

Información del curso

Pedro O. Pérez M., PhD.

Programación de estructuras de datos y algoritmos fundamentales
Tecnológico de Monterrey

pperezm@tec.mx

07-2021

Información del profesor

Información del profesor

Información del curso

Competencias

Contenidos de aprendizaje

Metodología

Evaluación

Normas de clase

Bibliografía

Información del profesor

- ▶ Pedro Oscar Pérez Murueta
 - ▶ ISC Mayo 1994
 - ▶ MTI Mayo 2002
 - ▶ DCC Diciembre 2019
- ▶ Correo: pperezm@tec.mx
- ▶ Oficina: Edificio 2, Piso 3
- ▶ Horario de asesoría: Se encuentra en la puerta de mi oficina.



SICT0300 - Solución de problemas con computación

Soluciona problemas de diferentes niveles de complejidad mediante la aplicación de metodologías computacionales y de tecnologías de información en ambientes controlados y de incertidumbre.

Subcompetencia	Nivel de dominio
SICT0301 - Evaluación del problema: Evalúa los componentes que integran una problemática de acuerdo a principios y procesos computacionales	B
SICT0302 - Toma de decisiones: Toma decisiones en la solución de problemas en condiciones de incertidumbre y diferentes niveles de complejidad con base metodologías de investigación y de cómputo.	B
SICT0303 - Implementación de acciones:: Implementa acciones científicas e ingenieriles o procesos computacionales que cumplen con el tipo de solución requerida.	B

SEG0700 - Transformación Digital

Genera soluciones a las problemáticas de su ámbito profesional, con la incorporación inteligente y oportuna de tecnologías digitales de vanguardia.

Subcompetencia	Nivel de dominio
SEG0702 - Tecnologías de Vanguardia: Evalúa diversas tecnologías con apertura a la búsqueda e implementación de alternativas relevantes en la transformación de la práctica profesional.	A

¿Qué necesito aprender?

Para desarrollar las subcompetencias, alcanzar los niveles de dominio, y responder a las situaciones problema planteadas en esta materia, es necesario aprender diferentes conceptos, teorías, procedimientos, técnicas, actitudes y valores, entre otros contenidos.

Conceptuales

- ▶ Abstracción de Datos.
- ▶ Recursión.
- ▶ Análisis de Complejidad.
- ▶ Algoritmos de Ordenamiento.
- ▶ Algoritmos de Búsqueda.
- ▶ Estructuras de Datos Lineales
- ▶ Estructuras de Datos No Lineales:
 - ▶ Jerárquicas.
 - ▶ Red (Grafos).
 - ▶ Conjunto (Técnicas de Hashing)

Procedimentales

- ▶ Identifica los principales elementos de la problemática.
- ▶ Realiza una abstracción de los elementos con el fin de diseñar las estructuras que mejor representen al problema.
- ▶ Aplica análisis de complejidad en la búsqueda de una solución óptima.
- ▶ Selecciona, en función a las características del problema, el algoritmo y/o la estructura de datos más conveniente.
- ▶ Codifica soluciones en lenguajes de programación de alto nivel.
- ▶ Desarrolla los algoritmos que mejor se adapten a la estructura seleccionada.
- ▶ Utiliza dispositivos programables en el proceso de solución.
- ▶ Analiza y demuestra matemáticamente los diferentes comportamiento de los algoritmos computacionales, en las diferentes estructuras de datos.
- ▶ Desarrolla soluciones eficientes a situaciones reales utilizando los diferentes estructuras de datos y algoritmos.

Actitudinales

- ▶ Aprecia la necesidad de un análisis sistemático de una situación problemática con la finalidad de identificar adecuadamente los factores que dan origen a las desviaciones observadas.
- ▶ Asume responsabilidad en las decisiones que toma para la solución de problemas consciente que ésta debe estar basada en metodologías y herramientas ingenieriles y de investigación.
- ▶ Es consciente de la necesidad de una exploración sistemática de posibles soluciones para identificar las mejores opciones.
- ▶ Valora la necesidad de realizar procesos computacionales eficientes, entendiendo la diferencia en los tiempos de ejecución de cada tipo de proceso.
- ▶ Concientiza la importancia del uso eficiente de algoritmos y estructuras de datos a problemáticas de la vida cotidiana.
- ▶ Acepta la responsabilidad del uso correcto de los procesos computacionales en situaciones que apoyen a la vida cotidiana.

