Ministerio de Cultura y Educación **(Programa del año 2026)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
PROGRAMACION II	TEC.REDES COMP.	12/15	2026	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
BERON, MARIO MARCELO	Prof. Responsable	P.Adj Exc	40 Hs
SANCHEZ, HECTOR ENRIQUE	Responsable de Práctico	JTP Exc	40 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
2 Hs	Hs	Hs	4 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
11/03/2026	23/06/2026	15	90

IV - Fundamentación

Conocer y aprovechar al máximo las características principales que posee un sistema operativo es una tarea esencial para el técnico en redes. Entre una de estas características se encuentran los intérpretes de comandos que el sistema provee. Estos intérpretes proporcionan una serie de comandos que permiten la realización de tareas de administración de recursos. En general, estos comandos se pueden combinar con el propósito de realizar tareas complejas en pocas líneas de código. Estas líneas de código pueden ser almacenadas en un archivo el cual a posteriori se puede ejecutar. Este tipo de programas son conocidos en la jerga como scripts del intérprete de comandos. Debido a la relevancia que tienen este tipo de programas para el técnico en redes, en esta materia se incluye una unidad cuyo principal objetivo es que el alumno conozca en profundidad los fundamentos y las distintas estrategias que se utilizan para programar scripts. Además se explican las bases para que el alumno pueda trasladar los conceptos brindados para la construcción de otros tipos de scripts. Por otra parte, con el propósito de incrementar los conceptos referentes a programación, en el presente programa se incluye una unidad que consiste en la definición e implementación de tipos de datos abstractos usando como lenguaje de estudio al lenguaje C. En esta unidad se pretende hacer énfasis en la construcción y manipulación de diversas estructuras de datos que son ampliamente utilizadas en los sistemas de software. Actualmente, el lenguaje C es muy usado para el desarrollo de software de base como así también para la interacción con diversos dispositivos. Por esta razón, ampliar los conocimientos de programación usando C como lenguaje base es relevante para el técnico en redes. El Paradigma Imperativo no es el único, existen otros como por ejemplo el Orientado a Objetos que permiten concebir a la programación desde otras perspectivas. Actualmente, el Paradigma Orientado a Objetos ha alcanzado una amplia popularidad debido a las ventajas que proporciona para la construcción de software. Particularmente, este paradigma se adapta muy bien para el desarrollo de aplicaciones de red. Por esta razón los lenguajes que son orientado a objetos o tienen soporte orientado a objeto son ampliamente usados para el desarrollo de estas clases de aplicaciones. Python es uno de tales lenguajes que tiene la particularidad de ser muy usado en el contexto

académico y empresarial por las facilidades que proporciona para la construcción de aplicaciones y por la rapidez y robustez con la que las mismas se construyen. Por las razones antes mencionadas, la enseñanza del Paradigma Orientado a Objetos y de Python se considera esencial en Programación II. Es importante notar que Programación II presenta temáticas de propósito general, remarcando como las mismas pueden ser aplicadas en el contexto de las redes, y prepara el camino para que el alumno pueda asimilar los tópicos de cursos superiores e introducirse en el mercado laboral sin mayores inconvenientes. Para finalizar, cabe destacar que el programa aquí presentado cubre los contenidos necesarios descritos en la ordenanza 024-08, Anexo II.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso el alumno deberá:
* Programar Scripts.
* Conocer técnicas avanzadas de programación.
* Conocer las bases de la Programación Orientada a Objetos.
* Construir programas utilizando el Soporte Imperativo y Orientado a objetos del lenguaje de programación Python.

