

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



Applicazione Cloud per la riproduzione video on-demand a qualità dinamica

Tesi di laurea

Relatore

Prof. Ombretta Gaggi

Laureando

Mattia Brunello

ANNO ACCADEMICO 2022-2023

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, dal laureando Mattia Brunello presso l'azienda AdMaioraStudio S.r.l.

Gli obbiettivi da raggiungere erano molteplici.

In primo luogo lo studio e la documentazione delle varie opzioni per lo sviluppo di un'applicazione cloud, tecniche e tecnologie utilizzate e i loro possibili futuri sviluppi.

In secondo luogo, lo sviluppo e l'analisi di un PoC di una WebApp per la gestione dello streaming di video di prodotti di vari espositori nelle fiere mondiali.

In terzo luogo le conclusioni con la documentazione completa degli artefatti sviluppati e definizione dei possibili casi d'uso futuri.

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. Ombretta Gaggi, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

Desidero ringraziare con affetto i miei genitori e i miei nonni per il sostegno, il grande aiuto e per essermi stati vicini in ogni momento durante gli anni di studio.

Padova, Luglio 2023

Mattia Brunello

Indice

1	Introduzione	1
1.1	L'idea	1
1.2	L'azienda	2
1.3	Struttura della tesi	2
2	Fondamenti teorici	3
2.1	Concetti di video on-demand e streaming	3
2.1.1	Tipologie di video on-demand	3
2.1.2	Concetti di streaming video	4
2.1.3	Protocolli e tecnologie di streaming video	4
2.2	I problemi dello streaming	5
2.2.1	Latenza	5
2.2.2	Qualità del video	5
2.2.3	Sicurezza	5
2.2.4	Gestione del carico del server	6
2.3	Tecnologie utilizzate	6
2.3.1	React	6
2.3.2	C#	7
2.3.3	Azure	7
3	Analisi dei requisiti	8
3.1	Descrizione delle esigenze dell'azienda	8
3.2	Individuazione e specifica dei requisiti funzionali e non funzionali	8
3.3	Casi d'uso	8
4	Progettazione	9
4.1	Descrizione dell'Architettura del sistema	9
4.2	Progettazione del frontend in React	9
4.3	Progettazione del backend in C Sharp	9
4.4	Integrazione con Azure Media Services	9
5	Implementazione	10
5.1	Descrizione delle principali scelte implementative	10
5.2	Dettagli sull'implementazione del frontend	10
5.3	Dettagli sull'implementazione del backend	10
5.4	Dettagli sull'implementazione dell'integrazione con Azure Media Services	10
6	Testing e validazione	11
6.1	Descrizione delle attività di testing	11

<i>INDICE</i>	vi
6.2 Verifica dei requisiti	11
6.3 Analisi dei risultati	11
6.4 Valutazione delle prestazioni della webapp	11
7 Conclusioni	12
7.1 Consuntivo finale	12
7.2 Raggiungimento degli obiettivi	12
7.3 Conoscenze acquisite	12
7.4 Valutazione personale	12
A Appendice A	13
Bibliografia	15

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

1.1 L'idea

Nell'attuale panorama delle fiere e degli eventi commerciali, le aziende partecipanti hanno manifestato un crescente interesse nella promozione innovativa dei propri prodotti. Attualmente, essa avviene principalmente attraverso la distribuzione di materiale pubblicitario, come brochure, volantini e cataloghi, lasciando al visitatore il compito di informarsi autonomamente sui prodotti offerti dai vari espositori.

In questa tesi verrà descritto lo sviluppo di una WebApp attraverso la quale gli espositori avranno la possibilità di caricare i propri video relativi ai prodotti esposti, rendendoli successivamente disponibili per la riproduzione on-demand da parte dei visitatori. L'idea rappresenta un passo avanti verso l'innovazione della promozione dei prodotti nelle fiere, perché offre ai partecipanti un'esperienza interattiva e coinvolgente.

L'applicazione è stata sviluppata utilizzando come linguaggio di backend C#, per lo sviluppo del frontend il framework JavaScript React, ¹ mentre per la gestione dell'archiviazione e streaming dei video sono stati utilizzati i servizi di Microsoft Azure: Media Service e Account Storage. ²

L'obiettivo principale di questo POC, è stato quello di studiare e verificare la fattibilità di un prodotto di questo tipo.

¹<https://reactjs.org/>

²<https://azure.microsoft.com/>

1.2 L'azienda

Ad Maiora Studio è una software house nata nel 2013 per operare nel campo del mobile e che negli anni si è specializzata anche nello sviluppo di software.

La sua mission è centrata sull'attenzione verso i clienti e lo sviluppo di software moderni, scalabili e progettati ad hoc per soddisfare le loro esigenze.

