

**UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO**

**FACULTAD DE CIENCIAS EXACTAS Y  
TECNOLOGÍAS**

**PLANIFICACIÓN ANUAL 2023**

ASIGNATURA:

**FUNDAMENTOS DE LA  
PROGRAMACIÓN**

**LICENCIATURA EN SISTEMAS DE  
INFORMACIÓN**

**Plan de Estudio: 2011  
Innovación Curricular 2022**

**Equipo cátedra**

Profesor adjunto: **Ing. Carmen Silva**

JTP: **Ing. Beatriz Fernández Reuter**

Auxiliares Docentes Diplomados:

**Ing. Aldo Roldan, Lic. Víctor Diaz, Prof. Walter Toledo**

Ayudantes Estudiantiles: **Martin Atia, Lucca Urso, Martin Palavecino**



## PLANIFICACIÓN DE LA ASIGNATURA

### 1- IDENTIFICACIÓN:

1.1- **Nombre de Asignatura:** Fundamentos de la Programación

1.2- **Carrera:** Licenciatura en Sistemas de Información

1.3- **Plan de Estudios:** 2011 – Innovación Curricular 2022

1.4- **Año académico:** 2023

1.5- **Carácter:** Obligatoria

1.6- **Ubicación de la Asignatura en el Plan de Estudios**

1.6.1- **Módulo – Año:** anual – 1° Año

1.6.2- **Trayecto al que pertenece la Asignatura/Obligación Curricular**

TRAYECTO	CARGA HORARIA PRESENCIAL
Ciencias Básicas y Específicas	
Algoritmos y Lenguajes	180 hs.
Arquitectura, Sistemas Operativos y Redes	
Ingeniería de Software, Bases de Datos y Sistemas de Información	
Aspectos Sociales y Profesionales	
Otros contenidos	
<b>CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR</b>	<b>180 hs.</b>

Tabla 1: Carga horaria por trayecto

1.6.3-**Correlativas**

1.6.3.1 **Anteriores:** no tiene

1.6.3.2. **Posteriores:**

Débiles: Estructuras de datos y programación I

Organizador del Computador

Teoría de Sistemas y Organizaciones

Fuertes: Base de Datos I

Investigación Operativa



### 1.7- Carga horaria:

**1.7.1. Carga horaria semanal total:** 6 horas.

**1.7.2. Carga horaria semanal destinada a la formación práctica:** 4 horas.

**1.7.3. Carga horaria total dedicada a las distintas actividades de formación práctica:**  
120 hs.

**1.8. Ámbitos donde se desarrollan las actividades de formación práctica a las que se hace referencia en el punto anterior:** aulas y laboratorio de computadoras del Dpto. de Informática - FCEyT.

**1.9. Indique la cantidad de comisiones en las que se dicta la asignatura:** 3 comisiones

## 2- PRESENTACIÓN

### 2.1. Ubicación de la Asignatura como tramo de conocimiento de una disciplina

Esta asignatura corresponde al Trayecto Algoritmos y Lenguajes. Está orientada fundamentalmente a la resolución de problemas computacionales mediante un lenguaje de programación, con el enfoque del paradigma imperativo; donde se abordan las estrategias de solución bajo criterios de calidad, eficiencia y corrección, para finalmente llegar a una implementación acorde a las especificaciones planteadas.

Se utilizan estructuras de datos, algoritmos fundamentales y técnicas de verificación.

Los conceptos abordados en la asignatura permitirán al alumno familiarizarse con los conceptos básicos de la disciplina y aplicarlos en la resolución de problemas de computación.

### 2.2. Conocimientos y habilidades previas que permiten encarar el aprendizaje de la Asignatura.

Los conocimientos y habilidades previas requeridas para cursar esta asignatura son los construidos y evaluados (matemática-Informática), en el curso de ingreso dictado por la Facultad de Ciencias Exactas y Tecnologías.

### 2.3. Aspectos del Perfil Profesional del Egresado a los que contribuye la asignatura

Esta asignatura contribuye a que el egresado posea:

- Los conocimientos básicos (lógico-matemáticos y computacionales) para una adecuada fundamentación teórica de su quehacer profesional específico.



En esta asignatura se proporciona al alumno la capacitación básica necesaria para:

- Analizar y seleccionar las estructuras de datos, necesarias para los diferentes Sistemas de Información.

## 2.4. Integración horizontal y vertical con otras asignaturas.

Esta asignatura está vinculada con otras asignaturas del Plan de Estudio, en forma vertical y horizontal, según se muestra en la siguiente figura.

