PREUNIVERSITARIO

Programa de cátedra

UTN UNIVERSIDAD TECNOLÓGICA NACIONAL	**Asignatura:**	**Departamento:**
	Bloque:	**Área:**
	Régimen:	**Horas semanales:**
	Tipo:	**Horas anuales:**
	Carrera: Tecnicatura universitaria en programación a distancia	**Nivel (año):** ☐ 1° ☐ 2°
	Ciclo lectivo: 2025	

Integrantes de la cátedra:

- Docentes:

Nombre del profesor	Periodo	Cantidad horas semanales
Cinthia Rigoni		

a) Fundamentación del preuniversitario

El preuniversitario comienza con una introducción a la lógica y estructura de la programación, proporcionando a los estudiantes una base sólida para ingresar al mundo de la programación y abordar la resolución de problemas de manera estructurada. A través del estudio de conceptos clave como algoritmos y estructuras secuenciales, condicionales y repetitivas, los estudiantes desarrollan habilidades fundamentales para comprender la construcción lógica de un programa y la interrelación de sus componentes, utilizando herramientas como el pseudocódigo y la aplicación PSeint.

Este curso está diseñado para que los estudiantes se familiaricen con el razonamiento lógico, la toma de decisiones en el código y el flujo de control en diversas estructuras, sentando una base esencial para futuros estudios en programación y cursos avanzados. Además de los conceptos básicos y estructurales de la programación, el preuniversitario introduce a los estudiantes al concepto de recursividad, una técnica esencial en programación que permite resolver problemas mediante soluciones que se llaman a sí mismas bajo ciertas condiciones. Esto facilita la solución de determinados tipos de problemas y enriquece el pensamiento lógico y la comprensión de la programación estructurada.

b) Objetivos del curso

Objetivos Generales:

- Proporcionar a los estudiantes una base sólida en lógica y fundamentos de la programación, esenciales para el desarrollo de algoritmos y programas en cualquier lenguaje.
- Fomentar el desarrollo de habilidades de pensamiento crítico y lógico para resolver problemas computacionales de manera estructurada y eficiente.
- Proporcionar una base sólida en lógica y fundamentos de programación, esencial para resolver problemas complejos.

Objetivos Específicos:

1. Comprender la estructura de algoritmos y su representación a través de pseudocódigo y diagramas de flujo.
2. Aprender a declarar y utilizar variables y operadores básicos.
3. Comprender y aplicar estructuras secuenciales, condicionales y repetitivas en algoritmos básicos.
4. Aplicar correctamente los operadores aritméticos, relacionales y lógicos en la construcción de programas.

c) Contenidos mínimos

- Estructuras Secuenciales: Declaración de variables, operadores y estructura secuencial.
- Condicionales: Condicionales simples, compuestas y múltiples, uso de estructura según-caso.
- Repetitivas: Ciclos para, mientras y repetir-mientras.

d) Programa analítico

Unidad temática	**Contenidos**
Unidad 1: Estructuras secuenciales	**Objetivo:** Comprender y aplicar el concepto de algoritmo secuencial para realizar operaciones básicas en programación. - Introducción a los algoritmos y pseudocódigo. - Declaración y uso de variables. - Operadores aritméticos y de asignación. - Estructura secuencial en pseudocódigo y diagramas de flujo.
Unidad 2: Estructuras condicionales	**Objetivo:** Aplicar la lógica condicional en programación para tomar decisiones dentro del código. - Condicionales simples (SI) y compuestas (SI – SINO). - Operadores relacionales y lógicos. - Uso del operador MOD para divisiones enteras. - Estructuras condicionales múltiples (anidadas). - Estructura según-caso.
Unidad 3: Estructuras repetitivas	**Objetivo:** Implementar estructuras de repetición para ejecutar bloques de código múltiples veces. - Ciclo “PARA”. - Ciclo “MIENTRAS”. - Ciclo “REPETIR-MIENTRAS”.

