

Programa de Asignatura

00 Código:	F	C	Asign
	T4	17	25

01. Facultad: Tecnología Informática / **Carrera:** Analista programador – Modalidad a Distancia

02. Asignatura: T4-17- 18 - PROGRAMACIÓN II

03. Año lectivo: 2022

04. Año de Cursada: 3°

05. Cuatrimestre: 1°

06. Carga horaria semanal: 5

07. Créditos:

08. Equipo Docente

Profesor Titular: Weingand Gastón

Eje socioprofesional: Los modelos como representación abstracta de la realidad.
(Análisis y Lenguajes)

Coordinador de eje: Ing. Vilaboa Pablo

Eje Epistémico: Teoría de Sistemas y Lenguajes de Programación.

09. Asignaturas correlativas previas y posteriores:

Correlativas previas: Programación Orientada a Objetos

Correlativas posteriores: No posee

10. Fundamentación:

a- Aporte específico de la asignatura a la formación académico profesional

La formación de un buen profesional abarca las habilidades metódicas y críticas que sirvan de base para comprender los preceptos básicos de la algoritmia. A lo largo de los años los paradigmas fueron cambiando posicionándose en la actualidad como un pilar fundamental a la hora de bosquejar la arquitectura de un programa. Esta materia aporta los conocimientos básicos de clases y desarrolla los fundamentos principales de la programación orientada a objetos como la herencia, la encapsulación y la abstracción. También se incorporan nociones elementales de la programación funcional como el cómputo determinista, los efectos colaterales, las funciones de orden superior y la recursividad. Incorporando lineamientos específicos del lenguaje

de programación que le permita al futuro profesional representar virtualmente los elementos observados de la vida real.

Prepara al alumno en las técnicas avanzadas de programación fomentando el uso de buenas prácticas entrenándolo en el desarrollo de sistemas y la depuración de errores, facilitándole los elementos esenciales para lograr una óptima gestión del código fuente.

b- El /los marco/s conceptual/es que sustenta/n el enfoque de la asignatura

La asignatura aborda fundamentalmente el modelo orientado a objetos como metodología que optimiza la forma de producir software. Además de este paradigma, se hace una exploración de diferentes paradigmas previamente aprendidos como el estructurado o procedimental y se introducen conceptos del paradigma funcional y heurísticos. Todas estas visiones permiten enriquecer las herramientas y metodologías que todo profesional debe poseer para producir software de calidad.

c- La articulación de la asignatura con asignaturas previas y posteriores

Articulación Vertical:

Las asignaturas Programación Estructurada, Programación I y Programación orientada a objetos integran los contenidos sobre el concepto de programa, la introducción a la programación, las estructuras de datos y los algoritmos más importantes para el ordenamiento, clasificación y búsqueda de datos, los que son utilizados por esta asignatura, además de la teoría de objetos, tan importante para el diseño de sistemas complejos.

Le provee a Metodología de Desarrollo de Sistemas II una visión práctica y teórica sobre orientación a objetos lo cual facilita la tarea de aprendizaje de las formas de modelado estándar desarrolladas en la asignatura con la que articula.

A Lenguajes de Última Generación le provee las herramientas de programación para encarar los aspectos relacionados con el acceso a datos desde un programa informático y la aplicación de técnicas de programación refinadas.

Articulación Horizontal:

La articulación con la asignatura Metodología de Desarrollo de Sistemas I permite que se puedan trabajar los aspectos referidos a la documentación y los métodos para desarrollar software.

Programación II aporta además a la asignatura Trabajo Final. Los contenidos del modelo orientado a objetos, la programación orientada a objetos, el aprovechamiento del acceso a dispositivos remotos, la programación funcional y el uso de sistemas heurísticos para la resolución de problemas complejos.

