

# Información del curso

Pedro O. Pérez M., PhD.

Análisis y diseño de algoritmos avanzados  
Tecnológico de Monterrey

*pperezm@tec.mx*

01-2022

## 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](mailto:pperezm@tec.mx)
- ▶ Horario de asesoría: <https://shorturl.at/flxGT>



# SICT0100 - Fundamentación de sistemas computacionales

Fundamenta el comportamiento de procesos computacionales y de tecnologías de información con base en principios de ciencias naturales y las matemáticas.

Subcompetencia	Nivel de dominio
SICT0101 - Explica el funcionamiento de sistemas computacionales por medio de argumentaciones sustentadas en las interacciones entre los componentes y su entorno creando modelos conceptuales donde se describan los componentes y la relación con su entorno.	C

## SICT0400 - Compromiso con la sustentabilidad

Aplica estándares, normas y principios de sustentabilidad en el desarrollo de sistemas computacionales y de tecnologías de información.

Subcompetencia	Nivel de dominio
SICT0401 - Aplica los estándares y normas propios de su profesión contrastándolos contra las restricciones de uso de acuerdo al proceso, producto o servicio donde se va a aplicar usando las normas y estándares más relevantes al dominio del problema que se va a resolver, distinguiendo claramente entre ambos.	C

# STC0100 - Desarrollo de Algoritmos Computacionales

Soluciona problemas generando algoritmos computacionales eficientes bajo modelos y herramientas de las ciencias computacionales.

Subcompetencia	Nivel de dominio
STC0101 - Implementa algoritmos computacionales confiables y correctos que solucionan problemas.	B
STC0102 - Optimiza algoritmos computacionales robustos y eficientes que se aplican en el desarrollo de soluciones.	B

# SEG0500 - Razonamiento para la Complejidad

Integra diferentes tipos de razonamiento en el análisis, sintexis y solución de problemas, con disposición al aprendizaje continuo.

Subcompetencia	Nivel de dominio
SEG0502 - Pensamiento científico: Resuelve problemas e interrogantes de la realidad, a partir de metodologías válidas y confiables.	B

# Conceptuales

- ▶ Técnicas de diseño de algoritmos.
- ▶ Manejo de Strings.
- ▶ Grafos.
- ▶ Geometría computacional.
- ▶ Técnicas de búsqueda avanzada.



# Procedimentales

- ▶ Proceso de programación, estrategias para planificación de CPU, para administración de memoria, y para manejo de dispositivos.
- ▶ Fundamentación del desempeño de un algoritmo a través de teoría de análisis.

# Actitudinales

- ▶ Reconocer la importancia de hacer buen uso de los elementos de un lenguaje de programación y de los recursos de una computadora, apertura a nuevos paradigmas de programación.
- ▶ Reconocer la relevancia de generar código eficiente y uso adecuado de los recursos de cómputo.

# Metodología

- ▶ Esta materia consta de 5 partes (distribuidas en 5 módulos) que te permitirán conocer algoritmos avanzados, siguiendo en la línea de las estructuras de datos. Los contenidos detallados los puedes encontrar en la sección de contenidos.
- ▶ En esta Unidad de Formación, trabajaremos principalmente con las técnicas de Aula Invertida y Aprendizaje colaborativo.

- En cada uno de los periodos se trabajará siguiendo una secuencia de actividades y recursos que denominamos "secuencia de aprendizaje". La secuencia de aprendizaje será la siguiente:
1. Presentación de conceptos teóricos por parte del profesor.
  2. Programación en grupo (dirigido por el profesor) de algunos conceptos teóricos del punto 1.
  3. Programación de manera individual o en equipo de algunos otros conceptos expuestos en el punto 1.
  4. Solución en equipos a la actividades o situación problema.
  5. Presentación en equipo de la situación problema

Se cuenta con dos situaciones problema planteadas, para las cuales se requerirá la entrega de una actividad integradora para cada una, en las que tendrás la oportunidad de poner en práctica lo aprendido durante esta Unidad de Formación, tomando en cuenta todo lo aprendido en Unidades de Formación pasadas.

- ▶ La primera situación problema, dura las primeras 5 semanas, es relativa al uso en general de Técnicas de programación y problemas de manejo de Strings.
- ▶ La segunda situación problema, abarca todo el segundo periodo (las 5 semanas intermedias) y parte del tercero. Esta situación problema se relaciona con el uso de grafos y Geometría Computacional.

# Evaluación

<b>Actividades</b>	<b>Evidencia de dominio</b>
Actividades de aprendizaje	45 %
Actividad integradora 1 (Evidencia)	20 %
Actividad integradora 2 (Evidencia)	20 %
Reflexión final (Evidencia)	5 %
Póster (vídeo) de reflexión (Evidencia)	10 %

# Normas de clase

## 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 (9:05).

## Tareas

- ▶ Toda tarea 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

Recursos base

[Recursos de consulta](#)

## Recursos base



**Nombre:** Introduction to Algorithms

**Autor:** Thomas H. Cormen

**Edición:** 3

**Año:** 2009

**Editorial:** MIT Press

**ISBN-** 978-0-262-03384-8

Biblioteca digital, se requiere usar las credenciales del Tec como alumno para realizar la consulta: [Introduction to Algorithms](#)



**Nombre:** Introduction to The Design and Analysis of Algorithms

**Autor:** Anany Levitin

**Edición:** 3

**Año:** 2012

**Editorial:** PEARSON

**ISBN-13:** 978-0-13-231681-1

**ISBN-10:** 0-13-231681-1

Bibliotec Tec: [Introduction to The Design and Analysis of Algorithms](#)



**Nombre:** Algoritmos computacionales, Introducción al análisis y diseño

**Autor:** Sara Baase, Allen Van Gelder

**Edición:** 3

**Año:** 2002

**Editorial:** Pearson

**ISBN-10:** 970-26-0142-8

Biblioteca Tec (versión inglés): [Computer algorithms : introduction to design and analysis](#)

## Zoom



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

## Google calendar



<https://shorturl.at/ertE1>

## Google drive



<https://shorturl.at/pIN49>

## Github



`https:  
//github.com/Manchas2k4/tc2038`

## Remind



`https:  
//www.remind.com/join/4ac82f`

## ¿Cómo vamos a trabajar?

- ▶ Equipos de trabajo: 2 personas. ¿Al azar o quieren conformarlos?
- ▶ Ambiente de trabajo: Linux (Windows: WLS2).
- ▶ IDE: Atom.
- ▶ Github Classroom (revisar uso).