

TALLER DE PROGRAMACIÓN**Año 2025****Carrera/ Plan:***Licenciatura en Informática Planes 2021/2015**Licenciatura en Sistemas Planes 2021/2015**Analista Programador Universitario Planes 2021/2015**Analista en Tecnologías de la Información y la Comunicación Planes 2021/2017***Año:** 1°**Régimen de Cursada:** Semestral**Carácter (Obligatoria/Optativa):** Obligatoria**Correlativas:** Conceptos de Algoritmos, Datos y Programas**Profesor responsable:** Laura De Giusti**Hs. semanales:** 6 (clases teórico-prácticas)**FUNDAMENTACIÓN**

Se trabaja a partir de plantear situaciones problemáticas basadas en la vida real, abordando 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 implementados a partir de aplicar 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

- Desarrollar programas simples en el paradigma imperativo.
- Extender el manejo de datos a datos no lineales (Árboles).
- Incorporar conceptos básicos de un segundo paradigma de programación (orientación a objetos) con énfasis en la noción de reusabilidad.
- Desarrollar programas simples en un lenguaje orientado a objetos.
- Incorporar conceptos básicos de la Programación Concurrente
- Desarrollar 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 la asignatura 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 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 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 y ordenación.

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 INGRESANTES 2025

Se consideran ingresantes 2025 a aquellos alumnos que han obtenido la cursada de CADP EN 2025, o hayan aprobado CADP en el redictado del segundo semestre en el año 2024 (no hayan cursado el redictado de Taller de Programación 2025).

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 1.30 hs presencial cada una, a esto debe adicionarse el tiempo que el alumno necesite para comprender el material didáctico disponible para cada semana. 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 determinado.

Cada turno está a cargo de dos docentes responsables del dictado de los contenidos teóricos-prácticos y cuenta con auxiliares docentes para las consultas de las actividades en máquina.

Cada alumno será responsable de una notebook que utilizará para el desarrollo de los ejercicios prácticos.

Se publicará en IDEAS el material del curso (video y diapositivas) que contiene el tema a abordar en la clase, conforme un cronograma establecido. El alumno debe acceder al material previa asistencia a la clase.

Asistencia a Clases

La asistencia a las clases teórico-prácticas es obligatoria.

En cada clase de Taller de Programación los alumnos tendrán presente o ausente.

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.

Aprobación de un módulo

Para dar por aprobado un módulo el alumno debe:

- Cumplir con el 50% de asistencia a las clases teórico-práctica.
- Asistir el día correspondiente al examen del módulo.
- Aprobación del examen con nota igual o superior a 6.

El alumno que no cumpla con la asistencia no podrá rendir el examen y 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 de Programación será AUSENTE.

Si un alumno no concurre el día del examen tendrá como nota de ese módulo INSUFICIENTE.

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

Aprobación de la cursada

Se otorgará la cursada del Taller de Programación a aquellos alumnos que:

- Hayan APROBADO los tres módulos en la primera instancia ó
- Hayan APROBADO dos módulos y hayan obtenido DESAPROBADO/INSUFICIENTE en uno de los tres módulos en la primera instancia.

En el caso que, el alumno tenga sólo uno de los tres módulos APROBADO y haya obtenido DESAPROBADO/INSUFICIENTE en los otros dos módulos, tendrá una instancia recuperatoria en la que se evaluarán temas correspondientes a la materia. La aprobación de esta instancia recuperatoria implicará la aprobación de la cursada.

Aprobación del Examen Final (PROMOCION)

Se otorgará el final del Taller de Programación a aquellos alumnos que hayan APROBADO los tres módulos (con nota 6 o más en cada uno), en la primera instancia.

Además, para aquellos alumnos que hayan APROBADO dos módulos y DESAPROBADO el restante, habrá una instancia en la cual rendirá una evaluación que incluirá los temas vistos en la materia, en caso de aprobar esta instancia el alumno obtiene el final de la materia.

Para que el final sea pasado efectivamente al SIU-GUARANI el alumno deberá inscribirse en una mesa de examen final antes de junio de 2026 inclusive. Hay que recordar que para poder pasar en SIU-GUARANI el final de Taller de Programación el alumno debe tener aprobado el examen final de CADP (hasta mayo de 2026 inclusive).

