POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria del Cinema e dei Mezzi di Comunicazione

Tesi di Laurea Magistrale

Progetto di una interfaccia utente per la creazione e la modifica di SmartVideo

Relatore:
Prof. Giovanni Malnati

Candidata:
Valeria Milan Matr. 256877

Anno accademico 2020/2021

Ringraziamenti

Vorrei ringraziare il prof. Malnati e l'azienda inRebus per avermi dato la possibilità di realizzare il progetto di tesi, per cui ho potuto mettermi alla prova come mai fino ad ora.

Sono tante le persone che vorrei ringraziare in realtà. Primi fra tutti i miei genitori e la mia famiglia. Nonostante tutto ci sono sempre, mi supportano e credono in me. Senza di loro non sarei mai arrivata fino a qui e spero di renderli fieri di me fino alla fine.

Al secondo posto gli amici, sia vecchi che nuovi, sia in Italia che sparsi per il mondo. Sono stati i cinque anni più belli, più ricchi di emozioni di tutta la mia vita e lo devo soprattutto a loro. Troppi i nomi da citare, dovrei aggiungere un'appendice solo per nominarli tutti. E quindi li elenco lo stesso: la Cleme, le mie "ragazze", il gruppo di Moncalieri, le compagne di squadra, i Geek, gli amici dell'Erasmus e tutte le persone stupende conosciute al Politecnico di cui farò tesoro anche in futuro.

In particolare un super grazie al mitico Denny per avermi guidato fuori dal tunnel della disperazione di Redux, a Cesare tester #1 dell'applicazione, a Mara e a Giulia per il supporto emotivo e mentale immenso nei momenti di crisi.

Tutti hanno contribuito a rendere fantastici questi anni e a tutti dico

"Grazie di cuore, vi voglio bene"

Abstract

Con il termine SmartVideo si vuole indicare un ipervideo, cioè un prodotto multimediale interattivo all'interno del quale sono presenti collegamenti a parti del video stesso e/o a risorse e contenuti esterni al media principale. Di conseguenza è possibile affermare che l'idea di base è simile al concetto di ipertesto, ovvero un testo con ancore cliccabili che permettono all'utente di navigare attraverso i diversi contenuti in modo non lineare. Parallelamente è importante notare il ruolo del "tempo": la dinamicità propria del video, infatti, può influenzare la presenza dei collegamenti durante la riproduzione e quindi richiedere l'utilizzo di nuove tecniche e strategie per presentare i contenuti all'utente. Lo scopo della seguente tesi è quello di presentare lo SmartVideo come strumento per esporre in modo sintetico e compatto dei contenuti attraverso uno storytelling non lineare, che sia innovativo nell'ambito della formazione aziendale e, più in generale, come possibile soluzione alla didattica a distanza. Per raggiungere tale obiettivo è stata costruita un'interfaccia web che guidi passo passo nella creazione del proprio SmartVideo: al termine del percorso il video sarà usufruibile da pagina web e di conseguenza facilmente condivisibile dall'utente attraverso link. Inoltre, ai fini formativi è prevista una sezione riguardante quiz a risposta multipla che fungono da feedback sull'apprendimento delle competenze. La tesi tratta quindi di tre macroargomenti: il primo introduce il concetto degli ipervideo e i campi di applicazione, il secondo è dedicato a un'analisi approfondita dell'interfaccia dal punto di vista concettuale e progettuale e infine le conclusioni riassumono i risultati ottenuti e le possibili ottimizzazioni future.

Indice

Elenco delle figure **9**

Introduzione **13**

1 SmartVideo come stumento per la formazione professionale **15**
1.1 Definizione di SmartVideo 15
1.1.1 Perchè il video? 17
1.2 Ipervideo come strumento di ausilio alla formazione 18
1.2.1 Sistema cognitivo 18
1.2.2 Video interattivi per la formazione aziendale attuale 20
1.3 Lavori correlati 22

2 Analisi e concetti alla base degli SmartVideo **30**
2.1 Obiettivi di progetto 30
2.1.1 Limitazioni della situazione attuale 31
2.1.2 Analisi dei miglioramenti apportabili 31
2.2 Target 32
2.2.1 Utenti-creatori 32
2.2.2 Utenti-spettatori 34
2.3 La soluzione: un'applicazione web 35
2.4 Requisiti di sistema 38
2.5 Scelte stilistiche 39

3 Applicazione web **44**
3.1 La base dati e il modello relazionale 44
3.1.1 MySQL e il modello relazionale 44
3.1.2 InnoDB 45

3.1.3 Linguaggio SQL 45
3.1.4 Organizzazione delle tabelle 47
3.2 La base dati e il modello relazionale 52
3.2.1 Introduzione a Xampp, PHP e Axios 53
3.2.2 Gestione delle richieste HTTP attraverso i Redux Thunk. . 55
3.2.3 Gestione degli endpoint dell'interfaccia 60
3.2.4 Gestione della registrazione e del login dell'utente 68
3.2.5 Gestione del caricamento dei file 71
3.2.6 Gestione degli errori 73
3.3 Interfaccia web 76
3.3.1 Single Page Application 76
3.3.2 React: concetti base 77
3.3.3 Componenti 77
3.3.4 React Hooks 80
3.3.5 Redux 90
3.3.6 React Router 92
3.3.7 Librerie di supporto 98
3.3.8 Descrizione delle pagine dell'applicazione 103

4 SmartVideo: la voce del target 121
4.1 Dati anagrafici del target 121
4.2 Analisi dei fruitori 123
4.3 Analisi dei creatori 125
4.4 Analisi dei contenuti 127
4.5 *"Lo consiglieresti?"* 129
4.6 Ottimizzazioni future 130
4.6.1 Ottimizzazioni previste nello sviluppo del progetto 130
4.6.2 Ottimizzazioni suggerite dal target 131

Conclusioni 133

Bibliografia e sitografia 135

Elenco delle figure

1.1 Logo H5P 23
1.2 H5P: visualizzazione del video interattivo 23
1.3 H5P: schermata di editing 24
1.4 Logo Camtasia 25
1.5 Camtasia: schermata di editing 25
1.6 Logo Kaltura 25
1.7 Kaltura: visualizzazione dei quiz 26
1.8 Kaltura: schermata di editing 26
1.9 Vimeo: anteprima capitoli 27
1.10 Anteprima piattaforma ON 27
1.11 Logo Edpuzzle 28
1.12 Edpuzzle: schermata di editing 28
1.13 Edpuzzle: visualizzazione del video interattivo 29
2.1 Categorizzazione dei link in base alle condizioni di tempo e spazio 37
2.2 Versioni del logo SmartVideo 39
2.3 Grafica banner SmartVideo 39
2.4 Palette SmartVideo 40
2.5 Esempi della grafica parte 1 40
2.6 Esempi della grafica parte 2 41
2.7 Esempi della grafica parte 3 41
2.8 Logo relativo alla cartella di un progetto 41
2.9 Logo relativo agli SmartLink 42
2.10 Logo relativo agli SmartQuiz 42
2.11 Rappresentazione delle icone relative a un singolo progetto 43
3.1 Modello concettuale della base dati 49
3.2 Vincolo d'integrita referenziale violato 52
3.3 Rappresentazione di una generica richiesta client/server 54
3.4 Schema delle fasi di una promise 55

3.5 Screenshot email di verifica 70
3.6 Visualizzazione grafica dell'errore 75
3.7 Ciclo di vita di un componente 80
3.8 Ciclo di vita dei React Hooks 82
3.9 Esempio pratico del funzionamento degli Hook 85
3.10 Gestione della visualizzazione del menù laterale 95
3.11 Material UI Switch 102
3.12 Material UI feedback con Snackbar 103
3.13 Schermata di login 104
3.14 Feedback tramite Formik e server 104
3.15 Verifica dell'account tramite OTP 105
3.16 Schermata "I miei progetti" 106
3.17 Creazione di un nuovo progetto 106
3.18 Schermata pagina principale del progetto 107
3.19 Schermata "Panoramica" 108
3.20 Schermata "Dettagli del progetto" 109
3.21 Schermata "Impostazioni video" 110
3.22 Schermata "Contenuti smart" 111
3.23 Eliminazione di un contenuto 111
3.24 Schermata "SmartLink" 112
3.25 Posizionamento di un link 112
3.26 Visualizzazione delle risorse caricate 113
3.27 Schermata "SmartQuiz" 114
3.28 Schermata "Glossario" 115
3.29 Schermata "SmartVideo" 116
3.30 Visualizzazione del glossario e delle informazioni 116
3.31 Vista dei contenuti satellite prima dell'interazione dell'utente 117
3.32 Vista dei contenuti satellite in seguito all'interazione dell'utente 118
3.33 Visualizzazione delle risorse 119
3.34 Visualizzazione dei quiz 119
3.35 Visualizzazione dei risultati 120
4.1 Sesso del target 121
4.2 Età del target 122
4.3 Professione del target 122
4.4 Settori aziendali raggiunti 123
4.5 Conoscenze e interesse da parte degli studenti 123
4.6 Conoscenze e interesse da parte dei dipendenti aziendali 124
4.7 Motivi di utilizzo di video interattivi 124
4.8 Opinioni sullo SmartVideo (fruitori) 125
4.9 Conoscenze e interesse da parte dei professori 125

4.10 Conoscenze e interesse da parte dei formatori aziendali 126
4.11 Opinioni sullo SmartVideo (creatori) 126
4.12 Ottimizzazioni future 131

Introduzione

Con l'arrivo del digitale e la diffusione di Internet, emerge un nuovo modo di approcciarsi all'apprendimento: l'*e-learning*. A partire dagli anni '90, con l'avvento del web 2.0, il concetto di formazione viene integrato in un ambiente online, interattivo e collaborativo, di cui hanno favorito soprattutto le aziende [1]. Di seguito sono elencati alcuni dei principali vantaggi dell'e-learning in ambito aziendale [2]:

- Aggiornamento continuo e in tempo reale. Essere sempre aggiornati sulle novità nel campo aumenta la competitività dell'azienda;
- Monitoraggio del processo di apprendimento da parte delle risorse. Accedere facilmente ai risultati permette, in caso di necessità, un tempestivo cambiamento nella struttura del corso, come sopra citato;
- Costi ridotti;
- Abbattimento delle distanze.

Oggigiorno questo concetto non è più strettamente legato al mondo aziendale: l'emergenza COVID-19, infatti, ha favorito l'esplosione della formazione a distanza. I sistemi educativi di tutti i Paesi hanno dovuto interfacciarsi con questa nuova realtà per poter garantire un'educazione continua, equa, inclusiva e di qualità [3]. Il tradizionale concetto di istruzione

[1] https://www.dyndevice.com/it/news/come-nasce-come-si-sviluppato-l-learning-ELN-445/

[2] https://www.randstad.it/knowledge360/gestione-del-personale/e-learning-come-si-evolve-il-concetto-di-formazione-in-azienda/

[3] https://www.eni.com/it-IT/carriere/formazione-online-coronavirus.html

in "presenza" ha subito una repentina *digital transformation*[1] che ha solo anticipato quella che sarà, nel futuro post-Covid, una realtà affermata, parallela alla didattica in aula.

La tesi è stata svolta presso l'azienda inRebus Technologies (Gruppo FOS) con l'obiettivo di proporre lo SmartVideo come un supporto digitale alla formazione a distanza: uno strumento per esporre in modo sintetico e compatto gli argomenti principali dei corsi attraverso uno storytelling non lineare. Si tratta, in sintesi, di affiancare al video, che nella società moderna si sta affermando come uno dei medium dominanti, contenuti di qualsiasi natura: link a risorse web, documenti, slideshow, immagini, audio, video e, per chi ne avesse la necessità, anche un test di verifica delle competenze.

In quest'ottica la tesi è stata strutturata in quattro sezioni principali.

Il capitolo 1 introduce il concetto di "ipervideo", fornisce una definizione contemporanea del termine e mostra alcuni esempi di integrazione nell'ambito dell'istruzione.

Il capitolo 2 esamina le caratteristiche, e soprattutto le limitazioni, del prototipo di SmartVideo utilizzato dall'azienda inRebus Technologies per definire l'insieme di specifiche e di scelte stilistiche su cui si è basata l'applicazione web, derivandone i requisiti del sistema.

Il capitolo 3 affronta nel dettaglio l'applicazione sotto tre aspetti principali: l'organizzazione del database, l'architettura del progetto lato server e l'architettura lato client.

Il capitolo 4 riporta i risultati di un'indagine condotta sulle conoscenze e le opinioni del prodotto da parte del target di riferimento e ne ricava un insieme di possibili ottimizzazioni future che renderebbero più completo ed efficiente il prodotto.

Infine, il capitolo conclusivo serve a ripercorre in sintesi il percorso logico esposto nei capitoli precedenti.

[1] Insieme di cambiamenti prevalentemente tecnologici, culturali, organizzativi e creativi". L. Corona, Delivery Coordinator Randstad Italia, *https://www.randstad.it/knowledge360/gestione-del-personale/il-ruolo-dellhr-nella-digital-transformation/*

Capitolo 1

SmartVideo come strumento di formazione

1.1 Definizione di SmartVideo

Il termine SmartVideo è stato coniato dall'azienda inRebus Technologies per indicare un ipervideo di loro utilizzo, impiegato per erogare corsi formativi ai loro clienti. In altre parole, si tratta di un prodotto multimediale interattivo in cui la struttura tradizionalmente lineare del video viene rivisitata aggiungendo aree sensibili che ne influenzano la fruizione, ovvero un ipervideo. Un video in cui sono inseriti particolari collegamenti, chiamati hyperlinks, che possono presentarsi sotto forma o di vere e proprie regioni sensibili all'interno dei frame o di hotspot[1], conferendo un aspetto interattivo al media. Generalmente, quando si sente parlare di video interattivo non esiste una chiara definizione riconosciuta da tutti gli individui. I più identificano con questo termine i video live dei social media come Facebook o Instagram in cui l'utente può interagire lasciando like, reazioni o commenti [4]. In alternativa vengono riconosciuti come i prodotti mediali in cui all'interno della narrazione è possibile effettuare scelte su come la storia dovrebbe proseguire, il cosiddetto *branching*. Caso esemplare il film "Black Mirror: Bandersnatch"[2] (2018) dove in base alle interazioni dello spettatore era mostrato un finale differente. Paradossalmente qualsiasi

[1] Elemento rappresentato da simboli, icone, scritte cliccabili con una visualizzazione a comparsa in sovrimpressione sul video.

[2] *Black Mirror: Bandersnatch*, diretto da David Slade (2018, Netflix)

[4] https://www.videosmart.com/blog/post/interactive-black-mirror-bandersnatch

video in cui è possibile compiere le azioni di riproduzione, pausa, stop o avanzamento veloce è da considerarsi interattivo. La realtà dietro al termine hypervideo è quindi molto più vasta di quanto sembri.

Proprio a causa della mancanza di una definizione accurata, Britta Meixner nel suo trattato "Hypervideo and interactive multimedia presentation" (2017) propone una categorizzazione degli ipervideo basata sulla funzione che svolgono gli hyperlink all'interno del video, distinguendoli in:

- omogenei: in questa sezione rientrano tutti gli hypervideo in cui siano presenti collegamenti a parti del video stesso o tra video diversi, come nel caso di Black Mirror;

- eterogenei: la narrazione del video è lineare, ma sono presenti link a informazioni esterne, aggiuntive e di qualsiasi forma (immagini, video, audio,etc...);

- ibridi: contiene tutti gli hypervideo che presentano un connubio tra le caratteristiche degli omogenei e quelle degli eterogenei.

È plausibile pensare che la definizione più completa per identificare un ipervideo si avvicini a quella usata per gli ipervideo ibridi della Meixner.

L'idea di base riprende quindi il concetto di ipertesto, ovvero un testo con ancore cliccabili che permettono all'utente di navigare attraverso i diversi contenuti in modo non lineare. La differenza eclatante tra le due realtà risiede nel ruolo del "tempo". La dinamicità propria del video, infatti, può influenzare la presenza dei collegamenti durante la riproduzione e quindi richiedere l'utilizzo di nuove tecniche e strategie per presentare i contenuti all'utente, diversamente dalla staticità del testo i cui i link sono sempre presenti e spetta all'utente scegliere se e quando esplorarli.

Nel corso della trattazione ci si riferirà a SmartVideo come a un video all'interno del quale sono presenti particolari collegamenti indicati con il termine "contenuto satellite", che rappresentano approfondimenti ai temi trattati, arrichiti da risorse web, documenti, immagini, audio, video ed eventulamente quiz di verifica delle competenze.

[5] Meixner B., *Hypervideos and Interactive Multimedia Presentations*, in *ACM Computing Surveys*,**n. 9**, marzo 2017, DOI: 10.1145/3038925

1.1.1 Perchè il video?

Dagli albori del web 2.0 si è assistito a un continuo evolversi delle possibili forme di interazione delle persone nei confronti dei media. Partendo dall'ipertesto di Tim Berners-Lee, con l'affermarsi della multimedialità e di Internet sono state sviluppate nuove forme di fruizione e condivisione di contenuti che trascendono la dimensione spaziale e temporale, portando alla nascita di media interattivi detti ipermedia [6]. Infine, l'integrazione di elementi audiovisivi con questi ultimi ha segnato la comparsa degli ipervideo.

Al giorno d'oggi il video è considerato un elemento essenziale e discriminante in diversi ambiti della vita quotidiana.

Secondo una ricerca svolta da Wavemaker[3] l'Italia risulta al quarto posto nella classifica globale e al primo in Europa per la fruizione video. Sempre più spesso si assiste a un fenomeno di contaminazione delle piattaforme, una sovrapposizione tra tv lineare, streaming e on-demand. Oggigiorno esistono più di 360 canali digitali e 8,5 milioni di smart tv, di cui più del 70% connesse, che permettono l'accesso a piattaforme all'interno delle quali ogni minuto sono pubblicate 400 ore di video [6].

Non solo, un secondo ambito notevolmente influenzato da questa tecnologia riguarda il marketing aziendale. Le strategie di marketing basate sul video hanno molto successo in quanto [7][8][9]:

- la comunicazione avviene in maniera semplice, chiara e personalizzata;
- è possibile sfruttare lo strumento per comunicare sia con i clienti sia con i propri dipendenti (comunicazione interna, mostrare i risultati della propria azienda ai collaboratori, etc…);
- l'utilizzo del video genera un numero maggiore di risultati rispetto a un testo scritto;

[3] *Quattro trend che parlano dell'evoluzione del video*, di Wavemaker, agenzia di media, content & technology, su Wired.it, Febbraio 2020

[6] https://www.iperdesign.com/it/2016/11/18/dallipertesto-al-video-interattivo-nuove-frontiere-per-la-formazione-line/

[7] https://www.telemaco.it/blog/strategia-di-comunicazione/

[8] https://www.mark-up.it/comunicare-un-brand-il-futuro-e-nel-video-marketing/

[9] https://www.comunicaresulweb.com/web-marketing/motivi-per-usare-video-marketing/

- il video aumenta la visibilità sui social, e in generale su Internet, del proprio brand;
- l'engagement dei clienti è molto elevato.

Tutto questo senza nulla togliere ai contenuti di qualità forniti in formato testuale o infografico, i quali sono e rimarranno indispensabili anche in futuro per le campagne di marketing.

1.2 Ipervideo come strumento di ausilio alla formazione

"I believe that the motion picture is destined to revolutionise our educational system and that in a few years it will supplant largely, if not entirely, the use of textbooks."

Thomas Edison, 1922

"Humans learned life's lessons by seeing real things or pictures with their eyes for ages, before they began learning through written or spoken words, so it is not strange that they still learn most readily that way" … "The cartoon is a good medium to stimulate interest. It is an ideal medium for teaching."..."Educational films will never replace the teacher... but their advancement by means of the motion picture screen will give more people in this world an opportunity to learn. Pictures can make both teaching and learning a pleasure. And educators agree that when a student has begun to learn and like it, half their problem is solved."

Disney, 1994

1.2.1 Sistema cognitivo

Prima di proseguire con la spiegazione dell'implementazione dell'ipervideo come supporto alla formazione è necessario fare un passo indietro e prendere coscienza dei processi mentali e cognitivi che legano i media all'apprendimento.

Teresa Chambel, José Bidarra, Nuno M. Guimarães nel loro articolo "Learning with Video in Hypermedia"[10] hanno affrontato tematiche

[10] Chambel T., Bidarra J., Guimarães N. M., *Learning with Video in Hypermedia*, Tech. Report DI/FCUL, TR-01-16, Univ. of Lisbon, Dicembre 2001, www.di.fc.ul.pt/tech-reports/ 01-16.pdf.

simili per lo sviluppo di "UNIBASE - Interactive Multimedia for Open and Distance Learning", un progetto congiunto della Universidade Aberta (Open University) and Faculdade de Ciências per lo sviluppo di un video interattivo per mantenere attiva la partecipazione degli studenti nella didattica in Portogallo. Basandosi sugli studi di Norman[4] sono stati identificati due processi cognitivi strettamente legati alla fruizione dei media, ovvero la cognizione esperienziale e la cognizione riflessiva. Si tratta di forme di apprendimento essenziali, non totalmente indipendenti l'uno dall'altro ma per molti aspetti distinti e per questo legati a media differenti.

Experiential Cognition

In questa categoria rientrano tutti i processi che sono conseguenza di una reazione del soggetto ad eventi che si verificano intorno a lui, necessari per le prestazioni che richiedono qualche tipo di abilità. Sono rapidi, immediati, avvengono senza sforzo e senza pensare a un modo per risolvere il problema. Grazie a questa modalità è possibile fare nuove esperienze, ma non ottenere nuove idee, pensieri indispensabili per poter progredire nell'apprendimento e nella comprensione dei concetti. Esempi di media esperienziali sono la televisione o i video in generale, ovvero strumenti che per natura rendono passivo l'individuo vincolandolo a una mera visione dei contenuti [10].

Reflective Cognition

L'apprendimento riflessivo è caratterizzato da paragoni e contrasti, pensieri, ragionamenti e decision making e per questo molto più difficile da realizzare rispetto a quello esperienziale. Sono i processi che portano una persona ad avere nuove idee, risposte ai problemi, richiedono quindi tempo per compiersi. Nonostante le riflessioni e i ragionamenti siano parte della natura umana, per ottenere una riflessione efficace è necessario ricorrere a determinate strategie. Un medium finalizzato alla cognizione riflessiva deve dare il tempo per riflettere all'audience, come nei casi di libri e testi [10].

[4] Norman D., Things that make us smart: defending human attributes in the age of the machine, Addison Wesley Publishing Company, 1993

Ogni individuo apprende, fa esperienza in maniera differente dagli altri. Allo stesso modo media diversi possono essere usati per supportare differenti stili di apprendimento, sia per la tipologia del mezzo, sia per come viene integrato nel processo formativo. Qualora le informazioni da trasmettere fossero le stesse per supporti mediali diversi non è scontato che il risultato sia lo stesso in tutti i casi. Secondo Norman infatti il medium non è neutrale nella trasmissione del messaggio, anzi ne influenza la percezione e l'impatto che ha sugli individui.

Nella realtà effettiva non esiste una distinzione netta tra medium esperienziali e riflessivi: riorganizzando la modalità di fruizione è possibile ottenere soluzioni ibride che permettono entrambe le tipologie di processi cognitivi. Esempio eclatante l'ipervideo: si tratta di un media esperienziale all'interno del quale sono inseriti controlli, che ne rendono interattiva la riproduzione, ed elementi che offrono all'utente spunti di riflessione secondo la legge del "stop, think, and correlate" [11]. Mentre il video cattura l'attenzione, gli hyperlink spingono lo studente ad agire e a ragionare sui contenuti che serviranno per rispondere alle domande sulla verifica delle competenze.

1.2.2 Video interattivi per la formazione aziendale attuale

Grazie ai loro device le persone accedono quotidianamente a miliardi di informazioni online: data la ricchezza di contenuti, i formati privilegiati attraverso cui possono essere raggiunti e condivisi nel piccolo schermo sono sicuramente quelli multimediali. Risulta evidente la tendenza a considerare più piacevole guardare un video piuttosto che pagine di testo o slide, la gente si sente più coinvolta da immagini in movimento, suoni e voci.

Rapid e-learning [12], *Video-Learning*: si tratta di alcuni dei tanti termini con cui viene indicata questa nuova metodologia di formazione affermatasi negli ultimi anni. Come accennato poc'anzi si sta vivendo in una società iperconnessa, social e iperformata in cui le competenze digitali sono diventate strategiche ed essenziali per interagire e sfruttare le possibilità

[11] Chambel T., Bidarra de Almeida J., Guimarães N., *Multimedia Artifacts That Help Us Learn: Perspectives of the UNIBASE Project on Distance Learning*, Workshop su Multimedia and Educational Practice, ACM Multimedia, Bristol, UK 1998.

[12] https://www.mosaicoelearning.it/blog/facili-video-interattivi-le-learning/

offerte dal canale telematico [13]. Non solo, investire sui video permette di guadagnare tempo prezioso sia a chi sta promuovendo il corso, sia a chi deve seguirne le lezioni: grazie alla disponibilità di software e applicazioni sempre più all'avanguardia i formatori possono produrre video di qualità eccellente semplicemente utilizzato una webcam o persino la fotocamera di uno smartphone, con i dovuti accorgimenti. Inoltre, vengono fornite più informazioni in un minor tempo [14].

Se venisse aggiunta la componente interattiva al semplice video le possibilità che si presenterebbero aumenterebbero notevolmente. La fruizione da passiva diventa altamente produttiva e attiva [13]: per evitare la passività degli utenti si rendono interattivi i video e i loro contenuti.

Per quanto riguarda la formazione aziendale, le funzionalità e le potenzialità di utilizzo di un video interattivo sono molteplici; tra le tipologie di strumenti realizzati con gli ipervideo sono presenti [15]:

- riprese di persone e/o attrezzature, strumentazione;
- video d'animazione in 3D, stop motion o tecniche affini, impiegati soprattutto per il forte impatto emotivo e per l'utilità nel rappresentare processi o viste di oggetti altamente difficili da mostrare altrimenti;
- simulazioni per istruire sull'utilizzo di determinati software attraverso la cattura dello schermo.

La componente interattiva viene fornita dall'aggiunta di illustrazioni e di elementi "attivi" al loro interno.

Con questi materiali possono essere successivamente realizzati ad esempio webinar, e-seminar, video lezioni, case study, presentazioni di prodotti o cataloghi di prodotti. Non solo, in ambito didattico un ipervideo può essere utilizzato per mostrare come svolgere, o risolvere, i compiti o per spiegare concetti e procedure specifici.

Aggiungendo la componente mobile alla formazione i benefici aumentano; si guadagnano: portabilità, connessione in qualunque momento e luogo, flessibilità, immediatezza delle comunicazioni, maggiore

[13] https://www.nextre.it/usare-video-interattivi-e-learning/
[14] https://www.docebo.com/it/blog/formazione-video-on-demand-come-e-perche/
[15] https://www.emathe.it/video-per-lapprendimento-formazione-aziendale/

indipendenza degli studenti [14].

Ovviamente usare video per l'apprendimento risulta vantaggioso in quanto può essere messo in pausa, riavviato, rivisto più volte, quindi essere riusabile dallo studente o dal dipendente. Ma è sbagliato pensare di affidarsi totalmente a questo mezzo. È importante saperlo usare e sfruttare a fine educativo, ovvero essere coscienti di come integrarlo e collegarlo al percorso formativo che si intende proporre [16]. Inoltre, è possibile che all'interno dell'audience siano presenti individui non totalmente favorevoli all'utilizzo di video, soprattutto se di lunga durata.

Si tratta, in sintesi, di affiancare e/o alternare al video materiali e contenuti di natura diversa come: link a risorse web, documenti, slides, immagini, audio e video...e, per chi ne avesse la necessità, anche test di verifica delle competenze.

Riassumendo, l'uso dell'ipervideo in contesti scolastici e di training aziendale può favorire la comprensione e il transfer delle conoscenze, soprattutto quando è necessario rendere osservabili processi o sistemi di formazione altrimenti troppo complessi, migliorare la connettività delle risorse e diminuire il gap fra i luoghi dell'apprendimento [17].

1.3 Lavori correlati

Nel panorama europeo dagli inizi degli anni 2000 ad oggi si sono alternati diversi esperimenti, ricerche, prototipi di progetti in cui si è tentato di inserire l'ipervideo nei processi di formazione online. Alcuni di essi sono stati trattati solo nella teoria, altri sono stati sviluppati e testati solo per un periodo determinato, altri invece si sono affermati e vengono tutt'oggi adoperati. Di seguito vengono riportati alcuni esempi di software impiegati dal 2015 a questa parte che sono stati analizzati per struttura e per contenuti ai fini del progetto. Come verrà affrontato nel capitolo 2 lo SmartVideo non

[5] www.smart-video.it

[16] https://www.elearningvincente.it/video-elearning-microlearning-videolearning/ https://www.mosaicoelearning.it/blog/facili-video-interattivi-le-learning/

[17] https://www.iperdesign.com/it/2016/11/18/dallipertesto-al-video-interattivo-nuove-frontiere-per-la-formazione-line/

è stato creato da zero durante il lavoro di tesi: si tratta di un prototipo di strumento esistente impiegato dall'azienda inRebus Technologies (Gruppo FOS)[5] che presenta attualmente diverse limitazioni. Dalle analisi di lavori correlati è stato possibile attuare miglioramenti al prodotto e integrare un'interfaccia ad hoc per realizzarlo.

H5P[6]

H5P è un plugin per sistemi di pubblicazione esistenti che permette di creare video interattivi, presentazioni per corsi, giochi, quiz. Ha una versione a pagamento e una gratuita: nel caso si scegliesse la seconda opzione è necessario aver configurato php e possedere Drupal, Wordpress o Moodle per poter condividere il proprio lavoro. Nel particolare per quanto riguarda i video interattivi è possibile aggiungere quiz di vario genere, risorse (immagini, tabelle, link), drag and drop di elementi, testi pop-up che possono comparire o sotto forma di bottone o direttamente interrompendo il video. Tra le altre caratteristiche si annoverano: l'aggiunta di un sommario interattivo finale, la suddivisione in capitoli del video, la possibilità di condizionare la riproduzione in base alle interazioni e alle risposte date dell'utente.

Figura 1.1 - Logo H5P

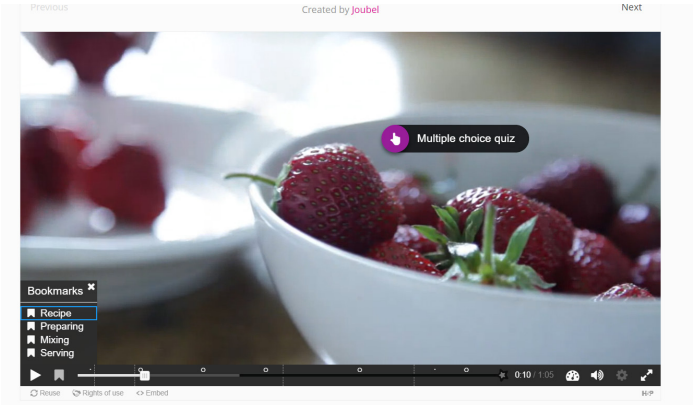


Figura 1.2 - H5P: visualizzazione del video interattivo

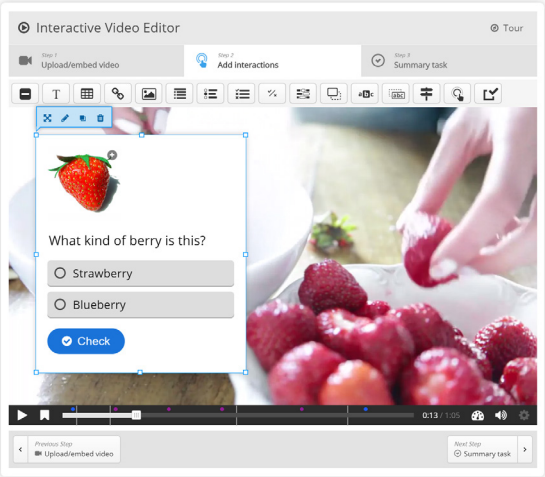


Figura 1.3 - H5P: schermata di editing

Le figure 1.3 e 1.2 mostrano rispettivamente l'editor del video interattivo e il risultato finale con la visualizzazione di un quiz a pop-up e dell'elenco di segnalibri, o capitoli, del video. Sulla timeline sono presenti riferimenti differenti per quiz e per contenuti (pallini di colore e di forma diversi) e i capitoli del video (linee tratteggiate). Le voci che permettono l'aggiunta di elementi interattivi sono rappresentate da semplici icone nella barra di modifica della figura 1.3: una scelta sensata per lo spazio a disposizione ma che potrebbe complicare l'usabilità e l'intuitività dell'interfaccia.

