

PROGETTO PROGRAMMAZIONE JAVA

Gioco di carte: 7 e mezzo

Anno Accademico 2017-2018

Angel Chelaru

# ANALISI

## Gioco di carte: 7 e mezzo

Il progetto consiste nella realizzazione del celebre gioco di carte *7 e mezzo*, in cui il giocatore (l'utente) sfida il mazziere (il computer).

Il progetto è composto da:

- Server: non ha interfaccia e si occupa di ricevere e conservare i vari messaggi di log;
- Client: ha interfaccia e serve a realizzare il gioco effettivo, la tabella delle classifiche e il diagramma a torta sugli utenti più assidui.

## MOCKUP

Campo Username

START

Carta

Mazzo

+

✓

Carta

Carta

Carta

**CLASSIFICA**

Username	Punteggio
Username	10
Username	9
Username	8
Username	7
Username	6
Username	5
Username	4
Username	3
Username	2
Username	1

**Giocatori Assidui**

User

User

User

User

## Vista Dinamica

1. All'avvio dell'applicazione, il sistema visualizza l'username dell'ultimo giocatore e lo stato dell'eventuale gioco non finito in precedenza, se l'ultimo accesso dell'utente risale a meno di un certo numero di ore.
2. L'utente inserisce il suo username, se non è già stato inserito automaticamente all'avvio, e poi preme il tasto START.
3. Il Sistema mostra una carta scoperta all'utente e una carta coperta al mazziere.
4. A questo punto l'utente sceglie se ricevere una o più carte in più cliccando il pulsante + oppure fermarsi pigliando il pulsante ✓.

**IF** Il Giocatore ha superato il punteggio di 7 e mezzo **THEN**

Il Mazziere ha vinto.

**ELSE**

**WHILE** Il punteggio del Mazziere è inferiore a quello del giocatore **THEN**

Il Mazziere riceve una nuova carta coperta.

**IF** Il Mazziere supera 7 e mezzo **THEN**

Il Mazziere perde e vince il Giocatore.

Vengono mostrate le carte del mazziere.

**ELSE IF** Il Mazziere supera il Giocatore **THEN**

Il Mazziere vince.

Vengono mostrate le carte del mazziere.

5. L'username, il punteggio del giocatore e del mazziere e l'esito della partita vengono archiviati.
6. Dopo 5 secondi, il Sistema rimuove tutte le carte mostrate all'utente e al mazziere.
7. L'utente può pigiare il pulsante START per iniziare a giocare nuovamente.

## Cache Locale

Se il gioco è in corso e l'applicazione viene chiusa, verrà salvato nella lo stato attuale della partita. All'avvio dell'applicazione, se l'utente ha chiuso l'applicazione non più di un certo numero di ore precedentemente, viene ripristinato lo stato del gioco del rispettivo utente.

## Archivio

Il Sistema archivia alla fine di ogni gioco:

1. L'username del giocatore
2. Il punteggio finale del giocatore
3. Il punteggio finale del mazziere
- 4.

## File di Log in XML

Viene inviata una riga di log per i seguenti eventi.