11. Competencias / sub-competencias y resultados de aprendizaje a las que tributa la asignatura:

Competencias del perfil	Subcompetencia	Nivel de dominio de la subcompetencia	Resultados de aprendizaje
2- Colaborar en el análisis, planificación, programación, implementación y mantenimiento de desarrollo de software de forma efectiva.	2-1- Capacidad para seleccionar el paradigma de programación más adecuado para enfocar los procesos de construcción de un software	3	T4-17-18-2-1-3-RA1: [Aplica] + [los paradigmas de programación con lenguaje Python]+ [para resolver problemáticas de sistemas de información general] + [demostrando el conocimiento de la sintaxis del lenguaje conforme a guía de codificación y buenas prácticas estipuladas en la Guía de estilo para el código Python “PEP 8”]
2- Colaborar en la análisis, planificación, programación, implementación y mantenimiento de desarrollo de software de forma efectiva.	2-3- Capacidad para reutilizar y adaptar rutinas y programas existentes, integrándolas con código propio.	3	T4-17-18-2-3-3-RA2: [Aplica] + [librerías de Python relacionadas con seguridad informática]+ [para resolver problemáticas de seguridad de sistemas de información general] + [demostrando el conocimiento de la sintaxis del lenguaje conforme a guía de codificación y buenas prácticas estipuladas en la Guía de estilo para el código Python “PEP 8”] T4-17-18-2-3-3-RA3: [Comprende]+ [como estructurar software de propósito general]+ [aplicando conceptos de definición de arquitectura, cohesión y acoplamiento] + [vinculando conceptos de análisis, diseño y programación]

12. Unidades de desarrollo de los contenidos:

Unidad 1: INTRODUCCION A PYTHON.

El Software.

Entorno integrado de desarrollo. Programas Vs. Proyectos. Aspectos distintivos de la programación orientada a objetos Vs. La programación estructurada. Vs. Programación funcional. Ventajas y desventajas.

Abstracción de datos. Objetos y Clases. Herencia. Herencia múltiple. Polimorfismo. Lenguajes orientados a objetos/funcionales. El lenguaje Python. Historia y evolución del lenguaje.

Tiempo: 10 hs

Unidad 2: CLASES Y OBJETOS.

Sentencias condicionales e iteradores. Desarrollo y utilización de clases y objetos. Variables de clases e instancias. Constructores y destructores. Herencia simple y herencia múltiple. Polimorfismo. Encapsulamiento. Clases de “nuevo-estilo”. Métodos especiales. Tipos en Python.

Tiempo: 20 hs

Unidad 3: PROGRAMACIÓN FUNCIONAL Y EXCEPCIONES.

Funciones de orden superior. Iteraciones sobre listas. Funciones lambda. Listas,

diccionarios y cadenas. Generadores y decoradores. Módulos y paquetes para excepciones. Ejemplos de su utilización. Integración con clases y objetos.
Tiempo: 15 hs.

Unidad 4: ARCHIVOS Y EXPRESIONES REGULARES.

Entrada y salida estándar. Archivos: Lectura y escritura. Cierre y tratamiento de errores en la gestión de archivos. Expresiones regulares, patrones. Módulo re.
Tiempo: 15 hs.

Unidad 5: LIBRERÍAS E INTRODUCCIÓN AL MACHINE LEARNING.

Descripción y ejemplos de algunas de las librerías más utilizadas: Pygame, RE, Collections, NumPy, SQLAlchemy, Request, Pillow, etc.
Introducción al machine learning. Introducción al Deep learning. Algoritmos de redes neuronales. Práctica de generación de red neuronal sobre placa microbit y/o simulación. Framework Anaconda y miniconda, descripción y uso práctico.
Tiempo: 20 hs.

13. Metodología:

a. Encuadre metodológico de la asignatura

Las actividades académicas tendrán una sección para la teoría y otra para la práctica, brindadas por el docente a través de videoconferencias semanales o quincenales. La forma de desarrollar los encuentros sincrónicos permitirá la articulación de los conocimientos teóricos y prácticos. Los encuentros poseen dos momentos marcados, el de abordaje metodológico hacia el marco teórico y el de desarrollo e implementación de la práctica.

Se tendrá especial atención en la participación activa de los alumnos, para poder rescatar la elaboración conceptual que construyan de los temas tratados.

