

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

PROFESORADO EN INFORMÁTICA

FUNDAMENTOS DE LA PROGRAMACIÓN

PLANIFICACIÓN

Equipo Cátedra

Profesoras adjuntas: Msc. Marilena Maldonado - Msc. Silvina Unzaga

Jefes de trabajos: Ing. Claudia Avila

Ayudantes de primera: P.U. Carmen Silva y P.U. Aldo Roldan

AÑO 2013

1. IDENTIFICACIÓN

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

1.2. Carreras: Profesorado en Informática (Plan de Estudios 2000)

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 ciclo Básico.

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: 6 hs.

1.3.5- Correlativas Anteriores: no tiene

1.3.6- Correlativas Posteriores: Programación

Organizador del Computador

Teoría de Sistemas

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.

Solución de problemas con las computadoras. Especificación, descomposición y representación de problemas. Algoritmos: definición y propiedades. Método heurístico en la elaboración de algoritmos. Diagrama de flujo. Tablas de decisiones. Pseudocódigos. Los lenguajes de programación. Manejo de datos en programación. Instrucciones, constantes, variables y arreglos. Estructuras de control y diseño de programas. Técnicas de Programación. Programación Modular y Estructurada. Técnicas de prueba de software. Fundamentos. Diseño de casos de prueba: orientados a la caja blanca y a la caja negra. Prueba por módulos.

1.6. Año Académico: 2013

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 es el tramo inicial de la línea curricular de Programación. Está orientada fundamentalmente a brindar técnicas de análisis, diseño y construcción de algoritmos mediante el uso de diversas estructuras de datos, 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 a consolidar el aprendizaje de la programación imperativa, tendiendo a la especificación y expresión de algoritmos, tendiente a programas concretos verificables.

Además se especifican herramientas o técnicas para el diseño de casos de pruebas en la verificación de software.

La figura 1 ilustra las estrategias de articulación horizontal y vertical de la asignatura teniendo en cuenta que pertenece al ciclo básico del Plan de Estudios de la carrera de Profesorado en Informática.

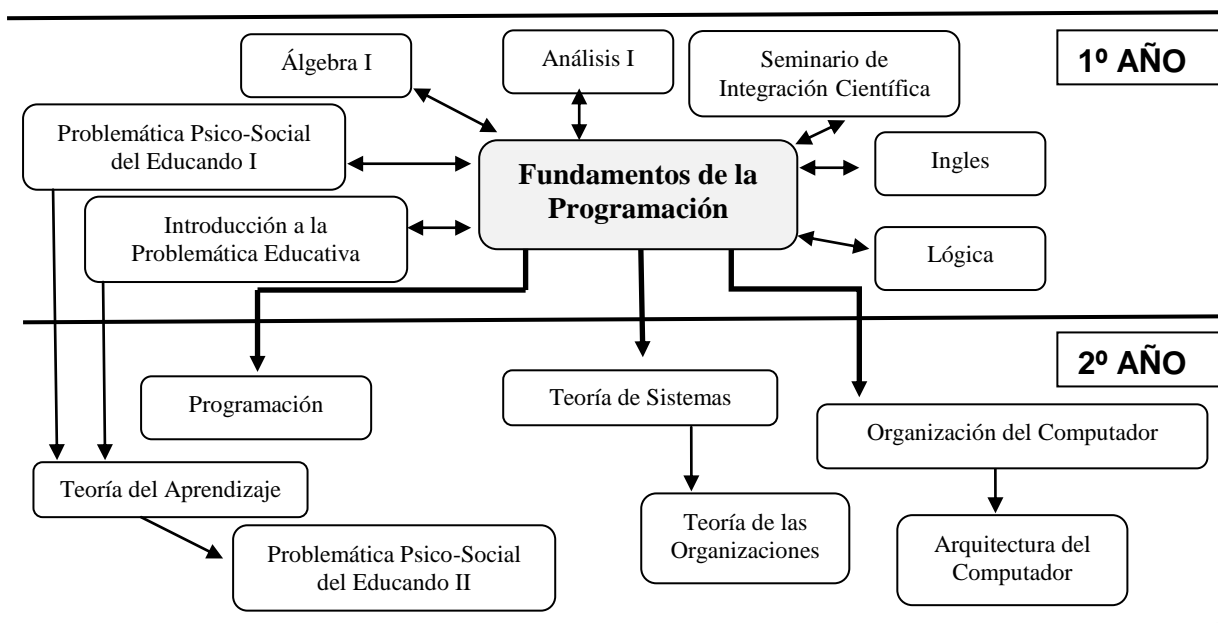


Figura 1

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 como profesional docente de la disciplina Informática.

