



**Analista Programador en Desarrollo
de Aplicaciones**

Plan 2003 -Res: 6164/03

PROGRAMA DE LA MATERIA:

11106 Programación I

Area: Sistemas

Equipo Docente: Cherencio, Guillermo.

Cantidad de horas semanales : 4 hs.

Días de dictado:

L	M	M	J	V	S	Primer Año
	2		2			

Materias Antecedentes

Ninguna

Materias Consecuentes

11213 Programación Orientada a Objetos

11214 Seminario de Programación

11216 Base de Datos

1.0 Objetivos, Expectativas de Logro

Que el alumno sea capaz de:

1. Reconocer del concepto de algoritmo.
2. Dominar diferentes estructuras de datos y control
3. Diseñar Algoritmos
4. Caracterizar el concepto de programa
5. Codificar algoritmos en un lenguaje de programación estructurado.
6. Resolver problemas de programación, adecuando los tipos de datos, aplicando los criterios de programación estructurada y los de refinamiento sucesivo

Contenidos mínimos:

1. Comprensión de problemas. Concepto de Algoritmo. Diagramación de Algoritmos.
2. Flujogramas. Constantes y Variables. Estructuras Elementales. Contadores y Acumuladores.
3. Estructuras de repetición y de control, estructura de selección simple y múltiple. Subalgoritmos.
4. Arreglos uni y bidimensionales. Apareo de elementos de un arreglo.
5. Métodos de ordenamiento y búsqueda.
6. Archivos. Registros. Direccionamiento.
7. Estructuras de Datos de Alto Nivel. Indexación.

Abril 2021

2.0 Programa de Estudio

1.0 Introducción

- 1.1 ¿Qué es un Computador? ¿Para qué sirve?
- 1.2 Modelo conceptual de computador. La máquina de Von Neumann.
- 1.3 Entrada. Proceso. Salida.
- 1.4 Resolución de Problemas. ¿Qué tipo de problemas puedo resolver utilizando un Computador?. Análisis de los problemas. Similitudes con el modelo conceptual de Von Neumann.
- 1.5 Algoritmo. Algoritmo y resolución de problemas.
- 1.6 Programa. Programa computacional.
- 1.7 Lenguajes de Programación. Breve evolución histórica.
- 1.8 Lenguaje de máquina. Código de máquina. Ensambladores.
- 1.9 Intérpretes y Compiladores.
- 1.10 TP I: Introducción a la programación. Ejemplo de compilación y ejecución de programa computacional.

2.0 Lenguajes de programación

- 2.1 Breve repaso de la unidad anterior.
- 2.2 Metodología de la programación.
- 2.3 Paradigma. Paradigma procedural. Lenguajes Procedurales.
- 2.4 Elementos de un lenguaje: Palabras clave, Identificadores, Sintaxis, Operadores, Tipos de Datos.
- 2.5 El lenguaje de programación C. Breve historia. Clasificación.
- 2.6 Estructura básica de un programa C. Identificación de elementos.
- 2.7 Aprender las estructuras de un lenguaje de programación: Secuencia, Selección, Repetición.
- 2.8 TP II: Metodología de la Programación. Uso del cd-rom de la asignatura, instalación del compilador y preparación del entorno de trabajo.

3.0 Estructura de Secuencia

- 3.1 Breve repaso de la unidad anterior.
- 3.2 Bloque de código. Sintaxis.
- 3.3 Constantes y Variables. Declaración, valores por defecto, alcance. Aplicación de los elementos del lenguaje.
- 3.4 Variables: Asignación, Conversión de tipos (casting).
- 3.5 Expresiones y Operadores aritméticos. Orden de evaluación.
- 3.6 Ejercitación. Resolución de problemas de secuencias usando pseudocódigo.
- 3.7 Entrada y Salida STD. Interacción con el usuario.
- 3.8 Ejercitación. Resolución de problemas de secuencias usando pseudocódigo.
- 3.9 TP III: Resolución de problemas de secuencia.

