



# UNIVERSITÀ DI PISA

DIPARTIMENTO DI FILOLOGIA, LETTERATURA E LINGUISTICA

Corso di Laurea in  
Informatica Umanistica

Tesi di Laurea Triennale

## **Esportazione video per PH-Remix: tracce, ritagli e crediti autoriali**

Candidata:

Valentina Galli

Relatori:

Prof.ssa Enrica Salvatori

Dr.ssa Chiara Mannari

Dr. Davide Italo Serramazza

A.A. 2021/2022

# Indice generale

<b>1. Introduzione</b>	<b>2</b>
1.1 La <i>public history</i>	2
1.2 La <i>remix culture</i>	3
1.3 Il progetto PH-Remix	5
<b>2. Illustrazione linguaggi e librerie utilizzati</b>	<b>9</b>
2.1 Node.js	9
2.2 FFmpeg	9
2.3 fluent-ffmpeg	10
2.4 ffmpeg-concat	11
<b>3. Implementazione dell'esportazione video</b>	<b>13</b>
3.1 I dati in input	13
3.2 Architettura dello <i>script</i> di esportazione	14
3.3 Tracce video e audio	15
3.3.1 Illustrazione codice	15
3.4 Ritaglio	18
3.4.1 Illustrazione codice	18
3.5 Titoli di testa e coda con crediti autoriali	20
3.5.1 L'importanza dei crediti e la questione legale	20
3.5.2 Illustrazione codice	22
<b>4. Beta testing della piattaforma</b>	<b>25</b>
<b>5. Conclusioni</b>	<b>27</b>
<b>6. Glossario</b>	<b>28</b>
<b>7. Bibliografia</b>	<b>30</b>
Classica	30
Digitale	30

# 1. Introduzione

La presente relazione tratterà il mio contributo al progetto PH-Remix, in particolare la realizzazione dello *script* di esportazione dei video remixati dall'utenza. Il progetto PH-Remix, finanziato dalla Regione Toscana, è infatti finalizzato eminentemente a creare una piattaforma in grado di estrarre, tramite intelligenza artificiale, brevi video tematici e quindi remixarli, al fine di creare prodotti originali. Prima di proseguire e di entrare al cuore del progetto e del mio specifico compito, è tuttavia necessario presentare PH-Remix come punto d'incontro tra *public history* e *remix culture*.

## 1.1 La *public history*

Con il termine *public history* si indica generalmente l'impiego di storici e del metodo storico all'infuori dell'ambito accademico. Non si tratta dell'esposizione di un professore verso i propri studenti di concetti studiati e approfonditi: la *public history* condivide la storia con i diversi pubblici, legandola concretamente al territorio, alle comunità e ai problemi attuali, con l'obiettivo di mostrare questi ultimi come conseguenza diretta del passato. Il *public historian* offre la sua conoscenza storiografica, crea o ricerca le fonti, ma i prodotti della *public history* non si possono ricondurre esclusivamente al lavoro dello storico, bensì devono essere il risultato di uno sforzo congiunto di più autori appartenenti alla cittadinanza. L'obiettivo non è rendere la storia più semplice, ma individuare i bisogni di storia che si ripropongono in modo ricorrente in una comunità, ed educare ad affrontarli tramite gli insegnamenti del passato.<sup>1</sup>

Per questi motivi il termine '*public history*' vede la sua corretta traduzione in 'storia condivisa', e non 'pubblica' – termine che invece rimanda al cosiddetto 'uso pubblico della storia', causa di strumentalizzazioni e manipolazioni del passato messe in atto per scopi politici e ideologici.<sup>2</sup> Fare *public history* vuol dire condividere la storia con una cittadinanza attiva e rendere gli individui consapevoli dell'eredità territoriale che appartiene loro.

---

<sup>1</sup> Salvatori 2012, pp. 29-32

<sup>2</sup> Borghi 2016, pp. 245-246

L'intento è innanzitutto quello di coinvolgere la popolazione. Per fare ciò nel secolo della comunicazione digitale è necessario approcciarsi a nuovi metodi ancora largamente inesplorati dalla storia. Siti web, attività di *social networking*, video e mappe interattive sono tutti strumenti utili alla storia condivisa per raggiungere un pubblico più ampio possibile.

## 1.2 La *remix culture*

Per *remix culture* si intende l'attitudine sociale che consente e incoraggia la creazione di nuove opere derivate combinando o modificando materiali esistenti.<sup>3</sup> Si tratta di uno scambio creativo reso possibile dalle tecnologie digitali e che consiste essenzialmente in una pratica di 'copia e incolla'.

L'innovazione tecnologica favorisce lo sviluppo del *remix*, che ha origini ben più lontane di quello che si potrebbe immaginare.

La tecnica del remix racchiude in sé due grandi mondi, dalle origini simili ma ben distinte: quello dell'audio remix o remix musicale, e quello del video remix.

Per definizione, il remix si basa sull'operazione di *sampling*, in italiano tradotto come 'campionamento'. Esso è descritto da Gallagher come "l'atto di appropriazione della sezione di un artefatto registrato, [con l'intento di] usarlo nuovamente in un lavoro originale".<sup>4</sup> Sia l'audio remix che il video remix sfruttano il campionamento, ma a causa della sostanziale differenza nel supporto fisico, essi non si sono sviluppati parallelamente nella storia.

Si può ricondurre l'origine del campionamento video ai montatori cinematografici russi, in particolare a Sergej Michajlovič Ėjzenštejn e alla sua teoria del 'montaggio delle attrazioni'. Attraverso l'utilizzo di materiale extra-diegetico, spesso campionato da film americani degli anni '20, il suo cinema si differenziava dal classico *a découpage*. Egli, infatti, ricercava il conflitto tra le immagini al fine di suscitare forti emozioni negli spettatori.

Non è casuale che il materiale venisse campionato da pellicole statunitensi. Il loro messaggio veniva maneggiato ed alterato, spesso per promuovere messaggi

---

<sup>3</sup> Wikipedia, voce in lingua inglese Remix Culture

<sup>4</sup> Gallagher 2017, p. 25

appartenenti all'ideologia comunista. Ciò vuol dire che fin dalla nascita del remix video esso veniva utilizzato anche con fini politici, di propaganda o critica.

Lo stampo critico-politico può essere identificato come un vero e proprio genere, anche nei remix a noi contemporanei. Per trattare il remix video, molti studiosi si sono dedicati all'identificazione dei suoi sotto-generi, e quello riconosciuto univocamente da tutti è proprio il remix critico-politico.

McIntosh si occupa di individuare le caratteristiche fondamentali per classificare un remix video come critico-politico, con il fine di dimostrare che questi non hanno avuto origine semplicemente grazie a Internet o YouTube e alla vasta disponibilità di materiale video televisivo. Egli riporta come primo esempio rilevante il breve interludio comico realizzato dal britannico Charles A. Ridley nel 1941, dal titolo *Lambeth Walk: Dancing Nazis*. Ridley preleva i campioni per questa breve pellicola interamente dal film di propaganda nazionalsocialista *Il trionfo della volontà* del 1935, diretto da Leni Riefenstahl: la marcia dei militari nazisti viene tagliata, ripetuta e montata in modo che vada a tempo con la musica ed imiti un ballo.<sup>5</sup>

Il remix audio, invece, fece la sua comparsa più tardi. Questo è dovuto principalmente al supporto su cui era registrata la traccia audio: non era possibile 'tagliare' e 'montare' un disco fonografico, come invece accadeva con la pellicola cinematografica. Solo con l'introduzione del nastro magnetico, negli anni '40, nascono i primi 'collagisti di suoni', come i parigini Pierre Schaeffer e Pierre Henry, parte del movimento *musique concrète* fortemente influenzato dalle teorie del montaggio cinematografico. Nonostante il ruolo da pionieri della musica elettronica, i campioni utilizzati erano originali, creati e registrati dalle stesse persone che poi li avrebbero remixati. Per questo motivo l'esempio più lontano di remix musicale risale al 1961, ovvero *Collage #1* di James Tenney, creato con campioni della canzone di Elvis Presley *Blue Suede Shoes*, del 1956. Ma è all'inizio degli anni '70 che il *sampling* diventa un vero e proprio fenomeno di massa, grazie alla tecnica del *turntablism* – l'utilizzo di giradischi e mixer da DJ per creare i *break* musicali di sottofondo alla musica hip hop.<sup>6</sup>

---

<sup>5</sup> McIntosh 2012

<sup>6</sup> Gallagher 2017, pp. 26-27

La *remix culture* è viva, sempre in evoluzione ed eternamente contemporanea: moltissime applicazioni social usate da milioni di utenti ogni giorno si basano direttamente sul remix video. L'esempio più calzante è quello del social network cinese TikTok: esso permette agli utenti di usare la stessa traccia audio per creare video simili tra loro, oppure di 'duettare' con altri utenti unendo insieme due video.<sup>7</sup>

Secondo il professor Navas, “[II] Remix non è un movimento, ma un raccoglitore – una colla culturale. (...) è più come un virus che muta e assume diverse forme a seconda delle necessità di diverse culture. (...) è sempre non originale. Ma allo stesso tempo, quando implementato con successo, può diventare uno strumento di autonomia.” Egli aggiunge inoltre che le operazioni digitali di copia e incolla possono essere considerate la forma più popolare di *sampling* attuata da chiunque usi un computer o uno smartphone.<sup>8</sup>

Come verrà analizzato nel dettaglio in 3.5, la *remix culture* trova la sua opposizione più forte nell'ambito dei diritti d'autore, e ne sono evidenti le motivazioni.

