Uno

Alunos: Gabriel Dolzan, Luis Felipe da Silva e Murilo Goedert.

SUMÁRIO

1. INTRODUÇÃO

O sistema a ser desenvolvido neste trabalho será baseado no conhecido jogo “Uno”. O jogo será desenvolvido para mobile, exclusivamente para Android, tendo seu ***backend*** desenvolvido na linguagem Java e utilizando comunicação via ***Java Sockets***. As mensagens trocadas entre os ambientes seguem o formato ***JSON*** e se utiliza uma conhecida biblioteca para facilitar essa troca de mensagens, chamada de ***Google.GSON***.

Dentro do sistema, estará disponível o recurso de cadastro de usuários, possibilitando a manutenção de dados de perfil dos usuários e histórico dos jogos jogados. Após realizar seu cadastro, o usuário poderá visualizar os jogos disponíveis e selecionar aquele no qual ele deseja se conectar. Dentro da sala de jogo, irá exibir quais jogadores estão esperando para iniciar a partida, permitindo que o usuário defina se está pronto para começar. Há ainda a opção de criar uma sala, para poder convidar seus amigos para jogar, por exemplo.

Dentro do jogo, as regras se baseiam nas clássicas já conhecidas do jogo “Uno”, o jogo se desenrola em turnos nos quais os jogadores tem a opção de comprar ou descartar cartas, vide regras do jogo original. Além disso, há ainda a opção de “pedir Uno”, uma mecânica conhecida na qual o jogador indica ter somente uma carta em mãos e estar prestes a vencer.

Finalmente, ao terminar o jogo, serão exibidos resultados a respeito da partida jogada, indicando não só o vencedor, mas também os demais jogadores e algumas características do jogo que foi finalizado.

2. REQUISITOS FUNCIONAIS

Esta seção conta com os requisitos funcionais do projeto em nível de usuário, são eles que ditam o que será desenvolvido ao longo do semestre. Ressaltamos que os mesmos podem ser alterados ou incrementados durante o decorrer do projeto.

ID	Descrição	Versão	Data	Fonte	Desenvolvedor	Status
RF01	O sistema manterá dados de jogadores (nome, foto, estatísticas de jogo).	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado
RF02	O sistema manterá dados de quais jogadores se conectaram em cada servidor.	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado
RF03	O jogador poderá se conectar a um dos servidores disponíveis.	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado
RF04	O jogador, quando conectado a um servidor, verá o status (conectado ou desconectado) dos demais jogadores.	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado
RF05	O jogador competirá contra outros jogadores que estiverem conectados ao mesmo servidor.	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado
RF06	O sistema manterá uma sequência de turnos que define qual jogador deve jogar.	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado
RF07	O jogador, estando em seu turno, pode optar por descartar uma de suas cartas seguindo as regras do jogo.	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado
RF08	O jogador, estando em seu turno, pode optar por comprar uma carta do monte de cartas.	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado
RF09	O jogador, estando em seu turno, pode indicar que está	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado

	em “UNO”, seguindo as regras do jogo.					
RF10	O jogador pode interferir antes do seu rival indicar que está em “UNO” (RF09), seguindo as regras do jogo.	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado
RF11	O jogador deve visualizar quantas cartas cada um dos demais jogadores possuem.	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado
RF12	O sistema indicará informações do jogo quando o mesmo terminar.	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado
RF13	O jogador pode indicar que deseja uma revanche, iniciando um novo jogo com os mesmos oponentes.	1.0	28/08	Reunião com a equipe 28/08	Equipe	Elencado

3. MENSAGENS DO SISTEMA

Esta seção conta com as mensagens que serão trocadas entre os componentes atuantes no projeto, indicando seu conteúdo e resposta esperada (quando houver).

