



- ▶ Progetto di Sistemi Distribuiti
- ▶ Alex Siroli - Alice Mastrilli

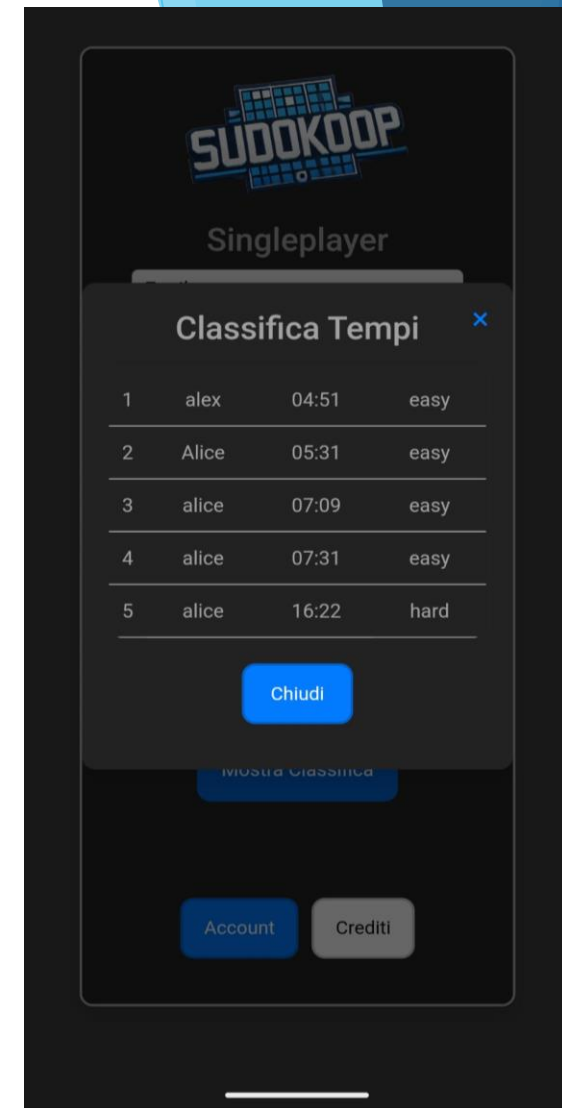
Obiettivo

- ▶ Realizzare un Sudoku online con funzionalità in tempo reale
- ▶ Modalità disponibili: giocatore singolo, cooperativa e sfida a squadre
- ▶ Interfaccia web responsive, compatibile anche con dispositivi mobili



