

**Batería de Pruebas Exhaustiva para Aplicación de Conversión de Documentos**

**Resumen Ejecutivo**

La batería de pruebas propuesta comprende **246 casos de prueba** distribuidos en **20 formatos de documentos diferentes**, organizados en **5 categorías principales** que abarcan desde documentos válidos hasta casos críticos de estrés. El sistema está diseñado para evaluar la resistencia, precisión y capacidad de recuperación de su aplicación de conversión bajo condiciones reales de uso.

**Metodología de Testing**

**Categorías de Pruebas**

**1. DOCUMENTOS\_VÁLIDOS (40 casos)**

* **Propósito**: Verificar que la aplicación procesa correctamente documentos bien formados
* **Resultado esperado**: Conversión exitosa sin errores
* **Prioridad**: Alta

**2. ERRORES\_CORREGIBLES (60 casos)**

* **Propósito**: Evaluar la capacidad de la aplicación para detectar y corregir errores menores
* **Resultado esperado**: Conversión exitosa con advertencias
* **Prioridad**: Alta

**3. ERRORES\_CRÍTICOS (46 casos)**

* **Propósito**: Verificar el manejo robusto de errores que no pueden ser corregidos
* **Resultado esperado**: Fallo controlado con mensajes informativos
* **Prioridad**: Crítica

**4. CASOS\_LÍMITE (60 casos)**

* **Propósito**: Probar el comportamiento en los límites de especificaciones
* **Resultado esperado**: Variable según implementación
* **Prioridad**: Media

**5. STRESS\_TEST (40 casos)**

* **Propósito**: Evaluar resistencia bajo carga y condiciones extremas
* **Resultado esperado**: Variable según configuración
* **Prioridad**: Alta

**Formatos de Documentos Incluidos**

La batería incluye los siguientes 20 formatos de documentos con sus errores más comunes identificados:

**Documentos de Prueba por Categoría**

**Categoría 1: Documentos Válidos**

**PDF - Documento Válido Simple**

**Archivo**: valid\_pdf\_simple.pdf

%PDF-1.4  
1 0 obj  
<<  
/Type /Catalog  
/Pages 2 0 R  
>>  
endobj  
...

*Documento PDF con estructura básica correcta para verificar funcionamiento normal*

**JSON - Documento Válido Simple**

**Archivo**: valid\_json\_simple.json

{  
 "documento": "JSON válido simple",  
 "version": "1.0",  
 "datos": {  
 "titulo": "Documento de prueba",  
 "autor": "Sistema de Testing"  
 }  
}

**CSV - Documento Válido Simple**

**Archivo**: valid\_csv\_simple.csv

id,nombre,edad,ciudad,salario  
1,Juan Pérez,30,Madrid,45000  
2,María García,25,Barcelona,38000  
3,Pedro López,35,Valencia,42000

**Categoría 2: Errores Corregibles**

**PDF - Estructura Mal Formada (Corregible)**

**Archivo**: pdf\_malformed\_structure.pdf

%PDF-1.4  
...  
startxref  
INVALID\_REFERENCE  
%%EOF

*PDF con referencia xref inválida que puede ser reparada*

**JSON - Sintaxis Inválida (Corregible)**

**Archivo**: json\_invalid\_syntax.json

{  
 "documento": "JSON con sintaxis inválida",  
 "version": "1.0",  
 "datos": {  
 "titulo": "Documento de prueba",  
 "autor": "Sistema de Testing",  
 "contenido": [  
 {  
 "tipo": "texto",  
 "valor": "Este JSON tiene una coma de más"  
 },  
 {  
 "tipo": "numero",  
 "valor": 42,  
 }  
 ]  
 }  
}

*JSON con coma final extra que puede ser corregida automáticamente*

**CSV - Delimitadores Inconsistentes (Corregible)**

**Archivo**: csv\_inconsistent\_delimiters.csv

id,nombre,edad,ciudad,salario  
1,Juan Pérez,30,Madrid,45000  
2;María García;25;Barcelona;38000  
3,Pedro López,35,Valencia,42000  
4|Ana Martínez|28|Sevilla|40000

*CSV con diferentes delimitadores que pueden ser normalizados*

**Categoría 3: Errores Críticos**

**PDF - Protegido por Contraseña (No Corregible)**

**Archivo**: pdf\_password\_protected.pdf

%PDF-1.4  
...  
4 0 obj  
<<  
/Filter /Standard  
/V 1  
/R 2  
/O (ENCRYPTED\_OWNER\_PASSWORD\_32\_BYTES)  
/U (ENCRYPTED\_USER\_PASSWORD\_32\_BYTES)  
/P -60  
>>

