

LABORATORIO ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 2 - SECCIÓN B

CÓDIGO:	781	PUNTEO NETO LABORATORIO:	X
ESCUELA DE INGENIERÍA EN:	CIENCIAS Y SISTEMAS	ÁREA A LA QUE PERTENECE:	CIENCIAS DE LA COMPUTACIÓN
PRE REQUISITO:	777 – Organización de lenguajes y compiladores 1 772 – Estructuras de datos	POST REQUISITO:	281 – Sistemas operativos 1 972 - Inteligencia Artificial 1
CATEGORÍA:	OBLIGATORIO	VIGENCIA:	PRIMER SEMESTRE 2026

Descripción del Laboratorio

El laboratorio de Organización de Lenguajes y Compiladores 2, adscrito al área de Ciencias de la Computación, se fundamenta en los conceptos teóricos abordados en la cátedra y avanza hacia la implementación práctica de un intérprete y un compilador, aplicando de manera sistemática las distintas etapas necesarias para su desarrollo adecuado.

A lo largo del trabajo en laboratorio se detalla el proceso de construcción, que integra teoría, análisis, diseño, eficiencia e implementación. Asimismo, se desarrollan prácticas relacionadas con temas como la traducción dirigida por la sintaxis Incluyendo analizadores ascendentes y descendentes, así como la generación de traducciones durante el análisis sintáctico, el manejo de tablas de símbolos y sistemas de tipos con verificación estática y dinámica, la generación de código intermedio de tres direcciones para expresiones aritméticas y relacionales, conversiones, estructuras de control iterativas, secuencias de escape y arreglos multidimensionales. También se abordan los entornos de ejecución, considerando ámbitos globales y locales, el paso de parámetros y la gestión de temporales dentro del registro de activación. Estas competencias trascienden el campo específico de los compiladores y resultan aplicables en áreas como la ingeniería de software y el diseño de sistemas. Para el desarrollo de las actividades prácticas se emplearán herramientas destinadas a la generación automática de analizadores léxicos y sintácticos.

Resumen de Ponderaciones y Tiempo de Auto-aprendizaje

TIPO	PONDERACIÓN	HORAS DE AUTO-APRENDIZAJE
Proyectos	95	80
Prácticas	0	4
Tareas	5	33
TOTAL	100	117

Equipo Académico

Coordinador del Área

Nombre: M.Sc. Luis Fernando Espino Barrios	Correo electrónico: usac.sistemas@gmail.com
---	--

Docente

LUIS FERNANDO ESPINO BARRIOS	luisespino@yahoo.com
------------------------------	----------------------

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Día		x				
Horario		07:10 - 10:30				
Lugar		meet				

Tutor(es)

Nombre del Tutor	Rubén Alejandro Ralda Mejía	
Correo electrónico institucional	3629952490101@ingenieria.usac.edu.gt	

Tipo		Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Clase	Día						
	Horario						
	Lugar						
Atención al Estudiante	Día						
	Horario						
	Lugar						

Índice

Descripción del Laboratorio.....	1
Resumen de Ponderaciones y Tiempo de Auto-aprendizaje.....	1
Equipo Académico.....	2
Coordinador del Área.....	2
Docente.....	2
Tutor(es).....	2
Competencias Vinculadas al Perfil del Egresado.....	4
Competencias Específicas.....	4
Competencias Generales.....	4
Competencias del Laboratorio.....	4
Competencia(s) Específica(s).....	4
Competencia(s) General(es).....	5
Diseño Didáctico.....	5
Sesión de Diagnóstico.....	5
Sesión No. 1, Unidad No. 1 - Fases de la Compilación.....	6
Sesión No. 2, Unidad No. 1 - Interpretación de instrucciones básicas.....	7
Sesión No. 3, Unidad No. 1 - Manejo de entornos dentro del intérprete.....	8
Sesión No. 4, Unidad No. 1 - Estructuras cíclicas y de control de flujo.....	8
Sesión No. 5, Unidad No. 2 - Funciones y cerraduras en un intérprete.....	9
Sesión No. 6, Unidad No. 2 - Programación orientada a objetos en un intérprete.....	10
Sesión No. 7, Unidad No. 2 - Representaciones intermedias y generación de código, Tipos primitivos a bajo nivel.....	11
Sesión No. 8, Unidad No. 3 - Entornos, variables globales y locales en un compilador, Traducción de estructuras de control en un compilador.....	12
Sesión No. 9, Unidad No. 3 - Traducción de funciones de primera clase en un compilador, Programación orientada a objetos.....	13
Sesión No. 10, Unidad No. 3 - Programación orientada objetos en un compilador, Unidad No. 4 - Optimización de código.....	14
Sesión No. 11, Unidad No. 3 - Programación orientada a objetos en un compilador, Unidad No. 4 - Optimización de código, Recolección de basura.....	15
Rúbrica de Evaluación.....	16
Normativa Académica y Ética del Curso.....	16
Bibliografía.....	17
E-Grafía.....	17

