

# Scalavelli project

Daniele Tentoni "Part Time Team"

25 agosto 2020

### **Abstract**

Progetto scritto in Scala per giocare al celebre gioco Machivelli sul proprio computer e sfidare altri giocatori.

## **Contents**

# 1 Processo di sviluppo

## 1.1 Metodologia

Sin da subito abbiamo deciso di adottare una metodologia di sviluppo *Agile-Scrum*, seppur non scegliendo un vero e proprio Scrum Master. Chi più e chi meno, a seconda dei vari sprint, praticamente tutti hanno avuto l'occasione, il modo e la motivazione di ricoprire tale ruolo, in modo tale da favorire il proseguimento del progetto coordinando il team con l'aiuto degli strumenti a disposizione. In accordo con tale metodologia, il lavoro è stato suddiviso in sprint al termine dei quali si sarebbero dovute sviluppare un numero minimo di funzionalità dell'applicativo, della durata media di una settimana e mezzo. I meeting sono stati frequenti all'inizio e ne sono stati svolti alcuni saltuariamente all'interno di ogni sprint. Questo perché, alternandosi con un periodo di lavoro autonomo, abbiamo ritenuto necessario e produttivo confrontarsi anche durante gli sprint su scelte sintattiche e pattern di sviluppo tra tutti i membri del team, in modo da condividere conoscenze ed entusiasmo.

## 1.2 Strumenti adottati

Come strategia di workflowing è stata scelta questa variante molto semplice, schematizzata di seguito: Come strumenti di *Build Automation* si è deciso di usare *SBT*, dato che durante il corso lo si è sempre preferito rispetto al cugino *Gradle* per lo svolgimento degli elaborati, nonostante il fatto che alcuni componenti del gruppo lo utilizzino in ambito aziendale. La comunicazione all'interno del gruppo è avvenuta su *Telegram* per quanto concerne meri aspetti organizzativi, mentre la condivisione di appunti e blocchi di codice durante le sessioni di sviluppo è stato sfruttato *Microsoft Teams*. Per la *Continuous Integration* è stato scelto *Travis CI*. Consisteva nell'unica soluzione freeware che i componenti del gruppo abbiano mai usato, introdotta proprio in questo corso. Tuttavia, è stato necessario approfondire l'argomento tramite studio autonomo, dato che ciò che si era appreso a lezione è stato ritenuto insufficiente per la buona riuscita del progetto per come è stato pensato. Ad esso è stata affidata l'esecuzione dei test necessari per verificare la correttezza del lavoro svolto e poter poi effettuare dei rilasci senza regressioni. Tramite lo stesso servizio, è stata effettuata la *Continuous Delivery*, rilasciando dei pacchetti compilati su *Github Releases*.

## **2 Requisiti**

### **2.1 Business**

### **2.2 Utente**

### **2.3 Funzionali**

### **2.4 Non Funzionali**

### **2.5 Implementativi**

## **3 Design Architettuale**

### **3.1 Model**

### **3.2 View**

### **3.3 Controller**

## 4 Design di Dettaglio

## **5 Implementazione**

### **5.1 Matteo**

### **5.2 Luca**

### **5.3 Lorenzo**

### **5.4 Daniele**

#### **5.4.1 Continuous Integration**

Io mi sono occupato di configurare opportunamente l'ambiente di CI scelto in modo da poter verificare la correttezza di ogni singola build, compilando ad ogni push su ogni branch e pull request. Inoltre, viene effettuato anche un upload sul sito Codecov.io (che si occupa di mantenere delle statistiche sulla copertura dei test sul progetto), del rilascio della Doc e Scaladoc ad ogni rilascio sul branch di sviluppo sull'ambiente di Github-Pages in modo da renderlo disponibile per tutti e di effettuare un rilascio dei pacchetti eseguibili degli applicativi server e client ad ogni push sul master che sia stata etichettata da un tag. Questa parte ha richiesto molto lavoro ancora prima di iniziare a sviluppare, ma successivamente tutto il team ne ha tratto beneficio. Successivamente, in corso d'opera, si è trattato solamente di adattare mano a mano la configurazione già esistente alle crescenti esigenze del progetto. In primo luogo, il bisogno di stringere i tempi di compilazione sull'ambiente remoto, che già all'inizio del progetto, quando quindi era ancora relativamente piccolo, iniziavano già a richiedere diversi minuti per ogni singolo passaggio, andandosi a sommare in lunghe compilazioni tra i 10 e i 15 minuti. Successivamente ha richiesto di risolvere qualche problema con la compilazione e l'impacchettamento degli eseguibili.

#### **5.4.2 Build automation**

Mi sono occupato inoltre degli script necessari per eseguire agilmente una compilazione di tutto il progetto o di parte di esso in base alle esigenze. Ho configurato tutto il progetto in modo che fosse logicamente diviso in moduli in modo da aumentare l'incapsulamento e l'esposizione ad altre porzioni di progetto di solo le parti necessarie. Ho inoltre configurato tutti i pacchetti e le librerie aggiuntive da importare nel progetto.

#### **5.4.3 Model**

Assieme a Lorenzo mi sono occupato dello sviluppo delle entità base del progetto e di tutti quegli elementi di gioco che riguardavano le regole del gioco e l'iterazione tra di esse.

## **6    Retrospettiva**

### **6.1   Problemi riscontrati**

### **6.2   Sviluppi futuri**