Conferenza GARR 2026
La forma del cambiamento

Web archiving "attivo": preservare portali di ricerca legacy con IA e Docker

Michele Fiaschi
Scuola Normale Superiore

Un patrimonio digitale da conservare

Fine anni '90

CRIBECU

Nascono le prime piattaforme per le Digital Humanities della Scuola Normale Superiore

2000 → oggi

Laboratori Lettere

Evoluzione verso ambienti complessi: collezioni, archivi digitali, strumenti di ricerca online

Il problema

Legacy & obsolescenza

IT e HPC mantengono sistemi datati, tecnologie superate, stack non aggiornabili: costo crescente, rischio perdita del patrimonio

Da questa necessità concreta nasce la sperimentazione sul web archiving

Il processo: da piattaforma legacy ad archivio riproducibile

Focus principale

Containerizzazione

Portale: *La fortuna visiva di Pompei (2012)*
Frontend PRADO - backend DbSite

	Stack originale	→ Docker
PHP	4 / 5 (mysql_*)	**8.3**
MySQL	5.5 (MyISAM)	**8.0**
Deploy	bare metal / VM	**docker run**

Analisi del codice legacy, riscrittura delle dipendenze incompatibili, scrittura del Dockerfile, test e debugging → ambiente riproducibile con docker run.

Traiettoria evolutiva

Staticizzazione

Crawling e generazione sito statico: contenuto preservato, indipendente dall'infrastruttura, distribuibile e consultabile.

Traiettoria evolutiva

Astrazione del dato di ricerca

Il dataset astratto dalla piattaforma e depositato su Zenodo con DOI, citabile e riusabile da qualsiasi ricercatore, indipendentemente dall'interfaccia web. Prototipo di agente conversazionale per l'interrogazione.

AI come strumento operativo: il ruolo di Codex

Codex (OpenAI) è stato il co-pilota dell'intera sperimentazione

analisi codice legacy · scrittura Dockerfile · debugging · script di crawling · astrazione del dato di ricerca

Reverse engineering

Analisi dello stack legacy (PHP, CMS datati) e ricostruzione delle dipendenze e configurazioni per il Dockerfile

Scrittura & iterazione

Script di crawling e staticizzazione generati e affinati con prompt successivi, nessuna scrittura manuale di codice

Astrazione del dato

Open research data: il dataset astratto dalla piattaforma e depositato su Zenodo con DOI, citabile e riusabile indipendentemente dall'interfaccia

Non serviva saper programmare, serviva saper fare le domande giuste

Web archiving attivo: perché non basta lo snapshot

Snapshot tradizionale

WARC · Wayback Machine · crawl statico

- Navigazione per indici non funziona
- Query al database non eseguibili
- Relazioni tra record perse
- Provider OAI-PMH inaccessibile

E se il sito staticizzato perde funzionalità essenziali? E se il servizio originale non è pubblicabile per runtime obsoleti e vulnerabilità note? **L'unica alternativa è spegnere il servizio**.

Web archiving attivo

preservare il servizio, non solo i contenuti

- Runtime aggiornato e funzionante
- Database interrogabile
- Navigazione per indici preservata
- Ambiente riproducibile con docker run

L'artefatto di conservazione è un servizio eseguibile, non uno snapshot.

Takeaway

1. ~6 ore con Codex per portare PHP 4/5 + MySQL 5.5 su PHP 8.3 + MySQL 8 in Docker senza uno sviluppatore dedicato
2. Docker + staticizzazione + Zenodo: stack minimo, riproducibile, trasferibile. Chiunque abbia un legacy può partire da qui
3. Primo passo del gruppo di lavoro: deploy pubblico → pentest reale → iterazione con AI sulle criticità → hardening documentato
4. Obiettivo: linee guida per docenti e ricercatori per gestire il ciclo di vita del prodotto web già nel contratto

È un punto di partenza: cerchiamo collaboratori e confronto con chi ha esperienze simili

Grazie

Per le domande: wooclap.com
Codice evento: **CONFGARR26**