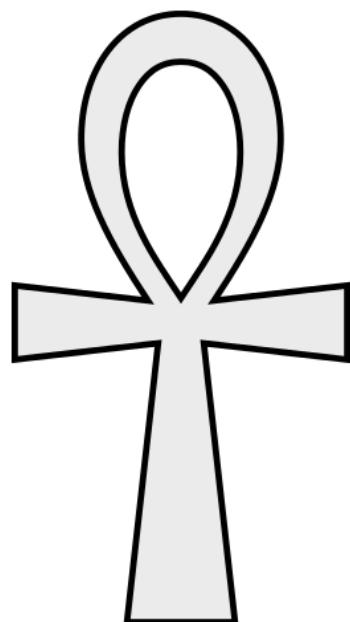


Ankh



**Projeto: Ankh
Versão: 2.0**

**Curitiba
2017**

Ankh

**Responsáveis pelo projeto e
desenvolvimento:**

**Nicolas Abril, Lucca Rawlyk Holosbach,
Ricardo Luis Souza Ardisson, Caique
Destro**

**Curitiba
2017**

Histórico de Modificações

Data	Versão	Descrição	Autores
12/2016	1.0	Versão inicial do jogo	Guilherme Triches, Matheus Giovanni Dias, Natan Almeida Junges, Ricardo Ardisson
09/2017	2.0	Save/Load, níveis e progressão de jogo	Caíque Destro, Nicolas Abril, Ricardo Ardisson, Lucca Rawlyk

SUMÁRIO

1 INTRODUÇÃO	13
1.1 OBJETIVO GERAL	13
1.2 OBJETIVOS ESPECÍFICOS	13
1.3 CONTEÚDO DO PLANO DO PROJETO	13
2 LEVANTAMENTO DE REQUISITOS	14
2.1 QUESTÕES ORGANIZACIONAIS	14
2.2 QUESTÕES ECONÔMICAS	15
2.3 QUESTÕES TÉCNICAS	16
2.4 QUESTÕES OPERACIONAIS	17
2.5 REQUISITOS FUNCIONAIS, NÃO FUNCIONAIS E RESTRIÇÕES DE PROJETO	18
3 ESTUDO DE VIABILIDADE	19
3.1 VIABILIDADE ORGANIZACIONAL	19
3.2 VIABILIDADE ECONÔMICA	19
3.3 VIABILIDADE TÉCNICA	20
3.4 VIABILIDADE OPERACIONAL	21
3.5 RECURSOS A SEREM UTILIZADOS	21
4 RESULTADOS	23
4.1 CASOS DE USO	23
4.1.1 RELAÇÃO ENTRE OS REQUISITOS FUNCIONAIS E OS CASOS DE USO	23
4.1.2 DIAGRAMA DE CASOS DE USO	24
4.1.3 CENÁRIO DOS CASOS DE USO	25
4.2 DIAGRAMA DE CLASSES	41
4.2.1 DIAGRAMA DE INSTÂNCIAS	49
4.2.1.1 DIAGRAMA WORLD	50
4.2.1.2 DIAGRAMA GUI/LIB	50
4.3 DICIONÁRIO DE INFORMAÇÕES	51
4.4 DIAGRAMA DE SEQUÊNCIA	63
4.5 DIAGRAMA DE COMUNICAÇÃO	68
4.6 DIAGRAMA ENTIDADE RELACIONAMENTO	75
4.7 DIAGRAMA ESTADOS	76
5 CONCLUSÕES	99
6 REFERÊNCIAS BIBLIOGRÁFICAS	100

LISTA DE FIGURAS

Figura 1. Diagrama de Casos de Usos	21
Figura 2. Diagrama de Classes 1	38
Figura 3. Diagrama de Classes 2	38
Figura 4. Diagrama de Classes 3	39
Figura 5. Diagrama de Classes 4	39
Figura 6. Diagrama de Classes 5	40
Figura 7. Diagrama de Classes 6	41
Figura 8. Diagrama de Classes 7	42
Figura 9. Diagrama de Classes 8	42
Figura 10. Diagrama de Classes 9	43
Figura 11. Diagrama de Classes 10	44
Figura 12. Diagrama de Classes 11	45
Figura 13. Diagrama de Classes 12	45
Figura 14. Diagrama de Classes 13	46
Figura 16. Diagrama de Instância World	47
Figura 17. Diagrama de Instância GUI/LIB	47
Figura 18. Diagrama de Sequência Fechar Programa	
Figura 19. Diagrama de Sequência Fim de Jogo	
Figura 20. Diagrama de Sequência Gravar Highscores	
Figura 21. Diagrama de Sequência Inicialização	
Figura 22. Diagrama de Sequência Menu de Jogo	
Figura 23. Diagrama de Sequência Menu Principal	
Figura 24. Diagrama de Sequência Mostrar Créditos	
Figura 25. Diagrama de Sequência Mostrar Highscores	
Figura 26. Diagrama de Sequência Mudar Configurações	
Figura 27. Diagrama de Sequência Novo Jogo	
Figura 28. Diagrama de Sequência Passar de Nível	
Figura 29. Diagrama de Sequência Salvar Jogo	
Figura 30. Diagrama de Sequência Carregar Jogo	
Figura 31. Diagrama de Sequência Executar Movimento	
Figura 32. Diagrama de Comunicação Carregar jogo	
Figura 33. Diagrama de Comunicação Executar movimento	
Figura 34. Diagrama de Comunicação Fechar programa	
Figura 35. Diagrama de Comunicação Fim de jogo	
Figura 36. Diagrama de Comunicação Gravar highscores	
Figura 37. Diagrama de Comunicação Menu de jogo	
Figura 38. Diagrama de Comunicação Menu principal	
Figura 39. Diagrama de Comunicação Mostrar créditos	
Figura 40. Diagrama de Comunicação Mostrar highscores	
Figura 41. Diagrama de Comunicação Mudar configurações	
Figura 42. Diagrama de Comunicação Novo jogo	
Figura 43. Diagrama de Comunicação Passar de nível	
Figura 44. Diagrama de Comunicação Salvar jogo	

- Figura 45.** Diagrama de Entidade-relacionamento
- Figura 46.** Diagrama Estados - Carregar jogo - game
- Figura 47.** Diagrama Estados - Carregar jogo - load files table
- Figura 48.** Diagrama Estados - Carregar jogo - load return button
- Figura 49.** Diagrama Estados - Carregar jogo - load save button
- Figura 50.** Diagrama Estados - Carregar jogo - mm load game button
- Figura 51.** Diagrama Estados - Carregar jogo - Tudo
- Figura 52.** Diagrama Estados - Executar movimento - Game
- Figura 53.** Diagrama Estados - Executar movimento - per
- Figura 54.** Diagrama Estados - Executar movimento - selected_ability
- Figura 55.** Diagrama Estados - Executar movimento - source_effects
- Figura 56.** Diagrama Estados - Fechar programa - target_effects
- Figura 57.** Diagrama Estados - Fechar programa - game
- Figura 58.** Diagrama Estados - Fechar programa - mm exit button
- Figura 59.** Diagrama Estados - Fechar programa - Tudo
- Figura 60.** Diagrama Estados - Fim de jogo - eg return button 0
- Figura 61.** Diagrama Estados - Fim de jogo - eg savescore button
- Figura 62.** Diagrama Estados - Fim de jogo - eg textbox
- Figura 63.** Diagrama Estados - Fim de jogo - endgame GUI
- Figura 64.** Diagrama Estados - Fim de jogo - Game
- Figura 65.** Diagrama Estados - Gravar highscores - endgame GUI
- Figura 66.** Diagrama Estados - Gravar highscores - game
- Figura 67.** Diagrama Estados - Gravar highscores - savescores button
- Figura 68.** Diagrama Estados - Gravar highscores - highscores textbox
- Figura 69.** Diagrama Estados - Gravar highscores - Tudo
- Figura 70.** Diagrama Estados - Inicialização - Tudo
- Figura 71.** Diagrama Estados - Menu de jogo - game
- Figura 72.** Diagrama Estados - Menu de jogo - pause GUI
- Figura 73.** Diagrama Estados - Menu principal - game
- Figura 74.** Diagrama Estados - Menu principal - Main menu GUI
- Figura 75.** Diagrama Estados - Mostrar creditos - credits button
- Figura 76.** Diagrama Estados - Mostrar creditos - credits GUI
- Figura 77.** Diagrama Estados - Mostrar creditos - Tudo
- Figura 78.** Diagrama Estados - Mostrar highscores -
guiLibrary[MAIN_MENU][HIGH_SCORES]
- Figura 79.** Diagrama Estados - Mostrar highscores - openGUI
- Figura 80.** Diagrama Estados - Mostrar highscores - Tudo
- Figura 81.** Diagrama Estados - Mudar configurações - game
- Figura 82.** Diagrama Estados - Mudar configurações - ge
- Figura 83.** Diagrama Estados - Mudar configurações - openGUI
- Figura 84.** Diagrama Estados - Novo jogo - game
- Figura 85.** Diagrama Estados - Novo jogo -
guiLibrary[MAIN_MENU][NEW_GAME_BUTTON]
- Figura 86.** Diagrama Estados - Passar de nível - game
- Figura 87.** Diagrama Estados - Salvar jogo - Game

Figura 88. Diagrama Estados - Salvar jogo -

guiLibrary[SAVELOAD][SAVE_BUTTON]

Figura 89. Diagrama Estados - Salvar jogo - Tudo

LISTA DE TABELAS E QUADROS

Tabela 1. Descrição dos campos dos cenários de casos de uso	22
Tabela 2. Caso de Uso 1 - Menu principal	23
Tabela 3. Caso de Uso 2 - Fechar programa	24
Tabela 4. Caso de Uso 3 - Mostrar highscores	25
Tabela 5. Caso de Uso 4 - Mudar configurações	25
Tabela 6. Caso de Uso 5 - Novo jogo	26
Tabela 7. Caso de Uso 6 - Salvar jogo	27
Tabela 8. Caso de Uso 7 - Carregar jogo	28
Tabela 9. Caso de Uso 8 - Executar movimento	29
Tabela 10. Caso de Uso 9 - Fim de jogo	32
Tabela 11. Caso de Uso 10 - Menu de jogo	33
Tabela 12. Caso de Uso 11 - Gravar highscore	34
Tabela 13. Caso de Uso 12 - Passar e nível	35
Tabela 14. Caso de Uso 13 - Inicialização	36
Tabela 15. Caso de Uso 14 - Mostrar créditos	37
Tabela 16. Dicionário da Classe PlayerCharacter	48
Tabela 17. Dicionário da Classe AICharacter	49
Tabela 18. Dicionário da Classe Ability	50
Tabela 19. Dicionário da Classe Buff	50
Tabela 20. Dicionário da Classe TemporaryBuff	51
Tabela 21. Dicionário da Classe RandomAI	51
Tabela 22. Dicionário da Classe EADisplay	52
Tabela 23. Dicionário da Classe RotateToTarget	52
Tabela 24. Dicionário da Classe EndTurn	52
Tabela 25. Dicionário da Classe FriendlyStatsEffect	52
Tabela 26. Dicionário da Classe HostileStatsEffect	53
Tabela 27. Dicionário da Classe RandomTeleport	53
Tabela 28. Dicionário da Classe Animate	53
Tabela 29. Dicionário da Classe AnimatelfDead	53
Tabela 30. Dicionário da Classe Move	53
Tabela 31. Dicionário da Classe BuffEffect	54

Tabela 32. Dicionário da Classe SourceEffectGroup	54
Tabela 33. Dicionário da Classe TargetEffectGroup	55
Tabela 34. Dicionário da Classe BUTTON	55
Tabela 35. Dicionário da Classe SLidebar	55
Tabela 36. Dicionário da Classe GUI	56
Tabela 37. Dicionário da Classe Logger	57
Tabela 38. Dicionário da Classe Screen	57
Tabela 39. Dicionário da Classe LIB_BITMAP	58
Tabela 40. Dicionário da Classe LIB_GUI	58
Tabela 41. Dicionário da Classe LIB_SFX	58
Tabela 42. Dicionário da Classe SFX_PLAYER	58
Tabela 43. Dicionário da Classe World	59

LISTA DE ABREVIATURAS E SIGLAS

AP: Action Points (pontos de ação). Representa quantos movimentos o jogador ainda pode fazer em seu turno. Cada movimento gasta uma quantidade diferente e pontos de ação.

HP: Health Points (pontos de vida). Representa quanto dano o jogador ainda pode receber até morrer. Alguns movimentos aumentam ou diminuem o HP.

MP: Magic Points (pontos de magia). Representa quantos ataques mágicos o jogador ainda pode realizar naquele turno. Cada movimento custa uma quantidade diferente de MP.

IDE: Integrated Development Environment (ambiente de desenvolvimento integrado). Programa que reúne diversas ferramentas de desenvolvimento de software, como editor, depurador e compilador.

1 INTRODUÇÃO

O projeto consiste num jogo de batalha em formato de turnos². O jogador tem armas variadas e diferentes formas de lidar com os inimigos. A temática egípcia foi escolhida por sua mitologia interessante, com criaturas e seres místicos. Os clientes do sistema são os próprios integrantes do grupo.

1.1 Objetivo Geral

Desenvolver um jogo de luta em turnos, com temática egípcia.

1.2 Objetivos Específicos

Entreter o usuário e fornecer experiência e conhecimento para a equipe no desenvolvimento do projeto.

1.3 Conteúdo do Plano do Projeto

Capítulo 2: Levantamento de Requisitos.

Capítulo 3: Estudo de Viabilidade.

Capítulo 4: Resultados.

Capítulo 5: Conclusões.

Capítulo 6: Referências Bibliográficas.

2 LEVANTAMENTO DE REQUISITOS

Esta seção contém o levantamento de requisitos do projeto e análise dos mesmos, separado em seções separadas por tipo. O levantamento foi feito utilizando entrevistas e reuniões.

2.1 Questões Organizacionais

- O jogo é um projeto desejado pela equipe como projeto para a disciplina?
 - Sim, ele é de interesse da equipe e também é complexo o suficiente para ser possível aplicar os conhecimentos da disciplina.
- Realizar este projeto garantirá aprovação dos integrantes na disciplina?
 - Provavelmente, se todos os alunos se dedicarem ao projeto, espera-se a aprovação do grupo.
- A realização deste projeto enriquecerá o conhecimento da equipe sobre a matéria?
 - Sim, pois criaremos o projeto do sistema com base nos assuntos estudados na matéria e os exemplos apresentados pelo professor.
- Qual é o objetivo a ser atingido pelo projeto?
 - A finalidade do jogo é entreter o usuário com um modo simples de interação.
- Qual é o benefício que o projeto dá ao usuário?
 - O usuário fará uso do nosso sistema apenas com finalidade de entretenimento, neste caso, o benefício é a diversão.
- O projeto está dentro das capacidades de realização do grupo?

- Sim, o projeto em questão tem como base um jogo preliminar realizado em outra matéria, sendo assim, qualquer novo desafio que surgir ainda está dentro das nossas capacidades.
- O grupo de desenvolvimento do projeto está satisfeito com o formato de trabalho proposto?
 - Sim, definimos tarefas para cada integrante e estamos mantendo uma uniformidade no desenvolvimento do projeto.

2.2 Questões Econômicas

- O projeto trará alguma renda?
 - Dentro dos objetivos atuais não.
- O projeto poderá trazer alguma renda?
 - Dificilmente, para este projeto ser comercial seria necessário um planejamento mais profundo e muito mais tempo de trabalho para ficar apresentável.
- É preciso comprar algum software para o desenvolvimento do projeto?
 - Não. Usaremos as bibliotecas allegro e sqlite, gratuitas, e o IDE Code::Blocks, também gratuito.
- A organização já tem o hardware necessário para fazer o projeto?
 - Sim, cada integrante possui seu computador pessoal em casa e a equipe também possui acesso a computadores na universidade.
- Os recursos necessários, dinheiro e tempo, são compatíveis com a capacidade econômica da equipe?
 - Sim. Não há salários a serem pagos e o tempo para a entrega é longo o suficiente.

- O potencial usuário precisará despender de recursos financeiros extras?
 - Não, o jogo não consome muitos recursos e só tem dependências gratuitas.
- Para avançar o projeto para um nível onde seria possível comercializado que recursos seriam necessários?
 - Continuando no caminho que foi usado para fazer o jogo atual não seria necessário adquirir nenhum software pago ou hardware a mais para o desenvolvimento de um jogo comercializável, porém teriam que ser investido muitas horas de trabalho a mais.

2.3 Questões Técnicas

- A equipe tem conhecimento necessário para o desenvolvimento de um jogo?
 - Sim, todas as ferramentas utilizadas são de domínio da equipe.
- A organização precisa manter algum sistema ativo, como um servidor, para que os clientes possam utilizar o jogo?
 - A princípio, não. O jogo é totalmente offline e depende apenas da máquina local.
- A equipe precisa fazer algum tipo de instalação manual para os usuários utilizarem o jogo?
 - Não, o jogo será feito para compilar diretamente no hardware dos usuários.
- Será preciso providenciar algum tipo de manutenção para o jogo?
 - Sim, é possível corrigir bugs no código, além de ser preciso atentar-se com o hardware dos usuários.
- O projeto exigirá do usuário uma máquina muito poderosa?
 - Não, o jogo consome poucos recursos e deve executar normalmente em qualquer computador doméstico.

- O usuário precisa de algum acessório extra para interagir com o jogo?
 - Não, o mouse e teclado que estão presentes em qualquer computador doméstico são o suficiente para a interação com o jogo.
- O jogo é compatível com quais sistemas operacionais?
 - É possível executar o jogo em máquinas com Windows ou Linux.

2.4 Questões Operacionais

- Qual é o público alvo do jogo?
 - Jogadores casuais e usuários comuns de computadores.
- O jogo será disponibilizado em alguma plataforma de distribuição?
 - A princípio, não há planos de disponibilizar o jogo em meios comuns de distribuição de jogos.
- Caso alguém tenha interesse no jogo, como essa pessoa poderia testá-lo?
 - A princípio, somente os integrantes da equipe têm acesso ao código, o usuário teria que entrar em contato com o grupo para poder jogar.
- Qual é o nível de complexidade do jogo?
 - As mecânicas básicas do jogo são simples, mas a dificuldade aumenta juntamente com a progressão de fases.
- Qual é a dificuldade que o usuário terá para testar o jogo?
 - Se o usuário tiver as dependências instaladas, será apenas preciso executar o programa compilado para o seu sistema.
- Qual é a expectativa de engajamento dos usuários?
 - Esperamos que usuários consumam o jogo de forma esporádica e descompromissada, apenas se entretenham com o jogo de forma casual.

- Existe algum requisito referente ao governo?
 - Não, o jogo tem escopo muito pequeno e não há interesses governamentais envolvidos.

2.5 Requisitos Funcionais, Não Funcionais e Restrições de Projeto

Requisitos Funcionais:

- FR01: O jogador pode abrir o menu principal.
- FR02: O jogador pode fechar o programa.
- FR03: O jogador pode ver os highscores.
- FR04: O jogador pode mudar as configurações do jogo.
- FR05: O jogador pode criar um novo jogo.
- FR06: O jogador pode salvar uma partida.
- FR07: O jogador pode carregar uma partida.
- FR08: O jogador pode realizar um movimento em seu turno.
- FR09: Quando o personagem morrer, o jogo acaba.
- FR10: O jogador pode abrir um menu de jogo.
- FR11: O jogador pode gravar seu score.
- FR12: Avança a fase após derrotar todos os inimigos
- FR13: Quando o sistema é aberto, inicializa e vai para o menu principal.
- FR14: O jogador pode ver os créditos.

Requisitos Não Funcionais:

- NFR01: Os menus são navegáveis por mouse.
- NFR02: O botão de fechar o programa é acessível em qualquer momento.
- NFR03: O menu de configurações é acessível a qualquer momento.
- NFR04: Cada partida é gerada aleatoriamente.

- NFR05: O score só é gravado se for um dos melhores.
- NFR06: Só é possível salvar se houver uma partida ativa.
- NFR07: Só passa de nível quando todos os inimigos forem mortos.

Restrições de Projeto:

- PR01: A linguagem de programação a ser utilizada é C++.
- PR02: O acesso a bancos de dados é feito usando a biblioteca SQLite.
- PR03: Os diagramas são feitos com Astah.
- PR04: Será utilizada a biblioteca gráfica Allegro.

3 ESTUDO DE VIABILIDADE

3.1 Viabilidade Organizacional

O objetivo da organização não envolve questões monetárias ou estratégicas, mas sim a absorção de conhecimento e a aprovação na disciplina, combinado com a realização de um projeto que leve ao entretenimento dos integrantes da equipe e possivelmente de outros usuários do sistema.

Portanto o jogo pareceu ser uma solução adequada para o projeto, sendo algo dentro do espectro do exequível e apresentando o potencial de utilizar os conhecimentos da disciplina, além de ser de interesse dos integrantes da equipe.

3.2 Viabilidade Econômica

Não há planos para o projeto trazer nenhum lucro para a organização, seria possível disponibilizar o jogo em alguma plataforma, porém não está nos objetivos do projeto.

O custo do projeto não é algo tão relevante, já que não foi necessário adquirir nenhum produto para sua realização, permitindo a equipe a não gastar muito mais dinheiro que o normal para desenvolver o jogo. Somente o tempo é algo considerável, porém esse já é algo esperado para um estudante, que pode ser considerado um investimento pessoal.

Mas podemos estimar os gastos que haveriam em um projeto profissional equivalente. Estimando-se 50 horas para cada um dos 4 integrantes, e comparando com um estágio de 20 horas semanais com salário de R\$ 1000,00, por cerca de R\$ 12,50 a hora, temos um custo de pessoal de R\$ 2500. Há um único software utilizado que não é livre e gratuito, o Astah community. Seria necessário comprar uma licença de uso profissional do Astah por US\$ 1190/ano. O custo de desenvolvimento substituindo os bibliotecas e softwares de desenvolvimento utilizados presentemente por soluções proprietárias pagas pode ser difícil de estimar, pois dependeria de negociação com empresas de desenvolvimento, e seriam possivelmente percentuais do faturamento. Mas por exemplo, a popular engine Unity cobra US\$ 35/mês para projetos de até US\$ 200.000 de faturamento por ano. Outro custo seria uma sala comercial pequena em Curitiba, custando cerca de R\$ 400,00 por mês. E finalmente, um PC de desenvolvimento simples pode ser obtido por cerca de R\$ 1500 reais cada, totalizando R\$ 6000,00 para os quatro integrantes da equipe. O custo total do projeto, se fosse necessário começar sem nenhum equipamento ou recurso, seria então em torno de R\$ 13000,00.

Mesmo assim, este projeto poderia ser muito bem aproveitado para fins financeiros, já que o mercado de jogos eletrônicos no Brasil é muito grande e rentável⁵.

3.3 Viabilidade Técnica

O jogo foi projetado para ser algo que não necessita de muita especificação de software ou hardware. Só é necessário um sistema operacional Linux ou Windows e um computador com uma capacidade

básica de processamento, já que o jogo é bem leve. Por enquanto não há dependência da rede para o jogo, já que não há multiplayer em rede.

Para desenvolver a codificação do projeto foi necessário conhecimento de C++ e da biblioteca gráfica Allegro, que a equipe já possuía de disciplinas passadas. O único requisito do projeto que não estava no repertório da equipe foi o de banco de dados, decidindo usar o SQLite, porém não foi algo que restringiu muito o seu avanço.

3.4 Viabilidade Operacional

Como o projeto é um jogo, a própria equipe é o modelo para os usuários. E, na nossa opinião, mesmo que não seja possível desenvolver todos as ideias que a equipe considerou como interessantes, ainda sim o jogo atende os requisitos básicos de entretenimento. Presume-se dos usuários que possuam conhecimentos básicos de informática, além de alguma experiência com jogos eletrônicos. Isso é necessário devido a ausência de um tutorial ou manual.

3.5 Recursos a serem utilizados

Como citados anteriormente os custos do projeto são basicamente os custos de manutenção dos equipamentos utilizados, e de deslocamento e tempo dos integrantes da equipe.

Estimando valores, o projeto tomaria cerca de 50 horas de cada integrante, totalizando cerca de 200 horas de trabalho. E, usando como o custo de trabalho da equipe como 0,30 reais por hora, proveniente apenas do custo da eletricidade, desgaste dos computadores e outros pequenos gastos similares, já que nenhum dos integrantes recebe dinheiro pela realização do projeto, o custo total serial em torno de 60 reais no total.

Material	Custo
Code::Blocks e compilador(gcc)	Software livre (GPL)

Astah community	Software proprietário de uso gratuito para fins não comerciais
Allegro	Software livre (zlib)
SQLite	Software livre (domínio público)
PC simples	Cerca de R\$1500,00 cada

4 RESULTADOS

4.1 Casos de Uso

Nesta seção é feita a análise de casos de uso do projeto, relacionados com os requisitos funcionais de projeto, mostrando as funcionalidades do software. A Figura 1 contém o diagrama de caso de uso e as Tabelas 1-14 contém os cenários dos casos de uso.

4.1.1 Relação entre os requisitos funcionais e os casos de uso

- FR01: O jogador pode abrir o menu principal. -> UC01: Menu principal
- FR02: O jogador pode fechar o programa. -> UC02: Fechar programa
- FR03: O jogador pode ver os highscores. -> UC03: Mostrar highscores
- FR04: O jogador pode mudar as configurações do jogo. -> UC04: Mudar configurações
- FR05: O jogador pode criar um novo jogo. -> UC05: Novo jogo
- FR06: O jogador pode salvar uma partida. -> UC06: Salvar jogo
- FR07: O jogador pode carregar uma partida. -> UC07: Carregar jogo
- FR08: O jogador pode realizar um movimento em seu turno. -> UC08: Executar movimento
- FR09: Quando o personagem morrer, o jogo acaba. -> UC09: Fim de jogo
- FR10: O jogador pode abrir um menu de jogo. -> UC10: Menu de jogo
- FR11: O jogador pode gravar seu score. -> UC11: Gravar highscore
- FR12: Avança a fase após derrotar todos os inimigos -> UC12: Passar de nível
- FR13: Quando o sistema é aberto, inicializa e vai para o menu principal. -> UC13: Inicialização
- FR14: O jogador pode ver os créditos -> UC14: Mostrar créditos

4.1.2 Diagrama de casos de uso

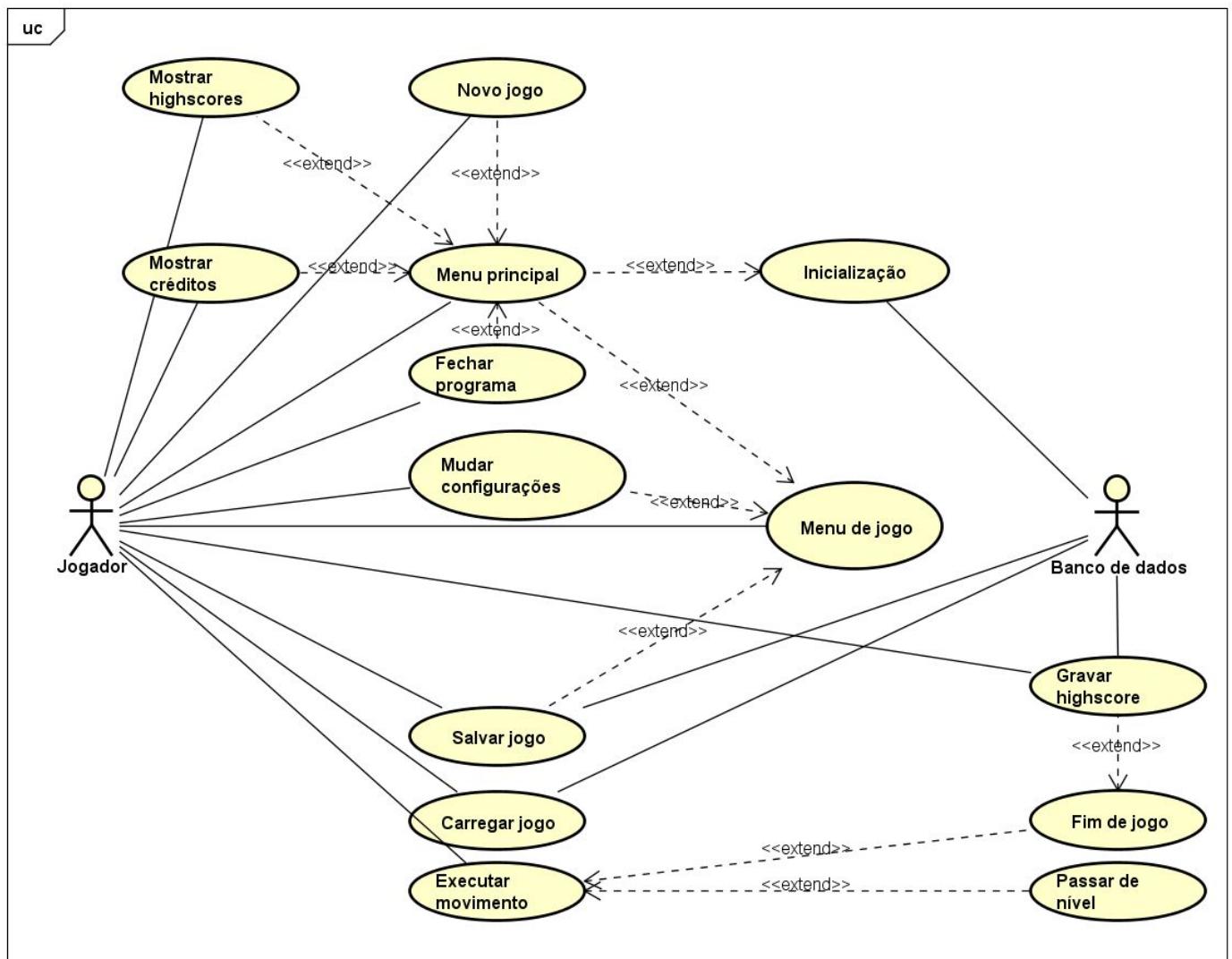


Figura 1. Diagrama de Casos de Uso

4.1.3 Cenário dos casos de uso

Os cenários são compostos por 9 atributos: Nome, Descrição, Atores, Pré-condições, Pós-condições, Fluxo Básico, Fluxo Alternativo, Fluxo de Exceção e Regras de Negócio. A Tabela 0 contém a descrição de cada um dos campos.

Nome	[Código do caso de uso] : Nome do caso de uso
Atores	[Tipo de ator]: Nome do ator
Descrição	Breve descrição do caso de uso
Pré-condições	Condições que precisam ser atendidas para o caso de uso ser executado
Pós-Condições	Condições que precisam ter sido atendidas ao término do caso de uso

Fluxo básico - Fluxo padrão de execução do caso de uso	
Ações dos atores	Ações do sistema
	1 - Ordem sequencial de execução do caso de uso. A direita, ações realizadas pelo sistema
2 - Ação realizada por um dos atores explicitados no campo "Atores"	
Regras de negócio	
Lista de todas as regras de negócio do caso de uso	
Fluxo alternativo - Caminho alternativo de execução do caso de uso, que pode ser executado ou não ao percorrer o fluxo básico	
1 - Ordem sequencial da execução do fluxo alternativo. A esquerda ações tomadas por atores	
	2 - A direita, ações tomadas pelo

	sistema
Fluxo de exceção - Fluxo de ações a serem tomadas caso algo errado ou inesperado aconteça	
[Número do passo do fluxo onde ocorreu a exceção][Letra identificadora da exceção] - Descrição da exceção e ações a serem tomadas quando ela ocorre	

Tabela 1 : Descrição dos campos dos cenários de casos de uso

Nome	UC01: Menu principal
Atores	Autor principal: Jogador
Descrição	Abre o menu principal, interrompendo a partida, se estiver ativa.
Pré-condições	O jogo acaba de ser inicializado ou o jogador apertou o botão de retornar ao menu principal
Pós-Condições	O menu principal foi aberto.

