# Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica



# Analisi ed implementazione prototipale di un software di condivisione audio/video per la collaborazione remota

*Tesi di laurea*


*Relatore*

Prof.Davide Bresolin

*Laureando*

Denis Salviato



Anno Accademico 2021-2022

a Gessica, ai miei nonni e
a me stesso.

# Ringraziamenti

*Desidero ringraziare il professor Davide Bresolin per la professionalità, la disponibilità e il sostegno fornitomi durante la stesura del lavoro.*

*Desiderio ringraziare l'azienda per avermi permesso di svolgere il percorso di tirocinio, aiutandomi a crescere professionalmente.*

*Desidero ringraziare i miei genitori e mia sorella per il supporto durante gli anni di studio.*

*Desidero ringraziare i genitori della mia compagna, per avermi compreso ed aiutato a superare i momenti difficili, motivandomi e trattandomi sempre come un figlio.*

*Desidero ringraziare Tommaso per essere stato un amico, per essere stato presente anche quando la distanza non lo permetteva e di essere ancora oggi una fonte di ispirazione.*

*Desidero ringraziare Veronica per essere stata una vera amica, per essere stata un punto di riferimento e per essere stata sempre presente come una sorella.*

*Desidero ringraziare Gessica, la mia findanzata e compagna da una vita, per essere la persona che sono oggi, per essere riuscito ad arrivare in fondo, per la pazienza sopportata e i sacrifici fatti, ma soprattutto per non aver mai smesso di credere in me.*

*Padova, Luglio 2022* Denis Salviato

# Sommario

Questo documento presenta il lavoro svolto durante il percorso di tirocinio presso l'azienda ospitante Innovative Digital eXperience S.r.l. che ha sede a Padova.

Oggigiorno stiamo vivendo un periodo storico caratterizzato da profonde trasformazioni economiche, politiche e sociali, che hanno subito però una notevole accelerazione a causa dell'emergenza sanitaria in atto. Le misure restrittive e le limitazioni ai diritti fondamentali, stabilite dai governi per contrastare la diffusione della pandemia, hanno alterato profondamente le abitudini di vita delle persone, impedendo, per un certo periodo, qualsiasi contatto con i familiari e gli amici. In questo panorama è stata così riscoperta l'utilità di usare le tecnologie e ne sono emersi i molteplici vantaggi: per molti, infatti, da strumento di alienazione è diventato un vero e proprio alleato per migliorare il benessere personale e combattere questa emergenza. Questi dati sono stati confermati dall'incremento di iniziative digitali e dall'aumento di installazioni di applicazioni di vario genere. Per quanto riguarda le aziende, indipendentemente dal settore di appartenenza, molte di esse sono state costrette ad avviare un programma di trasformazione digitale che ha previsto l'integrazione di nuove tecnologie.

Nell'ambito di questo percorso di digital transformation si colloca questo progetto di stage che mira a realizzare un prototipo di applicazione che abiliti la collaborazione remota tra operatori addetti alla manutenzione di prodotti. Lo scopo è consentire all'operatore di osservare in modalità "see-what-I-see" quello che vede il suo collega sul campo, in modo tale da poterlo guidare fornendogli istruzioni. Il lavoro svolto comprende sia la progettazione sia lo sviluppo di un software cloud di condivisione video e audio per l'interazione remota.

# Indice

# Elenco delle figure

# Elenco delle tabelle

# Capitolo 1

# Introduzione

Negli anni '60 Marshall McLuhan definì il fenomeno del "villaggio globale", in cui le nuove tecnologie avrebbero ricoperto un ruolo centrale nelle vite delle persone, diventando parte di esse. Lo studioso immaginò un mondo dove il passaggio di informazioni fosse istantaneo e i confini nulli. Le sue convinzioni sono state confermate dall'evolversi degli eventi e spesso si parla di una vera e propria rivoluzione informatica, vista la potenza e la repentinità dei progressi verificatesi, che hanno cambiato profondamente le società mondiali.

In questo nuovo mondo globalizzato risulta perciò fondamentale innovarsi, al fine di diventare sempre più competitivi sul mercato. Questa necessità è emersa in maniera decisiva in questi ultimi mesi, a causa dell'emergenza sanitaria in atto: molte aziende pubbliche e private, infatti, sono state costrette ad innovarsi e a modificare i propri sistemi organizzativi. Diverse imprese hanno così attivato lo smart working, o lavoro agile, e hanno incrementato le forme di assistenza tecnica da remoto viste le difficoltà legate al raggiungimento di alcuni paesi.

L'ideazione del progetto, svolto presso l'azienda Innovative Digital eXperience S.r.l., nasce proprio dall'esigenza di comunicare con i propri clienti per supportarli, superando limiti territoriali e ottimizzando le tempistiche. L'obiettivo dell'impresa è proprio quello di non negare la possibilità, quando questa fosse necessaria, di fornire un'assistenza completa ad un cliente. Pertanto, l'attività di stage ha riguardato la creazione di un'applicazione che permettesse la condivisione audio e video tra un cliente e un operatore e fornisse la possibilità di interagire virtualmente anche attraverso l'invio di messaggi e l'utilizzo di una lavagna virtuale.

## 1.1 Struttura del documento

Il presente documento è stato suddiviso in quattro parti che illustrano il percorso di stage. Si riporta, di seguito, l'organizzazione del testo.

Nella prima parte (Introduzione e Descrizione dello stage) si mira a presentare il progetto di stage, il contesto applicativo, a definire gli obiettivi e ad illustrare i prodotti attesi. Infine, si proporrà la pianificazione del lavoro.

Nella seconda parte (Strumenti e tecnologie) si intendono approfondire le tecnologie e gli strumenti, tra cui i linguaggi di programmazione, utilizzati durante il percorso di stage.

Nella terza parte (Realizzazione del progetto) si mira a descrivere il processo implementativo dell'applicazione, le scelte intraprese e i problemi riscontrati nella fase di sviluppo. Successivamente, si illustreranno le soluzioni individuate.

Nella quarta parte (Conclusioni) si intende riflettere sui risultati ottenuti riguardanti sia gli obiettivi specifici del progetto sia quelli formativi dello studente.

# Capitolo 2

# Descrizione dello stage

## 2.1 L'azienda

Innovative Digital eXperience S.r.l. è un'azienda italiana, nata il 24 marzo 2014 con sede legale in via Flavia 23/1 a Trieste e sede operativa in via dei Ronchi 21 a Padova.


**Figura 2.1:** Logo Innovative Digital eXperience S.r.l.

La missione dell'azienda si fonda sull'idea che "c'è un vero progresso solo quando i vantaggi di una nuova tecnologia sono accessibili a tutti"; da qui il desiderio e l'obiettivo del gruppo di supportare le aziende che intendono intraprendere un percorso di digital trasformation, sostenendole nelle varie fasi progettuali (dalla definizione dei requisiti all'implementazione di sistemi che integrano nuove tecnologie dirompenti) e aiutandole a comprendere i benefici e i vantaggi della quarta rivoluzione industriale.

### 2.1.1 Dominio applicativo

IDX è un'azienda che agisce in un dominio centrato principalmente nell'industria 4.0 e sviluppa soluzioni software multipiattaforma in ottica Smart Manufacturing e Sales Promotion. L'impresa è una start up innovativa, con un gruppo giovane e dinamico, ideale per seguire le aziende manifatturiere con esigenze di vendita, rivendita e assistenza. L'organizzazione ricerca e realizza la soluzione ideale in base ai bisogni delle imprese, integrando realtà aumentata, wearable device e IoT e sviluppando soluzioni software multipiattaforma per servizi, vendite e marketing.

