

***Plan Anual de Actividades Académicas
a completar por el Director de Cátedra***

Departamento: Ciencias Básicas.....
 Asignatura: Programación en Computación.....
 Titular: Ing. Pablo Daniel Longhi.....
 Asociado:
 Adjunto:
 JTP: Ing. José Sartor.....
 Auxiliares:

Planificación de la asignatura

FUNDAMENTACIÓN DE LA MATERIA DENTRO DEL PLAN DE ESTUDIOS

Es de conocimiento general, la influencia de la informática en las distintas áreas del desarrollo humano, en especial en el área ingenieril. Si bien, cada vez más aparecen en el mercado informático, softwares desarrollados para aplicaciones de gran diversidad destinada a asistir a los ingenieros electrónicos, mecánicos, eléctricos, etc. en sus tareas específicas. Estos softwares abarcan un gran porcentaje de los requerimientos de estos profesionales.

Pero, pueden presentarse casos particulares, donde las herramientas disponibles en el mercado no satisfagan los requerimientos para resolver los problemas que se le plantean al profesional, debiendo recurrir al desarrollo de las aplicaciones, es decir los programas necesarios. En caso de que el desarrollo de las aplicaciones se delegue a profesionales informáticos, los conocimientos brindados por la asignatura le permitirán al profesional integrar un grupo de trabajo con el fin de realizar el análisis de los requisitos del sistema, desarrollo de los algoritmos necesarios para poder implementar las aplicaciones, y el desarrollo de estas.

PROPÓSITOS U OBJETIVOS DE LA MATERIA

Que el alumno logre:

- Conocer los conceptos fundamentales para el desarrollo de aplicaciones.
- Reconocer un problema determinado que se le presente, plantear el algoritmo que lo resuelva, además de capacitarlo en el aprendizaje de una herramienta de desarrollo, que le permita implementar los algoritmos que resuelvan el problema.
- Plantear el algoritmo de la manera correcta y más eficiente posible, abstrayéndose de la implementación del mismo, de manera que la resolución del mismo no dependa de la herramienta a utilizar.
- Obtener conocimientos básicos de una de las herramientas actuales, existentes en el mercado (C Sharp), para que pueda implementar esos algoritmos.

CONTENIDOS

PROGRAMA ANALITICO

UNIDAD 1: INTRODUCCION. HARDWARE Y SOFTWARE

Nociones sobre hardware y software. Software de base y software de aplicación. Ventajas del procesamiento informático. Programas empaquetados y desarrollo a medida, ventajas y desventajas, costos. Tipos de sistemas operativos. Redes de computadoras, ventajas, hardware y software necesario. Uso de la plataforma Moodle.

UNIDAD 2: ALGORITMOS

Definiciones. Primitivas (tipos de datos, operadores y operandos, expresiones, identificadores de datos en memoria). Teoría de la resolución de problemas. Ejemplos de resolución de problemas con algoritmos matemáticos y no matemáticos. Fases en la creación de programas. Programas y subprogramas. Características de un buen diseño. Modularidad. Refinamiento.

UNIDAD 3: ENTIDADES PRIMITIVAS PARA EL DESARROLLO DE ALGORITMOS

Tipos de datos. Expresiones. Operadores y operandos. Identificadores. Constantes y variables. Instrucciones básicas de programación: Instrucción de asignación, instrucción de entrada, instrucción de salida. Expresiones aritméticas. Expresiones lógicas.

UNIDAD 4: TECNICAS PARA LA FORMULACION DE ALGORITMOS

Diagramación Lógica. Diagramas de Flujo: Simbologías. Variantes en la simbología a utilizar en la cátedra. Algoritmos en Pseudocódigo. Reglas sintácticas y semánticas. Primitivas. Ventajas y desventajas de cada una. Diagramas estructurados (Nassi-Schneiderman).

UNIDAD 5: ESTRUCTURAS SECUENCIALES

Estructuras de secuencia. Uso de asignaciones, entradas y salidas de datos. Resolución de problemas usando estructuras secuenciales para la formulación de sus algoritmos.

UNIDAD 6: ESTRUCTURAS CONDICIONALES

Estructuras de selección o condicionales, estructuras de selección múltiple. En caso de, ejemplos de problemas usando estructuras condicionales, simples, dobles, en cascada. Resolución de problemas usando condicionales para la formulación de sus algoritmos.

UNIDAD 7: ESTRUCTURAS REPETITIVAS

Estructuras repetitivas: HACER-PARA, HACER-MIENTRAS, REPETIR-HASTA. Ejemplos de cada tipo de estructuras. Resolución de problemas usando todos los tipos de estructuras para la formulación de sus algoritmos.

