



UNIVERSIDAD NACIONAL DEL COMAHUE

COMPLEJO UNIVERSITARIO REGIONAL ZONA ATLÁNTICA Y SUR

PROGRAMA DE LA ASIGNATURA: Introducción a la Programación

CARRERA: Tecnicatura Universitaria en Administración de Sistemas y Software Libre

CURSO: Primer año

ORDENANZA: Plan: 0895/12 - Modificatorias: 1216/13

AÑO: 2026

CUATRIMESTRE: 2°

EQUIPO DE CATEDRA:

Profesor Adjunto: Mgter. Carolina Juárez

Ayudante de Primera: Tec. Manuel López Jove - Lic Karina Carrion - An. Prog. Neri Castro

1. FUNDAMENTACIÓN

Introducción a la Programación constituye una de las asignaturas troncales de la carrera 'Tecnatura en Administración de Sistemas y Software Libre', y pertenece al 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

Objetivos:

- Conocer principios generales de programación de computadoras
- Conocer la forma de resolver problemas mediante algoritmos y su traducción lenguajes de programación

2.1 OBJETIVOS DEL PROGRAMA

Objetivo General:

Debido a la brecha conceptual entre un requerimiento operacional o problema de infraestructura de servidores y su resolución automatizada por computadora, el proceso de aprendizaje se aborda a través de las siguientes etapas:

1. Detección y Análisis Operacional: Adquirir habilidad en la identificación de necesidades de administración de sistemas (mantenimiento, despliegue, monitoreo) y en el planteamiento de estrategias eficientes de solución.
2. Diseño de Algoritmos y Scripts: Desarrollar secuencias lógicas de automatización para la resolución de problemas en servidores, utilizando un lenguaje de diseño o pseudocódigo bajo el paradigma procedimental/estructurado.
3. Implementación con Herramientas de Software Libre: Traducir la solución diseñada a un script ejecutable en el lenguaje elegido (ej. Bash, Python) priorizando el uso de herramientas y entornos de Software Libre.
4. Despliegue y Validación: Ejecutar el script en el servidor objetivo, validando su correcta ejecución, seguridad y manejo de errores dentro del sistema operativo.

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. Tecnicas de recorrido e iteración /(list/dict comprehensions). Principales librerías en Python para gestión del sistema operativo, manipulación de archivos, automatización, parseo y formatos de configuración.

5. BIBLIOGRAFÍA

Toda la bibliografía obligatoria y complementaria se encuentra disponible para su descarga en la carpeta 'Bibliografía Digital' dentro de la plataforma PEDCO

Bibliografía básica

- Apuntes de cátedra.
- Ayala San Martín, G. (2020). Algoritmos y programación: mejores prácticas: (ed.). Fundación Página 4 de 7Universidad de las Américas Puebla (UDLAP).
<https://elibro.net/es/lc/uncoma/titulos/180290>
- Gahe te Díaz, J. L. Vallez Fernández, C. M. & Fernández-Pacheco Sánchez-Migallón, A. (2024). Fundamentos de programación con Python: (1 ed.). McGraw-Hill España.

<https://elibro.net/es/lc/uncoma/titulos/295333>

- Chacón Sartori, C. (2023). Principios de programación: (1 ed.). Marcombo.
<https://elibro.net/es/lc/uncoma/titulos/281682>
- Muñoz Guerrero, L. E. (II.) & Trejos Buriticá, O. I. (2021). Introducción a la programación con Python: (1 ed.). RA-MA Editorial. <https://elibro.net/es/lc/uncoma/titulos/230298>

Soporte en Internet:

- <https://www.python.org/>
- <https://docs.python.org/es/3/tutorial/>

Bibliografía de consulta

- Ramírez Marín, J. H. (2019). Fundamentos iniciales de lógica de programación I. Algoritmos en PseInt y Python: (1 ed.). D - Institución Universitaria de Envigado.
<https://elibro.net/es/lc/uncoma/titulos/226488>
- Juganaru Mathieu, M. (2015). Introducción a la programación: (ed.). Grupo Editorial Patria.
<https://elibro.net/es/lc/uncoma/titulos/39449>

6. PROPUESTA METODOLÓGICA

Se establece un régimen único de cursada para estudiantes de Sede y Nodos, bajo la modalidad de presencialidad con mediaciones tecnológicas, eliminando distinciones en el dictado y evaluación.

Se utilizan clases invertidas (flipped classroom) donde el material teórico se consulta en PEDCO antes de los encuentros híbridos sincrónicos.

Se promueve el aprendizaje basado en problemas mediante trabajos grupales colaborativos.

Interacciones Previstas:

- Docente-Estudiante: Se realizan de forma sincrónica durante las clases híbridas y de consulta, y de forma asincrónica mediante foros de novedades y mensajería de PEDCO
- Estudiante-Estudiante: En la presencialidad remota y asincronía, se fomentan mediante el uso de foros de debate por unidad, wikis colaborativas para la resolución de prácticos y la conformación de grupos de trabajo para el desarrollo de software

Cada unidad del curso se aborda de manera teórico-práctica e incluye un trabajo práctico y cuestionario correspondiente. Tanto las teorías como los trabajos prácticos están disponibles en la plataforma virtual.

Hay un total de seis cuestionarios, uno por cada unidad temática del programa de Introducción a la Programación. Solo podrán rendir el examen parcial aquellos alumnos que hayan aprobado los cuestionarios correspondientes.

Como preparación previa a un examen parcial, se lleva a cabo un simulacro con el mismo nivel de dificultad que el examen real, permitiendo así que los alumnos se familiaricen con la complejidad de los ejercicios a evaluar. La cátedra proporciona la solución del simulacro, la cual se publica en la plataforma para el acceso de todos los estudiantes.