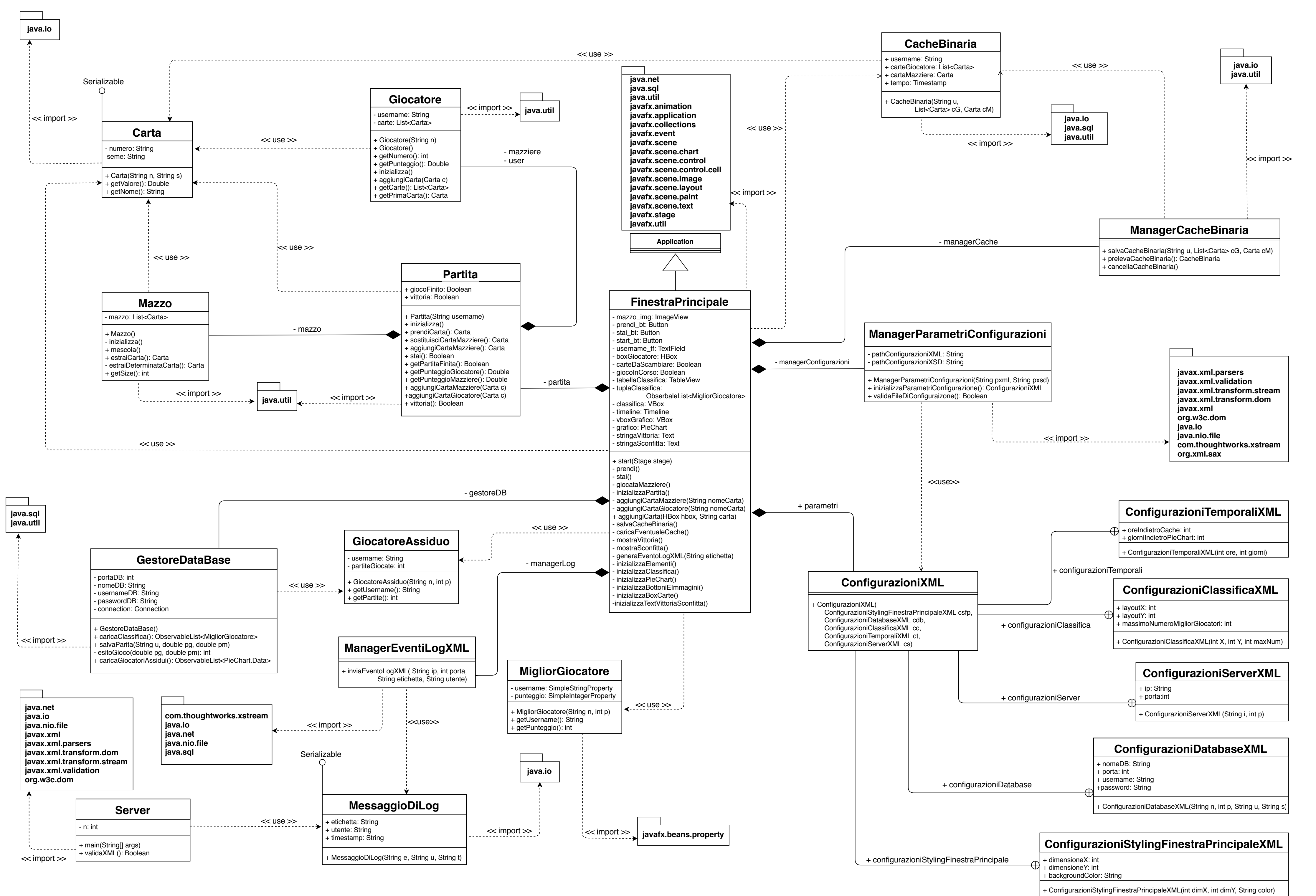
1. Avvio dell'applicazione ("APERTURA")
2. Pressione dei pulsanti "START", "STAI", "PRENDI"
3. Chiusura dell'applicazione ("CHIUSURA")

La riga di log contiene: indirizzo IP del client o il suo username, data-ora corrente e l'etichetta associata all'evento.

## File di Configurazione XML

All'avvio dell'applicazione, il Sistema legge dal file di configurazione i seguenti dati:

- font, dimensioni e colore background;
- stile della classifica e del diagramma a torta
- numero massimo di utenti mostrati nella classifica
- giorni da considerare all'indietro per prelevare i dati per la classifica e per il diagramma a torta
- IP e porta del database
- IP e porta del server di log



# PROGETTAZIONE

Classe *Carta*: rappresenta una delle 40 carte del mazzo; ogni carta è distinta da un numero e un seme.

Classe *Mazzo*: contiene le 40 carte che verranno usate in una partita; può mescolare le carte, estrarre la prima carta dal mazzo oppure una carta specifica.

Classe *Giocatore*: rappresenta uno dei due giocatori(mazziere e user); mantiene le informazioni delle carte che sono state estratte.

Classe *Partita*: gestisce gli eventi della partita: inizializza un mazzo, gestisce l'estrazione di una carta e determina la fine della partita.

Classe *FinestraPrincipale*: è la GUI del gioco e contiene, oltre agli elementi per la realizzazione effettiva del gioco, la tabella con la classifica dei migliori giocatori e il grafico a torta con le statistiche dei giocatori più assidui.

