

Planificación de Programación II



Código/s: R-123

Identificación y características del Espacio Curricular

Carrera/s:	Licenciatura en Ciencias de la Computación		
Plan de Estudios:	2010, TO2024	Carácter:	Obligatoria
Bloque/Campo:		Área:	Algoritmos y Lenguajes
Régimen de cursado:	Cuatrimestral		
Cuatrimestre:	2º [LCC], 2º [LCC]		
Carga horaria:	135 hs. / 9 hs. semanales	Formato curricular:	Asignatura
Escuela:	Ciencias Exactas y Naturales	Departamento:	Ciencias de la Computación
Docente responsable:	SEVERINO GUIMPEL, Federico		

Programa Sintético

Historia y evolución de los lenguajes de programación. Introducción a los lenguajes imperativos. El modelo iterativo. ¿Cómo pensar un programa? Buenas prácticas de programación. Formas de representar la información: introducción a estructuras de datos. Manejo de archivos. Gestión dinámica de la memoria.

Espacios Curriculares Relacionados

Previos Aprobados:	R-113 - Programación I
Simultaneos Recomendados:	
Posteriores:	R-212 - Estructuras de Datos y Algoritmos I, R-213 - Lenguajes Formales y Computabilidad, R-223 - Lógica

Vigencia desde 2024

Firma Profesor

Fecha

Firma Aprob. Escuela

Fecha

Con el aval del Consejo Asesor:

Fundamentación

La asignatura se ubica en el segundo semestre del 1° año de la carrera. Esta actividad curricular continúa y complementa la formación dada en la materia correlativa Programación I. El estudio de cómo representar la información continúa en la materia Estructuras de Datos y Algoritmos I.

Los algoritmos y las técnicas de programación tienen una importancia preponderante dentro de la carrera. En esta materia se pone especial atención en la construcción de algoritmos y su traducción eficiente a los lenguajes de programación Python y C, aplicando en esta traducción técnicas de diseño de programas y buenas prácticas de programación.

Se pretende que mediante los contenidos dados en la materia las y los estudiantes sean capaces de resolver de manera eficiente distintos problemas de carácter general, con independencia del lenguaje de programación utilizado y problemas de mayor dificultad, más cercanos a los entornos laborales.

Resultados del aprendizaje

Al finalizar el cursado los/las estudiantes serán capaces de:

RA1 Definir y diseñar programas que cumplan con requisitos establecidos.
RA2 Describir un determinado problema en un nivel alto de abstracción.
RA3 Escoger estructuras de datos apropiadas para la resolución de un problema.
RA4 Aplicar buenas prácticas de diseño de software en la creación de programas para obtener propiedades de software como claridad y adaptabilidad a cambios

Competencias / Ejes transversales y Resultados del Aprendizaje

Competencia/Eje transversal al que tributa	Nivel	Resultados del Aprendizaje
CGT1-Identificación, formulación y resolución de problemas de informática	Bajo	RA1, RA2
CGT2-Concepción, diseño y desarrollo de proyectos de informática	Bajo	RA3
CGT5-Generación de desarrollos tecnológicos y/o innovaciones tecnológicas	Bajo	RA3, RA4
CGS5-Fundamentos para el aprendizaje continuo	Bajo	RA4

Programa Analítico

Unidad 1

- 1.1. Historia y evolución de los lenguajes de programación
- 1.2. Paradigmas.
- 1.3 El proceso de construcción de programas

Unidad 2

- 2.1. Introducción a Python.
- 2.2. Características del lenguaje.
- 2.3 Tipos básicos
- 2.4 Operadores
- 2.5 Funciones
- 2.6 Entrada estándar
- 2.7 Condicionales

2.8 Funciones Recursivas

2.9 Uso de pytest

Unidad 3

3.1 El modelo iterativo

3.2 Listas

3.2.1 Definición

3.2.2 Operador []

3.3 Iteración

3.3.1 sobre listas

3.3.2 sobre rangos

3.4 El bloque while

Unidad 4

4.1 Introducción al manejo de la información

4.2 ¿Cómo representar la información?