### **1.3 Il progetto PH-Remix**

Il progetto di ricerca PH-Remix è coordinato dal Laboratorio di Cultura Digitale (LabCD) dell'Università di Pisa, in collaborazione con Festival dei Popoli, Mediateca Toscana e Fondazione Sistema Toscana.<sup>9</sup>

Il fine del progetto è quello di studiare nuove soluzioni per la valorizzazione e rivitalizzazione dei documentari appartenenti all'archivio del Festival dei Popoli, dando loro una nuova e originale funzione nell'ambito della *public history*.

I film vengono digitalizzati con lo scopo di conservazione e di riutilizzo tramite la piattaforma online, che permette la loro visualizzazione e il nuovo impiego grazie alle *clip* generate dinamicamente da algoritmi di IA.<sup>10</sup>

---

<sup>7</sup> Ferguson 2021

<sup>8</sup> Navas 2012, p. 4

<sup>9</sup> Sito web Laboratorio di Cultura Digitale - PH-Remix

<sup>10</sup> Paper Intelligenza artificiale e archivi audiovisivi: potenzialità e sfide del progetto “PH-Remix”

In figura 1 è mostrata l'architettura della piattaforma, che è basata sui seguenti microservizi – moduli sviluppati indipendentemente l'uno dall'altro che comunicano tra loro attraverso API (*application programming interface*):

- ▶ CONFER
  - ▷ In un primo momento attraverso un'interfaccia per l'upload dei video e la creazione delle schede film, vengono caricati i documentari che andranno a costituire il database della piattaforma.
- ▶ EXTRACT
  - ▷ Sui film caricati, gli algoritmi di intelligenza artificiale attuano l'estrazione di dati di diversa origine, ottenendo le *clip*.
  - ▷ È importante sottolineare che nell'interfaccia di remix accessibile agli utenti finali non saranno mai disponibili i film nella loro interezza, ma soltanto segmenti di essi selezionati tramite IA.
- ▶ REMIX
  - ▷ L'applicazione web di remix accessibile all'utente finale permette la ricerca delle clip e la creazione del montaggio.
- ▶ RENDER
  - ▷ Il modulo di esportazione genera il video lato server e lo restituisce all'utente per il download.

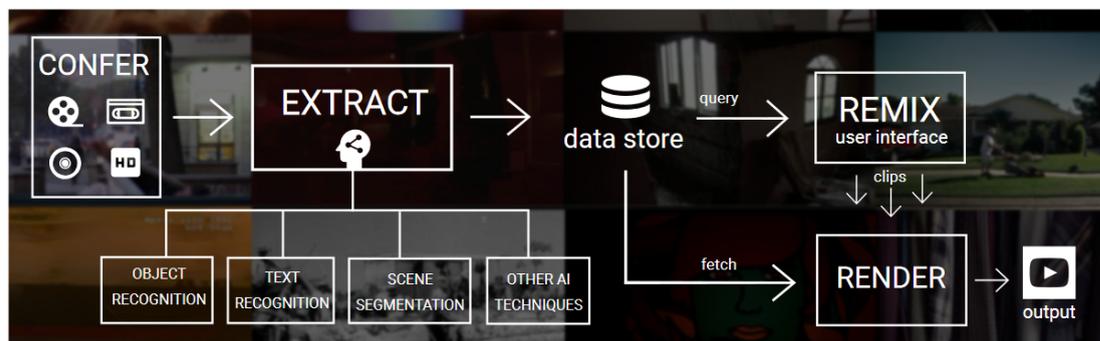


Figura 1. Illustrazione del funzionamento completo di PH-Remix<sup>11</sup>

<sup>11</sup> Sito web Laboratorio di Cultura Digitale - Poster

Nell'archivio del progetto confluiscono dati di due origini differenti: metadattazione tradizionale, attraverso l'importazione di schede catalografiche, realizzate nell'ambito del parallelo progetto di digitalizzazione dell'archivio; estrazione automatica di *clip* significative di durata variabile, tramite l'impiego di intelligenza artificiale. Queste sono estratte dai film automaticamente in base a tre tipi di contenuti:

- ▶ gli 'oggetti' contenuti nella scena,
  - ▷ tramite riconoscimento di oggetti,
  - ▷ ciò che per l'intelligenza artificiale è un 'oggetto riconoscibile' non è limitato agli oggetti inanimati: l'algoritmo può riconoscere anche persone, animali, parti del volto, ecc.
- ▶ le parole contenute nei sottotitoli,
  - ▷ tramite l'estrazione mediante OCR<sup>12</sup>,
- ▶ il colore dominante nella scena,
  - ▷ in base all'etichettatura delle inquadrature tramite la segmentazione delle scene.

Al momento della stesura di questo documento, nel database della piattaforma vi sono più di 300 film e documentari, provenienti da 67 differenti nazioni, per un ammontare di 316 ore totali di contenuti. A partire da questi sono state ottenute più di 800.000 *clip* utilizzabili per creare un numero all'atto pratico infinito di remix. Questo grande numero di segmenti è facilmente navigabile tramite un motore di ricerca integrato nella piattaforma: gli algoritmi di IA differenziano 489 oggetti e 16 colori dominanti, per un totale di 505 possibili chiavi di ricerca. Combinate alla ricerca per testo libero, che restituisce i segmenti con il testo corrispondente in sottotitolo, vi sono innumerevoli possibilità di ricerca. La ricerca può inoltre essere ristretta tramite quattro filtri: antroponomi (registi, produttori, attori), toponimi, titolo, paese di produzione. Essi limitano la ricerca delle *clip* all'interno dei film che contengono uno o più valori associati a questi filtri

---

<sup>12</sup> I sistemi di Optical Character Recognition, ovvero 'riconoscimento ottico dei caratteri', permettono di rilevare il testo contenuto in un documento e di trasferirlo in testo digitale leggibile da una macchina.

Come mostrato in figura 2, alla versione corrente (beta), la piattaforma si presenta lato client come un sito web *one-page*. L'interfaccia è composta da tre elementi principali: a sinistra, uno spazio per l'anteprima di clip o del remix in lavorazione; a destra, la barra di ricerca clip e l'elenco dei risultati; in basso, le tracce audio e video. Cliccando invece sul logo in alto a sinistra comparirà in sovrapposizione una guida dettagliata all'utilizzo della piattaforma.

Questa relazione tratterà nello specifico la mia esperienza di tirocinio presso il LabCD e lo sviluppo del conseguente progetto di tesi.

Durante il tirocinio mi sono occupata dell'implementazione del modulo di esportazione video. Si tratta di un lavoro svolto esclusivamente in *back-end* e lato server: il codice viene invocato al momento della richiesta, effettuata dall'utente, di esportazione del prodotto creato. Fino a quel momento il video è visualizzabile solo tramite un'anteprima; il processo di esportazione genera invece il video completo, e lo rende disponibile al download. Questo non include solamente la sequenza di *clip* così come visualizzate in anteprima, bensì è corredato da informazioni aggiuntive riguardanti il progetto PH-Remix, i produttori dei film, o maggiori informazioni catalografiche riguardo ciascuna *clip*.

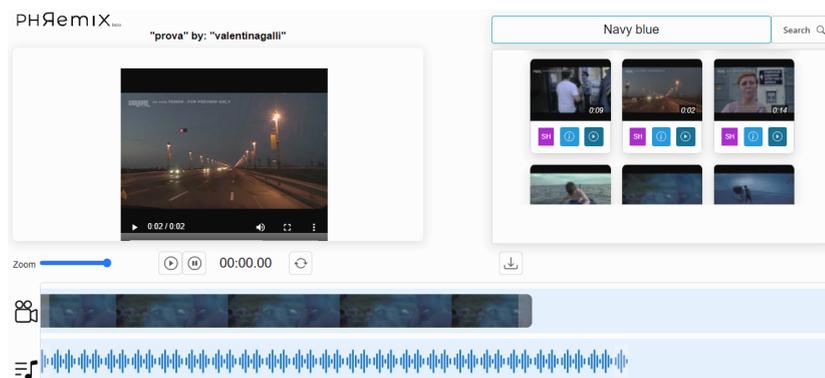


Figura 2. Schermata della piattaforma PH-Remix (beta) lato client.

