

Carrera: INGENIERIA EN INFORMATICA		
Asignatura: 3635 - Tópicos de programación		
Trayecto Programación		
Año académico 2023		
Responsable / Jefe de cátedra: Lic. Guatelli, Renata Silvia		
Carga horaria semanal 4 hs.	Carga horaria total 64 hs.	Créditos -----
Modalidad: Presencial		
Correlativas anteriores PROGRAMACION ESTRUCTURADA BASICA - MATEMATICA DISCRETA	Correlativas posteriores ALGORITMOS Y ESTRUCTURAS DE DATOS – TALLER DE INTEGRACION - SEGURIDAD DE LA INFORMACION	
Conocimientos necesarios -----		

Descripción de la asignatura

La asignatura “Tópicos de programación”, refuerza los conocimientos adquiridos en asignaturas previas introduciendo a la programación imperativa estructurada soluciones alternativas mediante recursividad. Además presenta el concepto de reusabilidad, a través del diseño de algoritmos independientes del tipo de dato y de una acción particular de forma de crear soluciones realistas. Se presentan estrategias de diseño de algoritmos para mejorar soluciones ineficientes y diferentes implementaciones para un problema en particular, trabajando sobre tipos de datos simples, compuestos, estáticos y dinámicos.

Metodología de enseñanza

Se utilizan metodologías activas de enseñanza, especialmente aprendizaje basado en problemas (ABP). Se presenta cada unidad temática introduciendo los conceptos fundamentales realizando analogías con ejemplos reales, que permite relacionar los contenidos de la materia con las herramientas habituales de trabajo.

Los contenidos de la asignatura se presentan de forma iterativa e incremental, de forma que le permitan al estudiante, construir sus propios procedimientos para resolver una situación problemática, lo que implica que sus ideas puedan verse modificadas y de esta forma siga construyendo nuevos conocimientos.

La materia tiene una fuerte carga práctica. En la misma se resuelven problemas, desarrollando pequeños sistemas, aplicando lo visto en las clases, trabajando de forma colaborativa, simulando un entorno de trabajo real.

Se motiva a los estudiantes en el uso de los foros de la plataforma MIEI y recursos de Teams, para la resolución de dudas, tanto de conceptos teóricos como prácticos, permitiendo desarrollar las capacidades de comunicación y afianzar el uso del lenguaje técnico. Además,

la cátedra cuenta con soporte digital de los contenidos, que los estudiantes pueden consultar luego de haber asistido a la clase.

Objetivos de aprendizaje

A través de esta asignatura, el alumno habrá adquirido los conocimientos necesarios y suficientes para estar en condiciones de:

Objetivos Generales:

La materia se desarrolla teniendo en cuenta los siguientes objetivos generales:

- analizar, plantear y resolver situaciones problemáticas.
- organizar y planificar su trabajo.
- hacer transferencia de los conocimientos teóricos a la práctica.
- adquirir la capacidad de trabajar en equipo.
- identificar un problema a partir de una situación problemática presentada.
- diseñar un algoritmo eficiente para la resolución de una situación problemática analizada.
- desarrollar un algoritmo utilizando un lenguaje de programación.
- plantear casos de prueba de forma tal de ver los casos generales y particulares de cada situación problemática planteada.
- expresar los contenidos teóricos de la materia y su vinculación con situaciones de la vida real.
- detectar la fuerte vinculación de esta asignatura con materias de años anteriores y posteriores del plan de carrera.
- integrar grupos de trabajo, potenciando su propio aprendizaje a través de la interacción y cooperación con sus pares.
- utilizar con fluidez el lenguaje técnico relacionado con la materia.

Objetivos Específicos:

- Aplicar los principios de la Programación Estructurada.
- Aplicar el concepto de reusabilidad.
- Manejar con solvencia los diferentes tipos de pasaje de parámetros, tanto de variables como de funciones.
- Aplicar sus conocimientos de algoritmia a colecciones de datos.
- Implementar algoritmos realistas independientes del tipo de dato.
- Diseñar e implementar soluciones que no dependan de un tipo de dato en particular, de forma que puedan utilizarse para diversos tipos de datos y puedan realizar diferentes acciones en función de la acción que reciba como parámetro, administrando correctamente la memoria en tiempo de ejecución.
- Entender, decidir y defender sobre alternativas de implementación conociendo las implicancias de cada una.
- Entender las diferentes estrategias de persistencia y aplicarla en ejemplos concretos.

Contenidos mínimos

Algorítmica. Estructuras de control. Tipos de datos simples y compuestos. (estáticos y dinámicos). Procedimientos y funciones. Concepto de reusabilidad. Estrategias de diseño de algoritmos.

