

Proyecto Final Compiladores

Descripción General

La idea del proyecto final es identificar una aplicación que sea interesante de compiladores, estudiarla, implementarla y tratar de extenderla o incluirle algo nuevo que sea interesante.

El proyecto puede ser la implementación de alguna parte del curso de compiladores que les haya gustado más o que quieran ver como se hace (lexer, parser, calculadora de first-follow sets, análisis semántico) o utilizar una implementación existente.

Se puede tomar una implementación, pero se debe implementar o reescribir su código con el fin de entenderlo y para obtener una calificación perfecta, tratar de implementar algo nuevo (una mejora de diseño, una extensión, una modificación que lo haga más interesante, un operador más, etc...).

El proyecto se puede realizar en grupos de máximo 4 personas.

Entregables:

Para el proyecto se debe entregar:

- Código fuente de la solución.
- Video de 5 minutos donde se muestre:
 - Ejecución de la aplicación
 - Detalles principales de diseño
 - Extensión o cambio realizado a la implementación existente
 - Referencias de la aplicación existente.

Registro de proyectos: Se debe realizar un registro del proyecto (título, resumen, integrantes) a para el día 4 de diciembre de 2025 en el enlace: <https://forms.gle/VVKNButCxdFy7p228>

Evaluación: El martes 9 de diciembre en el horario de clase se realizará la reproducción de los videos. Al finalizar cada video, se tendrá una sección de preguntas de 5 minutos por parte de los estudiantes y del profesor.

La rúbrica de evaluación es la siguiente:

Rubrica de evaluación proyecto final Compiladores			
Item	porcentaje	Nota	Explicación item
Video	30%		El video explica bien la solución y muestra dominio sobre la aplicación de compiladores
Preguntas	30%		Muestra dominio del tema y responden adecuadamente las preguntas
Extensión	20%		Extiende o implementa algo adicional a la aplicación existente.
Referencias	20%		Referencia adecuadamente en el video y en el código los recursos utilizados
Total	100%		

Proyectos en 2024-2

Estos son algunos proyectos presentados en el curso durante 2024-2

Título del proyecto	Resumen del Proyecto	Autores	Video
Juego de aventuras de texto en Python	Crear un intérprete que permita a los usuarios interactuar con un juego de aventuras de texto mediante comandos simples. El intérprete debe ser capaz de entender y ejecutar acciones como mover al personaje, interactuar con objetos, etc.	Daniel Esteban Tobar Lozano Maria Jose Jara Herrera Carlos Ivan Muñoz Villazon Valeria Mora Serrano	https://drive.google.com/file/d/1qjeo-SjnJ3wtMXKnCgZDAVNAAdAArQmIP/view?usp=sharing https://github.com/camunozv/CakeDelivery
Plugin de GCC para para la detección de vulnerabilidades	Es un plugin para GCC que combina análisis sintáctico y semántico para identificar y prevenir vulnerabilidades relacionadas con el manejo inseguro de buffers. Inicialmente, el plugin recorre el árbol de sintaxis abstracta (AST) para detectar llamadas a funciones consideradas peligrosas, como strcpy o gets. Posteriormente, extiende el análisis evaluando el contexto semántico: verifica el tamaño de los buffers, analiza el flujo de datos y comprueba la existencia de validaciones antes de la ejecución de operaciones críticas. Este enfoque integral permite no solo	David Alejandro Sepulveda Ama ya Diego Felipe Cabrejo Suarez Daniel Santiago Cocinero Jiménez	https://drive.google.com/file/d/1MIJysMQSuBRxYyYcTVKWCFX-z7lipqTE/view?usp=sharing https://colab.research.google.com/drive/169WcRNHc1sQEeqIJD3Vn0p_KDjfJyoD1?usp=sharing

