

**Carrera/ Plan:****TALLER DE PROGRAMACION**

*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 TIC Plan 2017*

**Año:** 1°

**Régimen de Cursada:** Semestral

**Carácter (Obligatoria/Optativa):** Obligatoria

**Correlativas:** Conceptos de Algoritmos, Datos y Programas

**Profesor/es:** Gladys Gorga, Laura De Giusti, Virginia Ainchil,  
Waldo Hasperué, Victoria Sanz, Eduardo Ibañez

**Año 2019**

**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 (Arboles).
- 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.

**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 interfaces 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 interfaces humano computador y computador-computador.

### **CONTENIDOS MINIMOS (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**

El curso 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 taller se divide en diferentes turnos dependiendo de la cantidad de alumnos inscriptos, 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 (considerando que se trata de alumnos del segundo semestre del primer año de sus carreras) 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 de Taller 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.

### 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 Taller.

## **EVALUACIÓN**

### **Aprobación de un módulo**

- Cumplir con el 80% de asistencia a las clases teórico-práctica
- Asistir los días correspondientes a la resolución del trabajo final
- Resolver un trabajo final (a definir) durante el horario del Taller.
- Aprobación del trabajo final y Coloquio con nota igual o superior a 4.

El alumno que no cumpla con la asistencia y/o no asista los días destinados a la resolución del trabajo final tendrá AUSENTE en el módulo correspondiente. En este caso, el alumno no podrá cursar los restantes módulos y el resultado de la cursada de Taller será AUSENTE.

Cada módulo tendrá como posibles resultados "AUSENTE", "APROBADO" o "DESAPROBADO". El resultado "AUSENTE" en alguno de los módulos implicará que el alumno esté AUSENTE en la cursada de la asignatura. En este caso, el alumno no podrá cursar el Redictado de Taller en el primer cuatrimestre de 2020.

### **Aprobación de la cursada**

- Se otorgará la cursada del Taller a aquellos alumnos que hayan APROBADO los tres módulos en la primera instancia.
- Se otorgará la cursada del Taller a aquellos alumnos que hayan APROBADO dos módulos y DESAPROBADO uno de los tres módulos en la primera instancia.
- En el caso que, el alumno tenga solo uno de los tres módulos APROBADO y haya DESAPROBADO los otros dos módulos, tendrá una UNICA instancia recuperatoria en la que se evaluarán temas correspondientes a esos módulos. La aprobación de esta instancia recuperatoria implicará la aprobación de la cursada.

### **Aprobación del Examen Final**

- Aquellos alumnos que hayan aprobado los tres módulos (con nota 6 o más en cada uno), en la primera instancia, deberán inscribirse y presentarse en una mesa de examen final para dar por cumplida la última etapa del Taller. Para cumplir con este paso, tienen tiempo hasta el mes de julio de 2020 inclusive. Para acceder a esta última etapa, el alumno debe tener aprobados los exámenes finales de las materias correlativas.

**CRONOGRAMA DE CLASES Y EVALUACIONES**

Clase	Fecha	Contenidos/Actividades
1	21/08	<b>Presentación del Taller</b>
2	26/08	Operación de Merge en Listas y Vectores
3	28/08	Ordenación en Arreglos. Método de ordenación por intercambio.
4	2/09	Recursión. Concepto. Características. Ejercitación.
5	4/09	Arboles Binarios Ordenados. Concepto. Operaciones. Ejercitación.
6	9/09	Arboles Binarios Ordenados. Ejercitación
7	16/09	Consultas
8	18/09	Resolución del Trabajo Final
9	23/09	Coloquios
10	25/09	Conceptos básicos del lenguaje Java
11	30/09	Introducción a la POO
12	2/10	Conceptos básicos de POO utilizando Java.
13	7/10	Constructores, Interacción entre objetos.
14	9/10	El concepto de herencia.
15	16/10	Consultas
16	21/10	Resolución del Trabajo Final
17	23/10	Coloquios
18	28/10	Conceptos básicos de Concurrencia
19	30/10	Entorno R-info
20	4/11	Memoria distribuida
21	6/11	Memoria compartida
22	11/11	Memoria distribuida y compartida
23	13/11	Consultas
24	20/11	Resolución del Trabajo Final
25	25/11	Coloquios
26	27/11	Consultas



27	2/12	Consultas
28	4/12	Consultas

Evaluaciones previstas	Fecha
Parcial (Fecha Recuperatoria)	9/12

**Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):**Contacto: [progra@lidi.info.unlp.edu.ar](mailto:progra@lidi.info.unlp.edu.ar)Página: <http://weblidi.info.unlp.edu.ar/catedras/adp>Blog: <http://blogs.unlp.edu.ar/adp>Entorno virtual: <http://ideas.info.unlp.edu.ar>

Firma del/los profesor/es



**REDICTADO TALLER DE  
PROGRAMACION**

Año 2019

**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 TIC* 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 (Arboles).
- 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.