UNIDAD 8: VECTORES

Vectores. Utilización del tipo de dato vector. Ejemplos. Índice de un vector. Recorridos ascendentes y descendentes. Mayor y menor de un vector. Promedio de un vector. Porcentajes en un vector, etc. Algoritmos con más de un vector. Algoritmos utilizando vectores y datos simples.

UNIDAD 9: MATRICES

Matrices. Índices. Recorridos por filas y columnas. Búsquedas de mayores y menores por filas, columnas y de la matriz, etc. Casos de utilización de matrices. Ejemplos. Ejercicios combinados de vectores y matrices. Resolución de ejercicios de exámenes.

UNIDAD TEMATICA 10: APLICACIONES ESPECIFICAS.

Introducción al cálculo numérico. Graficación. Algoritmos de métodos numéricos elementales.

METODOLOGÍA DE ENSEÑANZA

DESARROLLO DE LAS CLASES

La actividad curricular se divide en dos clases semanales una de carácter teórico-práctica y otra de carácter parte teórica:

DESARROLLO DE CLASES TEORICAS – PRÁCTICAS: TEORIA DE ALGORITMOS Y PROGRAMACION CON C SHARP

El contenido teórico a desarrollar en estas clases será:

- Técnicas para el desarrollo de algoritmos.
- Enseñanza de conocimientos específicos de C Sharp para la implementación de los problemas resueltos y desarrollados en las prácticas.

Estas clases estarán destinadas a la enseñanza y aprendizaje de las técnicas de desarrollo de algoritmos (diagramas de flujo y pseudocódigos) de manera de que el alumno pueda, en base a los problemas planteados, buscar la mejor solución algorítmica que lo resuelva. En estas clases, además de los contenidos teóricos, el dictado se complementará con la resolución de algunos problemas, de manera que el alumno, adquiera progresivamente la capacidad de entenderlos y plantar una solución (a través de un algoritmo), lo más óptima posible para los mismos.

Otro aspecto meramente práctico a desarrollar será la de capacitar al alumno en el manejo de la herramienta de desarrollo de aplicaciones, C Sharp para que el alumno adquiera los conocimientos necesarios para poder implementar los algoritmos planteados en clase, tanto aquellos resueltos en el aprendizaje de los conceptos teóricos, como los que se resolverán en las horas de práctica.

DESARROLLO DE CLASES PRACTICAS: DESARROLLO DE ALGORITMOS ASISTIDOS CON RAPTOR

Las clases prácticas se irán desarrollando concurrente al desarrollo de las clases teórico-prácticas de algoritmia y C Sharp con la finalidad de que el alumno pueda afianzar los conocimientos adquiridos en el aprendizaje teórico práctico.

Durante estas clases, las cuales se destinarán a la resolución de las guías de ejercicios de las unidades temáticas del programa analítico, se complementarán con el software RAPTOR, software que permite la realización de diagramas de flujo y que ayudan al usuario a comprender la lógica computacional y de esta manera resolver de manera sencilla problemas que se puedan implementar algorítmicamente.

Además Raptor cuenta con una modalidad que nos permite convertir nuestro diagrama de flujo a código (C Sharp, C++, Ada, Java, entre otros.). Esto permitirá, a los alumnos, relacionar la simbología de los diagramas de flujo con las sentencias específicas de C Sharp, y de esta manera disponer de un complemento para el aprendizaje en la implementación de los algoritmos en C Sharp.

CLASES DE CONSULTA

El alumno podrá solicitar clases de consultas ya sea referidas a la parte teórica-práctica (Algoritmia) como a la parte práctica (Resolución de las guías de trabajos prácticos), en

horarios a convenir con los docentes. Las consultas pueden estar referidas a cualquier tema de la materia, a temas de los trabajos prácticos, resolución de problemas de exámenes, etc.

METODOLOGÍA DE EVALUACIÓN.

TRABAJOS PRÁCTICOS

Durante el dictado de la asignatura los alumnos deberán desarrollar trabajos prácticos que serán considerados como evaluaciones parciales del desempeño del alumno a lo largo del transcurso del ciclo lectivo.

Trabajos Prácticos: Los estudiantes deberán llevar una Carpeta de Formación Práctica con el 100% de los Trabajos y con el 80% Aprobados o con sus instancias de Re-elaboración.

Para el desarrollo de los mismos, se deberán cumplimentar las siguientes etapas en cada uno de ellos:

- a) Análisis del problema. (Detectar Datos de Entrada, Salida, Auxiliares y Proceso).
- b) Desarrollo del Diagrama de Flujo, Pseudocódigo y codificación final del Algoritmo.
- c) Depuración del Programa.

