

**UNIVERSIDAD NACIONAL DE RÍO CUARTO**

**FACULTAD DE CIENCIAS EXACTAS, FÍSICO-QUÍMICAS Y NATURALES**

**DEPARTAMENTO DE COMPUTACIÓN**

**CARRERA/S:** Profesorado y Licenciatura en Matemática

**PLAN DE ESTUDIOS:** Profesorado en Matemática (Plan 2001 versión 3)

 Licenciatura en Matemática (Plan 2008 versión 1)

**ASIGNATURA:** Taller de Informática **CÓDIGO:** 1927

**DOCENTE RESPONSABLE:** Dr. Renzo Degiovanni

**EQUIPO DOCENTE:**

**AÑO ACADÉMICO:** 2017

**REGIMEN DE LA ASIGNATURA:** Cuatrimestral

**RÉGIMEN DE CORRELATIVIDADES:**

|  |  |
| --- | --- |
| ***Aprobada*** | ***Regular*** |
| - | Matemática Discreta Cód. 1925 (Sólo para el Profesorado)  |

**CARGA HORARIA TOTAL:** 90 horas modalidad Taller (Teórico/Prácticos/Laboratorios)

**CARÁCTER DE LA ASIGNATURA:** Obligatoria

1. **CONTEXTUALIZACIÓN DE LA ASIGNATURA**

*Taller de Informática* es una materia de primer año de las carreras Profesorado y Licenciatura en Matemática, y consiste principalmente en un curso inicial de programación orientado a la resolución de problemas matemáticos.

1. **OBJETIVOS PROPUESTOS**

El principal objetivo de esta materia es que los alumnos adquieran conocimientos de resolución de problemas usando algoritmos, y sean capaces de codificar dichos algoritmos en algún lenguaje ejecutable.

Para lograrlo se estudiarán técnicas de resolución de problemas como: representación de datos, división en subproblemas, abstracción en funciones/procedimientos. Además, se estudiará extensivamente la codificación de algoritmos en el paradigma de programación imperativo (incluyendo las nociones de variable, asignación, estructuras de control, definición e invocación a funciones y procedimientos, pasaje de parámetros, etc.), y se realizará una breve introducción al paradigma funcional (involucrando principalmente la noción de recursión).

Otro de los objetivos es que los alumnos puedan aplicar los conocimientos de programación adquiridos a la resolución de problemas matemáticos simples y de mediana complejidad. La materia contará con un taller, en donde se utilizará un lenguaje de programación específico (Octave) para atacar este tipo de problemas. Cabe aclarar que los conceptos de programación y resolución de problemas mediante algoritmos se presentarán en forma independiente del lenguaje particular, de modo que el alumno pueda aplicarlos en el futuro a la codificación de programas en cualquier lenguaje de programación.

Finalmente, se espera que al final del curso los alumnos conozcan diversas herramientas provistas por las TICs que sirven de soporte a la enseñanza y al aprendizaje de la matemática.

1. **CONTENIDOS BÁSICOS DEL PROGRAMA A DESARROLLAR**

**Unidad I: Introducción.** Sistemas de computación.Algoritmos y resolución de problemas mediante algoritmos. Lenguajes de programación y programas. Compiladores e intérpretes.

**Unidad II: Programación imperativa.** Expresiones y operadores. Variables y tipos básicos. El operador de asignación. Sentencias. Estructuras de control (composición secuencial, alternativa, ciclos, etc.).

**Unidad III: Representación de datos.** Tipos de datos básicos. Secuencias (vectores). Cadenas de caracteres. Matrices. Operadores vectoriales y matriciales. Tipos de datos definidos por el usuario (Registros). Representación gráfica de datos. Entrada y salida básica (lectura y escritura de archivos).

**Unidad IV: Modularización.** Descomposición de problemas en subproblemas. Definición de funciones y procedimientos. Pasaje de parametros. Introducción al paradigma funcional. Definición de funciones recursivas simples.

**Unidad V: Panorama general de conceptos avanzadas.** Nociones básicas de computabilidad y complejidad. Eficiencia de los programas. Aplicaciones avanzadas de análisis numérico.

**Unidad VI: Programación en Octave.** Se llevará a cabo en conjunto con todas las unidades anteriores. Luego de introducir los conceptos teóricos se requerirá al alumnado que codifique sus soluciones en Octave.

**Unidad VII: Tecnologías de la Información y las Comunicaciones para Matemática.** Lenguajes de programación y otros lenguajes informáticos. Visualización de información en problemas de matemática. Ejemplos. Gráficas de funciones. Visualización de estructuras discretas. El uso de TICs en la producción de materiales multimediales de enseñanza de la matemática. El caso Khan Academy. Otras herramientas basadas en TICs para enseñanza y aprendizaje de matemática. Asistentes de prueba. Constraint solvers.