### 2.1.2 Strumenti e tecnologie aziendali

Per supportare lo sviluppo dei progetti approvati, IDX utilizza alcuni strumenti utili per gestire le proprie attività e il lavoro di team. Prima di iniziare l'esperienza di tirocinio nell'azienda, il personale ha pertanto provveduto ad attivare un'e-mail con dominio aziendale, grazie alla quale è stato possibile creare un account idoneo per accedere agli strumenti di supporto disponibili. Di seguito, vengono descritti i software che sono stati messi a disposizione dall'azienda per la durata complessiva del tirocinio.

#### Git


**Figura 2.2:** Logo Git

Git è un software per il controllo di versione, creato da Linus Torvalds nel 2005. Questo strumento permette di gestire il codice del proprio progetto, lasciando traccia delle proprie modifiche eseguite nel tempo. Il vantaggio del suo utilizzo è quello di poter lavorare in team sullo stesso progetto, offrendo la possibilità di creare più ramificazioni, di unirle e, in seguito, di gestirne i conflitti. Per utilizzare

Git, da riga di comando su sistema operativo Windows, è necessario il supporto di Git bash. Oltre che dal terminale, sono disponibili diversi tool che permettono di gestire il sistema di Git graficamente, secondo una struttura ad albero composta da nodi univoci.

### GitHub

**Figura 2.3:** Logo GitHub

GitHub è un'azienda che offre servizio di hosting per repository Git. Questo strumento permette agli sviluppatori di gestire in maniera efficace il sistema di Git tra più utenti facilitandone l'utilizzo. Inoltre, GitHub offre molte funzionalità per gestire il repository cloud direttamente dalla piattaforma, grazie ad un'interfaccia molto intuibile. GitHub viene spesso utilizzato anche per condividere progetti con persone non necessariamente del proprio gruppo di lavoro e consente di personalizzare la visibilità del repository, selezionando se pubblica o privata e scegliendo i soggetti con cui condividerla.

### Jira


**Figura 2.4:** Logo Jira

Jira è un software creato dal gruppo Atlassian, studiato per aiutare il team a gestire e svolgere il lavoro. Inizialmente questo prodotto svolgeva per lo più un compito di monitoraggio e gestione delle attività sotto forma di ticket, mentre attualmente permette ad ogni membro del gruppo di lavoro di collaborare in modo efficace. Jira è consapevole che ogni team ha un proprio percorso specifico per creare un software, perciò offre anche la possibilità di scegliere se usare un flusso di lavoro predefinito oppure di crearne uno corrispondente al proprio modo di lavorare.

### Confluence


**Figura 2.5:** Logo Confluence

Per quanto riguarda la documentazione del lavoro svolto, è stato utilizzato Confluence, un software appartenente al gruppo Atlassian. Questo strumento fornisce la possibilità di avere uno spazio di lavoro in remoto, accessibile solo dopo autenticazione, dove creare, modificare e avviare discussioni sul proprio lavoro, condividere e reperire informazioni tra i team e gestire e organizzare i contenuti del lavoro svolto.

Tutta la documentazione riguardante il progetto è stata scritta mediante Word e condivisa con il gruppo aziendale su Confluence.

### Slack

**Figura 2.6:** Logo Slack

Slack è un software di messaggistica ideale per la comunicazione all'interno di un'azienda. Questo strumento permette di creare un dominio ristretto di collaboratori sul quale è possibile comunicare in diversi modi. Consente, infatti, di creare canali dedicati a settori specifici con accesso libero oppure privato, nel caso sia dedicato ad un gruppo ristretto di persone. Fornisce, inoltre, la possibilità di comunicare via chat sia privatamente sia con più membri contemporaneamente. Questo software è scaricabile sia nella versione desktop che per altri dispositivi, come per esempio mobile Android.

### Zoom

**Figura 2.7:** Logo Zoom

Zoom è una nota applicazione che fornisce un servizio di videoconferenza basato su cloud. Permette di creare una sessione, chiamata room, dov'è possibile avviare delle riunioni tra i partecipanti e consente di organizzare videochiamate sia tramite video sia utilizzando solo audio. Questa applicazione offre servizi di messaggio via chat privata tra due utenti oppure condivisa con il gruppo e, inoltre, permette di condividere lo schermo per presentare il desktop del proprio computer oppure delle singole schede. Durante il percorso di tirocinio, soprattutto quand'erano in vigore le misure restrittive stabilite a causa dell'emergenza sanitaria, Zoom è stato utilizzato più volte nel corso delle varie settimane per confrontarsi e condividere le informazioni riguardanti lo stato di avanzamento del progetto e lo sviluppo delle funzionalità.

## 2.2 Il progetto

Lo scopo del progetto di stage consiste nell'analisi e nell'implementazione prototipale di un software di condivisione audio/video per la collaborazione remota, che integri elementi in realtà aumentata.

### 2.2.1 Contesto applicativo

Nel panorama attuale molte imprese hanno intrapreso o intendono iniziare un percorso di trasformazione digitale, attraverso l'implementazione di sistemi che utilizzano nuove tecnologie dirompenti. La diffusione di questi strumenti digitali comporta però una sempre maggiore richiesta di supporto e assistenza per la gestione e la manutenzione degli stessi, che spesso si scontra con limiti territoriali e temporali. Pertanto, per poter migliorare l'efficienza, aumentare la produttività e al contempo abbassare i costi e i tempi di manutenzione, risulta sempre più necessario possedere degli strumenti che consentano ai tecnici di fornire assistenza da remoto ai clienti indipendentemente dai luoghi in cui sono ubicate le aziende richiedenti supporto.

Alla luce di ciò, si evince come il presente progetto, che mira a garantire alle aziende un supporto tecnico su tecnologie e macchinari, risponda a questa esigenza sempre più sentita e diffusa. Il lavoro svolto, infatti, intende stabilire una connessione stabile e sicura a qualsiasi distanza con la possibilità di condividere audio e video tra due client. Pertanto, come riportato precedentemente, l'assistenza da remoto risulta vantaggiosa, in quanto abbatte le distanze e quindi riduce i costi in termini di tempo e di denaro, poiché consente allo stesso tecnico di seguire più aziende,

ubicate in luoghi lontani fra loro, senza la necessità di raggiungerle fisicamente.

È stato inoltre deciso di implementare aspetti del mondo front-end, tra cui la manipolazione del DOM (Domain Object Model), la gestione di eventi e la comunicazione attraverso servizi http con il back-end per rendere la grafica accattivante e offrire così un'esperienza gratificante agli utenti.

### 2.2.2 Fasi di progettazione

Si riportano di seguito le fasi di progettazione e sviluppo dell'applicazione:

- Individuazione di una tecnologia open source per la comunicazione remota tra due o più persone;
- Instaurazione di una connessione stabile nel tempo e apertura di un canale di comunicazione che offre la possibilità, alle persone connesse, di interagire e scambiarsi dati;
- Scelta dell'architettura progettuale più adatta alle esigenze del prodotto;
- Avvio della collaborazione remota attraverso la piattaforma desktop, fornendo la possibilità di renderla in seguito responsive, ovvero adattabile a diverse risoluzioni dello schermo ed a diversi dispositivi;
- Implementazione dell'interfaccia grafica.

### 2.2.3 Obiettivi

All'inizio dello stage sono stati scelti degli obiettivi formativi con il tutor interno dell'azienda. Per definirli si farà riferimento alle seguenti notazioni:

**O**: per i requisiti obbligatori, vincolanti in quanto obiettivo primario richiesto dal committente;

**D**: per i requisiti desiderabili, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto;

**F**: per i requisiti facoltativi, rappresentanti valore aggiunto non strettamente competitivo.

Le sigle precedentemente indicate saranno seguite da una coppia sequenziale di numeri, identificativo del requisito. Gli obiettivi concordati sono i seguenti:

**Obbligatori**

- O01: studio delle tecnologie disponibili;
- O02: individuazione delle tecnologie adatte al contesto applicativo individuato;
- O03: implementazione di un prototipo per una delle tecnologie individuate;

**Facoltativi**

- F01: implementazione di un secondo prototipo per un'altra delle tecnologie individuate.