Fluxo básico	
Ações dos atores	Ações do sistema
	<p>1 - Exibe o menu principal, que contém o título do jogo e os botões:</p> <ul style="list-style-type: none"> • New Game • Highscores • Credits • Config • Load Game • Exit <p>Exibidos nesta ordem, separados em duas colunas</p>
2 - O jogador seleciona uma das opções	

	<p>3 - Se o jogador selecionou:</p> <ul style="list-style-type: none"> • New Game : Vai para o caso de uso “Novo jogo” • Highscores: Vai para o caso de uso “Mostrar highscores” • Config: Vai para o caso de uso “Mudar configurações” • Load Game: Vai para o caso de uso “Carregar jogo” • Credits: Vai para o caso de uso “Mostrar créditos” • Exit: Vai para o caso de uso “Fechar programa”
--	---

Tabela 2. Caso de Uso 1

Nome	UC02: Fechar programa
Atores	Autor principal: Jogador
Descrição	Encerra todos os processos ativos e fecha o programa
Pré-condições	O jogador pressionou um botão de fechar o programa
Pós-condições	O programa foi encerrado

Fluxo Básico	
Ações dos atores	Ações do sistema
	1 - Elimina todas as entidades
	2 - Faz log do encerramento
	3 - Salva as configurações
	4 - Limpa todos os objetos e encerra as bibliotecas
	5 - Fecha o programa

Tabela 3. Caso de Uso 2

Nome	UC03: Mostrar highscores
Atores	Autor principal: Jogador Aktor de suporte: Banco de dados
Descrição	Mostra uma tabela com os maiores scores gravados
Pré-condições	O jogador pressionou o botão de highscores ou morreu
Pós-condições	Os scores foram exibidos

Fluxo Básico	
Ações dos atores	Ações do sistema
	1 - Troca para o menu de highscores, contendo um botão para voltar ao menu principal dizendo “Return”
	2 - Pede acesso a tabela de scores ao banco de dados
3 - O banco de dados informa o conteúdo da tabela de scores	
	4 - A tabela é exibida
5 - O jogador pressiona o botão “Return”	

Tabela 4. Caso de Uso 3

Nome	UC04: Mudar configurações
Atores	Autor principal: Jogador
Descrição	Permite ao jogador modificar algumas configurações do jogo
Pré-condições	O jogador apertou um botão de configurações
Pós-condições	O menu de configurações foi exibido

Fluxo Básico	
Ações dos atores	Ações do sistema
	1 - Troca para o menu de configurações
	2 - Exibe as barras de ajuste de som, com os textos de descrição ao lado “SFX” e “Music”
3 - O jogador pressiona o botão de voltar	
	4 - Retorna ao menu principal
Fluxo Alternativo	
1 - O jogador modifica a posição de uma das barras de ajuste	
	2 - A configuração correspondente à barra é modificada

Tabela 5. Caso de Uso 4

Nome	UC05: Novo jogo
Atores	Autor principal: Jogador
Descrição	Começa uma partida nova
Pré-condições	O jogador apertou um botão de novo jogo
Pós-condições	Uma partida nova foi iniciada

Fluxo Básico	
Ações dos atores	Ações do sistema
	1 - Vai para a tela de jogo
	2 - Gera o personagem e inimigos
	3 - Exibe os movimentos possíveis ao jogador, com o nome do

	movimento, seu custo de AP e de MP e a tecla para executá-lo
	4 - Exibe o HP, MP e AP do jogador no canto da tela
	5 - Exibe os buffs equipamentos do jogador, com o nome e descrição de seus efeitos
6 - O jogador seleciona um dos movimentos possíveis	
	7 - Vai para o caso de uso “Executar movimento”

Tabela 6. Caso de Uso 5

Nome	UC06: Salvar jogo
Atores	Autor principal: Jogador Aktor de Bastidor: Banco de dados
Descrição	Salva a partida ativa ou carrega uma partida salva
Pré-condições	O jogador apertou o botão de salvar e está em uma partida ativa
Pós-condições	O jogo foi salvo ou uma partida salva foi carregada

Fluxo Básico	
Ações dos atores	Ações do sistema
	<p>1 - Troca para o menu de salvar o jogo contendo:</p> <ul style="list-style-type: none"> • Um botão de voltar “Return” • Um botão de salvar “Save score” • Uma caixa de inserção de texto para o nome do arquivo com a descrição “Save name” • Uma tabela com os saves

	existentes, cada um com um botão para carregá-lo escrito “Load”
2 - O jogador entra com o nome do save	
3 - O jogador pressiona o botão de salvar	
	4 - O jogo envia as informações da partida para o banco de dados
5 - O banco de dados guarda as informações em uma tabela com o nome inserido pelo jogador	
	6 - Retorna para o menu de jogo
Regras de negócio	
[RN01] O nome do arquivo é obrigatório	
[RN02] O nome do arquivo só pode conter letras, números e ‘_’.	
Fluxo Alternativo 1 - Return	
1 - O jogador pressiona o botão de voltar “Return”	
	2 - Volta para o menu de jogo
Fluxo Alternativo 2 - Load	
1 - O jogador pressiona um dos botões “Load”	
	2 - Carrega a partida correspondente à partida selecionada
Fluxo de exceção	
3a. O jogador tenta salvar sem inserir um nome para o arquivo. Nada ocorre	
3b - O jogador tenta salvar com um nome que contém um caractere inválido	

Tabela 7. Caso de Uso 6

Nome	UC07: Carregar jogo
Atores	Autor principal: Jogador Aktor de suporte: Banco de dados
Descrição	Carrega um jogo que havia sido salvo previamente
Pré-condições	O jogador selecionou o menu de carregar jogo
Pós-condições	A partida salva foi carregada e iniciada

Fluxo Básico	
Ações dos atores	Ações do sistema
	1 - Troca para a tela de carregamento de partidas salvas contendo um botão para voltar ao menu principal “Return”
2 - Pega a lista de saves com o banco de dados	
	3 - Exibe uma lista com todas as partidas salvas ao jogador
4 - O jogador seleciona um dos jogos para ser carregados e aperta o botão de carregar	
	5 - Pede ao banco de dados o conteúdo da save selecionada
6 - O banco de dados entrega o conteúdo das tabelas	
	7 - Troca para a tela de jogo
	8 - Carrega os inimigos e o jogador
	9 - Exibe os movimentos possíveis ao jogador, com o nome do

	movimento, seu custo de AP e de MP e a tecla para executá-lo
	10 - Exibe o HP, MP e AP do jogador no canto da tela
	11 - Exibe os buffs equipamentos do jogador, com o nome e descrição de seus efeitos
12 - O jogador seleciona um dos movimentos	
	13 - Vai para o caso de uso “Executar movimento”
Fluxo Alternativo	
1 - O jogador seleciona o botão de voltar no menu de carregamento	
	2 - Volta ao menu principal

Tabela 8. Caso de Uso 7

Nome	UC08: Executar movimento
Atores	Autor principal: Jogador
Descrição	Realiza uma das ações possíveis para o jogador
Pré-condições	O jogador selecionou um movimento para fazer em seu turno
Pós-condições	O movimento foi executado

Fluxo Básico	
Ações dos atores	Ações do sistema
	1 - Executa a ação correspondente à tecla selecionada
	2 - Subtrai o AP e o MP correspondentes ao movimento realizado

3 - O jogador seleciona um novo movimento	
	4 - Volta ao passo 1
Fluxo Alternativo 1 - Matou todos os inimigos	
	1 - Após executar a ação selecionada, todos os inimigos estavam mortos.
	2 - Vai para o caso de uso “Passar de nível”
Fluxo Alternativo 2 - Acabou pontos de ação	
	1 - Após reduzir os pontos de ação do jogador, ele está com 0 AP.
	2 - Passa para o turno do inimigo
	3 - Executa o movimento de um inimigo
	4 - Aplica o dano dos efeitos de status do inimigo
	5 - Se ainda tem inimigos para executar um movimento volta para o passo 3. Se não, recupera o AP do jogador e vai para o passo 5 do fluxo básico.
Fluxo Alternativo 3 - Jogador morreu	
	1 - Após um inimigo executar seu movimento, o jogador está sem HP.
	2 - Vai para o caso de uso “Fim de jogo”
Fluxo Alternativo 4 - Menu de jogo	
1 - O jogador pressiona ESC	
	2 - Vai para o caso de uso: “Abrir menu de jogo”
Fluxo Alternativo 5 - Movimento com Alvo	

1 - O jogador seleciona um movimento que requer um alvo	
	2 - Mostra os alvos possíveis para aquele movimento, mostrando a tecla que o seleciona
3 - O jogador seleciona um dos alvos possíveis	
	4 - Vai para o passo 1 do fluxo básico
Fluxo de exceção	
1a - O jogador selecionou um movimento que custa mais AP ou MP do que ele tem. Nada ocorre	

Tabela 9. Caso de Uso 8

Nome	UC09: Fim de jogo
Atores	Autor principal: Jogador
Descrição	Termina uma partida
Pré-condições	O jogador deve ter acabado de morrer em uma partida
Pós-condições	Retornou ao menu principal

Fluxo Básico	
Ações dos atores	Ações do sistema
	1 - Mostra a tela de fim de jogo para o jogador, com os botões “Return” e “Save score”.
	2 - Exibe na tela o turno em que o jogo acabou, o nível e quantos inimigos o jogador matou.
	3 - Exibe uma caixa de texto para inserir o nome do jogador, com a descrição “Insert name”

	4 - Vai para o caso de uso “Gravar highscore”
--	---

Tabela 10. Caso de Uso 9

Nome	UC10: Abrir menu de jogo
Atores	Autor principal: Jogador
Descrição	Abre o menu de jogo durante uma partida em andamento sem interrompê-la.
Pré-condições	O jogador apertou ESC durante uma partida em andamento
Pós-condições	O menu de jogo foi aberto

Fluxo Básico	
Ações dos atores	Ações do sistema
	1 - Pausa a partida em andamento
	2 - Exibe o menu de jogo que contém as seguintes opções: <ul style="list-style-type: none">● Resume● Config● Save/Load● Main Menu● Restart● Exit
3 - O jogador seleciona um dos botões	
	4 - Caso tenha selecionado: <ul style="list-style-type: none">● “Resume” ou tenha pressionado ESC esconde o menu de jogo e retoma a partida.● “Config” vai para o caso de uso “Mudar configurações”.● “Save/Load” vai para o caso de uso “Salvar jogo”.● “Main Menu” vai para o caso

	<ul style="list-style-type: none"> de uso “Abrir menu principal”. “Restart” vai para o caso de uso “Novo jogo” “Exit” vai para o caso de uso “Fechar programa”.
--	--

Tabela 11. Caso de Uso 10

Nome	UC11: Gravar Highscore
Atores	Autor principal: Jogador Aktor de suporte: Banco de dados
Descrição	Grava o score da partida finalizada no banco de dados
Pré-condições	Uma partida acabou
Pós-condições	O Highscore foi salvo, retorna ao menu principal

Fluxo Básico	
Ações dos atores	Ações do sistema
1 - Banco de dados fornece os highscores anteriores	
	2 - Exibe os highscore anteriores na tela
3 - Jogador insere seu nome	
4 - Jogador pressiona o botão “Save score”	
5 - Banco de dados salva o score atingido com o nome inserido Jogador pressiona o botão “Save score”	
	6 - Retorna ao menu principal
Regras de negócio	
[RN01] O nome é obrigatório para inserir um highscore	

Fluxo Alternativo - Score não é um Highscore	
	1 - Apenas exibe os highscores anteriores na tela
2 - Jogador pressiona “Return”	
	3 - Retorna ao menu principal
Fluxo Alternativo - Não quer highscore	
1 - Jogador pressiona o botão “Return”	
	2 - Retorna ao menu principal

Tabela 12. Caso de Uso 11

Nome	UC12: Passar de nível
Atores	Autor principal: Jogador
Descrição	Gera um novo nível do jogo
Pré-condições	Todos os inimigos morreram
Pós-condições	Um novo nível foi gerado

Fluxo Básico	
Ações dos atores	Ações do sistema
	1 - Exibe a mensagem de passar de nível
2 - O jogador pressiona qualquer tecla	
	3 - Recupera HP, AP e MP do jogador
	4 - Posiciona o jogador no canto do mapa
	5 - Gera inimigos e coloca-os no mapa.
	6 - Exibe os movimentos possíveis

	ao jogador, com o nome do movimento, seu custo de AP e de MP
	7 - Exibe o HP, MP e AP do jogador no canto da tela
	8 - Exibe os buffs equipamentos do jogador, com o nome e descrição de seus efeitos
9 - O jogador seleciona uma tecla correspondente a um movimento	
	10 - Vai para o caso de uso “Executar movimento”
Fluxo Alternativo - Menu de jogo	
1 - O jogador pressiona a tecla ESC em seu turno	
	2 - Vai para o caso de uso “Menu de jogo”

Tabela 13. Caso de Uso 12

Nome	UC13: Inicialização
Atores	Autor principal: Jogador
Descrição	Inicializa o programa
Pré-condições	O jogador executou o programa
Pós-condições	As interfaces foram carregadas

Fluxo Básico	
Ações dos atores	Ações do sistema
	1 - Carrega a biblioteca gráfica
	2 - Carrega a biblioteca de som
	3 - Inicializa as entradas de mouse e teclado

	4 - Carrega as configurações base
	5 - Abre o banco de dados
	6 - Cria a janela do jogo
	8 - Começa a tocar a música tema
	9 - Vai para o caso de uso “Abrir menu principal”
Fluxo de exceção	
	1a - A biblioteca gráfica não está presente. Um log é feito e o programa é terminado.
	2a - A biblioteca de som não está presente ou não pode ser iniciada. Um log é feito e o programa é terminado.
	3a - Nenhum mouse ou teclado foram identificados. Um log é feito do ocorrido.

Tabela 14. Caso de Uso 13

Nome	UC14: Mostrar créditos
Atores	Autor principal: Jogador
Descrição	Mostra os créditos do jogo
Pré-condições	O jogador pressionou o botão de mostrar créditos
Pós-condições	Os créditos foram exibidos

Fluxo Básico	
Ações dos atores	Ações do sistema
	1 - Abre a tela de créditos, com o botão “Return” e os créditos
2 - O jogador pressiona o botão	

“Return”	
	3 - Vai para o caso de uso “Menu principal”

Tabela 15. Caso de Uso 14

4.2 Diagrama de Classes

Por serem grandes demais para estas páginas, os diagramas de classes também encontram-se no arquivo Classes.asta

