

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

***PROFESORADO EN INFORMATICA***

# **PROGRAMACIÓN**

**PLANIFICACIÓN  
(Plan 2000)**

**EQUIPO CÁTEDRA**

**Msc. Ing. Rosa Palavecino – Profesor Asociado - SemiExclusivo**

**Msc. Ing. Raquel Zarco – Jefe de Trabajos Prácticos- Exclusivo**

**Esp. Ing. Analia Mendez – Auxiliar de 1ª – Semi Exclusivo**

**AÑO 2011**

## 1. IDENTIFICACIÓN

**1.1. Nombre de la Asignatura:** Programación

**1.2. Nombre de la Carrera:** Profesorado en Informática (Plan de estudios 2000)

**1.3. Ubicación de la Asignatura**

### 1.3.1. Módulo y Año en que está Ubicada la Asignatura

La asignatura está ubicada en el primer y segundo módulo, correspondiente al segundo 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	5 hs.
Algoritmos y Lenguajes	175 hs.
Sistemas Operativos y Redes	
Ingeniería de Software, Bases de Datos y Sistemas de Información	
Aspectos Profesionales y Sociales	
Otra	
<b>CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR</b>	<b>180 hs.</b>

**1.3.4- Carga horaria semanal:** 6 hs.

**1.3.5- Correlativas Anteriores:** Fundamentos de la Programación

Algebra I

Ingles I

**1.3.6- Correlativas Posteriores:** Sistemas de Información

Sistemas Operativos

Bases de datos

Tecnologías Informáticas Avanzadas (PI)

## 1.4. Objetivos del Plan de Estudios para la Asignatura

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

## 1.5. Contenidos Mínimos

Representación de datos. Estructuras Estáticas y Dinámicas Organización física y lógica de las estructuras de datos. Programación concurrente, funcional y lógica. Fundamentos de los lenguajes de Programación. Datos y estructuras de control. Clases de lenguajes. El estilo de la codificación.

Clasificación de los lenguajes: científicos, de gestión, polivalentes, lenguajes para el procesamiento de listas y cadenas. Lenguajes especiales. Eficiencia en código, memoria y en E/S.

### 1.6. Año Académico: 2011

## 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 corresponde a la línea curricular de Programación. Está orientada fundamentalmente al estudio, análisis y manejo de estructuras de datos y a la aplicación de distintas estrategias de diseño de algoritmos.

Se profundiza el uso de las estructuras de datos fundamentales y se desarrollan estructuras de datos avanzadas, analizando su organización física y lógica.

Se incorpora un concepto importante y poderoso en programación, como es la recursividad, y se incluyen diversos métodos de recorrido, búsqueda, ordenamiento y actualización sobre todas las estructuras de datos.

Se presentan distintos métodos de organización de archivos y criterios para efectuar la selección más adecuada a las distintas aplicaciones.

Se presentan los conceptos y se aplica el paradigma orientado a objeto.

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

- Matemáticos adquiridos en Álgebra
- Enfoque algorítmico y lenguaje de programación C, adquiridos en Fundamentos de la Programación
- Conocimientos básicos de Inglés

### 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:

- Realizar tareas de investigación, tanto a nivel básico como de aplicación en el ámbito que es específico de su competencia profesional.
- Analizar y seleccionar las estructuras de datos, necesarias para los diferentes Sistemas de Información.

### 3. OBJETIVOS

Que el alumno desarrolle las siguientes competencias:

#### a) Competencias básicas

- Capacidad de investigación
- Capacidad para actuar en nuevas situaciones
- Capacidad para buscar, seleccionar y procesar la información necesaria para la resolución de problemas
- Capacidad para verificar las soluciones encontradas

#### b) Competencias transversales

- Capacidad de abstracción
- Capacidad de análisis y síntesis
- Capacidad para la comunicación oral y escrita
- Capacidad para identificar, plantear y resolver problemas
- Capacidad para relacionar la teoría a la práctica

#### c) Competencias Específicas

- Capacidad para realizar la búsqueda creativa de soluciones algorítmicas
- Habilidad para identificar las distintas estructuras de datos y capacidad para seleccionar las más adecuadas para la resolución de problemas.
- Capacidad para utilizar el concepto de recursión.
- Capacidad para usar, en forma eficiente, los distintos métodos de clasificación y de búsqueda.
- Capacidad para resolver problemas aplicando las distintas organizaciones de archivos.
- Destreza para diseñar, ejecutar e interpretar programas usando un lenguaje orientado a objetos.

