

PLANIFICACIÓN DE CÁTEDRA

2 0 2 4



TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN

UTN – F.R. Resistencia

Resistencia - Chaco

Asignatura: PROGRAMACIÓN II

Nivel: 1° SEGUNDO CUATRIMESTRE

Carga horaria: HORAS/SEMANA: 8 HS

Coordinador de la Carrera

Ing. Claudia Laclau

tup@frre.utn.edu.ar

www.frre.utn.edu.ar/tup

Docente/s:

Comisión 1

Ing. Blas Cabas Geat (Jefe de Trabajos Prácticos)

blasc147@gmail.com

Comisión 2

Ing. Teresita I. Vallejos (Jefe de Trabajos Prácticos)

ing_vallejos@yahoo.com.ar

Ing. Uferer Ferreyra Facundo

juanfacundouf@gmail.com

Comisión 3

TSP Jorge Wirz (Jefe de Trabajos Prácticos)

jwirz@frre.utn.edu.ar

bercha9@hotmail.com

Ing. Diana Carpintero

carpinterodiana@gmail.com

Reglamento de Estudio: Ord. 1622/2018

Diseño Curricular: Ord. 2018/2024

www.frre.utn.edu.ar

Tel 0362-4432928

French 414 (3500) Resistencia – Chaco

PLANIFICACIÓN DE CÁTEDRA

INTRODUCCIÓN

La intención que persigue la materia es lograr que los alumnos conozcan los principios básicos de la resolución de problemas apreciando el papel central que juega la abstracción en la tarea de programar proporcionando para esto los fundamentos teóricos y prácticos básicos para cursar posteriores estudios en programación. Durante el cursado los alumnos aprenderán uno de los paradigmas de programación más utilizados para la construcción y desarrollo de aplicaciones modernas de forma ágil, siendo este además el que les brindará las bases para aplicarlo con Lenguajes de Programación actuales como JAVA y .NET cubiertos durante los Laboratorios de Programación.

OBJETIVOS GENERALES DE LA ASIGNATURA

- Preparar los esquemas mentales para la resolución de problemas.
- Aprender un lenguaje de programación de alto nivel.
- Reconocer y emplear los conceptos básicos de la programación orientada a objetos.
- Identificar cuestiones de ingeniería de software que aplican a grandes sistemas.
- Mantener una cultura de trabajo en equipo.

PROGRAMA ANALÍTICO

UNIDAD 1 – INTRODUCCIÓN A JAVA Y VERSIONAMIENTO.

Contenidos:

Complejidad Algorítmica

Algoritmia

Fases en el desarrollo de un algoritmo.

Verificación y Análisis de Algoritmos.

Eficiencia de Algoritmos

Tiempo de ejecución y uso de memoria.

Comportamiento asintótico

Funciones de complejidad en tiempo más usuales.

Subalgoritmos o Subprogramas

Funciones

Procedimientos

Ámbito de variables Locales y Globales

Paso de Parámetros

Paso por Valor.

Paso por Parámetro.

Recursividad

Concepto y Tipos de Recursividad.

Cuando usar y cuándo no usar Recursividad.

Introducción a la Plataforma JAVA.

IDEs (on-line y locales) para JAVA. La API de JAVA.

Control de Versiones con Git.

Manejo básico de repositorios GIT y similares. Introducción a Git: Conceptos de control de versiones, instalación de Git, configuración de un repositorio. Uso de Git: Manejo de conflictos, ramas, fusiones y rebase.

UNIDAD 2 - PROGRAMACIÓN ORIENTADA A OBJETOS

Contenidos:

Fundamentos de la Programación Orientación a Objetos.

Clases y Objetos

Definición de clases. Instanciación de objetos. Diferencias.

Implementación de tipos de relaciones entre clases.

Definición del ciclo de vida de clases.

Definición del ciclo de vida de objetos.

Creación de variables de referencia y asignaciones.

Constructores

¿Qué es un constructor? Definición de constructor por defecto y constructores parametrizados.

Propiedades

Implementación de getters y setter para encapsular atributos de clases.

Modificadores de acceso

Definición de public, private y protected.

Pilares de la Orientación a Objetos

Herencia

Definición de herencia entre clases. Herencia Simple.

Polimorfismo

Implementación de polimorfismo de métodos. Ejemplos.