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 buscar, seleccionar y procesar la información adecuada para el aprendizaje de los contenidos.
- ✓ Capacidad para manejar y articular de manera eficaz distintos lenguajes (usual, formal, simbólico y gráfico).

Competencias Transversales

- ✓ Capacidad para lograr autonomía en el aprendizaje.
- ✓ Capacidad para reconocer la necesidad de un aprendizaje continuo.
- ✓ Capacidad para evaluar el propio desempeño y encontrar los recursos necesarios para mejorarlo.
- ✓ Capacidad para comportarse con responsabilidad, honestidad e integridad personal.
- ✓ Capacidad para identificar las metas y responsabilidades individuales y colectivas; y actuar de acuerdo con ellas.

Competencias Específicas

- ✓ Capacidad para realizar la búsqueda creativa de solución/es algorítmica/s, si es que existe/n, y seleccionar la alternativa más adecuada.
- ✓ Capacidad para emplear estructuras de control y estructuras de datos en la resolución de problemas.
- ✓ Capacidad para realizar un diseño adecuado de una situación problemática.
- ✓ Capacidad para emplear, según la situación problemática, algoritmos fundamentales.
- ✓ Capacidad para realizar una adecuada verificación de algoritmos.
- ✓ Capacidad para emplear Lenguaje C como herramienta computacional.

4. SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

4.1. Programa sintético sobre la base de los contenidos mínimos

Unidad 1: Paradigmas y lenguajes

Paradigmas y lenguajes. Lenguajes de programación. Paradigma de programación imperativo.

Unidad 2: Algoritmos

Solución de problemas con las computadoras. Especificación, descomposición y representación de problemas. Etapas en la resolución de problemas con computadora. Algoritmo. Método heurístico en la elaboración de algoritmos. Datos simples. Estructuras de control y diseño de programas. Algoritmos y lógica. Formas de expresar un algoritmo.

Unidad 3: Estructuras de datos I

Manejo de datos en programación. Clasificación de las estructuras de datos. Estructura de datos compuesta.

Unidad 4: Modularización

Programación modular y estructurada.

Unidad 5: Verificación

Técnicas de prueba de software. Fundamentos. Diseño de casos de prueba.

Unidad 6: Estructuras de datos II

Pilas. Colas.

Unidad 7: Algoritmos fundamentales

Búsqueda. Ordenamiento

Unidad 8: Lenguaje de programación C

Entorno y elementos para la programación en lenguaje C.

4.2. Articulación Temática de la Asignatura / Obligación Curricular

La figura 2 presenta los principales conceptos a tratar en la asignatura y la relación entre los mismos.

