

## PROGRAMA DE LABORATORIO

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS



### Organización de Lenguajes y Compiladores 2

CÓDIGO:	781	PONDERACIÓN:	6
ESCUELA DE INGENIERÍA EN:	CIENCIAS Y SISTEMAS	ÁREA A LA QUE PERTENECE:	DESARROLLO DE SOFTWARE
PRE REQUISITO:	772 – Estructuras de datos 777 – Organización de lenguajes y compiladores 1	POST REQUISITO:	281 – Sistemas operativos 1 972 – Inteligencia Artificial 1
CATEGORÍA:	OBLIGATORIO	VIGENCIA:	SEGUNDO SEMESTRE 2025
HORAS POR SEMANA DEL CURSO:	4	HORAS POR SEMANA DEL LABORATORIO:	2
HORAS DE AUTOAPRENDIZAJE:	2	TOTAL DE HORAS DE APRENDIZAJE:	8
CATEDRÁTICO (A):	Luis Fernando Espino Barrios	AUXILIAR:	Rubén Alejandro Ralda Mejía
EDIFICIO:	T3	SECCIÓN:	B
SALÓN DEL CURSO:	meet	SALON DEL LABORATORIO:	meet
DÍAS QUE SE IMPARTE EL CURSO:	Martes	DÍAS QUE SE IMPARTE EL LABORATORIO:	Martes
HORARIO DEL CURSO:	07:10 a 10:30	HORARIO DEL LABORATORIO:	17:20 a 19:00

### Breve descripción del Laboratorio

El laboratorio de organización de lenguaje y compiladores 2 perteneciente al área de ciencias de la computación, parte de una base teórica obtenida en la clase magistral y continua con la implementación de un intérprete o compilador aplicando distintas fases para su correcto desarrollo. Durante el desarrollo de laboratorio se explica el proceso para la construcción el cual consta de la teoría, análisis, diseño, eficiencia e implementación, se repasan conceptos de la clase aplicados de forma práctica como la traducción dirigida por la sintaxis, generación y optimización de código intermedio; que son útiles en muchos contextos más allá de compiladores los cuales pueden ser ingeniería de software y diseño de sistemas.

## Índice

<b>Competencias Vinculadas al Perfil del Egresado .....</b>	<b>4</b>
Competencias Específicas.....	4
Competencias Generales .....	4
<b>Competencias del Laboratorio .....</b>	<b>4</b>
Competencia(s) Específica(s) .....	4
Competencia(s) General(es).....	5
<b>Diseño Didáctico por Competencias .....</b>	<b>5</b>
Sesión de Diagnóstico.....	6
Evaluación de conocimientos previos .....	6
Presentación del tutor.....	6
Presentación de los estudiantes .....	6
Presentación del programa del curso .....	6
Evaluación de conocimientos del laboratorio actual.....	6
Sesión No. 1, Unidad No. 1 - La computadora , Unidad 2 - Arquitectura y organización VNA....	7
Valor de la semana (Saber ser).....	7
Conocimiento (Saber).....	7
Habilidades ( Saber Hacer) .....	7
Sesión No. 2, Unidad No. 2 - Arquitectura y organización VNA.....	8
Valor de la semana (Saber ser).....	8
Conocimiento (Saber).....	8
Habilidades ( Saber Hacer) .....	8
Sesión No. 3, Unidad No. 2 - Arquitectura y organización VNA.....	9
Valor de la semana (Saber ser).....	9
Conocimiento (Saber).....	9
Habilidades ( Saber Hacer) .....	9
Sesión No. 4, Unidad No. 4 - Codificación de la información.....	10
Valor de la semana (Saber ser).....	10
Conocimiento (Saber).....	10
Habilidades ( Saber Hacer) .....	10
Sesión No. 5, Unidad No. 4 - Fundamentos de Algoritmos.....	11
Valor de la semana (Saber ser).....	11
Conocimiento (Saber).....	11
Habilidades ( Saber Hacer) .....	11
Sesión No. 6, Unidad No. 4 - Fundamentos de Algoritmos.....	12
Valor de la semana (Saber ser).....	12
Conocimiento (Saber).....	12
Habilidades ( Saber Hacer) .....	12