La figura mostra una ricerca del colore 'navy blue'.

## 2. Illustrazione linguaggi e librerie utilizzati

Prima di proseguire con la presentazione del codice sviluppato per l'implementazione del modulo di esportazione, è necessario presentare i linguaggi e le librerie utilizzati per lo sviluppo.

### 2.1 Node.js

Node.js è un *run-time system* JavaScript guidato da eventi asincroni.<sup>13</sup> Esso è progettato per creare applicazioni di rete scalabili, ovvero in grado di adattarsi a più contesti in base alle necessità. In particolare, Node.js viene utilizzato perché consente di eseguire il codice JavaScript lato server e creare applicazioni web basate sull'interazione in tempo reale.

Node.js permette inoltre l'importazione di librerie JavaScript, che agevolano la programmazione rendendo possibile anche la realizzazione di operazioni complesse. Per lo sviluppo del modulo di esportazione del video remix sono state utilizzate due librerie fondamentali: *fluent-ffmpeg* e *ffmpeg-concat*, entrambe con l'obiettivo di semplificare l'utilizzo della *suite* FFmpeg.

### 2.2 FFmpeg

FFmpeg è una *suite software* completa sostenuta dalla libreria di codifica *libavcodec*<sup>14</sup>. Essa permette di registrare, convertire e riprodurre file audio e video.<sup>15</sup> Alla sua base troviamo lo strumento da linea di comando `ffmpeg` che permette l'accesso alla libreria, corredato da numerose opzioni che specificano le necessità dell'utente. In particolare, l'opzione fondamentale è `-i`, che specifica un numero qualsiasi di file in input, mentre tutto ciò che nel comando non può essere interpretato come un'opzione viene considerato un *path* di output.

---

<sup>13</sup> Wikipedia, voce *Node.js*

<sup>14</sup> *libavcodec* è una libreria di codec gratuita e open source per la codifica e decodifica di dati audio e video, scritta in C. Diversi lettori multimediali, come ad esempio VLC, lo utilizzano come motore di decodifica integrato che consente la riproduzione di molti formati audio e video.

<sup>15</sup> Wikipedia, voce *FFmpeg*

Il processo di transcodifica può essere così descritto:

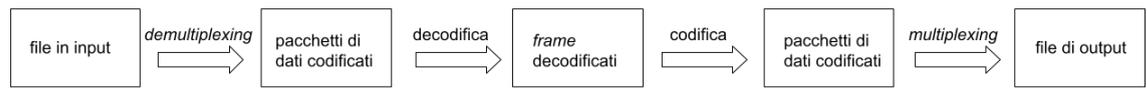


Figura 3. Illustrazione del processo di transcodifica attuato da FFmpeg

Prima della codifica, FFmpeg può processare i *frame* in traccia audio o video attraverso diversi filtri appartenenti alla libreria `libavfilter`<sup>16</sup>. I filtri possono essere semplici o complessi, a seconda del numero di input richiesti per l'applicazione del filtro: un esempio di filtro complesso è *overlay*, che richiede in input il file video originale e ciò che si vuole sovrapporre ad esso.

Le opzioni per includere i filtri in riga di comando sono `-filter` e `-filter_complex`. FFmpeg consente anche di controllare manualmente la selezione delle tracce tramite l'opzione `-map`, accompagnata dalle opzioni `-vn`, `-an`, `-sn` e `-dn`, che possono essere usate per ignorare rispettivamente le tracce video, audio, sottotitoli e *data*.

Tutte queste informazioni vanno fornite a FFmpeg nel terminale, risultando in comandi complessi, come ad esempio:

```
ffmpeg -i input -i logo -filter_complex  
'overlay=10:main_h-overlay_h-10' output
```

Questo comando inserisce come *overlay* un'immagine nell'angolo in basso a destra del video in input.

Un'opzione fondamentale per il progetto in questione è `-filter 'drawtext...'` (approfondimento al paragrafo 3.5).<sup>17</sup>

Come detto in precedenza, per la scrittura dello *script* di esportazione del remix si è ricorso a due librerie, `fluent-ffmpeg` e `ffmpeg-concat`.

## 2.3 fluent-ffmpeg

La libreria `fluent-ffmpeg` si pone come obiettivo quello di astrarre il complesso utilizzo da riga di comando di `ffmpeg` per renderlo, appunto, fluido e leggibile.

Sfruttando i metodi della libreria, è possibile eseguire comandi FFmpeg.<sup>18</sup>

<sup>16</sup> In modo equivalente, `libavfilter` è una libreria che fornisce un framework generico di filtraggio audio/video. Essa permette di migliorare e modificare filtri.

<sup>17</sup> Sito web FFmpeg documentation

<sup>18</sup> Sito web `fluent-ffmpeg` npm

Portando lo stesso esempio, per inserire un'immagine in sovrainpressione, invece di scrivere un unico comando in *console* si scriverà:

```
FFmpeg({ source: './input.mp4' })  
  .addInput('./logo.jpg')  
  .withVideoFilter('overlay=10:main_h-overlay_h-10')  
  .saveToFile('./output.mp4')
```

Altri metodi della classe `FFmpeg()` importanti per il progetto sono:

- ▶ `.setStartTime(...)`:
  - ▷ permette di saltare direttamente al marcatore temporale indicato.
- ▶ `.setDuration(...)`:
  - ▷ interrompe la traccia al marcatore temporale indicato.
- ▶ `.withNoVideo()`:
  - ▷ salta la traccia video.

## 2.4 ffmpeg-concat

La libreria `ffmpeg-concat` permette di concatenare video tramite `FFmpeg` in modo intuitivo, gestendo in modo semplice le transizioni, altrimenti estremamente difficili da realizzare.<sup>19</sup> A questo scopo, `ffmpeg-concat` adotta la collezione di transizioni di `GL Transitions`, un'iniziativa open source che utilizza il linguaggio `GLSL` per creare una raccolta universale di transizioni. Attualmente essa contiene 67 transizioni rilasciate con licenza gratuita da molti contributori.<sup>20</sup>

La funzione base della libreria è così strutturata:

```
concat ({  
  output: './output.mp4',  
  videos: [ './input0.mp4',  
            './input1.mp4',  
            './input2.mp4' ],  
  audio: './audio.mp3',  
  transition: { name: 'fade',  
                duration: 500 }  
})
```

---

<sup>19</sup> Sito web `ffmpeg-concat` - npm

<sup>20</sup> Sito web `GL transitions`

Ci sono diversi elementi da notare immediatamente:

- ▶ Il parametro `videos` è un array di *paths*.
- ▶ Il parametro `audio` ammette un solo *path*, rigorosamente in mp3:
  - ▷ se il parametro non è fornito, ffmpeg-concat presume che si voglia concatenare la traccia audio dei video in parametro `videos`.
- ▶ Il parametro `transition` vede come alternativa `transitions`, nel quale si possono specificare diverse transizioni per ogni nuovo video concatenato.
- ▶ A differenza di fluent-ffmpeg, ffmpeg-concat prevede una struttura ben definita da rispettare ogni volta che si vuole concatenare un nuovo video.

### 3. Implementazione dell'esportazione video

Come mostrato precedentemente in 1.3, l'architettura della piattaforma è basata su microservizi. In figura 1 è possibile osservare che il servizio di esportazione è l'ultimo, in quanto restituisce all'utente il prodotto finale della piattaforma.

#### 3.1 I dati in input

A partire dall'interazione dell'utente con l'interfaccia di remix sono generati diversi dati di input, contenenti le informazioni per la creazione del remix in formato .mp4 e necessari allo *script* di esportazione. Tali informazioni sono memorizzate all'interno di un dizionario JSON. Una chiamata alla API, lanciata tramite il pulsante *export* dell'interfaccia, avvia lo *script* di esportazione ed invia al server il dizionario.

Il dizionario ricevuto in input è così strutturato:

```
{
  "title": "..",
  "author": "..",
  "segments": [ {..},{..},{..},{..} ]
}
```

Il primo campo `"title"` contiene il titolo scelto dall'utente per il proprio remix.

Il campo `"author"` contiene l'*username* scelto dall'utente.

L'array `"segments"` contiene a sua volta un dizionario per ogni *clip* inclusa nel remix.

