

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

HOMEWORK 1

DOCUMENTO DEI REQUISITI

03/05/2020

Corso di **Ingegneria del Software**



Indice

1	Introduzione	3
1.1	Consegna iniziale del progetto.....	3
1.2	Scopo del progetto	4
1.3	Scopo del documento dei requisiti	4
2	Requisiti dell'utente.....	5
3	Requisiti di sistema	6
3.1	Requisiti funzionali	6
3.2	Requisiti non funzionali.....	10
4	Caratteristiche del sistema	12
4.1	Architettura del sistema.....	12
4.2	Evoluzione del sistema	13
4.3	Modelli di sistema	13
5	Interviste committenti	14
5.1	Committente 1	14
5.2	Committente 2	14
6	Appendice.....	15
6.1	Stakeholders.....	15
6.2	Cronologia versioni documento	15
6.3	Glossario	15

1 Introduzione

1.1 Consegna iniziale del progetto

Si consideri una mappa rettangolare su cui sia sovrapposta una griglia di coordinate intere x, y che la divide in $M \times N$ celle.

In ciascuna cella è possibile posizionare sino a cinque “pezzi”.

Ogni pezzo è caratterizzato da un tipo {elfo/nano/orco}.

- 1) È necessario sviluppare una classe specifica per ciascuna tipologia di pezzo.
- 2) Ogni cella della mappa ha una tipologia (pianura/bosco/montagna).
- 3) Esiste un riferimento temporale (giorno/notte).
- 4) I valori di attacco e difesa sono inizialmente definiti per tipologia (A/D):
 - Elfo 5/2;
 - Nano 2/5;
 - Orco 4/4.
- 5) I pezzi sono caratterizzati da modificatori di combattimento (+A%/+D%):
 - elfi bosco +0%/+100%;
 - nani montagna +100%/+0%;
 - orco giorno -50%/-50%;
 - orco notte +50%/+50%.

Si leggano i dati relativi ad i pezzi da disporre sulla mappa da un file contenente gruppi di linee di testo secondo il formato:

X

Y

tipo

e lì si disponga in una struttura dati adatta. Letti i dati, si calcoli:

- 1) Il numero di pezzi presenti sulla mappa per ciascuna tipologia.
- 2) La casella con il maggior valore di difesa di giorno.
- 3) La casella con il maggior valore di difesa di notte.
- 4) La casella con il maggior valore di attacco di giorno.
- 5) La casella con il maggior valore di attacco di notte.
- 6) La casella con il maggior numero di pezzi dello stesso tipo.

1.2 Scopo del progetto

“Homework1” è un progetto universitario che prevede la creazione della base di un videogioco con funzionalità basate da una descrizione iniziale del gioco e da interviste a due committenti. Il progetto prevede: lo sviluppo del software in linguaggio Java, un documento dei requisiti e un documento di architettura del software.

1.3 Scopo del documento dei requisiti

Questo è un documento di specifica dei requisiti del progetto “Homework1”.

In seguito alle interviste con i due committenti viene fatta la stesura di questo documento contenente tutto ciò che dovrà fare il sistema.

Il documento sui requisiti software è la dichiarazione ufficiale di ciò che è richiesto agli sviluppatori di sistema.

Include sia una definizione dei requisiti dell'utente che una specifica dei requisiti di sistema e le informazioni contenute nel documento sui requisiti dipendono dal tipo di sistema e dall'approccio allo sviluppo utilizzato.

In particolare, questo documento include quanto segue:

- Introduzione:
 - Consegna iniziale del progetto
 - Scopo del progetto
 - Scopo del documento dei requisiti
- Requisiti dell'utente
- Requisiti del sistema:
 - Requisiti funzionali
 - Requisiti non funzionali
- Caratteristiche del sistema
 - Architettura del sistema
 - Evoluzione del sistema
 - Modelli di sistema
- Interviste committenti:
 - Committente 1
 - Committente 2
- Appendice:
 - Stakeholder
 - Cronologia versioni documento
 - Glossario

2 Requisiti dell'utente

1	Correttezza delle funzionalità
Descrizione	Il programma deve rispettare le caratteristiche fornite dalla consegna iniziale del progetto. Deve inoltre visualizzare a schermo le dimensioni della mappa creata e l'avvenuto inserimento dei pezzi nella mappa.

