

Progetto senza titolo

Gabriele Masini #108456

2019-12-10

Indice

1	Preambolo	1
2	Introduzione	1
2.1	Scopo	1
2.2	Campo di applicazione	1
2.3	Definizioni, acronimi, abbreviazioni	1
2.4	Riferimenti	2
2.5	Vista d'insieme del documento	2
3	Descrizione generale	2
3.1	Prospettiva del prodotto	2
3.2	Funzionalità del prodotto	3
3.3	Caratteristiche degli utenti	4
3.4	Vincoli e limiti	5
3.5	Presupposti e dipendenze	5
3.6	Requisiti futuri	6
4	Specifiche dei requisiti	6
4.1	Interfacce con l'esterno	6
4.2	Requisiti funzionali	6
4.3	Requisiti non funzionali	6
5	Appendice	6

1 Preambolo

Grazie alla diffusione dell'*Internet of Things* e con l'introduzione sul mercato di dispositivi come *Amazon Echo* si sono resi disponibili al consumatore una molitudine di prodotti *smart* tra cui telecamere ip, videocitofoni e sensori di vario genere atti a "digitalizzare" la propria esperienza casalinga. Che si critichi o no la cosa, questa diffusione di dispositivi rende disponibile all'uomo tecnologie multimediali a basso prezzo. È esattamente in questo contesto che si inserisce il progetto *Progetto senza titolo*, il quale si pone come obiettivo la realizzazione di un sistema di gestione e registrazione dei propri dispositivi *smart* per il monitoraggio di un determinato ambiente.

2 Introduzione

2.1 Scopo

Il presente documento si riferisce alla definizione e descrizione delle informazioni necessarie per lo sviluppo del progetto *Progetto senza titolo* e interessa tutte le entità coinvolte nella progettazione, sviluppo e utilizzo dello stesso. Il documento viene redatto secondo le direttive espresse nel *IEEE Recommended Practice for Software Requirements Specifications* [1], ovvero secondo lo standard *IEEE Std 830-1998*.

2.2 Campo di applicazione

Il progetto nominato *Progetto senza titolo* si inserisce nel contesto odierno di forte sviluppo di tecnologie relative all'*Internet of Things* e si pone come obiettivo la gestione dei dispositivi *smart* multimediali per scopi di monitoraggio e videosorveglianza della propria abitazione, nonché di metodi comuni di registrazione dei sopra menzionati dispositivi. *Progetto senza titolo* non si interfacerà utilizzando protocolli proprietari dei singoli produttori, ma utilizzerà protocolli comuni e di pubblico dominio di utilizzo, incluso, ma non limitato a, *http*, *rtsp*, *udp*.

2.3 Definizioni, acronimi, abbreviazioni

Nel presente documento vengono spesso utilizzate abbreviazioni e acronimi comuni nel contesto di sviluppo del progetto. Di seguito vengono riportate le interpretazioni delle suddette abbreviazioni:

IOT	”Internet of Things”, l’aggregazione in internet di dispositivi di poca capacità di calcolo destinati come controllori per determinati elettrodomestici o utilizzati come sensori (telecamere, termometri, microfoni etc).
Ipcam	Telecamera, spesso utilizzata per videosorveglianza, che rende disponibile verso l’esterno una interfaccia ip per la gestione della stessa e la visione dello stream video tramite un protocollo comune (rtsp, udp etc).
Stream Server	Flusso di dati di vario tipo (audio, video, audio-video etc). Elaboratore che gestisce in modo centralizzato un insieme di dati e espone una interfaccia per interagire con più <i>Client</i> tramite uno o più protocolli.
Client	Elaboratore che si collega ad un <i>Server</i> per poter usufruire dei servizi da esso offerti.
Utente	Persona o ente che tramite un <i>Client</i> utilizza l’applicazione.
Amministratore	Persona o ente responsabile del mantenimento del <i>Server</i> . Spesso è anche un <i>Client</i> .

2.4 Riferimenti

- [1] IEEE, “IEEE Recommended Practice for Software Requirements Specifications,” Software Engineering Standards Committee of the IEEE Computer Society, rapp. tecn., 1998.
- [2] E. Gamma, R. Helm, R. Johnson e J. Vlissides, *Design Patterns, Elements of Reusable Object-Oriented Software*. 1994.

2.5 Vista d’insieme del documento

Il presente documento contiene le specifiche e le relative descrizioni dei requisiti del progetto *Progetto senza titolo*. Il progetto viene inquadrato secondo quanto citato in questa sezione e si prosegue nella sezione successiva a descriverne la prospettiva e le funzionalità, nonché la caratterizzazione dei potenziali utenti, dei vincoli e delle dipendenze. Nella sezione *Specifica dei requisiti*, verranno spiegate le interfacce con l’esterno e i requisiti funzionali e non. Alla fine del documento viene proposto in *Appendice* un possibile utilizzo dei celeberrimi *design pattern* [2] per la programmazione ad oggetti del progetto.