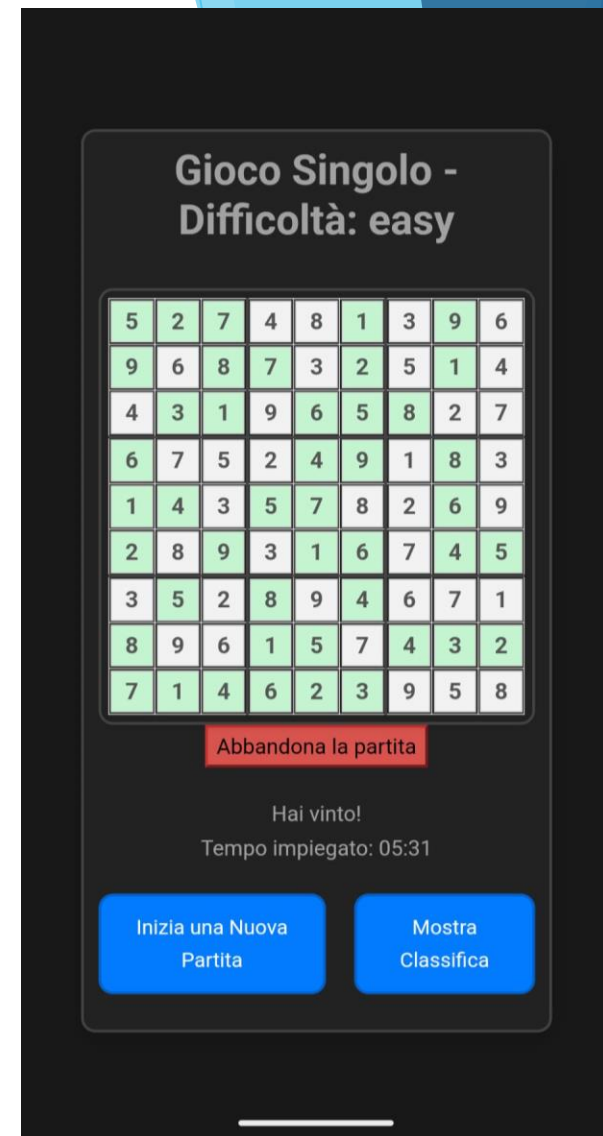
Funzionalità chiave

- ▶ Login/registrazione
- ▶ Creazione o ingresso in lobby (max 10)
- ▶ Chat in lobby
- ▶ Leaderboard single-player (ordinata per tempo di gioco)
- ▶ Statistiche personali (vittorie, sconfitte)



Single-Player

- ▶ 3 vite iniziali e timer sempre attivo
- ▶ A ogni errore si perde 1 vita e la cella diventa temporaneamente rossa
- ▶ All'inserimento di un numero corretto, la cella diventa verde
- ▶ Si vince se si completa il sudoku con almeno una vita rimasta



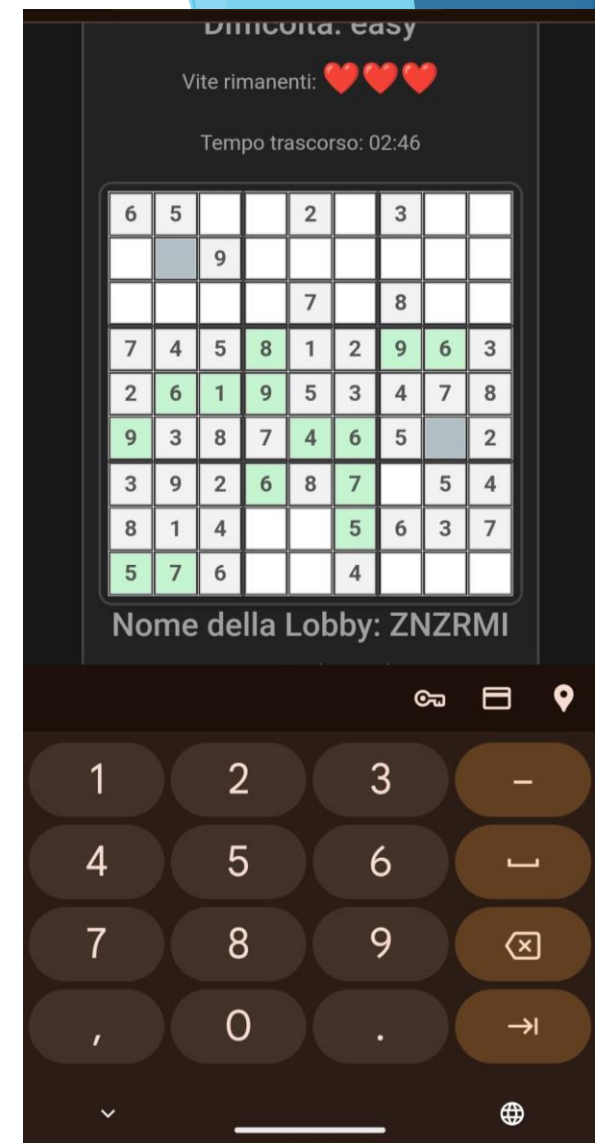
Lobby

- ▶ Un giocatore crea la lobby e sarà il master
- ▶ Viene generato un codice della lobby da comunicare ad altri
- ▶ Altri giocatori possono unirsi inserendo il codice univoco
- ▶ I giocatori in lobby possono comunicare utilizzando una chat apposita
- ▶ Il master sceglie la difficoltà e la modalità di gioco (almeno 2 giocatori in lobby)