Ciascun elemento dell'array possiede tutte le informazioni necessarie circa quella *clip*, tra cui:

- ▶ `timeStart` e `timeEnd`,
  - ▷ marcatori temporali del primo e dell'ultimo *frame* della *clip* rispetto al remix finale;
- ▶ `duration` e `originalSize`,
  - ▷ valori espressi in secondi circa la durata della *clip* nel remix finale e la sua durata originale nel caso sia stata ritagliata;
- ▶ `trackRef`,
  - ▷ che può assumere due valori, `videoTrack1` o `musicTrack1`, a seconda della traccia a cui appartiene la *clip*;

- ▶ `videoId`,
  - ▷ valore numerico che identifica il film da cui proviene il segmento all'interno del database;
- ▶ `segmentId`,
  - ▷ valore numerico che identifica lo specifico segmento all'interno del film.

### 3.2 Architettura dello *script* di esportazione

Una volta ricevuti in input i dati necessari all'esportazione, lo *script* svolge una sequenza di operazioni ordinate.

Per iniziare l'esportazione, vengono ritagliate tutte le *clip* secondo le indicazioni dell'utente e viene registrato quali appartengono alla traccia audio e quali alla traccia video.

Nel caso in cui la traccia audio non sia vuota, si procede con la concatenazione di `videoPerAudio`. Si tratta di un file .mp4 temporaneo generato dalla concatenazione dei segmenti selezionati per la traccia audio. Esso viene poi convertito in .mp3 e va a costituire il file in input audio nello step di concatenazione del video completo.

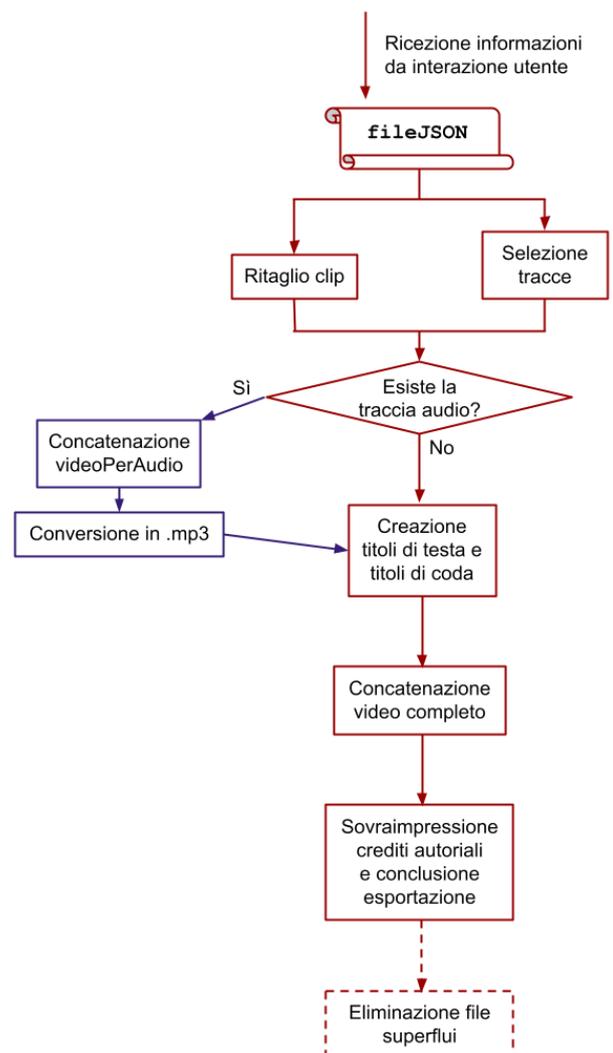


Figura 4. Schema semplificato del funzionamento del codice di esportazione.

Questa operazione è opzionale, in quanto nel caso in cui la traccia video risultasse vuota, lo *script* proseguirebbe con lo step successivo, ovvero la creazione di titoli di testa e titoli di coda contenenti i crediti autoriali.

Segue poi la concatenazione del video finale, che corrisponde alla sequenza: titoli di testa, *clip* scelte per la traccia video, titoli di coda, eventualmente con traccia audio.

Infine, l'esportazione si conclude con la sovrapposizione dei dati circa il film di provenienza su ogni *clip*.

Al termine dell'operazione, i file temporanei creati durante l'esportazione vengono eliminati per risparmiare lo spazio nel server.

### 3.3 Tracce video e audio

La natura stessa del video remix (si veda la descrizione nel par. 1.2) comporta che all'autore del remix sia fornita la possibilità di eseguire alcune operazioni fondamentali nel montaggio dei propri video: ogni segmento disponibile nella piattaforma deve poter essere recuperato, visualizzato in anteprima, e selezionato per la traccia audio o video (o entrambe). In aggiunta, deve essere fornita la possibilità di ritagliare entrambi i tipi di segmenti a proprio piacimento (si veda par. 3.4).

Se lato client l'applicazione di remix è stata sviluppata per consentire le operazioni sopra descritte, l'obiettivo primario dello *script* lato server è quello di selezionare i segmenti scelti dall'utente, estrarli e concatenarli sia in traccia audio che video, restituendo il video completo pronto per il download.

#### 3.3.1 Illustrazione codice

```
async function videoEsportazione (fileJson) {  
  [...]  
  var datiVideo = []  
  var datiAudio = []  
  [...]  
  var j = 0  
  var k = 0  
  for (var i = 0; i < fileJson["segments"].length; i++) {  
    if (fileJson["segments"][i]["trackRef"] == 'musicTrack1') {  
      datiAudio[j] = fileJson["segments"][i]  
      j++  
    } else if (fileJson["segments"][i]["trackRef"] == 'videoTrack1') {  
      datiVideo[k] = fileJson["segments"][i]  
      k++  
    }  
  }  
}
```

```

    } else {
      console.log("error")
    }
  }
  [...] }

```

① La distribuzione delle clip nelle rispettive tracce si svolge interamente all'interno del ciclo *for* eseguito dalla funzione **videoEsportazione**:

- ▶ Il codice scorre il dizionario `fileJson` alla ricerca della chiave `trackRef`, che differenzia le *clip* inserite nella traccia audio da quelle inserite nella traccia video.
- ▶ A seconda del valore della chiave `trackRef` le informazioni riguardo quel segmento saranno copiate rispettivamente nell'array `datiVideo` o `datiAudio`, in quanto a seconda della loro funzione i segmenti subiscono trattamenti diversi.

Come visto in precedenza in 2.4, `ffmpeg-concat` non permette di concatenare tracce audio, ma al contrario pretende che il parametro corrisponda ad un unico file `.mp3`. Questa attitudine essenzialmente contraddittoria rispetto alle esigenze degli autori di `remix` (v. 3.1) ha richiesto una programmazione ulteriore che aggirasse il problema.

La soluzione individuata è la seguente:

```

async function createVideo(params, fileJson) {
  try {
    if (params.arrayPathsAudio.length > 0) {
      await concat({
        output: params.videoPerAudio,
        videos: params.audioString.split(','),
        transition: {
          name: "fade",
          duration: 0
        }
      })
      await Promise.all([makeAudio(params.videoPerAudio,
        params.audio_path), [...] ])
    }
    await concat({
      output: params.videoConcat,
      videos: params.videoString.split(','),
      audio: params.audio_path,
      transition: {
        name: "fade",
        duration: 0
      }
    })
  }
}

```

①

②

③

④

```

    }
  })
  [...]
}

function makeAudio(videoPerAudio, audio) {
  return new Promise((resolve, reject) => {
    FFmpeg({ source: videoPerAudio })
      .withNoVideo()
      .saveToFile(audio).on('end', resolve).on('error', reject)
  })
}

```

① Questa porzione di codice suddivide la funzione principale in due opzioni alternative: se la traccia audio è presente, proseguirà con il codice ora in analisi; altrimenti, passerà direttamente alla concatenazione della traccia video senza alcuna traccia audio. Di seguito è preso in considerazione il primo caso in quanto pertinente all'analisi dell'implementazione della traccia audio.

② Per procedere alla creazione della singola traccia audio da sottomettere a FFmpeg per la concatenazione finale, vengono prima concatenati i video i cui *path* si trovano in un array. La traccia audio non viene specificata, così che il video risultante, denominato in variabile `videoPerAudio`, sia la concatenazione dei segmenti selezionati per la traccia audio.

③ Il passaggio successivo consiste nel convertire il video ottenuto in una traccia .mp3. Prima di proseguire, viene invocata la funzione `makeAudio`. La conversione avviene attraverso `fluent-ffmpeg` grazie all'opzione `.withNoVideo()`. In questo modo si ottiene una singola traccia .mp3 a partire da uno o più segmenti selezionati dall'utente come traccia audio.

④ Una volta ottenuta la traccia .mp3 a partire da `videoPerAudio`, è semplice generare il file finale: vengono passati a `ffmpeg-concat` l'array `params.videoString` contenente i *path* dei segmenti selezionati per il video e la traccia audio appena realizzata (`params.audio_path`).

