

# Entregables del Proyecto - Analizador de Oraciones

---

**Proyecto:** Analizador de Lenguaje Natural Simple

**Autores:** Ricardo Méndez, Emiliano Ledesma, Diego Jiménez, Abraham Velázquez

**Fecha:** Noviembre 2024

---

## Descripción General

Esta carpeta contiene los **6 entregables formales** del proyecto, todos basados en la **versión simplificada** del analizador ([version\\_simplificada.py](#)). Cada documento está diseñado como un entregable académico independiente que puede presentarse o evaluarse por separado.

---

## Índice de Entregables

### ENTREGABLE 1: Gramática Formal del Mini-Lenguaje

**Archivo:** [1\\_GRAMATICA\\_FORMAL.md](#)

**Contenido:**

- Definición formal completa de la gramática  $G = (V, T, P, S)$
- Alfabeto terminal de 45 palabras organizadas en 6 categorías
- 9 producciones con explicación detallada
- Ejemplos de derivación paso a paso
- Propiedades de la gramática (tipo 2, no ambigua, no recursiva)
- Validación formal

**Páginas:** ~12

**Nivel:** Fundamentos teóricos

---

### ENTREGABLE 2: Árboles de Derivación

**Archivo:** [2\\_ARBOLES\\_DERIVACION.md](#)

**Contenido:**

- Fundamento teórico de árboles de derivación
  - 6 ejemplos completos con:
    - Oración analizada
    - Componentes identificados
    - Árbol en formato ASCII
    - Reglas gramaticales aplicadas
    - Derivación paso a paso
  - Análisis de complejidad de los árboles
  - Representación visual en el sistema
-

- Verificación de unicidad (no ambigüedad)

**Páginas:** ~15

**Nivel:** Aplicación teórica

---

## ENTREGABLE 3: Ejemplos de Oraciones Válidas e Inválidas

**Archivo:** 3\_EJEMPLOS\_ORACIONES.md

### Contenido:

- 12 oraciones válidas organizadas en 5 categorías:
  - Con determinante + sustantivo
  - Con pronombres
  - Con nombres propios
  - Con adverbios
  - Con verbos intransitivos
- 14 oraciones inválidas en 5 categorías:
  - Sin sujeto
  - Sin verbo
  - Palabras no reconocidas
  - Estructura incorrecta
  - Incompletas
- Análisis detallado de cada caso
- Tablas resumen
- Análisis estadístico
- Patrones identificados

**Páginas:** ~24

**Nivel:** Casos prácticos

---

## ENTREGABLE 4: Reporte con Análisis del Modelo

**Archivo:** 4\_REPORTE\_ANALISIS\_MODELO.md

### Contenido:

- Resumen ejecutivo
- Modelo teórico completo
- Análisis del AFD (determinismo, completitud, minimalidad)
- Análisis de la gramática
- Análisis de complejidad computacional (temporal y espacial)
- Resultados experimentales
- Evaluación del modelo (fortalezas y debilidades)
- Limitaciones y restricciones
- Trabajo futuro
- Referencias académicas

**Páginas:** ~28

**Nivel:** Análisis técnico profundo

---

## ENTREGABLE 5: Implementación del Autómata Finito Determinista

**Archivo:** 5\_IMPLEMENTACION\_AFD.md

### Contenido:

- Especificación formal del AFD  $M = (Q, \Sigma, \delta, q_0, F)$
- Tabla de transiciones completa
- Diagrama de estados
- Propiedades verificadas (determinismo, completitud, minimalidad, alcanzabilidad)
- 5 fases del procesamiento explicadas
- Vocabulario del alfabeto detallado
- 6 casos de prueba paso a paso
- Análisis de complejidad
- Ventajas y limitaciones
- Comparación con alternativas
- Extensiones futuras

**Páginas:** ~22

**Nivel:** Implementación práctica

---

## ENTREGABLE 6: Código Completo Comentado con Ejecuciones

**Archivo:** 6\_CODIGO\_COMENTADO\_EJECUCIONES.md

### Contenido:

- Estructura general del código
- 23 fragmentos de código explicados línea por línea:
  - Clase AnalizadorSimple (9 fragmentos)
  - Clase AFDSimple (9 fragmentos)
  - Función main() (5 fragmentos)
- 8 ejecuciones completas simuladas con salida paso a paso:
  - "el gato come pescado"
  - "yo camino por el parque"
  - "María estudia matemáticas"
  - "los niños juegan"
  - "el perro corre rápidamente"
  - "come pescado" (rechazo)
  - "el gato pescado" (rechazo)
  - "el dinosaurio come pescado" (rechazo)
- 5 casos de uso avanzados
- Análisis de cada línea de código
- Complejidad de cada operación

- Simulaciones internas de ejecución

**Páginas:** ~55

**Nivel:** Documentación completa del código

---

## Organización por Propósito

### Para Presentación Teórica

1. ENTREGABLE 1: Gramática Formal
  2. ENTREGABLE 2: Árboles de Derivación
  3. ENTREGABLE 4: Reporte de Análisis

### Para Demostración Práctica

1. ENTREGABLE 3: Ejemplos de Oraciones
  2. ENTREGABLE 5: Implementación del AFD
  3. ENTREGABLE 6: Código Comentado

### Para Evaluación Académica

Todos los entregables (orden 1-6)