Metodología

Esta materia consta de cinco partes o módulos que te permitirán aprender y dominar tanto algoritmos importantes o fundamentales como estructura de datos, La combinación de algoritmos y estructura de datos es una mancuerna que no se puede disociar, Nikalaus Wirth publico en 1976 un libro que a la fecha es toda una referencia: $\text{Algorithms} + \text{Data Structures} = \text{Programs}^*$. Este libro clásico de la computación pone de manifiesto que los algoritmos y las estructura de datos están íntimamente relacionados, así por ejemplo, si se tiene un arreglo ordenado, se utilizará un algoritmo óptimo de búsqueda para arreglos ordenados.

Evaluación

Ver sitio en Canvas.

Normas de clase

Exámenes

- ▶ Los exámenes podrán ser presentados solamente en la fecha estipulada. El no presentar un examen implica una calificación de NP (No Presentó).

Calificaciones

- ▶ Las calificaciones parciales y final se expresan en escala de uno a cien.
- ▶ La calificación mínima aprobatoria es 70 (SETENTA).

Asistencia a clases

En lo que respecta a esta clase:

- ▶ La sesión de clase inicia 5 minutos después del horario establecido (11:05). Si no estás al inicio de la misma, se considerará que no asististe a esa sesión. Asimismo, también se considera inasistencia si te retiras, sin permiso del profesor, antes de terminar la sesión de clase.
- ▶ No podrás acreditar, bajo ningún concepto, las actividades (tareas y/o exámenes) de las sesiones a las cuales no hayas asistido. Además, será tu responsabilidad estudiar el material visto en esas sesiones.

Tareas y Proyectos

- ▶ Toda tarea y/o proyecto tendrá su fecha y horario de entrega que es inamovible. Vencido el término de entrega no se recibirán tareas y/o proyectos.
- ▶ Todas las tareas son individuales a menos que explícitamente se pida trabajar en grupo.

Redacción y Organización

- ▶ La mala redacción, organización y ortografía en la elaboración de tareas, proyectos, presentaciones y exámenes, será causa de penalización en la calificación correspondiente.

Faltas a la Integridad Académica en Tareas, Proyectos o Exámenes

- ▶ Las faltas a la integridad académica, como la copia o tentativa de copia en cualquier tipo de examen o actividad de aprendizaje; el plagio parcial o total; facilitar alguna actividad o material para que sea copiada y/o presentada como propia; la suplantación de identidad; falsear información; alterar documentos académicos; vender o comprar exámenes o distribuirlos mediante cualquier modalidad; hurtar información o intentar sobornar a un profesor o cualquier colaborador de la institución; entre otras acciones más son consideradas faltas grave. Cuando un alumno cometa un acto contra la integridad académica, se le asignará una calificación reprobatoria a la actividad, examen, período parcial o final. La calificación reprobatoria asignada por el profesor será inapelable, y a esta sanción se sumarán las otras posibles que determine el Comité de Integridad Académica de Campus. Esto tal como lo indica el Reglamento Académico en su CAPÍTULO IX: Faltas a la integridad académica.

Baja de materias

- ▶ Artículo 4.8 Los alumnos podrán darse de baja en una o varias unidades de formación inscritas durante el periodo académico semestral o periodos intensivos, **antes del último día de clases de la unidad de formación a dar de baja, conforme lo establezca el Calendario Escolar correspondiente.** Las unidades de formación dadas de baja no se registrarán como reprobadas. En el caso de que el alumno solicite la baja de todas sus unidades de formación, deberá pasar nuevamente por un proceso de admisión, en el que se tomará en cuenta su historial académico para la decisión. El proceso relacionado con la baja de las unidades de formación se llevará a cabo de acuerdo con las políticas y procedimientos administrativos definidas por la Dirección de Servicios Escolares para ese fin.

Bibliografía

- ▶ **Libro de texto:** Weiss, M. A. (2012). Data structures & algorithm analysis in C++. Pearson Education.
- ▶ **Libro de referencia:** Drozdek, A. (2012). Data Structures and algorithms in C++. Cengage Learning.
- ▶ **Libro de referencia:** Clifford, A. S. (2013). Data structure and algorithm analysis. Department of Computer Science Virginia Tech. Blacksburg.

Zoom



<https://itesm.zoom.us/my/pperezm>

Google calendar



<https://shorturl.at/ertE1>

Google drive



<https://shorturl.at/sMOW1>

Github



`https:`
`//github.com/Manchas2k4/TC1031`

Remind



`https:`
`//www.remind.com/join/d433hc`

¿Qué necesitamos instalar?

- ▶ Cuenta en Github: <https://github.com/>
- ▶ Cuenta en Repl.it: <https://replit.com/>
- ▶ Windows: Instalación de WSL2 (https://youtu.be/NDcT5_21j0s)
- ▶ Atom (<https://atom.io/>) o Sublime Text (<https://www.sublimetext.com/>)