



CARRERA	Tecnicatura Superior en Análisis de Sistemas con Orientación en Redes Informáticas				
PLAN DE ESTUDIOS	Resolución 2561/06				
DOCENTE	Gustavo Daniel Reyes				
UNIDAD CURRICULAR	Período	Año	Régimen	Hs. Cátedra	
Programación II (PII)	2023	2do.	Cuatrimestral (1er. cuat.)	5 (cinco)	

1. FUNDAMENTACIÓN

La cátedra de Programación II es un espacio esencial de la carrera Tecnicatura Superior en Análisis de Sistemas con Orientación en Redes Informáticas, puesto que brindará al futuro profesional, el conocimiento y dominio de los diferentes enfoques para la construcción de piezas de software (programas).

Intentar conjugar los distintos aspectos teóricos y prácticos, con el objetivo de que el alumno complete el ciclo del problema hasta obtener su solución con computadora, seleccionando adecuadamente algoritmos y datos, es una tarea que no puede estar sujeta a improvisaciones, y que conlleva una serie de actividades de decisión y planificación complejas y variadas, para las cuales es necesario tener una adecuada respuesta metodológica.

Con tal motivo, en esta asignatura se introducirán los conceptos fundamentales y necesarios para abstraer un problema del mundo real, convertirlo en un modelo computable, escribir una solución simbólica del mismo y traducirla en una solución concreta y ejecutable. La suma de estos conocimientos, permitirá a los futuros profesionales aplicar diferentes enfoques en el desarrollo de programas, en función de las características específicas de los problemas identificados.

En tal sentido, la propuesta que realizo tiene como objetivo principal posibilitar que el alumno disponga de un conjunto de conocimientos teórico-prácticos, que sean aplicables a su futura labor profesional y que le permitan entender las técnicas orientadas a la resolución de problemas utilizando una computadora, incorporando el concepto de modelo como representación del entorno, los datos y los procesos del mundo real, y de la noción de programa como la expresión ordenada, completa y correcta de la especificación de una solución computable de dicho problema.

Dentro de la UC se incorporarán las habilidades de modelado y diseño que se integrarán con las capacidades adquiridas en los trayectos formativos de las cátedras “Programación I” y “Introducción a la Informática”, completando las competencias de los alumnos para el desarrollo de programas.

2. OBJETIVOS

- Identificar las características básicas de los diferentes paradigmas de programación, en general, y de los paradigmas de Programación Imperativa y Estructurada, en particular.
- Aplicar el proceso de abstracción necesario para obtener una solución computable, a partir del análisis de un problema del mundo real.
- Reconocer la importancia de utilizar las metodologías de Programación Estructurada y Modular en el proceso de escritura de programas.
- Dominar los principios sintácticos y semánticos de un lenguaje de programación de alto nivel (C).
- Identificar las distintas etapas del proceso necesario para traducir un programa escrito en lenguaje de alto nivel a uno comprensible por una computadora.
- Seleccionar las estructuras de datos adecuadas para la construcción de programas de computación específicos.
- Comprender la importancia de la selección de las estructuras de datos adecuadas para el diseño de programas.

3. NÚCLEOS PROGRAMÁTICOS

A) PARADIGMAS Y LENGUAJES DE PROGRAMACIÓN

Resolución de problemas con computadoras. ¿Qué es la programación? ¿Qué es un paradigma? Paradigmas de programación. El Teorema del Programa Estructurado. Principales paradigmas de programación. Paradigma Imperativo vs. Declarativo. Programación Modular. Programación No Estructurada. Paradigma Estructurado. Ventajas de la programación estructurada.

¿Qué es un programa? Lenguajes de Programación. Tipos de lenguajes de programación: Lenguaje de máquina, Lenguaje ensamblador, Lenguaje de alto nivel. Traductores del lenguaje: intérpretes y compiladores. El proceso de compilación del lenguaje C. El Entorno de Desarrollo Integrado (IDE). ¿Por qué utilizar un IDE? Errores en el proceso de programación. Errores de compilación. Errores en tiempo de ejecución. Errores lógicos. Recomendaciones para programar.

