Curso de selección para ingreso a las Maestrías

2024 B

Centro de Investigación en Computación
Instituto Politécnico Nacional

ESTE PROGRAMA ES PÚBLICO, AJENO A CUALQUIER PARTIDO POLÍTICO. QUEDA PROHIBIDO EL USO PARA FINES DISTINTOS A LOS ESTABLECIDOS EN EL PROGRAMA.

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

I. Objetivos

- Ofertar una opción adicional para la evaluación de las personas aspirantes en el marco de los procesos de admisión a los programas de maestría del CIC-IPN.

- Evaluar el nivel general de las personas aspirantes a los programas de maestría del CIC-IPN, a partir de la opción de Curso de Selección para ingreso a las maestrías.

- Identificar los mejores aspirantes a partir de un proceso de evaluación de su desempeño en el marco del curso de selección para ingreso a las maestrías.

II. Contenido y Duración

El curso de selección para ingreso a las maestrías 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 hábiles, de lunes a viernes, iniciando el 8 de abril del 2024 para concluir el 14 de junio de 2024.

Cada módulo del Curso de Selección se impartirá en 2 y 3 sesiones semanales de 2 horas durante 10 semanas, haciendo un total de 100 horas por todo el curso. Las asignaturas y horarios se distribuyen de la siguiente forma:

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

III. Requisitos de Inscripción

La inscripción estará abierta a partir de la publicación de esta convocatoria para los aspirantes interesados en el Proceso de Admisión B24 para los programas de maestría del CIC-IPN, para lo cual deberán:

I. Realizar su registro al Curso de Selección en la siguiente liga:
https://forms.gle/dgtq6pgorhhgkGn46

II. Una vez revisado su registro, se le enviarán los datos para que realice el pago por la cantidad de $500 pesos 00/100 M. N. en una cuenta de Fundación Politécnico (no realizar ningún pago previo);

III. Estarán exentos de pago aquellos aspirantes que comprueben un promedio general mayor o igual que 9.0 en su licenciatura.

IV. Fechas Importantes

El proceso de inscripción al Curso de Selección, así como el Proceso de Admisión al semestre B24 deberá apegarse al siguiente calendario.

Descripción	**Periodo / Fecha (GMT -6)**
Registro e Inscripción al Curso de Selección	A partir de la publicación y hasta el 25 de marzo de 2024
Inicio del curso	8 de abril de 2024
Conclusión del curso	14 de junio de 2024
Examen final presencial módulo de matemáticas para ciencias de la computación	19 de junio de 2024 16:00 horas

Se recibirán inscripciones hasta el 25 de marzo del año en curso o hasta alcanzar el cupo máximo de 60 participantes; en caso de tener

dudas con respecto al cupo, pueden enviar un correo a la dirección: hanya@cic.ipn.mx

a) Una vez realizado el pago, recibirá un correo de confirmación de su inscripción.

V. Contenido temático por módulo

Matemáticas para Ciencias de la Computación

1. Lógica simbólica
   - 1.1 Conceptos fundamentales
   - 1.2 Equivalencia lógica.
   - 1.3 Formas normales
   - 1.4 Sistemas formales de razonamiento.
   - 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

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. Velleman D. J. *How to prove it. A structured approach.* (Third edition). Cambridge University Press. (2019).

2. Oleary M. L. 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, N. C. Recuperado de: http://www.appstate.edu/~hirstjl/primer/hirst.pdf. (2015)

4. Susanna S. Epp. *Discrete Mathematics with Applications.* Cengage Learning; (2019).

5. Rosen K. *Discrete Mathematics and its Applications.* Mc Graw Hill (2018).

6. Chimal Eguía J. C. Notas de clase. Centro de Investigación en Computación, IPN. (2020).

7. Luna García R. Notas de clase. Centro de Investigación en Computación, IPN. (2021).

8. Godoy Calderón S. Notas de clase. Centro de Investigación en Computación, IPN. (2021).

9. Téllez Castillo G. 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
   - 2.1 Sintaxis y control de flujo en Python.
   - 2.2 Tipos de datos y funciones en Python.
   - 2.3 Principales bibliotecas en Python.
   - 2.4 Estructuras de datos en Python.
   - 2.5 Clases y objetos en Python.
   - 2.6 Proyectos de programación en Python

3. Análisis de algoritmos y notación asintótica.
   - 3.1 Demostración de que un algoritmo es correcto.
   - 3.2 Caracterización del tiempo de ejecución: Uso de función potencial.
   - 3.3 Orden de crecimiento asintótico y notación asintótica.
   - 3.4 Ejemplos.

4. Algoritmos para grafos
   - 4.1 Definición de grafo y su representación computacional
   - 4.2 Búsqueda por amplitud
   - 4.3 Búsqueda por profundidad
   - 4.4 Presentación de problemas difíciles: Conjunto independiente, cobertura de vértices, el problema del viajero.

5. Algoritmos voraces
   - 5.1 Planificación de intervalos.
   - 5.2 Algoritmo de Dijkstra para el cálculo de caminos cortos.
   - 5.3 Árboles de expansión mínimos.
   - 5.4 Una mirada a los algoritmos de aproximación: Balanceo de carga.

6. Programación dinámica.
   - 6.1 Conceptos y estrategias de la programación dinámica.
   - 6.2 El problema de planificación de intervalos con peso.
   - 6.3 El problema de la mochila.
   - 6.4 Predicción de estructura secundaria de Redes Neuronales Artificiales

7. Algoritmos de aproximación y problemas NP-completos.
   - 7.1 Problemas NP y NP completos.
   - 7.2 Conceptos y estrategias de los algoritmos aproximados.
   - 7.3 El problema de los k-centros.
   - 7.4 El problema de satisfacibilidad.

8. Ejemplos y desarrollo de programas.
   - 8.1 El problema a resolver.
   - 8.2 Contexto y estado del arte.
   - 8.3 Datos de entrada y salida.
   - 8.4 Complejidad teórica y práctica.

Bibliografía

1. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

2. Kleinberg, Jon, and Tardos, Eva. *Algorithm design*. Addison Wesley, 2005.

3. Dasgupta, Sanjoy, Papadimitrou, Christos, and Vazirani Umesh. *Algorithms*. McGraw-Hill Science/Engineering/Math; 2006.

4. Matthes, Eric. Python crash course: *A hands-on, project-based introduction to programming. no starch press,* 2023.

5. Lutz, Mark. Learning python: *Powerful object-oriented programming*. "O'Reilly Media, Inc.", 2013.

6. Ayeva, Kamon, and Sakis Kasampalis. *Mastering Python Design Patterns: A guide to creating smart, efficient, and reusable software*. Packt Publishing Ltd, 2018.

VI. Estrategia de Evaluación

Para el módulo de **Programación y Algoritmia,** se tienen 10 semanas de clase (40 horas) y una semana extra para la presentación del proyecto final. Se evaluará con participaciones (10%); tareas (20%); proyecto final (70%).

Para el módulo de **Matemáticas para Ciencias de la Computación**, se tienen 10 semanas de clase (60 horas) y una sesión para evaluación. La evaluación será con un **examen presencial** aplicado una semana después de terminar el curso.

La calificación final del Curso de Selección será calculada como el promedio de la calificación obtenida en cada uno de los módulos. No se proporcionará retroalimentación específica o un desglose parcial de la calificación. Al final de cada unidad se realizará una encuesta de evaluación del profesor que la impartió.

VII. Aprobación para el semestre B24

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