



IBF

Istituto di Biofisica

CNR – Consiglio Nazionale delle Ricerche

Rapporto tecnico

Sviluppo di un software in Python per la generazione automatizzata dei prodotti scientifici dei Ricercatori e Tecnologi del CNR mediante protocollo OAI-PMH dalla piattaforma CNR Explora.

Ing. Francesco Impallari

Palermo, 09/08/2021

INDICE

1. Abstract	3
2. Introduzione	3
3. Obiettivi	4
4. Architettura del software	4
5. Tecnologie utilizzate	7
6. Algoritmo implementato	7
7. Risultati	8
8. Innovazione	8
9. Conclusioni	9
10. Codice Sorgente	10

1. Abstract

Il presente rapporto tecnico descrive la progettazione, lo sviluppo e l'implementazione di un software realizzato in linguaggio Python finalizzato all'acquisizione automatica dei prodotti scientifici dei Ricercatori e Tecnologi del Consiglio Nazionale delle Ricerche (CNR) attraverso il protocollo standard OAI-PMH (Open Archives Initiative – Protocol for Metadata Harvesting), utilizzando come sorgente dati la piattaforma istituzionale CNR Explora.

Il software consente di interrogare automaticamente il repository documentale del CNR, estrarre i metadati relativi ai prodotti della ricerca, identificarli mediante matricola e nominativo del ricercatore, elaborare le informazioni recuperate ed infine generare automaticamente una documentazione strutturata in formato Microsoft Word pronta per essere utilizzata nella compilazione di curricula scientifici, relazioni e documentazione amministrativa.

L'applicativo è stato progettato con l'obiettivo di eliminare le attività manuali di ricerca e trascrizione dei prodotti scientifici, migliorando qualità, uniformità e rapidità nella predisposizione della documentazione.

2. Introduzione

Nell'ambito delle attività tecnico-specialistiche di supporto alla gestione della produzione scientifica dell'Istituto di Biofisica del CNR è emersa l'esigenza di automatizzare il reperimento e la catalogazione dei prodotti scientifici dei Ricercatori e Tecnologi presenti nella piattaforma istituzionale CNR Explora. Tale esigenza nasce dalla frequente predisposizione di documentazione amministrativa e scientifica (curricula, procedure concorsuali, relazioni tecnico-scientifiche, rendicontazioni e valutazioni) che richiede il recupero sistematico delle informazioni bibliografiche.

La valorizzazione della produzione scientifica rappresenta un elemento fondamentale nelle attività istituzionali del CNR.

La predisposizione di curriculum, relazioni annuali, documentazione per concorsi, valutazioni comparative e rendicontazioni richiede frequentemente la consultazione della piattaforma CNR Explora, con successiva estrazione manuale delle informazioni bibliografiche.

Per automatizzare tale processo è stato sviluppato un software completamente realizzato in Python capace di interrogare direttamente il repository istituzionale mediante protocollo OAI-PMH e generare automaticamente un documento riepilogativo dei prodotti scientifici.

3. Obiettivi

Gli obiettivi principali del progetto sono stati:

- automatizzare il recupero dei prodotti scientifici;
- eliminare l'inserimento manuale dei dati;
- garantire uniformità nella documentazione prodotta;
- ridurre tempi ed errori;
- produrre automaticamente documentazione pronta per l'utilizzo amministrativo;
- valorizzare la produzione scientifica dei Ricercatori e Tecnologi.

4. Architettura del Software

Il software è stato sviluppato in linguaggio Python utilizzando un'architettura modulare composta dai seguenti componenti.

Interrogazione OAI-PMH

Connessione automatica al repository CNR tramite endpoint:

<http://www.cnr.it/oai-pmh/oai2.php>

mediante la libreria:

`oaipmh.client`

e gestione del protocollo OAI-PMH.

Gestione dei metadati

Il software acquisisce automaticamente i metadati Dublin Core dei prodotti scientifici, tra cui:

- autori;
- titolo;
- descrizione;
- rivista;
- anno;
- identificativo;
- tipologia del prodotto;
- editore;
- parole chiave;
- relazioni.

Ricerca dei prodotti

L'identificazione dei prodotti avviene mediante doppio criterio:

- matricola del Ricercatore/Tecnologo;
- nominativo autore.

