

**REDICTADO TALLER DE  
PROGRAMACIÓN****Año 2021****Carrera/ Plan:**

*Licenciatura en Informática* Plan 2015/Plan 2012/Plan 2003-07  
*Licenciatura en Sistemas* Plan 2015/Plan 2012/Plan 2003-07  
*Analista Programador Universitario* Plan 2015/Plan 2007  
*Analista en Tecnologías de la Información y la Comunicación*  
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 2020.
- Ingresantes (en agosto 2020) que hayan APROBADO el curso de CADP 2020 (2do semestre)
- Alumnos que hayan obtenido la cursada de CADP durante 2020 y hayan aprobado el EXAMEN FINAL al 28 de febrero de 2021.

Visto el contexto actual, las clases se llevarán a cabo en modalidad a distancia. Se utilizará el entorno virtual de enseñanza-aprendizaje IDEAS, donde se publicará todo el material necesario para el desarrollo de las clases y avisos. Para las actividades sincrónicas se utilizará Webex o similar.

La asignatura se estructura con clases teóricas y prácticas experimentales, en las cuales se desarrolla el contenido de los 3 módulos/paradigmas (Programación Imperativa, Programación Orientada a Objetos y Programación Concurrente) del Programa Analítico.

### **Teoría:**

Una vez a la semana se publicará en IDEAS la teoría (video y diapositivas) que contiene el tema a abordar en la semana. Este material puede ser accedido asincrónicamente por los alumnos.

En forma posterior, se planificará un encuentro sincrónico semanal donde los alumnos podrán realizar consultas y despejar dudas sobre el material teórico publicado de la semana. Asimismo el docente resolverá colaborativamente con los alumnos un ejercicio práctico. Los alumnos deben haber leído/visualizado previamente los contenidos de la clase para un efectivo aprovechamiento del encuentro.

La asistencia a los encuentros sincrónicos de teoría NO ES OBLIGATORIA.

### **Práctica:**

Se planificará un encuentro sincrónico semanal, en el cual los alumnos podrán realizar consultas sobre los trabajos prácticos y sobre el uso de las herramientas de programación.

La asistencia a los encuentros sincrónicos de práctica NO ES OBLIGATORIA.

## **EVALUACIÓN**

### **Auto-evaluaciones (Teórico-Prácticas):**

- Cada semana se publicará en el entorno IDEAS (o medio similar) una evaluación teórico-práctica con los temas vistos en la semana.
- El alumno tendrá un tiempo determinado para responder la autoevaluación.
- En total son 9 autoevaluaciones (3 por módulo/paradigma).
- Las calificaciones obtenidas en las auto-evaluaciones determinan la habilitación a rendir el parcial. Además las calificaciones podrán tenerse en cuenta (a favor del alumno) en la corrección de su parcial si resulta dudoso.

Parcial:

- Para poder rendir el parcial, el alumno deberá haber rendido al menos 6 autoevaluaciones (de las 9 disponibles) y tener al menos 1 auto-evaluación APROBADA por cada paradigma.
- 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 5 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 7 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 2022, inclusive.

**CRONOGRAMA DE CLASES Y EVALUACIONES**

Clase	Semana	Contenidos/Actividades
1	29/3	Presentación del Taller. Ordenación en Arreglos. Método de ordenación por intercambio. Ejercitación.
2	29/3	Práctica
3	5/4	Recursión. Concepto. Características. Árboles Binarios Ordenados. Concepto. Operaciones. Ejercitación.
4	5/4	Práctica
5	12/4	Operación de Merge en Listas y Vectores. Ejercitación.
6	12/4	Práctica
7	19/4	Conceptos básicos del lenguaje Java. Introducción a la POO
8	19/4	Práctica
9	26/4	Conceptos básicos de POO utilizando Java. Constructores, Interacción entre objetos.
10	26/4	Práctica
11	3/5	Concepto de herencia.
12	3/5	Práctica



13	10/5	Conceptos básicos de Concurrencia. Entorno R-info.
14	10/5	Práctica
15	17/5	Memoria distribuida
16	17/5	Práctica
17	24/5	Memoria compartida
18	24/5	Práctica
19	31/5	CONSULTA GENERAL

Evaluaciones previstas	Semana
Parcial	7/6
1era Fecha Recuperatoria	21/6
2da Fecha Recuperatoria	12/7

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

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

Dra. Victoria M. Sanz  
Prof. Adjunto

Dr. Waldo Hasperué  
Prof. Adjunto