



INSTRUMENTO DE EVALUACIÓN
LISTA DE CHEQUEO: Programación Estructurada Algoritmos (Python)
TECNÓLOGO ANÁLISIS Y DESARROLLO DE SOFTWARE

IDENTIFICACIÓN DE LA LISTA DE CHEQUEO			
Programa de Formación:	Tecnólogo Análisis y Desarrollo de Software	Código	228118
Competencia:	Análisis de la especificación de requisitos del software	Código	220501093
Resultado (s) de Aprendizaje Asociado (s):	Desarrollar procesos lógicos a través de la implementación de algoritmos.	Código	220501093 03
Criterios (s) de Evaluación Asociado (s):	<ul style="list-style-type: none">• Soluciona problemas de lógica proposicional incorporando habilidades propias en el como programador.• Crea soluciones a problemas mediante algoritmos que incluyen estructuras secuenciales, condicionales y cíclicas• Crea funciones y procedimientos en la solución de algoritmos para ordenar y simplificar los códigos.• Manipula arreglos en diferentes dimensiones para dar solución a problemas reales.• Administra la información de los usuarios por medio de archivos, permitiendo el ingreso, modificación y eliminación de los datos		
Evidencia(s) de Aprendizaje a evaluar:	Esta lista de chequeo está diseñada para que los instructores que orienten Programación Estructurada Algoritmos (Python), en el tecnólogo en análisis y desarrollo de software, desarrollen los conceptos de conocimientos, principios y procesos en cada resultado de aprendizaje de acuerdo con el trimestre y competencia, con su respectiva intensidad horaria.		
Instructor:	Alvaro Perez Niño		

OBJETIVO
Verificar el desarrollo de conocimientos de conceptos, principios y procesos por resultado de aprendizaje, con su respectiva intensidad horaria de acuerdo con el diseño curricular.
INSTRUCCIONES PARA SU DILIGENCIAMIENTO
Este instrumento está diseñado para verificar el desarrollo de conocimientos de conceptos, principios y procesos el resultado de aprendizaje, la competencia asociada al mismo y el diseño curricular.
De acuerdo a la programación, cada instructor debe desarrollar los conocimientos de conceptos y principios asociados al resultado de aprendizaje, teniendo en cuenta las horas estimadas para el desarrollo de la formación respectiva.
El Instructor debe diligenciar en el campo cumple o no cumple de acuerdo a las temáticas a desarrollar y si tiene observaciones debe registrarlas en el campo observaciones del ítem respectivo.

Trimestre	Resultado de Aprendizaje	Conocimientos de Conceptos y Principios	Horas Estimadas	Cumple		Observaciones
				Si	No	
Primer Trimestre	Desarrollar procesos lógicos a través de la implementación de algoritmos.	Introducción a la Competencia	2			
		Presentación del resultado de aprendizaje				
		Autoevaluación de conocimientos previos mediante ejercicios introductorios				
		Fundamentos de Lógica y Algoritmia	8			
		Fundamentos de lógica proposicional				
		Definición de algoritmo				
		Lenguajes algorítmicos (pseudocódigos) y diagramas de flujo				
		Pruebas de escritorio y/o validación				
		Estructuras Secuenciales	15			
		Gestión de entornos de desarrollo (Crear - Activar)				
		Variables, constantes y convenciones de nomenclatura.				
		Tipos de datos básicos: enteros, flotantes, cadenas y booleanos.				
		Conversión de tipos de datos.				
		Operadores aritméticos.				
		Manejo de entrada del usuario (input()).				
		Salida de datos en consola (print() con formateo).				
		Estructuras de Control de Flujo	25			
		Tipos de Operadores: asignación, comparación y lógicos				
		Sentencias condicionales (if, elif, else).				
		Sentencias de casos (match-case).				
		Ciclos e iteraciones:				
		Bucles for y while.				
		Control de flujo con break, continue y pass.				
		Estructuras de Datos	25			
		Listas, tuplas, arreglos y diccionarios				
		Operaciones y métodos sobre estructuras de datos				
		Funciones y Modularización del Código	20			
		Creación y uso de funciones en Python.				
		Parámetros y valores de retorno.				
		Alcance de variables (local y global).				
		Importación de módulos y reutilización de código.				
		Manejo de Excepciones y Depuración de Código	15			
		Tipos de errores en Python.				
		Uso de try-except para manejar excepciones.				

		Depuración con print() y herramientas de depuración.				
		Gestión de archivos	10			
		Lectura y escritura de archivos (open(), read(), write()).				
TOTAL			120			

CONTROL DEL DOCUMENTO

Autores:

Nombre	Cargo	Área	Fecha
Alvaro Pérez Niño	Instructor	Análisis y Desarrollo de Software	16 Febrero de 2025

SOCIALIZACIÓN Y VERIFICACIÓN LISTA DE CHEQUEO

Nombre	Cargo	Programa de Formación	Etapas	Fecha
	Instructor(a)	Técnico en Programación de Software v2	Socialización	
	Vocero(a)	Técnico en Programación de Software v2	Socialización	
Liliana Ma. Galeano Zea	Instructor(a)	Técnico en Programación de Software v2	Socialización	
	Instructor(a)	Técnico en Programación de Software v2	Verificación	
	Vocero(a)	Técnico en Programación de Software v2	Verificación	
Liliana Ma. Galeano Zea	Instructor(a)	Técnico en Programación de Software v2	Verificación	

CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes al formato y al contenido de la lista de chequeo)

Autores:

Nombre	Cargo	Razón del Cambio	Fecha