

Equipe: André Cristen e Lucas Levi Gonçalves

Linha de pesquisa: Desenvolvimento de jogos de tabuleiro digitais educacionais

Título do projeto: Desenvolvimento de um Jogo de tabuleiro

1. Detalhamento do projeto e técnicas	2
2. Requisitos Funcionais	3
3. Requisitos Não Funcionais	3
4. Regras de Negócio	3
5. Mensagens	4
6. Detalhamento dos Casos de Uso	6
6.1. UC01 - Criar conta	6
6.2. UC02 - Realizar login	7
6.3. UC03 - Criar Sala	7
6.4. UC04 - Entrar em uma sala	8
6.5. UC05 - Sair da sala	8
6.6. UC06 - Iniciar uma partida	9
6.7. UC07 - Sair da partida	9
6.8. UC08 - Comprar cartas	10
6.9. UC09 - Realizar Jogada	11
6.10. UC10 - Gerenciar usuários	11
6.11. UC11 - Gerenciar tabuleiros/mapas	12
7. Matriz de rastreabilidade	12
7.1. Requisitos x Casos de uso	12
7.2. Regras de Negócio x Casos de uso	13
8. Protótipos Não-funcionais	13
8.1. Login	13
8.2. Salas	15
8.3. Criar nova sala	15
8.4. Aguardando em uma sala	15
8.5. Na Partida	16
8.6. Minha Conta	16
9. Diagrama de caso de uso	17
10. Diagrama de sequência	18

10.1. Login	19
10.1.1. Fluxo principal	19
10.1.2. Fluxo de exceção	19
11. Diagrama de classes	20
12. Diagrama de Atividades	21
12.1. Login	21
13. Dimensionamento do software	23
13.1 UAW – Unadjusted ActorWeight.	23
13.2. UUCW – UnadjustedUse CaseWeight.	23
13.3. TCF - Technical Complexity Factor	23
13.3. UUCP - Unadjusted Use Case Points	25

1. Detalhamento do projeto e técnicas

O projeto consiste em um jogo de tabuleiro para ensino do conceito de algoritmos na educação básica, a ideia é totalmente baseada em um artigo publicado na Revista Brasileira de Informática na Educação, chamado de “Desenvolvimento e Avaliação de um Jogo de Tabuleiro para Ensinar o Conceito de Algoritmos na Educação Básica” onde o trabalho realmente consistia em desenvolver um jogo de tabuleiro físico, onde tendo essa ideia como base iremos desenvolver uma versão online multijogador com as alterações e personalizações que forem necessárias para o cumprir as exigências das disciplinas de Projeto Integrador III e Desenvolvimento de Sistemas Paralelos e Distribuídos.



Figura 1 - Splash Code

Vídeo de demonstração de como jogar a versão original a qual iremos nos basear <https://www.youtube.com/watch?v=LVKKEDUIuNo>.

2. Requisitos Funcionais

RF01	O sistema deve manter jogadores
RF02	O sistema deve manter salas
RF03	O sistema deve manter partidas
RF04	O sistema deve manter cartas
RF05	O sistema deve manter jogadas/movimentos
RF06	O sistema deve manter tabuleiros/mapas
RF07	O sistema deve manter entidades do tabuleiro/mapa

3. Requisitos Não Funcionais

RNF01	O sistema deve ter o backend e frontend em projetos separados
RNF02	O backend deve ser desenvolvido em Python
RNF03	O frontend deve ser desenvolvido em JavaScript
RNF04	O sistema deve comportar no mínimo 3 jogadores por sala
RNF05	O sistema deve comportar no máximo 5 jogadores por sala
RNF06	O sistema deve apresentar o status dos usuários na sala(online/offline)

4. Regras de Negócio

RN01	O usuário deve realizar o login antes de entrar em uma sala
RN02	Uma sala deve estar vinculada a uma ou mais partidas

RN03	Para iniciar uma partida é necessário que ao menos 3 jogadores aceitem o início da mesma
RN04	Caso durante a partida ocorram de jogadores serem desconectados o sistema deve aguardar 30 segundos antes de remover o jogador da sala automaticamente
RN05	A partida deve encerrar automaticamente caso restem apenas 2 jogadores online
RN06	Ao final da partida os jogadores devem ser redirecionados para a mesma sala que iniciou a partida

5. Mensagens

Nova conexão na sala

Conteúdo	{ room: ID da sala user: ID do Usuário }
Descrição	Quando um usuário entrar na sala
Retorno	{ event: "new_player", user: ID do Usuário }

Desconexão na sala

Conteúdo	{ room: ID da sala user: ID do Usuário }
Descrição	Quando um usuário for desconectado de uma sala
Retorno	{ event: "disconnected_player", message: string, user: ID do Usuário }

Remoção de usuário

Conteúdo	{ event: "remove_player", room: ID da sala, user: ID do Usuário }
-----------------	---

Descrição	Ação executada pelo dono da sala para remover um usuário da sala ou pelo próprio usuário ao sair da sala.
Retorno	{ event: string, message: string, user: ID do Usuário }

Início da partida

Conteúdo	{ event: "start_match", room: ID da Sala }
Descrição	Ação executada pelo dono da sala quando tiverem usuários suficientes para jogar
Retorno	{ event: string, message: string }

