Lista de Formatos Obsoletos e em Risco baseada no PRONOM

1. Script para Consultar e Classificar Formatos PRONOM

```
#!/usr/bin/env python3
# pronom_obsolescence_checker.py

import requests
import json
import csv
from datetime import datetime
from typing import Dict, List
import xml.etree.ElementTree as ET

class PRONOMObsolescenceChecker:
    """
    Consulta o registro PRONOM e classifica formatos por risco de obsolescência
    """

    def __init__(self):
        self.base_url = "https://www.nationalarchives.gov.uk/PRONOM"
        self.api_url = "https://www.nationalarchives.gov.uk/pronom"

        # Categorias de risco
        self.risk_categories = {
            'CRÍTICO': [],
            'ALTO': [],
            'MÉDIO': [],
            'BAIXO': [],
            'MÍNIMO': []
        }

        # Palavras-chave para classificação
        self.obsolete_keywords = [
            'obsolete', 'deprecated', 'discontinued', 'legacy',
```

```
            'superseded', 'replaced', 'outdated', 'no longer'
        ]

        self.high_risk_vendors = [
            'WordPerfect', 'Lotus', 'Corel', 'Borland', 'Novell',
            'PageMaker', 'Ventura', 'QuarkXPress', 'FrameMaker',
            'Harvard Graphics', 'Aldus', 'Claris'
        ]

        self.proprietary_vendors = [
            'Microsoft', 'Adobe', 'Apple', 'Autodesk'
        ]

    def load_pronom_database(self):
        """
        Carrega lista de formatos do PRONOM
        """
        formats_database = self._get_known_formats_database()
        return formats_database

    def _get_known_formats_database(self) -> List[Dict]:
        """
        Base de dados de formatos conhecidos com classificação de risco
        Baseado no PRONOM Registry
        """

        formats = [
            # FORMATOS DE ALTO RISCO / OBSOLETOS
            {
                'puid': 'x-fmt/44',
                'name': 'WordPerfect Document',
                'version': '5.x',
                'extension': ['wpd', 'wp', 'wp5'],
                'risk_level': 'CRÍTICO',
                'vendor': 'WordPerfect Corporation',
                'reason': 'Formato proprietário descontinuado, suporte limitado',
                'migration_target': 'PDF/A, DOCX, ODT'
            },
            {
                'puid': 'x-fmt/45',
```

```
        'name': 'WordPerfect Document',
        'version': '6.x',
        'extension': ['wpd', 'wp6'],
        'risk_level': 'ALTO',
        'vendor': 'Corel',
        'reason': 'Formato proprietário com suporte declinante',
        'migration_target': 'PDF/A, DOCX, ODT'
    },
    {
        'puid': 'x-fmt/274',
        'name': 'Lotus 1-2-3 Worksheet',
        'version': '2-5',
        'extension': ['wk1', 'wk3', 'wk4', 'wks'],
        'risk_level': 'CRÍTICO',
        'vendor': 'IBM/Lotus',
        'reason': 'Software descontinuado, formato obsoleto',
        'migration_target': 'XLSX, ODS, CSV'
    },
    {
        'puid': 'fmt/341',
        'name': 'dBase Database',
        'version': 'III-V',
        'extension': ['dbf'],
        'risk_level': 'ALTO',
        'vendor': 'dBase LLC',
        'reason': 'Formato legado com suporte limitado',
        'migration_target': 'SQLite, CSV, XLSX'
    },
    {
        'puid': 'fmt/467',
        'name': 'Harvard Graphics Presentation',
        'version': '2.x-3.x',
        'extension': ['ch3', 'sho'],
        'risk_level': 'CRÍTICO',
        'vendor': 'Software Publishing Corporation',
        'reason': 'Software descontinuado em 2001',
        'migration_target': 'PDF/A, PPTX, ODP'
    },
    {
        'puid': 'x-fmt/88',
```

```
        'name': 'PageMaker Document',
        'version': '6.x-7.x',
        'extension': ['pmd', 'pm6', 'p65'],
        'risk_level': 'ALTO',
        'vendor': 'Adobe (descontinuado)',
        'reason': 'Substituído pelo InDesign em 2004',
        'migration_target': 'INDD, PDF/A'
    },
    {
        'puid': 'fmt/1214',
        'name': 'Ventura Publisher Document',
        'version': '4.x-10.x',