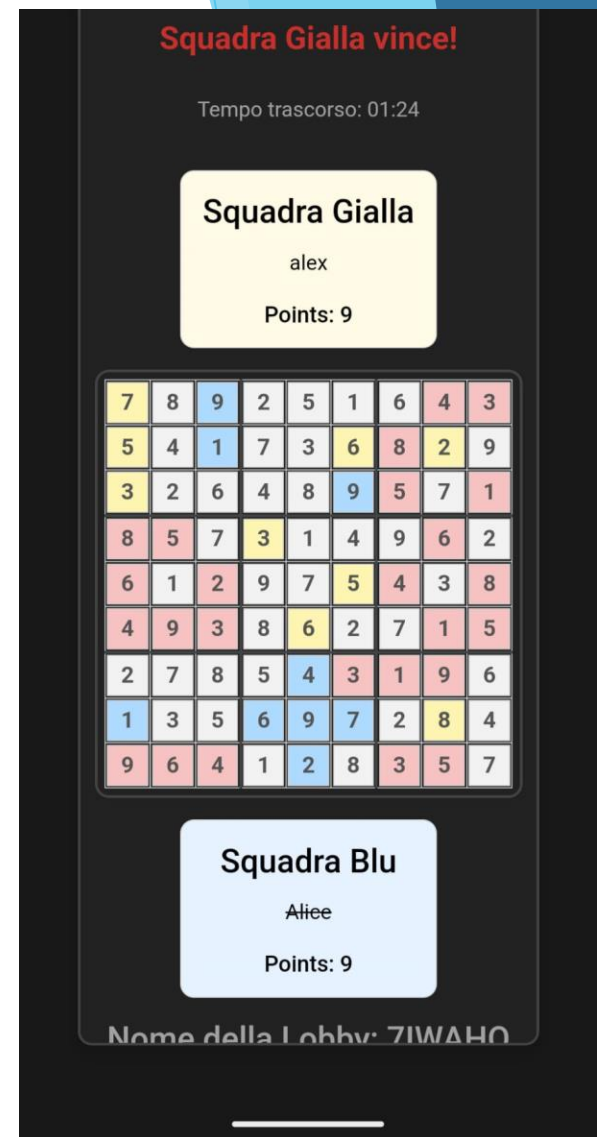
Coop (multiplayer)

- ▶ Stesso sudoku per tutti con 3 vite condivise
- ▶ Stessa logica di gioco del Single-Player
- ▶ Alla selezione di una cella, questa viene evidenziata per tutti



Versus (multiplayer)

- ▶ Prima di giocare, i giocatori devono dividersi in due squadre: blu e gialla
- ▶ Non ci sono vite. Se un giocatore inserisce un numero sbagliato viene eliminato
- ▶ Se viene inserito un numero corretto, la squadra guadagna un punto e la cella si colora del colore di quella squadra
- ▶ Alla selezione di una cella, questa viene evidenziata del colore di quella squadra
- ▶ Vince chi ha più punti o l'unica squadra rimasta



Architettura

- ▶ MEVN: MongoDB, Express.js, Vue.js, Node.js
- ▶ REST API per auth e single-player
- ▶ Socket.IO per lobby, chat, multiplayer



Testing & Deployment

- ▶ Unit test con Jest su controller e modelli
- ▶ Test manuali con utenti reali per verificare ritardi e sincronizzazione
- ▶ Utilizzo di Docker per ambiente di sviluppo e distribuzione



Sviluppi futuri

- ▶ Nuove modalità di gioco, ad esempio sfida contro un bot
- ▶ Creazione di tornei senza limiti di giocatori
- ▶ Rendere l'esperienza dell'utente più personalizzabile, ad esempio consentendogli di modificare il proprio username dopo la registrazione o di scegliere un colore personalizzato.



Demo

Login

[Non hai un account? Registrati](#)