Come visto al paragrafo 2.4, `ffmpeg-concat` richiede che venga specificato il parametro `transition`, scelto dalla libreria `GL Transitions`. È importante notare che ogni volta che `ffmpeg-concat` viene utilizzato in questo *script*, il parametro `transition` presenta la `duration` uguale a `0`. Durante lo sviluppo è stato scelto di non

usare alcuna transizione tra le *clip*, per rendere il risultato finale più aderente possibile al video che l'utente può visualizzare in anteprima. Inoltre, transizioni troppo invasive avrebbero rischiato di oscurare clip di durata inferiore al secondo. Ciò non implica che le transizioni non possano essere integrate nel prototipo in una fase successiva dello sviluppo (approfondimento al capitolo 5).

Va sottolineato il ruolo fondamentale dell'operatore `await`. Come osservabile in figura 4, le operazioni di concatenazione devono rispettare un ordine sequenziale. Per questo motivo la maggior parte delle funzioni sono asincrone o impiegano le *promises*.

### 3.4 Ritaglio

Come spiegato in precedenza, un obiettivo dello *script* era permettere il ritaglio delle clip secondo le indicazioni dell'utente. Nell'interfaccia, una volta scelta e posizionata la *clip* sulla traccia desiderata, essa può essere ritagliata trascinandone l'inizio o la fine. Durante questa operazione viene mostrato un indicatore di tempo che segna dinamicamente il punto di inizio e fine della *clip* all'interno del remix. Come mostrato in figura 5, al termine di questo procedimento appariranno delle linee azzurre ad indicare dove è stato effettuato il ritaglio.

La selezione di un ritaglio all'inizio o alla fine di una *clip* viene memorizzata nel dizionario JSON che verrà poi passato in input per l'esportazione: rispettivamente vengono modificati i valori di `resizedTimeStart` e di `duration`.



Figura 5. Traccia video con una clip, prima e dopo aver effettuato due operazioni di ritaglio all'inizio e alla fine.

#### 3.4.1 Illustrazione codice

```
function cutVids(fileJson, dir_name) {  
  var start = 0  
  var duration = 0  
  let array = []
```

```

for (var i = 0; i < fileJson["segments"].length; i++) { ①
  array.push(
    new Promise((resolve, reject) => {
      start = fileJson["segments"][i]["resizedTimeStart"] / 100 ②
      duration = fileJson["segments"][i]["duration"] / 100 ②
      FFmpeg({ source: '/data/segments/' +
fileJson["segments"][i]["videoId"] + '_' +
fileJson["segments"][i]["segmentId"] + '.mp4' }) ③
        .setStartTime(start)
        .setDuration(duration)
        .saveToFile(dir_name + "/" +
fileJson["segments"][i]["videoId"] + '_' +
fileJson["segments"][i]["segmentId"] + '_cut.mp4').on('end',
resolve).on('error', reject) ④
      })
    )
  }
  return array
}

```

① Il ritaglio dei segmenti secondo la preferenza dell'utente avviene all'interno della funzione `cutVids`. Innanzitutto, si seleziona all'interno del dizionario JSON ricevuto dall'API l'array `segments` contenente le informazioni riguardanti i segmenti scelti.

Tutti i segmenti, ritagliati o meno, vengono passati a questa funzione: se i valori di `["segments"][i]["resizedTimeStart"]` e `["segments"][i]["duration"]` corrispondono a quelli di `["segments"][i]["timeStart"]` e `["segments"][i]["originalSize"]`, la *clip* viene copiata nella cartella temporanea senza essere ritagliata.

② Le informazioni riguardanti il ritaglio della parte iniziale (`["resizedTimeStart"]`) e finale (`["duration"]`) della *clip* sono espressi in centesimi di secondo. Prima di essere assegnati alle variabili `start` e `duration`, i valori vengono convertiti in secondi.

③ Attraverso il ciclo *for* vengono presi in input tutti i segmenti facenti parte del remix: tramite le opzioni `.setStartTime` e `.setDuration` si assegnano i valori di ritaglio selezionati dall'utente.

④ Tutti i nuovi file vengono salvati in una cartella temporanea il cui *path* è definito in `.saveToFile` e saranno eliminati al termine dell'esportazione.

### 3.5 Titoli di testa e coda con crediti autoriali

Seguendo gli obiettivi della *public history*, requisito fondamentale è che sia indicata la provenienza di ogni singola *clip* presente sulla piattaforma e che tali informazioni siano inserite anche all'interno del remix esportato. Al momento della stesura della presente relazione, tutti i film e documentari utilizzabili appartengono all'archivio del Festival dei Popoli, ma potenzialmente il progetto potrebbe essere esteso alla collaborazione con altri enti, ottenendo così una grande varietà di risorse video provenienti da archivi diversi. Per questo motivo, nello *script* di esportazione è stata integrata la creazione automatica di titoli di testa e titoli di coda con informazioni che permettono di risalire ai film dell'archivio.

#### 3.5.1 L'importanza dei crediti e la questione legale

Oltre ad essere un dovere rispetto alla metodologia della *public history*, l'inclusione di crediti autoriali risulta necessaria anche sul piano legislativo. Come anticipato in precedenza, la *remix culture* si pone in contrapposizione rispetto alla tutela del diritto d'autore<sup>21</sup> e al principio di integrità dell'opera<sup>22</sup>: da un punto di vista legislativo il remix video è difficile da inquadrare, poiché esso si basa sull'operazione di campionamento e utilizza costantemente materiale altrui, sia video che sonoro. Allo stesso tempo, la pratica del remix rappresenta un esercizio del diritto di libertà di espressione artistica e critica. La libertà di espressione è un diritto fondamentale riconosciuto da tutte le democrazie e come tale deve essere tutelato.

Come indicato nel documento redatto dallo studio legale Mustilli che affianca il progetto nell'elaborazione di soluzioni per la diffusione dei contenuti dell'archivio, “sebbene agire sulla base di liberatore parrebbe la via più sicura e priva di rischi legali, nella pratica ottenerle sarebbe più complesso di quanto potrebbe sembrare”, in

---

<sup>21</sup> Il diritto d'autore è disciplinato dalla legge 22 aprile 1941, n. 633. Non è necessario alcun tipo di registrazione dell'opera per godere della tutela del diritto d'autore (art. 106). La SIAE si occupa di tenere due specifici registri, uno dedicato alle opere cinematografiche, l'altro ai software. Per Art. 8 della l. 633/41, si identifica come soggetto del diritto d'autore la persona fisica che ha creato l'opera. Per le opere cinematografiche, i diritti patrimoniali durano per 70 anni dalla morte dei coautori e titolari dei diritti: direttore artistico, autore della sceneggiatura, autore dei dialoghi, autore della musica (Art. 32). I diritti patrimoniali spettano a chi ha organizzato la produzione dell'opera (il produttore), che non può però effettuare trasformazioni dell'opera salvo le modifiche necessarie per l'adattamento cinematografico (Art. 45). Per approfondire: Seminario di Cultura Digitale “Fruizione dei beni immateriali” con Caterina Murgio.

<sup>22</sup> “L'autore ha il diritto di opporsi a qualsiasi deformazione o modifica dell'opera che possa danneggiare la sua reputazione: chi crea un'opera ha il diritto di essere giudicato dal pubblico per l'opera così come l'ha concepita [...]” Per approfondire: Diritto all'integrità dell'opera - Legge sul diritto d'autore - SIAE

quanto i detentori dei diritti potrebbero cambiare nel tempo, o potrebbero crearsi situazioni di stallo.<sup>23</sup> Se la soluzione basata sulla richiesta di liberatorie ai detentori dei diritti risulta poco percorribile, notevoli aperture sono invece fornite dalle modifiche apportate dal Decreto legislativo 8 novembre 2021<sup>24</sup>, n. 177 alla legge 22 aprile 1941, n. 633, con le quali l'Italia ha recepito la Direttiva 2019/790<sup>25</sup> del Parlamento europeo sul Copyright.

Va specificato che in nessun caso l'utente potrà visionare film e documentari per intero tramite la piattaforma. Inoltre, le *clip* non sono messe a disposizione per uso commerciale, ma solo ed esclusivamente per uso personale o didattico.