        'extension': ['vp'],
        'risk_level': 'CRÍTICO',
        'vendor': 'Corel (descontinuado)',
        'reason': 'Descontinuado em 2002',
        'migration_target': 'PDF/A, INDD'
    },
    {
        'puid': 'fmt/998',
        'name': 'QuarkXPress Document',
        'version': '3.x-4.x',
        'extension': ['qxd'],
        'risk_level': 'MÉDIO',
        'vendor': 'Quark',
        'reason': 'Versões antigas com compatibilidade limitada',
        'migration_target': 'Versão atual do QXD, PDF/A, INDD'
    },

    # FORMATOS DE MÉDIO RISCO
    {
        'puid': 'fmt/39',
        'name': 'Microsoft Word Document',
        'version': '97-2003',
        'extension': ['doc'],
        'risk_level': 'MÉDIO',
        'vendor': 'Microsoft',
        'reason': 'Formato legado, substituído por DOCX',
        'migration_target': 'DOCX, PDF/A, ODT'
    },
```

```
    {
        'puid': 'fmt/61',
        'name': 'Microsoft Excel Spreadsheet',
        'version': '97-2003',
        'extension': ['xls'],
        'risk_level': 'MÉDIO',
        'vendor': 'Microsoft',
        'reason': 'Formato legado, substituído por XLSX',
        'migration_target': 'XLSX, ODS, CSV'
    },
    {
        'puid': 'fmt/126',
        'name': 'Microsoft PowerPoint Presentation',
        'version': '97-2003',
        'extension': ['ppt'],
        'risk_level': 'MÉDIO',
        'vendor': 'Microsoft',
        'reason': 'Formato legado, substituído por PPTX',
        'migration_target': 'PPTX, ODP, PDF/A'
    },
    {
        'puid': 'fmt/101',
        'name': 'Adobe Photoshop',
        'version': '3.x-6.x',
        'extension': ['psd'],
        'risk_level': 'MÉDIO',
        'vendor': 'Adobe',
        'reason': 'Versões antigas com recursos limitados',
        'migration_target': 'PSD atual, TIFF, PNG'
    },
    {
        'puid': 'fmt/124',
        'name': 'Macromedia Flash',
        'version': 'SWF',
        'extension': ['swf'],
        'risk_level': 'CRÍTICO',
        'vendor': 'Adobe (descontinuado)',
        'reason': 'Flash Player descontinuado em 2020',
        'migration_target': 'HTML5, MP4, WebM'
    },
```

```
    {
        'puid': 'fmt/175',
        'name': 'Windows Media Video',
        'version': '7-9',
        'extension': ['wmv'],
        'risk_level': 'MÉDIO',
        'vendor': 'Microsoft',
        'reason': 'Formato proprietário com suporte declinante',
        'migration_target': 'MP4, WebM, MOV'
    },
    {
        'puid': 'fmt/134',
        'name': 'RealMedia',
        'version': 'RM, RMVB',
        'extension': ['rm', 'rmvb'],
        'risk_level': 'ALTO',
        'vendor': 'RealNetworks',
        'reason': 'Formato obsoleto, software descontinuado',
        'migration_target': 'MP4, WebM'
    },

    # FORMATOS DE BAIXO RISCO
    {
        'puid': 'fmt/412',
        'name': 'Microsoft Word Document',
        'version': '2007-2019 (OOXML)',
        'extension': ['docx'],
        'risk_level': 'BAIXO',
        'vendor': 'Microsoft',
        'reason': 'Formato moderno e amplamente suportado',
        'migration_target': 'PDF/A para preservação de longo prazo'
    },
    {
        'puid': 'fmt/214',
        'name': 'Microsoft Excel Spreadsheet',
        'version': '2007-2019 (OOXML)',
        'extension': ['xlsx'],
        'risk_level': 'BAIXO',
        'vendor': 'Microsoft',
        'reason': 'Formato moderno e amplamente suportado',
```

```
        'migration_target': 'ODS, CSV para preservação'
    },
    {
        'puid': 'fmt/215',
        'name': 'Microsoft PowerPoint Presentation',
        'version': '2007-2019 (OOXML)',
        'extension': ['pptx'],
        'risk_level': 'BAIXO',
        'vendor': 'Microsoft',
        'reason': 'Formato moderno e amplamente suportado',
        'migration_target': 'PDF/A para preservação'
    },

    # FORMATOS DE RISCO MÍNIMO
    {
        'puid': 'fmt/95',
        'name': 'PDF/A',
        'version': '1a, 1b, 2, 3',
        'extension': ['pdf'],
        'risk_level': 'MÍNIMO',