### 4. SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

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

##### Unidad 1: *Estructuras de datos básicos I*

- Introducción a las estructuras de datos. Estructura de datos encadenadas. Algoritmos fundamentales: recorrido, ordenamiento, búsqueda y actualización.

##### Unidad 2: *Estructura de Datos Avanzadas*

- Recursividad. Árboles. Grafos. Algoritmos fundamentales: recorrido, ordenamiento, búsqueda y actualización.

##### Unidad 3: *Algoritmos de Ordenamiento y Búsqueda*

- Ordenamiento y búsqueda, interna y externa. Algoritmos fundamentales.

#### Unidad 4: Archivos

- Archivos. Organización secuencial. Organización de archivos relativos. Organización de archivos indexados. Algoritmos fundamentales: recorrido, ordenamiento, búsqueda y actualización.

#### Unidad 5: Programación orientada a objetos

- Programación orientada a objetos. Clases. Polimorfismo. Herencia.

#### 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.

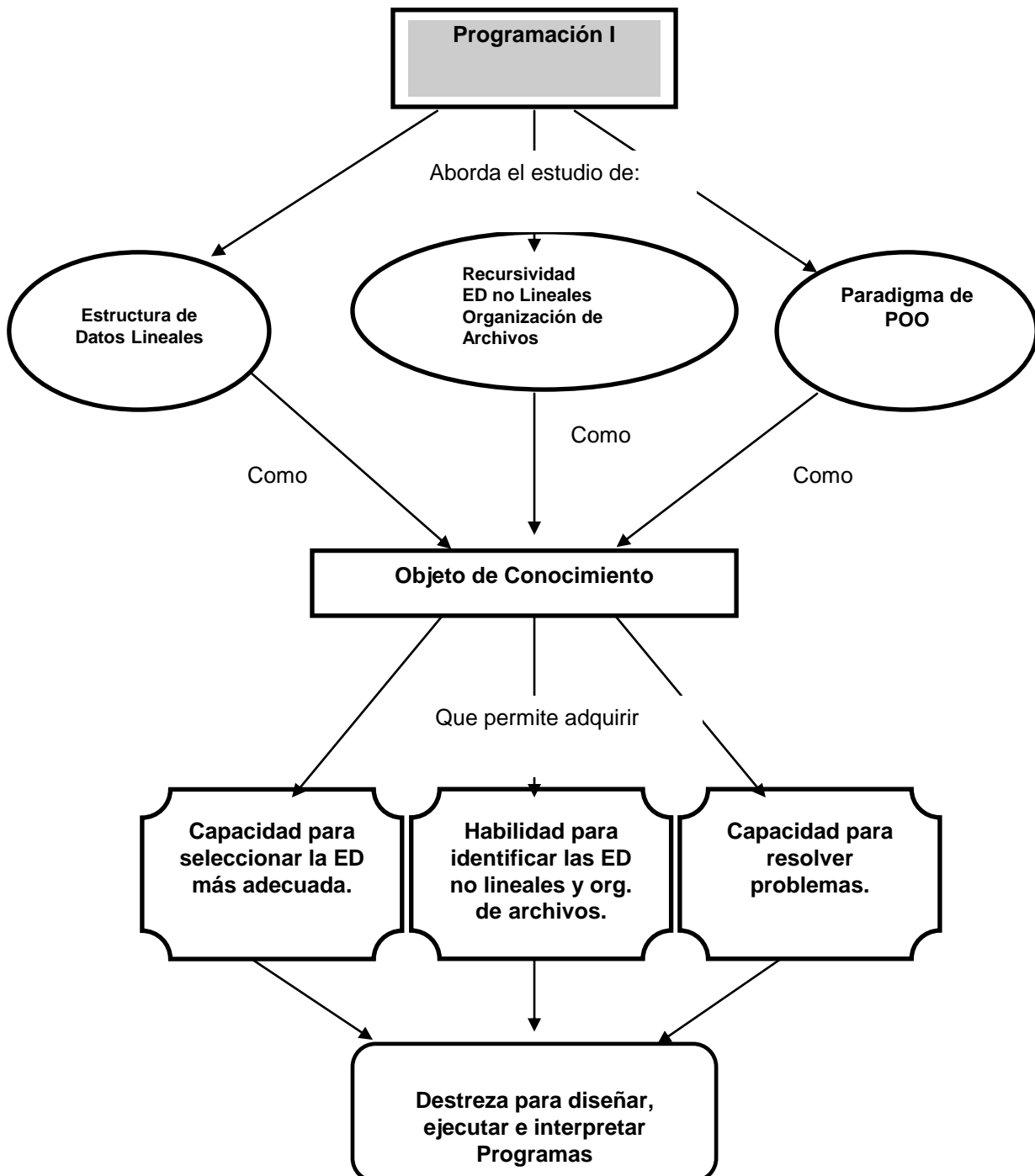


Figura 1

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

**Integración vertical**

La integración vertical se efectúa con la asignatura Fundamentos de la Programación. Esta integración se realiza con la incorporación en la asignatura de un práctico que refuerce lo aprendido y se logre un mayor alcance de las situaciones problemáticas aplicando la Programación Orientada a Objetos.

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.

