
 UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS	FORMATO DE SYLLABUS		Código: AA-FR-003	 Sistema Integrado de Gestión
	Macroproceso: Direccionamiento Estratégico		Versión: 01	
	Proceso: Autoevaluación y Acreditación		Fecha de Aprobación: 27/07/2023	

FACULTAD:	Tecnológica		
PROYECTO CURRICULAR:	Tecnología en Electrónica Industrial		CÓDIGO PLAN DE ESTUDIOS:

I. IDENTIFICACIÓN DEL ESPACIO ACADÉMICO

NOMBRE DEL ESPACIO ACADÉMICO: INFORMÁTICA Y ALGORITMOS

Código del espacio académico:	1206	Número de créditos académicos:				2
Distribución horas de trabajo:	HTD	2	HTC	2	HTA	2
Tipo de espacio académico:	Asignatura		Cátedra			

NATURALEZA DEL ESPACIO ACADÉMICO:

Obligatorio Básico	X	Obligatorio Complementario		Electivo Intrínseco		Electivo Extrínseco	
--------------------	---	----------------------------	--	---------------------	--	---------------------	--

CARÁCTER DEL ESPACIO ACADÉMICO:

Teórico		Práctico		Teórico-Práctico	X	Otros:		Cuál: _____
---------	--	----------	--	------------------	---	--------	--	-------------

MODALIDAD DE OFERTA DEL ESPACIO ACADÉMICO:

Presencial	X	Presencial con incorporación de TIC		Virtual		Otros:		Cuál: _____
------------	---	-------------------------------------	--	---------	--	--------	--	-------------

II. SUGERENCIAS DE SABERES Y CONOCIMIENTOS PREVIOS

Se espera que el estudiante tenga conocimientos básicos en operaciones matemáticas, pensamiento lógico y análisis de problemas. No se requiere experiencia previa en programación, pero sí disposición para el trabajo autónomo y colaborativo, así como curiosidad por resolver problemas tecnológicos usando herramientas computacionales.

III. JUSTIFICACIÓN DEL ESPACIO ACADÉMICO

La habilidad para estructurar soluciones mediante algoritmos es esencial para cualquier tecnólogo en electrónica. Esta asignatura brinda las bases del pensamiento computacional utilizando un lenguaje de alto nivel como Python, que hoy en día es ampliamente utilizado en ciencia de datos, electrónica embebida, automatización y desarrollo web. A través del diseño e implementación de algoritmos eficientes, el estudiante fortalecerá habilidades para resolver problemas, estructurar datos, interactuar con sensores o dispositivos, e incluso visualizar información de manera automatizada.

IV. OBJETIVOS DEL ESPACIO ACADÉMICO (GENERAL Y ESPECÍFICOS)

Objetivo General:

Desarrollar competencias en diseño algorítmico, lógica computacional y programación básica usando Python, para resolver problemas aplicados a la electrónica y a los sistemas digitales actuales.

Objetivos Específicos:

Comprender los principios de la informática, la codificación y el pensamiento computacional.
Identificar estructuras algorítmicas para resolver problemas reales.
Implementar programas en Python usando estructuras de control, datos y funciones.
Desarrollar pequeños scripts que automatizen tareas básicas y procesen datos.

V. PROPÓSITOS DE FORMACIÓN Y DE APRENDIZAJE (PFA) DEL ESPACIO ACADÉMICO

Propósitos de Formación Relacionados:

Fortalecer el pensamiento lógico, la creatividad computacional y la solución algorítmica de problemas reales.
Desarrollar habilidades en el uso de Python como herramienta transversal para electrónica, control y análisis de datos.
Promover la capacidad de estructurar código modular, reutilizable y documentado.

Resultados de Aprendizaje:

Diseña algoritmos para resolver problemas aplicados a electrónica, automatización y análisis.
Programa en Python utilizando estructuras de control, listas, funciones y módulos.