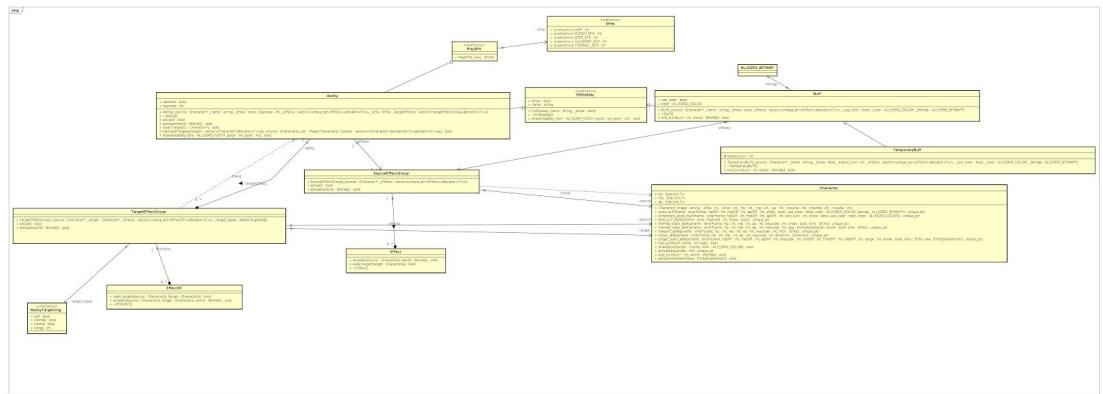


Figura 2. Diagrama de Classes 1

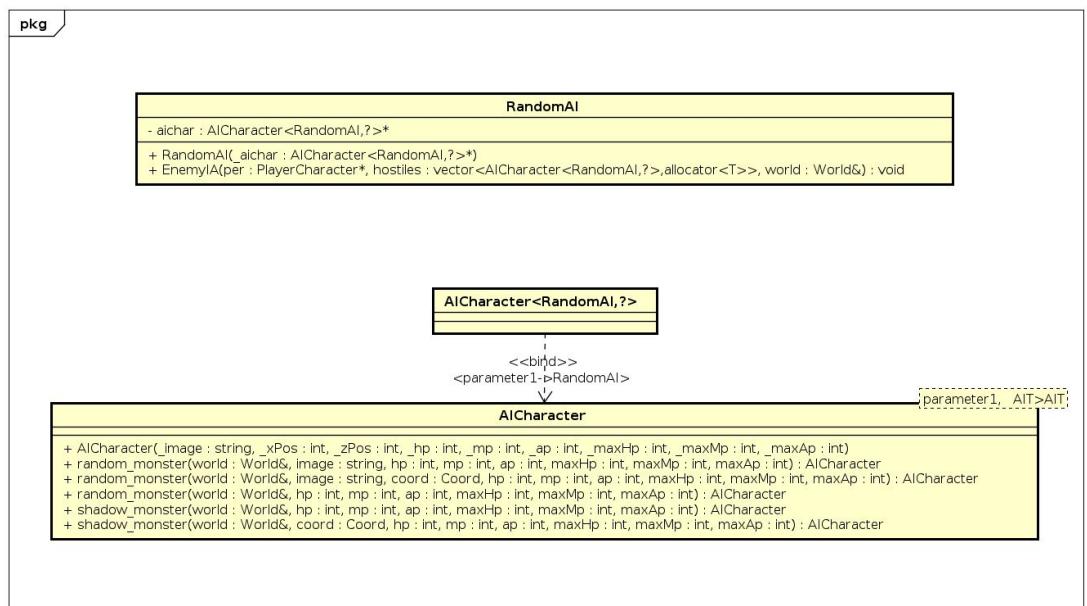


Figura 3. Diagrama de Classes 2

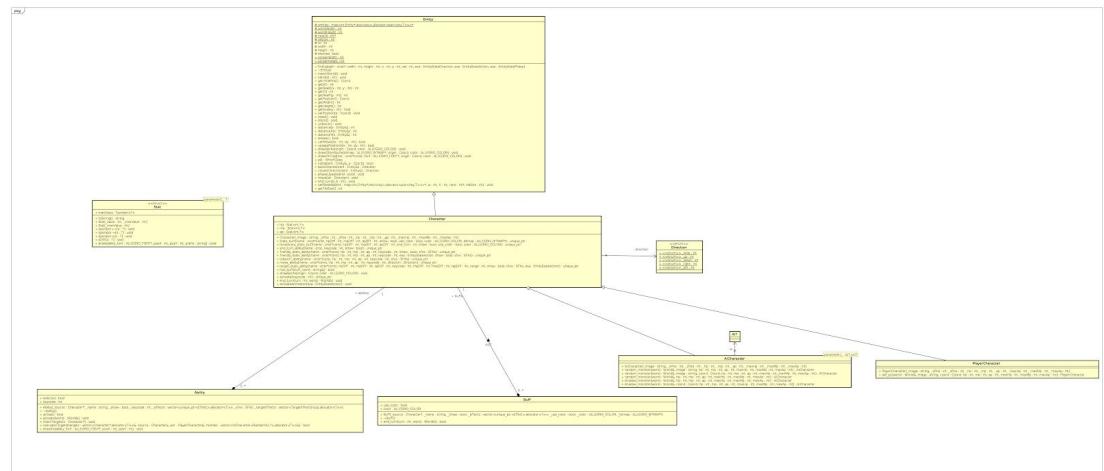


Figura 4. Diagrama de Classes 3

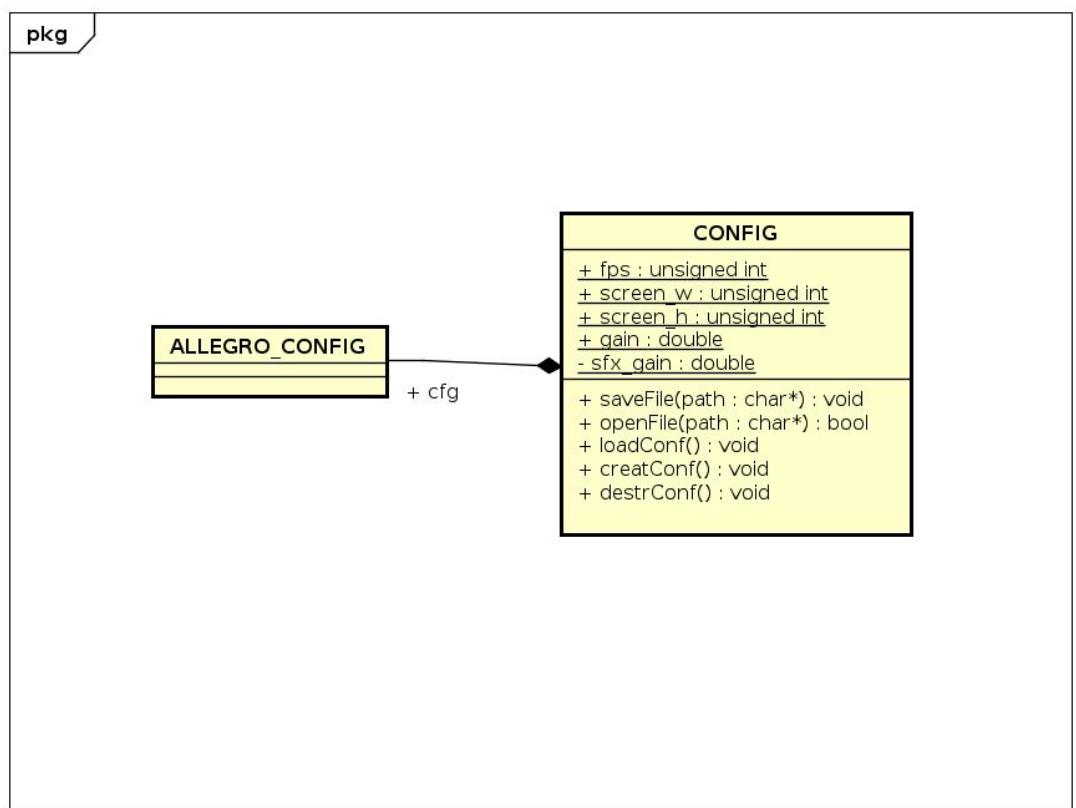


Figura 5. Diagrama de Classes 4

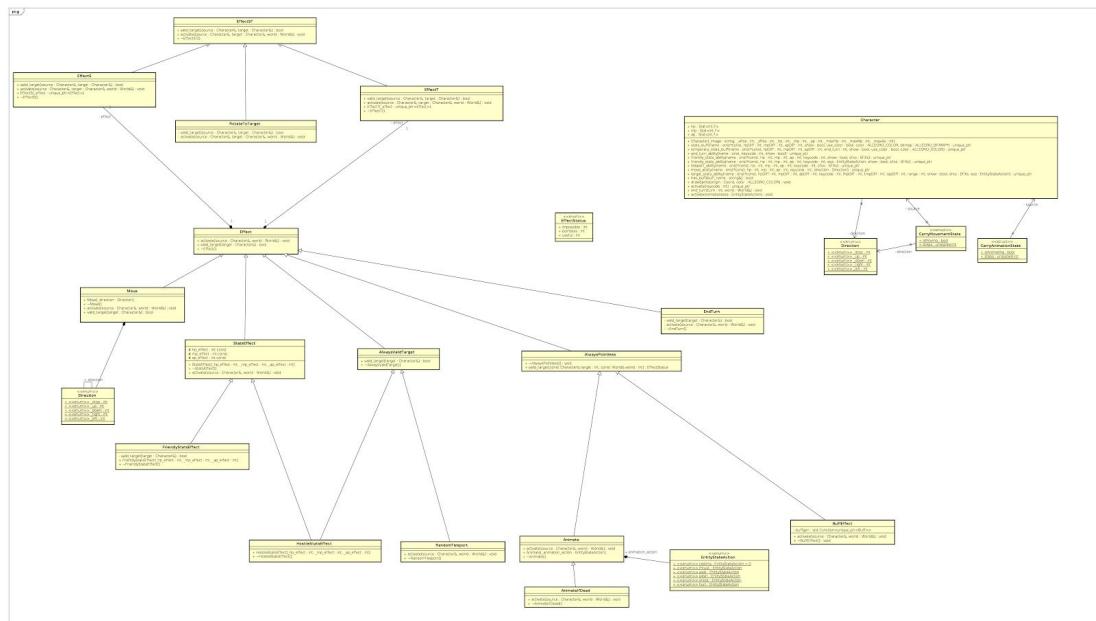


Figura 6. Diagrama de Classes 5

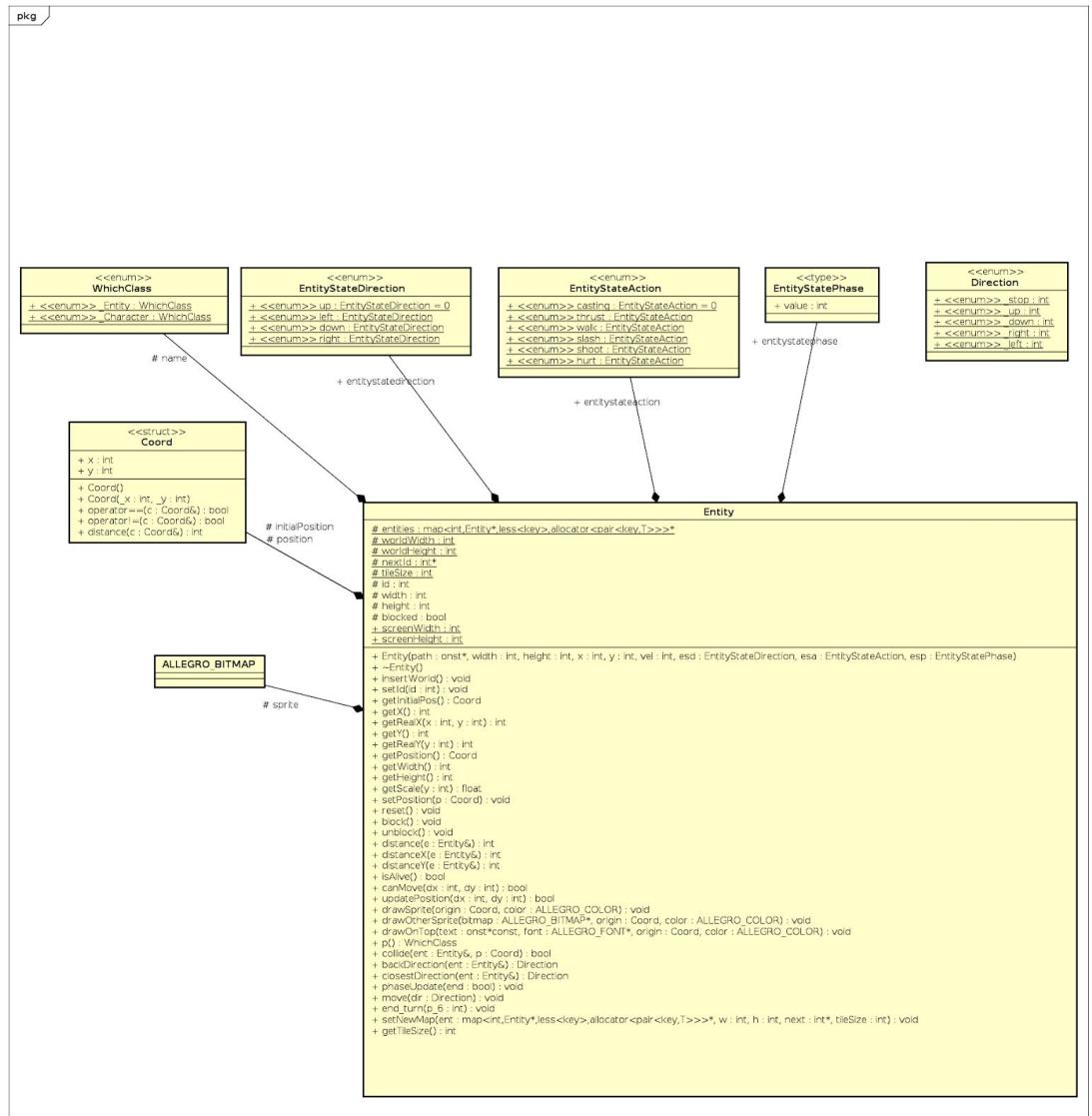


Figura 7. Diagrama de Classes 6

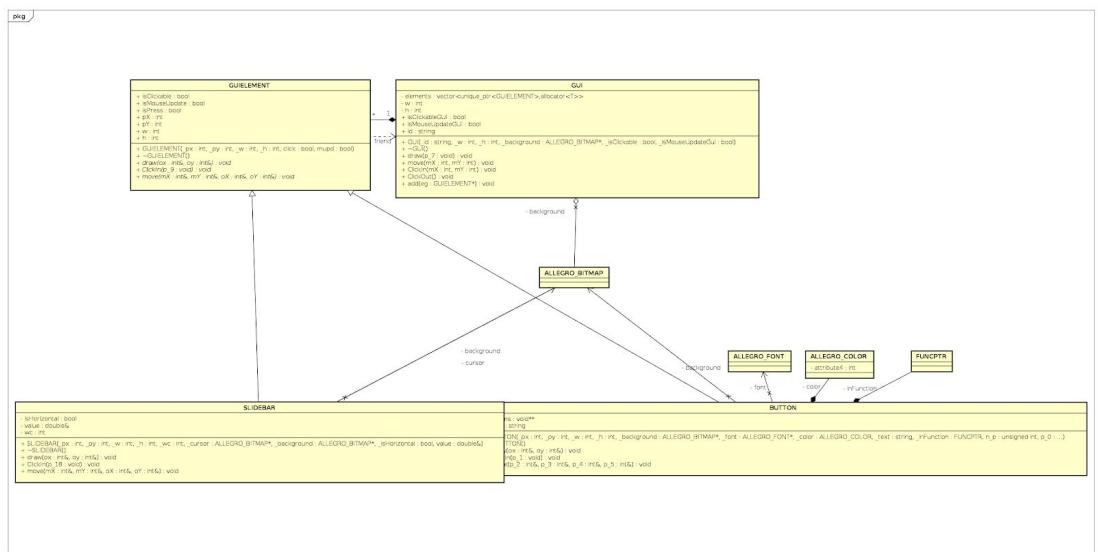


Figura 8. Diagrama de Classes 7

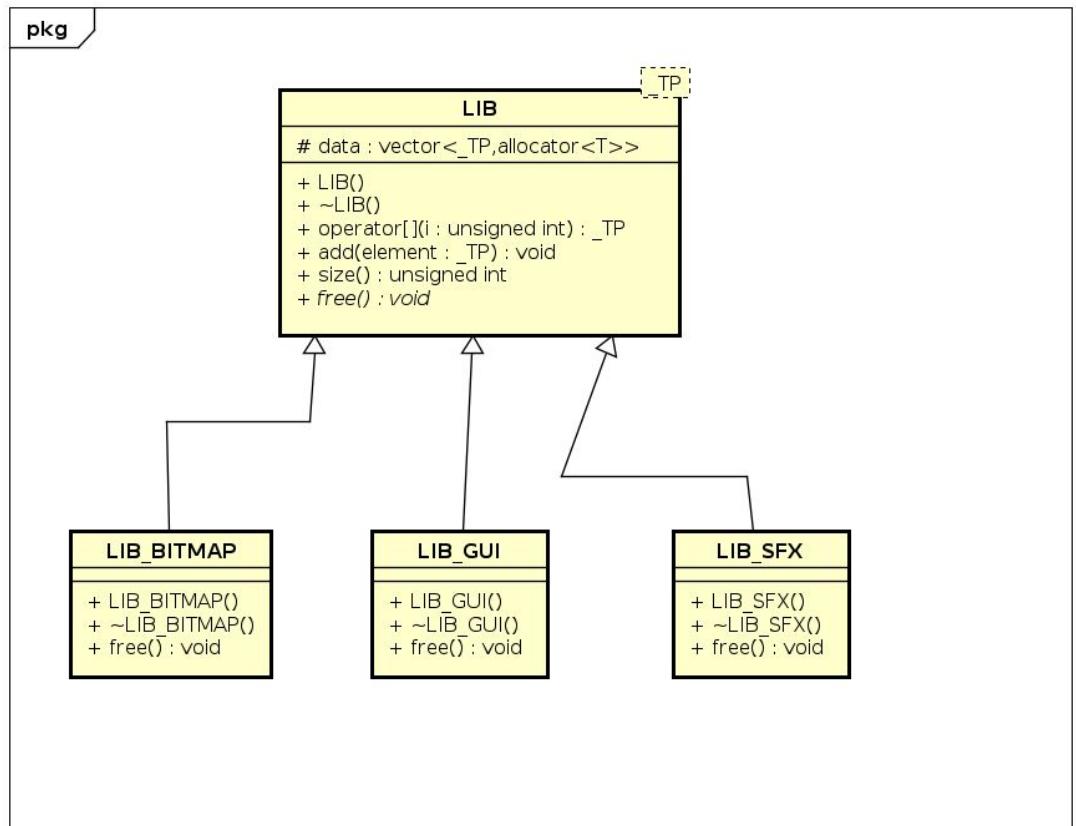


Figura 9. Diagrama de Classes 8

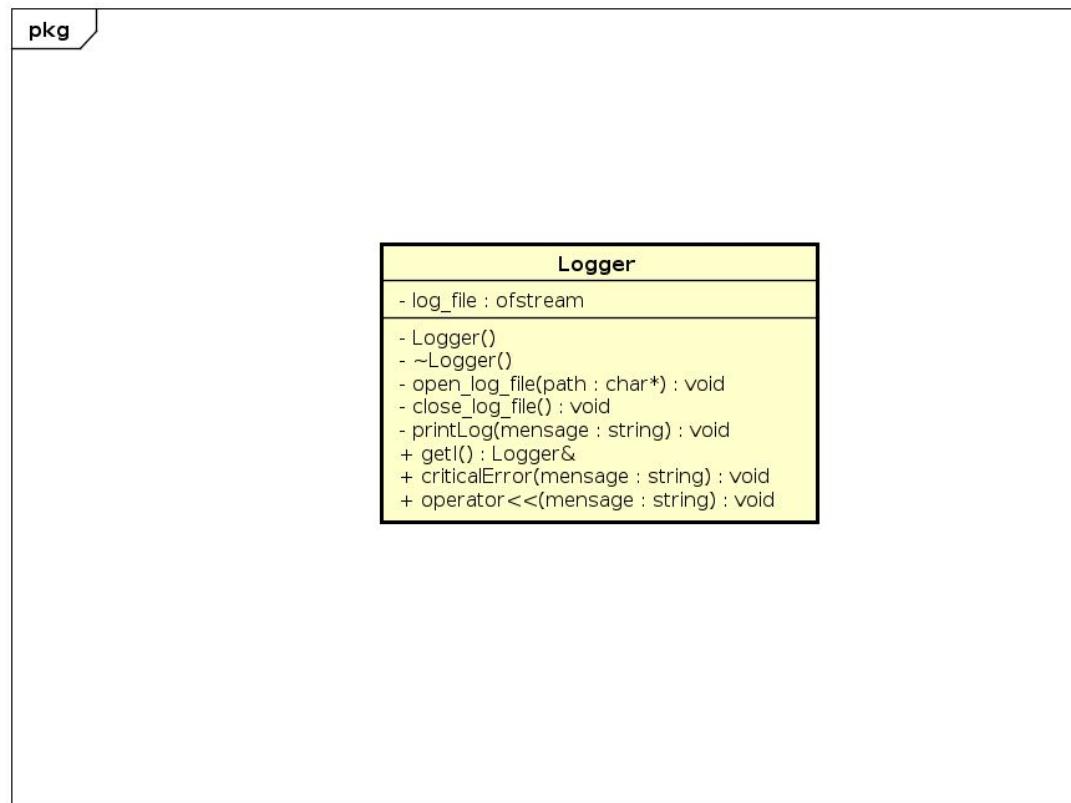


Figura 10. Diagrama de Classes 9

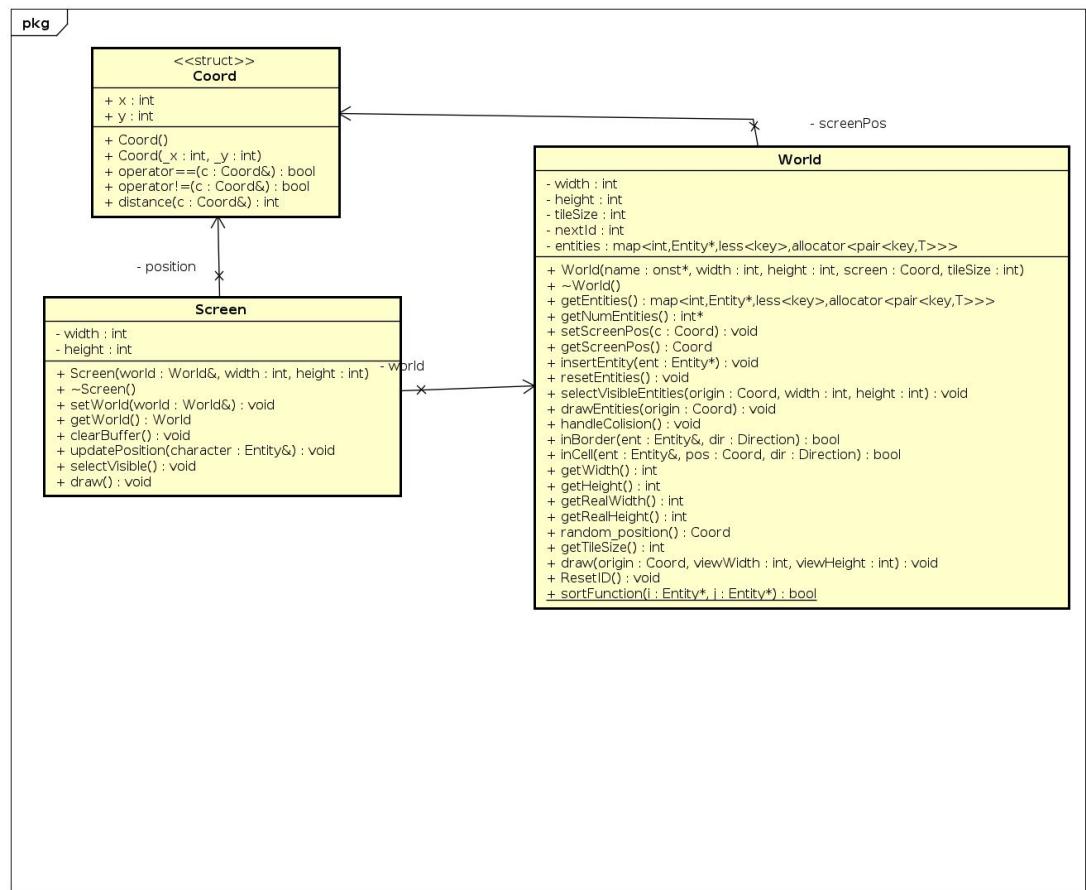


Figura 11. Diagrama de Classes 10

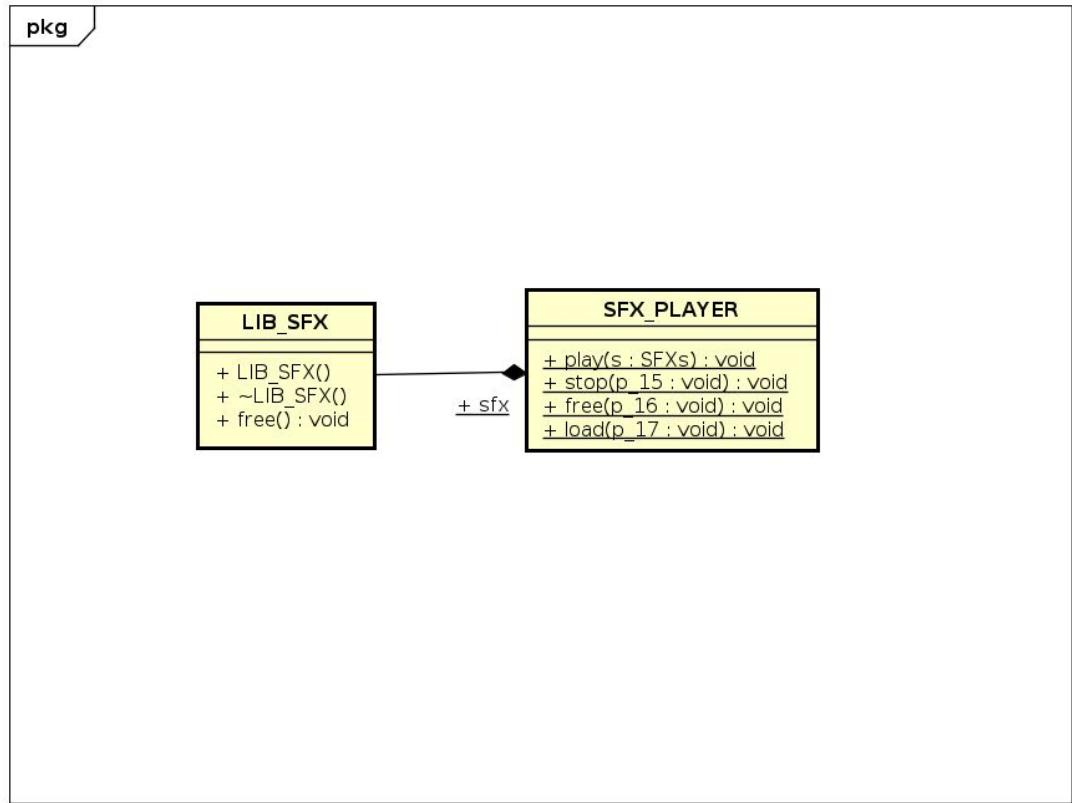


Figura 12. Diagrama de Classes 11

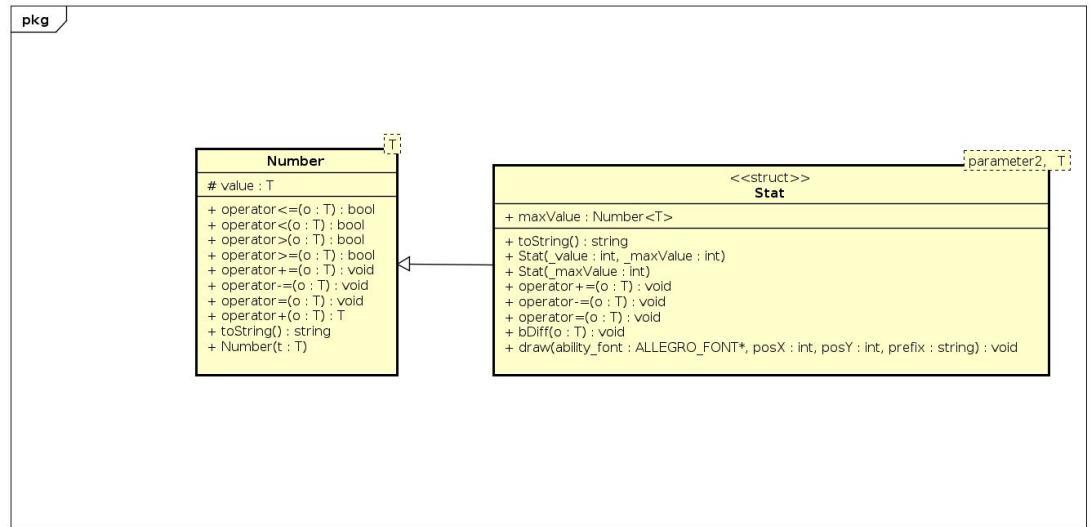


Figura 13. Diagrama de Classes 12



Figura 14. Diagrama de Classes 13

4.2.1 Diagrama de Instâncias

Esta seção contém os diagramas de instâncias. Primeiro, na Figura 16 o diagrama das instâncias relacionadas ao mundo e ao jogador. Na Figura 17 o diagrama das instâncias relacionadas a GUI.

4.2.1.1 Diagrama World

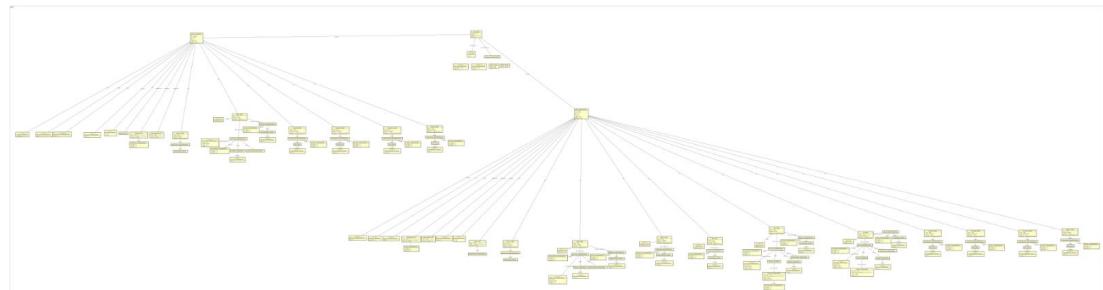


Figura 16. Diagrama de Instância World

4.2.1.2 Diagrama GUI/LIB

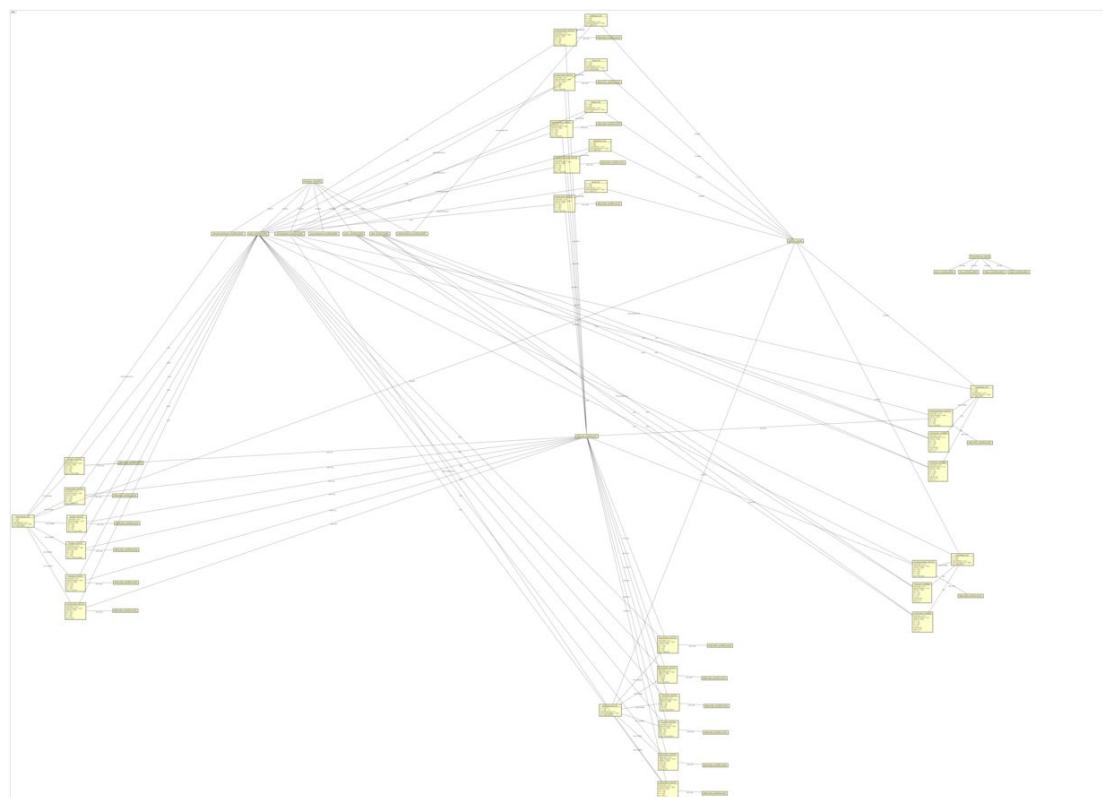


Figura 17. Diagrama de Instância GUI/LIB

4.3 Dicionário de Informações

Nas tabelas 15 a 42 a seguir, está o dicionário das classes do projeto, onde são apresentadas e explicadas todas as classes do programa e seus atributos para maior facilidade de compreensão e organização dos desenvolvedores.

Tabela 16. Dicionário da Classe PlayerCharacter

PlayerCharacter: Personagem do jogador					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
id	Identificador do personagem	≥ 0	Numérico	{9}	Contínuo
width	Largura do personagem	≥ 0	Numérico	{9}	Contínuo
height	Altura do personagem	≥ 0	Numérico	{9}	Contínuo
blocked	Personagem está bloqueado ou não		Booleano	B	Discreto [True False]
position	Posição atual do personagem	≥ 0	Par numérico	({9},{9})	
initialPosition	Posição inicial do personagem	≥ 0	Par numérico	({9},{9})	
sprite			Bitmap		
velocity	Velocidade do personagem	≥ 0	Numérico	{9}	Contínuo
hp	HP atual e máxima do personagem	≥ 0	Par numérico	({9},{9})	Contínuo
mp	MP atual e	≥ 0	Par	({9},{9})	Contínuo

	máxima do personagem		numérico		
ap	AP atual e máxima do personagem	≥ 0	Par numérico	({9},{9})	Contínuo, não negativo
abilities	Lista das habilidades disponíveis do personagem		Lista		
buffs	Lista dos buffs do personagem		Lista		

Tabela 17. Dicionário da Classe AICharacter

AICharacter: Personagem inimigo					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
id	Identificador do personagem	≥ 0	Numérico	{9}	Contínuo
width	Largura do personagem	≥ 0	Numérico	{9}	Contínuo
height	Altura do personagem	≥ 0	Numérico	{9}	Contínuo
blocked	Personagem está bloqueado ou não		Booleano	B	Discreto [True False]
position	Posição atual do personagem	≥ 0	Par numérico	({9},{9})	
initialPosition	Posição inicial do personagem	≥ 0	Par numérico	({9},{9})	
hp	HP atual e	≥ 0	Par	({9},{9})	Contínuo

	máxima do personagem		numérico		
mp	MP atual e máxima do personagem	≥ 0	Par numérico	({9},{9})	Contínuo
ap	AP atual e máxima do personagem	≥ 0	Par numérico	({9},{9})	Contínuo,
abilities	Lista das habilidades disponíveis do personagem		Lista		
buffs	Lista dos buffs do personagem		Lista		

Tabela 18. Dicionário da Classe Ability

Ability: Habilidade que pode ser executada por um personagem					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
keycode	Código da habilidade	≥ 0	Numérico	{9}	Contínuo
selected	Habilidade está selecionada para uso ou não		Booleano	B	Discreto [True False]

Tabela 19. Dicionário da Classe Buff

Buff: Buff recebido por um personagem					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
use_color	Usa ou não o efeito visual de cor do buff		Booleano	B	Discreto [True False]

effects	Efeitos que o buff causa		Efeito		
color	Cor do efeito visual		Cor	(HHH,HH H,HHH)	Contínuo
bitmap	Sprite do efeito visual		Bitmap		

Tabela 20. Dicionário da Classe TemporaryBuff

TemporaryBuff: Buff temporário					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
use_color	Usa ou não o efeito visual de cor do buff		Booleano	B	Discreto [True False]
effects	Efeitos que o buff causa		Lista		
color	Cor do efeito visual		Cor	(HHH,HH H,HHH)	Contínuo
bitmap	Sprite do efeito visual		Bitmap		
expire_turn	Turnos para que o buff acabe	> 0	Numérico	{9}	Contínuo

Tabela 21. Dicionário da Classe RandomAI

RandomAI: Inteligência de um inimigo do tipo aleatória					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio

aichar	Personagem controlado pela inteligência		Personagem		
--------	---	--	------------	--	--

Tabela 22. Dicionário da Classe EADisplay

EADisplay: Mostrador de efeitos e habilidades					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
name	Nome do efeito/habilidade	> 0	Alfabético	{A}	Contínuo
show	Está sendo exibido ou não		Booleano	B	Discreto [True False]

Tabela 23. Dicionário da Classe RotateToTarget

RotateToTarget: Efeito que rotaciona um personagem					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
Nenhum					

Tabela 24. Dicionário da Classe EndTurn

EndTurn: Efeito que termina o turno					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
Nenhum					

Tabela 25. Dicionário da Classe FriendlyStatsEffect

FriendlyStatsEffect: Efeito de status bom					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
hp_effect	Variação da HP	≥ 0	Numérico	{9}	Contínuo
mp_effect	Variação	≥ 0	Numérico	{9}	Contínuo

	da MP				
ap_effect	Variação da AP	>= 0	Numérico	{9}	Contínuo

Tabela 26. Dicionário da Classe HostileStatsEffect

HostileStatsEffect: Efeito de status ruim					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
hp_effect	Variação da HP	>= 0	Numérico	{9}	Contínuo
mp_effect	Variação da MP	>= 0	Numérico	{9}	Contínuo
ap_effect	Variação da AP	>= 0	Numérico	{9}	Contínuo

Tabela 27. Dicionário da Classe RandomTeleport

RandomTeleport: Efeito que teleports o personagem para algum lugar					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio

Tabela 28. Dicionário da Classe Animate

Animate: Efeito que realiza uma animação					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio

Tabela 29. Dicionário da Classe AnimatelfDead

AnimatelfDead: Efeito que realiza animação se o personagem morreu					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio

Tabela 30. Dicionário da Classe Move

Move: Efeito que move um personagem					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
direction	Direção do movimento		Direção		Discreto [Cima Baixo Esquerda Direita]

Tabela 31. Dicionário da Classe BuffEffect

BuffEffect: Efeito que ativa um buff					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
buffgen	Gerador do buff		Buff		

Tabela 32. Dicionário da Classe SourceEffectGroup

SourceEffectGroup: Grupo de efeitos afetando uma fonte					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
source	Fonte dos efeitos		Personagem		
effects	Lista dos efeitos afetando a fonte		Lista		

Tabela 33. Dicionário da Classe TargetEffectGroup

TargetEffectGroup: Grupo de efeitos afetando um alvo					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
source	Fonte dos efeitos		Personagem		
target	Alvo dos efeitos		Personagem		
effects	Lista de efeitos		Lista		

	afetando o alvo				
--	-----------------	--	--	--	--

Tabela 34. Dicionário da Classe BUTTON

BUTTON: Botão da interface gráfica					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
isClickable	É um objeto clicável		Booleano	B	Discreto [True False]
isMouseUp date	É modificável com mouse		Booleano	B	Discreto [True False]
isPress	Está pressionado		Booleano	B	Discreto [True False]
pX	Posição no eixo X		Numérico	{9}	Contínuo
pY	Posição no eixo Y		Numérico	{9}	Contínuo
w	Largura		Numérico	{9}	Contínuo
h	Altura		Numérico	{9}	Contínuo
background	Imagen de fundo		Bitmap		
font	Fonte do texto do botão		Fonte		
color	Cor do botão		Cor	(HHH,HH H,HHH)	
inFunction	Ação executada pelo botão		Função		

Tabela 35. Dicionário da Classe SLIDEBAR

SLIDE BAR: Barra de ajuste da interface gráfica					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
isClickable	É um objeto clicável		Booleano	B	Discreto [True False]
isMouseUpDate	É modificável com mouse		Booleano	B	Discreto [True False]
isPress	Está pressionado		Booleano	B	Discreto [True False]
pX	Posição no eixo X		Numérico	{9}	Contínuo ≥ 0
pY	Posição no eixo Y		Numérico	{9}	Contínuo ≥ 0
w	Largura		Numérico	{9}	Contínuo ≥ 0
h	Altura		Numérico	{9}	Contínuo ≥ 0
cursor	Sprite do cursor da barra		Bitmap		
background	Imagen de background		Bitmap		
isHorizontal	É uma barra horizontal		Booleano		
value	Valor da posição do cursor		Numérico	{9}	Contínuo, [0, 1]
WC					

Tabela 36. Dicionário da Classe GUI

GUI: Interface gráfica de usuário
--

Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
background	Background da GUI		Bitmap		
elements	Lista de elementos da GUI		Lista		
w	Largura		Numérico	{9}	Contínuo >= 0
h	Altura		Numérico	{9}	Contínuo >= 0
extraDraw	Coisas extras para serem desenhadas		Função		

Tabela 37. Dicionário da Classe Logger

Logger: Faz log das coisas					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
log_file	Arquivo onde vai ser escrito o log		Arquivo		

Tabela 38. Dicionário da Classe Screen

Screen: Contém características da tela do jogo					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
position	Posição da tela		Par numérico	({9},{9})	Contínuo >= 0
width	Largura		Numérico	{9}	Contínuo > 0
height	Altura		Numérico	{9}	Contínuo > 0

world	Mundo que está sendo exibido na tela		Mundo		
-------	--------------------------------------	--	-------	--	--

Tabela 39. Dicionário da Classe LIB_BITMAP

LIB_BITMAP: Faz a interface entre a biblioteca gráfica e os bitmaps exibidos					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
data	Conjunto de dados do pacote		Lista		

Tabela 40. Dicionário da Classe LIB_GUI

LIB_GUI: Faz a interface entre a biblioteca gráfica e a interface gráfica					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
data	Conjunto de dados do pacote		Lista		

Tabela 41. Dicionário da Classe LIB_SFX

LIB_SFX: Faz a interface da biblioteca de som					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
data	Conjunto de dados do pacote		Lista		

Tabela 42. Dicionário da Classe SFX_PLAYER

SFX_PLAYER: Toca os efeitos sonoros					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
sfx	Efeitos sonoros do jogador		LIB_SFX		

Tabela 43. Dicionário da Classe World

World: Contém todas as entidades e informações sobre o mapa					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
width	Largura do mapa		Numérico	{9}	Contínuo >= 0
height	Altura do mapa		Numérico	{9}	Contínuo >= 0
tileSize	Tamanho de uma casa		Numérico	{9}	Contínuo >= 0
nextId	id da próxima entidade a ser inserida no mundo		Numérico	{9}	Contínuo >= 0
screenPos	Posição do mapa na tela		Par numérico	({9},{9})	Contínuo >= 0
background	Imagen de fundo do mapa		Bitmap		
visibleEntities	Lista das entidades visíveis		Lista		
entities	Mapa de todas as entidades existentes		Mapa		

4.4 Diagrama de Sequência

Nas imagens 18 a 31 é apresentado os diagramas de sequência, mostrando a interação no tempo entre vários objetos. O arquivo astah está em anexo e contém o diagrama original.