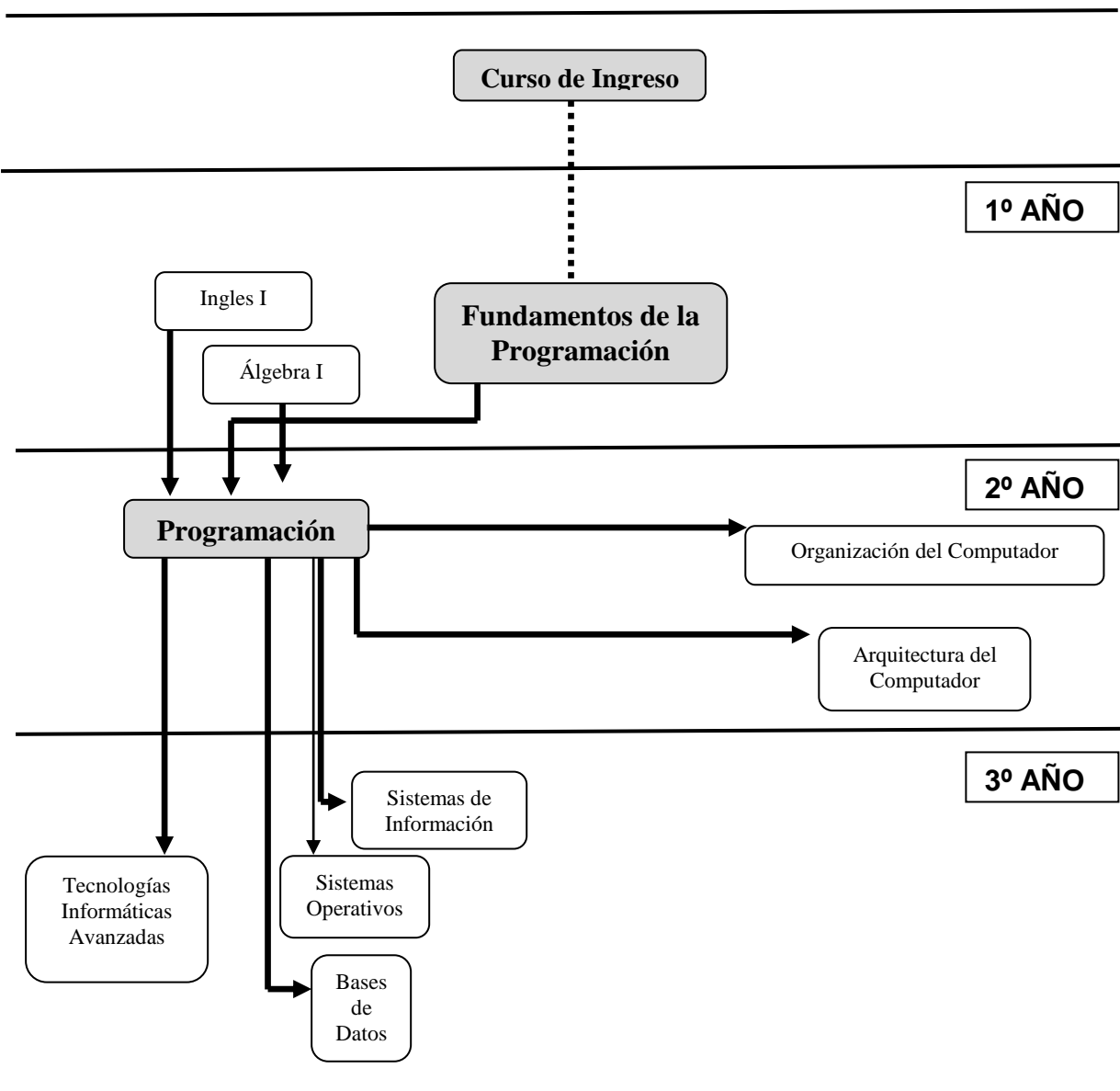


Figura 2

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 alumno adquiere conocimientos de estructuras de datos avanzadas y el paradigma de programación orientada a objetos. 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 de la estructura de datos pilas dictados en el primer módulo se brinda los conocimientos que permita a los alumnos entender la organización interna de una parte de la memoria en la asignatura Organización del Computador y para simular un modo de direccionamiento, en la asignatura Arquitectura del Computador.

Además los temas que se dictan en simultáneo en Organización del Computador le sirven al alumno para poder interpretar como se ejecutan las instrucciones de los programas que ellos escriben en Lenguaje JAVA.

## **4.4. Programa Analítico**

### **Unidad I: Estructuras de Datos I**

- i) **Introducción a las estructuras de datos:** Introducción. Abstracción en los lenguajes de programación: abstracciones de control. Abstracciones de datos. Tipos abstractos de datos: ventajas, implementación, especificación.
- ii) **Estructura de Datos Encadenadas:** Estructura de datos dinámicas. Punteros. Listas encadenadas: concepto. Clasificación. TAD Lista. Lista Ordenada. Listas doble encadenadas. Listas circulares. Pilas y colas. TAD pila, TAD cola. Diferentes formas de implementación. Colas de prioridades. Montículos. Bicolos. Operaciones: creación, inserción, eliminación, búsqueda, recorrido. Estrategias para el diseño de los algoritmos.

#### *Bibliografía recomendada*

- [Cairo, 2006]
