

**REDICTADO TALLER DE
PROGRAMACIÓN**

Año 2022

Carrera/ Plan:*Licenciatura en Informática Plan 2021/Plan 2015/Plan 2012/Plan 2003-07**Licenciatura en Sistemas Plan 2021/Plan 2015/Plan 2012/Plan 2003-07**Analista Programador Universitario Plan2021/Plan 2015/Plan 2007**Analista en Tecnologías de la Información y la Comunicación Plan2021/Plan 2017***Año:** 1°**Régimen de Cursada:** Semestral**Carácter (Obligatoria/Optativa):** Obligatoria**Correlativas:** Conceptos de Algoritmos, Datos y Programas**Profesor/es:** Victoria Sanz, Waldo Hasperué**Hs. semanales:** 6 (teoría y práctica)**FUNDAMENTACIÓN**

Se trabaja a partir de diferentes situaciones problemáticas de la vida real y se abordan las estrategias de solución bajo criterios de calidad, eficiencia y corrección, para finalmente llegar a una implementación acorde a las especificaciones planteadas. Todos los aspectos mencionados son abordados a partir del trabajo con diferentes paradigmas de programación.

Los conceptos abordados en la asignatura permitirán al alumno familiarizarse con todos estos aspectos vinculados a la tarea de la programación básica ya que se constituyen en uno de los conceptos básicos de la disciplina.

OBJETIVOS GENERALES

- Realizar desarrollo de programas simples en el paradigma imperativo.
- Extender el manejo de datos a datos no lineales (Árboles).
- Introducción de los conceptos básicos de un segundo paradigma de programación (orientación a objetos) con énfasis en la noción de reusabilidad.
- Desarrollo de programas simples en un lenguaje orientado a objetos.
- Introducción de los conceptos básicos de la Programación Concurrente
- Desarrollo de programas simples con un lenguaje de programación concurrente que permita interpretar los conceptos de comunicación y sincronización entre procesos.
- Combinar los elementos estudiados previamente en Conceptos de Algoritmos, Datos y programas con las tareas experimentales en diferentes lenguajes de programación, a fin de que el alumno complete el ciclo del problema a su solución con computadora.

RESULTADOS DE APRENDIZAJE

- 2.1. Utilizar una serie de técnicas con las que identificar las necesidades de problemas reales, analizar su complejidad y evaluar la viabilidad de las posibles soluciones mediante técnicas informáticas (Básico)
- 2.2. Describir un determinado problema y su solución a varios niveles de abstracción (Básico)
- 2.4. Escoger los patrones de solución, algoritmos y estructuras de datos apropiados (Básico)
- 3.5. Aplicar las correspondientes competencias prácticas y de programación en la creación de programas informáticos y/u otros dispositivos informáticos (Básico).

5.5. Diseñar y llevar a cabo investigaciones prácticas (por ejemplo, de rendimientos de sistemas) para interpretar datos y extraer conclusiones (Básico).

COMPETENCIAS

- CGS1- Desempeñarse de manera efectiva en equipos de trabajo, con capacidad para organizarlos y liderarlos.
- CGS2- Comunicarse con efectividad en forma oral y escrita.
- CGS4- Aprender en forma continua y autónoma, con capacidad de planificar este aprendizaje.
- CGT1- Identificar, formular y resolver problemas de Informática.
- CGT10- Capacidad para realizar investigaciones bibliográficas y de diferentes fuentes de información a fin de obtener conocimiento actualizado en temas de la disciplina.
- CGT5- Utilizar de manera efectiva las técnicas y herramientas de aplicación de la Informática.
- LI- CE4 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LS- CE1 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.

