Información del curso

Pedro O. Pérez M., PhD.

Desarrollo de aplicaciones avanzadas de ciencias computacionales
Tecnológico de Monterrey

pperezm@tec.mx

03-2025

Pedro Oscar Pérez Murueta

- Pedro Oscar Pérez Murueta
  - ISC Mayo 1994
  - MTI Mayo 2002
  - DCC Diciembre 2019
- Correo: pperezm@tec.mx
- Oficina: Parque Tecnológico, Piso 8.
- Horario de asesoría: Sitio en Canvas > Mis profesores.

- Benjamín Valdéz Aguirre
- Correo: bvaldesa@tec.mx
- Oficina: Parque Tecnológico, Piso 8.

- Horario de asesoría: `https://qrgo.page.link/M6HNX`

- Manuel Casillas
- Correo: manuel_casillas@tec.mx
- Horario de asesoría: Previa cita.

Objetivos generales

Al terminar la unidad de formación el alumno:

- Implementa algoritmos computacionales que solucionan problemas.
- Optimiza algoritmos computacionales que se aplican en el desarrollo de soluciones.
- Genera modelos computacionales para la solución de problemas.
- Implementa modelos computacionales en la solución numérica de un problema.

Metodología

- Este bloque se desarrolla en un periodo de 5 semanas, a lo largo de él se presentan módulos con contenidos y actividades de aprendizaje, con el fin de que cada estudiante construya los contenidos conceptuales, procedimentales y actitudinales que le permitan proponer una solución al reto de aplicación tecnológica planteado.
- La organización de las etapas permite al estudiante ir construyendo un pensamiento crítico y creativo que vincule fundamentos teóricos, matemáticos y herramientas tecnológicas para diseñar un prototipo de un sistema de detección de similitud de código.
- Para ello, nos apoyaremos en las metodologías de Aprendizaje Basados en Retos y Aula Invertida que te permitirán la resolución de un problema real mediante acciones concretas.

- Los contenidos del bloque se presentan en módulos que básicamente se refieren a la fundamentación y aplicación de cuatro grandes temas: metodología de investigación, inteligencia artificial, compiladores y métodos cuantitativos.

Para el desarrollo de solución al reto, se consideran períodos de inmersión total:

- 8 horas en las semanas 1 y 2 para el módulo 1.
- 22 horas en las semanas 1, 2, 3 y 4 para el módulo 2.
- 22 horas en las semanas 1, 2, 3 y 4 para el módulo 3.
- 20 horas en las semanas 1, 2, 3 y 4 para el módulo 4.
- 48 horas en la Semana 5. Para finalizar la integración, documentación y presentación de la solución propuesta al reto en un prototipo de software.

<table>
<tr><th>Cronograma</th><th>Semana 1</th><th>Semana 2</th><th>Semana 3</th><th>Semana 4</th><th>Semana 5</th></tr>
<tr><td rowspan="2">4</td><td>M1</td><td>M1</td><td>M1</td><td>M2</td><td rowspan="10">RETO</td></tr>
<tr><td>M1</td><td>M2</td><td>M2</td><td>M2</td></tr>
<tr><td rowspan="2">8</td><td>M2</td><td>M2</td><td>M2</td><td>M2</td></tr>
<tr><td>M2</td><td>M3</td><td>M2</td><td>M2</td></tr>
<tr><td>12</td><td colspan="4" rowspan="2">RETO</td></tr>
<tr><td>16</td></tr>
<tr><td></td><td>M3</td><td>M3</td><td>M3</td><td>M3</td></tr>
<tr><td>20</td><td>M3</td><td>M3</td><td>M3</td><td>M3</td></tr>
<tr><td rowspan="2"></td><td>M4</td><td>M4</td><td>M3</td><td>M3</td></tr>
<tr><td rowspan="2">24</td><td>M4</td><td>M4</td><td>M4</td><td>M4</td></tr>
<tr><td>M4</td><td>M4</td><td>M4</td><td>M4</td></tr>
</table>

Actividades	Ponderación
Actividades del Módulo 1	10 %
Actividades del Módulo 2	10 %
Actividades del Módulo 3	10 %
Actividades del Módulo 4	10 %
Presentación final del modelo ML mejorado	10 %
E1. Análisis de similitud empleando Inteligencia Artificial	20 %
E2. Reflexión sobre las ventajas/desventajas del modelo mejorado.	30 %

Descripción del problema

- El desarrollo de software mediante la copia ilegal del trabajo de otra persona o de código fuente abierto, y luego disfrazarlo de software original, se denomina plagio de software. Los piratas informáticos copian la lógica del software orinal en la versión pirateada. Esto representa una gran amenaza para una industria del software que aumenta rápidamente cada año y genera una gran pérdida económica para las empresas de software.
- La piratería del software es una violación de los derechos de autor del software original debido a que los piratas informáticos crean y venden la versión modificada. Emplean ingeniería inversa para acceder al código fuente del software original. De acuerdo con el informa de software mundial realizado por Business Software Alliance (BSA) en 2016, la tasa de piratería de software es del 39 %, lo que genera daños financieros de alrededor \$52,200 millones de dólares. Por tal motivo, los medio de detección de plagio son necesarios para superar tales pérdidas económicas.

- Varios estudios han demostrado que cada software contiene código plagiado en el rango del 5 % al 20 % y este tipo de herramientas son difíciles de mantener. Se han desarrollado diferentes herramientas de detección de plagio, por ejemplo, detección de similitud de código fuente, soluciones de errores de software, detección de clones y marcas de nacimiento de software, que se centran, principalmente, en el análisis basado en texto y estructura.

Reto

El objetivo de esta unidad de formación será desarrollar una propuesta para la detección del plagio de software. Para ello, será necesario que desarrollen:

1. Un modelo computacional que empleando herramientas del área de Compiladores, Método Cuantitativos y Aprendizaje Automatizado sea capaz de detectar el plagio de software en códigos desarrollados en Python.
2. Así como, un marco de referencia que permita evaluar el correcto funcionamiento del modelo propuesto.

Forma de trabajo:

- Equipos de tres personas.