VI - Contenidos

Unidad I: Shell Script con Bash
Introducción a la Programación de Script. Conceptos básicos de Bash. Construcción de Shell Script Básicos. Uso de Múltiples Comandos. Creación de un Archivo de Script. Mensajes. Variables. Redirección de la Entrada y la Salida. Pipes. Operaciones Matemáticas. Finalización de un Script. Uso de Comandos Estructurados. If-then. If-then-else. Ifs anidados. El comando test. Condiciones compuestas. Características Avanzadas del if. El comando case. Más Comandos Estructurados. El comando for. El comando for al estilo C. El comando while. El comando until. Ciclos anidados. Iteraciones sobre archivos. Entrada de Usuario. Parámetros en la Línea de Comandos. Variables de Parámetro Especiales. Opciones. Estandarización de Opciones. Entrada de Usuario. Presentación de Datos. Entrada-Salida. Creación de Funciones. Funciones de Script Básicas. Retorno de un Valor. Uso de Variables en Funciones. Arreglos y Funciones. Funciones Recursivas. Creación de una Librería. Uso de Funciones en la Línea de Comandos.

Unidad II: Tipos de Datos Abstractos
Introducción. Tipos de Datos Abstractos. Implementación de Tipos de Datos Abstractos. Aplicación de los conceptos antes mencionados usando el lenguaje de programación C.

Unidad III: Python Imperativo
Lenguaje de Programación Python. Historia. Características y Filosofía. El Intérprete de Python. Tipos Básicos: Enteros, Booleanos, Flotantes, Complejos, Decimal, String. Tipos de Datos Colección: Listas, Tuplas, Conjuntos, Diccionarios. Sentencias: Secuencia, Selección, Iteración, Excepciones. Funciones: Definición, Invocación, Pasaje de Parámetros.

Unidad IV: Paradigma de Programación Orientada a Objeto - Python Orientado a Objetos
Introducción. Génesis de la Programación Orientada a Objetos: La Vía de la Simulación. La Vía de la Ingeniería de Software. Abstracción de Datos e Información Oculta. El Concepto de Objeto. El Encapsulamiento: Propiedad Fundamental. Mensajes. Objetos en Programación Orientada a Objetos: Instancias y Clases. Clases e Instancias. Definición de Clases. Creación de Clases. Requisitos Iniciales. Definición de la Estructura. Definición del Comportamiento. Constructores. Métodos de Instancia. Importación de Clases. Reglas de Accesibilidad a una Clase. Reglas de Acceso a Variables y Métodos de Instancia. Composición en la Definición de una Clase. Métodos y Variables de Clase. Jerarquías de Clases. El Mecanismo de la Herencia. Algoritmo de Búsqueda de Métodos. Superposición de Métodos y Variables. Modificadores y Re-definición de Métodos. Clases y Constructores. Creación de Clases por medio de la Herencia. Compatibilidad entre Clases y Subclases. Polimorfismo. Excepciones. Aplicación de los conceptos previamente mencionados usando el lenguaje de programación Python.

Unidad V: Conceptos Avanzados de Python
Interfaces con el Sistema Operativo. Manejo de Archivos. Depuración y Pruebas de Programas con Python.

VII - Plan de Trabajos Prácticos

Unidad I: Shell Script con Bash

Construcción de: i) Scripts básicos para la familiarización con los comandos de bash; ii) Scripts que utilicen los recursos del sistema operativo; iii) Scripts que permitan realizar la instalación de aplicaciones.

Unidad II: Tipos de Datos Abstarctos
Especificación e Implementación de Tipos de Datos Abstractos comúnmente usados en programación.

Unidad III: Lenguaje de Programación Python - Python Imperativo
Realización de ejercicios que impliquen el uso de los tipos y estructuras de control proporcionados por el soporte imperativo de Python.

Unidad IV: Programación Orientada a Objetos
Realización de ejercicios que impliquen: i) La definición de clases; ii) El uso de herencia, composición y polimorfismo.

Unidad V: Aplicaciones de Script con Python
Realización de scripts que permitan: Copiar, mover archivos en diferentes directorios, borrar archivos y directorios, crear árboles de directorios, leer y escribir archivos. Elaborar casos de prueba de los scripts desarrollados.

Laboratorios

Laboratorio 1: Desarrollo de scripts usando bash.

