

**UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS**

LICENCIATURA EN SISTEMAS DE INFORMACIÓN

***FUNDAMENTOS
DE LA PROGRAMACIÓN***

PLANIFICACIÓN

Equipo Cátedra

Profesora adjunta responsable cátedra: Msc. Marilena Maldonado

Profesores práctica: Carmen Silva y Aldo Roldan

AÑO 2016

1. IDENTIFICACIÓN

1.1. Nombre de la Asignatura / Obligación Curricular: Fundamentos de la Programación

1.2. Carreras: Licenciatura en Sistemas de Información (Plan de Estudios 2011)

1.3. Ubicación de la Asignatura/ Obligación Curricular en el Plan de Estudios

1.3.1. Módulo - Año

La asignatura está ubicada en el primer y segundo módulo, correspondiente al primer año de la carrera.

1.3.2- Ciclo al que pertenece la Asignatura/ Obligación Curricular

La asignatura pertenece al Primer Ciclo.

1.3.3- Área a la que pertenece la Asignatura/ Obligación Curricular

ÁREAS	CARGA HORARIA EN HORAS RELOJ
Ciencias Básicas	
Teoría de la Computación	
Algoritmos y Lenguajes	180 hs.
Sistemas Operativos y Redes	
Ingeniería de Software, Bases de Datos y Sistemas de Información	
Aspectos Profesionales y Sociales	
Otra	
CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR	180 hs.

1.3.4- Carga horaria semanal: la asignatura tiene una carga horaria semanal de 6 hs. (2 hs. de teoría y 4 hs. de práctica)

1.3.5- Correlativas Anteriores: no tiene

1.3.6- Correlativas Posteriores: Estructuras de datos y programación
Organizador del Computador
Teoría de Sistemas y Organizaciones

1.4. Objetivos establecidos en el Plan de Estudios para la Asignatura/ Obligación Curricular

En el Plan de Estudios no se han definido objetivos para la asignatura.

1.5. Contenidos mínimos establecidos en el Plan de Estudios para la Asignatura / Obligación Curricular.

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.

1.6. Año Académico: 2016

2. PRESENTACIÓN

2.1. Ubicación de la asignatura como tramo de conocimientos de una disciplina / ubicación de la obligación curricular como actividad o herramienta

Esta asignatura está orientada fundamentalmente a la resolución de problemas con computadoras, mediante la realización de las etapas de análisis, diseño, especificación de algoritmos, escritura de programas y verificación; utilizando diversas estructuras de datos, algoritmos fundamentales, técnicas de verificación, así como, reglas para la escritura de programas como expresión ordenada, completa y correcta de la especificación de una solución computable mediante un lenguaje de programación.

Se pondrá especial atención en consolidar el aprendizaje expresado en el párrafo anterior, con el enfoque del paradigma imperativo.

2.2. Conocimientos y habilidades previas que permiten encarar el aprendizaje de la asignatura / obligación curricular

Los conocimientos y habilidades previas requeridas para cursar esta asignatura son los adquiridos, 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

➤ Posee:

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

➤ Está capacitado para:

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

3. OBJETIVOS

Que el alumno desarrolle las siguientes competencias:

Competencias Básicas

- ✓ Capacidad para adquirir los conceptos básicos.
- ✓ Capacidad para acceder a las fuentes de información.
- ✓ Capacidad para relacionar conceptos.
- ✓ Capacidad para manejar y articular de manera eficaz distintos lenguajes comunicación (usual, formal, simbólico y gráfico).

Competencias Transversales

- ✓ Capacidad para lograr autonomía en el aprendizaje.
- ✓ Capacidad para identificar las metas y responsabilidades individuales y grupales; y actuar de acuerdo con ellas.
- ✓ Capacidad para evaluar el propio desempeño y encontrar los recursos necesarios para mejorarlo.