---

## Tabla Resumen de Entregables

| #     | Título                | Páginas | Tipo            | Código   |
|-------|-----------------------|---------|-----------------|----------|
| 1     | Gramática Formal      | 12      | Teoría          | No       |
| 2     | Árboles de Derivación | 15      | Teoría+Práctica | No       |
| 3     | Ejemplos de Oraciones | 24      | Práctica        | No       |
| 4     | Reporte de Análisis   | 28      | Análisis        | No       |
| 5     | Implementación AFD    | 22      | Técnico         | Parcial  |
| 6     | Código Comentado      | 55      | Técnico         | Completo |
| TOTAL |                       | 156     |                 |          |

---

## Información Técnica

### Base del Proyecto

- **Archivo fuente:** `version_simplificada.py` (373 líneas)
- **Lenguaje:** Python 3.7+
- **Dependencias:** Ninguna (solo módulos estándar)

## Gramática Implementada

```
S → SN SV
SN → DET N | PRON | N
SV → V SN | V SP | V ADV | V
SP → PREP SN
```

## AFD Implementado

```
Estados: Q = {q0, q1, q2, q3, qr}
Alfabeto: Σ = {SN, V, COMPLEMENTO}
Estado inicial: q0 = q0
Estados de aceptación: F = {q3}
```

## Vocabulario

- **Total:** 45 palabras
- **Categorías:** 6 (DET, N, V, PRON, PREP, ADV)
- **Oraciones generables:** ~10,000-50,000 (estimado)

---

## Guía de Uso

### Para Estudiantes

1. Comienza con **ENTREGABLE 1** para entender la gramática
2. Revisa **ENTREGABLE 2** para ver árboles de derivación
3. Practica con **ENTREGABLE 3** (ejemplos)
4. Profundiza con **ENTREGABLE 4** (análisis)

### Para Desarrolladores

1. Empieza con **ENTREGABLE 5** (especificación del AFD)
2. Estudia **ENTREGABLE 6** (código completo)
3. Consulta **ENTREGABLE 3** para casos de prueba
4. Usa **ENTREGABLE 4** para análisis de complejidad

### Para Profesores

1. **ENTREGABLE 1-4:** Material teórico completo
2. **ENTREGABLE 5-6:** Material práctico/código
3. Todos incluyen ejemplos evaluables

## Estándares de Documentación

Todos los entregables siguen:

- ✓ **Formato Markdown** para compatibilidad
  - ✓ **Estructura jerárquica** con índices
  - ✓ **Ejemplos concretos** en cada sección
  - ✓ **Notación formal** cuando corresponde
  - ✓ **Diagramas ASCII** para visualización
  - ✓ **Código comentado** con explicaciones
  - ✓ **Referencias académicas** cuando aplica
- 

## Palabras Clave por Entregable

| Entregable | Palabras Clave                                |
|------------|---|
| 1          | Gramática, GIC, Producciones, Vocabulario     |
| 2          | Árboles, Derivación, Sintaxis, Reglas         |
| 3          | Ejemplos, Válidas, Inválidas, Casos de prueba |
| 4          | Análisis, Complejidad, Evaluación, AFD        |
| 5          | Implementación, Estados, Transiciones, Fases  |
| 6          | Código, Fragmentos, Ejecuciones, Simulaciones |

---

## Compatibilidad

### Formatos de Exportación

- ✓ Markdown (.md) - Formato nativo
- ✓ PDF - Conversión con pandoc o herramientas online
- ✓ HTML - Visualización en navegador
- ✓ DOCX - Conversión para Word

### Herramientas Recomendadas

- **Editor:** VS Code, Typora, Obsidian
  - **Visualizador:** GitHub, GitLab, cualquier viewer MD
  - **Conversión:** pandoc, markdown-pdf, grip
- 

## Notas Importantes

1. **Todos los entregables están basados en `version_simplificada.py`**

- No confundir con `primer_AFD.py` (versión con spaCy)
- La versión simplificada no requiere instalación de dependencias

## 2. Coherencia entre documentos

- Los 6 entregables son coherentes entre sí
- Los ejemplos son los mismos a través de todos
- El vocabulario es consistente (45 palabras)

## 3. Independencia de documentos

- Cada entregable puede leerse de forma independiente
- Todos tienen introducción y conclusiones propias
- No es necesario leer en orden (excepto para aprendizaje)

## 4. Nivel académico

- Adecuado para nivel universitario
- Incluye fundamentos teóricos sólidos
- Equilibrio entre teoría y práctica

---

## Contacto y Créditos

**Autores:** Ricardo Méndez, Emiliano Ledesma

**Proyecto:** Analizador de Lenguaje Natural Simple

**Curso:** Teoría de Autómatas y Lenguajes Formales

**Fecha:** Noviembre 2024

---

## Licencia

Estos documentos son material académico del proyecto. Pueden ser utilizados con fines educativos citando la fuente.

---

**Última actualización:** 27 de Noviembre de 2024