Camtasia[7]

Camtasia è uno software per la creazione di screencast a pagamento. Attraverso lo strumento è infatti possibile catturare lo schermo e contemporaneamente registrare l'audio del computer, di un microfono e effettuare riprese con la webcam. Sono inoltre previsti opzioni di editing e miglioramento del video ottenuto e strumenti per aggiungere quiz o segnalibri. Tutti i contenuti aggiuntivi e interattivi inseriti sono visibili solo se il prodotto è visualizzato attraverso il TechSmith Smart Player.

[6] https://h5p.org/integrations#plugins

[7] https://www.techsmith.com/tutorial-camtasia-quizzing.html

Figura 1.4 - Logo Camtasia

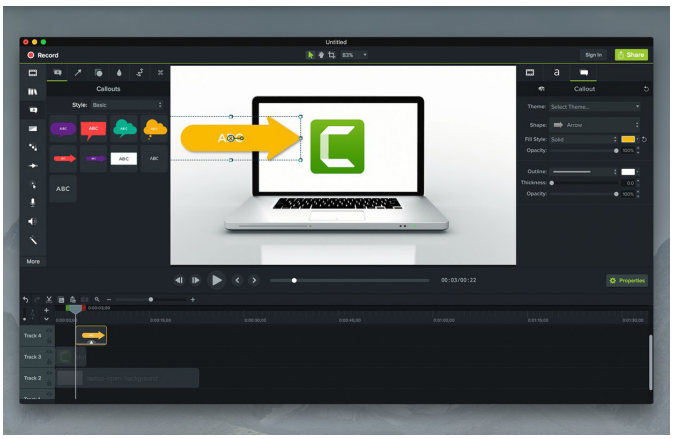


Figura 1.5 - Camtasia: schermata di editing

Kaltura Interactive Video Paths[8]

Sempre sullo stesso stampo dei precedenti è il progetto di Kaltura sui video interattivi caricabili sulla piattaforma social Kaltura MediaSpace [18]. Tramite lo strumento è possibile personalizzare i propri contenuti per creare un'esperienza differente ("scegli la tua avventura") per ogni cliente. I video realizzati possono essere condivisi e incorporati all'interno del proprio sito web, LMS o Mediaspace Video Portal per aumentare il coinvolgimento, migliorare l'apprendimento e aumentare la conversione [18].

Figura 1.6 - Logo Kaltura

[8] https://corp.kaltura.com/it/product/formazione-video-interattivi/

[18] https://corp.kaltura.com/video_resource/kaltura-mediaspace-overview/

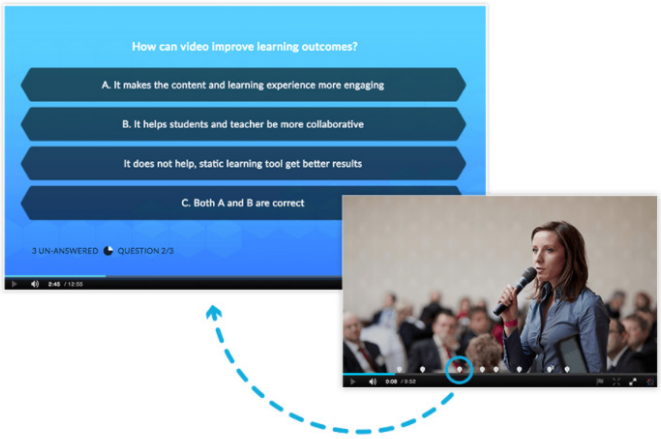


Figura 1.7 - Kaltura: visualizzazione dei quiz

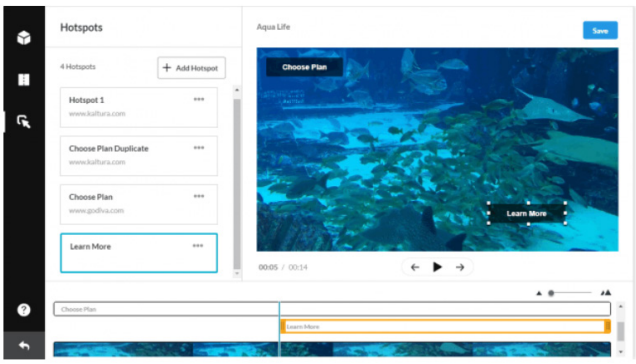


Figura 1.8 - Kaltura: schermata di editing

Vimeo Interaction Tools[9]

Vimeo è una delle principali piattaforme di video sharing utilizzate oggigiorno, competitor di Youtube. I membri con un account Vimeo Pro, Business, Premium e Enterprise possono accedere a diversi servizi aggiuntivi rispetto a un profilo standard tra cui strumenti per rendere interattivi i video. Gli elementi inseribili possono essere link a siti esterni, risorse, form dei contatti, card e capitoli. Per rendere visibili i contenuti interattivi è necessario che il video sia riprodotto tramite Showcase, altrimenti in qualsiasi altro modo l'esperienza dell'utente è ridotta alla semplice visualizzazione del media.

[9] https://vimeo.zendesk.com/hc/en-us/articles/115007469788-Video-interaction-tools-overview

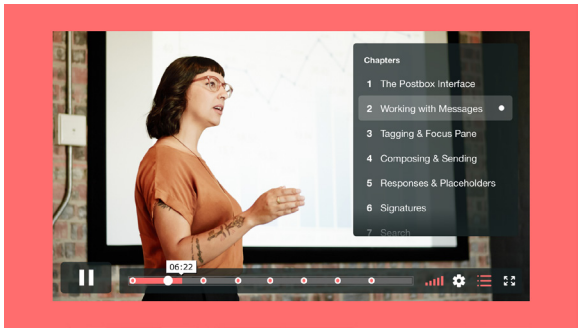


Figura 1.9 - Vimeo: anteprima capitoli

ON[10]

ON è una piattaforma social sia in formato web app sia mobile app per la formazione aziendale realizzato da Newton Management Innovation S.p.A grazie a Iperdesign. Per molti aspetti si presenta come un vero e proprio social network in cui è possibile pubblicare post, video, immagini e interagire con i contenuti di altri iscritti, effettuare sondaggi e televoti. La piattaforma comprende anche la creazione e l'utilizzo di video interattivi in cui è possibile assemblare sequenze di filmati e impostare domande a scelta multipla al termine di ogni clip.

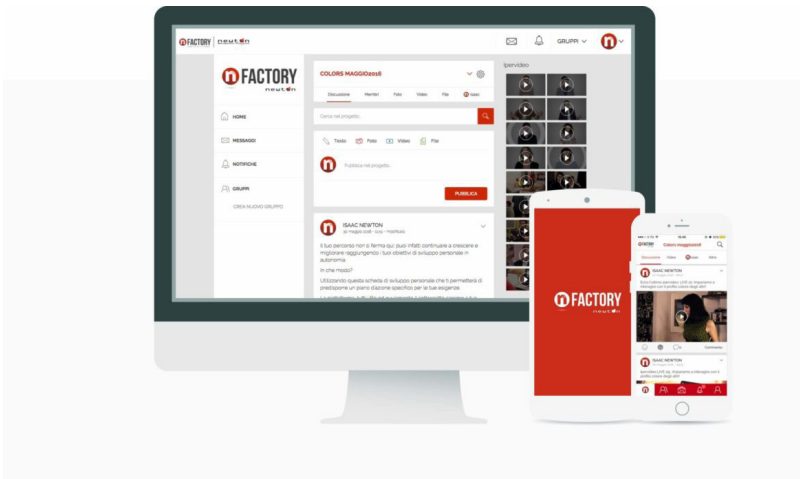


Figura 1.10 - Anteprima piattaforma ON

[10] https://www.iperdesign.com/it/2016/11/18/dallipertesto-al-video-interattivo-nuove-frontiere-per-la-formazione-line/

EdPuzzle[11]

EdPuzzle è uno strumento gratuito reso disponibile da Monica Morelli del gruppo *#daProfPerProf* sulla piattaforma di *e-learning* Aretè Formazione per la formazione del personale docente. Al suo interno è possibile creare e personalizzare videolezioni interattive e Video Quiz a partire da video e materiale presente in rete o nel proprio pc, inserendo quiz, note audio o aggiungendo il Voiceover ad esempio su parti in lingua straniera [19]. Attraverso il portale è possibile creare e gestire classi virtuali, assegnare esercizi (programmati e non) e monitorare l'attività degli studenti (attraverso voti e statistiche). Tra le funzionalità incluse è presente anche la possibilità di condividere i propri lavori con altri docenti iscritti alla piattaforma. L'ipervideo ottenuto è condivisibile sia tramite link agli studenti sia incorporabile nel proprio sito o blog.

Figura 1.11 - Logo Edpuzzle

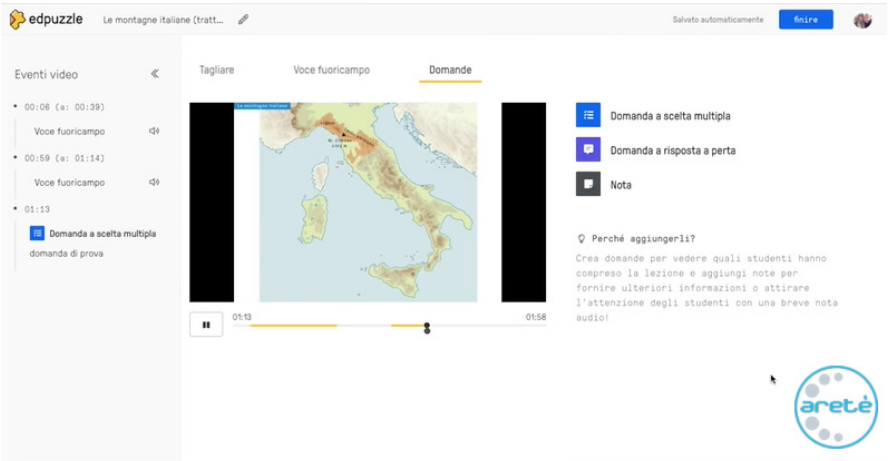


Figura 1.12 - Edpuzzle: schermata di editing

[11] https://edpuzzle.com/

[19] https://aretepiattaforma.it/tutorial/18/Come-si-fa-per-creare-video-lezioni-con-Edpuzzle

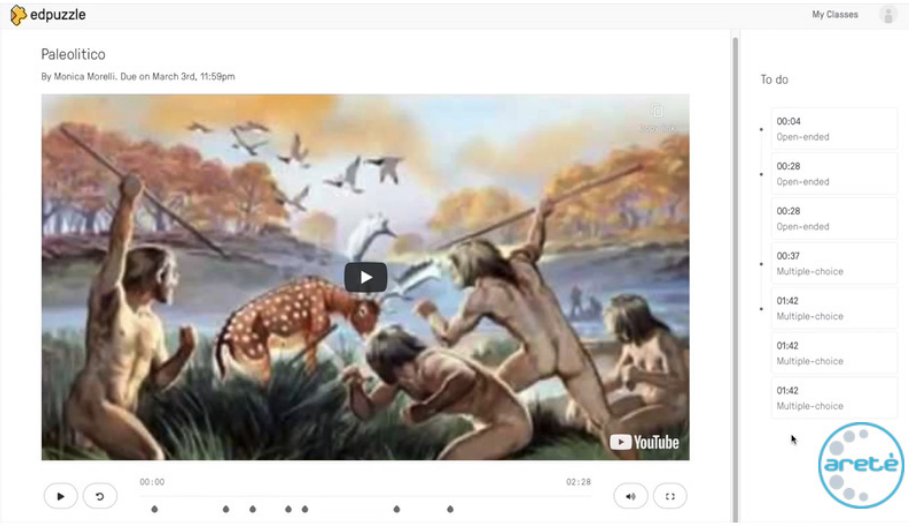


Figura 1.13 - Edpuzzle: visualizzazione del video interattivo

Nella figura 1.13 è mostrata la visualizzazione da parte dello studente del video interattivo: lungo la timeline è possibile accedere ai contenuti e ai quiz inseriti dal docente, mentre nella sezione laterale ne è mostrato un riepilogo. Nella figura 1.12 è invece rappresentata l'interfaccia dal punto di vista del professore per effettuare le modifiche al file sorgente e inserire i contenuti extra e interattivi.

Capitolo 2

Analisi e concetti alla base degli SmartVideo

2.1 Obiettivi di progetto

Le motivazioni che possono spingere a scegliere di utilizzare uno SmartVideo sono molteplici; alcuni esempi riguardano presentare un catalogo di prodotti o nello specifico le caratteristiche di uno di essi, illustrare un procedimento o un processo di sviluppo, mostrare gli *highlights* di un evento o di una conferenza o, ancora, realizzare un "porte aperte" virtuale di un complesso.

Lo scopo dell'applicazione è quello di fornire un supporto concreto ad aziende e ad istituzioni nella realizzazione di corsi formativi e/o didattici. In entrambi i casi è necessario possedere un video master che contenga le informazioni generali sul tema da trattare e un set in contenuti aggiuntivi più specifici che ne completino l'esposizione nel corso della presentazione. In ambito didattico è inoltre possibile che venga richiesta una verifica delle competenze acquisite tramite la lezione, motivo per cui è necessario prevedere una sezione adibita a tale scopo.

Quello che l'applicazione non si propone di fare è creare i singoli contenuti. Il progettista deve essere provvisto di tutto il materiale necessario al suo scopo: l'interfaccia svolge esclusivamente il ruolo di supporto per assemblarli.

2.1.1 Limitazioni della situazione attuale

Ai fini della realizzazione del progetto l'azienda Inrebus Technologies ha gentilmente fornito delle bozze relative a un prototipo di strumento da loro impiegato, chiamato "hyperfilm". Questo si componeva di un foglio XML[1] in cui era presente la struttura base e di due file Javascript, scritti in Mootools, per gestire l'assemblamento e le interazioni dei vari componenti grafici. Ogni hyperfilm era costruito localmente tramite i propri file e ciò comportava una serie di limitazioni. Prima fra tutti i clienti avrebbero dovuto affidare contenuti privati a terze parti per realizzare il proprio SmartVideo, rinunciando alla possibilità di personalizzazione del prodotto e vincolandoli a fornire un rapporto dettagliato sulla composizione desiderata. In secondo luogo, essendo puro codice, era necessario incaricare personale competente con conoscenze informatiche, in grado di tradurre il contenuto dei file, per l'inserimento dei vari elementi e delle risorse nella sezione appropriata del progetto. Tra gli aspetti negativi si annovera inoltre l'utilizzo di Mootools[2]: si tratta purtroppo di un framework antiquato difficile da interpretare e modificare se non si possiede una base di programmazione a oggetti. Infine, meno importanti ma comunque rilevanti nella user experience, sono stati riscontrati problemi relativi alle scelte grafiche e al modo in cui venivano visualizzati i contenuti e le risorse associate durante la riproduzione del video.

2.1.2 Analisi dei miglioramenti apportabili

Dal materiale analizzato è stata stilata una lista di obiettivi essenziali per lo sviluppo dello SmartVideo. Primo fra tutti l'esigenza di avere un'interfaccia atta ad assemblare il proprio lavoro, che fosse comprensibile, facilmente interpretabile da qualsiasi utente anche sprovvisto di conoscenze informatiche. Uno strumento che fosse disponibile per tutti, senza affidarsi a terze parti, e che permettesse di personalizzare il proprio prodotto.

A livello strutturale l'interfaccia dovrebbe: prevedere l'introduzione di un database, e di conseguenza la gestione del caricamento dei file,

[1] https://support.microsoft.com/it-it/office/informazioni-di-base-suxml-a87d234d-4c2e-4409-9cbc-45e4eb857d44

[2] https://mootools.net/

in cui salvare ilmateriale e le informazioni d'interesse per facilitare l'organizzazione dei singoli lavori; essere suddivisa in sezioni per consentire all'utente di compiere modifiche mirate; avere un'area distinta dedicata alla visualizzazione dello SmartVideo, sia in fase di sviluppo sia a lavoro terminato.

A livello di contenuto, tra le implementazioni realizzabili a fine formativo, lo sviluppo di una funzione che permetta di aggiungere quiz per monitorare l'acquisizione delle competenze e, facoltativamente, un glossario relativo a termini tecnici o di poco uso impiegati nella lezione. In secondo luogo, la necessità di evidenziare la presenza dei contenuti di approfondimento senza distogliere totalmente l'attenzione dalla lezione. Infine, inserire una sezione opzionale relativa alle informazioni tecniche del progetto per dare lustro ai processi alla base della creazione del materiale.

2.2 Target

Partendo dalle assunzioni fatte nel paragrafo 2.1 è possibile suddividere il target in due categorie: gli utenti-creatori degli SmartVideo e gli utenti-spettatori che ne usufruiscono.

2.2.1 Utenti-creatori

I primi sono rappresentati da dipendenti di aziende e potenzialmente da personale scolastico. Si tratta di una categoria di target formata, competente, che possiede una conoscenza base delle tecnologie attuali ed è abituata all'organizzazione di materiale tecnico e specifico del settore. Gli utenti che rientrano in questa categoria devono poter accedere a tutti i servizi dell'applicazione. L'approccio nella realizzazione dell'interfaccia deve quindi essere schematico e immediato da comprendere per guidare l'utente nell'assemblaggio del prodotto finale.

Persona

Luca è un dirigente aziendale di 45 anni. Lavora nel settore automobilistico da più di 15 anni. Ha un figlio piccolo di 8 anni a cui deve badare nonostante il lavoro occupi gran parte del suo tempo. Ama le auto da corsa e le passeggiate in montagna nei weekend.

User Scenario

Luca è incaricato della formazione dei dipendenti sulle novità proposte dall'azienda. Recentemente gli è stata affidata la promozione di un nuovo prodotto. Ha bisogno di uno strumento attraverso cui istruire i suoi collaboratori sulle caratteristiche del prodotto e dare riferimenti tecnici per le strategie di marketing senza dover essere fisicamente presente. L'azienda gli propone di utilizzare lo SmartVideo, una nuova tecnologia da poco integrata nel sistema aziendale. Essendo un'applicazione web Luca si deve preoccupare solo di caricare i contenuti e le informazioni di interesse, risparmiando tempo prezioso e l'utilizzo di difficili programmi di editing video per inserire notazioni grafiche superflue. Tramite lo SmartVideo Luca può mostrare il video demo sul lancio del prodotto con all'interno note di approfondimento e di riflessione su questioni su cui i suoi colleghi dovranno porre l'attenzione durante lo sviluppo della campagna.

Persona

Giada ha 40 anni e vive a Ravenna. Ha una laurea in scienze della formazione e da quando ha terminato gli studi è diventata professoressa di scienze a tempo pieno nel liceo scientifico locale. I suoi hobby sono il bricolage e la pittura. Si offre spesso come volontaria per gli eventi di raccolta fondi per il centro dei disabili della città.

User Scenario

Giada deve cominciare a spiegare ai suoi alunni in DAD la scissione molecolare e tutti i procedimenti che avvengono nell'organismo umano che portano alla formazione del feto nel grembo della donna. Ha preparato dei video d'esempio, ma avrebbe bisogno di poter monitorare se i suoi studenti stiano effettivamente seguendo la lezione, magari inserendo dei checkpoint durante la riproduzione. Una collega le propone come soluzione l'utilizzo di uno SmartVideo, in quanto rimasta colpita dall'efficienza dello strumento impiegato in una presentazione di un suo superiore durante un corso di aggiornamento. Giada è un po' perplessa perché non crede di riuscire a realizzare un prodotto del genere, soprattutto per la

sua poca dimestichezza con l'informatica. Appena accede all'interfaccia si rende conto però che le istruzioni solo molto chiare e riesce a caricare tutto il materiale necessario senza problemi.

2.2.2 Utenti-spettatori

Riguarda gli utenti che ricoprono il ruolo di fruitori del prodotto. In questa categoria rientrano il personale aziendale e, nel caso della didattica a distanza, gli studenti. Il target è molto più vario del caso precedente, sia per competenze sia per età. Il prodotto finale deve quindi risultare di semplice comprensione a livello di funzionamento e pratico da utilizzare; i contenuti devono essere mostrati in modo chiaro e devono essere facilmente usufruibili e condivisibili.

Persona
Cesare ha 35 anni, vive a Bologna con la compagna e ha da poco cambiato lavoro. È un programmatore esperto in iOS e Android, ma volendo arricchire il suo bagaglio di conoscenze, ha deciso di intraprendere un percorso più improntato sul marketing e la promozione dei prodotti.

User Scenario
Trovandosi in una nuova realtà vorrebbe essere aggiornato sulla situazione attuale aziendale per potersi integrare meglio con i colleghi e iniziare al più presto la sua nuova esperienza lavorativa. Per colmare le lacune nelle sue conoscenze, Cesare deve seguire più di un corso formativo. Durante le lezioni si rende conto che i corsi preparati con il semplice supporto di slide sono complessi e scomodi nel ricercare le informazioni per rispondere alle domande conclusive. L'impressione è totalmente differente, invece, per le lezioni rilasciate tramite SmartVideo: non solo l'utilizzo del video cattura più la sua attenzione, ma la possibilità di mettere in pausa il corso per esplorare in tranquillità i contenuti di approfondimento gli permette un apprendimento a 360° gradi semplice, efficace e più veloce dei metodi tradizionali. Essendo un ambito lavorativo differente dal suo precedente impiego, Cesare è contento di poter accedere in ogni momento a un glossario in cui vengono spiegati termini di uso comune nella nuova realtà, ma a lui sconosciuti.

Persona

Chiara è una studentessa di 17 anni al terzo anno di liceo scientifico e vive a Nichelino. È una ragazza solare e intelligente. Ama trascorrere il tempo libero in compagnia delle amiche e tenendosi in forma facendo ginnastica all'aperto.

User scenario

Purtroppo, con l'arrivo del COVID-19, Chiara ha dovuto rinunciare alla vita quotidiana con i suoi compagni di classe. Le lezioni in DAD sono noiose ed è facile distrarsi. Vorrebbe trovare un modo per rimanere al passo con le lezioni che la invogli a studiare e che non faccia calare i suoi voti. I professori hanno deciso di preparare delle prove facsimile sulle videolezioni degli argomenti trattati fino a quel momento in preparazione delle verifiche e delle interrogazioni. Al loro interno sono state inserite annotazioni importanti e possibili domande per guidare lo studente al ripasso della materia. Lo strumento utilizzato è uno SmartVideo: in questo modo Chiara può rivedere e riascoltare i punti cruciali spiegati dai professori e al tempo stesso verificare le sue conoscenze attuali rispondendo ai quiz. Essendo uno strumento interattivo Chiara si sente più stimolata a esplorare i contenuti e a rafforzare le sue competenze.

2.3 La soluzione: un'applicazione web

Dall'analisi del target e degli obiettivi è emerso che il progetto avrebbe dovuto essere facilmente accessibile e reperibile da diversi utenti, evitando installazioni fastidiose o procedimenti macchinosi. Per questi motivi si è optato per una soluzione interamente online a portata di tutti, facilmente utilizzabile persino per chi non possiede competenze informatiche o computer super performanti, in cui l'utente non entri in diretto contatto con il codice, ovvero un'applicazione web (abbreviato "web app").

Gli aspetti principali con cui le applicazioni web si differenziano dai software "tradizionali" riguardano ad esempio l'assenza di installazione in locale (essendo eseguite direttamente sul browser necessitano solo di autenticarsi sulla piattaforma), la maggior sicurezza dei dati e la disponibilità di aggiornamenti veloci per tutti i clienti trattandosi di operazioni svolte un'unica volta sul server ospitante e non sulle singole macchine degli utenti.

Soprattutto dal punto di vista aziendale, è risultato necessario introdurre un processo di profilazione degli utenti dal momento che vengono condivisi e utilizzati contenuti privati o coperti da copyright per la realizzazione di uno SmartVideo. A tal proposito è essenziale che la sezione dell'applicazione dedicata alla creazione del prodotto sia privata e quindi non visibile ad utenti esterni, a differenza del video finale che deve risultare facilmente condivisibile e accessibile a tutti. Deve essere quindi possibile gestire separatamente contenuti pubblici e privati. Per rispecchiare l'esigenza di semplicità di comprensione dell'interfaccia si è scelto inoltre di basare le varie pagine sull'utilizzo di form e di feedback visivi in risposta alle azioni dell'utente. Soprattutto in ambito business i form rappresentano infatti un modo semplice e efficace per ottenere i dati desiderati dai potenziali clienti [20].

A livello dello SmartVideo finale è stato definito che il video debba presentare due menù interattivi: il primo con riferimento ai contenuti delle singole sezioni dell'applicazione e il secondo relativo all'accesso rapido ai contenuti satellite. In particolare, le scelte organizzative effettuate per i link sono state attuate prendendo spunto dal lavoro svolto dagli studiosi Chambel, Correi e Guimarães relativamente agli *hypervideo* nel web [21]. Dalle ricerche risulta che la presenza di collegamenti ipertestuali all'interno del video aiuti a creare momenti di riflessione per un miglior apprendimento dei contenuti durante la sua fruizione. Gli *hyperlink* individuati possono essere suddivisi in quattro categorie in base al rapporto spazio-tempo all'interno del video principale:

- Link incondizionati: si tratta di link sempre attivi in qualsiasi parte del video;

- Link spaziali: anch'essi sempre attivi, dipendono esclusivamente dalla componente spazio, permettono di suddividere il video in "regioni" all'interno delle quali sono presenti dei collegamenti esterni;

[20] https://blog.hubspot.com/marketing/web-forms#:~:text=Why%20Should%20I%20Create%20Web,until%20it%27s%20ready%20for%20analysis

[21] Chambel T., Correia N., Guimarães, N., *Hypervideo on the Web: Models and Techniques for Video*, in *Integration. International Journal of Computers & Applications*, Acta Press, Vol. 23, #2, 2001

- Link temporali: dipendono solo dalla componente temporale, quindi possono essere attivi per certi intervalli di tempo;
- Link spazio-temporali: dipendono da entrambe le grandezze

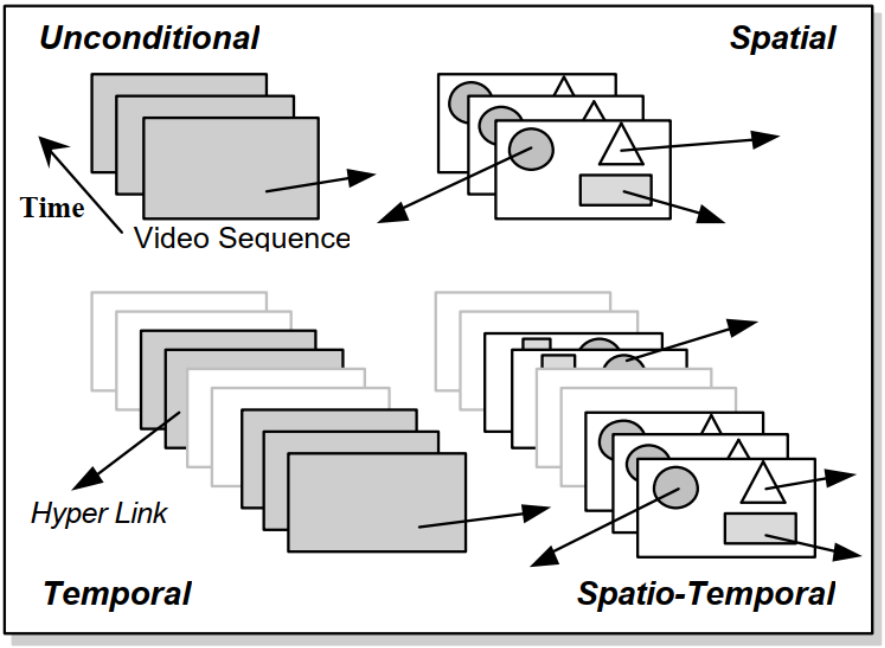


Figura 2.1 - Categorizzazione dei link in base alle condizioni di tempo e spazio [21]

Dai loro studi si evince oltretutto l'importanza della link awareness, ovvero la percezione che l'utente ha dei contenuti satellite. Come affermato nel paragrafo 2.2.2 è importante che la comparsa dei link da un lato non distragga il soggetto, dall'altro lo guidi nella navigazione del prodotto. Per questi motivi si è scelto di impiegare piccoli hotspot temporanei (link spazio-temporali) per la comparsa dei link sul video e un menù esterno al video (link incondizionati) per consentire costantemente un accesso rapido e diretto ai contenuti. A livello di applicazione la sezione riguardante i link e i quiz avrà quindi una funzione tramite cui sarà possibile gestire il posizionamento del contenuto nel video sia temporale, per definire la durata degli hotspot, sia spaziale per determinare la loro collocazione nei frame.

2.4 Requisiti di sistema

Nel corso della realizzazione dell'applicazione si sono sperimentate diverse opzioni sia a livello di organizzazione dei contenuti sia per quanto riguarda i framework impiegati.

La prima versione è stata sviluppata costruendo il database sulla struttura interna del file XML dell'hyperfilm e usando HTML5 per l'organizzazione degli elementi statici nella pagina web e Mootools per le interazioni dell'utente. Non essendo più stato aggiornato negli ultimi anni, il framework presentava diverse lacune e, lunghi procedimenti onerosi nella gestione delle chiamate al database, in particolar modo per il caricamento dei file e i relativi controlli sulla sicurezza dei dati. Essendo l'interazione con il database e la gestione delle risorse uno dei pilastri della nuova applicazione, il framework è stato presto abbandonato.

Una caratteristica positiva di Mootools è possedere una logica a metà fra la classica programmazione a oggetti di Java e la sintassi di jQuery. Motivo per cui il secondo tentativo è stato effettuato mantenendo la struttura statica in HTML dell'applicazione e traducendo i file "dinamici" in Vanilla Javascript. Le interazioni con il database sono state gestite tramite le chiamate ajax con jQuery. Anche in questo caso il progetto è stato scartato per la poca flessibilità del codice, in particolare per problematiche riscontrate nell'implementazione del drag and drop.

La versione finale è stata ottenuta introducendo React per la gestione dei contenuti al front end. I file HTML e i file Javascript sono stati quindi interamente tradotti nella nuova logica a componenti, lasciando invece invariato il database. Il framework essendo molto attuale ha reso decisamente più dinamica e fluida la costruzione della pagina web.

Requisiti di sviluppo

Per la realizzazione dell'applicazione il database è gestito tramite Mysql e Xampp, il cui funzionamento verrà illustrato nel capitolo 3. Lato server le operazioni di interpretazione ed elaborazione delle richieste proveniente dal client sono gestire con php puro nella versione 7.2.34. Lato client il framework impiegato è React (v 17.0.1) con il supporto di Node.js (v 14.15.1), sviluppabile tramite software quali ad esempio Jetbrains Webstorm o simili

come Visual Studio Code.

Requisiti di utilizzo
Per poter essere utilizzata l'applicazione Web implica ovviamente di possedere una connessione a Internet e un pc. I browser testati[3] fino ad adesso sono le versioni recenti di Chrome e Firefox in cui sia installato Javascript e sia supportato HTML5. Temporaneamente la fruizione può avvenire sono tramite computer: l'adattamento responsive dello SmartVideo per mobile e tablet è prevista, ma ancora purtroppo in fase di sviluppo.

2.5 Scelte stilistiche

Figura 2.2 - Versioni del logo SmartVideo

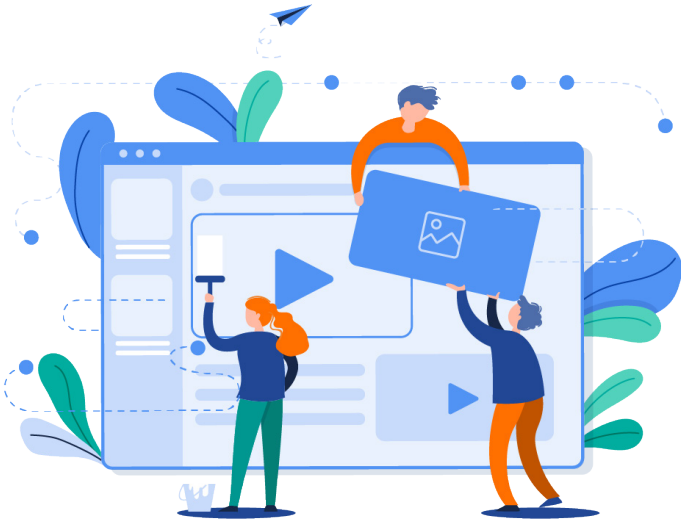


Figura 2.3 - Grafica banner SmartVideo

[3] Testati tramite CanIUse, https://caniuse.com/ciu/index

Le scelte stilistiche alla base dell'applicazione sono state tratte analizzando delle grafiche proposte dall'azienda ai fini del progetto (figura 2.2 e figura 2.3). Per mantenere la coerenza cromatica la palette ricercata è caratterizzata principalmente dal contrasto tra blu e arancio, valorizzandone le singole sfumature e aggiungendo, dove necessario, giochi di opacità per ottenere un miglior effetto visivo. Dall'analisi delle reference si è ottenuta la palette indicata nella figura 2.4.

