



**A.N.E.P.**

**Consejo de Educación Técnico Profesional**  
(Universidad del Trabajo del Uruguay)

	DESCRIPCIÓN	CÓDIGO
<b>TIPO DE CURSO</b>	EDUCACIÓN MEDIA TECNOLÓGICA	049
<b>PLAN</b>	2004	2004
<b>ORIENTACIÓN</b>	INFORMÁTICA	480
<b>SECTOR DE ESTUDIOS</b>	INFORMÁTICA	013
<b>AÑO</b>	1er	1er
<b>MÓDULO</b>	N/C	N/C
<b>ÁREA DE ASIGNATURA</b>	TÉCNICAS INFORMÁTICAS	925
<b>ASIGNATURA</b>	PROGRAMACIÓN I	3501
<b>ESPACIO CURRICULAR</b>	ESPACIO CURRICULAR TECNOLÓGICO	-----

<b>TOTAL DE HORAS/CURSO</b>	64
<b>DURACIÓN DEL CURSO</b>	32 semanas
<b>DISTRIB. DE HS /SEMANALES</b>	2

<b>FECHA DE PRESENTACIÓN</b>	25-06-2012
<b>FECHA DE APROBACIÓN</b>	
<b>RESOLUCIÓN CETP</b>	

**PROGRAMA PLANEAMIENTO EDUCATIVO**  
**ÁREA DISEÑO Y DESARROLLO CURRICULAR**

## 1-FUNDAMENTACIÓN

Este curso pretende introducir al estudiante en el aprendizaje de la programación y la aplicación de sus conocimientos en el Lenguaje Java.

Se ha elegido Java como lenguaje de programación por ser el lenguaje multiplataforma más difundido del mercado, que ha ido avanzando junto con la tecnología actual y se aplica en diversos medios como PC, Celular, Tablet, Domótica y otros. Las características principales que nos ofrece Java son:

- **Lenguaje Simple:** Viene de la misma estructura de C y C++; ya que C++ fue un referente para la creación de Java por eso utiliza determinadas características del mismo y se han eliminado otras.
- **Orientado a Objeto:** La programación en Java en su mayoría está orientada a objetos, ya que al estar agrupados en estructuras encapsuladas es más fácil su manipulación.
- **Distribuido:** Permite conectarse con servidores o clientes remotos lo que facilita la creación de aplicaciones distribuidas a través de una colección de clases para aplicaciones en red.
- **Robusto:** Es altamente confiable en comparación con C, se han eliminado muchas características inestables como la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.
- **Seguro:** La seguridad es una característica muy importante en Java ya que se han implementado barreras de seguridad en el lenguaje y en el sistema de ejecución de tiempo real.
- **Indiferente a la arquitectura:** Java es compatible con los más variados entornos, cualquiera sean estos, desde Windows 95, Unix, Windows Nt y Mac. Es muy versátil ya que utiliza byte-codes que es un formato intermedio que sirve para transportar el código a diferentes plataformas (Hardware - Software).
- **Portable:** Por ser indiferente a la arquitectura sobre la cual está trabajando, esto hace que su portabilidad sea muy eficiente, sus programas son iguales en cualquiera de las plataformas, ya que Java corre a través de una máquina virtual.
- **Interpretado y compilado a la vez:** Java puede ser compilado e interpretado en tiempo real, ya que cuando se construye el código fuente este se transforma en una especie de código de máquina.
- **Multihilo:** Usando su función de multihilos, por cada hilo que el programa tenga se ejecutarán en tiempo real muchas tareas al mismo tiempo.
- **Dinámico:** En la fase de enlazado, sus clases solamente actuarán en medida en que sean requeridas o necesitadas; esto permitirá que los enlaces se puedan incluir incluso desde fuentes muy variadas desde la red.
- **Applets:** Se pueden crear aplicaciones independientes y applets. Independientes porque se pueden comportar como cualquier programa escrito en cualquier lenguaje. Los applets considerados pequeños programas, tienen la capacidad de ejecutar funciones muy complejas.
- **Alto rendimiento:** Es considerado de alto rendimiento por ser tan veloz en el momento de correr los programas y por ahorrarse muchas líneas de código.

