



# >Three Way Milkshake\_

---

## Verbale esterno 8

### 13 Aprile 2021

---

#### Three Way Milkshake - Progetto "PORTACS"

threewaymilkshake@gmail.com

<b>Versione</b>	1.0.0
<b>Stato</b>	Approvato
<b>Uso</b>	Esterno
<b>Approvazione</b>	Chiarello Sofia
<b>Redazione</b>	Greggio Nicolò
<b>Verifica</b>	Zuccolo Giada
<b>Destinatari</b>	Three Way Milkshake Prof. Vardanega Tullio Prof. Cardin Riccardo

#### **Descrizione**

Verbale del meeting del 2021-04-13 del gruppo Three Way Milkshake con il  
proponente Sanmarco Informatica.



## Registro delle modifiche

Vers.	Descrizione	Data appr.	Approvazione
1.0.0	Approvazione del verbale	data-appr	Chiarello Sofia

Vers.	Descrizione	Redazione	Data red.	Verifica	Data ver.
0.1.0	Stesura e verifica del verbale	Greggio Nicolò	2021-04-15	Zuccolo Giada	data-ver



## Indice

<b>1</b>	<b>Informazioni generali</b>	<b>3</b>
1.1	Dettagli sull'incontro . . . . .	3
1.2	Ordine del giorno . . . . .	3
<b>2</b>	<b>Verbale della riunione</b>	<b>4</b>
2.1	Facades . . . . .	4
2.2	Singletons . . . . .	4
2.3	Observer . . . . .	4
2.4	MVC nel client . . . . .	4
2.5	Ulteriori note . . . . .	4
<b>3</b>	<b>Tracciamento temi affrontati</b>	<b>5</b>



# 1 Informazioni generali

## 1.1 Dettagli sull'incontro

- **Luogo:** Incontro telematico tramite piattaforma Google Meet;
- **Data:** 2021-04-13;
- **Ora di inizio:** 15:15;
- **Ora di fine:** 16:10;
- **Partecipanti interni:** (5/6)
  - Chiarello Sofia;
  - Crivellari Alberto;
  - De Renzis Simone;
  - Greggio Nicolò;
  - Zuccolo Giada.
- **Partecipanti esterni:** (1)
  - Beggiato Alex (Sanmarco Informatica).

## 1.2 Ordine del giorno

La riunione prevede la discussione con il proponente delle criticità emerse nella Product Baseline riguardanti la nostra architettura, in particolare:

- uso del pattern facade per creare unici punti di interazione fra livelli diversi;
- presenza di diversi singleton gestiti manualmente;
- uso di observer strutturato da noi;
- identificazione del pattern mvc nella struttura dei client.



## 2 Verbale della riunione

### 2.1 Facades

Si è deciso di rimuovere i facade presenti nell'architettura, i quali centralizzavano l'accesso rispettivamente alle liste di client e alla persistenza. Al loro posto si faranno fluire le dipendenze verso le interfacce o classi necessarie direttamente.

### 2.2 Singletons

Si è deciso di adottare Spring Boot per la gestione delle dipendenze e risolvere così il problema della gestione manuale dei singleton ed i conseguenti problemi correlati allo unit testing di tali componenti. Questo influenza i componenti:

- WarehouseMap;
- UsersList;
- ForkliftsList;
- TasksSequencesList.

### 2.3 Observer

Dopo il confronto con il proponente e la decisione congiunta di adottare Spring Boot per gestire le dipendenze ed eliminare i singleton, non si è rilevata alcuna criticità nel mantenere il pattern observer per la mappa, per cui rimarrà tale.

### 2.4 MVC nel client

Il gruppo approfondirà internamente come rivedere l'architettura del client e il pattern dichiarato per sanare la criticità.

### 2.5 Ulteriori note

Il proponente ha sollevato i seguenti punti:

- l'introduzione di Spring Boot per la sola gestione delle dipendenze dovrebbe essere abbastanza rapida;
- Spring è uno strumento molto ampio e potente, iniziare a sfruttare altre caratteristiche arrivati a questo punto potrebbe essere troppo costoso;
- Spring semplifica un po' la codifica, ma i costrutti di base rimangono gli stessi;
- riferimenti a lavori dei gruppi dello scorso anno che hanno avuto Sanmarco Informatica come proponente:
  - [Red Round Robin](#);
  - [Heptagon](#).



### 3 Tracciamento temi affrontati

Codice	Domanda	Risposta
VE_6.1	Facade	Rimuovere
VE_6.2	Singletons	Utilizzare Spring Boot per gestire le dipendenze
VE_6.3	Observer	Mantenere, non ci sono problemi, è adatto
VE_6.4	MVC nel client	Provare a rivedere architettura, eventualmente non chiamarlo mvc dato che non lo è strettamente, ma mostrare solo come abbiamo strutturato le parti