

DEPARTAMENTO Y/O DELEGACION: COMPUTACIÓN APLICADA

PROGRAMA DE CATEDRA: **ALGORITMOS MATEMATICOS Y PROGRAMACION**

OPTATIVA: NO

CARRERA/S A LA QUE PERTENECE Y/O SE OFRECE :
PROFESORADO UNIVERSITARIO EN MATEMATICA

AREA: MÉTODOS COMPUTACIONALES

ORIENTACION: MÉTODOS NUMÉRICOS

PLAN/ES DE ESTUDIOS:
1467/14, 0699/16 - 0695/20

TRAYECTO (PEF): N/C

CARGA HORARIA SEMANAL SEGÚN PLAN DE ESTUDIOS: 4

CARGA HORARIA TOTAL: 64

REGIMEN: Segundo Cuatrimestre

EQUIPO DE CATEDRA: (Apellido y Nombres - Cargo)

- **Viviana Zimmerman - PAD2**
- Martín Vilugron - AYP3

ASIGNATURAS CORRELATIVAS (Según Plan/es de Estudios):

- PARA CURSAR:

-

- PARA RENDIR EXAMEN FINAL:

-

1. **FUNDAMENTACIÓN:**

El desarrollo de algoritmos y su aplicación en la programación de computadoras son herramientas que en la actualidad se han mostrado muy valiosas para la enseñanza de diversos aspectos

fundamentales en la resolución de problemas en general, y en particular de problemas matemáticos. Uno de los beneficios intercurriculares más comunes de la programación de computadoras es que simplifica el aprendizaje de contenidos matemáticos, ya que permite introducir conceptos matemáticos en forma natural y en general más atractiva. Por otro lado, también estimula algo que es tan fundamental como lo es la creatividad.

Como consecuencia de esto, resulta fundamental para estudiantes del Profesorado en Matemáticas tener un buen manejo de la programación de computadoras para poder luego volcar estos conocimientos en sus futuros estudiantes.

Teniendo en mente este propósito, resulta conveniente introducir la programación empleando un lenguaje como es mBlock. mBlock es un lenguaje de programación libre, basado en Bloques, con un entorno de programación visual y multimedia que permite crear programas de forma fácil e intuitiva mediante una interfaz gráfica. La programación con bloques es en general más fácil de trabajar que la programación basada en texto, ya que los bloques no necesitan ser memorizados como los comandos escritos y no pueden ocurrir errores de sintaxis. mBlock está basado (y es casi idéntico) a Scratch, que se puede considerar como un sucesor del lenguaje Logos, fue originariamente diseñado para expresar ideas de forma creativa con el objetivo de incentivar el desarrollo de habilidades de pensamiento lógico y computacional principalmente en niñas/os.

Sin embargo, la programación basada en texto es más flexible, ya que el texto se puede editar fácilmente sin tener que arrastrar bloques al editor, y también en general es más eficaz. Teniendo esto en cuenta, en la segunda parte de la materia se introduce el lenguaje de programación Python. Python es un lenguaje de alto nivel relativamente fácil de aprender y muy actual. Esta última característica lleva a que se encuentre mucha información, tutoriales, etc., pero sobre todo, muchas librerías que se pueden utilizar para simplificar el desarrollo de nuevos códigos.

2. OBJETIVOS:

- * Comprender qué es un algoritmo, cómo son implementados en forma de programa y cómo se almacenan y ejecutan sus instrucciones.
- * Adquirir habilidad en el planteo y resolución de problemas mediante el uso de estructuras de datos básicas y procesos algorítmicos.
- * Lograr habilidad en la utilización de los lenguajes de programación mBlock y Python.
- * Diseñar, crear, probar y depurar programas completos en mBlock y Python que permitan resolver diferentes tipos de problemas.

3. CONTENIDOS SEGÚN PLAN DE ESTUDIOS:

- Resolución de problemas y algoritmos. Algoritmos y Lenguajes. Estrategias y recursos para el diseño de algoritmos. Estructuras de control. Tipos de datos simples.
- Expresiones lógicas y su uso en programación. Mecanismos básicos de abstracción. Modularización.
- Algoritmos matemáticos y numéricos simples. Entornos de Programación basados en bloque y basados en texto.