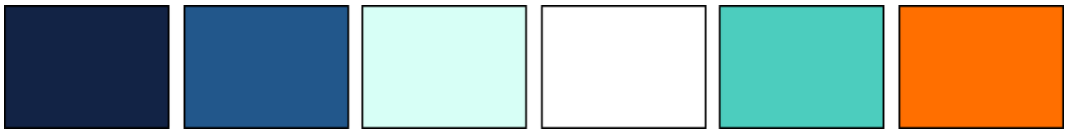


Figura 2.4 - Palette SmartVideo

L'assegnazione delle tonalità ai diversi elementi della pagina è dipesa da studi sugli effetti psicologici provocati dai colori freddi e caldi.

Nel primo caso si tratta di cromaticità associate a sensazioni di tranquillità e serenità, motivo per cui sono stati utilizzati per rappresentare le parti statiche e passive dell'interfaccia, ovvero gli sfondi delle varie videate. Scegliere di utilizzare colori scuri come base dell'applicazione è dovuto anche a ragioni riguardanti l'occhio umano: impiegare tonalità chiare tendenti al bianco stancherebbe facilmente la vista degli utenti.

L'arancione, essendo un colore caldo, è impiegato nelle situazioni in cui è richiesto all'utente di compiere un'azione oppure per rappresentare elementi attivi dell'interfaccia. In generale è utilizzato per stimolare l'attenzione dell'utente. Alcuni esempi di utilizzo di tonalità calde riguardano: la selezione delle voci nella barra laterale, il menù interattivo dello SmartVideo, il bottone per caricare il file relativo al video master.

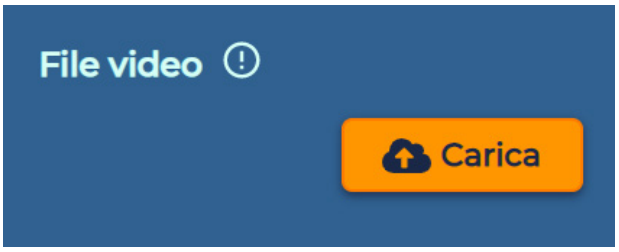


Figura 2.5 - Esempi della grafica parte 1

Figura 2.6 - Esempi della grafica parte 2

Infine, il verde è stato impiegato per elementi neutri, non totalmente passivi ma allo stesso tempo non stimolanti quanto i precedenti. Si tratta di tutti i bottoni relativi alla navigazione dell'utente tra le varie sezioni dell'interfaccia e all'aggiunta di nuovi contenuti quali progetti, link o quiz.

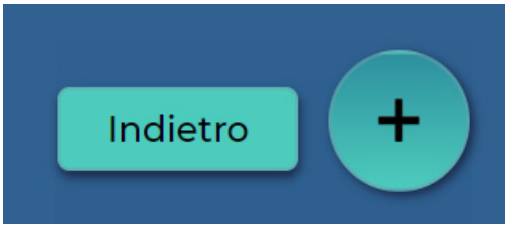


Figura 2.7 - Esempi della grafica parte 3

Sulla base dei loghi della figura 2.2 e della palette sono stati concepiti i concept delle grafiche impiegate per rappresentare progetti, link e quiz.

Figura 2.8 - Logo relativo alla cartella di un progetto

Per raffigurare le cartelle relative ai singoli progetti è stata abbinata alla tradizionale icona della "folder" la pellicola del film per riprendere l'elemento alla base dello SmartVideo, ovvero il video.

Figura 2.9 - Logo relativo agli SmartLink

Nei casi dei link e dei quiz la scelta di distinguere graficamente i due elementi è dovuta al fatto che vengono gestiti all'interno della stessa vista. Per mantenere la coerenza con lo stile dell'interfaccia si è optato per riprendere la versione circolare del logo dello SmartVideo, sia per forma che per colori.

Il tratto caratterizzante è rappresentato dall'immagine posta al centro della circonferenza: una lampadina accesa, come simbolo di curiosità, per identificare i contenuti satellite di approfondimento e un punto interrogativo per i test a risposta chiusa.

Figura 2.10 - Logo relativo agli SmartQuiz

In generale il riferimento principale per la realizzazione dell'interfaccia dal punto di vista grafico è stato Material Design[4], ovvero il sistema di design creato da Google. Tipografia, stile e disposizione degli elementi nella pagina sono stati ispirati al suo modello.

Relativamente alla prima, il font maggiormente impiegato per il progetto è il Montserrat di Julieta Ulanovsky nelle sue versioni Light e Medium. La scelta di font sans serif, ovvero senza grazie, è sempre più comune nelle applicazioni web e nei software caratterizzati da design minimalisti [22] e basati sugli stili di Google. Infatti, oltre a dare un'impronta moderna e attuale al sito, sono diffusamente impiegati nel web per la loro linearità e leggibilità [23].

Una diretta conseguenza dell'ampio utilizzo del modello di Google oggi giorno è la presenza di una serie di "standard" a cui l'utente medio è abituato: comportamenti, azioni, stili che si ripetono sempre uguali nelle applicazioni

[4] https://material.io/design/introduction

[22] https://www.html.it/articoli/i-migliori-font-sans-serif-per-il-web/

[23] https://www.codingcreativo.it/font-per-il-sito-web/

e che permettono una facile comprensione dell'interfaccia. Primo fra tali elementi il bottone "+" in basso a destra che permette di aggiungere contenuti di vario genere nel classico stile circolare, in sovrimpressione rispetto al resto dei componenti della pagina. In secondo luogo, l'uso di ombre per rendere una percezione di profondità tra gli elementi o di cartoline (o *card*) per presentare oggetti singoli o, nel caso specifico dell'applicazione, progetti, link e quiz. Come verrà descritto nel paragrafo 3.3.7 del capitolo 3 relativamente alle librerie utilizzate come supporto al frontend, tramite Material-UI è stato possibile inserire direttamente elementi propri delle grafiche di Google come switch, bottoni, snackbar per i feedback.

L'analisi dello stile ha riguardato anche la disposizione di elementi e di singole voci nella pagina. Nella cultura occidentale la lettura dei documenti avviene partendo da sinistra proseguendo verso destra e dall'alto verso il basso, motivo per cui nella parte inferiore a destra della vista sono di solito posti tutti i bottoni principali come "Conferma", "Salva", "Avanti" o, come citato poc'anzi, "Aggiungi". Per questa ragione è importante la collocazione degli elementi all'interno della pagina.

Particolare attenzione è stata posta ad esempio alle icone delle card dei progetti e dei contenuti satelliti. Nel dettaglio della figura 2.11 la rappresentazione delle possibili azioni relative a un singolo progetto.

Figura 2.11 - Rappresentazione delle icone relative a un singolo progetto

Tale disposizione delle icone porta, infatti, l'attenzione dell'utente sull'opzione "visualizza lo SmartVideo", indicato con il simbolo del play, piuttosto che sull'azione di eliminazione del progetto.

Capitolo 3

Applicazione web

Come affermato nel capitolo 2, il progetto si presenta sotto forma di applicazione web (abbreviato "web app"). Si tratta di un'applicazione ospitata su un server accessibile e usufruibile via web attraverso un sistema di autenticazione, un software applicativo basato su un'architettura client-server, ovvero un'architettura all'interno della quale all'utente ("client") vengono forniti servizi tramite un server esterno al sito, ospitato da una terza parte [24][25]. Rispetto alle applicazioni tradizionali le web app presentano diversi vantaggi, tra i quali, per esempio, ritroviamo l'assenza di installazione dovuta al fatto che, essendo eseguite direttamente sul browser, necessitano solo di effettuare il login; la sicurezza dei dati; aggiornamenti veloci per tutti i clienti trattandosi di operazioni svolte un'unica volta sul server ospitante e non sulle macchine locali.
In particolare, in questo capitolo l'applicazione verrà analizzata sotto tre aspetti fondamentali: l'organizzazione del database, l'architettura del progetto lato server e l'architettura lato client.

3.1 La base dati e il modello relazionale

3.1.1 MySQL e il modello relazionale

Dal punto di vista della gestione dei dati, l'applicazione si appoggia a MySQL, uno dei RDBMS open source e liberi più diffusi al mondo: i dati

[24] https://it.qaz.wiki/wiki/Web_application
[25] https://www.sviluppati.com/2016/04/05/web-application-quali-i-vantaggi/

sono memorizzati e gestiti in tabelle, ognuna delle quali è costituita da righe identificate da un codice univoco detto chiave. Tali tabelle possono essere indipendenti tra di loro o legate da relazioni logiche. In particolare, le relazioni tra due tabelle o entità si differenziano per cardinalità in corrispondenze 1:1, corrispondenze 1:n o corrispondenze n:n, in base alla forza del vincolo che le lega.

3.1.2 InnoDB

Nel caso specifico del progetto si è scelto di utilizzare per MySQL come storage engine InnoDB per il salvataggio dei dati. InnoDb [26] è un sottosistema di archiviazione che esegue le transazioni dei dati in modo indipendente l'una dall'altra, salvandole sul supporto di memorizzazione solo una volta terminate, in modo tale da prevenire accessi incompleti al database. Tra i motivi principali per la scelta di questo storage engine vi sono la possibilità di ripristino dei dati in caso di modifiche errate, grazie a un sistema di salvataggio dei log delle transizioni, e la gestione delle chiavi esterne che, in caso di eliminazione di dati, aiuta l'utente a rispettare l'integrità referenziale del database. Ovviamente il modello relazionale presenta delle limitazioni che saranno trattate nel capitolo conclusivo parlando delle possibili ottimizzazioni future del progetto.

3.1.3 Linguaggio SQL

Per poter interagire con il DBMS è stato utilizzato il linguaggio SQL (Structured Query Language), un formalismo che indica quali operazioni svolgere sul database [27]. Si tratta di un linguaggio dichiarativo in quanto descrive un'azione e non come compierla.

Le funzionalità primarie si possono suddividere in:

- Definizione della struttura della base dati (DDL)
- Manipolazione dei dati (DML)

Entrambe le sottocategorie presentano un set di operazioni determinate; i seguenti listati ne mostrano degli esempi pratici.

[26] https://www.ionos.it/digitalguide/hosting/tecniche-hosting/che-cose-innodb/
[27] https://www.html.it/pag/32137/introduzione-ai-rdbms

```
SET storage_engine=InnoDB;
SET FOREIGN_KEY_CHECKS=0;

CREATE DATABASE IF NOT EXISTS smartvideo;
use smartvideo;

DROP TABLE IF EXISTS PROGETTO;
CREATE TABLE IF NOT EXISTS PROGETTO(
   IdProg VARCHAR(5) UNIQUE NOT NULL,
   Nome_Progetto VARCHAR(255) NOT NULL,
   Proprietario VARCHAR(255) NOT NULL,
   DataCreazione DATE NOT NULL,
   DataUltModifica DATE NOT NULL,
   PRIMARY KEY (IdProg)
);
```

- Esempio linguaggio DDL -

La figura riporta un frammento di codice utilizzato per la creazione del database smartvideo e nel particolare la tabella PROGETTO e la definizione di tutti i suoi attributi e chiavi. Le operazioni DROP TABLE e CREATE TABLE rientrano nel set di azioni possibili per definire la base dati.

```
UPDATE PROGETTO
SET Nome_Progetto='$t'
WHERE IdProg='$p' AND
Proprietario='$ut'
```

(a)

```
SELECT *
FROM PROGETTO
WHERE IdProg='$idP'
```

(b)

```
INSERT INTO PROGETTO
(IdProg, Nome_Progetto,
 Proprietario,
 DataCreazione,
 DataUltModifica)
VALUES('$idP','$t','$ut',
'$dtC','$dtM')
```

(c)

```
DELETE FROM PROGETTO
WHERE Proprietario='$ut'
AND IdProg='$id'
```

(d)

- Operazioni del linguaggio DML -

In questo caso ad essere mostrate sono le operazioni consentite e più frequenti quando si ha a che fare con la manipolazione dei dati, ovvero rispettivamente l'inserimento (*c*), la lettura (*b*), la modifica (*a*) e la cancellazione (*d*), comunemente riassunte dall'acronimo CRUD (Create-Read-Update-Delete).

3.1.4 Organizzazione delle tabelle

A monte della costruzione del database è stata effettuata un'attenta analisi delle caratteristiche e delle funzionalità che avrebbe dovuto possedere un hypervideo per rispondere ai requisiti della formazione aziendale e didattica. Dai risultati del rapporto sono state individuate tre funzioni principali da sviluppare attorno al video master: la gestione di contenuti satelliti che permettano una navigazione non lineare dello storytelling del media di base; la possibilità di verificare le competenze acquisite attraverso quiz a risposta chiusa; in caso di contenuti molto tecnici, avere uno spazio dedicato in cui mostrare all'utente un glossario per guidarlo nella comprensione degli argomenti trattati.

Per rispondere a tali esigenze la base dati è stata strutturata nel seguente modo:

Tabella PROGETTO Tabella cardine dell'interfaccia; al suo interno sono contenute informazioni anagrafiche dei vari progetti (titolo, proprietario, data di creazione) a cui faranno riferimento le altre entità. Si tratta di una tabella costruita ad hoc con informazioni non direttamente modificabili dall'utente, identificato con l'attributo Proprietario, eccezione fatta per il nome del progetto.

Tabella VIDEO Entità in cui viene inserito il video che fungerà da base per lo SmartVideo; oltre al file sono presenti alcune opzioni atte a influenzare la riproduzione del media, tra cui la possibile interruzione forzata alla comparsa di un contenuto satellite.

Tabella LINK Attraverso di essa è possibile agire sui contenuti satelliti sopraccitati. La visualizzazione degli stessi avviene tramite hotspot a comparsa. Per questo motivo, oltre a poter inserire un titolo per singolo link e una descrizione testuale del suo contenuto, è possibile impostare

la posizione temporale (frame di inizio e frame di fine) e spaziale del satellite all'interno dell'hypervideo (coordinate x e y all'interno dei frame selezionati).

Tabella RISORSA I contenuti satellite svolgono la funzione di approfondimento di un determinato argomento. A supporto della trattazione testuale è possibile inserire delle risorse mediali. Per ognuna di esse è specificato il file di riferimento, il formato (immagine, video, audio, URL o documento) del contenuto inserito, un titolo e una didascalia.

Tabella QUIZ Come suggerito dal nome si tratta della tabella all'interno della quale sono gestite le domande da porre durante la riproduzione del video. Allo stesso modo dei contenuti satelliti, la loro comparsa è gestita attraverso hotspot temporanei di cui è possibile impostare il momento di apparizione (frame di comparsa, singolo a differenza dei link) e la posizione all'interno del frame selezionato. I quiz per semplicità si presentano come test a risposta chiusa con un numero di risposte minimo pari a due (es. vero o falso, quiz a risposta multipla).

Tabella RISPOSTA Tabella dedicata alla registrazione delle singole risposte, in cui è possibile indicarne il quiz di riferimento e la correttezza.

Tabella GLOSSARIO Nel caso in cui sia necessario enfatizzare e/o sottolineare termini specifici o tecnici, la tabella si occupa di stilare un elenco di glosse con rispettiva definizione.

Tabella INFO Tabella contenente i dati generali sullo SmartVideo. In aggiunta alle funzionalità essenziali è stato ipotizzato uno spazio dedicato a informazioni a corredo, come ad esempio eventuali contributori e software utilizzati nella realizzazione del progetto.

Tabella USER Entità contenente l'anagrafica degli utenti. Per poter usufruire dell'applicazione sono state previste la registrazione e l'autenticazione dell'utente sulla piattaforma ospitante. Di conseguenza le fasi di signup, verifica delle credenziali e login sono tutte gestite interagendo con la tabella USER. Inoltre, l'username dell'utente corrisponde al valore dell'attributo Proprietario nella tabella

PROGETTO. Attualmente uno SmartVideo può essere modificato solo ed esclusivamente dal suo creatore. Una delle possibili ottimizzazioni potrebbe essere quella di prevedere la condivisione della gestione dei lavori.

La figura 3.1 mostra una rappresentazione concettuale del database attraverso una estrema semplificazione del modello ER (Entità-Relazione) in cui la usuale raffigurazione delle relazioni è stata omessa per mostrare più facilmente la correlazione/corrispondenza tra le chiavi delle tabelle.

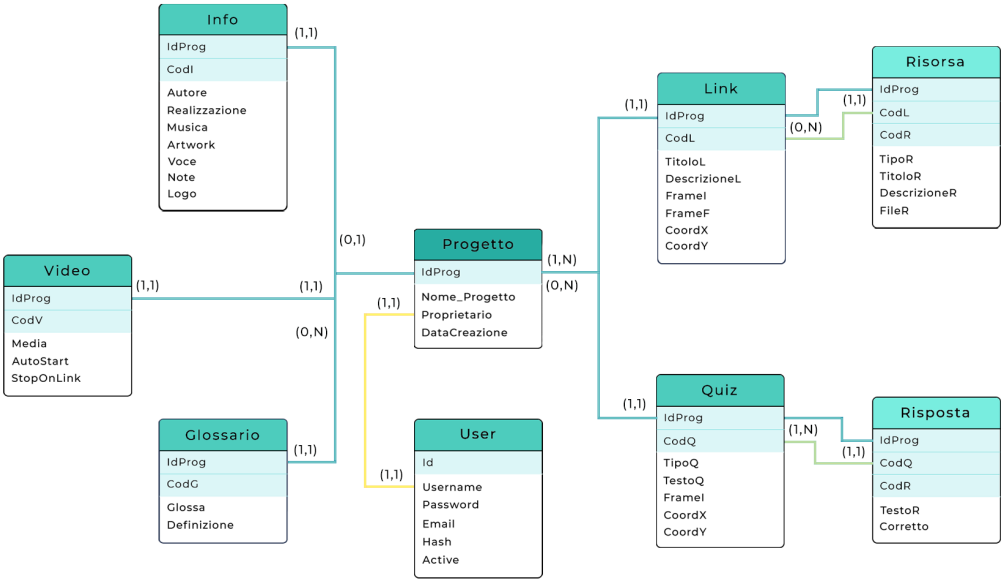


Figura 3.1 - Modello concettuale della base dati

Gli attributi selezionati in celestino rappresentano le chiavi primarie (nota: attributo o un insieme di attributi che identifica in modo univoco le righe o tuple di una tabella. L'aggettivo "primaria" indica che, oltre alle proprietà di chiave, non ammette valori nulli, a differenza delle altre chiavi, denominate chiavi candidate.) delle singole tabelle.

Ognuna di esse è rappresentata da un codice alfanumerico: una lettera, che identifica la tabella di riferimento, e un valore numerico incrementale. Ipotizzando un esempio pratico: nella tabella PROGETTO è presente un progetto P2, avente un contenuto satellite nella tabella LINK identificato dalla chiave P2 – L0 e caratterizzato da due risorse P2 - L0 - R0 e P2 - L0 - R1

della tabella RISORSA. Nello specifico nei casi di glossario, link e quiz i codici sono stati ideati per ricominciare sempre da 0 per ogni progetto, così come le risorse e le risposte per i rispettivi approfondimenti e test, per limitare la crescita repentina dei valori. All'inserimento di una nuova chiave viene calcolato il massimo valore esistente in quel momento e successivamente aumento di una unità.

La struttura della base dati è frutto di un processo di normalizzazione delle informazioni: quando una relazione è nella forma non normalizzata vi è il rischio che si presentino ripetizioni tra tuple e che si verifichino comportamenti poco desiderabili durante gli aggiornamenti [28]. Per questa ragione la suddivisione in tabelle e la scelta delle chiavi è stata pensata per eliminare le possibili ridondanze all'interno del database e per rendere mirate le modifiche allo stesso.

Dallo schema si evince, infine, come le chiavi siano per la maggior parte dei casi composte da un insieme di attributi, in cui almeno uno di essi richiama un attributo di un'altra tabella. Per garantire la correttezza dei riferimenti è stato imposto un vincolo di integrità referenziale.

In generale per vincolo si intende l'insieme delle regole che devono essere rispettate da tutte le righe (tuple) di una tabella. Esistono diversi tipi di vincolo in base al fatto che si stia considerando i valori all'interno di una singola tabella (intra-relazionali) o tra tabelle differenti (inter-relazionali); alcuni esempi sono [29]: vincolo di chiave primaria, vincolo di unicità, vincolo di dominio o di tupla. In questo caso specifico è importante evidenziare il vincolo di integrità referenziale, ovvero la condizione necessaria per cui ogni entità referenziata da un'altra tabella debba esistere nel database [30].

[28] http://www-db.disi.unibo.it/courses/BDPG/normaliz.pdf
[29] https://www.seneta.it/2018/08/linguaggio-sql-vincoli-di-integrita-dei-dati/

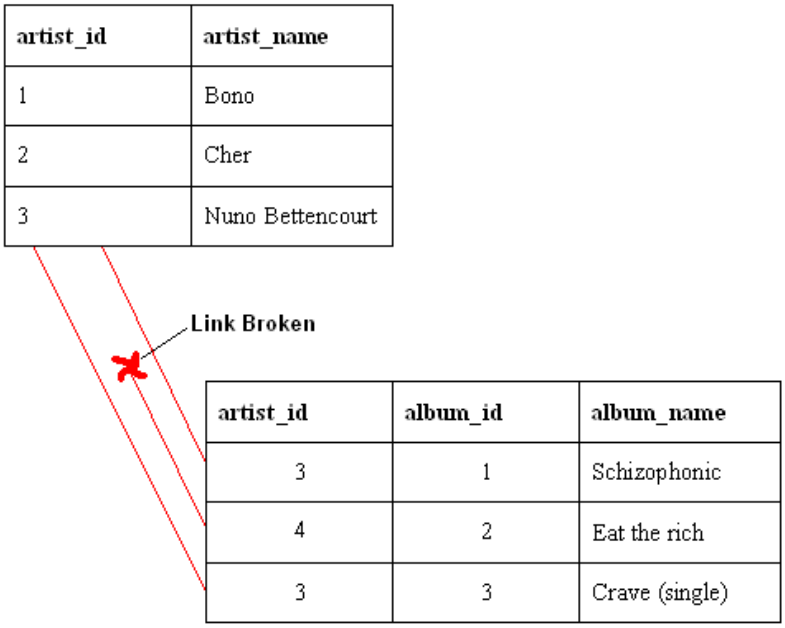


Figura 3.2 - Vincolo d'integrita referenziale violato

La figura 3.2 mostra un esempio di mancata integrità referenziale. La tabella relativa agli album musicali presenta un riferimento all'attributo artist_id della tabella degli artisti. Il vincolo non è rispettato in quanto è stata aggiunta una tupla nella tabella referenziata il cui l'attributo referenziato non ha un corrispondente nella tabella referenziante o di origine.

A livello di codice l'insieme di attributi a cui è applicato il vincolo è rappresentato dalla chiave esterna. Il frammento di codice mostra il modo in cui è stato applicato il vincolo: l'attributo chiave IdProg della tabella INFO referenzia l'attributo IdProg della tabella PROGETTO e in caso di cancellazioni o modifiche all'interno di quest'ultima l'operazione viene propagata anche alla tabella INFO.

[30] https://www.britannica.com/science/computer-science/Information-management#ref1278029

```
CREATE TABLE IF NOT EXISTS INFO(
     IdProg VARCHAR(5) NOT NULL,
     CodI VARCHAR(5) NOT NULL,
     //codice omesso
     PRIMARY KEY (IdProg, CodI),
     FOREIGN KEY (IdProg)
           REFERENCES PROGETTO(IdProg)
           ON DELETE CASCADE
           ON UPDATE CASCADE
);
```

- Dichiarazione del vincolo d'integrità -

Il controllo del vincolo avviene dopo ogni istruzione SQL che potrebbe causarne la violazione, come appunto l'inserimento di tuple non referenziate.

Nel caso opposto, se si verificassero modifiche o cancellazioni nelle informazioni della tabella referenziante allora sarebbe possibile scegliere come queste operazioni vadano a influire sulle tabelle referenziate. Le opzioni attuabili sono [30]:

- CASCADE: permette la propagazione dell'operazione di aggiornamento o cancellazione a tutte le tabelle referenziate;
- SET NULL/DEFAULT: sostituisce in tutte le colonne delle tuple che hanno valori non più presenti nella tabella referenziata null o un valore di default, specificato in precedenza;
- NO ACTION: l'azione di modifica o cancellazione viene impedita, generalmente segnalando all'utente il tentativo di violazione del vincolo attraverso un messaggio di errore.

3.2 Gestione dell'interfaccia lato server

Come affermato nell'introduzione del capitolo, l'applicazione si basa su un'architettura client-server [31]:

- Il client effettua una richiesta a un server remoto e fornisce la risposta ricevuta all'utente.

- Il server, attraverso un linguaggio di scripting, interpreta la richiesta del client e manda indietro una risposta

Nel seguente paragrafo saranno presentati nel dettaglio gli strumenti adottati per la gestione del back end e degli endpoint alla base delle varie funzionalità dell'interfaccia.

3.2.1 Introduzione a Xampp, PHP e Axios

Lato server, per la gestione delle interazioni tra la base dati e il client, è stato utilizzato Xampp. Xampp è considerato uno degli ambienti di sviluppo open source, distribuito da Apache, più popolari attualmente. L'acronimo riporta alcune delle sue componenti principali ed evidenzia il suo aspetto multipiattaforma [32]:

- X – Cross-Platform
- A – Apache
- M – MySQL
- P – PHP
- P – Perl

In particolare, ai fini dell'applicazione, è stata adoperata la componente phpMyAdmin, un tool gratuito in grado di gestire l'amministrazione di MySQL sul Web [33].

Per la parte di scripting, invece, è stato utilizzato PHP. PHP si presenta come un linguaggio installato all'interno del server che, in fase di esecuzione, acquisisce le informazioni ricevute dal client sotto forma di richieste HTTP, grazie al Web server, e le restituisce come risposta dopo averle adeguatamente interpretate [31].

[31] https://www.html.it/pag/16673/cos-php/ 12/03
[32] https://blog.dangerx82.it/joomla/121-cosa-xampp.html
[33] https://www.phpmyadmin.net/

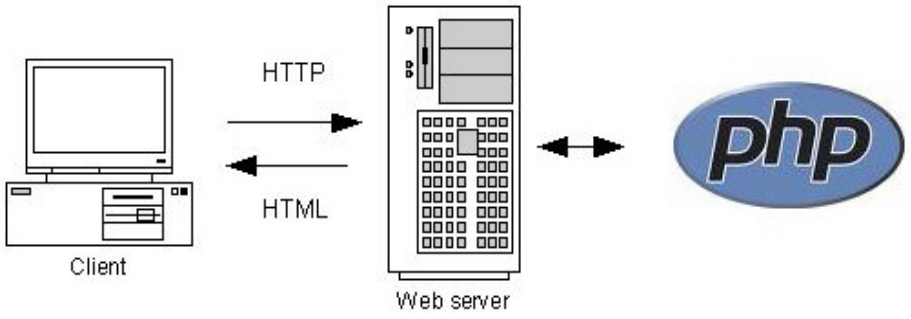


Figura 3.3 - Rappresentazione di una generica richiesta client/server

Il fatto che PHP sia un linguaggio di scripting server side implica che lato client sarà possibile accedere solo ai dati ottenuti tramite richieste HTTP e non direttamente al codice sorgente.

Lato client le interazioni con il server avvengono attraverso axios, una libreria libera e open source per le richieste HTTP basate su promises. Ciò significa che è possibile gestire il successo o il fallimento di un'azione asincrona attraverso handlers e che il valore di completamento si otterrà solo una volta terminata la promise [34]. Gli stati che una promise può assumere sono:

- *pending* (attesa): stato iniziale dopo l'invio della richiesta.
- *fulfilled* (soddisfatto): l'operazione si è conclusa con successo.
- *rejected* (respinto): l'operazione è fallita; viene restituito il messaggio d'errore corrispondente.

A ognuna di queste soluzioni è associato un handler: *then* recupera il valore della risposta, mentre *catch* rileva eventuali errori. Le varie fasi di una promise sono mostrate nella figura 3.4.

[34] https://developer.mozilla.org/it/docs/Web/JavaScript/Reference/Global_Objects/Promise

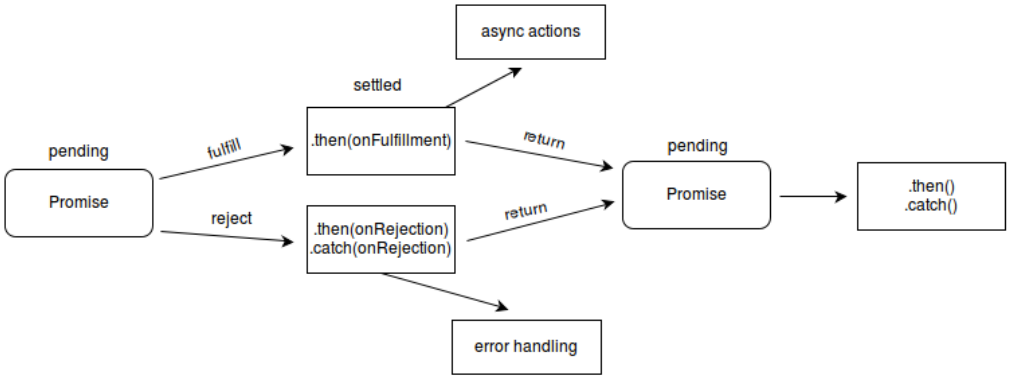


Figura 3.4 - Schema delle fasi di una promise

Axios permette di inviare richieste HTTP asincrone a URL esterne all'applicazione e di eseguire le operazioni CRUD. Per citare un esempio pratico, il formato standard di una richiesta POST tramite axios è

```
axios.post(url[, data[, config]])
```

La risposta ottenuta dal server si presenta come un oggetto costituito dai seguenti attributi [35]:

- data: contiene i dati effettivi ricevuti dal server
- status: il codice HTTP relativo alla risposta, rappresentato da un numero che può assumere valori compresi tra 100 e 509
- statusText: il messaggio relativo al codice ricevuto (ad es. 200 operazione andata a buon fine, 400 Bad Request)
- headers: lista delle header inviate dal server
- config: configurazione originale della risposta
- request: l'oggetto richiesto

All'interno del progetto axios è stato importato come libreria all'interno di React e Node.js, che saranno approfonditi successivamente nel paragrafo dedicato all'analisi dell'interfaccia lato client.

[35] https://zetcode.com/javascript/axios/

3.2.2 Gestione delle richieste HTTP attraverso i Redux Thunk

Entrando nell'analisi pratica dell'applicazione, l'interfaccia è costruita in modo tale da avere un continuo dialogo con il server per mostrare i dati costantemente aggiornati in seguito alle azioni dell'utente. La fase di login sarà temporaneamente tralasciata in quanto verrà approfondita nella sezione 3.2.3 parlando della gestione della creazione dell'account dell'utente e della sua autenticazione.

Come verrà spiegato nel paragrafo 3.3 relativo alla gestione dell'interfaccia lato client, l'applicazione è interamente gestita da Redux, una libreria associata a framework di javascript, come React, che permette di salvare lo stato globale dell'applicazione all'interno di uno store [36]. Per poter agire su di esso esistono particolari funzioni chiamate reducer. Ognuno di essi è caratterizzato dall'old state dello store e da un'azione, che indica come quest'ultimo debba essere aggiornato, e viene attivato tramite il dispatch dell'azione. Generalmente questo processo avviene in modo sincrono e non è indicato per gestire le comunicazioni con una API esterna o nel caso in cui siano presenti effetti collaterali [37]. A tal proposito Redux mette a disposizione dei middleware che permettono, ad esempio, di aggiornare lo stato dello store in seguito a una chiamata asincrona.

