

PROGRAMA DE ASIGNATURA

CARRERA: INGENIERÍA EN INFORMÁTICA	PLAN: PII2016_V001	AÑO ACADÉMICO: 2023
ASIGNATURA: INFORMÁTICA II		COD: 000455
DPTO: COMPUTACIÓN E INFORMÁTICA	TIPO: OBLIGATORIA	BLOQUE: TECNOLOGÍA BÁSICAS
HORAS TOTALES: 96	CURSO: I	SEMESTRE: SEGUNDO

OBJETIVOS

- Resolver problemas utilizando las estructuras básicas del lenguaje C/C++.
- Comprender el concepto del tipo abstracto de datos
- Comprender la utilización de punteros.
- Resolver problemas utilizando las estructuras de datos estáticas y dinámicas

TEMA	CONTENIDOS
I	<p>PUNTEROS Y MEMORIA DINÁMICA</p> <p>Puntero a void y casteo o conversión de tipos. Aritmética de punteros. Relación entre punteros y arreglos. Punteros a funciones. Concepto y ejemplo de callback. Arreglos de punteros. Indirección múltiple. Manejo de memoria dinámica. Distribución de memoria de un programa: stack y heap. Las funciones malloc, realloc, free. El operador sizeof. Operadores New y delete. Detección de leaks (valgrind). Creación de arreglos multidimensionales utilizando arreglos de punteros y memoria dinámica. Estructuras. Definición. Campos de una estructura. Declaración. Utilización de typedef. Arreglos de estructuras. Punteros a estructuras. Paso de estructuras a funciones por valor y referencia. Anidación de estructuras.</p>
II	<p>PERSISTENCIA DE DATOS</p> <p>Secuencias y archivos. Puntero a archivos. Modos de apertura de un archivo. Cierre y borrado de archivos. Escritura y lectura de caracteres: el código ascii. Escritura y lectura de cadenas. Escritura y lectura de más de un byte. Persistencia de estructuras. Archivos de modo texto y Archivos binarios.</p>
III	<p>TIPOS ABSTRACTOS DE DATOS</p> <p>Concepto. Implementación de tipos abstractos de datos. Estructuras autoreferenciadas. Definición. Lista simple y doblemente enlazada. Pila. Cola. Buffer circulares. Algoritmos de push, pop, print, find, remove según corresponda al tipo de estructura. Implementación utilizando funciones: necesidad de la utilización de indirección múltiple.</p>

IV	<p>RECURSIÓN</p> <p>Definición. Búsqueda del caso recursivo y el caso base. Comparación entre recursión vs iteración. Quicksort. Calificadores de acceso para funciones y variables: static, volatile, extern. Separación de archivos en .h y .c</p>
V	<p>OPERACIONES DE BAJO NIVEL</p> <p>Operadores AND, OR, XOR y complemento. Operadores de desplazamiento. Tipos de datos signados y no signados. Uso de librería stdint. Campos de bits. Definición. Acceso a los miembros. Problemas de portabilidad. Uso de sizeof. Implementación. Máscaras y corrimientos. Uniones. Definición. Declaración e inicialización de uniones. Acceso a miembros de la unión. Utilización junto a estructuras y campos de bits. Ejemplos de decodificación de tramas RS232, satelitales, GPS, etc donde se necesite el acceso a bits individuales, cálculo de CRC. Enumeraciones. Definición. Declaración y uso.</p>

BIBLIOGRAFÍA BÁSICA

- Deitel H, Deitel P. Cómo programar en C / C++ y Java. Ed. Pearson Educación.
- Weiss Mark, Estructuras de datos en Java. Ed Pearson.

BIBLIOGRAFÍA COMPLEMENTARIA

- Bronson G. C++ para Ingeniería y ciencias. Ed. Thomson.