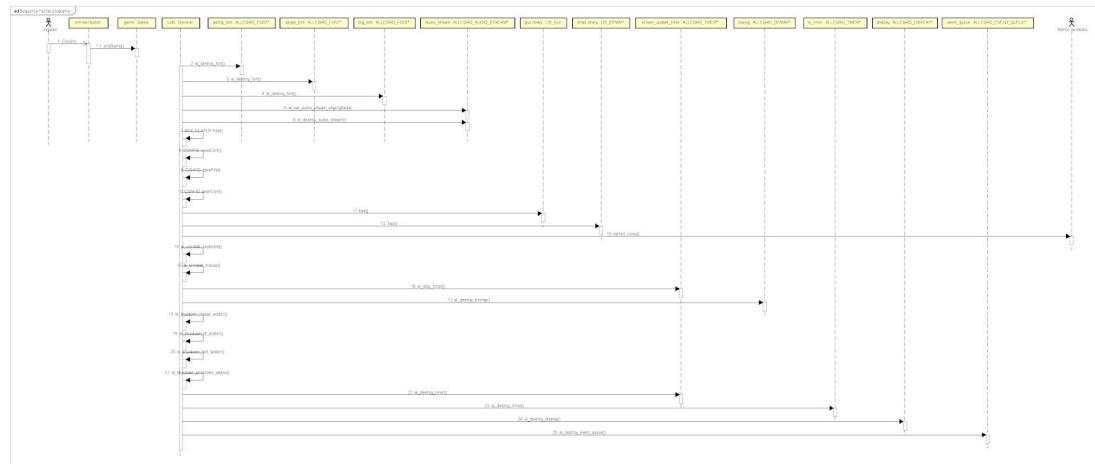


Figura 18. Diagrama de Sequência Fechar Programa

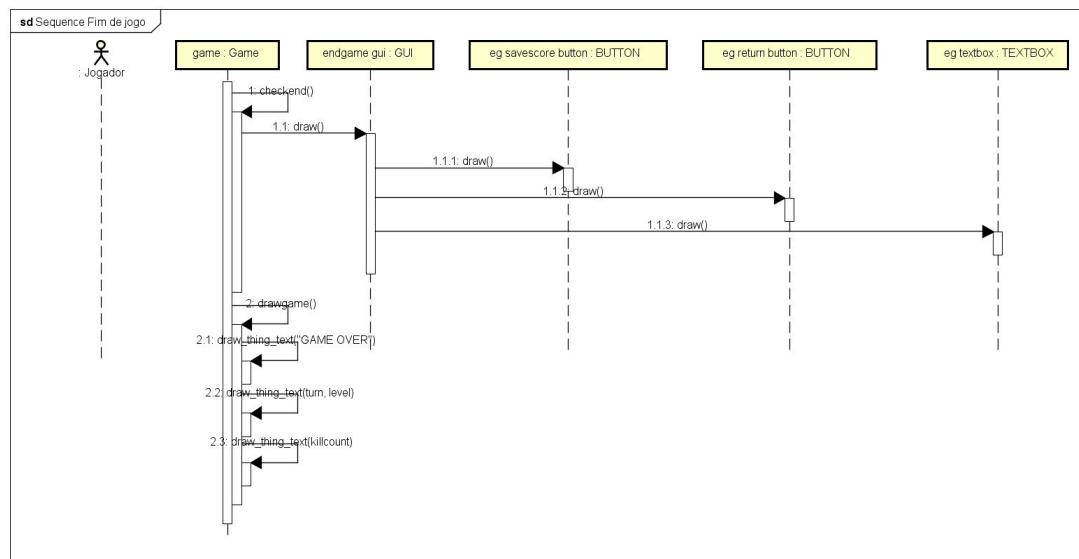


Figura 19. Diagrama de Sequência Fim de Jogo

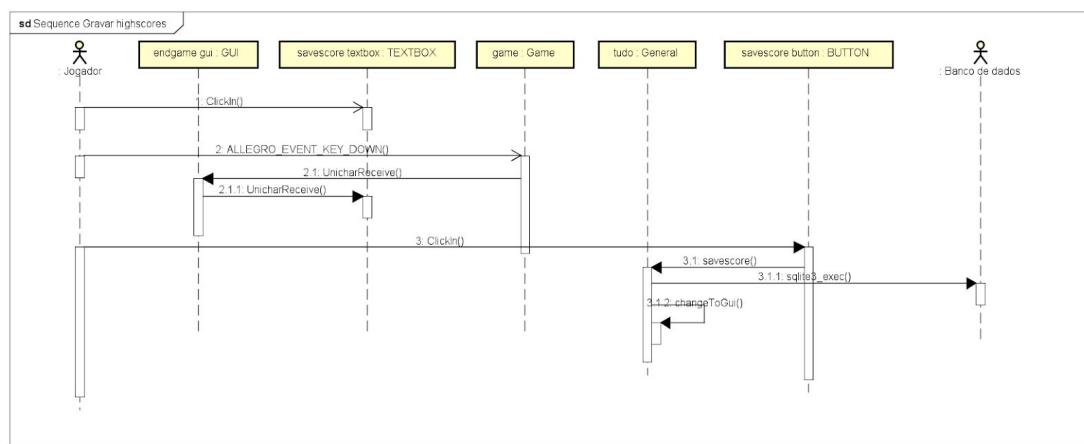


Figura 20. Diagrama de Sequência Gravar *Highscores*

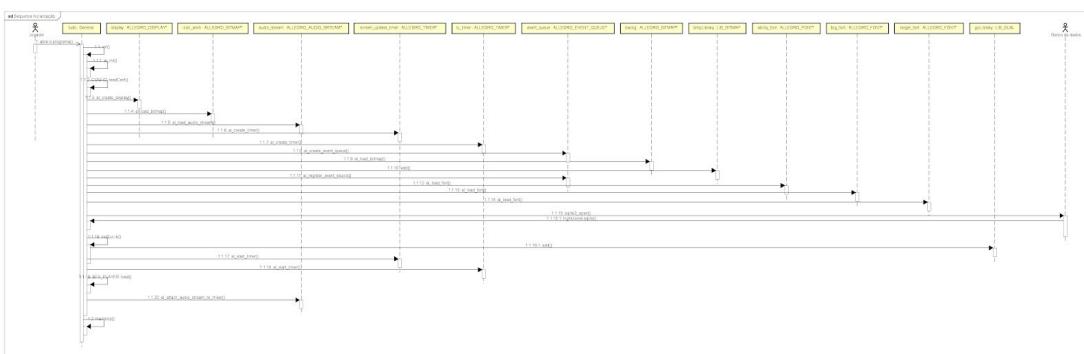


Figura 21. Diagrama de Sequência Inicialização

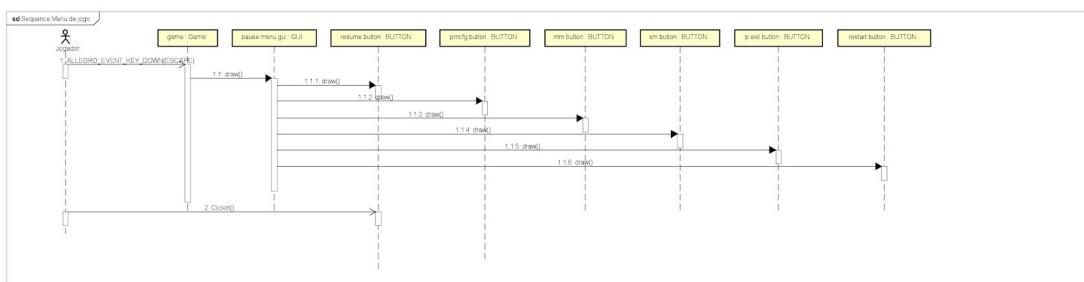


Figura 22. Diagrama de Sequência Menu de Jogo

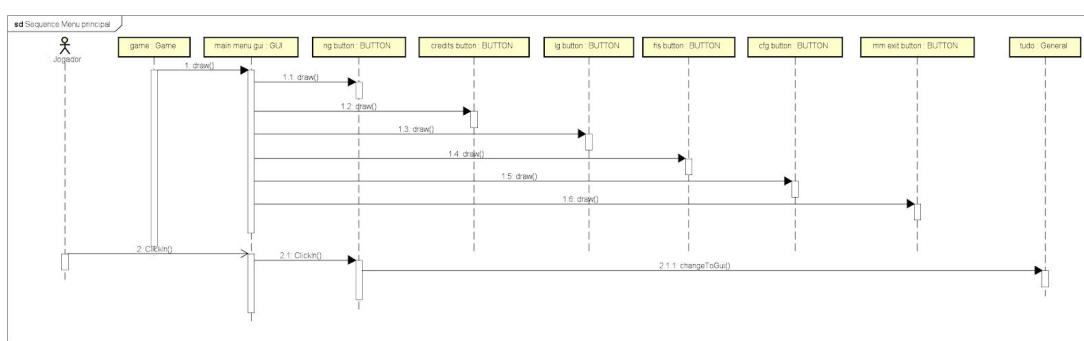


Figura 23. Diagrama de Sequência Menu Principal

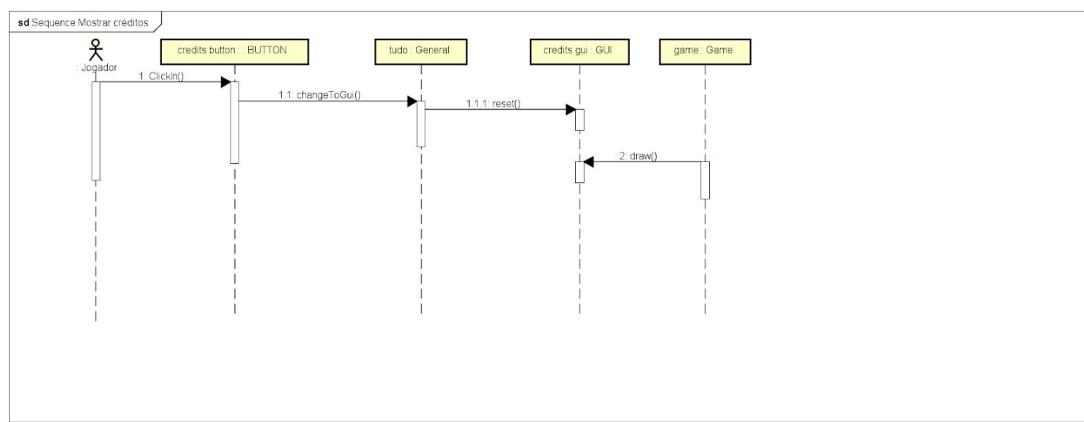


Figura 24. Diagrama de Sequência Mostrar Créditos

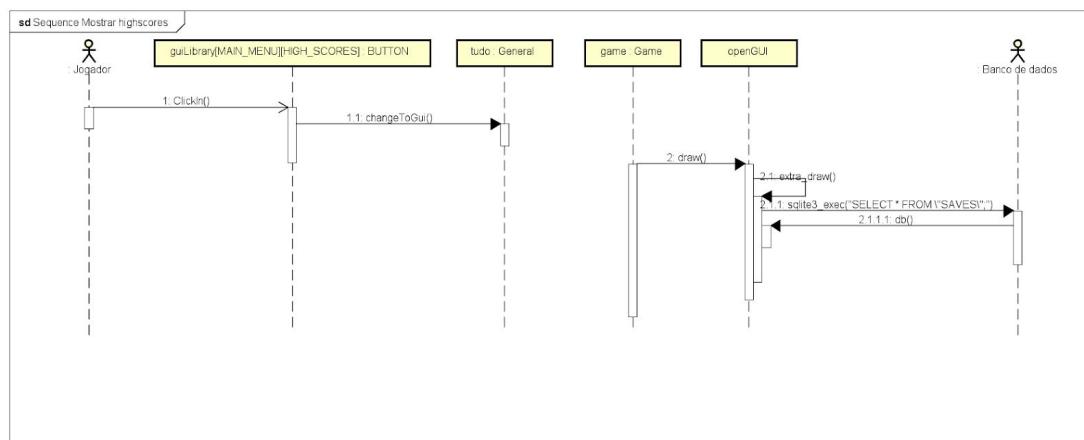


Figura 25. Diagrama de Sequência Mostrar *Highscores*

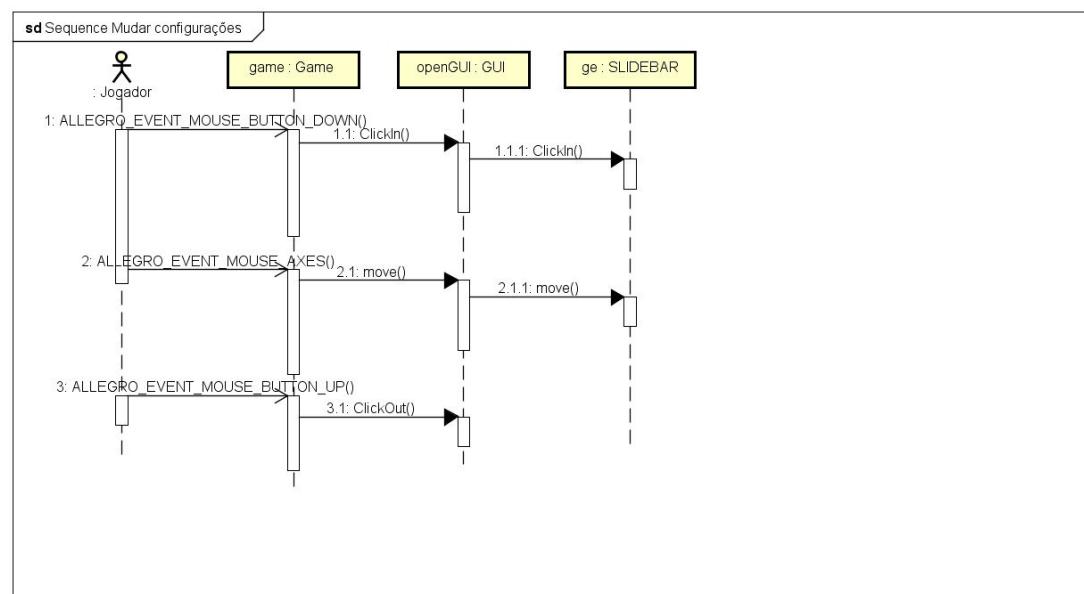


Figura 26. Diagrama de Sequência Mudar Configurações

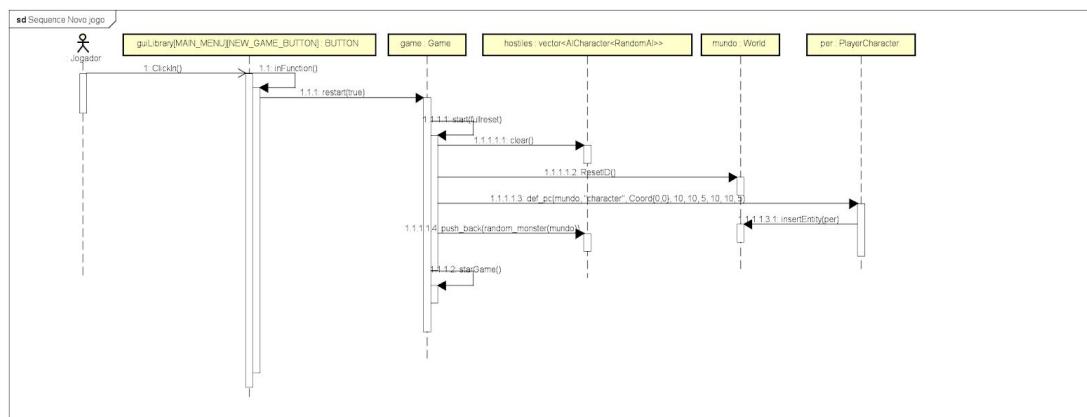


Figura 27. Diagrama de Sequência Novo Jogo

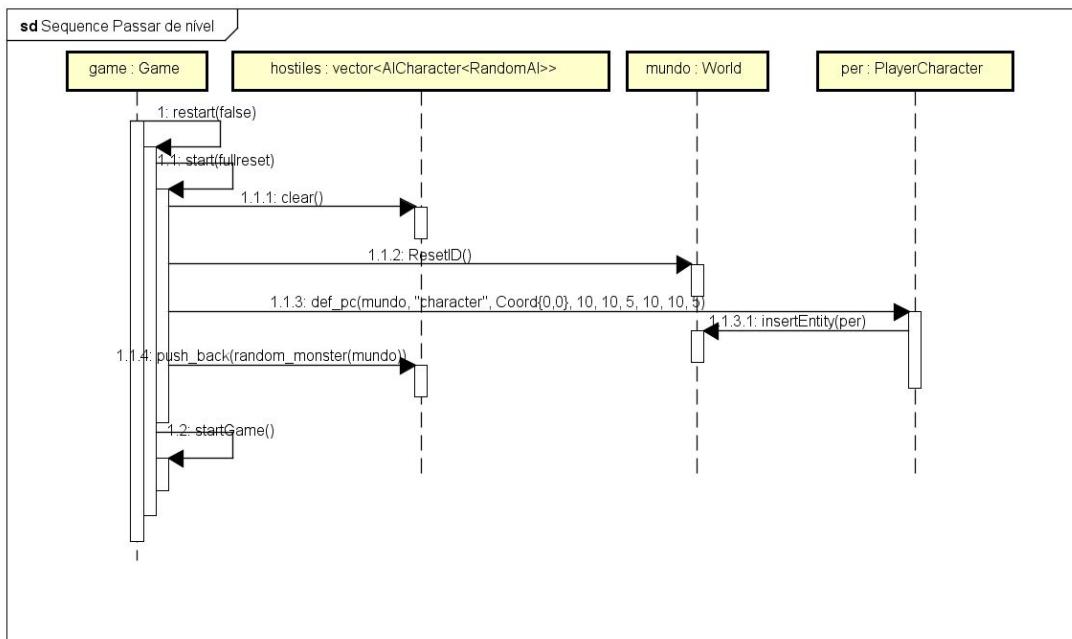


Figura 28. Diagrama de Sequência Passar de Nível

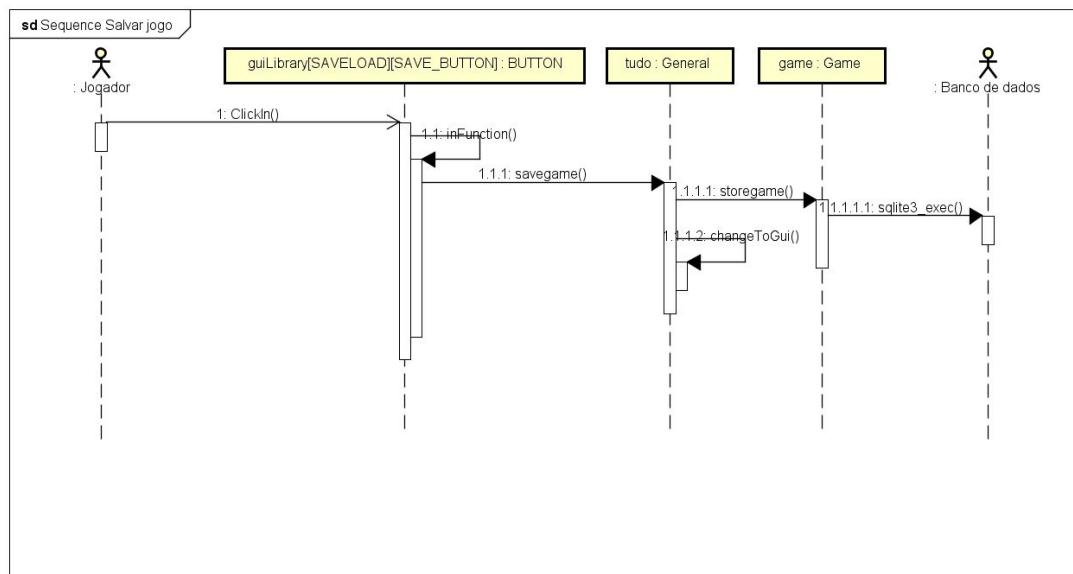


Figura 29. Diagrama de Sequência Salvar Jogo

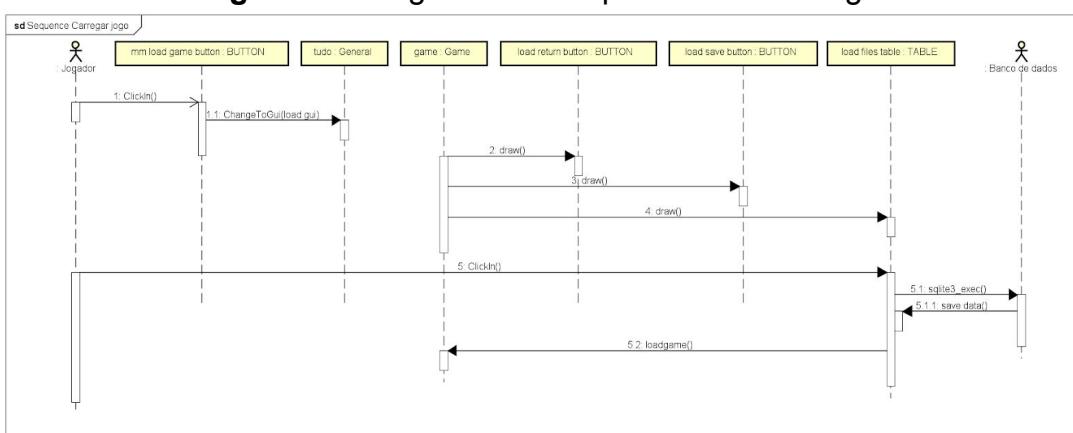


Figura 30. Diagrama de Sequência Carregar Jogo

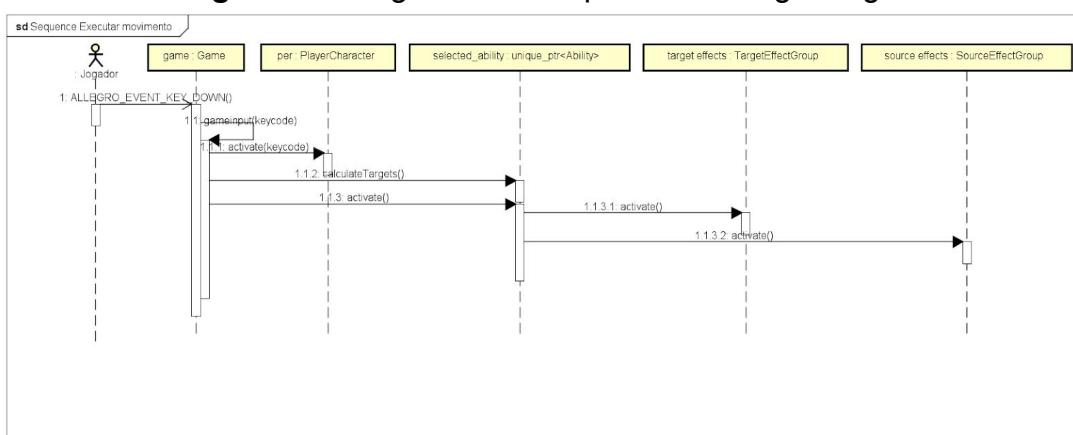


Figura 31. Diagrama de Sequência Executar Movimento

4.5 Diagrama de Comunicação

Segue nas figuras 32 a 44 o diagrama de comunicação, derivado do diagrama de sequência. Estes diagramas apresentam a forma e organização da comunicação e interação entre os vários objetos em vários casos de uso.