- [Joyanes, 2008]
- [Joyanes, 2002]

### **Unidad II: Estructuras de Datos Avanzadas**

- i) **Recursividad:** Recursividad: definición y características. Requisitos para la recursión. Algoritmos recursivos. Creación, seguimiento, convocatoria. Eficiencia de la recursión.

**ii) Árboles:** Árboles generales. Conceptos básicos. Formas de representación. Árboles binarios: terminología. TAD árbol binario. Árboles binario, lleno y completo. Operaciones: creación, inserción, eliminación. Representación. Recorrido de árboles binarios. Árboles de búsqueda binaria. Operaciones en árboles de búsqueda binaria: creación, inserción, eliminación, búsqueda. Árboles balanceados por su altura (AVL). Inserción, eliminación. Rotación. Árboles balanceados por un límite. Árboles multicaminos: Árboles B, Árboles  $B^+$ : Algoritmos básicos:

**iii) Grafos:** Grafos: conceptos y definiciones. TAD grafo. Métodos de Representación: matriz de adyacencias, listas de adyacencia. Operaciones de: creación, inserción, eliminación. Recorrido de grafos: en amplitud, en profundidad, comparaciones. Matriz de caminos: algoritmo de Warshall. Caminos más cortos con un solo origen: Algoritmo de Dijkstra. Árboles de expansión: Algoritmo de Kruskal. Algoritmo de Prim.

#### *Bibliografía recomendada*

- [Cairo, 2006]
- [Joyanes, 2008]
- [Joyanes, 2002]

### **Unidad III: Algoritmos de Ordenamiento y Búsqueda**

**i) Ordenamiento interno:** concepto. Ordenación rápida (Quicksort). Ordenamiento por apilamiento (Heapsort). Ordenamiento por mezcla (MergeSort). Ordenamiento topológico.

