

UNIVERSITA' DEL PIEMONTE AMEDEO AVOGADO

Corso di Laurea in Informatica

## MagicMirror

**Relatore**

Prof. Marco Guazzone

**Relazione della prova**

**finale di:**

Riccardo Berto

Matricola: 20003275

Anno Accademico 2016/2017

# Abstract

L'obiettivo dello stage era lo sviluppo di moduli in diversi linguaggi che permettesse l'interazione tra uomo-macchina da parte di un dispositivo, e di un'interfaccia che permettesse di configurare quest'ultimo da una pagina web.

Tra i moduli sviluppati vi sono il riconoscimento di presenza umana o di uno specifico volto (tramite l'utilizzo di una telecamera) e il riconoscimento vocale, ovvero l'impartizione di comandi specifici tramite voce per mezzo di un microfono.

Nello sviluppo di questi moduli si sono incontrate moderne tipologie di software come OpenCV e Google SpeechToText che sono state implementate con diverse tipologie di linguaggi.

La pagina web sviluppata aveva il compito di modificare la configurazione dei moduli (anche quelli creati da terzi) senza dover andare a modificare il file in locale rendendo pi facile aggiornare le features del dispositivo.

# Contents

<b>Abstract</b>	<b>I</b>
<b>Introduzione</b>	<b>III</b>
<b>1 Scopi e Problemi</b>	<b>IV</b>
1.1 Moduli del MagicMirror . . . . .	IV
1.2 Interfaccia di controllo del MagicMirror . . . . .	IV

# Introduzione

Durante

prova

prova

prova

prova

# Chapter 1

## Scopi e Problemi

### 1.1 Moduli del MagicMirror

Nello sviluppo dei moduli del MagicMirror si é dovuto affrontare il problema di far comunicare i diversi dispositivi hardware (tra cui microfono e telecamera) con il calcolatore principale, utilizzando tecnologie e software diversi tra loro e linguaggi differenti, dal momento che ognuno era risultato pi facile da implementare con uno specifico linguaggio rispetto ad un altro.

Lo scopo dei moduli é di permettere un'interazione umana tra il software principale e l'essere umano, e di poter controllare la macchina senza bisogno di usare input meccanici come mouse e tastiera.

### 1.2 Interfaccia di controllo del MagicMirror

Nello sviluppo della pagina Web di controllo si é affrontato il problema di dover modificare il documento di configurazione del software principale da una pagina web senza eliminarne parti essenziali o modificarlo in un formato sbagliato (gestito tramite validatore). Lo scambio dei messaggi tra il Backend ed il Frontend avviene in formato JSON, lo stesso formato con cui é stipulato il file di configurazione del Magic Mirror.

Lo scopo dell'interfaccia di permettere ad un qualsiasi utente (con i permessi) di gestire i diversi moduli implementati nel software modificandone il JSON, senza accedere fisicamente alla macchina. La pagina, inoltre, permette di configurare moduli creati anche da terzi: inserendoli semplicemente nella directory del dispositivo il Backend li cerca e li indicizza nell'interfaccia, da

dove possono essere attivati.

L'ambiente in cui é stato svolto lo stage é Raspian una distribuzione Debian che gira sul calcolatore Rasperry.

# Bibliography

- [1] Docker official website, <https://www.docker.com>, Ultimo accesso: 05/10/2016
- [2] runC Open Container Project, <https://runc.io>, Ultimo accesso: 05/10/2016
- [3] Jrme Petazzoni (Docker), <http://www.slideshare.net/jpetazzo/implementing-separation-of-concerns-with-docker-and-containers>, Ultimo accesso: 05/10/2016
- [4] Wes Felter - Alexandre Ferreira - Ram Rajamony - Juan Rubio (IBM), An Updated Performance Comparison of Virtual Machines and Linux Containers, July 21 2014
- [5] Aaron Grattafiori (NCC Group Whitepaper), Understanding and Hardening Linux Containers, 2016 April 20
- [6] Golang website, <http://golang.org>, Ultimo accesso: 05/10/2016
- [7] runC Open Container Project, <https://golang.org/ref/spec>, Ultimo accesso: 05/10/2016
- [8] Tiobe Programming languages index , <http://www.tiobe.com/tiobe-index/>, Ultimo accesso: 05/10/2016
- [9] GIT website, <https://git-scm.com>, Ultimo accesso: 05/10/2016
- [10] Tommi Virtanen article (Computer Scientist), <http://eagain.net/articles/git-for-computer-scientists/>, Ultimo accesso: 05/10/2016
- [11] Manifesto agile, <http://agilemanifesto.org>, Ultimo accesso: 05/10/2016

- [12] Metodologie agili, [http://www.umsl.edu/~sauterv/analysis/6840\\_f09\\_papers/Nat/Agile.html](http://www.umsl.edu/~sauterv/analysis/6840_f09_papers/Nat/Agile.html), Ultimo accesso: 05/10/2016
- [13] Roger Pressman e Bruce Maxim, Software Engineering: A Practitioner's Approach, 7th edition
- [14] Roger Pressman e Bruce Maxim, Software Engineering: A Practitioner's Approach, 7th
- [15] Kanban, <https://www.atlassian.com/agile/kanban>, Ultimo accesso: 05/10/2016
- [16] Trello, virtual kanban board, <https://trello.com>, Ultimo accesso: 05/10/2016
- [17] Libreria libcompose, <https://github.com/docker/libcompose>, Ultimo accesso: 05/10/2016
- [18] Libreria libcompose, <http://yaml.org>, Ultimo accesso: 05/10/2016
- [19] Libreria spf13/Cobra, <https://github.com/spf13/cobra>, Ultimo accesso: 05/10/2016
- [20] Alan A. A. Donovan Brian W. Kernighan, The Go Programming Language, 2016 October 26
- [21] Go Testing Package, <https://golang.org/pkg/testing>, Ultimo accesso: 05/10/2016