- TP 1 - INTRODUCCIÓN: Bajar Planificación de la Materia desde la Plataforma Moodle.
- TP 2 - ALGORITMOS: Resolución de Problemas con Algoritmos Matemáticos.
- TP 3 – PRIMITIVAS: Resolución de Problemas con Datos, Operadores e Instrucciones Básicas de Programación.
- TP 4 - FORMULACIÓN DE ALGORITMOS: Diagramas de Flujo, Pseudo-Código y Diagramas Estructurados Nassi-Schneiderman.
- TP 5 – ESTRUCTURAS SECUENCIALES: Resolución de Problemas usando Estructuras Secuenciales.
- TP 6 – ESTRUCTURAS CONDICIONALES: Resolución de Problemas usando Estructuras condicionales
- TP 7 – ESTRUCTURAS REPETITIVAS: Resolución de Problemas usando Estructuras repetitivas.
- TP 8 – VECTORES: Resolución de Problemas utilizando el tipo de Dato Vector.
- TP 9 – MATRICES: Resolución de Problemas utilizando el tipo de Dato Matriz. Resolución de Ejercicios de Exámenes.
- TP 10 – APLICACIONES ESPECÍFICAS: Algoritmos de Métodos Numéricos.

Los Trabajos Prácticos se irán desarrollando en forma concurrente al desarrollo de las clases Teórico-Prácticas, con la finalidad de que el alumno pueda afianzar los conocimientos adquiridos en el aprendizaje teórico y pueda implementar las soluciones algorítmicas de los problemas resueltos.

METODOLOGÍA DE EVALUACIÓN

Se propone la evaluación continua en el proceso de enseñanza-aprendizaje con el objetivo de que se pueda valorar y mejorar durante el cursado, el proceso de aprendizaje del alumnado y obtener de este proceso, la información necesaria para la correcta transferencia y asimilación de los conocimientos

Su objetivo es perfeccionar el propio proceso de formación tanto para el docente como para el alumno.

La metodología aplicada para complementar la evaluación continua será:

- **Trabajos prácticos:** para cada unidad temática del programa analítico de la materia, se evaluará al alumno con la presentación de un trabajo práctico.
- **Evaluaciones parciales individuales:** se realizarán 3 evaluaciones parciales cuyos contenidos a evaluar serán los siguientes:

Parcial N° 1: algoritmos con estructuras condicionales con uso de RAPTOR.
Parcial N° 2: algoritmos con estructuras repetitivas con uso de RAPTOR.
Parcial N° 3: resolución de problemas con estructuras estructurados (vectores y matrices y codificación con C Sharp)

La evaluación continua permitirá al alumno demostrar sus conocimientos y certificar sus saberes de manera parcial, específicamente en lo referido a la utilización de las planillas de cálculos para la resolución de problemas generales y específicos de la carrera.

Régimen de cursado

- 1) Asistencia a por lo menos el 75% las clases dictadas (25 % de inasistencias).
- 2) Excepciones: el alumno podrá tener hasta el 40% de inasistencias en carácter de opcional siendo la Secretaria Académica de la FRRQ quien dé lugar a esta solicitud, como lo estipula el punto 7.1.1.2 de la Ordenanza 1549.

Régimen de aprobación

Se logrará ésta condición, bajo un régimen de evaluación continua a través de exámenes parciales y aprobación de trabajo prácticos. El alumno dispondrá de una instancia de recuperatorio para todas las evaluaciones propuestas (3).

Aprobación Directa: SIN examen final.

Se considerara esta condición en el caso de que el alumno obtenga en cada evaluación parcial un 70 % de promedio o más de los 3 parciales y aprobación de la carpeta de trabajos prácticos.

Aprobación No-Directa, Con Examen Final

Se considerara esta condición en el caso de que el alumno obtenga en cada evaluación parcial un porcentaje entre el 40 y el 69 % de promedio en las evaluaciones parciales junto con la carpeta de trabajos prácticos aprobada y las condiciones de asistencia antes detallada.

Alumno con No Aprobación

En caso de que el alumno no logre uno o más de los requisitos mínimos para el caso de aprobación no directa deberá recurrir la materia. Estos casos serian:

- No alcanzar el porcentaje de asistencia estipulado.
- No haber llegado el mínimo establecido para una o más de las 3 evaluaciones parciales pasada la instancia de recuperatorio.
- No haber cumplimentado con la presentación de carpeta de trabajos prácticos.

RECURSOS DIDÁCTICOS A UTILIZAR COMO APOYO A LA ENSEÑANZA.

Las clases prácticas de Laboratorio se dictan en una sala de informática con recursos informáticos diversos, los alumnos disponen de PCs. Con:

- Pizarrón y fibrones para las clases teórico prácticas de desarrollo de algoritmos, junto con las guías de ejercicios.
- Software de desarrollo C Sharp Express o Visual Studio 2010 o superior.
- Conexión a Internet con el objetivo de:

- Disponer en clase del material de cátedra que se encuentra en la plataforma educativa (Moodle) de la facultad.
- Consulta de información en general.
- Proyector para el dictado de clases.