La asignatura Programación 1 mantiene desde 1997 en el bachillerato de Informática, un contenido orientado al Lenguaje C que es necesario modificar para mantener moderno y actualizado este curso.

Su característica principal de ser orientado a objetos genera un problema importante al enseñarse por el nivel de abstracción del alumno que recién empieza a programar. Esto plantea algunas problemáticas sobre el curso:

- La opción de dictar un lenguaje orientado a objetos pretende preparar de mejor manera al alumnado cuando llega a Programación III, donde se encontrará con un lenguaje Visual orientado a objetos. De esta manera se intenta obtener resultados más óptimos en este año culminante del bachillerato preparando al alumno desde primero.
- Como la mayoría del alumnado recién se inicia en la programación, comenzar trabajando con profundidad en objetos agrega más problemáticas a nuestro trabajo, por lo que se pretende apostar a la programación estructurada, trabajando los objetos Java al mínimo, dejando para Programación 2 una profundización del tema. La propuesta consiste en enseñar objetos, pero la mínima cantidad posible.
- La documentación adecuada para programación orientada a objetos es UML, pero se considera agregar demasiada complejidad para el primer curso de programación, por lo tanto se aconseja trabajar con Pseudocódigo (y solo Pseudocódigo) como herramienta de documentación y razonamiento para el alumno que se inicia.

Existe en Internet y en formato impreso mucho material sobre Java. Este curso tiene contenidos pensados y ordenados para que el estudiante de primero aprenda los conocimientos básicos de programación aplicado a Java.

Como apoyo del docente se puede descargar desde el sitio web de UTU ([www.utu.edu.uy](http://www.utu.edu.uy) / Publicaciones / Didácticas) una Guía para el docente de Programación 1.

## 2-COMPETENCIAS

- Desarrollar un razonamiento lógico aplicado a la resolución de algoritmos.
- Implementar algoritmos en Lenguaje Java.
- Aprender conocimientos básicos de programación Java en memoria estática.
- Desarrollar en el laboratorio hábitos y costumbres que le permitirán ubicarse adecuadamente en un ámbito profesional, respetando el trabajo de otros tanto como el propio, así como el equipamiento disponible, observando las normas establecidas al respecto.
- Reconocer la falibilidad de los programas realizados, aprendiendo del error y actuando en consecuencia para mejorar los mismos.
- Desarrollar capacidades de comunicación, cooperación y trabajo en equipo.

## 3- CONTENIDOS

### Introducción

#### Unidad 1: ¿Qué es un programa?

Lenguaje de Programación  
Software. Formas. Licencia. Software Libre  
Una clasificación de los lenguajes de programación  
Pseudocódigo

#### Unidad 2: Algoritmos y Programación Estructurada

Algoritmos  
Programación Estructurada  
GOTO

#### Unidad 3: Introducción a Java

El Lenguaje Java

Conceptos Básicos de Objetos  
Eclipse  
JDK

#### **Unidad 4: Primer Programa en Java**

#### **Unidad 5: Variables**

Conceptos básicos  
Nombre de la variable  
Tipos de datos  
Declaración de Variables  
Asignación  
Listar  
Cargar  
CAST

#### **Unidad 6: Sentencia IF**

Instrucciones de control

#### **Unidad 7: Estructuras repetitivas**

La estructura while  
La estructura do while

#### **Unidad 8: Menú**

Sentencia switch  
Indentado  
Estilo de indentación

#### **Unidad 9: Operadores Lógicos**

Los operadores relacionales  
Los operadores lógicos

#### **Unidad 10: Control de datos ingresados**

Banderas  
WHILE(TRUE)

#### **Unidad 11: La estructura for**

Contadores  
For Anidado  
ASCII  
Números al azar

#### **Unidad 12: Métodos (de la clase)**

¿Qué es un método?  
Variables locales y de clase

#### **Unidad 13: Array**

Declaración  
Inicializar  
Recorrer  
Cargado Total  
Asignación Directa

Constantes

#### **Unidad 14: String**

Obtener información del string  
Comparación de strings  
Extraer un substring de un string  
Convertir un número a string  
Convertir un string en número  
Scanner y String  
Reverse  
Tratamiento de Excepciones

#### **Unidad 15: Objetos dentro de array**

Calificadores: Public, Private, Protected, Static y Final.  
Arrays con clases

#### **Unidad 16: Obligatorio y defensa**

Tutoría del Obligatorio

### **4-PROPUESTA METODOLÓGICA**

Los objetivos antes mencionados se desarrollarán en clases teórico-prácticas dictadas en el laboratorio de informática, con una carga horaria de 2 horas semanales.