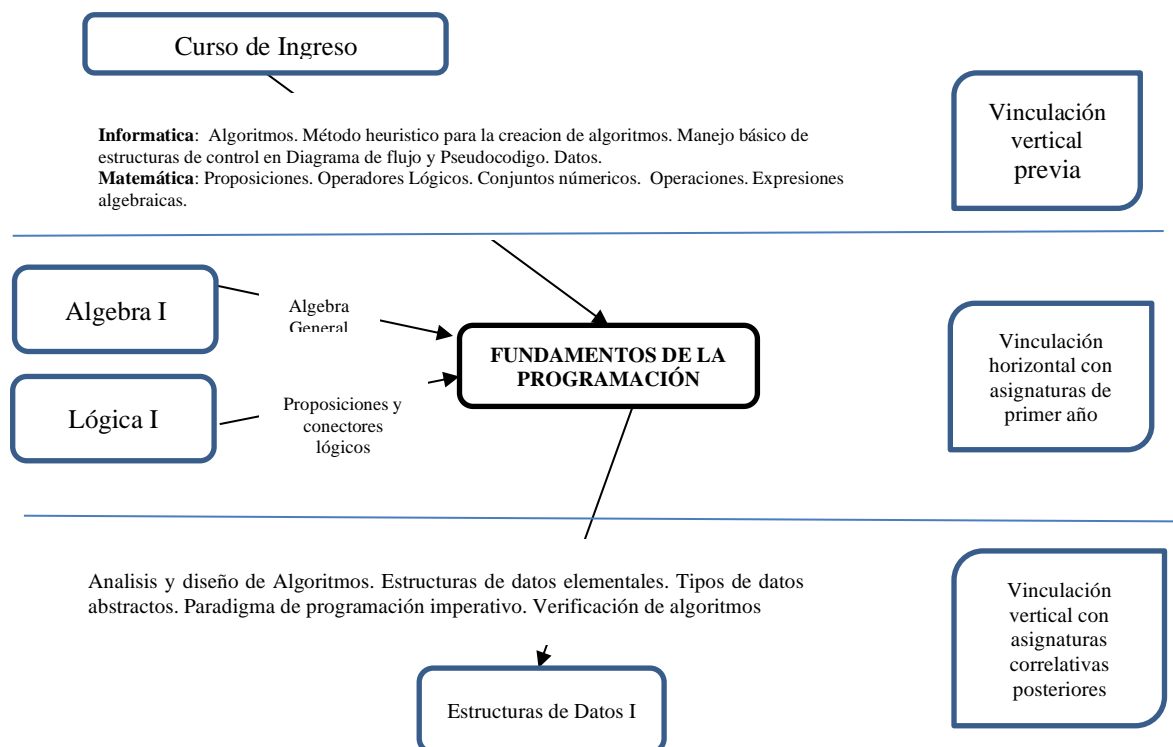


Figura 1. Integración horizontal y vertical de Fundamentos de la Programación

## 3- OBJETIVOS

Se expresan los objetivos de la asignatura en términos de competencias

### 3.1. Objetivos Generales

- Desarrollar creativamente soluciones algorítmicas seleccionando criteriosamente la alternativa adecuada, aplicando distintos paradigmas de programación.
- Emplear estructuras de control y estructuras de datos en la resolución de problemas, aplicando distintos paradigmas de programación.
- Verificar algoritmos desarrollados en los distintos paradigmas de programación.
- Aplicar distintos lenguajes de programación como herramientas computacionales para la resolución de problemas.



### 3.2. Objetivos Específicos

- Que el alumno desarrolle las siguientes competencias básicas:
  - Selecciona y procesa la información necesaria para la resolución de problemas y verifica las soluciones encontradas.
- Que el alumno desarrolle las siguientes competencias específicas:
  - Diseña e implementa programas, conforme a los principios o fundamentos específicos del paradigma de programación procedimental.
  - Construye la solución de problemas de computadora mediante la diagramación de flujo.
  - Desarrolla aplicaciones de software en lenguaje de programación Python, empleando la programación procedimental, aplicando sus conceptos básicos,
  - Resuelve correctamente situaciones problemáticas aplicando los algoritmos fundamentales de recorrido, búsqueda, ordenamiento, y actualización.
  - Aplica reglas de estilo de programación y documentación de programas.
- Que el alumno desarrolle las siguientes competencias transversales:
  - Relaciona la teoría con la práctica.
  - Participa colaborativamente en equipos de trabajo.
  - Identifica, plantea y resuelve problemas con iniciativa, toma de decisiones, autonomía y creatividad.
  - Fortalece la comunicación oral y escrita en el intercambio y transmisión de los conocimientos.

## 4- SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

### 4.1. Contenidos mínimos establecidos en el Plan de Estudios para la Asignatura

Paradigmas y Lenguajes. Análisis y diseño de algoritmos. Tipos de datos simples. Estructuras de datos elementales. Tipos abstractos de datos. Paradigma de programación imperativo. Resolución de problemas y algoritmos. Estructuras básicas de control. Procedimientos y funciones. Algoritmos fundamentales: recorrido, búsqueda, ordenamiento, actualización. Verificación de Algoritmos: pruebas orientadas a la caja negra y pruebas orientadas a la caja blanca. Lenguaje de programación imperativo.