4.3 Tuplas

4.4 Tuplas vs Listas

4.4 Estructuras de datos mutables y no mutables

4.5 Diccionarios

4.6 Conjuntos

Unidad 5

5.1 Manejo de archivos

5.2 Formas de apertura

5.3 Lectura de archivos

5.4 Escritura en archivos

Unidad 6

6.1 Historia del lenguaje C

6.2 Características del lenguaje

6.3 Lenguajes tipados y no tipados

6.4 Lenguajes compilados e interpretados

6.5 Tipos básicos

6.6 Operadores

6.7 Salida estándar

6.8 Bloques condicionales

6.9 Entrada estándar

6.10 Funciones

6.11 Funciones Recursivas

6.12 El uso de la sentencia assert

Unidad 7

7.1 Estructuras Condicionales

7.2 Arrays

7.3 Strings

7.4 Entrada estándar

7.5 Bloque for

7.6 La palabra clave struct

Unidad 8

8.1 Gestión de la memoria

- 8.2 Introducción
- 8.3 Operador &
- 8.4 Punteros
- 8.5 El tipo array
- 8.6 El operador []
- 8.7 Manipulando direcciones
- 8.8 Pasaje de argumentos a funciones
- 8.9 Tipos de memoria
- 8.10 Aritmética de punteros
- 8.11 Archivos en C

Modalidades de enseñanza

Se dan clases teóricas en aula, clases prácticas en laboratorio y clases prácticas en aula donde se resuelven problemas para pensar, usando el concepto de aula invertida y se trabaja en grupos.

CGT1. Identificación, formulación y resolución de problemas de informática. Para esta competencia la modalidad de enseñanza es mediante la resolución de problemas en las clases de aula y laboratorios.

CGT2. Concepción, diseño y desarrollo de proyectos de informática. Para esta competencia la modalidad de enseñanza es mediante la resolución de problemas en las clases de aula y laboratorios.

CGT5. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas. Para esta competencia la modalidad de enseñanza es mediante las clases teóricas y de laboratorio.

CGS5. Fundamentos para el aprendizaje continuo. Para esta competencia la modalidad de enseñanza es mediante las clases teóricas y de laboratorio.

Recursos

Las clases teóricas y de resolución de problemas se realizan en aula. Las clases de laboratorio se realizan en laboratorio informático. Para las clases de práctica y laboratorio se utiliza un proyector.

En el Campus virtual de la UNR se encuentra el material teórico, las prácticas, soluciones de ejercicios seleccionados y referencias bibliográficas.

Para la comunicación con los alumnos se utiliza un canal de Zulip que se encuentra instalado en un servidor del Departamento de Ciencias de la Computación.

Actividades de Formación Práctica

Nº	Título	Descripción
1	Familiarizándose con Python	Primeros programas usando números. Primeros programas usando cadenas. Primeros programas interactivos.
2	El modelo iterativo	Bucles For. Bucles While. Programas usando random

3	Cadenas Listas y Tuplas	Listas. Cadenas. Tuplas. Programas que almacenen información utilizando estos recursos para representarlas
4	Estructuras de Datos	Listas y Secuencias. Diccionarios. Conjuntos.
5	Archivos	Persistencia de la información. ¿Cómo funciona un archivo? ¿Cómo acceder y recuperar datos de un archivo?
6	Primeros pasos en C	Entrada/salida. Selección. Bucles While. Funciones.
7	Programando en C	Switch. For. Arrays. Cadenas de caracteres.
8	Estructuras y Punteros	Estructuras. Punteros. Arreglando programas. Estructuras y punteros.
9	Punteros, Funciones y Memoria dinámica	Bloques de programa que se deben analizar y marcar errores de manejo de memoria y/o salidas obtenidas.
10	Repaso Final de C	Gestión de la memoria y punteros. Estructuras y punteros.
11	TP Final	

Evaluación

La evaluación se realiza mediante tres parciales teórico-prácticos de los cuáles se pueden recuperar dos de los mismos.