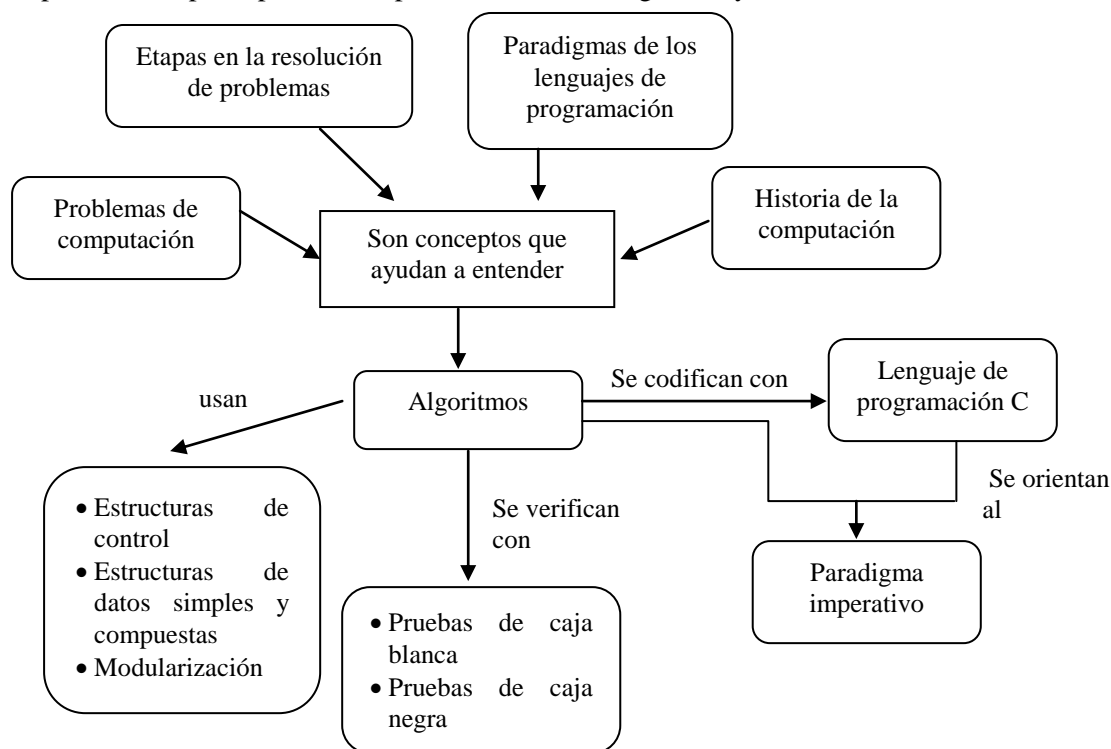


Figura 2

4.3. Integración horizontal y vertical con otras asignaturas**Integración vertical**

La integración vertical se efectúa con el curso de ingreso que la Facultad de Ciencias Exactas y Tecnologías dicta a los ingresantes de la carrera. Esta integración se realiza determinando los contenidos mínimos a dictarse y el alcance de los mismos; y con la incorporación en la asignatura de un práctico que refuerce lo aprendido y se logre un mayor alcance de las situaciones problemáticas.

Por otra parte, si bien, el Plan de Estudios de la carrera de Profesorado en Informática no está dividido en áreas, esta asignatura corresponde a la línea curricular de Programación que incluye las asignaturas Fundamentos de la Programación y Programación.