Ação de movimento/jogada

Conteúdo	{ event: "execute_throw", room: ID da Sala, user: ID do Usuário, cards: [Array de ID's das cartas que foram escolhidas] }
Descrição	Ação de jogada de movimento do personagem
Retorno	{ event: string, message: string }

Cancelar partida

Conteúdo	{ event: "stop_match", room: ID da Sala }
Descrição	Ação executada pelo dono da sala caso ele decida parar a partida
Retorno	{ event: string, message: string }

Finalização da partida

Conteúdo	{ event: "end_match", room: ID da Sala }
Descrição	Operação executada quando uma partida chega ao final
Retorno	{ event: string, message: string }

6. Detalhamento dos Casos de Uso

6.1. UC01 - Criar conta

Objetivo	Permitir que um usuário que ainda não possua uma conta possa se registrar
Requisitos	RF01
Atores	Jogador
Prioridade	Alta
Pré-condições	O usuário não poderá estar logado em uma conta existente
Pós-condições	O usuário será redirecionado para a tela de login
Frequência de uso	Regular
Criticalidade	Média
Fluxo Principal	Usuário irá preencher as informações solicitadas em tela e clicar no botão "Registrar-se"
Fluxo Alternativo	
Fluxo de Exceção	<ul style="list-style-type: none">• Usuário já cadastrado: Caso alguma das informações únicas como email e apelido forem preenchidas, o cadastro não poderá ser concluído e a tela se manterá a mesma informando o erro ao usuário.• Senhas não coincidem: Caso a senha e confirmação de senha forem digitadas de forma diferentes, o cadastro não poderá ser

	concluído e a tela se manterá a mesma informando o erro ao usuário.
--	---

6.2. UC02 - Realizar login

Objetivo	Permitir que um usuário já cadastrado possa entrar na plataforma
Requisitos	RF01
Atores	Jogador
Prioridade	Alta
Pré-condições	O usuário não poderá estar logado em uma conta existente
Pós-condições	O usuário será redirecionada para a tela inicial
Frequência de uso	Recorrente
Criticalidade	Média
Fluxo Principal	Usuário preenche as informações solicitadas em tela e clicar no botão “Entrar”
Fluxo Alternativo	
Fluxo de Exceção	<ul style="list-style-type: none"> • Informações não coincidem: Caso as informações preenchidas pelo usuário não representam nenhum cadastro, o login não poderá ser efetuado e a tela se manterá a mesma informando o erro ao usuário.

6.3. UC03 - Criar Sala

Objetivo	Permitir um usuário criar uma nova sala para organização de uma partida
Requisitos	RF02
Atores	Jogador
Prioridade	Média
Pré-condições	O usuário precisa estar logado em uma conta existente
Pós-condições	O usuário será redirecionado para a tela que representa a nova sala criada

Frequência de uso	Regular
Criticalidade	Média
Fluxo Principal	Usuário preenche as informações solicitadas em tela e clicar no botão “Criar”
Fluxo Alternativo	
Fluxo de Exceção	

6.4. UC04 - Entrar em uma sala

Objetivo	Permitir que o usuário entre em uma sala já existente
Requisitos	RF02
Atores	Jogador
Prioridade	Média
Pré-condições	<ul style="list-style-type: none"> • O usuário precisa estar logado em uma conta existente • A sala desejada não pode estar cheia • A sala desejada não pode ter sido iniciada sua partida
Pós-condições	O usuário será redirecionado para a tela que representa a sala selecionada
Frequência de uso	Constante
Criticalidade	Baixa
Fluxo Principal	Usuário seleciona a sala deseja e deve clicar no botão “Entrar”
Fluxo Alternativo	
Fluxo de Exceção	<ul style="list-style-type: none"> • Sala cheia: Caso durante o período de solicitação do usuário entrar na sala outro usuário entrar antes, deverá ser exibida a mensagem de que a sala está cheia; • Partida iniciada: Caso durante o período de solicitação do usuário entrar na sala a partida tenha sido iniciada, deverá ser exibida a mensagem de que a partida já foi iniciada.

6.5. UC05 - Sair da sala

Objetivo	Permitir que o usuário saia da sala
-----------------	-------------------------------------

Requisitos	RF02
Atores	Jogador
Prioridade	Média
Pré-condições	Estar em uma sala
Pós-condições	O usuário será redirecionado a tela inicial
Frequência de uso	Constante
Criticalidade	Baixa
Fluxo Principal	Clicar no botão “Sair”
Fluxo Alternativo	Caso o usuário que sair da sala seja o dono da mesma, essa função deve ser repassada para outro usuário que se mantiver na sala
Fluxo de Exceção	

6.6. UC06 - Iniciar uma partida

Objetivo	Permitir que os usuários iniciarem a partida
Requisitos	RF03
Atores	Jogador
Prioridade	Média
Pré-condições	<ul style="list-style-type: none"> • A sala possui no mínimo 3 jogadores • O usuário ser o criador da sala
Pós-condições	Os usuários presentes na sala serão redirecionados para a tela da partida
Frequência de uso	Constante
Criticalidade	Média
Fluxo Principal	O dono da sala deverá clicar no botão “Iniciar Partida”
Fluxo Alternativo	
Fluxo de Exceção	<ul style="list-style-type: none"> • Caso algum usuário tenha saído no exato momento da iniciação da partida e a sala assim continuar com menos que três jogadores, deverá apresentar uma mensagem dizendo que não é possível iniciar uma partida com menos de três jogadores

