante será capaz de aplicar estructuras de control secuencial y selectivas, en la construcción de programas computacionales usando un lenguaje de programación.		
<b>Ejes temáticos:</b>	1. Algoritmo: definición, características y representación 2. Variables y tipos de datos 3. Introducción a la programación: programas traductores, programación estructurada 4. Estructuras de control para la programación: estructuras de control secuencial 5. Estructuras de control para la programación: estructuras de control selectiva: simple, compuesta y múltiple		

<b>Unidad 2</b> <b>Estructuras de control para la programación: repetitiva</b>		Duración en horas	<b>24</b>
<b>Resultado de aprendizaje:</b>	Al finalizar la unidad, el estudiante será capaz de aplicar estructuras de control repetitivas, en la construcción de programas computacionales usando un lenguaje de programación.		
<b>Ejes temáticos:</b>	1. Estructuras de control para la programación: estructuras de control repetitivas: mientras, hacer-mientras y desde/para 2. Creación de menú de opciones 3. Contador y acumulador		

<b>Unidad 3</b> <b>Módulos para la programación: función y procedimiento</b>		Duración en horas	<b>24</b>
<b>Resultado de aprendizaje:</b>	Al finalizar la unidad, el estudiante será capaz de aplicar módulos de programación y librerías de programación, en la construcción de programas computacionales.		
<b>Ejes temáticos:</b>	1. Modularización de programas: definición y características 2. Paso de parámetros en los módulos de programa 3. Tipos de módulos: función y procedimiento 4. Creación de menú de opciones con módulos de programa 5. Librerías de programación		

<b>Unidad 4</b> <b>Módulos para la programación: funciones propias del lenguaje de programación – recursividad</b>		Duración en horas	<b>24</b>
<b>Resultado de aprendizaje:</b>	Al finalizar la unidad, el estudiante será capaz de aplicar las funciones propias del lenguaje de programación, en la construcción de programas computacionales.		
<b>Ejes temáticos:</b>	1. Funciones propias del lenguaje de programación: definición 2. Recursividad		

#### IV. Metodología

El proceso de aprendizaje se desarrolla siguiendo la secuencia teórico-práctica, planteando situaciones problemáticas de organizaciones reales, donde los estudiantes proponen alternativas de solución, ensayan sus propuestas, debaten y se consolida el saber, y se aplica a situaciones nuevas, para el efecto se utilizará:

##### a. Modalidad presencial:

- Método de aprendizaje cooperativo
- Aprendizaje basado en problemas
- Aula invertida
- Aprendizaje basado en TIC

##### b. Modalidad semipresencial

- Método de aprendizaje cooperativo
- Aprendizaje basado en problemas
- Aprendizaje basado en TIC
- Aula invertida

##### c. Modalidad a distancia

- Método de aprendizaje cooperativo
- Aprendizaje basado en problemas
- Aprendizaje basado en TIC

#### V. Evaluación

##### Modalidad presencial

Rubros	Unidad a evaluar	Fecha	Entregable/Instrumento	Peso parcial	Peso Total
Evaluación de entrada	Prerrequisito	Primera sesión	Evaluación teórico-práctica / <b>Prueba objetiva</b>	0 %	
Consolidado 1 <b>C1</b>	1	Semana 4	Evaluación individual teórico-práctica / <b>Prueba mixta.</b>	70%	20%
	2	Semana 7	Evaluación práctica de modo individual / <b>Prueba de desarrollo</b>		
			Actividades de trabajo autónomo en línea.	30%	
Evaluación parcial <b>EP</b>	1 y 2	Semana 8	Exposición de los programas según los casos propuestos y será de modo grupal. / <b>Rúbrica de evaluación</b>	25%	
Consolidado 2 <b>C2</b>	3	Semana 12	Evaluación práctica de modo individual/ <b>Prueba de desarrollo.</b>	70%	20%
	4	Semana 15	Exposición de las funciones según los casos propuestos: grupal. / <b>Rúbrica de evaluación</b>		
			Actividades de trabajo autónomo en línea.	30%	
Evaluación final <b>EF</b>	Todas las unidades	Semana 16	Exposición de los programas de cómputo con las estructuras de control, módulos de programa, librerías de programación y uso de menús de opciones, según los casos propuestos: grupal/ <b>Rúbrica de evaluación</b>	35%	
Evaluación sustitutoria			No aplica		

**Modalidad a distancia**

Rubros	Unidad a evaluar	Fecha	Entregable/Instrumento	Peso
Evaluación de entrada	Prerrequisito	Primera sesión	Evaluación teórico-práctica / <b>Prueba objetiva</b>	0 %
Consolidado 1 <b>C1</b>	1	Semana 2	Evaluación individual teórico-práctica de modo individual/ <b>Prueba mixta</b>	20 %
Evaluación parcial <b>EP</b>	1 y 2	Semana 4	Evaluación práctica de modo individual/ <b>Prueba de desarrollo</b>	25 %
Consolidado 2 <b>C2</b>	3	Semana 6	Evaluación práctica de modo individual/ <b>Prueba de desarrollo</b>	20 %
Evaluación final <b>EF</b>	Todas las unidades	Semana 8	Evaluación práctica de modo individual/ <b>Prueba de desarrollo</b>	35 %
Evaluación sustitutoria			No aplica	

**Modalidad semipresencial**

Rubros	Unidad a evaluar	Fecha	Entregable/Instrumento	Peso Total
Evaluación de entrada	Prerrequisito	Primera sesión	Evaluación teórico-práctica / <b>Prueba objetiva</b>	0 %
Consolidado 1 <b>C1</b>	1	Semana 1-3	Evaluación individual teórico-práctica / <b>Prueba mixta.</b> Evaluación práctica de modo individual / <b>Prueba de desarrollo</b>	20 %
Evaluación parcial <b>EP</b>	1 y 2	Semana 4	Evaluación práctica de modo individual/ <b>Prueba de desarrollo</b>	25 %
Consolidado 2 <b>C2</b>	3	Semana 5-7	Evaluación práctica de modo individual/ <b>Prueba de desarrollo.</b> Exposición de las funciones según los casos propuestos: grupal. / <b>Rúbrica de evaluación</b>	20 %
Evaluación final <b>EF</b>	Todas las unidades	Semana 8	Exposición de los programas de cómputo con las estructuras de control, módulos de programa, librerías de programación y uso de menús de opciones, según los casos propuestos: grupal/ <b>Rúbrica de evaluación</b>	35 %
Evaluación sustitutoria			No aplica	

**Fórmula para obtener el promedio:**

$$PF = C1 (20\%) + EP (25\%) + C2 (20\%) + EF (35\%)$$

**VI. Bibliografía**

**Básica**

- Joyanes, L. (2013). Fundamentos generales de programación. (1° ed.) México: McGraw-Hill.

**Complementaria:**

- Acera, M. (2017). Curso de programación C/C ++. (4° ed.) Madrid: Anaya Multimedia.
- Priale, L. (2009). Algoritmos y diagramas de flujo aplicaciones en C++. (1° ed.) Perú: Megabyte.