



Centro de Investigación  
en Computación

Instituto Politécnico Nacional

# Curso Propedéutico Maestrías **2021 B**

---

## **MCC – MCIC**



## Objetivos

---

- Ofertar una opción adicional para la evaluación de aspirantes en el marco de los procesos de admisión a los programas de Maestría del CIC-IPN.
- Promover una homogeneización del nivel general de los estudiantes de nuevo ingreso que sean admitidos a los programas de Maestría del CIC-IPN a partir de la opción de curso propedéutico.
- Identificar los mejores aspirantes a partir de un proceso de evaluación de su desempeño en el marco del curso propedéutico.

## Contenido y Duración

---

El curso propedéutico se oferta a través de una modalidad a distancia (plataforma digital), en idioma español, y contempla dos módulos a impartirse:

- Matemáticas para Ciencias de la Computación
- Programación y Algoritmia

La duración del curso es de diez semanas, de lunes a viernes, a través de una modalidad a distancia (plataforma digital), en idioma español, por profesores del CIC, del 17 de mayo al 23 de julio.

Cada módulo del propedéutico se impartirá en sesiones semanales durante 10 semanas, haciendo un total de 100 horas por todo el curso, tal como se detalla en la siguiente tabla:

Módulo	Horario	Duración
Matemáticas para Ciencias de la Computación	Lunes, miércoles y viernes 18-20 horas	60 horas
Programación y Algoritmia	Martes y jueves 20-22 horas	40 horas

# Inscripción

---

La inscripción será abierta a todos los aspirantes registrados en el proceso de Admisión para los programas de Maestría del CIC-IPN, para lo cual deberán haber enviado el expediente de identidad, descrito en la convocatoria vigente, así como cubrir la cantidad de \$400 pesos 00/100 M.N a través de un depósito a la siguiente cuenta:

Banco: BBVA No. de Convenio: 1089986  
Fundación Politécnico A. C. Referencia **CICCPRO9**

Se recibirán inscripciones hasta el día 14 de mayo o hasta alcanzar el cupo máximo, el cual es de 50 participantes.

## Contenido temático por módulo

---

### Matemáticas para Ciencias de la Computación

#### 1. Lógica simbólica

##### 1.1 Conceptos fundamentales

- 1.1.1 Proposiciones y valor de verdad
- 1.1.2 Lenguaje del Cálculo Proposicional
- 1.1.3 Sintaxis: Fórmulas bien formadas
- 1.1.4 Semántica: Tablas de verdad.
- 1.1.5 Pragmática: Representación del conocimiento.

##### 1.2. Equivalencia lógica.

- 1.2.1 Equivalencia y sustituciones.
- 1.2.2 Leyes de idempotencia, identidad y dominación.
- 1.2.3 Leyes de conmutatividad y asociatividad.
- 1.2.4 Leyes de distributividad.
- 1.2.5 Leyes de DeMorgan y negación.

#### 1.3. Formas Normales

- 1.3.1 Forma normal negativa.
- 1.3.2 Forma normal conjuntiva
- 1.3.3 forma normal disyuntiva
- 1.3.4 Conversión entre formas normales

#### 1.4 Sistemas formales de razonamiento.

- 1.4.1 Inferencia, deducción, inducción y abducción
- 1.4.2 Reglas de inferencia en el cálculo proposicional
- 1.4.3 Técnicas de demostración
  - 1.4.3.1 Vacuidad, trivialidad y contraejemplo.
  - 1.4.3.2 Demostración condicional directa.
  - 1.4.3.3 Demostración indirecta.
  - 1.4.3.4 Subdemostraciones

#### 1.5 Teorema de inducción lógica

#### 1.6 Principio de inducción matemática

### 2. Conjuntos

- 2.1 Introducción
- 2.2 Definiciones
- 2.3 Operaciones sobre conjuntos
- 2.4 Propiedades
- 2.5 Producto cartesiano
- 2.6 Propiedades

### 3. Relaciones

- 3.1 Introducción
- 3.2 Definiciones
- 3.3 Propiedades de las relaciones.
- 3.4 Relaciones de orden
- 3.5 Relaciones de equivalencia
  - 3.5.1 Clases de equivalencia
  - 3.5.2 Conjunto cociente.

### 4. Funciones

- 4.1 Introducción
- 4.2 Definiciones
- 4.3 Clasificación de funciones.