Competencias Específicas

- ✓ Capacidad para el diseño e implementación de programas conforme a los principios o fundamentos específicos del paradigma de programación procedimental.
- ✓ Capacidad para realizar la búsqueda creativa de solución/es algorítmica/s, si es que existe/n, y seleccionar la alternativa más adecuada.
- ✓ Capacidad para seleccionar estructuras de datos adecuadas para la resolución de problemas.
- ✓ Capacidad para emplear, según la situación problemática, algoritmos fundamentales de recorrido, búsqueda, ordenamiento y actualización.
- ✓ Capacidad para realizar una adecuada verificación de algoritmos.
- ✓ Capacidad para emplear Lenguaje C como herramienta computacional.
- ✓ Capacidad para documentar correctamente trabajos de programación.
- ✓ Capacidad para comprobar errores de implementación en lenguaje C.
- ✓ Capacidad manejar el entorno de desarrollo de lenguaje C.

4. SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

4.1. 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 I

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.

Unidad 3: Estructuras de datos I

Estructuras de datos elementales: estructura de datos compuesta.

Unidad 4: Programación 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: Estructuras de datos II

Pilas. Colas.

Unidad 7: Algoritmos II

Algoritmos fundamentales: recorrido, búsqueda, ordenamiento, actualización.

Unidad 8: Lenguaje de programación

Entorno y elementos para un lenguaje de programación imperativo.

4.2. Articulación temática de la Asignatura / Obligación Curricular

La figura 1 presenta los principales conceptos a tratar en la asignatura y la relación entre los mismos.

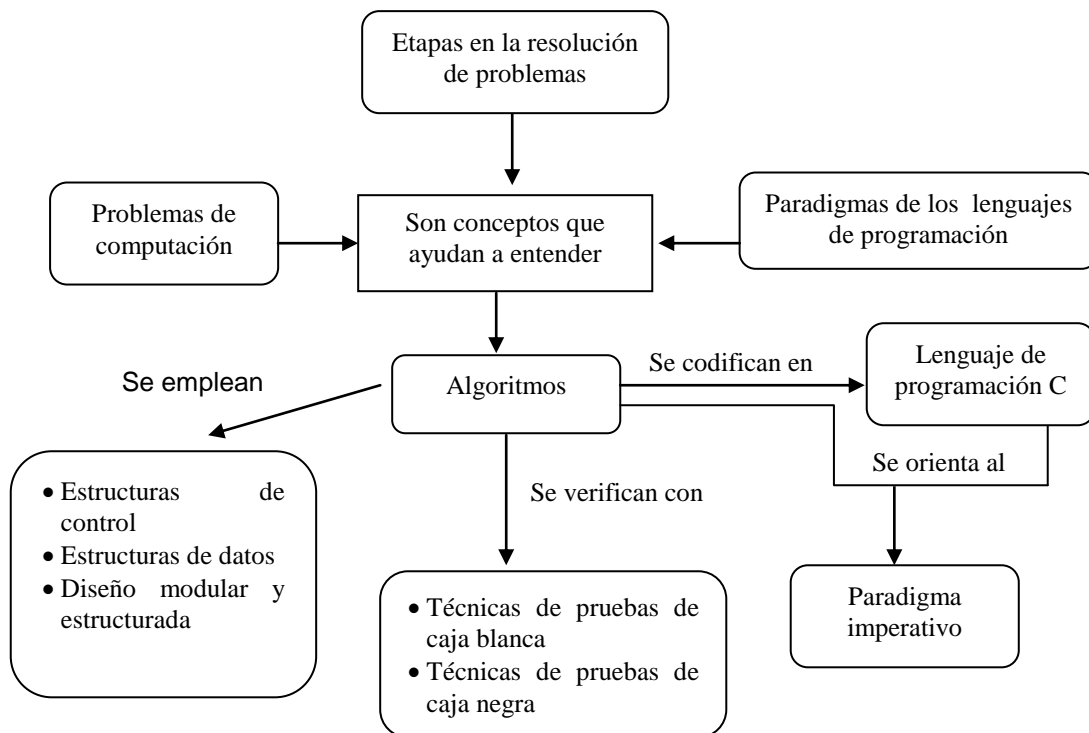


Figura 1

4.3. Integración horizontal y vertical con otras asignaturas

Integración vertical

Esta asignatura corresponde al área Algoritmos y Lenguajes del Plan de Estudios de la carrera, junto con las asignaturas: Estructuras de datos y Programación, Programación lógica y Funcional, Programación Avanzada y Métodos Numéricos.