Competencias Vinculadas al Perfil del Egresado

Competencias Específicas

No.	Competencia
1	Demuestra pensamiento crítico, actitud investigativa y rigor analítico en el planteamiento y la resolución de problemas complejos.
2	Toma decisiones profesionales con base en fundamentos teóricos, datos e información pertinente, válida y confiable.
3	Identifica oportunidades y riesgos para la innovación y adaptación de conocimientos y tecnologías para resolver problemas.

Competencias Generales

No.	Competencia
1	Aplica principios básicos de ingeniería, ciencias de computación y sistemas de información y comunicación, en la formulación y resolución adecuada de problemas complejos.
2	Aplica estándares de calidad, eficiencia y seguridad en la implementación adecuada de soluciones de software, hardware y TIC en general.
3	Aplica conocimientos tecnológicos con ética profesional, respetando y cuidando los recursos naturales, humanos y financieros.

Competencias del Laboratorio

Competencia(s) Específica(s)

No.	Competencia	Nivel de Aprendizaje
1	El estudiante evalúa el desempeño de cada fase del compilador a través de métricas de tiempo y uso de memoria para comparar implementaciones como GCC vs. Clang	Evaluar
2	El estudiante aplica técnicas de optimización de código como optimización local y global para reducir tamaño de código y mejorar rendimiento	Aplicar
3	El estudiante genera código de bajo nivel y ensamblador a partir de plantillas de generación y asignación de registros considerando convenciones de llamada y optimización de registros	Crear
4	El estudiante analiza representaciones intermedias de código mediante código de tres direcciones para identificar oportunidades de optimización	Analizar

5	El estudiante explica la eliminación de recursión por la izquierda con ejemplos paso a paso para preparar gramáticas para análisis sintáctico.	Comprender
---	--	------------

Competencia(s) General(es)

No.	Competencia	Nivel de Aprendizaje
1	El estudiante comprende las fases de un compilador a través de diagramas de flujo y modelos teóricos con ejemplos de compilación real en lenguajes como C, Go o Java.	Comprender
2	El estudiante utiliza herramientas generadoras de lexers y parsers tales como Flex, Bison y ANTLR en proyectos de análisis léxico y sintáctico	Aplicar

Diseño Didáctico

Sesión de Diagnóstico

Evaluación de conocimientos previos

Se aplicará una actividad diagnóstica con el objetivo de identificar el nivel de conocimientos y habilidades que los estudiantes poseen al inicio del curso. No influye en la nota final, pero es obligatoria para todos los estudiantes.

Tipo de Actividad	Descripción
Cuestionario	

Presentación del tutor

El tutor se presenta formalmente al grupo, compartiendo su formación académica, experiencia profesional y educativa, así como sus expectativas sobre el curso. También se abordan aspectos como normas de convivencia, canales de comunicación, disponibilidad para consultas y métodos de acompañamiento.

Presentación de los estudiantes

Se escogen un grupo de estudiantes al azar. En su presentación, se les pedirá que compartan información básica como su nombre, intereses personales o profesionales, experiencias previas relacionadas con el curso y sus expectativas. Esta actividad busca promover la interacción, el reconocimiento entre pares y la construcción de un entorno participativo y respetuoso.

Presentación del programa del curso

Se presenta el contenido del programa del curso, se aclaran dudas y se fomenta el compromiso del estudiante con su aprendizaje.

Evaluación de conocimientos del laboratorio actual

Se realiza una evaluación o práctica que permite conocer el grado de familiaridad de los estudiantes con las herramientas, entornos o competencias técnicas necesarias para el laboratorio actual.