4. CONTENIDO PROGRAMA ANALÍTICO:

Unidad 1: Introducción y conceptos básicos.

Importancia de la programación de computadoras en la enseñanza de la matemática. Lenguajes de programación. Lenguajes de programación por bloques y por texto. Diagramas de flujo y pseudocódigo. Algoritmos.

Unidad 2: Entorno de mBlock.

Instalación y configuración del entorno. Utilización del entorno web. Creación de usuario. Descripción del entorno de desarrollo: Escenario, Objetos, Fondos, Sección de categorías de bloques, Área de programación.

Unidad 3: Introducción a los Bloques.

Tipos de bloques. Introducción a las primeras Categorías de Bloques: Movimiento, Apariencia y Sonido. Ubicación, movimiento, orientación, deslizamiento. Disfraces. Edición de disfraces y fondos. Sonidos, edición de sonidos. Galerías de Objetos, Sonidos y Fondos. Tamaño de los Objetos. Mensajes de salida de los Objetos.

Unidad 4: Introducción a estructuras del código y Extensiones.

Categoría Eventos. Formas de iniciar la ejecución de una secuencia de bloques. Categoría Control. Bucles o repeticiones de bloques. Bloques adicionales: Extensiones. Lápiz y Música.

Unidad 5: Datos.

Manejo de directo de datos ingresados en los bloques. Datos numéricos y de texto. Bloques informadores predefinidos. Definición de Variables y Listas. Manejo de datos y variables mediante Operadores. Operaciones aritméticas. Ingreso de datos: categoría Sensores.

Unidad 6: Estructuras de control de flujo del programa.

Categoría Control. Estructuras condicionales: Si, Si / Si no. Bucles con condiciones: Repite hasta que. Operaciones Lógicas. Categoría Sensores para manejo de condiciones. Categoría Mis Bloques: Definición de nuevos bloques (funciones). Nuevos bloques con y sin argumentos. Categoría Eventos: Control de flujo de programa mediante mensajes. Categoría Control: Multiplicación de Objetos como Clones.

Unidad 7: Introducción a la programación en Python.

Instalación y configuración del entorno. Conceptos básicos de Python: sintaxis, indentación, comentarios. Primeros ejemplos básicos de programas en Python.

Unidad 8: Manejo de datos.

Variables y tipos de datos. Tipos de datos numéricos, lógicos y de cadena de caracteres. Operadores aritméticos y de asignación. Conversión de tipos. Métodos. Operadores lógicos. Precedencia de operadores. Entrada y salida de datos. Formateo de de resultados. Estructuras de datos: listas, tuplas, diccionarios y conjuntos: definición, manipulación y acceso a elementos.

Unidad 9: Estructuras de control de flujo.

Condicionales: if, elif, else. Loops e iteraciones: while, for. Iteraciones sobre estructuras de datos. Funciones. Definición de funciones. Distintos tipos de argumentos. Recursividad. Módulos.

Unidad 10: Bibliotecas o paquetes. Introducción a Numpy. Variables de tipo arreglo (arrays): Creación manual y mediante funciones. Arrays multidimensionales. Selección de partes de arreglos. Operaciones y métodos sobre arrays. Operaciones entre vectores. Introducción a visualización con Matplotlib. Creación de gráficos simples. Títulos, leyendas, y otros. Introducción de fórmulas. Gráficos múltiples.

