

## PROGRAMA DE ASIGNATURA

FPY1101

Fundamentos de programación

14

Crédito  
s Duoc

5

Crédito  
s SCT

Presencial



## 1. ANTECEDENTES GENERALES

Horas Totales	Docencia Dirigida Presencial	Horas Pedagógicas	Docencia Dirigida No Presencial	Horas Pedagógicas	Prerrequisitos	Horas Cronológicas de Trabajo Autónomo
126	Laboratorio de PC	84	No Requiere	0	No aplica	56



## 2. DESCRIPCIÓN DE LA ASIGNATURA

Capacidad/es Institucionales	0
Capacidad/es y Competencia/s del Perfil de Egreso	Desarrollar software seguro y de calidad, analizando el ciclo de vida de éste, según las características del proyecto, las mejores prácticas y estándares de la industria.

RESULTADO DE APRENDIZAJE	INDICADOR DE LOGRO
RA1 Construye algoritmos, utilizando estrategias de abstracción por medio de pseudocódigos, para dar solución a la problemática planteada en el contexto de negocio y considerando aspectos éticos en el algoritmo.	IL 1.1 Identifica las componentes de un algoritmo (variables, estructuras y expresiones), sus funciones básicas e interacción hardware de sistemas para el funcionamiento del software.
	IL 1.2 Identifica un algoritmo considerando las entradas, procesos y salidas para dar solución a un problema planteado, considerando aspectos éticos en el algoritmo.
	IL 1.3 Utiliza expresiones aritméticas, variables de control, asignación de resultados de expresiones relacionales y lógicas para desarrollar un algoritmo y posterior pseudocódigo.
	IL 1.4 Identifica la funcionalidad de las estructuras condicionales para el desarrollo de un algoritmo en pseudocódigo.
	IL 1.5 Identifica la funcionalidad de las estructuras de repetición para el desarrollo de un algoritmo en pseudocódigo.
	IL 1.6 Representa algoritmos a través de diagramas de flujo y pseudocódigo, para obtener un resultado según el requerimiento.
RA2 Desarrolla programas, utilizando un lenguaje de programación, para dar solución a la problemática planteada en el contexto del negocio.	IL 2.1 Utiliza variables para almacenar diversos tipos de datos en el lenguaje seleccionado.
	IL 2.2 Utiliza expresiones aritméticas, relacionales y lógicas para resolver el o los problemas planteados.
	IL 2.3 Programa estructuras de control y validación de acuerdo con las reglas de negocio planteadas.
	IL 2.4 Programa estructuras de repetición de acuerdo con los requerimientos para dar solución del caso planteado.
	IL 2.5 Utiliza un sistema de control para administrar las versiones código (GitHub) en el desarrollo de una aplicación.
RA3 Construye estructuras de datos temporales, utilizando arreglos para almacenar datos, con el fin de darle persistencia a éstos y trabajar con problemas que necesiten guardar información temporalmente.	IL 3.1 Identifica colecciones y arreglos que permitan el almacenamiento de datos según los requerimientos del problema planteado.
	IL 3.2 Utiliza arreglos y matrices para la inserción, eliminación, modificación y búsqueda de datos temporales para cumplir con los requerimientos del problema planteado.
	IL3.3 Utiliza archivos para lograr la persistencia de los datos de la aplicación según el caso planteado.
RA4 Modulariza programas, mediante funciones con y sin parámetros, para simplificar y darle eficiencia a las soluciones construidas.	IL 4.1 Utiliza librerías de sistema para la optimización del código según el caso planteado.
	IL 4.2 Programa funciones que permitan la reutilización de código según el problema planteado.
	IL 4.3 Integra funciones invocadas desde el programa principal para dar solución al problema planteado.