Tipo de Actividad	Descripción
Cuestionario	

Sesión No. 1, Unidad No. 1 - Fases de la Compilación

Área Actitudinal (Saber ser)

Nombre del valor: Curiosidad académica
Comprender las motivaciones y aplicaciones de los compiladores despierta interés en el diseño de lenguajes de programación y herramientas de desarrollo.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante comprende las fases de un compilador a través de diagramas de flujo y modelos teóricos con ejemplos de compilación real en lenguajes como C, Go o Java.	
El estudiante explica la eliminación de recursión por la izquierda con ejemplos paso a paso para preparar gramáticas para análisis sintáctico	
Tema	Subtema
Fases de la compilación	Motivaciones para estudiar compiladores
Fases de la compilación	Recomendaciones para el curso
Fases de la compilación	¿Qué es un compilador?
Fases de la compilación	Análisis Léxico
Fases de la compilación	Análisis Sintáctico
Fases de la compilación	Análisis Semántico
Fases de la compilación	Compilar vs Interpretar vs Transpilar

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante explica la eliminación de recursión por la izquierda con ejemplos paso a paso para preparar gramáticas para análisis sintáctico	
Tipo de Actividad	Ponderación

Ejemplo práctico	0
------------------	---

Sesión No. 2, Unidad No. 1 - Interpretación de instrucciones básicas

Área Actitudinal (Saber ser)

Nombre del valor: Rigor Técnico
Ejecutar de manera rigurosa las reglas de interpretación y traducción de instrucciones, garantizando la obtención de resultados correctos.

Área de Conocimiento (Saber)

Competencia(s)	
Describir el funcionamiento mediante el cual un intérprete ejecuta instrucciones básicas, aplicando reglas de precedencia y acciones semánticas durante su procesamiento.	
Tema	Subtema
Interpretación de instrucciones básicas	Repaso de fases del compilador
Interpretación de instrucciones básicas	Precedencia de operadores
Interpretación de instrucciones básicas	Calculadora básica
Interpretación de instrucciones básicas	Acciones semánticas
Interpretación de instrucciones básicas	Tipos de analizadores sintácticos

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante describe y justifica el proceso de transformación de gramáticas con recursión por la izquierda, desarrollando ejemplos detallados paso a paso, con el propósito de adecuarlas para su análisis sintáctico.	
Tipo de Actividad	Ponderación
Ejemplo práctico	0

Sesión No. 3, Unidad No. 1 - Manejo de entornos dentro del intérprete

Área Actitudinal (Saber ser)

Nombre del valor: Organización y estructura
El manejo de entornos y tabla de símbolos requiere una organización sistemática para gestionar correctamente el alcance de variables y sus valores.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante comprende las fases de un compilador a través de diagramas de flujo y modelos teóricos con ejemplos de compilación real en lenguajes como C, Go o Java.	
Tema	Subtema
Manejo de entornos dentro del intérprete	Patrón Visitor
Manejo de entornos dentro del intérprete	Tabla de símbolos
Manejo de entornos dentro del intérprete	Parent Pointer Tree
Manejo de entornos dentro del intérprete	Declaraciones
Manejo de entornos dentro del intérprete	Asignaciones

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante comprende las fases de un compilador a través de diagramas de flujo y modelos teóricos con ejemplos de compilación real en lenguajes como C, Go o Java.	
Tipo de Actividad	Ponderación
Ejemplo práctico	0

Sesión No. 4, Unidad No. 1 - Estructuras cíclicas y de control de flujo

Área Actitudinal (Saber ser)

Nombre del valor: Responsabilidad computacional
Entender el poder de un lenguaje implica usarlo de forma ética y consciente.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante emplea herramientas de generación de analizadores léxicos y sintácticos, como Flex, Bison y ANTLR, en el desarrollo de proyectos de análisis léxico y sintáctico.	
Tema	Subtema
Estructuras cíclicas y de control de flujo	Lenguajes Turing completos
Estructuras cíclicas y de control de flujo	Estructuras de control
Estructuras cíclicas y de control de flujo	Ciclos
Estructuras cíclicas y de control de flujo	Sentencias de transferencia

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante hace uso de herramientas para la creación automática de analizadores léxicos y sintácticos, como Flex, Bison y ANTLR, en la realización de proyectos de análisis léxico y sintáctico.	
Tipo de Actividad	Ponderación
Ejercicio práctico	0

Sesión No. 5, Unidad No. 2 - Funciones y cerraduras en un intérprete

Área Actitudinal (Saber ser)