5. BIBLIOGRAFÍA BÁSICA Y DE CONSULTA:

- Página principal de mBlock: <https://mblock.makeblock.com/en/>
- <https://mblock.makeblock.com/en/download/>

- Aprende a programar con mBlock: <https://www.robotix.es/es/ebook-mblock>
- mBlock 5 Online Help: <https://www.yuque.com/makeblock-help-center-en/mblock-5>
- Guía de referencia de Scratch: <http://www.eduteka.org/pdfdir/ScratchGuiaReferencia.pdf>
- Wiki de Scratch: https://test.scratch-wiki.info/wiki/Spa:Scratch_Wiki_Español
- Sitio web del lenguaje python: <http://www.python.org/>
- Tutorial oficial del lenguaje Python: <https://docs.python.org/3/tutorial/index.html>
- Bibliotecas: funciones y métodos de la librería estándar: <http://docs.python.org/library/>
- Referencia del Lenguaje: sintaxis y semántica del lenguaje: <http://docs.python.org/reference/>
- Curso de Introducción a Python para Ciencias e Ingenierías: <https://fiolj.github.io/intro-python-IB/>

6. PROPUESTA METODOLÓGICA MODALIDAD PRESENCIAL:

Clases de teoría:

Las clases se plantean como teórico-prácticas con una explicación de los temas en el pizarrón o con proyección de pantalla para demostración de los temas de la asignatura, seguido de ejemplos concretos implementados por las y los estudiantes en las computadoras.

Clases prácticas de problemas:

Espacio de trabajo individual o en parejas de resolución de problemas en computadora, con discusión grupal de problemas de mayor complejidad y/o fundamentales para la comprensión de la unidad, en aquellos casos que sea necesario. En las clases prácticas se trabajará con guías de trabajos prácticos y ejercicios. Tanto las clases teóricas como las clases prácticas se desarrollarán en el laboratorio de informática para poder implementar en forma práctica todos los conocimientos que se vayan adquiriendo a lo largo de la cursada.

7. EVALUACIÓN Y CONDICIONES DE ACREDITACIÓN:

Alumnos Regulares:

Para aprobar la asignatura las y los estudiantes deben:

- asistir a clases ya que se realizará una evaluación continua de su avance en la asignatura;
- aprobar en primera o segunda instancia cada uno de los exámenes parciales que se tomen durante el cursado. Cada examen parcial tendrá una instancia de recuperación en caso de ser desaprobado;
- entregar las resoluciones de los ejercicios solicitados en cada una de las guías de problemas;
- una vez cumplidos los requisitos anteriores, deberán rendir y aprobar con puntuación igual o mayor a cuatro (4) un examen final en alguna de las fechas previstas por la Universidad para tal fin. Este examen consistirá en la creación de un programa que abarque los contenidos aprendidos en la materia. El tema del del programa será acordado con las/los docentes, y lo deberán desarrollar en sus casas previamente a la fecha del examen, fecha en la cual deberán exponer y explicar lo desarrollado, en forma oral.

Alumnos Libres:

El Examen Libre constará de dos partes. Una de estas partes consistirá en la presentación y exposición de un proyecto previamente desarrollado en sus casas. El tema del proyecto será elegido por las/os docentes del tribunal examinador. La segunda parte consistirá en resolver en el aula una serie de códigos también indicados por las/los docentes. Es necesario aprobar con nota mayor a cuatro (4) la primera instancia para poder pasar a la siguiente. La nota final es el promedio de las dos instancias evaluadas.

8. DISTRIBUCIÓN HORARIA:

Clases de teorías: Miércoles de 13: 30 a 15:30 hs.

Clases prácticas de problemas: Jueves de 16 a 18 hs.

9. CRONOGRAMA TENTATIVO:

Semana 1	Unidades 1 y 2
Semana 2	Unidad 3 y 4
Semana 3	Unidad 5
Semana 4	Unidad 6 (primera parte)
Semana 5	Examen parcial
Semana 6	Unidad 6 (continuación)
Semana 7	Unidad 6 (continuación)
Semana 8	Unidad 6 (continuación)
Semana 9	Examen parcial
Semana 10	Unidad 7
Semana 11	Unidad 8
Semana 12	Unidad 9
Semana 13	Unidad 9 (continuación)
Semana 14	Unidad 10
Semana 15	Examen parcial
Semana 16	Recuperación Exámenes parciales

PROFESORA

Viviana A. Ramirez

Dra. MARIANA PUETA
Secretaria Académica
Centro Regional Univ. Bariloche
Univ. Nacional del Comahue

CONFORMIDAD DEL
DEPARTAMENTO

CONFORMIDAD SECRETARÍA ACADÉMICA