**ii) Ordenamiento externo:** concepto. Mezcla directa. Fusión natural. Mezcla equilibrada múltiple. Método polifásico de ordenación externa.

**iii) Búsqueda externa:** búsqueda en archivos secuenciales. Búsqueda secuencial. Búsqueda secuencial mediante bloques. Búsqueda secuencial con índices. Búsqueda Binaria. Búsqueda por transformación de clave (hash). Soluciones de colisiones. Hashing dinámico: búsqueda dinámica por transformación de claves. Método de las expansiones totales. Método de las expansiones parciales. Listas invertidas. Multilistas.

#### *Bibliografía recomendada*

- [Cairo, 2006]
- [Joyanes, 2008]
- [Joyanes, 2002]
- [Joyanes, 2003]

### **Unidades IV: Archivos**

**i) Archivos:** Definiciones. Campos, registros, archivos, bases de datos. Estructura jerárquica. Conceptos: claves, registro físico o bloque. Factor de bloque. Organización de archivos: secuencial, directa, secuencial indexada. Modos de acceso. Algoritmos fundamentales: recorrido, actualización, búsqueda.

#### *Bibliografía recomendada*

- [Cairo, 2006]
- [Joyanes, 2008]
- [DeGiusti, 2001]



**Unidades V: Programación Orientada a Objetos**

**i) Programación Orientado a objetos:** Lenguajes orientados a objetos: sus características. Conceptos básicos: Programación orientado a Objetos. Objeto. Clases. Encapsulamiento. Herencia: tipos de herencia. Polimorfismo. Modularidad. Objetos: propiedades, estructura interna, atributos, métodos, mensajes. Instancia de un objeto. Clase, estructura interna, diferencia entre clase y objeto. Eventos. Tipo de eventos. Entorno visual. Excepciones: Manejo de Excepciones, Lanzamiento de Excepciones, Captura de Excepciones. Estrategia de diseño de algoritmos.

*Bibliografía recomendada*

- [[Joyanes, 2008]
- [DeGiusti, 2001]
- [Weiss, 2006]
- [Burns, 2003]

**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.

Practico nro.	Temas a desarrollar	Fecha
1	Practico Introdutorio: Arreglos, pilas, colas estáticas.	Semanas del 28 de marzo a la semana del 18/4
2	Listas simple encadenadas. Manejo de pilas y colas dinámicas.	Semanas del 25 de abril al 23 de mayo
3	Listas doble encadenadas- Listas circulares- Recursividad	Semanas del 30 de mayo al 20 de junio
4	Árboles	Semanas del 8 de agosto al 5 de septiembre
6	Grafos	Semana del 12 de septiembre al 3 de octubre
7	Archivos	Semana del 10 de octubre al 4e noviembre

Tabla 1

**4.6. Programa y Cronograma de Actividades de Formación Experimental****Taller Lenguaje de Programación Java***Contenidos*

Contenidos programáticos del Taller de Lenguaje Java

**i) Programación Orientado a Objetos:**

- Lenguajes orientados a objetos: sus características. Conceptos básicos: Programación orientado a Objetos. Objeto. Clases. Encapsulamiento. Herencia: tipos de herencia. Polimorfismo.

Modularidad. Objetos: propiedades, estructura interna, atributos, métodos, mensajes. Instancia de un objeto. Clase, estructura interna, diferencia entre clase y objeto.

- Características Principales de Java. La maquina virtual Java. La plataforma Java. El JDK de Sun
- El lenguaje Java: Sintaxis, Comentarios, Identificadores, Operadores, Expresiones, Arrays, Strings, Control de flujo. Objetos, Clases. Operador New, Constructores, Destrucción de Objetos, Garbage Collector. Control de Acceso. Herencia y Subclases. Clase Object. Clases Abstractas. Interfaces. Paquetes.
- Archivos en Java, entrada y salida, Paquete Java.io, Clase File, Manejo de Byte y Char, Acceso de lectura, Acceso de escritura, Acceso aleatorio.
- Clases básicas, Clases de Java.lang, Clases utilitarias
- Excepciones, Manejo de Excepciones, Lanzamiento de Excepciones, Captura de Excepciones.
- Entorno visual. SWING Application