Sesión No. 7, Unidad No. 5 - Algoritmos .....	13
Valor de la semana (Saber ser).....	13
Conocimiento (Saber).....	13
Habilidades ( Saber Hacer) .....	13
Sesión No. 8, Unidad No. 5 - Algoritmos .....	14
Valor de la semana (Saber ser).....	14
Conocimiento (Saber).....	14
Habilidades ( Saber Hacer) .....	14
Sesión No. 9, Unidad No. 5 - Algoritmos .....	15
Valor de la semana (Saber ser).....	15
Conocimiento (Saber).....	15
Habilidades ( Saber Hacer) .....	15
Sesión No. 10, Unidad No. 5 - Algoritmos .....	16
Valor de la semana (Saber ser).....	16
Conocimiento (Saber).....	16
Habilidades ( Saber Hacer) .....	16
Sesión No. 11, Unidad No. 6 - Administración y representación de Algoritmos .....	17
Valor de la semana (Saber ser).....	17
Conocimiento (Saber).....	17
Habilidades ( Saber Hacer) .....	17
<b>Tiempo de Auto-aprendizaje .....</b>	<b>18</b>
<b>Rúbrica de Evaluación .....</b>	<b>18</b>
<b>Resumen de Ponderaciones .....</b>	<b>18</b>
<b>Normativa Académica y Ética del Curso.....</b>	<b>19</b>
<b>Equipo Académico .....</b>	<b>20</b>
Coordinador del Área.....	20
Sección A.....	20
Sección B.....	21
Sección C.....	22
<b>Bibliografía.....</b>	<b>23</b>
<b>E-Grafía.....</b>	<b>23</b>

## 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.
4	Demuestra destreza y habilidad en la selección, uso y adaptación de herramientas metodológicas, tecnológicas, equipos especializados y en la lectura e interpretación de datos, pertinentes al contexto de su ejercicio profesional.
5	Aplica los conocimientos de su disciplina en la elaboración, fundamentación y defensa de argumentos para prevenir y resolver problemas complejos en su campo profesional, identificando y aplicando innovaciones.

### Competencias Generales

No.	Competencia
1	Aplica conocimientos tecnológicos con ética profesional y respetando y cuidando los recursos naturales, humanos y financieros.
2	Aplica principios básicos de ingeniería, ciencias de computación y sistemas de información y comunicación, en la formulación y resolución adecuada de problemas complejos.
3	Aplica estándares de calidad, eficiencia y seguridad en la implementación adecuada de soluciones de software, hardware y TIC en general.
4	Construye soluciones integrales trabajando en forma colaborativa y propositiva en equipos interdisciplinarios, en forma presencial o utilizando plataformas virtuales.
5	Actualiza permanente sus conocimientos relacionados con TIC en general, apoyándose en las estrategias de aprendizaje apropiadas.

## 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 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
3	El estudiante analiza representaciones intermedias de código mediante código de tres direcciones para identificar oportunidades de optimización.	Analizar

## Diseño Didáctico por Competencias

Esta sección organiza las sesiones del laboratorio en función de las competencias que el estudiante debe desarrollar. Cada clase incluye valores (saber ser), contenidos teóricos (saber) y habilidades prácticas (saber hacer), permitiendo un aprendizaje integral y aplicado. Las actividades están alineadas con los objetivos del curso y el perfil del egresado.

## 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	Actividad de diagnóstico en Kahoot que permite al tutor evaluar de forma inmediata e interactiva el nivel de dominio de los conceptos fundamentales para el curso de compiladores 1.

### 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	Actividad de diagnóstico en Kahoot con preguntas relacionadas al curso actual.

## Sesión No. 1, Unidad No. 1 - Traducción dirigida por la sintaxis

### Valor de la semana (Saber ser)

<b>Nombre: Comprensión integral</b>
Identificar y diferenciar cada fase de la compilación fortalece la visión global del proceso de traducción de lenguajes.
<b>Nombre: Pensamiento crítico</b>
Comparar enfoques como compilar, interpretar y transpilar fomenta la capacidad de análisis y toma de decisiones técnicas fundamentadas.