6.7.UC07 - Sair da partida

Objetivo	Permitir que o usuário saia de uma partida
Requisitos	RF03
Atores	Jogador
Prioridade	Baixa
Pré-condições	O usuário precisa estar em uma partida em andamento
Pós-condições	O usuário será redirecionado para a tela inicial
Frequência de uso	Constante
Criticalidade	Baixa
Fluxo Principal	Clicar no botão de opções da partida e selecionar a opção “Sair”
Fluxo Alternativo	Caso ao sair restem menos que três jogadores na partida a mesma deve ser encerrada e todos os usuários restantes redirecionados para a tela da sala
Fluxo de Exceção	

6.8. UC08 - Comprar cartas

Objetivo	Permitir que o usuário realize a compra de cartas para montagem de jogadas
Requisitos	RF03, RF04
Atores	Jogador
Prioridade	Média
Pré-condições	O usuário precisa estar em uma partida em andamento e precisa ser o jogador da vez na rodada
Pós-condições	Usuário receberá cartas aleatórias
Frequência de uso	Constante
Criticalidade	Média
Fluxo Principal	O usuário deverá clicar em “Comprar Cartas” e selecionar a quantidade de 1 a 3.
Fluxo Alternativo	

Fluxo de Exceção	
-------------------------	--

6.9. UC09 - Realizar Jogada

Objetivo	Permitir que o usuário realize uma jogada
Requisitos	RF03, RF04, RF05, RF06, RF07
Atores	Jogador
Prioridade	Média
Pré-condições	O usuário precisa estar em uma partida em andamento e precisa ser o jogador da vez na rodada
Pós-condições	O usuário terá realizado uma jogada
Frequência de uso	Constante
Criticalidade	Média
Fluxo Principal	O usuário deverá clicar em “Montar Jogada” e selecionar as cartas que ele tem a disposição na ordem que deverão ser jogadas
Fluxo Alternativo	
Fluxo de Exceção	<ul style="list-style-type: none"> • Movimento inválido: Caso a jogada do usuário seja inválida por motivos como, se chocar com: outro jogador, fim do mapa, entidade/obstáculo, o sistema deverá apresentar uma mensagem dizendo que o movimento é inválido e dar a opção de montar uma nova jogada ou pular a vez

6.10. UC10 - Gerenciar usuários

Objetivo	Permitir que sejam gerenciados os cadastros de usuários
Requisitos	RF01
Atores	Administrador
Prioridade	Alta
Pré-condições	Administrador deve estar logado no servidor
Pós-condições	
Frequência de uso	Regular

Criticalidade	Alta
Fluxo Principal	Selecionar o usuário e a ação desejada
Fluxo Alternativo	
Fluxo de Exceção	

6.11. UC11 - Gerenciar tabuleiros/mapas

Objetivo	Permitir que sejam gerenciados os cadastros de tabuleiros/mapas
Requisitos	RF06
Atores	Administrador
Prioridade	Alta
Pré-condições	Administrador deve estar logado no servidor
Pós-condições	
Frequência de uso	Baixa
Criticalidade	Alta
Condição de Entrada	
Fluxo Principal	Selecionar o tabuleiro/mapa e a ação desejada
Fluxo Alternativo	<ul style="list-style-type: none"> • Criar um novo tabuleiro/mapa: preencher as informações solicitadas em tela e clicar no botão “Confirmar”
Fluxo de Exceção	

7. Matriz de rastreabilidade

7.1. Requisitos x Casos de uso

Requisitos x Casos de uso	RF01	RF02	RF03	RF04	RF05	RF06	RF07
UC01	X						

UC02	X						
UC03		X					
UC04		X					
UC05		X					
UC06			X				
UC07			X				
UC08				X			
UC09					X		
UC10	X						
UC11						X	X

7.2. Regras de Negócio x Casos de uso

Regras de Negócio x Casos de uso	RN01	RN02	RN03	RN04	RN05	RN06
UC01						
UC02	X					
UC03						
UC04						X
UC05						
UC06		X	X			
UC07				X	X	
UC08						
UC09						
UC10						
UC11						

8. Protótipos Não-funcionais

8.1. Login

A wireframe of a login page displayed within a browser window. The browser's address bar shows "Home". The page content is enclosed in a large rounded rectangle. On the left side of this rectangle, the text "Bem-vindo" is followed by "Nome Do Jogo" in a larger font. Below this, a smaller line of text reads: "O que acha de aprender se divertindo? Entre e divirta-se com seus amigos agora mesmo sem pagar nada!". On the right side, the heading "Login" is positioned above two input fields labeled "Email" and "Senha". Below these fields, a link "Crie uma conta!" is shown, followed by a blue button labeled "Entrar".

Home

Bem-vindo
Nome Do Jogo

O que acha de aprender se divertindo? Entre e divirta-se com seus amigos agora mesmo sem pagar nada!