CRIAR PARTIDA

<table>
<tr><td>Conteúdo da Mensagem</td><td><table>
<tr><th>Campo</th><th>Conteúdo</th></tr>
<tr><td>operação</td><td>“create-match”</td></tr>
<tr><td>name</td><td>Nome da partida</td></tr>
<tr><td>qtdPlayers</td><td>quantidade máxima de jogadores na partida</td></tr>
</table></td></tr>
<tr><td>Descrição</td><td>Cria uma nova partida no servidor</td></tr>
<tr><td>Retorno</td><td>{
matchId: id da partida,
name: nome da partida,
qtdPlayers: quantidade máxima de jogadores,
status: status da partida (aguardando ou jogando)
}</td></tr>
</table>

BUSCA DE AVATARES

<table>
<tr><td>Conteúdo da Mensagem</td><td><table>
<tr><th>Campo</th><th>Conteúdo</th></tr>
<tr><td>operação</td><td>“get-avatars”</td></tr>
</table></td></tr>
<tr><td>Descrição</td><td>Busca no servidor uma lista de avatares</td></tr>
<tr><td>Retorno</td><td>[
{
id: id do avatar,
imageUrl: nome da constante do drawable no android
},
…
]</td></tr>
</table>

LISTAGEM DE PARTIDAS

<table>
<tr><td>Conteúdo da Mensagem</td><td><table><tr><th>Campo</th><th>Conteúdo</th></tr><tr><td>operação</td><td>“get-matches-list”</td></tr></table></td></tr>
<tr><td>Descrição</td><td>Busca no servidor a lista de partidas em execução, esta busca é feita de tempos em tempos enquanto o usuário não está definitivamente jogando.</td></tr>
<tr><td>Retorno</td><td>[
{
matchId: o Id da partida,
name: o nome da partida,
qtdPlayers: a quantidade máxima de jogadores na partida,
status: o estado da partida podendo ser “aguardando” ou “jogando”
},
…
]</td></tr>
</table>

INGRESSAR EM UMA PARTIDA

<table>
<tr><td>Conteúdo da Mensagem</td><td><table><tr><th>Campo</th><th>Conteúdo</th></tr><tr><td>operação</td><td>“join-match”</td></tr><tr><td>userId</td><td>O id do próprio usuário</td></tr><tr><td>matchId</td><td>O id da partida que se quer ingressar</td></tr></table></td></tr>
<tr><td>Descrição</td><td>Envia uma solicitação para ingressar em uma partida</td></tr>
<tr><td>Retorno</td><td>{
matchId: o Id da partida,
name: o nome da partida,
qtdPlayers: a quantidade máxima de jogadores na partida,
status: o estado da partida podendo ser “aguardando” ou “jogando”,
players: lista de jogadores na partida
}</td></tr>
</table>

REALIZAR O LOGIN

<table>
<tr><td>Conteúdo da Mensagem</td><td>
<table>
<tr><th>Campo</th><th>Conteúdo</th></tr>
<tr><td>operação</td><td>“login”</td></tr>
<tr><td>username</td><td>O id do usuário</td></tr>
<tr><td>password</td><td>A senha do usuário</td></tr>
</table>
</td></tr>
<tr><td>Descrição</td><td>Envia uma solicitação para realizar o login</td></tr>
<tr><td>Retorno</td><td>{
id: {O id do usuário “criado”}
name: {o nome do jogador},
password: {A senha do usuário},
avatar: {json do avatar}
}</td></tr>
</table>

INFORMAÇÕES DO PERFIL

<table>
<tr><td>Conteúdo da Mensagem</td><td>
<table>
<tr><th>Campo</th><th>Conteúdo</th></tr>
<tr><td>operação</td><td>“my-profile”</td></tr>
<tr><td>userId</td><td>{O nome do usuário}</td></tr>
</table>
</td></tr>
<tr><td>Descrição</td><td>Retorna as informações do perfil do usuário</td></tr>
<tr><td>Retorno</td><td>{
id: {O id do usuário “criado”}
name: {o nome do jogador},
password: {A senha do usuário},
avatar: {json do avatar}
}</td></tr>
</table>