### 2.2.4 Contenuti formativi

Prima di iniziare l'attività di stage sono state definite le conoscenze da approfondire nell'ambito dello sviluppo di applicazioni principalmente lato client, con la possibilità di avvicinarsi a tematiche legate alla realtà aumentata. Nello specifico è stata prefissa l'acquisizione delle conoscenze che trattano i seguenti argomenti:

- analisi e progettazione;
- back-end cloud;
- progressive web app o mobile app;
- realtà aumentata;
- streaming audio/video.

## 2.3 Prodotti attesi

Al termine del periodo di tirocinio i prodotti attesi dall'azienda sono i seguenti:

- una relazione scritta sulle tecnologie analizzate con pregi e difetti di ogni piattaforma e le motivazioni della scelta di una di esse;
- un file compresso con estensione a scelta contenente i codici sorgenti dell'applicazione funzionante e una copia della relazione prodotta.

## 2.4 Pianificazione del lavoro

Prima di iniziare l'esperienza in azienda sono state concordate le ore lavorative con una suddivisione delle attività da svolgere. Nel piano di lavoro concordato è stata inclusa la seguente pianificazione settimanale:

**Prima Settimana (40 ore)**:

- incontro con persone coinvolte nel progetto per discutere i requisiti e le richieste relativamente al sistema da sviluppare;
- verifica credenziali e strumenti di lavoro assegnati;
- presa visione dell'infrastruttura esistente;
- formazione sulle tecnologie adottate;
- studio delle tecnologie disponibili;

**Seconda Settimana (40 ore)**:

- studio delle tecnologie disponibili;
- individuazione delle tecnologie più adatte al contesto applicativo;

**Terza Settimana (40 ore)**:

- progettazione e implementazione del prototipo;

**Quarta Settimana (40 ore)**:

- progettazione e implementazione del prototipo;

**Quinta Settimana (40 ore)**:

- progettazione e implementazione del prototipo;

**Sesta Settimana (40 ore)**:

- progettazione e implementazione del prototipo;

**Settima Settimana (40 ore)**:

- progettazione e implementazione del prototipo;
- stesura e formalizzazione della documentazione finale del progetto;

**Ottava Settimana (32 ore)**:

- stesura e formalizzazione della documentazione finale del progetto.

# Capitolo 3

# Strumenti e tecnologie

## 3.1 Html


**Figura 3.1:** Logo HTML5

Per la realizzazione del front-end di questo progetto sono stati utilizzati diversi strumenti, tra cui Html5. L'Hyper Text Markup Language, da cui l'acronimo Html, è un linguaggio di markup. Questo strumento viene utilizzato per creare la struttura logica della pagina Web e la sua rappresentazione. Le pagine Web che si trovano nel browser sono scritte in Html combinate con altri linguaggi per renderle dinamiche. Quando si chiede di mostrare una pagina web scrivendo l'Url su un client, come ad esempio il browser, viene effettuata una chiamata ad un Web server. Il compito di quest'ultimo è quello di fornire al client la pagina richiesta specificata nell'Url. Quindi il Web server passa il codice Html al client che si occupa di fare il rendering della pagina. Questo è il motivo per cui l'utente riesce a vedere una pagina web comprensibile.

## 3.2 CSS

**Figura 3.2:** Logo CSS3

Per gestire lo stile della pagina è stato adoperato il noto Cascading Style Sheets, da cui la sigla CSS. Questo linguaggio viene utilizzato per creare degli stili e personalizzare così la pagina Web a proprio piacere. Questo strumento permette di togliere le responsabilità estetiche alla pagina Html, lasciando ad essa il compito di gestire principalmente la struttura della pagina. Per lo stile invece lo strumento più potente e adatto per esprimerlo è il CSS, che è stato creato appunto per questo scopo. Separare la struttura dallo stile non è un problema grazie a questo strumento in quanto, attraverso delle classi oppure degli identificativi, riesce a individuare elementi Html ben precisi. Per cui, quando viene specificato un ID nel foglio di stile, si è certi che il CSS applica le istruzioni del blocco solo a quel preciso componente. Nell'applicazione, sviluppata durante il percorso di stage, è stato necessario creare un unico foglio di stile nominato style.css e successivamente includerne il percorso nel file Html.

## 3.3 Javascript


**Figura 3.3:** Logo Javascript

Per creare la pagina di visualizzazione lato client, oltre ad Html ed al CSS, è stato utilizzato il linguaggio di programmazione Javascript. Questo linguaggio appartiene alla EMAC dal 1997 e ad oggi è uno dei più diffusi al mondo. Javascript è un linguaggio, inizialmente nato lato web, utilizzato nella realizzazione front-end delle applicazioni. Una delle sue caratteristiche è quella di rendere una pagina web dinamica, permettendo di gestirne gli eventi e di manipolarne i componenti Html. Non a caso al giorno d'oggi alcuni dei più famosi framework come ReactJs, creata da Facebook, sono scritti su base Javascript. Un altro punto di forza del linguaggio è la sua familiarità sintattica con i principali linguaggi di programmazione, come Java e C++. Javascript supporta la programmazione orientata agli oggetti ed è stato utilizzato anche per la parte back-end del progetto, al fine di avere un unico linguaggio principale.

## 3.4 JQuery


**Figura 3.4:** Logo JQuery

Per questa applicazione è stato adoperato anche JQuery, una delle librerie più utilizzate di Javascript. Anch'esso, come il linguaggio descritto precedentemente, è lato client in quanto permette di modellare il DOM del browser, ovvero il Domain Object Model, rendendo dinamica la pagina web. Questa libreria rende possibile all'utente l'interazione con la pagina web, visualizzata nel browser, attraverso funzioni ed eventi. In merito all'applicazione, oggetto del progetto di stage, l'utilizzo di JQuery ha reso il relativo codice più leggibile, riducendo così il numero di righe e ha permesso di gestirne il codice CSS con semplici funzionalità messe a disposizione dalla relativa libreria, evitando di creare funzioni superflue e ridondanti.

## 3.5 Node.js


**Figura 3.5:** Logo Nodejs

Per il back-end dell'applicazione è stato utilizzato Node.js. Questo strumento non è altro che un runtime di Javascript costruito su V8 Chrome, ovvero l'inter-

prete Javascript del noto browser Chrome, da cui il nome. Quindi Node.js nasce dall'estrapolazione del motore dell'interprete di Javascript V8 Chrome, che è stato poi portato lato back-end per le operazioni usuali di qualsiasi altro linguaggio di programmazione. Javascript viene spesso considerato solo come un linguaggio lato web, senza conoscerne veramente le potenzialità. Grazie a Node.js è possibile utilizzarlo come unico linguaggio di programmazione sia per il lato back-end sia per il front-end dell'applicazione. Nel progetto di stage è stato utilizzato per gestire il componente server dell'applicazione, importando una libreria di Openvidu scritta in Javascript.

## 3.6 WebRTC


**Figura 3.6:** Logo Web RTC

La libreria scelta per l'applicazione si basa su un progetto che sfrutta WebRTC, ovvero Web Real Time Communication. Dalla definizione dell'acronimo, è possibile intuire che WebRTC è una tecnologia che consente di comunicare in tempo reale. Essa è utilizzata per rendere possibile la comunicazione peer-to-peer, permettendo a due client, come ad esempio due browser, di comunicare direttamente tra loro. Un punto di forza di questa tecnologia è l'essere open source e supportata dai principali browser. Una volta, prima della diffusione di questo sistema per comunicare con un client, era necessario connettersi ad un server ed utilizzarlo come tramite per passare il flusso di dati. Dall'altro lato il secondo client veniva interrogato dal server e, in seguito, rispondeva allo stesso server.