Competencias a desarrollar

Genéricas:

- Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.
- Utilización de técnicas y herramientas de aplicación en la ingeniería en sistemas de información / informática.
- Desempeño en equipos de trabajo.
- Comunicación efectiva.
- Actuación profesional ética y responsable.
- Aprendizaje continuo.
- Desarrollo de una actitud profesional emprendedora.
- Concepción, diseño y desarrollo de proyectos de ingeniería en sistemas de información / informática.

Específicas:

- Especificación, proyecto y desarrollo de software.
- Especificación, proyecto y desarrollo de sistemas de información.
- Establecimiento de métricas y normas de calidad de software.

Programa analítico	
Unidad 1	Arrays Arrays unidimensionales. Concepto de puntero. Aritmética de Punteros. Manejo de arrays unidimensionales con diferentes notaciones. Arrays numéricos. Cadenas. Funciones de biblioteca. Arrays multidimensionales. Manejo de arrays multidimensionales con diferentes notaciones. Uso de arrays.
Unidad 2	Tipos de memorias Tipos de memoria. Memoria de pila. Memoria Heap. Manejo de memoria en tiempo de ejecución. Ventajas y desventajas de cada tipo de memoria.
Unidad 3	Pasaje de parámetros Concepto de pasaje de parámetros. Pasaje de parámetros por copia o valor. Pasaje de parámetros por dirección. Punteros a variables. Punteros a funciones. Funciones de biblioteca de ordenamiento y búsqueda. Creación de funciones independientes del tipo de dato. Concepto de funciones map, filter y reduce. Argumentos en el main. Macro reemplazo. Similitudes y diferencias entre funciones y macros.
Unidad 4	Tipo de Dato Abstracto (TDA)

	Concepto de Tipo de Dato Abstracto. Creación e implementación de un TDA. Diferencia entre TDA y estructuras de datos.
Unidad 5	Persistencia de memoria Archivos binarios y de texto, su creación, modos de acceso, posicionamiento, cierre, eliminación, y funciones relacionadas. Creación de archivos como lotes de prueba. Merge o apareo de archivos.
Unidad 6	Introducción a la Recursividad Concepto de recursión. Estructura de funciones recursivas. Diferencias entre recursividad e Iteración.

Planificación de actividades [15/16 semanas dependiendo del calendario académico universitario – UNLaM]					
Semana	Unidad /de s	Actividad	Tipo	Duración estimada	
Semana 1	1	Presentación de los docentes del curso, breve explicación de las pautas generales de la materia. Breve repaso de conceptos adquiridos en “Programación Estructurada Básica”. Trabajar en forma modular. Creación de funciones. Macro reemplazo. Similitudes y diferencias entre macros y funciones. Pasaje de parámetros por copia o valor.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	3
Semana 2	2	Breve explicación de cómo trabajar con proyectos y cómo documentar. Edición, compilación, depuración y ejecución de programas, pautas de trabajo en laboratorio. Arrays unidimensionales. Concepto de puntero. Aritmética de punteros. Relación entre punteros y vectores. Arrays numéricos. Uso de punteros y aritmética de punteros en lugar del uso de índices para arrays unidimensionales. Pasaje de parámetros por dirección. Variables dimensionadas como parámetros. Presentación del trabajo práctico.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	1 - 3
Semana 3	3	Arrays de caracteres. Uso de punteros y aritmética de punteros en lugar del uso de índices para arrays de caracteres. Funciones de conversión. Revisión de funciones de biblioteca,	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	1

		reemplazo de las mismas por versiones propias. Estrategias para buscar, contar, etc. palabras en una cadena. Estrategia para resolver una función que permite 'normalizar' una cadena de caracteres.			
Semana 4	4	Arrays Multidimensionales. Ejemplificación del uso de arrays bidimensionales. Arrays de punteros. Aritmética de punteros en lugar del uso de índices para arrays bidimensionales. Punteros a punteros. Desarrollar funciones sobre matrices. Recorridos y procesamiento de arrays de más de una dimensión. Argumentos a main.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	1
Semana 5	5	Tipos de memorias. Gestión de memoria dinámica, funciones de solicitud y devolución de memoria. Ejemplificación. Creación de arrays en tiempo de ejecución. Memoria de Pila. Memoria heap.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	2
Semana 6	6	Punteros a funciones. Concepto de punteros a función. Definición de punteros a función. Uso de punteros a función. Función de ordenamiento. Desarrollo de funciones propias que permitan ordenar vectores de cualquier tipo de dato, utilizando diferentes métodos de ordenamiento. Desarrollo de funciones independientes del tipo de dato.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	3
Semana 7	7	Estructuras de datos. Punteros a estructuras de datos. Concepto de archivo. Tipos de archivos. Archivos binarios y de texto, modos de apertura. Archivos binarios. Acceso secuencial. Ejemplificación. Desarrollo de funciones genéricas para operar sobre archivos.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	5
Semana 8	8	Actualización masiva de archivos binarios. Búsqueda de un registro por su clave o por su posición relativa dentro del archivo y actualización del mismo. Desarrollo de funciones genéricas para operar sobre archivos.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	3 - 5