Ai fini dell'applicazione i middleware impiegati sono stati i Redux Thunk. Le azioni principali svolte dai thunk possono essere riassunte in: fetch o lettura delle informazioni del database, crea/aggiorna una voce ed elimina voce. Di seguito è mostrato un esempio di funzionamento del thunk relativo al recupero dei progetti di un dato user, analizzando i passaggi dalla creazione della richiesta HTTP alla ricezione della risposta del server.

La parte di set up per poter utilizzare un Redux Thunk consiste nell'importare la libreria axios e dichiarare lo stato iniziale dei progetti all'interno dello store.

[36] https://redux.js.org/introduction/getting-started
[37] https://www.digitalocean.com/community/tutorials/redux-redux-thunk
[38] https://redux.js.org/tutorials/essentials/part-5-async-logic

```
import axios from "axios";
const initialState = {
   projects: [],
   status: 'idle',
   error: null
}
```

- Impostazione dello stato -

Successivamente è possibile definire il corpo del middleware. Il metodo *createAsyncThunk* permette di generare un thunk che gestisce automaticamente le chiamate delle azioni relative allo stato della promise. Accetta due argomenti: una stringa che rappresenta il prefisso delle azioni sopraccitate (anche detta action generator per questa ragione) e una funzione callback che contiene una Promise [38]. A prova di quanto detto, se la chiamata andasse a buon fine verrebbe effettuato il dispatch dell'azione *projects/fetchProjects/fulfilled.*

```
export const fetchProjects = createAsyncThunk('projects/
  fetchProjects',
  async (utente,thunkApi) => {
     try{
         const response = await
         axios.post('http://localhost/smartvideo_app/php/
            progetti.php', utente)
         return response.data
     }catch (error){
         return thunkApi.rejectWithValue(error.response)
     }
})
```

- Definizione del corpo del Thunk -

La funzione asincrona presenta a sua volta due argomenti. Il primo riguarda i dati passati come parametro al momento del dispatch: rappresentano le informazioni inviate nel corpo della richiesta HTTP, che verranno interpretate all'interno dell'endpoint indicato dall'URL. Il secondo valore, invece, permette di ritornare messaggi d'errore descrittivi specifici in caso di fallimento.

Dal momento che il thunk non conosce a priori quale sarà il risultato della promise e di conseguenza l'action chiamata, è possibile gestire i vari casi costruendo una logica simile a quella del reducer, all'interno della quale viene aggiornato lo stato salvato nello store in base alle caratteristiche dell'action corrispondente. Prendendo in considerazione l'esempio della richiesta andata a buon fine, il comportamento dell'azione *projects/fetchProjets/fulfilled* è descritto dal frammento di codice *[fetchProjects.fulfilled]*: la variabile project memorizzata nello store verrà aggiornata con la lista dei progetti ricevuti dalla promise.

```
const projectsSlice = createSlice({
  name: 'projects',
  initialState,
  extraReducers: {
       [fetchProjects.pending]: (state, action) => {
           state.status = 'loading'
       },
       [fetchProjects.fulfilled]: (state, action) => {
           state.status = 'succeeded'
           state.projects = state.projects.concat(action.
                            payload)
       },
       [fetchProjects.rejected]: (state, action) => {
           state.status = 'failed'
            alert( state.message = action.payload.statusText
               +' '+action.payload.status+': '+action
               .payload.data)
           state.error = action.error.message
```

- Gestione della risposta della promise -

Il metodo POST indica che è possibile inviare al server un set di dati in forma chiave/valore necessari per la corretta interpretazione della richiesta [39]. Ad esempio, nel momento in cui viene effettuato il dispatch di *fetchProjects*, viene passato come parametro un form data contenente le informazioni utili al recupero della lista dei progetti dell'utente collegato.

```
const fd = new FormData()
fd.append('token', username)
dispatch(fetchProjects(fd))
```

- Chiamata della funzione *fetchProjects* -

Il valore della costante "fd" all'interno del dispatch corrisponde al parametro "utente" indicato nella funzione asincrona del Thunk.

Lato server, il file *progetti.php* riceve il valore "token" passato all'interno del form data, lo salva in locale e apre la connessione al database.

```
//controllo sui dati ricevuti dal client, con gestione di
eventuali errori
if (!isset($_POST['token'])) {
     http_response_code(400);
     echo json_encode("Indicare un'utente specifico");
}else{
    $ut = $_REQUEST['token'];

//apertura connessione al db
  $con = mysqli_connect('localhost','root','','smartvideo1');
 if (mysqli_connect_errno()){
   #die ('Failed to connect to MySQL: ' . mysqli_connect_
         error());
   echo "Failed to connect to MySQL: "
         . mysqli_connect_error();
  }


//istruzione di selezione dei progetti
  $sql_check = "SELECT *
                FROM PROGETTO
                WHERE Proprietario='$ut'";
  $res_check=mysqli_query($con,$sql_check);

//traduzione del risultato della query in un oggetto JSON
  $rows = array();
```

[39] https://www.html.it/pag/62463/le-richieste-http-get-e-post/

```
  while($r = mysqli_fetch_assoc($res_check)){
     $rows[]=$r;
  }

  echo json_encode($rows);

//chiusura connessione
  mysqli_close($con);
}
```

- Gestione della richiesta HTTP lato server -

La lettura della tabella PROGETTO avviene attraverso l'istruzione SELECT in cui viene specificato il valore dell'attributo Proprietario per ottenere la lista dei progetti creati e gestiti esclusivamente dall'utente desiderato. Il risultato della query viene poi inserito all'interno di un array e successivamente tradotto in oggetto JSON prima di essere inviato al client. In caso l'operazione termini con un successo, la lista dei progetti verrà aggiunta alla lista *projects* salvata nello stato.

3.2.3 Gestione degli endpoint dell'interfaccia

Dopo aver effettuato il login, l'utente è reindirizzato al suo spazio personale all'interno del quale è possibile aggiungere, modificare, eliminare i singoli progetti o visualizzare gli SmartVideo corrispondenti. L'interfaccia che ne gestisce la creazione e la modifica è stata strutturata in modo tale da avere una sezione editabile per ogni elemento dell'hypervideo.

Tutte le interazioni che comportano un salvataggio di informazioni all'interno della base dati sono state gestite come nell'esempio precedente: client side è presente un Redux Thunk tramite cui è possibile modificare lo stato globale dell'applicazione, mentre server side viene implementata una delle istruzioni CRUD. Lo schema riassume per ogni sezione la tabella del database a cui fa riferimento e le azioni che si possono compiere su di essa:

- sezione "I miei progetti" (tabella PROGETTO):
    - lettura del database per il recupero della lista dei progetti (SELECT)

  - inserimento di un nuovo progetto (UPDATE)
  - eliminazione di un progetto (DELETE)

- sezione "Dettagli del progetto" (tabella INFO):
  - lettura della tabella (SELECT)
  - aggiunta (INSERT), in caso di nuovo progetto, o aggiornamento (UPDATE), in caso siano già presenti, delle informazioni
  - aggiunta (INSERT) o aggiornamento (UPDATE) del file relativo al logo dell'azienda

- sezione "Impostazioni del video" (tabella VIDEO):
  - lettura della tabella (SELECT)
  - aggiunta (INSERT) o aggiornamento (UPDATE) delle impostazioni
  - aggiunta (INSERT) o aggiornamento (UPDATE) del file relativo al video master del progetto

- sezione "SmartLink" (tabelle LINK e RISORSA):
  - lettura della tabella LINK (SELECT)
  - aggiunta (INSERT) o aggiornamento (UPDATE) del contenuto di un link
  - aggiunta (INSERT) o aggiornamento (UPDATE) della posizione di un link
  - eliminazione di un contenuto satellite (DELETE)
  - lettura della tabella RISORSA (SELECT)
  - aggiunta (INSERT) dei file relativi alle risorse del link
  - eliminazione di una risorsa (DELETE)

- sezione "SmartQuiz" (tabelle QUIZ e RISPOSTA):
  - lettura della tabella QUIZ (SELECT)
  - aggiunta (INSERT) o aggiornamento (UPDATE) del contenuto di un quiz

  - aggiunta (INSERT) o aggiornamento (UPDATE) della posizione di un quiz
  - eliminazione di un contenuto satellite (DELETE)
  - lettura della tabella RISPOSTA (SELECT)
  - aggiunta (INSERT) di una nuova risposta
  - eliminazione di una risposta (DELETE)

- sezione "Glossario" (tabella GLOSSARIO):
  - lettura della tabella GLOSSARIO (SELECT)
  - aggiunta (INSERT) o aggiornamento (UPDATE) di una voce e della sua definizione
  - eliminazione di una glossa (DELETE)

- sezione "SmartVideo"
  - fetch dei dati relativi al progetto
  - fetch delle informazioni
  - fetch delle caratteristiche del video principale
  - fetch dei link e delle risorse a essi associate
  - fetch dei quiz e delle risposte
  - fetch del glossario

Nel caso si acceda a un progetto creato in precedenza, la lettura a priori del database permette di mostrare i dati relativi all'ultima modifica salvata. Effettuare il fetch delle informazioni come prima azione è altresì importante per la gestione delle chiavi nei casi di elementi multipli, come i link e i quiz, per un singolo progetto.

Di seguito è mostrato il ciclo delle operazioni applicate nel caso dei link. (nota: le parti analoghe all'esempio relativo a fetchProjects sono state omesse per semplicità).

1. Fetch dei contenuti satelliti (link)

```
dispatch(fetchContents(form_data))
export const fetchContents = createAsyncThunk('contents/
   fetchContents',
   async (project,thunkApi) => {…})
(…)
[fetchContents.pending]: (state, action) => {
   state.hyperlink_status = 'loading'
},
[fetchContents.fulfilled]: (state, action) => {
   state.hyperlink_status = 'succeeded'
   state.links = state.links.concat(action.payload)
   let l = state.links.length
   if(l>1) {
       state.links = state.links.sort((a, b) => {
           let c1 = a.CodL
           let c2 = b.CodL
           return c1.localeCompare(c2)
       })
       state.maxL = state.maxL.filter(q => q.idp !== action.
                    payload[0].IdProg)
       state.maxL.push({idp: state.links[l - 1].IdProg,
                        cod: state.links[l - 1].CodL});
   }else if(l===1){
      state.maxL = state.maxL.filter(q => q.idp !== action.payload.
                   IdProg)
      state.maxL.push({idp: action.payload[0].IdProg,
                        cod: state.links[0].CodL});
   }else{
       state.maxL = state.maxL.filter(q => q.idp !== state.proj)
       state.maxL.push({idp: state.proj, cod: 'L0'});
   }
},
[fetchContents.rejected]: (state, action) => {
   state.hyperlink_status = 'failed'
   alert( state.message = action.payload.statusText +'
                '+action.payload.status+': '+action.payload.data)
   state.error = action.error.message
},
```

- Algoritmo riassuntivo dei passaggi nel Thunk -

Al momento del salvataggio in locale dell'elenco dei link è effettuato un controllo sul suo contenuto per determinare il valore della chiave del prossimo link inseribile. Ricordando la figura 3.1 la chiave primaria della tabella LINK è composta da due attributi: il codice del progetto di riferimento e un valore univoco identificativo del singolo contenuto. Di conseguenza, per ottenere il valore massimo inserito è sufficiente ordinare in ordine alfabetico la lista dei link per il loro codice: la chiave ottenuta viene successivamente memorizzata nello store insieme al progetto di riferimento sotto forma della variabile maxL.

Sul server l'operazione di lettura delle informazioni avviene attraverso un controllo incrociato tra le chiavi delle tabelle LINK e PROGETTO in rispetto del vincolo d'integrità citato nel paragrafo 3.1.4.

```
//salvataggio dei dati ricevuti dal client
$p = $_REQUEST['projectid'];
$u = $_REQUEST['token'];

/* connessione al database */

//istruzione di lettura condizionata della tabella LINK
$sql_check = "SELECT *
           FROM LINK, PROGETTO
           WHERE LINK.IdProg = PROGETTO.Idprog AND LINK.IdProg='$p'
              AND Proprietario='$u'";
```

- Recupero dell'elenco dei link lato server -

2. Inserimento e aggiornamento di un contenuto satellite

Quando l'utente desidera inserire un nuovo link, viene calcolato il valore della sua chiave lato client, leggendo il contenuto della variabile *maxL* relativa a il progetto selezionato e aumentandolo di una unità. Di conseguenza all'interno della richiesta HTTP saranno sicuramente presenti: il codice del progetto, l'utente che sta effettuando l'operazione e la chiave del link che si vuole creare o modificare.

```
const fdata = new FormData()
fdata.append('chiave', link.CodL)//chiave calcolata in locale
fdata.append('projectid', project.IdProg)
fdata.append('titolo',formRef.current.values.titolo)
fdata.append('descrizione',formRef.current.values.descrizione)

dispatch(inviaContent(fdata))

export const inviaContent = createAsyncThunk(
   'contents/inviaContent',
    async (data,thunkApi) => {
       try{
           const res = await
           axios.post('http://localhost/smartvideo_app/php/
               nuovo_contenuto.php', data)
           return res.data[0]
       }catch(error){
           return thunkApi.rejectWithValue(error.response)
       }
   }
)
```

- Richiesta di inserimento di un nuovo progetto -

In caso la richiesta termini con successo, il nuovo link viene aggiunto alla lista; in particolare, se si tratta di un'operazione di aggiornamento la voce relativa al link viene rimossa dall'elenco per essere reinserita con le modifiche effettuate.

Sul server le operazioni di INSERT e UPDATE sono state gestite all'interno dello stesso endpoint *nuovo_contenuto.php*. Per determinare quale delle due istruzioni debba essere applicata viene fatto un controllo sulla tabella LINK: se esiste una tupla avente come chiave la coppia (id_progetto, codice_link) verrà applicata l'UPDATE, altrimenti si procederà con l'inserimento di una nuova voce.

```
$p = $_REQUEST['projectid'];
$c = $_REQUEST['chiave'];
$ti = $_REQUEST['titolo'];
$d= $_REQUEST['descrizione'];

 //query di controllo
 $sq = "SELECT * FROM LINK WHERE IdProg='$p' AND CodL='$c'";
 $rq = mysqli_query($con,$sq);


 if(mysqli_num_rows($rq) > 0){ --> esiste già la tupla relativa
     $sm = "UPDATE LINK                                al link
           SET TitoloL='$ti',DescrizioneL='$d'
           WHERE IdProg='$p' AND CodL='$c'";
     /*…*/


 }else{ --> creazione di una nuova voce
     $sql = "INSERT INTO LINK (IdProg,CodL,TitoloL,DescrizioneL)
           VALUES('$p','$c','$ti','$d')";
      /*…*/
 }
```

- Aggiunta e aggiornamento di un link -

In entrambi i casi, per ottenere i dati aggiornati nella risposta al client è necessario effettuare un'istruzione di SELECT e seguire i passaggi di conversione del formato del risultato illustrati nell'esempio relativo al fetchProjects. Il frammento di codice mostra solo le principali operazioni CRUD connesse all'azione di inserimento/aggiornamento. In generale, è importante tenere presente che spesso sono effettuati controlli aggiuntivi a fronte di alcuni ragionamenti logici: uno stesso utente non può essere proprietario di due progetti identificati dallo stesso nome, così come, nel caso specifico che si sta analizzando, non ha senso avere due contenuti satelliti con lo stesso titolo perché causerebbe solo confusione nell'utente finale.

3. Eliminazione di un link dalla base dati

```
dispatch(deleteContent(fd)

export const deleteContent = createAsyncThunk(
   'contents/deleteContent',
   async (proj,thunkApi) => {
       try{
           const response = await
           axios.post('http://localhost/smartvideo_app/php/
              elimina_contenuto.php',proj)
           return response.data[0]
       }catch(error){
           return thunkApi.rejectWithValue(error.response)
       }
   }
)
```

- Richiesta di eliminazione di un link -

Server side prima di poter eliminare la tupla dalla tabella, ne viene verificata l'effettiva esistenza. In caso positivo, i dati del link vengono salvati e tradotti nell'oggetto JSON che verrà inviato al client e solo successivamente viene applicata l'istruzione di DELETE.

```
$s_check = "SELECT*
     FROM LINK
     WHERE IdProg='$p' AND CodL='$c'";
$r_check = mysqli_query($con,$s_check);
if(!$r_check){
     http_response_code(400);
     echo json_encode("La voce selezionata non esiste nel db");
}else{
     $risultato = array();
     while($riga = mysqli_fetch_assoc($r_check)){
         $risultato[]=$riga; }
$sql_del = "DELETE FROM LINK WHERE IdProg='$p' AND CodL='$c'";
$res=mysqli_query($con,$sql_del);
```

- Eliminazione di un link -

Se l'operazione di eliminazione del contenuto dal database termina in successo, i dati contenuti nella risposta del server vengono utilizzati per selezionare e rimuovere il link dalla lista dei contenuti satellite.

In generale, gestire l'interfaccia interamente con l'ausilio di Redux presenta alcune limitazioni, tra cui la perdita di tutte le informazioni presenti nello store al refresh della pagina. Per ovviare a questo problema sono presenti, all'interno di ogni sezione, delle chiamate axios indipendenti dai thunk che garantiscono la corretta visualizzazione dei dati qualora fosse necessario ricaricare la pagina.

3.2.4 Gestione della registrazione e del login dell'utente

Per poter creare uno SmartVideo, all'utente è richiesto di registrarsi e successivamente di autenticarsi. Trattandosi di un prototipo la parte riguardante la sicurezza e il trattamento di dati sensibili è stata solo abbozzata per mostrare l'idea di base, ma non implementata completamente. Gli step caratterizzanti questa prima fase sono: registrazione dell'account, verifica della email tramite codice OTP e accesso con le credenziali create.

Per potersi iscrivere l'utente è tenuto a inserire un indirizzo email, che utilizzerà per autenticarsi, un username identificativo e una password. Alla conferma delle informazioni inserite, i dati sono inviati sul server all'endpoint "…/smartvideo_app/php/signup.php".

All'interno del file le prime operazioni riguardano il controllo dell'email e dell'username: in caso siano già presenti nel database una notifica di errore viene inviata al client. Superata la fase di verifica viene aggiunto l'account nella tabella USER.

```
$e=$_REQUEST['email'];
$u=$_REQUEST['username'];
$p=$_REQUEST['password'];
$hash = rand(1000,5000);

$s = "INSERT INTO USER (Username,Password,Email,Hash)
     VALUES ('". mysqli_escape_string($con,$u) ."',
          '".mysqli_escape_string($con,password_hash($p,PASSWORD_
           DEFAULT)).".',
           '".mysqli_escape_string($con,$e)."',
           '".mysqli_escape_string($con,$hash)."')";
```

Per quanto riguarda la sicurezza dei dati sono stati fatti alcuni piccoli accorgimenti preliminari in vista della sua implementazione definitiva.

Tra questi l'utilizzo della funzione mysqli_escape_string evita che si inseriscano inavvertitamente dei caratteri speciali che potrebbero causare dei malfunzionamenti all'interno della base dati, come l'apice, il doppio apice o lo slash, tenendo in considerazione il set di caratteri specifico della connessione che è stata aperta [40]. Basti notare l'utilizzo costante dell'apostrofo o singolo apice per definire ciascuna delle istruzioni php.

In secondo luogo, la password non è salvata in chiaro, ma subisce un processo di criptatura prima di essere inserita nella base dati: la funzione password_hash riceve come parametri la password inserita dall'utente e l'algoritmo con cui si desidera criptare la stringa (nel caso specifico del progetto è stato impiegato l'algoritmo bcrypt[1]).

L'account creato sarà quindi costituito dai seguenti elementi:

- Id_utente incrementale
- Indirizzo di posta
- Username identificativo
- Password criptata
- Codice hash: sequenza numerica casuale di quattro cifre impiegata per la verifica dell'account
- Flag del valore di 1 o di 0 a seconda che l'account sia già verificato o meno

Se l'inserimento avviene con successo viene inviata una mail di verifica delle credenziali all'indirizzo specificato dall'utente da un indirizzo di posta creato ad hoc ai fini del prototipo, come mostrato nella figura 3.5.

[1] funzione di hashing di password progettata da Niels Provos e David Mazières, basata sulla cifratura Blowfish e presentata a USENIX nel 1999, https://it.wikipedia.org/wiki/Bcrypt

[40] https://www.w3schools.com/php/func_mysqli_real_escape_string.asp

```
$to = $e; // indirizzo email dell'utente
$subject = 'Signup | Verifica account'; // oggetto della mail
$message = 'Grazie per aver preso parte alla community Smartvideo!
          Il tuo account è stato creato, puoi effettuare il login
          con le seguenti credenziali dopo aver inserito il codice
          di verifica.

          ------------------------
          Username: '.$u.'
          Password: '.$p.'
          ------------------------

          Il codice di verifica è: '.$hash.' ';
$headers = 'From:smartvideoassistant@gmail.com' . "\r\n";

mail($to, $subject, $message, $headers)//funzione di invio del
                                        messaggio
```

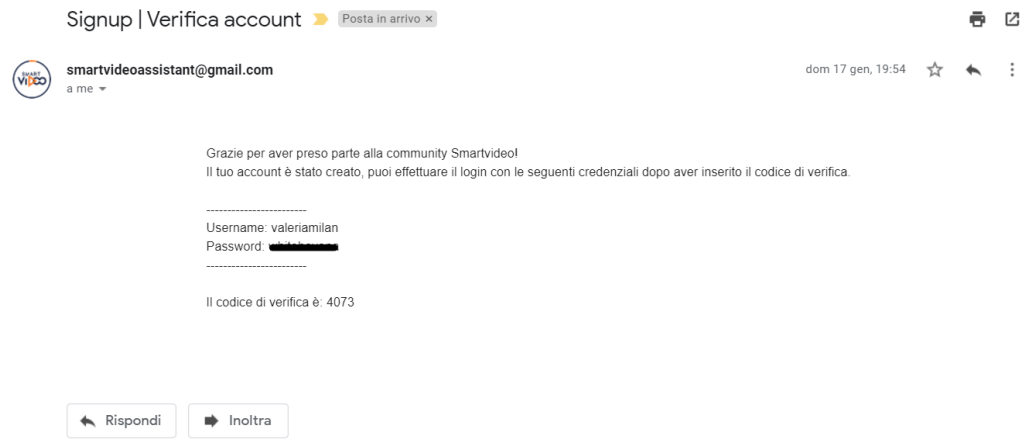


Figura 3.5 - Screenshot email di verifica

La fase di verifica dell'account è stata implementata sfruttando la tecnologia dell'OTP (One Time Password). Si tratta di un codice temporaneo e valido per una singola sessione impiegato comunemente per poter accedere a un sistema protetto o per gestire delle transazioni. Essendo temporaneo, dopo esser stato utilizzato non compromette le operazioni effettuate neanche in caso di intercettazione da parte di intrusi [41].

Se il codice inserito dall'utente corrisponde al valore dell'attributo Hash viene aggiornato lo stato dell'attributo Active da 0 a 1 ed è possibile procedere con la fase di login; in caso contrario verrà mostrato un messaggio

[41] https://www.trendoo.it/blog/codice-otp-via-sms/

d'errore all'utente.

L'ultimo step della fase preliminare per poter usufruire dell'applicazione web consiste nell'effettuare il login. Dopo i dovuti controlli su username e indirizzo email, per verificare il valore della password è utilizzata password_verify, una funzione che accetta due parametri, la password inserita dall'utente e la password criptata presente nella base dati, e che restituisce un valore booleano per identificare il successo o il fallimento dell'operazione. Per evitare di far autenticare l'utente ogni qualvolta che desidera accedere alla sua area personale, si è optato per l'utilizzo di un token per mantenere la sessione aperta. Generalmente il token è rappresentato da un codice temporaneo che viene creato al momento del login e che viene memorizzato all'interno del localStorage del browser che si sta utilizzando. Per il caso specifico dell'applicazione questo meccanismo è stato simulato utilizzando l'username dell'utente, restituito al client se il controllo della password è superato con successo. Particolarmente importante è il suo impiego nella gestione del routing tra pagine private e pubbliche, affrontato nel paragrafo dedicato a React Routing.

3.2.5 Gestione del caricamento dei file

Tra gli aspetti caratterizzanti dell'interfaccia, uno su cui porre l'attenzione riguarda il caricamento dei file da parte degli utenti. A differenza delle normali informazioni testuali, il trattamento di risorse complesse necessita dei dovuti accorgimenti. È infatti possibile inserire l'oggetto intero all'interno del database, ma ciò comporterebbe sia un aumento della dimensione della base dati, sia tempi più lunghi nel caricamento della pagina web che ospita la risorsa. Generalmente, la soluzione adottata consiste nel fare l'upload del file su una directory all'interno del server relativa a una cartella predefinita (ad esempio per le risorse, la cartella di riferimento è raggiungibile grazie alla directory "/resources") e memorizzare nella base dati solo il suo nome, ovvero un valore di tipo stringa [42].

[42] https://www.codexworld.com/upload-store-image-file-in-database-using-php-mysql/

Nell'applicazione questa operazione è effettuata quando si desidera salvare:

- Il video master alla base dello SmartVideo
- Le risorse dei contenuti satelliti (fatta eccezione per quelle di tipo URL)
- Il logo dell'azienda nella sezione delle informazioni del progetto

Il caricamento del file, lato client, avviene attraverso l'utilizzo di un input di tipo file all'interno di un form. Dopo esser stato inviato tramite form data al server, il file si presenta nel formato [43]

```
[file1] => Array
        (
            [name] => MyFile.txt
            [type] => text/plain
            [tmp_name] => /tmp/php/php1h4j1o
            [error] => UPLOAD_ERR_OK  (= 0)
            [size] => 123
        )
```

ed è accessibile tramite l'istruzione php *$_FILES['nome_input_file']*. Nel caso si volesse, ad esempio, controllare la dimensione della risorsa è possibile ottenere il suo valore tramite *$_FILES['nome_input_file']['size']*.

Il frammento di codice mostra i passaggi per la memorizzazione del video principale all'interno della directory "/videos" del server e il salvataggio nella tabella VIDEO della base dati.

```
$uploadDir = 'videos/'; //directory nella quale si desidera
                          caricare il file
$allowTypes = array('mp4', 'webm', 'ogg'); //formati consentiti

if (!empty($_FILES['bgvideo'])){
   $v = trim($_FILES['bgvideo']['name']); //file inviato dal client
    $fileNameM = basename($v);
    $uploadedFile = '';
    $targetFilePath = $uploadDir . $fileNameM;
```

[43] https://www.php.net/manual/en/reserved.variables.files.php

```
// recupero del formato del file
$fileType = pathinfo($targetFilePath, PATHINFO_EXTENSION);
$fileType = strtolower($fileType);

$e = '.'.$fileType;
$new_fileName = $p.'_'.$chiave.$e; //nome con cui verrà salvato
                                    nel server
$targetFilePath = $uploadDir . $new_fileName;

//controllo del formato
if(in_array($fileType, $allowTypes)){
  //caricamento di una copia del file sul server
    if(move_uploaded_file($_FILES['bgvideo']
                          ["tmp_name"],$targetFilePath)){
       $uploadedFile .= $fileNameM;

       //inserimento del nome nella base dati
       $sqlvid="INSERT INTO VIDEO (IdProg,CodV,Media,Estensione)
                VALUES ('$p','$codV', '$v','$e')";
  }
}
```

- Spostamento del file nella directory *videos/* -

Prima di essere salvato nella cartella specifica del server, il file è stato rinominato utilizzando la chiave primaria della tupla in cui è stato memorizzato. Il fine alla base di questa scelta è garantire una corrispondenza univoca tra il file inserito nella pagina web client side e quello presente sul server. Con un esempio pratico: l'utente desidera visualizzare all'interno del suo SmartVideo, indicato con il codice progetto P1, un video denominato "myvideo.mp4". Nella base dati la tupla corrispondente alle informazioni del video ha chiave (P1,V0). A seguito dei dovuti controlli di formato il file viene copiato all'interno della directory "/videos" con il nome P1_V0.mp4, mentre all'interno della tabella VIDEO è inserita la voce "myvideo.mp4" in formato stringa; per poter facilitare l'accesso alla risorsa client side è inoltre prevista una colonna dedicata al salvataggio dell'estensione del file.

Se le operazioni si concludono positivamente, il server invia come risposta al client tutte le informazioni relative al video caricato. Per inserire il file, salvato nella directory, all'interno della pagina web è sufficiente creare

un oggetto <video> e ricostruire il percorso della source con le informazioni ricevute dal php, come mostrato nel frammento di codice.

```
<video id='smartvid'>
 <source src={ http://localhost/smartvideo_app/php/videos/'+video.
    IdProg+'_'+video.CodV+video.Estensione}/>
</video>
```

- Definizione dell'elemento video nella pagina web -

La URL del video sopraccitato risulterà essere quindi

`http://localhost/smartvideo_app/php/videos/P1_V0.mp4`

3.2.6 Gestione degli errori

Durante l'elaborazione dei dati ricevuti dalla richiesta HTTP del client è possibile che si verifichino degli errori. Quando si presenta questa condizione è necessario interrompere le operazioni sul server e restituire un messaggio di fallimento all'utente. Lato server la funzione utilizzata per tale scopo è *http_response_code* attraverso cui è possibile impostare lo status della response manualmente. I codici che indicano la presenza di errori nella richiesta del client sono quelli compresi tra 400 (*Bad Request*) e 451 (*Unavailable For Legal Reasons*) e vengono scatenati qualora i dati inviati non siano corretti o falliscano le istruzioni di aggiornamento del database. Prendendo d'esempio il file *login.php* oltre alla definizione del codice è possibile associare un messaggio testuale con la spiegazione dell'errore.

```
if(password_verify($p, $enc_p)){
     //la password inserita dall'utente è corretta
}else{
     http_response_code(400);
     echo json_encode('La password non è corretta.');
}
```

- Definizione dell'errore nel corpo del costrutto if -

In questi casi lato client è scatenata l'azione relativa allo stato della promise *rejected*. Come accennato nel paragrafo 3.2.2, all'interno dei Redux Thunk le situazioni di fallimento sono gestite grazie al *thunkApi*: la funzione

rejectWithValue propria del thunkApi permette di recuperare i messaggi d'errore definiti nel php e di impostarli come payload della risposta del server. Di seguito è mostrato come viene gestito e tradotto visivamente il messaggio relativo all'inserimento di una password errata al momento del login.

```
export const logIn = createAsyncThunk(
   'user/login',
   async (user, thunkApi) =>{
       try {
           const response = await
              axios.post('http://localhost/smartvideo_app/php/
                          login.php', user)
           return response
       } catch (error) {
           return thunkApi.rejectWithValue(error.response)
       }
   }
)
(…)
[logIn.rejected]: (state, action) => {
   state.loginStatus = 'failed'
   state.message = action.payload.statusText +' '+action.payload.
                   status+':'+action.payload.data
},
```

- Gestione dell'errore lato client -

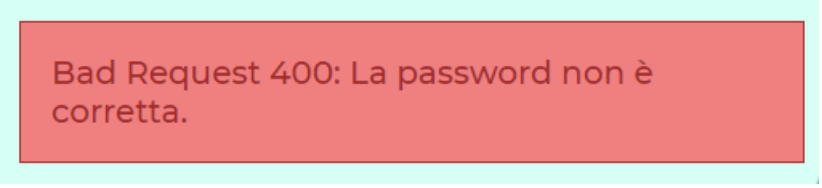


Figura 3.6 - Visualizzazione grafica dell'errore

3.3 Interfaccia web

3.3.1 Single page Application

Il progetto è stato realizzato come Single Page Application (SPA), ovvero un'unica pagina web in cui vengono caricati dinamicamente i contenuti tramite script. Un'applicazione che opera sul browser e le cui risorse, necessarie per la navigazione (codice HTML, Javascript e CSS), vengono caricate un'unica volta appena si visita la pagina [42][43]. Per questo motivo, a fronte della riduzione della latenza nella visualizzazione delle viste secondarie, il primo caricamento della pagina potrebbe risultare più o meno lento in base alle dimensioni dell'applicazione e alle ottimizzazioni delle prestazioni.

