

PROGRAMACIÓN II**Carrera:***Ingeniería en Computación***Año:** *primero***Régimen de Cursada:** *Semestral (2do semestre)***Carácter (Obligatoria/Optativa):** *obligatoria***Correlativas:** *Programación I***Profesores:** *Alejandro Héctor Gonzalez***Hs. semanales teoría:** *3 hs***Hs. semanales práctica:** *5 hs***Año 2023****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 (Árboles).*
- *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 MÍNIMOS (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 combinan 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 los 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: PROGRAMACIÓN ORIENTADA A OBJETOS CON JAVA USANDO B (Spanish Edition) (Español) 1st Edición
Autores: David Barnes (Author), Michael Kolling (Author), S.L. Gestión editorial Avanzada (Translator)
Editorial: Pearson
Año de edición : 2017

Título: Lógica de programación: Solucionario en pseudocódigo–Ejercicios resueltos.
Autores: Buriticá, O. I. T. (2021).
Editorial: Ediciones de la U.

Año: 2021

Título: Programación
Autores: Alfonso Jiménez Marín; Francisco Manuel Pérez Montes
Editorial: Ediciones Paraninfo, S.A. ISBN: 9788428342865
Año de edición: 2021

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

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

Título: Aprende a programar con Java :un enfoque práctico partiendo de cero
Autores: Jiménez Marín, A., y Pérez Montes, F..
Editorial: Paraninfo
Año de edición: 2a ed., rev., 2016.

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

BIBLIOGRAFIA COMPLEMENTARIA

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

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.

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

Introduction to algorithms.
Comen, Leiserson. MIT Press 2001.

SITIOS DE INTERES:

<http://csunplugged.org>
<http://www.eduteka.org>

CONDICIONES DE LOS ESTUDIANTES PARA ACCEDER

Pueden cursar el redictado:

- ✓ Los alumnos que se le vencieron los trabajos prácticos de la asignatura en Julio del año en corriente.
- ✓ Los alumnos que tienen aprobado la cursada y/o el examen final de las correlativas del plan de estudio de la asignatura.

METODOLOGÍA DE ENSEÑANZA

La asignatura Programación II se organiza en 4 módulos: M1-Programación Imperativa, M2-Programación Orientada a Objetos, M3-Programación en Assembler y M4-Programación Concurrente.

Se propone la modalidad de trabajo de aulas-talleres virtuales y desarrollo de actividades en línea. 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 trabaja con la modalidad de Aula-Taller donde cada estudiante utiliza su computadora desde la casa, se conecta a una sesión Webex con el ayudante para desarrollar ejercicios en forma grupal y poder consultar. Cada taller virtual 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 taller virtual y también fuera del horario de clase utilizando el entorno virtual de enseñanza y aprendizaje (EVEA) 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 tutorías virtuales de forma sincrónica usando Webex y/o otras herramientas de videoconferencia de uso en común..

La dinámica de las clases comprenderá 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. Estas acciones serán tanto sincrónicas como asincrónicas.

En las clases se conformarán equipos de 10 alumnos integrados en grupos de 2 o 3 alumnos para el desarrollo conjunto de actividades. Cada equipo estará supervisado por un ayudante.

El material teórico y la ejercitación práctica utilizados en el curso están disponibles en el EVEA (Entorno virtual de enseñanza y aprendizaje), se dispone de videos de explicación teórico-práctica de cada tema en cada módulo y de actividades y materiales de lectura.

Asistencia a los talleres virtuales:

- *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 será el equivalente a estar conectado en Webex durante el desarrollo de las clases, se buscarán soluciones alternativas en casos de problema de conectividad.*
- *Pueden justificarse ausentes que serán evaluados por la cátedra.*

EVALUACIÓN

Durante la clase, se propone la realización de ejercicios prácticos que los alumnos deben resolver y enviar, por el EVEA, al auxiliar a su cargo. La evaluación de estas actividades forma parte de la evaluación del proceso. 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 revisado con los ayudantes en el aula, un ejercicio completo resuelto por computadora para el módulo a evaluar.
- Asistir los días correspondientes a la resolución del trabajo final (TP final).
- Resolver el TP final que se desarrolla durante el horario del Taller.
- Aprobar el TP final.