Semana 9	9	Archivos de texto. Archivos de texto de longitud fija y variable. Convertir información de archivos de texto en binarios y viceversa. Ejercitación sobre conversión de archivos.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	3 - 5
Semana 10	10	Proceso de merge. Creación de archivos como lotes de prueba. Ejercitación sobre archivos de diferentes tipos.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	3 - 5
Semana 11	11	Presentación del concepto de TDA. Conveniencia de trabajar con tipos abstractos de datos. Diferencia entre estructura de datos y TDA. Presentación del TDA fecha. Breve explicación de cómo documentar. Armado de un proyecto, pruebas de las distintas operaciones.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	4
Semana 12	12	Ejercitación sobre todos los temas vistos hasta el momento. Consultas sobre el trabajo práctico.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	1 - 2 -3 - 4 -5
Semana 13	13	Parcial 1 Entrega trabajo práctico	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	1 - 2 -3 - 4 -5
Semana 14	14	Devolución parcial 1. Defensa del TP. Introducción a la recursividad. Concepto. Elementos característicos de las funciones recursivas. Ventajas y desventajas de implementaciones recursivas vs. iterativas.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	6
Semana 15	15	Recuperatorio Parcial 1 Defensa del TP.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	1 - 2 -3 - 4 -5 - 6
Semana 16	16	Entrega de notas finales. Defensa trabajo práctico	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	1 - 2 -3 - 4 -5 - 6

Evaluación

Descripción del proceso evaluativo desarrollado por la cátedra

El proceso de evaluación consta de:

- un examen parcial presencial incluye las unidades 1 a 5.
- un trabajo práctico grupal.
- un examen recuperatorio presencial incluye las unidades 1 a 6.

Las evaluaciones podrán ser teórico - prácticas. o únicamente prácticas o teóricas.

Las evaluaciones teóricas constan de preguntas para desarrollar y/o preguntas del tipo opción múltiple y/o verdadero / falso, justificando la respuesta o no.

Las evaluaciones prácticas constan de la resolución de ejercicios en máquina. Desarrollando el código necesario para resolver el problema solicitado y el lote de pruebas correspondiente.

Una vez entregado el parcial, el mismo es corregido por los docentes, probando que cumpla con la consigna solicitada. A la semana siguiente, se hace una devolución grupal del mismo. Luego se hace la devolución individual.

El trabajo practico grupal se evalúa en diferentes niveles, autoevaluación por parte de los integrantes del grupo; evaluación entre pares, cada grupo verifica el trabajo de otro grupo y hace las correcciones que crea conveniente; por último, los docentes verifican el trabajo original y las correcciones realizadas. Se realiza una puesta en común del ejercicio planteado y se construye una solución grupal.

Primera evaluación	Aproximadamente Semana 13	teórico - práctica	4 hs
Aprobación TP Grupal	A partir de la semana 13	teórico - práctica	4 hs
Recuperatorio	Aproximadamente Semana 15	teórico - práctica	4 hs

Bibliografía obligatoria

Título	Autor	Editorial	Edición	Año
El Lenguaje de Programación C	Kernighan y Ritchie	Prentice Hall	2.ed.	1991
Cómo Programar en C / C++	Deitel y Deitel	Prentice Hall	2a. ed.	c1994
Apuntes de cátedra	Docentes de la cátedra	Disponible en MIEI		

Bibliografía complementaria recomendada

Título	Autor	Editorial	Edición	Año

Otros recursos obligatorios [Videos, enlaces, otros. Incluir una fila por cada recurso]

Nombre	

Otros recursos complementarios [Videos, enlaces, otros. Incluir una fila por cada recurso]	
Nombre	
https://pythontutor.com/c.html#mode=edit	Permite visualizar en el navegador lo que la computadora está haciendo paso a paso mientras ejecuta un programa.
https://loom.wecode.io	Plataforma para realizar revisión entre pares