Questo passaggio indiretto di dati, utilizzando un server intermediario, creava un ritardo di latenza provocando problemi nelle comunicazioni audio, video e nelle condivisioni dello schermo.

Grazie a WebRTC viene usato un server intermediario solo per stabilire la connessione tra due client. Inizialmente il client invia i suoi dati al server con delle informazioni riguardanti la sua posizione e il nodo da raggiungere. Quest'ultimo interpreta i dati ricevuti e contatta il secondo client mandando le informazioni della richiesta. Il client può accettare la richiesta ed in caso positivo rispondere al server con le sue informazioni. Ricevuti i dati, il server li inoltra al primo client che potrà così comunicare direttamente con il secondo client, senza più necessitare di un intermediario.

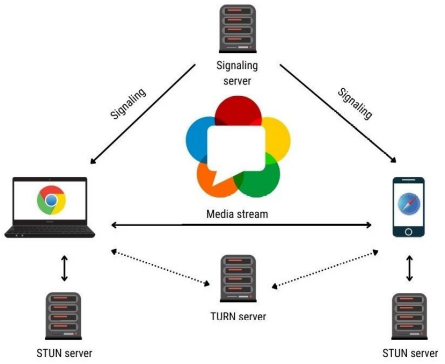


**Figura 3.7:** Funzionamento processo WebRTC

Un altro aspetto importante di questa tecnologia è la sicurezza. Per trasferire in tempo reale i dati vengono prima crittati utilizzando il metodo Datagram Transport Layer Security. Questa funzionalità è integrata nativamente da ogni browser che supporta WebRTC. Inoltre, viene utilizzato anche il Secure Real Time Protocol per crittare audio e video in modo tale che non possano essere visti né sentiti.

## 3.7 Kurento


**Figura 3.8:** Logo Kurento

Kurento è un framework open source che espone le funzionalità, per mezzo di un API, di un media server chiamato Kurento Media Server, dal suo acronimo KMS. Questo framework viene utilizzato nel progetto indirettamente, in quanto esso è rielaborato da OpenVidu, che è la libreria effettivamente usata. Kurento è utilizzato per sviluppare applicazioni audio e video avanzate per piattaforme compatibili con WebRTC.

Per poter far condividere tra più utenti dei media stream, che consistono nel flusso di dati da un client all'altro, non è più sufficiente WebRTC, che funziona peer-to-peer, ma viene utilizzato il media server di Kurento. Per stabilire una connessione è necessario un primo passaggio per far conoscere le parti coinvolte. Quindi viene emesso un web socket, che è un pacchetto con dati relativi al chiamante, ad un apposito server. In questa fase viene ottenuto il client di Kurento e viene creata una media pipeline, la quale permette e stabilisce il flusso dati dei media. Oltre alla pipeline viene creato anche un WebRTC endpoint per l'uscita dello stream dati. Terminati questi passaggi viene creato l'user che resta in ascolto per connessioni WebRTC. Il client quando riceve il segnale di una nuova connessione, grazie alla configurazione WebRTC precedente, può creare l'elemento video, solitamente un tag Html, per mostrare il flusso di dati ricevuto dal client remoto. Ad oggi, nel sito ufficiale Kurento viene segnalato come deprecato in quanto è stato preso in carico da OpenVidu.

## 3.8 OpenVidu


**Figura 3.9:** Logo OpenVidu

Openvidu è una libreria open source ed è il cuore centrale dell'applicazione del progetto di stage. Essa consiste in una piattaforma che permette di aggiungere e

gestire videochiamate nella propria applicazione web. Il suo scopo è di permettere la comunicazione real-time tra client incorporando una serie di funzionalità per facilitarne l'uso. Questa tecnologia è stata creata sopra il framework Kurento. La differenza è che Openvidu aggiunge ulteriori funzionalità a quelle di Kurento, creando un sistema che nasconde tutte quelle operazioni a basso livello di cui il programmatore non deve più preoccuparsene.

Openvidu mette a disposizione due componenti importanti: il file openVidu-server.js ed il file openVidu-browser. Il primo elemento crea tutte le funzionalità necessarie per gestire le operazioni di openVidu-browser lato server, mentre il secondo è una libreria che consente di gestire lato client tutte le funzionalità di una videochiamata e di gestirne l'approccio con gli utenti. Il programmatore è libero di sviluppare la logica della vista utilizzando un framework, come ad esempio Angular oppure Javascript. Per fare ciò è sufficiente creare un oggetto Openvidu che dispone di metodi utili a gestire sessioni, user, altre funzionalità come per esempio configurare lo stream video con dei parametri. OpenVidu segue l'architettura publish/subscribe. Questo design pattern permette la comunicazione tra componenti attraverso un sistema di segnali con un intermediario, chiamato dispatcher.


**Figura 3.10:** Design Publish and Subscribe

Con questo sistema un publisher, cioè il componente che emette il segnale, non deve preoccuparsi dell'identità dei riceventi. A sua volta i destinatari eseguono una funzione di subscribe per segnalare al dispatcher che restano in ascolto di un determinato tipo di segnale. In questo modo è possibile gestire e personalizzare il comportamento degli eventi che si possono manifestare.

## 3.9 Visual Studio Code

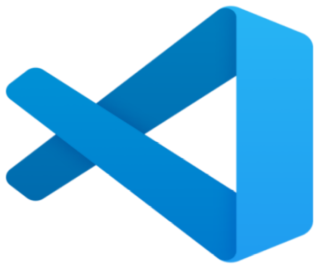

**Figura 3.11:** Logo Visual Studio Code

Visual Studio Code è un editor di codice sorgente compatibile con i principali sistemi operativi, tra i quali Windows, MacOS e Linux. Questo strumento viene fornito con un supporto integrato per Javascript, Typescript e Node.js. Oltre a questo, offre un vasto ecosistema di estensioni per altri linguaggi di programmazione. VSC è ben strutturato ed è molto intuibile, offrendo numerose funzionalità per la realizzazione di progetti anche complessi. Una volta avviato l'editor è disponibile una sessione dedicata a guide e tutorial per un apprendimento veloce, oltre ad avere pieno supporto dalla documentazione del sito ufficiale.

# Capitolo 4

# Realizzazione del progetto

## 4.1 Implementazione

L'implementazione del software è la parte che ho trovato più entusiasmante e competitiva come programmatore, dove ogni sfida diventa un'occasione per accrescere le proprie competenze.

### 4.1.1 Inserimento nel team di sviluppo

La prima giornata all'interno dell'azienda ha riguardato l'inserimento nel team di sviluppo, composto da un responsabile, un software developer e due tirocinanti. Il primo ha svolto il ruolo di project manager, occupandosi principalmente di supervisionare il processo di sviluppo del progetto, organizzandolo e scandendone i tempi di implementazione, e di partizionarne le attività tra i componenti del gruppo. Il software developer ha fornito supporto durante le attività, consigliando per esempio delle strategie per superare le difficoltà, dovute anche all'inesperienza dei tirocinanti. Infine, i due tirocinanti si sono occupati dell'analisi del progetto e dell'implementazione del software.

### 4.1.2 Organizzazione delle attività

Nella prima fase il responsabile del progetto ha redatto un'analisi dei requisiti, contenente le funzionalità chiave del prototipo finale, e ha creato un template che fungesse da linea guida per l'elaborazione dell'analisi tecnica, sviluppata in seguito dai tirocinanti. Il risultato è stato un documento che spiegava nel dettaglio com'era costruito il prodotto dal punto di vista tecnologico e del software.