### Conocimiento (Saber)

Competencia(s)	
<ol style="list-style-type: none"> <li>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.</li> <li>2. 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.</li> <li>3. El estudiante explica la eliminación de recursión por la izquierda con ejemplos paso a paso para preparar gramáticas para análisis sintáctico.</li> <li>4. El estudiante utiliza herramientas generadoras de lexers y parsers tales como Flex, Bison y ANTLR en proyectos de análisis léxico y sintáctico.</li> </ol>	
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

### Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
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.	Actividad en clase	0

## Sesión No. 2, Unidad No. 1 - Traducción dirigida por la sintaxis

### Valor de la semana (Saber ser)

<b>Nombre: Rigor técnico</b>
Aplicar con precisión las reglas de interpretación y traducción de instrucciones, asegurando resultados exactos.
<b>Nombre: Responsabilidad</b>
Ser consciente del impacto que tiene una interpretación correcta o incorrecta en el funcionamiento del sistema.

### Conocimiento (Saber)

Competencia(s)	
<ol style="list-style-type: none"> <li>El estudiante explica la eliminación de recursión por la izquierda con ejemplos paso a paso para preparar gramáticas para análisis sintáctico</li> <li>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.</li> <li>El estudiante utiliza herramientas generadoras de lexers y parsers tales como Flex, Bison y ANTLR en proyectos de análisis léxico y sintáctico</li> </ol>	
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

### Habilidades (Saber Hacer)

Competencia	Tipo de	Ponderación
-------------	---------	-------------



	Actividad	
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.	Ejercicio	0

### Sesión No. 3, Unidad No. 1 - Traducción dirigida por la sintaxis

#### Valor de la semana (Saber ser)

<b>Nombre: Trabajo en equipo</b>
Resolver problemas entre varios, repartiéndose el análisis de cada parte, hace que todo sea más fácil y se aprende más.
<b>Nombre: Claridad al explicar</b>
Aprender a resolver y compartir ideas sobre entornos y variables mejora la forma en que nos comunicamos en equipo.

#### Conocimiento (Saber)

Competencia(s)	
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. 2. El estudiante utiliza herramientas generadoras de lexers y parsers tales como Flex, Bison y ANTLR en proyectos de análisis léxico y sintáctico.	
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

#### Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
El estudiante evalúa el desempeño de cada fase del	Otros	0

compilador a través de métricas de tiempo y uso de memoria para comparar implementaciones como GCC vs. Clang		
--	--	--

## Sesión No. 4, Unidad No. 1 - Traducción dirigida por la sintaxis

### Valor de la semana (Saber ser)

<b>Nombre: Responsabilidad computacional</b>
Entender el poder de un lenguaje implica usarlo de forma ética y consciente.
<b>Nombre: Optimización responsable</b>
Buscar eficiencia sin sacrificar legibilidad ni corrección.

### Conocimiento (Saber)

Competencia(s)	
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. 2. El estudiante utiliza herramientas generadoras de lexers y parsers tales como Flex, Bison y ANTLR en 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

### Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
El estudiante comprende las fases de un compilador a través de diagramas de flujo y modelos teóricos con ejemplos de compilación real	Ejercicio	0
El estudiante evalúa el desempeño de cada fase del compilador a través de métricas de tiempo y uso de memoria para comparar diferentes implementaciones	Actividad	0

El estudiante aplica técnicas de optimización de código como optimización local y global para reducir tamaño de código y mejorar rendimiento	Otros	0
--	-------	---

## Sesión No. 5, Unidad No. 1 - Traducción dirigida por la sintaxis

### Valor de la semana (Saber ser)

<b>Nombre: Responsabilidad en la interoperabilidad</b>
Llamar funciones externas exige control y seguridad.
<b>Nombre: Ética del alcance</b>
Mantener el uso de variables dentro de su contexto lógico y justo.