#### 1.1. Programa Sintético sobre la base de los contenidos mínimos

##### **Unidad 1: Paradigmas y lenguajes**

Software. Lenguajes de programación. Paradigmas. Paradigma de programación imperativo.

##### **Unidad 2: Algoritmos**

Problemas de computación. Algoritmos. Estructuras básicas de control. Formas de expresar un algoritmo. Etapas en la resolución de problemas con computadora. Tipos de datos simples.



Tipos abstractos de datos. Algoritmos fundamentales: recorrido, búsqueda, ordenamiento, actualización.

**Unidad 3: Estructuras de datos**

Estructuras de datos elementales: estructura de datos compuesta. Pilas. Colas.

**Unidad 4: Programación modular y estructurada**

Análisis y diseño de algoritmos. Programación modular y estructurada. Procedimientos y funciones.

**Unidad 5: Verificación**

Estrategias de prueba del software. Verificación de Algoritmos: pruebas orientadas a la caja negra y pruebas orientadas a la caja blanca

**Unidad 6: Lenguaje de programación**

Entorno y elementos para un lenguaje de programación imperativo.

**4.3- Articulación Temática de la Asignatura**

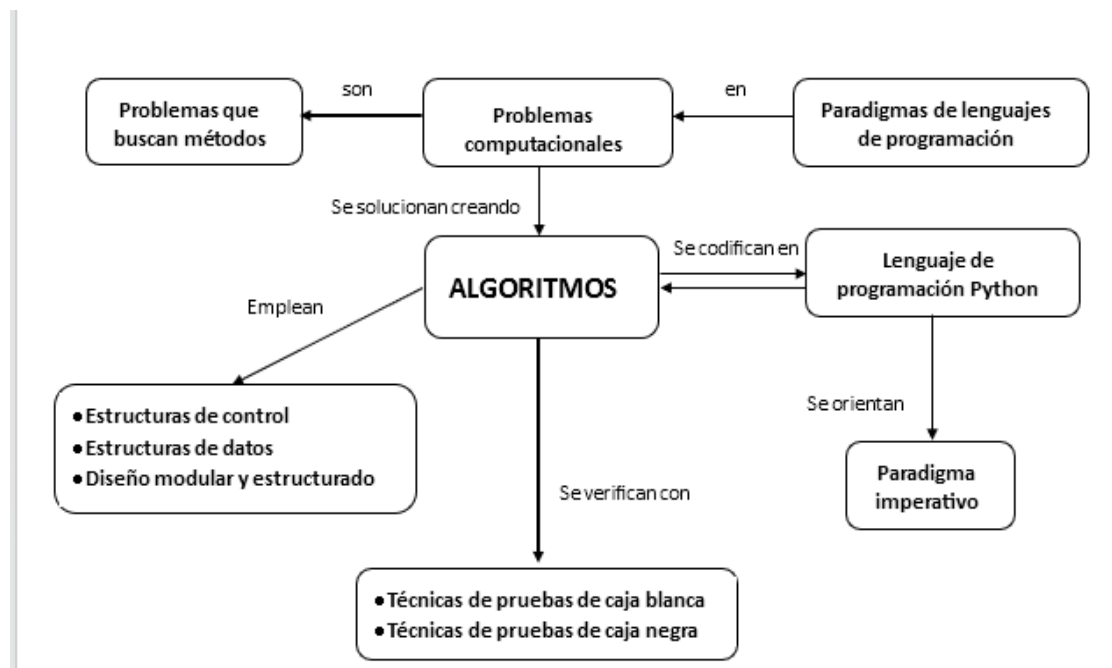


Figura 2. Articulación temática de la asignatura



#### 4.4. Programa Analítico

##### **Unidad 1: Paradigmas y lenguajes**

Software: concepto, clasificación, programas. Lenguajes de programación: concepto, tipos de lenguajes y traductores. Paradigmas: concepto, tipos y características. Paradigma de programación imperativo: concepto y principales características.

##### **Unidad 2: Algoritmos**

Algoritmo: concepto, propiedades, dominio, errores de compilación, errores de ejecución y errores de lógica. Estructuras básicas de control. Formas de expresar un algoritmo: pseudocódigo, lenguajes de programación y diagramas de flujo. Etapas en la resolución de problemas con computadora: análisis del problema, diseño del algoritmo, codificación del algoritmo, verificación, mantenimiento y documentación. Tipos de datos simples: tipo de dato numérico, lógico, carácter, definido por el usuario. Tipos abstractos de datos. Algoritmos fundamentales. Recorrido: concepto, acceso secuencial. Búsqueda: concepto, métodos de búsqueda lineal, binaria y por transformación de claves. Ordenación: concepto, métodos de ordenación selección, intercambio o burbuja. Actualización: concepto, añadir, insertar y eliminar.