Successivamente sono state assegnate le attività da svolgere attraverso il software Jira, che veniva utilizzato dal contesto aziendale per gestire e monitorare l'implementazione dei progetti. Il flusso di sviluppo di questi veniva frammentato in attività che spesso venivano intercambiate con il termine di issue. Nel software Jira il responsabile aveva i privilegi per generare e creare delle issue, identificate da un nome ed un codice univoco, che descrivevano lo scopo dell'attività da svolgere nella sezione dedicata alle note.

Una volta assegnate le issue, si è proceduto con l'attribuzione della scadenza temporale che consiste nell'individuazione di una data futura entro la quale si stima venga conclusa l'attività. Per indicare chi doveva svolgerla è stato sufficiente assegnare l'issue ad un membro appartenente al progetto, con conseguente notifica della nuova mansione da svolgere.

### 4.1.3 Individuazione delle tecnologie

In questa fase sono state studiate varie tecnologie per l'implementazione prototipale di un software di condivisione audio/video per la collaborazione remota che integri elementi in realtà aumentata. Successivamente, con l'aiuto del responsabile, sono state individuate le tecnologie essenziali per creare le basi sulle quali il progetto è stato sviluppato.

La prima scelta è stata WebRTC, ovvero Web Real Time Communication, che consiste in una tecnologia che consente a due client di comunicare fra loro in tempo reale.

Grazie allo studio è stato compreso che la libreria più adatta per supportare il progetto era Openvidu, una piattaforma che permette di aggiungere e gestire video chiamate nella propria applicazione web. Essa metteva a disposizione del codice a scopo illustrativo e dimostrativo sia server side che client side.

Il linguaggio di sviluppo scelto per il progetto è Javascript ed è stato utilizzato come linguaggio sia client side sia server side. Tale scelta è risultata vantaggiosa in quanto ha richiesto al programmatore la conoscenza di un unico linguaggio di programmazione e, di conseguenza, è stato così dimezzato il tempo dedicato allo studio e alla formazione, che è stato possibile investire nell'implementazione dell'interfaccia grafica, rendendola maggiormente accattivante.

### 4.1.4 Predisposizione dell'ambiente di lavoro

Stabilite le principali tecnologie e concordati con il team gli strumenti da utilizzare, è stato definito e predisposto l'ambiente di sviluppo.

Nell'analisi dei requisiti era stato specificato di utilizzare una macchina con sistema operativo Ubuntu. Pertanto, è stata scelta la versione Ubuntu 18.04 LTS Bionic Beaver e su questo sistema operativo è stato installato Openvidu.

La prima fase ha riguardato l'installazione di Kurento Media Server (acronimo KMS), che è l'elemento centrale di Kurento, in quanto è responsabile della trasmissione, dell'elaborazione, del caricamento e della registrazione dei media. Il suo utilizzo è stato indispensabile poiché Openvidu è una libreria implementata su KMS e, quindi, usufruisce già in partenza delle sue funzionalità.

In seguito, è stato installato Coturn, il cui uso è stato fondamentale perché consiste in un progetto open source che mette a disposizione un'implementazione dei server STUN e TURN. Oltre a Coturn, nella fase di configurazione è stato indispensabile installare anche Redis, ovvero un Database non relazionale disponibile nelle versioni ufficiali per sistemi Linux.

Per completare la preparazione dell'ambiente di lavoro è stato necessario inserire delle proprietà in un file di configurazione, che sono servite a dare specifiche indicazioni al sistema come, per esempio, il numero di porta e l'indirizzo IP pubblico.

Ogniqualvolta si desiderava utilizzare la libreria Openvidu, bisognava avviare i server Redis, Coturn e KMS, e successivamente il comando per eseguire il JAR del server di Openvidu.

L'editor scelto per lo sviluppo del software è stato Visual Studio Code. Per un corretto funzionamento del progetto è stato necessario installare alcune estensioni disponibili nella sezione extensions nell'editor. Inoltre, sono stati installati alcuni pacchetti reperibili da Npm, un package manager per Javascript.

Infine, su richiesta del capo progetto, è stata elaborata una guida, redatta direttamente su Confluence, che illustrava tutti i passaggi di configurazione dell'ambiente di sviluppo server side. Il documento è diventato parte integrante del manuale sviluppatore.

### 4.1.5 Struttura del progetto

Terminata la generazione del progetto con tutte le sue dipendenze, è stato strutturato nella parte backend e in quella frontend.

Il backend comprende il software che si occupa di gestire tutte le operazioni del server che vengono incluse nel file server.js.

Il frontend, invece, comprende il software dedicato alla parte web e a tutto ciò che riguarda la Graphic User Interface (acronimo GUI). Sono stati importati così quattro file principali:

- app.js: contiene le funzioni Javascript;

- index.html: contiene la struttura della pagina web;
- style.css: contiene tutte le classi per lo stile della pagina web;
- openvidu-browser-2.15.2.js: è una libreria Openvidu per il browser.

### 4.1.6 Architettura software

L'architettura del software è un sistema REST che sfrutta le chiamate HTTP per ottenere ed interagire con le risorse. Il protocollo HTTP consente l'utilizzo di numerosi metodi, tra cui i più comuni si suddividono in GET, POST, PUT e DELETE. Nel prototipo viene eseguita solamente la chiamata GET.

### 4.1.7 Gestione dell'utente e della sessione

Openvidu è una libreria che presenta un'unità centrale che consente a più utenti di interagire attraverso una connessione instaurata in una sessione, ovvero un'unità che svolge il ruolo di stanza dove gli utenti possono comunicare e condividere stream audio e video al suo interno.

Si riporta di seguito la descrizione di alcune funzionalità di Openvidu con le implementazioni effettuate.

Una volta raggiunta la pagina iniziale di questo software, è innanzitutto necessario autenticarsi inserendo un nome utente e la password, mediante la maschera in figura 4.1.


**Figura 4.1:** Maschera di richiesta autenticazione

Cliccando sul bottone nominato Join viene chiamata una funzione Javascript che esegue il primo metodo post come viene illustrato nella figura 4.2 del file app.js. Nel file server.js è implementato il metodo post con lo stesso url della chiamata http precedente figura 4.3.

```
function logIn() {
    var user = $("#user").val(); // Username
    var pass = $("#pass").val(); // Password

    httpPostRequest(
        'AppApi-login/login',
        {user: user, pass: pass},
        'LogIn WRONG',
        (response) => {
            console.log("response value",response);
            $("#name-user").text(user);
            //joinSession();
            $("#not-logged").hide();
            $("#logged").show();
        }
    )
}
```

**Figura 4.2:** Funzione di logIn() lato client

```
// Login
app.post('/AppApi-login/login', function (req, res) {

    // Retrieve params from POST body
    var user = req.body.user;
    var pass = req.body.pass;
    console.log("Logging in | {user, pass}={" + user + ", " + pass + "}");

    if (login(user, pass)) { // Correct user-pass
        // Validate session and return OK
        // Value stored in req.session allows us to identify the user in future requests
        console.log("'" + user + "' has logged in");
        req.session.loggedUser = user;
        res.status(200).send();
    } else { // Wrong user-pass
        // Invalidate session and return error
        console.log("'" + user + "' invalid credentials");
        req.session.destroy();
        res.status(401).send('User/Pass incorrect');
    }
});
```

**Figura 4.3:** Funzione di Login lato server

Il controllo, eseguito nel server, avviene mediante una funzione che verifica l'esistenza delle credenziali inserite in un mock, che consiste in un oggetto simulato che riproduce il comportamento di una tabella salvata nel database. Il valore ritornato è un boolean che equivale a true, se riscontra una corrispondenza nei dati salvati, altrimenti viene ritornato il valore a false. Quando il controllo va a buon fine, nella response della funzione di login, si mostra la seconda maschera all'utente come in figura 4.4.