Sobrecarga

Implementación de sobrecarga de métodos. Ejemplos.

Conceptos avanzados

Clases Abstractas e Interfaces

Implementación de clases abstractas.

Implementación de interfaces.

Usos comunes de ambas. Ejemplos.

Diferencias entre clases abstractas e interfaces.

Paquetes y Namespaces

Comprender la agrupación de clases por funcionalidades.

Agrupaciones lógicas y físicas de clases.

Importar funcionalidades de otras clases a través de paquetes o namespaces.

Aplicación de la Programación Orientada a Objetos

Resolver escenarios de la vida real a través del modelado de clases.

Entender el concepto de contextos.

UNIDAD 3 – TDA. ESTRUCTURAS LINEALES. ESTRUCTURAS COMPLEJAS DE DATOS.

Contenidos:

Clasificación. Listas particulares y generalizadas. Organización y acceso.

Concepto de restricción. Noción de puntero.

Operaciones con Listas: recorrido, inserción, borrado.

Tipos Abstractos de Datos (TAD) y Diseño Orientado a Objetos.

Tipos de datos, estructuras de datos y tipos abstractos de datos.

Especificación de TADs.

El TAD conjunto.

Principios del diseño Orientado a Objetos.

Abstracción.

Encapsulamiento.

Modularidad.

Estructuras de Datos Dinámicas.

TAD Lista

Especificación formal del TAD Lista.

Implementación del TAD Lista con estructuras estáticas.

Implementación del TAD Lista mediante variables dinámicas.

Listas Doblemente Enlazadas.

Lista Circular mediante variables dinámicas.

TAD Pila

Especificación formal del TAD Pila.

Implementación del TAD Pila con arreglos.

Implementación del TAD Pila mediante variables dinámicas.

Aplicación de Pilas

Llamada a subprogramas o procedimientos.

Recursión.

Ordenamiento.

TAD Cola

Especificación formal del TAD Cola

Implementación del TAD Cola con arreglos.

Implementación del TAD Cola con Listas enlazadas y circulares.

Colas de Simulación.

Introducción a las Colecciones.

Clasificación y jerarquías de las Colecciones. Set o Conjuntos. List o listas. Queue o Cola. Map o Mapas. Iteradores. TreeSet. LinkedList o Lista Enlazada. ArrayList. Ordenamiento de Colecciones, Priority Queue, Métodos hashCode () y equals (). HashSet. HashMap.

UNIDAD 4 - TAD ÁRBOL ARCHIVOS Y FICHEROS

Contenidos:

Implantación de Árboles N-arios.

Transformar árboles n-arios a binarios.

Árboles Binarios

Definiciones recursivas.

Características de los Árboles: profundidad, niveles, nodos.

Implementación de Árboles Binarios.

Recorrido de un árbol binario.

Reconstrucción de árboles binarios a partir de dos de sus recorridos: Anchura y Profundidad.

Tipos de recorridos: pre-orden, in-orden o post-orden.

Árbol binario de búsqueda.

Implementación de Árboles binarios con vectores.

Manejo de ficheros y directorios.

Streams de datos. Serialización. Distribuir aplicaciones JAVA con archivos JAR y EXE. Java I/O: Lectura y escritura de archivos, uso de las clases File, FileReader, FileWriter, BufferedReader, BufferedWriter, manejo de excepciones de I/O.

UNIDAD 5 - TAD GRAFO y TABLAS HASH. PROGRAMACIÓN CONCURRENTE. TESTING, DEPURACIÓN Y BUENAS PRÁCTICAS

Contenidos:

Conceptos y Definiciones relacionadas con Grafos

Representación de los grafos.

Matriz de Adyacencia.

Lista de Adyacencia.

Recorridos de un Grafo.

Recorrido en Anchura o Amplitud.

Recorrido en Profundidad.

Algoritmos fundamentales con Grafos

Algoritmo de Dijkstra.

Algoritmo de Floyd.

Tablas Hash

Estructura interna de una Tabla Hash.

Estrategias de soluciones de Colisiones.

Concurrency en Java.

Introducción a los hilos: Creación de hilos mediante la interfaz Runnable y la clase Thread, control del ciclo de vida de un hilo. Sincronización de hilos: Manejo de problemas de concurrencia, uso de synchronized, deadlock. Creación de pools de hilos, ejecución de tareas.

