

Programa de Asignatura

00 Código:

F	C	Asign.
T	109	10
T	108	25

01. Facultad: Tecnología Informática / Carrera: Ingeniería en Sistemas Informáticos

02. Asignatura: *T109 – 10 - PROGRAMACIÓN I.*
T108 – 25 - PROGRAMACIÓN I.

03. Año lectivo: 2020

04. Año de Cursada: 1º

05. Cuatrimestre: 2º

06. Hs. Semanales: 6

07.

Profesor Titular: Brizuela, Rafael	
Buenos Aires	Rosario
Titular a cargo: Brizuela, Rafael	Asociado a cargo: Hernández, Alejandro
Adjuntos: Martinez Jorge Viñuales Carlos Roda Jorge Weingand Gastón Battaglia Nicolás Martinez Pedro Cardacci Dario	Adjuntos: Bracamonte, Lucas Auxiliar: Figueroa, Telma
Eje socio profesional: Los Modelos como representación abstracta de la realidad (Análisis y Lenguajes)	
Coordinador de eje: Ing. Vilaboa, Pablo	Coordinador de eje: Ing. Vaquero, Marcelo
Eje Epistémico: Teoría de Sistemas y Lenguajes de Programación.	

Fundamentación

En esta asignatura se tomará conocimiento de cómo almacenar datos en estructuras dinámicas y mediante el manejo de estructuras aprendido en programación estructurada se logrará afianzar los conocimientos previos al estudio de base de datos en Programación orientada a objetos. Así mismo en esta asignatura tendrán una visión global de los distintos paradigmas de programación.

08 Ítems del perfil que se desarrollarán:

- Desarrollar habilidades, conocimientos y destrezas en la aplicación de técnicas de manejo de archivos mediante la programación estructurada, apoyado en lenguaje C.
- Desarrollar habilidades, conocimientos y destrezas en la aplicación de técnicas en la programación orientada a objetos, apoyada en un entorno de programación.
- Desarrollar destrezas en el desarrollo de ejercitación en grupos, la interrelación personal, aplicando enfoques múltiples a un mismo problema.
- Capacitarse para la implementación de estas soluciones dentro de la problemática del hombre y la sociedad, comprendiendo los avances tecnológicos.

09. Correlativas

Previas: Programación Estructurada.

Posteriores: Programación Orientada a Objetos

10. Articulación con otras asignaturas:

A Cálculo Infinitesimal I le proveerá las bases de programación para desarrollar la representación informatizada de límites, derivadas y funciones de mediante el paradigma orientado a objetos.

A Sistemas de Computación II de proveerá los conocimientos de las estructuras dinámicas, bases para la programación en Assembler.

Articula con Historia de la Ciencia y la Técnica para el desarrollo Tramo Inicial de Orientación vocacional de los alumnos de 1er año.

De Programación Estructurada, Sistemas de Computación I, Laboratorio de Cálculo recibe los lineamientos básicos para su dictado desde el punto de vista de programación, uso de los recursos de computación y cálculos matemáticos.

A Programación Orientada a Objetos, Lenguaje de Última Generación les brinda las bases del paradigma.

11. Objetivos:

- Incorporar los conocimientos para identificar las diferencias entre las metodologías de programación estructurada, la orientada a objetos y orientadas a eventos.
- Dominar el manejo de punteros y archivos desde ambas metodologías.
- Incorporar los elementos brindados por la interfaz de los lenguajes orientados a eventos con el objeto de poder aplicarlos en la construcción de software.
- Dominar los aspectos lógicos y algorítmicos de la programación orientada a eventos con el objeto de poder aplicarlos en la construcción de software.
- Desarrollar la idea fundamental de objeto, las propiedades que la definen y los

- eventos que lo controlan.
- Comprender las técnicas de acceso a archivos, su administración y las ventajas y las desventajas que cada una representa.

12. Unidades de desarrollo de los contenidos:

Unidad de repaso y conocimientos necesarios:

- a) Evaluación Diagnóstica.
- b) Repaso:
 - Vectores y matrices.
 - Funciones y procedimientos.
- c) Conocimientos necesarios:
 - Estructuras de control.
 - Estructuras de repetición.
 - Estructuras de decisión.
 - Diagramación lógica.

Tiempo: 6 Hs.

Unidad 1:

Introducción a la programación visual
--

CTS Sistemas común de tipos. Tipos de valor y por referencia, Boxing y Unboxing.
El tipo String. El tipo Date. Tipos Numéricos. Conversión de tipos. Generación de números aleatorios.
Introducción a los Formularios. Formularios MDI. Menues.
Controles Windows Forms. Textbox. Label. Linklabel. Checkbox. Radiobutton. Listbox. Checkedlistbox. Combobox. DateTimePicker. ListView. Monthcalendar. NotifyIcon. NumericUpDown. PictureBox. ProgressBar. RadioButton. RichtextBox. ToolTip. TreeView. WebBrowser. GrupBox. MenuStrip. DataGridView. ColorDialog. FontDialog. FolderBrowseDialog. OpenFileDialog. SaveFileDialog. PrintDialog. Impresión.