*PDF con protección por contraseña que no puede ser procesado*

**JSON - Brackets Incorrectos (No Corregible)**

**Archivo**: json\_incorrect\_brackets.json

{  
 "documento": "JSON con brackets incorrectos",  
 "version": "1.0",  
 "datos": {  
 "titulo": "Documento de prueba",  
 "contenido": [  
 {  
 "tipo": "texto",  
 "valor": "Este JSON tiene brackets mal balanceados"  
 }  
 ]  
 }  
]

*JSON con brackets no balanceados que causa error de parsing*

**CSV - Registros Mal Formados (No Corregible)**

**Archivo**: csv\_malformed\_records.csv

id,nombre,edad,ciudad,salario  
1,Juan Pérez,30,Madrid,45000  
2,María García,25,Barcelona  
3,Pedro López,35,Valencia,42000,EXTRA\_FIELD  
4,Ana Martínez,28,Sevilla,40000

*CSV con registros inconsistentes que no pueden ser reparados automáticamente*

**Categoría 4: Casos Límite**

**Documentos Vacíos**

* **PDF vacío**: Solo headers PDF sin contenido
* **JSON vacío**: {}
* **CSV vacío**: Solo headers sin datos
* **XML vacío**: Solo declaración XML
* **TXT vacío**: Archivo de 0 bytes

**Documentos Muy Grandes**

* **PDF grande**: >100MB con múltiples páginas e imágenes
* **JSON grande**: Estructuras profundamente anidadas
* **CSV grande**: Millones de registros
* **XML grande**: Estructura jerárquica compleja

**Caracteres Especiales**

* **Unicode complejo**: Emojis, caracteres asiáticos, RTL
* **Caracteres de control**: Bytes nulos, caracteres no imprimibles
* **Encoding mixto**: UTF-8, Latin-1, ASCII mezclados

**Categoría 5: Stress Test**

**Procesamiento Concurrente**

* **100 documentos simultáneos** de cada formato
* **Diferentes tamaños** (pequeño, medio, grande)
* **Mezcla de válidos y erróneos**

**Estructuras Complejas**

* **Anidamiento profundo**: >1000 niveles en JSON/XML
* **Tablas masivas**: CSV con >1 millón de columnas
* **PDFs complejos**: Múltiples layers, transparencias, fuentes

**Plan de Implementación**

**Fase 1: Preparación del Entorno**

1. **Configurar infraestructura de testing**
2. **Crear archivos de prueba** usando el contenido proporcionado
3. **Implementar métricas de rendimiento**
4. **Configurar logging detallado**

**Fase 2: Ejecución de Pruebas**

1. **Documentos válidos**: Verificar funcionamiento base
2. **Errores corregibles**: Evaluar capacidad de reparación
3. **Errores críticos**: Probar manejo robusto de fallos
4. **Casos límite**: Verificar comportamiento en extremos
5. **Stress test**: Evaluar resistencia bajo carga

**Fase 3: Análisis y Reporte**

1. **Métricas de rendimiento**
   * Tiempo de procesamiento por formato
   * Uso de memoria y CPU
   * Tasa de éxito/fallo
   * Capacidad de recuperación
2. **Indicadores de calidad**
   * Precisión de conversión
   * Detección de errores
   * Capacidad de reparación
   * Estabilidad bajo estrés

**Criterios de Evaluación**

**Métricas Clave**

* **Tasa de éxito**: >95% para documentos válidos
* **Detección de errores**: >90% para errores conocidos
* **Capacidad de reparación**: >70% para errores corregibles
* **Tiempo de respuesta**: <5 segundos para documentos <10MB
* **Uso de memoria**: <2GB para documentos <100MB

**Umbrales de Aceptación**

* **Crítico**: Fallos en documentos válidos
* **Mayor**: Fallos en detección de errores críticos
* **Menor**: Rendimiento por debajo del 80% esperado
* **Informativo**: Advertencias en errores corregibles

**Recomendaciones de Implementación**

**Automatización**

* Integrar con CI/CD pipeline
* Ejecutar pruebas en cada release
* Monitorear métricas en tiempo real
* Generar reportes automáticos

**Escalabilidad**

* Usar contenedores para aislamiento
* Implementar paralelización
* Configurar recursos dinámicos
* Monitorear uso de sistema

**Mantenimiento**

* Actualizar casos de prueba regularmente
* Añadir nuevos formatos según necesidad
* Revisar métricas de rendimiento
* Documentar cambios y mejoras

Esta batería de pruebas exhaustiva le permitirá evaluar de manera integral la robustez, precisión y capacidad de su aplicación de conversión antes del lanzamiento al mercado, asegurando una experiencia de usuario óptima y un manejo adecuado de situaciones excepcionales.

⁂