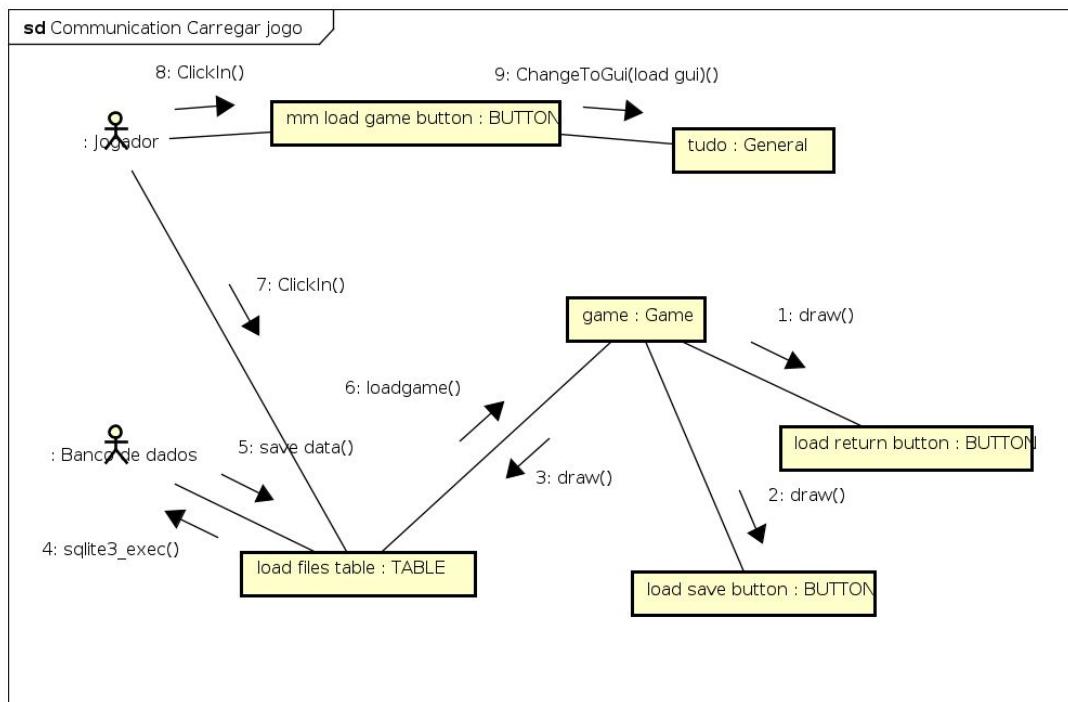


Figura 32. Diagrama de Comunicação Carregar jogo

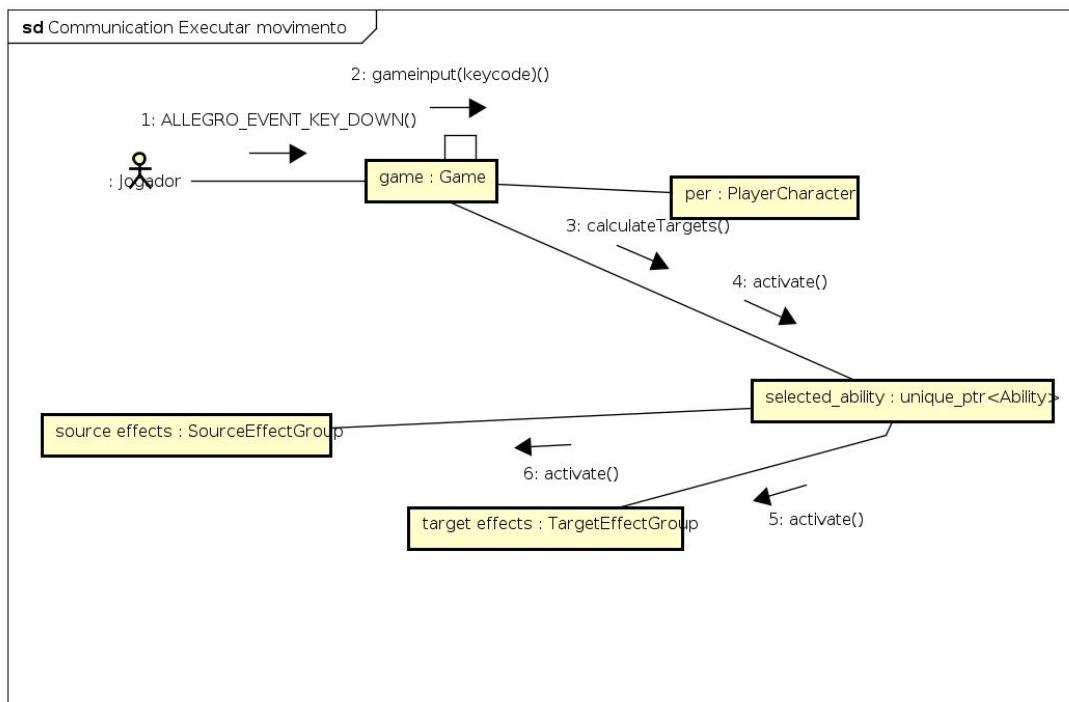


Figura 33. Diagrama de Comunicação Executar movimento

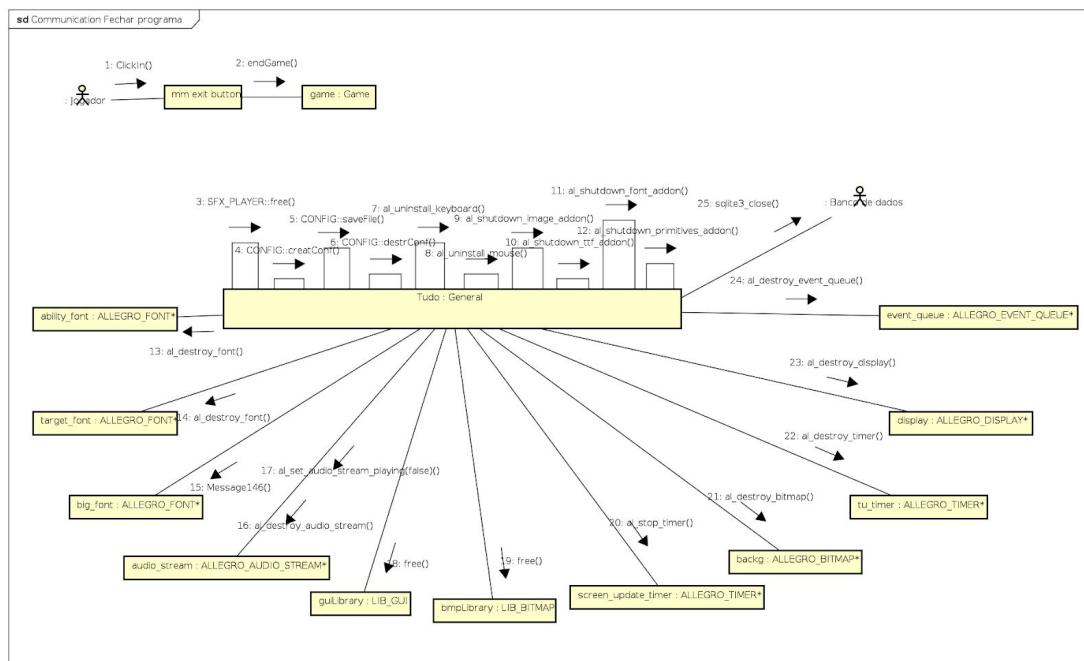


Figura 34. Diagrama de Comunicação Fechar programa

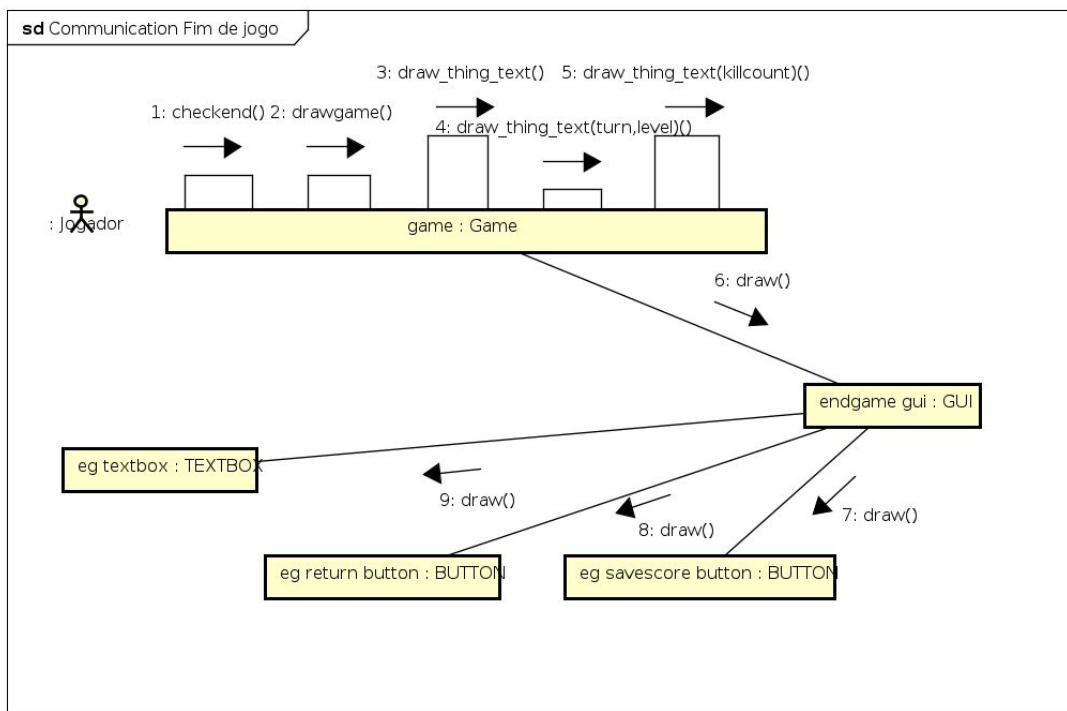


Figura 35. Diagrama de Comunicação Fim de jogo

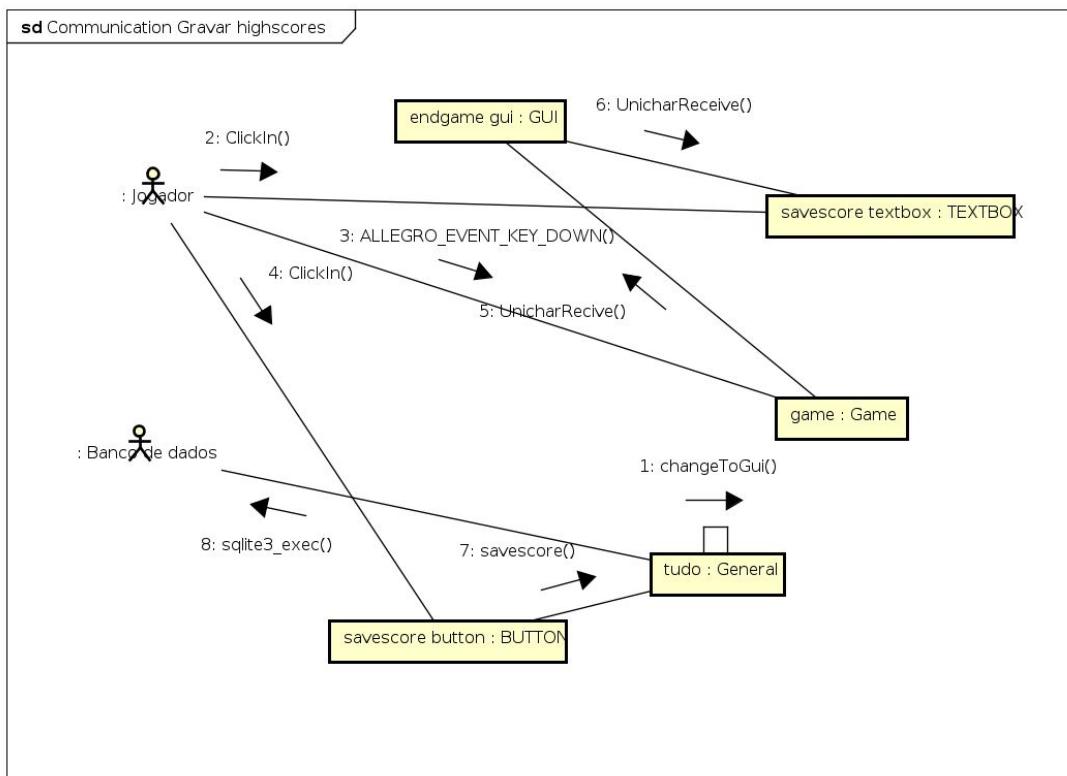


Figura 36. Diagrama de Comunicação Gravar highscores

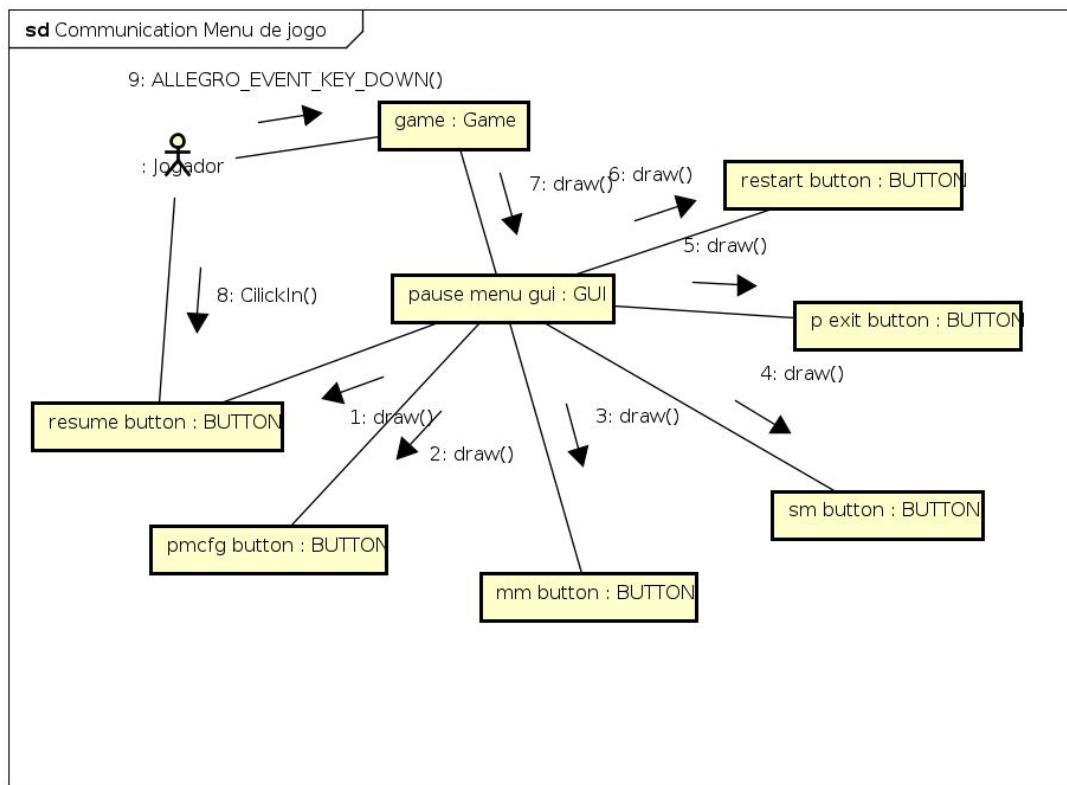


Figura 37. Diagrama de Comunicação Menu de jogo

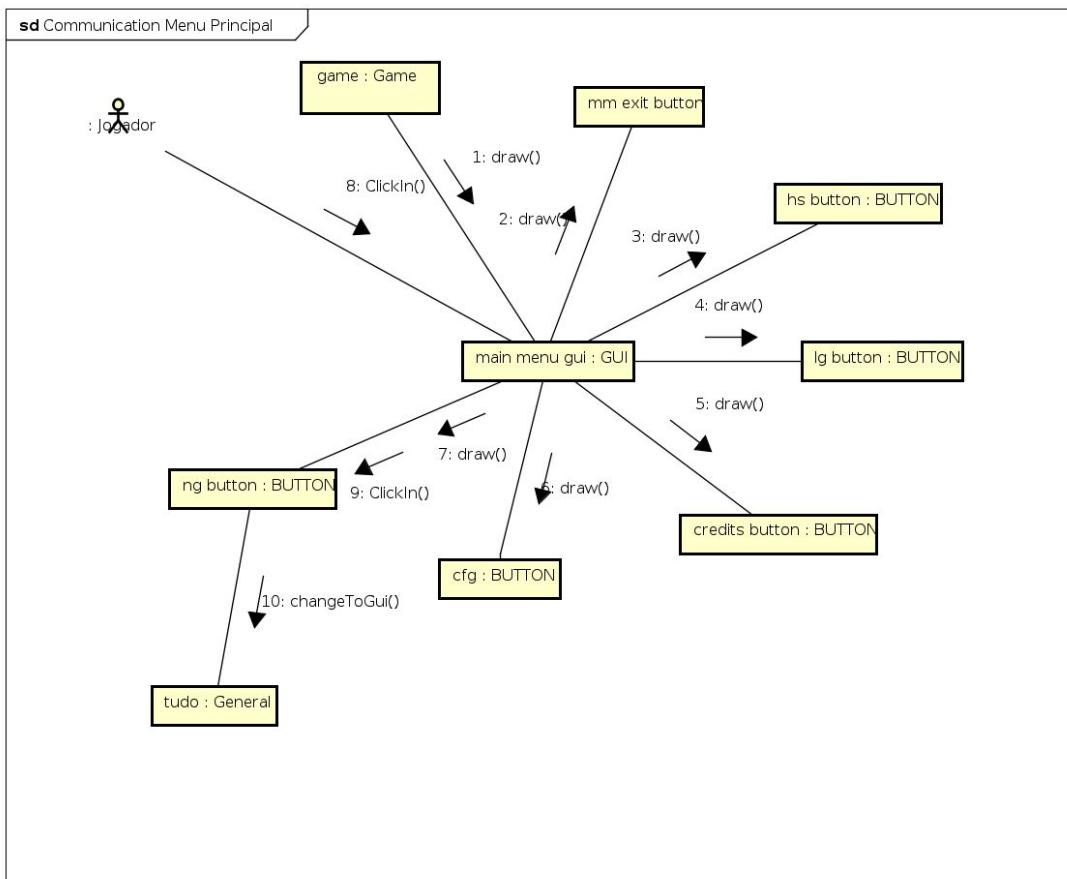


Figura 38. Diagrama de Comunicação Menu principal

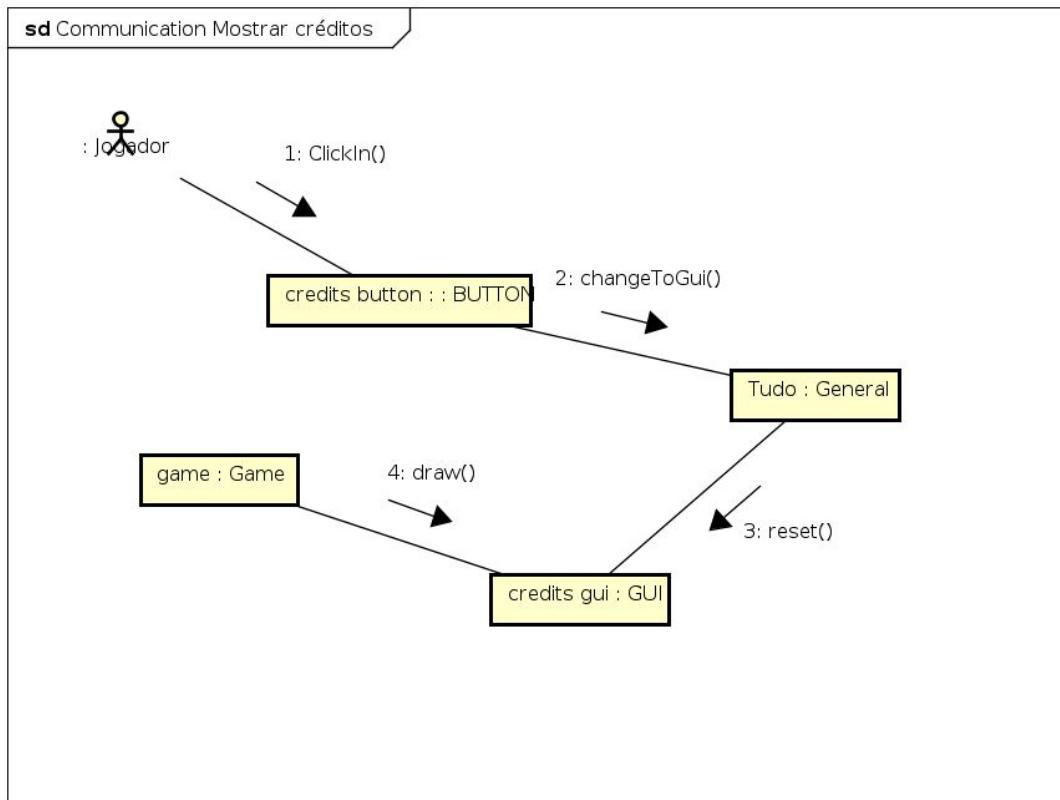


Figura 39. Diagrama de Comunicação Mostrar créditos

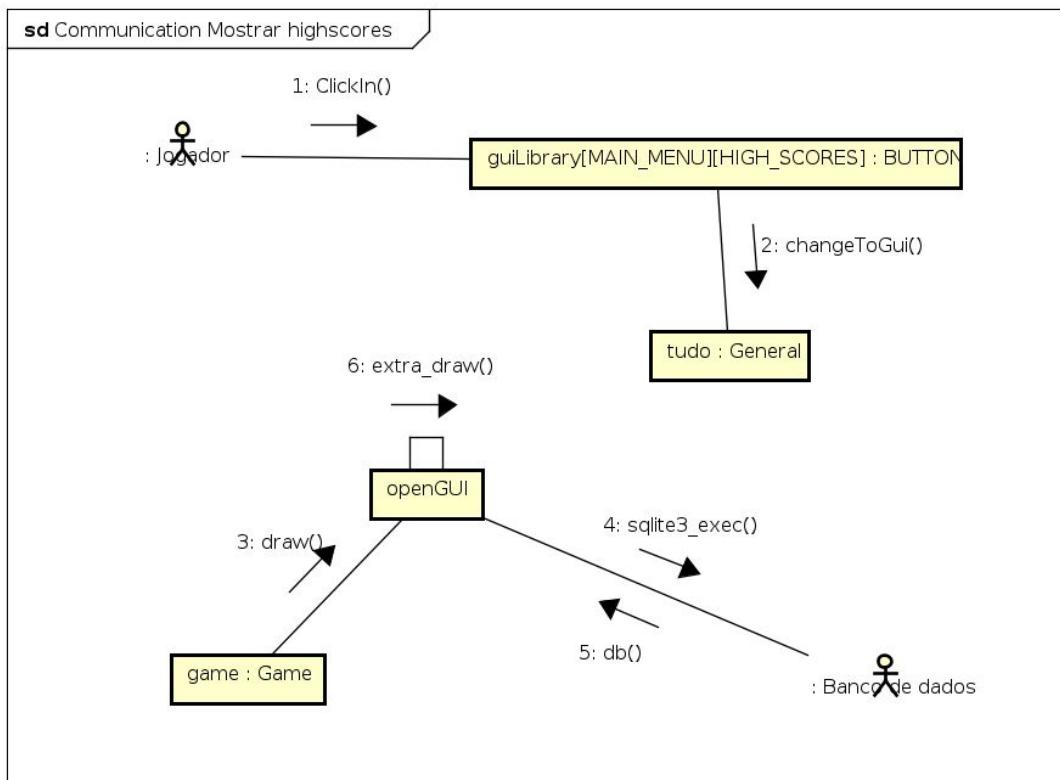


Figura 40. Diagrama de Comunicação Mostrar highscores

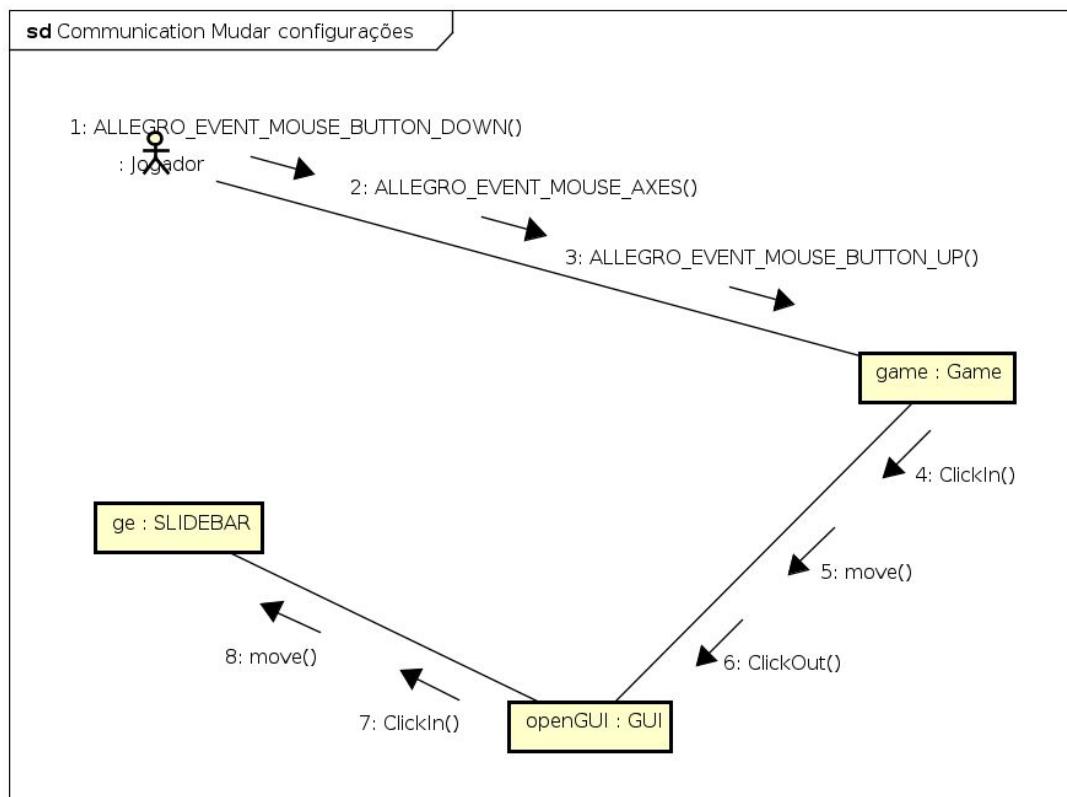


Figura 41. Diagrama de Comunicação Mudar configurações

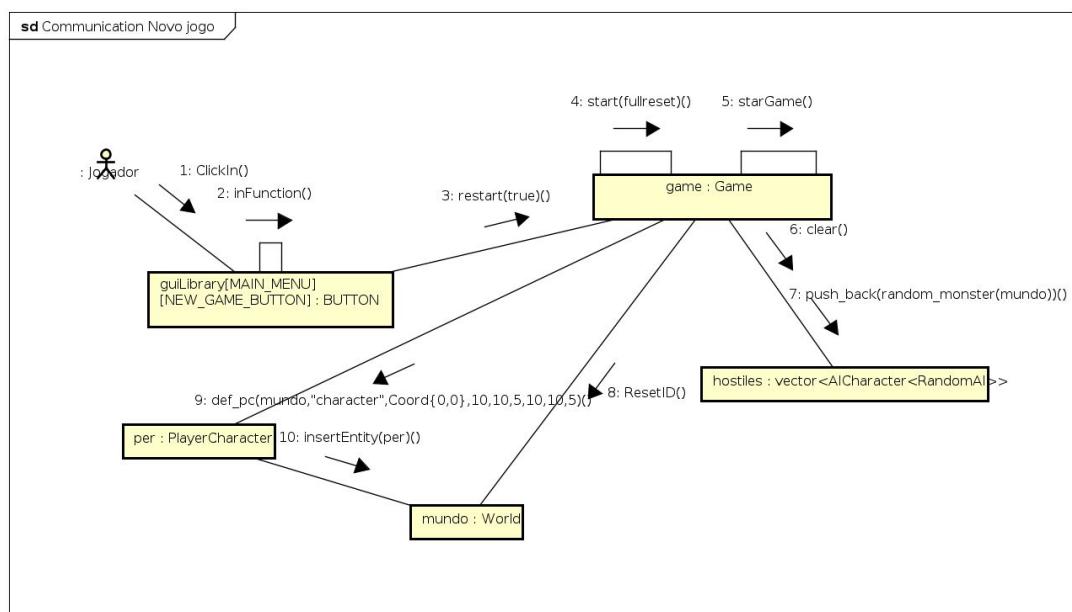


Figura 42. Diagrama de Comunicação Novo jogo

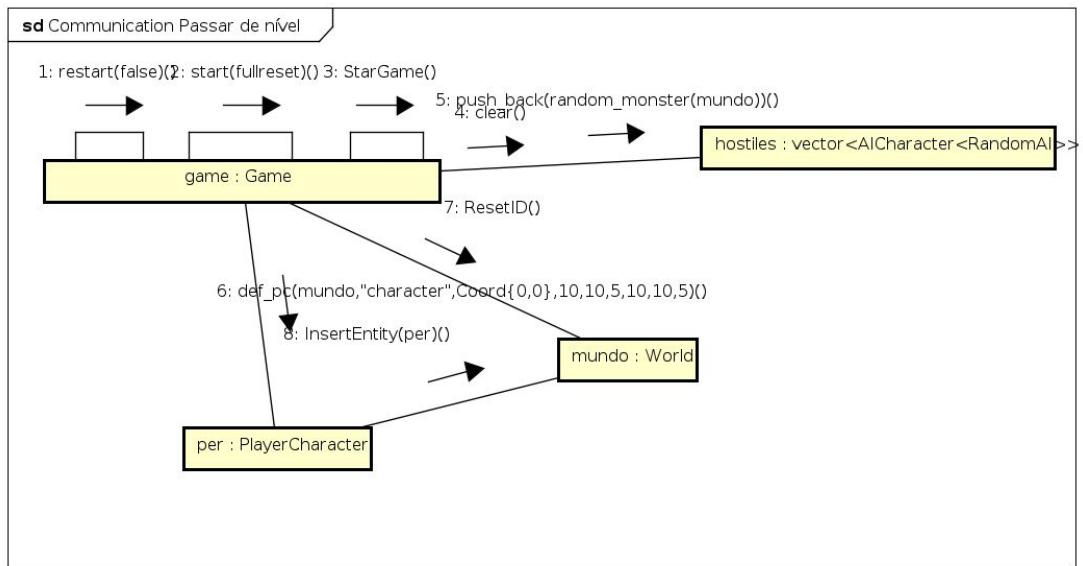


Figura 43. Diagrama de Comunicação Passar de nível

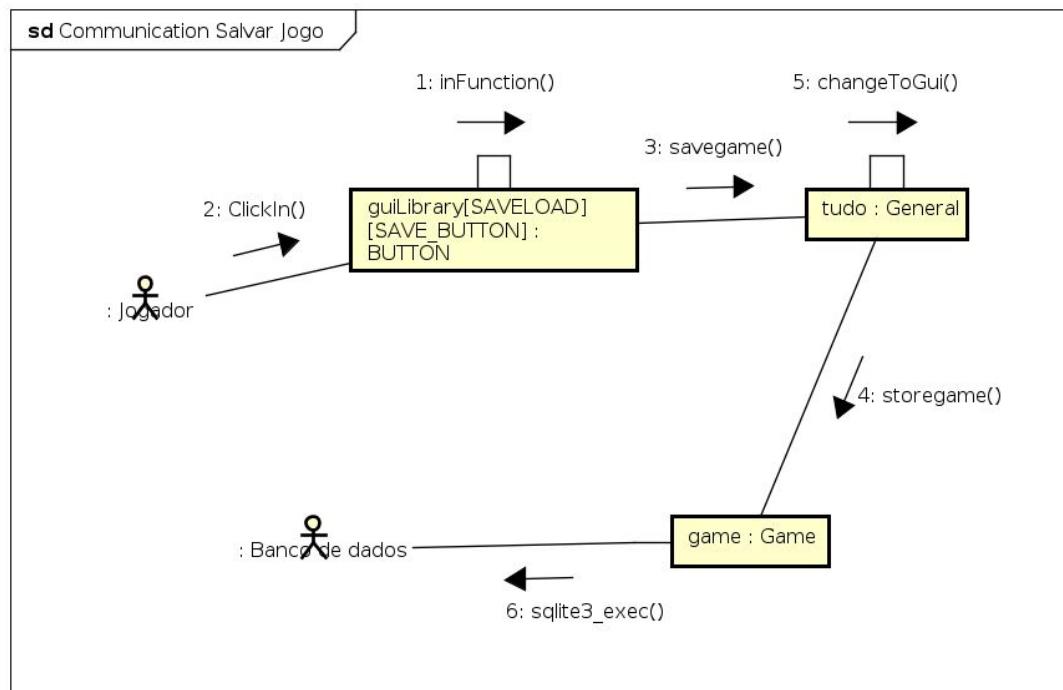


Figura 44. Diagrama de Comunicação Salvar jogo

4.6 Diagrama Entidade Relacionamento

Segue o diagrama entidade relacionamento na figura 45, que mostra a estrutura do banco de dados utilizado para guardar as informações de recordes e de jogos salvos.

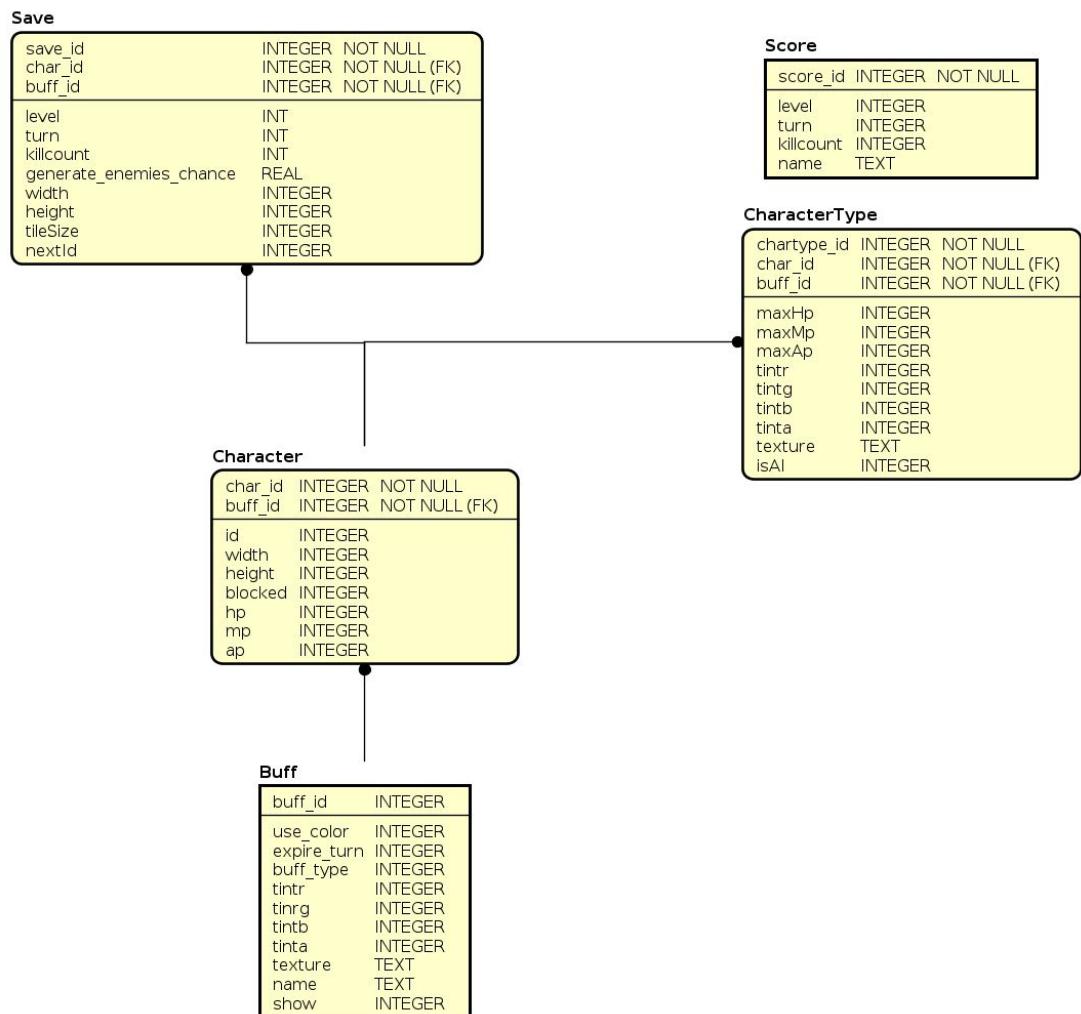


Figura 45. Diagrama de Entidade-relacionamento

4.7 Diagrama Estados

Segue nas figuras 46 a 89, o diagrama de estados, apresentando o comportamento das classes em relação a cada caso de uso.

