



>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



>Three Way
Milkshake_

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	5
1.1	Scopo del documento	5
1.2	Scopo del prodotto	5
1.3	Riferimenti	5
1.3.1	Normativi	5
1.3.2	Informativi	6
2	Tecnologie e librerie	7
2.1	Server	7
2.1.1	Tecnologie	7
2.1.2	Librerie e Framework	7
2.2	Client	7
2.2.1	Tecnologie	7
2.2.2	Librerie e Framework	7
2.3	Versionamento e Continuous Integration	7
3	Setup	8
3.1	Requisiti di sistema	8
3.1.1	Requisiti Hardware	8
3.1.2	Requisiti Software	8
3.2	Installazione	8
4	Testing	9
4.1	JUnit	9
4.2	Libreria test frontend	9
5	Architettura del sistema	10
5.1	Server	10
5.1.1	Diagramma delle classi	10
5.1.1.1	Persistence layer	10
5.1.1.2	Business layer	10
5.1.1.2.1	Mappa	10
5.1.1.2.2	Clients	10
5.1.1.2.3	Tasks	10
5.1.1.2.4	Collisioni	10
5.1.1.3	Communication layer	10
5.2	Client	10
5.3	Comunicazione	10
5.3.1	Diagrammi di sequenza	10
5.3.2	Protocollo di comunicazione	10
6	Estendere PORTACS	11
6.1	Algoritmo alternativo per il path finding	11
6.2	Introdurre nuove tipologie di utenti	11
6.2.1	Lato server	11
6.2.2	Lato client	11
6.3	Implementare tipi di persistenza alternativi	11
6.4	Modificare handler nell'algoritmo di gestione delle collisioni	11



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 Versionamento e Continuous Integration

- **GitHub**
- **GitHub Action**



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