

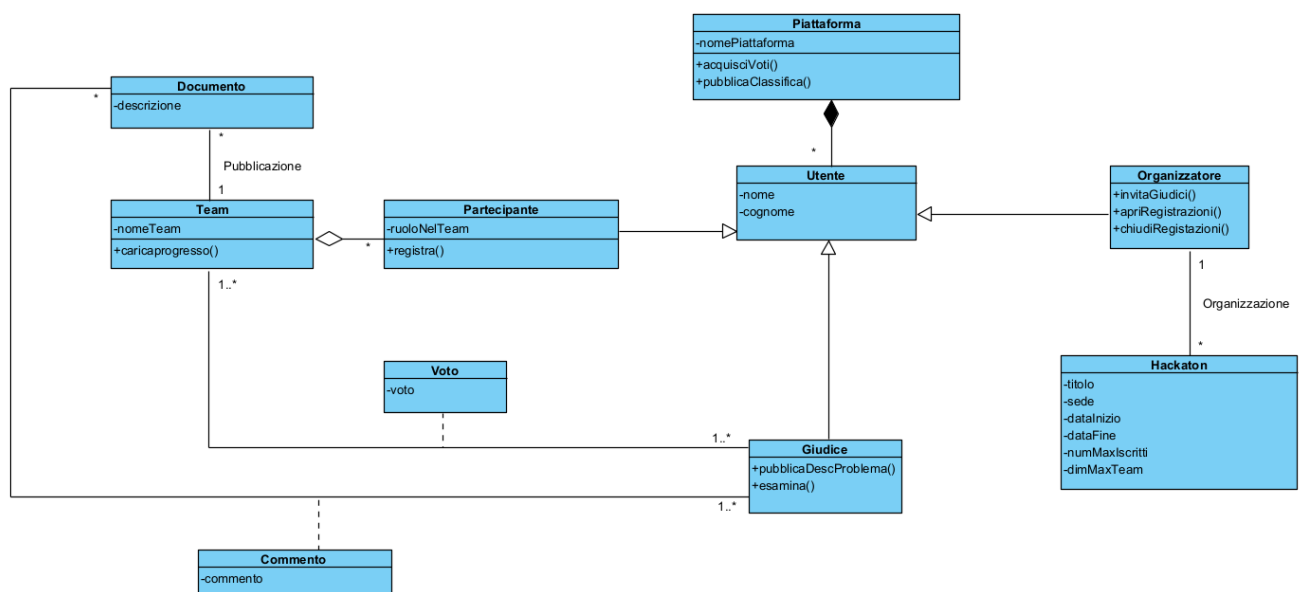
Federico Padulano N86005043

Fabrizio Padulano N86005058

## Codice

<https://github.com/fabpadulano-unina/progetto-hackaton>

## Diagramma UML



## Ragioni scelta classi

Hackaton: rappresenta l'evento, entità principale, come attributi ha titolo, sede, date di inizio/fine, numero massimo di iscritti e dimensione massima dei team che definiscono l'evento. L'evento è organizzato da un Organizzatore per cui si definisce una classe organizzatore responsabile di creare l'evento, invitare giudici, aprire e chiudere le iscrizioni. E' presente un'associazione tra Organizzatore e Hackaton chiamata Organizzazione. Un organizzatore può organizzare più Hackaton, un Hackaton è organizzato da un solo organizzatore. L'organizzatore da testo deve essere iscritto alla piattaforma. Conviene dunque definire una classe Piattaforma formata da utenti che si occupa di pubblicare classifiche e acquisire i voti. Poiché è formata da utenti, definiamo allora una classe Utente che ha come attributi nome e cognome così da identificarlo. Essa è collegata alla piattaforma tramite una composizione (se eliminiamo la piattaforma ha senso eliminare tutti i suoi utenti, una piattaforma è formata da 0 o più utenti). In particolare, Utente è una superclasse per

Organizzatore, Giudice e Partecipante, poiché un utente registrato potrebbe assumere uno di questi ruoli. La classe Partecipante avrà come attributo ruoloNelTeam e può registrarsi all'hackaton. I partecipanti possono formare team. Team sarà una classe avente come attributo il nome del team. Il partecipante appartiene a un team tramite aggregazione. Periodicamente i team devono caricare gli aggiornamenti come progressi, dunque questa classe avrà come metodo caricaProgresso, il progresso verrà mostrato sottoforma di documento che sarà dunque una classe chiamata Documento, che come attributo avrà la descrizione del documento stesso. E' presente un'associazione "Pubblicazione" tra Documento e Team. Un documento è relativo ad un solo team e un team può pubblicare più progressi, dunque più documenti. Il documento verrà esaminato e commentato dai giudici. Definiamo quindi una classe chiamata Giudice che avrà il compito di esaminare i documenti e pubblicare la descrizione del problema dell'evento da affrontare. Alla fine dell'hackathon, ogni giudice assegna un voto a ciascun team, quindi è presente una classe di associazione tra Giudice e Team chiamata Voto, che ha come attributo il voto assegnato dal giudice. Un giudice può assegnare più voti a diversi team, un team può ricevere più voti da diversi giudici.