Laboratorio 2: Implementación y corrida de programas imperativos y orientados a objetos usando el lenguaje de programación Python.

Laboratorio 3: Implementación y corrida de scripts que trabajen con archivos e interactúen con el sistema operativo.

VIII - Regimen de Aprobación

Condiciones de Regularización:
a) Aprobar un parcial o sus respectivas recuperaciones con una nota mayor que o igual que 6 (seis). Se otorgarán dos recuperaciones a todos los alumnos.

b) Aprobar un práctico de laboratorio indicado por la cátedra con una nota mayor o igual que 6 (seis). Se otorgarán dos recuperaciones del práctico de laboratorio a todos los alumnos.

Condiciones de Aprobación:
a) Por promoción, para lo cual el alumno debe:
a.1) Aprobar un parcial o sus respectivas recuperaciones con una nota mayor o igual que 7. Se otorgarán dos recuperaciones a todos los alumnos.
a.2) Aprobar un práctico de laboratorio indicado por la cátedra con una nota mayor o igual que 7 (siete). Se otorgarán dos recuperaciones del práctico de laboratorio a todos los alumnos.
a.3) Aprobar una evaluación teórica integradora con una nota mayor o igual que 7. Se otorgarán dos recuperaciones de esta evaluación a todos los alumnos.
b) Por examen final.

Alumnos Libres: Según la reglamentación vigente (Art. 27 de Ord. 13/03 CS).

IX - Bibliografía Básica

[1] Richar Blum. Linux Command Line and Shell Scripting. Wile Publishing Inc. ISBN: 978-0-470-25128-7. 2008.
[2] Material de la Materia Programación Orientada a Objetos dictada en la carrera Licenciatura e Ingeniería en Ciencias de la Computación. Universidade do Minho. Profesor Responsable: Mario Martins.
[3] Guido van Rossum. El Tutorial de Python. Editor: Fred L. Drake. Octubre de 2017.
[4] Brian W. Kernighan, Dennis Ritchie. El lenguaje de programación C (Segunda Edición - Español). Prentice Hall. 1995. ISBN-13: 978-9688802052.
[5] Material proporcionado por la cátedra.

X - Bibliografia Complementaria

[1] David Fiedler, Bruce Hunter. Unix System V Release 4.
[2] Administration, Second Edition. Editorial: SAMS. ISBN: 0-672-22810-6. 1991.
[3] Mangus Lie Hetlanfd. Beginning Python from Novice to Professional. ISBN: 1-59059-519-X. 2005.
[4] John Goerzen. Fundation of Python Network Programming. Editorial: Apress. ISBN: 1-59059-371-5. 2004.
[5] Carl Albing, J. P. Vossen, Cameron Newhan. Bash Cookbook. Editorial: O'Reilly. ISBN 10: 0-596-52678-4. Mayo 2007.
[6] Chet Ramey, Brian Fox. Bash Reference Manual. Publicado por: Free Software Foundations. Diciembre de 2009.
[7] Summerfield, Mark. Programming in Python 3: A Complete Introduction to the Python Language. Second Edition. 2009.

XI - Resumen de Objetivos

*Programar Scripts.
*Conocer técnicas avanzadas de programación.
*Conocer las bases de la Programación Orientada a Objetos.
*Construir programas utilizando el Soporte Imperativo y Orientado a objetos del lenguaje de programación Python.

XII - Resumen del Programa

Unidad I: Shell Scripts con Bash.
Unidad II: Tipos de Datos Abstractos.
Unidad III: Python Imperativo.
Unidad IV: Programación Orientada a Objetos - Python Orientado a Objetos.
Unidad V: Conceptos Avanzados de Python.

XIII - Imprevistos

Serán resueltos por la cátedra cuando los mismos se presenten. Contacto:
*Dr. Mario Marcelo Berón
*Correo electrónico: mberon@email.unsl.edu.ar

XIV - Otros

-