Para la creación de programas más sólidos y eficientes, es necesario abordar Paradigmas de programación en las Ciencias de la Computación; que se fueron perfilando mediante diferentes teorías y grupos de lenguajes disímiles para construir soluciones algorítmicas. Por lo tanto esta área tiene como principal objetivo que el alumno conozca y comprenda los distintos paradigmas de programación como un modelo básico de diseño e implementación de programas, mediante el uso de distintos lenguajes de programación como herramienta computacional.

En Fundamentos de la Programación, el alumno adquiere los conocimientos básicos de la programación aplicando el paradigma de programación imperativo para el diseño de un algoritmo; se enseña un lenguaje imperativo para su implementación y se lo capacita para verificar la solución de un problema. En Estructura de Datos y Programación el alumno adquiere conocimientos de estructuras de datos avanzadas, se presentan distintos métodos de organización de archivos y criterios para efectuar la selección más adecuada a las distintas aplicaciones. Se resuelve problemas utilizando lenguajes de programación orientados al paradigma orientado a objetos.

En Programación Lógica y Funcional y en Programación Avanzada se avanza en el aprendizaje de nuevos paradigmas de programación, con la finalidad que el alumno formalice y extienda sus conocimientos y aplicaciones con respecto a la programación.

Además, en métodos numéricos se adquiere conocimientos sobre propagación de errores y algoritmos numéricos

Integración horizontal

La asignatura se relaciona con el curso de ingreso, coordinado por la Secretaría Académica de la Facultad de Ciencias Exactas y Tecnologías, ya que se enseña los conocimientos básicos sobre algoritmos, que son necesarios para la enseñanza y aprendizaje de los contenidos de esta asignatura.

Esta asignatura está vinculada con otras asignaturas del Plan de Estudio, en forma vertical y horizontal, según se muestra en figura 2.

