

<b>CARRERA:</b> Ingeniería en Informática	<b>CURSO LECTIVO:</b> 2024
<b>ASIGNATURA:</b> Informática General	<b>CURSO:</b> 1° año – 1° semestre
<b>DURACIÓN:</b> Semestral	<b>Hs. TOTALES:</b> 80 Hs. Reloj totales
<b>SEMANAS:</b> 16	<b>Hs. TEÓRICAS:</b> 32 Hs. Reloj totales <b>Hs. PRÁCTICAS:</b> 48 Hs. Reloj totales

**COORDINADOR:** Trigila, Mariano.

**PROFESOR TITULAR:** Hamkalo, José.

**PROFESOR PROTITULAR:** Trigila, Mariano / Mariño, Hernán.

**PROFESOR ADJUNTO:** De Arambarri, Alejandra / Acquesta, Alejandro.

**PROFESOR ASISTENTE:** Pérez Catuogno, Nahuel / Dibarbora, Carlos.

## 1. OBJETIVOS DE LA ASIGNATURA

Que el alumno logre:

- Adquirir los conceptos fundamentales de la programación estructurada y modular.
- Interpretar la redacción (especificación) de un problema.
- Resolver problemas aplicando conceptos de lógica y conceptos de algoritmia.
- Implementar la resolución de problemas en un lenguaje de programación de alto nivel.
- Ejecutar, validar y verificar el programa desarrollado en el lenguaje de programación utilizando una computadora y el software específico.
- Adquirir habilidades de programación orientada a datos.
- Aplicar herramientas de búsqueda y autoaprendizaje.

## 2. COMPETENCIAS

Eje / Enunciado	Bajo	Medio	Alto
1. Especificación, proyecto y desarrollo de sistemas de información.			
2. Especificación, proyecto y desarrollo de sistemas de comunicación de datos.			
3. Especificación, proyecto y desarrollo de software.		X	
4. Proyecto y dirección en lo referido a seguridad informática.			
5. Establecimiento de métricas y normas de calidad de software.			
6. Procedimientos y certificaciones del funcionamiento, condición de uso o estado de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software.			
7. Dirección y control de la implementación, operación y mantenimiento de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software.		X	
8. Identificación, formulación y resolución de problemas de ingeniería en sistemas de información / informática.	X		
9. Concepción, diseño y desarrollo de proyectos de ingeniería en sistemas de información / informática.			

10. Gestión, planificación, ejecución y control de proyectos de ingeniería en sistemas de información / informática.			
11. Utilización de técnicas y herramientas de aplicación en la ingeniería en sistemas de información / informática.	X		
12. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.			
13. Desempeño en equipos de trabajo.			
14. Comunicación efectiva.		X	
15. Actuación profesional ética y responsable.			
16. Evaluación y actuación en relación con el impacto social de su actividad profesional en el contexto global y local.			
17. Aprendizaje continuo.			X
18. Desarrollo de una actitud profesional emprendedora.			

### 3. UNIDADES TEMÁTICAS

#### **Unidad Temática 1 – INTRODUCCIÓN A LA PROGRAMACIÓN. ENTRADA, SALIDA, VARIABLES Y OPERADORES.**

Esquema y elementos de una computadora (entrada, salida, procesador, memoria, almacenamiento). Etapa en la resolución de problemas (análisis, diseño, algoritmo/programa, codificación, compilación/interpretación, ejecución, prueba). Sistemas numéricos (decimal, binario, hexadecimal, octal), conversión. Método directo de conversión (binario a decimal y decimal a binario).

Entorno de desarrollo (IDE). Lenguaje de programación Python, concepto palabra reservada, Elementos del lenguaje de programación Python. Escritura y ejecución de programas en Python. Conceptos de errores de sintaxis, errores de ejecución y errores de semántica.

Entrada y salida de datos. Uso de variables. Tipos variables enteras (int), cadena de caracteres (str), reales (float), lógicas (bool). Conversión de tipos. Formato de salida de texto. Operadores aritméticos y de concatenación.

#### **Unidad Temática 2 – FUNCIONES.**

Concepto de Funciones, Declarar una función propia. Pasaje de Parámetros. Valores de retorno. Alcance de las variables: locales, no locales y globales. Definir función principal y funciones auxiliares en el entorno de un programa. Función booleana (con retorno tipo bool). Bibliotecas y uso de funciones de biblioteca. Funciones internas (built-in) y su uso. Generación de números aleatorios. Funciones / Métodos usuales recomendados, utilizados en la cátedra.