Tiempo: 18 hs

Unidad 2:

Estructuras dinámicas I

Recursividad.

Listas. Fundamentos y clasificación,

Listas simplemente enlazadas. Listas doblemente enlazadas.

Listas simplemente enlazadas circulares. Listas doblemente enlazadas circulares.

Operaciones con listas.

Pilas. Conceptos. Operaciones con pilas.

Colas. Conceptos. Operaciones con colas.

Tiempo: 18 hs

Unidad 3

Estructuras dinámicas II

Árboles, representación, arboles binarios, su estructura, operaciones con árboles binarios, recorrido de un árbol binario de búsqueda, operaciones en arboles binarios de búsqueda, balanceo de un árbol.

Arboles AVL, manejo de un AVL, representación de un TDA en árbol AVL.

Arboles B. Operaciones. Ventajas y desventajas.

Tiempo: 18 Hs.

Unidad 4:

Archivos

Introducción al manejo de archivos. Apertura de archivos. Lectura de archivos. Escritura de Archivos. Cierre de archivos. Uso de StreamReader y StreamWriter.

Introducción al sistema de objetos de archivos. Modelo FSO. Creación y administración del objeto FSO. Trabajar con unidades, archivos y carpetas.

Creación, lectura, escritura, inserción, eliminación y búsqueda con archivos binarios, aleatorios y secuenciales.

Concepto de índice.

Tiempo: 6 hs

Unidad 5:

Corte de control y apareo de archivos

Corte de control. Corte de control de 1 nivel. Corte de control de 'n' niveles. Diagramación de las distintas técnicas y métodos de corte de control a un archivo.

Apareo de Archivos. Diagramación de las distintas técnicas y métodos para el apareamiento y manejo de archivos.

Diagramación de ejercicios combinados de apareo y corte de control.

Tiempo: 24 hs

Unidad 6:

Estilos de Programación

Paradigmas de programación. Características principales, ventajas y desventajas de los paradigmas.

Paradigma lógico. Paradigma imperativo. Paradigma declarativo. Paradigma funcional. Paradigma estructurado. Paradigma orientado a objetos. Definición de objetos, clases, instancias.

Inteligencia artificial. Características principales y aspectos distintivos.

Análisis comparativo y combinación entre los distintos estilos.

Tiempo: 6 Hs.

13. Metodología de trabajo:

Las clases se dividirán en teoría y práctica. La ejercitación reflejará los conceptos transmitidos en la clase teórica, más los conocimientos de las clases anteriores y serán aplicados en la resolución de situaciones problemáticas reales de acuerdo al nivel alcanzado en el momento por el grupo de alumnos, lo que permitirá la articulación de los conocimientos teóricos y prácticos.

Con el fin de verificar la construcción y la adquisición de conocimientos se promoverá la participación activa de los alumnos esperando puedan formular soluciones a los problemas propuestos.

Se tendrá especial atención en la participación activa de los alumnos, para poder rescatar la elaboración conceptual que construyan de los temas tratados.

Teoría: 50% del tiempo de la clase. Exposición docente. Diálogo con los alumnos. Puesta en común. Conclusiones grupales. Interconsultas. Revisión de temas. Relación de temas

nuevos con conocimientos adquiridos de la asignatura y de otras asignaturas relacionadas.
Práctica: 50% del tiempo de la clase. La ejercitación deberá reflejar los conceptos transmitidos en la clase teórica, más los conocimientos de las clases anteriores. Se propondrán ejercicios cuya resolución será grupal en algunos casos e individual en otros. Los alumnos prepararán trabajos de investigación que incluirán la exposición oral de los mismos.

Se fomentará la participación de los alumnos en la formulación de soluciones a los problemas propuestos.

Se desarrollarán ejercicios que denoten un alto grado de relación grupal para aprovechar la sinergia de la relación interpersonal

Herramientas:

- Diagramación Lógica.
- Entorno de programación estructurada
- Entorno de programación lógica
- Entorno de programación en eventos
- Entorno de programación en objetos

14 Trabajos prácticos:

- Un trabajo domiciliario de investigación
- Realización de los problemas presentados en clase.
- Entrega de la carpeta de ejercicios prácticos de la materia.
- Todos los trabajos prácticos se acompañarán con el software ejecutable.