Login

Email

Senha

Ainda não está registrado?
[Crie uma conta!](#)

Entrar

8.3. Criar nova sala

Criar Nova Sala

Criar Nova Sala

Nome da sala

Senha da sala

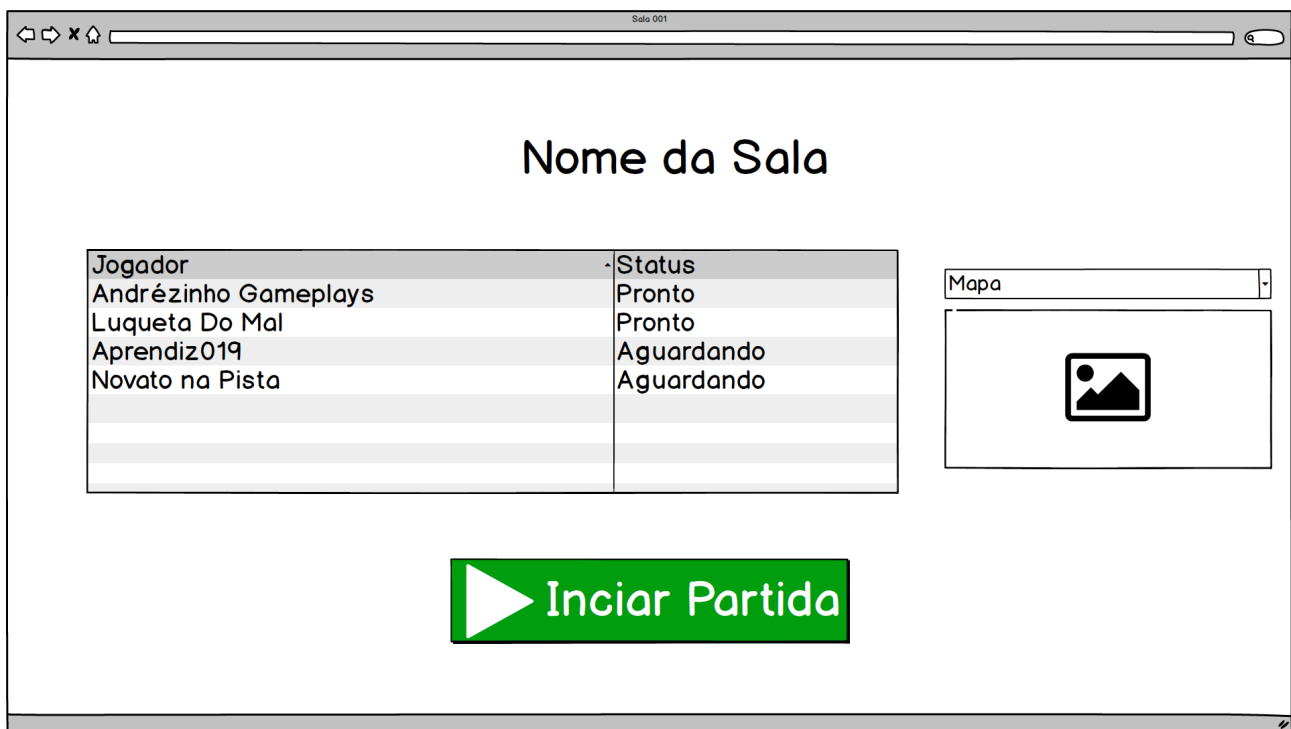