#### **Unidad Temática 3 – CONDICIONALES.**

Lógica proposicional, operadores lógicos y relacionales, tabla de Verdad. Operadores de comparación, operadores de membresía (operador in). Concepto de flujo y control de flujo, condicionales simples, condicionales anidadas y condicionales múltiples. Algoritmos de: búsquedas del máximo, búsquedas del mínimo, búsqueda por igualdad. Funciones booleanas complejas, uso de múltiples relaciones lógicas dentro de la condición.

#### **Unidad Temática 4 – CICLOS DE REPETICIÓN.**

Concepto de ciclos. Ciclos de repetición (for, while), su uso. Función range(). Comparativa entre el ciclo for y el ciclo while. Concepto de iterable. Uso operador in en ciclo for. Ciclos anidados. Algoritmos de búsqueda y procesamiento integrando de: máximo, mínimo, condicional anidada, ciclos anidados, acumuladores, sumadores y banderas.

#### **Unidad Temática 5 – CADENA DE CARACTERES.**

Concepto de cadena de caracteres. Concepto de inmutabilidad. Acceso por índice, notación con Slices. Recorrido de cadenas con ciclos. Operaciones de concatenación con string "+" y "\*".

Caracteres, codificación ASCII, conversión. Algoritmos procesamiento, modificación y de búsqueda en string.

#### **Unidad Temática 6 – COLECCIONES.**

Listas y Listas Anidadas, Tuplas, Diccionarios. Concepto de mutabilidad. Procedimiento de crear, obtener, agregar / insertar, eliminar en colección. Concatenar colección (tupla, lista). Recorrido y procesamiento de datos en colección. Algoritmo búsquedas y ordenamiento (tupla, lista). Notación con Slices (tupla, lista). Métodos usuales recomendados. Listas de listas, implantación de matrices con listas de listas. Procesamiento de datos en colecciones, procesar una tupla, una lista, un diccionario.

#### **Unidad Temática 7 – ARCHIVOS.**

Archivo de texto y con formato CSV (Valores separados por comas). Apertura de archivos (de escritura, de lectura, de apéndice), cierre de archivos. Recorrido de archivos y procesamiento de datos sin colecciones. Carga datos de archivo en colecciones para procesamiento de datos en colección. Salida de datos en archivos.

### **4. BIBLIOGRAFÍA**

#### **4.1 BIBLIOGRAFÍA GENERAL OBLIGATORIA**

- Miller, B., Ranum, D., Elkner, J., Wentworth, P., Downey, A. B., Meyers, C., & Mitchell, D. How to Think Like a Computer Scientist: Interactive Edition. Runestone, 2014. [último acceso 08/02/2023]: <https://runestone.academy/ns/books/published/thinkcspy/index.html>
- Wentworth, P., Elkner, J., Downey, A. B., & Meyer, C. How to think like a computer scientist: Learning with Python 3. O'Reilly Media, Inc., 2015. [último acceso 08/02/2023]: <http://greenteapress.com/wp/think-python-2e/>
- Python 3.X documentación oficial (*Idioma Inglés*), 2022. [último acceso 08/02/2023]: <https://docs.python.org/3/>

#### **4.2 BIBLIOGRAFÍA GENERAL COMPLEMENTARIA**

- Guttag, J. V. Introduction to Computation and Programming Using Python: With Application to Computational Modeling and Understanding Data. Mit Press, 2021.

### **5. METODOLOGÍA DE ENSEÑANZA**

La materia se organiza para su dictado en clases teóricas y en clases prácticas. Y para el apoyo se cuenta con clases de consulta que no son obligatorias.

Clases Teóricas: El docente desarrolla la exposición de los conceptos teóricos, sus fundamentos y su aplicación. Motiva el debate grupal y el análisis. Propone analogías, ejemplos y casos prácticos, entre otros recursos didácticos.

El docente expone los temas con una combinación de transmisión oral, la utilización de pizarrón y recursos tecnológicos audiovisuales. Los alumnos participan planteando sus dudas, consultas, comentarios, fundamentando sus propios análisis. Al estudiante se le provee para cada unidad didáctica un material de lectura obligatoria desarrollado por la cátedra.

Clases Prácticas: Se implementa en la praxis los conceptos teóricos. Se promueve el entrenamiento de las herramientas claves de la materia. Se motiva la aplicación, el trabajo en grupo, la participación, la reflexión y el autoaprendizaje.