I prodotti sviluppati da Ad Maiora Studio si basano sulle più recenti tecnologie, integrano componenti e librerie eterogenee e pongono una forte enfasi sull'esperienza utente e l'interfaccia grafica.

L'azienda si distingue per l'approccio continuativo di assistenza ai clienti, garantendo un partner sempre accessibile e in grado di rispondere tempestivamente alle richieste.

Ad Maiora Studio si concentra principalmente su piccole e medie imprese operanti nei settori industriale e dei servizi, incoraggiandole a intraprendere un percorso di modernizzazione iniziando dal software.

L'obiettivo è supportare efficacemente l'adattamento alle mutevoli esigenze di mercato e di business, fornendo strumenti innovativi e personalizzati, che rappresentano il cuore dell'attività di Ad Maiora Studio.

Grazie alla competenza Full Stack del team di sviluppatori, l'azienda è in grado di realizzare ogni tipo di software, coprendo l'intero processo di sviluppo, dalla progettazione all'implementazione.

La qualità delle soluzioni software offerte è sempre un punto focale, al fine di soddisfare appieno le aspettative dei clienti e garantire il massimo risultato.

1.3 Strutta della tesi

Il secondo capitolo descrive i fondamenti teorici

Il terzo capitolo descrive l'analisi dei requisiti

Il quarto capitolo approfondisce la progettazione

Il quinto capitolo approfondisce l'implementazione

Il sesto capitolo approfondisce la fase di testing e validazione

Il settimo capitolo descrive le conclusioni e i possibili sviluppi futuri

Capitolo 2

Fondamenti teorici

In questo capitolo verranno descritti i fondamenti teorici necessari per la comprensione del lavoro svolto.

2.1 Concetti di video on-demand e streaming

Video on-demand (tradotto come video su richiesta) o VOD, è un sistema che permette di accedere a contenuti multimediali (video, audio, immagini) in qualsiasi momento e in qualsiasi luogo tramite una connessione internet. Contrariamente alla trasmissione televisiva tradizionale, nella quale gli utenti sono limitati da un palinsesto predefinito, il VOD dà la possibilità agli utenti di scegliere quale contenuto guardare e quando usufruirne.

Nel contesto della webapp sviluppata, questo concetto ha una funzione chiave, infatti consente agli utenti di accedere a una vasta gamma di selezione video riguardanti i prodotti esposti nelle fiere mondiali, permettendo di scegliere i video di loro interesse in base alle loro preferenze e necessità, eliminando le limitazioni spazio-temporali delle fiere fisiche.

Nel corso della tesi, verranno analizzate le caratteristiche e le sfide associate all'implementazione del video on-demand nella webapp, quindi le strategie di gestione e organizzazione dei contenuti, nonché la scalabilità e la qualità dello streaming per garantire un'esperienza fluida e coinvolgente per gli utenti.

2.1.1 Tipologie di video on-demand

Esistono diverse tipologie di video on-demand, sotto elencate:

- **Subscription VOD** ovvero i servizi con un canone periodico come ad esempio Netflix, Amazon Prime Video ecc..
- **Transactional VOD** ossia servizi che permettono di acquistare o noleggiare contenuti, come ad esempio Google Play, Apple TV, Chili ecc..
- **Advertising VOD** ossia servizi gratuiti che mostrano annunci pubblicitari durante la riproduzione dei contenuti, come ad esempio Youtube, RaiPlay e

Mediaset Play

- **Premium VOD** ovvero la trasmissione di contenuti Premium, come anteprime cinematografiche, eventi sportivi ecc., proposti da piattaforme come ad esempio Curzon Cinemas

2.1.2 Concetti di streaming video

Lo streaming video è un metodo di trasmissione di dati multimediali, in particolare di video e audio. Esistono due principali categorie di streaming video:

- **Video on-demand** è la trasmissione di contenuti pre-registrati, come ad esempio film, serie TV, documentari ecc., i quali vengono compressi e memorizzati su un server come file, e vengono trasmessi agli utenti che ne fanno richiesta senza la necessità che il contenuto venga scaricato sul dispositivo dell'utente. Infatti i dati ricevuti dalla richiesta vengono decompressi e riprodotti in tempo reale.
- **Live streaming** è simile alle trasmissioni televisive tradizionali, in cui gli utenti guardano i contenuti in tempo reale. Viene utilizzato per trasmettere eventi in diretta come ad esempio concerti, eventi sportivi ecc., vengono anch'essi leggermente compressi e memorizzati su un server, ma vengono trasmessi in tempo reale agli utenti che ne fanno richiesta.

