



# UNIVERSIDAD DE GUADALAJARA

Centro Universitario de Ciencias Exactas e Ingenierías  
División de Ciencias Básicas  
LICENCIATURA EN MATEMÁTICAS

## 1. INFORMACIÓN DEL CURSO:

<b>Nombre:</b> Programación para ciencias		<b>Número de créditos:</b> 5 (cinco)	
<b>Departamento:</b> Matemáticas		<b>Horas teoría:</b> 0 (cero)	<b>Horas práctica:</b> 68 (sesenta y ocho)
		<b>Total de horas por cada semestre:</b> 68 (sesenta y ocho)	
<b>Tipo:</b> Taller	<b>Prerrequisitos:</b> No tiene prerrequisitos	<b>Nivel:</b> Básica común, se recomienda en el 2 <sup>do</sup> semestre	

## 2. DESCRIPCIÓN

### Objetivo General:

Conocer y aplicar los elementos básicos de algoritmia en diversos algoritmos de complejidad moderada. Conocer y diferenciar las fases en el diseño de un programa. Conocer la estructura general de un programa. Diseñar y codificar algoritmos de complejidad moderada en el lenguaje de programación Fortran. Utilizar los elementos básicos de la programación estructurada en el diseño de programas en Fortran. Compilar y ejecutar los programas diseñados en Fortran en el compilador *GFortan*. Diseñar, codificar y ejecutar algoritmos en el software libre para cálculo numérico *Octave*.

### Contenido temático sintético (que se abordará en el desarrollo del programa y su estructura conceptual)

<p><b>Encuadre (2 horas)</b></p> <p><b>Unidad 1. Introducción a la algoritmia (3 horas)</b></p> <p>1.1 Introducción</p> <p>1.2 Algoritmos: generalidades</p> <p>1.3 Definición común de algoritmo</p> <p>1.3.1 Ejemplo de un algoritmo conocido</p> <p>1.4 Definición formal de algoritmo</p> <p>1.5 Complejidad de un algoritmo: Tiempo y Espacio</p> <p>1.5.1 Peor caso y caso probabilístico</p> <p>1.6 Análisis asintótico de funciones</p> <p>1.6.1 Relaciones de dominación</p> <p>1.6.2 Notaciones asintóticas</p> <p>1.6.3 Funciones de complejidad de tiempo más usuales</p> <p>1.7 Análisis de diversos algoritmos</p> <p>1.7.1 Reglas generales para el cálculo de <math>T(n)</math></p> <p>1.7.2 ¿Cómo obtener <math>T(n)</math>?</p> <p>1.7.3 Análisis asintótico de las funciones obtenidas</p> <p>1.7.4 Completando el análisis asintótico</p> <p>1.8 Actividades a realizar</p> <p><b>Unidad 2. Diseño de programas (3 horas)</b></p> <p>2.1 Introducción</p> <p>2.2 Fases para el diseño de un programa</p> <p>2.2.1 Análisis</p> <p>2.2.2 Programación</p> <p>2.2.3 Codificación</p> <p>2.3 Herramientas de programación</p> <p>2.3.1 Reglas para la creación de diagramas de flujo</p> <p>2.4 Ejemplo del diseño de un programa en <i>FORTTRAN</i></p> <p>2.5 Fases para la puesta a punto del programa (depuración)</p> <p>2.5.1 Edición</p> <p>2.5.2 Compilación</p>
---

- 2.5.3 Montaje
- 2.5.4 Ejecución
- 2.6 Características de un programa
  - 2.6.1 Características generales de un programa
- 2.7 Programación estructurada
  - 2.7.1 Definición de las estructuras básicas
- 2.8 Objetos de un programa: constantes y variables
  - 2.8.1 Atributos de un objeto
  - 2.8.2 Constantes
  - 2.8.3 Variables
- 2.9 Expresiones
  - 2.9.1 Tipos de expresiones
  - 2.9.2 Operadores
  - 2.9.3 Tabla de verdad de los operadores lógicos
  - 2.9.4 Orden de evaluación de los operadores
- 2.10 Tipo de errores de cálculo
  - 2.10.1 Cifras significativas
  - 2.10.2 Exactitud y precisión
  - 2.10.3 Error
  - 2.10.4 Error de redondeo
  - 2.10.5 Error de truncamiento
  - 2.10.6 Error numérico total
- 2.11 Actividades a realiza

