PROGRAMACIÓN II
(Redictado)

Carrera:

Ingeniería en Computación

Año 2022

Año: *primero*
Régimen de Cursada: *Semestral (primer semestre)*
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 (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.

BIBLIOGRAFÏA

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: Pretince 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: Programación orientada a objetos con JAVA usando BlueJ
Autores : Barnes, David; Kölling, Michael
Editorial: Prentice Hall
Año de edición : 2013

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

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: Pretince Hall
Año de edición : 2006

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

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.

Introducción a la programación con Pascal.
José Manuel Iñesta Quereda. Universitat Jaume I, 2000

Introduction to algorithms.

Comen, Leiserson. MIT Press 2001.

Fundamentos de Programación.
Joyanes Aguilar L., Fernandez M., Rodríguez L. Mc Graw Hill. 1999.

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

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:

- ✔ 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 Guaraní 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 recursar 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).