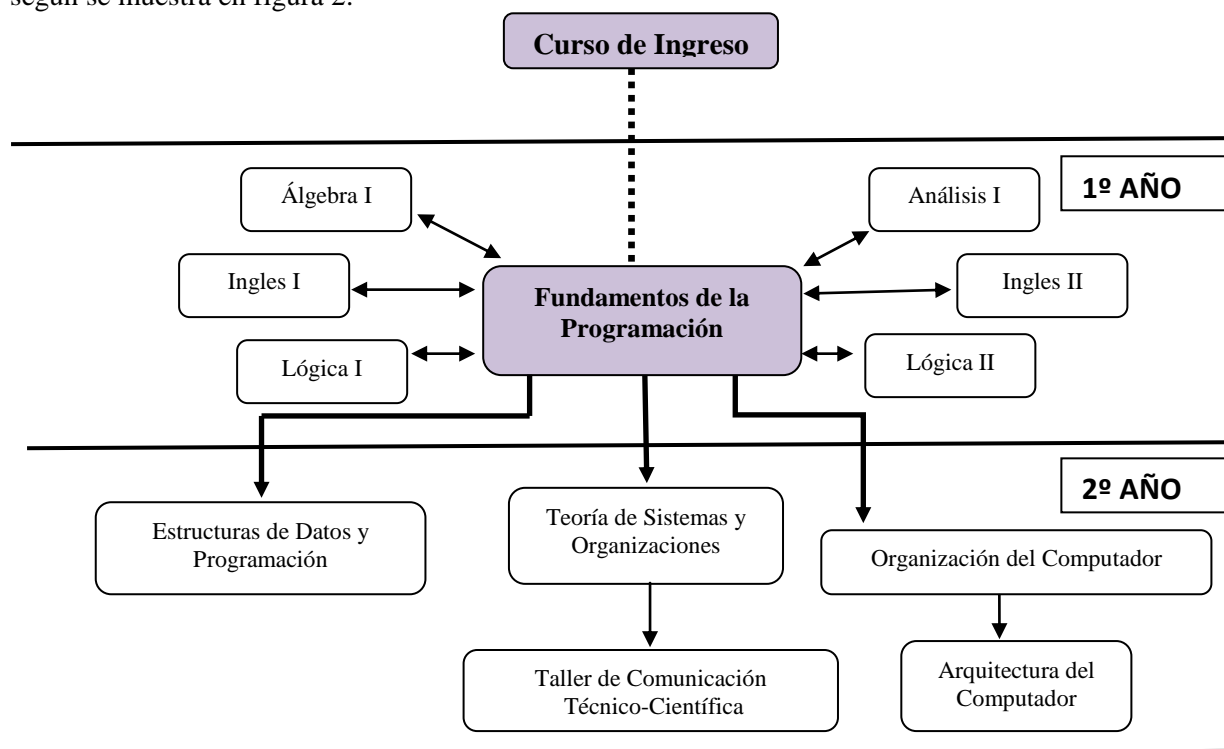


Figura 2

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

Problemas de computación: problemas que buscan respuestas, problemas que buscan pruebas, problemas que buscan métodos. 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.

Unidad 3: Estructuras de datos I

Estructura de datos elementales: estructura de datos, estructuras de datos simples o compuestos, 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.

Unidad 4: Programación 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: Estructuras de datos II

Pilas: concepto, operaciones e implementación estática. Colas: concepto, operaciones e implementación estática. Aplicaciones.

Unidad 7: Algoritmos II

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, shell, quicksort. Actualización: concepto, añadir, insertar y eliminar.

Unidad 8: Lenguaje de programación

Lenguaje de programación imperativo C: 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. Programa y cronograma de trabajos prácticos

El desarrollo de los trabajos prácticos se realizará conforme el plan que se muestra en la siguiente tabla.

Trabajos Prácticos	Contenidos	Cantidad Horas
T.P.1 Algoritmos-Estructuras de control	Unidad 1, 2	8 hs.
T.P.2 Estructura de Datos I – Arreglos	Unidad 3 y Unidad 7	8 hs.
T.P.3 Estructura de Datos I – Arreglos y Registro	Unidad 3 y Unidad 7	14 hs.
T.P.4 Programación Estructurada y Modular	Unidad 4 y Unidad 7	12 hs.
T.P.5 Verificación	Unidad 5	10 hs.
T.P.6 Estructura de Datos II	Unidad 6	8 hs.
TOTAL HORAS PRÁCTICAS		60 HORAS

Tabla 1

Esta actividad tiene una dedicación de sesiones semanales de dos horas, totalizando en el año académico **30 hs.**, se desarrollarán en aulas de la F.C.E.yT. de la UNSE y se habilitan 2 (dos) comisiones, una en horario de la mañana y otra en horario de la tarde.

4.6. Programa y cronograma de actividades de formación experimental

Taller Lenguaje de Programación C

Contenidos

Para el desarrollo de esta actividad el alumno necesita conocer los siguientes contenidos:

- ✓ Lenguaje de programación C: 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.
- ✓ Los temas desarrollados en las unidades 2, 3, 4, 5, 7 del programa analítico de la planificación de la asignatura.

Objetivos

- ✓ Adquirir destrezas fundamentales de programación que permitan la implementación de algoritmos usando el lenguaje de programación C como herramienta computacional.
- ✓ Capacidad para emplear reglas de estilo de programación y documentación de programas.
- ✓ Capacidad manejar el entorno de desarrollo del lenguaje de programación C.
- ✓ Capacidad para comprobar la presencia de errores de codificación en lenguaje de programación C.
- ✓ Desarrollar un sentido de responsabilidad por el propio comportamiento.
- ✓ Capacidad para identificar las metas y responsabilidades individuales y grupales; y actuar de acuerdo con ellas.

Metodología

Las clases serán teóricas y prácticas, para lo cual se aplicará el método de Resolución de Ejercicios y Problemas con el fin de afianzar los conceptos teóricos, aclarar dudas y que el alumno alcance un buen desarrollo en su aprendizaje.

Evaluación

Para aprobar el Taller de lenguaje de Programación C, el alumno debe aprobar un parcial o su correspondiente recuperatorio. La escala de valoración a emplear será cuantitativa (escala de 1 a 10), el puntaje mínimo para aprobar el parcial es de 5 (cinco) puntos y para el recuperatorio 6 (seis) puntos.