La tecnologia SPA è molto diffusa al giorno d'oggi, utilizzata da alcuni giganti del Web come Gmail, Google Maps o Facebook, ed è consigliata se si desidera sviluppare ad esempio piattaforme basate su Saas[2] o comunità chiuse in cui è richiesto un login per poter accedere, come i portali universitari [44]. Inoltre, grazia alla loro configurazione, che prevede una gestione separata di front end e back end, è possibile avere il back end condiviso tra mobile app e web app, riducendo di gran lunga i tempi di sviluppo o consentendo la condivisione (cercherei un'altra parola perché sopra c'è condiviso) dei dati. D'altro canto tra gli aspetti negativi sono presenti la minor sicurezza ad attacchi rispetto alle app tradizionali, l'indicizzazione SEO, una certa difficoltà nel valutare il tracking per la *web analytics* e la necessità di avere Javascript abilitato sul browser [45]. Contrapposte alle SPA esistono le Multi-Page Application (o MPA) che, a differenza delle prime, seguono un approccio più tradizionale: la scelta di utilizzo di una rispetto all'altra dipende sostanzialmente dai fini dell'applicazione.
Dal punto di vista del codice può essere implementata con vari framework di Javascrit, come AngularJS, Vue Js o React, e solitamente si compone di un unico file html che svolge il ruolo da contenitore all'interno del quale la struttura del DOM viene riempita dinamicamente in base alle interazioni

[2] https://www.redhat.com/it/topics/cloud-computing/what-is-saas#:~:text=Software%-2Das%2Da%2DService,infrastruttura%20IT%20che%20la%20supportano.

[44] https://www.aalpha.net/blog/know-about-single-page-application/
[45] https://www.tagmanageritalia.it/glossario-single-page-application/

dell'utente.

3.3.2 React: concetti base

Il framework scelto per implementare l'applicazione lato client è React. Si tratta di una libreria Javascript, open source, sviluppata ad hoc da Facebook per lo sviluppo di interfacce grafiche. Si basa su un approccio modulare e dichiarativo simile all'HTML e per questo facile da imparare ed applicare [46]. Con React è possibile sviluppare non solo applicazioni web, ma anche mobile app, grazie alla versione Native. Inoltre, è facile eseguire il debug e i test sulle viste. Basato sullo standard ES6 (ECMAScrypt 6), il linguaggio utilizzato da React è JSX (Javascript Syntax eXtension), ovvero una variante del Javascript, simile all'HTML per la sintassi, non interpretabile nativamente da qualsiasi browser. Per questo motivo è necessaria la presenza di un precompilatore che permetta, in fase di sviluppo, la traduzione del codice sorgente in una sintassi comprensibile da un browser generico. Gli strumenti incaricati di svolgere tale funzione sono Babel e Webpack [47].

3.3.3 Componenti

Il codice sorgente è organizzato in moduli indipendenti e potenzialmente riutilizzabili; in altre parole è costituito da componenti. I componenti sono blocchi di codice che vengono costruiti dinamicamente a partire da due fattori: dati immutabili forniti in ingresso e stato interno. I primi sono definiti come *properties*, o più comunemente props. Le props sono caratterizzate da nomi univoci e possono assumere la forma di qualsiasi tipo di dato definito da Javascript. All'interno del componente si può accedere al loro valore in sola lettura, in quanto tutti i componenti React devono comportarsi come funzioni pure rispetto alle proprie props [48].

In generale, una componente è definita attraverso una funzione (*stateless component*) o una classe (*stateful component*) e può contenere al suo interno altre componenti [49]. Nel caso specifico analizzato, l'intera applicazione è

[46] https://www.html.it/pag/55052/react-introduzione/

[47] https://siddharthac6.medium.com/getting-started-with-react-js-using-webpack-and-babel-66549f8fbcb8

[48] https://it.reactjs.org/docs/components-and-props.html

costituita da componenti stateless.

Di seguito è mostrato il modello con cui sono stati definiti i componenti dell'interfaccia. L'esempio riguarda il componente "TimeAgo", impiegato per visualizzare, all'interno di un progetto di SmartVideo, il tempo trascorso dalla sua creazione.

```
//librerie importate
import React from "react";
import {parseISO, formatDistanceToNow} from 'date-fns'

export const TimeAgo = ({timestamp}) => {
   //operazioni Javascript
   let timeAgo = ''
   if(timestamp){
       const date = parseISO(timestamp)
       const timePeriod = formatDistanceToNow(date)
       timeAgo = `${timePeriod} ago`
   }

   return( //parte del componente visualizzato nella pagina web
       <span title={timestamp} style={{fontSize: '0.8em'}}>
                <i>{timeAgo}</i>
       </span>
   )
}
```

- Definizione della componente TimeAgo -

Il nome che identifica il componente è univoco e solitamente è caratterizzato dall'iniziale maiuscola. Il codice è diviso in tre blocchi principali: le prime righe segnalano le librerie e i componenti importati, necessari alla definizione di TimeAgo; il secondo racchiude delle istruzioni Javascript per l'elaborazione dei dati; infine, il terzo rappresenta il contenuto ritornato e visualizzato effettivamente nella pagina, definito in JSX. Il parametro timestamp rappresenta la prop del componente e assume come valore il dato che viene passato al momento della chiamata del componente, ovvero, in questo caso, la data di creazione del progetto.

[49] https://www.educative.io/edpresso/what-is-a-react-component

```
import {TimeAgo} from "../features/projects/TimeAgo";

function Project({project}){
    return(
        <div>
             <TimeAgo timestamp={project.DataCreazione}/>
        </div>
    )
}
```

- Chiamata della componente TimeAgo dentro alla componente Project -

Come accennato all'inizio del paragrafo, ogni componente si presenta come un blocco indipendente e isolato dal resto del codice. Per poter essere utilizzata e inclusa in altre componenti è necessario esplicitare le istruzioni di *export*, ove definita, e *import* dove necessaria.

Per concludere la presentazione dei componenti è importante sottolineare la loro mutabilità nel tempo. Ogni componente è infatti caratterizzato da un ciclo di vita. L'istruzione attraverso cui vengono generati è *ReactDOM.render(<Componente/>, nodo_radice)* come mostrato dal frammento di codice.

```
ReactDOM.render(
  <React.StrictMode>
     <Provider store={store}>
         <App />
       </Provider>
  </React.StrictMode>,
  document.getElementById('root')
);
```

- Definizione istruzione generatrice -

Il nodo radice "root" fa riferimento al div principale, vuoto, situato nel file HTML alla base della SPA: l'operazione di render permette di riempire la pagina principale con il contenuto del componente App e di tutti i suoi figli.

Il ciclo è costituito in totale da tre fasi, ognuna delle quali caratterizzata da un metodo specifico (elementi verdi nella figura 3.7):

- Mounting, o fase iniziale di creazione. Corrisponde al momento in cui la funzione viene eseguita per la prima volta e il contenuto renderizzato viene inserito all'interno dell'albero DOM;
- Updating. Le componenti sono caratterizzate dalle props e dallo stato interno; il mutamento di questi fattori nel tempo rende la componente dinamica, causando la ri-esecuzione della funzione e di conseguenza anche l'aggiornamento dell'albero DOM;
- Unmounting, o fase di eliminazione. Corrisponde all'evento in cui un componente è rimossa dalla struttura dell'albero.

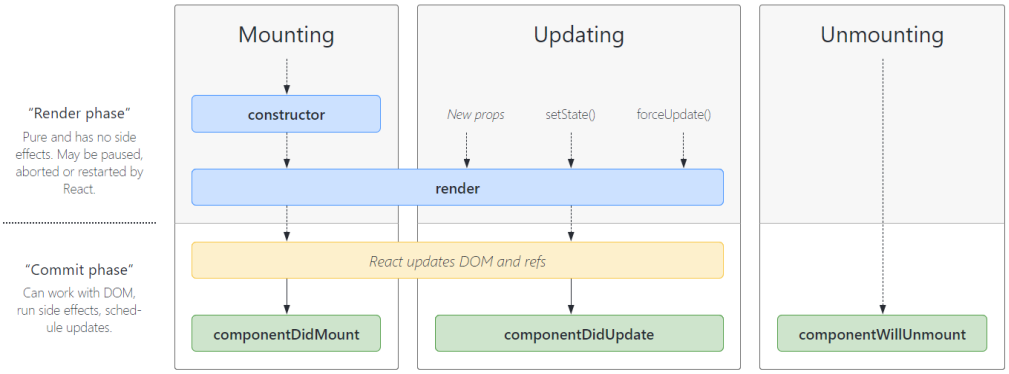


Figura 3.7 - Ciclo di vita di un componente[3]

3.3.4 React Hooks

Gli sviluppatori di React si sono accorti che con il procedere degli aggiornamenti si stavano presentando alcuni problemi di una certa rilevanza, impossibili da ignorare. In primo luogo, utilizzare i componenti classi può creare confusione soprattutto in chi è novizio nell'uso del framework, in quanto sottintende una conoscenza pregressa di costrutti Javascript, come ad esempio il "this". Inoltre, possono portare il programmatore a utilizzare dei patterns che allungano e rendono più complesso il lavoro, invece di ottimizzarlo. Un altro problema è la gestione di componenti stateful complessi con i metodi di lifecycle: sono operazioni che spesso contengono

logiche non del tutto correlate tra loro e per questo possono introdurre bug ed inconsistenze. In molti casi non è neppure possibile suddividerli in blocchi più piccoli e compatti e ciò ne rende difficile la fase di testing. Infine, un altro aspetto importante consiste nella difficoltà di riutilizzare la logica dipendente dallo stato in componenti diversi.

Per questi motivi sono stati introdotti con la versione 16.8.0 gli Hooks. Si tratta di funzioni particolari che incapsulano al loro interno parti non funzionali come la memorizzazione e la gestione dell'evoluzione di uno stato o l'esecuzione di effetti collaterali [50]. Generalmente si riconoscono per il prefisso use- e non possono essere utilizzati all'interno dei componenti classi, ma rappresentano piuttosto una loro alternativa. Fungono da supporto quando si ha a che fare con lo stato dei componenti e come quest'ultimo può influenzare l'aggiornamento dell'albero DOM. Non è necessario usufruirne e non comportano alcun cambiamento che possa danneggiare le funzioni già esistenti, sono cioè retrocompatibili. I React Hook permettono di riutilizzare la logica stateful senza dover manipolare la gerarchia di componenti e rendono possibile la suddivisione di un componente in funzioni più piccole basate su logiche correlate.

Per utilizzarli è necessario rispettare due regole molto importanti:

- Gli Hooks non possono essere chiamati all'interno di costrutti condizionali, cicli o funzioni annidate, ma solo al livello più alto;
- Non è possibile inserire un Hook all'interno di funzioni Javascript pure, ma sono in funzioni React, ovvero i componenti.

[3] https://projects.wojtekmaj.pl/react-lifecycle-methods-diagram/

[50] Materiale corso Interfacce Grafiche

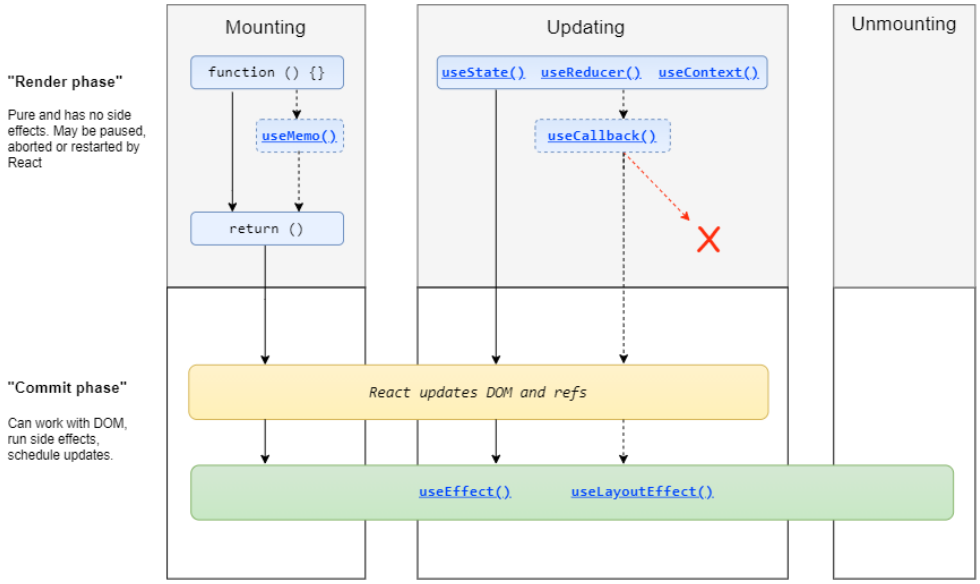


Figura 3.8 - Ciclo di vita dei React Hook[4]

I React Hooks utilizzati ai fini dell'applicazione sono *useState* e *useEffects*.

useState

Il primo Hook analizzato è useState. Si tratta di un costrutto che permette di mantenere uno stato locale all'interno dei componenti funzioni, prima presente solo all'interno delle classi. La differenza sostanziale consiste nel numero di variabili stato dichiarabili nello stesso momento. I due frammenti di codice mostrano rispettivamente lo stato dichiarato all'interno di una classe e l'uso di useState all'interno di un componente funzione.

```
class Message extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      message: '',
      list: [],
    };
  }
  /* ... */
}
```

- Componente classe -

```
import React, { useState } from 'react';

const Message= () => {
   const [messageState, setMessageState] = useState( '' );
   const [listState, setListState] = useState( [] );
}
```

- Componente funzione -

Mentre per una classe è possibile gestire più variabili insieme all'interno dello stato, nelle funzioni a ognuna è associato un Hook distinto [51]. La funzione *useState* riceve in ingresso un dato arbitrario e restituisce una lista contenente rispettivamente lo stato corrente e una funzione necessaria a modificarlo. Un motivo per preferire l'Hook alle classi è la flessibilità e la facilità d'uso intrinseca dell'array rispetto all'oggetto della classe.

Il valore restituito sarà, al primo render del componente, lo stato iniziale, altrimenti quello relativo all'ultimo aggiornamento dello stato corrente. Esistono due modi per modificare lo stato: il primo consiste semplicemente nel passare in ingresso un nuovo valore alla funzione; se invece si desidera utilizzare lo stato precedente è necessario definire una arrow function che, dato il vecchio valore, produce il nuovo stato. Relativamente all'esempio precedente:

Metodo 1 – `setMessaggeState('nuovo messaggio')`

Metodo 2 – `setMessageState( prev_msg  => prev_msg + 'something')`

È altresì probabile che più componenti annidati necessitino di utilizzare lo stesso Hook. In tali situazioni risulta conveniente dichiararlo nel componente padre e definire funzioni locali che ne condizionino il comportamento nei componenti figli. Di seguito viene mostrato un esempio pratico di quanto detto relativamente alla visualizzazione delle risorse dei contenuti satelliti all'interno dello SmartVideo.

[4] https://wavez.github.io/react-hooks-lifecycle/

[51] https://blog.logrocket.com/a-guide-to-usestate-in-react-ecb9952e406c/

```
import React, {useState} from "react";

function PreviewRisorsa({risorsa, mostra}){ //componente figlio
   return(
       <div className='risorsa_modale'>
          <div onClick={()=> mostra(false)}> //al click si chiude
              <MdClose className='chiudi'/>
          </div>
          <div className='risorsa_modale_content'>
               {contenuto}
          </div>
       </div>
   )
}
function RisorsaElenco({risorsa}){ //componente padre
   const [mostraContenuto,setMostraContenuto] = useState(false)
   const mostra = (m) => setMostraContenuto(m)
   return(
     <div>
        <div onClick={()=>setMostraContenuto(true)}> //al click si
           {icona} {risorsa.TitoloR}                  apre
        </div>
        {mostraContenuto && <PreviewRisorsa risorsa={risorsa}
                                mostra={mostra}/>}
     </div>
   )
}
```

- Comparsa di un componente gestita con useState -

In particolare, il contenuto delle risorse è mostrato in overlay sullo SmartVideo, all'interno di una finestra modale. Per gestire la visibilità di questa finestra è stato definito, dentro RisorsaElenco, uno stato *mostraContenuto* inizialmente impostato a false (contenuto della risorsa non visibile). La comparsa/scomparsa è veicolata dalla funzione *setMostraContenuto*.

[52] https://dmitripavlutin.com/react-useeffect-explanation/
[53] https://it.reactjs.org/docs/hooks-effect.html

Cliccando sul nome della risorsa lo stato passa da *false* a *true* e si apre la finestra con la preview: per chiuderla è necessario interagire con un elemento dichiarato all'interno di *PreviewRisorsa*, ovvero la "X".

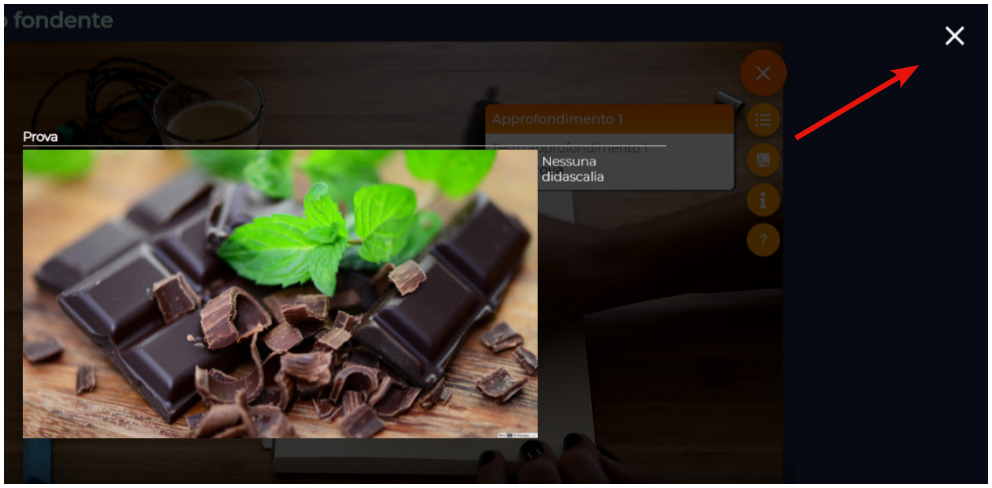


Figura 3.9 - Esempio pratico di funzionamento dell'Hook

Per aggiornare lo stato *mostraContenuto*, definito nel componente padre, da dentro il componente figlio è stata costruita una funzione ad hoc "mostra" che viene passata come prop a *PreviewRisorsa*. Il clic sulla "X" scatena indirettamente l'evento *setMostraContenuto* e chiude la preview. Riassumendo la funzione "mostra" è definita nel componente padre e viene implementata nel componente figlio.

useEffect

Durante lo sviluppo di un'applicazione è probabile che sia necessario eseguire operazioni del tipo il fetch di dati esterni, manipolare l'albero DOM o usare funzioni temporali come *setTimeout()*. Questi casi sono comunemente indicati con il termine side effects in quanto possono alterare altri componenti e non possono essere eseguiti durante la renderizzazione: il render del componente e la logica dei side effects sono indipendenti l'uno dall'altro [52][53]. L'Hook che si occupa della gestione e dell'esecuzione degli effetti all' interno dei componenti funzione è useEffect (nelle classi React lo stesso ruolo era svolto dalle funzioni *componentDidMount*, *componentDidUpdate* e *componentWillUnmount*): essendo dichiarati dentro la funzione React hanno diretto accesso alle props e al suo stato. Gli effetti vengono eseguiti a ogni renderizzazione, inclusa la prima.

La sintassi dell'Hook è rappresentata nel frammento di codice

```
useEffect(callback[, dependencies])
```

Il primo parametro è una funzione callback che contiene la logica dell'effetto specifico, mentre il secondo è costituito da un array opzionale che rappresenta le condizioni per cui l'effetto si avveri. A seconda del valore delle dependencies si possono verificare tre situazioni differenti [52]:

- Dipendenze non specificate; l'effetto è eseguito sempre a ogni render del componente;
- [] array vuoto; l'effetto viene eseguito una sola volta dopo il primo render;
- [prop1, prop2, …, stato1, stato 2] array di props o stati del componente; l'effetto è eseguito solo quando tutti i valori passati come dependencies variano.

Alcuni effetti possono restituire un valore all'interno della callback. Nel caso sia ritornata una funzione, React la interpreta come una funzione di cancellazione o cleanup dell'effetto, che verrà eseguita sempre come prima azione qualora si attivasse nuovamente l'effetto.

Di seguito sono presentati due esempi di utilizzo di useEffect all'interno dell'applicazione, relativi a due casi sopraccitati di dipendenze.

```
const [video, setVideo] = useState({})

useEffect(()=> {
    const fpr = new FormData()
    fpr.append('token', token)
    fpr.append('projectid', id)
    axios.post('http://localhost/smartvideo_app/php/
               recupero_video.php',fpr)
         .then((response)=>{ setVideo(response.data[0])})
         .catch((error)=> alert(error))
},[])
```

- Chiamata per recuperare le informazioni sul video di base al primo render del componente -

```
useEffect(() => {
   if (glossaryStatus === 'idle'){
       const fd = new FormData();
       fd.append('projectid',project.IdProg)
       dispatch(fetchGlosses(fd))
   }
},[glossaryStatus, dispatch])
```

- Chiamata di *fetchGlosses* condizionata -

Negli esempi la funzione callback è costituita da una chiamata asincrona al server. In particolare, il primo caso si riferisce alla lettura delle informazioni sul video per la realizzazione dello SmartVideo. L'operazione viene effettuata una sola volta, come indicato dalla dependency dell'Hook, e il risultato della richiesta HTTP viene utilizzato per aggiornare lo stato video del componente, attraverso *setVideo()*. Il secondo caso invece si riferisce alla lettura delle glosse di un progetto: l'effetto è eseguito ogni qualvolta che una glossa viene aggiunta o eliminata dall'elenco in seguito alle interazioni con l'utente.

useReducer

Un ulteriore Hook su cui porre l'attenzione è useReducer. A livello dell'applicazione non è esplicitamente utilizzato, ma la logica alla base del suo funzionamento serve da introduzione a Redux (paragrafo 3.3.5). È impiegato quando la complessità dello stato cresce e non è possibile gestirla interamente con useState, come ad esempio in casi in cui bisogna tenere traccia di molte informazioni che dipendono l'una dall'altra. Un caso pratico consiste nel carrello della spesa: il totale della spesa dipende dal numero di prodotti inseriti, quindi necessita una gestione più solida e strutturata.

Vengono introdotte le azioni, ovvero delle operazioni che hanno come fine quello di modificare lo stato del componente. Sono identificate da un type, generalmente una stringa che ne indica il comportamento, e da altri parametri opzionali. Stati e azioni sono legati all'interno di particolari funzioni chiamate reducer o riduttori: dato lo stato precedente e un'azione, la funzione ritorna lo stato successivo. Il risultato dell'operazione risulta essere un nuovo oggetto contenente le informazioni del "vecchio stato" che non sono state modificate e i dati aggiornati in seguito all'azione. Lo stato precedente e l'azione rimangono quindi immutati. Di seguito è mostrato come potrebbero essere gestiti alcune funzioni di Monopoli attraverso un reducer.

```
//Definizione dello stato iniziale o "precedente"
const InitialState = {
   player: [
       {
           id:1,
           avatar:'candela',
           casella: 'via!',
           denaro: 1000,
           terreni: []
       },
       {
           id:2,
           avatar:'bottiglia',
           casella: 'via!',
           denaro: 1000,
           terreni: []
       }
   ],
```

```
   turno: 1,
   costruzioni:{
       'vicolo corto': 0,
       'vicolo stretto':0,
   },
}
```

```
//Definizione delle azioni
const compra = {type:'compra', terreno: 'Viale dei giardini',
               player: 1,costo: 200}
const vendi = {type: 'vendi', terreno:'corso…', valore: 230,
               player: 1}
```

```
//Definizione del reducer
function reducer(oldState, action){
   switch (action.type){
       case 'compra':{
           return {
               ...oldState,
               player: //modifiche allo stato player
       }
       case 'edifica':{
           return {
               ...oldState,
               player: //altre modifiche allo stato player
           }
       }
   }
}
```

- Esempio del Hook Reducer applicato al Monopoli -

Per eseguire un'azione bisogna chiamare il metodo dispatch e passare come argomento il nome dell'azione interessata.

Nel paragrafo 3.3.5 il concetto di reducer verrà ampliato per poter gestire lo stato dell'intera applicazione, e non più dei singoli componenti, grazie alla libreria Redux.

3.3.5 Redux

Redux è una libreria Javascript open source, creata nel 2015 da Dan Abramov e Andrew Clark. Il suo scopo principale è quello di ottimizzare la gestione di applicazioni molto grandi e complesse.

Come visto finora, i componenti sono caratterizzati da due elementi, le props e lo stato interno, e possono essere organizzati in gerarchie alla cui radice è presente il componente padre <App />. È plausibile che a un certo punto lo state di un componente figlio sia necessario a componenti dello stesso livello o di livello superiore. Una prima soluzione sarebbe quella di gestire i dati di interesse al componente padre, che racchiude tutti gli altri, e passarli ai figli come props, ma risulterebbe troppo oneroso per applicazioni molto complesse e superfluo per tutti i componenti figli intermedi non interessati a tali informazioni. Redux risolve il problema introducendo uno store, separato dalla struttura principale, contenente lo stato globale dell'applicazione. In questo modo i dati sono accessibili per tutte le componenti individualmente. I principi alla base della libreria possono essere riassunti in:

- Lo stato globale dell'applicazione è memorizzato in un oggetto ad albero all'interno dello store;
- Lo stato è accessibile in sola lettura, a meno che non venga eseguita un'azione;
- Le modifiche allo state tree vengono effettuate tramite speciali funzioni pure, ovvero i reducer.

Redux, in sintesi, prende i concetti dell'Hook Reducer e li estende aggiungendo la memorizzazione dello stato nello store.

Un altro elemento utile per descrivere il funzionamento di Redux è rappresentato dagli slices: una collezione di logiche reducer e azioni relativa a una singola feature dell'app. In Redux ai reducer non è permesso di modificare lo stato corrente direttamente. Per far ciò sono necessari i cosiddetti *immutable updates* che restituiscono lo stato aggiornato in un oggetto differente rispetto a quello contenente lo stato precedente. Generalmente queste operazioni sono difficili e complicate da scrivere a mano. Grazie ai Redux Slice risulta più facile curare questi aspetti. Al loro

interno vengono inoltre gestiti i Thunk che, come affrontato nel paragrafo 3.2.2, svolgono il ruolo da tramite tra front end e back end.
Un componente può eseguire essenzialmente due operazioni nei confronti dello store:

- accedere a un dato memorizzato nello store;
- chiamare l'esecuzione di un'azione per aggiornare lo stato.

Non potendo agire direttamente sullo store, Redux mette a disposizione due Hook.

Nel primo caso è necessario utilizzare l'Hook *useSelector* della libreria react-redux. Al metodo viene passata una funzione di selezione che permette di ritornare un valore specifico dello stato. Il listato mostra due modi per chiamare l'Hook: nel primo la funzione è esplicitata e ritorna il valore di info_status all'interno dello stato dei contents; il secondo utilizza la funzione *selectProjectById* per prelevare le informazioni relative ad uno specifico progetto dalla lista dei progetti salvata nello stato dei projects.

```
import {useSelector} from "react-redux";
const infoStatus = useSelector(state=> state.contents.info_status)
```

```
import {useSelector} from "react-redux";
import {selectProjectById} from "../../features/projects/
         ProjectsSlice";
const project = useSelector(state => selectProjectById(state, id))
-------------------------------------------------------------------
//definizione della funzione all'interno di ProjectsSlice,js
export const selectProjectById = (state, projectId) =>
    state.projects.projects.find(project =>project.IdProg ===
    projectId)
```

- Esempio di utilizzo di useSelector -

Nel secondo caso bisogna effettuare il dispatch dell'azione richiesta. L'Hook fornito da Redux per simulare il metodo dei reducer è useDispatch.

```
import {useDispatch} from "react-redux";
import {inviaInfo} from "../../features/content/ContentSlice";

function Info(){
  const dispatch =useDispatch()
  dispatch(inviaInfo(finf)) --> Thunk rispettivo descritto nello
                                slice dei contenuti
 /*…*/
}
```

- Esempio di utilizzo di useDispatch -

Tramite Redux è stato inoltre possibile sovrintendere la questione del token.

Essendo un'applicazione sviluppata temporaneamente in locale, la parte relativa alla sicurezza è solo abbozzata. Per simulare una corretta autenticazione dell'utente tramite token, è stato salvato l'username sia nel localStorage, sia nello store, come verrà affrontato nel paragrafo 3.3.6 relativo al routing e alla gestione di pagine pubbliche e pagine private.

Per concludere, è importante sottolineare la differenza tra stato globale e stato locale, in quanto un abuso del sistema potrebbe sovraccaricare lo storage. Per evitare problematiche simili è importante considerare se un determinato dato sia utile a una singola componente o a tutto l'albero. Nel primo caso infatti è corretto utilizzare l'Hool State.

3.3.6 React Router

Nel paragrafo 3.3.1 è stato esordito che l'applicazione è una SPA. Nonostante sia "Single Page" l'applicazione è costituita in realtà da un insieme di viste o "pagine" differenti generate lato client. I contenuti al loro interno sono realizzati sia dal client, sia dal risultato di richieste specifiche verso il server. Per poter navigare tra le varie viste è utilizzato un processo detto routing. A ogni "pagina" viene associato un URL. Il router è responsabile di leggere la URL e di visualizzare il contenuto corrispondente.

La libreria utilizzata per la gestione delle rotte è react-router-dom. A tal scopo il router viene posto alla radice della gerarchia di visualizzazione [54].

```
import { BrowserRouter } from "react-router-dom";

ReactDOM.render(
   <BrowserRouter>
       <App />
   </BrowserRouter>,
   document.getElementById("root")
);
```

- Disposizione del router nella funzione generatrice -

Il componente BrowserRouter indica che la navigazione dell'interfaccia è basata sull'URL completa (a differenza di HashRouter che implicherebbe l'uso di fragment).

Successivamente alla definizione del router è necessario specificare esplicitamente tutte le rotte (<Route />) possibili associando a un path il componente da visualizzare. L'operazione è possibile ponendo un costrutto <Switch/> all'interno di <App />: tale componente visualizza esclusivamente il primo elemento il cui inizio del path combacia con l'URL corrente; ciò significa che se due componenti hanno una rotta simile è necessario specificare l'attributo "exact" affianco all'URL per evitare incomprensioni. Il listato mostra differenti modi per definire le rotte all'interno dello *switch*.

```
import {Switch, Route} from 'react-router';

const MyComponent = (props) => {
   return (
       <Switch>
           <Route path="/about">
               <About />
           </Route>
           <Route path="/contact/:id" component={Contatct} />
           <Route path="/contact/"
                 render = {(props) => <AllContacts {…props} />} />
           <Route path="/"> <Home/> </Route>
       </Switch>
   );
}
```

- Definizione delle rotte tramite lo switch -

[54] https://reactrouter.com/web/guides/primary-components

Per renderizzare il componente della rotta è possibile impiegare uno dei seguenti metodi: component, children o render. Ognuno di essi passa al nuovo componente tre props: match, location e history [55]. Di particolare importanza per lo sviluppo dell'interfaccia sono stati:

- Match. Oggetto che contiene le informazioni su come il path di una certa <Route> combaci con l'URL del browser. Tra le proprietà peculiari della prop è rilevante params il quale permette di gestire le URL dinamiche tramite l'Hook useRouteMatch.

- History. Anch'esso è un oggetto e rappresenta una delle due dependencies più importanti di React Route; grazie ad esso è possibile manipolare con una serie di metodi l'URL, e di conseguenza il contenuto della pagina.

A partire dalla versione 5.x il pacchetto react-router-dom offre quattro Hook a supporto dell'instradamento [50]:

- usehistory(): Hook attraverso cui è possible accedere all'oggetto history;

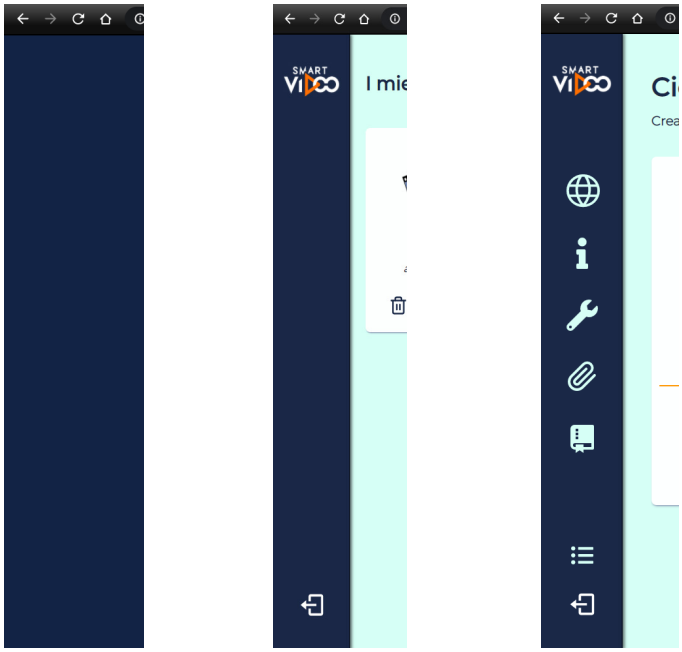
- useLocation(): restituisce l'oggetto location all'interno del quale è presente la URL corrente in forma destrutturata;

- useParams(): restituisce un oggetto contenente un set di coppie chiave/valore impiegate per la determinazione della corrispondenza tra path del componente e URL del browser;

- useRouteMatch(): restituisce un oggetto match risultato della corrispondenza tra l'espressione fornita come argomento dell'Hook e l'URL corrente.