4.0 Estructura de Selección

- 4.1 Breve repaso de la unidad anterior.
- 4.2 Control de flujo de un programa.



- 4.3 Operadores relacionales. Expresiones relacionales simples.
- 4.4 Operaciones lógicas AND, OR, XOR. Tablas de verdad.
- 4.5 Ejercitación. Resolución de problemas de selección usando pseudocódigo. Implementación en C.
- 4.6 Selecciones anidadas.
- 4.7 Ejercitación. Resolución de problemas de selección anidadas usando pseudocódigo. Implementación en C.
- 4.8 Estructura de selección múltiple.
- 4.9 TP IV: Resolución de problemas de selección.
- 5.0 **Estructuras Repetitivas**
 - 5.1 Breve repaso de la unidad anterior.
 - 5.2 Control de Flujo de un programa. Necesidad de repetir instrucciones.
 - 5.3 ¿Qué es una estructura repetitiva? ¿Cuándo utilizarlas?.
 - 5.4 Variable de control: inicialización, actualización, evaluación.
 - 5.5 Estructura mientras. Contadores. Acumuladores.
 - 5.6 Ejercitación. Resolución de problemas de repetición usando pseudocódigo. Implementación en C.
 - 5.7 Estructura repetir-hasta. Ejemplos – pseudocódigo - implementación en C.
 - 5.8 Estructura para. Ejemplos – pseudocódigo - implementación en C.
 - 5.9 ¿Cuándo utilizar una u otra estructura repetitiva?
 - 5.10 Algoritmo del menor y del mayor.
 - 5.11 TP V: Resolución de problemas de iteración.
- 6.0 **Modularidad y Reusabilidad**
 - 6.1 Breve repaso de la unidad anterior.
 - 6.2 Descomposición de problemas.
 - 6.3 Funciones: declaración, implementación, uso. Enfoque Top-Down versus Down-Top.
 - 6.4 Biblioteca. Biblioteca standard. Reutilización de código.
 - 6.5 Funciones y modelo de memoria: pasaje de parámetros por valor y por referencia.
 - 6.6 Punteros: forma de simular pasaje por referencia.
 - 6.7 Ejemplos – pseudocódigo - implementación en C. Implementación modular de programas ya hechos.
 - 6.8 Arreglos. ¿Qué son? ¿Para qué sirven? ¿Cuándo usarlos?.
 - 6.9 Arreglos y estructuras repetitivas.
 - 6.10 Ejercitación. Resolución de problemas de conteo, sumariación, promedios, apareo de arreglos unidimensionales.
 - 6.11 Cadenas de caracteres y modelo de memoria. Asignación dinámica de memoria.
 - 6.12 Cadenas de caracteres y arreglos unidimensionales.
 - 6.13 Cadenas de caracteres y punteros. Pasaje de parámetros a funciones.
 - 6.14 Cadenas de caracteres y entrada – salida standard.
 - 6.15 TP VI: Resolución modular de problemas de uso de Cadenas, Arreglos y Funciones.



7.0 **Búsqueda y Ordenamiento**

- 7.1 Breve repaso de la unidad anterior.
- 7.2 El problema de la búsqueda y el ordenamiento.
- 7.3 Algoritmos de búsqueda y ordenamiento sobre vectores unidimensionales.
- 7.4 Ejercitación. Resolución de problemas de búsqueda y ordenamiento. Implementación en C.
- 7.5 Arreglos bidimensionales. Ejemplos de utilización de matrices.
- 7.6 Arreglos tridimensionales.
- 7.7 Arreglos de n dimensiones.
- 7.8 TP VII: Resolución de problemas utilizando arreglos, búsquedas y ordenamiento.

