

PROGRAMACIÓN II**Carrera/ Plan:** (Dejar lo que corresponda)

Ingeniería en Computación

Año 2019

Año: primero**Régimen de Cursada:** Semestral**Carácter (Obligatoria/Optativa):** obligatoria**Correlativas:** Programación I**Profesor/es:** Alejandro Héctor Gonzalez**Hs. semanales:** 8 hs**FUNDAMENTACIÓN**

Los conceptos y actividades abordadas en la asignatura permitirán al alumno familiarizarse con los aspectos vinculados a la tarea de la programación ya que se constituyen en uno de los conceptos básicos de la disciplina. Se promueve trabajar 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 los diversos paradigmas de programación para entender sus diferencias y similitudes.

OBJETIVOS GENERALES

Profundizar el análisis de problemas resolubles con computadora, poniendo énfasis en la modelización, abstracción de funciones y en la modularización de los mismos utilizando diferentes paradigmas de programación

Objetivos específicos:

- Desarrollo de programas simples en el paradigma imperativo articulando los conceptos aprendidos en Programación I.
- Incorporación del manejo de estructuras de datos a estructuras no lineales (Arboles).
- Introducción de los conceptos básicos de la programación orientación a objetos, con énfasis en la noción de reusabilidad.
- Desarrollo de programas simples en un lenguaje orientado a objetos.
- Analizar y desarrollar programas simples en bajo nivel.
- 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.
-

CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)

- Tipos de datos compuestos (estáticos y dinámicos, lineales y no lineales).
- Recursividad. Introducción a los tipos abstractos de datos. Computadoras digitales.
- Organización funcional. CPU. Nociones de circuitos combinatorios y secuenciales.
- Memoria interna y externa. Periféricos. Representación de datos a nivel de máquina.
- Nociones básicas de manejo de interrupciones.

PROGRAMA ANALÍTICO**A- Recursividad**

- Características.
- Ejecución de un programa y la pila de activación.
- Análisis comparativo entre soluciones iterativas y recursivas.
- Ejemplos.

B- Estructura de datos compuestas no lineales: árboles.

- ✓ Introducción al concepto de datos no lineales.
- ✓ Terminología y definiciones básicas del tipo de dato árbol.
- ✓ Árboles binarios. Representación y operaciones.
- ✓ Árboles binarios ordenados. Representación y operaciones.
- ✓ Problemas que combinen árboles, listas y arreglos.

C- Memoria y periféricos

- ✓ Tipos de memorias, clasificación.
- ✓ Memoria principal, formas de organización. Memoria Cache.
- ✓ Memoria secundaria, organización y formato de datos. Organización jerárquica de la memoria.
- ✓ Tipos de periféricos comparaciones.

D- Organización funcional de la computadora

- ✓ Manejo de Instrucciones.
- ✓ Modos de direccionamiento.
- ✓ Lenguaje ensamblador.
- ✓ Práctica sobre simulador de un procesador.
- ✓ Subrutinas y pasaje de parámetros.
- ✓ Conceptos de interrupciones.

E- Introducción a la programación orientada a objetos

- ✓ Abstracción de datos y conceptos sobre tipos de datos.
- ✓ Introducción a la Programación Orientada a Objetos
- ✓ Reusabilidad de soluciones.
- ✓ Abstracción de datos y procesos.
- ✓ La noción de Objeto. Operaciones (métodos) aplicables a un objeto.
- ✓ Concepto de clases e instancias.
- ✓ Noción de herencia y polimorfismo. Relación con el re-uso.
- ✓ Introducción al lenguaje JAVA

F- Introducción a la Concurrencia.

- ✓ Impacto del cambio de los procesadores en el software.
- ✓ Diferentes arquitecturas de procesadores.
- ✓ Concurrencia.
- ✓ Comunicación y Sincronización entre procesos.
- ✓ Relación con la evolución de las arquitecturas de procesadores.
- ✓ Expresión de la concurrencia en los lenguajes de programación.