Asimismo cada alumno deberá dedicar un promedio de 3 horas semanales más al estudio domiciliario y práctica, en el laboratorio o en su casa.

El estudiante analizará problemas que le serán presentados con un grado creciente de dificultad, utilizando distintas metodologías. Se sugiere usar casos de estudio que se refieran a situaciones que se presentan en la vida real para su mejor apropiación.

Se sugiere a los docentes realizar un curso lo más práctico posible, partiendo de bases teóricas, aplicando las mismas mediante la resolución de problemas: “a programar se aprende programando”.

Se sugiere hacer una rápida introducción de las 3 primeras unidades en pocas clases, pasando a la resolución de problemas directamente en Java.

El curso debe ir desde una introducción de algoritmos y programación, hasta el manejo de arrays unidimensionales.

A su vez el coordinar con las asignaturas paralelas de Lógica e Introducción permiten desde otra dimensión educativa complementar los conocimientos que los estudiantes necesitan, potenciando el aprendizaje en esta asignatura

### **5-EVALUACIÓN**

Se entiende por Evaluación al proceso sistemático y planificado de recogida de información relativa al proceso de aprendizaje de los alumnos, al proceso de enseñanza, al centro educativo, etc., para su posterior valoración, de modo que el docente pueda tomar las decisiones oportunas sobre la base de los datos recabados (reconducción, ajuste, etc.)

La evaluación como componente básico en los procesos de enseñanza y aprendizaje, tiene que atender todos los aspectos a partir de los cuales se ha iniciado el mismo. Por eso ha de ser coherente con las características de las etapas, con los objetivos del curso y con la metodología que se ha usado a lo largo del proceso.

La evaluación permite averiguar ¿cómo va el curso? ¿cuánto aprenden los alumnos? ¿dónde tienen carencias? ¿si hay que darle mayor o menor profundidad a los contenidos? ¿quiénes necesitan un trabajo más personalizado?...

La evaluación se puede clasificar en tres etapas:

1. **Diagnóstica/Inicial.** Orientada a recabar información sobre las capacidades de partida y sus conocimientos previos en relación con un nuevo aprendizaje, para de este modo adecuar el proceso de enseñanza a sus posibilidades. Suele utilizarse normalmente con finalidades pronósticas, y por lo tanto al inicio de un período de aprendizaje.

Se debe recabar información sobre:

- Los conocimientos técnicos previos de los alumnos
- Datos sociales (del alumno y de su entorno)

En esta etapa es necesario determinar: ¿qué sabe el alumno? y ¿en qué contexto estudia? para de esa forma poder planificar ¿cómo les voy a enseñar?

2. **Formativa/Continua/Procesual.** Cuando se refiere a los aprendizajes de los alumnos, se orienta al ajuste y adaptación continuos del proceso de enseñanza a los procesos de aprendizaje de los alumnos en el momento en que estos se producen. Supone por tanto la recogida y el análisis continuo de información, de modo que se puedan introducir las reorientaciones y autocorrecciones precisas.

En este tipo de evaluación interesa, por tanto, verificar los errores, dificultades, ritmos de aprendizaje, logros, etc. de los alumnos, de modo que se pueda proporcionar de modo eficaz ayuda y refuerzo a la construcción de los aprendizajes.