#### *Bibliografía recomendada*

- [Joyanes, 2008]
- [Joyanes, 2002]
- [DeGiusti, 2001]
- [Weiss, 2006]
- [Goodrich, 2002]
- [Exkel, 2002]

#### Objetivos

- ✓ Analizar el paradigma de Programación Orientado a Objetos.
- ✓ Aprender a analizar distintas situaciones problemática y a diseñar las soluciones algorítmicas correspondientes.
- ✓ Adquirir destrezas fundamentales de programación que permitan la implementación de algoritmos usando el lenguaje de programación Java.
- ✓ Emplear reglas de estilo de programación y documentación de programas.
- ✓ Organizar eficazmente su trabajo.
- ✓ Desarrollar un sentido de responsabilidad por el propio comportamiento.

#### Metodología

Las clases serán teóricas y practicas, para lo cual se aplicaran los métodos 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 y se desarrollaran en el Laboratorio de Informática.

#### Evaluación

La evaluación del Taller de lenguaje de programación Java, se realizará en los tres parciales previstos en la asignatura, donde se solicitará codificación de los algoritmos correspondientes. La evaluación será cualitativa dicotómica (aprobado – desaprobado)

#### Instrumentos para la actividad

Para desarrollar esta actividad de formación experimental es necesario contar con el siguiente software:  
Lenguaje de Programación JAVA:

Cronograma

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

**5. BIBLIOGRAFÍA****5.1. Bibliografía Específica**

<b>Título</b>	<b>Autor(es)</b>	<b>Editorial</b>	<b>Año y Lugar de edición</b>	<b>Disponible en</b>	<b>Cantidad de ejemplares disponibles</b>
Algoritmos, datos y programas	[DeGiusti, 2001] De Giusti, Armando	Prentice Hall	2001, Argentina	Biblioteca Dpto. Informática. F.C.E.y T.	1
Fundamentos de Programación: Algoritmos. Estructuras de datos y Objetos 3° edición	[Joyanes, 2003] Joyanes Aguilar, Luis	McGraw-Hill	2003, España	Biblioteca Dpto. Informática. F.C.E.y T.	1
“Programación en JAVA 2 - Algoritmos, Estructuras de Datos y Programación orientada a objetos”	[Joyanes, 2002] Joyanes Aguilar Luis, Zahonero Martinez Ignacio	McGraw-Hill	2002, España	Biblioteca Dpto. Informática. F.C.E.y T.	1
Estructuras de datos. 3° edición	[Cairo, 2006] Cairo Osvaldo – Guardati Silvia	McGraw-Hill	2006, México	Biblioteca Dpto. Informática. F.C.E.y T.	1
Estructuras de datos en Java	[Joyanes, 2008] Joyanes Aguilar Luis, Zahonero Martinez Ignacio	McGraw-Hill	2008, México	Biblioteca Dpto. Informática. F.C.E.y T.	1

Tabla 2

**5.2. Bibliografía General o de Consulta**

<b>Título</b>	<b>Autor(es)</b>	<b>Editorial</b>	<b>Año y Lugar de edición</b>	<b>Disponible en</b>	<b>Cantidad de ejemplares disponibles</b>
Estructura de datos en JAVA"	[Weiss, 2006] Weiss Mark Allen	Addison- Wasley	2006, Madrid. Esoaña	Biblioteca Dpto. Informática. F.C.E.y T	1
Estructuras de Datos y	[Goodrich, 2002] Goodrich- Tamassia	CECSA 2° edición-	2002	Biblioteca Dpto. Informática. F.C.E.y T	1

Título	Autor(es)	Editorial	Año y Lugar de edición	Disponible en	Cantidad de ejemplares disponibles
Algoritmos en JAVA					
Estructura de Datos y Organización de Archivos	[Loomis, 1991] Loomis Mary	Prentice Hall- Hispanoamericana	1991.	Biblioteca Dpto. Informática. F.C.E.y T	1
Piensa en JAVA	[Exkel, 2002] Bruce Eckel	Prentice Hall	Madrid 2002	Biblioteca Dpto. Informática. F.C.E.y T	1