3 Descrizione generale

3.1 Prospettiva del prodotto

Il progetto *Progetto senza titolo* è inteso come una applicazione *stand alone* programmata in *Java* che permetta di gestire una aggregazione di dispositivi atti a monitorare lo stato di uno o più ambienti (ad esempio: una abitazione,

un ufficio, un parco etc). La applicazione è un programma che viene eseguito su una macchina *Server* ed essa espone all'esterno una interfaccia web (*http*) per l'interazione con l'utente; inoltre il programma presenta una interfaccia sul server di tipo "a riga di comando" per la gestione da parte dell'amministratore.

La natura di Java permette che la applicazione sia *cross-platform* in modo da supportare ambienti Microsoft Windows, Linux e macOS. Anche se Java permette l'utilizzo della applicazione su piattaforme diverse, le uniche supportate ufficialmente dal progetto sono le 3 sopra citate con riferimento a macchine ad architettura *x86* e *amd64*.

Il software utilizza un database MariaDB per la memorizzazione dei dati degli utenti e delle risorse. Questo servizio database deve essere installato sulla macchina stessa del server o comunque essere da esso accessibile tramite la rete internet. Per quanto concerne l'elaborazione degli *stream* audio e video, l'applicazione si avvale del progetto *ffmpeg*.

L'applicazione necessita di almeno una interfaccia di rete installata e configurata correttamente sulla macchina *Server* per la comunicazione coi dispositivi da monitorare, l'accesso da parte dei *Client* e la connessione al servizio di database (nel caso questo sia installato su una macchina remota).

Il *Server* deve disporre di sufficiente spazio di archiviazione per la memorizzazione dei dati temporali dei vari dispositivi (ad esempio: registrazioni video, registrazioni audio, andamento dei sensori etc) per il periodo di mantenimento indicato dall'amministratore. Per completezza è disponibile anche l'archiviazione su server esterno via rete internet, anche se il suo utilizzo è sconsigliato per enti e privati che non dispongono di una banda larghissima in upload, la quale è necessaria per il trasporto di stream di grandi dimensioni come audio e video.

Per aggregazioni di piccole dimensioni (il numero effettivo dei dispositivi audiovisivi dipende dal numero di elaborazioni intermedie e la risoluzione dei singoli dispositivi, in generale 15 è un buon compromesso) è sufficiente disporre sul server di un processore *multi core* per l'elaborazione intermedia degli stream audio-video (se necessaria). Per aggregazioni di grandi dimensioni è necessaria una scheda video Nvidia® che supporti l'elaborazione tramite CUDA e utilizzare la corretta versione di *ffmpeg*.

3.2 Funzionalità del prodotto

La applicazione dovrà disporre di due interfacce separate, una accessibile tramite *http* e una via riga di comando.

All'interfaccia *http* avranno accesso gli utenti abilitati a visionare in *real time* lo stato dei vari dispositivi, nonché accedere alle registrazioni. Ogni utente deve essere prima autenticato con apposite credenziali.

L'interfaccia via riga di comando è riservata al solo amministratore del sistema. Anch'essa richiede l'inserimento di una password d'amministratore per evitare eventuali manomissioni.

La gerarchia di dispositivi è gestita sottoforma di albero, ovvero ogni dispositivo può essere aggregato in un gruppi che può a sua volta appartenerne ad un'altro gruppo. Non sono ammesse relazioni ricorsive tra gruppi. Tutti i

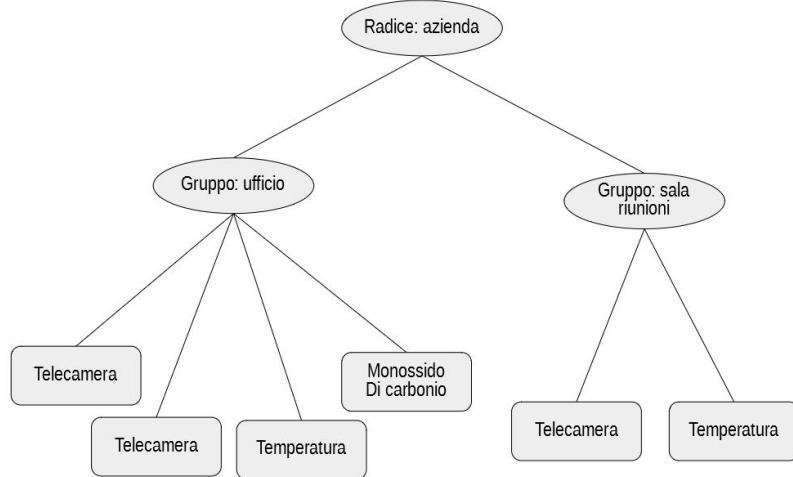
gruppi/dispositivi sono figli (diretti o indiretti) di un gruppo radice. In figura 1 si può osservare un esempio di organizzazione gerarchica di dispositivi e gruppi.