e) Programa de examen

El programa de examen será similar al programa analítico. Los estudiantes deberán demostrar un conocimiento claro de las estructuras secuenciales, condicionales y repetitivas, aplicándolos en ejercicios prácticos.

f) Trabajos prácticos

Unidad	Título del trabajo	Objetivo	Temas a cubrir
1	Práctico 1: Algoritmos secuenciales	Desarrollar la capacidad de representar problemas de forma estructurada mediante algoritmos y pseudocódigo.	Estructura secuencial, variables y operadores.
2	Práctico 2: Estructuras Condicionales	Aplicar condicionales en la programación para resolver problemas que requieran decisiones.	Condicionales simples, compuestas y múltiples.
3	Práctico 3: Estructuras Repetitivas	Implementar ciclos para resolver problemas que requieran repetición de acciones.	Ciclo para, ciclo mientras, ciclo repetir-mientras.

g) Estrategias de evaluación

El proceso evaluativo de la materia **Introducción a la Programación** es progresivo, con un enfoque en la evaluación continua y la retroalimentación constante. Las evaluaciones están diseñadas para que los estudiantes consoliden los conocimientos adquiridos en cada unidad y avancen de manera estructurada a lo largo del curso.

Actividades en el Aula

1. **Evaluaciones por Actividad**
   - Cada actividad presentada en el aula virtual cuenta con una evaluación asociada.
   - Los estudiantes tendrán **intentos ilimitados** para aprobar estas evaluaciones, alcanzando una calificación mínima de **7 (siete)**.
   - La aprobación de cada actividad habilita automáticamente la siguiente evaluación dentro de la unidad.
2. **Trabajos Prácticos**
   - Cada módulo incluye un **trabajo práctico obligatorio**, que debe ser entregado antes de poder rendir las **autoevaluaciones** correspondientes al módulo.
   - Las autoevaluaciones constan de **2 (dos) oportunidades** para alcanzar la calificación mínima de **7 (siete)**.
3. **Secuencia de Unidades**
   - La aprobación de todas las actividades y evaluaciones de una unidad es requisito para avanzar a la siguiente unidad.
4. **Esparcimiento y Juegos**
   - Los estudiantes participarán en actividades lúdicas como juegos, salas de escape y otros espacios de esparcimiento.

Examen Final

- Una vez completadas todas las unidades, los estudiantes deberán rendir un **Examen Final**, que evaluará de manera integral los contenidos abordados durante el curso.
- La calificación mínima para aprobar el Examen Final es **6 (seis)**.
- Los estudiantes que aprueben el Examen Final con esta calificación alcanzarán la **aprobación del curso**.

Condiciones de Aprobación

1. **Aprobación de Unidades**
    - Completar y aprobar con una calificación mínima de **7 (siete)**:
        - Todas las actividades de evaluación asociadas a cada unidad.
        - Todos los trabajos prácticos obligatorios.
    - La aprobación de una unidad es necesaria para acceder a la siguiente.
2. **Aprobación del Curso**
    - Aprobar todas las unidades según lo descrito anteriormente.
    - Rendir y aprobar el **Examen Final** con una calificación mínima de **6 (seis)**.
3. **Condición de Aprobación Directa**
    - No aplica en este modelo evaluativo progresivo. Los estudiantes deben cumplir con las condiciones específicas para cada evaluación y el Examen Final.

Instancias de Recuperación

- Las evaluaciones por actividad cuentan con **intentos ilimitados** hasta alcanzar la calificación mínima.
- Las autoevaluaciones tendrán **2 (dos) oportunidades** para aprobar.
- En el caso de los trabajos prácticos, las entregas podrán reintentarse si es necesario, siempre dentro de los plazos establecidos.

Retroalimentación Personalizada

A lo largo del curso, se proporcionará retroalimentación individualizada en cada instancia de evaluación, con el objetivo de reforzar el aprendizaje, identificar áreas de mejora, y acompañar a los estudiantes en su proceso de formación.