Questa doppia verifica aumenta l'affidabilità dell'identificazione.

Eliminazione duplicati

Durante l'elaborazione viene confrontato l'identificativo del prodotto per evitare duplicazioni nella documentazione finale.

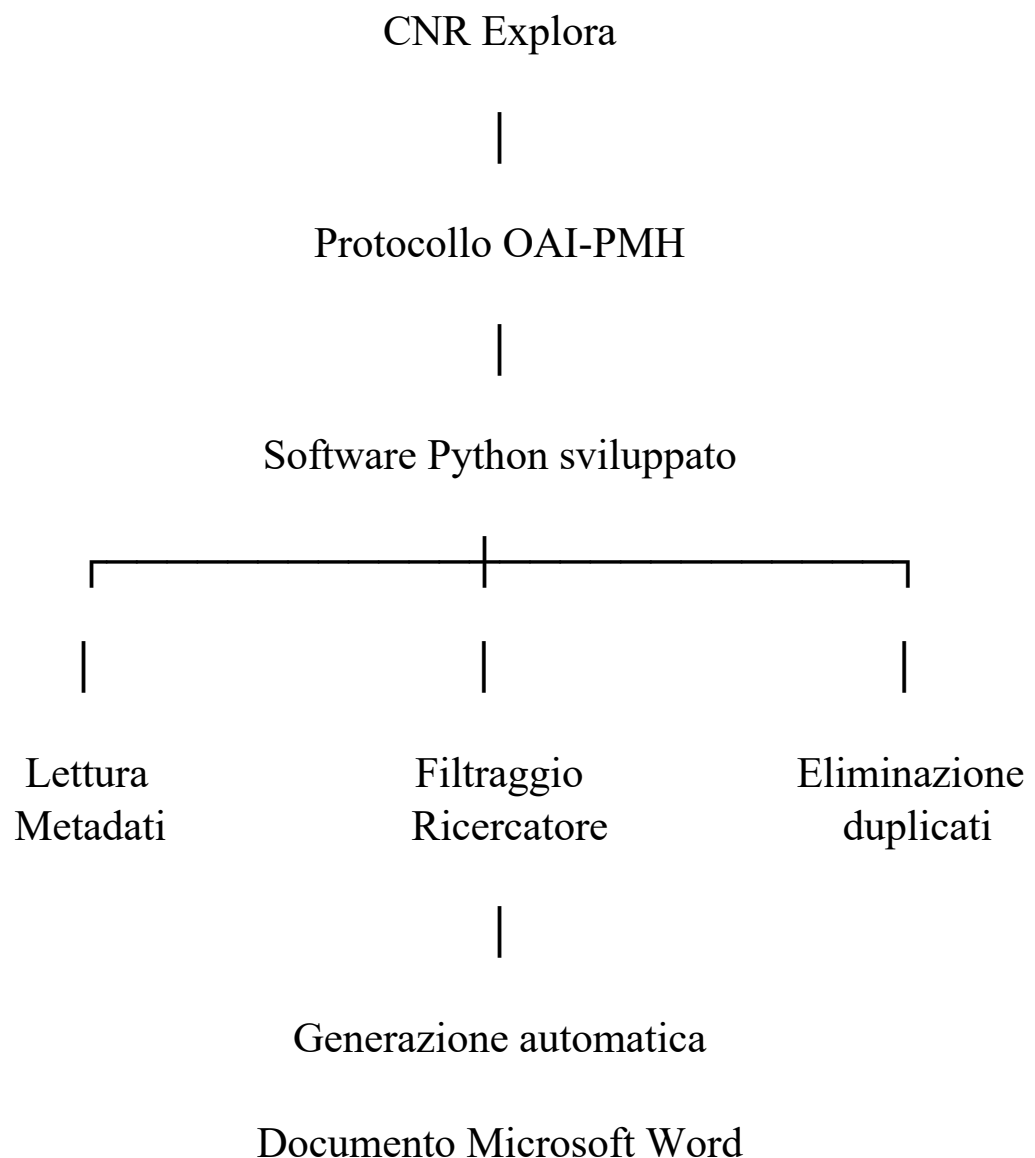
Generazione automatica della documentazione

Il software produce automaticamente un documento Microsoft Word (.docx) mediante la libreria **python-docx**, organizzando ogni prodotto scientifico all'interno di una tabella contenente:

- numero progressivo;
- tipologia;
- elenco autori;
- titolo;
- abstract;
- rivista;
- anno;
- ulteriori informazioni;
- identificativo del prodotto.