ARTICULACIÓN HORIZONTAL Y VERTICAL CON OTRAS MATERIAS

La articulación vertical se da con la cátedra “Matemática para Ingeniería Electromecánica” y con los talleres del Software “Mathematica”. Además con las cátedras de Análisis Matemático y Álgebra, ya que algunos de los métodos numéricos enseñados en estas cátedras son aquellos que se pueden implementar algorítmicamente y que corresponden a la unidad N° 10 de esta cátedra si bien algunos métodos se dan en el 3º año de la carrera.

CRONOGRAMA ESTIMADO DE CLASES

UNIDAD TEMÁTICA	SEMANAS	CARGA HORARIA
UNIDAD 1	1	3 horas
UNIDAD 2	2	3 horas
UNIDAD 3	3	6 horas
UNIDAD 4	4	6 horas
UNIDAD 5	5,9	15 horas
UNIDAD 6	10,13	18 horas
UNIDAD 7	14,20	12 horas
UNIDAD 8	21,22,23,24,25	15 horas
UNIDAD 9	26,27,28,29,30	15 horas
UNIDAD 10	31,32	6 horas

BIBLIOGRAFÍA

A la bibliografía necesaria la podemos dividir en dos grupos: la relacionada con la parte teórica, específicamente a algoritmos y la parte relacionada con la herramienta a utilizar para el desarrollo de aplicaciones, C Sharp y, VB .NET y VB 6.0 (para alumnos de años anteriores).

Respecto a la resolución de algoritmos, se dispone bibliografía actualizada y tutoriales bajados de Internet los cuales aportan una gran variedad de ejercicios y casos interesantes.

Respecto a la bibliografía relacionada con la parte de C Sharp, se utilizarán recursos disponibles en Internet como ser: apuntes, monografías, tutoriales, etc. Actualmente la biblioteca de la Facultad no dispone bibliografía específica sobre C Sharp debido a que no se consigue bibliografía en el mercado.

Cabe aclarar que el alumno podrá disponer del siguiente material, ya sea bajándolo desde la página disponible en la plataforma Moodle de esta facultad. Este material consta de:

- Apuntes de algoritmos utilizados en el dictado de clases.
- Tutoriales de algoritmos y de Visual Basic 6.0 y Visual Basic .NET.
- Ejercicios de la guía de ejercicios codificados en Visual Basic 6.0.
- Guías de ejercicios de las clases teórico prácticas (algoritmos).
- Trabajos prácticos realizados en años anteriores.
- Gran número de modelos de exámenes.

Material disponible en la biblioteca de la universidad

- **Burden Richard L. Faires J. Douglas** - Análisis Numérico - CENGAGE LEARNING (2 ejemplares)
- **Cairó Battistutti, Osvaldo** - Metodología de la programación: algoritmos, diagramas de flujo y programas - México : Alfaomega - 2005 – (2 ejemplares)
- **Deitel, Harvey M.; Deitel, Paul J** -PROGRAMAR EN C / C++ y Java - Pearson Educación 2004 - (1 ejemplar)
- **Dale, Nell; Weems, Chip** Programación y resolución de problemas con C++ McGraw-Hill 2007 - (1 ejemplar)
- **Halvorson, Michael** - Aprenda Visual Basic 6.0 ya - McGraw-Hill – 1998 – (2 ejemplares)
- **Joyanes, Aguilar Luis; Rodriguez Baena Luis** - Fundamentos de programación: algoritmos, estructuras de datos y objetos- McGraw-Hill - 2003 - (2 ejemplares)
- **Mackenzie, Duncan; Sharkey, Kent** -Aprendiendo Visual Basic .NET en 21 días - Pearson Educación – 2003 -(2 ejemplares)
- **Marti Oliet Narciso** - Especificación Derivación y Análisis de Algoritmos. Ejercicios Resueltos - Prentice-Hall – 2006 - (2 ejemplares)
- **Francisco Charte** - VISUAL BASIC 2010 (GUIA PRACTICA)- ANAYA MULTIMEDIA, 2010 (1 ejemplar)
- **Halvorson, Michael** - Aprenda Visual Basic 6.0 ya - McGraw-Hill – 1998 – (2 ejemplares)
- **Mackenzie, Duncan; Sharkey, Kent** -Aprendiendo Visual Basic .NET en 21 días - Pearson Educación – 2003 -(2 ejemplares)
- **Perry, Greg** - Aprendiendo Visual Basic 6 en 21 días - Prentice-Hall – 1999 - (1 ejemplar)
- **Reselman; Peasley; Pruchnia** Descubre Visual Basic 6 - Prentice-Hall- 1999 - (1 ejemplar)