



## Verbale interno del 31/05/2019

*Gruppo Cyber13 - Progetto P2PCS*

### Informazioni sul documento

<b>Versione</b>	1.0.0
<b>Data Redazione</b>	31/05/2019
<b>Responsabile</b>	Ilaria Rizzo
<b>Redazione</b>	Elena Pontecchiani
<b>Verifica</b>	Fabio Garavello
<b>Approvazione</b>	Ilaria Rizzo
<b>Uso</b>	Interno
<b>Destinatari</b>	Cyber13 Prof. Tullio Vardanega Prof. Riccardo Cardin
<b>Mail di contatto</b>	swe.cyber13@gmail.com

## Diario delle modifiche

Vs	Data	Descrizione	Autore	Ruolo
1.0.0	31/05/2019	Approvazione del verbale per rilascio in RQ	Ilaria Rizzo	Responsabile
0.1.0	31/05/2019	Verifica con esito positivo del verbale	Fabio Garavello	Verificatore
0.0.1	31/05/2019	Creazione scheletro e redazione del verbale	Elena Pontecchiani	Programmatore

## Indice

<b>1</b>	<b>Informazioni sulla riunione</b>	<b>3</b>
<b>2</b>	<b>Resoconto</b>	<b>3</b>
<b>3</b>	<b>Conclusioni e decisioni prese</b>	<b>4</b>

## 1 Informazioni sulla riunione

- **Luogo della riunione:** LabTA, Torre Tullio Levi-Civita, Via Trieste 63, Padova;
- **Data della riunione:** 31 Maggio 2019, dalle ore 08.00 alle ore 12.00;
- **Partecipanti della riunione:**
  - Daniel Mirel Bira;
  - Casagrande Andrea;
  - Garavello Fabio;
  - Pontecchiani Elena;
  - Rizzo Ilaria;
  - Squeri Matteo.
- **Ordine del giorno:** Il team ha indetto una riunione interna con lo scopo di aggiornare il responsabile di progetto Ilaria Rizzo in merito allo svolgimento delle attività previste.  
Nello specifico la riunione si è improntata sull'utilizzo dei design pattern da integrare al codice e della conseguente riorganizzazione delle classi (sia per quanto riguarda il front-end sia per il back-end).

## 2 Resoconto

La riunione si è improntata sulla riorganizzazione delle classi in base ai design pattern che si sono decisi di integrare.

Nello specifico, per quanto riguarda il front-end:

- Si è deciso di organizzare ogni  $activity_{|g|}$  del front-end secondo il pattern Model-View-Presenter. Per questo sarà necessaria una ristrutturazione dei metodi, soprattutto alla luce del fatto che la View non deve avere conoscenza diretta in merito al Model (cosa che alla data attuale non è rispettata, dal momento che i metodi della parte che verrà strutturata come View richiedono esplicitamente il  $context_{|g|}$ );
- Per la creazione di liste con un numero di elementi variabili di elementi si è scelto il pattern  $Adapter_{|g|}$  per adattare la View dinamicamente al numero di elementi.

Per quanto riguarda il back-end i file PHP ospitati nel server web sono stati strutturati secondo il modello API-REST. Pertanto i file sono suddivisi in due cartelle, in linea con il modello.

### 3 Conclusioni e decisioni prese

ID Decisione	Descrizione
VER-2019-05-31.01	Si è deciso di adottare il pattern MVP per le <i>activity</i> <sub> g </sub> per quanto riguarda la parte front-end
VER-2019-05-31.02	Si è deciso di adottare il pattern Adapter per l'implementazione della <i>scroll-view</i> <sub> g </sub> per quanto riguarda la parte front-end
VER-2019-05-31.03	Per implementare il pattern MVP per ogni activity, ogni file verrà scorporato in due diversi file, uno per la View e uno per il Presenter
VER-2019-05-31.04	La nomenclatura per i file scorporati sarà Nome-activity-originalePresenter per la parte Presenter e Nome-activity-originaleView per la parte della View
VER-2019-05-31.05	I file PHP vengono scorporati in due cartelle, in modo da seguire l'architettura API-REST