Java y Bases de Datos

JDBC (Java Database Connectivity): Conexión a bases de datos, ejecución de consultas SQL. SQL y Java: Consultas SELECT, INSERT, UPDATE, DELETE. Preparación y ejecución de declaraciones: Uso de Prepared Statement para prevenir inyección SQL.

Introducción al testing.

Concepto de testing, niveles de prueba, introducción a JUnit, creación de pruebas unitarias. Depuración de programas Java: Uso de depuradores, seguimiento de variables y pilas de llamadas. Buenas prácticas de programación en Java: Principios de código limpio, refactorización de código.

ESTRATEGIAS METODOLÓGICAS

Desarrollo de la Asignatura.

Las instancias para el desarrollo de la asignatura serán clases teóricas y prácticas. Inicialmente se tratará de familiarizar al alumno con los conceptos básicos que se deben dominar para poder acceder al entendimiento de las bases de la programación. Para ello es necesario que el alumno adquiera una primera aproximación a la resolución de problemas, especialmente a aquellos del tipo algorítmico.

Dinámica del Dictado de las Clases Híbridas

El dictado de la cátedra será de carácter teórico y práctico a través de Clases Híbridas, utilizando Videoconferencia con los Alumnos remotos y el Aula Informática para los presenciales, estableciéndose en primer lugar los conceptos teóricos seguido de una aplicación inmediata por medio de estudio dirigido, brindando ejemplos presentados en la pizarra, así también como compartiendo la Pantalla del Monitor para que los Alumnos puedan apreciar correctamente los conceptos, posteriormente estos podrán realizar trabajos prácticos en sus respectivas PCs. Los alumnos podrán en todo momento comunicarse con el Profesor durante el cursado utilizando un Chat. El Profesor procederá a describir técnicas, características y pondrá ejemplos. Asimismo, habrá una parte de la clase dedicada a la aplicación de los conceptos vistos durante la parte teórica. Se desarrollarán problemas con creciente nivel de dificultad. Debido al contexto de carácter remoto, no se realizarán trabajos grupales, pero se brindarán ejercicios prácticos a través del Aula Virtual, además de Cuestionarios para Evaluaciones de Nivel. Los alumnos contarán además con una Cuenta de Correo Electrónico para comunicarse con el Profesor de forma asincrónica.

Ejercicios Prácticos.

Las actividades se plantean con presentación y resolución de problemas de baja complejidad con el objetivo de reafirmar conceptos vistos en teoría. En los laboratorios se trabajará sobre una guía de trabajos prácticos que abarcan los temas vistos en teoría. Los alumnos podrán resolver los ejercicios de la guía y revisar sus soluciones en clases con el Profesor, de esta manera se realizará una evaluación continua del progreso de los alumnos, estimulando además la autoevaluación por parte del estudiante, como medio idóneo para aumentar su rendimiento.

CRONOGRAMA COMISIÓN 1:

CLASE Nº	FECHA	CONTENIDO A SER DESARROLLADO
CLASE 1	06-08-2024	UNIDAD 1
CLASE 2	08-08-2024	UNIDAD 1
CLASE 3	09-08-2024	UNIDAD 1
CLASE 4	13-08-2024	UNIDAD 1
CLASE 5	15-08-2024	UNIDAD 1
CLASE 6	16-08-2024	UNIDAD 1
CLASE 7	20-08-2024	UNIDAD 1
CLASE 8	22-08-2024	UNIDAD 1
CLASE 9	23-08-2024	UNIDAD 2
CLASE 10	27-08-2024	UNIDAD 2(Exámenes Finales 4º llamado)
CLASE 11	30-08-2024	UNIDAD 2(Exámenes Finales 4º llamado)
CLASE 12	03-09-2024	UNIDAD 2
CLASE 13	05-09-2024	UNIDAD 2
CLASE 14	06-09-2024	UNIDAD 2
CLASE 15	10-09-2024	UNIDAD 2
CLASE 16	12-09-2024	UNIDAD 2
CLASE 17	13-09-2024	UNIDAD 3
CLASE 18	17-09-2024	UNIDAD 3
CLASE 19	19-09-2024	UNIDAD 3
CLASE 20	20-09-2024	UNIDAD 3
CLASE 21	24-09-2024	UNIDAD 3
CLASE 22	26-09-2024	UNIDAD 3
CLASE 23	27-09-2024	UNIDAD 3
CLASE 24	01-10-2024	Exámenes Finales 5º llamado.
CLASE 25	03-10-2024	Exámenes Finales 5º llamado.
CLASE 26	04-10-2024	Exámenes Finales 5º llamado.
CLASE 27	08-10-2024	PRIMER PARCIAL TEÓRICO/PRÁCTICO
CLASE 28	10-10-2024	UNIDAD 4
CLASE 29	11-10-2024	UNIDAD 4
CLASE 30	15-10-2024	UNIDAD 4
CLASE 31	17-10-2024	UNIDAD 4
CLASE 32	18-10-2024	UNIDAD 4
CLASE 33	22-10-2024	UNIDAD 4
CLASE 34	24-10-2024	UNIDAD 4
CLASE 35	25-10-2024	UNIDAD 4
CLASE 36	29-10-2024	UNIDAD 5
CLASE 37	31-10-2024	UNIDAD 5
CLASE 38	01-11-2024	UNIDAD 5
CLASE 39	05-11-2024	UNIDAD 5
CLASE 40	07-11-2024	UNIDAD 5
CLASE 41	08-11-2024	UNIDAD 5
CLASE 42	12-11-2024	UNIDAD 5
CLASE 43	14-11-2024	UNIDAD 5
CLASE 44	15-11-2024	UNIDAD 5
CLASE 45	19-11-2024	UNIDAD 5
CLASE 46	21-11-2024	UNIDAD 5
CLASE 47	22-11-2024	UNIDAD 5
CLASE 48	26-11-2024	SEGUNDO PARCIAL TEORÍA
CLASE 49	28-11-2024	SEGUNDO PARCIAL PRACTICA
CLASE 50	29-11-2024	PRIMER RECUPERATORIO
CLASE 51	03-12-2024	SEGUNDO RECUPERATORIO

CRONOGRAMA COMISIÓN 2:

CLASE Nº	FECHA	CONTENIDO A SER DESARROLLADO
CLASE 1	05-08-2024	UNIDAD 1
CLASE 2	07-08-2024	UNIDAD 1
CLASE 3	12-08-2024	UNIDAD 1
CLASE 4	14-08-2024	UNIDAD 1
CLASE 5	19-08-2024	UNIDAD 1 Día de la UTN
CLASE 6	21-08-2024	UNIDAD 2
CLASE 7	26-08-2024	Exámenes finales 4 llamado
CLASE 8	28-08-2024	Exámenes finales 4 llamado
CLASE 9	02-09-2024	UNIDAD 2
CLASE 10	04-09-2024	UNIDAD 2
CLASE 11	09-09-2024	UNIDAD 2
CLASE 12	11-09-2024	UNIDAD 2
CLASE 13	16-09-2024	UNIDAD 3
CLASE 14	18-09-2024	UNIDAD 3
CLASE 15	23-09-2024	UNIDAD 3
CLASE 16	25-09-2024	UNIDAD 3
CLASE 17	30-09-2024	Exámenes Finales 5º llamado.
CLASE 18	02-10-2024	Exámenes Finales 5º llamado.
CLASE 19	07-10-2024	UNIDAD 4
CLASE 20	9-10-2024	UNIDAD 3
CLASE 21	14-10-2024	UNIDAD 3
CLASE 22	16-10-2024	UNIDAD 3
CLASE 23	21-10-2024	UNIDAD 3
CLASE 24	23-10-2024	PRIMER PARCIAL TEORÍA/PRACTICO
CLASE 25	28-10-2024	UNIDAD 4
CLASE 26	30-10-2024	UNIDAD 4
CLASE 27	04-11-2024	UNIDAD 4
CLASE 28	06-11-2024	UNIDAD 4
CLASE 29	11-11-2024	UNIDAD 4
CLASE 30	13-11-2024	UNIDAD 5
CLASE 31	18-11-2024	UNIDAD 5 Día de la Soberanía Nacional
CLASE 32	20-11-2024	UNIDAD 5
CLASE 33	25-11-2024	UNIDAD 5
CLASE 34	27-11-2024	UNIDAD 5
CLASE 35	02-12-2024	SEGUNDO PARCIAL TEORÍA/PRÁCTICA
CLASE 36	04-12-2024	PRIMER RECUPERATORIO / SEGUNDO RECUPERATORIO/EXAMEN INTEGRADOR