SAIR DA PARTIDA

<table>
<tr><td>Conteúdo da Mensagem</td><td>
<table>
<tr><th>Campo</th><th>Conteúdo</th></tr>
<tr><td>operação</td><td>“quit-match”</td></tr>
<tr><td>matchId</td><td>{Id da partida para sair}</td></tr>
<tr><td>userId</td><td>{Id do usuário que quer sair}</td></tr>
</table>
</td></tr>
<tr><td>Descrição</td><td>Envia uma solicitação para sair da partida</td></tr>
<tr><td>Retorno</td><td>{
matchId: {o Id da partida},
name: {o nome da partida},
qtdPlayers: {a quantidade máxima de jogadores na partida},
status: {o estado da partida podendo ser “aguardando” ou “jogando”},
players: {lista de jogadores na partida}
}</td></tr>
</table>

PRONTO PARA JOGAR

<table>
<tr><td>Conteúdo da Mensagem</td><td>
<table>
<tr><th>Campo</th><th>Conteúdo</th></tr>
<tr><td>operação</td><td>“ready-to-play”</td></tr>
<tr><td>userId</td><td>{Id do usuário que está pronto}</td></tr>
<tr><td>matchId</td><td>{Id da partida que está pronto}</td></tr>
</table>
</td></tr>
<tr><td>Descrição</td><td>Envia uma solicitação informando estar pronto para jogar, isso então inicia uma conexão contínua com o servidor, que utiliza múltiplos threads para gerenciar múltiplos clientes conectados simultaneamente.</td></tr>
<tr><td>Retorno</td><td>{
id: {O id do usuário “criado”}
name: {o nome do jogador},
password: {A senha do usuário},
avatar: {json do avatar},
status: {o estado da partida “aguardando” ou “jogando”}
}</td></tr>
</table>

REGISTRAR NO SERVIDOR

<table>
<tr><td>Conteúdo da Mensagem</td><td>
<table>
<tr><th>Campo</th><th>Conteúdo</th></tr>
<tr><td>operação</td><td>“sign-up”</td></tr>
<tr><td>username</td><td>{O nome do usuário}</td></tr>
<tr><td>password</td><td>{Senha do usuário}</td></tr>
<tr><td>avatarId</td><td>{Avatar escolhido}</td></tr>
</table>
</td></tr>
<tr><td>Descrição</td><td>Solicita registro no servidor</td></tr>
<tr><td>Retorno</td><td>{
id: {O id do usuário “criado”}
name: {o nome do jogador},
password: {A senha do usuário},
avatar: {json do avatar},
status: {o estado da partida “aguardando” ou “jogando”}
}</td></tr>
</table>

4. DIAGRAMA DE CLASSES

Nesta seção são apresentados os diagramas a respeito do projeto desenvolvido, uma divisão dos mesmos foi feita para facilitar a compreensão do leitor. Há três tipos de diagramas, **Diagrama de Modelo**, **Diagrama de Controle** e **Diagrama Android**.

É importante destacar que esses objetos podem sofrer pequenas alterações até o fim do desenvolvimento, levando assim os mesmos a serem atualizados neste documento.

Diagrama de Classes – Modelo

As classes referentes ao modelo da aplicação são compartilhadas entre o ***backend*** e o ***frontend***, isso facilita o tratamento dos objetos que "navegam" entre os dois campos.