Se aplicarán diversas técnicas y metodologías en el plano teórico y práctico entre las cuales podemos mencionar:

Teoría: 40% del tiempo de cada encuentro. Exposición docente. Diálogo con los alumnos. Puesta en común. Conclusiones grupales. Ínter consultas. Revisión de temas. Relación de temas nuevos con conocimientos adquiridos de la asignatura y de otras asignaturas relacionadas.

Práctica: 60% del tiempo de cada encuentro. La ejercitación deberá reflejar los conceptos transmitidos en la sección para teoría, más los conocimientos de encuentros anteriores. Se propondrán ejercicios cuya resolución será grupal en algunos casos e individual en otros. Los alumnos prepararán trabajos de investigación que incluirán la exposición oral de los mismos.

Los medios audiovisuales ocuparán un importante lugar en el desarrollo de los encuentros para lograr un mayor estímulo en el alumno y de esa forma fijar mejor los contenidos directrices.

Los textos de actualidad y las notas periodísticas sobre tecnología de punta, serán propuestos para ser leídos y tratados durante los encuentros.

Se fomentará la participación de los alumnos en la formulación de soluciones a los problemas de la práctica profesional.

Se desarrollarán ejercicios que denoten un alto grado de relación grupal para aprovechar la sinergia de la relación interpersonal.

b. Contextos de desarrollo de las actividades formativas

Las actividades formativas se llevarán a cabo a través de la plataforma de acceso remoto UAI Ultra. A su vez, se ofrecerán encuentros sincrónicos a través de videoconferencias y actividades asincrónicas en la plataforma mencionada.

Actividades en plataforma digital:

La asignatura contará con un aula virtual en el LMS institucional, UAIOonline Ultra, donde podrán acceder a los contenidos analíticos y actividades de aprendizaje sincrónicas y asincrónicas alineadas a los resultados de aprendizajes.

El acceso a las aulas virtuales se realizará desde la WEB APP miUAI con las credenciales institucionales y será tutorizada por el docente a cargo.

Se propone como complemento de estrategias de aprendizaje, el empleo de tecnologías digitales que favorezcan el desarrollo de actividades (Individuales y grupales) participativas y colaborativas que promuevan la autonomía del estudiante en el proceso de enseñanza sobre la base de siguientes actividades:

- Desarrollo de los contenidos: Bibliografía, presentaciones, tutoriales, videos, lecturas complementarias.
- Resolución de problemas: Indagación de respuestas sobre problemas simulados a fin de crear una solución.
- Trabajo por proyectos: Indagación de respuestas sobre preguntas o problemas del mundo real a fin de crear una solución.
- Foros: Discusión colaborativa sobre temas de interés de la asignatura.

c. Recursos didácticos

Se utilizarán los siguientes recursos didácticos:

- a. Guía de orientación por unidad
- b. Guía de revisión conceptual
- c. Guía de trabajos prácticos
- d. Ejercicios Integradores
- e. Material multimedial
- f. Código de programación
- g. Foros

- h. Docs de Microsoft
- i. Learn de Microsoft
- j. IDE Visual Studio Code

14. Procedimiento de evaluación y criterios de promoción:

a. Modalidad y Criterios de Evaluación

La evaluación se sustenta en la participación durante los encuentros, la producción domiciliaria (TP's y trabajos de indagación) y los parciales. Para lograr una evaluación continua se trabajará con rúbricas que permitirán observar a lo largo del cuatrimestre el grado de apropiación de los temas tratados en la asignatura.

A continuación, se detallan los criterios aplicados para cada instrumento:

Parciales.

Los exámenes parciales serán como mínimo dos, cada uno con articulación teórica y práctica. Ambos son obligatorios, individuales y escritos. El primero de ellos se efectuará entre la quinta y séptima semana del cuatrimestre y el segundo entre la semana doce y catorce. Los parciales deberán tener un 60% de contenido correcto para estar aprobados. En este último caso la nota deberá quedar establecida entre 4 (cuatro) y 10 (diez) puntos. Si el alumno no se presenta a un examen parcial se calificará con una nota insuficiente.