Il controllo dell'URL è di rilevante importanza all'interno dell'applicazione perché condiziona la presenza di alcuni elementi quando viene mostrata la pagina. In dettaglio nella figura 3.9 sono mostrate le tre visualizzazioni possibili della navbar che funge da menù laterale all'interno dell'interfaccia.

[55] https://reactrouter.com/web/api



(a) (b) (c)

Figura 3.10 - Gestione della visualizzazione del menù laterale

Esaminando i vari casi: la figura 3.9 a mostra la navbar nascosta per le pagine relative alla fase di registrazione e login e per la visualizzazione dello SmartVideo finale; la figura 3.9 b mostra la barra laterale quando si accede alla propria area personale, come è anche intuibile dall'icona in basso relativa al logout dell'utente; infine, il terzo caso (figura 3.9 c) rappresenta il menù completo di tutte le sue voci, accessibile quando si vuole modificare il contenuto di un singolo progetto. Dal punto di vista del codice questa operazione è possibile grazie alla combinazione della prop *match* e dell'Hook *useRouteMatch*.

Le rotte corrispondenti ai casi sopraccitati sono rispettivamente:

```
//in App.js all'interno dello Switch che veicola le rotte possibili
 <Route exact path="/login" component={LoginForm} /> --> a
 <PrivateRoute exact path="/myProjects" render={() => ( --> b
    <React.Fragment>
       {mostra && <AddProject/>}
       <MyProjects />
    </React.Fragment>
 )}/>
<PrivateRoute path="/myProjects/:projectId"
                         component={Interfaccia}/> --> c
```

- Definizione delle route -

Il frammento di codice mostra come è gestita la visualizzazione delle voci nel menù completo. In particolar modo la loro comparsa è condizionata dal valore della variabile booleana "visualizza".

```
//in Navbar.js
  let visualizza
  const match0 = useRouteMatch("/myProjects")

  const match = useRouteMatch("/myProjects/:projectId")
  const id = match? match.params.projectId : undefined

  if(match0 && match){
    visualizza = true
  }else{
    visualizza = false
  }
```

- Definizione della condizione di visibilità -

Per consentire la navigazione verso un determinato path esistono diverse soluzioni. La prima consiste nell'utilizzare l'Hook *useHistory()*. La funzione infatti accetta alcuni metodi tra cui ad esempio *push* e *replace* che provocano l'instradamento in seguito all'attivazione di un evento, come il click del mouse su un bottone.

```
<button type='button' className='indietro'
   onClick={() => history.push(`/myProjects/${project.IdProg}`)}>
       Indietro
</button>
```

- Instradamento in seguito al click sul bottone -

Alternativamente la libreria mette a disposizione due componenti per veicolare la navigazione dell'applicazione: Link e Redirect.

<Link /> permette di navigare verso una URL come effetto di un click. Il path può essere passato sia come stringa, sia come oggetto tramite l'attributo to. Una sua variante è rappresentata da <NavLink> che permette di aggiungere dello stile quando il link diventa attivo.

```
import {Link} from "react-router-dom";

//componente Project.js
<Link to={`/myProjects/${project.IdProg}`}>Modifica progetto</Link>
```

- Instradamento con Link -

<Redirect /> costrutto che forza la navigazione verso la propria destinazione, definita tramite to analogamente al caso precedente. È utilizzato soprattutto in situazioni in cui l'utente cerca di visualizzare contenuti senza aver rispettato certe condizioni, come ad esempio tentare di accedere a dati personali senza aver effettuato il login: il componente forza l'utente alla pagina di autenticazione per proseguire.

Un altro aspetto importante di React Router infatti è la possibilità di configurare certe rotte come "private". In una web app qualsiasi rotta è visualizzabile modificando l'URL dalla barra degli indirizzi del browser. Se non si crea un qualche tipo di "blocco" chiunque può accedere indiscriminatamente a qualsiasi contenuto. Per evitare che si verifichino certe situazioni è necessario effettuare un controllo sull'utente. Se è autenticato allora la visualizzazione procede senza problemi, altrimenti il router scatena il reindirizzamento alla pagina di login. Il controllo può avvenire in maniera sincrona, attraverso l'uso dei token e del localStorage, o in modo asincrono, effettuando una richiesta al backend. Ciò consente dal punto di vista opposto di ottenere direttamente la propria pagina personale

sia quando si accede all'applicazione ed è stato mantenuto il login attivo, sia al refresh senza dover ripassare per la fase di autenticazione.

3.3.7 Librerie di supporto

Ai fini dello sviluppo dell'applicazione sono state impiegate una serie di librerie sia proprie di React,così come esterne. Di seguito vengono citate le principali e il loro utilizzo.

Formik, Yup e Reactstrap

L'interfaccia è strutturata in modo tale che le informazioni e le risorse, che l'utente desidera all'interno del suo SmartVideo, siano raccolti tramite form. Ogni sezione relativa a un'area di modifica di un singolo progetto si presenta quindi come una pagina al cui interno sono presenti input di testo, radio button, input di tipo file, ecc…

Per crearli è utilizzato Formik, una libreria dichiarativa, adattabile e specializzata nella gestione dei form in React. Grazie ad essa è possibile risolvere facilmente problematiche del tipo:

- Manipolazione dello stato
- Validazione del form e visualizzazione dei messaggi d'errore
- Invio dei dati

Alla base del funzionamento della libreria è presente l'oggetto *formik* che contiene, e attraverso cui è possibile reperire, tutte le informazioni relative a un particolare form tra cui ad esempio i valori degli input, se il form è stato modificato e/o inviato o se i valori inseriti sono validi.

I tre elementi principali di cui si compone Formik sono: l'oggetto initialState, all'interno del quale sono definiti i valori iniziali degli input; la funzione onSubmit che regola l'invio dei dati e la visualizzazione del form tramite particolari props chiamate onSubmitProps; l'oggetto validationSchema all'interno del quale sono definiti le condizioni di validità dei vari input. In particolare, per la convalida è stata utilizzata la libreria Yup attraverso cui è possibile costruire schemi estremamente espressivi che consentono sia di verificare la forma di un valore esistente, sia di trasformare valori [56].

```
const initialValues = {
   email: '',
   username: '',
   password: '',
   conferma: '',
}
const validationSchema = Yup.object({
  email: Yup.string().email("Il formato dell'indirizzo email non è
                        incorretto").required('Obbligatorio'),
   username: Yup.string().required('Obbligatorio'),
   password: Yup.string().required('Obbligatorio'),
   conferma: Yup.string().oneOf([Yup.ref('password'),''],
      'Le password non corrispondono').required('Obbligatorio')
});
const onSubmit = (values,onSubmitProps) => {
   console.log('Form data', values);
   const form = document.getElementById('f_signup')
   const formData = new FormData(form);
   onSubmitProps.setSubmitting(false);
   // resetForm();
   dispatch(selezEmail(values.email))
   onSubmitProps.setStatus('sent');
   dispatch(signUP(formData));
}
```

- Impostazione della libreria Formik -

Questi tre dati vengono poi passati come attributi al componente <Formik/>.

La libreria fornisce anche altri componenti atti a semplificare la gestione dei form: Form, Field, che ricopre il ruolo del tag <input> HTML, e ErrorMessage. Come mostrato dal listato, i componenti sono renderizzati come figli del componente Formik; la funzione riceve delle props (formik) e restituisce il corpo del form in JSX.

Un altro aspetto rilevante di Formik è la riusabilità. Per evitare di riscrivere ogni elemento da zero per ogni form dell'interfaccia, è stato creato il componente FormikControl capace di mostrare un determinato input in

[56] https://www.npmjs.com/package/yup

base alle sue props. In tal modo si è definito una sola volta tutti i tipi di input necessari; all'interno della componente FormikControl.js è stato posto uno switch per scegliere quale elemento renderizzare. Di esempio il form relativo alla registrazione dell'utente: la prop control determina il tipo di elemento desiderato.

```
//componente RegistrationForm.js
return (
    <Formik initialValues={initialValues}
            validationSchema={validationSchema}
            onSubmit={onSubmit}>
       {
           formik => {
              return <Form id='f_signup' className="form">
               <FormikControl
                 control='input'
                 type='email'
                 label='Indirizzo email'
                 name='email'
               />
               <FormikControl
                 control='input'
                 type='text'
                 label='Username'
                 name='username'
               />
               <FormikControl
                 control='input'
                 type={type}
                 label='Password'
                 name='password'
               />
           </Form>
           }
       }
   </Formik>
);
```

- Gestione del form per il login -

I singoli elementi sono costruiti a partire dallo stesso schema: label, input e error. In particolare, l'esempio mostra come un input di tipo testo è definito.

```
function Input(props){
   const {label, name, placeholder,value, ...rest} = props;
   return (
       <div style={{position:'relative'}}>
           {label && <label htmlFor={name}>{label}</label>}
            <Field id={name} name={name} placeholder={placeholder}
                  {...rest} />
            <ErrorMessage name={name} type='input' style={{}}
                  component={TextError}/>
        </div>
    )
}
```

- Implementazione degli input partendo dagli elementi base di Formik -

Uno dei limiti di Formik è rappresentato dalla gestione degli input di tipo file. Questi ultimi sono stati gestiti attraverso Reactstrap, ovvero la libreria di Bootstrap 4 per React. In questo caso l'aggiornamento del valore dell'input e gli errori sono stati gestiti manualmente tramite gli attributi della prop formik.

Material-UI

Material-UI è un framework di React open source, basato sulla libreria di Google Material Design [57]. I suoi componenti sono indipendenti e non si affidano a fogli di stile globale: ognuno di essi può essere opportunamente stilizzato attraverso i metodi withStyles e makeStyles. Il frammento di codice mostra la personalizzazione del componente Switch.

[57] https://dev.to/amberjones/5-delightful-things-about-material-ui-5402

```
const OrangeSwitch = withStyles({
   switchBase: {
       color: orange[300],
       '&$checked': {
           color: orange[500],
       },
       '&$checked + $track': {
           backgroundColor: orange[500],
       },
   },
   checked: {},
   track: {},
})(Switch);
```

- Personalizzazione dello Switch tramite Material UI -

I vari componenti forniti dal framework possono essere suddivisi in nove categorie: Layout, Input, Navigation, Surfaces, Feedback, Data Display,Data Grid, Utils e Lab. Tra questi il supporto di Material è stato rilevante per la realizzazione di componenti come radio button, switch o slider, non ancora implementati o difficilmente personalizzabili con Formik. Spesso si è optato per l'utilizzo di switch a fronte di situazioni in cui sarebbero richiesti i checkbox sia per lo stile, sia per la facilità di elaborazione dei suoi valori.

Material è stato impiegato inoltre per la visualizzazione di feedback positivi visivi, tramite l'utilizzo di Snackbar e Alert.

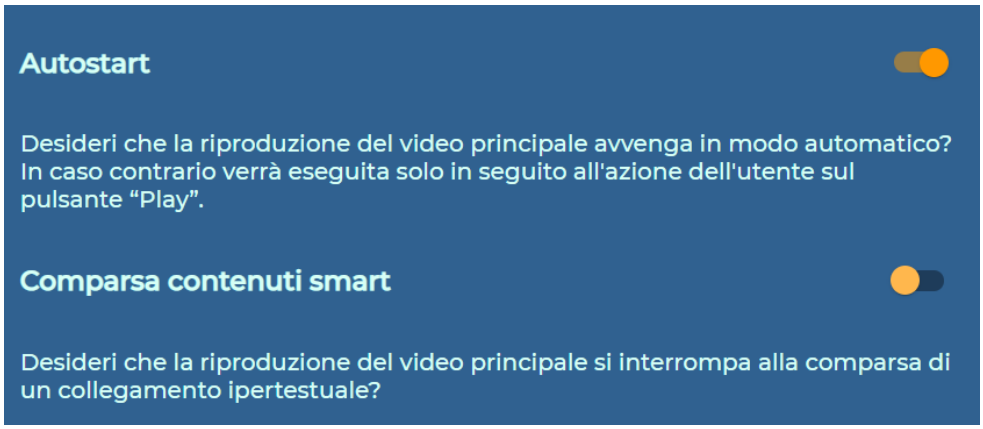


Figura 3.11 - Material UI Switch

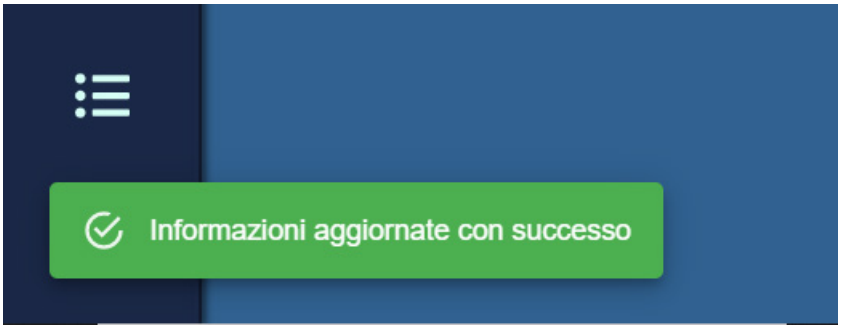


Figura 3.12 - Material UI feedback con Snackbar

I casi in cui si manifesta sono generalmente due: il feedback si attiva ogni volta che l'utente accede a una pagina dell'interfaccia e la lettura dei dati dal database va a buon fine oppure nei casi in cui si desidera salvare delle modifiche e l'operazione termina col successo. Per i feedback di errore si è scelto l'utilizzo standard di alert.

3.3.8 Descrizione delle pagine dell'applicazione

Il percorso dell'utente all'interno dell'applicazione è suddiviso in tre step principali: autenticazione e registrazione, creazione e modifica di un progetto e visualizzazione dello SmartVideo finale. Di seguito vengono descritti dal punto di vista grafico e funzionale.

Registrazione e autenticazione dell'utente

All'apertura dell'applicazione all'utente è presentata la schermata di login, attraverso cui è possibile accedere alla propria area personale o eventualmente alla sezione di registrazione per crearsi un account, tramite l'apposito link.

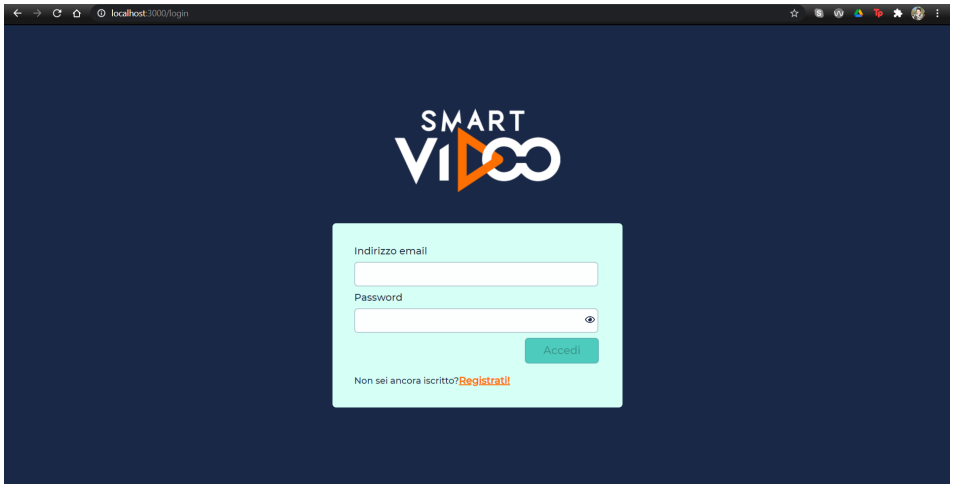


Figura 3.13 - Schermata di login

Come accennato nel paragrafo 3.3.5, i form sono gestiti tramite la libreria Formik. Quando un valore inserito non rispetta le regole di validità viene mostrato un avviso in corrispondenza dell'input errato. I controlli generali riguardano il formato dell'input e l'obbligatorietà.

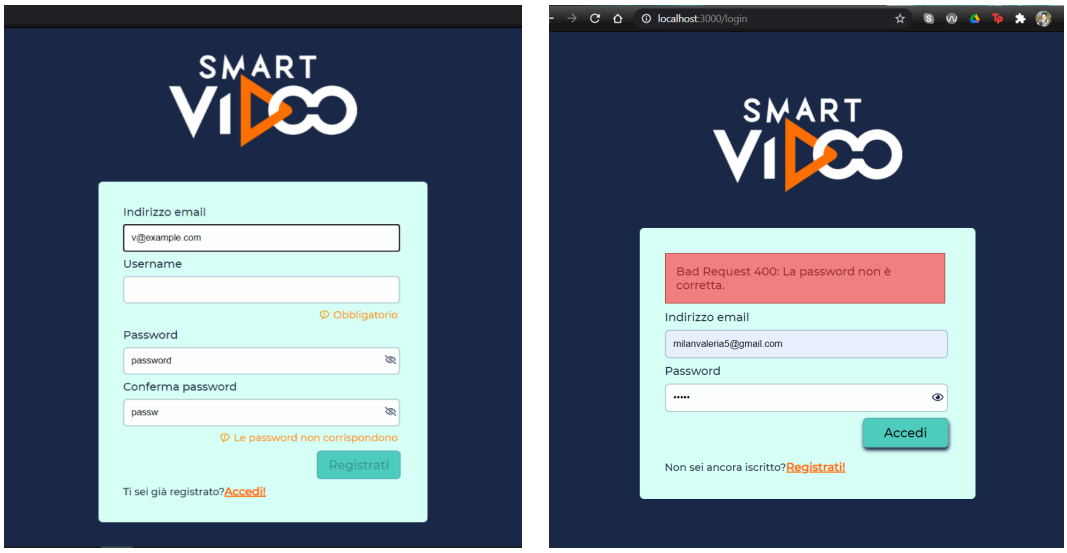


Figura 3.14 - Feedback tramite Formik e server

L'inserimento della mail e della password presentano controlli aggiuntivi in quanto nel primo caso la stringa deve rispettare il formato standard degli indirizzi email "example@something.com", mentre nel secondo il controllo è effettuato sulla conferma della password che deve combaciare con il valore precedentemente inserito (figura 3.14).

Una volta inseriti tutti i termini richiesti, il bottone relativo all'invio dei dati viene abilitato per proseguire con le operazioni al backend. I controlli sui valori infatti possono avvenire anche in seguito all'invio: in caso di risposta positiva l'utente è reindirizzato alla sua pagina personale, mentre in caso contrario i Thunk di Redux restituiscono il messaggio d'errore relativo sotto forma di avviso nella parte superiore del form, come mostrato nella seconda immagine della figura 3.14.

Quando l'utente si registra e l'account viene creato correttamente, viene mostrata la schermata di verifica dell'account in cui è necessario inserire il codice inviato per email (paragrafo 3.2.2).

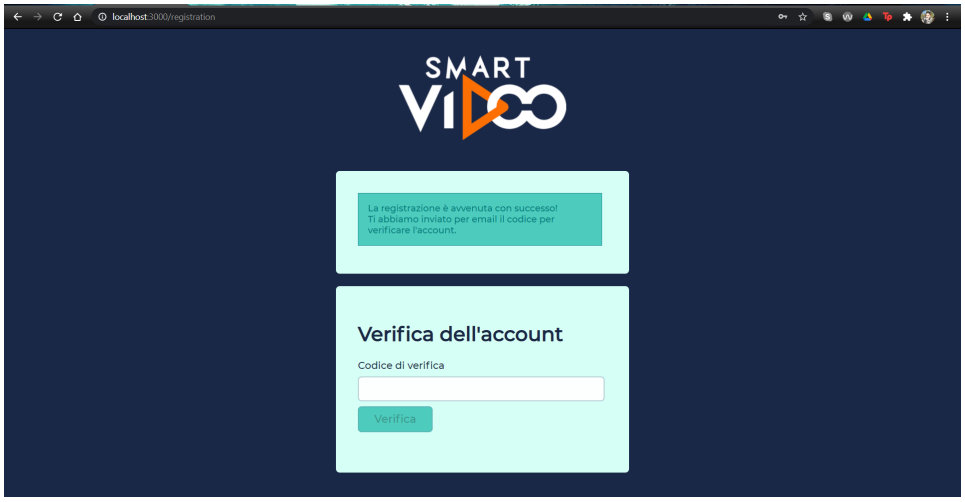


Figura 3.15 - Verifica dell'account tramite codice OTP

Nel caso in cui anche la verifica avvenga con successo, l'utente è rimandato alla schermata di login, altrimenti viene mostrato un messaggio d'errore come nell'esempio precedente. All'invio delle credenziali viene aggiornato il token salvato nel localStorage ed è reso disponibile l'accesso all'area personale.

Creazione e modifica di un progetto

La prima schermata raggiungibile è quella relativa ai propri progetti. Al render del componente viene avviata una richiesta al database grazie a cui viene restituito al client l'elenco dei progetti creati. La lista viene memorizzata nello store e successivamente presentata visivamente.

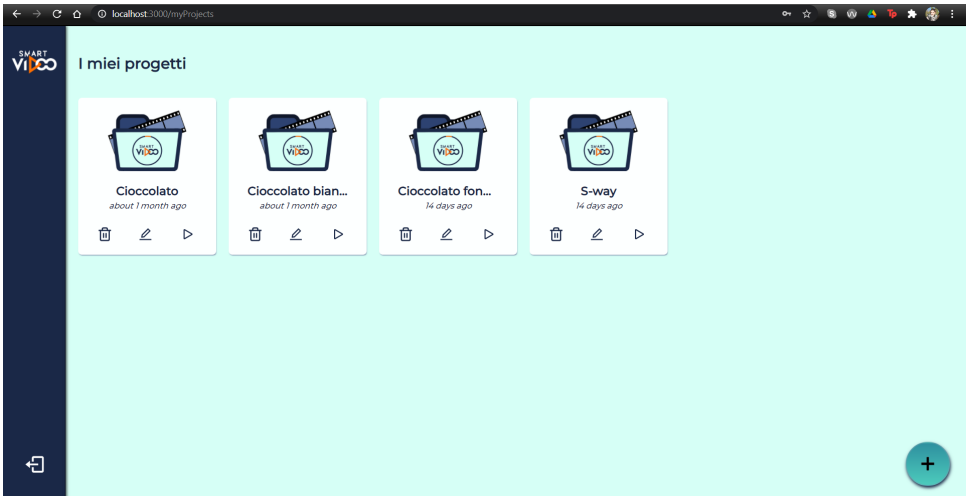


Figura 3.16 - Schermata "I miei progetti"

Ogni progetto è presentato sotto forma di cartolina composta dal titolo, dal lasso di tempo trascorso dalla sua creazione e da tre icone volte rispettivamente a eliminare il progetto, modificarlo e visualizzare lo SmartVideo corrispondente.

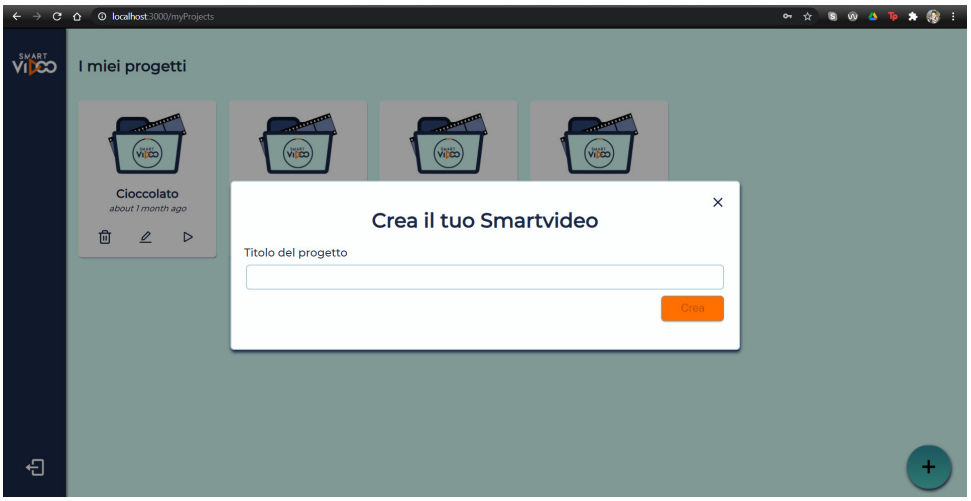


Figura 3.17 - Creazione di un nuovo progetto

All'interno della pagina è altresì possibile aggiungere un nuovo progetto tramite il bottone in basso a destra. La schermata che si manifesta in questo caso è costituita da una finestra modale che compare in overlay e in cui è possibile inserire il titolo del nuovo SmartVideo.

Prima che l'operazione venga terminata è effettuato un controllo sul nome introdotto in modo tale che un utente non inserisca due progetti

identificati dallo stesso titolo. È inoltre presente la navbar laterale, elemento fondamentale per aiutare l'utente a orientarsi nell'applicazione. Nel caso specifico l'unica azione possibile all'interno del menù è effettuare il logout e ritornare alla finestra del login.

Accedendo a un progetto, la schermata che compare è costituita da un menù con accesso rapido alle sezioni editabili. L'URL relativo alla pagina web è costruito a partire dalla chiave del progetto selezionato: quindi se si vuole modificare il progetto P20 l'URL della barra degli indirizzi risulterà "…/myProjects/P20".

Le voci dei quadranti principali corrispondano a quelle insite nella navbar in modo da permette una navigazione non lineare e personalizzata da parte dell'utente all'interno delle viste. Le icone utilizzate sono state scelte attingendo al catalogo di React, React Icons, all'interno del quale sono raccolte diverse librerie popolari di icone come FontAwesome o Material Design per garantire un aspetto attuale e all'avanguardia.

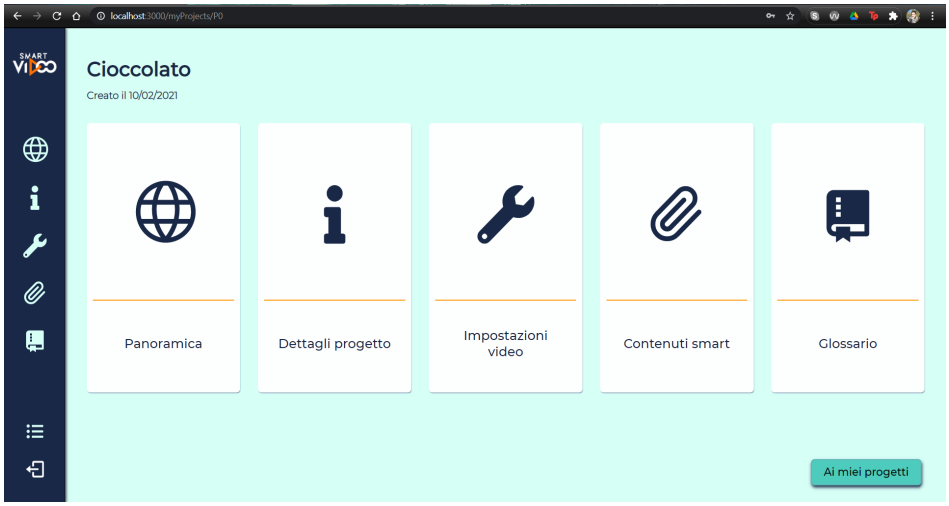


Figura 3.18 - Schermata pagina principale del progetto

La figura 3.18 mostra la presenza di un'ulteriore icona attraverso cui è possibile navigare indietro alla schermata relativa ai propri progetti.

Di seguito verranno presentate nel dettaglio le singole sezioni. Per facilitare l'orientamento all'interno dell'interfaccia è stato posto un controllo sull'URL nell'implementazione del componente Navbar.js che permette di selezionare visivamente l'icona corrispondente alla sezione visualizzata.

"Panoramica". La prima sezione presenta un riassunto dei contenuti inseriti attraverso cui l'utente può controllare l'effettivo salvataggio delle modifiche apportate. Ogni qualvolta che si accede alla schermata vengono effettuati dei controlli sulle informazioni presenti sulla base dati; i risultati vengono elaborati e successivamente tradotti nello schema presentato nella figura 3.19.

Figura 3.19 - Schermata "Panoramica"

I dati relativi a link, quiz e voci del glossario sono stati riassunti mostrandone solo la quantità effettiva per ottenere una visualizzazione ordinata e pulita della schermata, evitando di provocare confusione nell'utente a causa delle troppe informazioni. La sezione relativa alla panoramica dei contenuti è l'unica non direttamente editabile dall'utente, ma al tempo stesso influenzata dai suoi comportamenti all'interno dell'applicazione.

"Dettagli del progetto". La schermata presenta un form all'interno del quale è possibile aggiungere informazioni relative ai metodi di realizzazione del progetto o citare eventuali contributori. È inoltre possibile inserire il logo dell'azienda o dell'istituto per cui si sta realizzando lo SmartVideo qualora si fosse interessati a mostrarlo come copertina del video finale. Tutte le informazioni relative a questa pagina sono facoltative; l'unico dato obbligatorio è il nome del progetto: se venisse modificato verrebbe

effettuato il dispatch dell'azione incaricata di modificarlo in tutte le pagine in cui compare.

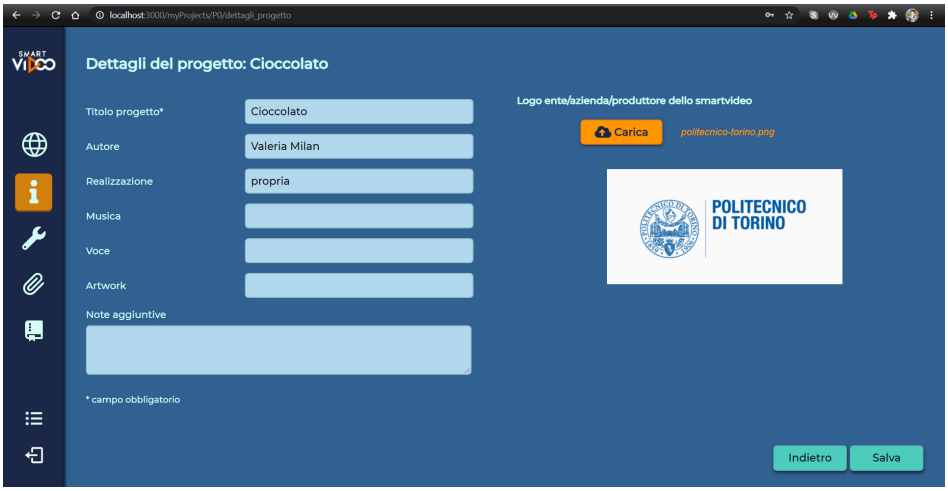


Figura 3.20 - Schermata "Dettagli del progetto"

Come accennato nel paragrafo 3.3.6, gli input di tipo file sono stati gestiti con una libreria differente da Formik perché non totalmente supportati da essa. Durante lo sviluppo sono sorte diverse problematiche a riguardo. Primo fra tutti l'azzeramento del contenuto dell'input ogni volta che la pagina viene visualizzata. Di conseguenza non è possibile assegnare un valore predefinito come ad esempio in seguito alla lettura del database. Inoltre, qualsiasi modifica effettuata a uno degli altri input testuali provocava una reazione d'errore della validazione del file ("formato de file non supportato") nonostante l'input specifico non venisse alterato o esistesse già una risorsa corrispondente nella base dati. Per queste ragioni il form testuale e l'input di tipo file sono gestiti indipendentemente sia a livello spaziale nella pagina, ovvero in due componenti distinte, sia a livello di chiamate al database: le voci "Salva" e "Carica" sono rispettivamente relative al questionario e alla risorsa. Se il caricamento dell'immagine avviene con successo, una copia viene salvata all'interno del server cloud (paragrafo 3.3…). In questo modo l'anteprima risulta sempre visibile nella pagina perchè costruita facendo riferimento all'indirizzo sorgente della copia presente nel server.