Confirmar Senha da sala

Quantidade Máxima de Jogadores: 3

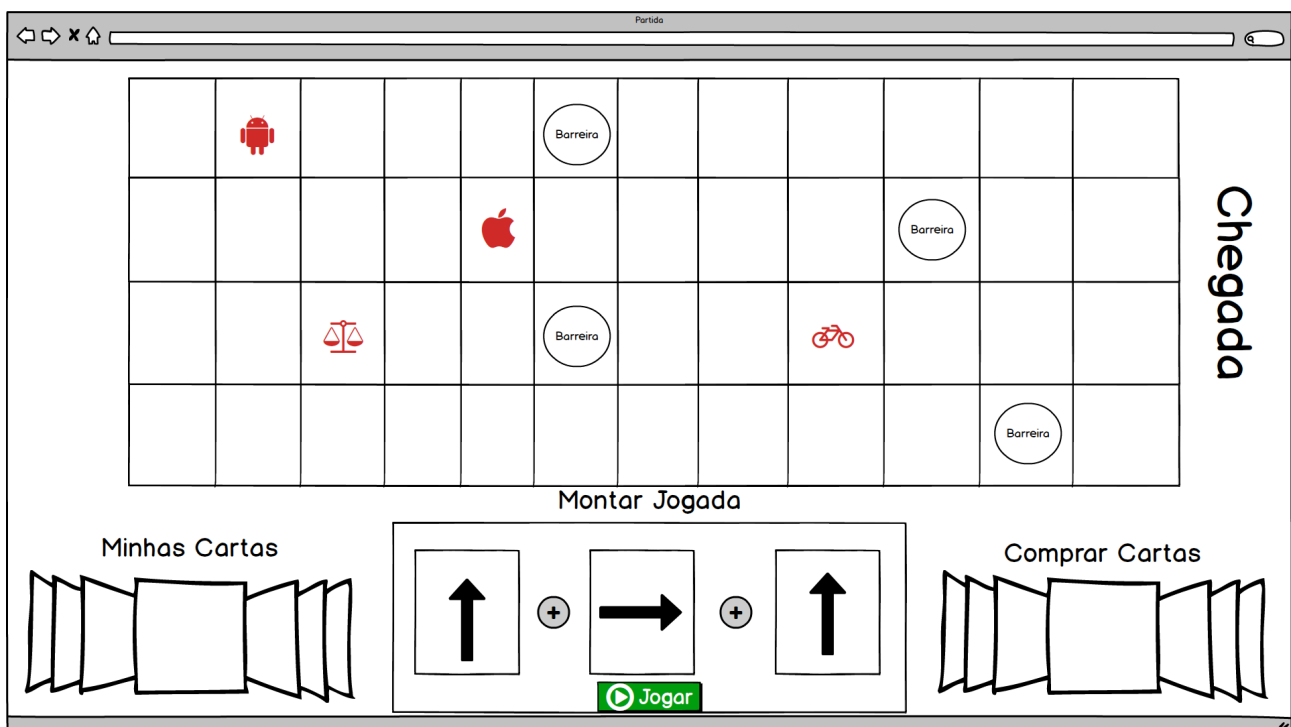
Mapas

Criar

8.4. Aguardando em uma sala



8.5. Na Partida



8.6. Minha Conta

Minha Conta

SalasMinha ContaSair