L'impaginazione è completamente automatica e uniforme.

Diagramma dell'Architettura



5. Tecnologie utilizzate

Lo sviluppo del software ha previsto l'utilizzo delle seguenti tecnologie:

- Python;
- OAI-PMH;
- Dublin Core Metadata;
- python-docx;
- espressioni regolari (Regular Expressions);
- elaborazione XML;
- gestione automatica documenti Microsoft Word.

Il software sviluppato ha consentito:

- riduzione dei tempi di elaborazione;
- standardizzazione della documentazione;
- eliminazione della trascrizione manuale;
- maggiore affidabilità;
- riutilizzabilità del software;
- esportazione automatica dei risultati;
- supporto ai procedimenti amministrativi.

6. Algoritmo implementato

L'algoritmo sviluppato esegue automaticamente le seguenti operazioni:

1. connessione al repository CNR Explora;
2. recupero dei record bibliografici;
3. lettura dei metadati;
4. individuazione dei prodotti associati al ricercatore;
5. eliminazione dei duplicati;

6. organizzazione dei dati;
7. costruzione delle tabelle;
8. impaginazione del documento;
9. generazione automatica del file Word finale.

Una volta impostati i parametri identificativi del Ricercatore o Tecnologo, il processo di acquisizione, elaborazione e generazione del documento viene eseguito automaticamente.

7. Risultati

Il software consente di ottenere automaticamente:

- elenco completo delle pubblicazioni;
- uniformità della documentazione prodotta;
- significativa riduzione dei tempi di elaborazione;
- diminuzione degli errori di trascrizione;
- riutilizzo della documentazione per concorsi, curricula e relazioni.

L'applicativo può essere utilizzato per qualsiasi Ricercatore o Tecnologo semplicemente modificando i parametri identificativi.

8. Innovazione

L'elemento innovativo del progetto consiste nell'aver sviluppato un sistema completamente automatico che integra:

- interoperabilità tramite protocollo OAI-PMH;
- harvesting dei metadati;
- filtraggio intelligente dei record;
- normalizzazione delle informazioni;
- produzione automatica della documentazione.

Il software trasforma una semplice consultazione del repository in un processo completamente automatizzato.

9. Conclusioni

L'intero progetto è stato ideato, progettato, sviluppato e collaudato dal sottoscritto.

Le attività svolte hanno riguardato:

- analisi dei requisiti;
- studio del protocollo OAI-PMH;
- progettazione dell'architettura software;
- implementazione degli algoritmi di acquisizione dei metadati;
- sviluppo delle funzioni di ricerca e filtraggio;
- implementazione della generazione automatica dei documenti Microsoft Word;
- attività di test e validazione dei risultati.

Il software rappresenta uno strumento tecnico-specialistico sviluppato per supportare la gestione della produzione scientifica del CNR mediante tecniche di interoperabilità tra sistemi informativi.

L'automazione introdotta consente una significativa riduzione dei tempi di predisposizione della documentazione scientifica e amministrativa, aumentando qualità, affidabilità e uniformità delle informazioni prodotte.

L'architettura modulare adottata rende inoltre il sistema facilmente estendibile e riutilizzabile per ulteriori esigenze di gestione della produzione scientifica.

Sviluppi futuri

Il software è stato progettato con architettura modulare e può essere esteso mediante:

- esportazione PDF;
- esportazione Excel;
- interfaccia grafica;
- interrogazione simultanea di più ricercatori;

- integrazione con IRIS CNR;
- aggiornamento automatico periodico.

Il software costituisce un prodotto originale sviluppato internamente al CNR per supportare le attività tecnico-specialistiche di gestione della produzione scientifica e documentale dell'Ente, introducendo un processo automatizzato di interoperabilità tra sistemi informativi basato su standard internazionali (OAI-PMH e Dublin Core). L'applicativo rappresenta un esempio di trasferimento di competenze informatiche a supporto delle attività istituzionali del personale di ricerca.