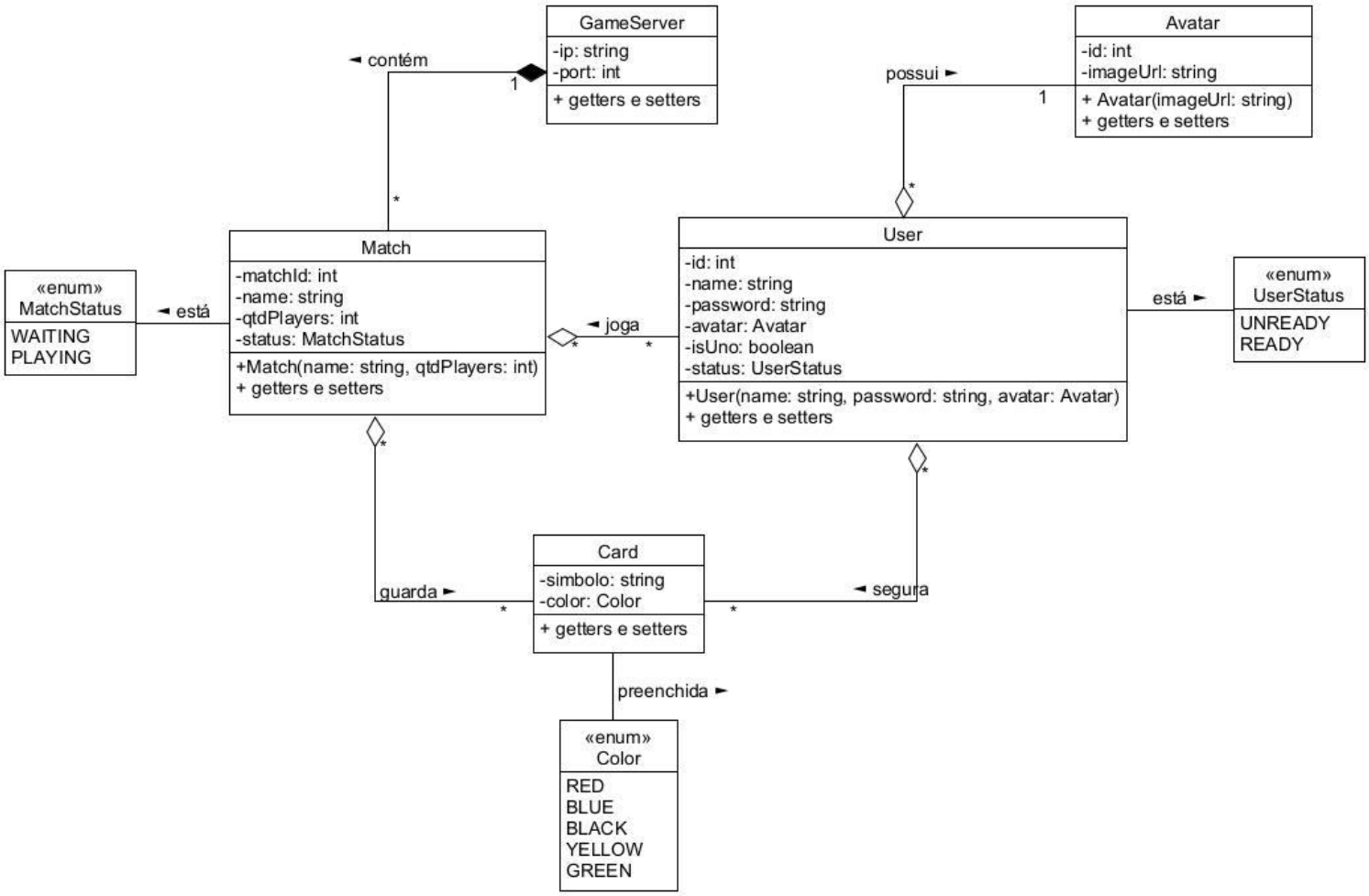

Diagrama de Classes – Controle

As classes referentes ao controle da aplicação, classes que usam o modelo e processam as mensagens recebidas, passam pela regra de negócio e devolvem a resposta quando necessário.