Email


Nome Completo

Apelido do Jodador

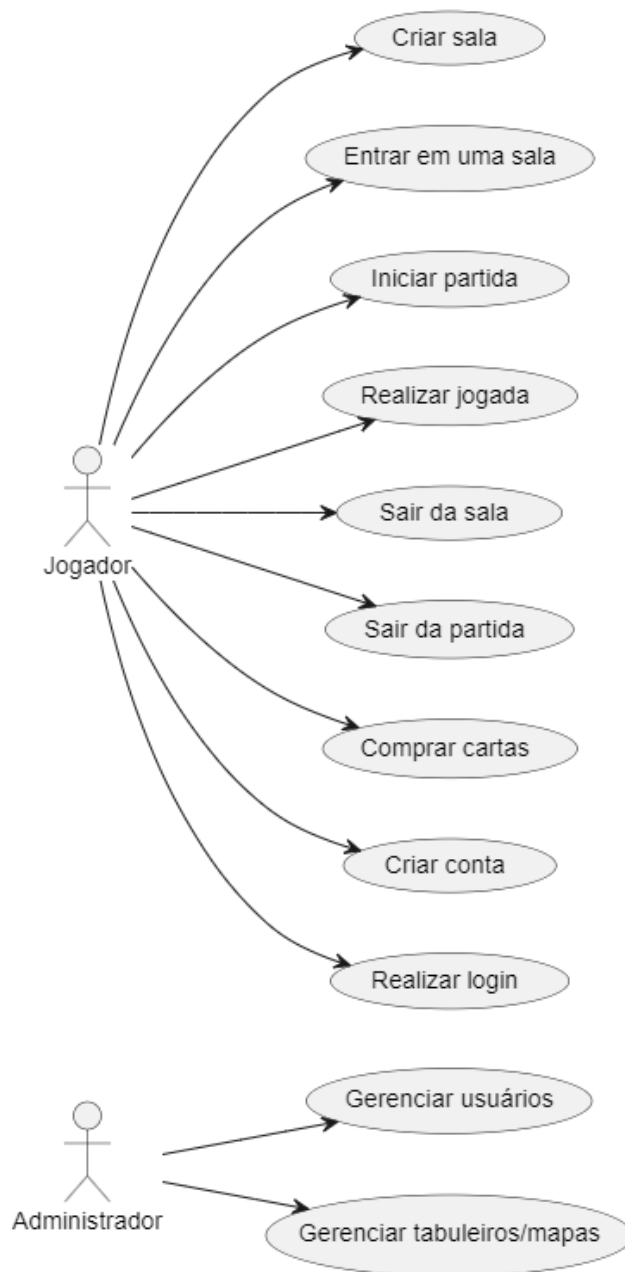
Data de Nascimento

Senha

Confirmar Senha

 **Salvar Dados**

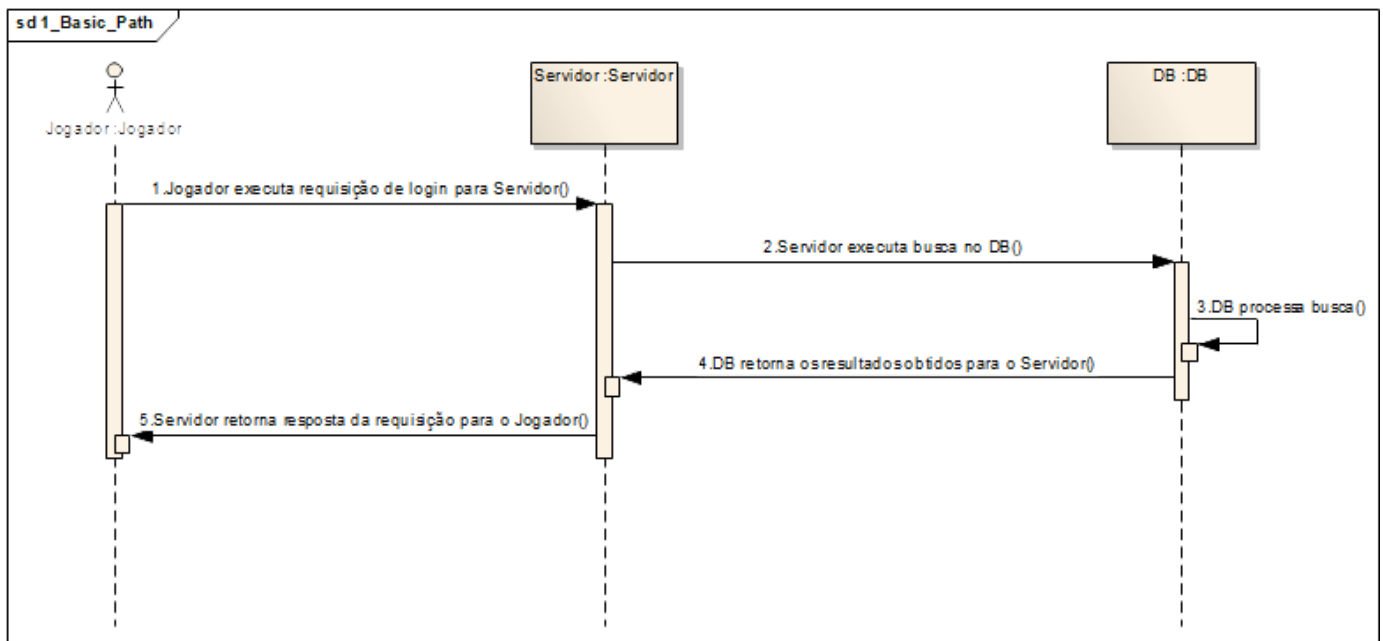