Nombre del valor: Ética del alcance
Mantener el uso de variables dentro de su contexto lógico y justo.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante evalúa el desempeño de cada fase del compilador a través de métricas de tiempo y uso de memoria para comparar implementaciones como GCC vs. Clang	
El estudiante utiliza herramientas generadoras de lexers y parsers tales como Flex, Bison y ANTLR en proyectos de análisis léxico y sintáctico	
Tema	Subtema
Funciones y cerraduras en un intérprete	Funciones embebidas
Funciones y cerraduras en un intérprete	Funciones foráneas
Funciones y cerraduras en un intérprete	Closures

Funciones y cerraduras en un intérprete	Resolución de variables
Funciones y cerraduras en un intérprete	Hoisting
Funciones y cerraduras en un intérprete	Call Stack

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante comprende las fases de un compilador a través de diagramas de flujo y modelos teóricos con ejemplos de compilación real en lenguajes como C, Go o Java.	
Tipo de Actividad	Ponderación
Ejemplo práctico	0

Sesión No. 6, Unidad No. 2 - Programación orientada a objetos en un intérprete

Área Actitudinal (Saber ser)

Nombre del valor: Precisión
Ya que al aplicar técnicas de optimización y análisis de código, se requiere atención a los detalles para lograr mejoras reales y evitar errores.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante identifica y explica las distintas etapas de un compilador mediante diagramas de flujo y modelos teóricos, apoyándose en ejemplos reales de compilación en lenguajes como C, Go o Java.	
Tema	Subtema
Programación orientada a objetos en un intérprete	Clases y Objetos
Programación orientada a objetos en un intérprete	Prototipos
Programación orientada a objetos en un intérprete	Constructores
Programación orientada a objetos en un intérprete	Sobrecarga de métodos

Área de Habilidades (Saber Hacer)

Competencia
El estudiante reconoce y describe las etapas que conforman un compilador utilizando diagramas de flujo y marcos teóricos, complementados con ejemplos prácticos de compilación en lenguajes como C, Go o Java.

Tipo de Actividad	Ponderación
Ejemplo práctico sobre el contenido impartido de la sesión, mejorando el código del ejemplo de la sesión anterior.	0

Sesión No. 7, Unidad No. 2 - Representaciones intermedias y generación de código, Tipos primitivos a bajo nivel

Área Actitudinal (Saber ser)

Nombre del valor: Rigor
Porque se trabaja con conceptos técnicos de bajo nivel, donde la precisión y estructura lógica son clave, especialmente en temas como ensamblador o representación en memoria.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante utiliza herramientas generadoras de lexers y parsers tales como Flex, Bison y ANTLR en proyectos de análisis léxico y sintáctico	
El estudiante comprende las fases de un compilador a través de diagramas de flujo y modelos teóricos con ejemplos de compilación real en lenguajes como C, Go o Java.	
Tema	Subtema
Representaciones intermedias y generación de código	Código de 3 direcciones
Representaciones intermedias y generación de código	Bytecode
Representaciones intermedias y generación de código	Assembly: RISC-V
Tipos primitivos a bajo nivel	Tipos enteros
Tipos primitivos a bajo nivel	Representación de punto flotante
Tipos primitivos a bajo nivel	Char
Tipos primitivos a bajo nivel	Strings

Área de Habilidades (Saber Hacer)

Competencia
El estudiante utiliza herramientas generadoras de lexers y parsers tales como Flex, Bison y ANTLR en

proyectos de análisis léxico y sintáctico	
Tipo de Actividad	Ponderación
Ejemplo práctico	0

Sesión No. 8, Unidad No. 3 - Entornos, variables globales y locales en un compilador, Traducción de estructuras de control en un compilador

Área Actitudinal (Saber ser)

Nombre del valor: Orden
La correcta organización de los entornos asegura un acceso coherente a variables globales y locales.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante produce código de bajo nivel y en lenguaje ensamblador utilizando esquemas de generación y asignación de registros, respetando las convenciones de llamada y aplicando criterios de optimización en el uso de los registros.	
Tema	Subtema
Entornos, variables globales y locales en un compilador	Tabla de símbolos
Entornos, variables globales y locales en un compilador	Manejo del stack para variables locales
Traducción de estructuras de control en un compilador	Resolución de variables
Traducción de estructuras de control en un compilador	Etiquetas
Traducción de estructuras de control en un compilador	Saltos condicionales, Saltos incondicionales