CRONOGRAMA COMISIÓN 3:

CLASE Nº	FECHA	CONTENIDO A SER DESARROLLADO
CLASE 1	06-08-2024	UNIDAD 1
CLASE 2	08-08-2024	UNIDAD 1
CLASE 3	13-08-2024	UNIDAD 1
CLASE 4	15-08-2024	UNIDAD 2
CLASE 5	20-08-2024	UNIDAD 2
CLASE 6	22-08-2024	UNIDAD 2
CLASE 7	27-08-2024	Exámenes finales 4º llamado
CLASE 8	29-08-2024	Exámenes finales 4º llamado
CLASE 9	03-09-2024	UNIDAD 2
CLASE 10	05-09-2024	UNIDAD 2
CLASE 11	10-09-2024	UNIDAD 2
CLASE 12	12-09-2024	UNIDAD 3
CLASE 13	17-09-2024	UNIDAD 3
CLASE 14	19-09-2024	UNIDAD 3
CLASE 15	24-09-2024	UNIDAD 3
CLASE 16	26-09-2024	UNIDAD 3
CLASE 17	01-10-2024	Exámenes Finales 5º llamado.
CLASE 18	03-10-2024	Exámenes Finales 5º llamado.
CLASE 19	08-10-2024	PRIMER PARCIAL TEORÍA/PRACTICO
CLASE 20	10-10-2024	UNIDAD 4
CLASE 21	15-10-2024	UNIDAD 4
CLASE 22	17-10-2024	UNIDAD 4
CLASE 23	22-10-2024	UNIDAD 4
CLASE 24	24-10-2024	UNIDAD 4
CLASE 25	29-10-2024	UNIDAD 5

CLASE 26	31-10-2024	UNIDAD 5
CLASE 27	05-11-2024	UNIDAD 5
CLASE 28	07-11-2024	UNIDAD 5
CLASE 29	12-11-2024	UNIDAD 5
CLASE 30	14-11-2024	UNIDAD 5
CLASE 31	19-11-2024	UNIDAD 5
CLASE 32	21-11-2024	UNIDAD 5
CLASE 33	26-11-2024	UNIDAD 5
CLASE 34	28-11-2024	UNIDAD 5
CLASE 35	03-12-2024	SEGUNDO PARCIAL TEORIA/PRÁCTICO
CLASE 36	05-12-2024	PRIMER RECUPERATORIO / SEGUNDO RECUPERATORIO/EXAMEN INTEGRADOR

ORGANIZACIÓN DE ESPACIOS DENTRO Y FUERA DEL ÁMBITO UNIVERSITARIO

Para las clases se requerirá un laboratorio de computación que permita la utilización de proyector multimedia o pantalla LED/LCD, cuente con conectividad a Internet, y que permita la conformación de grupos de trabajo.

RECURSOS NECESARIOS:

Laboratorio de Computación con proyector/LED/LCD y conexión a Internet.

Software de desarrollo instalado en la PCs

- JAVA JDK o JAVA SDK versión 8.0 o superior
- IntelliJ
- BlueJ
- NetBeans (opcional)
- Eclipse (opcional)

EVALUACIÓN: INSTRUMENTOS Y MODALIDAD- RÉGIMEN DE PROMOCIÓN

Se contemplan 5 instancias de evaluación (IE) organizadas de la siguiente manera:

- **IE1:** Primer examen parcial (parte teórica)
- **IE2:** Primer examen parcial (parte práctica)
- **IE3:** Segundo examen parcial (parte teórica)
- **IE4:** Segundo examen parcial (parte práctica)
- **IE5:** Elaboración y presentación de un Trabajo Práctico Integrador

APROBACIÓN DIRECTA

Las condiciones para aprobar directamente la materia son:

- 75% de asistencia a las clases teóricas y prácticas.
- Cumplir con todas las tareas encomendadas en clases.
- La aprobación de **TODAS las IE** con nota igual a 6 (seis) o superior.
- Se podrá **recuperar una sola IE** desaprobada.
- En el caso en que no haya aprobado alguna de las instancias de evaluación parcial, tendrá al menos una instancia de recuperación.
- La nota promedio de las instancias de evaluación así obtenida será la calificación definitiva de aprobación de la asignatura.