2	Lettura dati da file
Descrizione	Il programma deve avere accesso a dati forniti da file. Sia per la creazione della mappa, sia per l'inserimento dei pezzi nella mappa.

3	Controllo formattazione dei file
Descrizione	I dati letti da file devono essere formattati correttamente così da far funzionare correttamente il programma, in caso contrario il programma verrà terminato visualizzando a video la descrizione della problematica riscontrata.

4	Consistenza programma
Descrizione	I dati letti da file devono essere controllati e in caso non rientrino nei valori prestabiliti l'utente sarà avvisato della problematica riscontrata e se essa ha determinato la terminazione del programma o meno.

5	Utilizzabilità
Descrizione	Il software deve poter essere accessibile da terminale ed essere intuitivo all'utilizzo.

6	Affidabilità
Descrizione	Il programma non deve ammettere errori di calcolo o errori di visualizzazione dei dati richiesti.

7	Sicurezza privacy
Descrizione	Il sistema deve garantire sicurezza della privacy dell'utente non richiedendo alcun inserimento di dati e alcuna connessione ad internet

3 Requisiti di sistema

3.1 Requisiti funzionali

Homework1 / Functional Requirments / FR1	
Funzione	Creazione mappa
Descrizione	Creazione di una mappa rettangolare su cui sia sovrapposta una griglia di coordinate intere x, y che la divide in M x N celle. Ogni cella della mappa ha una tipologia (pianura/bosco/montagna).
Input	M (numero colonne), N (numero righe), tipologia di ogni cella.
Fonte	File di testo.
Output	Dimensioni della mappa.
Destinazione	Per l'utente, output standard.
Azione	Legge i dati in input e genera la mappa con le celle della tipologia assegnata.
Requisiti	File di testo formattato correttamente e con valori che rientrano nei range prestabiliti.
Pre-condizioni	File di testo con il nome corretto e nella cartella corretta del progetto.
Post-condizioni	Mappa creata.
Effetti collaterali	Nessuno.

<i>Homework1 / Functional Requirments / FR2</i>	
Funzione	Riempimento mappa
Descrizione	Inserimento dei pezzi in input nelle celle della mappa corrispondenti. In ciascuna cella è possibile posizionare sino a cinque “pezzi”. Ogni pezzo è caratterizzato da un tipo {elfo/nano/orco}.
Input	Per ogni pezzo da inserire: tipo di pezzo, coordinate dove inserirlo.
Fonte	File di testo.
Output	Avviso dell’avvenuta o non avvenuta dell’inserimento.
Destinazione	Per l’utente, output standard.
Azione	Legge i dati in input e inserisce ogni pezzo nella corrispondente cella, se c’è spazio, se no non lo aggiunge.
Requisiti	Mappa creata. File di testo formattato correttamente.
Pre-condizioni	File di testo con il nome corretto e nella cartella corretta del progetto.
Post-condizioni	Mappa riempita con i pezzi forniti in input.
Effetti collaterali	Nessuno.

<i>Homework1 / Functional Requirments / FR3</i>	
Funzione	Stampa risultati 1
Descrizione	Calcolare e visualizzare: 1) Il numero di pezzi presenti sulla mappa per ciascun tipo. 2) La cella con il maggior numero di pezzi dello stesso tipo.
Input	Dimensioni mappa, informazioni pezzi inseriti.
Fonte	Programma stesso.
Output	1) Il numero di pezzi presenti sulla mappa per ciascuna tipologia. 2) La casella con il maggior numero di pezzi dello stesso tipo.
Destinazione	Per l'utente, output standard.
Azione	1) Con 3 contatori differenti conta per ogni cella della mappa quanti pezzi ci sono per ogni tipo sommandone ogni volta il numero per ogni cella al rispettivo contatore così da avere alla fine il numero di pezzi per ogni tipo di tutta la mappa. 2) Conta per ogni cella il numero di pezzi per ogni tipo e ricerca quale di queste celle ha il maggior numero di pezzi dello stesso tipo
Requisiti	Mappa generata e riempita con i pezzi forniti in input.
Pre-condizioni	Nessuna.
Post-condizioni	Nessuna.
Effetti collaterali	Nessuno.

