



>Three Way Milkshake_

Manuale Manutentore

Three Way Milkshake - Progetto "PORTACS"

threewaymilkshake@gmail.com

Versione	0.0.1
Stato	Non approvato
Uso	Esterno
Approvazione	_____
Redazione	_____
Verifica	_____
Destinatari	Sanmarco Informatica Prof. Vardanega Tullio Prof. Cardin Riccardo Three Way Milkshake

Descrizione

Manuale di supporto allo sviluppo e manutenzione del software G
PORTACS



Registro delle modifiche

Vers.	Descrizione	Redazione	Data red.	Verifica	Data ver.
0.1.0	Stesura § 1	Tessari Andrea	2021-04-22	_____	_____
0.0.1	Impaginazione	De Renzis Simone	2021-04-15	_____	_____



Indice

1	Introduzione	6
1.1	Scopo del documento	6
1.2	Scopo del prodotto	6
1.3	Riferimenti	6
1.3.1	Normativi	6
1.3.2	Informativi	7
2	Tecnologie e librerie	8
2.1	Server	8
2.1.1	Tecnologie	8
2.1.2	Librerie e Framework	8
2.2	Client	8
2.2.1	Tecnologie	8
2.2.2	Librerie e Framework	8
2.3	Version Control System e Continuous Integration	8
2.3.1	Git e gitflow	8
2.3.2	GitHub	9
2.3.3	GitHub Actions	9
3	Setup	10
3.1	Requisiti di sistema	10
3.1.1	Requisiti Hardware	10
3.1.2	Requisiti Software	10
3.2	Installazione	10
4	Testing	11
4.1	JUnit	11
4.2	Libreria test frontend	11
5	Architettura del sistema	12
5.1	Server	12
5.1.1	Diagramma delle classi	12
5.1.1.1	Persistence layer	12
5.1.1.2	Business layer	12
5.1.1.2.1	Mappa	12
5.1.1.2.2	Clients	12
5.1.1.2.3	Tasks	12
5.1.1.2.4	Collisioni	12
5.1.1.3	Communication layer	12
5.2	Client	12
5.3	Comunicazione	12
5.3.1	Diagrammi di sequenza	12
5.3.2	Protocollo di comunicazione	12



6	Estendere PORTACS	13
6.1	Algoritmo alternativo per il path finding	13
6.2	Introdurre nuove tipologie di utenti	13
6.2.1	Lato server	13
6.2.2	Lato client	13
6.3	Implementare tipi di persistenza alternativi	13
6.4	Modificare handler nell'algoritmo di gestione delle collisioni	13



Elenco delle figure



Elenco delle tabelle



1 Introduzione

1.1 Scopo del documento

Lo scopo di questo documento è presentare tutte le informazioni necessarie al mantenimento e all'estensione del software PORTACS, mostrando nel dettaglio l'architettura del sistema e l'organizzazione del codice sorgente.

In questo documento saranno presentate le varie tecnologie usate, sia lato front end che back end, come anche le varie librerie e framework. Verrà inoltre mostrato il sistema di versionamento utilizzato e la Continuous Integration applicata.

1.2 Scopo del prodotto

Il capitolato_G C5 propone un progetto_G in cui viene richiesto lo sviluppo di un software per il monitoraggio in tempo reale di unità che si muovono in uno spazio definito. All'interno di questo spazio, creato dall'utente per riprodurre le caratteristiche di un ambiente reale, le unità dovranno essere in grado di circolare in autonomia, o sotto il controllo dell'utente, per raggiungere dei punti di interesse posti nella mappa. La circolazione è sottoposta a vincoli di viabilità e ad ostacoli propri della topologia dell'ambiente, deve evitare le collisioni con le altre unità e prevedere la gestione di situazioni critiche nel traffico.