Classe *ManagerParametriConfigurazioni*: legge i parametri contenuti all'interno del file Configurazioni.xml, invoca la validazione dei dati contenuti, deserializza il contenuto XML come oggetto Java, inizializza la classe ConfigurazioniXML con i parametri letti.

Classe *ConfigurazioniXML*: mette a disposizione i parametri di sistema contenuti all'interno del file Configurazioni.xml .

Classe *MigliorGiocatore*: è la classe bean per la tabella della classifica dei miglior giocatori.

Classe *GiocatoreAssiduo*: è la classe che funge da struttura dati per passare i dati al diagramma a torta.

Classe *GestoreDataBase*: interagisce con il database MySQL e restituisce la classifica dei migliori giocatori e i dati del grafico a torta riguardo i giocatori più assidui.

Classe *CacheBinaria*: viene usata come classe serializable per salvare la cache della partita in corso.

Classe *ManagerCacheBinaria*: salva la cache della partita in corso su un file binario, la legge, la restituisce e la cancella dal file.

Classe *MessaggioDiLog*: viene usata come classe serializable per inviare i messaggi di log in formato xml al server.

Classe *ManagerEventiLogXML*: invia i messaggi di log al server.

Classe *Server*: riceve i messaggi di log, invoca la validazione dei dati e li salva sul file FileLogs.txt .

# MANUALE D'USO

Per il collaudo seguire i seguenti passi.

1. Scrivere l'username di un utente presente nella classifica (es. "l.lupi") e premere il pulsante START. (Figura 1)
2. In base alla carta uscita, decidere se premere il pulsante + ed estrarre un'altra carta (senza però superare il punteggio di 7.5, altrimenti si perde), o se premere il pulsante ✓ per restare. (Figura 2)
3. Non appena si decide di rimanere, la carta del mazziniere viene scoperta (Figura 3) e il mazziniere inizierà ad estrarre finché non si supera o eguaglia il punteggio dell'utente. (Figura 4)
4. Infine apparirà la scritta "Hai vinto!" o "Hai perso!" per indicare la fine della partita. (Figure 5 e 6)
5. Per iniziare una nuova partita, premere START e tornare al punto 2.