### Conocimiento (Saber)

Competencia	
1. El estudiante utiliza herramientas generadoras de lexers y parsers tales como Flex, Bison y ANTLR en proyectos de análisis léxico y sintáctico 2. 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
Funciones y cerraduras en un intérprete	Funciones embebidas
Funciones y cerraduras en un intérprete	Funciones foráneas
Funciones y cerraduras en un intérprete	Clousures
Funciones y cerraduras en un intérprete	Resolución de variables
Funciones y cerraduras en un intérprete	Hoisting
Funciones y cerraduras en un intérprete	Call Stack

### Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
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	Actividad	0

registros		
El estudiante explica la eliminación de recursión por la izquierda con ejemplos paso a paso para preparar gramáticas para análisis sintáctico	Ejercicio	0

## Sesión No. 6, Unidad No. 2 – Generación de código intermedio

### Valor de la semana (Saber ser)

<b>Nombre: Responsabilidad</b>
porque al implementar conceptos de POO, los estudiantes deben asumir el diseño correcto de clases y métodos, comprendiendo su impacto en el mantenimiento del código.
<b>Nombre: Precisión</b>
Ya que, al aplicar técnicas de optimización y análisis de código, se requiere atención a los detalles para lograr mejoras reales y evitar errores.

### Conocimiento (Saber)

Competencia	
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.	
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

### Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
El estudiante aplica técnicas de optimización de código como optimización local y global para reducir tamaño de código y	Ejercicio	0

mejorar rendimiento		
El estudiante analiza representaciones intermedias de código mediante código de tres direcciones para identificar oportunidades de optimización	Práctica	0

## Sesión No. 7, Unidad No. 2 – Generación de código intermedio

### Valor de la semana (Saber ser)

<b>Nombre: Rigor</b>
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.
<b>Nombre: Autonomía</b>
dado que muchos de estos temas requieren que el estudiante investigue, practique e identifique por su cuenta cómo se comportan las representaciones internas del código.

### Conocimiento (Saber)

Competencia	
1. El estudiante analiza representaciones intermedias de código mediante código de tres direcciones para identificar oportunidades de optimización.	
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

### Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
El estudiante explica la eliminación de recursión por la izquierda con ejemplos paso a paso para preparar gramáticas para análisis sintáctico	Ejercicio	0
El estudiante analiza representaciones intermedias de código mediante código de tres direcciones para identificar oportunidades de optimización	Actividad	0

## Sesión No. 8, Unidad No. 2 – Generación de código intermedio

### Valor de la semana (Saber ser)

<b>Nombre: Responsabilidad computacional</b>
El uso correcto de las tablas de símbolos ayuda a la eficiencia del programa
<b>Nombre: Optimización responsable</b>
El uso correcto de memoria ayuda a que los programas se desempeñen de forma adecuada

### Conocimiento (Saber)

Competencia	
1. 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
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
Entornos, variables globales y locales 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

## Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
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	Actividad	0

## Sesión No. 9, Unidad No. 2 – Generación de código intermedio

### Valor de la semana (Saber ser)

<b>Nombre: Rigor técnico</b>
La traducción de funciones e instrucciones ocupa la mayor parte de complejidad en un compilador por lo que mejora su uso.
<b>Nombre: Optimización</b>
La traducción adecuada de instrucciones permite que los programas generados por el compilador tengan un rendimiento adecuado.

### Conocimiento (Saber)

Competencia	
1. 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. 2. El estudiante analiza representaciones intermedias de código mediante código de tres direcciones para identificar oportunidades de optimización.	
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

## Habilidades (Saber Hacer)

Competencia	Tipo de Actividad	Ponderación
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	Ejercicio	0
El estudiante analiza representaciones intermedias de código mediante código de tres direcciones para identificar oportunidades de optimización	Ejercicio	0

## Sesión No. 10, Unidad No. 2 – Generación de código intermedio

### Valor de la semana (Saber ser)

<b>Nombre: Responsabilidad computacional</b>
El uso de la memoria permite que los programas generados por el compilador tengan un rendimiento adecuado.
<b>Nombre: Ética</b>
El uso adecuado de la memoria permite que los costos de utilización del programa sean menores al ser optimizados.