La condición de aprobación del módulo puede ser Aprobado, Desaprobado o coloquio. La instancia de coloquio es para defender el TP final que no alcanza para ser aprobado. El resultado del coloquio puede ser aprobado o desaprobado.

Aprobación de la cursada de la asignatura

Se otorgará la cursada de Programación II a aquellos alumnos que hayan cumplido con el 80% de asistencia a cada uno de los módulos y se hayan presentado a rendir y aprobado los tres primeros módulos (Imperativo, Objetos y Assembler).

Si el alumno aprobó solo uno o dos módulos podrá presentarse a 2 (dos) instancias recuperatorias globales. Si el alumno aprobó 0 debe recurrar la asignatura.

Aprobación del final con promoción

Aquellos alumnos que aprobaron los 3 (tres) módulos tendrán que entregar la resolución de un problema para el módulo de Programación Concurrente. Aprobada esta entrega, se promociona la asignatura donde se promedian las notas de los 4 módulos.

Final de la asignatura

El final deberá ser rendido por aquellos alumnos que sólo aprobaron los 3 (tres) módulos mencionados en la aprobación de la cursada (Imperativo, Objetos y Assembler).

ALTERNATIVA VIRTUAL

En caso de que la situación lo amerite se prevé un funcionamiento virtual donde la forma de desarrollarse la cursada es similar a la presencial con las siguientes modificaciones:

En el caso de distanciamiento total (aislamiento) se trabajará con Webex y se intensificará el envío de actividades por entorno virtual

En caso de distanciamiento parcial y cantidad de aforo del aula se organizan dos grupos de trabajo, uno virtual y otro presencial de manera de generar dos burbujas. Cada grupo tendrá una semana clase presencial y la otra virtual, luego se rotan.

En el caso de contar aula híbrida se planifican sesiones donde se transmitirá la clase en forma remota para los que no se encuentran en el aula

En todos los casos:

- *La asistencia será el equivalente a estar conectado en Webex durante el desarrollo de las clases, se buscarán soluciones alternativas en casos de problema de conectividad.*
- Haber revisado con los ayudantes el envío a través de la plataforma virtual de un ejercicio completo resuelto por computadora para el módulo a evaluar.

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
1	14/08	Presentación del Taller.	Anotarse en el EVEA Armado de grupos de trabajo

2	16/08	Recursión. Concepto. Características Ejercitación.	Resolución de ejercicios básicos utilizando recursión.
	21/08	Feriado puente turístico	
3	23/08	Árboles Binarios Ordenados. Concepto. Operaciones. Ejercitación	Implementación de un árbol binario ordenado.
4	28/08	Árboles Binarios Ordenados. Borrado. Ejercitación	Implementación de las operaciones básicas de árboles binarios ordenados en Pascal.
	30/08	Feriado Memoria por la verdad	
5	04/09	Ejercicios combinados de árboles binarios ordenados y listas	Árboles y listas ejercicios combinados
6	06/09	Ejercicios combinados de árboles binarios ordenados y lista. Segunda parte	Ejercicios combinados
	11/09	Asueto académico	
7	13/09	Repaso para el TP final	
8	18/09	Resolución del Trabajo Final 1	Evaluación

Módulo 2 - Programación Orientada a Objetos (JAVA)