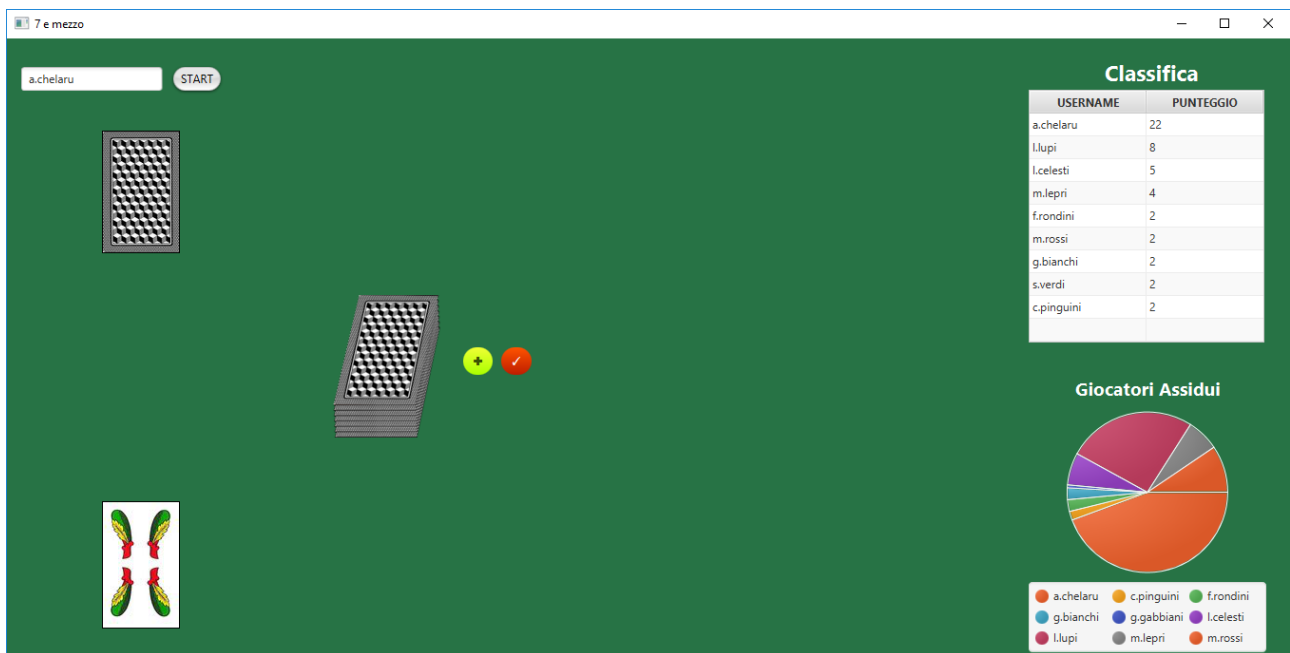


Figura 1



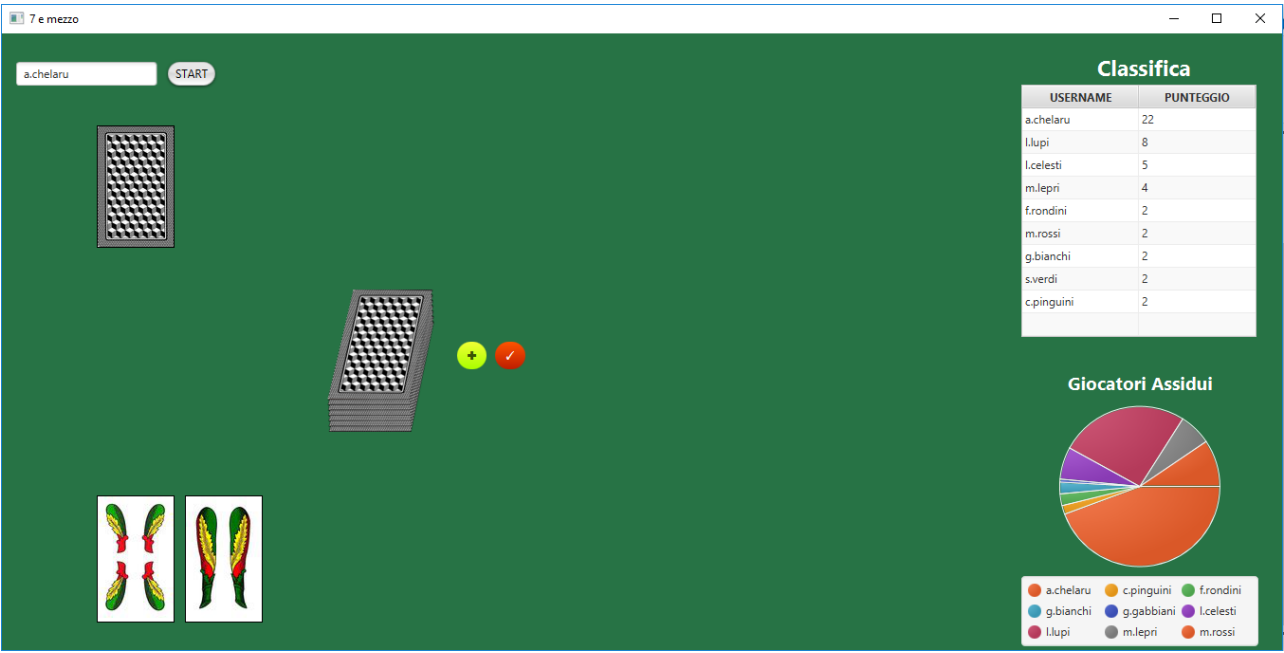


Figura 2

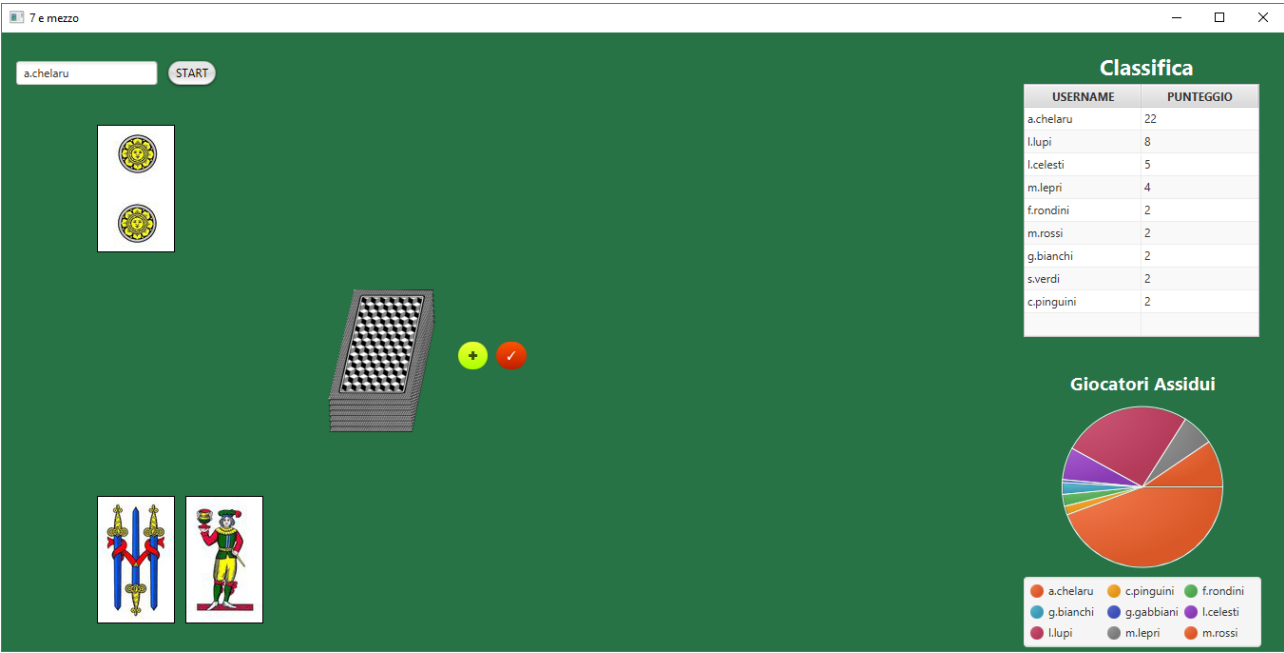


Figura 3



Figura 4

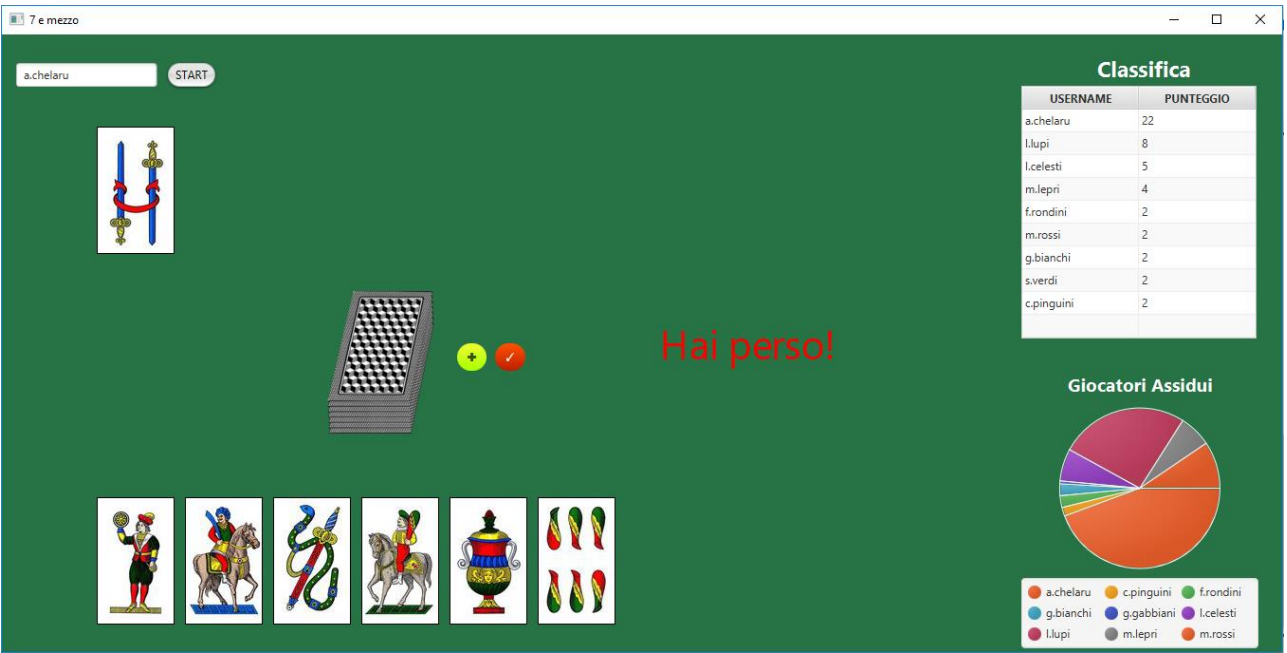


Figura 5

## File di configurazione XML

L'applicazione utilizza un file di configurazione in formato XML per inizializzare vari aspetti:

- le configurazioni per determinare alcuni parametri della GUI, come le dimensioni della finestra e il colore di background;
- le configurazioni del database;
- le configurazioni dello stile della tabella;
- le configurazioni temporali, le quali indicano per quanto mantenere la cache e per quanti giorni all'indietro considera i dati per il grafico dei giocatori assidui;
- le configurazioni del server;

```
<?xml version="1.0"?>
- <ConfigurazioniXML>
  - <configurazioniStylingFinestraPrincipale>
    <dimensioneX>1350</dimensioneX>
    <dimensioneY>650</dimensioneY>
    <backgroundColor>#277345</backgroundColor>