"Impostazioni video". La terza voce del menù riguarda una serie di impostazioni relative al video master dello SmartVideo. L'architettura dei componenti è analoga al caso precedente, soprattutto per quanto riguarda il caricamento del file del video, gestito separatamente dalle altre opzioni. Anche in questo ne è visualizzata un'anteprima per fare in modo che l'utente abbia un feedback sia sulla corretta selezione della risorsa, sia sull'avvenuto salvataggio.

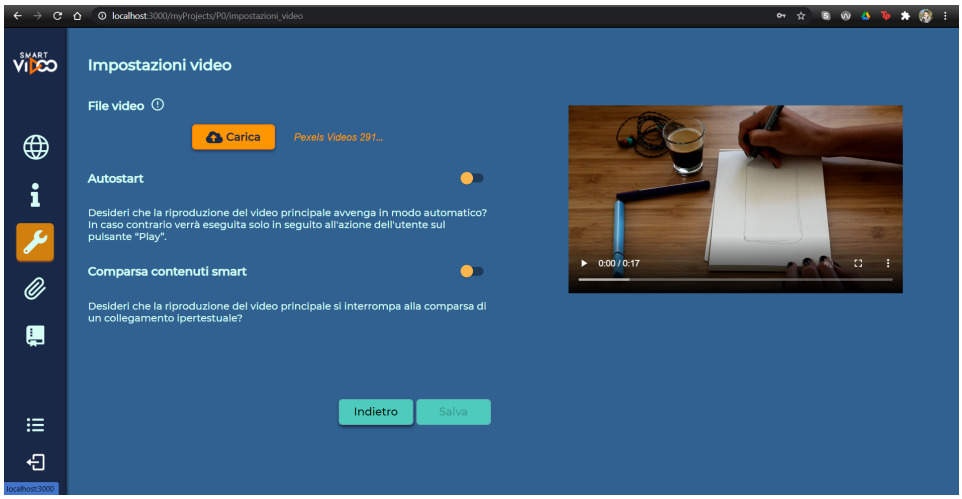


Figura 3.21 - Schermata "Impostazioni video"

In particolare, le opzioni nella prima metà della pagina servono a influenzare la riproduzione del video nei momenti di avvio dello SmartVideo e di comparsa dei contenuti satelliti. La scelta di utilizzare degli switch è stata guidata dal fatto che restituiscono un valore booleano a differenza delle checkbox o dei radio button che effettuano controlli solo su stringhe. In questo modo è risultato più semplice e immediato fare i confronti e gli aggiornamenti ai valori presenti nel database.

"Contenuti smart". La schermata funge da contenitore per i contenuti smart, i quali possono essere distinti in link, o semplicemente approfondimenti, e quiz. In alto è presente un filtro per facilitare la visualizzazione dei contenuti di interesse.

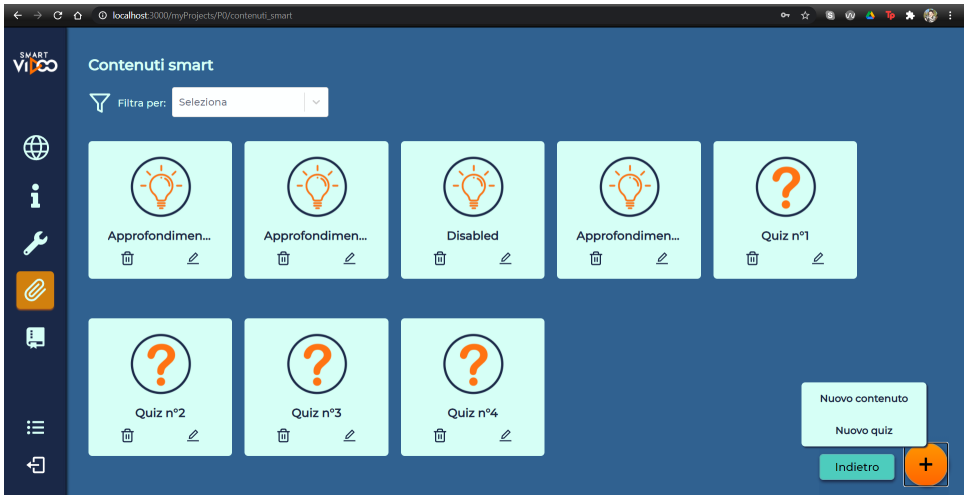


Figura 3.22 - Schermata "Contenuti smart"

I contenuti satelliti di tipo link sono identificati dal nome dell'approfondimento e da due icone con funzione simile a quella presentata parlando dei progetti, ovvero l'eliminazione e la modifica. Quando si tenta di eliminare un elemento (progetto, link, quiz, voce del glossario) viene visualizzato un messaggio di conferma per evitare che l'utente cancelli inavvertitamente informazioni importanti. La figura 3.23 mostra l'avviso in overlay, all'interno di una finestra modale come nel caso di aggiunta di un nuovo progetto. Il messaggio presentato è identificativo per l'elemento che si desidera eliminare, ovvero in questo caso il "contenuto Approfondimento 1".

Figura 3.23 - Eliminazione di un contenuto

Accedendo alla schermata di modifica, la pagina mostrata è suddivisa in due sottosezioni (figura 3.24). La metà superiore della videata è relativa alle caratteristiche testuali del contenuto satellite e alla sua posizione all'interno dello SmartVideo. In particolare, uno SmartLink esiste se è presente almeno una descrizione teorica e testuale dell'approfondimento: il titolo e il contenuto per questo motivo sono segnati come obbligatori, a differenza delle risorse che possono essere opzionali.

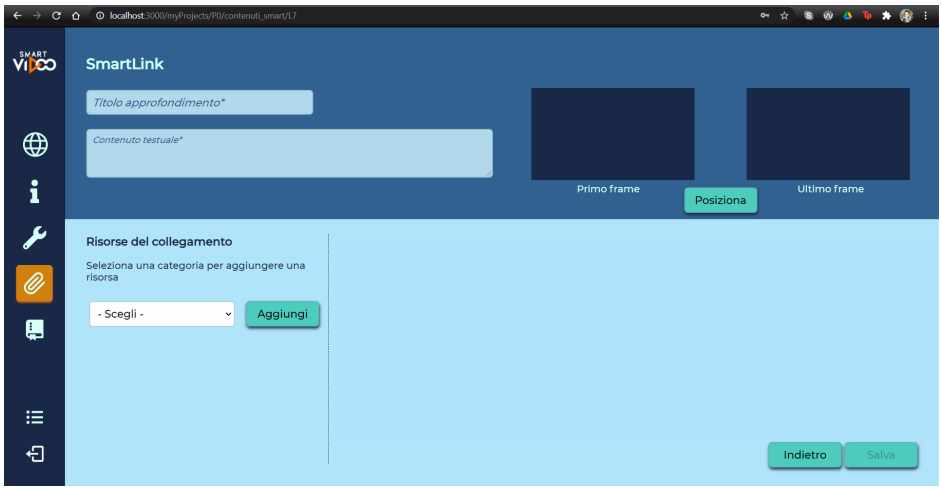


Figura 3.24 - Schermata "SmartLink"

In dettaglio nella figura 3.25 è mostrato il posizionamento del link all'interno del video.

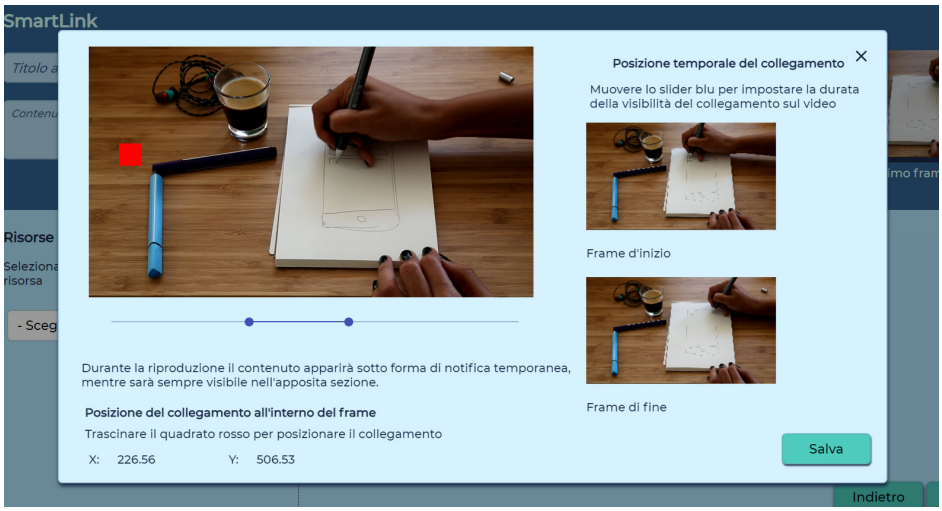


Figura 3.25 - Posizionamento del link

Per poter collocare un contenuto all'interno del video sono effettuate essenzialmente due operazioni.

La prima consiste nell'indicare il frammento di video in cui si desidera far apparire l'hotspot relativo al link. A tal scopo è possibile selezionare i frame tramite lo spostamento delle estremità dello slider posto al di sotto dell'anteprima: al trascinamento dei "pallini" la seekbar del video aggiorna l'immagine visualizzata nella preview e disegna il frame corrispondente nella colonna a destra della finestra. Inoltre, viene creata una copia dell'immagine disegnata anche nella schermata in background in modo tale che l'utente abbia un costante riferimento visivo della collocazione del link anche mentre ne sta modificando il contenuto.

La seconda azione possibile consiste nel posizionare l'hotspot, rappresentato dal quadretto rosso nella figura 3.25, all'interno dell'area disponibile, tramite un'operazione di drag & drop. Le coordinate del riferimento vengono visualizzate e aggiornate automaticamente alle interazioni dell'utente nella parte inferiore della finestra modale. Premendo "Salva" i dati vengono inviati al database, indipendentemente dalle azioni eseguite nella schermata in background.

La metà inferiore della finestra di modifica del contenuto smart ospita i form relativi al caricamento delle risorse. In particolare, il select permette di scegliere il tipo di file da inserire (immagine, video, audio, documento, URL). Cliccando su "Aggiungi" viene mostrato un form creato ad hoc per la risorsa, all'interno del quale inserire il file, il titolo e opzionalmente una didascalia. L'opzione "Cambia" serve per poter accedere ad un'altra categoria di risorsa. Una volta caricata, la risorsa appare nello spazio a lato identificata da un'icona, che ne rappresenta la tipologia, e dai dati inseriti, con riferimento al nome originale del file.

Figura 3.26 - Visualizzazione delle risorse caricate

Sempre a causa dei problemi relativi alla gestione del file, invece di inserire un'opzione di modifica della risorsa, si è optato per la semplice eliminazione. In caso ci siano errori nel materiale caricato l'utente deve quindi rieseguire i passaggi da capo dalla selezione del file all'interno del suo pc.

Per quanto riguarda gli SmartQuiz l'organizzazione della schermata è simile al caso precedente: è presente un select che permette di scegliere il tipo del test da aggiungere (vero o falso, risposta multipla) e il pulsante relativo al posizionamento del contenuto nel video. A differenza dei link, i quiz necessitano un solo frame in quanto appaiono in un determinato istante: la loro comparsa interrompono forzatamente la riproduzione del video, obbligando l'utente a rispondere per poter proseguire con la visualizzazione.

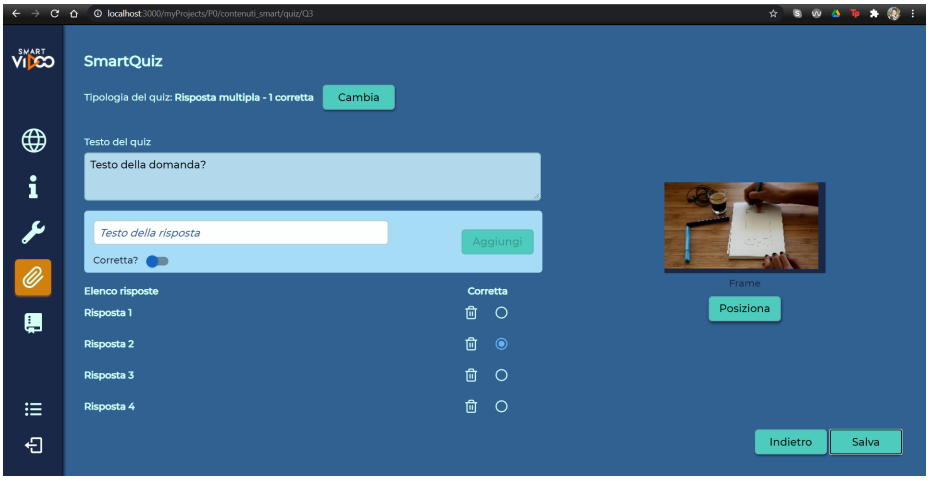


Figura 3.27 - Schermata "SmartQuiz"

Nel caso si desideri inserire un test a risposta multipla è presente un form aggiuntivo rispetto ai vero o falso in cui impostare sia il testo della risposta, sia l'opzione "corretta" tramite uno switch. Quest'ultima opzione è altresì modificabile tramite i radio button a lato dell'elenco riepilogativo delle risposte inserite.

"Glossario". Compilare la sezione riguardante il glossario è opzionale come quella riguardante i dettagli del progetto. Viene utilizzata nei casi in cui siano presenti termini tecnici, complessi, acronimi, non di uso quotidiano o eventualmente stranieri.

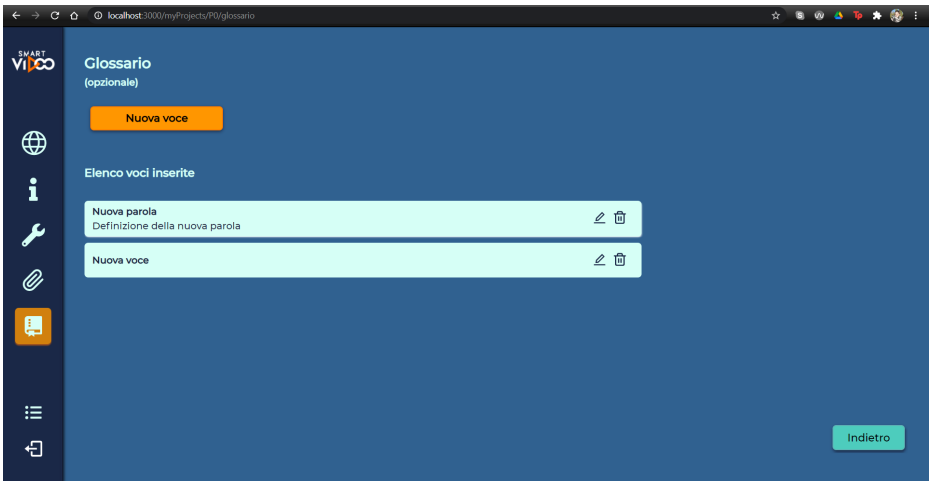


Figura 3.28 - Schermata "Glossario"

Quando una nuova parola viene inserita, viene aggiunta all'interno di un elenco in ordine alfabetico. Cliccando sulla voce è possibile visualizzare la sua definizione normalmente nascosta. Il motivo alla base di questa scelta è simile a quello dell'organizzazione della sezione "Panoramica": rendere la schermata pulita, mostrando solo le informazioni essenziali per l'utente. Per ogni glossa sono inoltre presenti due opzioni: l'eliminazione e la modifica del suo contenuto. Nel secondo caso il form utilizzato per inserire la voce viene mostrato compilato con le informazioni relative aggiornate per permette all'utente una modifica mirata dei contenuti.

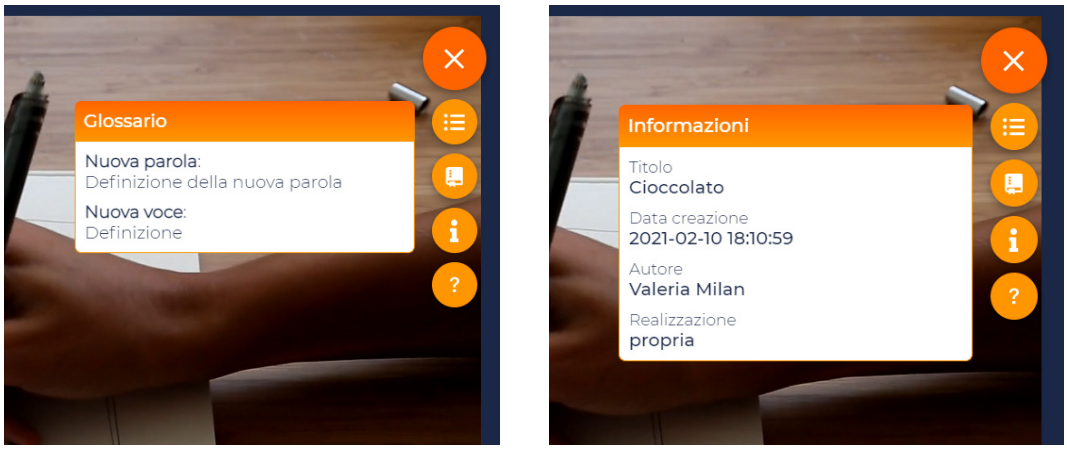
Visualizzazione SmartVideo

Premendo sull'icona play nella schermata dei progetti (figura 3.29) si può accedere allo SmartVideo assemblato con i file e le informazioni inserite tramite l'interfaccia. La pagina che viene mostrata è costituita da tre elementi: il video principale al centro, una sezione dedicata all'accesso rapido ai contenuti satelliti nella parte inferiore e un menù animato in alto a destra, in sovraimpressione sul video.

Figura 3.29 - Schermata "SmartVideo"

Le voci presenti nel menù rispecchiano la struttura dell'interfaccia, in particolare sono stati inseriti: l'elenco dei contenuti di tipo link, il glossario, se è presente almeno una voce al suo interno, le informazioni tecniche relative al progetto e una voce aggiuntiva di aiuto per guidare l'utente all'utilizzo dello strumento. Il menù è stato creato tramite i floating menù di Material-UI a cui sono stati aggiunti i tooltip per dare un riferimento testuale delle varie opzioni.

Come affermato poc'anzi le informazioni reperibili esclusivamente tramite il menù sono le voci presenti nel glossario e i dati descrittivi del progetto come mostrato nelle figure 3.30 a e 3.30 b.



(a) (b)

Figura 3.30 - Visualizzazione del glossario e delle informazioni

Cliccando su una voce del menù si apre una finestra, anch'essa in sovrimpressione, caratterizzata dal titolo della sezione corrispondente, evidenziato da uno sfondo arancione, e dal contenuto relativo vero e proprio, mostrato su sfondo bianco. A differenza del glossario, la cui visualizzazione è dettata solo dalla presenza o meno di voci inserite al suo interno, la sezione inerente alle informazioni sul progetto sarà sempre visibile in quanto indicati il titolo e la data di creazione, ovvero dati sempre definiti nella base dati.

Diversamente dalle voci sopra citate gli SmartLink non sono sempre accessibili. Il loro inserimento all'interno del menù è condizionato dalla riproduzione del video principale. Così come la comparsa dell'hotspot è vincolata a solo certi frame, il link corrispondente diventa attivo, e di conseguenza esplorabile nei due menù, solo in seguito alla sua prima apparizione nel corso della lezione. Se i contenuti satellite fossero sempre presenti verrebbe meno l'obiettivo principale dello SmartVideo, in quanto l'utente potrebbe leggere subito il contenuto dei link e rispondere ai quiz senza effettivamente aver seguito il video principale.

La visualizzazione di un contenuto satellite, invece, può avvenire in più modi: attraverso l'apposita voce nel floating menù, tramite l'accesso rapido nella parte inferiore del video o cliccando direttamente sugli hostpost quando appaiono sul video. Le figure 3.31 e 3.32 mostrano la vista dello SmartVideo prima e dopo l'interazione dell'utente con un contenuto satellite.

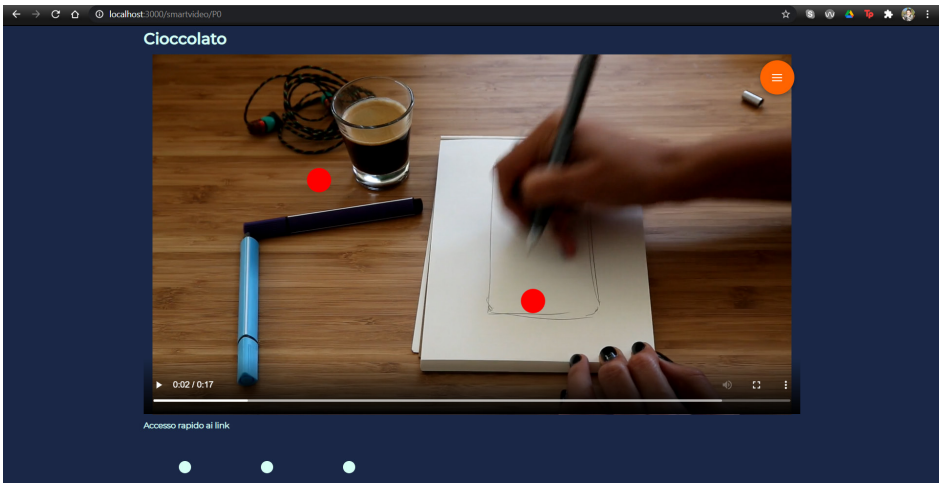


Figura 3.31 - Vista dei contenuti satellite prima dell'interazione dell'utente

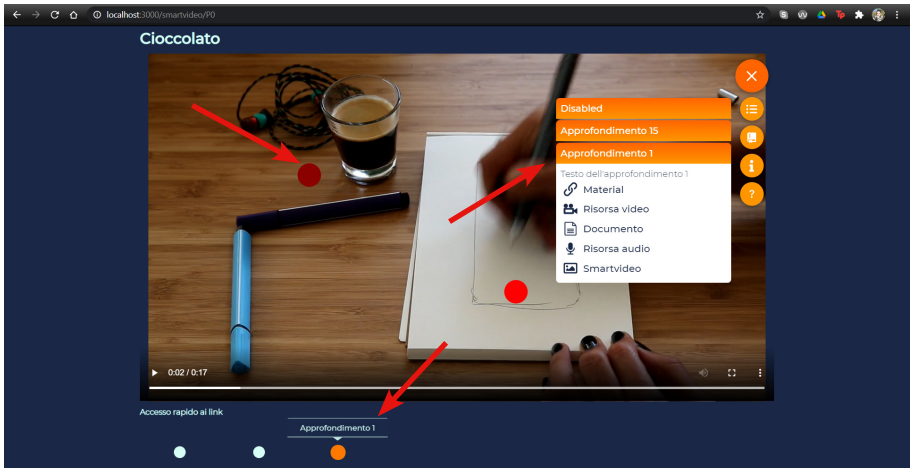


Figura 3.32 - Vista dei contenuti satellite in seguito all'interazione dell'utente

Quando si accede a un approfondimento tramite hotspot o voce nel menù di l'accesso rapido, graficamente accade che: il link selezionato viene aperto nel menù laterale; l'hotspot viene evidenziato assegnandogli uno sfondo più scuro; il collegamento al link nella barra inferiore si colora di arancione e cambia dimensione; la seekbar del video si aggiorna, posizionandosi nel punto in cui l'hotspot è comparso (ovvero il frame di inizio impostato nella schermata relativo alla collocazione del link all'interno del video). Procedendo in questo ordine le quattro azioni sono collegate tra di loro. L'unico modo per navigare i contenuti satellite indipendentemente e senza influenzare la riproduzione del video è attraverso il menù laterale.

Gli approfondimenti smart si presentano sotto forma voci all'interno di un elenco ordinate per apparizione cronologica. Così come per il glossario, il contenuto normalmente nascosto ed è visibile solo in seguito al click dell'utente sul titolo del link. In particolare, nella figura 3.32 sono mostrate le informazioni relative all'*Approfondimento 1*: ad una parte introduttiva descrittiva seguono le risorse corrispondenti. Così come nella figura 3.26 un'icona ne caratterizza la tipologia, seguita dal titolo della risorsa. In seguito al click sul nome si apre una finestra modale in cui viene mostrata a tutta pagina il file corrispondente, provvisto di nome ed eventualmente della didascalia corrispondente (figura 3.33).

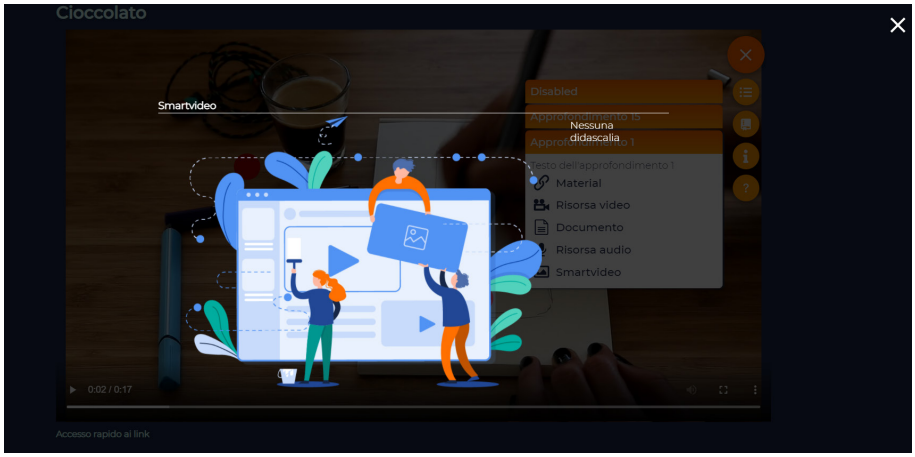


Figura 3.33 - Visualizzazione delle risorse

L'unica eccezione è rappresentata dalle risorse di tipo link le quali, all'azione dell'utente, scatenano l'apertura della pagina corrispondente in una finestra esterna nel browser.

Infine, l'ultimo elemento distintivo dello SmartVideo sono i Quiz. Come accennato durante la spiegazione della creazione di un quiz, le domande compaiono in overlay in un determinato momento del video. Il contenuto è visualizzato all'interno di una finestra modale in cui l'utente deve selezionare una delle risposte possibile e cliccare su "Conferma" per poter proseguire con la riproduzione del video.

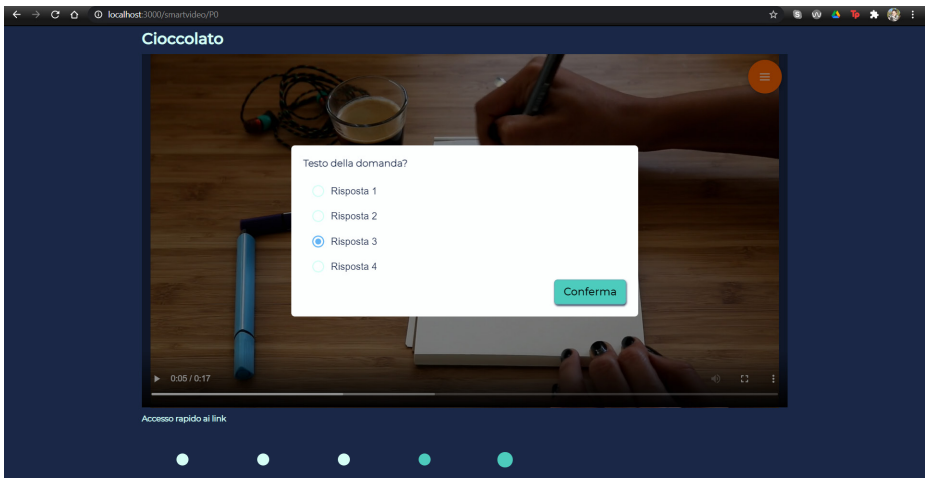


Figura 3.34 - Visualizzazione dei quiz

Al termine dello SmartVideo è possibile accedere ai risultati dei test cliccando sull'opzione "Visualizza i risultati". La finestra mostrata è quella visualizzata nella figura 3.35 in cui è presente il punteggio dell'utente in percentuale, il numero di risposte corrette, il numero di risposte sbagliate. In caso si siano fatti degli errori è mostrata la domanda incriminata con la risposta data e l'opzione corretta.

Figura 3.35 - Visualizzazione dei risultati

Capitolo 4

SmartVideo: la voce del target

Ai fini della tesi è stata condotta un'indagine che ha riguardato le conoscenze e le opinioni del prodotto da parte del target di riferimento. I dati sono stati raccolti grazie a una survey, realizzata con i Moduli di Google, che è stata condivisa insieme a un link alla demo del progetto su vari social, tra cui Linkedin per poter ottenere più facilmente riscontri dal personale aziendale. Il pubblico raggiunto si compone di 86 individui tra formatori, professori, dipendenti fruitori del prodotto e studenti.

4.1 Dati anagrafici del target

Dall'analisi generale sul target risulta che circa il 90% dei soggetti raggiunti sono possibili fruitori dello SmartVideo, di cui il 60% rappresentato da individui giovani con meno di 30 anni, ovvero studenti o ragazzi alle prese con le prime esperienze lavorative. Di conseguenza l'analisi corposa dei dati riguarderà soprattutto il prodotto dello SmartVideo e meno l'applicazione sviluppata per crearlo.

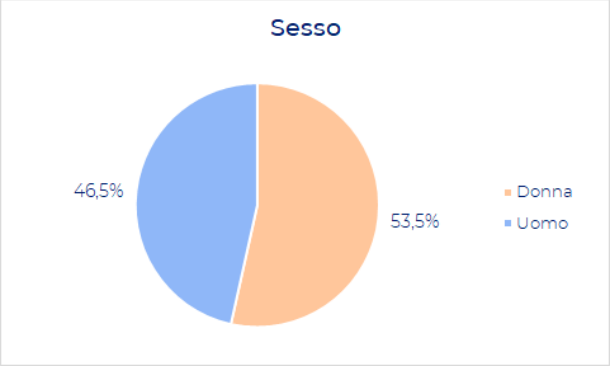


Figura 4.1 - Sesso del target

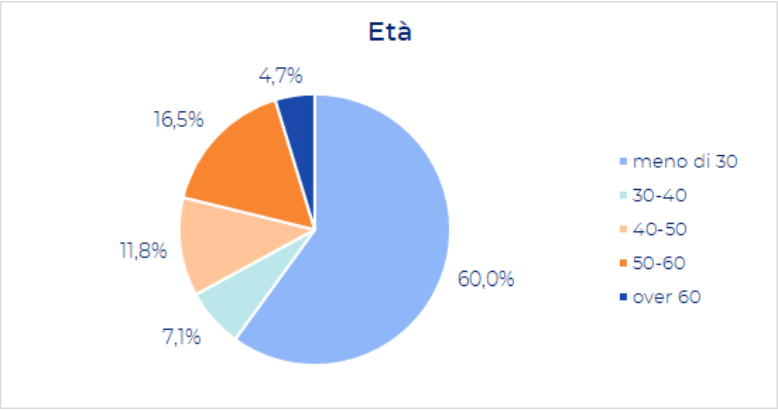


Figura 4.2 - Età del target

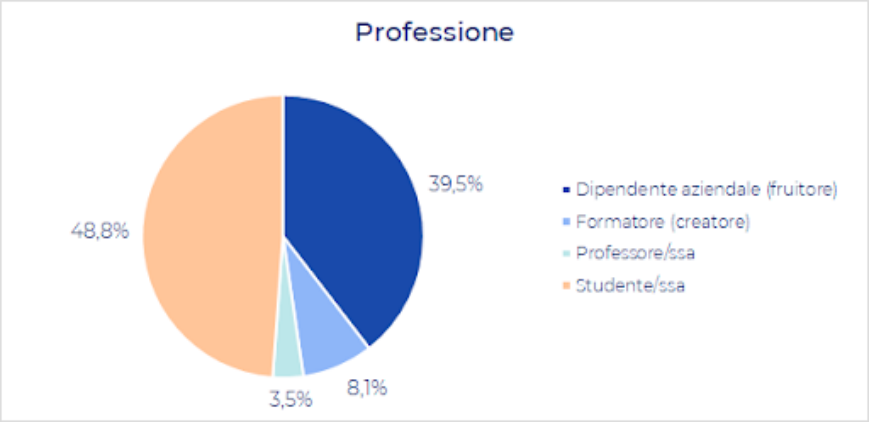


Figura 4.3 - Professione del target

In particolare, essendo un progetto finalizzato principalmente alla formazione aziendale è stato richiesto agli utenti provenienti da tale realtà di indicare il loro settore di provenienza, anche per confrontare il target raggiunto dalla survey con quello attualmente cliente dell'azienda inRebus Technologies. Nonostante su 41 soggetti solo in 27 abbiano reso pubblico tale informazione, dal grafico è possibile notare una considerevole varietà di ambiti.