DESCRIPCIÓN DE LA ACTIVIDAD CURRICULAR

Actividades a desarrollar por el/los docente/s:

- Atender al alumno ante consultas propias de la asignatura como así también las de relación docente-alumno.
- Fomentar el estudio independiente.
- Preparar clases teóricas y/o prácticas.
- Asistir periódicamente a reuniones de cátedra.
- Preparar evaluaciones y actividades de seguimiento del alumno.
- Confeccionar, revisar y actualizar el programa de la asignatura a su cargo.
- Confeccionar, revisar y actualizar las guías de ejercicios prácticos.
- Revisar y actualizar la bibliografía y material didáctico en aula virtual.
- Cumplir además con toda otra función que surja de su tarea para garantizar el correcto desarrollo de la carrera.

Actividades a desarrollar por el alumno:

- Asistir a clases teóricas y prácticas.
- Realizar consultas.
- Realizar autoevaluaciones.

- Realizar guías de ejercicios prácticos.
- Desarrollar un trabajo integrador grupal.

Materiales, recursos, soporte:

Espacios físicos:

- Aula
- Laboratorio de Informática

Recursos tecnológicos:

- Aula virtual de Moodle
- Equipamiento informático PC o computadora portátil;
- Proyector multimedia;
- Conectividad a internet;
- Software: IDE de desarrollo para C/C++

METODOLOGÍA DE ENSEÑANZA

En esta asignatura se enfoca en proporcionar a los estudiantes los conocimientos fundamentales necesarios para comprender los conceptos básicos de la informática y la programación. La metodología de enseñanza empleada en esta asignatura se basa en una combinación de técnicas pedagógicas diseñadas para facilitar el aprendizaje y la retención de los contenidos.

En las clases teóricas, se utilizará como estrategia de enseñanza la *lección magistral participativa* con ejemplos prácticos para que los estudiantes puedan entender de manera clara y concisa los temas y conceptos. Además, se realizarán las primeras implementaciones, para que los estudiantes puedan ver cómo funcionan los conceptos teóricos en la práctica.

Además, se empleará el *aprendizaje invertido* en donde los estudiantes tendrán material bibliográfico y/o audiovisual en el aula virtual antes de asistir a la clase, para luego en la clase, el docente retomará los contenidos para seguir profundizando en los conceptos y realizar actividades prácticas que les permitan aplicar lo aprendido.

En las clases prácticas, se utilizará la estrategia de *resolución de ejercicios* en la cual tendrán en el aula virtual guías de ejercicios que cubran los temas tratados en la teoría. Estas guías irán desde ejercicios más simples hasta ejercicios de mayor complejidad, de manera que los estudiantes puedan ir progresando en su aprendizaje de manera gradual y sistemática. De esta manera, los estudiantes tendrán la oportunidad de aplicar los conocimientos adquiridos en la teoría y practicar habilidades en programación.

A modo de integración de los contenidos y las habilidades desarrolladas por los estudiantes, se implementará la estrategia de *aprendizaje cooperativo en grupos pequeños*, en la cual formarán equipos que deberán desarrollar un trabajo integrador a lo largo de la cursada y concluirá con una *presentación oral*, fortaleciendo la comunicación efectiva en la oralidad.

CRITERIOS Y FORMAS DE EVALUACIÓN

La evaluación es una instancia más de aprendizaje. Se considera su dimensión diagnóstica, para evaluar los saberes previos acerca del diseño e implementación de algoritmos utilizando las estructuras de control de la programación, como también la manipulación de arreglos y programación modular a través de funciones; su dimensión formativa, para un seguimiento continuo y de aprendizaje controlado tanto en las actividades propuestas en las clases teóricas y prácticas a lo largo de la cursada como por ejemplo discusiones en clases, cuestionarios de autoevaluación, ejercicios prácticos individuales o de a pares; y

sumativa, para que las calificaciones (juicio de valor) reflejen los resultados de aprendizaje, para ello se realizarán dos instancias parciales durante la cursada en la cual luego de cada una de ellas se realizará una puesta en común con los errores más comunes y se elaborará una solución con la participación de todos los estudiantes y la presentación y defensa oral del trabajo integrador grupal. Finalmente se evaluará el examen final de la asignatura.