Clase	Fecha	Contenidos	Actividades
Clase 1 (Objetos)	20/09	Conceptos básicos del lenguaje Java	Implementación de programas simples imperativos en Java para ejercitar la sintaxis.
Clase 2 (Objetos)	25/09	Introducción a la POO utilizando Java. Manejo de arreglos	Ejercitación que comprende instanciación de objetos y envío de mensajes.
Clase 3 (Objetos)	27/09	POO utilizando Java	Ejercitación que comprende programación de nuevas clases, instanciación de objetos de dichas clases, envío de mensajes a dichos objetos.
Clase 4 (Objetos)	02/10	Constructores	Ejercitación que comprende la incorporación de constructores a las clases implementadas con anterioridad.
Clase 5 (Objetos)	04/10	Concepto de herencia y polimorfismo	Ejercitación con herencia.
Clase 6 (Objetos)	09/10	Repaso	Repaso de todos los conceptos de POO
Clase 7 (Objetos)	11/10	Resolución del Trabajo Final 2	Evaluación

Módulo 3 - Programación en bajo nivel (Assembler)

Clase	Fecha	Contenidos	Actividades
	16/10	Feriado Día del Respeto a la Diversidad Cultural	
Clase 1 (Assembler)	18/10	Assembler del MSx88	Presentación del simulador para comprender el funcionamiento interno de la PC.
Clase 2 (Assembler)	23/10	Assembler Modos de direccionamiento y saltos	Resolución de ejercicios que requieren diferentes modos de direccionamiento.
Clase 3 (Assembler)	25/10	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)	30/10	Pasaje de Parámetros	Resolución de problemas que requieren comunicación entre módulos utilizando diferentes implementaciones de pasaje de parámetros
Clase 5 (Assembler)	01/11	Resolución del Trabajo Final 3	Evaluación

Módulo 4 - Programación Concurrente

Clase	Fecha	Contenidos	Actividades
Clase 1 (Concurrencia)	06/11	Conceptos básicos de Concurrencia. Entorno CMRE	Ejercicios que muestran los problemas de concurrencia.
Clase 2 (Concurrencia)	08/101	Memoria distribuida	Trabajo con el entorno CMRE aplicando los conceptos vistos a ejercicios con memoria distribuida.
Clase 3 (Concurrencia)	13/11	Memoria compartida	Trabajo con el entorno CMRE aplicando los conceptos vistos a ejercicios con memoria compartida.
Clase 4 (Concurrencia)	15/11	Memoria distribuida y compartida	Trabajo con el entorno CMRE aplicando los conceptos vistos a ejercicios con memoria distribuida y compartida.
	20/11	Feriado Dia de la Soberanía Nacional	
Clase 5 (Concurrencia)	22/11	Asignación de trabajos de promoción	Evaluación
	27/11	Asueto dia trabajador No docente	
Consulta	29/11	Consulta para recuperatorio global 1	
Recuperatorio Global 1	04/12	Recuperatorio 1	
Consulta	06/12	Consulta para recuperatorio global 2	
Consulta	11/12	Consulta para recuperatorio global 2	
Recuperatorio Global 2	13/12		

Fechas de Evaluaciones

Tema	Fecha
Imperativo	05/09
Objetos	03/10
Assembler	24/10
Concurrencia	09/11
Recuperatorio global 1	16/11
Recuperatorio global 2	04/07

Contacto de la cátedra:

Profesor: Alejandro Héctor Gonzalez

Mail: agonzalez@lidi.info.unlp.edu.ar

Sitio Web: <https://ic.info.unlp.edu.ar/alcances-del-titulo-y-competencias-profesionales-de-ingenieria-en-computacion/>

Plataforma virtual: EVEA (Moodle) de la Facultad de Informática Curso 2022-Redictado de Programación 2

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

Firma del/los profesor/es



Prof Asociado Alejandro Héctor Gonzalez

PROGRAMACIÓN II
(Redictado)

Año 2023

Carrera:*Ingeniería en Computación***Año:** *primero***Régimen de Cursada:** *Semestral (primer semestre)***Carácter (Obligatoria/Optativa):** *obligatoria***Correlativas:** *Programación I***Profesores:** *Alejandro Héctor Gonzalez***Hs. semanales de teoría:** *5 hs***Hs. semanales de práctica:** *3 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 (Árboles).
- 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 MÍNIMOS (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 combinan 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 los 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: PROGRAMACIÓN ORIENTADA A OBJETOS CON JAVA USANDO B (Spanish Edition) (Español) 1st Edición
Autores: David Barnes (Author), Michael Kolling (Author), S.L. Gestión editorial Avanzada (Translator)
Editorial: Pearson
Año de edición : 2017

Título: Lógica de programación: Solucionario en pseudocódigo–Ejercicios resueltos.
Autores: Buriticá, O. I. T. (2021).
Editorial: Ediciones de la U.