##### **Unidad 3: Estructuras de datos**

Estructura de datos elementales: estructura de datos, estructuras de datos simples o compuestas, estructuras de datos homogéneas o heterogéneas, estructuras de datos estáticas o dinámicas. Estructuras de datos compuesta: concepto, arreglos: concepto, tipos de arreglos, operaciones. Registros: concepto, notación y operaciones. Pilas: concepto, operaciones e implementación estática. Colas: concepto, operaciones e implementación estática. Aplicaciones.

##### **Unidad 4: Programación modular y estructurada**

Análisis y diseño de algoritmos. Programación modular y estructurada: diseño de problemas, módulo: concepto y características. Procedimientos. Funciones. Parámetros. Variables locales y globales.



### Unidad 5: Verificación

Estrategias de pruebas de software: concepto, organización, estrategias para arquitecturas convencionales. Verificación de algoritmos: concepto, objetivos y principios. Diseño de casos de prueba orientados a la caja blanca: concepto, método del camino básico. Diseño de casos de prueba orientados a la caja negra: concepto, método de particiones de equivalencia y método de análisis de valores límite.

### Unidad 6: Lenguaje de programación

Lenguaje de programación imperativo Python: entorno. Creación de programa. Tipos de datos simples. Constantes y variables. Expresiones aritméticas, lógicas y relacionales. Estructuras de Control. Estructura de datos compuestos. Funciones: estructura, parámetros, ámbito de una variable. Depuración de un programa. Pruebas.

## 4.5- Cronograma para el desarrollo de las Unidades Temáticas

UNIDAD	CARGA HORARIA	CRONOGRAMA DE DICTADO
Unidad 1: Paradigmas y lenguajes	2	1º semana (1º módulo)
Unidad 2: Algoritmos	4 12	2º y 3 semana (1º módulo) 9º, 10º, 11º y 12º semana (1º módulo)3º y 4º semana (2º módulo)
Unidad 3: Estructuras de datos	10 4 4	4º, 5º, 6º, 7º y 8º semana (1º módulo) 1º y 2º semana (2º módulo) 10º y 11º semana (2º módulo)
Unidad 4: Programación modular y estructurada	12	13º, 14º y 15º semana (1º Módulo) 5º, 6º y 7º semana (2º módulo)
Unidad 5: Verificación	4	8º y 9º semana (2º módulo)
Unidad 6 Lenguaje de programación	8	1º y 2º módulos
<b>TOTAL</b>	<b>60</b>	

Tabla 2: Cronograma para el desarrollo teórico de las unidades temáticas

## 5. FORMACIÓN PRÁCTICA

### 5.1. Descripción de las actividades de formación práctica

Las actividades practicas se organizan en Trabajos Prácticos organizados temáticamente, En ellos, los alumnos resuelven problemas básicos y problemas similares a los del mundo real desarrollando soluciones en computadoras, mediante la aplicación del paradigma imperativo.





Los trabajos prácticos contienen ejercicios donde deberán resolverlos desarrollando...

- Diagramación de flujo
- Desarrollo de la codificación utilizando el lenguaje de programación Python
- La verificación de la solución propuesta

Las actividades prácticas se llevan a cabo tanto en forma individual, procurando desarrollar habilidades de autogestión y superación frente al error, como también en modalidad grupal, fomentando la interacción con sus iguales, la cooperación y la discusión productiva con el objetivo de resolver el problema planteado.

En las actividades de formación práctica se pretende que el alumno desarrolle habilidades de eficiencia en la programación, observando organización y claridad en la codificación y documentación de los proyectos elaborados. Particularmente, se fomenta la prueba y verificación del código desarrollado.

### 5.1.1. Cuadro sintético

	DESCRIPCIÓN	APLICANDO...	MODALIDAD
<b>ACTIVIDADES DE FORMACIÓN PRÁCTICA</b>	Resolución de problemas usando las estructuras de datos en estudio	Paradigma procedimental	Individual
		Diagramación de flujo estructurada	
	Desarrollo de proyectos de programación que hacen uso de las estructuras de datos en estudio	Diagramación de flujo estructurada	Individual y grupal
		Lenguaje de programación Python	

Tabla 3: Descripción de las Actividades de Formación Práctica



## 5.2.-Formación en Ejes Transversales

En la Tabla 3 se relacionan cada uno de los Ejes Transversales de Formación con las actividades y los resultados de aprendizaje esperados.