15. Bibliografía:

Obligatoria:

Harvey Deitel-Paul Deitel, “**C# como programar**”, Mexico, Pearson Prentice Hall, 2007
Brizuela, Rafael , “**Apuntes de programacion I**” , Buenos Aires, UAI , 2016.
Ceballos, Fco Javier, “**Microsoft C# lenguaje y aplicaciones**”, Mexico, Alfaomega RA-MA 2008
Nilsson, Nils J. **Inteligencia artificial: una nueva síntesis**.-- Madrid: McGraw-Hill, c2001

Ampliatoria:

Cardacci, Darío - Booch,Grady -. "**Orientación a objetos- Teoría y Práctica**" .Argentina, Pearson Argentina - UAI Universidad Abierta Interamericana, 2013 -400 páginas

López, Gustavo; Jeder, Ismael; Vega, Augusto. **Análisis y diseño de algoritmos: implementaciones en PC y Pascal.**-- México, DF: Alfaomega, 2009

Quero Catalinas, Enrique. **Programación en lenguajes estructurados.**-- Madrid: Thomson Editores, 2003. xii, 309 páginas

Escolano, Francisco; Cazorla, Miguel Angel; Alfonso Galipienso, María Isabel; y otros. **Inteligencia artificial: modelos, técnicas y áreas de aplicación** -- Madrid: Thomson Editores, c2003. Martin, James; Odell, James J. **Análisis y diseño orientado a objetos** -- México, DF: Prentice Hall Hispanoamericana, c1994. Martin, James; Odell, James J. **Métodos orientados a objetos: conceptos fundamentales** -- México, DF: Prentice Hall Hispanoamericana, 1997

López Herranz, José; Quero Catalinas, Enrique. **Fundamentos de programación.** 2a. ed.-- Madrid: Paraninfo, c2000.

López Román, Leobardo. **Programación estructurada: un enfoque algorítmico.** 2a. ed.-- México, DF: Alfaomega, c2003

Cairó Battistutti, Osvaldo. **Metodología de la programación: algoritmos, diagramas de flujo y programas.** 2a. ed.-- México, DF: Alfaomega, c2003

16. Procedimiento de evaluación y criterio de promoción:

Parciales.

Los exámenes parciales serán como mínimo dos, cada uno con teoría y práctica. Ambos son obligatorios, individuales y escritos. El primero de ellos se efectuará en la octava semana del cuatrimestre y el segundo entre la semana doce y catorce. Los parciales deberán tener un 50% de contenido correcto para estar aprobados. En este último caso la nota deberá quedar establecida entre 4 (cuatro) y 10 (diez) puntos. Si el alumno no se presenta a un examen parcial se calificará con una nota insuficiente.

Recuperatorios.

Los exámenes recuperatorios se desarrollarán a razón de uno por cada parcial con teoría y práctica. Se podrá optar por tomar un recuperatorio integral en aquellos casos que se justifique y el mismo deberá ser equivalente al alcance de los recuperatorios individuales. Recuperarán los parciales que correspondan, aquellos alumnos que obtuvieron una calificación inferior a 4 (cuatro) puntos en alguno de sus exámenes parciales. Los recuperatorios son individuales y por escrito. El docente dispondrá la fecha para realizar esta actividad que podrá ser con posterioridad a cada uno de los parciales o en una fecha común para todos. Las notas de los exámenes recuperatorios no reemplazan las calificaciones insuficientes obtenidas en los exámenes regulares que se están recuperando, las mismas serán tomadas como parte de las calificaciones que intervienen en el promedio.

Aspectos comunes a parciales y recuperatorios.

La confección de los exámenes parciales y recuperatorios deberán contemplar diversas técnicas para poder observar desde distintos ángulos el objeto de evaluación. Se evaluará la

capacidad de razonamiento del alumno y el marco teórico que lo sustenta. Se recomienda ser objetivo y extremar la claridad de las consignas no dejando lugar a duda de lo que se está intentando evaluar.

Trabajos Prácticos.

Se desarrollarán los trabajos prácticos enunciados en el punto 14 (trabajos prácticos). El trabajo práctico en Vb.Net deberá ser defendido por el alumno. El trabajo y la defensa será individual y la evaluación del mismo estará conformada por un promedio de notas que involucren la producción y el desempeño. Para ambos trabajos prácticos los aspectos a tener en cuenta en dicha evaluación son: calidad del trabajo, preparación y clasificación del material, producción del material, calidad de la defensa, aprovechamiento de los medios y administración del tiempo.

Nota conceptual.

La nota conceptual estará sustentada en la participación en clase del alumno, el interés demostrado en la asignatura y el empeño puesto de manifiesto en las actividades de estudio.

Otros aspectos generales de la evaluación.

También se confeccionarán grillas de seguimiento las cuales deberán reflejar la evolución de los alumnos en los aspectos evaluados.

Los cuestionarios de autoevaluación elaborados para el curso poseen como objetivo el desarrollo de un espíritu crítico en el alumno sobre su desempeño en la futura práctica profesional.

MAPA CONCEPTUAL PROGRAMACIÓN I