### **Unidad 3. Estructura general de un programa (3 horas)**

- 3.1 Introducción
- 3.2 Partes principales de un programa
- 3.3 Clasificación de instrucciones
  - 3.3.1 Instrucciones de inicio/fin
  - 3.3.2 Instrucciones de declaración
  - 3.3.3 Instrucciones de asignación
  - 3.3.4 Instrucción de entrada
  - 3.3.5 Instrucción de salida
  - 3.3.6 Instrucciones de control
- 3.4 Elementos auxiliares de un programa
- 3.5 Actividades a realizar

### **Unidad 4. Fundamentos de programación en *FORTRAN* (27 horas)**

- 4.1 Breve historia de *FORTRAN*
- 4.2 Compilar y ejecutar programas en *FORTRAN*
- 4.3 Elementos básicos de un programa en *FORTRAN*
- 4.5 Constantes
  - 4.5.1 Constantes enteras
  - 4.5.2 Constantes reales
  - 4.5.3 Constantes de doble precisión
  - 4.5.4 Constantes lógicas
  - 4.5.5 Constantes complejas
  - 4.5.6 Constantes carácter
  - 4.5.7 Constantes con nombre: PARAMETER
- 4.6 Declaración de variables
  - 4.6.1 Conversión entre tipos
- 4.7 Operaciones en *FORTRAN*
- 4.8 Arreglos
  - 4.8.1 Arreglos de tamaño fijo
  - 4.8.2 Asignación dinámica de memoria
- 4.9 Funciones intrínsecas
- 4.10 Control de flujo del programa
  - 4.10.1 Ciclos
  - 4.10.2 IF
  - 4.10.3 SELECT CASE

- 4.10.4 Control lógico de ciclos
- 4.10.5 EXIT y CYCLE
- 4.11 Actividades a realizar

**Unidad 5. Diseño descendente de programas en FORTRAN (12 horas)**

- 5.1 Programación descendente o modular en FORTRAN
- 5.2 Implementación de subprogramas en FORTRAN
  - 5.2.1 Funciones
  - 5.2.2 Subrutinas
  - 5.2.3 Módulos
- 5.3 Características de las subrutinas y funciones
- 5.4 Paso de valores por referencia
- 5.5 Compilación por separado de las unidades del programa.
- 5.6 Actividades a realizar

**Unidad 6. Entrada y salida por archivos en FORTRAN (6 horas)**

- 6.1 Entrada y salida de datos estándar
- 6.2 Unidades de entrada y salida
- 6.3. Entrada y salida de datos por archivo
- 6.4. Formato de entrada y salida
- 6.5 Actividades a realizar

**Unidad 7. Fundamentos de programación en OCTAVE (12 horas)**

- 7.1 Breve historia de Octave
- 7.2 Funciones básicas de Octave
  - 7.2.1 Operadores aritméticos y funciones matemáticas elementales
  - 7.2.2 Funciones matemáticas elementales
  - 7.2.3 Operadores de comparación
  - 7.2.4 Operadores booleanos
  - 7.2.5 Operadores booleanos "short-circuit"
  - 7.2.6 Operador de asignación
- 7.3 Vectores y matrices
  - 7.3.1 Matrices especiales
  - 7.3.2 Operaciones con matrices y vectores
- 7.4 Gráficas
  - 7.4.1 Gráficas en dos dimensiones
  - 7.4.2 Gráficas tridimensionales
  - 7.4.3 Múltiples gráficas
- 7.5. Programación en Octave
  - 7.5.1 Condicionales y ciclos
  - 7.5.2 Sentencia IF
  - 7.5.3 Sentencia SWITCH
  - 7.5.4 Sentencia FOR
  - 7.5.5 Sentencia DO-UNTIL
  - 7.5.6 Sentencia WHILE
  - 7.5.7 Sentencia BREAK y CONTINUE
- 7.6 Archivos \*.m
  - 7.6.1 Archivos de Comandos (SCRIPTS)
  - 7.6.2 Definición de Funciones
  - 7.6.3 HELP para las funciones de usuario
- 7.7 Actividades a realizar

