



Facultad de Ingeniería

Ingeniería de Sistemas de Información

Ingeniería de Software

Ciencia de Datos

Trabajo Final

“Compiladores”

2025

1. Competencia del curso.

- a) CG3. N2 Realiza trabajos básicos de investigación que contribuyan al desarrollo científico, aplicando los conocimientos, habilidades y el método lógico y científico en el marco de la ética y las líneas de investigación de la Universidad.
- b) CP2. N3 Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al ciclo de vida del Software.

2. Logro esperado del curso.

- a) Desarrolla una solución computacional utilizando el analizador léxico y sintáctico, demostrando la resolución de problemas.
- b) Aplica fundamentos de compiladores para solucionar problemas usando recursos computacionales de forma óptima.

3. Formación de equipos.

- a) Los equipos estarán conformados por 4 integrantes.
- b) La elección será por afinidad de los estudiantes.
- c) Los integrantes deben elegir un representante líder, quien hará las coordinaciones.

4. Actividades del proyecto.

- a) El equipo formado de estudiantes debe reunirse y seleccionar una organización de un rubro de negocio.
- b) De dicha organización seleccionar, por cada integrante, un área o proceso de negocio a analizar.
- c) Describir el caso de estudio.
- d) Analizar el caso y proponer una solución. Desarrollar un subconjunto de un compilador o procesador de lenguaje específico que permita interpretar o traducir un lenguaje diseñado para automatizar tareas dentro de una empresa de cualquier rubro.
- e) Deberán diseñar e implementar un sistema que interprete un lenguaje creado por el equipo, orientado a resolver tareas específicas dentro de una empresa. Este lenguaje puede ser usado, por ejemplo, para:
 - a. Generar reportes automáticos.
 - b. Controlar procesos de producción.
 - c. Gestionar inventarios.
 - d. Automatizar tareas administrativas.
 - e. Simular procesos de negocio.
- f) Incluir en su aplicación:
 - a. Diseño del lenguaje
 - i. Definición de la gramática (léxica, sintáctica y semántica).
 - ii. Ejemplos de programas escritos en el lenguaje.
 - b. Análisis léxico
 - i. Implementación de un analizador léxico.

- c. Análisis sintáctico
 - i. Implementación de un parser
- d. Análisis semántico:
 - i. Verificación de tipos, estructuras y reglas del lenguaje.
- e. Traducción dirigida por la sintaxis
 - i. Generación de código intermedio o ejecución directa de instrucciones.
- f. Aplicación práctica
 - i. Demostración de cómo el lenguaje resuelve un problema real en el rubro elegido.
- g) Deben probar su solución con dos códigos de ejemplo.
- h) Algunos ejemplos de aplicación se encuentran en el Anexo 3.

5. Informe.

El Informe será presentado en un documento con las siguientes consideraciones:

Contenido
a) Carátula. Debe incluir el nombre de la universidad, el nombre de su facultad, el nombre de su carrera, el título de su trabajo, el nombre del curso, el nombre del profesor, el código y nombre de los integrantes.
b) Tabla de contenidos.
c) Introducción.
d) Resumen.
e) Descripción de caso de estudio.
f) Objetivos.
g) Desarrollo. <ul style="list-style-type: none"> i. Identificación de requerimientos. ii. Solución del caso. <ul style="list-style-type: none"> - Diseño del lenguaje - Implementación de análisis léxico - Implementación de análisis sintáctico - Implementación de análisis semántico - Traducción dirigida por la sintaxis - Aplicación práctica.
h) Conclusiones y recomendaciones.
i) Referencias.

6. Cronograma.

Hitos	Semana	Puntaje
Presentación del 1 Avance. Ítems a – f del informe. (Excepción d.)	9	3
Presentación del 2 Avance. Ítems e - g. (Hasta el analizador sintáctico)	12	6

Presentación del Informe Final. Archivo de código fuente. Exposición.	15	11
Puntaje		20

7. Presentación.

- a) Cada avance será presentado a través de su aula virtual.
- b) En caso se detecte copia/plagio en el Trabajo Final, éste tendrá calificación 0, se reportará a los integrantes del equipo al Comité de Disciplina de la Facultad de Ingeniería para la evaluación de las sanciones respectivas según reglamento.
- c) En caso un equipo no presente un hito, éste tendrá puntaje 0, no podrá ser evaluado en posteriores presentaciones, cada hito tiene sus criterios de evaluación, revisar Anexo 1.
- d) El avance 1 y 2 serán entregados el último día de clase de la semana respectiva.
- e) El Informe Final y el código de la solución serán entregados como máximo al inicio de la 1 sesión de clase de la semana 15.
- f) El líder del equipo presenta los avances del trabajo, considerando la siguiente codificación del archivo: "Grupo0x.rar" o "Grupo0x.zip". que contiene todos los entregables de su trabajo final.
- g) No se aceptarán enlaces a repositorios de archivos.