Criterios de Evaluación

Implementación de Estructuras de Datos Dinámicas:

- Utilizar punteros y gestión de memoria dinámica en la implementación de estructuras de datos dinámicas para resolver problemas específicos.
- Demostrar la capacidad de seleccionar y aplicar la estructura de datos adecuada según los requisitos del problema.
- Identificar y solucionar posibles problemas de gestión de memoria y punteros en la implementación.

Manipulación de Archivos para la Gestión de Datos:

- Programar operaciones básicas de lectura, escritura y manipulación de archivos para gestionar datos persistentes.
- Aplicar conocimientos de manipulación de archivos en contextos prácticos, como la creación, edición y eliminación de datos almacenados.

Diferenciación entre Iteración y Recursión:

- Identificar situaciones adecuadas para aplicar estrategias de iteración y recursión en la resolución de problemas repetitivos.
- Comparar y contrastar las ventajas y desventajas de la iteración y la recursión en términos de eficiencia y legibilidad del código.

Implementación de Operaciones de Bajo Nivel para la Optimización:

- Diseñar y programar operaciones de bajo nivel que optimicen el uso de recursos como memoria y tiempo de ejecución.
- Demostrar comprensión de las técnicas de optimización de código, como la manipulación directa de registros y la reducción de instrucciones redundantes.

Exposición Oral de Programas Elaborados:

- Comunicar efectivamente las soluciones de programas desarrollados de manera oral, utilizando el código fuente como apoyo visual.
- Explicar de manera clara y concisa los conceptos clave detrás de las soluciones implementadas y demostrar su comprensión.
- Utilizar presentaciones visuales y ejemplos para respaldar la exposición oral y fomentar la comprensión de los espectadores.

Evidencias de aprendizaje o instrumentos de evaluación

- Realizar los cuestionarios de autoevaluación en el Aula Virtual.
- Resolución de las guías de ejercicios, las entregas se realizan de manera individual a través del aula virtual.
- Realizar y aprobar dos parciales con opción de recuperar ambos.
- Realizar y aprobar el trabajo integrador grupal.
- Realizar el examen final de la asignatura.

Condiciones Generales

Para obtener la regularidad de la asignatura, el estudiante debe:

- Realizar el 100% de las actividades propuestas en la cátedra;
- Aprobar dos instancias parciales teórico-prácticas con la posibilidad de recuperar ambos. La nota mínima de aprobación es “cuatro” y corresponde al 50% de la prueba.
- Aprobar la presentación y defensa del trabajo integrador. La nota mínima de aprobación es “cuatro” y es individual.
- Participar del 80% de los encuentros presenciales;

Para acreditar la asignatura el estudiante debe:

- Poseer la regularidad en la actividad curricular;
- Rendir el examen final de rigor con un rendimiento igual o superior al 50%. Se calificará de acuerdo a la norma establecida, de 1 a 10, siendo el porcentaje anteriormente mencionado como criterio mínimo para su aprobación.

Tanto las instancias parciales como el examen final consisten en una parte teórica en donde habrá preguntas acerca de los conceptos vistos y una parte práctica en donde tendrá uno o dos ejercicios a resolver utilizando un lenguaje de programación. Se deberá aprobar el 50% de cada parte para la aprobación del examen final.

Consultas

Fuera del ámbito de clase, el canal de comunicación de la cátedra es a través del aula virtual y para avisos importantes por correo electrónico. Además, se tendrán abiertos foros de consultas en el aula virtual que estarán habilitados en forma permanente.

Cumplimiento

Todas las actividades y tareas tienen fecha de vencimiento de presentación. En caso de incumplimiento, con la debida justificación, se otorgará prórroga.