

# Universidad Autónoma de Sinaloa

## Facultad de Informática Culiacán

Guía para examen

### De respuesta a cada uno de los siguientes cuestionamientos:

1. Defina el concepto de Lenguaje de programación (LP) y cite al menos 5 LP más utilizados.
2. Hablando en términos de Lenguajes de Programación (LP), ¿Cuál es la ventaja de estudiar más de un LP diferente?
3. Explique cada una de las cualidades o características más importantes que deben tener un buen Lenguaje de Programación.
4. Explique qué diferencia existen entre los lenguajes que son Compilados e Interpretados y cite ejemplos de lenguajes en cada clasificación.
5. Defina al menos 5 elementos sintácticos básicos que conforman a un lenguaje.
6. Explique al menos 4 tipos de aplicaciones que pueden desarrollarse, utilizando como LP a C#.
7. Explique el Modelo de compilación y cite ejemplos de lenguajes que utilizan un compilador.
8. Hablando en términos de edición, compilación y depuración de códigos fuentes, Mencione que herramientas se pueden utilizar para desarrollar programas según el LP.
9. Defina los lenguajes que son Declarativos, Imperativos y Orientados a Objetos y cite ejemplos en cada una de las clasificaciones.
10. Explique cada etapa del proceso de traducción de un programa fuente (Analizador léxico, sintáctico y semántico).
11. Describa cual es la función del analizador léxico, sintáctico y semántico, en el análisis de un programa fuente.
12. Describa los aspectos más importantes sobre el Lenguaje de programación (LP) C#.
13. Año de creación y Creador del lenguaje C#.
14. Características del Lenguaje C#.
15. Aplicaciones que pueden crearse o desarrollarse con los lenguajes C#
16. Descripción de los componentes utilizados para la Edición, Compilación e interpretaciones según el lenguaje, de un programa fuente (Todas las etapas que atraviesa un programa fuente hasta ejecutarse en una PC).
17. Historia de Versiones del lenguaje C#
18. Arquitectura de la plataforma .Net Framework
19. Explicación del Modelo de Ejecución del CLR para C#
20. Herramientas utilizadas para la edición y compilación de programas fuentes tanto para C#  
Describe la estructura de un programa en C#
21. Como se agregan referencias de clases/bibliotecas a un programa fuente en C#
22. ¿Cuáles son los Comandos utilizados para operaciones de Entrada/Salida (IO) en C#?
23. ¿Cuáles son los formatos más comunes para salidas en C#?
24. ¿Cómo se comenta en C# en un programa fuente?
25. ¿Cómo se aplica (sintaxis) y cuál es la intención del tratamiento de Excepciones en C#?

26. ¿Cuál es el procedimiento de compilación, ejecución y depuración por medio de comandos para un programa fuente en C#?
27. Tipos de Datos más comunes del lenguaje C#
28. Listado de Operadores del lenguaje C#
29. Definición de constantes y variables en el lenguaje C#.
30. Valores por defecto al definir variables en ambos lenguajes.
31. Conversión de tipos de datos en ambos lenguajes
32. Lectura de datos de todos los tipos de datos en lenguaje c#.
33. Sintaxis de las estructuras de selección y repetición en c# (Ejemplos de aplicación en problemas según la estructura).
34. Definición de arreglos de tamaño constante o variable según la decisión del usuario en c# tanto para unidimensionales como bidimensionales (ejercicios de cada caso).
35. Métodos existentes en la clase arreglos en el lenguaje C#, que nos permiten ordenar, conocer las dimensiones de las longitudes, entre otros.
36. Sintaxis de declaración en c#.

## Codificación de algoritmos

Elabore el programa correspondiente a cada uno de los siguientes problemas, utilizando el lenguaje C#.

1. Dado el siguiente pseudocódigo, codifíquelo en el lenguaje C# y Java

Objetivo: Imprimir en pantalla, si un número dado por el usuario es: positivo, negativo o nulo.

Programador: MC. Gálvez Gámez Gerardo

Fecha: \_\_/octubre/2012

INICIO

    //Definición de Variables y Constantes

CADENA TipoNumero

ENTERO Numero

    //Lectura de Datos no Conocidos

IMPRIMIR "Proporcione el valor para número:"

LEER Numero

SI Numero1 > Numero2 ENTONCES

    SI Numero1 > Numero3 ENTONCES

        NumeroMayor = "Número 1 es Mayor"

    SI\_NO

        NumeroMayor = "Número 3 es Mayor"

    FIN\_SI

SI\_NO

    SI Numero1 > Numero3 ENTONCES

        NumeroMayor = "Número 1 es Mayor"

    SI\_NO

        NumeroMayor = "Número 2 es Mayor"

    FIN\_SI

FIN\_SI

    //Imprimir como resultado el tipo de número

IMPRIMIR "El Número ", Numero, " es: ", TipoNumero

FIN

**Nota:** Este algoritmo presenta una falla? Debe encontrarlas y solucionarlos.

2. Elabore un programa en C# y otro en Java, tal que dado un número entero proporcionado por el usuario calcule de ese número, haciéndolo sólo por sumas. El cuadrado de un número N es la suma de los N primeros números impares. Debe asegurarse de que el número sea positivo. (use la programación modular).

Por ejemplos:

Sea el número  $3^2 = 1 + 3 + 5 = 9$

Sea el número  $9^2 = 1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 + 17 = 81$

3. Elabore un programa en C# que solicite al usuario los datos para dos arreglos Bidimensionales de N Renglones X N Columnas cada uno, obtenga e imprima la suma (use la programación modular).