Anexo 3

DENOMINACIÓN DE LA ASIGNATURA O UNIDAD DE APRENDIZAJE
PROGRAMACIÓN Y ALGORITMOS

"No aplica"
CICLO ESCOLAR

SC1221
CLAVE DE LA ASIGNATURA

FINES DE APRENDIZAJE O FORMACIÓN
Al finalizar la asignatura, el alumno será capaz de analizar e identificar la forma en que un problema puede ser resuelto por medio de la computadora, definiendo las especificaciones y diseñando la solución mediante algoritmos, diagramas de flujo y pseudocódigo, así como utilizando un lenguaje de programación de alto nivel para su implementación en metodología, estatutos, expresiones, declaraciones, entrada y salida de datos, estructuras de control y arreglos, con la finalidad de que le permita posteriormente el desarrollo de aplicaciones para pc, celulares y microprocesadores.

CONTENIDO TEMÁTICO
1. **Metodología para la resolución de problemas**. 1.1. ¿Qué es un problema? 1.2. Tipos de problemas 1.3. Fases en la resolución de un problema **2. Programación y los conceptos** 2.1. Definición de un programa computacional 2.2. Concepto de un lenguaje de programación maquinal y de alto nivel 2.3. Traductores de lenguaje 2.3.1. interpretadores y compiladores **3. Fases para desarrollar un programa** 3.1. Análisis del problema 3.1.1. Entradas

3.1.2. Salidas
3.2. Diseño de la solución
3.3. Codificación
3.4. Ejecución, pruebas y depuración
3.4.1. Errores semánticos
3.4.2. Errores de sintaxis
3.4.3. Errores de lógica
3.4.4. Errores de ejecución
3.4.5. Implantación

4. Diseño de la solución

4.1. Algoritmo
4.1.1. Definición
4.1.2. Formas de representar la solución
4.1.3. Pseudocódigo
4.1.4. Diagrama de flujo
4.2. Prueba de escritorio
4.2.1. Ejemplos de herramientas a utilizar en el diseño de programas
4.2.1.1. PSeInt
4.2.1.2. Raptor
4.2.1.3. Otros

5. Estructuras de control a través de pseudocódigo y diagramas de flujo

5.1. ¿Qué es una estructura de control?
5.2. Variables y constantes
5.3. Estructura secuencial
5.4. Estructura selectiva
5.4.1. Simple
5.4.2. Doble
5.4.3. Múltiple
5.4.4. Anidamientos
5.5. Estructura cíclica
5.5.1. Repetición controlada-Desde
5.5.2. Mientras-Hacer
5.6. Control de acceso y seguridad en el programa
5.6.1. Tipos de privilegios
5.6.2. Tamaño y caracteres utilizados para el password
5.7. Integridad y validez de datos.
5.7.1. Validación de tipo de datos

5.7.2. Rango de datos
5.7.3. Restricciones

6. Lenguaje y herramientas para la construcción de programas
6.1. Medio ambiente de desarrollo integrado (IDE)
6.2. Lenguaje de programación de alto nivel
6.2.1. Historia
6.2.2. Características

7. Iniciando en el lenguaje: variables, expresiones y declaraciones
7.1. Variable y constantes
7.2. Tipos de datos
7.3. Asignación de valores
7.3.1. Asignación múltiple
7.4. Nombre de identificadores y sus reglas
7.5. Palabras reservadas
7.6. Sentencias o instrucciones en el lenguaje

8. Manejo de operadores
8.1. Aritméticos
8.1.1. Jerarquía
8.1.2. Operadores de atajo
8.2. Relación
8.3. Lógicos o booleanos
8.4. Jerarquía

9. Manejo de comentarios
9.1. Comentarios cortos
9.2. Comentarios largos
9.3. Estándares
9.3.1. Indentación
9.3.2. Uppercase: constantes
9.3.3. CamelCase
9.3.3.1. Upper: nombre de la clase
9.3.3.2. Lower Camel Case: variables, funciones
9.3.3.3. Snake Case, uso de (_)

10. Entrada y salida
10.1. Entrada
10.1.1. Conversión de datos de string a tipos de datos nativos

10.2. Salida
10.2.1. Uso de strings

11. Estructuras de control
11.1. *If else*
11.2. *For*
11.3. *While*

12. **Librería matemática**
12.1. Constantes pi y e
12.2. Raíz cuadrada
12.3. Redondeo
12.4. Potencias
12.5. Números aleatorios

ACTIVIDADES DE APRENDIZAJE BAJO CONDUCCIÓN DE UN ACADÉMICO

1. Solucionar problemas aplicando los conceptos explicados en clase
2. Analizar los diversos problemas mostrando el detalle de las entradas y salidas.
3. Realizar el diseño de la solución de problemas identificando las variables, constantes y sus tipos, así como el desarrollo de pseudocódigo y diagramas de flujo para cada una de las estructuras de control, considerando los aspectos de seguridad y validación de datos.
4. Codificar las soluciones propuestas en el lenguaje de alto nivel seleccionado para la clase, incluyendo el análisis, diseño y prueba de escritorio correspondientes

ACTIVIDADES DE APRENDIZAJE INDEPENDIENTES

1. Realizar investigaciones de lo siguiente:
   - El significado de cada una de las fases del desarrollo de programas.
   - Las palabras utilizadas en la construcción de pseudocódigo y la simbología y su significado para la construcción de diagramas de flujo.

- Lo que representa un programa computacional y el uso de lenguajes de programación como Python, Java, C++, para su desarrollo.
- Lo que representa una prueba de escritorio y como esta debe ser elaborada.
- La evolución de un lenguaje de programación describiendo su historia, características y usos en la actualidad.
- Las diferentes librerías que pueden utilizarse en el programa para trabajar con ambiente gráfico.
- Ejemplos de programas con los diferentes tipos de comentarios y el uso de la indentación.

2. Desarrollar programas
   - Para practicar la estructura secuencial involucrando las instrucciones de entrada y salida.
   - Para practicar las estructuras condicionales y cíclicas involucrando las instrucciones de necesarias para resolver problemas.
   - Para utilizar las funciones matemáticas necesarias para resolver problemas.

3. Resolver problemas, diseñando la solución, seleccionando y aplicando las estructuras de control apropiadas.
   - Se deberá de contemplar el desarrollo de pseudocódigo y diagramas de flujo para cada una de las estructuras de control, considerando los aspectos de seguridad y validación de datos.

CRITERIOS DE EVALUACIÓN	
Investigaciones y/o tareas	15%
Prácticas y exámenes rápidos	15%
Exámenes parciales (2)	30%
Proyecto final	15%
Examen final	25%
Total	100%