Con l'entrata in vigore del provvedimento alla legge 22 aprile 1941, n. 633 è stato inserito l'Art. 70-bis, il cui testo recita:

“1. Sono liberi il riassunto, la citazione, la riproduzione, la traduzione e l'adattamento di brani o di parti di opere e di altri materiali e la loro comunicazione al pubblico se effettuati con mezzi digitali, esclusivamente per finalità illustrative ad uso didattico, nei limiti di quanto giustificato dallo scopo non commerciale perseguito, nonché sotto la responsabilità di un istituto di istruzione, nei suoi locali o in altro luogo o in un ambiente elettronico sicuro, accessibili solo al personale docente di tale istituto e agli alunni o studenti iscritti al corso di studi in cui le opere o gli altri materiali sono utilizzati. 2. Il riassunto, la citazione e la riproduzione di brani o di parti di opere e di altri materiali e la loro comunicazione al pubblico sono sempre accompagnati dalla menzione del titolo dell'opera, dei nomi dell'autore, dell'editore e del traduttore, qualora tali indicazioni figurino sull'opera”.

In questo testo viene riconosciuta la legittimazione della creazione di *clip* da mettere a disposizione per uso didattico, se accompagnate dalla menzione del titolo, autore, eventuali editori e traduttori. Viene però sottolineata anche la limitazione di accesso al materiale ai soli docenti e studenti di un determinato istituto di istruzione.<sup>26</sup>

---

<sup>23</sup> Memorandum Recap della Call con Fondazione Sistema Toscana e Festival dei Popoli e considerazioni integrative al parere datato 28 luglio 2021, avvocati Marcello Mustilli e Matteo Roiz

<sup>24</sup> Decreto legislativo 8 novembre 2021, n. 177

<sup>25</sup> Direttiva (UE) 2019/790 del Parlamento Europeo e del Consiglio del 17 aprile 2019

<sup>26</sup> Decreto legislativo 8 novembre 2021, n. 177

Nonostante le aperture rispetto al passato, il testo della legge continua ad essere in contrapposizione sia con l'idea di realizzare un progetto di *public history*, pensato per un pubblico non necessariamente accademico, sia con i principi della *remix culture*, che si basa sul riutilizzo e la condivisione del materiale senza porsi limiti relativi alle modalità di accesso e di condivisione dei prodotti del remix.

La piattaforma di PH-Remix è stata modellata sulla base delle possibilità fornite dalla legge. Al momento, come verrà illustrato nel capitolo 4, essa è aperta ad un ristretto gruppo di utenti per i test sulla piattaforma beta. Il sito è protetto da una password, e per accedere alla piattaforma di remix è necessario inserire la propria e-mail.

Al momento del download l'utente accetta implicitamente i termini d'uso, che sono consultabili nella pagina di accesso, nella guida, e dopo l'esportazione. Lo sviluppo sta procedendo per includere un'accettazione più esplicita che obblighi l'utente a scorrere i termini d'uso per poter continuare.

Inoltre, come anticipato, durante l'esportazione vengono generati automaticamente dei titoli di coda che contengono i riferimenti al film di provenienza da cui è stata estratta ciascuna *clip*.<sup>27</sup> Di seguito è illustrato il codice che permette la realizzazione dei titoli di testa e coda.

### 3.5.2 Illustrazione codice

```
function makeTitle(testoTitolo, lunghBase, videoTitolo) {
  return new Promise((resolve, reject) => {
    FFmpeg({ source: 'files_export/intro.mp4' })
      .addInput('files_export/audio.mp3')
      .withVideoFilter("drawtext=text='" + testoTitolo + "':
fontSize=23: fontcolor=white: x=(w-text_w)/2:y=(h-text_h)/2:
enable='between(t,0," + (lunghBase + 2) + ")'")
      .saveToFile(videoTitolo).on('end', resolve).on('error', reject)
  })
}
[...]
```

```
var testoTitolo = "\"" + titoloRemix + "\",\n\ncreato da " + nomeAutore
+ "\n\n" + getDate()
```

---

<sup>27</sup> Per ciascuna vengono indicati titolo, regista, paese di produzione, anno di produzione e locazione in archivio. Questi dati possono essere ampliati - ad esempio per includere produttore, distributore - una volta aggiunti i campi desiderati nella scheda in archivio del film.

La creazione dei titoli di testa avviene tramite la funzione `makeTitle`. Al suo interno viene invocato `fluent-ffmpeg`, che prende in input un video pre-esistente (`files_export/intro.mp4`) e aggiunge in sovrapposizione il valore della variabile `testoTitolo`, ovvero `"title"` e `"author"` (rispettivamente nelle variabili `titoloRemix` e `nomeAutore`), insieme alla data corrente (`getDate()`).

La sovrapposizione del testo è realizzata tramite l'opzione `drawtext` di `FFmpeg`, che richiede diversi parametri che vanno ad influenzare l'aspetto estetico del blocco testuale e la sua durata.

```
function makeCredits(testoCredits, lunghCred, videoCredits) {
  return new Promise((resolve, reject) => {
    FFmpeg({ source: 'files_export/sfondo_credits.mp4' })
      .addInput('files_export/audio.mp3')
      .setDuration(lunghCred + 2)
      .withVideoFilter("drawtext=text='" + testoCredits① + "':
fontSize=20: fontcolor=white: x=(w-text_w)/2:y=h-100*t:
enable='between(t,0,'" + (lunghCred + 2) + "')")
      .saveToFile(videoCredits).on('end', resolve).on('error', reject)
  })
}
[...]
```

```
if (video_params.arrayPathsAudio.length > 0) {
  testoCredits① = "\nTracce video \n" + creditsVideo②.join(" ") +
"\n\n\nTracce audio\n" + creditsAudio②.join(" ")
} else {
  testoCredits① = "\nTracce video \n" + creditsVideo②.join(" ")
}
[...]
```

```
url = "http://remix.unipi.it:3000/api/0.1/film_info/" +
datiVideo[i]["videoId"]
let value = await axios.post(url)
creditsVideo②.push("\n\n" + (i + 1) + ". " + value.data.phx_title + ",
" + value.data.phx_director + "\n" + value.data.phx_country + ", " +
value.data.phx_year + "\n" + value.data["Archivio di conservazione"] + "
(" + value.data.Inventario + ")")
[...]
```

```
url = "http://remix.unipi.it:3000/api/0.1/film_info/" +
datiAudio[i]["videoId"]
let value = await axios.post(url)
creditsAudio②.push("\n\n" + (i + 1) + ". " + value.data.phx_title + ",
" + value.data.phx_director + "\n" + value.data.phx_country + ", " +
value.data.phx_year + "\n" + value.data["Archivio di conservazione"] + "
(" + value.data.Inventario + ")")
```

La creazione di titoli di coda avviene tramite la funzione `makeCredits` in modo pressoché uguale ai titoli di testa, fatta eccezione per il testo in sovraimpressione, che invece di essere statico scorre dal basso verso l'alto e include crediti autoriali per ogni segmento usato.

① `testoCredits` è la stringa stampata in sovraimpressione. A seconda della presenza o meno della traccia audio, essa può assumere due valori, come indicato nel costrutto *if-else*: nel primo caso la stringa contiene le informazioni degli array `creditiVideo` e `creditiAudio`, nel secondo caso solo dell'array `creditiVideo`.

② `creditiVideo` e `creditiAudio` sono array di stringhe contenenti le informazioni autoriali di ogni segmento. Queste informazioni sono ricavate da metadati, ovvero dati che i segmenti ereditano dalle schede del film genitore.

`value.data.phx_title` (titolo), `value.data.phx_director` (regista), `value.data.phx_country` (paese di produzione), `value.data.phx_year` (anno di produzione), `value.data["Archivio di conservazione"]` (archivio di conservazione del film) e `value.data.Inventario` (numero di inventario nell'archivio) sono tutti dati ricavati grazie ad una chiamata alla API realizzata tramite Axios<sup>28</sup>.



Figura 6. Esempio di titoli di testa e titoli di coda di un remix realizzato tramite la piattaforma.

<sup>28</sup> Axios è un client HTTP basato su promesse per il browser e node.js. Per più informazioni visitare il sito [axios-http.com](https://www.axios-http.com)

## 4. *Beta testing* della piattaforma

Lo sviluppo del microservizio di *export* ha consentito il completamento del prototipo, che si avvia ora alla fase finale, ovvero il *testing*. Si tratta di un lavoro ancora in corso, che sta avvenendo contemporaneamente alla stesura di questa relazione. Lo stato attuale del prototipo PH-Remix con i dati dell'archivio del Festival dei Popoli rappresenta un caso di studio, tramite il *beta testing* si potranno effettuare valutazioni e analizzare i risultati ottenuti. La collaborazione con l'archivio FdP rappresenta un mezzo per mettere alla prova strumenti innovativi che possono essere riadattati per altre e nuove occasioni.

Questa operazione sta coinvolgendo in primo luogo un gruppo di sette persone che hanno lavorato a diverse parti del progetto – assegnisti di ricerca, tirocinanti e tesisti – e due *public historians*. I principali obiettivi in questa fase sono: identificare punti di forza e punti critici; comprendere l'intuitività della piattaforma e l'efficienza della guida; individuare alternative di utilizzo; ricevere opinioni su funzionalità aggiuntive e implementazioni future. Infine, i *tester* sono invitati a fare rapporto di qualsiasi bug incontrino, al fine di facilitare lo sviluppo successivo.