	<p>alertar sobre posibles riesgos, sino también ofrecer recomendaciones prácticas para mejorar la seguridad del código, reduciendo falsos positivos y orientando al desarrollador hacia mejores prácticas de programación. Este se configura para integrarse en el proceso de compilación, convirtiéndose en una herramienta útil para la detección temprana de vulnerabilidades en proyectos de software.</p>		
Cool Wasm	Implementacion del lenguaje cool en webassembly	Camilo Andres Cuello Romero Sergio Sanchez Moreno	https://drive.google.com/drive/folders/1OBPEMFqK5ec-ThKsBKQ6XI1cCrKX59RB?usp=sharing https://github.com/sechmo/cool-wasm
Generador de Autómatas Determinísticos	Se busca crear una página web en la cual, al ingresar una gramática libre de contexto, se cree, de forma gráfica, el autómata determinístico con sus ítems y transiciones correspondientes.	Julio César Bedoya Gutierrez, Juan Andrés Bueno Ramírez, Nicolás Alejandro Belli Pinzón, Juliette Sofía Lizarazo Ardila	https://drive.google.com/file/d/1kqmb6GqhRWCU8HIDeFwvIo05-fZMks1i/view?usp=sharing https://github.com/jubuenor/compiler-project
Calculadora de First-Follow Sets	Calculadora de First y Follow, permitiendo ingresar una gramática y obtener los conjuntos First y Follow con 1 lookahead.	Danny Marcelo Yaluzan Acosta Obed Felipe Espinosa Angarita Andrea Katherine Calderon Garcia David Steven Rodriguez Guzman	https://drive.google.com/file/d/1wl8vWARV93neX4qXW8DpmEGUZlxJaIDd/view?usp=sharing https://colab.research.google.com/drive/1p4BGnNGY60pD5MioKhpgoDTZ6BHWvYbw?usp=sharing

Parser de Python a JavaScript	El proyecto consiste en realizar un parser sencillo de python a JavaScript, es decir, transforma código python en código JavaScript.	Martin Alonso Gomez Uribe, Ronald Daniel Jacanamejoy Mutumbajoy	https://drive.google.com/file/d/14_fXWI4psi8Wbbm8TjbYCF65Zp-6TqVv/view?usp=sharing https://colab.research.google.com/drive/1vJMtHgV5yrMIGUb9PwnYrHUfjjaDS7FO?usp=sharing
Translating JSON to XML	Se va a implementar un traductor de JSON a XML y viceversa. La idea es recibir una entrada de JSON o XML y traducirlo a su equivalente.	Grevy Joner Rincon Mejia	https://drive.google.com/file/d/1KEJK202FxxQQ2C4ANL_iRMhLOUZSi6bla/view?usp=sharing https://drive.google.com/file/d/19vk3TVDYp23XolArB_4jxKDBd53ZbFPd/view
jl2wasm	<p>Propongo implementar un compilador de un pequeño subset de Julia a WebAssembly. El compilador se implementará en Julia como un programa literario escrito en el entorno de cuadernos computacionales Pluto.jl.</p> <p>El subset de Julia incluye funciones recursivas, condicionales y operadores aritméticos. Además tendrá un sistema de tipos estático con tipos booleanos, enteros y números de coma flotante.</p> <p>Este proyecto está basado en un pequeño compilador de Scheme a WebAssembly demostrado por Andy Wingo en la presentación Compiling To Web Assembly (FOSDEM 2021).</p>	Sergio Alejandro Vargas Quintero	https://youtu.be/uhugO18NhRM https://github.com/savq/jl2wasm
Traductor de Expresiones LaTeX a Python (LaPyX)	Este proyecto traduce ecuaciones matemáticas escritas en LaTeX a código Python utilizando la librería ANTLR. Para lograrlo, primero se define una gramática ANTLR que reconoce	Gabriel Santiago Delgado Lozano Fabio Esteban Murcia Martínez Marcos Pinzón	https://youtu.be/nn0sUj-DX88

	<p>expresiones como fracciones ($\frac{a}{b}$), raíces cuadradas (\sqrt{x}) y sumatorias ($\sum_{i=1}^n i^2$). Luego, se genera un analizador sintáctico en Python con ANTLR y se implementa un visitor que recorre el árbol de análisis para convertir las expresiones a código Python equivalente, como a / b, $\text{math.sqrt}(x)$, y $\text{sum}(i**2 \text{ for } i \text{ in range}(1, n+1))$. Finalmente, el traductor procesa ecuaciones LaTeX y genera código ejecutable en Python de forma automática.</p>	<p>Pardo Oscar Julián Rodríguez Cárdenas</p>	<p>https://github.com/gdelgadol/CompilersProject/tree/main</p>
--	---	--	--