### Conocimiento (Saber)

Competencia	
1. 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. 2. El estudiante aplica técnicas de optimización de código como optimización local y global para reducir tamaño de código y mejorar rendimiento.	
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

### Habilidades (Saber Hacer)

Competencia	Tipo de	Ponderación
-------------	---------	-------------



	Actividad	
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	Ejercicio	0
El estudiante aplica técnicas de optimización de código como optimización local y global para reducir tamaño de código y mejorar rendimiento	Ejercicio	0

## Sesión No. 11, Unidad No. 3 - Optimización de código

### Valor de la semana (Saber ser)

<b>Nombre: Precisión</b>
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.
<b>Nombre: Responsabilidad</b>
Se deben de atender los procesos de optimización aplicando todas las reglas necesarias para que esta se lleve a cabo con éxito.

### Conocimiento (Saber)

Competencia	
1. El estudiante aplica técnicas de optimización de código como optimización local y global para reducir tamaño de código y mejorar rendimiento.	
Tema	Subtema
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

### Habilidades (Saber Hacer)

Competencia	Tipo de	Ponderación
-------------	---------	-------------

	Actividad	
El estudiante aplica técnicas de optimización de código como optimización local y global para reducir tamaño de código y mejorar rendimiento	Ejercicio	0

## Tiempo de Auto-aprendizaje

Tipo	Horas de Auto-aprendizaje
Proyectos	80
Prácticas	0
Tareas	29
<b>Total</b>	<b>109</b>

## 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.

## Resumen de Ponderaciones

Tipo	Valor
Actividades en Clase	0
Proyectos	100

Prácticas	0
Tareas	0
Examen Final	0
<b>Total</b>	<b>100</b>

## 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.

## Equipo Académico

### Coordinador del Área

Nombre: <b>LUIS FERNANDO ESPINO BARRIOS</b>	Correo electrónico: <b>luisespino@yahoo.com</b>
--	--

## Sección A

### Docente

Nombre del Docente <b>BAYRON WOSVELY LOPEZ LOPEZ</b>	Correo electrónico <b>blopezw@yahoo.com</b>
---	--

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Día	X					X
Horario	7:10 - 8:50					12:10 - 13:50
Lugar	meet					meet

### Tutor(es)

Nombre del Tutor	<b>Jhonatan Josúe Tzunún Yax</b>	
Correo electrónico institucional	<b>3227943820801@ingenieria.usac.ed</b>	

Tipo		Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Clase	Día		X				
	Horario		17:20 - 19:00				
	Lugar		meet				
Atención al Estudiante	Día						
	Horario						

	Lugar						
--	-------	--	--	--	--	--	--

## Sección B

### Docente

Nombre del Docente <b>Luis Fernando Espino Barrios</b>	Correo electrónico luisespino@yahoo.com
---	--

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Día		X				
Horario		07:10 a 10:30				
Lugar		Meet				

### Tutor(es)

Nombre del Tutor	<b>Rubén Alejandro Ralda Mejía</b>	
Correo electrónico institucional	<a href="mailto:3629952490101@ingenieria.usac.edu.gt">3629952490101@ingenieria.usac.edu.gt</a>	

Tipo		Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Clase	Día		X				
	Horario		17:20 a 19:00				
	Lugar		Meet				
Atención al Estudiante	Día						
	Horario						
	Lugar						

## Sección C

### Docente

Nombre del Docente <b>EDGAR RUBEN SABAN RAXON</b>	Correo electrónico ticosabrax@gmail.com
--	--

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Día		X		X		
Horario		19:00 - 20:40		19:00 - 20:54		
Lugar						

### Tutor(es)

Nombre del Tutor	<b>Ronaldo Javier Posadas Guerra</b>	
Correo electrónico institucional	3561774830101@ingenieria.usac	

Tipo		Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Clase	Día		X				
	Horario		17:20 - 19:00				
	Lugar		meet				
Atención al Estudiante	Día						
	Horario						
	Lugar						

## Bibliografía

Libros de texto • Compiladores, Principios, Técnicas y Herramientas Aho, Sethi y Ullman. PEARSON ADDISON- WESLEY, 2008, segunda edición

## E-Grafía