8. Exposición.

- a) La exposición final será presencial para todos los integrantes del equipo.
- b) Todos los integrantes deben estar preparados para la exposición.
- c) La exposición será realizada por un integrante del equipo.
- d) El integrante será seleccionado por sorteo, de los que estén presentes en la fecha prevista.
- e) Los integrantes que no salgan sorteados podrán responder preguntas realizadas durante y al final de la exposición.
- f) El estudiante que no esté presente en la exposición tendrá calificación de "0" en Trabajo Final del Curso.

9. Evaluación.

La Evaluación se realizará con los criterios del Anexo 1. Considerar las siguientes indicaciones:

- Respetar los criterios de redacción según la guía de redacción USIL. (Ortografía, citaciones, referencias, etc.)
- La redacción debe tener coherencia, pertinencia, etc.
- La solución planteada debe incluir estructuras de control secuenciales, condicionales, repetitivas, funciones.
- La solución planteada debe incluir variables, constantes, arreglos.

Anexo 01

Evaluación del Trabajo Final de Curso

Curso/Clase	“Curso”		Semestre	2025 - 1
Evaluación	Trabajo Final.		Ciclo	I
Docente			Tipo	“A”
Duración	20 minutos de exposición.		Fecha	00/00/0000

Tema/Título	Título del trabajo de los estudiantes	Informe(I)	Exposición(E)	Participación(P)	Promedio (I*P+E)/2
Integrantes	Apellidos, Nombres	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0

1) Primer Avance:

Criterios	Puntaje Máximo	Puntaje Obtenido
Introducción.	0.5	
Descripción del caso.	2.0	
Objetivos.	0.5	
Total	3	

2) Segundo Avance:

Criterios	Puntaje Máximo	Puntaje Obtenido
Descripción del caso.	0.5	
Objetivos.	0.5	
Desarrollo.		
- Requerimientos	2	
- Solución del caso (Avance hasta el analizador sintáctico).	3	
Total	6	

3) Presentación Final:

a) Informe Final:

Criterios	Puntaje Máximo	Puntaje Obtenido
Resumen.	1	
Introducción.	0.5	
Desarrollo.	8	
Conclusiones.	0.5	
Recomendaciones.	0.5	
Referencias.	0.5	

Incumple reglas de ortografía, normas APA y presenta incoherencias en su redacción.	-2	
Total	11	0

b) Exposición (Resultado individual):

Criterios	Puntaje Máximo	1	2	3	4
Posee una presentación adecuada.	1				
Sustenta el conocimiento teórico aplicado en el trabajo presentado.	7				
Muestra dominio de conocimiento del trabajo presentado.	6				
Demuestra un desenvolvimiento fluido en su exposición.	4				
Utiliza adecuadamente material y herramientas para su exposición.	2				
Total	20				

c) Evaluación Grupal – Cumplimiento de Tareas Asignadas en el Proyecto:

Cada integrante evaluará a sus compañeros de equipo en una escala de 0% a 100% respecto al **cumplimiento de tareas asignadas, solo asignadas**. Debe sustentar además el motivo de las calificaciones que está asignando a sus compañeros en base a las tareas que fueron delegadas y cumplidas completa y correctamente (100%). La asignación de tareas es decisión y responsabilidad de todo el equipo y el cumplimiento de estas es individual. El promedio de esta evaluación se aplicará directamente al informe final del proyecto. Por ejemplo: Si el resultado del informe es 20 y un integrante A del equipo fue calificado por sus compañeros con un promedio de 80%, la nota de A sería de $20 \times 80\% = 16$.

Observaciones / Comentarios del Docente

Anexo 02

Material de Apoyo

Introducción:

- Descubre cómo hacer una introducción matadora con estas 17 técnicas:
<https://rockcontent.com/es/blog/como-hacer-una-introduccion/>
- Cómo elaborar una introducción:
http://comunicacionacademica.uc.cl/images/recursos/espanol/escritura/recurso_en_pdf_extenso/14_Como_elaborar_una_introduccion.pdf

Resumen:

- Cómo elaborar resúmenes:
<https://www.uma.es/publicadores/biblioteca/wwwuma/ElaborarResumenes.pdf>
- ¿Qué es un resumen lingüístico?: https://ceo.uai.cl/wp-content/uploads/2019/03/como_hacer_un_resumen_ceo_uai.pdf

Exposiciones:

- Guía del elevator pitch perfecto con ejemplos prácticos y plantilla:
<https://asana.com/es/resources/elevator-pitch-examples>
- Ejemplo de Elevator Pitch para presentarte tú o un proyecto:
<https://www.youtube.com/watch?v=uv357YzY7-k>
- Elevator Pitch: Presentaciones efectivas: <https://www.youtube.com/watch?v=r8J6oi3afds>

Anexo 03

Ejemplos de aplicaciones

Educación

Lenguaje para generar horarios académicos: permite definir cursos, profesores y restricciones, y genera automáticamente un horario válido.