10. Disponibilità e documentazione del codice sorgente

Il codice sorgente dell'applicativo, sviluppato in linguaggio Python, è riportato integralmente nell'**Appendice A** del presente rapporto tecnico.

Il sorgente documenta le principali fasi del processo implementato:

- configurazione del client OAI-PMH e registrazione del lettore dei metadati Dublin Core;
- connessione all'endpoint istituzionale del CNR;
- acquisizione dei record relativi all'Istituto di Biofisica mediante il set OAI-PMH CDS008;
- estrazione dei campi bibliografici e descrittivi dei prodotti scientifici;
- individuazione dei prodotti associati al singolo Ricercatore o Tecnologo mediante numero di matricola e nominativo;
- gestione dei prodotti per i quali la matricola non risulta presente nei metadati storici;
- esclusione dei record duplicati attraverso il confronto dell'identificativo del prodotto;
- conversione e normalizzazione delle principali tipologie documentali;
- generazione automatica del documento Microsoft Word mediante la libreria python-docx;
- produzione di una scheda strutturata e numerata per ciascun prodotto scientifico individuato.

La presenza del codice sorgente consente di documentare in maniera verificabile il contributo personale fornito nell'analisi del problema, nella progettazione dell'algoritmo, nell'implementazione software, nella gestione dei metadati e nella produzione automatizzata dell'output documentale.