Proyectos en 2023-2

Estos son algunos proyectos presentados en el curso durante 2023-2

Proyecto	Integrantes	Video	Enlace Fuentes
Parser JSON a Java	Julián Steven Vega Daza, Santiago Rodríguez Vallejo, Johan Sebastián Lemus Pedraza	https://youtu.be/W--JUcNLTJc	https://github.com/sarodriguezva/JsonToJavaConverter/tree/main
SWLang	Sebastián Daniel Moreno Martínez	https://drive.google.com/file/d/1uC2rQY3PI2fCunPmIK-H5rJUpU26Pib4/view?usp=sharing	https://drive.google.com/file/d/1N2SO_hVuAzQqx5yiyCRV9w70DZDhuA_0/view?usp=sharing
Graphex	Juan Carlos Garavito Higuera, David Santiago Cruz Hernandez, Omar David Toledo Leguizamón,	https://youtu.be/4x_F31PPTD8	https://colab.research.google.com/drive/1xd-AVxiUEKbmexETEChtp0MXJNobUKqw?usp=sharing

	Santiago Restrepo Rojas		
Traductor LPP a C++	Anderson Andrés Llanos Quintero, Juan Diego Lozano Colmenares, Luis Alfonso Díaz Vergel, Pedro Grisales	https://youtu.be/6cLy-b IQyY	https://drive.google.com/file/d/1nICepEPWbRJku8awShjozhBbMJNQJz6q/view?usp=sharing
Nuevas acciones para una tortuga con código generado en Unity -	Daniel Felipe Ahumada Hernández, Cristian David Santos Gil, Andrés Felipe Patio Camargo, Joseph Nicolay Sarmiento Molano	https://youtu.be/5q2uqWy52to	https://github.com/sadipvp/Compilador_CSharp/tree/main
Pequeño DBMS en Español	David Alarcón, César Lemos	https://youtu.be/0kPMBqKLjnQ?si=T-PI2jmCEuf-XGB	https://github.com/cesarALS/Proyecto_DB_Compiladores.git

Algunas ideas adicionales de proyectos

Aunque el grupo pueda seleccionar cualquier proyecto que deseen con el fin de ayudar (quizás) a determinar qué proyectos podrían ser, se proponen aquí algunas ideas de proyecto:

1. **Un api pequeño para un lenguaje de consulta utilizando ANTLR en Python:**
<https://medium.com/@ab.rhmn97/build-an-api-query-language-with-antlr-in-python-7313dba222e7>

En este proyecto se diseña una API pequeña para realizar consultas a una estructura de tipo diccionario. Las consultas son del estilo:

User eq "john" and age eq 34

2. Una aplicación de compiladores para realizar juegos en Unity:

<https://github.com/jewagenaar/antlrExampleUnity>

Este es un proyecto de ejemplo que muestra cómo obtener un lenguaje de scripting creado con Antlr corriendo en Unity. También hay una pequeña escena de demostración (Assets/Scenes/TortoiseScene) que demuestra cómo se puede utilizar en un juego.

3. Pet projects are fun

<https://medium.com/securenative/pet-projects-are-fun-763ee39a732b>

Este es un proyecto que toma un código en go y actualiza sus dependencias.

4. Ejemplo web en ANTLR: un ejemplo simple

<https://www.javacodegeeks.com/2016/05/antlr-web-simple-example.html>

Incluye un ejemplo de listas de tareas y parsing de las mismas en javascript.

5. Creando un lenguaje de script con ANTLR:

<https://hackernoon.com/creating-a-scripting-language-with-antlr-part-1-1b42c3e4d718>

<https://medium.com/hackernoon/creating-a-scripting-language-with-antlr-part-2-ba4d8d6eab55>

<https://medium.com/@thosakwe/creating-a-scripting-language-with-antlr-part-2-e41fc68c60ba>

Una explicación paso a paso de cómo crear un lenguaje sencillo que traduce a JavaScript.

6. Escribiendo tu propio lenguaje de programación en Python:

<https://medium.com/@marcelogdeandrade/writing-your-own-programming-language-and-compiler-with-python-a468970ae6df>

Es una guía paso a paso para crear un pequeño lenguaje con Python

7. Aplicaciones prácticas de compiladores en ANTLR (cap 8) del libro:

<https://dl.icdst.org/pdfs/files3/a91ace57a8c4c8cdd9f1663e1051bf93.pdf> (Este libro se encuentra también en biblioteca)