1. **FUNDAMENTACIÓN DE LOS CONTENIDOS**

En los últimos años el uso de software se ha convertido en un elemento central en la resolución de problemas en diversas disciplinas científicas. En particular, existen problemas de análisis matemático, relevantes a la física, la ingeniería y la matemática, para los cuales no se conocen soluciones analíticas apropiadas. Sin embargo, muchos de estos problemas pueden resolverse mediante técnicas de análisis númerico (también llamado cálculo numérico), como las que se verán en este curso.

Es de suma importancia que los alumnos adquieran conocimientos de programación, para crear programas que solucionen nuevos problemas, o adaptar soluciones existentes a nuevos requerimientos.

En el curso se introducirán los conceptos subyacentes a la programación de forma independiente del lenguaje, de manera que el alumno sea capaz de adaptarse en el futuro a nuevos lenguajes sin mayores inconvenientes. De esta manera, se presentarán los conceptos generales de sistemas de computación,algoritmos y resolución de problemas mediante algoritmos. Se discutirá la codificación de algoritmos en lenguajes de programación para transformarlos en programas ejecutables, y las herramientas de soporte para llevar a cabo esta tarea (compiladores e intérpretes). Se presentarán los conceptos centrales a la programación imperativa, tipos de datos básicos comunes a la mayoría de los lenguajes de programación, y se explicará como representar datos para su manipulación eficiente mediante programas. Se discutirán los mecanismos de modularización básicos subyacentes a los lenguajes de programación modernos: la definición de funciones y procedimientos para abstraer comportamiento común. También se introducirá brevemente el paradigma funcional, y se ejercitará la definición de funciones recursivas simples.

A diferencia de los cursos básicos de programación tradicionales, en el curso se hará un uso intensivo de vectores, matrices y sus operaciones, debido a su importancia en la resolución problemas de análisis matemático. Adicionalmente, se estudiarán técnicas de visualización de datos, que también son muy útiles en este contexto.

También se presentarán brevemente las nociones básicas de computabilidad y complejidad de programas, y se mostrará la importancia del desarrollo de algoritmos eficientes.

Al final del curso se dará un panorama general sobre los usos de las TICs en la enseñanza y el aprendizaje de la matemática.

1. **ACTIVIDADES A DESARROLLAR**

La materia tendrá la modalidad de taller. Durante las clases se introducirán los conceptos teóricos de programación, y se proveerán guías de ejercicios prácticos ejercicios para codificar en Octave. La intención es que los alumnos apliquen los conceptos teóricos al desarrollo de algoritmos, y luego los codifiquen en el lenguaje de programación Octave.

Octave es un lenguaje de programación con construcciones del estado del arte tanto del paradigma imperativo como del funcional. Además, provee funcionalidades especiales para resolver problemas de análisis númerico (matrices, operaciones eficientes sobre las mismas, soporte para gráficos, numerosas funciones matemáticas predefinidas, etc.). Octave cuenta con herramientas de código abierto (compiladores, bibliotecas, entorno gráfico), que pueden ser descargadas y utilizadas gratuitamente.

Si bien Matlab es el lenguaje más utilizado para resolver problemas de análisis matemático, tanto en la industria como en la academia, su costo lo hace prohibitivo para el curso actual. De cualquier manera, Octave provee las funcionalidades más importantes de Matlab, y en particular todo lo requerido para el correcto desarrollo de este curso.

Se intentará confeccionar guías prácticas con problemas que puedan resultar de interés para los alumnos, y se fomentará la lectura de material adicional y la autoorganización de los alumnos en sus actividades.

1. **NÓMINA DE TRABAJOS PRÁCTICOS**

La evaluación de la asignatura constará de dos trabajos prácticos, de carácter obligatorio, que tienen como objetivo aplicar la teoría aprendida al desarrollo de programas para resolver problemas interesantes de análisis matemático, de complejidad media. Cada trabajo práctico contará con una defensa individual, en donde se evaluará la comprensión de los conceptos teóricos aplicados a la resolución del trabajo.

El primer trabajo práctico evaluará los temas presentados en las unidades I, II y parte de la unidad III. El segundo abarcará la parte restante de la unidad III, junto con la unidad IV. Ambos trabajos evaluarán la capacidad de los alumnos de desarrollar programas (unidad VI).

Debido a la complejidad de los temas tratados en las unidades previas, las unidades V y VII será informativas y no tendrán ejercitación práctica ni evaluación.