Debe presentar las siguientes características:

- **Múltiple.** Se deben realizar muchas evaluaciones usando múltiples técnicas (orales, por observación, escritos, trabajos prácticos, trabajos en equipo, ...)
- **Continua** en cuanto que está inmersa en los procesos de enseñanza y aprendizaje del alumno o alumna. Se sugiere ir registrando en la libreta estas evaluaciones.
- **Integral** para considerar tanto la adquisición de nuevos conceptos como de procedimientos, actitudes, capacidades de relación y comunicación así como del desarrollo autónomo de cada estudiante.
- **Individualizada** centrándose en las particularidades de cada alumno y en su evolución.
- **Integradora** para lo cual tiene en cuenta las características del grupo a la hora de seleccionar los criterios de evaluación.
- **Cualitativa** ya que además de los aspectos cognitivos, se evalúan de forma equilibrada los diversos niveles de desarrollo del alumno.
- **Orientadora**, dado que aporta al alumnado la información precisa para mejorar su aprendizaje y adquirir estrategias apropiadas.

La evaluación, a pesar de registrarse en la libreta, no necesariamente se traduce en un número (calificación). Siéntase libre de registrar conceptos, características y cualidades, aspectos indispensables de las particularidades de nuestros estudiantes.

**3. Sumativa/Final.** Cuando se refiere a los aprendizajes de los alumnos, se orienta a determinar el grado de logro que un alumno ha obtenido en relación con los objetivos fijados para una área o etapa. Se realiza habitualmente, por tanto, al final de un proceso de enseñanza-aprendizaje, y se vincula a las decisiones de exoneración (nota del 7 al 12), examen diciembre (4, 5 o 6) o examen febrero (1, 2 y 3).

- También cubre finalidades estrictamente pedagógicas en la medida que permite establecer las situación de un alumno en relación con los objetivos y contenidos necesarios para afrontar con éxito futuros aprendizajes, constituyendo en este sentido el primer paso de un nuevo ciclo de evaluaciones diagnósticas, formativas y sumativas.
- Ante la duda si un alumno exonera el docente debe hacerse 2 preguntas:
- ¿El alumno llegó al “piso” del curso? (¿Sabe lo mínimo necesario para este curso?)
- ¿El alumno está preparado para el siguiente nivel? (¿Tiene las herramientas mínimas necesarias para cursar Programación II?)

Al finalizar el curso, se solicitará un trabajo (llamado Obligatorio), que deberá ser realizado por equipos, y defendido en forma individual. Este proyecto será propuesto a los alumnos con tiempo suficiente para una correcta realización, y será integrador de los conocimientos del curso.

Luego de entregada la letra (post vacaciones de primavera) el docente debe tutoriar los equipos atendiendo sus preguntas y generando conocimientos que le permitan exonerar la asignatura.

El docente debe darle todas las oportunidades para que el alumno exonere. No lo harán aquellos que teniendo todas las oportunidades no realicen los trabajos mínimos y no obtengan los conocimientos básicos del curso.

La realización y entrega del Obligatorio es condición sine qua non para exonerar la asignatura. El obligatorio consiste en realizar un programa que realice una tarea y que debe entregarse el programa (software) junto con el código fuente documentado e impreso.

En la defensa (noviembre) el docente puede considerar la eventualidad de no tomarle defensa a alguno de los integrantes de un equipo, si considera que este ha demostrado ampliamente los conocimientos mínimos requeridos para el curso.

### CRONOGRAMA

	Mar	Abr	May	Jun	Jul	Ago	Set	Oct	Nov
Ev Diagnóstica									
Contenidos									
Vectores									
Obligatorio									
Defensa									