Tabla 3

\*sitios de interés en la Web

[www.sun.com](http://www.sun.com)

[www.javaworld.com](http://www.javaworld.com)

[www.javasoft.com](http://www.javasoft.com)

## 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 usando el pizarrón. Utilizando como recursos didácticos: esquemas, diagramas de flujo, pseudo código, gráficos etc. Cada contenido desarrollado es mediado para su mejor comprensión y con el fin de propiciar el diálogo y discusión; además se realizan ejercicios de aplicación con el fin de aclarar conceptos, técnicas y métodos a utilizar y se anticipa los temas para que los alumnos realicen la investigación bibliográfica correspondiente.

Las exposiciones teóricas se complementan con las clases prácticas, para lo cual se elaboran Trabajos Prácticos para cada unidad temática con el fin de afianzar los conceptos teóricos, aclarar dudas y que el alumno alcance un buen desarrollo en su aprendizaje.

En simultáneo se desarrolla un Taller de Lenguaje de Programación JAVA a los efectos de lograr una práctica de formación experimental que permita encontrar las soluciones a los problemas planteados en las clases prácticas.

## 6.2. Actividades de los Alumnos y de los Docentes

### Actividades de los docentes

La asignatura esta a cargo de un equipo docente conformado por:

Una Profesora asociada: Msc. Ing. Rosa Palavecino

Una Jefe de Trabajos Prácticos: Msc. Ing. Raquel Zarco

Una Ayudante de Primera Diplomado: Esp. Ing. Analía Mendez

Dos ayudantes estudiantiles.

El rol que desempeñen las 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 Asociada
  - Preparar la planificación de la asignatura.
  - 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 de formación experimental.
  - Supervisar la preparación de los trabajos prácticos, de formación experimental y evaluativos.
  - Atender consultas de los alumnos
- ✓ Jefe de trabajos prácticos
  - Coordinar y desarrollar clases prácticas.
  - Coordinar y desarrollar clases de formación experimentales.
  - Preparar trabajos prácticos y evaluativos.
  - Atender consultas de los alumnos.
  - Colaborar en la preparación de material didáctico.
  - Participar y colaborar en el proceso de evaluación.
- ✓ Ayudantes de Primera Diplomado
  - Desarrollar las clases prácticas.
  - Desarrollar las clases de formación experimental.
  - Colaborar en la preparación de los trabajos prácticos y evaluativos.
  - Colaborar en la preparación del trabajo de formación experimental.
  - Atender consultas de los alumnos.
  - Colaborar en la preparación de material didáctico.

- Colaborar y participar en el proceso de evaluación.
- ✓ Ayudantes estudiantiles
  - Colaborar en el desarrollo de las clases prácticas y experimentales.
  - Atender consultas de los alumnos.
  - Colaborar en la preparación de material didáctico.

#### Actividades de los alumnos

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

### 6.3. Mecanismos para la integración de docentes

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

TEORICAS	FORMACION PRACTICA					
	FORMACIÓN EXPERIMENTAL	RESOLUCIÓN DE PROBLEMAS DEL MUNDO REAL	ACTIVIDADES DE PROYECTO Y DISEÑO DE SISTEMAS DE INFORMACIÓN	INSTANCIAS SUPERVISADAS DE FORMACIÓN EN LA PRACTICA EXPERIMENTAL	OTRAS	TOTAL
60 HS. <sup>1</sup>	60 hs.	-----	-----	-----	60 hs.	180 hs.

**Tabla 4**

1) 48 hs. de teoría y 12 de evaluaciones parciales

### 6.5. Recursos Didácticos

Se utilizarán como recursos didácticos:

- ✓ Lenguaje de programación Java, equipo computacional del Laboratorio de Informática, consultas a Internet, para contribuir al desarrollo de la fase de formación experimental.
- ✓ Software PowerPoint, PC, cañón, tiza y pizarrón se usarán para clases teórico/práctica y prácticas.
- ✓ Bibliografía actualizada, que los alumnos utilizarán para adquirir habilidad de sintetizar e integrar informaciones e ideas.