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante elabora código de bajo nivel y código ensamblador a partir de plantillas para la generación y asignación de registros, teniendo en cuenta las convenciones de llamada y la optimización del uso de los registros.	
Tipo de Actividad	Ponderación

Ejemplo práctico	0
------------------	---

Sesión No. 9, Unidad No. 3 - Traducción de funciones de primera clase en un compilador, Programación orientada a objetos

Área Actitudinal (Saber ser)

Nombre del valor: Rigor técnico
La traducción de funciones e instrucciones ocupa la mayor parte de complejidad en un compilador por lo que mejora su uso

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante genera código de bajo nivel y ensamblador a partir de plantillas de generación y asignación de registros considerando convenciones de llamada y optimización de registros	
Tema	Subtema
Traducción de funciones de primera clase en un compilador	Paso de argumentos
Traducción de funciones de primera clase en un compilador	Call Frames
Traducción de funciones de primera clase en un compilador	Valores de retorno
Programación orientada a objetos en un compilador	Propiedades

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante genera código de bajo nivel y ensamblador a partir de plantillas de generación y asignación de registros considerando convenciones de llamada y optimización de registros	
Tipo de Actividad	Ponderación
Ejercicio práctico	0

Sesión No. 10, Unidad No. 3 - Programación orientada objetos en un compilador, Unidad No. 4 - Optimización de código

Área Actitudinal (Saber ser)

Nombre del valor: Eficiencia
Buscar mejorar el rendimiento y el uso de recursos del código generado.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante desarrolla código de bajo nivel y en ensamblador empleando plantillas para la generación y asignación de registros, tomando en cuenta las convenciones de llamada y la optimización en el uso de dichos registros.	
Tema	Subtema
Programación orientada a objetos en un compilador	Métodos
Programación orientada a objetos en un compilador	Manejo del heap
Optimización de código	Eliminación de código muerto
Optimización de código	Propagación de constantes

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante emplea técnicas de optimización de código, tanto a nivel local como global, con el objetivo de disminuir el tamaño del código y mejorar su rendimiento.	
Tipo de Actividad	Ponderación
Ejercicio práctico	0

Sesión No. 11, Unidad No. 3 - Programación orientada a objetos en un compilador, Unidad No. 4 - Optimización de código, Recolección de basura

Área Actitudinal (Saber ser)

Nombre del valor: Precisión
La optimización es el paso que nos permite corregir aquello generado en el proceso de compilación lo que permite que el resultado final siempre sea óptimo

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante genera código de bajo nivel y ensamblador a partir de plantillas de generación y asignación de registros considerando convenciones de llamada y optimización de registros	
Tema	Subtema
Programación orientada a objetos en un compilador	Métodos
Programación orientada a objetos en un compilador	Manejo del heap
Optimización de código	Eliminación de código muerto
Optimización de código	Propagación de constantes
Optimización de código	Optimización de bucles
Optimización de código	Simplificación Algebraica
Optimización de código	Optimización de saltos condicionales
Recolección de basura	Garbage Collection
Recolección de basura	Memory Leaks

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante genera código de bajo nivel y ensamblador a partir de plantillas de generación y asignación de registros considerando convenciones de llamada y optimización de registros	
Tipo de Actividad	Ponderación
Ejercicio práctico	0

Rúbrica de Evaluación

Cada una de las actividades del laboratorio (proyectos, prácticas, tareas y otras) cuenta con una rúbrica de evaluación específica, la cual está detallada en el documento que se entrega al estudiante al momento de asignar la actividad. Estas rúbricas describen los criterios de evaluación, niveles de desempeño esperados y la ponderación correspondiente de cada aspecto evaluado.

Es **responsabilidad del estudiante** leer detenidamente la rúbrica asignada antes de iniciar el desarrollo de la actividad. Comprender los criterios de evaluación no solo permite orientar adecuadamente el trabajo, sino también mejorar el desempeño académico y fomentar la autorregulación del aprendizaje.

En caso de no recibir la rúbrica al momento de la asignación, el estudiante **debe solicitarla directamente al tutor académico**, ya que constituye una herramienta esencial para el cumplimiento de los objetivos de aprendizaje y la evaluación transparente.