VI. CONTENIDOS TEMÁTICOS

<p>Pensamiento computacional y lógica algorítmica (4 semana)</p> <p>Introducción al pensamiento computacional y resolución de problemas.</p> <p>Algoritmos y diagramas de flujo.</p> <p>Conceptos fundamentales de informática.</p> <p>Lenguajes de programación y el ecosistema Python</p> <p>Estructuras básicas de programación en Python (4 semana)</p> <p>Tipos de datos, variables y operadores.</p> <p>Estructuras condicionales (if, elif, else).</p> <p>Estructuras iterativas (while, for).</p> <p>Entrada y salida de datos.</p> <p>Buenas prácticas en programación estructurada</p> <p>Modularidad y estructuras de datos (4 semanas)</p> <p>Funciones: definición, parámetros, retorno.</p> <p>Listas, tuplas, diccionarios y conjuntos.</p> <p>Manipulación de cadenas y conversiones de tipos.</p> <p>Manejo de errores y excepciones.</p> <p>Introducción al paradigma orientado a objetos</p> <p>Automatización, análisis y visualización de datos (4 semanas)</p> <p>Lectura y escritura de archivos</p> <p>Estructuras de control</p> <p>Uso de funciones y procedimientos</p> <p>Matrices y Vectores</p>
VII. ESTRATEGIAS DE ENSEÑANZA QUE FAVORECEN EL APRENDIZAJE
<p>Se usará un enfoque de aprendizaje activo y por proyectos. Las clases combinarán teoría práctica inmediata con ejercicios guiados, retos semanales, trabajo colaborativo y autoevaluación. Se usarán entornos modernos (Google Colab, Jupyter Notebooks, Thonny, VSCode), y se promoverá el uso de recursos abiertos, videotutoriales, plataformas interactivas y foros técnicos.</p>
VIII. EVALUACIÓN
<p>De acuerdo con el estatuto estudiantil vigente (Acuerdo No. 027 de 1993 expedido por el Consejo Superior Universitario y en su Artículo No. 42 y al Artículo No. 3, Literal d) el profesor al presentar el programa presenta una propuesta de evaluación como parte de su propuesta metodológica.</p> <p>Para dar cumplimiento a lo dispuesto en el estatuto estudiantil, los porcentajes por corte se definen como se indica a continuación, con base en las fechas establecidos por el Consejo Académico en el respectivo calendario académico.</p> <p>Primer corte (hasta la semana 8) à 35%</p> <p>Segundo corte (hasta la semana 16) à 35%</p> <p>Proyecto final (hasta la semana 18) à 30%</p> <p>En todo caso, la evaluación será continua e integral, teniendo en cuenta los avances del estudiante en los siguientes aspectos: i) comprensión conceptual (pruebas escritas, talleres); ii) aplicación práctica (laboratorios, informes técnicos); iii) proyecto integrador final (análisis, diseño, montaje y presentación); y iv) participación y trabajo en equipo. Asimismo, se debe valorar el desarrollo de competencias comunicativas, resolución de problemas, uso de instrumentos, pensamiento lógico y creatividad. Las pruebas se concertarán con el grupo y se ajustarán a las fechas establecidas en el respectivo calendario académico.</p>
IX. MEDIOS Y RECURSOS EDUCATIVOS
<p>Para el adecuado desarrollo de este espacio académico, se requiere el uso de medios institucionales y recursos individuales que faciliten los procesos de enseñanza y aprendizaje, tanto en ambientes presenciales como virtuales. Las actividades teóricas se apoyarán en aulas de clase dotadas de medios audiovisuales (tablero, videobeam, sillas) y plataformas virtuales institucionales como Microsoft Teams o Google Meet. Además, será fundamental el acceso a presentaciones digitales, bibliografía especializada, textos base, hojas de datos, artículos técnicos y bibliotecas digitales.</p> <p>En cuanto al trabajo práctico, se utilizarán aulas de laboratorio equipadas con equipos de cómputo en donde se encontrarán el software (Python 3.x, JupyterLab, VSCode, Thonny, Google Colab), las librerías (NumPy, matplotlib, pandas, turtle) y recursos (plataformas como W3Schools, Python Tutor, Real Python). Cada estudiante deberá contar con sus elementos y/o dispositivos básicos necesarios para el desarrollo de las prácticas. En algunos casos, se requerirán sensores, microcontroladores (Arduino, ESP32, etc.) y módulos de comunicación, cuando sean necesarios.</p> <p>Como recursos propios, el estudiante debe disponer de una calculadora científica, conexión estable a internet que la universidad proporciona, un sistema para la toma de apuntes (cuaderno, tablet o computador) y acceso a los materiales de clase. Será responsabilidad del estudiante descargar los insumos digitales y contar con los elementos necesarios que serán especificados previamente en cada práctica o proyecto.</p>
X. PRÁCTICAS ACADÉMICAS - SALIDAS DE CAMPO
<p>Durante el curso se pueden organizar visitas a laboratorios especializados de la universidad para observar la aplicación de principios de programación en la industria. También se promoverá la participación en ferias académicas y encuentros estudiantiles que sean desarrollados en la institución educativa. En todo caso, las salidas estarán orientadas a fortalecer el vínculo entre teoría y realidad industrial.</p>
XI. BIBLIOGRAFÍA
<p>Swiegart, Al. Automatiza tareas aburridas con Python. No Starch Press.</p> <p>Zelle, John. Python Programming: An Introduction to Computer Science. Franklin, Beedle & Associates</p> <p>Lutz, Mark. Learning Python. O'Reilly Media</p> <p>Downey, Allen. Think Python: How to Think Like a Computer Scientist. Green Tea Press</p> <p>W3Schools Python [https://www.w3schools.com/python/]</p> <p>Python Tutor: https://pythontutor.com</p>

XII. SEGUIMIENTO Y ACTUALIZACIÓN DEL SYLLABUS

Fecha revisión por Consejo Curricular:			
Fecha aprobación por Consejo Curricular:		Número de acta:	