Para tener derecho a rendir la evaluación parcial, el alumno debe aprobar tres de cuatro prácticos evaluativos, el último práctico es obligatoria su aprobación.

A continuación se expresan en forma genérica los criterios de evaluación, los cuales serán refinados al momento de diseñar los evaluativos correspondientes.

- ✓ Capacidad para emplear Lenguaje C como herramienta computacional.
- ✓ Capacidad para emplear los recursos del Lenguaje de Programación C.
- ✓ Capacidad para emplear Lenguaje C como herramienta computacional.

- ✓ Capacidad para documentar correctamente trabajos de programación.
- ✓ Capacidad para comprobar errores de implementación en lenguaje C.
- ✓ Capacidad manejar el entorno de desarrollo de lenguaje C.
- ✓ El adecuado uso de técnicas de prueba para la verificación de un algoritmo.

Instrumentos para la actividad

Para desarrollar esta actividad de formación experimental es necesario contar con los siguientes actores:

- ✓ Profesores
- ✓ Estudiantes
- ✓ Material didáctico
- ✓ Entorno de Desarrollo Integrado (IDE) Netbeans (se ejecuta en Windows y Linux y se utiliza la familia de herramientas GCC)
- ✓ Actividades de aprendizaje

Esta actividad de formación experimental tiene una dedicación de sesiones semanales de dos horas, totalizando en el año académico **30 hs.**, se desarrollan en el Laboratorio de Informática de la F.C.E.yT. de la UNSE y se habilitan 2 (dos) comisiones, una en horario de la mañana y otra en horario de la tarde.

5. BIBLIOGRAFÍA

5.1. Bibliografía específica

Título	Autor(es)	Editorial	Año y Lugar de edición	Disponible en	Cantidad de ejemplares disponibles
Algoritmos, datos y programas	De Giusti, Armando	Prentice Hall	2001, Argentina	Biblioteca Dpto. Informática. F.C.E.y T.	1 (uno)
Fundamentos de Programación	Joyanes Aguilar, Luis	McGraw-Hill	2003, España	Biblioteca Dpto. Informática. F.C.E.y T.	1 (uno)
Introducción a la Informática	Abásolo Guerrero María J. Perales López Francisco J.	UIB	2011, Palma	Biblioteca Dpto. Informática. F.C.E.y T.	1 (uno)
Programación en C. Metodología Algoritmos y Estructura de Datos	Joyanes Aguilar Luis, Zahonero Martinez Ignacio	McGraw-Hill	2005, España	Biblioteca Dpto. Informática. F.C.E.y T.	1 (uno)
Estructura de Datos	Cairo, Osvaldo y Guardati, Silvia	McGraw Hil	2006, Mexico	Biblioteca Dpto. Informática. F.C.E.y T.	1 (uno)
Ingeniería del software	Pressman, Roger	McGraw Hil	2006, España	Biblioteca Dpto. Informática. F.C.E.y T.	1 (uno)
Introducción a la Programación	Abásolo Guerrero, Perales Lopez	Universitat de les Illes Balears	2011, España	Biblioteca Dpto. Informática. F.C.E.y T.	1 (uno)

Tabla 2

6. ESTRATEGIAS METODOLÓGICAS

6.1. 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.

En las clases teóricas se aplicará el método Expositivo/Lección Magistral donde la exposición explicativa se realizará con el diseño de las clases en powerpoint o con el uso del pizarrón, utilizando como recursos didácticos: esquemas, tablas, gráficos. Cada contenido desarrollado es mediado para su mejor comprensión y con el fin de propiciar la discusión dirigida y exposiciones abiertas; además se realizan ejercicios de aplicación con el fin de aclarar conceptos, técnicas y métodos a utilizar.

Las clases teóricas se complementan con las clases prácticas, para lo cual se elaboran trabajos prácticos con el fin de afianzar los conceptos teóricos y aclarar dudas.