        'vendor': 'ISO Standard',
        'reason': 'Padrão aberto para preservação digital',
        'migration_target': 'N/A - formato de preservação'
    },
    {
        'puid': 'fmt/291',
        'name': 'OpenDocument Text',
        'version': '1.x',
        'extension': ['odt'],
        'risk_level': 'MÍNIMO',
        'vendor': 'OASIS Standard',
        'reason': 'Padrão aberto ISO/IEC 26300',
        'migration_target': 'PDF/A para preservação'
    },
    {
        'puid': 'fmt/294',
        'name': 'OpenDocument Spreadsheet',
        'version': '1.x',
        'extension': ['ods'],
        'risk_level': 'MÍNIMO',
```

```
        'vendor': 'OASIS Standard',
        'reason': 'Padrão aberto ISO/IEC 26300',
        'migration_target': 'CSV para dados tabulares simples'
    },
    {
        'puid': 'fmt/295',
        'name': 'OpenDocument Presentation',
        'version': '1.x',
        'extension': ['odp'],
        'risk_level': 'MÍNIMO',
        'vendor': 'OASIS Standard',
        'reason': 'Padrão aberto ISO/IEC 26300',
        'migration_target': 'PDF/A para preservação'
    },
    {
        'puid': 'x-fmt/18',
        'name': 'Plain Text',
        'version': 'UTF-8',
        'extension': ['txt'],
        'risk_level': 'MÍNIMO',
        'vendor': 'Unicode Consortium',
        'reason': 'Formato universal e simples',
        'migration_target': 'N/A'
    },
    {
        'puid': 'x-fmt/384',
        'name': 'Comma Separated Values',
        'version': 'RFC 4180',
        'extension': ['csv'],
        'risk_level': 'MÍNIMO',
        'vendor': 'IETF Standard',
        'reason': 'Formato aberto e universal',
        'migration_target': 'N/A'
    },
    {
        'puid': 'fmt/11',
        'name': 'Portable Network Graphics',
        'version': '1.x',
        'extension': ['png'],
        'risk_level': 'MÍNIMO',
```

```
        'vendor': 'W3C Standard',
        'reason': 'Formato aberto amplamente suportado',
        'migration_target': 'N/A'
    },
    {
        'puid': 'fmt/353',
        'name': 'JPEG 2000',
        'version': 'JP2',
        'extension': ['jp2'],
        'risk_level': 'BAIXO',
        'vendor': 'ISO/IEC Standard',
        'reason': 'Padrão aberto para imagens de alta qualidade',
        'migration_target': 'PNG, TIFF para preservação'
    },
    {
        'puid': 'fmt/353',
        'name': 'TIFF',
        'version': '6.0',
        'extension': ['tif', 'tiff'],
        'risk_level': 'MÍNIMO',
        'vendor': 'Adobe (padrão aberto)',
        'reason': 'Padrão de facto para preservação de imagens',
        'migration_target': 'N/A'
    },
    {
        'puid': 'fmt/199',
        'name': 'MPEG-4',
        'version': 'Part 14',
        'extension': ['mp4'],
        'risk_level': 'BAIXO',
        'vendor': 'ISO/IEC Standard',
        'reason': 'Padrão aberto amplamente suportado',
        'migration_target': 'N/A'
    },
    {
        'puid': 'fmt/134',
        'name': 'WAV',
        'version': 'PCM',
        'extension': ['wav'],
        'risk_level': 'MÍNIMO',
```

```
        'vendor': 'Microsoft/IBM',
        'reason': 'Formato não comprimido padrão',
        'migration_target': 'FLAC para compressão sem perdas'
    },
    {
        'puid': 'fmt/279',
        'name': 'FLAC',
        'version': '1.x',
        'extension': ['flac'],
        'risk_level': 'MÍNIMO',
        'vendor': 'Xiph.Org (código aberto)',
        'reason': 'Código aberto, compressão sem perdas',
        'migration_target': 'N/A'
    },
    {
        'puid': 'fmt/134',
        'name': 'MP3',
        'version': 'MPEG-1/2 Layer III',
        'extension': ['mp3'],
        'risk_level': 'BAIXO',
        'vendor': 'ISO/IEC Standard',
        'reason': 'Amplamente suportado, mas com perdas',
        'migration_target': 'FLAC, WAV para preservação'
    },
    {
        'puid': 'x-fmt/263',
        'name': 'ZIP',
        'version': '2.0',
        'extension': ['zip'],
        'risk_level': 'BAIXO',
        'vendor': 'PKWARE',
