



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Lenguajes de Programación

Felipe Restrepo Calle

ferestrepoca@unal.edu.co

Departamento de Ingeniería de Sistemas e Industrial
Facultad de Ingeniería
Universidad Nacional de Colombia
Sede Bogotá

Felipe Restrepo Calle
Profesor Asociado

Sitio web: <http://dis.unal.edu.co/~ferestrepoca/>
Correo: ferestrepoca@unal.edu.co

Oficina: 114 Edificio 453 (Aulas de Ingeniería)
Horario de atención: mediante cita previa

Formación:

- 2004 **Ingeniero de Sistemas y Computación**
UNIVERSIDAD TECNOLÓGICA DE PEREIRA, Pereira, Colombia
- 2007 **Magíster en Instrumentación Física**
UNIVERSIDAD TECNOLÓGICA DE PEREIRA, Pereira, Colombia
- 2011 **Doctor por la Universidad de Alicante**
Tecnologías para la sociedad de la información
UNIVERSIDAD DE ALICANTE, Alicante, España
- 2012-13 **Postdoctorado Universidad de Sevilla**
Sevilla, España
- 2014 **Postdoctorado Universidad Federal Rio Grande do Sul (UFRGS)**
Porto Alegre, Brasil

Experiencia:

- 2004-2013 **Co-fundador**
DUTO S.A., Pereira, Colombia
- 2005 **Profesor Asistente**
UNIVERSIDAD TECNOLÓGICA DE PEREIRA, Pereira, Colombia
- 2005-2007 **Ingeniero de Sistemas**
Empresa de Acueducto y Alcantarillado de Pereira E.S.P., Pereira, Colombia
- 2007 **Analista Técnico**
Empresa de Telecomunicaciones de Pereira E.S.P., Pereira, Colombia
- 2012 **Profesor Master en Tecnologías de la Informática**
Universidad de Alicante, Alicante, España
- 2012-2013 **Investigador Postdoctoral**
Universidad de Sevilla, Sevilla, España
- 2014-1 **Investigador Postdoctoral**
Universidad Federal Rio Grande do Sul (UFRGS), Porto Alegre, Brasil
- 2014-2- Hoy **Profesor Asociado**
Universidad Nacional de Colombia, Bogotá, Colombia

Grupo: [PLaS - Programming Languages and Systems](http://plas.unal.edu.co/) <http://plas.unal.edu.co/>

Objetivo

Investigar acerca de los aspectos prácticos y teóricos del diseño e implementación de lenguajes de programación, y su integración con sistemas software y hardware.

Líneas de investigación

1. Lenguajes de Programación

- Lenguajes de programación de propósito general y dominio específico (DSL)
- Lenguajes para soportar el ciclo de vida de desarrollo de SW (metalenguajes, modelado, espec.)
- Análisis y manipulación de código fuente

2. Educación en programación de computadores

- Realimentación automática a estudiantes y profesores
- Sistemas de calificación automática
- Entornos didácticos de programación

3. Sistemas

- Diseño de sistemas embebidos tolerantes a fallos (técnicas, herramientas y métricas)
- Computación aproximada
- Sistemas inteligentes de transporte – ITS
- Agricultura de precisión

Lenguajes de Programación

<https://sites.google.com/unal.edu.co/lenguajesdeprogramacion-2019-1/>

Código: 2025966

Créditos: 3

Pre-requisitos: Introducción a la Teoría de la Computación
Estructuras de Datos

Descripción

El objeto de estudio de esta asignatura es conocer la estructura y esencia de los lenguajes de programación y, a partir de ahí, abordar el diseño y construcción de un lenguaje de programación.

Para el desarrollo satisfactorio de esta asignatura es importante que el estudiante tenga sólidos conocimientos de lógica de programación y matemáticas discretas; además, debe haber utilizado más de un lenguaje de programación.

Objetivos

General

Formar en el estudiante la capacidad de analizar, diseñar e implementar un lenguaje de programación en el marco de un sistema automático de procesamiento de lenguaje.