<i>Homework1 / Functional Requirments / FR4</i>	
Funzione	Stampa risultati 2
Descrizione	Calcolare e visualizzare: <ul style="list-style-type: none">- La cella con il maggior valore di difesa di giorno.- La cella con il maggior valore di difesa di notte.- La cella con il maggior valore di attacco di giorno.- La cella con il maggior valore di attacco di notte.
Input	Dimensioni mappa, informazioni pezzi inseriti.
Fonte	Programma stesso.
Output	<ul style="list-style-type: none">- La cella con il maggior valore di difesa di giorno.- La cella con il maggior valore di difesa di notte.- La cella con il maggior valore di attacco di giorno.- La cella con il maggior valore di attacco di notte.
Destinazione	Per l'utente, output standard.
Azione	Per ogni richiesta: per ogni cella viene calcolato il valore di attacco/difesa considerando i modificatori di combattimento presenti (tipologia di cella, giorno o notte) e nel frattempo viene fatta la ricerca della cella con il maggior valore richiesto.
Requisiti	Mappa generata e riempita con i pezzi forniti in input.
Pre-condizioni	Nessuna.
Post-condizioni	Nessuna.
Effetti collaterali	Nessuno.

3.2 Requisiti non funzionali

1	Formattazione file di testo: creazione mappa
Requisito	<p>Il file di testo "input_map.txt" che definisce il numero di colonne e di righe deve essere formattato correttamente, avendo nella prima riga il numero (Integer) di colonne (M), nella seconda riga il numero (Integer) di righe (N), alla riga successiva in poi ci sarà la stesura della mappa contrassegnando la tipologia di ogni cella in base alla lettera inserita (sono valide solo 3 lettere -> 'p': pianura, 'b': bosco, 'm': montagna). Le righe (String) destinate alla stesura saranno con tante righe quante le righe della mappa da creare e con la lunghezza di ogni riga pari al numero di colonne della mappa da creare.</p> <p>Esempio del file "input_map.txt":</p> <pre>6 3 mmppbp bmbbbm mppmbb</pre>

2	Formattazione file di testo: riempimento mappa
Requisito	<p>Il file di testo "input_data.txt" che definisce ogni pezzo che si vuole inserire con le corrispettive coordinate deve essere formattato correttamente, avendo le informazioni di ogni pezzo scritte su 3 righe destinate rispettivamente a: coordinata x (Integer), coordinata y (Integer), tipo di pezzo (String).</p> <p>Esempio del file "input_map.txt":</p> <pre>x y tipo x y tipo ...</pre>

3	Correttezza dati input: creazione mappa
Requisito	<p>I dati in input forniti dal file "input_map.txt" sulle dimensioni della mappa da creare devono rispettare dei range:</p> <ul style="list-style-type: none"> - Numero colonne: compreso tra 1 e 20. - Numero righe: compreso tra 1 e 20.

4	Struttura pezzi della mappa
Requisito	In ciascuna cella è possibile posizionare sino a cinque “pezzi”. Ogni pezzo è caratterizzato da un tipo {elfo/nano/orco}. È necessario sviluppare una classe specifica per ciascuna tipologia di pezzo. I valori di attacco e difesa sono inizialmente definiti per tipologia (A/D) Elfo 5/2; Nano 2/5; Orco 4/4.

5	Struttura celle e gestione temporale
Requisito	Ogni cella della mappa ha una tipologia (pianura/bosco/montagna). Esiste un riferimento temporale (giorno/notte).

6	Modificatori di combattimento
Requisito	Esiste un riferimento temporale (giorno/notte). I pezzi sono caratterizzati da modificatori di combattimento (+A%/+D%): elfi bosco +0%/+100%; nani montagna +100%/+0%; orco giorno -50%/-50%; orco notte +50%/+50%.