Año: 2021

Título: Programación
Autores: Alfonso Jiménez Marín; Francisco Manuel Pérez Montes
Editorial: Ediciones Paraninfo, S.A. ISBN: 9788428342865
Año de edición: 2021

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

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

Título: Aprende a programar con Java :un enfoque práctico partiendo de cero
Autores: Jiménez Marín, A., y Pérez Montes, F..
Editorial: Paraninfo
Año de edición: 2a ed., rev., 2016.

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

BIBLIOGRAFIA COMPLEMENTARIA

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

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.

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

Introduction to algorithms.
Comen, Leiserson. MIT Press 2001.

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

CONDICIONES DE LOS ESTUDIANTES PARA ACCEDER AL REDICTADO

Pueden cursar el redictado los alumnos que cumplan con:

- ✓ Tengan la condición de DESAPROBADO según SIU-GUARANI en los trabajos prácticos del cuatrimestre anterior (considerando el curso regular y/o la cursada de invierno/verano según corresponda al cuatrimestre en cuestión).
- ✓ Los alumnos que tienen la condición de “AUSENTE” o “ABANDONO” según el Siu Guarani en los trabajos prácticos del cuatrimestre anterior (considerando el curso regular y/o la cursada de invierno/verano según corresponda al cuatrimestre en cuestión).
- ✓ Los alumnos que se le vencieron los trabajos prácticos de la asignatura en febrero del año en corriente.
- ✓ Los alumnos que tienen aprobado el examen final de las correlativas del plan de estudio de la asignatura.

METODOLOGÍA DE ENSEÑANZA

La asignatura Programación II se organiza en 4 módulos: M1-Programación Imperativa, M2-Programación Orientada a Objetos, M3-Programación en Assembler y M4-Programación Concurrente.

Se propone la modalidad de trabajo de aulas-talleres virtuales y desarrollo de actividades en línea. 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 trabaja con la modalidad de Aula-Taller donde cada estudiante utiliza su computadora desde la casa, se conecta a una sesión Webex con el ayudante para desarrollar ejercicios en forma grupal y poder consultar. Cada taller virtual 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 taller virtual y también fuera del horario de clase utilizando el entorno virtual de enseñanza y aprendizaje (EVEA) 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 tutorías virtuales de forma sincrónica usando Webex y/o otras herramientas de videoconferencia de uso en común..

La dinámica de las clases comprenderá 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. Estas acciones serán tanto sincrónicas como asincrónicas.

En las clases se conformarán equipos de 10 alumnos integrados en grupos de 2 o 3 alumnos para el desarrollo conjunto de actividades. Cada equipo estará supervisado por un ayudante.

El material teórico y la ejercitación práctica utilizados en el curso están disponibles en el EVEA (Entorno virtual de enseñanza y aprendizaje), se dispone de videos de explicación teórico-práctica de cada tema en cada módulo y de actividades y materiales de lectura.

Asistencia a los talleres virtuales:

- *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 será el equivalente a estar conectado en Webex durante el desarrollo de las clases, se buscarán soluciones alternativas en casos de problema de conectividad.*
- *Pueden justificarse ausentes que serán evaluados por la cátedra.*

EVALUACIÓN

Durante la clase, se propone la realización de ejercicios prácticos que los alumnos deben resolver y enviar, por el EVEA, al auxiliar a su cargo. La evaluación de estas actividades forma parte de la evaluación del proceso. 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 revisado con los ayudantes en el aula, un ejercicio completo resuelto por computadora para el módulo a evaluar.
- Asistir los días correspondientes a la resolución del trabajo final (TP final).
- Resolver el TP final que se desarrolla durante el horario del Taller.
- Aprobar el TP final.