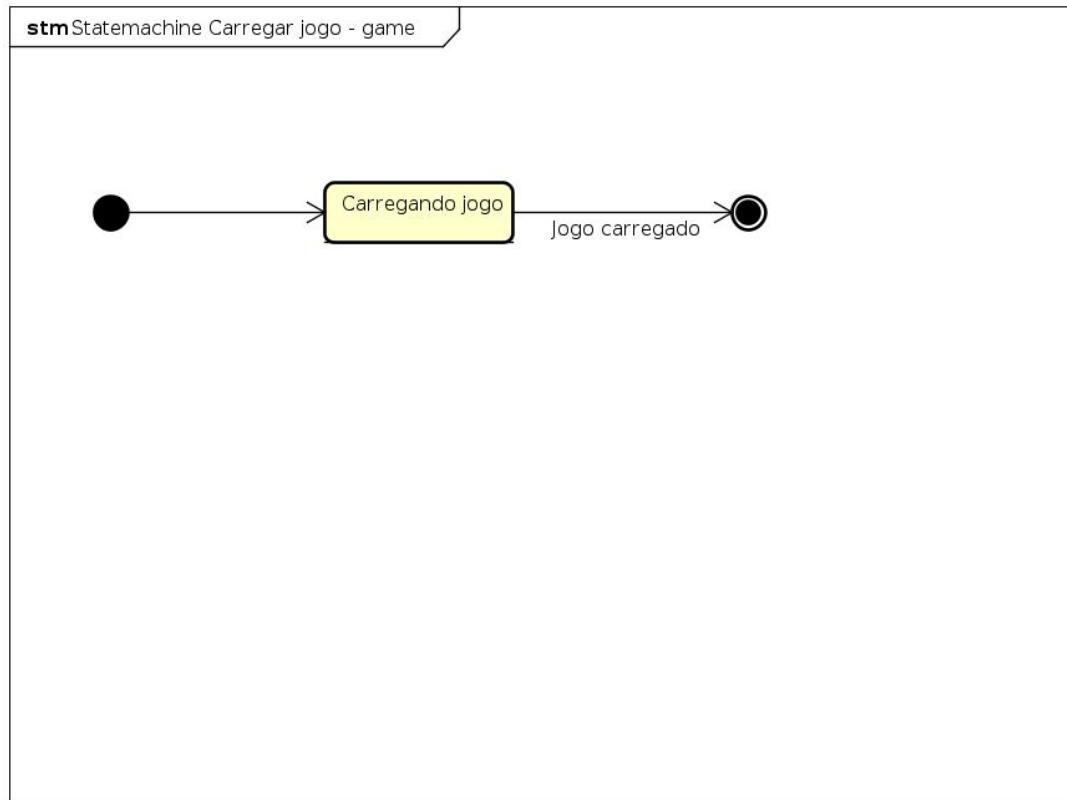


Figura 46. Diagrama Estados - Carregar jogo - game

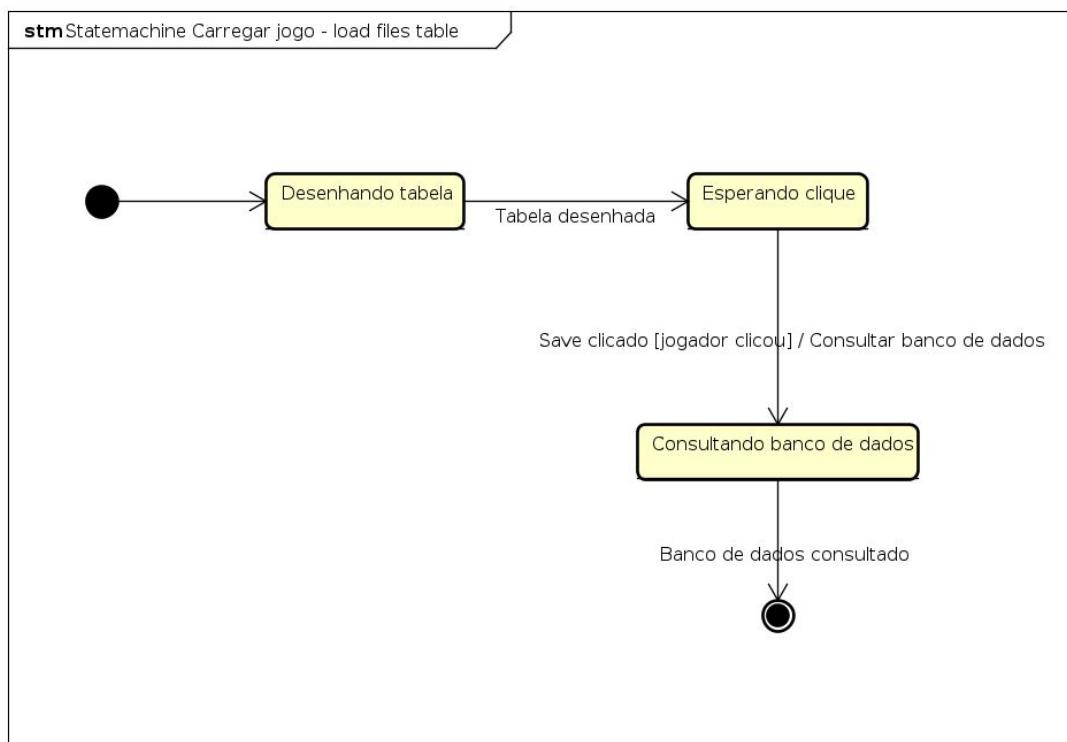


Figura 47. Diagrama Estados - Carregar jogo - load files table

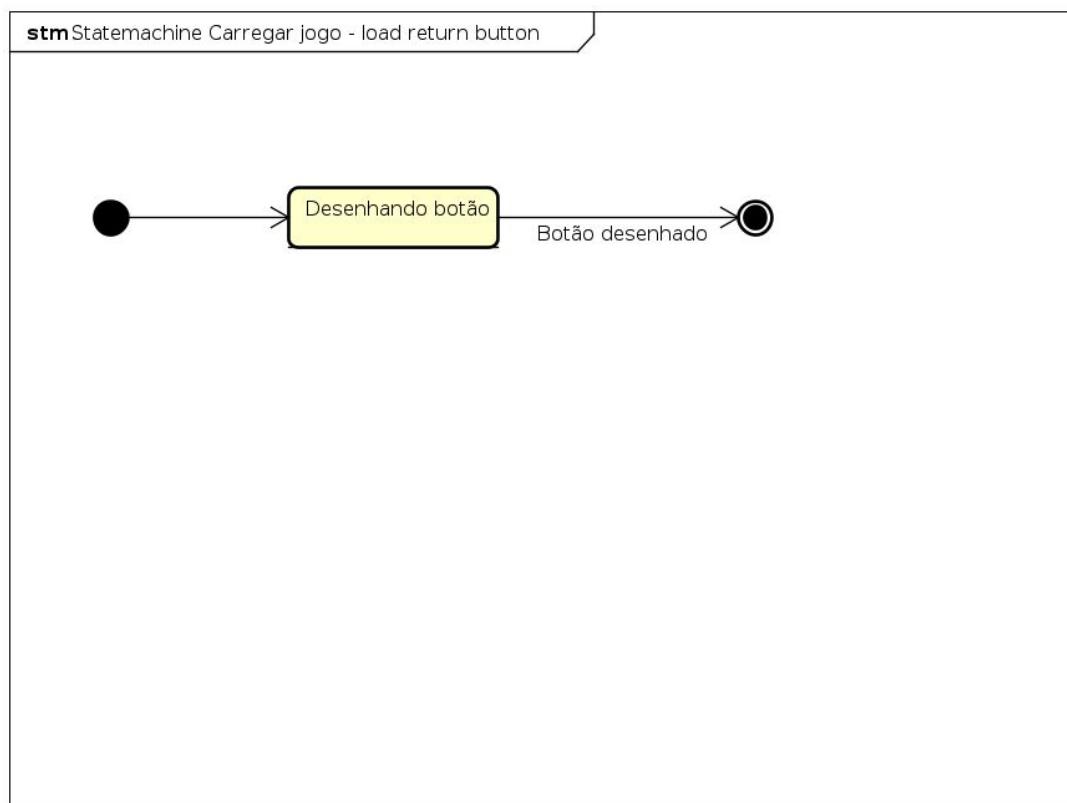


Figura 48. Diagrama Estados - Carregar jogo - load return button

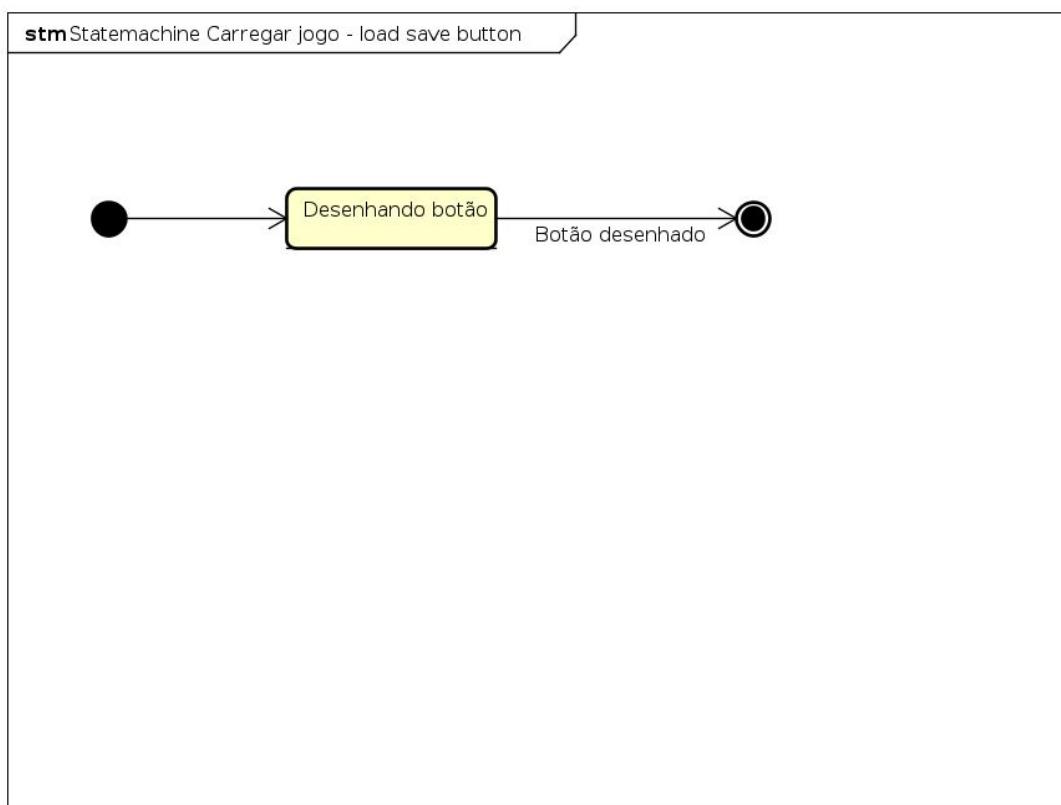


Figura 49. Diagrama Estados - Carregar jogo - load save button

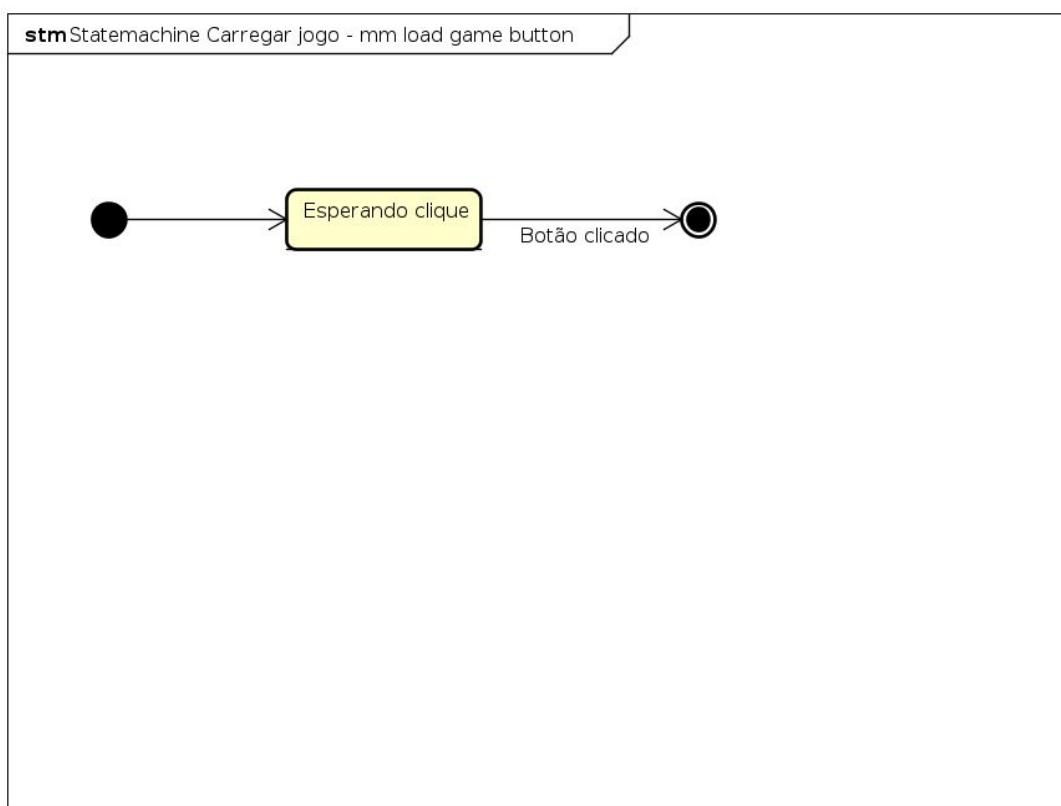


Figura 50. Diagrama Estados - Carregar jogo - mm load game button

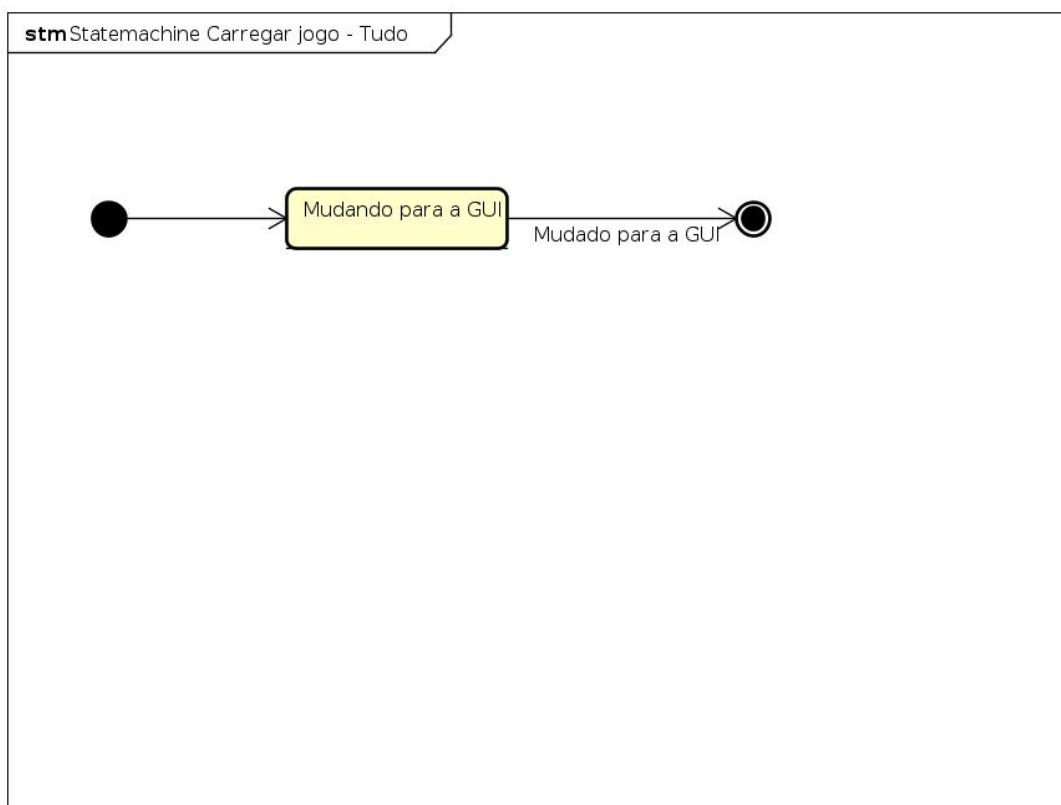


Figura 51. Diagrama Estados - Carregar jogo - Tudo

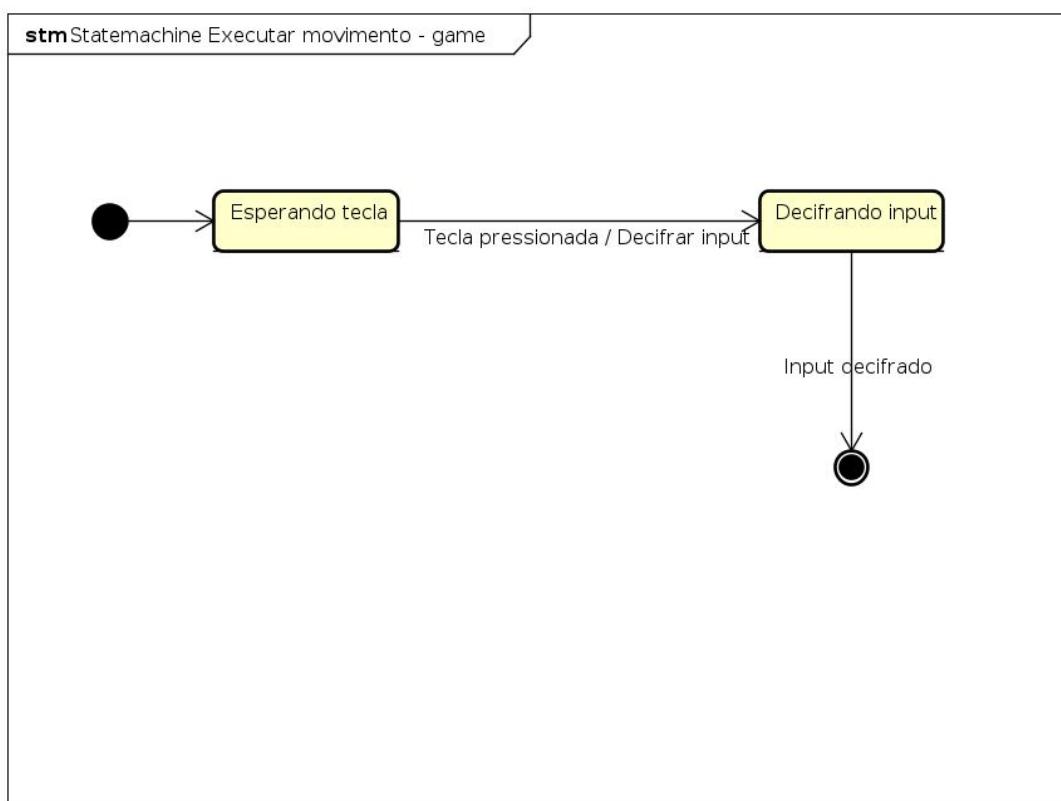


Figura 52. Diagrama Estados - Executar movimento - Game

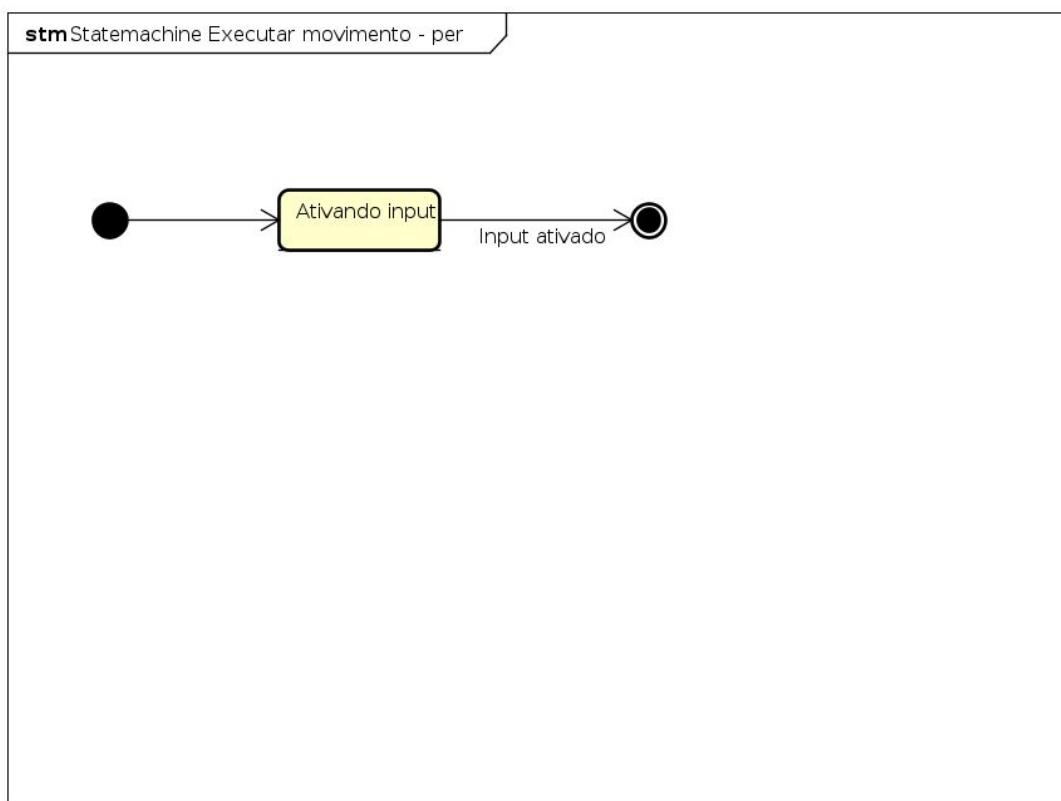


Figura 53. Diagrama Estados - Executar movimento - per

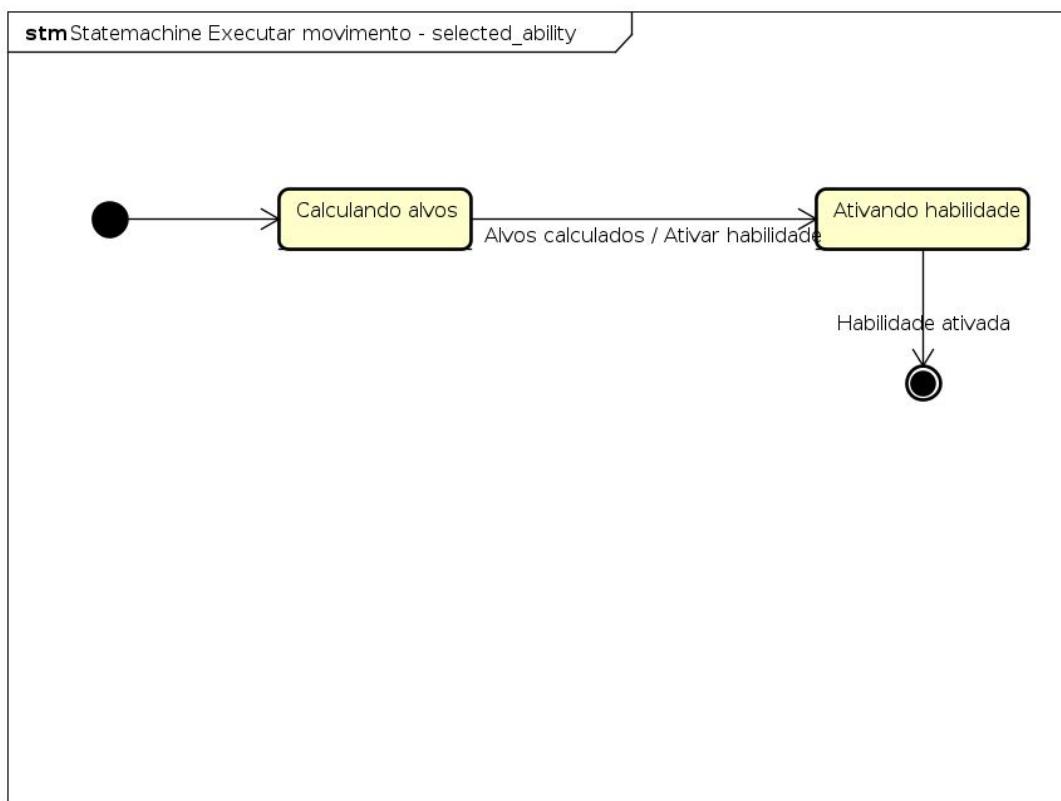


Figura 54. Diagrama Estados - Executar movimento - selected_ability

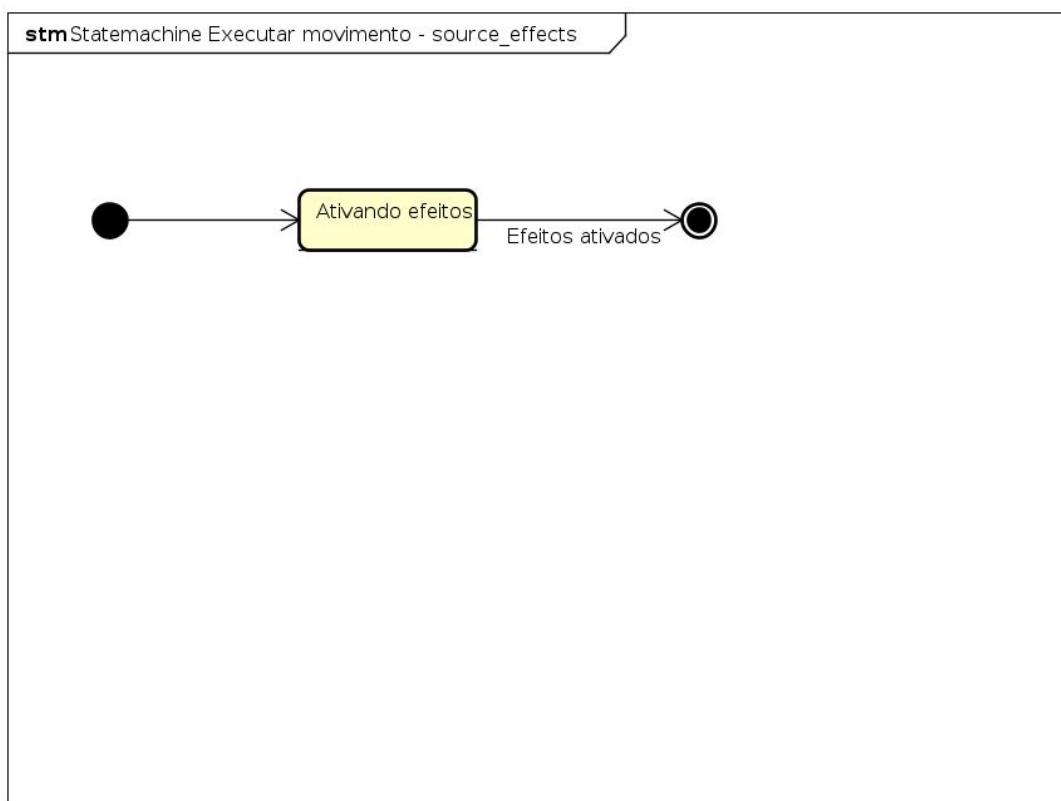


Figura 55. Diagrama Estados - Executar movimento - source_effects

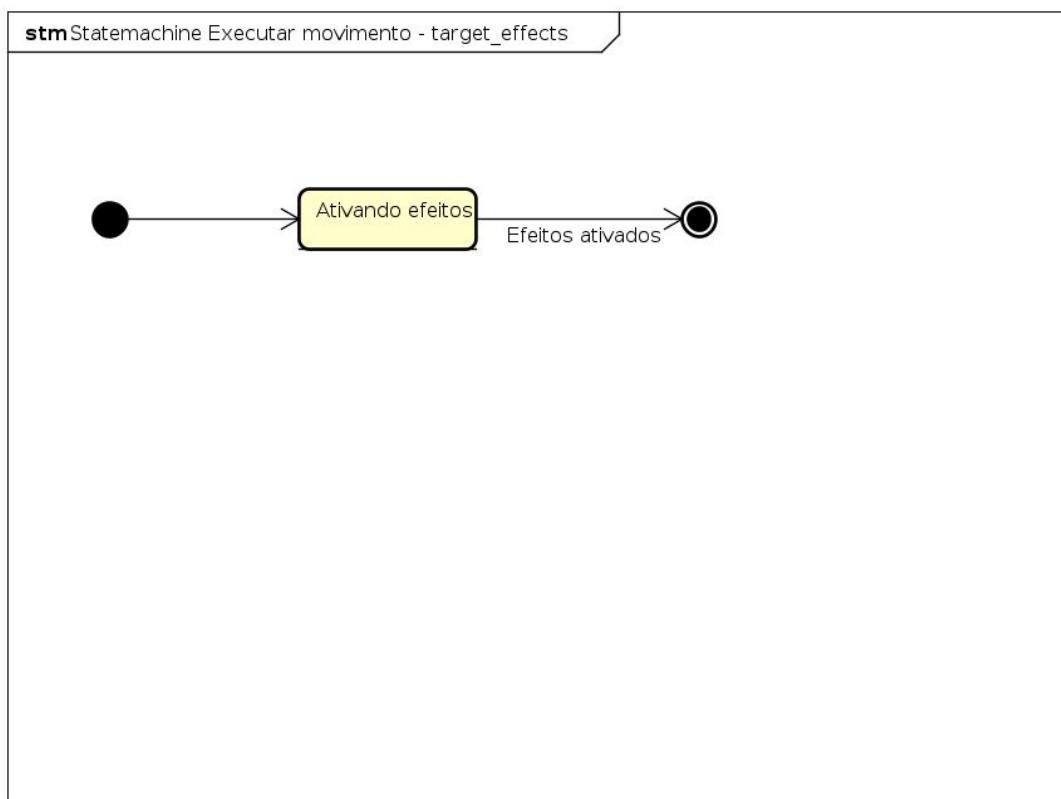


Figura 56. Diagrama Estados - Fechar programa - target_effects

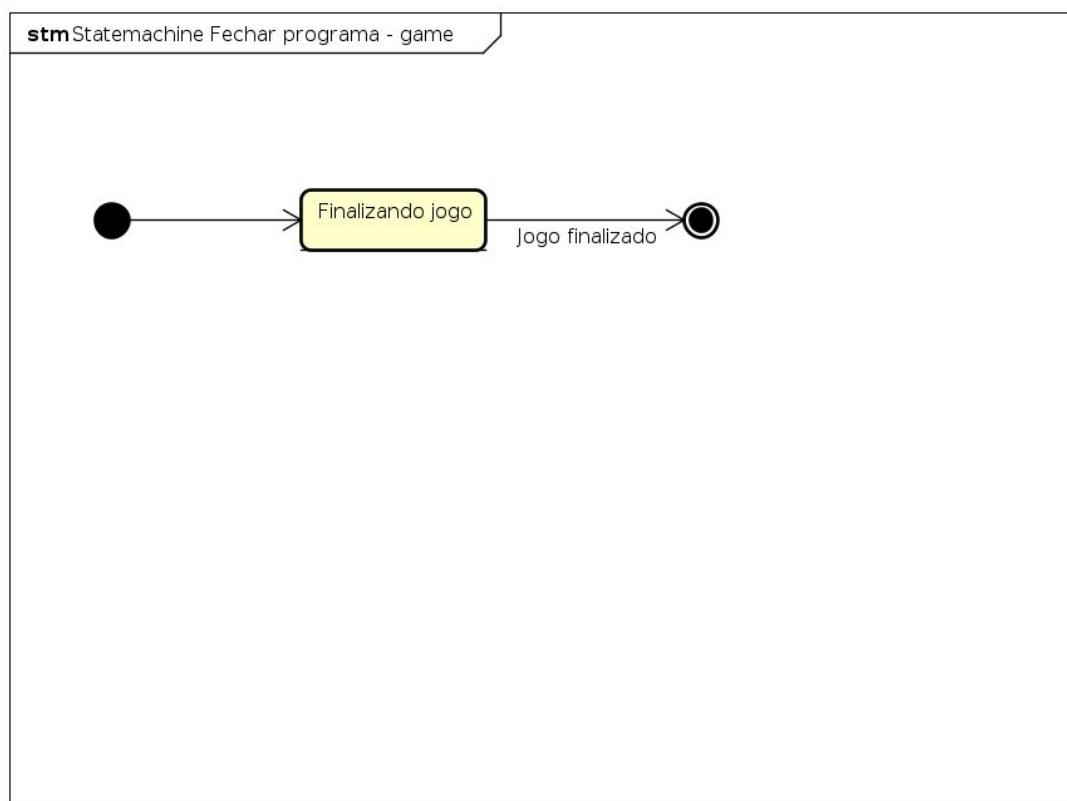


Figura 57. Diagrama Estados - Fechar programa - game

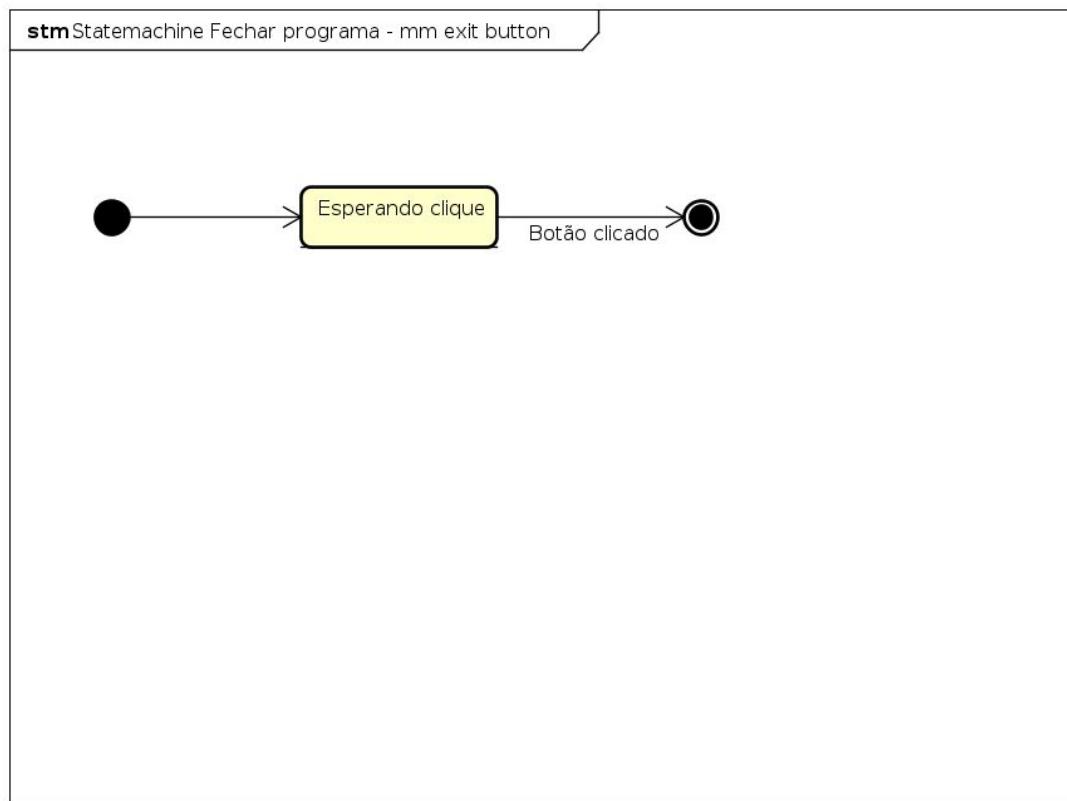


Figura 58. Diagrama Estados - Fechar programa - mm exit button

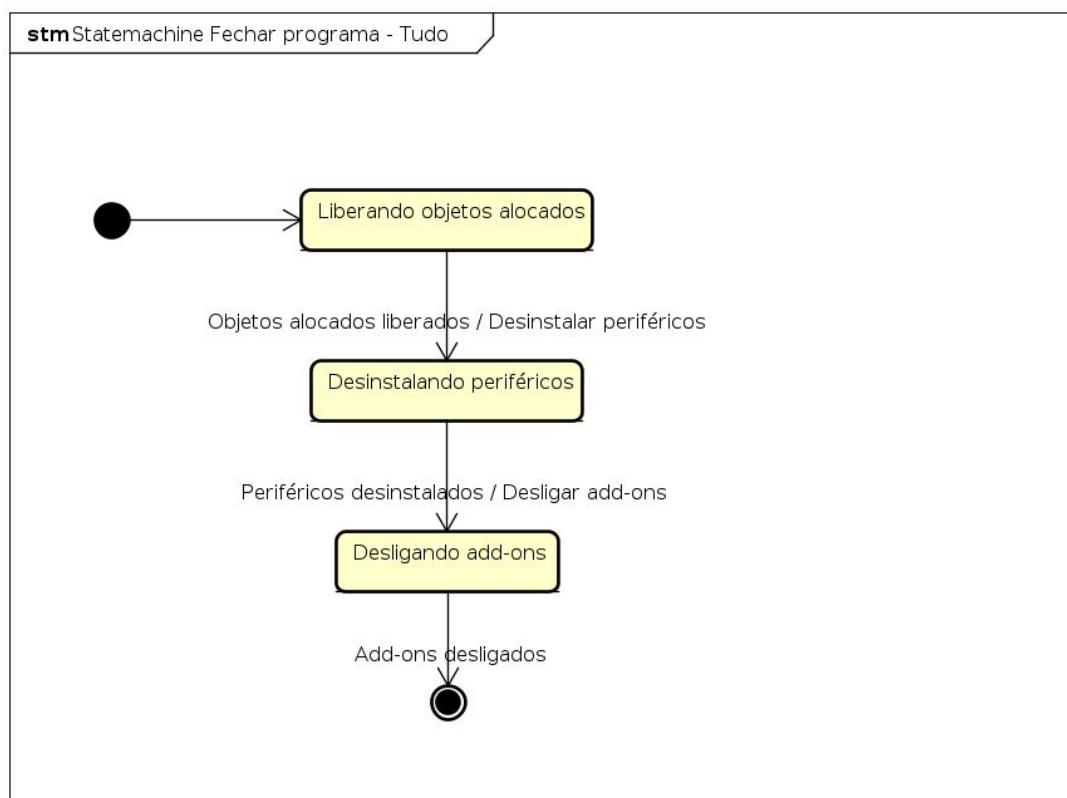