Normativa Académica y Ética del Curso

En concordancia con el perfil del estudiante de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, se espera un alto nivel de compromiso con la excelencia académica y la ética profesional. Por ello, que se establece los siguientes lineamientos de carácter obligatorio que regulan el comportamiento académico del estudiante:

Plagio y copias

- Todo proyecto será sometido a verificación para confirmar su autoría y originalidad, con la finalidad de evitar cualquier plagio, copia o que la actividad no haya sido realizada por el estudiante.
- Cualquier evidencia de lo antes descrito en las distintas actividades será sancionada con una calificación de 0 (cero) y el caso será reportado al Docente quien a su vez informará a la Escuela de Ciencias y Sistemas para su seguimiento institucional.

Prórrogas y reposiciones

- No se otorgarán prórrogas para entregas de actividades.
- No se permitirá la reposición de proyectos bajo ninguna circunstancia.

Requisitos para evaluación final del curso

- Es obligatorio aprobar el laboratorio para tener derecho a la evaluación final del curso.
- La calificación de prácticas, proyectos y otras actividades que se indique será asignada de forma presencial, en la fecha y hora establecidas por el tutor académico.

Asistencia

- Para obtener la nota del laboratorio, se requiere un mínimo del 80% de asistencia a las sesiones de laboratorio.
- En caso de inasistencia, sólo se aceptarán justificaciones válidas respaldadas por constancia oficial.

Entregas

- No se aceptarán entregas tardías de tareas, prácticas, exámenes cortos, exámenes finales o proyectos sin justificación.

Medio oficial de entrega

- La plataforma UEDI de la Facultad será el único medio oficial para la entrega de actividades del curso.

Bibliografía

- Compiladores, Principios, Técnicas y Herramientas Aho, Sethi y Ullmam.
PEARSON ADDISON- WESLEY, 2008, segunda edición

E-Grafía

Carter, Z. (2009). Jison – Documentación. Disponible en: <https://zaa.ch/jison/docs/>

Klein, G., Rowe, S. y Décamps, R. (1998). JFlex – Manual. Disponible en: <https://jflex.de/manual.html>

Viswanadha, S. y Sankar, S. (s. f.). JavaCC – Documentación. Disponible en: <https://javacc.github.io/javacc>

CUP – Documentación. Disponible en: <http://www2.cs.tum.edu/projects/cup/docs.php>

PLY – Documentación. Disponible en: <https://www.dabeaz.com/ply/>

Python – Ply

- <https://ericknavarro.io/2020/02/10/24-Mi-primer-proyecto-utilizando-PLY/>
- <https://ericknavarro.io/2020/03/15/26-Interprete-sencillo-utilizando-PLY/>
- http://kunusoft.com/slides/compi2/manual_ply_python/
- http://kunusoft.com/slides/compi2/guia_heredados_pila/index.php?pic=0 o

Java – JavaCC

- <https://ericknavarro.io/2020/02/10/23-Mi-primer-proyecto-utilizando-JavaCC/>
- <https://ericknavarro.io/2020/03/07/25-Interprete-sencillo-utilizando-Javacc/> o <https://ericknavarro.io/2019/04/26/02-Mi-primer-proyecto-utilizando-Jlex-y-Cup-Windows/>
- [https://ericknavarro.io/2019/04/26/05-Interprete-sencillo-utilizando-Java-Jlex-y-Cup/Javascript/NodeJS – Jison](https://ericknavarro.io/2019/04/26/05-Interprete-sencillo-utilizando-Java-Jlex-y-Cup/Javascript/NodeJS-Jison)
- <https://ericknavarro.io/2019/07/21/17-Mi-primer-proyecto-utilizando-Jison>
- <https://ericknavarro.io/2019/08/01/20-Interprete-sencillo-utilizando-Jison-con-Nodejs-Ubuntu/>

C# – Irony

- <https://ericknavarro.io/2019/07/24/19-Mi-primer-proyecto-utilizando-Irony-Windows/>
- <https://ericknavarro.io/2019/08/07/22-Interprete-sencillo-utilizando-Irony-con-CS/>

Visual Basic – Gold Parser

- <https://ericknavarro.io/2019/07/22/18-Mi-Primer-Proyecto-Utilizando-GOLD-Parser-Windows/>
- <https://ericknavarro.io/2019/08/07/21-Interprete-sencillo-utilizando-GOLD-Parser-y-Visual-Basic>