Además se desarrollará un taller de Lenguaje de Programación C a los efectos de lograr una práctica de formación experimental que permita codificar la solución de problemas, mediante el uso de un lenguaje de programación. Para lograr este objetivo se aplicará el método de Resolución de Ejercicios y Problemas con el fin de afianzar los conceptos teóricos, aclarar dudas y que el alumno alcance un buen desarrollo en su aprendizaje.

6.2. Actividades de los alumnos y de los docentes

Actividades de los docentes

La asignatura está a cargo de un equipo docente integrado por:

Una profesora adjunta. Responsable asignatura: Msc. Marilena Maldonado

Dos ayudantes de primera: P.U. Carmen Silva y P.U. Aldo Roldan

Tres ayudantes estudiantiles.

El rol que desempeñen los docentes en el aula será de:

- ✓ Facilitador del aprendizaje.
- ✓ Propiciador de la comunicación.
- ✓ Facilitador del proceso de aprendizaje.
- ✓ Consultor frente a las dudas e inquietudes de los alumnos.

Las funciones específicas de cada docente serán:

- ✓ Profesora adjunta, responsable asignatura.
 - Preparar planificación de la asignatura.
 - Preparar planificación de actividades del equipo cátedra.
 - Desarrollar clases teóricas.
 - Coordinar las actividades del equipo cátedra.
 - Seleccionar material bibliográfico.
 - Preparar material didáctico.
 - Supervisar el desarrollo de clases prácticas.

- Supervisar el desarrollo de clases prácticas de formación experimental.
 - Supervisar la preparación de trabajos prácticos, trabajos prácticos de formación experimental y evaluaciones parciales.
 - Atender consultas de los alumnos.
- ✓ Ayudantes de primera
- Desarrollar las clases prácticas.
 - Desarrollar las clases de formación experimental.
 - Preparación de los trabajos prácticos, trabajos de formación experimental y evaluaciones parciales.
 - Preparación de material didáctico.
 - Colaborar y participar en el proceso de evaluación.
 - Atender consultas de los alumnos.
- ✓ Ayudantes estudiantiles
- Colaborar en el desarrollo de las clases prácticas y prácticas experimentales.
 - Colaborar en la preparación de material didáctico.
 - Atender consultas de los alumnos.

Actividades de los alumnos

- ✓ Participar de las discusiones sobre los temas que se traten en cada clase.
- ✓ Resolver y presentar trabajos prácticos y experimentales que se les soliciten.

6.3. 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.

6.4. Cuadro Sintético

Teóricas	Formación Práctica					Total
	Formación experimental	Resolución de problemas del mundo real	Actividades de Proyectos y Diseño de Sistemas de Información	Instancias supervisadas de formación en la práctica profesional	Otras	
60	60	---	---	---	60	120

Tabla 3

De la carga horaria del tipo de clase Teórica, **10 hs.** corresponden a evaluaciones parciales.

6.5. Recursos didácticos

Se utilizarán como recursos didácticos:

- ✓ Se usarán PC, cañón, pizarrón para el desarrollo de clases teóricas y prácticas.

- ✓ Equipo computacional del Laboratorio de Informática, Entorno de Desarrollo Integrado (IDE), Netbeans para contribuir al desarrollo de la fase de formación experimental.
- ✓ Bibliografía actualizada y consultas a Internet que los alumnos utilizarán para adquirir habilidad de sintetizar e integrar informaciones e ideas.
- ✓ Plataforma Moodle.

7. EVALUACIÓN

7.1. Evaluación diagnóstica

No se prevee

7.2. Evaluación formativa

La evaluación formativa es de carácter continuo y está dirigida a evaluar el proceso de enseñanza-aprendizaje durante todo el desarrollo de la asignatura. Para ello se tendrá en cuenta el desempeño que demuestren los alumnos en la realización de los trabajos prácticos, trabajo de formación experimental y evaluaciones parciales, con el fin de determinar el grado de aciertos, desaciertos y problemas en general, con el propósito de encarar, si fuera necesario, acciones correctivas.

7.3. Evaluación parcial

7.3.1. Programa y cronograma de evaluaciones parciales

En la tabla 4 se muestra el programa de evaluaciones parciales para el presente año académico.