Figura 59. Diagrama Estados - Fechar programa - Tudo

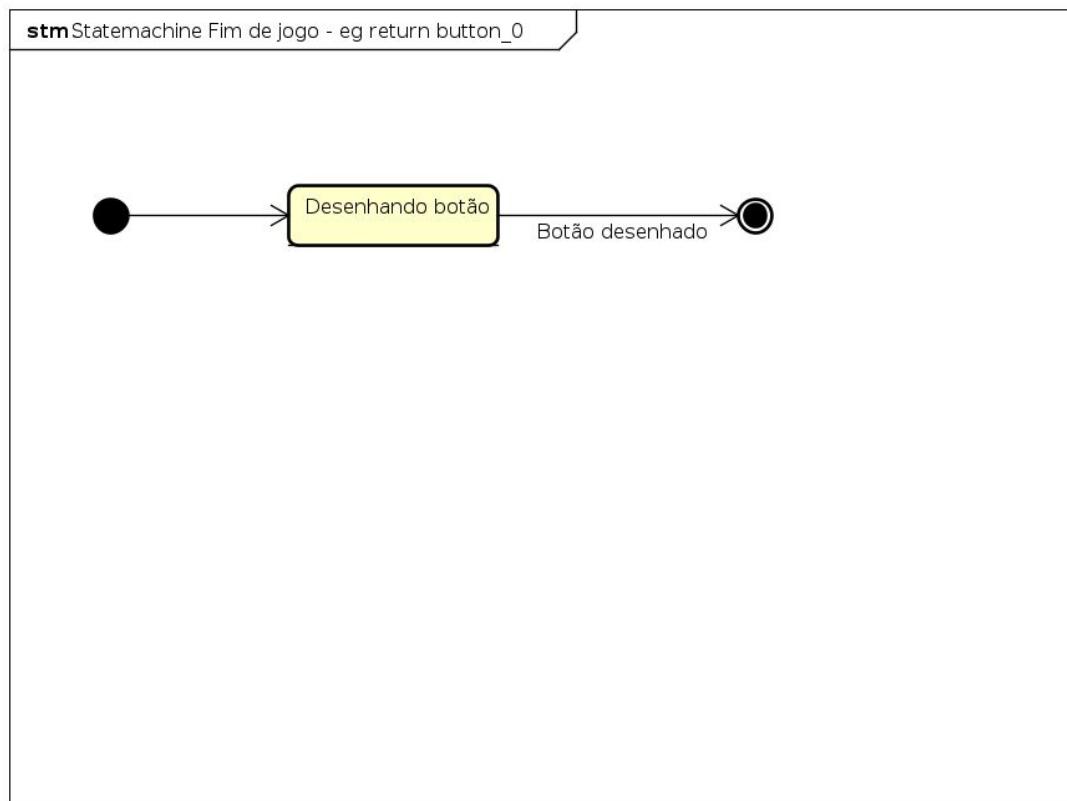


Figura 60. Diagrama Estados - Fim de jogo - eg return button 0

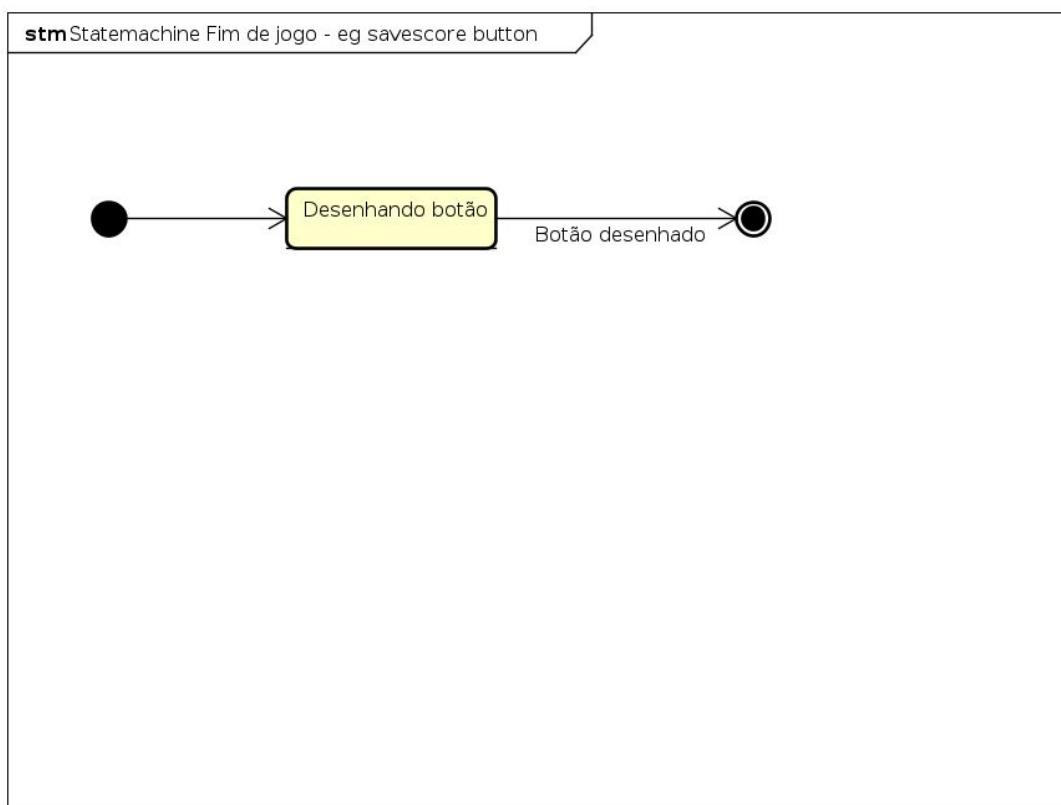


Figura 61. Diagrama Estados - Fim de jogo - eg savescore button

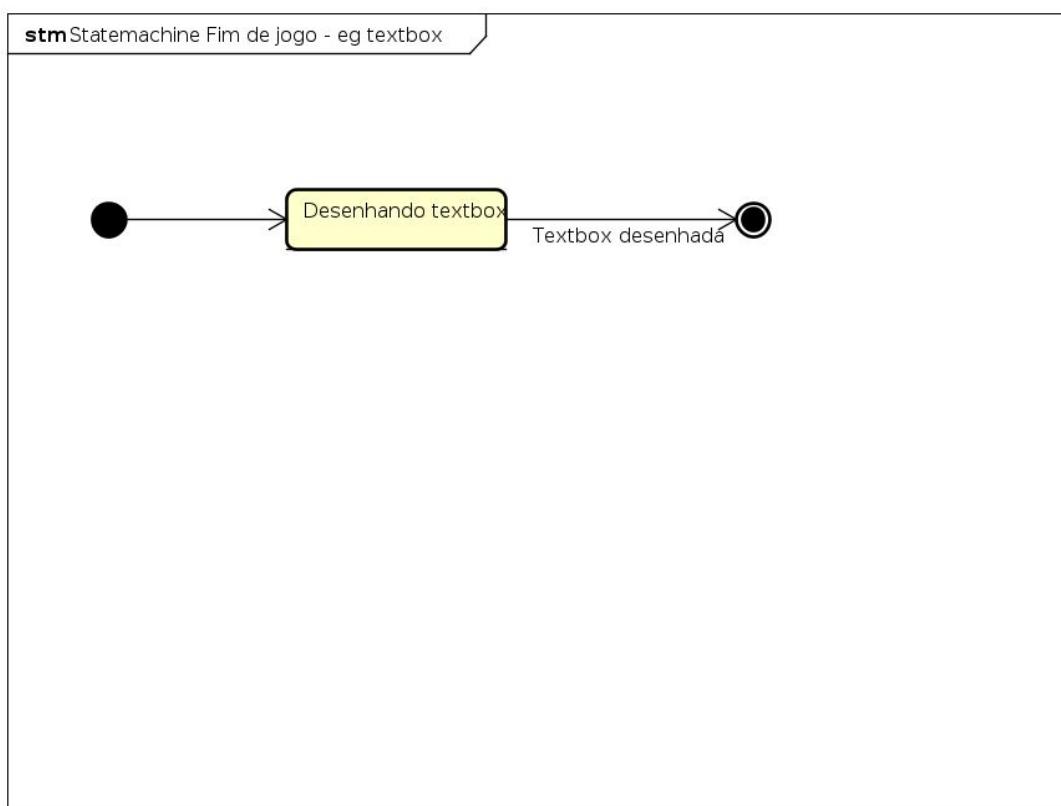


Figura 62. Diagrama Estados - Fim de jogo - eg textbox

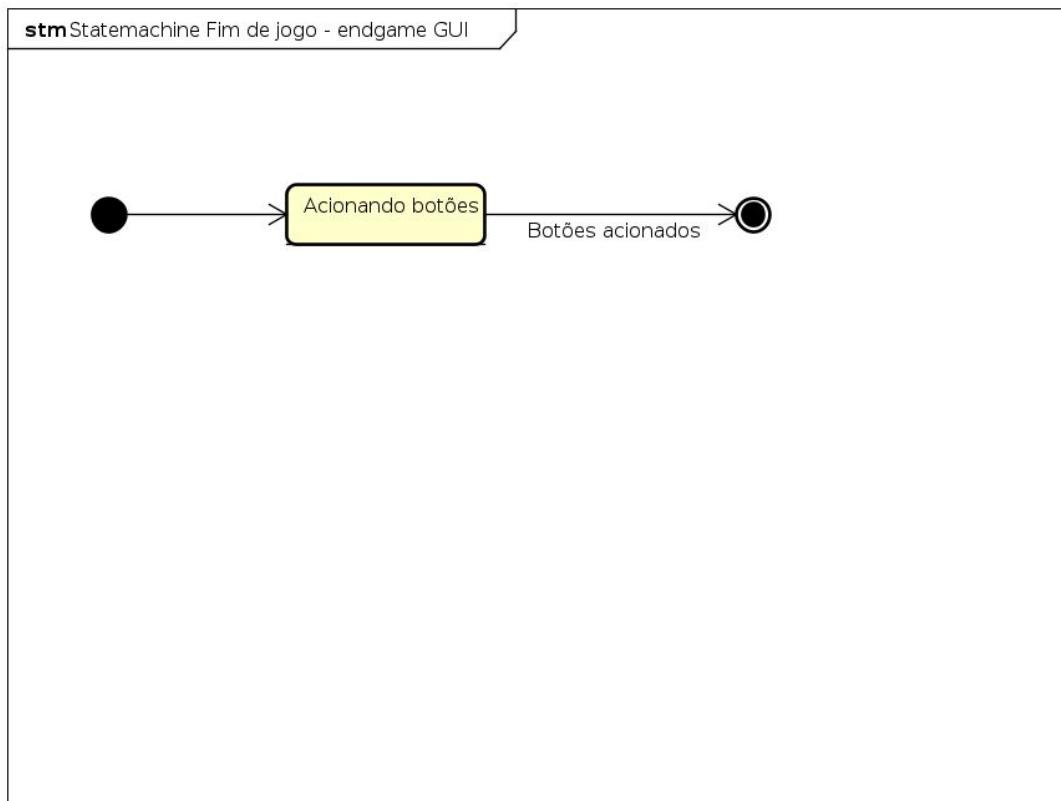


Figura 63. Diagrama Estados - Fim de jogo - endgame GUI

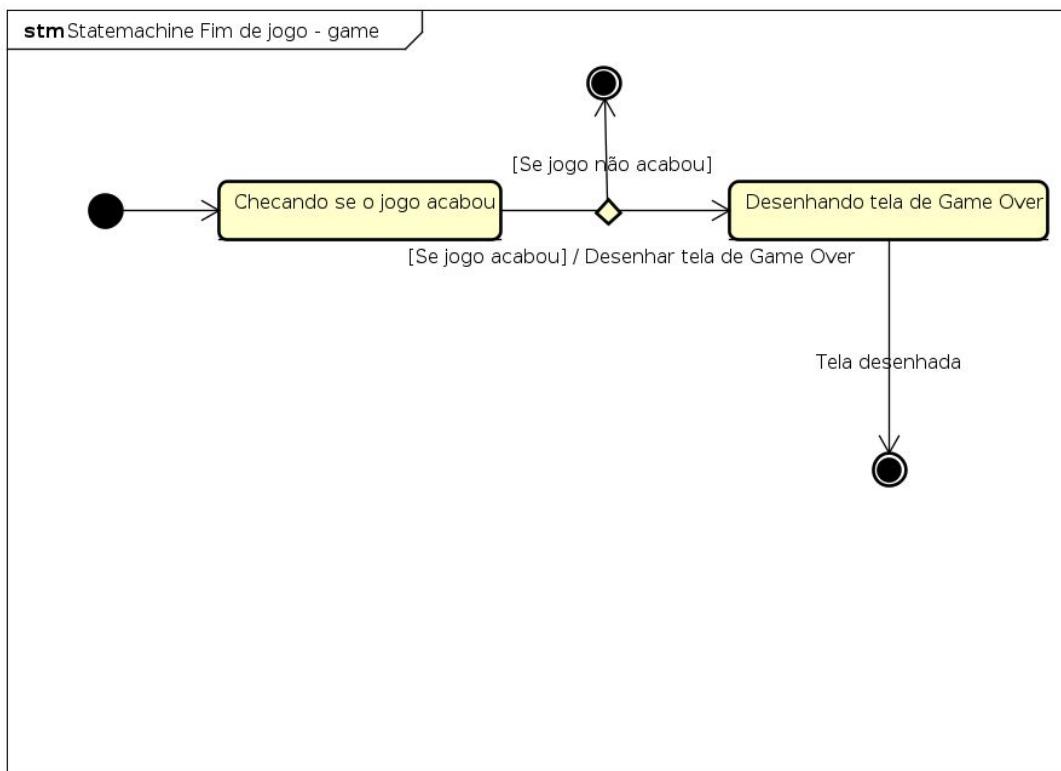


Figura 64. Diagrama Estados - Fim de jogo - Game

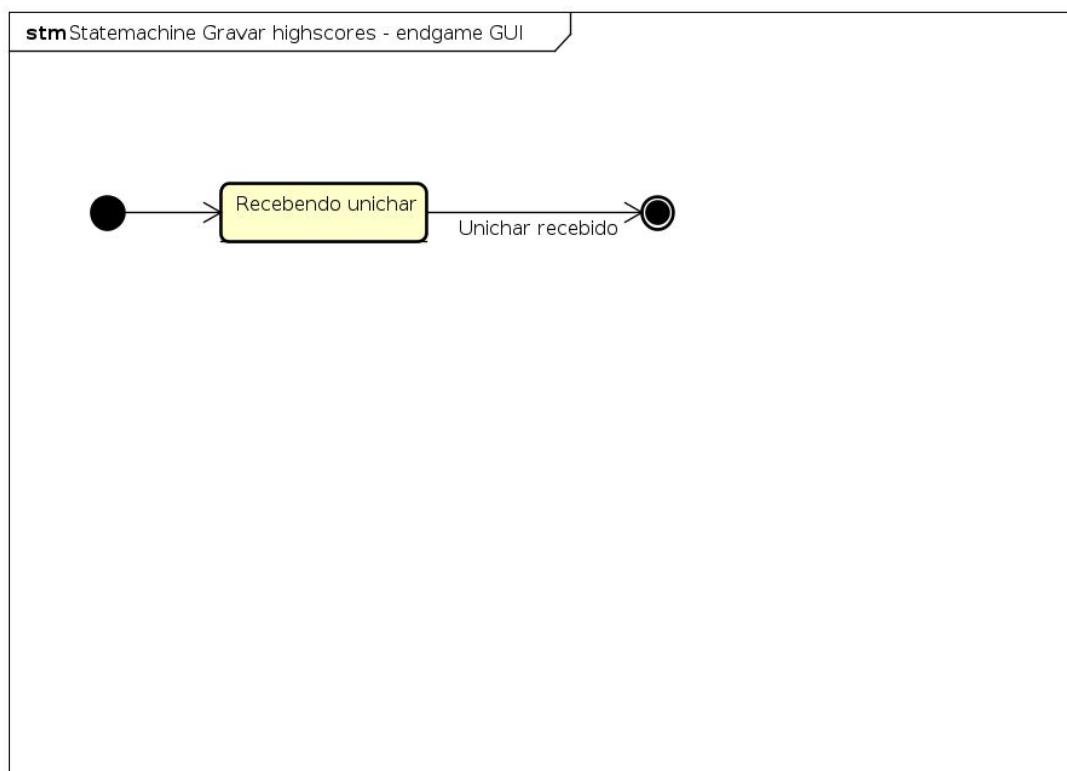


Figura 65. Diagrama Estados - Gravar highscores - endgame GUI

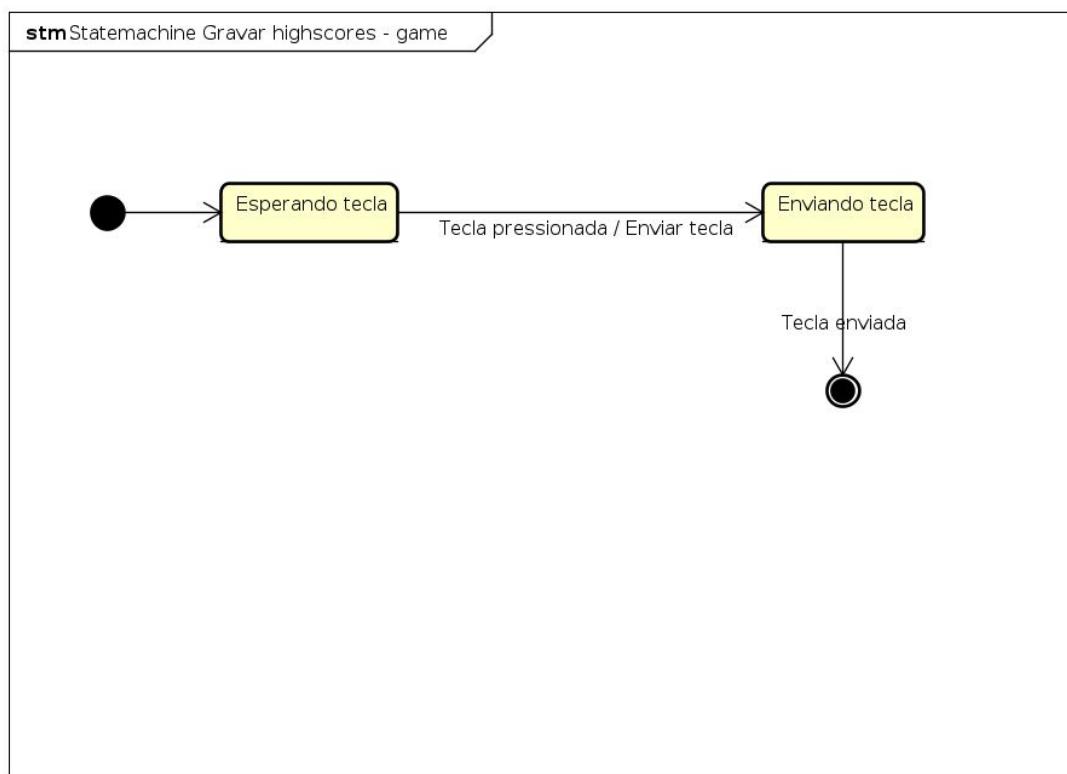


Figura 66. Diagrama Estados - Gravar highscores - game

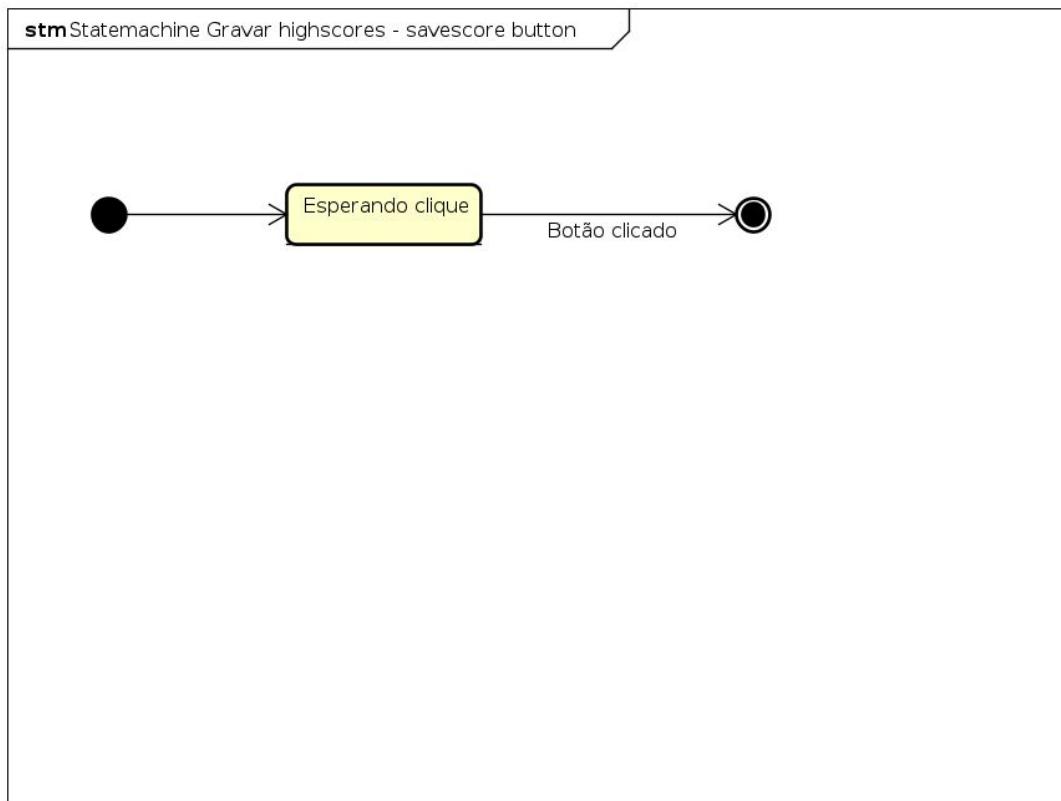


Figura 67. Diagrama Estados - Gravar highscores - savescores button

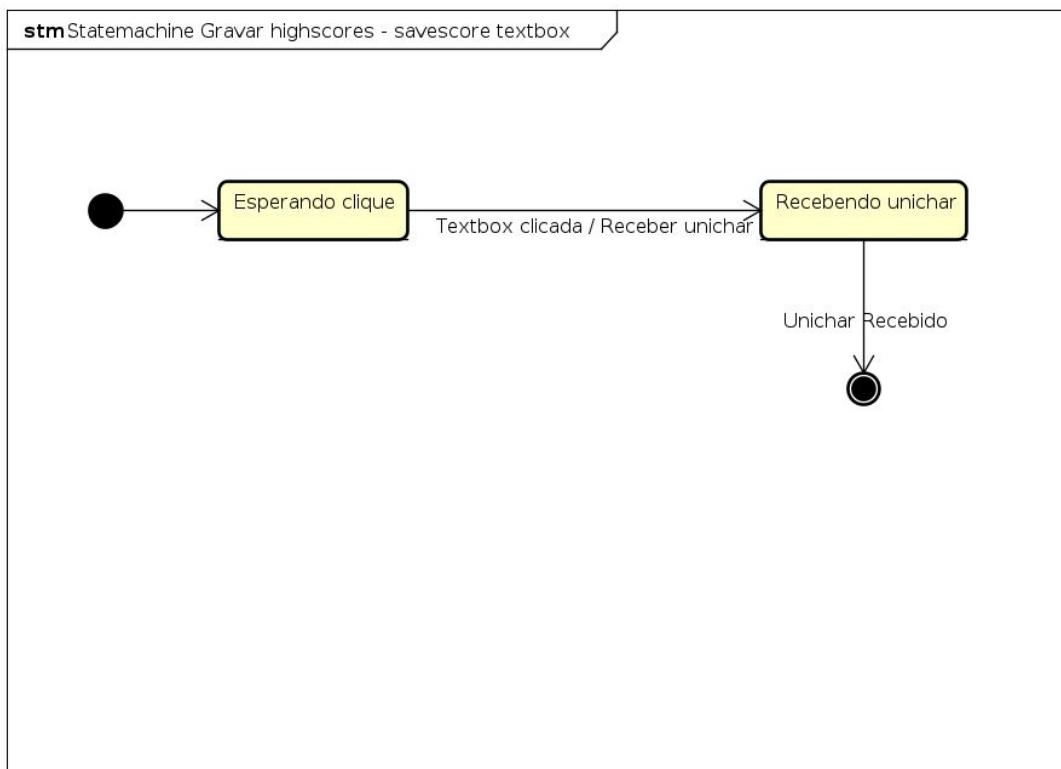


Figura 68. Diagrama Estados - Gravar highscores - highscores textbox

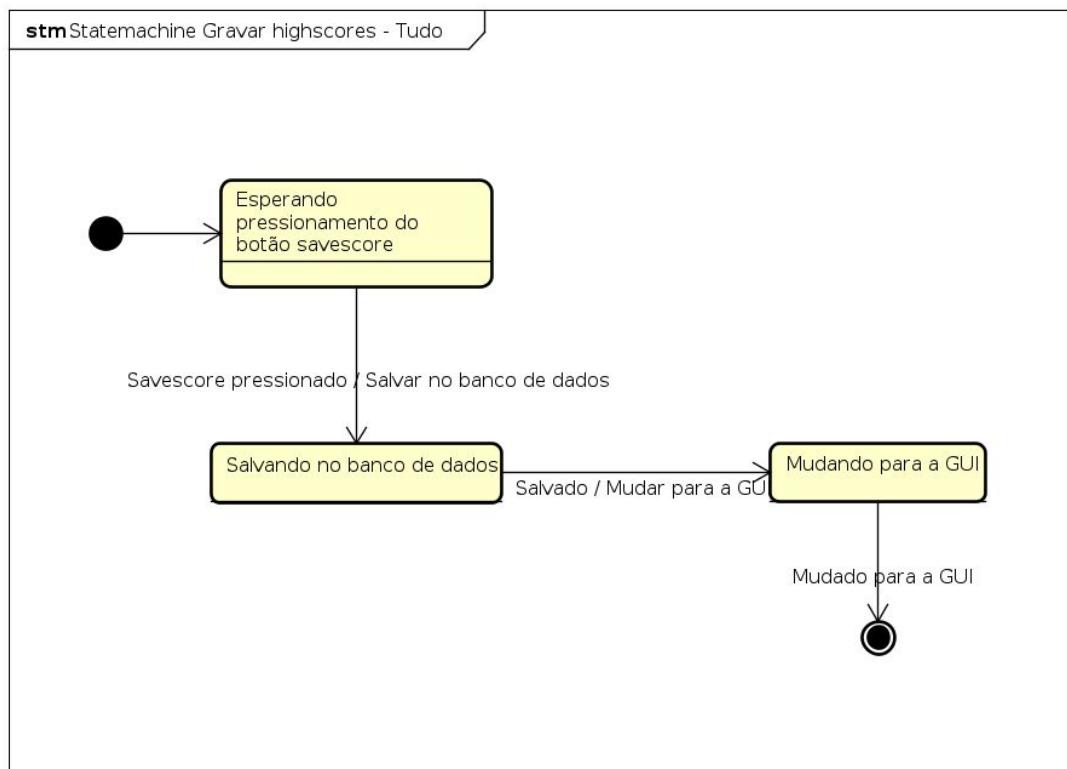


Figura 69. Diagrama Estados - Gravar highscores - Tudo

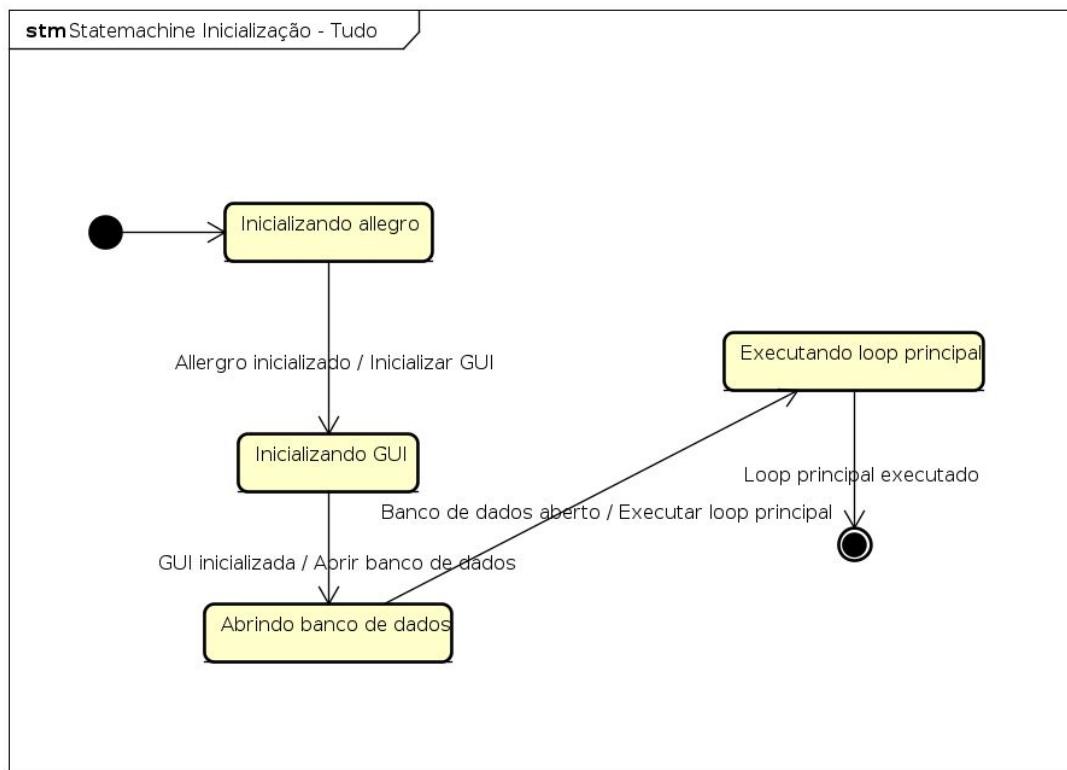


Figura 70. Diagrama Estados - Inicialização - Tudo

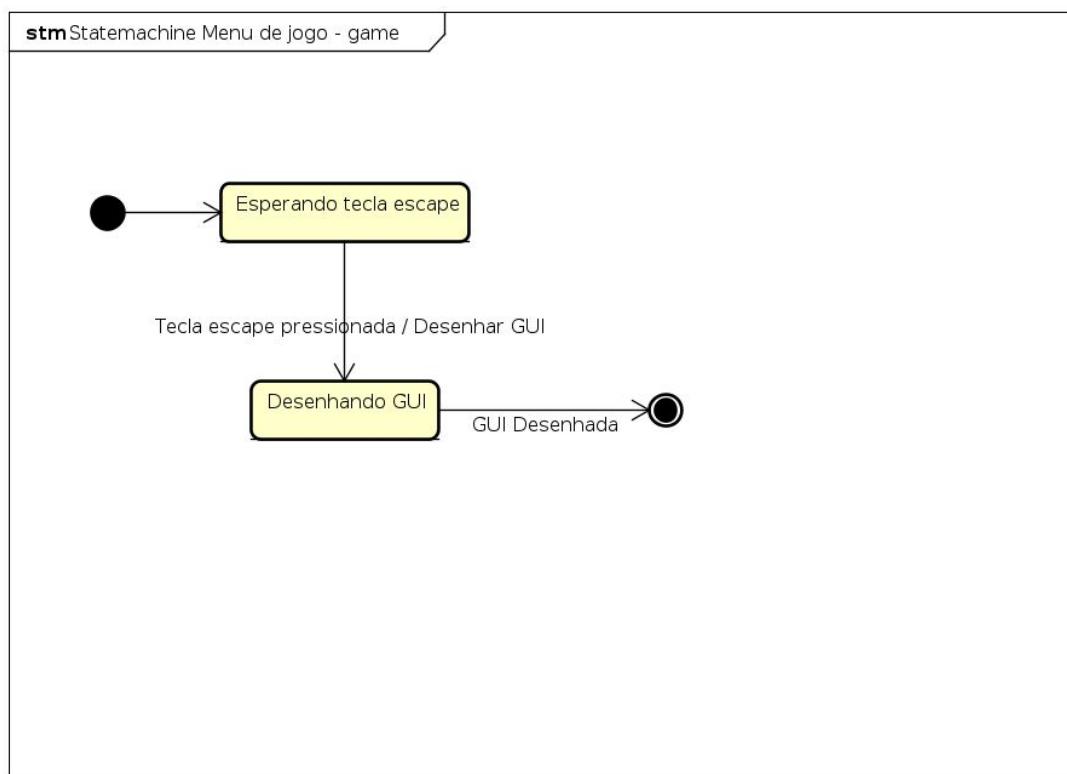


Figura 71. Diagrama Estados - Menu de jogo - game

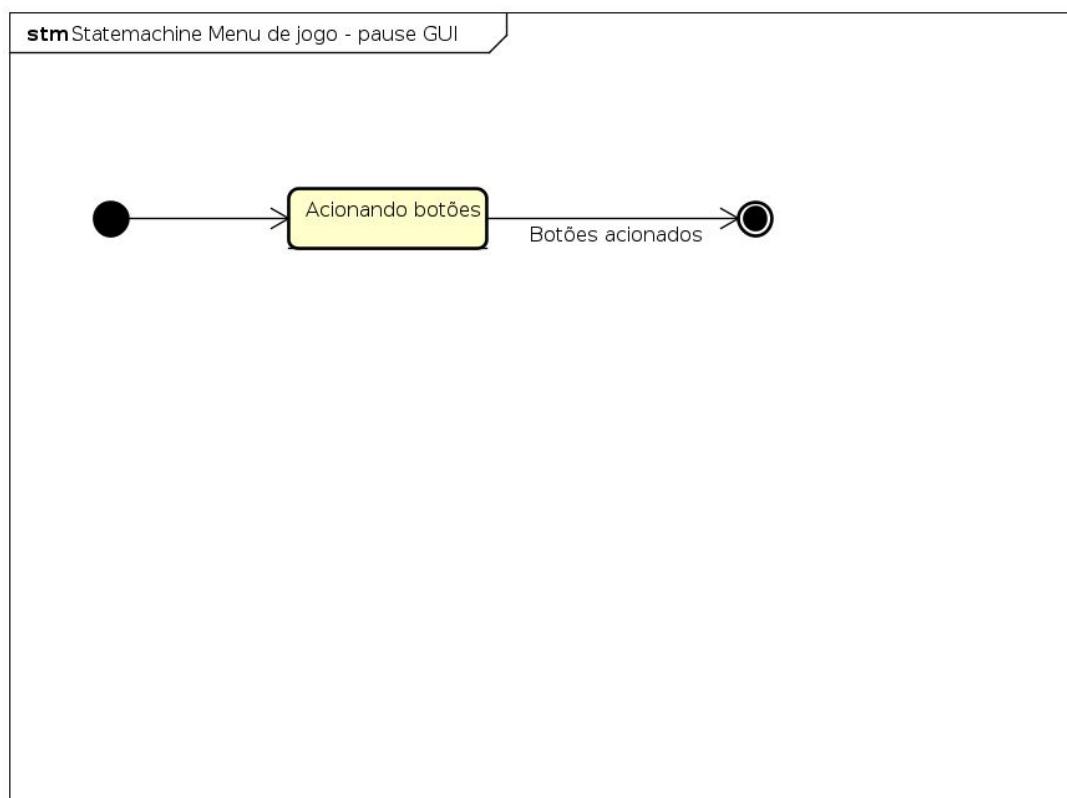


Figura 72. Diagrama Estados - Menu de jogo - pause GUI