APROBACIÓN NO DIRECTA (Regularidad y Exámen Final)

El estudiante que no alcanzó los objetivos de Aprobación Directa, pero que ha demostrado niveles mínimos y básicos de aprendizaje (Aprobación de la Cursada) estará habilitado a rendir un Examen Final.

- **Aprobación de la Cursada**

Las condiciones para aprobar la cursada de la materia son:

- 75% de asistencia a las clases teóricas y prácticas.

- Cumplir con todas las tareas encomendadas en clases.
 - La aprobación de al menos 3 (tres) IE con nota igual a 6 (seis) o superior, debiendo ser algunas de las 3 las siguientes:
 - Al menos una IE del primer parcial.
 - Al menos una IE del segundo parcial.
 - La IE 5 (TPI).
 - Existirá una instancia de recuperación para cada IE desaprobada, según cronograma.
-
- **Exámen Final**
Los alumnos que solo hayan aprobado la cursada, deberán rendir un examen final integrador, en las fechas asignadas para los exámenes finales debiendo desarrollar y aprobar primero una instancia de evaluación escrita y luego otra instancia oral.

NO APROBACIÓN (Libre)

El estudiante que no ha demostrado niveles mínimos y básicos de aprendizaje deberá recursar la asignatura. No está habilitado a rendir examen final.

CORRELATIVIDADES

Para Cursar

Cursada

Programación I

Aprobada

Ninguna

Para Rendir

Aprobada

Programación I

BIBLIOGRAFÍA

GENERAL

- García de Jalón de la Fuente, J. (1999). Aprende Java como si estuviera en primero. San Sebastián: Universidad de Navarra. Escuela Superior de Ingenieros Industriales.
- Schildt, H., & Patarroyo Puentes, J. (2002). Fundamentos de programación en Java 2. Bogotá: Osborne/McGraw-Hill.
- Jorge Martínez Ladrón de Guevara (2016). Fundamentos de programación en Java. Facultad de Informática Universidad Complutense de Madrid / Editorial EME.
- SUN MICROSYSTEMS INC. El lenguaje de programación Java. Guía Práctica. Sun Learning Services. SL – 275. Revisión F. 2005
- SUN MICROSYSTEMS INC. El lenguaje de programación Java. Manual del Estudiante. Sun Learning Services. SL – 275. Revisión F. 2005
- JOYANES AGUILAR, LUIS. Programación Orientada a Objetos. McGraw-Hill. 2002

DE CONSULTA

- MATERIAL PROVISTO POR EN CLASES QUE CONSTA DE TEORÍA, EJEMPLOS DE CÓDIGO JAVA Y EJERCICIOS, LOS CUALES SON DE LIBRE ACCESO PARA CUALQUIER ALUMNO.
- THINKING IN JAVA - 5ta EDICIÓN o Superior (Unidades 3, 4 y 5)
- JOYANES AGUILAR, LUIS, Fundamentos de Programación: Algoritmos y Estructura de Datos. Mcgraw-Hill. 1996.
- ALLEN WEISS, MARK. Estructura de Datos y Algoritmos. Addison-Wesley Iberoamericana. 1995.
- AHO, ALFRED V; HOPCROFT, JOHN E; ULLMAN, JEFFRY D. Estructura de Datos y Algoritmos. Addison-Wesley Iberoamericana. 1990.
- CAIRO, OSVALDO. Estructura de Datos. McGraw-Hill. 2002.
- HEILEMAN, GREGORY. Estructura de Datos, Algoritmos y Programación orientada a objetos. McGraw-Hill. 1998.
- RUSSELL. Introduction to Computing and Algorithms. Addison-Wesley. 1998.
- AHO, ALFRED. Estructura de datos y algoritmos. Addison-Wesley. 1988.
- OBJECT-ORIENTED ANALYSIS AND DESIGN WITH APPLICATIONS. BLOOCH, GRADY. SECOND EDITION, THE BENJAMIN/CUMMINGS PUBLISHING COMPANY, INC. 1994.
- OBJECT-ORIENTED MODELING AND DESIGN. JAMES RUMBAUGH ET AL. PRENTICE-HALL, INC. 1991.
- ARNOLD, GOSLING, HOLMES. Holmes. El lenguaje de programación Java. Addison-Wesley. 3ra. Edición 2002.
- ECKEL, BRUCE. Pensando en Java. Prentice-Hall. 2006.