

# PROGRAMACIÓN ESTRUCTURADA

2137075 • CZ12 • 9 CRÉDITOS • 50 HORAS

TRIMESTRE 24-O ◇ 21/OCTUBRE/24 AL 24/ENERO/25

LUNES, MIÉRCOLES Y VIERNES ◇ 9:30 A 11:00 HR

ASAEL FABIAN MARTÍNEZ MARTÍNEZ

AT321 ◇ amartinez@izt.uam.mx ◇ fabian@xanum.uam.mx

## Descripción del curso

---

### Descripción general

En este curso se proporcionarán las bases para el diseño e implementación de programas que resuelvan problemas algorítmicos bajo los paradigmas de la programación estructurada y la programación orientada a objetos. Para ello se estudiarán diversos ejemplos de problemas que pueden ser resueltos computacionalmente. Todos los conceptos estudiados se implementarán utilizando un lenguaje de programación de alto nivel (e.g. C++, Python o Java).

### Contenido sintético

#### 1. Principios de programación estructurada

Introducción a la computación y a los computadores — procesamiento de información — algoritmos — pseudocódigo — lenguajes de programación.

#### 2. Elementos de programación

Introducción — tipos de datos simples — estructuras de control secuencial, selectivas e iterativas — procedimientos, funciones y paso de parámetros — bibliotecas de funciones — tipos de datos estructurados — manejo de archivos — diseño ascendente y descendente — documentación.

#### 3. Recursividad

### Objetivo general

Plantear soluciones algorítmicas a problemas utilizando la programación estructurada o la programación orientada a objetos.

Utilizar adecuadamente los tipos de datos simples, estructurados y abstractos en el diseño de algoritmos.

Aplicar correctamente el diseño modular en la implementación de algoritmos.

#### 4. Programación orientada a objetos

El paradigma orientado a objetos — modelado e identificación de objetos — propiedades fundamentales de la orientación a objetos — Lenguaje Unificado de Modelado (UML).

#### 5. Diseño de clases y objetos

Declaración de objetos de clases — constructores — destructores — herencia — polimorfismo — representación gráfica con UML.

#### 6. Tipos de datos abstractos

Conceptos básicos — definición e implementación — tipos de datos abstractos lineales (listas, pilas, colas) — tipos de datos abstractos no lineales (árboles, grafos).

## Aspectos y criterios de evaluación

---

El total de la calificación se dividirá como sigue:

Prácticas y tareas	30%
Proyectos parciales	35%
Proyecto final	25%
Revisión de resúmenes	10%

Se tendrán prácticas semanales, tentativamente los días viernes, en donde se aplicarán los temas vistos a lo largo de la semana. Estos ejercicios tendrán que ser entregados para su evaluación.

Adicional a las tareas, se tendrán proyectos parciales, que consistirán en dar solución a problemas específicos. La demanda para resolverlos será mayor que la requerida para las prácticas y tareas. El último proyecto tendrá que ser presentado de manera oral.

A lo largo del curso, la revisión de resúmenes consistirá en elegir los trabajos del sitio arXiv ([arxiv.org](http://arxiv.org)) para explicarlos en el aula virtual y en el salón de clases.

En el proyecto final se aplicarán todos los conocimientos adquiridos durante el curso para dar solución a algún problema específico. Además de la entrega de los códigos correspondientes, se tendrá que realizar una presentación oral del problema y su solución.

La escala de calificación será de 0 a 10 y su equivalente en letra es la siguiente:

<6.0	⇒ NA
[6, 7.8)	⇒ S
[7.7, 8.9)	⇒ B
≥8.9	⇒ MB

### Lenguaje de programación

Todos los conceptos estudiados en clase, así como todas las entregas, se implementarán en el lenguaje C/C++.

Existe la libertad de trabajar con otro lenguaje imperativo u orientado a objetos, sólo será necesario informar al profesor del cambio.

Con relación a la entrega de actividades, se presentan algunas consideraciones mínimas a tener en cuenta.

1. Para tener derecho a una calificación aprobatoria, se deberán entregar todas las actividades, sin importar su calificación.
2. Cada tarea tendrá una fecha límite de entrega. No se aceptarán entregas posteriores.
3. Tareas iguales, muy parecidas o descargadas de internet, total o parcialmente, serán anuladas; se calificará con NA a quien tenga dos o más tareas anuladas.
4. Se cuenta con un aula virtual, alojada en Virtuami  
<https://virtuami.izt.uam.mx/aulas/avmacca/>  
en donde se podrán descargar algunos de los materiales para el curso y enviar tareas.
5. Considerar los posibles inconvenientes que puedan surgir (como la falta de internet o luz, o archivos demasiado grandes) para la entrega puntual de las tareas.
6. En ningún momento se dejarán trabajos adicionales para reponer o subir calificación.

Sobre la entrega de programas:

7. Se deberán entregar dos archivos: *código fuente* y *resultados*.
8. El *código fuente* deberá estar propiamente documentado; incluir al inicio de cada archivo el nombre de quien entrega.
9. Los *resultados* consisten en la salida del programa y se deberán entregar en un archivo en formato PDF.

## Bibliografía

---

1. Joyanes, L., 2020, Fundamentos de programación. Algoritmos, estructura de datos y objetos. Quinta edición. Mc Graw Hill.
2. Joyanes, L. y Zahonero, I., 2014, Programación en C, C++, Java y UML. Segunda edición. Mc Graw Hill.
3. Hinojosa, A. P. 2022, El lenguaje de programación Python de principio a fin.