CONTENIDOS MÍNIMOS (de acuerdo al Plan de Estudios)

- Estructuras de datos no lineales. Árboles.
- Conceptos de programación imperativa.
- Desarrollo de programas en un lenguaje imperativo.
- Implementación de algoritmos fundamentales (búsqueda, ordenación).
- Conceptos de programación orientada a objetos.
- Análisis de las características fundamentales de un lenguaje orientado a objetos.
- Desarrollo de programas en un lenguaje orientado a objetos.
- Conceptos básicos de concurrencia y paralelismo.
- Procesos. Comunicación y sincronización entre procesos.
- Desarrollo de programas concurrentes/paralelos.

PROGRAMA ANALÍTICO

A. Programación Estructurada

Estructuras de datos no lineales.

Tipo de dato Árboles. Definición y terminología asociada. Características. Operaciones.

Implementación de algoritmos fundamentales sobre estructuras de datos estáticas y dinámicas: búsquedas, ordenación, merge.

Desarrollo de programas en un lenguaje imperativo (Pascal).

B. Programación orientada a objetos

Introducción a la POO.

Concepto de Objeto (estado y comportamiento), Clase e Instancia, Constructores.

Concepto de Herencia.

Desarrollo de programas simples en un lenguaje orientado a objetos (Java)

C. Programación Concurrente

Conceptos básicos de concurrencia y paralelismo.

Procesos. Comunicación y sincronización entre procesos.

Desarrollo de programas concurrentes/paralelos utilizando el ambiente del multirobot (r-info).

BIBLIOGRAFÍA

Algoritmos, datos y programas con aplicaciones en Pascal, Delphi y Visual Da Vinci.

De Giusti, Armando et al. 1er edición. Prentice Hall 2001.

Estructuras de Datos y Algoritmos.

Hernández R., Dormido R., Lazaro J. Ros S. Pearson Education. 2000.

Introduction to algorithms

Comen, Leiserson. MIT Press 2001.

Estructuras de Datos y Algoritmos.

Aho Alfred, Hopcroft John y Ullman Jeffrey. Addison Wesley Publishing Company. EUA. 1998.

Programación en Pascal

Joyanes Aguilar, Luis. Mc Graw Hill. 2006

Fundamentos de Programación. Libro de Problemas.

Joyanes Aguilar L., Fernandez M., Rodríguez L. Mc Graw Hill. 2003.

Data structures, algorithms and software principles.

Standish, T. A. Addison Wesley Publishing Company. 1994.

Estructuras de Datos y Algoritmos

Weiss, M.A. Addison Wesley. 1995.

Fundamentos de Programación.

Joyanes Aguilar L., Fernandez M., Rodríguez L. Mc Graw Hill. 1999.

Algoritmos y estructuras de datos y programación orientada a objetos.

Flórez Rueda. Ecoe Ediciones. Bogotá. 2005. ISBN 958648394/0

Programación En C Metodología, Algoritmos Y Estructura De Datos.

Joyanes Aguilar Luis – Zahonero Martínez. Segunda Edición –Editorial Mc Graw Hill. España - Edición 2007

An introduction to object-oriented programming

Timothy Budd. Addison Wesley. 2002.

Programación orientada a objetos con Java

David J. Barnes, Michael Kolling. Tercera Edición. Pearson Educación, 2007 Bruce Eckel: Thinking in Java. Cuarta Edición. Prentice Hall, 2006.

Thinking in C++.

Bruce Eckel. Segunda Edición. Prentice Hall, 2000.

Data Structures and Algorithm Analysis in Java

Mark Allen Weiss. Tercera Edición. Addison-Wesley. 2012

Bibliografía Adicional

Introduction to Computer Science with applications in Pascal.

Garland, S.J. Addison Wesley Publishing Company. 1986.

Estructuras de Datos.

Franch Gutierrez, Xavier. Alfaomega Grupo Editor Argentino. 2002

Estructura de Datos.

Joyanes Aguilar C., Zahonero Martinez I. Mc Graw Hill. 1998.

Estructuras de Datos. Libro de Problemas.

Joyanes Aguilar L., Fernandez M., Rodríguez L. Mc Graw Hill. 1999.

Estructuras de Datos.

Lipschutz, S. Mc Graw Hill. 1997.