Además, se proveerán una serie de ejercicios adicionales (de resolución opcional) que acompañarán cada una de las clases teóricas, prácticas y laboratorios.

1. **HORARIOS DE CLASES:**

El dictado de la asignatura estará divido en dos clases de tres horas semanales, que se dictarán los días Martes de 16 a 19hs y los Viernes de 13 a 16hs.

**HORARIO DE CLASES DE CONSULTAS:** Las clases de consulta se darán semanalmente en horario a convenir con los alumnos.

1. **MODALIDAD DE EVALUACIÓN:**

**Evaluaciones Parciales:** La materia contará con dos trabajos prácticos obligatorios, cada uno con una instancia de recuperación. El plazo para la resolución de cada uno de los trabajos prácticos será de dos semanas. Cada trabajo práctico contará con una defensa oral, en la que se evaluarán los contenidos teóricos y su aplicación para la resolución de cada trabajo.

**Evaluación Final:** El examen final para alumnos regulares se llevará a cabo mediante una evaluación oral, si el número de alumnos evaluados así lo permite. Abarcará la totalidad de los contenidos de la asignatura, verificando que los alumnos hayan adquirido los conocimientos teóricos y puedan aplicarlos correctamente para resolver ejercicios prácticos y de programación.

**CONDICIONES DE REGULARIDAD:** Aprobación de los dos trabajos prácticos obligatorios y sus correspondientes defensas orales con nota igual o superior a 5.

**CONDICIONES DE PROMOCIÓN:** Aprobación de los dos trabajos prácticos obligatorios y sus correspondientes defensas orales con un promedio de 7, y nota superior a 6 en ambos.

**PROGRAMA ANALÍTICO**

1. **CONTENIDOS**

Son los indicados en el punto C (contenidos básicos).

1. **CRONOGRAMA DE CLASES Y PARCIALES**

|  |  |  |  |
| --- | --- | --- | --- |
| Semana | Día/Fecha | Clases con modalidad de taller (Teóricos/Prácticos/Laboratorios) |  Exámenes |
| 1 | 14/8 al 18/8  | Introducción. Sistemas de computación. Algoritmos. Programas. Lenguajes de programación. |  |
| 2 | 21/8 al 25/8 | Programación imperativa I: Expresiones, variables y tipos básicos, asignación. |  |
| 3 | 28/8 al 01/9 | Programación imperativa II: Composición secuencial, alternativa. |  |
| 4 | 4/9 al 8/9 | Programación imperativa III: Ciclos. |  |
| 5 | 11/9 al 15/9 | Tipos de datos. Secuencias y matrices. Operadores vectoriales y matriciales. Strings. |  |
| 6 | 18/9 al 22/9 | Gráficos.  | Presentación del Trabajo Práctico 1. |
| 7 | 25/9 al 29/9 | Entrada y salida. |  |
| 8 | 2/10 al 6/10 | Consulta y defensa Trabajo Práctico 1 | Defensa del Trabajo Práctico 1. |
| 9 | 09/10 al 13/10 | Modularización. Funciones y procedimientos. Pasaje de parámetros.  |  |
| 10 | 16/10 al 20/10 | Funciones y procedimientos II. |  |
| 11 | 23/10 al 27/10 | Introducción al paradigma funcional. Funciones recursivas. |  |
| 12 | 30/10 al 3/11 | Nociones básicas de computabilidad y complejidad. Aplicaciones avanzadas de Octave. | Presentación del Trabajo Práctico 2. |
| 13 | 6/11 al 10/11 | Uso de las TICs en la enseñanza de la matemática. |  |
| 14 | 13/11 al 17/11 | Consulta y defensa Trabajo Práctico 2. | Defensa del Trabajo Práctico 2. |

1. **BIBLIOGRAFÍA**

**Obligatoria**

*Material de clases teóricas.* Preparado por el docente.

*An Introduction to Programming in JAVA. An Interdisciplinary Approach.* R. Sedgewick, K. Wayne. 2008.

*MATLAB A Practical Introduction to Programming and Problem Solving.* S. Attaway. 2012.

**De consulta**

*Octave manual v4*.0.3. John W. Eaton, David Bateman, Søren Hauberg, Rik Wehbring. 2016. Link de descarga: https://www.gnu.org/software/octave/octave.pdf

# *Algorítmica y representación de datos.* Pierre-Claude Scholl, Jean-Pierre Peyrin. 1985.

# *Esquemas Algorítmicos Fundamentales: Secuencias e iteración*. Scholl, P. y J.-P. Peyrin. 1991.