



## **Verbale esterno - 18/05/2023**

---

Arena Ivan Antonino  
Baesso Nicola  
Bousapnameme Ruth Genevieve  
Calabrese Luca

Garon Martina  
Liva Noemi  
Marchiante Marco

## **Progetto Ingegneria del Software**

Dipartimento di Matematica  
**Università degli Studi di Padova**



## Indice

<b>1</b>	<b>Informazioni della riunione</b>	<b>2</b>
<b>2</b>	<b>O.D.G - Ordine del Giorno</b>	<b>2</b>
<b>3</b>	<b>Svolgimento</b>	<b>2</b>
3.1	Mantenimento e condivisione dello stato tra <i>client</i> e <i>server</i> ed utilizzo del pattern MVC . .	2
3.2	JAPI in diagramma UML . . . . .	2
<b>4</b>	<b>Risultati</b>	<b>3</b>

## 1 Informazioni della riunione

**Presenti:**

- Marchiante Marco
- Arena Ivan Antonino
- Bousapnameme Ruth Genevieve
- Calabrese Luca
- Garon Martina

**Assenti:**

- Baesso Nicola
- Liva Noemi

**Luogo:** Meeting Zoom.**Orario:** 8:50 - 9:10.**Partecipanti esterni:** Riccardo Cardin**Responsabile:** Martina Garon

## 2 O.D.G - Ordine del Giorno

1. Mantenimento e condivisione dello stato tra *client* e *server* ed utilizzo del pattern MVC;
2. JAPI in diagramma UML.

## 3 Svolgimento

La riunione è iniziata puntuale alle 8.50 e siamo partiti con l'esposizione dei due punti.

### 3.1 Mantenimento e condivisione dello stato tra *client* e *server* ed utilizzo del pattern MVC

Abbiamo esposto che l'applicazione è divisa in due parti: una API REST nel *back-end* ed un'applicazione React nel *front-end*.

Il prof. Cardin si è accertato che lo stato dell'applicazione sia nel *server*, cosa che abbiamo confermato. Inoltre, conferma che il pattern MVC deve rimanere sullo stesso lato della comunicazione (o *client* o *server*).

Ipotizziamo che si sia creato un fraintendimento tra "mantenimento dello stato" ed il fatto che il *pattern* MVC è nel *client*.

Nel caso specifico, l'unico invio di dati dal *client* al *server* è l'invio dei dati di *login*. Durante questo invio, vengono inviati sia i dati richiesti dal *form* di *login* che la soluzione del CAPTCHA.

Inoltre, la validazione dei dati avviene sia lato *client* che *server*. Lo stato è comunque mantenuto interamente dal *server*.

Secondo il prof. Cardin, non è chiara la dinamica di una possibile situazione critica. Si deve comunque rammentare che il *browser* è un'ambiente insicuro e non si può presumere nulla. Per il progetto, comunque, ci si aspetta un livello di sicurezza adeguato ai nostri studi.

Il prof. Cardin non è in grado di valutare la criticità, poiché la critica mossa è vaga. Lui consiglia nuovamente un confronto con Zucchetti, per chiarire esattamente il punto critico o fornire un esempio concreto.

### 3.2 JAPI in diagramma UML

Nel progetto, viene utilizzata JAPI come *repository*. Essa può essere inserita nel diagramma UML, ma la si deve evidenziare come "non facente parte del nostro codice". L'evidenza si fa nel diagramma, marcando la JAPI con un colore diverso.

## 4 Risultati

Il prof. Cardin sostanzialmente non ha avuto materiale sufficiente per valutare la criticità di sicurezza. Comunque, condivide l'architettura che stiamo presentando, agli scopi preposti. Pertanto, abbiamo deciso di chiarire il fraintendimento con Zucchetti, inviando una *email* a Gregorio.