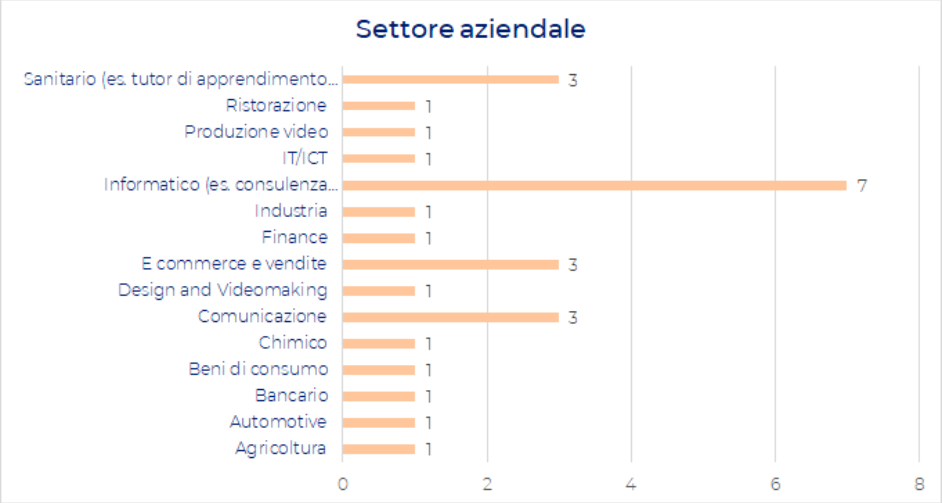


Figura 4.4 - Settori aziendali raggiunti

Fruitori e creatori sono stati gestiti separatamente nell'indagine. Ad entrambi sono state poste sia domande in comune come gli approcci avuti con video interattivi o opinioni su certi contenuti dello SmartVideo, sia domande più mirate soprattutto riguardo la configurazione dell'interfaccia. Di seguito sono riportati i dati registrati per le due categorie distinte.

4.2 Analisi dei fruitori

Le prime domande poste ai potenziali fruitori di SmartVideo sono servite ad avere un'idea base sul livello di conoscenza e di utilizzo dello strumento.

STUDENTI

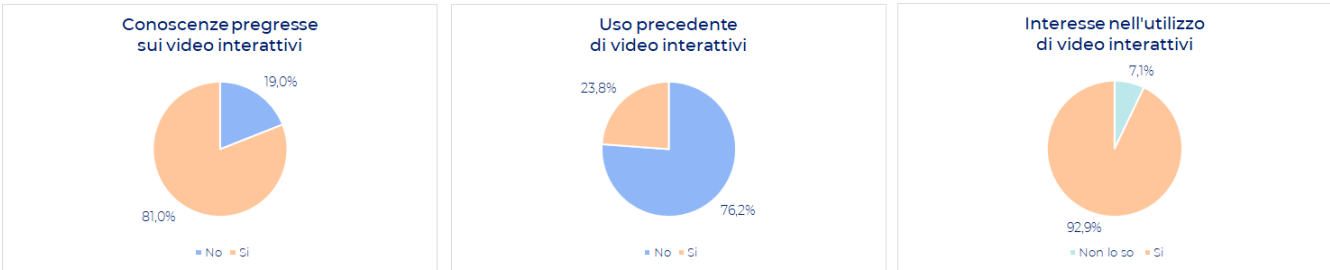


Figura 4.5 - Conoscenze e interesse da parte degli studenti

DIPENDENTI

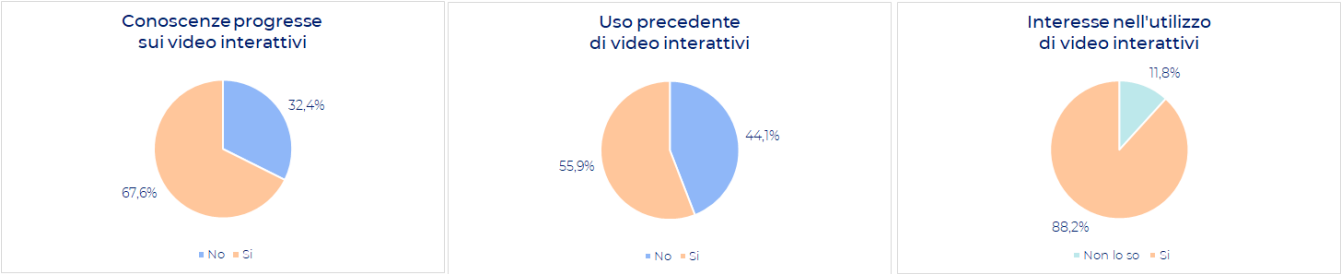


Figura 4.6 - Conoscenze e interesse da parte dei dipendenti aziendali

Nonostante la tecnologia sia maggiormente conosciuta dagli studenti, risulta evidente la disparità nel numero di soggetti che ne ha mai effettivamente usufruito: la correlazione tra "individui che conoscono" e "individui che usano" è decisamente più stretta nell'ambito aziendale rispetto a quello didattico in cui solo un limitato 23% ne fa o ne ha fatto uso. Aspetto molto positivo dell'indagine è rappresentato dal fatto che in entrambi i casi non ci sia stato nessuno totalmente disinteressato allo strumento, dando piuttosto spazio al beneficio del dubbio

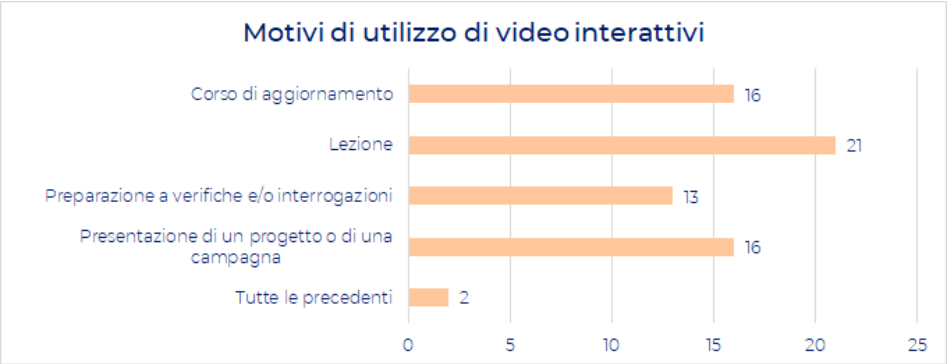


Figura 4.7 - Motivi di utilizzo di video interattivi

Ai soggetti interessati allo SmartVideo è stato richiesto anche di indicare un suo possibile utilizzo, dopo averne compreso le caratteristiche e le potenzialità. un picco di risposte nell'impiego del video interattivo come strumento di supporto alle lezioni, le restanti alternative sono circa allo stesso livello di interesse. In particolare un dipendente (over 60) ha espressamente indicato che userebbe lo SmartVideo per avere chiarimenti "poter comunicare efficacemente con il mio collega di un reparto diverso".

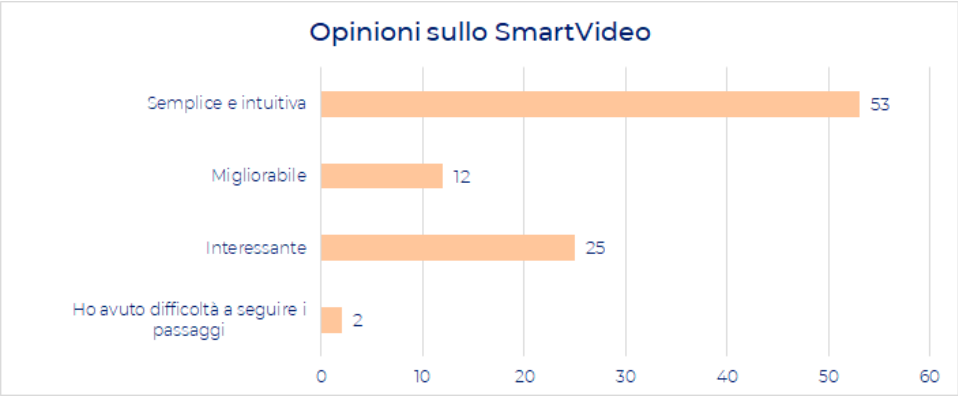


Figura 4.8 - Opinioni sullo SmartVideo (fruitori)

Infine, molto positivi sono stati i riscontri generali sul prodotto come si era sperato durante la fase di progettazione del prodotto. Su 69 fruitori ben 53 hanno trovato lo strumento "semplice e intuitivo", 25 "interessante". In questo caso le scelte selezionabili erano multiple e non mutuamente esclusive come nelle domande precedenti.

4.3 Analisi dei creatori

Analogamente ai casi dei fruitori è stato domandato per prima cosa le conoscenze dello strumento e in caso positivo gli ambiti di utilizzo.

PROFESSORI

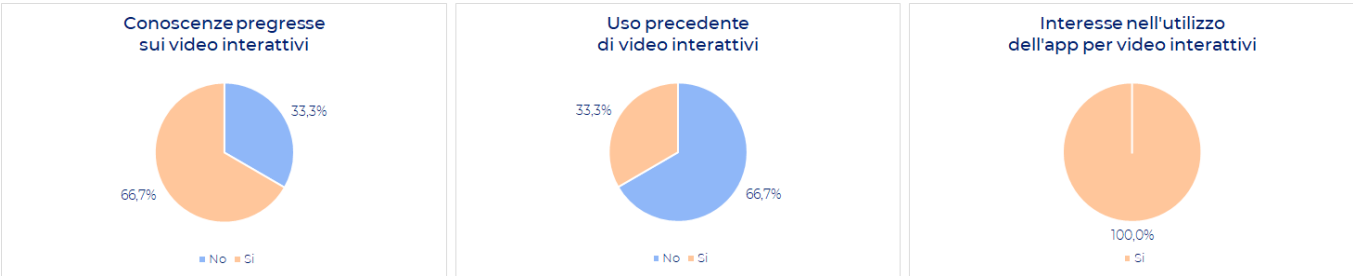


Figura 4.9 - Conoscenze e interesse da parte dei professori

FORMATORI

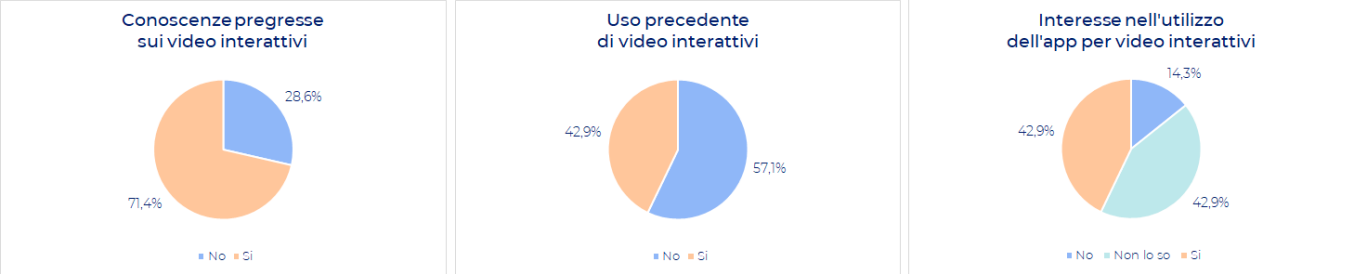


Figura 4.10 - Conoscenze e interesse da parte dei formatori aziendali

Riguardo i possibili creatori dello SmartVideo le differenze sostanziali tra formatori e professori sono rappresentate dal grado di interesse nell'impiego dell'applicazione. A differenza del caso studio di studenti e di dipendenti, è stato chiesto di esporre i motivi che spingerebbero a usufruire o meno dell'interfaccia: la maggior parte considera l'applicazione interessante, utile nello specifico ad esempio per la realizzazione di "tutorial per macchinari aziendali" e perchè "non troppo difficile da usare" o ancora "molto più completa e tutto sommato chiara e realizzabile, rispetto a quelle di cui sono a conoscenza"; in minoranza ma comunque presenti, alcuni soggetti preferiscono rimanere su mezzi tradizionali avvalendosi di "condivisione di risorse interattive real time - videoconferenze" o semplicemente perchè si considerano "refrattario a innovazioni così radicali". In generale, chi ha affermato di essersi già cimentato nell'utilizzo del prodotto ha dichiarato di averlo impiegato essenzialmente per lezioni, corsi di aggiornamento o per la presentazione di un progetto, ma nessuno ha saputo indicare un software o programma di riferimento attraverso cui realizzarli eccetto un caso in cui è stato indicato *Learningapps*.

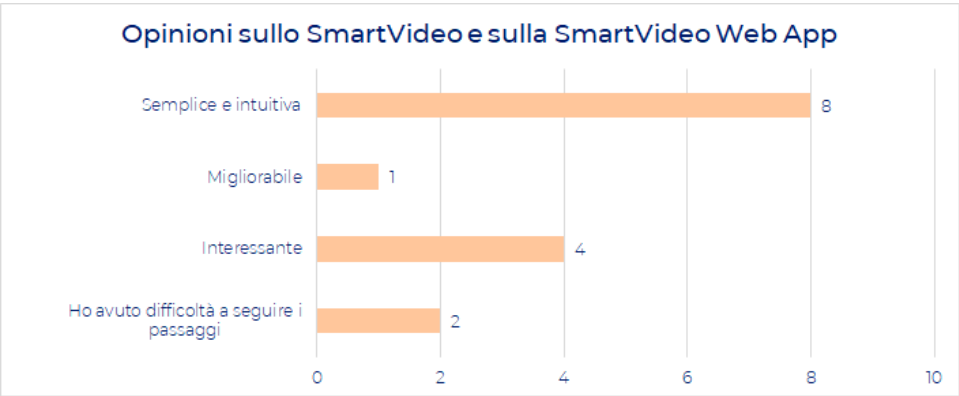


Figura 4.11 - Opinioni sullo SmartVideo (creatori)

Il grafico mostra che anche per quanto riguarda di formatori e professori il giudizio complessivo del prodotto risulta essere molto positivo. Ciò significa che dal punto di vista l'obiettivo è stato raggiunto a pieno. Gli unici commenti riportati riguardano sostanzialmente:

- personalizzazione del tema e dei colori dell'interfaccia, degli hotspot e dei link in generale;
- maggiore differenziazione tra hotspot di tipo approfondimento e quiz;
- implementazione di un tutorial "live" step by step realizzato con l'utilizzo di tooltip da mostrare al primo accesso al sito.

4.4 Analisi dei contenuti

La seconda parte della survey ha riguardato nello specifico le sezioni e i contenuti dell'applicazione e dello SmartVideo. Tutti i commenti ricevuti sono stati aggiunti all'elenco di ottimizzazioni future previste e suggerite che saranno presentate nel paragrafo 4.6.

Contenuti Smart

In generale la scelta di accostare al semplice testo un insieme di risorse di vario tipo ha sortito un effetto positivo: sia il pubblico creatore, sia quello fruitore hanno dato opinioni abbastanza simili riguardo i contenuti di approfondimento. In particolare una studentessa ha affermato

> "*Ho amato la possibilitá di poter visualizzare e scaricare un documento direttamente dal video!*"

Nonostante la presenza di accessi multipli ai contenuti, sia tramite hotspot sia grazie alle voci di accesso rapido poste nella parte inferiore del video, sia risultata abbastanza chiara circa un quinto del pubblico fruitore ha considerato superfluo l'impiego di entrambi gli strumenti o persino fastidioso (1 persona su 76) o ha affermato di preferire una riproduzione del video base sconnessa/incondizionata dalle interazioni dell'utente con gli hotspot.

Quiz

Formatori e professori hanno indicato che oltre ai quiz a risposta chiusa già implementati sarebbe utile prevederne anche a risposta multipla corretta e a risposte aperte. Tali tipologie risultano infatti ideali per test di avanzamento e per valutare competenze per cui la risposta "chiusa" è limitante, come la verifica delle conoscenze di una lingua straniera. Alcuni hanno aggiunto che sarebbe interessante poter inserire immagini nel testo della domanda o nelle risposte ad esempio mostrando un frammento di codice e chiedendo quale funzionalità rappresenti.

Anche dal punto di vista degli "studenti" le tipologie indicate sono simili a quelle dei formatori. In particolare, è stata richiesto un giudizio sulla visualizzazione, e in generale la gestione, attuale dei test di verifica. Non è presente una critica specifica, in numero abbastanza equo sono stati segnati:

- il desiderio di avere i quiz facoltativi durante la riproduzione e obbligatori a fine video;
- avere feedback immediati sulla risposta data;
- soddisfazione della presentazione attuale.

Glossario

Per quanto riguarda il Glossario, essendo puramento passivo attualmente, è stato chiesto di indicare quali potrebbero essere le possibili implementazioni da una serie di opzioni date. 25 persone su 86 hanno dichiarato di essere soddisfatti della condizione attuale, mentre dei restanti 61 soggetti è risultato che:

- il 34% vorrebbe una barra di ricerca all'interno dell'elenco delle voci;
- il 25% gradirebbe la comparsa di un hotspot o segnale visivo all'interno del video per indicare il momento e di conseguenza il contesto di utilizzo di una determinata voce;
- il 29% pensa sarebbe interessante avere un link diretto alla definizione e ai sinonimi nel web.

Una studentessa ha nello specifico indicato che sarebbe utile prevedere "*la possibilità di poter mettere in ogni voce del glossario uno o più link riguardanti*

degli articoli, o cose simili, che potrebbero essere interessanti e utili a chi sta visualizzando il video".

4.5 "Lo consiglieresti?"

Tra le ultime domande poste all'interno della survey è stata chiesta un'opinione al target sul consigliare il prodotto a professori, colleghi di lavoro, superiori. Eccetto casi rari restii o incerti sull'utilizzo dello SmartVideo e dell'applicazione per crearlo, i feedback sono stati tutti molto positivi. La maggior parte lo consiglierebbe per la semplicità d'uso, per l'utilità, per l'apprendimento di concetti difficili e perchè renderebbe più coinvolgente l'insegnamento e meno noiose le lezioni, aiutando a mantenere alta l'attenzione. Altri hanno affermato che considerano lo strumento un'ottimo mezzo di implementazione della formazione del personale e della didattica, "soprattutto nelle scuole primarie e secondarie di primo grado".

Di seguito sono riportati per intero alcuni commenti ricevuti.

"*Certo anzi sarebbe interessante utilizzarlo anche con gli studenti che fanno tirocinio per spiegare i vari argomenti ma non avendo basi di informatica non sono in grado di creare queste interessanti lezioni che catturano l'interesse, sono smart, e forniscono in tempo reale il feedback sono un'interfaccia attuale*"

- Donna, 50-60 anni, tutor di apprendimento clinico (infermieristica)

"*Potrebbe rivelarsi utile nella creazione di video tutorial interattivi dedicati a specifici argomenti di programmazione*"

- Uomo, 30-40 anni, formatore (finanza)

"*Certamente, anche perché in un periodo in cui le videolezioni sono diventate l'unico modo per continuare a portare avanti il programma scolastico, ritengo molto utile l'utilizzo di mezzi sempre nuovi.*"

- Uomo, meno di 30 anni, studente

4.6 Ottimizzazioni future

4.6.1 Ottimizzazioni previste nello sviluppo del progetto

L'applicazione è stata sviluppata implementando esclusivamente le funzioni base ritenute necessarie ai fini del progetto. In fase di realizzazione sono state stilate una serie di ottimizzazioni sia a livello strutturale sia a livello dei contenuti.

Nel primo caso rientrano:

- la traduzione della base dati passando a un database distribuito, document-based, non relazionale come MongoDb[1], ideale per lo sviluppo di applicazioni moderne e per una più efficiente gestione dei dati, soprattutto dei file, sia in fase di caricamento sia per i backup generali delle informazioni sul server;
- completamento della gestione della sicurezza dei dati dell'utente;
- implementazione dell'accesso condiviso (share edit) per la modifica dei progetti;
- lato server, passaggio a un framework php come Laravel o a linguaggi come Python;
- ottimizzazione dello store di Redux.

Dal punto di vista dei contenuti si prospettano:

- la visualizzazione degli SmartVideo in versione responsive per tablet e smartphone;
- l'implementazione di nuove tipologie di quiz;
- la suddivisione in capitolo (o segnalibri) dello SmartVideo per una navigazione più personalizzata e non lineare dei contenuti;
- eventualmente, l'aggiunta di una funzione di "Recensione" del prodotto per permettere a professori e formatori di avere dei feedback qualitativi sulla lezione o sul corso tenuti.

Nello specifico per i contenuti è stato chiesto al target quali funzione

avessero la priorità di realizzazione.

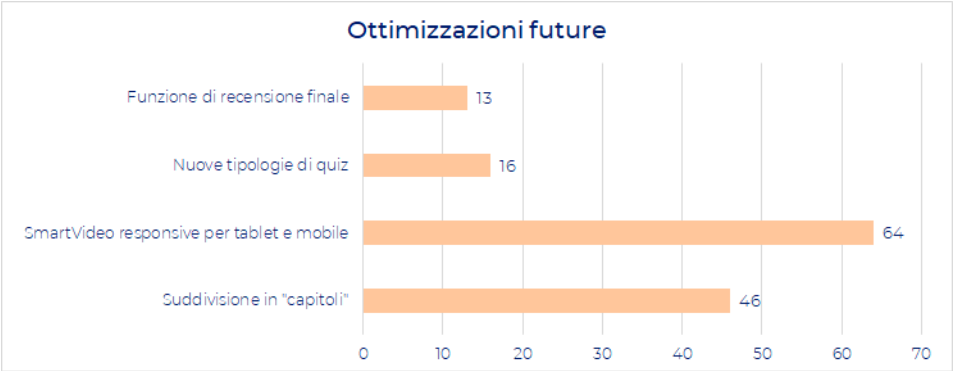


Figura 4.12 - Ottimizzazioni future

Il grafico riporta una richiesta considerevole di sviluppo delle versioni responsive (77,4%), seguita dall'implementazione dei segnalibri all'interno dello SmartVideo (56%).

4.6.2 Ottimizzazioni suggerite dal target

L'ultima domanda è stata riservata a suggerimenti e opinioni generali del target sul prodotto. Per la maggior parte si tratta di commenti sull'indice di gradimento del progetto sia in generale, sia relativo a singoli contenuti. Tra i 26 commenti rilasciati 6 sono relativi a possibili miglioramenti della gestione degli hotspot. I punti salienti evidenziati sono riportati nei seguenti commenti.

> *"Il cerchio che indica la comparsa di un hotspot lo metterei in un punto fisso e riconoscibile, come in alto a destra del video, e aggiungerei un'immagine riconoscibile a seconda del video (per esempio un computer per informatica). Farei anche in modo che uscendo e rientrando dal video gli hotspot rimangano e non si resettino"*
>
> - Uomo, meno di 30 anni, studente

[1] https://www.mongodb.com/it

"Se è possibile forse modificherei leggermente il pallino rosso dell'hotspot... Si potrebbe ad esempio semplicemente aggiungere un contorno di un altro colore (bianco o nero per dire) intorno al pallino così da renderlo più facilmente visibile."

- Uomo, meno di 30 anni, studente

"Secondo me come hotspot andrebbero bene anche tracce audio, semplici immagini schematiche, poche parole chiave"

- Uomo, meno di 30 anni, dipendente aziendale fruitore

"Mi rendo conto che l'hotspot debba essere ben visibile e riconoscibile, ma in alcuni casi l'ho trovato un po' invadente... Forse troverei un'icona/simbolo da far comparire (in maniera che possa comunque attirare l'attenzione, accompagnata da un suono o animazione o punto luce tipo lampadina che accende, etc.) in una posizione fissa e ai bordi del video che l'utente possa identificare immediatamente come fonte di informazioni."

- Donna, meno di 30 anni, studentessa

"Essendo daltonica, per la soluzione delle risposte proporrei colori come blu e giallo ocra"

- Donna, meno di 30 anni, studentessa

Conclusioni

Nella società moderna l'e-learning è una pratica oramai altamente diffusa da anni e sta continuando a crescere e ad affermarsi nella vita di tutti i giorni degli individui. Con l'esplosione del COVID-19 questo fenomeno ha assunto ancora più importanza: il tradizionale concetto di "presenza" è stato rivisitato con un'ottica digitale, favorendo pratiche online per poter garantire una continua educazione. É in questo panorama che si affermano tecnologie nuove, sempre aggiornate che si fanno carico di ottenere gli stessi risultati che si avrebbero con i mezzi tradizionali nonostante la componente della distanza.

Il progetto di tesi ha l'obiettivo di supportare i processi di formazione aziendale, e più in generale didattica, fornendo un prodotto interattivo alla portata di tutti e un'applicazione per realizzarlo. Ai fini dello sviluppo l'azienda inRebus Technologies ha fornito delle bozze relative a un prototipo di strumento da loro impiegato, chiamato "hyperfilm". Dall'analisi delle sue caratteristiche e soprattutto delle sue limitazioni è stato definito un insieme di specifiche che regolano il comportamento atteso del sistema, ne è stata fatta una progettazione, lo si è implementato e lo si è sottoposto ad un certo numero di utenti che lo hanno testato. Dalle valutazioni ottenute si evince che il sistema realizzato ha raggiunto completamente gli obiettivi preposti, ovvero essere semplice, intuitivo e di massima usabilità. I requisiti minimi posti all'inizio in fase di sviluppo sono stati rispettati e confermati dal target; ciò non toglie che per essere effettivamente integrata negli attuali processi di formazione siano necessari dei miglioramenti in diversi aspetti, sia a livello gestionale, sia a livello di contenuti per favorire una modalità di apprendimento a 360° gradi coinvolgente ed efficace. Primo fra tutti

l'implementazione della versione responsive per dispositivi mobile e tablet.

In conclusione, al lavoro di tesi, un augurio per il futuro. La tecnologia si evolve molto rapidamente, di conseguenza sarà sempre possibile realizzare progetti all'avanguardia per rispondere alle esigenze esistenti. L'augurio è per un futuro in cui sia possibile garantire a chiunque l'accesso alla formazione, in qualsiasi luogo e in qualsiasi momento.

Bibliografia e sitografia

Articoli

- Meixner B., *Hypervideos and Interactive Multimedia Presentations*, in *ACM Computing Surveys*,n. 9, marzo 2017, DOI: 10.1145/3038925
- Chambel T., Bidarra J., Guimarães N. M., *Learning with Video in Hypermedia*, Tech. Report DI/FCUL, TR-01-16, Univ. of Lisbon, Dicembre 2001, www.di.fc.ul.pt/tech-reports/ 01-16.pdf.
- Chambel T., Bidarra de Almeida J., Guimarães N., *Multimedia Artifacts That Help Us Learn: Perspectives of the UNIBASE Project on Distance Learning*, Workshop su Multimedia and Educational Practice, ACM Multimedia, Bristol, UK 1998.
- Chambel T., Correia N., Guimarães, N., *Hypervideo on the Web: Models and Techniques for Video*, in *Integration. International Journal of Computers & Applications*, Acta Press, Vol. 23, n. 2, 2001

Siti

- https://www.dyndevice.com/it/news/come-nasce-come-si-sviluppato-l-learning-ELN-445/, visitato il 23/02/2021
- https://www.randstad.it/knowledge360/gestione-del-personale/e-learning-come-si-evolve-il-concetto-di-formazione-in-azienda/, visitato il 23/02/2021
- https://www.eni.com/it-IT/carriere/formazione-online-coronavirus.html, visitato il 23/02/2021
- https://www.videosmart.com/blog/post/interactive-black-mirror-bandersnatch, visitato il 27/03/2021
- https://www.iperdesign.com/it/2016/11/18/dallipertesto-al-video-interattivo-nuove-frontiere-per-la-formazione-line/, visitato il 27/03/2021
- https://www.telemaco.it/blog/strategia-di-comunicazione/, visitato il 27/03/2021

- https://www.mark-up.it/comunicare-un-brand-il-futuro-e-nel-video-marketing/, visitato il 01/04/2021
- https://www.comunicaresulweb.com/web-marketing/motivi-per-usare-video-marketing/, visitato il 01/04/2021
- https://www.mosaicoelearning.it/blog/facili-video-interattivi-le-learn, visitato il 01/04/2021
- https://www.nextre.it/usare-video-interattivi-e-learning/ , visitato il 01/04/2021
- https://www.docebo.com/it/blog/formazione-video-on-demand-come-e-perche/, visitato il 02/04/2021
- https://www.emathe.it/video-per-lapprendimento-formazione-aziendale/, visitato il 02/04/2021
- https://www.elearningvincente.it/video-elearning-microlearning-videolearning/ , visitato il 01/04/2021
- https://www.iperdesign.com/it/2016/11/18/dallipertesto-al-video-interattivo-nuove-frontiere-per-la-formazione-line/, visitato il 03/04/2021
- https://corp.kaltura.com/video_resource/kaltura-mediaspace-overview/, visitato il 02/04/2021
- https://aretepiattaforma.it/tutorial/18/Come-si-fa-per-creare-video-lezioni-con-Edpuzzle , visitato il 02/04/2021
- https://blog.hubspot.com/marketing/web-forms#:~:text=Why%20Should%20I%20Create%20Web,until%20it%27s%20ready%20for%20analysis, visitato il 01/04/2021
- https://www.html.it/articoli/i-migliori-font-sans-serif-per-il-web/, visitato il 28/03/2021
- https://www.codingcreativo.it/font-per-il-sito-web/, visitato il 28/03/2021
- https://it.qaz.wiki/wiki/Web_application, visitato il 08/03/2021
- https://www.sviluppati.com/2016/04/05/web-application-quali-i-vantaggi/ , visitato il 08/03/2021
- https://www.ionos.it/digitalguide/hosting/tecniche-hosting/che-cose-innodb/ , visitato il 09/03/2021
- https://www.html.it/pag/32137/introduzione-ai-rdbms, visitato il 09/03/2021
- http://www-db.disi.unibo.it/courses/BDPG/normaliz.pdf, visitato il 11/03/2021
- https://www.seneta.it/2018/08/linguaggio-sql-vincoli-di-integrita-dei-dati/#:~:text=I%20vincoli%20di%20integrit%C3%A0%20su%20una%20tabella%20impongono%20delle%20regole,da%20due%20o%20pi%C3%B9%20colonne, visitato il 10/03/2021
- https://www.britannica.com/science/computer-science/Information-management#ref1278029, visitato il 11/03/2021
- https://www.html.it/pag/16673/cos-php/, visitato il 12/03/2021

- https://blog.dangerx82.it/joomla/121-cosa-xampp.html, visitato il 13/03/2021
- https://www.phpmyadmin.net/, visitato il 13/03/2021
- https://developer.mozilla.org/it/docs/Web/JavaScript/Reference/Global_Objects/Promise, visitato il 13/03/2021
- https://zetcode.com/javascript/axios/, visitato il 13/03/2021
- https://redux.js.org/introduction/getting-started , visitato il 14/03/2021
- https://www.digitalocean.com/community/tutorials/redux-redux-thunk, visitato il 13/03/2021
- https://redux.js.org/tutorials/essentials/part-5-async-logic, visitato il 14/03/2021
- https://www.html.it/pag/62463/le-richieste-http-get-e-post/, visitato il 152/03/2021
- https://www.w3schools.com/php/func_mysqli_real_escape_string.asp, visitato il 15/03/2021
- https://www.trendoo.it/blog/codice-otp-via-sms/, visitato il 12/03/2021
- https://www.codexworld.com/upload-store-image-file-in-database-using-php-mysql/ , visitato il 14/03/2021
- https://www.php.net/manual/en/reserved.variables.files.php, visitato il 15/03/2021
- https://siddharthac6.medium.com/getting-started-with-react-js-using-webpack-and-babel-66549f8fbcb8 , visitato il 17/03/2021
- https://it.reactjs.org/docs/components-and-props.html , visitato il 18/03/2021
- https://www.educative.io/edpresso/what-is-a-react-component, visitato il 19/03/2021
- materiale corso Interfacce grafiche , visitato il 19/03/2021
- https://blog.logrocket.com/a-guide-to-usestate-in-react-ecb9952e406c/, visitato il 19/03/2021
- https://dmitripavlutin.com/react-useeffect-explanation/ , visitato il 21/03/2021
- https://it.reactjs.org/docs/hooks-effect.html, visitato il 21/03/2021
- https://reactrouter.com/web/guides/primary-components, visitato il 21/03/2021
- https://redux.js.org/tutorials/essentials/part-2-app-structure , visitato il 24/03/2021
- https://reactrouter.com/web/api , visitato il 214/03/2021
- https://www.npmjs.com/package/yup , visitato il 24/03/2021
- https://dev.to/amberjones/5-delightful-things-about-material-ui-5402, visitato il 24/03/2021