Appendice A – Codice sorgente dell'applicativo

Nota tecnica: il codice riportato costituisce la versione prototipale funzionante sviluppata nel 2021. L'applicativo è stato realizzato per l'esecuzione in modalità a riga di comando e utilizza i parametri identificativi del Ricercatore o Tecnologo — numero di matricola e nominativo — per la selezione dei relativi prodotti scientifici. Il sorgente è riprodotto preservando la struttura della versione originale utilizzata per le prove di estrazione e generazione documentale.

```
-----
# -*- coding: utf-8 -*-

"""
Software per l'acquisizione dei prodotti scientifici del personale
Ricercatore e Tecnologo del CNR mediante protocollo OAI-PMH
e per la generazione automatica di un documento Microsoft Word.

Autore: Ing. Francesco Impallari
Struttura: CNR - Istituto di Biofisica
Data: 09 agosto 2021
"""

import re

from oaipmh.client import Client
from oaipmh.metadata import MetadataRegistry, oai_dc_reader
from docx import Document
from docx.shared import Cm, Pt

# -----
# CONFIGURAZIONE
# -----

# Endpoint OAI-PMH del Consiglio Nazionale delle Ricerche
OAI_PMH_URL = "http://www.cnr.it/oai-pmh/oai2.php"

# Set OAI-PMH relativo all'Istituto di Biofisica del CNR
```

```
SET_ISTITUTO = "CDS008"
```

```
# Dati identificativi del Ricercatore/Tecnologo
```

```
MATRICOLA = "560"
```

```
NOMINATIVO = "Martorana V"
```

```
# Nome del documento Word prodotto dal software
```

```
NOME_FILE_OUTPUT = "{}.docx".format(NOMINATIVO)
```

```
# -----
```

```
# FUNZIONI DI SUPPORTO
```

```
# -----
```

```
def imposta_larghezza_colonna(colonna, larghezza):
```

```
    """
```

```
    Imposta la larghezza di una colonna della tabella Word.
```

```
    Parametri:
```

```
        colonna: colonna della tabella python-docx;
```

```
        larghezza: larghezza espressa in centimetri.
```

```
    """
```

```
    colonna.width = larghezza
```

```
    for cella in colonna.cells:
```

```
        cella.width = larghezza
```

```
def normalizza_lista(valore):
```

```
    """
```

```
    Restituisce sempre una lista.
```

```
    I metadati Dublin Core possono essere restituiti come liste,  
    stringhe singole oppure valori nulli.
```

```
    """
```

```
    if valore is None:
```

```
        return []
```

```
if isinstance(valore, list):  
    return valore
```

```
return [valore]
```

```
def testo_da_lista(valore, separatore="; "):  
    """  
    Converte una lista di metadati in una stringa leggibile.  
    """  
    elementi = normalizza_lista(valore)  
  
    elementi_puliti = []  
  
    for elemento in elementi:  
        if elemento is not None:  
            elementi_puliti.append(str(elemento).strip())  
  
    return separatore.join(elementi_puliti)
```

```
def primo_valore(valore):  
    """  
    Restituisce il primo elemento disponibile di un metadato.  
    """  
    elementi = normalizza_lista(valore)  
  
    if elementi:  
        return str(elementi[0]).strip()  
  
    return ""
```

```
def contiene_valore(lista_valori, valore_cercato):  
    """  
    Verifica se il valore ricercato è presente in almeno uno  
    degli elementi della lista.  
    """
```

```
if not valore_cercato:
    return False
```

```
modello = re.compile(
    re.escape(str(valore_cercato)),
    flags=re.IGNORECASE
)
```

```
for elemento in normalizza_lista(lista_valori):
    if modello.search(str(elemento)):
        return True
```

```
return False
```

```
def normalizza_tipologia(tipologia):
    """
```

```
    Converte le tipologie documentali OAI-PMH in una
    denominazione più leggibile.
    """
```

```
    valore = primo_valore(tipologia)
```

```
    tipologie = {
        "info:eu-repo/semantics/article": "Article",
        "info:eu-repo/semantics/conferenceObject": "Conference",
        "info:eu-repo/semantics/bookPart": "Book Part",
        "info:eu-repo/semantics/book": "Book",
        "info:eu-repo/semantics/doctoralThesis": "Doctoral Thesis",
        "info:eu-repo/semantics/report": "Report",
        "info:eu-repo/semantics/workingPaper": "Working Paper"
    }
```

```
    return tipologie.get(valore, valore)
```

```
def configura_documento():
    """
```

```
    Crea il documento Word e ne configura lo stile generale.
```

```
"""
```

```
documento = Document()
```

```
stile_normale = documento.styles["Normal"]
```

```
stile_normale.font.name = "Calibri"
```

```
stile_normale.font.size = Pt(8)
```

```
titolo = (
```

```
    "Elenco dei prodotti scientifici di "
```

```
    "{} - Matricola {}".format(NOMINATIVO, MATRICOLA)
)
```

```
documento.add_heading(titolo, level=0)
```

```
return documento
```

```
def aggiungi_prodotto_al_documento(
```

```
    documento,
```

```
    numero,
```

```
    tipologia,
```

```
    autori,
```

```
    titolo,
```

```
    descrizione,
```

```
    rivista,
```

```
    anno,
```

```
    identificativo):
```

```
"""
```

```
Inserisce nel documento Word una tabella contenente  
i metadati di un prodotto scientifico.
```

```
"""
```

```
dati_prodotto = (
```

```
    ("Tipologia prodotto", tipologia),
```

```
    ("Elenco autori", autori),
```

```
    ("Titolo", titolo),
```

```
    ("Descrizione", descrizione),
```

```
    ("Rivista/Editore", rivista),
```

```

("Codice identificativo (ISSN)", ""),
("Anno pubblicazione", anno),
("Indice di classificazione", ""),
("Impact Factor rivista", ""),
("Ruolo svolto", ""),
("Numero citazioni", ""),
("Altre informazioni", identificativo)
)

tabella = documento.add_table(
    rows=1,
    cols=2,
    style="Table Grid"
)

tabella.autofit = False
tabella.allow_autofit = False

celle_intestazione = tabella.rows[0].cells

celle_intestazione[0].paragraphs[0].add_run(
    "Nr."
).bold = True

celle_intestazione[1].text = str(numero)

imposta_larghezza_colonna(
    tabella.columns[0],
    Cm(3)
)

imposta_larghezza_colonna(
    tabella.columns[1],
    Cm(12)
)

for denominazione, contenuto in dati_prodotto:
    celle = tabella.add_row().cells