7. EVALUACIÓN Y CONDICIONES DE ACREDITACIÓN

Acreditación de la asignatura como alumno regular:

El cursado de la asignatura tiene una validez de tres años. Para ser considerado alumno regular, se deben cumplir con los siguientes requisitos:

1. Cumplir con las fechas de entrega y obtener una calificación igual o superior a 60 en los cuestionarios y trabajos prácticos, tanto individuales como grupales. La participación en las wikis colaborativas y foros de debate de cada unidad será obligatoria para considerar el trabajo práctico como entregado.
2. Obtener una calificación igual o superior a 40 en dos parciales sincrónicos teórico-prácticos realizados durante la cursada, uno a mitad y otro al final.
3. Aprobar el examen final con una calificación igual o superior a 40.

Al cumplir con los requisitos 1 y 2, el alumno obtiene la condición de alumno regular y puede rendir el examen final para aprobar en la mesa de Examen Final según el calendario académico.

Acreditación de la asignatura por promoción:

Este régimen de acreditación aplica para los alumnos que aprueben en primera instancia:

1. Seis cuestionarios teórico-prácticos asincrónicos a través de PEDCO, con calificación igual o superior a 80 (uno por cada unidad temática del programa).
2. Trabajos prácticos obligatorios.
3. Obtener una calificación de 70/100 en el primer y segundo parcial. Modalidad Sincrónica.
4. Realizar un coloquio, modalidad presencial o remoto sincrónico, para evaluar manejo del lenguaje de la cátedra, comprensión y análisis de lo aprendido.

La aprobación por promoción de la materia se logra al cumplir con los cuatro requisitos mencionados anteriormente.

Acreditación de la asignatura como alumno libre:

La acreditación como alumno libre se rige por la ordenanza vigente. Para acceder a esta modalidad, además se debe cumplir con la siguiente condición previa:

- Aprobar, 15 días antes de la fecha del examen final, los seis cuestionarios teóricos/prácticos obligatorios disponibles en PEDCO; con calificación igual o superior a 60.

Luego, y solo si se ha cumplido con la condición previa, se deberá aprobar el examen final teórico práctico, el mismo será presencial en la sede correspondiente.

8. PARCIALES, RECUPERATORIOS Y COLOQUIOS

Los estudiantes deben aprobar todos los cuestionarios asincrónicos y Trabajos Prácticos Obligatorios, así como también dos exámenes parciales.

Los cuestionarios tienen dos instancias de recuperación, con un tiempo mínimo de espera de 72 hs entre fechas. de reintento.

En caso de no aprobar los exámenes parciales, presencial o remoto sincrónico, en su primera instancia, se brinda una oportunidad de recuperarlos al finalizar el cuatrimestre.

Los coloquios se llevan a cabo de forma presencial en la sede correspondiente, excepcionalmente la cátedra podrá autorizar el coloquio remoto sincrónico fuera de sede, con la condición que el alumno utilice tecnología de cámara y micrófono, siempre acompañados de un documento que verifique la identificación del alumno.

9. DISTRIBUCIÓN HORARIA

Modalidad	Minutos *	Porcentaje
Presencialidad en el establecimiento	120	25%
Presencialidad remota	120	25%
EAD Asincronica - PEDCO	120	25%
Modalidad de presencialidad híbrida/combinada	120	25%

* Minutos de dictado semanal

Observaciones

Lunes 16 a 18 hs - Teoría- (Sala 1 informática) - modalidad híbrida sincrónica- Carolina Juárez - Karina Carrión

Miércoles de 16 a 18 hs - Práctica- (Sala 2 Informática) - modalidad híbrida sincrónica- Ayudante Manuel Lopez Jove - Neri Castro

Jueves de 16 a 17 hs - Consulta - modalidad virtual - Ayudante Neri Castro

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	

Observaciones

Encuentros: Lunes y Miércoles 16 a 18hs híbridos (alumnos presencial y virtual sincrónico)

- Lunes: Clase Teórica
- Miércoles: Clase Práctica
- Jueves: Clase Consulta 16 a 17 hs

11. PLANIFICACIÓN DE ACTIVIDADES EXTRACURRICULARES

Trabajos de investigación temática: Tendencias y escalabilidad en el ecosistema Open Source

- Investigación sobre las tendencias actuales, automatización y escalabilidad de la infraestructura basada en Linux, Python y herramientas de Software Libre (Ansible, Docker, OpenStack) en el mercado laboral profesional de administración de sistemas y DevOps.

Análisis de código abierto en internet: Estudio de repositorios públicos

- Estudio de repositorios públicos de Software Libre (GitHub, GitLab, Salsa Debian) para

identificar el uso práctico de estructuras de datos y control en scripts reales de automatización, herramientas CLI y módulos de administración de sistemas de complejidad simple a intermedia.

Foros de resolución de retos lógicos y automatización (PEDCO)

- Espacios asincrónicos en PEDCO para debatir estrategias de resolución de algoritmos orientados a la administración de sistemas, procesamiento de logs, parseo de configuraciones y diagnóstico de fallas, fomentando la interacción colaborativa entre estudiantes de todas las sedes.

Talleres de administración y scripting colaborativo

- Encuentros sincrónicos de resolución colaborativa de problemas de infraestructura y automatización en equipo, utilizando entornos Linux y metodologías de trabajo colaborativo (Control de versiones con Git, documentación técnica libre y revisión de código) esenciales en el perfil SysAdmin/DevOps.

Carolina Juárez
Profesor Adjunto

Viedma, 10 de Agosto de 2026.-