9. Diagrama de caso de uso



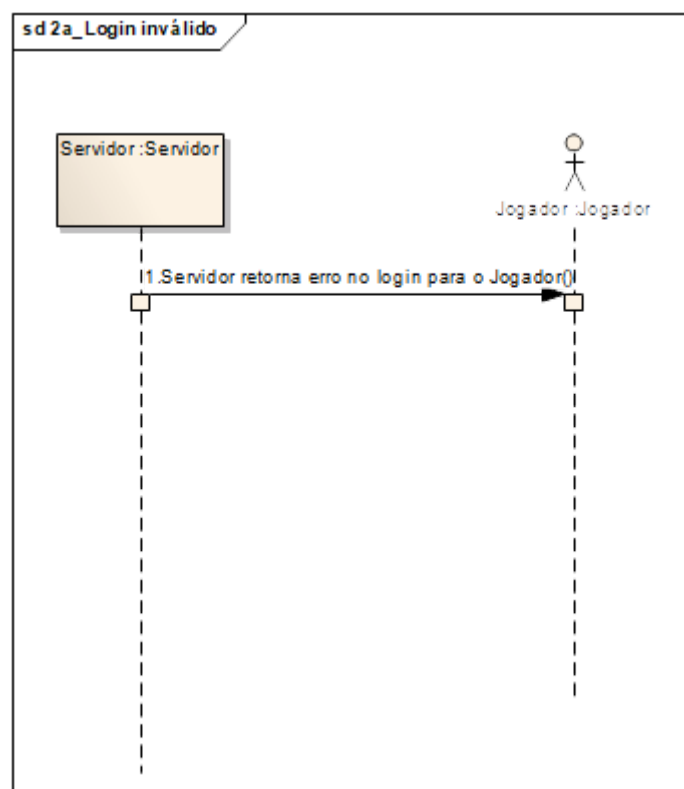
10. Diagrama de sequência

10.1. Login

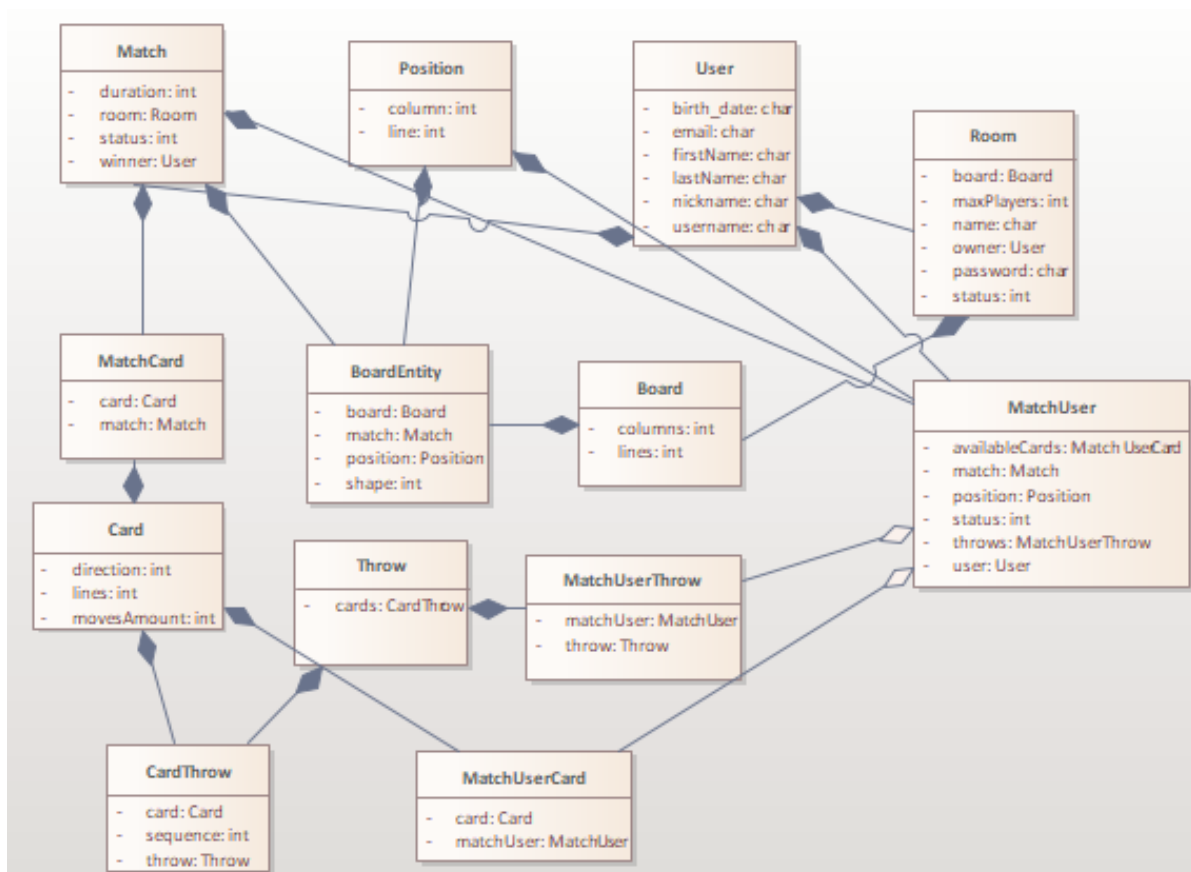
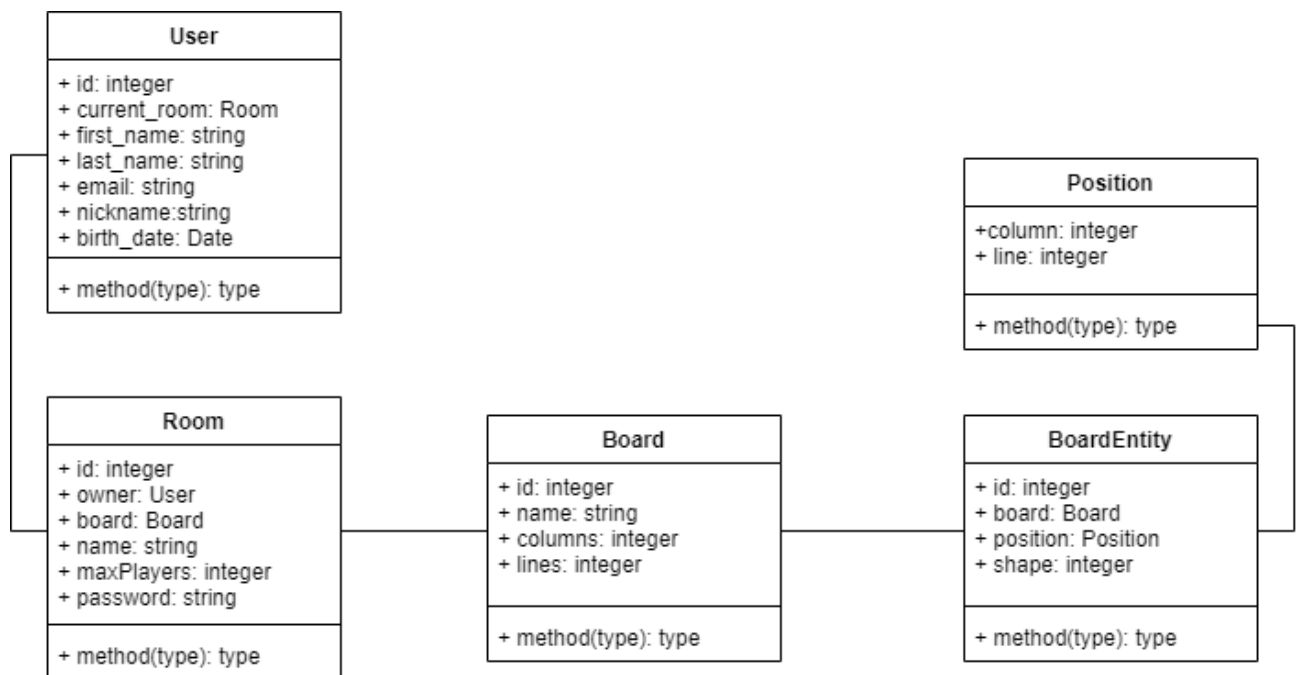
10.1.1. Fluxo principal