  </configurazioniStylingFinestraPrincipale>
  - <configurazioniDatabase>
    <nomeDB>setteemezzo</nomeDB>
    <porta>3306</porta>
    <username>root</username>
    <password/>
  </configurazioniDatabase>
  - <configurazioniClassifica>
    <layoutX>930</layoutX>
    <layoutY>30</layoutY>
    <massimoNumeroMigliorGiocatori>10</massimoNumeroMigliorGiocatori>
  </configurazioniClassifica>
  - <configurazioniTemporal>
    <oreIndietroCache>5</oreIndietroCache>
    <giorniIndietroPieChart>90</giorniIndietroPieChart>
  </configurazioniTemporal>
  - <configurazioniServer>
    <ip>127.0.0.1</ip>
    <porta>8080</porta>
  </configurazioniServer>
</ConfigurazioniXML>
```

## Server di Log

Alla pressione di un pulsante qualsiasi e alla apertura e chiusura dell'applicazione, viene inviata al server una stringa con il nome dell'azione svolta, il nome dell'utente e il timestamp in formato XML. Il server riceve il messaggio di log, lo convalida da un file in formato XSD e poi lo salva in un file di testo.

```
1) ricevuto:
<MessaggioDiLog>
  <etichetta>APERTURA</etichetta>
  <utente>192.168.43.105</utente>
  <timestamp>2018-07-20 22:39:13.905</timestamp>
</MessaggioDiLog>
```

Validazione avvenuta con successo

```
2) ricevuto:
<MessaggioDiLog>
  <etichetta>START</etichetta>
  <utente>a.chelaru</utente>
  <timestamp>2018-07-20 22:42:08.96</timestamp>
</MessaggioDiLog>
```

Validazione avvenuta con successo

```
3) ricevuto:
<MessaggioDiLog>
  <etichetta>START</etichetta>
  <utente>a.chelaru</utente>
  <timestamp>2018-07-20 22:42:14.677</timestamp>
</MessaggioDiLog>
```

Validazione avvenuta con successo

```
4) ricevuto:
<MessaggioDiLog>
  <etichetta>PRENDI</etichetta>
  <utente>192.168.43.105</utente>
  <timestamp>2018-07-20 22:42:16.409</timestamp>
</MessaggioDiLog>
```

Da notare che all'apertura dell'applicazione viene salvato nel campo utente l'ip del computer da cui si è fatto l'accesso. Ciò perché ancora non sappiamo chi sarà l'utente che deciderà di giocare.