Evaluación	Contenidos	Tipo	Fecha probable	Instrumento	Condiciones para rendir
Primer Parcial	Temas incluidos en Unidades 2, 3 y 7	Especialmente diseñada, individual, escrita, prueba de desempeño	Junio (2 hs)	Resolución documentada de problemas	-----
Recuperatorio Primer Parcial	Temas incluidos en Unidades 2, 3 y 7	Especialmente diseñada, individual, escrita, prueba de desempeño	Junio/Julio (2 hs)	Resolución documentada de problemas	Haber desaprobado o estar ausente en el primer parcial
Segundo Parcial	Temas incluidos en Unidades 4, 5, 6 y 7	Especialmente diseñada, individual, escrita, prueba de desempeño	Octubre (3 hs)	Resolución documentada de problemas	Haber aprobado el primer parcial o su correspondiente recuperatorio
Recuperatorio Segundo Parcial	Temas incluidos en Unidades 4, 5, 6 y 7	Especialmente diseñada, individual, escrita, prueba de desempeño	Noviembre (3 hs)	Resolución documentada de problemas	Haber aprobado el primer parcial o su correspondiente recuperatorio; y haber desaprobado o estar ausente en el segundo parcial

Tabla 4

Totalizando en el año académico **10 hs.** para evaluaciones parciales.

7.3.2. Criterios de evaluación

A continuación se expresan en forma genérica los criterios de evaluación, los cuales serán refinados al momento de diseñar la prueba correspondiente.

- ✓ Capacidad para buscar, seleccionar y procesar la información necesaria para la resolución de un problema.
- ✓ Capacidad para verificar la solución hallada aplicando las técnicas correspondientes.
- ✓ Capacidad para emplear un lenguaje de programación como herramienta computacional.

7.3.3. 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 recuperatorios de sesenta (60) puntos.

7.4. Evaluación integradora

No se prevee.

7.5. Autoevaluación

La autoevaluación se llevará a cabo por parte de los alumnos y del equipo cátedra. Para el primer caso se pretende lograr que el alumno desarrolle el juicio crítico, la capacidad de autoevaluarse, el valor de su trabajo y cultive un activo compromiso por la honestidad; para ello se diseñará una encuesta de opción múltiple, en la que el alumno se evaluará respecto de los conocimientos adquiridos en la asignatura, los procedimientos aprendidos, las actitudes profundizadas, etc. En el caso de los docentes permitirá una autoevaluación del equipo cátedra con el fin de corregir errores y fijar las pautas a seguir para lograr un mejor aprendizaje.

7.6. Evaluación sumativa

7.6.1. Condiciones para lograr la promoción sin examen final de la Asignatura/ obligación Curricular (Rige la Resolución HCD N°135/ 00)

No se prevee.

7.6.2. Condiciones para lograr la regularidad de la asignatura

Para regularizar la asignatura el alumno deberá:

- ✓ Tener un porcentaje de asistencia del 80%.
- ✓ Aprobar los dos parciales de práctica o sus correspondientes recuperatorios.
- ✓ Aprobar evaluación de formación experimental.

7.7. Examen final

La evaluación final será oral o escrita sobre los temas del programa analítico de la asignatura. En el caso de ser escrita la escala de valoración a emplear será cuantitativa politómica (escala de 1 a 100). El puntaje mínimo para aprobar será de cincuenta (50) puntos.

7.8. Examen libre

Los alumnos deberán cumplir las siguientes etapas, cada una de ellas eliminatória. En el caso de ser escrita la escala de valoración a emplear será cuantitativa politómica (escala de 1 a 100). El puntaje mínimo para aprobar será de cincuenta (60) puntos.

Primera etapa

- ✓ Aprobar una evaluación escrita de tipo teórica, de los temas del programa analítico de la asignatura.

Segunda etapa

- ✓ Aprobar una evaluación escrita de tipo práctica, de los temas del programa analítico de la asignatura.

Tercera etapa

- ✓ Resolución en computadora de un trabajo en Lenguaje de programación C. Unidad 8.

.....
Ing. Marilena Maldonado
Prof. Responsable Asignatura