



# UNIVERSIDAD NACIONAL DEL COMAHUE

## CENTRO UNIVERSITARIO REGIONAL ZONA ATLANTICA

**PROGRAMA DE LA ASIGNATURA:** Introducción a la Programación

**CARRERA:** Tecnicatura Universitaria en Administración de Sistemas y Software Libre

**CURSO:** Primer año1

**ORDENANZA:** 0895/12

**AÑO:** 2022

**CUATRIMESTRE:** 2°

**EQUIPO DE CATEDRA:**

Profesor:

Profesor Adjunto: Mgter. CAROLINA JUÁREZ

Ayudante de primera: Ing. ALBERTO ANDRÍA

## **1. FUNDAMENTACIÓN**

Introducción a la Programación constituye una de las asignaturas troncales de la carrera Tecnicatura en Administración de Sistemas y Software Libre, y forma parte del conjunto de materias del segundo cuatrimestre, está diseñada para formar al alumno en los conceptos básicos de programación, métodos de trabajo, técnicas y herramientas para abordar la resolución de problemas, como una manera de introducirlo en el desarrollo de su formación profesional, la cual se irá completando y profundizando en las materias correlativas del plan.

Como punto de partida, el alumno adquirirá herramientas para abordar las estrategias en la resolución de problemas, métodos de diseño de algoritmos y la introducción al lenguaje de programación Python.

Se ha elegido este lenguaje de programación porque es multiparadigma, ya que soporta orientación a objetos, programación imperativa y programación funcional.

## **2. OBJETIVOS SEGÚN PLAN DE ESTUDIOS**

El objetivo fundamental de la materia es la resolución de problemas de simple complejidad mediante el uso de la computadora.

Se pretende que el alumno adquiera conocimientos de los conceptos y terminología fundamentales en computación, y que resuelva problemas que serán básicos en asignaturas posteriores.

### **2.1 OBJETIVOS DEL PROGRAMA**

Debido a la gran brecha conceptual existente entre el enunciado del problema a resolver por computadora y su correspondiente programa, llegar a este objetivo se debe cumplir a través de las siguientes etapas:

- Adquirir habilidad en la detección de una situación de problema y el planteo de los posibles caminos de solución
- Desarrollar algoritmos como solución a problemas dados a través de un lenguaje de diseño de algoritmos orientado a un paradigma procedural.
- Traducir el algoritmo de programación diseñado a un programa escrito en el lenguaje elegido.
- Ejecutar el programa en computadora

## **3. CONTENIDOS SEGÚN PLAN DE ESTUDIOS**

Algoritmos. Resolución de problemas y clases de problemas con computadora. Programas. Estructura General de un programa. Tipos de Datos primitivos. Estructuras de control. Modularización.

## 4. CONTENIDOS ANALÍTICOS

**Unidad 1:** Resolución de Problemas y Algoritmos. Estrategias para resolver problemas. Razonamiento Lógico. Instalación de Pseudointérprete (Pseint). Diseño de Algoritmos. Análisis del algoritmo: Tratamiento de Datos, Datos de entrada y Salida. Variables, Tipos y Asignaciones. Metodología diagramas de líneas. Estructuras de control: secuenciales, alternativas, iterativas y combinadas. Ejecución y Verificación: Traza de un algoritmo.

**Unidad 2:** Lenguaje de Programación. Instalación entorno de trabajo. Introducción al Lenguaje. Estructura y conceptos básicos de un programa. Variables. tipos de datos. Tablas de verdad. Manejo de Strings. Sintaxis del lenguaje. Ámbito de influencia de las variables. Diseño de programas.

**Unidad 3:** Estructuras de Control Alternativas. Expresiones Booleanas. Estructuras de Control. Alternativas en la etapa de Análisis, Diseño, Implementación y Verificación. Alternativas Simples. Alternativas Dobles. Alternativas Múltiples. Alternativas Combinadas.

**Unidad 4:** Modularización Definición. Características. Propiedades. Encabezado de una función. Componentes del encabezado. Cuerpo y Valor retornado por una función. Parámetros Formales y Actuales. Invocación/Requerimiento de una función. Estrategia: Divide y Vencerás. Documentación de funciones. Modularización en Diagrama de Líneas. Modularización en Python.

**Unidad 5:** Estructuras de Control Repetitivas. Condiciones booleanas de la estructura repetitiva. Repetitivas en la etapa de Análisis, Diseño e Implementación. Break. Errores.

**Unidad 6:** Listas - Tuplas - Conjuntos - Diccionarios. Recorrido. Principales librerías en Python.

**Unidad 7:** Bash: introducción al lenguaje Bash y scripts. Shell de Linux.

## 5. BIBLIOGRAFÍA

### Bibliografía básica

- Apuntes de cátedra.
- Polya, George. How to solve it. Editorial: Princeton University Press. Princeton, New Jersey. 1973
- Savitch, Walter, Carrano, Frank. Java: An Introduction to Problem Solving & Programming 5th Ed. Editorial: Pearson Prentice Hall. 2008..
- Soporte en Internet:
  - <http://pedco.uncoma.edu.ar> (Centro Regional Zona Atlántica/Tecnicaturas en Informática)
  - <https://media.readthedocs.org/pdf/entrenamiento-python-basico/latest/entrenamiento-python-basico.pdf>
  - <http://fcaglp.unlp.edu.ar/~observacional/manuales/python-by-vilar.pdf>

### Bibliografía de consulta

- J. Mason, B. Leone y K. Starey. Pensar Matemáticamente. Editorial: MEC, Labor Barcelona 1992.
- M. Lucas, J.P. Peyrin y P.C. Scholl. Algorítmica y Representación de Datos. Tomo I. Editorial

Masson S.A. 1990.