El docente guía y asiste a los estudiantes, tanto de manera individual como grupal, en la resolución de ejercicios, cumpliendo un rol de tutor o colaborador a solicitud. Eventualmente, o de ser requerido por los estudiantes, el docente desarrolla una exposición sobre resolución de

ejercicios de manera grupal o individual a modo de tutoría. Al estudiante se le provee para cada unidad didáctica una guía de ejercicios obligatoria, para la resolución de problemas, desarrollado por la cátedra.

Clases de Consultas: Es un espacio donde el docente orienta al estudiante a la resolución de problemas y a resignificar tanto de los conceptos teóricos como su aplicación.

## **6. FORMACIÓN PRÁCTICA**

Clases prácticas (resolución de problemas):

En la práctica se desarrollan actividades dentro del ámbito de un Laboratorio de Informática. Consisten en la resolución de problemas y su implementación en computadora, en el lenguaje de programación elegido en la cátedra (Python), aplicando así todos los conceptos vistos en las clases teóricas. Con estas actividades el estudiante desarrollará un aprendizaje inductivo, cooperativo y por descubrimiento. Al estudiante se le provee de una guía práctica de ejercicios por cada unidad didáctica. Las guías de ejercicios son:

<b>Práctica</b>	<b>Tema</b>	<b>Actividad</b>	<b>Lenguaje Software</b>	<b>Laboratorio</b>
1	Introducción a la programación. (Unidad 01)	Resolución de problemas / ejercicios que involucran la entrada y salida, las variables, los tipos de datos y los operadores.	Lenguaje de programación Python ( <i>última versión</i> ). Thony es el IDE a utilizar.	Laboratorio de Informática y Redes (A, B). Laboratorio de Mecánica Computacional.
2	Funciones. (Unidad 02)	Resolución de problemas / ejercicios que involucran el uso de funciones en la implementación de la solución.	Lenguaje de programación Python ( <i>última versión</i> ). Thony es el IDE a utilizar.	Laboratorio de Informática y Redes (A, B). Laboratorio de Mecánica Computacional.
3	Condicionales. (Unidad 03)	Resolución de problemas / ejercicios que involucran el uso de condicionales simples, compuestas y múltiples; y el uso de funciones booleanas ( <i>qué retorna un valor lógico</i> ).	Lenguaje de programación Python ( <i>última versión</i> ). Thony es el IDE a utilizar.	Laboratorio de Informática y Redes (A, B). Laboratorio de Mecánica Computacional.
4	Ciclos de repetición. (Unidad 04)	Resolución de problemas / ejercicios que involucran el uso de ciclos de repetición tanto “for” como “while”.	Lenguaje de programación Python ( <i>última versión</i> ). Thony es el IDE a utilizar.	Laboratorio de Informática y Redes (A, B). Laboratorio de Mecánica Computacional.
5	Cadenas de caracteres.	Resolución de problemas / ejercicios que involucran	Lenguaje de programación	Laboratorio de Informática y

	(Unidad 05)	el uso y procesamiento de cadena de caracteres como ser búsquedas y reemplazos.	Python ( <i>última versión</i> ). Thony es el IDE a utilizar.	Redes (A, B). Laboratorio de Mecánica Computacional.
6	Colecciones. (Unidad 06)	Resolución de problemas / ejercicios que involucran el altas, bajas, modificaciones y consultas de información sobre las colecciones listas, tuplas y diccionarios .	Lenguaje de programación Python ( <i>última versión</i> ). Thony es el IDE a utilizar.	Laboratorio de Informática y Redes (A, B). Laboratorio de Mecánica Computacional.
7	Archivos. (Unidad 07)	Resolución de problemas / ejercicios que involucran el uso y el procesamiento de archivos de texto. Se lee, se escribe y se agrega información sobre los archivos.	Lenguaje de programación Python ( <i>última versión</i> ). Thony es el IDE a utilizar.	Laboratorio de Informática y Redes (A, B). Laboratorio de Mecánica Computacional.

Es importante destacar que cada guía de ejercicios está relacionada con el material de apunte teórico de la unidad correspondiente y que es provisto y explicado en las clases teóricas.

## 7. CRITERIO Y MODALIDAD DE EVALUACIÓN

La evaluación será formativa, en proceso, con el seguimiento de las actividades que realice el alumno y el monitoreo de la adquisición de las competencias establecidas. La evaluación continua se realiza a través de la participación en clase en las prácticas de laboratorio. Asimismo, se cuenta con tres instancias de evaluación sumativa: dos exámenes parciales y el examen final de la materia.