Programación estructurada en Turbo Pascal 7.

Lopez Roman, L. Alfaomega Grupo Editor Argentino. 1998.

Estructuras de Datos.

Martinez Román, Quiroga Elda. Thomson International. 2002

Estructura de Datos y Algoritmos.

Sisa, Alberto Jaime. Editorial Prentice. 2002.

Pascal Estructurado.

Tremblay, Jean Paul. Mc Graw Hill. 1980.

Data structures, algorithms and performance.

Wood, D. Addison Wesley Publishing Company. 1993.

Structures and Algorithm Analysis in Java

Weiss, M.A. Data, 3rd Edition, Pearson/Addison Wesley, 2011

Data Structures and Algorithms using C#.

M. McMillan. Cambridge University Press, 2006

Sitios de interés: <http://csunplugged.org>

<http://www.eduteka.org>

METODOLOGÍA DE ENSEÑANZA

A los fines de la organización de la cátedra se propone que los alumnos que podrán asistir a este curso son:

- Alumnos que figuren con nota DESAPROBADO en el SIU-GUARANI en la cursada de Taller de Programación 2021, habiéndose presentado a rendir los parciales en las 3 fechas.
- Ingresantes (en agosto 2021) que hayan APROBADO el curso de CADP 2021 (2do semestre)

El curso de Redictado de Taller de Programación se divide en 3 módulos: Programación Imperativa, Programación Orientada a Objetos y Programación Concurrente.

Cada módulo del Taller tiene una duración estimada de 8 clases con una carga semanal de 2 clases de 3 hs. cada una. Cada clase consta de contenidos teórico-prácticos con actividades en máquina para resolver en el aula y también fuera del horario de clase.

El curso se organiza en dos turnos, cubriendo horarios de mañana y tarde. Cada turno tiene asignado un aula y un horario determinados. Cada turno está a cargo de un docente responsable del dictado de los contenidos teóricos-prácticos y cuenta con auxiliares docentes para las consultas de las actividades en máquina.

En la cátedra se organizan actividades por equipos de trabajo, con 2 alumnos en las actividades teórico-prácticas. Este número puede incrementarse hasta 4 alumnos para los trabajos que se desarrollen fuera del aula. En principio los alumnos son "pares" sin roles determinados en el equipo, aunque dado un problema a resolver, ellos pueden autorregularse. Cada equipo será responsable de una notebook que utilizará para el desarrollo de los ejercicios prácticos.

Los equipos deben demostrar capacidad de aprender (a partir de casos problema) en forma continua y autónoma, teniendo la posibilidad de consultar a sus docentes en el aula y vía WEB por el entorno virtual de enseñanza y aprendizaje IDEAS.

Los alumnos de cada comisión/equipo deben realizar un trabajo escrito documentado de cada uno de los problemas que resuelven y defender en un coloquio (en el que las preguntas son individuales) la solución realizada. La cátedra mantiene planillas que permiten calificar diferentes aptitudes de los miembros del equipo (conocimientos / capacidad de autoaprendizaje / modo de expresarse / predisposición al trabajo colaborativo). Estas planillas son compartidas por los docentes responsables de la asignatura para ser tenidas en cuenta en las evaluaciones parciales y finales de los alumnos.

En el seguimiento y evaluación de los alumnos se trata de formarlos en una metodología de ir del "caso problema del mundo real" a su solución efectiva con herramientas informáticas. Para ello se pone énfasis en la selección del paradigma a utilizar según el tipo de problema y en la metodología de análisis, diseño, programación, documentación y verificación de cada algoritmo implementado por el alumno en cada lenguaje del Taller. Este seguimiento incluye el planteo de temas que el alumno debe buscar en la bibliografía (tales como soluciones alternativas o modos de documentar una solución o técnicas de debugging de código).

