

Universidad Tecnológica de la Mixteca

Clave DGP: 200089

Ingeniería en Computación

PROGRAMA DE ESTUDIOS

NOMBRE DE LA ASIGNATURA

Programación funcional

SEMESTRE	CLAVE DE LA ASIGNATURA	TOTAL DE HORAS
Séptimo	025075	85

OBJETIVO(S) GENERAL(ES)DE LA ASIGNATURA

Los objetivos de esta materia son a) que el estudiante conozca un paradigma de programación basado en la noción matemática de función, y b), aprender a aplicar este tipo de paradigma en problemas prácticos.

TEMAS Y SUBTEMAS

- Introducción
 - 1.1.El paradigma funcional.
 - 1.2. Características de la programación funcional.
 - 1.3.Datos inmutables como parte del paradigma.
 - 1.4. Ventajas del paradigma funcional.
 - 1.5.Estado del arte sobre lenguajes funcionales y sus usos.
- 2.Fundamentos matemáticos
 - 2.1. Conjuntos y operaciones: unión, complemento, conjunto potencia.
 - 2.2.Producto cartesiano, proyecciones.
 - 2.3.Definición descriptiva de conjuntos.
 - 2.4. Funciones matemáticas: composición e inversión.
 - 2.5. Nociones de computabilidad.
 - 2.6.Cálculo lambda.
 - 2.7. Teoría de categorías.
- 3. Fundamentos de la programación funcional.
 - 3.1.Declaraciones.
 - 3.2.Referencias.
 - 3.3. Expresiones e instrucciones.
 - 3.4. Variables locales.
- 4. Memoria y tipos de datos.
 - 4.1. Caracteres, números, booleanos.
 - 4.2.Cadenas de texto, listas: cabeza y cola.
 - 4.3. Colecciones (listas, vectores, arreglos, tuplas y diccionarios).
 - 4.4.Definición de nuevos tipos de datos.
 - 4.5. Sinónimos y datos enumerados.
 - 4.6.Definiciones recursivas de datos.

5. Funciones.

- 5.1.Sintaxis.
- 5.2. Funciones como valor de primera clase.
- 5.3.Composición de funciones e inversión.
- 5.4. Funciones recursivas.
- 5.5. Funciones Lambda (anónimas).

- 6.Funciones de orden superior y polimorfismo.
 - 6.1.Parcialización.
 - 6.2. Secciones. 6.3.Composición de funciones.
 - 6.4. Polimorfismos en listas y tuplas.
- 7.Estrategias de evaluación. 7.1. Evaluación perezosa.
 - 7.2. Tipos de datos infinitos.
 - 7.3. Evaluación rápida.
- 7.4. Mutabilidad en la programación funcional.

ACTIVIDADES DE APRENDIZAJE

- Realización de prácticas para fortalecer las técnicas especificadas en el temario.
- Investigación sobre técnicas no incluidas en este temario o sobre lenguajes funcionales/multiparadigma no cubiertos en clase.
- Realización de casos de estudio o proyectos. Por ejemplo, depuradores, intérpretes, editores de texto, aplicaciones web, videojuegos, graficación 3D, entre otros.

CRITERIOS Y PROCEDIMIENTOS DE EVALUACION Y ACREDITACIÓN

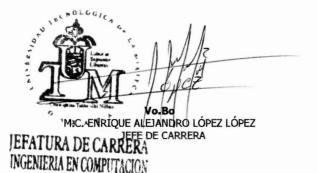
Para aprobar el curso se realizarán tres evaluaciones parciales (50 %) y una evaluación final (50%). Para cada evaluación se realizará un examen y se evaluarán tareas y proyectos. El examen tendrá un valor mínimo de 50% y las tareas y provectos un valor máximo de 50%. Adicionalmente se recomienda: Los proyectos prácticos serán implementados en un lenguaje funcional o multiparadigma pero siguiendo las técnicas de la programación funcional.

BIBLIOGRAFÍA (TIPO, TITULO, AUTOR, EDITORIAL Y AÑO)

- Básica:
 - Functional Programming using Haskell. Bird, R. Prentice Hall, 2ed. 1998. 1.
 - How To Design Programs. Felleisen, M. et al. MIT Press. 2002. 2. 3. The Joy of Clojure, Fogus, M., & Houser, C. Manning, 2ed. 2014.
- Consulta:
 - Structure and interpretation of computer programs. Abelson, H. et al. MIT Press, 2ed. 1996. 1.
 - The craft of functional programming, Thompson, S. Addison-Wesley, 3ed. 2011. 2. Functional Programming in Scala. Chiusano, P., & Bjarnason, R. Manning. 2013. 3.
 - Erlang programming. Cesarini, F, & Thompson, S. O'Reilly Media. 2009.

PERFIL PROFESIONAL DEL DOCENTE

Maestro o Doctor en Ciencias de la Computación o su equivalente en Ingeniería en Computación, de preferencia con experiencia laboral o de investigación en la Programación Funcional.



AUTORIZÓ
DR. AGUSTIN SANTIAGO ALYARADO
VICE-RECTOR ACADÉMICO
Chor yeard Telab vi

VICE-RECTORIA ACADÉMICA