### **Criterios de evaluación**

- *Claridad y precisión conceptual.*
- *Utilización de vocabulario específico.*
- *Claridad y precisión en la implementación del programa que resuelve el problema.*
- *Aplicación de conceptos teóricos pertinentes en la resolución de problemas.*
- *Coherencia entre la resolución del problema planteado.*
- *Aplicación adecuada de los procedimientos y técnicas que corresponde utilizar para la resolución del problema.*
- *Implementación correcta y funcional de la codificación que resuelve el problema.*
- *Desarrollo de habilidades de autoaprendizaje guiadas a través de la práctica y la experimentación.*

### **Criterios de calificación**

- Escala conceptual para los trabajos prácticos desarrollados en el laboratorio a través de las guías de ejercicios: Muy Bien, Bien, Regular, Mal, Incompleto, No realizó. Se desarrollan valoraciones dialogadas por observación directa, de forma individual, grupal y/o generalizadas a toda el aula a modo de guiar a los estudiantes en su proceso de formación.
- Escala de valoración en evaluaciones parciales para cada ejercicio: **A, A-, A--, D, I, NR.**  
**A, A-, A--:** Son distintos niveles de aprobación; **D:** Desaprobado; **I:** Desaprobado por incompleto; **NR:** Desaprobado por No realizar. Luego, con todos los resultados de los ejercicios

de las evaluaciones (parcial y recuperatorio) se realiza una transformación a valoración numérica para consignar la nota de cursada en actas de la Facultad.

- Escala de valoración en evaluaciones finales para cada ejercicio: Son las mismas escalas declaradas para evaluaciones parciales escritas en el punto anterior.

### **Instancias de evaluación sumativa**

Exámenes parciales: En la cátedra se toman dos parciales, cada uno con una oportunidad de recuperación. Cada parcial o recuperatorio contiene una parte práctica que consiste en 2 (dos) ejercicios donde cada uno tiene un problema a resolver. El examen se lleva adelante en computadora.

Por cada ejercicio los estudiantes deben escribir un programa en lenguaje de programación Python utilizando los conceptos teóricos permitidos para el parcial, de acuerdo con las unidades temáticas que se están evaluando.

Para aprobar, se debe tener bien resueltos como mínimo un problema. Un problema bien resuelto implica, que el código de programación funcione, es decir ejecute correctamente en una computadora y provea la salida esperada, respetando la totalidad de la especificación del problema propuesto.

En la corrección se tendrá en cuenta que los principios teóricos empleados sean los adecuados, que el procedimiento seguido sea correcto, que los resultados sean los esperados según la especificación del problema. Un error menor podría significar una baja de puntos; el profesor evaluará la gravedad de este para determinar baja de puntos o desaprobación del ejercicio. Un error conceptual grave, invalida el ejercicio.

La temática del parcial es anticipada por los Profesores en cada oportunidad. Se fijan de antemano fechas y temas a incluir.

### **FIRMA DE LA CURSADA**

Para aprobar la cursada es necesario tener los niveles de asistencia a clases que fija la Universidad, y tener aprobados dos parciales o sus recuperatorios.

### Examen final:

Modalidad de evaluación:

- El examen final consiste en una evaluación escrita y presencial, donde el alumno deberá demostrar conocimientos teóricos y prácticos.
- La parte teórica consiste en preguntas de opción entre respuestas múltiples, preguntas de indicar la respuesta correcta, entre otras.
- La parte práctica consiste en la resolución de 2 (dos) ejercicios.
- En el examen final se diferencia del parcial en que abarca todos los temas del programa totalmente integrados. Los ejercicios prácticos tendrán un carácter integrador y superador con respecto a los realizados en clase y en los parciales.
- En las preguntas teóricas se pretende que el estudiante demuestre un conocimiento profundo de los temas teóricos y su aplicación.

Criterios de evaluación:

- Adecuada respuesta a los contenidos teóricos.
- Relación de conceptos pertinentes.
- Respeto de las consignas presentadas.
- Resolución correcta de los problemas planteados.

- Implementación correcta de la codificación que resuelve el problema.

Criterio de aprobación:

- Su aprobación estará sujeta al nivel de cumplimiento de los criterios de evaluación.
- Un examen final será aprobado con una nota numérica superior o igual a 4 (cuatro). Caso contrario será desaprobado.