        'reason': 'Amplamente suportado',
        'migration_target': 'TAR.GZ, 7Z para melhor compressão'
    },
    {
        'puid': 'x-fmt/266',
        'name': 'RAR',
        'version': '4.x-5.x',
        'extension': ['rar'],
        'risk_level': 'MÉDIO',
```

```
        'vendor': 'RARlab (proprietário)',
        'reason': 'Formato proprietário',
        'migration_target': 'ZIP, 7Z, TAR.GZ'
    },
    {
        'puid': 'fmt/471',
        'name': 'HTML',
        'version': '5',
        'extension': ['html', 'htm'],
        'risk_level': 'MÍNIMO',
        'vendor': 'W3C Standard',
        'reason': 'Padrão web atual',
        'migration_target': 'PDF/A para preservação'
    },
    {
        'puid': 'fmt/817',
        'name': 'EPUB',
        'version': '3.x',
        'extension': ['epub'],
        'risk_level': 'BAIXO',
        'vendor': 'IDPF/W3C Standard',
        'reason': 'Padrão aberto para e-books',
        'migration_target': 'PDF/A para preservação'
    },
    {
        'puid': 'fmt/24',
        'name': 'AutoCAD Drawing',
        'version': 'DWG R14-2000',
        'extension': ['dwg'],
        'risk_level': 'MÉDIO',
        'vendor': 'Autodesk (proprietário)',
        'reason': 'Formato proprietário, versões antigas',
        'migration_target': 'DWG atual, DXF, PDF/A'
    },
    {
        'puid': 'fmt/64',
        'name': 'AutoCAD Drawing Exchange',
        'version': 'DXF',
        'extension': ['dxf'],
        'risk_level': 'BAIXO',
```

```
            'vendor': 'Autodesk',
            'reason': 'Formato de intercâmbio documentado',
            'migration_target': 'PDF/A para preservação'
        },
    ]

    return formats

def classify_formats(self):
    """Classifica formatos por nível de risco"""
    formats = self.load_pronom_database()

    for fmt in formats:
        risk_level = fmt['risk_level']
        self.risk_categories[risk_level].append(fmt)

    return self.risk_categories

def export_to_json(self, filename='pronom_obsolescence_list.json'):
    """Exporta lista para JSON"""
    categories = self.classify_formats()

    output = {
        'metadata': {
            'generated_date': datetime.now().isoformat(),
            'source': 'PRONOM Registry',
            'total_formats': sum(len(v) for v in categories.values())
        },
        'risk_categories': categories
    }

    with open(filename, 'w', encoding='utf-8') as f:
        json.dump(output, f, indent=2, ensure_ascii=False)

    print(f"Lista exportada para: {filename}")
    return filename

def export_to_csv(self, filename='pronom_obsolescence_list.csv'):
    """Exporta lista para CSV"""
    categories = self.classify_formats()
```

```
        with open(filename, 'w', newline='', encoding='utf-8') as f:
            writer = csv.writer(f)
            writer.writerow([
                'PUID', 'Nome', 'Versão', 'Extensões', 'Nível de Risco',
                'Fornecedor', 'Razão', 'Formato de Migração'
            ])

            for risk_level, formats in categories.items():
                for fmt in formats:
                    writer.writerow([
                        fmt['puid'],
                        fmt['name'],
                        fmt['version'],
                        ', '.join(fmt['extension']),
                        risk_level,
                        fmt['vendor'],
                        fmt['reason'],
                        fmt['migration_target']
                    ])

        print(f"Lista exportada para: {filename}")
        return filename

    def generate_markdown_report(self, filename='pronom_obsolescence_report.md'):
        """Gera relatório em Markdown"""
        categories = self.classify_formats()

        with open(filename, 'w', encoding='utf-8') as f:
            f.write("# Lista de Formatos - Risco de Obsolescência\n\n")
            f.write(f"**Data de Geração:** {datetime.now().strftime('%Y-%m-%d
%H:%M:%S')}\n\n")
            f.write(f"**Fonte:** PRONOM Registry (The National Archives, UK)\n\n")