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
Identificación, formulación y resolución de problemas de informática	Trabajos prácticos	<ul style="list-style-type: none"><li>• Selecciona y procesa la información necesaria para la resolución de problemas y verifica las soluciones encontradas.</li><li>• Realiza la búsqueda creativa de solución/es algorítmica/s, si es que existe/n, y selecciona la alternativa más adecuada.</li><li>• Verifica la solución propuesta</li></ul>	Básico
Utilización de técnicas y herramientas de aplicación en la informática	Trabajos prácticos	<ul style="list-style-type: none"><li>• Aplica estructuras de control y estructuras de datos en la resolución de problemas.</li><li>• Utiliza para la solución de los problemas algoritmos fundamentales de recorrido, búsqueda, ordenamiento y actualización.</li><li>• Realiza el diseño estructurado y modular para la solución/es algorítmica/s.</li><li>• Diseña y verifica soluciones algorítmicas mediante la creación de proyectos en lenguaje Python.</li></ul>	Básico
Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	Trabajos prácticos	<ul style="list-style-type: none"><li>• Desarrolla, ejecuta y verifica programas en lenguaje Python aplicando programación modular y estructurada.</li></ul>	Básico
Fundamentos para el desempeño en equipos de trabajo	Actividades teóricas. Trabajos Prácticos	<ul style="list-style-type: none"><li>• Integra participativamente el desarrollo de actividades de revisión teórica y actividades de resolución de problemas aportando sus habilidades y conocimientos al equipo de trabajo.</li><li>• Desarrolla los trabajos asignados de manera colaborativa favoreciendo la resolución de los problemas planteados y el desarrollo de la dinámica grupal.</li></ul>	Básico

Tabla 3: Formación en Ejes Transversales



### 5.3 Cronograma de formación práctica

Las actividades prácticas incluyen los trabajos prácticos con su correspondiente desarrollo de solución mediante la creación de proyectos en lenguaje Python.

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DESARROLLO
T.P.1 Estructuras Básicas de Control	12	Semanas 1,2 y 3
T.P.2 Estructura de Datos Arreglos	24	Semanas 4,5,6,7,8 y 9
T.P.3 Programación Estructurada y Modular	28	Semanas 10,11,12,13,14 y 15 Semanas 1
T.P.4 Estructura de Datos Registro	16	Semanas 2,3,4 y 5
T.P.5 Verificación	24	Semanas 6,7,8,9 10 y 11
T.P.6 Estructura de Datos Pilas y Colas	16	Semanas 12,13,14 y 15
<b>TOTAL</b>	<b>120</b>	

Tabla 4: Cronograma para el desarrollo de las Actividades Prácticas

### 6.BIBLIOGRAFÍA.

TÍTULO	AUTORES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Algoritmos, datos y programas	De Giusti, Armando	Prentice Hall	1 (uno)	2001, Argentina
Fundamentos de Programación	Joyanes Aguilar, Luis	McGraw-Hill	1 (uno)	2003, España
Introducción a la Informática	Abásolo Guerrero María J. Perales López Francisco J.	UIB	1 (uno)	2011, Palma
Estructuras de Datos	Cairo, Osvaldo y Guardati, Silvia	McGraw Hil	1 (uno)	2006, Mexico
Ingeniería del software	Pressman, Roger	McGraw Hil	1 (uno)	2006, España
Introducción a la Programación	Abásolo Guerrero, Perales Lopez	Universitat de les Illes Balears	1 (uno)	2011, España
Introducción práctica a la programación con Python	Algar Díaz, M. J. & Fernández de Sevilla Vellón, M.	Editorial Universidad de Alcalá	Biblioteca Digital eLibro.net	2019
	Link de acceso: <a href="https://elibro.net/es/lc/unsebiblio/titulos/124259">https://elibro.net/es/lc/unsebiblio/titulos/124259</a>			



TÍTULO	AUTORES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Lógica de programación	Ramírez Marín, J. H	Ediciones de la U.	Biblioteca Digital eLibro.net	2017
				Link de acceso: <a href="https://elibro.net/es/lc/unsebiblio/titulos/70315">https://elibro.net/es/lc/unsebiblio/titulos/70315</a>
Diseño y construcción de algoritmos	Mancilla Herrera, A.	Universidad del Norte	Biblioteca Digital eLibro.net	2015
				<a href="https://elibro.net/es/lc/unsebiblio/titulos/69931">https://elibro.net/es/lc/unsebiblio/titulos/69931</a>
Algoritmos y programación: mejores prácticas	Ayala San Martín, G.	Fundación Universidad de las Américas Puebla (UDLAP).	Biblioteca Digital eLibro.net	2020
				<a href="https://elibro.net/es/lc/unsebiblio/titulos/180290">https://elibro.net/es/lc/unsebiblio/titulos/180290</a>
Fundamentos iniciales de lógica de programación I. Algoritmos en PseInt y Python	Ramírez Marín, J. H	Institución Universitaria de Envigado	Biblioteca Digital eLibro.net	2019
				Link de acceso: <a href="https://elibro.net/es/lc/unsebiblio/titulos/226488">https://elibro.net/es/lc/unsebiblio/titulos/226488</a>

Tabla 5: Bibliografía

## 7. ESTRATEGIAS METODOLÓGICAS.

### 7.1- Aspectos pedagógicos y didácticos

La metodología de enseñanza que se aplicará en el aula será en un espacio de diálogo y construcción, en el que se trabaje interactuando permanentemente, y de esta manera tanto los alumnos como el docente se consideran fuente de información.