- J.D. Warnier. Síntesis de Programación Lógica (Los tratamientos y sus datos). Editores Técnicos Asociados S.A. Barcelona. 1979.

## 6. PROPUESTA METODOLÓGICA

Cada unidad se desarrolla en forma teórico/práctica, y cuenta con el trabajo práctico correspondiente. Las teorías y los trabajos prácticos están disponibles en la plataforma. Se crean foros para informar novedades de la materia y un foro por unidad a fin de agrupar las consultas de conceptos y trabajos prácticos. Por otro lado se cuenta con un foro activo sólo para profesores para comunicación y documentación de las acciones de la cátedra.

Cómo práctica previa a un examen parcial, se desarrolla un simulacro con el mismo nivel del examen para que el alumno conozca la complejidad de los ejercicios a evaluar. La Cátedra realiza la correspondiente resolución, que es publicada en la plataforma para que todos los alumnos cuenten con ella.

Dado que el alumno deberá adquirir habilidades para participar en carácter colaborativo, en proyectos de software e integrar equipos de trabajos para el desarrollo de aplicaciones centradas en la web, desde la asignatura se proponen algunas instancias de entrega de trabajos prácticos y evaluativas en forma grupal.

La materia cuenta con dos exámenes parciales y dos exámenes recuperatorios.

## 7. EVALUACIÓN Y CONDICIONES DE ACREDITACIÓN

### ***Modalidad Presencial***

#### Cursado de la asignatura

El cursado de la asignatura tiene validez según la ordenanza vigente y para obtenerlo, el alumno regular debe cumplir con las siguientes condiciones:

- Cumplir con la entrega y aprobar los trabajos prácticos: grupales e individuales. Para ello, el alumno debe desarrollar satisfactoriamente cada trabajo, entregar en tiempo y forma el material que se le solicite, realizar la evaluación correspondiente al tema.
- Aprobar 40/100 el examen final de la cursada, el cual es de carácter teórico/ práctico obligatorio.

#### Acreditación de la asignatura por promoción

Tienen derecho a este régimen de acreditación los alumnos de la modalidad presencial, debiendo aprobar cada parcial con 80/100 y un examen integrador al final de la cursada de carácter teórico - práctico obligatorio.

#### Acreditación. Alumnos libres

La acreditación de alumnos libres se ajustará a lo indicado al respecto en la ordenanza vigente. En este caso, el estudiante deberá aprobar un examen escrito (o en máquina) de la parte práctica de la signatura. Posteriormente, y solo si ha cumplido con la condición previa, deberá aprobar, en las mismas condiciones, el examen escrito y oral correspondiente a la parte teórica de la materia.

### ***Modalidad Semipresencial***

#### Cursado de la asignatura

El alumno regular debe cumplir con las siguientes condiciones:

- Cumplir con la entrega y aprobar los trabajos prácticos: grupales e individuales. Para ello, el alumno debe desarrollar satisfactoriamente cada trabajo, entregar en tiempo y forma el material que se les solicite y realizar la evaluación correspondiente al tema.
- Aprobar 40/100 un exámenes al final de la cursada de carácter teórico/práctico obligatorio

#### Acreditación. Alumnos libres

La acreditación de alumnos libres se ajustará a lo indicado al respecto en la ordenanza vigente. En este caso, el estudiante deberá aprobar un examen escrito (o en máquina) de la parte práctica de la signatura. Posteriormente, y solo si ha cumplido con la condición previa, deberá aprobar, en las mismas condiciones, el examen escrito y oral correspondiente a la parte teórica de la materia.

## **8. PARCIALES, RECUPERATORIOS Y COLOQUIOS**

Cualquiera sea la modalidad de cursado elegida, los estudiantes deberán aprobar todos los Trabajos Prácticos

Deberán aprobar dos exámenes parciales

Los exámenes parciales no aprobados tienen la posibilidad de ser recuperados posteriormente a la primer fecha de examen parcial.

Los alumnos de modalidad Presencial, que obtuvieron una nota 80/100 o superior, tienen la posibilidad de promocionar luego de ser evaluados en un examen teórico/práctico al final de la cursada.

## **9. DISTRIBUCIÓN HORARIA**

- Encuentros:

Lunes 15 a 17 hs - Presencial Sala 1 - Clase Teórica - Prof. Carolina Juárez

Miércoles 17 a 19 - Virtual Sincrónico - Clase Práctica - Prof. Alberto Andría

## 10. CRONOGRAMA TENTATIVO

Cuatrimestre					
Tiempo / Unidades	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Unidad 1	X				
Unidad 2	X	X			
Unidad 3		X			
Unidad 4			X		
Unidad 5			X		
Unidad 6				X	
Unidad 7				X	

## 11. PLANIFICACIÓN DE ACTIVIDADES EXTRACURRICULARES

- Trabajos de Investigación
- Análisis de Programas de Simple Complejidad publicados en Internet

Carolina Juárez  
Profesor a Cargo

Viedma, 5 de Agosto de 2022.-