```

```
celle[0].text = str(denominazione)
celle[1].text = str(contenuto)
```

```
documento.add_paragraph("")
```

```
# -----
# INIZIALIZZAZIONE DEL CLIENT OAI-PMH
# -----
```

```
registro_metadati = MetadataRegistry()
```

```
registro_metadati.registerReader(
    "oai_dc",
    oai_dc_reader
)
```

```
client_oai = Client(
    OAI_PMH_URL,
    registro_metadati
)
```

```
# -----
# CREAZIONE DEL DOCUMENTO WORD
# -----
```

```
documento_word = configura_documento()
```

```
numero_prodotto = 0
```

```
# Insieme utilizzato per evitare la duplicazione dei prodotti
identificativi_elaborati = set()
```

```
# -----
# ACQUISIZIONE ED ELABORAZIONE DEI METADATI
```

```

# -----

for record in client_oai.listRecords(
    metadataPrefix="oai_dc",
    set=SET_ISTITUTO):

    intestazione_record = record[0]
    metadati_record = record[1]

    # Alcuni record OAI-PMH potrebbero non contenere metadati
    if metadati_record is None:
        continue

    metadati = metadati_record.getMap()

    publisher = metadati.get("publisher", [])
    description = metadati.get("description", [])
    language = metadati.get("language", [])
    formato = metadati.get("format", [])
    tipologia = metadati.get("type", [])
    rights = metadati.get("rights", [])
    data_publicazione = metadati.get("date", [])
    relation = metadati.get("relation", [])
    source = metadati.get("source", [])
    coverage = metadati.get("coverage", [])
    contributor = metadati.get("contributor", [])
    titolo = metadati.get("title", [])
    identifier = metadati.get("identifier", [])
    creator = metadati.get("creator", [])
    subject = metadati.get("subject", [])

    # -----
    # IDENTIFICAZIONE DEI PRODOTTI DEL
    RICERCATORE/TECNOLOGO
    # -----

    prodotto_associato_alla_matricola = contiene_valore(
        relation,

```

```

    MATRICOLA
)

prodotto_associato_al_nominativo = contiene_valore(
    creator,
    NOMINATIVO
)

# Il prodotto viene selezionato se nei metadati è presente
# la matricola oppure il nominativo dell'autore.
prodotto_selezionato = (
    prodotto_associato_alla_matricola
    or prodotto_associato_al_nominativo
)

if not prodotto_selezionato:
    continue

# -----
# CONTROLLO DEI DUPLICATI
# -----

identificativo_prodotto = primo_valore(identifier)

# Qualora il metadato identifier non sia disponibile,
# viene utilizzato l'identificativo dell'intestazione OAI-PMH.
if not identificativo_prodotto:
    identificativo_prodotto = str(
        intestazione_record.identifier()
    )

if identificativo_prodotto in identificativi_elaborati:
    continue

identificativi_elaborati.add(
    identificativo_prodotto
)

```

```
numero_prodotto += 1
```

```
# -----  
# NORMALIZZAZIONE DEI METADATI  
# -----
```

```
tipologia_prodotto = normalizza_tipologia(  
    tipologia  
)
```

```
autori_prodotto = testo_da_lista(  
    creator,  
    separatore="; "  
)
```

```
titolo_prodotto = testo_da_lista(  
    titolo,  
    separatore=" "  
)
```

```
descrizione_prodotto = testo_da_lista(  
    description,  
    separatore=" "  
)
```

```
rivista_prodotto = testo_da_lista(  
    publisher,  
    separatore="; "  
)
```

```
anno_publicazione = testo_da_lista(  
    data_publicazione,  
    separatore="; "  
)
```

```
# -----  
# PRODUZIONE DELLA TABELLA WORD  
# -----
```

```

aggiungi_prodotto_al_documento(
    documento=documento_word,
    numero=numero_prodotto,
    tipologia=tipologia_prodotto,
    autori=autori_prodotto,
    titolo=titolo_prodotto,
    descrizione=descrizione_prodotto,
    rivista=rivista_prodotto,
    anno=anno_publicazione,
    identificativo=identificativo_prodotto
)

```

```

print(
    "Prodotto n. {} - Identificativo: {}".format(
        numero_prodotto,
        identificativo_prodotto
    )
)

```

```

# -----
# SALVATAGGIO DEL DOCUMENTO
# -----

```

```

documento_word.save(
    NOME_FILE_OUTPUT
)

```

```

print("")
print("Elaborazione completata.")
print(
    "Numero complessivo di prodotti individuati: {}".format(
        numero_prodotto
    )
)
print(
    "Documento generato: {}".format(

```

```
)  
    NOME_FILE_OUTPUT  
)
```