Además, para aquellos alumnos que no obtienen la promoción para aprobar el final deberán rendir un examen teórico práctico en cualquier mesa de final durante la validez de su cursada.

METODOLOGÍA DE ENSEÑANZA RECURSANTES 2025

Se consideran recursantes 2025 a aquellos alumnos que no han sido incluidos como Ingresantes 2025.

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 1.30 hs presencial cada una, a esto debe adicionarse el tiempo que el alumno necesite para comprender el material didáctico disponible para cada semana. 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 determinado.

Cada turno está a cargo de dos docentes responsables del dictado de los contenidos teóricos-prácticos y cuenta con auxiliares docentes para las consultas de las actividades en máquina.

Cada alumno será responsable de una notebook que utilizará para el desarrollo de los ejercicios prácticos.

Se publicará en IDEAS el material del curso (video y diapositivas) que contiene el tema a abordar en la clase, conforme un cronograma establecido. El alumno debe acceder al material previa asistencia a la clase.

Asistencia a Clases

La asistencia a las clases teórico-prácticas NO es obligatoria.

Aprobación de un módulo

Para dar por aprobado un módulo el alumno debe:

- Cumplir con el 75% de las autoevaluaciones propuestas en el entorno IDEAS realizadas y un 50% de aprobadas (nota 6 o más).
- Asistir el día correspondiente al examen del módulo.
- Aprobación del examen con nota igual o superior a 6.

El alumno que no cumpla con las autoevaluaciones no podrá rendir el examen y 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 de Programación será AUSENTE.

Si un alumno no concurre el día del examen tendrá como nota de ese módulo INSUFICIENTE.

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

Aprobación de la cursada

Se otorgará la cursada del Taller de Programación a aquellos alumnos que:

- Hayan APROBADO los tres módulos en la primera instancia ó

- Hayan APROBADO dos módulos y hayan obtenido DESAPROBADO/INSUFICIENTE en uno de los tres módulos en la primera instancia.

En el caso que, el alumno tenga sólo uno de los tres módulos APROBADO y haya obtenido DESAPROBADO/INSUFICIENTE en los otros dos módulos, tendrá una instancia recuperatoria en la que se evaluarán temas correspondientes a la materia. La aprobación de esta instancia recuperatoria implicará la aprobación de la cursada.

Aprobación del Examen Final (PROMOCION)

Se otorgará el final del Taller de Programación a aquellos alumnos que hayan APROBADO los tres módulos (con nota 6 o más en cada uno), en la primera instancia.

Además, para aquellos alumnos que hayan APROBADO dos módulos y DESAPROBADO el restante, habrá una instancia en la cual rendirá una evaluación que incluirá los temas vistos en la materia, en caso de aprobar esta instancia el alumno obtiene el final de la materia.

Para que el final sea pasado efectivamente al SIU-GUARANI el alumno deberá inscribirse en una mesa de examen final antes de junio de 2026 inclusive. Hay que recordar que para poder pasar en SIU-GUARANI el final de Taller de Programación el alumno debe tener aprobado el examen final de CADP (hasta mayo de 2026 inclusive).

Además, para aquellos alumnos que no obtienen la promoción para aprobar el final deberán rendir un examen teórico práctico en cualquier mesa de final durante la validez de su cursada.

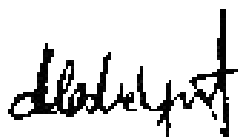
CRONOGRAMA DE EVALUACIONES

Las clases comienzan la semana del 25/08/2025 en los horarios y días correspondiente a cada turno.

Evaluaciones previstas	Semana
Evaluación primer módulo	15/09/2025 (el día depende del turno al que asiste el alumno)
Evaluación segundo módulo	13/10/2025 (el día depende del turno al que asiste el alumno)
Evaluación tercer módulo	10/11/2025 (el día depende del turno al que asiste el alumno)
Evaluación recuperatoria	5/12/2025 (igual para todos los turnos)
Evaluación recuperatoria PROMOCIÓN	5/12/2025 (igual para todos los turnos)

Mail de comunicación: tallerprogra@lidi.info.unlp.edu.ar

Entorno utilizado por la materia para la comunicación e información: IDEAS



De Giusti Laura