Específicos:

1. Aplicar los conceptos fundamentales de la teoría de la computación en el diseño de lenguajes de programación.
2. Conocer, desde una perspectiva histórica, el desarrollo de los diferentes lenguajes de programación y de los paradigmas de programación.
3. Determinar la estructura y el funcionamiento de un analizador léxico, sintáctico y semántico.
4. Aplicar metodologías para el desarrollo de analizadores léxicos, sintácticos y semánticos mediante el uso de herramientas de software.
5. Adquirir habilidades para desarrollar herramientas computacionales de procesamiento de lenguajes de programación como compiladores, traductores e intérpretes en diversas áreas de conocimiento.

Metodología

- Conferencias del profesor
- Laboratorios y trabajos prácticos
- Presentaciones de los estudiantes
- Proyecto final

Durante las sesiones se incluirán diferentes actividades de **aprendizaje activo** (lluvia de ideas, trabajo en parejas/equipos, videos, pruebas de concepto, programación en equipos, presentación de imágenes, explicación de textos, etc.). De esta forma, se pretende proveer una experiencia de aprendizaje variada, potencializando así, los diferentes estilos de aprendizaje de los estudiantes.

Contenido (1/3)

1. Introducción a los lenguajes

- Sistema de procesamiento del lenguaje
- Estructura de un lenguaje
- Intérpretes vs. Traductores, Máquinas virtuales
- Arquitecturas u organizaciones computacionales
- Rol de los lenguajes de programación

2. Análisis léxico

- Alfabeto, lexemas y tokens.
- Lenguajes regulares, expresiones regulares y reconocimiento de tokens.
- Generadores de analizadores léxicos.
- Autómatas finitos.
- Construcción de analizadores léxicos.

Contenido (2/3)

3. Análisis sintáctico

Conceptos de análisis sintáctico.

Jerarquía de gramáticas de Chomsky.

Lenguajes libres de contexto y gramáticas libres de contexto.

Diseño de gramáticas: ambigüedad, asociatividad y precedencia de operadores.

Análisis sintáctico ascendente y descendente.

Generadores de analizadores sintácticos.

Construcción de analizadores sintácticos.

4. Traducción dirigida por sintaxis

Gramáticas de atributos.

Traducción dirigida por sintaxis, definición dirigida por sintaxis (DDS).

Esquemas de traducción dirigida por sintaxis (ETDS).

Generación de código.

Contenido (3/3)

5. Paradigmas de programación

- Conceptos de paradigmas de programación
- Programación Imperativa/procedimental
- Programación Orientada a Objetos
- Programación Funcional
- Programación Lógica
- Programación Concurrente
- Programación Paralela

6. Retos en lenguajes de programación

Evaluación

Primer parcial: 20%

Segundo parcial: 20%

Prácticas: 25%

Exposición: 15%

Proyecto final: 20%

Recursos recomendados

- **MOOCs (Massive Open Online Courses)**

- Curso Udacity: Programming Languages

<https://www.udacity.com/course/programming-languages--cs262>

- Curso Coursera: Functional Programming Principles in Scala

<https://www.coursera.org/course/progfun>

- **Blog: Federico Tomassetti**

<https://tomassetti.me/>

Bibliografía

1. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman. **Compiladores : principios, técnicas y herramientas**. Addison Wesley Longman, 2007. 2nd edition.
2. Loudon, Kenneth C. **Construcción de compiladores: principios y práctica**. Thomson, 2004. ISBN: 970-686-299-4.
3. Parr Terence. **The Definitive ANTLR4 Reference**. Pragmatic bookshelf. 2012.
4. Pat Terry. **Compiling with C# and Java**. Pearson Education, 2005.
5. John C. Mitchell. **Concepts in programming languages**. Cambridge University Press, 2003. ISBN: 0-521-78098-5.
6. John R. Levine, Tony Mason, Doug Brown. **Lex & yacc**. O`Reilly, 1995.
7. Parr, Terence. **The definitive ANTLR reference : building domain-specific languages**. Pragmatic Bookshelf, 2007. ISBN: 978-09787392-4-9.
8. Loudon, Kenneth , **Programming Languages: Principles and Practice**, Second Edition, PWS Publishing, 2002.
9. Scott, Michael, **Programming Language Pragmatics**, 2nd Ed, Ac. Press, 2005.
10. Watt, David, **Programming Language Design Concepts**, Prentice Hall, 2004.

Listado

Sitio web

<https://sites.google.com/unal.edu.co/lenguajesdeprogramacion-2019-1/>