

Información del curso

Pedro O. Pérez M., PhD.

12-2025

Implementación de métodos computacionales

Tecnológico de Monterrey

pperezm@tec.mx

Información del profesor

Información del profesor

Información del curso

Metodología

Evaluación

Normas de clase

Bibliografía

Información del profesor

Información del profesor

- Pedro Oscar Pérez Murueta
 - ISC Mayo 1994
 - MTI Mayo 2002
 - DCC Diciembre 2019
- Correo: pperezm@tec.mx
- Zoom: <https://itesm.zoom.us/my/pperezm>
- Horario de asesoría: <https://calendar.app.google/Fo1j6BmYfGJwjctRA>



Información del curso

En esta unidad de formación aprenderás conceptos de la teoría de la computación y su aplicación a la creación de lenguajes de programación. Esto se hará por medio de la solución de un par de situaciones problema:

- Familiarizándose con un nuevo lenguaje de programación. Está relacionada con la forma en la que se diseñan e implementan los nuevos lenguajes de programación, utilizando herramientas de la Teoría de la Computación.
- El lenguaje de programación más popular. Trata de las características que hacen que un lenguaje de programación sea popular. Dichas características son parte de los diferentes paradigmas de lenguajes de programación que se han desarrollado a lo largo de la historia.

- A lo largo de la materia deberás repetir una y otra vez esta secuencia:
 1. Aprender conceptos teóricos y practicarlos haciendo ejercicios, de la misma forma que se hace en matemáticas.
 2. Explorar su relación con la creación de los lenguajes de programación.
 3. Diseñar, basado en los conceptos, el modelo utilizado para resolver el problema.
 4. Implementar el diseño por medio de la creación de un programa computacional en un lenguaje de programación específico, cumpliendo con las indicaciones dadas por el instructor en cuanto a la entrada que recibe el programa y la salida que se espera que tenga.
 5. Hacer pruebas para verificar que es correcto y que cumple con lo solicitado.
 6. Preparar la documentación adecuada para el manejo del programa y ponerlo listo para su publicación.
- Una vez que el programa o la tarea estén listos se deberá entregar por el medio indicado, antes de la fecha límite establecida por el instructor.

Actividades	Evidencia de dominio
Ejercicios de conceptos formales	12 %
Actividades de programación	30 %
E1. Convertir una expresión regular a DFA (Integradora)	16 %
E2. Validación con autómatas (Integradora)	16 %
E3. Programación Paralela (Integradora)	16 %
E4. Soluciones alternativas a las herramientas usadas para la solución de las situaciones problema (Reflexión)	10 %

Calificaciones

- Las calificaciones 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 (16:05).

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.

Tareas

- Todas las tareas son individuales a menos que explícitamente se pida trabajar en grupo.
- Toda tarea tendrá una fecha y horario de entrega **fijos e inamovibles**.
- Si no entregas la actividad a tiempo, **tendrás un plazo adicional de 2 semanas** para enviarla, pero se considerará **entrega tardía** y tendrá una penalización de **40 puntos sobre la calificación obtenida**.
- Después de ese plazo de 2 semanas, **ya no se aceptarán entregas de la actividad**.

Tareas

- Si decides utilizar herramientas de Inteligencia Artificial Generativa para completar esta actividad, **debes asegurarte de que las soluciones propuestas se basen exclusivamente en los conceptos, funciones y técnicas vistas en clase.**
- El uso de elementos o técnicas no abordadas en el curso **resultará en una reducción de puntos** y, en casos graves, podrías recibir una **calificación de cero** en la actividad.
- Además, es obligatorio **indicar explícitamente el uso de IA generativa en tu solución.** No hacerlo será considerado una falta de Integridad Académica.

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 4.9 Los estudiantes pueden darse de baja de una o varias unidades de formación durante el periodo académico establecido en el Calendario Escolar. Las fechas límite para darse de baja de una unidad de formación dependen de su duración. Por ejemplo, para una unidad de quince semanas, la fecha límite es el viernes de la décima semana. Si la fecha límite coincide con un día festivo, se adelanta al día hábil anterior. Es importante destacar que las unidades de formación dadas de baja no se registran como reprobadas. Si un estudiante se da de baja de todas las unidades, debe pasar por un nuevo proceso de admisión para reincorporarse, tomando en cuenta su historial académico. Este proceso se lleva a cabo según las políticas de la Dirección de Servicios Escolares.

[Recursos base](#)

[Recursos de consulta](#)

Recursos base



Nombre: Languages and machines: an introduction to the theory of computer science

Autor: Thomas Sudkamp

Edición: 3rd

Año: 2016

Editorial: Addison-Wesley

ISBN: 9780321322210



Nombre: Programming languages: principles and practice

Autor: Kenneth Loudon

Edición: 3rd

Año: 2011

Editorial: Cengage Learning

ISBN: 9780321322210



[Racket Programming Language](#) ➞



[Clojure Programming Language](#) ➞

Herramientas a utilizar

Herramientas a utilizar

1. IDE: Atom, Sublime, Visual Studio Code, cualquier IDE que les guste.
2. Python: <https://www.python.org/>
3. Racket: <https://racket-lang.org>
4. C++ en Windows:
<https://code.visualstudio.com/docs/cpp/config-mingw>
5. SWI-Prolog: <https://www.swi-prolog.org/>