## 7. EVALUACIÓN

### 7.1. Evaluación Diagnóstica

Teniendo en cuenta que la evaluación diagnóstica no sólo es una estimación, sino que tiene como propósito contribuir al aprendizaje, se llevará a cabo una única evaluación diagnóstica, al comienzo de

las clases, cuya finalidad será determinar el nivel de conocimientos y habilidades previas que permitan encarar el aprendizaje de la asignatura.

Los contenidos sobre los que se evaluará serán: Algoritmos, tipos de datos, arreglos, pilas y colas.

La evaluación diagnóstica será especialmente diseñada, individual, escrita y objetiva. Se hará una prueba de opción múltiple para que el alumno marque la opción correcta. El nivel de calificación será cualitativa politómica (nivel medio, bajo alto).

## 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, taller 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 a llevar a cabo durante el presente año académico.

Evaluación	Contenidos	Tipo	Fecha probable	Instrumento	Condiciones para rendir
Primer Parcial	Listas simples encadenadas- Manejo de Pilas y Colas implementadas dinámicamente	Especialmente diseñada, individual, escrita, prueba de desempeño	27/05	Resolución documentada de problemas	-----
Recuperatorio del Primer Parcial	Listas simples encadenadas- Manejo de Pilas y Colas implementadas dinámicamente	Especialmente diseñada, individual, escrita, prueba de desempeño	8/6	Resolución documentada de problemas	Haber desaprobado o estar ausente en el primer parcial
Segundo Parcial	Listas doble encadenadas, circulares, Recursividad, árboles, grafos	Especialmente diseñada, individual, escrita, prueba de desempeño	7/9	Resolución documentada de problemas	Haber aprobado el primer parcial o su correspondiente recuperatorio
Recuperatorio del Segundo Parcial	Listas doble encadenadas, circulares, Recursividad, árboles, grafos	Especialmente diseñada, individual, escrita, prueba de desempeño	28/9	Resolución documentada de problemas	Haber desaprobado o estar ausente en el segundo parcial
Tercer Parcial	Archivos-	Especialmente diseñada, individual, escrita, prueba de desempeño	9/11	Resolución documentada de problemas	Tener aprobado al menos un parcial anterior o sus correspondientes recuperatorios

<b>Evaluación</b>	<b>Contenidos</b>	<b>Tipo</b>	<b>Fecha probable</b>	<b>Instrumento</b>	<b>Condiciones para rendir</b>
Recuperatorio del Tercer Parcial	Archivos-	Especialmente diseñada, individual, escrita, prueba de desempeño	14/ 11	Resolución documentada de problemas	Haber desaprobado o estar ausente en el tercer parcial
Integral	Todos los temas	Especialmente diseñada, individual, escrita, prueba de desempeño	18/11	Resolución documentada de problemas	Tener aprobado dos parciales o sus correspondientes recuperatorios

Tabla 5

### 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. Para cada parcial y/o su recuperatorio se evaluará:

- ✓ Interpretación del problema e identificación de estructuras de datos adecuadas para su resolución.
- ✓ Diagramación y su aplicación en problemas de programación en computadora.
- ✓ Programación Orientada a Objetos y diagramas UML
- ✓ Adecuado uso del Lenguaje de Programación JAVA

### 7.3.3. Escala de Valoración

La escala de valoración a emplear en los parciales y recuperatorios será cualitativa dicotómica (aprobado – desaprobado)

## 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:

- Aprobar los parciales previstos o sus correspondientes recuperatorios o el recuperatorio integral
- Asistir al menos al 75 % de las clases teóricas, las clases prácticas y Taller

### 7.7. Examen Final

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

### 7.8. Examen Libre

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

Etapa 1: Evaluación escrita

- a) Desarrollo de un trabajo final en lenguaje JAVA, con evaluación en computador, cuya temática y planteo deberá ser solicitado a la cátedra con al menos 7 días de anticipación a la fecha de examen. El trabajo se presentará en la fecha del examen y deberá ser aprobado por el tribunal.
- b) Aprobar una evaluación escrita de tipo practica.
- c) Aprobar una evaluación de tipo practica en computador

Etapa 2: Evaluación oral

Se utiliza la misma modalidad que rige para los alumnos regulares.

*MSc.Ing. Rosa Palavecino*  
Prof. Asociado Interino