10.1.2. Fluxo de exceção



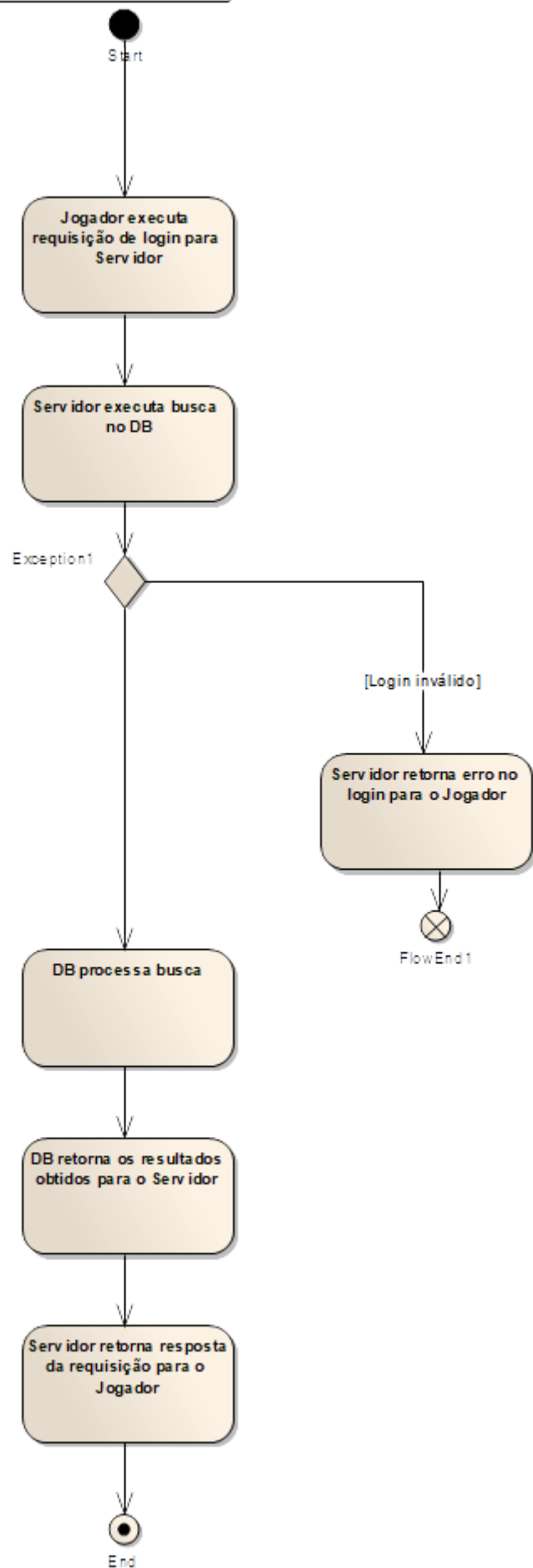
11. Diagrama de classes parcial



12. Diagrama de Atividades

12.1. Login

act Realizar login_ActivityGraph



13. Dimensionamento do software

13.1 UAW – Unadjusted ActorWeight.

- Jogador: 3
- Administrador: 3

Total: 6

13.2. UUCW – UnadjustedUse CaseWeight.

- Jogador
 - Criar Sala: 5
 - Entrar em uma sala: 5
 - Iniciar Partida: 5
 - Realizar Jogada: 5
 - Sair da sala: 5
 - Sair da partida: 5
 - Comprar cartas: 5
 - Criar conta: 5
 - Realizar Login:5
- Administrador:
 - Gerenciar usuários: 5
 - Gerenciar tabuleiros/mapas: 5

Total: $11 * 5 = 55$

13.3. TCF - Technical Complexity Factor

Fator	Requisito	Peso	Influência	Total
T1	Sistema distribuído	2	3	6
T2	Tempo de resposta	2	4	8
T3	Eficiência	1	2	2

T4	Processamento complexo	1	1	1
T5	Código reusável	1	4	4
T6	Facilidade de instalação	0.5	0	0
T7	Facilidade de uso	0.5	5	2.5
T8	Portabilidade	2	1	2
T9	Facilidade de mudança	1	2	2
T10	Concorrência	1	1	1
T11	Recursos de segurança	1	2	2
T12	Acessível por terceiros	1	3	3
T13	Requer treinamento especial	1	0	0
				33,5

$$TCF = 0.6 + (0.01 * 33,5) = 0,935$$

13.2. EF: Environmental Factor

E1	Facilidade com RUP ou outro processo formal 1	1.5	3
E2	Experiência com a aplicação em desenvolvimento	0.5	4
E3	Experiência em Orientação a Objetos 1	1	5
E4	Presença de analista experiente	0.5	5
E5	Motivação	1	3
E6	Requisitos estáveis	2	2
E7	Desenvolvedores em meio-expediente	-1	2
E8	Linguagem de programação difícil	-1	1

$$EF: 1.4 + (-0.03 * 18)$$

$$EF: 0.86$$

13.3. UUCP - Unadjusted Use Case Points

- Total: $6 + 55 = 61$

13.3. UCP

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF}$$

$$\text{UCP} = 61 * 0.935 * 0.86$$

$$\text{UCP} = 49,05$$

Horas estimadas: 981 h