2.1.3 Protocolli e tecnologie di streaming video

Per la trasmissione di contenuti multimediali, esistono diversi protocolli e tecnologie, sotto elencati:

- **HTTP Live Streaming** o HLS è un protocollo di streaming sviluppato da Apple nel 2009, permette la trasmissione in streaming di contenuti multimediali, frammentando il contenuto in segmenti di file HTTP e li distribuisce ai dispositivi client utilizzando il medesimo protocollo.
HLS è adattivo: il client può cambiare la qualità del video in base alla larghezza di banda disponibile senza interrompere la riproduzione. È nativamente compatibile con i dispositivi Apple e è supportato anche dalla maggior parte dei dispositivi e browser che supportano HTTP, non richiedendo l'installazione di plugin aggiuntivi.
- **Dynamic Adaptive Streaming over HTTP** o DASH è un protocollo di streaming sviluppato dal Moving Picture Experts Group (MPEG), permette la trasmissione di contenuti multimediali attraverso il protocollo HTTP.
DASH suddivide il contenuto in segmenti e li trasmette ai dispositivi client tramite HTTP, permettendo un adattamento dinamico della qualità del video in base alla larghezza di banda disponibile. Offre una vasta gamma di scelta del formato video e codec, permettendo di scegliere il formato più adatto per il dispositivo client.
- **Real Time Messaging Protocol** o RTMP è un protocollo di streaming sviluppato da Adobe nel 2012, permette la trasmissione di contenuti multimediali in

tempo reale, divide il contenuto in pacchetti e li trasmette ai dispositivi client tramite TCP o UDP, consente una comunicazione bidirezionale tra il server e il dispositivo client utilizzando un flusso continuo.

RTMP è un protocollo di streaming non adattivo, ovvero non permette di cambiare la qualità del video in base alla larghezza di banda disponibile, ma permette di trasmettere contenuti in tempo reale con una bassa latenza.

Dal 2020, con la deprecazione di Adobe Flash Player, RTMP è stato sostituito da protocolli di streaming adattivi come HLS e DASH.

2.2 I problemi dello streaming

2.2.1 Latenza

La latenza è il tempo di ritardo tra l'invio di un pacchetto e la ricezione di una risposta, è un problema comune nello streaming, in quanto può causare ritardi nella riproduzione del video. Può essere causata da diversi fattori, come ad esempio la velocità delle connessioni, la distanza tra il server e il dispositivo client, la compressione del video e la capacità di elaborazione del dispositivo client. Per ridurre la latenza si utilizzano protocolli efficienti in base alla tipologia del contenuto.

2.2.2 Qualità del video

Un aspetto importante dello streaming video è la qualità del video, essa dipende da diversi fattori, i due principali sono la compressione e la codifica del video.

- **La compressione** è un processo che riduce la dimensione del file video, rimuovendo le informazioni ridondanti o non necessarie, permettendo una trasmissione più rapida ed efficiente del video. Può essere di due tipi: lossless e lossy. La compressione lossless è un processo che riduce la dimensione del file video senza perdita di qualità, mentre la compressione lossy è un processo che riduce la dimensione del file video con una leggera perdita di qualità.
- **La codifica** è un processo che converte il video da un formato a un altro formato, consentendo la trasmissione e riproduzione dei video su diversi dispositivi e piattaforme. Esistono diversi formati video, i più comuni sono: H.264, H.265, VP9 e AV1.

2.2.3 Sicurezza

La sicurezza è un aspetto importante dello streaming video, in quanto i contenuti multimediali possono essere facilmente copiati e distribuiti senza autorizzazione. Per proteggerlo esistono diversi metodi, i più comuni sono: Digital Rights Management (DRM) e Watermarking.

Il DRM è un metodo di protezione da copie non autorizzate, impostando una chiave di protezione, che viene utilizzata per decodificare i contenuti multimediali.

Il Watermarking è un altro metodo di protezione che permette di proteggere i contenuti

con un watermark, ovvero un segno distintivo (come ad esempio un logo), che viene utilizzato per identificare il proprietario.

2.2.4 Gestione del carico del server

I server di streaming possono ricevere un elevato numero di richieste da tutto il mondo, questo può causare dei problemi di distribuzione del contenuto alle richieste più distanti geograficamente dal server, in quanto i pacchetti impiegano più tempo a raggiungere il dispositivo client e quindi causano ritardi nella riproduzione del video.

Inoltre per eventi particolarmente popolari come ad esempio eventi sportivi, il numero di richieste può aumentare notevolmente, causando un sovraccarico del server e quindi ritardi nella riproduzione del video, in quanto la banda disponibile è limitata e viene distribuita tra tutti i dispositivi client.

Per risolvere questi problemi si utilizzano dei servizi di Content Delivery Network (CDN), ovvero una rete di server presenti in tutto il mondo, che permette di distribuire i contenuti multimediali ai dispositivi client più vicini geograficamente, riducendo la latenza e il carico dei vari server.

