

Programación II**Carrera/ Plan:**

Ingeniería en Computación - Plan 2024

Año 2025

Año: Primero**Régimen de Cursada:** Semestral (segundo semestre)**Carácter (Obligatoria/Optativa):** Obligatorio**Correlativas:** Programación I**Profesor/es:** Alejandro Héctor Gonzalez**Hs. semanales:** 8**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.

Análisis e implementación de estructuras de datos no lineales. Tipos de datos y abstracción de datos.

Introducción a la Programación Orientada a Objetos.

EJES TRANSVERSALES

3. Especificación, proyecto y Desarrollo de Software y Sistemas Conjuntos de Hardware y Software haciendo uso de conceptos, métodos y herramientas de gestión de proyectos, ingeniería de software, base de datos, experiencia del usuario, elicitación, análisis, especificación y validación de requerimientos. (1-Bajo)
4. Desarrollo de Redes de Computadoras y de Redes de Computadoras de área amplia, locales, inalámbricas y móviles. (1-Bajo)
06. Proyecto, desarrollo, dirección, control, construcción, operación y mantenimiento de Sistemas de Procesamiento de Señales, Sistemas Embebidos y sus periféricos incluido en software de soporte, Sistemas Computarizados de automatización y control y Sistemas Conjuntos de Hardware y Software. (1-Bajo)
09. Identificación, formulación y resolución de problemas de ingeniería en computación. (1-Bajo)
12. Utilización de técnicas y herramientas de aplicación en la ingeniería en computación. (1-Bajo)
14. Fundamentos para el desempeño en equipos de trabajo. (1-Bajo)
15. Fundamentos para una comunicación efectiva. (1-Bajo)
18. Fundamentos para el aprendizaje continuo. (1-Bajo)

CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)

Recursividad. Introducción a los tipos abstractos de datos.

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

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- 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*

D- 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.*

E- Conceptos generales de programación

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

BIBLIOGRAFIA

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: 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

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:

https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html
<https://repositorio.umsa.bo/bitstream/handle/123456789/4146/teran.pdf?sequence=1&isAllowed=y>
<https://puntocomnoesunlenguaje.blogspot.com/2012/06/java-ejercicios-basicos-resueltos-1.html>
<https://www.discoduroderoer.es/ejercicios-propuestos-y-resueltos-programacion-orientado-a-objetos-java/>

METODOLOGÍA DE ENSEÑANZA

La asignatura Programación II se organiza en 3 módulos: M1-Programación Imperativa, M2-Programación Orientada a Objetos y M3-Programación Concurrente.

Se propone la modalidad de trabajo de aulas-talleres presenciales 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 una computadora provista por la Facultad o propia desarrollando ejercicios en forma grupal y consultando a los ayudantes. Cada módulo 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 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 en el aula taller y a través del EVEA.

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 30 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:

- 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 controlada durante el desarrollo de las clases.
- Pueden justificarse ausentes que serán evaluados por la cátedra.

Para alcanzar los ejes transversales se abordará el trabajo en grupos de 2 alumnos conformados en equipos de desarrollo más grandes supervisados por un ayudante para poder abordar dinámicas de trabajo grupal, responsabilidad y comunicación efectiva. Los problemas a resolver van aumentando en complejidad de forma de poder abordar características de la ingeniería de software abarcando temas básicos como licitación de requerimientos, modelos de desarrollo de software. Se buscará resolver situaciones que involucren el análisis del hardware disponible acorde al software a desarrollar, Se abordarán los temas de conexión , concurrencia y redes de datos.

El aprendizaje se desarrolla de forma continua tanto dentro como fuera del aula con trabajo sincrónico presencial y asincrónico asistido por tecnología digital.

EVALUACIÓN

Durante las clases, se propone la realización de ejercicios prácticos que los alumnos deben consultar y resolver. En determinados temas se solicitará enviar el código del programa resuelto a través del 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 un ejercicio completo resuelto por computadora para el módulo a evaluar.
- Asistir los días correspondientes a la resolución del trabajo final (TP final).
- Resolver y aprobar el TP final que se desarrolla durante el horario del Taller.
- 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. Se considera que el módulo está aprobado con nota 6 o superior.

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 al menos dos de los módulos.

Si el alumno aprobó uno o dos de los módulos podrá presentarse a la instancia recuperatoria global donde rendirá uno de los módulos.

Si el alumno aprobó 0 debe recurrar la asignatura y no tiene instancia recuperatoria.

Aprobación del final con promoción

Aquellos alumnos que aprobaron los 3 (tres) módulos tendrán promoción de la asignatura donde se promedian las notas de los 3 módulos.

Final de la asignatura

El final deberá ser rendido por aquellos alumnos que sólo aprobaron los 2 (dos) módulos.

En la evaluación son tenidos en cuenta los ejes transversales, se tiene en cuenta la comunicación en la forma de justificar y comentar los programas. La comprensión de los problemas, las estrategias de resolución, la adecuada licitación de los requerimientos para cada problema planteado, así como la organización de la arquitectura de computadora necesaria para poder llevar adelante la ejecución de los programas desarrollados.

CRONOGRAMA DE CLASES Y EVALUACIONES

Módulo 1 - Programación Imperativa (Pascal)



Prof Asociado Alejandro Héctor Gonzalez