Giovedì 31 marzo 2022 si è tenuta la prima riunione con tutti i *tester*, in cui si illustrava brevemente il progetto, la piattaforma, e gli obiettivi dei test. È stato richiesto a tutti di conservare i remix più significativi per condividerli alla fine dei test come esempio delle potenzialità della piattaforma. Al termine delle due settimane di *testing*, si è tenuta una seconda riunione dove tutti hanno potuto esprimere i loro risultati e le loro opinioni.

Questo gruppo test, oltre agli obiettivi già ben elencati, ha lo scopo di produrre materiale per la candidatura ad AIPH award. Si tratta di un concorso che punta a mettere in risalto i migliori progetti di Public History realizzati fra il 1° gennaio 2021 e il 25 aprile 2022.<sup>29</sup> La candidatura ha richiesto la realizzazione di una pagina web che documentasse il progetto e le attività svolte.<sup>30</sup> Ho collaborato direttamente alla realizzazione della pagina<sup>31</sup> e di un'infografica che illustra le informazioni presentate

---

<sup>29</sup> AIPH - Concorso per i migliori progetti di Public History realizzati fra il 1° gennaio 2021 e il 25 aprile 2022

<sup>30</sup> Sito web Laboratorio di Cultura Digitale - AIPH Award

<sup>31</sup> Il sito di LabCD è costruito tramite l'editor a blocchi di Wordpress, e lo stesso è stato utilizzato per la realizzazione della pagina per la candidatura.

al paragrafo 1.3. Inoltre, è in programma la realizzazione di un video riassuntivo della fase di *testing*, con scene tratte dalla registrazione delle riunioni.

**PHRemix**  
 Laboratorio di Cultura Digitale  
 UNIVERSITÀ DI PISA

HOME LABORATORIO PROGETTI INCENTRI FORMAZIONE SUMMER SCHOOL TUTORIAL EVENTI

Home > PH-Remix > Privato: AIPH award

**PRIVATO: AIPH AWARD**

**PUBLIC HISTORY REMIX**  
 La REMIX CULTURE incontra la PUBLIC HISTORY

**PH-REMIX**

Partecipazione  
 > coinvolgimento del pubblico nella creazione dei contenuti  
 Ideo di strumenti digitali  
 > conosciuti dal pubblico per la creazione di prodotti di digital public history

Verifica delle fonti  
 > utilizzo di materiale d'archivio per poter avere opportuna documentazione sulle fonti e relativo rimando ad esse

Condivisione  
 > pubblicazione del remix con modalità open

Per saperne di più:  
 > Gruppo di lavoro  
 > Poster  
 > Remix culture

PH-Remix è un progetto coordinato dal Laboratorio di Cultura Digitale (LabCD) dell'Università di Pisa, in collaborazione con Festival dei Popoli, Mediateca Toscana e Fondazione Sistema Toscana. Esso si pone come finalità la diffusione della storia, e la valorizzazione e rivitalizzazione del materiale d'archivio. PH-Remix permette di riutilizzare il materiale audiovisivo d'archivio per creare nuove narrazioni storiche attraverso la metodologia della public history.

**ANATOMIA DI UN PUBLIC HISTORY REMIX**

Intro con titolo, autore e data del remix  
 Sequenza di clip estratte da diversi film  
 Titoli di coda generati automaticamente con informazioni sui film da cui sono state estratte le clip: titolo, autore, anno, paese, nazione di provenienza e inventario  
 Presenza di didascalie inerenti dall'autore del remix estrapolate dalla clip  
 Ulteriori testi e didascalie in piena schiarita

LABCD  
 LABORATORIO DI CULTURA DIGITALE  
 Centro interdisciplinare di ricerca e formazione  
 Sede amministrativa: Dipartimento di Biologia, Letteratura, Linguistica  
 Piazza Tommaseo, 2 (Palazzo Matteucci)  
 56126 Pisa

CONTATTI  
 Via Colligio Ricci, 10 - 56126 Pisa  
 Tel. 050.2212021  
 info@labcd.unipi.it  
 Assistenza tecnica:  
 Polo 4.812  
 Piazza Matteucci  
 56126 Pisa

SERVIZI E COLLEGAMENTI  
 Dipartimenti afferenti:  
 Filologia Letteraria, Linguistica  
 Civiltà e Forme del Sapere  
 Economia e Management  
 Informatica  
 Scienze Politiche

SEGUICI  
 Facebook  
 YouTube  
 News

Figura 7. Schermata della pagina web realizzata per la candidatura ad AIPH award.

**PHRemix** dati numerici del prototipo di piattaforma, raccolti in data: 12 aprile 2022

300+ FILM SELEZIONATI DALL'ARCHIVIO FdP

316 ORE DI CONTENUTI

67 NAZIONI DI PROVENIENZA

CATALOGAZIONE DEI FILM

116 TEMI NEL SOGGETTARIO

ANTROPONIMI

TOPONIMI

ESTRAZIONE AUTOMATICA DI CLIP TRAMITE ALGORITMI DI INTELLIGENZA ARTIFICIALE

1. RICONOSCIMENTO DI OGGETTI
2. ESTRAZIONE TESTO DA SOTTOTITOLI
3. SEGMENTAZIONE IN SCENE BASATE SU COLORE DOMINANTE

800.000+ CLIP GENERATE

505 ELEMENTI DI RICERCA

FILTRI DI RICERCA SU FILM:  
 TITOLO, ANTRONIMI, TOPONIMI, PAESE DI PRODUZIONE

RICERCA TRAMITE TESTO NEI SOTTOTITOLI

COMBINAZIONI DI RICERCA

Figura 8. Particolare dell'infografica realizzata per la candidatura ad AIPH award.

## 5. Conclusioni

PH-Remix è un progetto estremamente innovativo, strettamente legato ai temi e agli ambiti dell'Informatica Umanistica, in quanto unisce *public history* e *remix culture* in un unico grande progetto di programmazione. Quest'ultima non si limita semplicemente alla mera scrittura di codice, ma presuppone in principio conoscenze sia umanistiche, per potersi meglio avvicinare ai contenuti e alle necessità della *public history*, sia informatiche, per comprendere le possibilità e i limiti di ogni sua componente.

Essendo un progetto di *public history*, ritengo personalmente che le restrizioni imposte dalla legge italiana sul diritto d'autore limitino le potenzialità di PH-Remix, che potrebbe andare ben oltre l'ambito accademico rivolgendosi direttamente a comunità di diversa estrazione. Nonostante sia stata recentemente aggiornata, la legge è ancora lontana dall'includere e considerare tutte le realtà di condivisione di media digitali, che sono invece in continuo sviluppo.

PH-Remix è ancora un prototipo, la cui struttura permette un gran numero di implementazioni future, molte delle quali necessiterebbero un conseguente sviluppo anche nello *script* di esportazione. FFmpeg lascia spazio ad un vasto numero di possibilità, in particolare per quanto riguarda gli aspetti più estetici del remix. Si potrebbe, ad esempio, permettere di scegliere il colore dei titoli di testa e coda, o il carattere del testo che compare in sovraimpressione. Una volta implementate più tracce video, l'utente potrebbe usarle per mettere un video in *overlay* ad un altro, con una modalità Picture-in-Picture (PiP).

Per quanto riguarda le transizioni, potrebbero essere implementate in modo simile al ritaglio delle *clip*: aggiungendo nel dizionario JSON dei campi specifici che memorizzano per ogni *clip* quale transizione è stata selezionata e la sua durata, si possono utilizzare quei dati all'interno della funzione `ffmpeg-concat` che controlla le transizioni per personalizzarle.

Infine, durante il periodo di *beta testing* ho avuto modo di sperimentare direttamente con il prototipo. Ritengo che PH-Remix sia uno strumento di grande potenzialità per aprire connessioni tra comunità e generazioni diverse.