### Modalidades de enseñanza aprendizaje

- Exposición didáctica por parte del docente
- Exposición por parte de los estudiantes de problemas, ejercicios, temas e investigaciones en el salón de clase.
- Resolución por parte de los estudiantes de ejercicios, problemas de manera individual o colectiva en el salón de clases.
- Realización de exámenes sin previo aviso, pero que, solamente tengan el carácter de examen diagnóstico.
- Utilización del software libre *GFortran* para codificación y ejecución de programas.
- Utilización del software libre *Octave* para codificación y ejecución de programas.
- Lectura de bibliografía en inglés.

### Modalidad de evaluación

Instrumento	Criterios de calidad	Ponderación
Trabajo personal	Autenticidad en el trabajo, uso correcto del lenguaje matemático y enmienda de errores.	40%
Participación en clase	Participación activa e interés de las intervenciones.	10%
Exposición en clase	Claridad y uso correcto de los elementos expuestos.	10%
Trabajo final	Autenticidad en el trabajo, uso correcto del lenguaje matemático, uso correcto en la aplicación de los elementos de programación en Fortran y en Octave.	40%

### Competencia a desarrollar

- G6. Usar el pensamiento cuantitativo y razonamiento analítico para identificar y analizar cantidades y magnitudes, sus formas y relaciones, a través de herramientas matemáticas modernas.
- G7. Usar herramientas de cómputo científico, entendiendo los algoritmos utilizados y las particularidades de los resultados obtenidos.
- G8. Construir un discurso comunicable de las ideas propias de acuerdo con el contexto en que se deba expresar (incluir idiomas extranjeros).
- G9. Auto gestionar el aprendizaje para el cumplimiento de las metas propias, identificando los recursos necesarios y logrando la disciplina requerida.
- G10. Crear y defender una postura propia ante los distintos fenómenos con base en el pensamiento crítico (la abstracción, el análisis y la síntesis) y privilegiando la investigación como método.
- G11. Plantear problemas de la realidad en términos del conocimiento científico disponible para su solución.

### Campo de aplicación profesional

Desarrollo de algoritmos que puedan ser llevados a un programa, ya sea en un lenguaje de programación o en programa de cálculo numérico para la resolución de diversos problemas matemáticos tanto elementales como avanzados.

### 3. BIBLIOGRAFÍA.

Título	Autor	Editorial, fecha	Año de la edición más reciente
Metodología de la Programación	Alcalde E. y García M.	McGraw Hill	1992 (2 <sup>da</sup> Ed.)
Fundamentos generales de Programación	Joyanes L.	McGraw Hill	2013 (1 <sup>ra</sup> Ed.)
Fundamentos de Algoritmia	Brassard G. y Bratley P.	Prentice Hall	1997 (1 <sup>ra</sup> Ed.)
Análisis de Algoritmos	Abellanas M. y Lodaes D.	Macrobit /Ra-Ma	1991 (1 <sup>ra</sup> Ed.)
Fundamentos de Algoritmia	Grassard G. y Bratley P.	Pearson	2000 (1 <sup>ra</sup> Ed.)
Manual de Programación para Ciencias	Villalpando Becerra J. F.	Edición Propia	2013

Formato basado en el Artículo 21 del Reglamento General de planes de estudios de la U.de G.