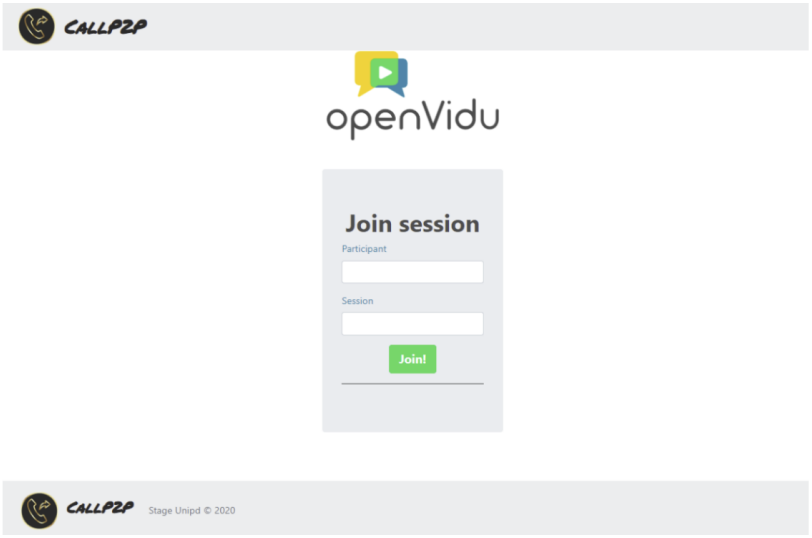


**Figura 4.4:** Maschera di richiesta partecipazione

La nuova maschera permette all'utente di scegliere la sessione dove collegarsi, con un opportuno nickname con cui si desidera essere identificati. Per proseguire si deve cliccare sul bottone Join, così da provocare l'evento della funzione per richiedere un token. Di seguito, nella figura 4.5, si illustra la funzione del file app.js che contiene la logica sopra descritta, mentre si riporta nella figura 4.6 l'implementazione della funzione del file server.js.

```
function joinSession() {
    getToken((token) => {

        // --- 1) Get an OpenVidu object ---
        OV = new OpenVidu();
        // --- 2) Init a session ---
        session = OV.initSession();
        // --- 3) Specify the actions when events take place in the session on every new Stream received...
        session.on('streamCreated', (event) => {
            // Subscribe to the Stream to receive it
            if ($('#btn-vid-stream').val() === "true") setUnpublishStream();
            subscriber = session.subscribe(event.stream, 'vid-cnv-str', { insertMode: 'PREPEND' });
            appendUserData(subscriber.stream.connection);
        });
        // On every Stream destroyed...
        session.on('streamDestroyed', (event) => {
            // Delete the HTML element with the user's name and nickname
            removeUserData(event.stream.connection);
        });
        // --- 4) Connect to the session passing the retrieved token and some more data from the client (in this case a JSON with the nickname chosen by the user) ---
        var nickName = $("#nickName").val();
        session.connect(token, { clientData: nickName })
            .then(() => {
                $('#session-title').text(sessionName);
                $('#not-logged').hide();
                $('#logged').hide();
                $('#cnv-write').hide();
                $('#session').show();
                setDafaultSetting();
                var userName = $("#user").val();
                if (isPublisher(userName)) {
                    publisher.on('videoElementCreated', (event) => {
                        // Init the main video with ours and append our data
                        var userData = {
                            nickName: nickName,
                            userName: userName
                        };
                        appendUserData(userData);
                        $(event.element).prop('muted', true); // Mute local video
                    });
                } else {
                    console.warn('You don\'t have permissions to publish');
                    initMainVideoThumbnail();
                    switchEnablePropcrty('#btn vid strcam');
                }
            })
            .catch(error => {
                console.warn('There was an error connecting to the session:', error.code, error.message);
            });
```

**Figura 4.5:** Funzione joinSession() lato client

```
// Get token (add new user to session)
app.post('/AppApi-sessions/get-token', function (req, res) {
    if (!isLogged(req.session)) {
        req.session.destroy();
        res.status(401).send('User not logged');
    } else {
        // The video-call to connect
        var sessionName = req.body.sessionName;
        console.log("sessionName", sessionName)
        // Role associated to this user
        var role = users.find(u => (u.user === req.session.loggedUser)).role;
        console.log("role", role)
        // Optional data to be passed to other users when this user connects to the video-call
        // In this case, a JSON with the value we stored in the req.session object on login
        var serverData = JSON.stringify({ serverData: req.session.loggedUser });
        console.log("serverData", serverData)
        console.log("Getting a token | {sessionName}={" + sessionName + "}");
        // Build tokenOptions object with the serverData and the role
        var tokenOptions = {
            data: serverData,
            role: role
        };
        console.log("tokenOptions", tokenOptions)

        if (mapSessions[sessionName]) {
            // Session already exists
            console.log('Existing session ' + sessionName);
            // Get the existing Session from the collection
            var mySession = mapSessions[sessionName];
            // Generate a new token asynchronously with the recently created tokenOptions
            mySession.generateToken(tokenOptions)
                .then(token => {
                    // Store the new token in the collection of tokens
                    mapSessionNamesTokens[sessionName].push(token);
                    // Return the token to the client
                    res.status(200).send({
                        0: token
                    });
                })
                .catch(error => {
                    console.error(error);
                });
        } else {
            // New session
            /* RECORDING */
            console.log('New session ' + sessionName);
            // Create a new OpenVidu Session asynchronously
            OV.createSession(properties)
                .then(session => {
                    // Store the new Session in the collection of Sessions
                    mapSessions[sessionName] = session;
                    //console.log("mapSessions[sessionName]", mapSessions[sessionName])
                    // Store a new empty array in the collection of tokens
                    mapSessionNamesTokens[sessionName] = [];
                    //console.log("mapSessionNamesTokens[sessionName]", mapSessionNamesTokens[sessionName])
                    // Generate a new token asynchronously with the recently created tokenOptions
                    session.generateToken(tokenOptions)
                        .then(token => {
                            // Store the new token in the collection of tokens
                            mapSessionNamesTokens[sessionName].push(token);
                            //console.log("mapSessionNamesTokens[sessionName]", mapSessionNamesTokens[sessionName])
                            // Return the Token to the client
                            res.status(200).send({
                                0: token
                            });
                        })
                        .catch(error => {
                            console.error("Generate Token error: ", error);
                        });
                })
                .catch(error => {
                    console.error("Create sessione error: ", error);
                });
        }
    }
});
```

**Figura 4.6:** Funzione getToken() lato server

Durante il processo di sviluppo del software sono state tenute in considerazione due diverse opzioni:

- Richiesta dell'utente di entrare in una nuova sessione: in questo caso è stato necessario creare nel file server.js una nuova sessione e mapparla per il nome precedentemente inserito.
- Richiesta dell'utente di entrare in una sessione già esistente nel sistema: si è proceduto con l'individuazione della sessione precedentemente mappata e la

generazione di un token valido con cui l'utente potesse accedere.

### 4.1.8 Design pattern Publish/Subscribe

Congiuntamente al sistema REST descritto, Openvidu integra il design pattern Publish Subscribe. Quest'ultimo è un pattern di messaggistica dove sono protagoniste due entità:

- il Publisher, il quale rappresenta un elemento che pubblica un messaggio;
- il Subscriber, che svolge il ruolo di iscriversi ad un particolare messaggio.