1.3 Riferimenti

1.3.1 Normativi

- NORME DI PROGETTO_G v3.0.0 : per qualsiasi convenzione sulla nomenclatura degli elementi presenti all'interno del documento;
- Regolamento progetto_G didattico:
<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/P1.pdf>;
- Model-View Patterns:
<https://www.math.unipd.it/~rcardin/sweb/2020/L02.pdf>;
- SOLID Principles:
<https://www.math.unipd.it/~rcardin/sweb/2020/L04.pdf>;
- Diagrammi delle classi:
https://www.math.unipd.it/~rcardin/swea/2021/DiagrammidelleClassi_4x4.pdf;
- Diagrammi dei package:
https://www.math.unipd.it/~rcardin/swea/2021/DiagrammideiPackage_4x4.pdf;
- Diagrammi di sequenza:
https://www.math.unipd.it/~rcardin/swea/2021/Diagrammidisequenza_4x4.pdf;
- Design Pattern Creazionali:
https://www.math.unipd.it/~rcardin/swea/2021/DesignPatternCreazionali_4x4.pdf;
- Design Pattern Strutturali:
https://www.math.unipd.it/~rcardin/swea/2021/DesignPatternStrutturali_4x4.pdf;



- Design Pattern Comportamentali:
https://www.math.unipd.it/~rcardin/swea/2021/DesignPatternComportamentali_4x4.pdf.

1.3.2 Informativi

- **GLOSSARIO**: per la definizione dei termini (pedice G) e degli acronimi (pedice A) evidenziati nel documento;
- Capitolato d'appalto C5-PORTACS:
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C5.pdf>
- Software Engineering - Iam Sommerville - 10th Edition.
- Angular:
<https://angular.io/>;
- Node.js:
<https://nodejs.org/en/>;
- PrimeNG:
<https://www.primefaces.org/primeng/>;
- Java:
<https://www.java.com/it/>;
- Spring:
<https://spring.io/>;
- Docker:
<https://www.docker.com/>.



2 Tecnologie e librerie

2.1 Server

2.1.1 Tecnologie

- **Java**
- **Json**
- **Docker**
- **Gradle**

2.1.2 Librerie e Framework

- **Spring**
- **Gson**
- **Junit**
- **Mockito**

2.2 Client

2.2.1 Tecnologie

- **Node.js**
- **HTML**
- **CSS**
- **Typescript**

2.2.2 Librerie e Framework

- **Angular**
- **PrimeNG**
- **Libreria di test1**
- **Libreria di test2**

2.3 Version Control System e Continuous Integration

2.3.1 Git e gitflow

Git è un sistema di controllo per il versionamento veloce ed efficiente. Gitflow è un workflow che aiuta lo sviluppo software dando delle linee guida sui branch che strutturano le repo e le operazioni per l'implementazione di feature e rilascio di releases.

Maggiori informazioni:

- **Git:** <https://git-scm.com/>;
- **gitflow:** <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>.



2.3.2 GitHub

GitHub è un provider di hosting internet per lo sviluppo di software e il controllo della versione utilizzando Git. Fornisce un intero ecosistema di strumenti (version control, issue tracking, project boards, continuous integration e delivery...) e permette la creazione di account personali o di organizzazioni.

- **Maggiori informazioni su GitHub:** <https://github.com/about>;
- **Three Way Milkshake su GitHub:** <https://github.com/Three-Way-Milkshake>.

2.3.3 GitHub Actions

È uno strumento integrato in ogni repo di GitHub, permette di creare singole attività combinabili al fine di realizzare complessi workflow personalizzati. Puoi creare le tue azioni o utilizzare e personalizzare le azioni condivise dalla community di GitHub. Si possono creare azioni da zero, pubblicarne di nuove o utilizzare il vasto catalogo di automazioni già realizzate dalla community.

- **Documentazione:** <https://docs.github.com/en/actions>.



3 Setup

3.1 Requisiti di sistema

3.1.1 Requisiti Hardware

3.1.2 Requisiti Software

3.2 Installazione



4 Testing

4.1 JUnit

4.2 Libreria test frontend



5 Architettura del sistema

Qui si potrebbe mettere uno schema simile a quello della slide iniziale per evidenziare l'architettura client-server

5.1 Server

Dire che è 3 layer architecture

Qui potrebbe esserci il diagramma minimale complessivo

5.1.1 Diagramma delle classi

5.1.1.1 Persistence layer

5.1.1.2 Business layer

5.1.1.2.1 Mappa

5.1.1.2.2 Clients

5.1.1.2.3 Tasks

5.1.1.2.4 Collisioni

5.1.1.3 Communication layer

5.2 Client

5.3 Comunicazione

5.3.1 Diagrammi di sequenza

5.3.2 Protocollo di comunicazione



6 Estendere PORTACS

6.1 Algoritmo alternativo per il path finding

6.2 Introdurre nuove tipologie di utenti

6.2.1 Lato server

6.2.2 Lato client

6.3 Implementare tipi di persistenza alternativi

6.4 Modificare handler nell'algoritmo di gestione delle collisioni