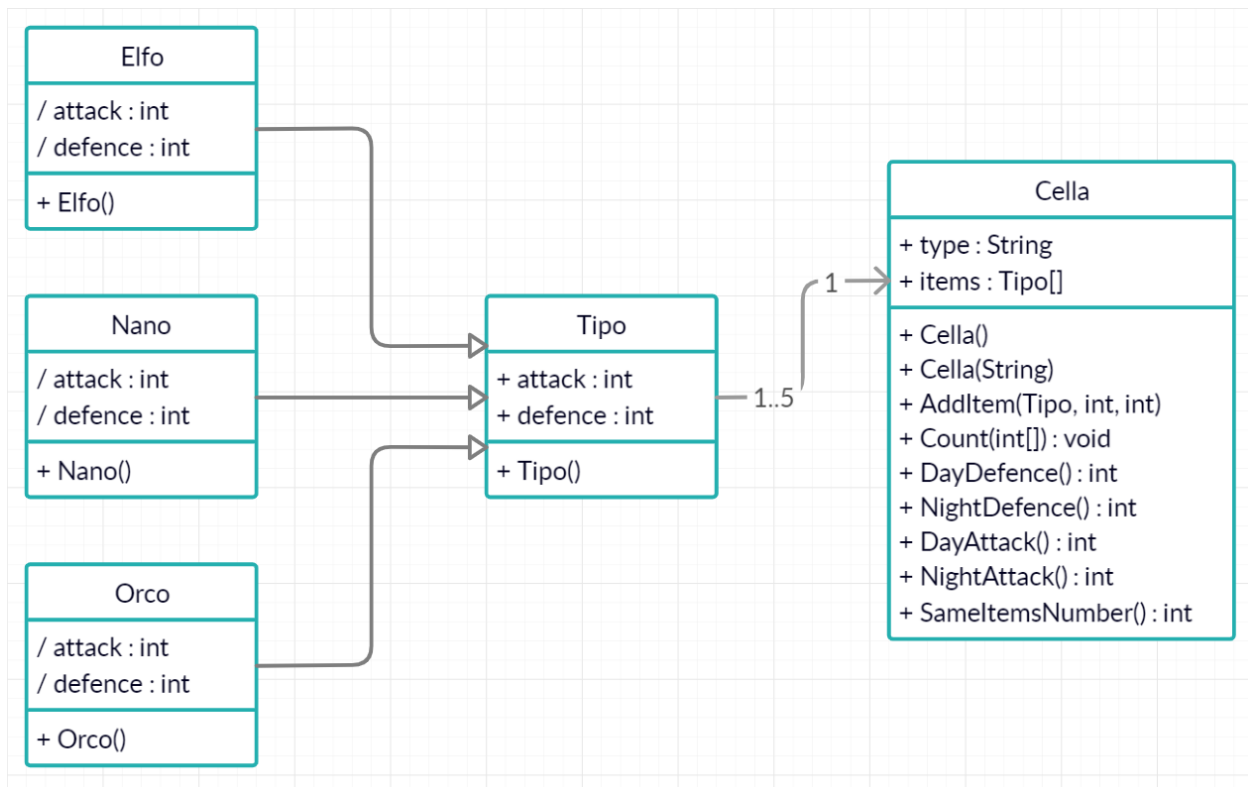
7	Utilizzabilità
Requisito	Il software deve poter essere accessibile da terminale ed essere intuitivo all’utilizzo.

8	Affidabilità
Requisito	Il programma non deve ammettere errori di calcolo o errori di visualizzazione dei dati richiesti.

9	Sicurezza privacy
Requisito	Il sistema deve garantire sicurezza della privacy dell’utente non richiedendo alcun inserimento di dati e alcuna connessione ad internet.

4 Caratteristiche del sistema

4.1 Architettura del sistema

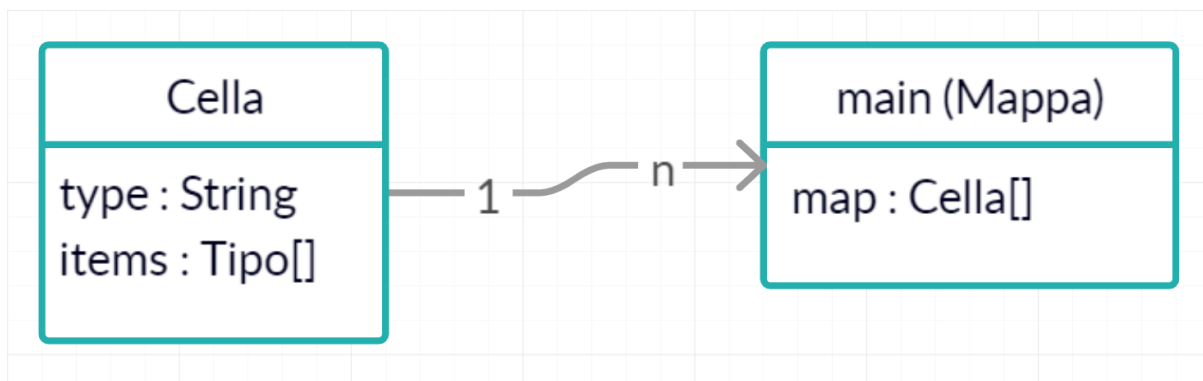


La mappa sarà presente nel main, creata come un array di oggetti Cella. Codice nel main:

```
> Cella[] map = new Cella[dim];
```

Funzioni per lettura dati da file, gestione mappa, elaborazione dei dati e stampa dei risultati saranno presenti nel main.