**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 interfaces 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 MINIMOS (de acuerdo al Plan de Estudios)**

- Estructuras de datos no lineales. Arboles.
- 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 Arboles. 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 multibot (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 2018.
- Ingresantes (en agosto 2018) que hayan APROBADO el curso de CADP 2018 (2do semestre)
- Alumnos que hayan cursado el Redictado CADP2018 y tengan aprobado el EXAMEN FINAL al 31 de marzo de 2019.

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, constando 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.

## **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**

### **Aprobación de un módulo**

- Cumplir con el 80% de asistencia a las clases teórico-práctica
- Asistir los días correspondientes a la resolución del trabajo final
- Resolver un trabajo final (a definir) durante el horario del Taller.
- Aprobación del trabajo final y Coloquio con nota igual o superior a 4.

El alumno que no cumpla con la asistencia y/o no asista los días destinados a la resolución del trabajo final tendrá AUSENTE en el módulo correspondiente. En este caso, el alumno no podrá cursar los restantes módulos y el resultado de la cursada del Redictado del Taller será AUSENTE.

Cada módulo tendrá como posibles resultados "AUSENTE", "APROBADO" o "DESAPROBADO". El resultado "AUSENTE" en alguno de los módulos implicará que el alumno esté AUSENTE en la cursada de la asignatura.

### **Aprobación de la cursada**

- Se otorgará la cursada del Redictado Taller a aquellos alumnos que hayan APROBADO los tres módulos en la primera instancia.
- Se otorgará la cursada del Taller a aquellos alumnos que hayan APROBADO dos módulos y DESAPROBADO uno de los tres módulos en la primera instancia.
- En el caso que, el alumno tenga solo uno de los tres módulos APROBADO y haya DESAPROBADO los otros dos módulos, tendrá una UNICA instancia recuperatoria en la que se evaluarán temas correspondientes a esos módulos. La aprobación de esta instancia recuperatoria implicará la aprobación de la cursada.

### **Aprobación del Examen Final**

- Aquellos alumnos que hayan aprobado los tres módulos (con nota 6 o más en cada uno), en la primera instancia, deberán inscribirse y presentarse en una mesa de examen final para dar por cumplida la última etapa del Redictado de Taller. Para cumplir con este paso, tienen tiempo hasta el mes de MARZO de 2020 inclusive. Para acceder a esta última etapa, el alumno debe tener aprobados los exámenes finales de las materias correlativas.

**CRONOGRAMA DE CLASES Y EVALUACIONES**

Clase	Fecha	Contenidos/Actividades
1	25/03	<b>Presentación del Taller</b>
2	27/03	Operación de Merge en Listas y Vectores
3	1/04	Ordenación en Arreglos. Método de ordenación por intercambio.
4	3/04	Recursión. Concepto. Características. Ejercitación.
5	8/04	Arboles Binarios Ordenados. Concepto. Operaciones. Ejercitación.
6	10/04	Arboles Binarios Ordenados. Ejercitación
7	15/04	Resolución del Trabajo Final
8	22/04	Coloquios
9	24/04	Coloquios
10	29/04	Conceptos básicos del lenguaje Java
11	6/05	Introducción a la POO
12	8/05	Conceptos básicos de POO utilizando Java.
13	13/05	Constructores, Interacción entre objetos.
14	15/05	El concepto de herencia.
15	20/05	Resolución del Trabajo Final
16	22/05	Coloquios
17	27/05	Coloquios
18	29/05	Conceptos básicos de Concurrencia
19	3/06	Entorno R-info
20	5/06	Memoria distribuida
21	10/06	Memoria compartida
22	12/06	Memoria distribuida y compartida
23	19/06	Resolución del Trabajo Final
24	24/06	Coloquios
25	26/06	Coloquios
26	1/07	Consultas

27	3/07	Consultas
28	10/07	Consultas

Evaluaciones previstas	Fecha
Parcial (Fecha Recuperatoria)	15/07

**Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):**Contacto: [progra@lidi.info.unlp.edu.ar](mailto:progra@lidi.info.unlp.edu.ar)Página: <http://weblidi.info.unlp.edu.ar/catedras/adp>Blog: <http://blogs.unlp.edu.ar/adp>Entorno virtual: <http://ideas.info.unlp.edu.ar>

Firma del/los profesor/es