- 4.4 Propiedades
- 4.5 Funciones inducidas por una función.
- 4.6 Propiedades.
- 4.7 Cardinalidad
- 4.8 Propiedades

## 5. Técnicas de conteo

- 5.1 Introducción
- 5.2 Principio de adición y de multiplicación.
- 5.3 Permutaciones
- 5.4 Binomiales y Combinaciones
- 5.5 Principio del palomar (Pigeonhole)

## Bibliografía

1. Daniel J. Velleman. *How to prove it. A structured approach*. (3rd edition). Cambridge University Press. (2019).
2. Michael L. O'Leary. *A First Course in Mathematical Logic and Set Theory*. Wiley. (2016).
3. Hirst, H. P., Hirst, J. L. *A primer for logic and proof*. (2015 edition). Appalachian State University, Boone, NC. Recuperado de: <http://www.appstate.edu/~hirstjl/primer/hirst.pdf> (2015)
4. Susanna S. Epp. *Discrete Mathematics with Applications*. Cengage Learning; (2019)
5. Kenneth Rosen. *Discrete Mathematics and Its Applications*. Mc Graw Hill (2018)
6. Juan Carlos Chimal Eguía. *Notas de clase*. Centro de Investigación en Computación, IPN. (2020)
7. René Luna García. *Notas de clase*. Centro de Investigación en Computación, IPN. (2021)
8. Salvador Godoy Calderón. *Notas de clase*. Centro de Investigación en Computación, IPN. (2021)
9. Germán Téllez Castillo. *Notas de clase*. Centro de Investigación en Computación, IPN. (2021)

## Programación y Algoritmia

1. Conceptos y técnicas de lenguajes de programación.
  - 1.1. Lenguajes de programación más representativos y sus características.
  - 1.2. Estructura léxica, sintáctica y semántica.

- 1.3. Tipos de datos, declaración y enlace dinámico.
- 1.4. Estructuras de control y bloques de instrucciones.
- 1.5. Subprogramas y funciones.
- 1.6. Tipos de datos abstractos y encapsulamiento.
- 1.7. Programación orientada a objetos.
- 1.8. Herencia, sobrecarga y polimorfismo.

## 2. Programación en Python y Java.

- 2.1. Características de Java y control de flujo.
- 2.2. Tipos de datos y operadores en Java.
- 2.3. Clases, métodos y programación de objetos en Java.
- 2.4. Python sintaxis y control de flujo.
- 2.5. Python tipos de datos y funciones.
- 2.6. Principales bibliotecas en Python.
- 2.7. Estructuras de datos en Java y Python.
- 2.8. Clases y objetos en Python.

## 3. Análisis de algoritmos y notación asintótica.

- 3.1. Análisis de algoritmos aspectos básicos y ejemplos.
- 3.2. Crecimiento asintótico comparativo.
- 3.3. La complejidad de un algoritmo y su estimación.
- 3.4. El mejor, el promedio y el peor caso en la ejecución de un algoritmo.
- 3.5. La cota mínima de un problema.
- 3.6. Hallar la cota mínima mediante la transformación de un problema.

## 4. Análisis de algoritmos representativos

- 4.1. Algoritmos de búsqueda y ordenamiento.
- 4.2. Algoritmos voraces.
- 4.3. Grafos: búsqueda en profundidad, amplitud y Dijkstra.
- 4.4. Problema del agente viajero y el cartero.

## 5. Programación dinámica.

- 5.1. Conceptos y estrategias de la programación dinámica.
- 5.2. El problema de asignación de recursos.
- 5.3. El problema de la mochila.
- 5.4. Cálculo de la distancia de edición.
- 5.5. Ciclos negativos en un grafo.
- 5.6. El problema del árbol binario óptimo.



6. Algoritmos de aproximación, y no polinomiales.
  - 6.1. Conceptos y estrategias de los algoritmos aproximados.
  - 6.2. El problema de la mochila aproximado.
  - 6.3. El problema de decisión y el de satisfacción.
  - 6.4. Problemas NP y NP completos, el teorema de Cook y demostraciones

#### Bibliografía

1. *Concepts of programming languages*, Robert W. Sebesta.
2. *Introduction to algorithms*, Thomas H. Cormen.
3. *Algorithm Design*, Jon Kleinberg y Eva Tardos.
4. *Introduction to the Design and Analysis of Algorithms*, R.C.T. Lee.

## Estrategia de Evaluación

Los profesores responsables de cada módulo emitirán una calificación correspondiente al tema o curso, la cual se calculará de acuerdo con la siguiente ponderación:

Módulo	Examen	Tareas
<b>Matemáticas para Ciencias de la Computación</b>		
- Lógica	50%	50%
- Conjuntos	50%	50%
- Relaciones	50%	50%
- Funciones	50%	50%
- Técnicas de conteo	80%	20%
<b>Programación y Algoritmia</b>	100%	--

Los exámenes serán realizados a través de una plataforma de entorno seguro especializada para tal efecto.

La calificación final del curso propedéutico será calculada como el pro-

medio de la calificación obtenida de cada uno de sus módulos. No se proporcionará retroalimentación específica o un desglose parcial de la calificación. Al final se realizará una encuesta de evaluación del fragmento del curso impartido por cada profesor.

## Aprobación para el proceso de admisión B21

La calificación final obtenida será utilizada para acreditar el examen de admisión del aspirante a los programas de maestría del CIC-IPN, para lo cual será aplicada una regla de equivalencia entre escalas entre el puntaje obtenido en el curso propedéutico y los puntajes GRE, EXANI III y EXADEP del resto de aspirantes.

