

Uno

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

SUMÁRIO

| | |
|--------------------------|---|
| 1. INTRODUÇÃO | 3 |
| 2. REQUISITOS FUNCIONAIS | 4 |
| 3. MENSAGENS DO SISTEMA | 5 |

1. INTRODUÇÃO

O sistema a ser desenvolvido neste trabalho será baseado no conhecido jogo “Uno”. O jogo será desenvolvido inicialmente para desktop, e possivelmente irá receber uma versão mobile, exclusivamente para Android. Para o sistema, será desenvolvido o recurso de cadastro dos usuários, permitindo manter dados de perfil e histórico dos jogos. Após o login, o sistema irá permitir que o usuário visualize as salas disponíveis e permita que ele selecione a sala em que deseja se conectar. Dentro da sala, irá listar quais jogadores estão esperando para iniciar a partida, permitindo que o usuário defina se está pronto para começar. Dentro do jogo, as regras se baseiam conforme o “Uno”.

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 (caso haja). Ressaltamos que as mesmas podem ser alteradas ou incrementadas durante o decorrer do projeto.

BUSCA DE AVATARES

| Conteúdo da Mensagem | Campo | | Conteúdo |
|----------------------|-----------------------------------------------------------------------------------------------|--|---------------|
| | operação | | "get_avatars" |
| Descrição | Busca no servidor uma lista de avatares | | |
| Retorno | <pre>[{ id: {id do avatar}, image: {base64 da imagem do avatar} }, ...]</pre> | | |

REGISTRAR NO SERVIDOR

| Conteúdo da Mensagem | Campo | | Conteúdo |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|----------------------------|
| | operação | | "logon" |
| | nome | | {O nome do usuário} |
| | avatar | | {O id do avatar escolhido} |
| Descrição | Solicita registro no servidor | | |
| Retorno | <pre>{ id_usuario: {O id do usuário "criado"} matches: [{ id: {o Id da partida} jogadores: {a quantidade de jogadores na partida}. status: {o estado da partida "aguardando" ou "jogando"} }, ...] }</pre> | | |

LISTAGEM DE PARTIDAS

| | | |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Conteúdo da Mensagem | Campo | Conteúdo |
| | operação | “partidas” |
| Descrição | Busca no servidor a lista de partidas em execução, esta busca é feita de tempos em tempos enquanto o usuário não está definitivamente jogando. | |
| Retorno | Uma String no formato JSON: <pre>{ id: {o Id da partida} jogadores: {a quantidade de jogadores na partida}. status: {o estado da partida podendo ser “aguardando” ou “jogando”} }</pre> | |

INGRESSAR EM UMA PARTIDA

| | | |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|
| Conteúdo da Mensagem | Campo | Conteúdo |
| | operação | “join” |
| | user | O id do próprio usuário |
| | id | O id da partida que se quer ingressar |
| Descrição | Envia uma solicitação para ingressar em uma partida | |
| Retorno | <p>Caso a partida esteja disponível, retorna para o usuário as informações da mesma, para que seja então renderizado em tela o deck de cartas e as informações dos demais jogadores. O usuário só poderá ingressar em uma partida que está “Aguardando”, o retorno é “false” caso não possa ingressar, ou uma String no formato JSON contendo as informações dos demais usuários na partida:</p> <p>usuários: um array de nomes de usuário e ids.</p> <p>token: Uma chave aleatória gerada pelo server para “autenticar” o usuário nas próximas requisições</p> | |

SAIR DA PARTIDA

| Conteúdo da Mensagem | Campo | | Conteúdo |
|----------------------|--------------------------------------------------|--|-----------------------------------------------|
| | operação | | "quit" |
| | token | | O token do próprio usuário previamente gerado |
| Descrição | Envia uma solicitação para sair da partida | | |
| Retorno | Retorna "true" caso saia da partida com sucesso. | | |

PRONTO PARA JOGAR

| Conteúdo da Mensagem | Campo | | Conteúdo |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|-----------------------------------------------|
| | operação | | "ready" |
| | token | | O token do próprio usuário previamente gerado |
| Descrição | 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. | | |
| Retorno | <p>Conforme a partida for acontecendo, o servidor poderá então encaminhar dados ao cliente sobre o status da partida, para que este então possa atualizar a interface para o usuário. os dados possíveis são todos do tipo JSON, como a seguir:</p> <p>Informação que alguém entrou na partida</p> <pre>{ tipo: 'info', value: 'somebody_enters', params: { user_id: {id do usuário que entrou}, user_avatar: {avatar do usuário que entrou} } }</pre> | | |

| | |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>Informação que a partida se iniciou.</p> <pre>{ tipo: 'info', value: 'match_started', params: { cards: [{o array das cartas sorteadas para o usuário}] } }</pre> <p>Informação que é a sua vez de jogar</p> <pre>{ tipo: 'info', value: 'your_turn', params: { last card: {id da carta jogada}, bought cards: [{cartas compradas (caso alguém por exemplo jogou um +2, ou alguém jogou uma cor ou número que o jogador atual não possui, sendo assim já virão todas estas cartas, e o usuário terá a “ilusão” de estar comprando do baralho)}] } }</pre> <p>Informação que é a vez de alguém jogar</p> <pre>{ tipo: 'info', value: 'other_turn', params: [user_id: {id_do_usuario_que_jogou}] }</pre> <p>Informação que alguém jogou</p> <pre>{ tipo: 'info', value: 'somebody_played', params: { card_played: {id_da_carta jogada} } }</pre> <p>Informação que alguém pediu Uno!</p> <pre>{ tipo: 'info', value: 'uno', params: { user_id: {id do usuário que pediu uno} } }</pre> |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| | |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>Informação que alguém ganhou</p> <pre>{ tipo: 'end', value: 'win', params: { user_id: {id do usuário que ganhou} } }</pre> |
| Mensagens Adicionais | <p>Durante a conexão aberta, também há a necessidade de mandar algumas informações para o servidor, que são encaminhadas pela mesma conexão, sendo elas:</p> <p>Jogar uma carta:</p> <pre>{ tipo: 'info', id: {id da carta jogada}, atributos: { color: {Cor da carta jogada quando utilizado +4 ou "Muda de cor"} } }</pre> <p>Pedir Uno!</p> <pre>{ tipo: 'info', value: 'say_uno' }</pre> |