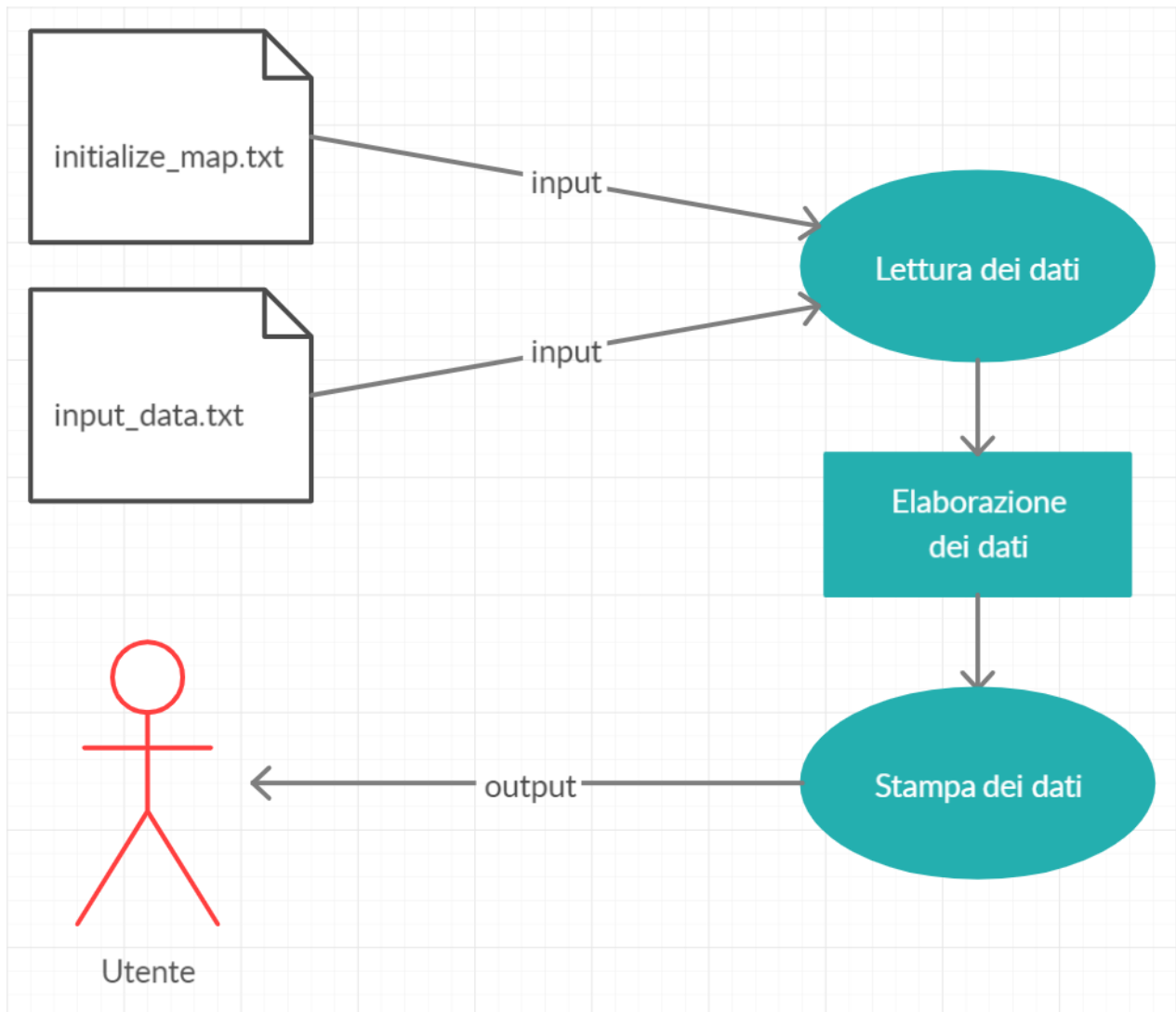
Relazione tra la classe Cella e il main (contenente la mappa):



4.2 Evoluzione del sistema

Dato che questo software che si basa su una mappa di dimensioni che in futuro potrebbero diventare notevoli in caso di modifica del massimo numero di righe e colonne, risulta più opportuno generare la mappa secondo i dati letti da un file di testo invece di impostarli uno ad uno durante l'esecuzione del programma.