Las actividades de estudio tanto prácticas como de tipo teórico serán apoyadas y desarrolladas mediante el aula virtual de la asignatura alojada en la plataforma Moodle del Centro Universitario Virtual (CUV - <http://cuv.unse.edu.ar>). Mediante esta plataforma se acercarán al estudiante diferentes recursos tales como videos tutoriales y materiales de estudio creados por los docentes, se utilizarán herramientas como cuestionarios evaluativos, así como la presentación digital de las actividades y proyectos propuestos en las prácticas. Cada contenido desarrollado es mediado para su mejor comprensión y con el fin de propiciar el diálogo y la discusión.



Tanto en las clases teóricas como en las clases prácticas, se aplicará la metodología de aula invertida. Los alumnos realizarán actividades de aprendizaje y de evaluación formativa previamente a los encuentros con los docentes. Durante las clases se completarán las actividades requeridas para el abordaje de cada tema y se organizarán tareas individuales y grupales colaborativas que favorezcan la apropiación del estudio, y que el alumno deberá completar y presentar o enviar mediante la plataforma del CUV.

Los conceptos teóricos y los aspectos metodológicos de la práctica requerida son abordados tanto en los encuentros teóricos como en los prácticos, a fin de fortalecer las habilidades de resolución algorítmica en el alumno. En la práctica se plantean a los alumnos diferentes problemas en los que se solicita la programación completa en lenguaje Python de la resolución del enunciado a fin de fortalecer las habilidades de programación en el alumno.

Se utiliza software libre en la tarea de desarrollar habilidades relacionadas con la programación en un lenguaje procedimental, habilidades que son las necesarias para la resolución de problemas y el diseño de algoritmos y además permite preparar a los alumnos a enfrentar los distintos cambios del mundo tecnológico.

En el desarrollo de las actividades de la asignatura se promueve por parte del alumno la integración participativa en donde aporte al grupo de estudio tanto sus habilidades como sus conocimientos.

Mediante actividades de aula y actividades aplicativas se propone al alumno la vinculación entre teoría y práctica. En estas actividades se fomenta el desarrollo de habilidades para la comunicación efectiva mediante la expresión oral y/o escrita de los conceptos teóricos relacionándolos con la práctica, así como el seguimiento documentado de sus procesos mentales en la resolución de problemas fundamentándolos teóricamente.

## **7.2- Mecanismos para la integración de docentes**

Se realizarán actividades de revisión y coordinación en el área Algoritmos y Lenguajes en el marco de la Comisión de Seguimiento del Plan de Estudios de la carrera de Licenciatura en Sistemas de Información. Además, se efectuarán reuniones periódicas con las cátedras del área Algoritmos y Lenguajes con el fin de aunar tareas conjuntas de integración.



La coordinación entre las asignaturas se realiza mediante encuentros anuales en los que se revisan los logros alcanzados en la comprensión y aplicación del paradigma de programación estudiado y las actividades prácticas.

### **7.3- Recursos Didácticos**

Se utilizarán como recursos didácticos:

- i. Bibliografía actualizada impresa y digital (Biblioteca eLibro) y apuntes elaborados por el equipo docente de la asignatura, disponibles en la plataforma del CUV y en el Centro de Documentación del Departamento de Informática.
- ii. En las clases teóricas y en las clases prácticas se utilizarán enunciados tomados de las actividades prácticas propuestas, diapositivas y videos elaborados por los docentes de la asignatura, demos, tutoriales, etc.
- iii. Se contará con el espacio de la asignatura en la plataforma Moodle del Centro Universitario Virtual (CUV), que permitirá al alumno acceder a los recursos digitales, apoyo de prácticas, etc. También brindará a los alumnos un canal de comunicación permanente donde podrán acceder a información actualizada de la asignatura: fechas de evaluaciones, resultados de parciales, condición final de la cursada, etc.

## **8. EVALUACIÓN**

### **8.1- Evaluación Diagnóstica**

No se prevé

### **8.2- Evaluación Formativa**

Se implementará un proceso de evaluación formativa de carácter sistemático, cualitativo y continuo, se pretende resaltar el papel de evaluación orientadora, reguladora y motivadora.

Se busca obtener información relevante respecto al proceso de enseñanza y el proceso de aprendizaje que permita comprender cómo se producen dichos procesos y tomar las decisiones pertinentes en vistas a mejorar tanto el desarrollo como los resultados del aprendizaje. Permitirá identificar la evolución en el aprendizaje de los alumnos y el grado de impacto de la propuesta educativa que lleva a cabo la cátedra.