### 3.SÍNTESIS DE LA RUTA DE APRENDIZAJE

RESULTADOS DE APRENDIZAJE	INDICADOR/ES LOGRO	EXPERIENCIA DE APRENDIZAJE	ACTIVIDADES Y EVALUACIONES	AMBIENTES DE APRENDIZAJE	N° HORAS
RA1	IL1.1 al IL1.6	EA 1 - Fundamentos para la creación de algoritmos	Act 1.1 - Algoritmos y su relación con la vida diaria	Laboratorio PC Avanzado	7
			Act 1.2 - Expresiones y estructuras de entrada y salida	Laboratorio PC Avanzado	7
			Act 1.3 - Tomando decisiones	Laboratorio PC Avanzado	7
			Act 1.4 - Repitiendo tareas	Laboratorio PC Avanzado	7
			Act 1.5 - Consolidando Experiencia 1	Laboratorio PC Avanzado	2
			Ev For 1 - Diagrama de flujo I	Laboratorio PC Avanzado	2
			Ev Parcial 1 - Diagrama de flujo II (30%)	Laboratorio PC Avanzado	3
RA2	IL2.1 al IL2.5	EA 2 - Programación de Aplicaciones con Python	Act 2.1 - Hello World!	Laboratorio PC Avanzado	7
			Act. 2.2. Estructuras de entrada y salida	Laboratorio PC Avanzado	7
			Act 2.3 - Estructuras de decisión en Python	Laboratorio PC Avanzado	7
			Act 2.4 - Estructuras de repetición en Python	Laboratorio PC Avanzado	7
			Act. 2.5. Resolviendo problemas con Python	Laboratorio PC Avanzado	2
			Ev For 2 - Selección y repetición I	Laboratorio PC Avanzado	2
			Ev Parcial 2 - Selección y repetición II (30%)	Laboratorio PC Avanzado	3
RA3Y RA4	IL3.1 al IL4.3	EA 3 - Optimizando el código con Python	Act 3.1 - Colecciones en Python	Laboratorio PC Avanzado	7
			Act 3.2 - Trabajando con diccionarios, set y tuplas	Laboratorio PC Avanzado	7
			Act 3.3. Uso de archivos en Python	Laboratorio PC Avanzado	7
			Act 3.4- Funciones en Python	Laboratorio PC Avanzado	7
			Act 3.5 Desarrollo colaborativo con GitHub y Python	Laboratorio PC Avanzado	14
			Act 3.6. Consolidando conocimientos de la asignatura	Laboratorio PC Avanzado	2

			Ev For 3 - Diseño de aplicaciones con Python I	Laboratorio PC Avanzado	2
			Ev Parcial 3 - Diseño de aplicaciones con Python II(30%)	Laboratorio PC Avanzado	3
			Ev Parcial 4- Skills for all (10%)	Laboratorio PC Avanzado	N/A
RA2 al RA4		Evaluación Final Transversal		Laboratorio PC Avanzando	7



#### 4. EVALUACIONES

EVALUACIÓN	SITUACIÓN EVALUATIVA	DESCRIPCIÓN	PESO DE LA EVALUACIÓN PARCIAL (%)	PONDERACIÓN FINAL
Ev Parcial 1-Diagrama de flujo II	Ejecución práctica sin presentación	Esta primera evaluación parcial, considera el diseño de un diagrama de flujo que represente un algoritmo y posterior desarrollo en pseudocódigo para resolver un problemática del entorno en el ámbito de la innovación. El desarrollo de la evaluación se debe realizar de forma local, sin acceso a internet.	30%	60%
Ev Parcial 2-Selección y repetición II	Ejecución práctica sin presentación	En esta segunda evaluación parcial, considera programar una solución que incluya ciclos de repetición y estructuras de control, en resolver un problemática del entorno en el ámbito de la innovación. El desarrollo de la evaluación se debe realizar de forma local, sin acceso a internet.	30%	
Ev Parcial 3- Diseño de aplicaciones con Python II	Ejecución práctica sin presentación	Esta tercera evaluación parcial, considera la programación de un software que incluya arreglos, matrices e integraciones de funciones en un programa principal para luego subir un control de versiones de GitHub, para resolver un problemática del entorno en el ámbito de la innovación. El desarrollo de la evaluación se debe realizar de forma local, sin acceso a internet.	30%	
Ev Parcial 4-Skills for All	Ejecución Práctica	Esta cuarta evaluación parcial, considera el puntaje obtenido por el estudiante desde la plataforma "Skills for All" de Cisco Networking Academy. Este proceso se inicia desde la experiencia 2 y finaliza en la experiencia 3.	10%	
Evaluación Final Transversal	Ejecución práctica sin presentación	Esta evaluación consiste en resolver un problemática del entorno en el ámbito de la innovación, de forma individual sobre gestión de información a través de estructuras de programación, colecciones (Arreglos, Matrices) y modularizado en funciones independientes que en su llamada desde el programa principal den cobertura a los requerimientos de la situación que se presente. El desarrollo de la evaluación se debe realizar de forma local, sin acceso a internet.		40%



#### 5. BIBLIOGRAFÍA OBLIGATORIA

Domínguez, D. Flores, M. & Rangel, O. (2017). Algoritmos y diagramas de flujo con Raptor. Omega grupo Editor

Robbins, P. (2023). Python Programming for Beginners: The Complete Guide to Mastering Python in 7 Days with Hands-On Exercises - Top Secret Coding Tips to Get an Unfair Advantage and Land Your Dream Job. Independently published.