G - Conceptos finales

- ✓ Metodologías de resolución de problemas.
- ✓ Noción de paradigmas de programación.
- ✓ Evolución de los lenguajes de programación. Alternativas.

BIBLIOGRAFÍA

Título: Diseño de algoritmos con implementaciones en Pascal y C

Autores: Nivo Ziviani, Joaquín Adiego

Editorial: Paraninfo

Año de edición : 2007

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

Autores: De Giusti, Armando et al.

Editorial: Prentice Hall

Año de edición : 2002

Título: Programación orientada a objetos con JAVA usando BlueJ

Autores: Barnes, David; Kölling, Michael

Editorial: Prentice Hall

Año de edición : 2013

Título: Programación estructurada y orientada a objetos

Autores: Lopez Roman Leobardo

Editorial: Alfa Omega Grupo editor

Año de edición : 2011

Título: Java. Como programar
Autores: Paul Deitel, Harvey Deitel
Editorial: Perason
Año de edición : 2012

Título: Organización y Arquitectura de Computadoras. 7ma edición
Autores: William Stallings
Editorial: Pretince Hall
Año de edición : 2016

Título: Organización de Computadoras
Autores: Andrew Tanenbaum
Editorial: Pretince Hall
Año de edición : 2006

Título: Problemas resueltos de estructura de computadores
Autores: Jesus Carretero, Perez Felix García Caballeira, Jose Daniel Garcia Sanchez, David Exposito Singh
Editorial: Paraninfo
Año de edición : 2015

BIBLIOGRAFIA COMPLEMENTARIA

Análisis y Diseño Orientado a Objetos con Aplicaciones.
Booch Grady. Addison Wesley. 1996.

Arquitectura de computadores- Un enfoque cuantitativo.
Hennessy & Patterson. Ed. Mc Graw Hill (1ra edición).2002

Estructuras de datos: referencia práctica con orientación a objetos.
Román Martínez, Elda Quiroga. Editorial Thomson, 2002.

Estructuras de Datos y Algoritmos.
Hernández R., Dormido R., Lazaro J. Ros S. Pearson Education. 2000.

Fundamentos de Programación.
Joyanes Aguilar L., Fernandez M., Rodríguez L. Mc Graw Hill. 1999.
Introducción a la programación con Pascal.
José Manuel Iñesta Quereda. Universitat Jaume I, 2000
Introduction to algorithms.
Comen, Leiserson. MIT Press 2001.

Programación Orientada a Objetos.
Joyanes Aguilar, L. Mc Graw Hill., 1998

Recursividad, Complejidad y Diseño de Algoritmos.
Jesús Bisbal Riera. Editorial UOC. España. 2009

SITIOS DE INTERES:
<http://csunplugged.org>
<http://www.eduteka.org>

METODOLOGÍA DE ENSEÑANZA

La asignatura Programación II se organiza en 4 módulos: Programación Imperativa, Programación en Assembler, Programación Orientada a Objetos y Programación Concurrente.

Se propone la modalidad de trabajo de talleres en el aula y desarrollo de actividades en línea como parte de la extensión del aula presencial. El trabajo en el aula-taller se fundamenta en un aprendizaje activo donde el alumno se apropia de los conocimientos, y el docente cumple el rol de coordinador y realiza actividades de tutoría.

Cada uno de los módulos se trabaja con la modalidad de Aula-Taller utilizando el laboratorio móvil de computadoras y tiene una duración estimada de 7 clases con una carga horaria de 8 horas semanales con 2 clases de 4 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 utilizando el entorno virtual de enseñanza y aprendizaje (EVEA) IDEAS y herramientas digitales específicas para cada caso.

Las clases se desarrollan de manera integral donde participan los diferentes integrantes de la cátedra. El docente responsable organiza la asignatura en general y define la dinámica del taller en conjunto con los docentes de la materia. Trabajan en forma coordinada con los auxiliares docentes para definir estrategias de tutoría en las clases para las actividades en máquina.