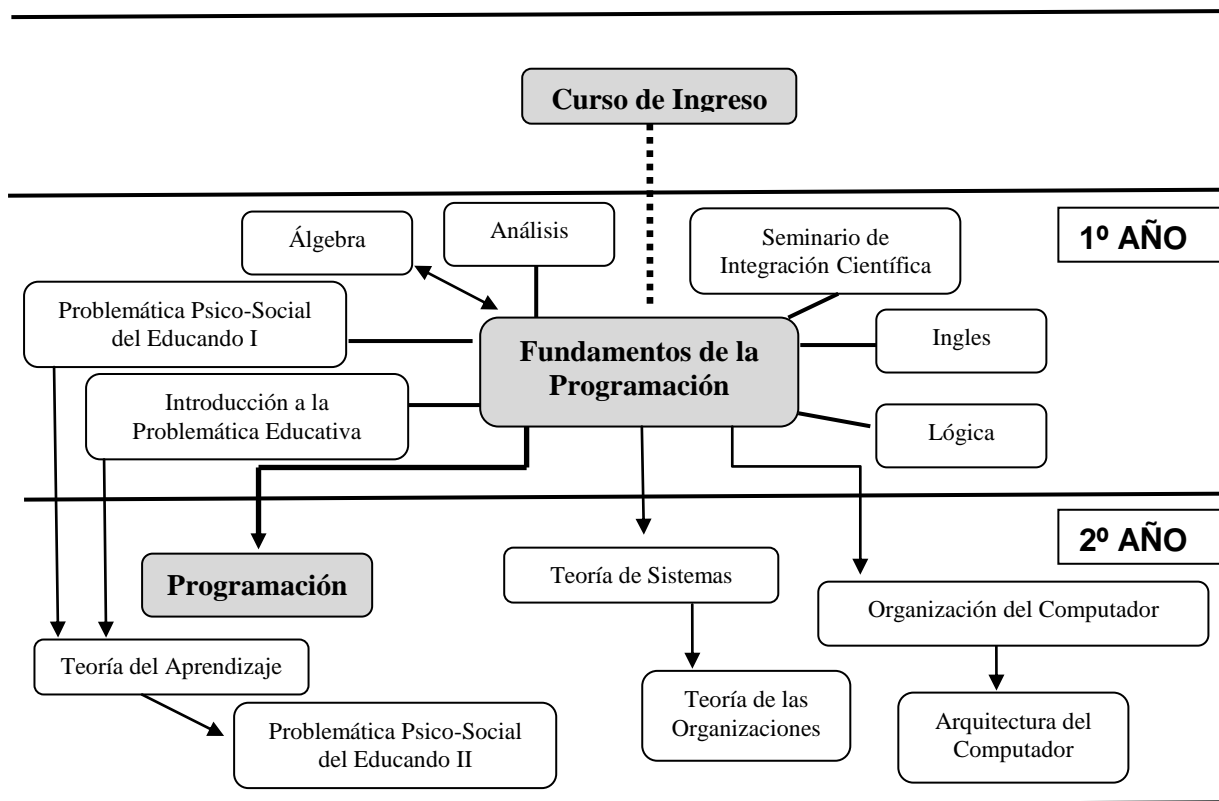


Figura 3

Esta línea curricular tiene como objetivos formar al alumnado para desarrollar aplicaciones software e informar acerca de las metodologías, técnicas y lenguajes de programación como herramientas básicas para el desarrollo de software.

En Fundamentos de la Programación, el alumno adquiere los conocimientos básicos de la programación, resuelve problemas y usa un lenguaje de programación como herramienta.

En Programación el alumnado adquiere conocimientos de estructuras de datos avanzadas, paradigmas de programación: programación orientada a objetos y programación orientada a eventos. Se presentan distintos métodos de organización de archivos y criterios para efectuar la selección más adecuada a las distintas aplicaciones. En esta asignatura se avanza en el conocimiento de los lenguajes de programación y se resuelve problemas utilizando nuevos lenguajes de programación.

Integración horizontal

A través del aprendizaje del Lenguaje de programación C y de la estructura de datos arreglos dictados en el primer módulo se brinda los conocimientos y una herramienta para la realización de programas que brinden soluciones a problemas desarrollados en asignaturas del año curricular.

4.4. Programa analítico

Unidad 1: Paradigmas y lenguajes

Paradigmas y lenguajes. Lenguajes de programación: concepto, traductores e intérpretes, tipos de lenguajes. Paradigma de programación imperativo: concepto y principales características.

Unidad 2: Algoritmos

Solución de problemas con las computadoras: problemas que buscan respuestas, problemas que buscan pruebas, problemas que buscan métodos. Especificación, descomposición y representación de problemas. Etapas en la resolución de problemas con computadora: análisis del problema, diseño de una solución, especificación de algoritmos, escritura de programas, verificación. Algoritmo: concepto, pasos elementales, propiedades (ausencia de ambigüedad, generalidad, tiempo de respuesta), dominio, errores en la construcción (errores de dominio y de lógica). Método heurístico en la elaboración de algoritmos. Estructura de datos simples: tipo de dato numérico, lógico, carácter, abstracto. Constantes y variables. Estructuras de control y diseño de programas: concepto, estructura secuencial, estructura de selección y estructura de iteración. Formas de expresar un algoritmo: lenguaje común, pseudocódigo, lenguajes de programación y diagramas de flujo, tablas de decisiones.

Unidad 3: Estructuras de datos I

Manejo de datos en programación. Clasificación de las estructuras de datos: estructuras de datos simples o compuestos, estructuras de datos homogéneas o heterogéneas, estructuras de datos estáticas o dinámicas. Estructura de datos compuesta.

Arreglos: concepto, tipos de arreglos, operaciones, representación y direccionamiento en memoria.

Registros: concepto, notación y operaciones.

Unidad 4: Modularización

Descomposición de problemas. Módulo: concepto y características. Procedimientos. Funciones. Parámetros. Variables locales y globales.

Unidad 5: Verificación

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. Prueba por módulos.

Unidad 6: Estructuras de datos II

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

Unidad 7: Algoritmos fundamentales

Búsqueda: concepto, métodos de búsqueda: lineal, binaria y por transformación de claves. Ordenación: concepto, métodos de ordenación: selección, intercambio o burbuja, shell, quicksort.