BIBLIOGRAFÍA ESPECÍFICA NUCLEO A			
AUTOR	TÍTULO	EDITORIAL	LUGAR Y AÑO DE ED.
Alonso Amo, F. y Segovia Pérez, F.	Entornos y Metodologías de Programación.	Paraninfo	España, 1995
Cairo Battistutti, O.	Metodología de la programación. Algoritmos, diagramas de flujo y programas, 3ra. edición.	Alfaomega	México, 2005



INSTITUTO DE EDUCACIÓN SUPERIOR N° 6.012

Pellegrini N° 295 - El Carril (Salta) - Rep. Argentina
FAX - T.E. N° 4-908015 - Email: isfd6012@gmail.com



Joyanes Aguilar, L.	Metodología de la Programación, 2da. Edición	Mc Graw-Hill	España, 2006
Joyanes Aguilar, L. y Zahonero Martínez, I.	Programación en C, C++, Java y UML, 2da. edición.	Mc Graw-Hill	México, 2014
Sethi, R.	Lenguajes de Programación: Conceptos y Constructores.	Addison-Wesley	España, 1992
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2022

B) INTRODUCCIÓN A LA PROGRAMACIÓN

Concepto de programa. Partes de un programa. Elementos básicos de un lenguaje de programación. El lenguaje de programación C. Estructura de un programa en C. La función main(). El archivo de código fuente. Comentarios. Identificadores. Palabras reservadas. Literales. Delimitadores. Identificadores. Tipos de datos en C. Tipos de datos simples predefinidos: ordinales, reales y carácter. Variables. Constantes. Tipos de constantes: literales, definidas y declaradas.

Expresiones y operadores. Operadores de asignación. Operadores aritméticos. Operadores de relación. Operadores lógicos. El operador condicional. Prioridad y asociatividad entre operadores. Conversión de tipos de datos (type casting): implícita y explícita. Entrada y salida estándar en C. Las función printf() y scanf(). Especificadores: de conversión y marcas de formato. Sentencias de control de flujo. Sentencias secuenciales. Sentencias condicionales if-else y switch. Anidamiento de sentencias condicionales. Sentencias repetitivas. Sentencia repetitivas condicionales while y do-while. La sentencia repetitiva incondicional for. Anidamiento de sentencias repetitivas. Funcionamiento un ciclo anidado. Diseño de ciclos. Fin de un ciclo. Estilo de programación. Instalación y configuración del IDE CodeBlock.

BIBLIOGRAFÍA ESPECÍFICA NUCLEO A

AUTOR	TITULO	EDITORIAL	LUGAR Y AÑO DE ED.
Cairo Battistutti, O.	Fundamentos de programación. Piensa en C.	Pearson Educación	México, 2006
Joyanes Aguilar, L.	Metodología de la Programación, 2da. Edición	Mc Graw-Hill	España, 2006
Joyanes Aguilar, L. y Zahonero Martínez, I.	Programación en C, C++, Java y UML, 2da. edición.	Mc Graw-Hill	México, 2014
Kemighan, B. y Ritchie, D.	El Lenguaje de Programación C, 2da. Edición.	Pearson Educación	España, 1989
García-Bermejo, J.	Programación Estructurada en C.	Prentice-Hall	
Márquez, G; Osorio, S. y Olvera, N.	Introducción a la Programación Estructurada en C.	Prentice-Hall	México, 2011
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2022

C) TIPOS DE DATOS ESTRUCTURADOS ESTÁTICOS

Introducción a las estructuras de datos. Clasificación de estructuras de datos. Arrays unidimensionales: concepto y características. Sintaxis. Principales operaciones con arrays unidimensionales: lectura/escritura, asignación, actualización, recorrido, copia, inserción, ordenación y búsqueda. Métodos de ordenación: Selección Directa, de la Burbuja, Inserción Directa. Métodos de búsqueda: Búsqueda secuencial, Búsqueda binaria.

Arrays bidimensionales: concepto y características. Sintaxis. Principales operaciones con arrays bidimensionales: lectura/escritura, asignación, actualización, ordenación y búsqueda. Suma y resta de matrices. Producto de matrices. Producto de un escalar por una matriz. Arrays de más de dos dimensiones.

Cadenas de caracteres: concepto, declaración y asignación. Operaciones con cadenas: manejo de los elementos de la cadena, longitud de una cadena, comparación de cadenas, unión de cadenas. Funciones y procedimientos para tratamiento de cadenas. Conversión de cadenas a números. Cadenas como parámetros de funciones.