8.0 **Persistencia**

- 8.1 Breve repaso de la unidad anterior.
- 8.2 Necesidad de persistencia de datos en dispositivo de almacenamiento secundario.
- 8.3 Archivo. Registro. Campo.
- 8.4 Contenido de los archivos: archivos delimitados versus no delimitados; archivos binarios versus archivos de texto. Ordenados. Desordenados.
- 8.5 Métodos de acceso: Archivos secuenciales versus Archivos Relativos.
- 8.6 Estructura de registro. Implementación en C.
- 8.7 Apertura, Proceso y Cierre de archivos. Flujos de datos (streams). Corte de Control.
- 8.8 Ejercitación. Resolución de problemas usando archivos secuenciales. Implementación en C.
- 8.9 Algoritmo de Corte de control. Ejemplo. Implementación en C.
- 8.10 Archivos Relativos. Planificación. Hashing.
- 8.11 Claves Sinónimas. Colisiones. Direccionamiento Abierto y Cerrado.
- 8.12 Ejercitación. Implementación de archivo relativo. Implementación en C.
- 8.13 Combinación de archivos secuenciales y relativos. Trabajo con más de un archivo a la vez.
- 8.14 TP VIII: Trabajo Practico Final Integrador: Proyecto individual en donde se apliquen conceptos vistos en cada una de las unidades didácticas. Aplicación que utilice una biblioteca propia, arreglos, cadenas de caracteres, búsqueda, ordenamiento, persistencia de datos.

9.0 **Estructuras de Alto Nivel**

- 9.1 Breve repaso de la unidad anterior.
- 9.2 Recursividad.
- 9.3 Alternativa al hashing: índices. Archivos Indexados. Búsqueda en Índices.
- 9.4 Listas.
- 9.5 Pilas.
- 9.6 Colas.
- 9.7 Acceso por clave secundaria: Índices e Índices invertidos. Árboles.
- 9.8 Depuración de Programas.

3.0 Bibliografía

Bibliografía Básica

- Braunstein Silvia L, Giogia Alicia B, “Introducción a la programación y a las estructuras de datos”, EUDEBA, Buenos Aires, 2da. Edición, 2001
- Herbert Schildt, “Programación en lenguaje C”, McGraw Hill, Madrid, 1992
- Joyanes Aguilar Luis, Castillo Sanz Andres, Sanchez Garcia Lucas, Zahonero Martinez Ignacio, “C. Algoritmos, programación y Estructura de Datos”, McGraw Hill , 2005
- Kernighan Brian W., Richie Dennis M., “El Lenguaje de Programación C” 2da edición, Prentice Hall, Murray Hill, New Jersey, 1991 ISBN 968-880-205-0
- Letvin Lozano R, “Diagramación y programación”, Mc Graw Hill, 3ra Edición, 2004.
- Stevens Richard W., “Advanced Programming in the UNIX Environment”, Addison-Wesley, 1993 ISBN 0-201-56317-7

Bibliografía Adicional

- Alcalde E. y García M., “Metodología de la Programación”, McGraw Hill, Madrid, 1992.
- Pressman, “Ingeniería de software: un enfoque práctico”, Mc Graw Hill, 1998
- Sethi R., “Lenguajes de Programación”, Addison-Wesley, Wilmington, Delaware, 1989
- Wirth N., “Introducción a la programación sistemática”, El Ateneo, Bs.As., 1986

Apuntes de la Asignatura

- Cherencio, Guillermo, “Trabajando con el Compilador GNU C/C++ GCC”, ISDFyT N° 189, asignatura: “Diagramación y Algoritmos I”, 2007
- Cherencio, Guillermo, “Administración de Memoria y Apuntadores en ANSI C”, ISDFyT N° 189, asignatura: “Diagramación y Algoritmos I”, 2007
- Cherencio, Guillermo, “Cuestiones sintácticas básicas del Lenguaje C”, ISFT N° 189, asignatura: “Programación I”, 2009
- Cherencio, Guillermo, “Trabajando con Librerías Estáticas en el Entorno de Desarrollo Geany”, ISFT N° 189, asignatura: “Programación I”, 2011
- Cherencio, Guillermo, “Traduciendo Diagramas de Flujo a Programas C”, ISFT N° 189, asignatura: “Programación I”, 2016
- Cherencio, Guillermo, “Trabajando con la Librería idxg”, ISFT N° 189, asignatura: “Programación I”, 2018