Lenguaje para evaluar reglas de matrícula: verifica si un estudiante cumple con los requisitos para inscribirse en un curso.

Lenguaje para generar reportes de rendimiento: interpreta datos de notas y asistencia para generar informes personalizados.

Logística y transporte

Lenguaje para gestionar inventario: interpreta comandos para agregar, mover o eliminar productos en almacenes.

Lenguaje para simular operaciones logísticas: permite modelar procesos como carga, descarga y transporte.

Salud

Lenguaje para registrar y consultar citas médicas: permite definir pacientes, médicos y horarios.

Lenguaje para validar recetas médicas: verifica compatibilidad de medicamentos y dosis.

Lenguaje para generar reportes clínicos: interpreta datos de pacientes y genera informes médicos.

Retail y comercio

Lenguaje para definir promociones y descuentos: permite establecer reglas como “si compras 3, el cuarto es gratis”.

Lenguaje para gestionar ventas: interpreta comandos para registrar ventas, aplicar impuestos y generar facturas.

Lenguaje para analizar comportamiento de clientes: permite definir patrones de compra y generar recomendaciones.

Ejemplos concretos

Lenguaje para generación de horarios escolares

Rubro: Educación

Nombre del lenguaje: HorarioScript

Propósito: Automatizar la asignación de horarios de clases según disponibilidad de profesores y aulas.

Ejemplo de código:

```
curso "Matemáticas" {  
    profesor: "Juan Pérez";  
    días: ["Lunes", "Miércoles"];  
    aula: "A-101";  
    duración: 2h;  
}
```

Aplicación: El compilador verifica conflictos y genera un horario semanal optimizado.

Lenguaje para gestión de citas médicas

Rubro: Salud

Nombre del lenguaje: CitaLang

Propósito: Registrar y validar citas médicas según disponibilidad.

Ejemplo de código:

```
cita {  
    paciente: "Ana Torres";  
    médico: "Dr. Ramírez";  
    fecha: "2025-09-20";  
    hora: "10:30";  
    especialidad: "Dermatología";  
}
```

Aplicación: El compilador verifica disponibilidad y genera confirmación o reprogramación.

Lenguaje para gestión de recursos humanos

Rubro: Administración / RRHH

Nombre del lenguaje: HRScript

Propósito: Automatizar tareas como asignación de vacaciones, control de asistencia, etc.

Ejemplo de código:

```
empleado "Carlos Ruiz" {  
    cargo: "Analista";  
    vacaciones: "2025-10-01" a "2025-10-15";  
    contrato: indefinido;  
}
```

Aplicación: El compilador valida fechas, genera reportes y simula procesos administrativos.

Lenguaje para gestión hotelera

Rubro: Turismo / Hotelería

Nombre del lenguaje: HotelScript

Propósito: Automatizar reservas, asignación de habitaciones y servicios.

Ejemplo de código:

```
reserva {  
    cliente: "Lucía Gómez";  
    habitación: "Suite 301";  
    entrada: "2025-09-20";  
    salida: "2025-09-25";  
    servicios: ["Desayuno", "Spa"];  
}
```

Aplicación: El compilador interpreta las reservas y genera confirmaciones y asignaciones.

Lenguaje para gestión de pedidos en almacén

Rubro: Logística / Comercio

Nombre del lenguaje: PedidoLang

Propósito: Automatizar la entrada, salida y seguimiento de pedidos.

Ejemplo de código:

```
pedido "P-2025-001" {  
    cliente: "Distribuidora Norte";  
    productos: ["Caja A", "Caja B"];  
    estado: "pendiente";  
    prioridad: alta;  
}
```

Aplicación: El compilador genera instrucciones para preparar el pedido y actualizar su estado.

Lenguaje para programación de eventos culturales

Rubro: Cultura / Entretenimiento

Nombre del lenguaje: EventoLang

Propósito: Definir y organizar eventos con horarios, artistas y recursos.

Ejemplo de código:

```
evento "Festival de Jazz" {  
    fecha: "2025-11-12";  
    lugar: "Parque Central";  
    artistas: ["Trío Azul", "Sax Libre"];  
    recursos: ["Escenario", "Sonido", "Iluminación"];  
}
```

Aplicación: El compilador genera el cronograma y asigna recursos automáticamente.

Cada uno de estos ejemplos permite aplicar:

- Análisis léxico: identificar tokens como producto, fecha, regla, reserva, empleado, experimento, etc.
- Análisis sintáctico: verificar que las instrucciones estén bien estructuradas.

- Análisis semántico: validar que los valores tengan sentido (por ejemplo, que la fecha sea válida, cantidades, tipos).
- Traducción dirigida por la sintaxis: generar código que automatice la tarea (por ejemplo, generar un horario, validar una factura, asignar una habitación, etc.).