## 6. Glossario

- ▶ API, Application Programming Interface, interfaccia di programmazione di applicazioni
  - ▷ Insieme di definizioni e protocolli, atte all'espletamento di un dato compito per la creazione e l'integrazione di software. Le API permettono a prodotti o servizi di comunicare con altri prodotti o servizi senza che sia necessario sapere come questi vengono implementati, semplificando così lo sviluppo di app.<sup>32</sup>
- ▶ Back-end
  - ▷ La parte di un programma informatico che permette il funzionamento delle interazioni utente. Solitamente corrisponde alla parte di gestione di un sito, accessibile solo ad amministratori.<sup>33</sup>
- ▶ Demultiplexing, DEMUX
  - ▷ Processo inverso al multiplexing, consiste nel separare diversi flussi di segnale digitale appartenenti allo stesso segnale. Nel caso di FFmpeg, esso genera pacchetti di informazione video/audio che verranno poi decodificati.
- ▶ Frame, fotogramma
  - ▷ Immagini ferme che compongono la pellicola cinematografica.
- ▶ Front-end
  - ▷ La parte visibile all'utente di un programma e con cui egli può interagire - tipicamente un'interfaccia utente. Essa è responsabile dell'acquisizione dei dati di ingresso e della loro elaborazione in modo tale da renderli utilizzabili dal back-end.<sup>34</sup>
- ▶ Lato client
  - ▷ Le operazioni di elaborazione effettuate da un utente in un'architettura client-server. Un'operazione tipica client-side ad esempio è quella del caricamento di un'interfaccia grafica utente, ad esempio il front-end di un'applicazione web o di un comune sito web.<sup>35</sup>

---

<sup>32</sup> Wikipedia, voce *Application programming interface*

<sup>33</sup> Wikipedia, voce *Front end e back end*

<sup>34</sup> Ibid.

<sup>35</sup> Wikipedia, voce *Lato client*

- ▶ Lato server
  - ▷ Le operazioni di elaborazione compiute dal server in un'architettura client-server. Per server si intende un programma software, che gira su una macchina remota (chiamata per estensione 'server') rimanendo in ascolto su determinate porte e raggiungibile da un computer client. Alcune operazioni necessitano l'esecuzione lato server in quanto richiedono l'accesso a informazioni o funzionalità non disponibili sul client, o richiedono misure di sicurezza che sarebbero inaffidabili se eseguite lato client.<sup>36</sup>
- ▶ Multiplexing, MUX
  - ▷ Processo inverso al demultiplexing, consiste nel combinare diversi flussi di segnale digitale precedentemente appartenuti allo stesso segnale. Nel caso di FFmpeg, esso consiste nella ricombinazione di pacchetti di informazione video/audio per ottenere il file di output.
- ▶ Path, pathname, percorso
  - ▷ Indica la posizione specifica di un elemento all'interno di un archivio dati. Può corrispondere ad un file o una cartella.
- ▶ Run-time system
  - ▷ Software che fornisce i servizi necessari all'esecuzione di un programma, pur non facendo parte in senso stretto del sistema operativo. Nel caso di Node.js, esso permette la scrittura di codice Javascript eseguibile sul server, in quanto di base Javascript sarebbe eseguibile solo lato client.
- ▶ Suite software
  - ▷ Collezione di programmi che condividono simili funzionalità e interfacce, e che possono comunicare tra loro in modo semplice.

---

<sup>36</sup> Wikipedia, voce *Lato server*

## 7. Bibliografia

### Classica

Borghi, Beatrice. 2016. *La Storia: indagare, apprendere, comunicare*. Bologna, Pàtron Editore.

Gallagher, Owen. 2017. *Reclaiming Critical Remix Video: the role of sampling in transformative works*. New York, Routledge.

Grasso, Giovanni, Chiara Mannari e Davide Serramazza. 2022. *Intelligenza artificiale e archivi audiovisivi: potenzialità e sfide del progetto "PH-Remix"* in "AIUCD 2022 - Proceedings", pp. 141-144.  
shorturl.at/qwzNS

Grasso, Giovanni, Chiara Mannari e Davide Serramazza. *PH-Remix. Enhancing cataloguing and promotion of film heritage through video remix and artificial intelligence* in "Florence Heri-Tech".  
shorturl.at/cfhxN

McIntosh, Jonathan. 2012. *A history of subversive remix video before YouTube: Thirty political video mashups made between World War II and 2005*. San Francisco.  
<https://journal.transformativeworks.org/index.php/twc/article/view/371/299>

Navas, Eduardo. 2012. *Remix Theory: the aesthetics of sampling*. Vienna, SpringerWienNewYork.

Salvatori, Enrica (a cura di). 2012. *Storia e territorio della Val di Vara*. Pisa, Felici.

### Digitale

Da Wikipedia:

- ▶ voce in lingua inglese *Remix Culture*
  - ▷ [https://en.wikipedia.org/wiki/Remix\\_culture](https://en.wikipedia.org/wiki/Remix_culture)
- ▶ voce *Node.js*
  - ▷ <https://it.wikipedia.org/wiki/Node.js>
- ▶ voce *FFmpeg*
  - ▷ <https://it.wikipedia.org/wiki/FFmpeg>
- ▶ voce *Application programming interface*
  - ▷ [https://it.wikipedia.org/wiki/Application\\_programming\\_interface](https://it.wikipedia.org/wiki/Application_programming_interface)

- ▶ voce *Front-end e back-end*
  - ▷ [https://it.wikipedia.org/wiki/Front-end\\_e\\_back-end](https://it.wikipedia.org/wiki/Front-end_e_back-end)
- ▶ voce *Lato client*
  - ▷ [https://it.wikipedia.org/wiki/Lato\\_client](https://it.wikipedia.org/wiki/Lato_client)
- ▶ voce *Lato server*
  - ▷ [https://it.wikipedia.org/wiki/Lato\\_server](https://it.wikipedia.org/wiki/Lato_server)

Da Youtube:

- ▶ Ferguson, Kirby. 2021. *Everything is a Remix Part 1*
  - ▷ [https://www.youtube.com/watch?v=MZ2GuvUWaP8&t=615s&ab\\_channel=KirbyFerguson](https://www.youtube.com/watch?v=MZ2GuvUWaP8&t=615s&ab_channel=KirbyFerguson)

Documentazione linguaggi e moduli:

- ▶ FFmpeg
  - ▷ <https://ffmpeg.org/ffmpeg.html>
  - ▷ <https://ffmpeg.org/ffmpeg-filters.html>
- ▶ fluent-ffmpeg
  - ▷ <https://www.npmjs.com/package/fluent-ffmpeg>
- ▶ ffmpeg-concat
  - ▷ <https://www.npmjs.com/package/ffmpeg-concat>
- ▶ GL transitions
  - ▷ <https://gl-transitions.com/>

Documentazione PH-Remix:

- ▶ Sito web Laboratorio di Cultura Digitale - PH-Remix
  - ▷ <http://www.labcd.unipi.it/ph-remix/>
- ▶ Sito web Laboratorio di Cultura Digitale - Poster
  - ▷ <http://www.labcd.unipi.it/ph-remix/poster/>
- ▶ Sito web Laboratorio di Cultura Digitale - AIPH Award
  - ▷ <https://www.labcd.unipi.it/ph-remix/aiph-award/>
- ▶ Seminario di Cultura Digitale “Remix Culture” con Chiara Mannari
  - ▷ <http://www.labcd.unipi.it/seminari/chiara-mannari-remix-culture/>

- ▶ Seminario di Cultura Digitale “Fruizione dei beni immateriali” con Caterina Murgo
  - ▷ <http://www.labcd.unipi.it/seminari/seminario-di-cultura-digitale-caterina-murgo-fruizione-dei-beni-immateriali/>
- ▶ Memorandum *Recap della Call con Fondazione Sistema Toscana e Festival dei Popoli e considerazioni integrative al parere datato 28 luglio 2021*, avvocati Marcello Mustilli e Matteo Roiz
  - ▷ [shorturl.at/amoEF](http://shorturl.at/amoEF)

Altro:

- ▶ BibliografiaDigitaleTesi/codice\_esportazione.js - Codice completo
  - ▷ [https://github.com/valentinagalli/BibliografiaDigitaleTesi/blob/main/codice\\_esportazione.js](https://github.com/valentinagalli/BibliografiaDigitaleTesi/blob/main/codice_esportazione.js)
- ▶ AIPH - Concorso per i migliori progetti di Public History realizzati fra il 1° gennaio 2021 e il 25 aprile 2022
  - ▷ <https://aiph.hypotheses.org/10494>
- ▶ Diritto all'integrità dell'opera - Legge sul diritto d'autore - SIAE
  - ▷ <https://www.siae.it/it/diritto-dautore/diritti-morali/diritto-allintegrità-dellopera>
- ▶ Decreto legislativo 8 novembre 2021, n. 177
  - ▷ <https://www.gazzettaufficiale.it/eli/id/2021/11/27/21G00192/sg>
- ▶ Direttiva (UE) 2019/790 del Parlamento Europeo e del Consiglio del 17 aprile 2019
  - ▷ <https://eur-lex.europa.eu/legal-content/IT/TXT/PDF/?uri=CELEX:32019L0790&from=RO>

Tutti i collegamenti presenti in bibliografia digitale sono stati visitati per l'ultima volta il 15 aprile 2022.