



PRINCIPIOS DE PROGRAMACIÓN



CERTIFICACIÓN ZTUDIA EN PRINCIPIOS DE PROGRAMACIÓN.

OBJETIVO DEL CURSO

Al finalizar el curso, el Ztudiante será capaz de aplicar un conocimiento amplio de los principios fundamentales de la programación para diseñar algoritmos, comprender estructuras de control, manipular tipos de datos primitivos y crear arreglos unidimensionales, lo que le permitirá desarrollar soluciones algorítmicas básicas a problemas sencillos y comprender la lógica de programas elementales.

2

1. Introducción a la Programación

- 1.1. Conocer la historia de la programación
- 1.2. Analizar el concepto de programa
- 1.3. Analizar el concepto de lenguaje de programación
- 1.4. Comprender el significado de lenguajes compilados
- 1.5. Comprender el significado de lenguajes interpretados
- 1.6. Conocer las herramientas básicas de programación
- 1.7. Conocer los entornos de desarrollo integrado

2. Diseño de algoritmos

- 2.1. Analizar el concepto de algoritmo
- 2.2. Estudiar las características de un algoritmo
- 2.3. Obtener el algoritmo de un proceso simple
- 2.4. Conocer el pseudocódigo
- 2.5. Representar algoritmos a través de pseudocódigo

3. Diagramas de flujo

- 3.1. Conocer los diagramas de flujo
- 3.2. Aprender la simbología de los diagramas de flujo
- 3.3. Aprender las reglas para la construcción de diagramas de flujo
- 3.4. Representar bifurcaciones en un diagrama de flujo
- 3.5. Representar procesos cíclicos en un diagrama de flujo
- 3.6. Representar algoritmos a través de diagramas de flujo

4. Tipos de datos primitivos

- 4.1. Analizar el tipo de dato Entero
- 4.2. Analizar el tipo de dato Punto Flotante
- 4.3. Analizar el tipo de dato Carácter
- 4.4. Analizar el tipo de dato Cadena
- 4.5. Analizar el tipo de dato Booleano

4.6. Analizar el tipo de dato Nulo

5. Operadores y representación de datos

- 5.1. Estudiar los operadores aritméticos
- 5.2. Estudiar los operadores relacionales
- 5.3. Estudiar los operadores lógicos
- 5.4. Estudiar los operadores de asignación
- 5.5. Declarar variables
- 5.6. Declarar constantes
- 5.7. Comprender qué es una literal
- 5.8. Comprender el significado de sentencias y bloques

6. Operaciones aritméticas

- 6.1. Sumar números
- 6.2. Restar números
- 6.3. Multiplicar números
- 6.4. Dividir números
- 6.5. Elevar números a la potencia n
- 6.6. Obtener la raíz n de un número
- 6.7. Obtener el módulo de una división

7. Estructuras selectivas

- 7.1. Utilizar la estructura secuencial de código
- 7.2. Comprender qué es una estructura selectiva
- 7.3. Utilizar la estructura selectiva simple (IF)
- 7.4. Utilizar la estructura selectiva doble (IF - ELSE)
- 7.5. Utilizar la estructura selectiva múltiple (IF - ELSEIF - ELSE)
- 7.6. Utilizar estructuras selectivas anidadas
- 7.7. Utilizar el operador ternario

8. Estructuras iterativas

- 8.1. Utilizar la estructura iterativa controlada por contador (FOR)
- 8.2. Utilizar la estructura iterativa controlada por condición (WHILE)
- 8.3. Utilizar la estructura iterativa controlada por condición (DO - WHILE)
- 8.4. Utilizar estructuras iterativas anidadas
- 8.5. Comprender las sentencias BREAK y CONTINUE
- 8.6. Interrumpir las estructuras iterativas con BREAK
- 8.7. Saltar un ciclo en estructuras iterativas con CONTINUE

9. Algoritmos clásicos y técnicas de resolución

- 9.1. Analizar el algoritmo de búsqueda lineal
- 9.2. Analizar el algoritmo de búsqueda binaria
- 9.3. Analizar el algoritmo de ordenamiento Bubble Sort
- 9.4. Analizar el algoritmo de ordenamiento Quick Sort
- 9.5. Analizar el algoritmo de ordenamiento Merge Sort
- 9.6. Analizar los algoritmos de fuerza bruta
- 9.7. Analizar los algoritmos recursivos

10. Manipulación de datos con arreglos unidimensionales

- 10.1. Declarar arreglos unidimensionales
- 10.2. Inicializar arreglos unidimensionales
- 10.3. Almacenar elementos en un arreglo unidimensional
- 10.4. Acceder a los elementos de un arreglo unidimensional
- 10.5. Recorrer los elementos de un arreglo unidimensional
- 10.6. Ordenar los elementos de un arreglo unidimensional

11. Manipulación de datos con matrices

- 11.1. Declarar matrices
- 11.2. Inicializar matrices
- 11.3. Almacenar elementos en matrices
- 11.4. Acceder a los elementos de una matriz
- 11.5. Recorrer los elementos de una matriz
- 11.6. Ordenar los elementos de una matriz

12. Diseño de métodos y funciones

- 12.1. Crear un método
- 12.2. Utilizar el paso de parámetros por valor
- 12.3. Utilizar el paso de parámetros por referencia
- 12.4. Crear una función
- 12.5. Devolver un resultado por parámetro de referencia
- 12.6. Devolver un resultado por retorno
- 12.7. Devolver un resultado por devolución implícita

13. Uso de métodos y funciones

- 13.1. Llamar a un método
- 13.2. Llamar a una función
- 13.3. Asignación de métodos a variables

- 13.4. Asignación de funciones a variables
- 13.5. Comprender qué es la recursividad
- 13.6. Llamar a una función recursivamente

14. Tipos de datos personalizados

- 14.1. Crear un tipo de dato personalizado
- 14.2. Definir los atributos de un tipo de dato personalizado
- 14.3. Asignar valores a los atributos de un tipo de dato personalizado
- 14.4. Obtener los atributos de un tipo de dato personalizado
- 14.5. Inicializar un arreglo para tipos de datos personalizados
- 14.6. Acceder a los elementos de un arreglo con tipos de datos personalizados

15. Aprendizaje de las estructuras de datos

- 15.1. Conocer las listas enlazadas simples
- 15.2. Conocer listas enlazadas dobles
- 15.3. Conocer pilas (LIFO / UEPS)
- 15.4. Conocer colas (FIFO / PEPS)
- 15.5. Conocer listas enlazadas circulares
- 15.6. Conocer colas circulares
- 15.7. Conocer árboles binarios

16. Técnicas de prevención, detección y corrección de errores

- 16.1. Detectar errores sintácticos
- 16.2. Detectar errores semánticos
- 16.3. Detectar errores de ejecución
- 16.4. Inspeccionar variables
- 16.5. Inspeccionar puntos de interrupción
- 16.6. Insertar mensajes de depuración

17. Buenas prácticas de programación

- 17.1. Dividir un problema complejo en subproblemas
- 17.2. Reutilizar código
- 17.3. Utilizar comentarios para documentar el código
- 17.4. Mostrar mensajes
- 17.5. Convertir datos de tipo texto a número
- 17.6. Convertir datos de tipo numérico a otro tipo de dato numérico
- 17.7. Refactorizar código