4.3 Modelli di sistema



Dispositivi input/output del sistema:

- Input esclusivamente da file di testo.
- Output esclusivamente da output standard, indirizzato all'utente utilizzatore del programma.

5 Interviste committenti

5.1 Committente 1

- Nome: Nicola Lorenzon (intervista del 18/04/2020)

Autore: "In base a cosa definisco le dimensioni della mappa?"

Committente: "In base ad un file 'initialize_map.txt', diverso da quello per l'inserimento dei pezzi che si chiamerà "input_data.txt", dove la prima riga del file equivale al numero di colonne che avrà la mappa, la seconda riga del file equivale al numero di righe che avrà la mappa."

Autore: "In base a cosa definisco la tipologia di ogni cella?"

Committente: "Sempre nel file 'initialize_map.txt' dalla terza riga in poi ci saranno righe e colonne formate dalle lettere 'p' per pianura, 'b' per bosco e 'm' per montagna. Ogni riga nel file di testo equivale ad ogni riga della mappa, andando a capo ogni fine riga, così da avere un numero di righe equivale al numero di righe della mappa e la lunghezza di ogni riga in base al numero di colonne."

Autore: "Ci sono dei limiti sulle dimensioni della mappa?"

Committente: "Sì, la mappa deve avere un minimo di 1 riga e 1 colonna e un massimo di 20 righe e di 20 colonne."

Autore: "Se il file 'initialize_map.txt' è formattato male cosa deve succedere?"

Committente: "Il programma deve terminare stampando l'errore."

Autore: "Se il file per l'aggiunta dei pezzi nella mappa è formattato male cosa deve succedere?"

Committente: "Il programma deve terminare stampando l'errore."

Autore: "Hai qualche altro requisito da comunicarmi?"

Committente: "Sì, come ultimo requisito il sistema deve garantire l'utilizzabilità del software, l'affidabilità dei risultati e la sicurezza della privacy di chi lo usa."

5.2 Committente 2

- Nome: Davide Colussi (intervista del 20/04/2020)

Autore: "Devo stampare qualcosa riguardo informazioni della mappa?"

Committente: "Sì, la stampa delle dimensioni della mappa quando viene generata."

Autore: "Come gestisco l'eventuale tentativo di aggiunta di un 6° pezzo in una casella che ne contiene già 5?"

Committente: "Non aggiungendolo e stampando il fatto che non è stata aggiunto il pezzo di troppo ma continuando comunque l'esecuzione del programma."

Autore: "Come va gestita il tentativo di aggiunta di un pezzo in una cella non esistente?"

Committente: "Non aggiungendolo e stampando il motivo per il quale non è stato aggiunto."

Autore: "Hai qualche altro requisito da comunicarmi?"

Committente: "Sì, l'ultimo requisito è che ogni possibile eccezione deve essere gestita terminando il programma e stampando l'errore per il quale si è generata l'eccezione."

6 Appendice

6.1 Stakeholders

Nome	Descrizione
Simone Bastasin	Proprietario
Nicola Lorenzon	Committente
Davide Colussi	Committente

6.2 Cronologia versioni documento

Versione	Data	Stato	Descrizione	Autore
1.0	20/04/2020	Bozza	Inizio della stesura del documento in seguito alle interviste con i committenti.	Simone Bastasin
1.1	21/04/2020	Quasi definitivo	Completamento della stesura del documento, necessita revisione.	Simone Bastasin
1.2	03/05/2020	Definitivo	Revisione del documento che non ha portato a modifiche importanti ma semplici correzioni a livello di organizzazione e ordine.	Simone Bastasin

Versione attuale: 1.1

6.3 Glossario

Termine / Acronimo	Definizione
Formattazione	Procedimento di strutturazione e decorazione grafica che rende un testo puro adeguato all'utilizzo al quale è mirato.
Input	Nel linguaggio scientifico e tecnico è l'insieme dei dati, informazioni e istruzioni immessi nella fase iniziale in un procedimento.
Output	Nel linguaggio scientifico e tecnico è il risultato finale di un procedimento.