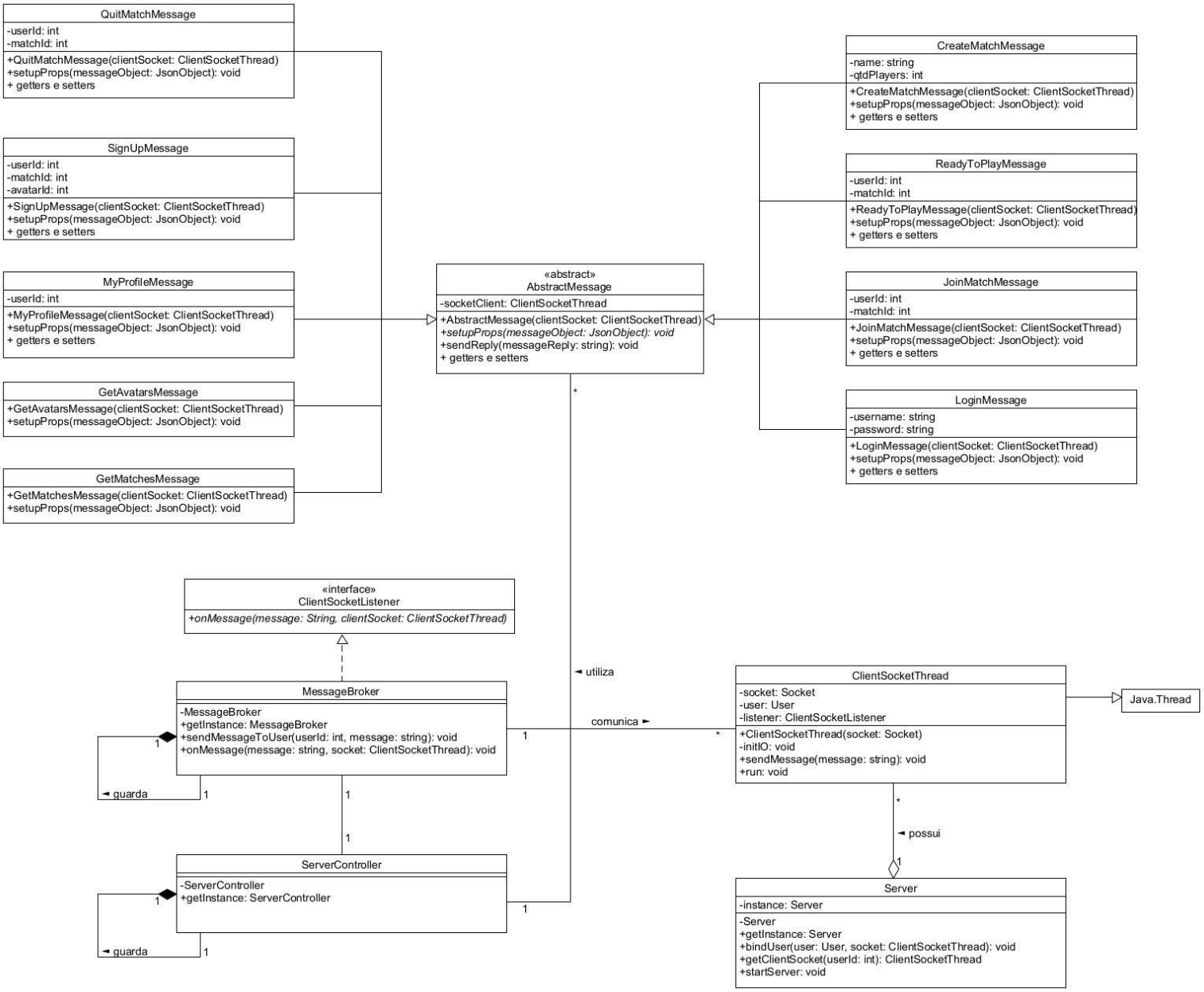

Diagrama de Classes – Android

As classes referentes a aplicação Android, na linha de frente da nossa aplicação. Mostram-se aqui somente as classes que tratam do envio e recebimento das mensagens, há ainda outros objetos referentes a componentes de tela, personalização de layouts, dentre outros.