La condición de aprobación del módulo puede ser Aprobado, Desaprobado o coloquio. La instancia de coloquio es para defender el TP final que no alcanza para ser aprobado. El resultado del coloquio puede ser aprobado o desaprobado.

Aprobación de la cursada de la asignatura

Se otorgará la cursada de Programación II a aquellos alumnos que hayan cumplido con el 80% de asistencia a cada uno de los módulos y se hayan presentado a rendir y aprobado los tres primeros módulos (Imperativo, Objetos y Assembler).

Si el alumno aprobó solo uno o dos módulos podrá presentarse a 2 (dos) instancias recuperatorias globales.

Si el alumno aprobó 0 debe recurrar la asignatura.

Aprobación del final con promoción

Aquellos alumnos que aprobaron los 3 (tres) módulos tendrán que entregar la resolución de un problema para el módulo de Programación Concurrente. Aprobada esta entrega, se promociona la asignatura donde se promedian las notas de los 4 módulos.

Final de la asignatura

El final deberá ser rendido por aquellos alumnos que sólo aprobaron los 3 (tres) módulos mencionados en la aprobación de la cursada (Imperativo, Objetos y Assembler).

ALTERNATIVA VIRTUAL

En caso de que la situación lo amerite se prevé un funcionamiento virtual donde la forma de desarrollarse la cursada es similar a la presencial con las siguientes modificaciones:

En el caso de distanciamiento total (aislamiento) se trabajará con Webex y se intensificará el envío de actividades por entorno virtual

En caso de distanciamiento parcial y cantidad de aforo del aula se organizan dos grupos de trabajo, uno virtual y otro presencial de manera de generar dos burbujas. Cada grupo tendrá una semana clase presencial y la otra virtual, luego se rotan.

En el caso de contar aula híbrida se planifican sesiones donde se transmitirá la clase en forma remota para los que no se encuentran en el aula

En todos los casos:

- *La asistencia será el equivalente a estar conectado en Webex durante el desarrollo de las clases, se buscarán soluciones alternativas en casos de problema de conectividad.*
- Haber revisado con los ayudantes el envío a través de la plataforma virtual de un ejercicio completo resuelto por computadora para el módulo a evaluar.

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
1	09/03	Presentación del Taller.	Anotarse en el EVEA

			Armado de grupos de trabajo
2	13/03	Recursión. Concepto. Características Ejercitación.	Resolución de ejercicios básicos utilizando recursión.
3	16/03	Árboles Binarios Ordenados. Concepto. Operaciones. Ejercitación	Implementación de un árbol binario ordenado.
4	20/03	Árboles Binarios Ordenados. Borrado. Ejercitación	Implementación de las operaciones básicas de árboles binarios ordenados en Pascal.
5	23/03	Ejercicios combinados de árboles binarios ordenados y listas	Árboles y listas ejercicios combinados
6	27/03	Ejercicios combinados de árboles binarios ordenados y lista. Segunda parte	Ejercicios combinados
7	30/03	Repaso para el TP final	
8	03/04	Resolución del Trabajo Final 1	Evaluación

Módulo 2 - Programación Orientada a Objetos (JAVA)