Mediante la evaluación formativa se motivará, reforzará y proporcionará ayuda a los estudiantes, reconociendo y observando sus aprendizajes, dificultades y posibilidades.

La evaluación formativa de los contenidos teóricos se llevará a cabo mediante la aplicación de herramientas disponibles en la plataforma Moodle del CUV. En particular se utilizará la herramienta Cuestionario (CD), siguiendo la técnica de la pregunta.

Regularmente, y como una actividad de motivación, al comienzo o finalización de clase teórica se implementarán dinámicas de evaluación formativa, tales como escrito de un minuto, tarjetas de aplicación, preguntas abiertas de respuesta breve, cuyos resultados serán luego discutidos en aula.

En cuanto a las competencias relacionadas a la práctica, la evaluación formativa se llevará adelante aplicando la técnica de casos y se estructurará mediante la entrega regulada de diferentes enunciados propuestos en cada trabajo práctico.

### 8.2.1. Programa y Cronograma de Evaluaciones Formativas.

		MESES			
ACTIVIDAD	TEMA	Mayo	Junio	Agosto	Noviembre
<b>Cuestionario 1</b>	Estructuras de Datos: Arreglos	2° Semana	-----		
<b>Cuestionario 2</b>	Programación Modular	-----	3ª Semana		
<b>Cuestionario 3</b>	Estructura de Datos: Registro			4° Semana	
<b>Cuestionario 4</b>	Verificación				1° Semana

Tabla 7: Cronograma de evaluaciones teóricas formativas

Los cuestionarios se llevarán a cabo mediante la plataforma del CUV o en su defecto mediante prueba escrita y se realizarán en horarios de clase de teoría.

Se detalla a continuación objetivos, modalidad y criterios de evaluación de los trabajos prácticos.



TRABAJOS PRÁCTICOS	OBJETIVOS
TP 1 Estructuras básicas de control	Resolver problemas utilizando las estructuras de control y estructuras de datos
TP 2 Estructuras de Datos: Arreglos	Resolver problemas utilizando las estructuras de datos adecuada aplicando para la solución de los problemas algoritmos fundamentales de recorrido, búsqueda, ordenamiento y actualización.
TP 3 Programación estructurada y modular	Aplicar el diseño estructurado y modular para la solución algorítmica identificando correctamente los subproblemas: procedimientos y funciones.
TP 4 Registros y Programación estructurada y modular	Aplicar el concepto de diseño estructurado y modular aplicando la estructura de datos registro.
TP 5 Verificación	Diseñar casos de prueba para los problemas planteados, aplicando la técnica de caja negra: Análisis de valores límites y la técnica de caja blanca: Método del camino básico
TP 6 Estructuras de datos: pilas	Reconocer las principales características de las estructuras de datos Pila, diseñando el algoritmo solución correspondiente mediante el desarrollo de sus procedimientos básicos.

ASPECTOS COMUNES A TODOS LOS TRABAJOS PRÁCTICOS	
<b>OBJETIVOS GENERALES</b>	Que el estudiante adquiera las competencias para: <ul style="list-style-type: none"><li>• Usar las estructuras de datos estudiadas con sus operaciones básicas en la resolución de problemas.</li><li>• Identificar las diversas relaciones posibles entre clases en problemas que utilizan estructuras de datos para representarlas gráficamente mediante diagramas UML de clases.</li><li>• Desarrollar aplicaciones de software mediante la utilización del paradigma procedimental.</li></ul>
<b>MODALIDAD</b>	<ul style="list-style-type: none"><li>• Resolución individual de problemas mediante la diagramación de flujo, aplicando el Paradigma procedimental.</li><li>• Desarrollo individual y grupal, de proyectos de programación que hacen uso de las estructuras de datos y las estructuras de control adecuadas</li></ul>
<b>CRITERIOS DE EVALUACIÓN</b>	Cada trabajo práctico recibirá una calificación de aprobado o desaprobado. Para aprobar la presentación debe cumplir como mínimo con los siguientes ítems: <ul style="list-style-type: none"><li>• El trabajo práctico debe estar desarrollado completamente.</li><li>• Los diagramas de flujo deben haber sido desarrollados por el alumno siguiendo las pautas fijadas en los Recursos Bibliográficos.</li><li>• La codificación en lenguaje Python debe realizarse siguiendo los principios de la programación estructurada.</li><li>• La presentación del código de los ejercicios solicitados deberá realizarse en tiempo y forma a través de la plataforma CUV.FCEyT.</li></ul>





### 8.3- Evaluación Parcial

#### 8.3.1- Programa de Evaluaciones Parciales

En la siguiente tabla se muestra el programa de evaluaciones parciales para el presente año académico.

