

**Planificación de la Asignatura:** Fundamentos de Programación - Bioinformática

**Fecha:** 23/10/2024 13:02

**Código:** L1305

**Carrera:** Licenciatura en Bioinformática

**Departamento Académico:** Informática

**Docente a cargo:**

**Correo del docente a cargo:** analia.cherniz@uner.edu.ar

**Régimen de Dictado:** Cuatrimestral doble oferta

**Carga Horaria Semanal:** 5 horas semanales

**Carga Horaria Total:** 70 horas

---

**Contenidos Mínimos:**

Algoritmos computacionales. Programación: diseño de programas para resolver problemas empleando un lenguaje de programación estándar: tipos de datos, constantes y variables, estructuras de control, estructuras de datos básicas (arreglos, registros), funciones, archivos. Introducción a la programación Orientada a Objetos: prueba y depuración de programas.

---

**Correlativas Regulares para cursar:**

No posee

**Correlativas Aprobadas para cursar:**

No posee

**Correlativas Aprobadas para promocionar o rendir el examen final:**

No posee

**Objetivo General:**

Que el alumno logre:

Integrar el recurso informático a su proceso de formación básica, científica- técnica.

Comprender y utilizar los conceptos de programación de computadoras.

Resolver problemas a través de programas, empleando un lenguaje de programación estándar de alto nivel.

Desarrollar el pensamiento lógico y crítico dentro de un contexto de trabajo colaborativo.

**Objetivos Particulares:**

Que el alumno logre:

Dominar el concepto de algoritmo y su importancia en la resolución de problemas.

Dominar y aplicar los conceptos de programación: diseño, edición, compilación, depuración y ejecución de programas.

Utilizar con destreza un lenguaje estándar de alto nivel.

Emplear un entorno de programación y conocer y utilizar sus principales comandos.

Adquirir la destreza necesaria para seleccionar estructuras de datos básicas y los tipos de datos adecuados para organizar la información en los programas.

Diseñar algoritmos de solución para distintos problemas y su programa correspondiente.

Aplicar el modelo de la programación modular y estructurada a la resolución de problemas.

Familiarizarse con el hardware, manuales y lenguaje técnico propios de los elementos empleados en el desarrollo de la asignatura.

Conocer las bases conceptuales del modelo de la programación orientada a objetos.

**Programa Analítico:****Unidad 1: Introducción a la Programación**

Concepto de algoritmo y algoritmos computacionales. Programa. Depuración y documentación de programas. Lenguajes de Programación: tipos y características. Paradigmas de programación.

**Unidad 2: Introducción al Lenguaje Python**

Estructura de un programa en Python y su ejecución por parte del Intérprete. Tokens del lenguaje. Variables. Ámbito de validez de las variables. Flujos de entrada y salida.

**Unidad 3: Expresiones y Lógica de Control**

Operadores en Python. Jerarquía de los operadores. Construcción de expresiones. Estructuras de control: condicional (if, elif, else) iterativas (while, for). Uso de las declaraciones pass, break, continue. Función range(). Módulo random. Resolución de problemas mediante programas en Python. Edición, prueba y depuración de programas empleando un Intérprete de Python.

**Unidad 4: Estructuras de datos**

Listas: creación, recorrido, añadir y quitar elementos, métodos integrados. Listas por comprensión. Tuplas. Problemas de programación empleando datos estructurados.

**Unidad 5: Funciones y módulos**

Diseño e implementación de funciones en Python. Argumentos, argumentos por defecto, argumentos arbitrarios. Funciones integradas en Python. Importación de módulos y paquetes. Modularidad: creación de módulos con funciones. Problemas de programación empleando funciones. Recursividad

**Unidad 6: Módulo String**

Definición y creación, eliminación, modificación, recorrido de objetos tipo String. Métodos integrados. Operadores y operaciones. Problemas de programación empleando cadenas de caracteres.

**Unidad 7: Archivos de texto**

Apertura. Lectura. Escritura. Cierre. Funciones integradas vinculadas al manejo de archivos. Archivos de texto estructurados: CSV

**Listado de Actividades de Formación Práctica:**

Listado de actividades prácticas:

Introducción a Python, variables y operadores

Estructuras de control de flujo

Cadenas en Python

Estructuras de datos

Funciones en Python

Entrada y Salida: archivos de texto



**Metodología de Evaluación Durante el cursado:**

Evaluaciones parciales

Se realizarán 2 evaluaciones parciales de 2 hs de duración. En la primera se evaluarán los temas hasta la unidad 3 de la asignatura. Al final del cuatrimestre se tomará un examen integrador final de las totalidad de los temas dictados.

Ambas instancias de evaluación se pueden recuperar.

La nota para aprobar las evaluaciones es 6 o superior.

**Metodología de Evaluación en Exámenes Finales:**

Examen final para alumnos regulares:

Se realizará una evaluación de 2 o 3 problemas a resolver mediante código en Python y preguntas conceptuales. Tiempo estimado de resolución: 2 Hs. Calificación mínima para la aprobación: 6.

Examen final para alumnos libres:

El alumno en condición de libre debe presentar la resolución de un trabajo práctico. Se brindarán con antelación el enunciado y los archivos para procesar. El día de la mesa, el estudiante deberá presentar un código funcional, que resuelva las actividades planteadas, las cuales se evaluarán mediante un cuestionario. Una vez aprobada esta instancia, pasará a rendir el mismo examen que el alumno regular. Calificación mínima para la aprobación: 6 en ambas instancias (examen escrito y trabajo práctico).

**Condiciones de Regularidad :**

## Condiciones de Regularidad

Para poder regularizar se exige que el alumno apruebe el primer examen parcial y el examen integrador final con nota igual o superior a 60 puntos y tenga un 75% de asistencia a las clases de práctica.

## Condiciones de promoción

La asignatura puede ser aprobada (promovida) durante el cursado, evitando la instancia del examen final.

La condición para promover es tener 75% de asistencia en la práctica y nota igual o superior a 75 puntos en el examen integrador final.

**Bibliografía Principal:**

El tutorial de Python. Python Software Foundation. Disponible online en: <https://docs.python.org/es/3/tutorial/>

La Biblioteca Estándar de Python. Python Software Foundation. Disponible online en:

<https://docs.python.org/es/3/library/index.html#library-index>

Referencia del Lenguaje Python. Python Software Foundation. Disponible online en:

<https://docs.python.org/es/3/reference/index.html#reference-index>

**Bibliografía Complementaria:**

Tutorial de Python: Guía básica de Python en español. Juan José Lozano Gómez. Disponible en:

<https://j2logo.com/python/tutorial/>

Programación en Python - Nivel básico. Repositorio de manuales y recursos del entrenamiento realizado por la empresa Covantec R.L. Disponible en:

<https://entrenamiento-python-basico.readthedocs.io/es/latest/index.html>.

Introducción a la programación con Python. Apuntes del Curso de iniciación a la programación en Python, del módulo Lenguaje de Marcas y Sistemas de Gestión de la Información del ciclo formativo Administración de Sistemas Informáticos en Red (ASIR), IES Abastos de Valencia (España), 2020/2021. Disponible en:

<https://www.mclibre.org/consultar/python/>.

Python Tutorial. Bernd Klein. Disponible en: <https://python-course.eu/python-tutorial/>

Lógica de Programación. Fundación Carlos Slim. Disponible en:

[https://aprende.org/pages.php?r=cfc\\_course&tagID=7929](https://aprende.org/pages.php?r=cfc_course&tagID=7929)