Queste due entità non si conoscono a vicenda, quindi non sanno a chi è rivolto il messaggio o da chi lo riceveranno. Il ruolo del Publisher è quello di creare un messaggio per un argomento specifico, mentre quello del Subscriber è di iscriversi ad un messaggio di un argomento. Per poter comunicare tra loro viene utilizzata un'entità che funge da broker. Sia il Publisher che il Subscriber comunicano con il broker: il primo invia il messaggio al broker, il quale filtrerà ed invierà i messaggi ai Subscriber individuati. Questa funzionalità è fondamentale per permettere agli utenti all'interno della sessione di condividere lo stream audio/video e di scambiare messaggi testuali. Nella figura 4.7 si illustrano sia la funzione che invia un segnale contenente il messaggio, sia quella che ne gestisce l'evento. Chi è in ascolto per il tipo di segnale prende il messaggio e lo mostra a video nell'elemento html designato come target.

```
function sendChatMessage() {
    var mss = $("#senderMs").val().trim();             // Message in local.
    var nik = $("#nickName").val();                  // sender's Nickname.
    if (mss === "") {
        $("#senderMs").val() = ""
        return;
    }
        var date = formatAMPM(new Date());
        var message = "<li><strong> " + nik + "</strong> : " + mss +
            "<p><small>" + date + "</small></p>" + "</li>";     // complete string { nikName : "message's user"}

        session.signal({
            data: message,              // Any string (optional)
            to: [],                     // Array of Connection objects (optional. Broadcast to everyone if empty)
            type: 'my-chat'             // The type of message (optional)
        })
            .then(() => {
                // $("#sendMsButton").attr("disabled",true);
                console.log('Message sent');
            })
            .catch(error => {
                console.error(error);
            });
}
// CONNECTION FEATURES
session.on('signal:my-chat', (event) => {
    $("#senderMs").val("");
    $("#chat-text").append(event.data);
});
```

**Figura 4.7:** Sviluppo e gestione del messaggio

### 4.1.9 La GUI

L'interfaccia grafica è stata implementata cercando di rendere l'esperienza dell'utente il più gratificante possibile. Per questo motivo sono state aggiunte opzioni per gestire la finestra video, come bloccare il video oppure silenziare l'audio. Nell'analisi dei requisiti, uno dei casi d'uso opzionali prevede lo sviluppo di contenuti in Realtà aumentata. Per creare un'esperienza che garantisca valore aggiunto al prodotto è stata implementata una lavagna per disegnare su video. Come si vede nella figura 4.8, per guidare un utente, che ha richiesto supporto nello svolgere un lavoro nel quale ha riscontrato delle difficoltà, grazie all'utilizzo di questa applicazione, l'operatore ha la possibilità di disegnare sul video condiviso dall'utente, fornendogli così assistenza.

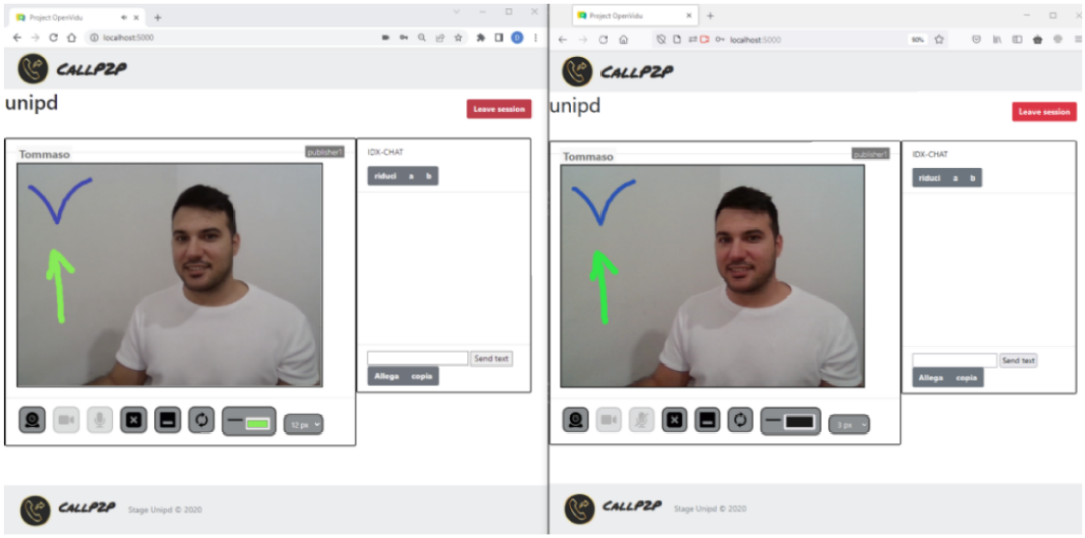


**Figura 4.8:** Disegno condiviso su stream video

Per creare questa funzionalità è stato scelto HTML Canvas con implementazione Javascript, che consente di disegnare condividendo il flusso e permettendo a tutti gli utenti connessi alla sessione di vedere in real-time le azioni svolte.

Oltre all'abilità di disegno, sono state implementate un set di funzioni per agevolare l'utente e rafforzare lo scopo del caso d'uso. Sono state così aggiunte alcune opzioni per il controllo del disegno come, per esempio, il colore e lo spessore del tratto della linea per disegnare oppure la possibilità di nascondere o cancellare il contenuto della canvas. In figura 4.9 è possibile visualizzare la toolbar e, in particolare, la realizzazione di tratti con colori e spessori diversi sul video.

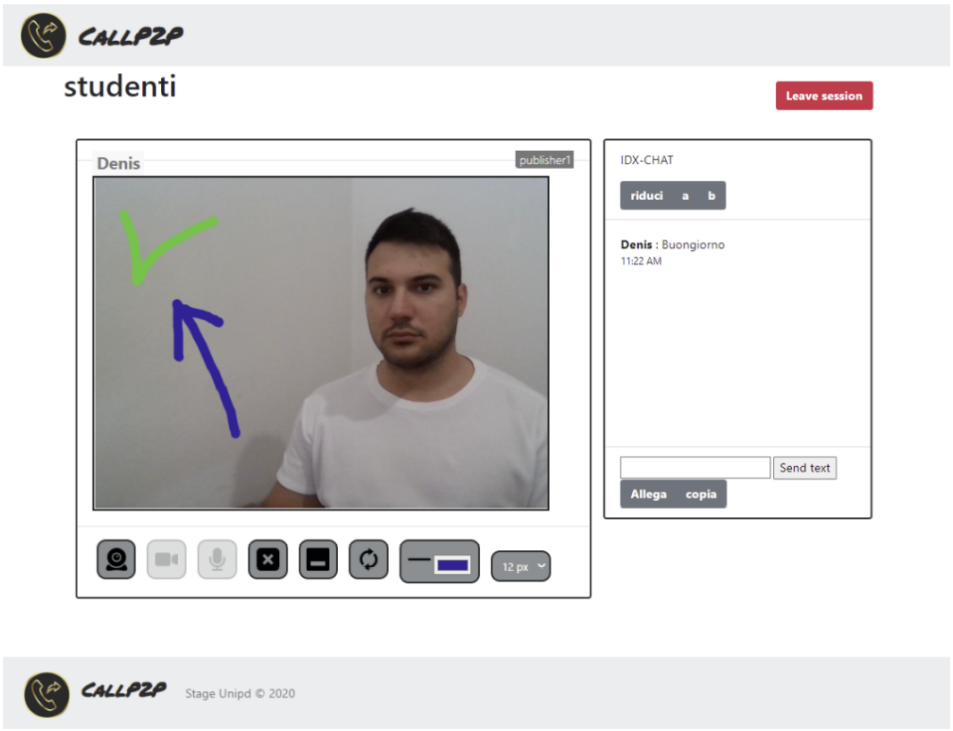


**Figura 4.9:** Canvas su video condiviso

## 4.2 Collaudo

### 4.2.1 Casi d'uso

Durante la stesura dell'analisi dei requisiti è necessario individuare i casi d'uso, ovvero una serie di interazioni tra un utente ed un sistema che permettono così all'utente di soddisfare un obiettivo o portare a termine un compito. I casi d'uso scelti con il responsabile del progetto sono stati i seguenti:

- Autenticazione effettuata tramite username e password;
- Richiesta di accesso alla sessione con scelta di nickname;
- Invio di messaggi testuali;
- Condivisione dell'audio e del video;
  - Disattivazione dell'audio;
  - Disattivazione del video;
- Condivisione di un tratto grafico in real-time;

  - Scelta dello spessore del tratto grafico;
  - Scelta del colore del tratto grafico;
  - Cancellazione del tratto grafico;
  - Possibilità di nascondere o mostrare il tratto; grafico realizzato;
  - Disattivazione del tratto grafico;

- Rimozione di un utente dalla sessione.