            f.write("---\n\n")

            risk_order = ['CRÍTICO', 'ALTO', 'MÉDIO', 'BAIXO', 'MÍNIMO']

            for risk_level in risk_order:
                formats = categories[risk_level]
                f.write(f"## {risk_level} RISCO ({len(formats)} formatos)\n\n")
```

```
                for fmt in formats:
                    f.write(f"### {fmt['name']} (v{fmt['version']})\n\n")
                    f.write(f"- **PUID:** {fmt['puid']}\n")
                    f.write(f"- **Extensões:** {', '.join(fmt['extension'])}\n")
                    f.write(f"- **Fornecedor:** {fmt['vendor']}\n")
                    f.write(f"- **Razão:** {fmt['reason']}\n")
                    f.write(f"- **Migração recomendada:** {fmt['migration_target']}\n\n")

                f.write("---\n\n")

        print(f"Relatório gerado: {filename}")
        return filename

    def get_statistics(self):
        """Retorna estatísticas sobre os formatos"""
        categories = self.classify_formats()

        stats = {
            'total_formats': sum(len(v) for v in categories.values()),
            'by_risk_level': {k: len(v) for k, v in categories.items()},
            'critical_extensions': [],
            'high_risk_extensions': []
        }

        for fmt in categories['CRÍTICO']:
            stats['critical_extensions'].extend(fmt['extension'])

        for fmt in categories['ALTO']:
            stats['high_risk_extensions'].extend(fmt['extension'])

        return stats


def main():
    """Função principal"""
    checker = PRONOMObsolescenceChecker()

    print("=== Gerador de Lista de Obsolescência PRONOM ===\n")
```

```
    print("Gerando arquivos de saída...\n")

    checker.export_to_json()
    checker.export_to_csv()
    checker.generate_markdown_report()

    stats = checker.get_statistics()
    print("\n=== ESTATÍSTICAS ===")
    print(f"Total de formatos catalogados: {stats['total_formats']}")
    print("\nDistribuição por nível de risco:")
    for level, count in stats['by_risk_level'].items():
        print(f"  {level}: {count} formatos")

    print(f"\nExtensões CRÍTICAS: {', '.join(set(stats['critical_extensions']))}")
    print(f"\nExtensões ALTO RISCO: {', '.join(set(stats['high_risk_extensions']))}")

    print("\n✓ Arquivos gerados com sucesso!")


if __name__ == "__main__":
    main()
```

2. Script de Integração com DROID

```
#!/usr/bin/env python3
# check_obsolescence_in_droid.py

import json
import csv
import pandas as pd
from collections import defaultdict

def check_obsolescence_in_collection(droid_csv, obsolescence_json):
    """
    Verifica quais formatos obsoletos existem na coleção
    """

    # Carregar lista de obsolescência
    with open(obsolescence_json, 'r') as f:
        obsolescence_data = json.load(f)
```

```
# Criar dicionário de extensões por risco
extension_risk = {}
for risk_level, formats in obsolescence_data['risk_categories'].items():
    for fmt in formats:
        for ext in fmt['extension']:
            extension_risk[ext.lower()] = {
                'risk_level': risk_level,
                'format_name': fmt['name'],
                'migration_target': fmt['migration_target'],
                'reason': fmt['reason']
            }

# Carregar relatório DROID
df = pd.read_csv(droid_csv)

# Analisar formatos encontrados
results = {
    'CRÍTICO': defaultdict(int),
    'ALTO': defaultdict(int),
    'MÉDIO': defaultdict(int),
    'BAIXO': defaultdict(int),
    'MÍNIMO': defaultdict(int),
    'NÃO_CLASSIFICADO': defaultdict(int)
}

total_files = len(df)
files_at_risk = 0

for idx, row in df.iterrows():
    ext = str(row.get('EXT', '')).lower()

    if ext in extension_risk:
        risk_info = extension_risk[ext]
        risk_level = risk_info['risk_level']
        results[risk_level][ext] += 1

        if risk_level in ['CRÍTICO', 'ALTO', 'MÉDIO']:
            files_at_risk += 1
    else:
```

```
        results['NÃO_CLASSIFICADO'][ext] += 1

# Gerar relatório
report = {
    'summary': {