La dinámica de las clases comprende una actividad inicial que estimule al estudiante y le aporte la problemática a tratar y el marco teórico involucrado. Un segundo momento de síntesis informativa que orienta y guía al alumno en la elaboración del conocimiento a través de ejemplos y casos. Y un tercer momento de actividades de integración y extensión con una guía de trabajo para ordenar las actividades individuales y grupales.

En las clases se conformarán equipos integrados por 2 alumnos para el trabajo en máquina. Cada equipo será responsable de una notebook del laboratorio móvil que utilizará para el desarrollo de los ejercicios prácticos.

El material teórico y la ejercitación práctica utilizados en el curso están disponibles en el EVEA IDEAS.

Asistencia a clases

- La asistencia a las clases teórico-prácticas es obligatoria
- En cada clase 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.

EVALUACIÓN

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 de la asignatura.

Aprobación de un módulo

- El alumno debe cumplir con el 80% de asistencia a las clases del módulo.
- Haber entregado dos ejercicios completos resueltos por computadora para el módulo a evaluar.
- Asistir los días correspondientes a la resolución del trabajo final.
- Resolver un trabajo final que se desarrolla durante el horario del Taller.
- Aprobación del trabajo final (teórico-práctico).

Cada módulo tiene su correspondiente instancia recuperatoria.

Condiciones de Aprobación de la asignatura

Se otorgará la cursada del Taller a aquellos alumnos que hayan cumplido con el 80% de asistencia a cada uno de los módulos y hayan presentado a rendir los cuatro módulos y aprobado al menos 3 (tres) de los 4 (cuatro) módulos. Si el alumno aprobó solo dos módulos podrá presentarse a una instancia recuperatoria global para alcanzar la cursada.

Si el alumno aprobó 0, 1 o 2 módulos debe recurrir a la asignatura.

Aprobación del final con promoción

Aquellos alumnos que aprobaron los 4 (cuatro) módulos tendrán promocionada la asignatura.

En caso de que el alumno tenga solo 3 (tres) de los módulos aprobados y haya cumplido con la asistencia y la presentación del trabajo final en los otros módulos tendrá una **UNICA instancia recuperatoria global** en la que se evaluará el tema correspondiente al módulo desaprobado y se le permitirá alcanzar la promoción.

Final de la asignatura

El final deberá ser rendido por aquellos alumnos que solo aprobaron 3 Módulos. El final es de carácter teórico y práctico.

CRONOGRAMA DE CLASES Y EVALUACIONES

Cada semana del curso incluye dos días de clases teórico-prácticas que se organizan de acuerdo al siguiente cronograma:

Módulo 1 - Programación Imperativa (Pascal)

Clase	Fecha	Contenidos	Actividades
Clase 1 (Imperativo)	07/08	Presentación del Taller.	Anotarse en Ideas Armado de grupos de trabajo
Clase 2 (Imperativo)	12/08	Recursión. Concepto. Características Ejercitación.	Resolución de ejercicios básicos utilizando recursión.
Clase 3 (Imperativo)	14/08	Arboles Binarios Ordenados. Concepto. Operaciones. Ejercitación	Implementación de un árbol binario ordenado.
Clase 4 (Imperativo)	21/08	Arboles Binarios Ordenados. Borrado. Ejercitación	Implementación de las operaciones básicas de árboles binarios ordenados en Pascal.
Clase 5 (Imperativo)	26/08	Ejercicios combinados de árboles binarios ordenados y listas	Árboles y listas ejercicios combinados
Clase 6 (Imperativo)	28/08	Ejercicios combinados de árboles binarios ordenados y lista. Segunda parte	Ejercicios combinados
Clase 7 (Imperativo)	02/09	Repaso para el TP final	
Clase 8 (Imperativo)	04/09	Resolución del Trabajo Final 1	Evaluación

Módulo 2 - Programación Orientada a Objetos (JAVA)