### 4.2.2 Preparazione del test case

Terminata l'implementazione del software, è stata effettuata una verifica dei casi d'uso del progetto a partire dalla predisposizione di un documento in formato excel su Confluence. In questo file è stata creata una tabella, dove ogni riga riguardava una funzionalità sviluppata da testare, mentre le colonne sono state così suddivise:

- id del test: è stato inserito un codice numerico che identificava il test;
- nome del test: è stato riportato l'identificativo testuale per fare riferimento al test;
- descrizione: è stato spiegato l'obiettivo della funzionalità e la previsione del risultato atteso;
- parametri: era un campo opzionale nel quale era possibile inserire i parametri necessari per eseguire il test;
- risultato: è stato inserito il risultato del test ottenuto dalla verifica.
- stato: è stato registrato lo stadio del test che si differenzia in pass, fail e pending;
- note: sono state aggiunte tutte le criticità riscontrate nella verifica del test.

### 4.2.3 Validità del software

Il processo conclusivo del progetto è consistito nel verificare il funzionamento dei casi d'uso descritti nell'analisi dei requisiti. L'obiettivo ha riguardato il controllo della qualità del prodotto e la conformità alle esigenze del committente del risultato atteso. I test svolti hanno mirato perciò a valutare il funzionamento logico del progetto, ma anche la solidità e la qualità del codice.

Alcune funzionalità del prodotto richiedevano che i test fossero eseguiti da due o più membri del team. Pertanto, la condivisione di audio e video all'interno dell'applicazione è stata testata connettendo fino a quattro dispositivi alla sessione. Ogni utente connesso ha provato tutte le funzionalità presenti in maschera, con l'accortezza di segnalare eventuali anomalie. Ogniqualvolta un test falliva, veniva registrata l'eccezione trovata e successivamente si ripeteva la verifica per tutti i test non passati.

Inizialmente i test riguardanti la richiesta di accesso alla sessione con scelta di nickname e quelli relativi alla condivisione di un tratto grafico in real-time sono falliti.

Nel primo caso è stato rilevato che non era possibile accedere alla sessione a causa del certificato SSL scaduto. Pertanto, è stato necessario installare un certificato SSL temporaneo gratuito chiamato Let's Encrypt.

Nel secondo caso è stato registrato che l'utente riusciva a tracciare correttamente il disegno, però non poteva condividere la sua illustrazione con gli altri utenti, i quali visualizzavano così solo il video. Per risolvere il problema è stato necessario condividere le coordinate del mouse tramite un segnale, che provocava agli altri utenti un evento che riproduceva il disegno realizzato.

Per arrivare alla versione finale del software è stato necessario ripetere due volte il test case.

### 4.2.4 Valutazione del prodotto

Raggiunta una versione stabile del software, è stata avviata la fase finale del progetto che ha riguardato il passaggio di consegna del prodotto. È stata così organizzata una riunione con tutte le parti interessate del progetto, nel corso della quale è stato mostrato il prototipo finale ed è stato verificato con il cliente che i requisiti richiesti fossero stati soddisfatti rispetto a quanto concordato. Terminato lo sviluppo del progetto, gli ultimi giorni sono stati dedicati alla redazione della documentazione.

# Capitolo 5

# Conclusioni

## 5.1 Ripartizione delle ore

Durante i primi giorni del percorso di stage è stato riscontrato che la pianificazione delle ore settimanali stilata necessitava di alcuni adattamenti, in quanto risultava poco efficace e proficuo il periodo inziale di studio teorico delle tecnologie.

È stato pertanto concordato di affiancare alla formazione la progettazione e l'implementazione del prototipo. In questo modo è stato possibile da un lato integrare le nuove conoscenze apprese con l'applicazione pratica delle stesse e dall'altro ottimizzare i tempi di lavoro.

## 5.2 Obiettivi raggiunti

Nella Tabella 5.1 e nella Tabella 5.2 si riportano gli obiettivi del progetto e la relativa valutazione. Tutti gli obiettivi obbligatori sono stati soddisfatti, mentre l'obiettivo facoltativo non è stato raggiunto perché i tempi necessari per la progettazione e l'implementazione del prototipo e per i relativi test, per verificarne la validità, hanno richiesto maggior tempo rispetto a quello preventivato.

| **OBBLIGATORI** | **ESITI** |
|---|---|
| Studio delle tecnologie disponibili | Soddisfatto |
| Individuazione delle tecnologie adatte al contesto applicativo individuato | Soddisfatto |
| Implementazione di un prototipo per una delle tecnologie individuate | Soddisfatto |

**Tabella 5.1:** Valutazione degli obiettivi obbligatori

| FACOLTATIVI | ESITI |
|---|---|
| Implementazione di un secondo prototipo per un'altra delle tecnologie individuate | Non soddisfatto |

**Tabella 5.2:** Valutazione degli obiettivi facoltativi

## 5.3 Considerazioni personali

Al termine del percorso di studi ritengo di aver sviluppato molteplici competenze grazie all'integrazione tra le diverse esperienze vissute.

In primo luogo, lo studio teorico delle tecnologie abbinato costantemente al relativo campo applicativo mi ha permesso di comprendere l'importanza di pianificare sempre il tempo a disposizione calibrandolo in base alla necessità di approfondire una specifica tematica. Ho compreso la rilevanza di collegare le mie nuove conoscenze alle esperienze di volta in volta vissute e ne ho individuato l'efficacia e l'utilità nel lavoro.

Ho potuto sperimentare metodi e tecniche di auto-apprendimento, selezionando quelli maggiormente adatti alla mia persona e ho rilevato la necessità di un costante e continuo aggiornamento formativo.

Durante lo stage ho potuto incrementare le mie competenze relative alla gestione tecnologica e l'implementazione di un progetto software. Nello specifico, lo studio di Openvidu mi ha permesso di capire come le sue componenti possano interagire con l'architettura sottostante e come un prodotto già esistente si possa adattare estendendone le funzionalità in base alle proprie esigenze. Ho sperimentato anche i possibili rischi connessi allo sviluppo del progetto. Durante l'emergenza sanitaria a causa del Covid-19, ho infatti contratto il virus e pertanto ho dovuto interrompere il percorso di stage, posticipando così le attività previste.

Oltre alle competenze tecniche, ho migliorato le mie competenze relazionali. Ho lavorato in un team giovane e dinamico, formato da persone disponibili e competenti, che mi ha permesso di crescere grazie al confronto e alla collaborazione. Ho imparato a comunicare perplessità e difficoltà, quando necessario, e a condividere le conoscenze apprese e i risultati raggiunti con i membri del team. Pertanto, ritengo che lo stage sia stato un percorso formativo significativo per la mia carriera professionale.

# Bibliografia

## Riferimenti bibliografici

Griffiths, Patrick. *XHTML e CSS. Il web secondo HTML Dog*. Pearson Education, 2007.

## Siti web consultati

*Guida Html*. URL: https://www.html.it/.

*IDX-Italy*. URL: https://www.idxitaly.com/.

*Kurento*. URL: https://www.kurento.org/.

*Openvidu*. URL: https://docs.openvidu.io/en/2.15.0/.

*W3Schools*. URL: /https://www.w3schools.com/.

*WebRTC*. URL: https://webrtc.org/.

*Wikipedia*. URL: https://it.wikipedia.org/.