Unidad 8: Lenguaje de programación C

Elementos básicos. Creación de programa. Técnicas de Programación. Tipos de datos. Constantes y variables. Operadores y expresiones. Estructura 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 2	8 hs.
T.P.2 Estructura de Datos I – Arreglos I	Unidad 3 y Unidad 7	4 hs.
T.P.3 Estructura de Datos I – Arreglos II	Unidad 2, Unidad 3 y Unidad 7	8 hs.
T.P.4 Estructura de Datos I – Registros	Unidad 2, , Unidad 3 y Unidad 7	8 hs.
T.P.5 Verificación	Unidad 5	10 hs.
T.P.6 Programación Estructurada y Modular	Unidad 4 y Unidad 7	12 hs.
TOTAL HORAS PRÁCTICAS		60 HORAS

Tabla 1

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.

Las clases se desarrollarán en los Laboratorios del Departamento de Informática.

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 politómica (escala de 1 a 100). El puntaje mínimo para aprobar los mencionados parciales será de cincuenta (50) puntos.

Para tener derecho a rendir la evaluación parcial, el alumno deberá realizar dos prácticas evaluativas entre las cuales deberá obtener una **Nota Final (NF)** ≥ 50 puntos, donde NF es igual a lo siguiente:

$$NF = (0,35 \times \text{Puntaje de Evaluativo 1}) + (0,65 \times \text{Puntaje de Evaluativo 2})$$

A su vez debe lograr un mínimo de 50 puntos en el Evaluativo 2.

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

- ✓ 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 utiliza la familia de compilador GCC). Esta práctica se realiza en Linux.
- ✓ Actividad de aprendizaje

Cronograma

Esta actividad experimental se desarrollará con una dedicación de sesiones semanales de dos horas, totalizando en el año académico **60 hs**.

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)
Programación en C. Metodología Algoritmos y Estruct. 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)

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

Una profesora adjunta: Msc. Silvina Unzaga

Un jefe de trabajo práctico: Ing. Claudia Avila

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.
- ✓ Profesora adjunta.
- Colaborar en la preparación de la planificación de actividades del equipo cátedra.
 - Desarrollar clases teóricas.
 - Preparar material didáctico.
 - Colaborar en la supervisión de la preparación de trabajos prácticos, trabajos prácticos de formación experimental y evaluaciones parciales.
 - Atender consultas de los alumnos.
- ✓ Jefe de trabajos prácticos
- Coordinar y desarrollar clases prácticas.
 - Coordinar y desarrollar clases prácticas de formación experimentales.
 - Preparar trabajos prácticos, trabajos prácticos de formación experimental y evaluaciones parciales.
 - Preparación de material didáctico.
 - Participar y colaborar en el proceso de evaluación.
 - Atender consultas de los alumnos.
- ✓ Ayudantes de primera
- Desarrollar las clases prácticas.
 - Desarrollar las clases de formación experimental.
 - Colaborar en la preparación de los trabajos prácticos, trabajos de formación experimental y evaluaciones parciales.
 - Colaborar en la 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 de la línea curricular de Programación con el fin de aunar tareas conjuntas de integración para ser realizadas en el próximo período académico.

6.4. Cuadro sintético

Teóricas	Formación Práctica					
	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	Total
60	60	---	---	---	60	120

Tabla 4

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.

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 5 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	17 / 06 (2 hs)	Resolución documentada de problemas	-----

Evaluación	Contenidos	Tipo	Fecha probable	Instrumento	Condiciones para rendir
Recuperatorio Primer Parcial	Temas incluidos en Unidades 2, 3 y 7	Especialmente diseñada, individual, escrita, prueba de desempeño	26 / 06 (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	04 / 11 (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	18 / 11 (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% para cada tipo de clase.
- ✓ Aprobar los tres parciales o sus correspondientes recuperatorios.
- ✓ Aprobar el trabajo 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 eliminatoria.

Primera etapa

- ✓ Primera instancia: aprobar una evaluación escrita de tipo teórica de los temas de las unidades 1, 2, 3 y 7 del programa analítico de la asignatura.
- ✓ Segunda instancia: aprobar una evaluación escrita de tipo práctica de los temas de las unidades 1, 2, 3 y 7 del programa analítico de la asignatura.

Segunda etapa

- ✓ Primera instancia: aprobar una evaluación escrita de tipo teórica de los temas de las unidades 4, 5, 6 y 7 del programa analítico de la asignatura.
- ✓ Segunda instancia: aprobar una evaluación escrita de tipo práctica de los temas de las unidades 4, 5, 6 y 7 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 de la asignatura