BIBLIOGRAFÍA ESPECÍFICA NUCLEO A

AUTOR	TITULO	EDITORIAL	LUGAR Y AÑO DE ED.
Cairo Battistutti, O.	Fundamentos de programación. Piensa en C.	Pearson Educación	México, 2006
Cairo Battistutti, O.	Estructuras de datos, 3ra. edición.	Mc Graw-Hill	México, 2006
Joyanes Aguilar, L.; Fernández Azuela, M.; Sánchez García, L. y Zahonero Martínez, I.	Estructuras de datos en C	Mc Graw-Hill	España, 2005
Joyanes Aguilar, L. y Zahonero Martínez, I.	Programación en C, C++, Java y UML, 2da. edición.	Mc Graw-Hill	México, 2014
Kemighan, B. y Ritchie, D.	El Lenguaje de Programación C, 2da. Edición.	Pearson Educación	España, 1989
García-Bermejo, J.	Programación Estructurada en C.	Pearson/Prentice-Hall	México, 2008
Márquez, G; Osorio, S. y Olvera, N.	Introducción a la Programación Estructurada en C.	Pearson/Prentice-Hall	México, 2011
Wirth, N.	Algoritmos y Estructuras de Datos.	Prentice-Hall	España, 1986
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2022

D) PROGRAMACIÓN MODULAR

El paradigma de Programación Modular: objetivos, propiedades, implementación. Técnica Divide y Vencerás. Subprogramas. Funciones: concepto y estructura. Funciones de biblioteca. Funciones definidas por el programador: prototipos, implementación y llamadas. Parámetros de la función. Parámetros actuales y formales. Paso de parámetros por valor y por referencia. Arreglos como parámetros de funciones. Ámbito de una variable. Variables locales y globales. Regla de uso de identificadores



locales y globales. Caso de llamadas a funciones. Anidamiento de funciones. Paso de funciones como parámetros. Ventajas de la Programación Modular. Recursividad: concepto. Tipos de recursividad. Ventajas y desventajas de la recursividad.

BIBLIOGRAFÍA ESPECÍFICA NUCLEO A			
AUTOR	TÍTULO	EDITORIAL	LUGAR Y AÑO DE ED.
Cairo Battistutti, O.	Fundamentos de programación. Piensa en C.	Pearson Educación	México, 2006
Cairo Battistutti, O.	Estructuras de datos, 3ra. edición.	Mc Graw-Hill	México, 2006
Joyanes Aguilar, L.; Fernández Azuela, M.; Sánchez García, L. y Zahonero Martínez, I.	Estructuras de datos en C	Mc Graw-Hill	España, 2005
Joyanes Aguilar, L. y Zahonero Martínez, I.	Programación en C, C++, Java y UML, 2da. edición.	Mc Graw-Hill	México, 2014
Kernighan, B. y Ritchie, D.	El Lenguaje de Programación C, 2da. Edición.	Pearson Educación	España, 1989
García-Bermejo, J.	Programación Estructurada en C.	Pearson/Prentice-Hall	
Márquez, G; Osorio, S. y Olvera, N.	Introducción a la Programación Estructurada en C.	Pearson/Prentice-Hall	México, 2011
Wirth, N.	Algoritmos y Estructuras de Datos.	Prentice-Hall	España, 1986
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2022

4. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Trabajos grupales e individuales.
- Elaboración de informes escritos.
- Exposiciones grupales e individuales
- Aprendizaje cooperativo.
- Aula invertida.

5. EVALUACIÓN

En las asignaturas que se ocupan de la capacidad de desarrollar programas de computadora presentan algunas particularidades que las hacen especialmente interesantes a la hora de aplicar instrumentos de evaluación no tradicionales. Estas características son: a) el resultado del aprendizaje es estrictamente una competencia; b) el ejercicio profesional de esta competencia se realiza usualmente en equipo y c) el proceso de evaluación puede ser más objetivo que en otras materias.

CRITERIOS

- Participación comprometida en los trabajos propuestos, ya sean individuales y/o grupales.
- Compromiso y responsabilidad en las tareas a desarrollar
- Presentación de trabajos en tiempo y forma
- Manejo adecuado de la bibliografía
- Coherencia, claridad conceptual, integración y pertinencia tanto en oratoria como en el discurso escrito
- Dominio, comprensión y transposición de los contenidos del programa.
- Capacidad de resolución de problemáticas presentadas.
- Capacidad de organizar, coordinar y elaborar la tarea encomendada
- Aplicación de conocimientos y dominio de técnicas, mediante la respuesta a preguntas y la resolución de problema.
- Participación pertinente en el desarrollo de las clases.