El estudiante alcanza la condición de Regular, si aprueba con nota mayor o igual a 6 los 3 parciales, caso contrario, obtiene la condición de Libre.

Para las o los estudiantes que hayan alcanzado la condición Regular, el examen final consiste en la entrega de un Trabajo Práctico final integrador. El enunciado de este trabajo práctico cambia cada año.

Luego de la aprobación del TP se realiza un coloquio donde se debe justificar las decisiones tomadas en el desarrollo del TP y se evalúan temas dados en la materia.

Para las o los estudiantes en condición Libre el examen final consiste en la resolución de un examen final práctico en computadora (que equivale a un mini proyecto de TP final) y un coloquio sobre temas dados en la materia.

Resultado de Aprendizaje	Actividades/Modalidad de Enseñanza	Modalidad de Evaluación
RA1	Resolución de ejercicios en clases de práctica y en laboratorio	Parciales y trabajo práctico final
RA2	Clases teóricas y de práctica	Ejercicios de parciales
RA3	Clases teóricas y de práctica	Ejercicios de parciales
RA4	Clases teóricas y prácticas. Clases de laboratorio	Parciales y trabajo práctico final. Coloquio del trabajo final

Bibliografía básica

Autores (Apellido, Inicial nombre)	Año de edición	Título de la obra	Editorial o Revista	Ejemplares disponibles o sitio web
Magnus Lie Hetland	2005	Beginning python : from novice to professional	Apress	1
Kernighan, Ritchie	1991	El lenguaje de programación C	Pearson Educación	6
K. Reitz and T. Schlusser	2016	The Hitchhiker's Guide to Python!	O' Relly Media	https://docs.python-guide.org/
B. Gottfried	2005	Programación en C	McGraw-Hill	1

Bibliografía complementaria

Autores (Apellido, Inicial nombre)	Año de edición	Título de la obra	Editorial o Revista	Ejemplares disponibles o sitio web

Distribución de la carga horaria**Presenciales**

Teóricas				55 Hs.
Prácticas			Formación Experimental	35 Hs.
			Resolución de Problemas vinculados a la Profesión	40 Hs.
			Resolución de Problemas y Ejercicios	
			Actividades de Proyecto y Diseño	
			Formación en la Práctica Profesional	
Evaluaciones				5 Hs.
			Total	135 Hs.

Dedicadas por el alumno fuera de clase

			Preparación Teórica	5 Hs.
			Preparación Práctica	15 Hs.
			Elaboración y redacción de informes, trabajos, presentaciones, etc.	55 Hs.
			Total	75 Hs.

Cronograma de actividades

Semana	Unidad	Tema	Actividad
1	1.2.1	Presentación de la materia. Historia de los lenguajes y el proceso de construir programas. Introducción al lenguaje Python	1
2	2.2 al 2.8	Características del lenguaje Python.	1
3	3	El modelo iterativo	2
4	4.1 al 4.3	¿Cómo se representa la información en un sistema?	3
5	4.4 al 4.6	Estructuras de datos mutables y no mutables. Características	4
6	5	Archivos	5
7	6.1 al 6.4, 7.4	Historia del lenguaje C. Evolución. C vs Python. Lenguajes compilados e interpretados.	6
8	6.5 al 6.9	El lenguaje C. Características y programas.	6
9	6.10, 6.11, 6.12, 7.1, 7.5	Funciones y testing en C. Estructuras condicionales.	7

10	7.2, 7.3, 7.6	¿Cómo representar información en C?	7 y 8
11	8.1 al 8.3	¿Cómo se almacena la información en la memoria?	8
12	8.4 al 8.7	Manejo dinámico de la memoria	8
13	8.8 al 8.10	Manejo dinámico de la memoria	8 y 9
14		Archivos en C	9
15		Repaso general de la materia y presentación del trabajo final.	