**Considerazioni front-end e back-end webApp**

## **Decisione del framework front-end**

Per lo sviluppo front-end della webApp il team ha considerato i tre principali frameworks: Angular, React e Vue. Questi frameworks servono a facilitare lo sviluppo lato client della webApp e le principali caratteristiche sono riportate nella seguente tabella:

|  |  |  |  |
| --- | --- | --- | --- |
| Caratteristiche | Angular | React | Vue |
| Manipolazione UI/DOM | Sì | Sì | Sì |
| Gestione stati | Sì | Parziale | Sì |
| Routing | Sì | No | Sì |
| Gestione e validazione form | Sì | No | No |
| Http Client | Sì | No | No |

Dopo aver fatto attente ricerche su queste tecnologie e anche da quanto è emerso dal documento “Verbale\_29\_11\_21”, il team Oberon ha deciso di utilizzare Vue che ha come punti di forza la semplicità e una curva di apprendimento molto meno ripida rispetto a Angular e React, tuttavia riesce a soddisfare pienamente i requisiti tecnici necessari per la buona riuscita della webApp.

Per quanto riguarda la comunicazione tra webApp e blockchain si è deciso di utilizzare web3.js a discapito di Avalanche.js in quanto la prima è una libreria molto diffusa, compatibile con la C-chain di Avalanche e supportata da una grande community, la seconda invece è una libreria più giovane e ancora non completa al 100%.

**Stack back-end**

Per l’implementazione della parte back-end si è deciso, insieme al proponente, di utilizzare il framework Spring del linguaggio Java, in particolare all’interno di questo framework verranno utilizzate principalmente le librerie Spring Core e Spring Data per una gestione dati di tipo REST.