INSTRUMENTOS DE EVALUACIÓN

Para valorar los aprendizajes de los alumnos y sistematizar dichas valoraciones, se utilizarán los siguientes instrumentos:

- Evaluaciones escritas
- Evaluaciones orales
- Trabajos prácticos
- Coloquios

PARA OBTENER LA REGULARIDAD

Para regularizar la asignatura, el alumno deberá:

- Tener el 80% de asistencia a las clases.
- Presentar a término y aprobar el 90% de los trabajos prácticos propuestos.
- Aprobar los exámenes parciales propuestos o sus respectivos exámenes recuperatorios. La nota mínima obtenida no deber ser menor de 5 (cinco), equivalente al 60% del puntaje puesto en juego, en una escala de 1 (uno) a diez (10).
- Obtener una calificación final mínima de 5 (cinco) puntos, en una escala de 1(uno) a 10(diez), al terminar el cursado.

Las condiciones de regularización descriptas serán expuestas a los alumnos durante la presentación de la Unidad Curricular y del plan de trabajo de la misma.



PARA OBTENER LA PROMOCIONALIDAD

Según lo establece la normativa vigente (RAM), la Unidad Curricular NO ES promocionable.

PARA RENDIR LA UNIDAD CURRICULAR EN CALIDAD DE ALUMNO LIBRE

En base a la normativa vigente, las condiciones establecidas para rendir esta Unidad Curricular en calidad de alumno libre son:

- Presentación y aprobación de la totalidad de los trabajos prácticos o de un trabajo práctico integrador equivalente, en ambos casos, propuestos por la Cátedra. Para ello, el alumno deberá requerirlo a la Cátedra con treinta (30) días de anticipación a la fecha del examen final, a fin de obtener el conjunto de trabajos prácticos que deberá resolver. Los mismos deberán ser entregados al docente responsable cinco (5) días antes de la fecha de examen final.
- Superada la instancia anterior, el alumno estará en condiciones de rendir un primer examen final escrito sobre aspectos teóricos (25%) y prácticos (75%) de la Unidad Curricular.
- En caso de aprobar el examen escrito, el alumno estará en condiciones de rendir un segundo examen final de características similares al que rinden los alumnos regulares, sobre aspectos teóricos de la materia. Este examen será tomado en forma oral y versará sobre las temáticas abordadas en el programa de la Unidad Curricular, desde la perspectiva de la bibliografía específica.

PARA RENDIR LA UNIDAD CURRICULAR EN CALIDAD DE ALUMNO REGULAR

En base a la normativa vigente (RAM), para rendir esta Unidad Curricular en calidad de alumno regular el alumno deberá rendir un primer examen final escrito sobre aspectos teóricos de la misma. Este examen versará sobre las temáticas comprendidas en el programa de la asignatura, desde la perspectiva de la bibliografía obligatoria. Los criterios de evaluación se basan en la acreditación de la lectura de la bibliografía, la argumentación sólida que utilice la terminología y los conceptos del campo de estudio y la apropiación crítica de los mismos.

6. MATERIALES PARA EL APRENDIZAJE

- Gabinete de Informática, con la suficiente cantidad de computadoras con el siguiente software debidamente instalado: Windows 7+ y con el Entorno de Desarrollo Integrado (EDI) CodeBlocks.
- Apuntes de Cátedra.
- Guía de trabajos prácticos
- Pizarra y fibrones.
- Notebook.
- Cañón proyector.

7. DISTRIBUCIÓN EN EL TIEMPO DE LOS CONTENIDOS PROGRAMADOS

Núcleo	Marzo			Abril			Mayo			Junio		
PARADIGMAS Y LENGUAJES DE PROGRAMACIÓN	R	R	F									
INTRODUCCIÓN A LA PROGRAMACIÓN				F	J		F	J				
TIPOS DE DATOS ESTRUCTURADOS ESTÁTICOS									F			
PROGRAMACIÓN MODULAR												
EXAMEN PARCIAL												
EXAMEN RECUPERATORIO												

Referencias:

- F** Día feriado o no laborable
- J** Jornada educativa
- R** Receso de verano/invierno

.....
 Firma del docente