Dado el contenido del programa que se enfoca en los paradigmas imperativo (utilizando PASCAL), orientado a objetos (utilizando JAVA) y Concurrente (utilizando CMRE) el alumno es evaluado por todos los aspectos relacionados con las Competencias Generales Tecnológicas que corresponden y por las cuestiones de la competencia específica que abarca la asignatura, constanding el resultado de esta evaluación en la corrección de las pruebas (parciales y finales) del alumno. Se pone énfasis en detallar los aspectos técnicos que debe perfeccionar hacia el futuro en asignaturas que correlacionan con Taller de Programación

Asistencia a Clases

- La asistencia a las clases teórico-prácticas es obligatoria.
- En cada clase del Redictado de Taller de Programación los alumnos tendrán presente, ausente, o ausente justificado. Los ausentes justificados no pasan a ser presentes.
- La asistencia a cada clase será tomada una única vez durante el horario de clase. Si un alumno no se encuentra en el aula por cualquier motivo, tendrá ausente.
- Pueden justificarse ausentes solamente por razones de salud, presentando certificado otorgado por Hospital Público.
- El certificado, para ser tenido en cuenta, debe ser entregado al docente encargado del aula INDEFECTIBLEMENTE la semana posterior a la reincorporación a clase, debiéndose respetar esta condición para que el certificado sea aceptado.
- El alumno que no cumpla con la condición de asistencia de un módulo no podrá cursar los restantes módulos y el resultado de la cursada del Redictado del Taller será AUSENTE.

Evaluaciones Breves en la clase

Durante la clase se propone la realización de ejercicios prácticos que los alumnos deben resolver y enviar, por el entorno IDEAS, al auxiliar a su cargo. La evaluación de estas actividades servirá de información para los docentes y de orientación para el alumno. El rendimiento satisfactorio de los alumnos en estas pruebas será considerado, a favor del alumno, durante la instancia de evaluación final del Redictado de Taller de Programación.

EVALUACIÓN

- La evaluación consiste de un parcial teórico-práctico al final del cuatrimestre.
- Para poder rendir el parcial, el alumno deberá cumplir con el 80% de asistencia a las clases teórico-práctica de cada módulo.
- El parcial contará con una parte práctica a resolver en máquina y preguntas teóricas a desarrollar sobre los paradigmas vistos.
- El parcial se APROBARÁ con nota 4 o superior.
- El parcial tendrá 2 recuperatorios, al que podrán acceder los alumnos que hayan DESAPROBADO en alguna instancia previa.

Aprobación de la cursada

Se otorgará la cursada a aquellos alumnos que hayan APROBADO el parcial en cualquiera de las 3 fechas disponibles.

Aprobación del Examen Final

Los alumnos que hayan aprobado el parcial en cualquiera de las 3 fechas disponibles con nota 6 o superior, tendrán aprobado el examen final. En este caso deberán inscribirse y presentarse en una mesa de examen final y para ello tienen tiempo hasta la mesa de marzo de 2023, inclusive.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Semana	Contenidos/Actividades
1	14/3	Presentación del Taller.
2	14/3	Ordenación en Arreglos. Método de ordenación por intercambio. Ejercitación
3	21/3	Merge
4	28/3	Merge acumulador
5	28/3	Recursión. Concepto. Características.
6	4/4	Árboles Binarios Ordenados. Concepto. Operaciones. Ejercitación
7	4/4	Conceptos básicos del lenguaje Java.
8	11/4	Introducción a la POO
9	18/4	Conceptos básicos de POO utilizando Java.
10	18/4	Constructores, Interacción entre objetos.
11	25/4	Concepto de herencia.
13	25/4	Conceptos básicos de Concurrencia. Entorno R-info.
14	2/5	Memoria distribuida
15	2/5	Memoria compartida
16	9/5	CONSULTA GENERAL
17	9/5	CONSULTA GENERAL

Evaluaciones previstas	Semana
Parcial	23/5
1era Fecha Recuperatoria	13/6
2da Fecha Recuperatoria	4/7

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

Contacto: <http://ideas.info.unlp.edu.ar>



Waldo Hasperué



Victoria Sanz