## 6-BIBLIOGRAFÍA

- **[Arnold y Gosling, 1997]** Ken Arnold y James Gosling. Addison-Wesley/Domo. "El lenguaje de Programación Java". Wesley Iberoamericana. 1997. 334 páginas. (Muy básico, escrito por el desarrollador del lenguaje).
- **[Cortés et al.,1996]** José Luis Cortés, Nuria González, Virginia Pérez, Paqui Villena y Ana Rosa Freire. "Java, el lenguaje de programación de Internet". Data Becker 1996.
- **[Cuenca, 1996]** Pedro Manuel Cuenca Jiménez. "Programación en Java para Internet". Anaya Multimedia. 1996.
- **[Cuenca, 1997]** Pedro Manuel Cuenca Jiménez,. "Programación en Java". Ediciones Anaya Multimedia. 1997.
- **[Eckel, 1997]** Bruce Eckel. "Hands -on Java Seminar". President MindView Inc. 1997. 577 páginas. (Tutorial completo en Inglés en formato PDF).
- **[Framiñán, 1997]** José Manuel Framiñán Torres. "Manual Imprescindible de Java". Ediciones Anaya Multimedia. 1997.
- **[Froufe, 1997]** Agustín Froufe. "Tutorial de Java". Facultad de Informática de Sevilla. 1997. <http://www.fie.us.es/info/internet/JAVA/>.
- **[García et al., 1999]**. Javier García de Jalón, José Ignacio Rodríguez, Iñigo Mingo, Aitor Imaz, Alfonso Brazález, Alberto Larzabal, Jesús Calleja y Jon García. "Aprenda Java como si estuviera en primero". Universidad de Navarra. 1999. 140 páginas. (Tutorial muy básico, en el que se trata la potencia de Java, pero sin profundizar en ningún tema en particular).
- **[García, 1997]** Francisco José García Peñalvo. "Apuntes de teoría de la asignatura Programación Avanzada del tercer curso de Ingeniería Técnica en Informática de Gestión". Universidad de Burgos. 1997. 216 páginas.
- **[Johnson, 1996]** Jeff Johnson. "Coding Standards for C, C++, and Java". Vision 2000 CCS Package and Application Team. 1996. 14 páginas. (Consejos para el formato de los fuentes Java, C y C++).
- **[Lalani, 1997]** Suleiman 'Sam' Lalani. "Java, biblioteca del programador". Ediciones McGraw Hill. 1997.
- **[Morgan, 1999]** Mike Morgan. "Descubre Java 1.2". Prentice Hall. 1999. 675 páginas. (Sin duda muy interesante, sobre todo por su actualidad al tratar con Java 2, y por su extensión al tratar todas las bibliotecas de Java).
- **[Naughton, 1996]** Patrick Naughton. "Manual de Java". Mc. Graw Hill 1996. 395 páginas. (Introduce todos los aspectos de la programación básica en Java).
- **[Piattini et al., 1996]** Mario G. Piattini, José A. Calvo-Manzano, Joaquín Cervera y Luis Fernández. "Análisis y diseño detallado de Aplicaciones informáticas de gestión". Ra-Ma. 1996.
- **[Rambaugh et al., 1998]** J. Rambaugh J., M. Blaha, W. Premerlani, F. Eddy y W. Lorensen. "Modelado y Diseño Orientados a Objetos. Metodología OMT". Prentice Hall, 2º reimpresión. 1998.
- **[Rational, 1997]** Rational Software Corporation. "The Unified Modeling Language Documentation Set 1.1". www.rational.com. Setiembre de 1997.
- **[Rifflet, 1998]** Jean-Marie Rifflet "Comunicaciones en UNIX". McGraw Hill. 1998.
- **[Rojo, 1998]** Ignacio Rojo Fraile. "Comparativa de herramientas Java". Artículo de la revista Solo Programadores nº49. Tower. Octubre 1998. (Compara 5 IDEs Java).
- **[Sanz, 1998]** Javier Sanz Alamillo. "Novedades y cambios con Java 2". Artículo de la revista Solo Programadores nº55. Tower. Marzo 1999. (Buen resumen de los cambios que han surgido en el JDK 1.2).
  - **[Schildt 2005]** Schildt, Herbert "LA BIBLIA DE JAVA 2 V5.0" ANAYA MULTIMEDIA 1152 páginas. Idioma: Español ISBN: 8441518653. 1ª edición

- **[Sun, 1998] Sun Microsystems Inc.** "JDK 1.2 Documentation". www.sun.com. 1997. (*Documentación de la API de Java del JDK*).
- **[van Hoff et al., 1996] Arthur van Hoff, Sami Shaioi y Orca Starbuck.** "Hooked on Java". Addison-Wesley. 1996. (*Todo lo que hace falta saber para crear applets, con muchos ejemplos. En inglés*).
- **[Wu 2005] Thomas C. Wu** "INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS CON JAVA" Editorial McGraw-Hill. 2005. Idioma: Español ISBN: 8448131940. 864 páginas; 25x20 cm
- **[Zolli, 1997] Andrew Zolli.** "La biblia de Java". Anaya multimedia. 1997. 814 páginas. (*Completo en lo a que bibliotecas del JDK 1.1 se refiere*).