Mentre per la gestione del carico del server, si utilizzano dei servizi di Load Balancing, ovvero dei servizi di bilanciamento di carico, come il bilanciamento di carico di rete (NLB) o il bilanciamento di carico applicativo (ALB), che permettono di distribuire il traffico più efficientemente tra i server, riducendone il carico.

Inoltre per gestire i picchi di richieste, si utilizzano infrastrutture server scalabili basate su cloud, come ad esempio Azure Media Services di Microsoft, che permette di scalare automaticamente le risorse in base alle esigenze del traffico di richieste, mantenendo il servizio sempre disponibile e riducendo i costi di gestione.

2.3 Tecnologie utilizzate

Per la realizzazione sono state utilizzate le seguenti tecnologie:

2.3.1 React

React è una libreria JavaScript open source, sviluppata da Meta nel 2013, permette la creazione di componenti riutilizzabili e la gestione dello stato dell'applicazione in modo efficiente. Scompone l'interfaccia utente in piccoli componenti modulari, ciascuno responsabile di una specifica parte dell'interfaccia, garantendo una maggior facilità di sviluppo e manutenibilità del codice.

Una caratteristica distintiva di React è la sua efficacia di rendering, infatti, grazie al virtual DOM (Dynamic Object Model), permette di aggiornare solo le parti dell'interfaccia che vengono modificate, senza dover ricaricare l'intera pagina, garantendo una maggiore efficienza e velocità di rendering.

È spesso utilizzato insieme ad altre librerie come React Router, che permette di gestire le rotte dell'applicazione.

2.3.2 C#

C# è un linguaggio di programmazione orientato agli oggetti, sviluppato da Microsoft nel 2000, è un linguaggio di programmazione multiparadigma, supporta i paradigmi di programmazione procedurale, funzionale, generica, orientata agli oggetti e asincrona. È un linguaggio di programmazione fortemente tipizzato, in quanto ogni variabile deve essere dichiarata con un tipo specifico, e supporta la programmazione generica, in quanto permette di creare classi e metodi generici, che possono essere utilizzati con tipi diversi.

È spesso utilizzato per lo sviluppo di applicazioni web, desktop e mobile, in quanto permette di creare applicazioni efficienti e affidabili.

Ha un ampio supporto per il framework .NET, il quale offre una vasta gamma di librerie e strumenti per lo sviluppo di applicazioni backend e non solo, inoltre, il framework .NET include ASP.NET, un framework utilizzato per lo sviluppo di applicazioni web, che consente la creazione di API RESTful con facilità, grazie anche alla gestione avanzata delle richieste HTTP.

2.3.3 Azure

Azure è una piattaforma di cloud computing, sviluppata da Microsoft nel 2010, permette di creare, testare, distribuire e gestire applicazioni e servizi tramite i data center di Microsoft. Offre una ampia varietà di servizi, come servizi di elaborazione, servizi di archiviazione e database, servizi di rete, servizi di intelligenza artificiale e servizi di sicurezza.

Il vantaggio principale di Azure è la scalabilità, infatti permette di scalare le risorse di elaborazione e di rete in base alle esigenze dell'applicazione, inoltre permette di ridurre i costi di gestione, in quanto ci sono diversi piani di pagamento, che permettono di pagare solo le risorse utilizzate.

Capitolo 3

Analisi dei requisiti

In questo capitolo verranno descritti i requisiti individuati per lo sviluppo del POC.

3.1 Descrizione delle esigenze dell'azienda

3.2 Individuazione e specifica dei requisiti funzionali
e non funzionali

3.3 Casi d'uso

Capitolo 4

Progettazione

- 4.1 Descrizione dell'Architettura del sistema
- 4.2 Progettazione del frontend in React
- 4.3 Progettazione del backend in C Sharp
- 4.4 Integrazione con Azure Media Services

Capitolo 5

Implementazione

In questo capitolo verrà descritta l'implementazione del POC.

- 5.1 Descrizione delle principali scelte implementative
- 5.2 Dettagli sull'implementazione del frontend
- 5.3 Dettagli sull'implementazione del backend
- 5.4 Dettagli sull'implementazione dell'integrazione con Azure Media Services

Capitolo 6

Testing e validazione

In questo capitolo verranno descritte le attività di testing e validazione svolte sul POC.

6.1 Descrizione delle attività di testing

6.2 Verifica dei requisiti

6.3 Analisi dei risultati

6.4 Valutazione delle prestazioni della webapp

Capitolo 7

Conclusioni

7.1 Consuntivo finale

7.2 Raggiungimento degli obiettivi

7.3 Conoscenze acquisite

7.4 Valutazione personale

Appendice A

Appendice A

Citazione

Autore della citazione

Bibliografia