	Evaluación	Contenidos	Tipo	Fecha probable	Instrumento	Condiciones para rendir
Primer cuatrimestre	Primer Parcial	Temas incluidos en los T.P.1 y T.P.2	Especialmente diseñada, individual, escrita, prueba de desempeño	Semana de 15 de Mayo (2 hs)	Resolución documentada de problemas	-----
	Segundo Parcial	Temas incluidos en T.P.3	Especialmente diseñada, individual, escrita, prueba de desempeño	Semana del 19 de Junio (2 hs)	Resolución documentada de problemas	-----
	Recuperatorio parciales 1,2 e Integral	Temas incluidos en los T.P.1, T.P.2 y T.P.3	Especialmente diseñada, individual, escrita, prueba de desempeño	Semana del 26 de Junio (2 hs)	Resolución documentada de problemas	Haber desaprobado o estar ausente (con justificación) en el 1er y/o 2do parcial
Segundo cuatrimestre	Cuarto Parcial	Temas incluidos en T.P.4	Especialmente diseñada, individual, escrita, prueba de desempeño	Semana de 25 de Septiembre (3 hs)	Resolución documentada de problemas	Haber aprobado el 1er y 2do parcial o su correspondiente parcial Integral.
	Quinto Parcial	Temas incluidos en los T.P.5 y T.P.6	Especialmente diseñada, individual, escrita, prueba de desempeño	Semana del 23 de Octubre (2 hs)	Resolución documentada de problemas	Haber aprobado el 1er y 2do parcial o su correspondiente parcial Integral.
	Recuperatorio parciales 4,5 e Integral	Temas incluidos en los T.P.4, T.P.5 y T.P.6	Especialmente diseñada, individual, escrita, prueba de desempeño	Semana del 13 de Noviembre (3 hs)	Resolución documentada de problemas	Haber desaprobado o estar ausente (con justificación) en el 3er y/o 4to parcial

Tabla 6



### 8.3.2- Criterios de Evaluación

ESTRUCTURA	OBJETIVO Determinar si los alumnos han adquirido las competencias para:	CRITERIOS DE EVALUACIÓN
Diagramación de flujo de operaciones básicas de la estructura de datos en estudio	<ul style="list-style-type: none"><li>• Emplear las estructuras de datos estudiadas, y sus operaciones básicas, en la resolución de problemas mediante diagramación de flujo.</li></ul>	<ul style="list-style-type: none"><li>• Correcta representación de las operaciones de la/s estructura/s de datos en estudio.</li></ul>
Diagramación de flujo de la solución de un problema planteado		<ul style="list-style-type: none"><li>• Selección eficiente de la estructura de datos adecuada a la resolución del problema.</li><li>• Correcta aplicación de estructuras de diagramación.</li><li>• Resolución algorítmica en tiempo y forma adecuados.</li></ul>
Desarrollo de un programa para la resolución de un problema planteado utilizando el lenguaje de programación Python	<ul style="list-style-type: none"><li>• Desarrollar una solución de programación mediante la creación de un programa en lenguaje Python, aplicando la reutilización de código.</li></ul>	<ul style="list-style-type: none"><li>• Correcto análisis y diseño de la solución del problema propuesto.</li><li>• Eficiente codificación en lenguaje Python del enunciado, verificación y funcionamiento de la aplicación.</li><li>• Capacidad para el análisis crítico y propuestas de mejoras sobre el código presentado.</li></ul>

### 8.3.3- Escala de Valoración

La escala de valoración a emplear en los parciales y recuperatorios será cuantitativa politómica (escala de 1 a 100). El puntaje mínimo para aprobar los parciales es de cincuenta (50) puntos, y para los parciales integrales de sesenta (60) puntos.

### 8.4- Evaluación Integradora

No se prevee.

### 8.5- Evaluación Sumativa

#### 8.5.1- Condiciones para lograr la promoción sin Examen Final de la signatura. (Rige la Resolución HCD N° 135/00)

No se prevee.



### 8.5.2- Condiciones para lograr la Regularidad de la Asignatura.

Para regularizar la asignatura el alumno deberá

- Aprobar los cuatro parciales de la práctica o sus correspondientes recuperatorios integrales, según tabla 6
- Completar la presentación de una selección de los ejercicios de Trabajos prácticos solicitados

### 8.6- Examen Final

En el examen final los alumnos serán evaluados sobre los contenidos teóricos previstos en el programa de la asignatura. El examen será individual y podrá ser oral o escrito.

### 8.7- Examen Libre

Para el examen Libre el alumno deberá aprobar las dos (2) instancias que se detallan a continuación, siendo cada una de ellas eliminatorias:

#### Primera etapa

Aprobar una evaluación escrita de tipo práctica con diagramación de flujo y en computador codificando en Lenguaje Python.

#### Segunda etapa

Aprobar una evaluación oral/escrita de los contenidos teóricos del programa analítico de la asignatura.

.....  
**Ing. Carmen Silva**