Gli utenti con i giusti permessi e l'amministratore possono gestire la gerarchia, operando spostamenti, rinomine, cancellazioni e aggiunte di dispositivi e gruppi.

Per ogni gruppo è sufficiente memorizzarne solamente il nome e una descrizione.

Per ogni dispositivo invece è necessario memorizzarne il nome, una descrizione e il percorso dove salvare le registrazioni. Inoltre l'utente può specificare anche una operazione intermedia, che per dispositivi audio-video può essere una operazione di transcodifica (utilizzando quindi ffmpeg) da un formato ad un altro oppure, nel caso di sensori con risultati numerici, una formula matematica (ad esempio il sensore restituisce i gradi su scala Celsius, ma l'utente vuole memorizzare i gradi in scala Kelvin).

Figura 1: Esempio di una possibile gerarchia di dispositivi (nei rettangoli) e gruppi (negli ovali)



3.3 Caratteristiche degli utenti

Gli utenti che si interfacceranno con l'applicazione sono l'amministratore e gli utenti generali. L'amministratore ha completo accesso su tutte le parti della applicazione (sia nell'interfaccia web, sia nell'interfaccia a riga di comando). Gli

utenti generali avranno dei determinati permessi a loro assegnati per l'utilizzo dell'interfaccia web.

L'utente amministratore deve avere un ottimo grado di dimestichezza con la macchina *Server* su cui configura l'applicazione, sia dal punto di vista del sistema operativo, sia in campo di reti e telecomunicazione, sia nell'ottica di configurazione di programmi via riga di comando, sia, infine, in campo di formati multimediali audio-video. È sufficiente un diploma di tecnico/perito informatico; è consigliata una laurea in campo informatico.

Gli utenti generali sono, invece, di tipo eterogeneo. È comunque necessario che essi sappiano utilizzare un *personal computer* nelle sue funzioni ad altissimo livello (nel senso di comprendere le interfacce grafiche e la loro navigazione) dimodoché siano in grado di utilizzare un *browser web* per l'accesso all'applicazione. Per essi non sono necessarie competenze avanzate in campo di informatica, ma sono consigliate.

3.4 Vincoli e limiti

L'applicazione è limitata e/o vincolata nel suo funzionamento da:

- La memoria massima di archiviazione dei dati.
- Il numero di thread o core del processore e/o la potenza di calcolo della GPU.
- Le funzioni e i metodi che *ffmpeg* mette a disposizione.
- La banda massima supportata dalla connessione ad internet, nonché la banda massima della rete locale.
- Le politiche di privacy necessarie per memorizzare dati basilari di utenti (e-mail, username e password).
- La presenza sul server di una *Java Virtual Machine*.
- La presenza sul client di un *browser web*

3.5 Presupposti e dipendenze

Il progetto dipende dai progetti *MariaDB* e *ffmpeg*, utilizza il protocollo ip e i protocolli che si appoggiano ad esso (http, udp, rtsp etc) e supporta solo macchine *x86* e *amd64* con sistema operativo Microsoft Windows, Linux, maxOS. I browser supportati dall'interfaccia web sono Microsoft Edge, Apple Safari, Mozilla Firefox, Google Chrome (e suoi derivati, es: Chromium, Opera etc). Queste dipendenze possono portare al cambiamento del presente documento e, di conseguenza, all'applicazione.

3.6 Requisiti futuri

In futuro può essere necessario supportare più protocolli e più dispositivi. Può essere utile implementare una comunicazione tra l'applicazione e sistemi di *smart home* come Amazon Echo.

4 Specifica dei requisiti

4.1 Interfacce con l'esterno

EI01	Database Gerarchia dispositivi
Descrizione	L'applicazione riceve in formato JSON la lista di dispositivi, gruppi e la loro organizzazione gerarchica.
Formato dati	<pre>{ { "nome": "dispositivo 1", "configurazione": { ... } }, { "nome": "gruppo 1", "children": { { "nome": "dispositivo 2", "configurazione": { ... } } } } }</pre>

4.2 Requisiti funzionali

RF01	Subsys	Func
Input	input descr	
Output	input descr	

4.3 Requisiti non funzionali

5 Appendice

// design pattern // tabella con dimesione video