        'total_files': total_files,
        'files_at_risk': files_at_risk,
        'risk_percentage': (files_at_risk / total_files * 100) if total_files > 0 else 0
    },
    'by_risk_level': {}
}

for risk_level, extensions in results.items():
    if extensions:
        report['by_risk_level'][risk_level] = {
            'count': sum(extensions.values()),
            'extensions': dict(extensions)
        }

# Salvar relatório
output_file = 'obsolescence_analysis_report.json'
with open(output_file, 'w') as f:
    json.dump(report, f, indent=2)

# Gerar relatório detalhado
generate_detailed_report(df, extension_risk, 'obsolescence_detailed_report.csv')

print(f"\n=== ANÁLISE DE OBSOLESCÊNCIA ===")
print(f"Total de arquivos: {total_files}")
print(f"Arquivos em risco: {files_at_risk} ({report['summary']['risk_percentage']:.2f}%)")
print(f"\nDistribuição por nível de risco:")

for risk_level in ['CRÍTICO', 'ALTO', 'MÉDIO', 'BAIXO', 'MÍNIMO']:
    if risk_level in report['by_risk_level']:
        count = report['by_risk_level'][risk_level]['count']
        print(f"  {risk_level}: {count} arquivos")

print(f"\n✓ Relatórios salvos:")
print(f"  - {output_file}")
print(f"  - obsolescence_detailed_report.csv")
```

```
    return report

def generate_detailed_report(df, extension_risk, output_file):
    """Gera relatório detalhado com todos os arquivos em risco"""

    with open(output_file, 'w', newline='', encoding='utf-8') as f:
        writer = csv.writer(f)
        writer.writerow([
            'Arquivo', 'Extensão', 'Nível de Risco', 'Formato',
            'Tamanho (bytes)', 'Razão', 'Migração Recomendada'
        ])

        for idx, row in df.iterrows():
            ext = str(row.get('EXT', '')).lower()
            filepath = row.get('FILE_PATH', row.get('URI', 'N/A'))
            filesize = row.get('SIZE', 0)

            if ext in extension_risk:
                risk_info = extension_risk[ext]

                # Apenas incluir arquivos de risco médio ou superior
                if risk_info['risk_level'] in ['CRÍTICO', 'ALTO', 'MÉDIO']:
                    writer.writerow([
                        filepath,
                        ext,
                        risk_info['risk_level'],
                        risk_info['format_name'],
                        filesize,
                        risk_info['reason'],
                        risk_info['migration_target']
                    ])

if __name__ == "__main__":
    import sys

    if len(sys.argv) < 2:
        print("Uso: python check_obsolescence_in_droid.py <droid_report.csv>
[obsolescence_list.json]")
        sys.exit(1)
```

```
    droid_csv = sys.argv[1]
    obsolescence_json = sys.argv[2] if len(sys.argv) > 2 else 'pronom_obsolescence_list.json'

    check_obsolescence_in_collection(droid_csv, obsolescence_json)
```

3. Uso Prático

```
#!/bin/bash
# run_obsolescence_check.sh

# 1. Gerar lista de formatos obsoletos do PRONOM
echo "Gerando lista de formatos obsoletos..."
python3 pronom_obsolescence_checker.py

# 2. Executar varredura DROID
echo "Executando varredura DROID..."
./scan_droid.sh

# 3. Verificar obsolescência na coleção
echo "Analisando obsolescência na coleção..."
LATEST_DROID=$(ls -t /OPT/droid/reports/report_*.csv | head -1)
python3 check_obsolescence_in_droid.py "$LATEST_DROID" pronom_obsolescence_list.json

echo "Análise completa!"
```

4. Saídas Geradas

Os scripts geram:

1. **pronom_obsolescence_list.json** - Lista completa em JSON
2. **pronom_obsolescence_list.csv** - Lista em CSV para Excel
3. **pronom_obsolescence_report.md** - Relatório legível em Markdown
4. **obsolescence_analysis_report.json** - Análise da sua coleção
5. **obsolescence_detailed_report.csv** - Lista de arquivos em risco

Esses scripts fornecem uma base sólida baseada no PRONOM para monitorar obsolescência tecnológica no seu acervo digital.

Revision #2
Created 23 October 2025 19:55:47 by Rondineli G. Saad
Updated 23 October 2025 20:10:49 by Rondineli G. Saad