Recuperatorios.

Los exámenes recuperatorios se desarrollarán a razón de uno por cada parcial con teoría y práctica. Se podrá optar por tomar un recuperatorio integral -en aquellos casos que se justifique- el cual será equivalente al alcance de los recuperatorios individuales. Recuperarán los parciales que correspondan, aquellos alumnos que obtuvieron una calificación inferior a 4 (cuatro) puntos en alguno de sus exámenes parciales. Los recuperatorios son individuales y por escrito. El docente dispondrá la fecha para realizar esta actividad que podrá ser con posterioridad a cada uno de los parciales o en una fecha común para todos. Las notas de los exámenes recuperatorios no reemplazan las calificaciones insuficientes obtenidas en los exámenes regulares que se están recuperando, las mismas serán tomadas como parte de las calificaciones que intervienen en el promedio.

Aspectos comunes a parciales y recuperatorios.

La confección de los exámenes parciales y recuperatorios contemplarán diversas técnicas para poder observar desde distintos ángulos el objeto de evaluación. Se intercalarán preguntas a desarrollar, del tipo múltiples casos, guiadas y lógicas. Se evaluará la capacidad de razonamiento del alumno y el marco teórico que lo sustenta.

Trabajos prácticos.

a. Elaboración de una carpeta de ejercicios prácticos. Se desarrollará de forma individual y el docente podrá seleccionar ejercicios para que se realicen en forma grupal. La carpeta

de trabajos prácticos a resolver es la **“Guía de Trabajos Prácticos de la Asignatura”** y la **“Guía de Revisión Conceptual”**. La primera consta de ejercicios, estos están agrupados de acuerdo con las unidades de la asignatura. Dentro de cada unidad el conjunto de ejercicios presenta problemas cuya resolución implica la aplicación de técnicas de programación y permiten abordar el problema tratado desde diferentes ópticas y con niveles graduales de dificultad. El docente podrá tomar algunos ejercicios emblemáticos del conjunto y desarrollarlos en el aula con los alumnos a modo de ejemplo. El docente sugerirá la resolución de cada conjunto de ejercicios en la medida que culmine con el desarrollo de la unidad correspondiente a la asignatura. La segunda guía son preguntas de autoevaluación sobre los temas abordados en cada unidad. El docente indicará la fecha y forma de entrega dentro del cuatrimestre.

b. Trabajo de indagación. Los trabajos de indagación estarán referidos a temas de la asignatura. Se realizarán en forma grupal y el número de sus integrantes será de cuatro a seis alumnos. La entrega de estos será fijada por el docente dentro del plazo en que se desarrolla el cuatrimestre.

Ejercicios integradores.

Los ejercicios integradores llevará una nota de calificación entre 1 (uno) y 10 (diez) puntos la cual se puede promediar con la nota de concepto.

Trabajo de indagación.

Se desarrollará un trabajo práctico de indagación bibliográfica obligatorio referido a temas de la asignatura y deberá ser expuesto por los alumnos. El trabajo y la exposición será grupal y la evaluación de este estará conformada por un promedio de notas que involucren la producción grupal y el desempeño individual. Los aspectos a tener en cuenta en dicha evaluación son: calidad de la investigación, preparación y clasificación del material, producción del material, calidad de la exposición oral, dominio del lenguaje específico, aprovechamiento de los medios y administración del tiempo.

Nota conceptual.

Una nota conceptual basada en la participación del alumno durante los encuentros, el interés demostrado en la asignatura y el empeño puesto de manifiesto en las actividades de estudio.

b. Evidencias

Las evidencias que darán cuenta del logro de los objetivos planteados en el curso son:

- a. Resolución de la guía de trabajos prácticos.
- b. Resolución de la guía de abordaje conceptual
- c. Trabajo de indagación.
- d. Rúbricas.
- e. Foros
- f. Resolución de los ejercicios integradores.

g. Parciales

c. Requisitos de aprobación de la asignatura

Para aprobar la asignatura y acceder al examen final o coloquio el alumno deberá tener sus dos exámenes parciales y el trabajo práctico aprobado, más la entrega de la carpeta con la resolución de los ejercicios de programación.