Tuttavia, se l'applicazione carica la cache in memoria e la partita che è stata interrotta viene

ripristinata, all'apertura verrà iniziato l'username dell'utente nel messaggio di log di apertura.

```
20) ricevuto:
<MessaggioDiLog>
  <etichetta>APERTURA</etichetta>
  <utente>a.chelaru</utente>
  <timestamp>2018-07-21 10:59:11.203</timestamp>
</MessaggioDiLog>

Validazione avvenuta con successo
```

## Struttura e contenuto del database

All'interno del database sono contenute due tabelle:

- la tabella “giocatori” che contiene la lista di tutti i giocatori salvati; ognuno è univocamente identificato dal suo username e per ognuno vengono salvati anche alcuni dati anagrafici;
- la tabella “partite” contengono l'username del giocatore, il timestamp della fine della partita, l'esito del gioco (1 per vittoria, 0 per sconfitta) e il punteggio sia del giocatore che del mazziere.

Result Grid

Filter Rows:

Edit:

Export/Import:

	Username	Nome	Cognome	DataDiNascita	Email	Sesso
	a.chelaru	Angel	Chelaru	1996-02-19	a.chela...	M
	c.pinguini	Chiara	Pinguini	1998-12-24	c.pingu...	F
	f.rondini	Francesca	Rondini	1989-10-21	f.rondi...	F
	g.bianchi	Giovanni	Bianchi	1990-11-30	g.bianc...	M
	g.gabbiani	Giada	Gabbiani	1992-09-11	g.gabbi...	F
	l.celesti	Luca	Celesti	1996-08-24	l.celesti...	M
	l.lupi	Laura	Lupi	1987-12-03	l.lupi@...	F
	m.lepri	Martina	Lepri	1994-07-16	m.lepri...	F
	m.rossi	Mario	Rossi	1997-03-22	m.rossi...	M
	s.verdi	Simone	Verdi	1993-01-14	s.verdi...	M
	NULL	NULL	NULL	NULL	NULL	NULL

Tabella “Giocatori”

Result Grid					
Filter Rows:					
Edit:					
Export/Import:					
	Username	OraFinePartita	Esito	PunteggioGiocatore	PunteggioMazziere
	l.celesti	2018-07-08 18:1...	1	6	9
	l.celesti	2018-07-08 18:1...	0	5	5
	l.celesti	2018-11-07 08:5...	1	7	11
	l.celesti	2018-12-07 08:5...	1	5	8.5
	l.celesti	2019-04-06 08:5...	0	6.5	12
	l.celesti	2019-04-07 08:5...	0	8	6
	l.lupi	2018-04-07 18:5...	1	6.5	8.5
	l.lupi	2018-05-09 21:5...	1	6.5	8
	l.lupi	2018-05-09 21:5...	1	7.5	8
	l.lupi	2018-05-09 21:5...	0	5.5	6.5
	l.lupi	2018-06-07 08:5...	0	8.5	3
	l.lupi	2018-06-14 12:1...	0	12	0.5
	l.lupi	2018-06-14 12:3...	0	13.5	0.5

Tabella "Partite"

La decisione dell'esito viene fatta nel lato server attraverso un trigger che decide l'esito della partita.

```

86
87 • DROP TRIGGER IF EXISTS determina_esito;
88
89 DELIMITER $$
90
91 • CREATE TRIGGER determina_esito
92 BEFORE INSERT ON partite
93 FOR EACH ROW
94
95 BEGIN
96     IF (NEW.PunteggioGiocatore > 7.5) THEN
97         SET NEW.esito = 0;
98     ELSEIF (NEW.PunteggioMazziere > 7.5) THEN
99         SET NEW.esito = 1;
100     ELSEIF(NEW.PunteggioGiocatore > NEW.PunteggioMazziere) THEN
101         SET NEW.esito = 1;
102     ELSE
103         SET NEW.esito = 0;
104     END IF;
105
106 END $$
107
108 DELIMITER ;
109
110

```