Abril 2021

4.0 Evaluación

Se considerará alumnos regulares en modalidad presencial a aquellos que:

- Aprueben el 1er y 2do parcial con nota mayor igual a 4.
- Todos los parciales tienen una instancia de recuperación
- Tengan aprobados todos los trabajos prácticos grupales propuestos: TP I, II, III, IV, V, VI, VII
- Tengan aprobado la primera versión del trabajo práctico final individual integrador: TP VIII.
- Cumplan con todos los requisitos de promoción indicados en el Plan Institucional de Evaluación, normativas de la DGyCE, DES, Pcia. Bs. As.

Se considerará alumnos regulares en modalidad no-presencial o híbrida y mientras dure el aislamiento social obligatorio a aquellos que:

- Mantengan el ingreso y participación en clases sincrónica y lectura de material asincrónico acorde con los registros del campus institucional
<http://campus.isft189.edu.ar/moodle>
- Tengan aprobados todas las actividades prácticas propuestas individuales y grupales: TP I, II, III, IV, V, VI, VII, a través de campus institucional
- Cumplan con todos los requisitos de promoción indicados en el Plan Institucional de Evaluación, normativas de la DGyCE, DES, Pcia. Bs. As.

Evaluación Final para Alumnos libres (inscriptos así a comienzo del año) y Regulares:

- Presentación del Trabajo Práctico Final Individual Integrador (aprobado previamente por el docente en la regularización de la asignatura), software aplicativo funcionando, completo y sobre el cual se solicitarán cambios a realizar. El alumno contará con una 1 hora para implementar los cambios.

Criterio de Evaluación:

- En cuanto a los contenidos teóricos se requerirá la comprensión básica de los mismos y su correcta aplicación.
- Los alumnos tendrán a su disposición la sintaxis básica del lenguaje de programación C, no siendo este punto de interés en la evaluación
- Es de interés para la evaluación la lógica de los procesos que deba implementar el alumno
- Es de interés para la evaluación el progreso que el alumno manifieste en cada etapa. La actitud frente a las dificultades, las mejoras introducidas a la aplicación entre la primera versión y la versión final de la misma.
- Es de interés para la evaluación la actitud del alumno frente al trabajo propuesto, las ganas, el interés demostrado en las actividades.
- Es de interés para la evaluación que el alumno adquiera destreza en la utilización de documentación y manuales técnicos en inglés.

Abril 2021



5.0 Cronograma

Semana	Actividad
1	Contrato Pedagógico. 1.1 – 1.6
2	1.7 – 1.10. Entrega TP I
3	2.1 – 2.5
4	2.6 – 2.12 Entrega TP II
5	3.1 – 3.6
6	3.7 – 3.9 Entrega TP III
7	4.1 – 4.5
8	4.6 – 4.9 Entrega TP IV
9	5.1 - 5.6
10	5.7 – 5.9
11	5.10 – 5.11 Entrega TP V
12	Primer Parcial (Unidades I, II, III, IV, V)
13	6.1 – 6.4
15	6.5 – 6.10
16	6.11 – 6.15 Entrega TP VI
17	7.1 – 7.4
18	7.5 – 7.8 Entrega TP VII
20	8.1 – 8.8
21	8.9 – 8.14 Entrega TP VIII (final integrador)
22	Segundo Parcial (Unidades VI, VII, VIII)
23	9.1 – 9.5
24	9.6 – 9.8
25	Recuperatorios, Trabajo Integrador
26	Recuperatorios, Trabajo Integrador, Cierre