Clase	Fecha	Contenidos	Actividades
Clase 1 (Objetos)	09/09	Conceptos básicos del lenguaje Java	Implementación de programas simples imperativos en Java para ejercitar la sintaxis.
Clase 2 (Objetos)	16/09	Introducción a la POO utilizando Java. Manejo de arreglos	Ejercitación que comprende instanciación de objetos y envío de mensajes.
Clase 3 (Objetos)	18/09	POO utilizando Java	Ejercitación que comprende programación de nuevas clases, instanciación de objetos de dichas clases, envío de mensajes a dichos objetos.
Coloquios TP1	23/09	Actividad de recuperación del TP1	
Clase 4 (Objetos)	25/09	Constructores	Ejercitación que comprende la incorporación de constructores a las clases implementadas con

			anterioridad.
Clase 5 (Objetos)	30/09	Concepto de herencia y polimorfismo	Ejercitación con herencia.
Clase 6 (Objetos)	02/10	Repaso	Repaso de todos los conceptos de POO
Clase 7 (Objetos)	07/10	Resolución del Trabajo Final 2	Evaluación

Módulo 3 - Programación en bajo nivel (Assembler)

Clase	Fecha	Contenidos	Actividades
Clase 1 (Assembler)	09/10	Assembler del MSx88	Presentación del simulador para comprender el funcionamiento interno de la PC.
Coloquio TP2	16/10	Actividad de recuperación del TP 2	
Clase 2 (Assembler)	21/10	Assembler Modos de direccionamiento y saltos	Resolución de ejercicios que requieren diferentes modos de direccionamiento.
Clase 3 (Assembler)	23/10	Subrutinas y Tipos de pasaje de parámetros	Resolución de problemas a bajo nivel que requieren la modularización de las soluciones. Utilización de CALL, RET
Clase 4 (Assembler)	28/10	Pasaje de Parámetros	Resolución de problemas que requieren comunicación entre módulos utilizando diferentes implementaciones de pasaje de parámetros
Clase 5 (Assembler)	30/10	Repaso	Repaso
Clase 6 (Assembler)	04/11	Resolución del Trabajo Final 3	Evaluación

Módulo 4 - Programación Concurrente

Clase	Fecha	Contenidos	Actividades
Clase 1 (Concurrencia)	06/11	Conceptos básicos de Concurrencia	Ejercicios que muestran los problemas de concurrencia.
Clase 2 (Concurrencia)	11/11	Entorno CMRE	Trabajo con el entorno CMRE.
Clase 3 (Concurrencia)	13/11	Memoria distribuida	Trabajo con el entorno CMRE aplicando los conceptos vistos a ejercicios con memoria distribuida.
Clase 4 (Concurrencia)	20/11	Memoria compartida	Trabajo con el entorno CMRE aplicando los conceptos vistos a ejercicios con memoria compartida.
Coloquio TP3	25/11	Actividad de recuperación del TP 3	
Clase 5 (Concurrencia)	27/11	Memoria distribuida y compartida	Trabajo con el entorno CMRE aplicando los conceptos vistos a ejercicios con

			memoria distribuida y compartida.
Clase 6 (Concurrencia)	02/12	Resolución del Trabajo Final 4	Evaluación
Coloquio	04/12	Actividad de recuperación de TP 4	
Consulta	09/12	Consulta para recuperatorio global	
Recuperatorio	11/12	Recuperatorio Global	

Fechas de Evaluaciones

Tema	Fecha	Actividad
Imperativo	04/09	Resolución del Trabajo Final
Imperativo	23/09	Coloquios
Objetos	07/10	Resolución del Trabajo Final
Objetos	16/10	Coloquios
Assembler	04/11	Resolución del Trabajo Final
Assembler	25/11	Coloquios
Concurrencia	02/12	Resolución del Trabajo Final
Concurrencia	04/12	Coloquios
Recuperatorio Global	11/12	Examen recuperatorio

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

Profesor: Alejandro Héctor Gonzalez agonzalez@lidi.info.unlp.edu.ar

Blog de catedra: <http://blogs.unlp.edu.ar/programacion2/>

Plataforma virtual: Ideas, curso Programación 2

Firma del/los profesor/es



Mg. Alejandro Héctor Gonzalez