El alumno deberá tener como mínimo cuatro notas para poder promediar sin contar los recuperatorios de parciales.

Para aprobar la asignatura los estudiantes deberán cumplir los siguientes requerimientos:

1. Asistencia: Los estudiantes deben contar con una asistencia mínima del 70%, que se complementa entre la participación de los encuentros y la realización de las actividades solicitadas.
2. Aprobar mediante alguna de estas modalidades:
 - a. Evaluación final “integradora coloquial”: accederán a este régimen de evaluación aquellos alumnos cuyo promedio de cursada se encuentre comprendido entre 6 y 10 puntos. Los alumnos podrán presentarse a la mesa examinadora en grupos de no más de tres personas. La evaluación se realizará en forma individual, debiendo demostrar el dominio de la asignatura como unidad y la capacidad de asociarla con otras materias del plan de estudio ya cursadas.
 - b. Examen final: acceden a este régimen de evaluación aquellos alumnos que han obtenido durante su cursada un promedio comprendido entre 4 y 5,99 puntos. El alumno se presentará en forma individual ante un tribunal examinador, el que interrogará sobre el programa de la asignatura, debiendo demostrar su capacidad de integrarla y relacionarla con otras asignaturas del plan de estudio.

Examen recuperatorio: En las carreras de grado, aquellos alumnos cuyo promedio sea inferior o igual a 3,99 puntos (obtenidos en las dos o tres evaluaciones parciales) y tengan una asistencia igual o mayor al 50%, deberán rendir un Examen Recuperatorio de Materia en la misma fecha que se indica para el Examen Final.

Habiendo aprobado el Examen Recuperatorio de la Materia, con nota mínima de 4 (cuatro) puntos, estarán en condiciones de acceder al Examen Final en el siguiente llamado, con la previa inscripción realizada 48 horas hábiles antes del mismo.

Para rendir Examen Recuperatorio de la Materia no es necesario presentar Permiso de Examen, pero deberá registrarse con no menos de 48 horas hábiles de anticipación. Los alumnos de la modalidad a distancia deberán registrarse con no menos 72 horas hábiles de anticipación.

El derecho a rendir Examen Recuperatorio de la Materia tendrá validez por dos años o dos presentaciones a Exámenes Recuperatorios de la Materia, lo que se produzca primero.

Objetivos de la evaluación.

Validar que el alumno alcanzó los resultados de aprendizaje con el nivel de dominio planteado para la asignatura y el grado de internalización de los contenidos.
Verificar los errores y cambios conceptuales en las estructuras cognitivas de los alumnos.
Garantizar la transferencia del conocimiento.
Reconocer la integración de aprendizajes significativos.

Contenidos de la evaluación.

Se tendrán en cuenta los contenidos conceptuales, procedimentales y actitudinales que denoten la apropiación de los resultados de aprendizaje esperados.

15. Bibliografía

Obligatoria:

Raúl González Duque - “Python para todos”. 1ª Ed. España. Creative Commons 2015.
David Beazley, Brian K. Jones - “Python Cookbook” 3ª Ed. USA. OReilly Media 2013.
Eric Matthes - “Python Crash Course” 2ª Ed. USA. No Starch Press 2019.
Martino Savia, Cathy Wang - “Python Tools for Visual Studio”. 1ª UK. Packt 2014.
Aurélien Géron – “Hands-On Machine Learning with Scikit-Learn & TensorFlow”. 1ª Ed USA. OReilly Media 2017

Ampliatoria:

Rafael Caballero, Enrique Martín y Adrián Riesco - “BIG DATA con PYTHON. Recolección, almacenamiento y proceso” 1ª Ed. España. RC Libros 2018.
Zed A. Shaw - “Learn More Python 3 The Hard Way” 1ª Ed. USA. Addison-Wesley Professional 2017
Luciano Ramalho - “Fluent Python”. 1ª Ed. USA. OReilly Media 2015.