Abril 2021

6.0 Observaciones

Inicialmente el alumno comprenderá algunos conceptos básicos para introducirlo hacia el mundo de la programación de computadores. Se destaca la necesidad de utilizar lenguajes de alto nivel para evitar la complejidad propia del computador y la necesidad de utilizar diagramas (en este caso, los diagramas de flujo) para una representación gráfica del algoritmo por implementar. A partir de la unidad 3 se introducen las estructuras básicas de todo lenguaje de programación: secuencia, selección, iteración. En la unidad 6 se introducen las funciones y la forma en que éstas nos ayudan a lograr programas modulares y reutilizables. Creación de bibliotecas propias y utilización de arreglos como una forma de facilitar la programación evitando la repetición de identificadores, combinando con las estructuras de iteración, el alumno contará con dos elementos fundamentales de la programación.

Por último, se indica la necesidad de almacenamiento secundario, puesto que todo lo realizado hasta el momento actúa en términos de la memoria principal del computador, la cual es volátil y no permite guardar el resultado de nuestro proceso. Se requiere de estructuras persistentes en disco. Se indican las limitaciones del acceso a los datos a través de la clave primaria, se desarrollan las estructuras de datos de alto nivel y de cómo éstas pueden resolver el problema para acceder a los datos a través de claves secundarias.

No se trata de un curso de lenguaje C/C++, de hecho sólo se mencionan sus elementos básicos a fin de que éste se utilice como una herramienta para implementar los algoritmos propuestos. Se optó por este lenguaje por ser altamente tipado, permite una programación estructurada, no requiere de ningún proceso de instalación por parte del alumno.

Todo el material de estudio (incluso este documento), más software de desarrollo, tutoriales, trabajos prácticos, programas de demostración, etc. requerido por los alumnos en esta asignatura se encuentran catalogados en <http://www.grch.com.ar/docs/p1>

Abril 2021



7.0 Intervención Didáctica –Recursos

Estrategias Didácticas a utilizar:

Según sean los objetivos de cada clase, las mismas serán:

- Clase expositiva para la explicación de conceptos teóricos.
- Trabajos de producción individual para ejercicios en clase en donde el alumno deba aplicar conceptos teóricos a casos particulares.
- Trabajos de producción grupal cuando el alumno deba debatir con sus pares las distintas propuestas de solución a una situación planteada.

Recursos a utilizar:

- Centro de Cómputos para realizar prácticas en donde el alumno deba aplicar conceptos teóricos a la resolución de situaciones problemáticas.
- Tiza, Pizarrón
- CD-ROM (1) y (2) de la asignatura en donde se encuentra digitalizado el material teórico a utilizar, así como también todo el software GNU requerido por la asignatura.
- Notebook personal

Abril 2021



8.0 Actividades de Extensión

Los alumnos deberán especificar el trabajo final integrador a implementar, aquí se espera que los alumnos puedan vincular esta asignatura con sistemas reales que se implementen en el mercado local. No se pretende que el alumno implemente un sistema completo, pero si se espera que pueda producir una aplicación de tipo consola que integre todos los conceptos vistos en la asignatura y tenga alguna utilidad real.

Abril 2021

9.0 Integración Curricular

Integración Curricular Vertical

Se espera que los contenidos teóricos de las primeras unidades permitan una base de sustento o punto de partida para las asignaturas Sistemas de Información I y Sistemas Operativos, ambas asignaturas de segundo año.

Integración Curricular Horizontal

Los contenidos están relacionados horizontalmente con la asignatura Arquitectura de Computadores. Las unidades 1 y 2 contienen conceptos que deberán ser profundizados en dicha asignatura. Se requiere de una coordinación entre ambas asignaturas para que ambos docentes acuerden el grado de profundidad de estos conceptos, eviten la duplicidad de contenidos y vez de ello, lo reutilicen haciendo explícita referencia a la asignatura que corresponda. Esto genera en el alumno la idea de articulación entre las asignaturas y no la ilusión de ser simples estancos de conocimiento que no tienen conexión alguna.

Abril 2021