Clase	Fecha	Contenidos	Actividades
	06/04	Feriado: Pascuas	
Clase 1 (Objetos)	10/04	Conceptos básicos del lenguaje Java	Implementación de programas simples imperativos en Java para ejercitar la sintaxis.
Clase 2 (Objetos)	13/04	Introducción a la POO utilizando Java. Manejo de arreglos	Ejercitación que comprende instanciación de objetos y envío de mensajes.
Clase 3 (Objetos)	17/04	POO utilizando Java	Ejercitación que comprende programación de nuevas clases, instanciación de objetos de dichas clases, envío de mensajes a dichos objetos.
Clase 4 (Objetos)	20/04	Constructores	Ejercitación que comprende la incorporación de constructores a las clases implementadas con anterioridad.
Clase 5 (Objetos)	24/04	Concepto de herencia y polimorfismo	Ejercitación con herencia.
Clase 6 (Objetos)	27/05	Repaso	Repaso de todos los conceptos de POO
	01/05	Feriado: Día del Trabajador	
Clase 7 (Objetos)	04/05	Resolución del Trabajo Final 2	Evaluación

Módulo 3 - Programación en bajo nivel (Assembler)

Clase	Fecha	Contenidos	Actividades
Clase 1 (Assembler)	08/05	Assembler del MSX88	Presentación del simulador para comprender el funcionamiento interno de la PC.
Clase 2 (Assembler)	11/05	Assembler Modos de direccionamiento y saltos	Resolución de ejercicios que requieren diferentes modos de direccionamiento.
Clase 3 (Assembler)	15/05	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)	18/05	Pasaje de Parámetros	Resolución de problemas que requieren comunicación entre módulos utilizando diferentes implementaciones de pasaje de parámetros
Clase 5 (Assembler)	22/05	Repaso	Repaso de todos los conceptos de Assembler. Resolución de ejercicios combinados.
	25/05	Feriado: Día de la Revolución de Mayo	
Clase 6 (Assembler)	29/05	Resolución del Trabajo Final 3	Evaluación

Módulo 4 - Programación Concurrente

Clase	Fecha	Contenidos	Actividades
Clase 1 + Clase 2 (Concurrencia)	1/06	Conceptos básicos de Concurrencia. Entorno CMRE + Áreas	Ejercicios que muestran los problemas de concurrencia + Trabajo con el entorno CMRE aplicando los conceptos vistos a ejercicios con áreas.
Clase 3 (Concurrencia)	05/06	Memoria distribuida	Trabajo con el entorno CMRE aplicando los conceptos vistos a ejercicios con memoria distribuida.
Clase 4 (Concurrencia)	06/06	Memoria compartida	Trabajo con el entorno CMRE aplicando los conceptos vistos a ejercicios con memoria compartida.
Clase 5 (Concurrencia)	08/06	Memoria distribuida y compartida	Trabajo con el entorno CMRE aplicando los conceptos vistos a ejercicios con memoria distribuida y compartida.
Clase 6 (Concurrencia)	12/06	Asignación de trabajos de promoción + Consulta para recuperatorio global 1	
Consulta	15/06	Consulta para recuperatorio global 1	
	19/06	Feriado: Puente Turístico	
Recuperatorio Global 1	22/06	Recuperatorio 1	
Entrega TP de Promoción	23/06	TP de promoción	Deben entregar el TP de promoción a través del entorno Asignaturas.
	26/06	NO HAY ACTIVIDADES	
Consulta	29/06	Consulta para recuperatorio global 2	
Consulta	03/07	Consulta para recuperatorio global 2	
Recuperatorio Global 2	06/07	Recuperatorio 2	
Muestra final	13/07		

Fechas de Evaluaciones

Tema	Fecha
Imperativo	03/04
Objetos	04/05
Assembler	29/05
Concurrencia	12/06
Recuperatorio global 1	22/06
Recuperatorio global 2	06/07

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

Contacto de la cátedra:

Profesor: Alejandro Héctor Gonzalez

Mail: agonzalez@lidi.info.unlp.edu.ar

Sitio Web: <https://ic.info.unlp.edu.ar/alcances-del-titulo-y-competencias-profesionales-de-ingenieria-en-computacion/>

Plataforma virtual: EVEA (Moodle) de la Facultad de Informática Curso 2022-Redictado de Programación 2

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

Firma del/los profesor/es

Prof Asociado Alejandro Héctor Gonzalez