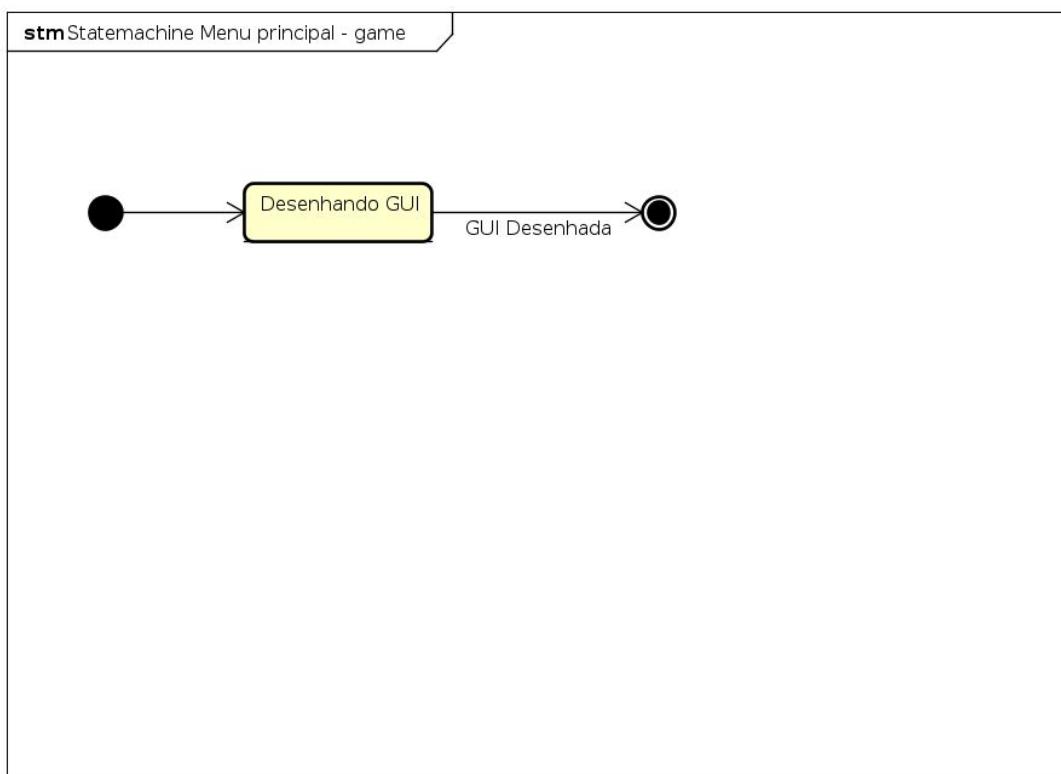


Figura 73. Diagrama Estados - Menu principal - game

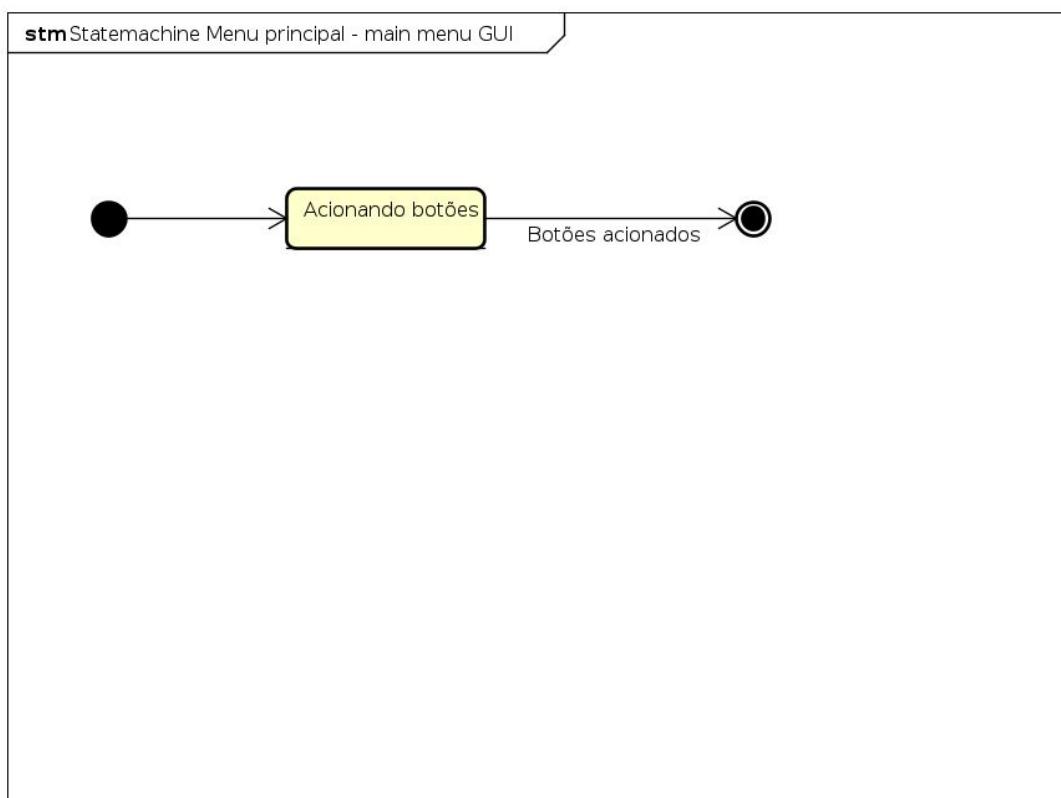


Figura 74. Diagrama Estados - Menu principal - Main menu GUI

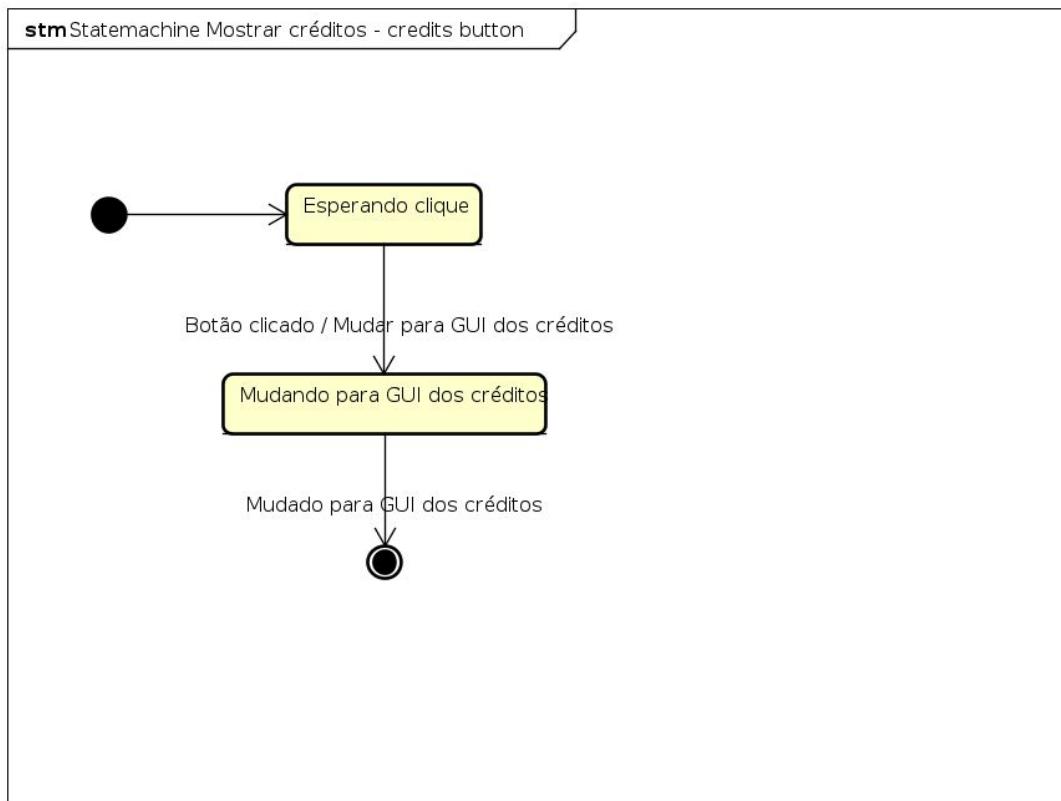


Figura 75. Diagrama Estados - Mostrar creditos - credits button

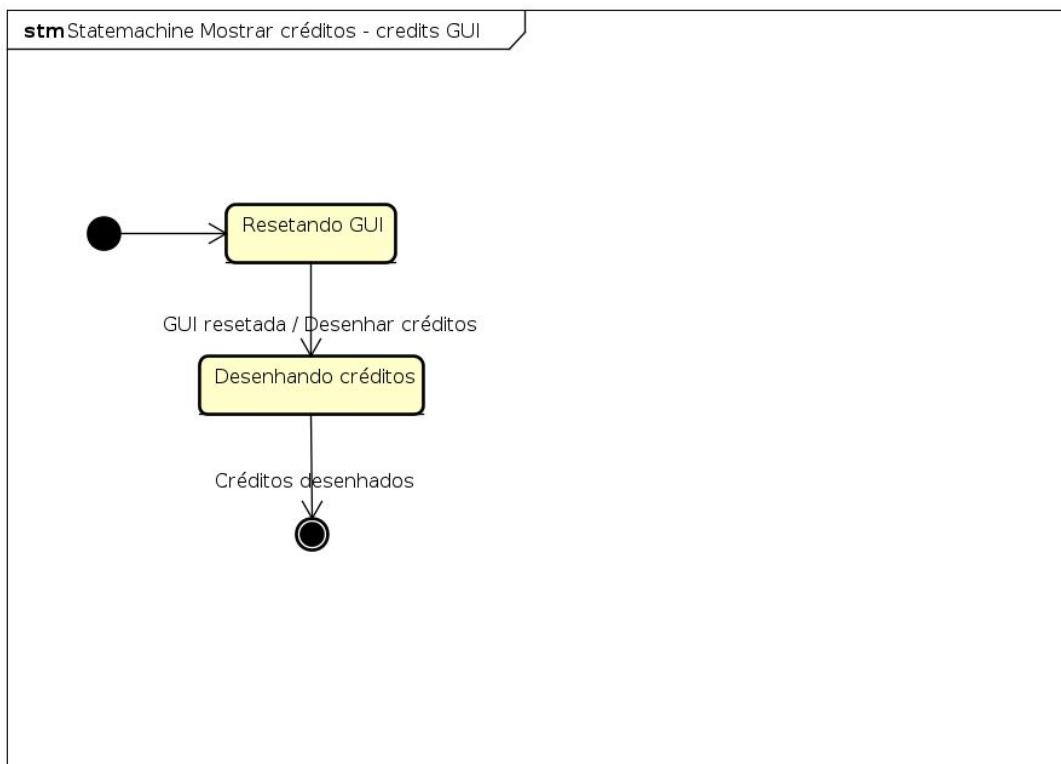


Figura 76. Diagrama Estados - Mostrar creditos - credits GUI

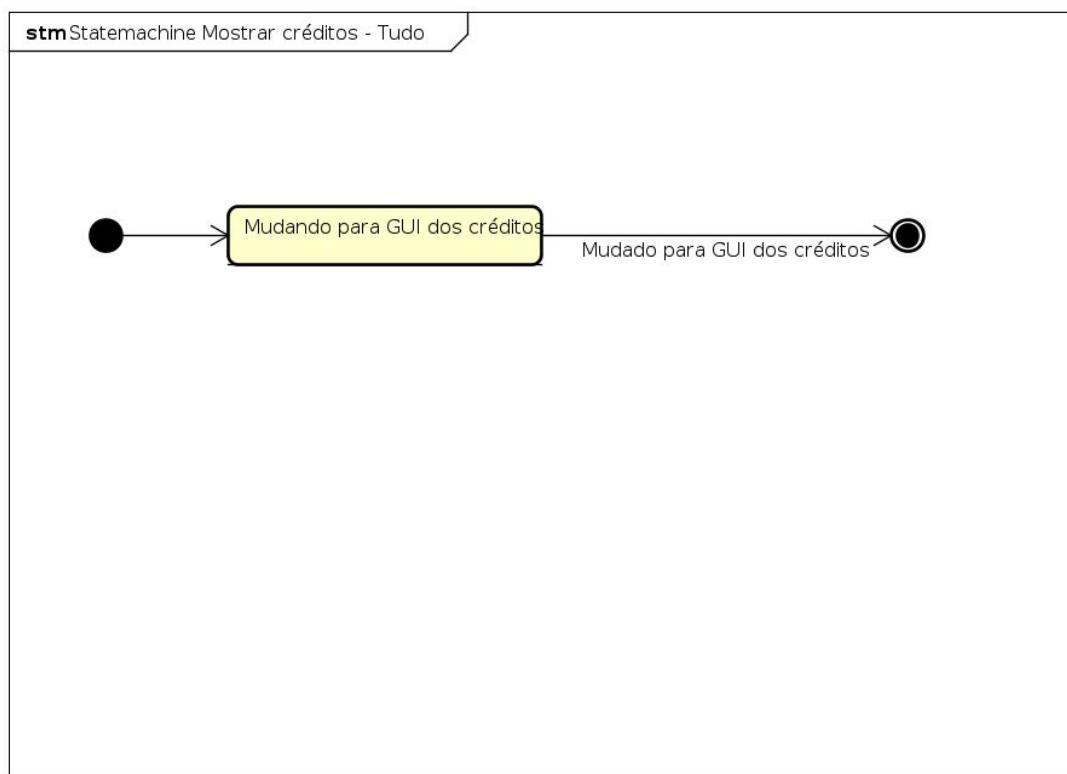


Figura 77. Diagrama Estados - Mostrar creditos - Tudo

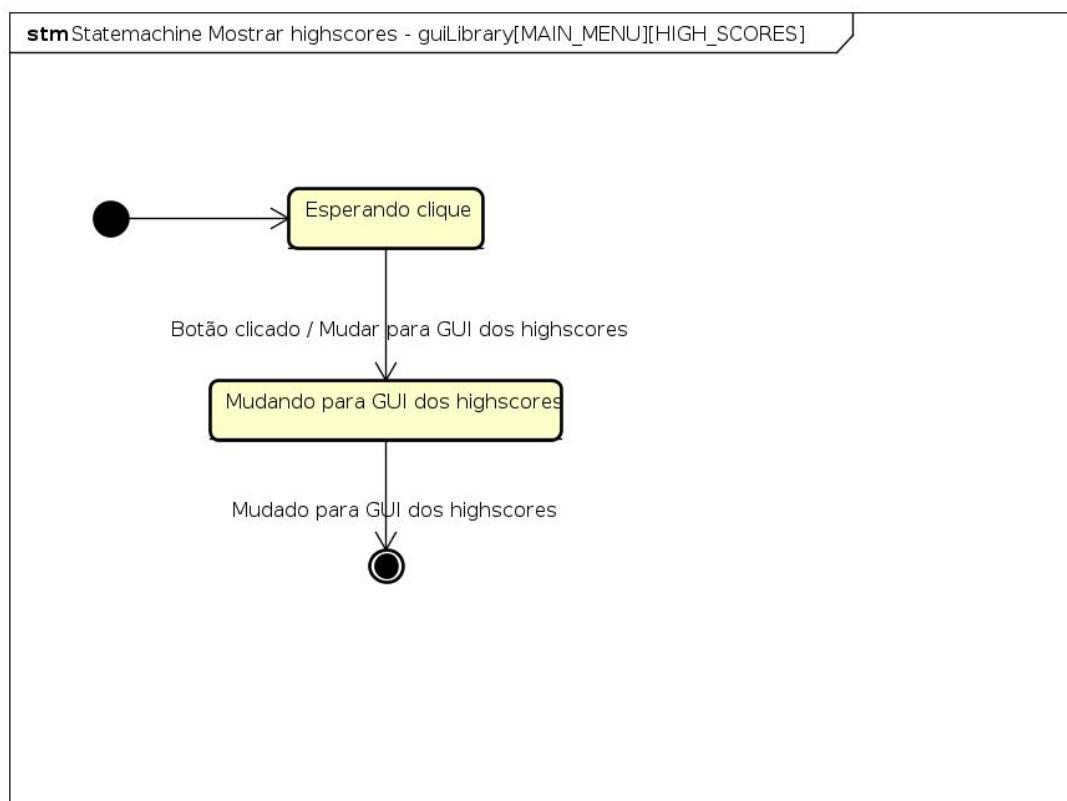


Figura 78. Diagrama Estados - Mostrar highscores -

guiLibrary[MAIN_MENU][HIGH_SCORES]

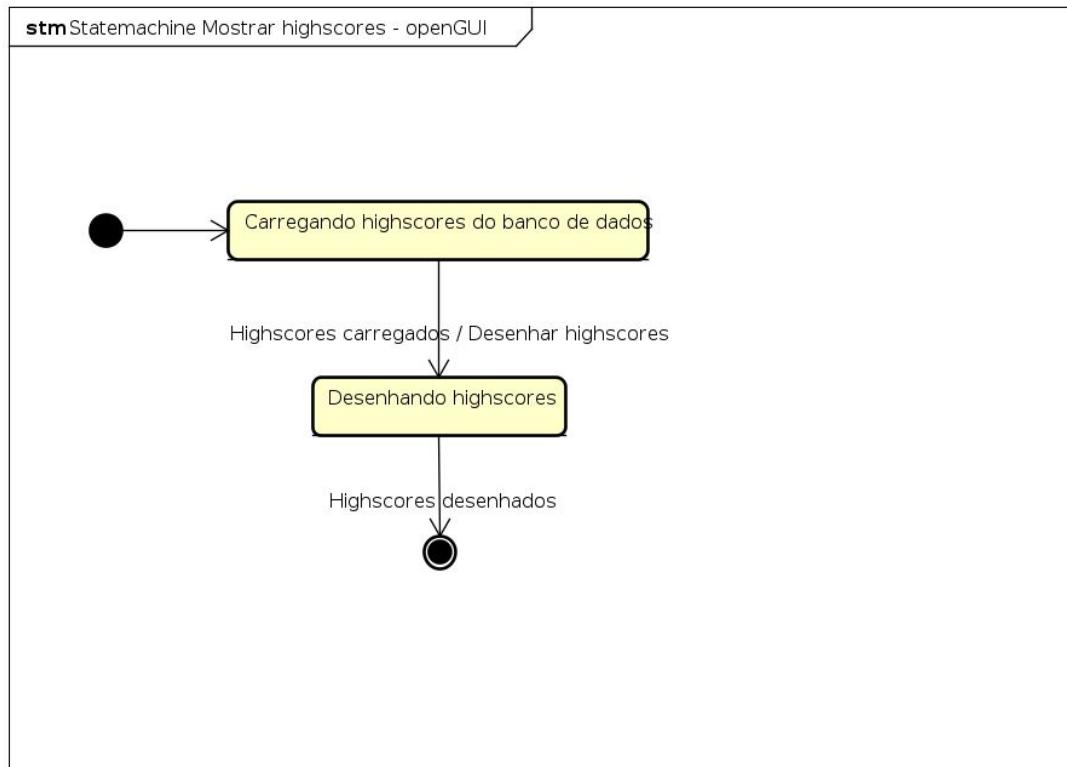


Figura 79. Diagrama Estados - Mostrar highscores - openGUI

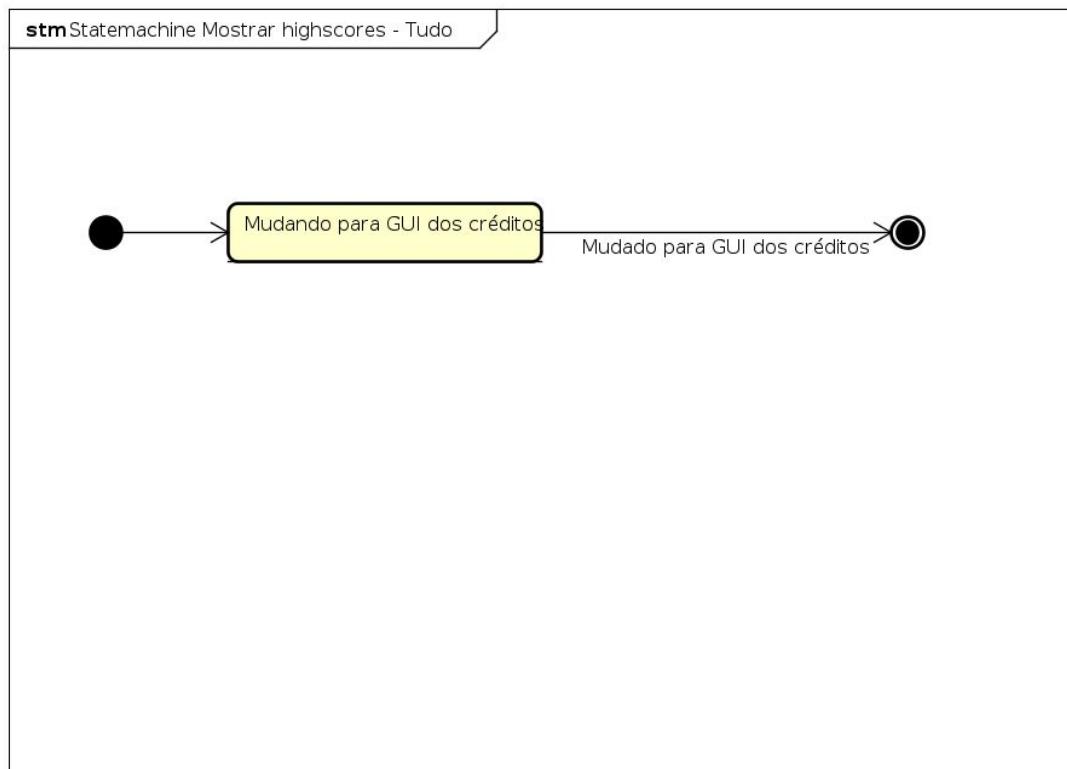


Figura 80. Diagrama Estados - Mostrar highscores - Tudo

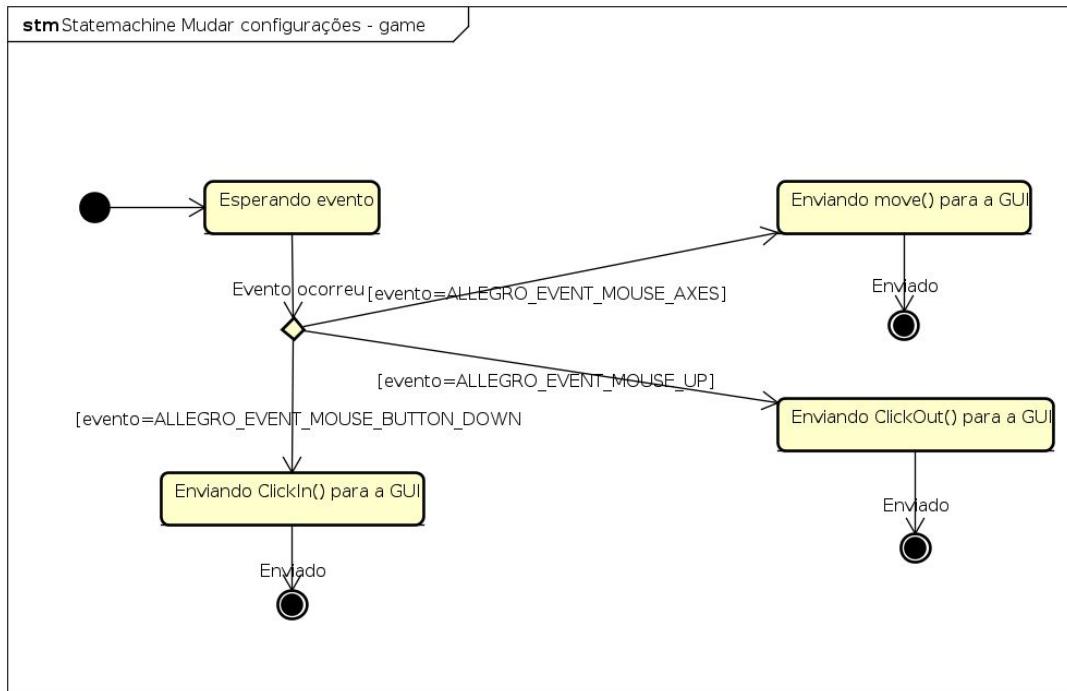


Figura 81. Diagrama Estados - Mudar configurações - game

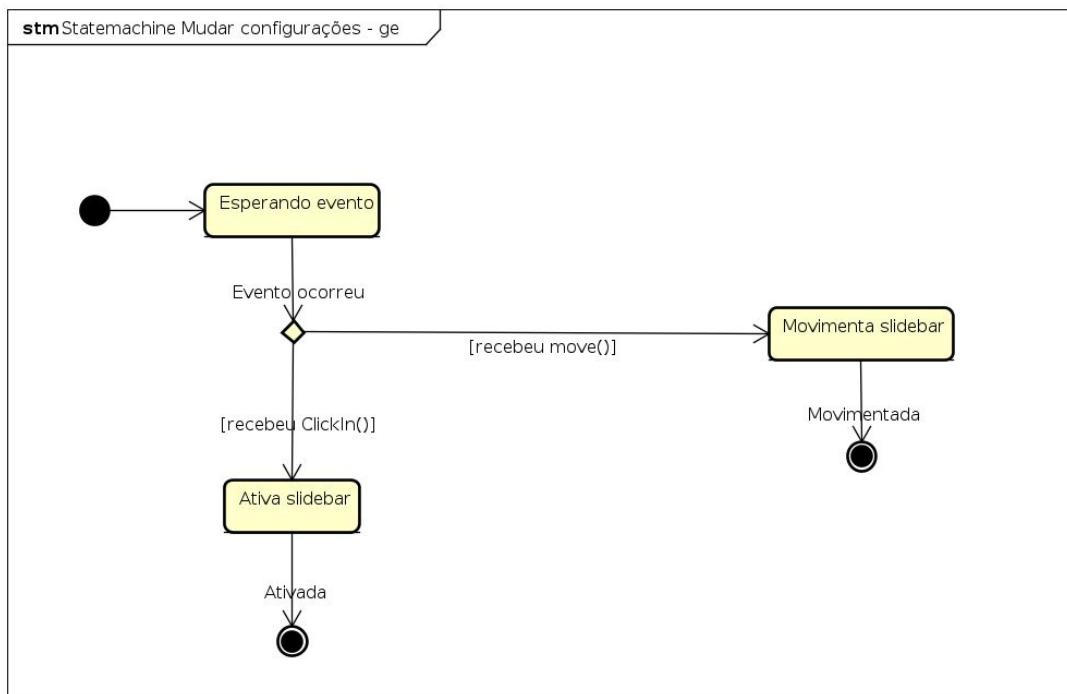


Figura 82. Diagrama Estados - Mudar configurações - ge

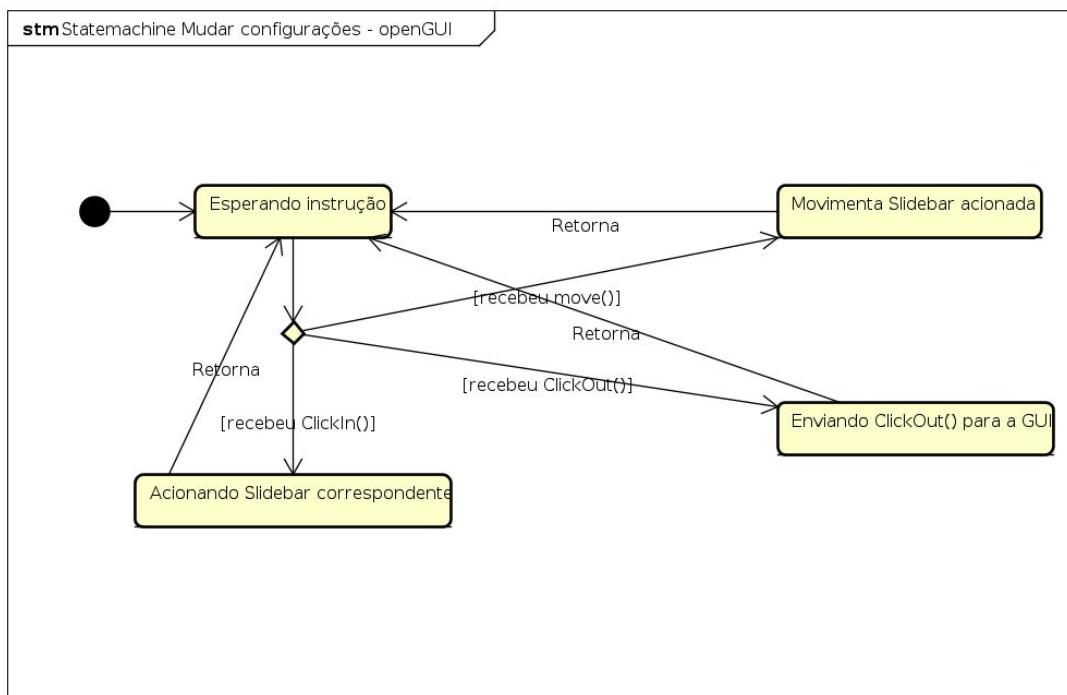


Figura 83. Diagrama Estados - Mudar configurações - openGUI

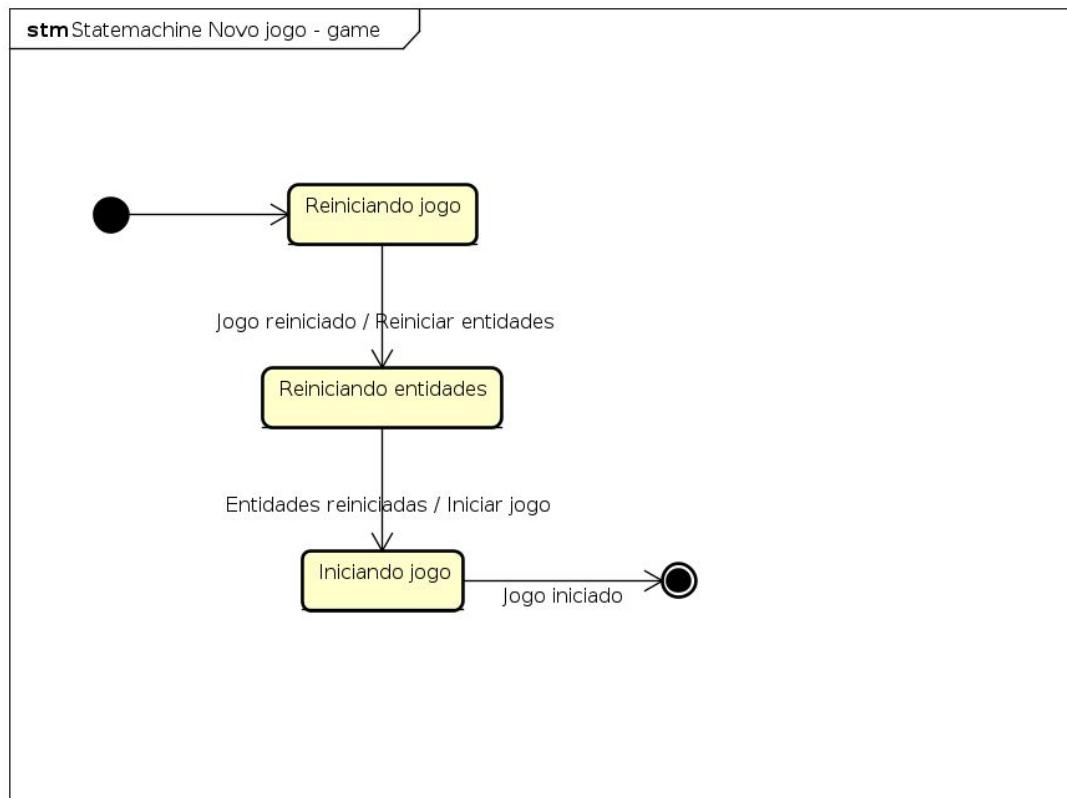


Figura 84. Diagrama Estados - Novo jogo - game

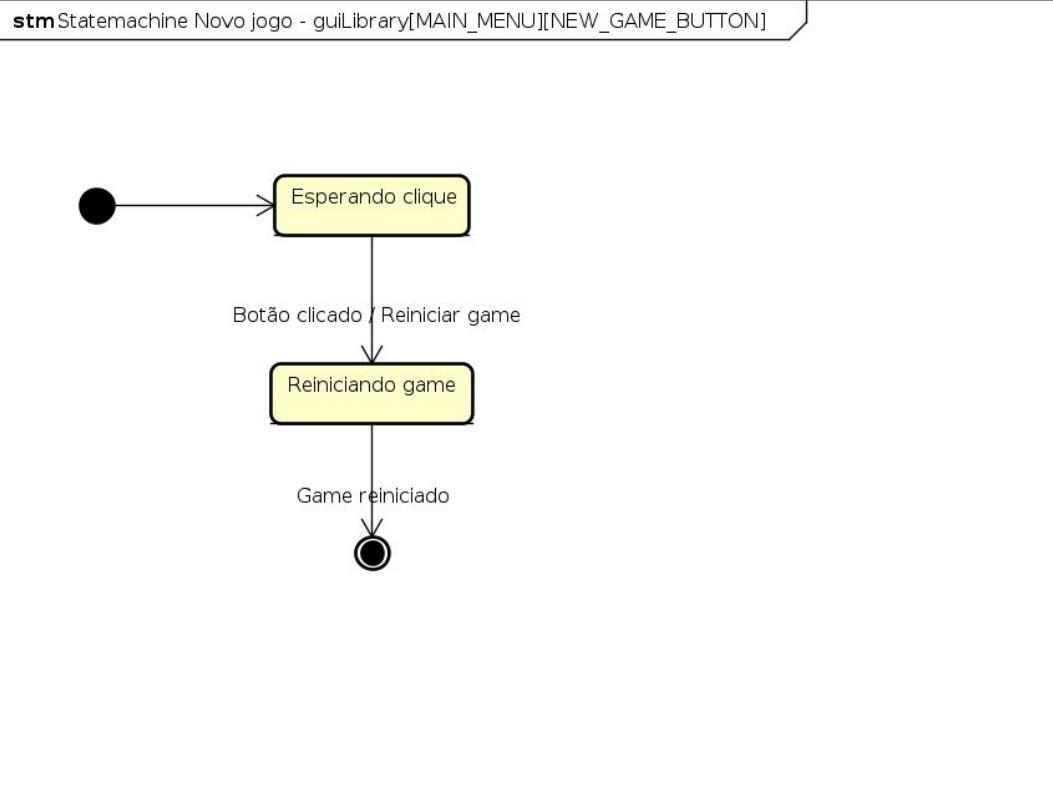


Figura 85. Diagrama Estados - Novo jogo -
guiLibrary[MAIN_MENU][NEW_GAME_BUTTON]

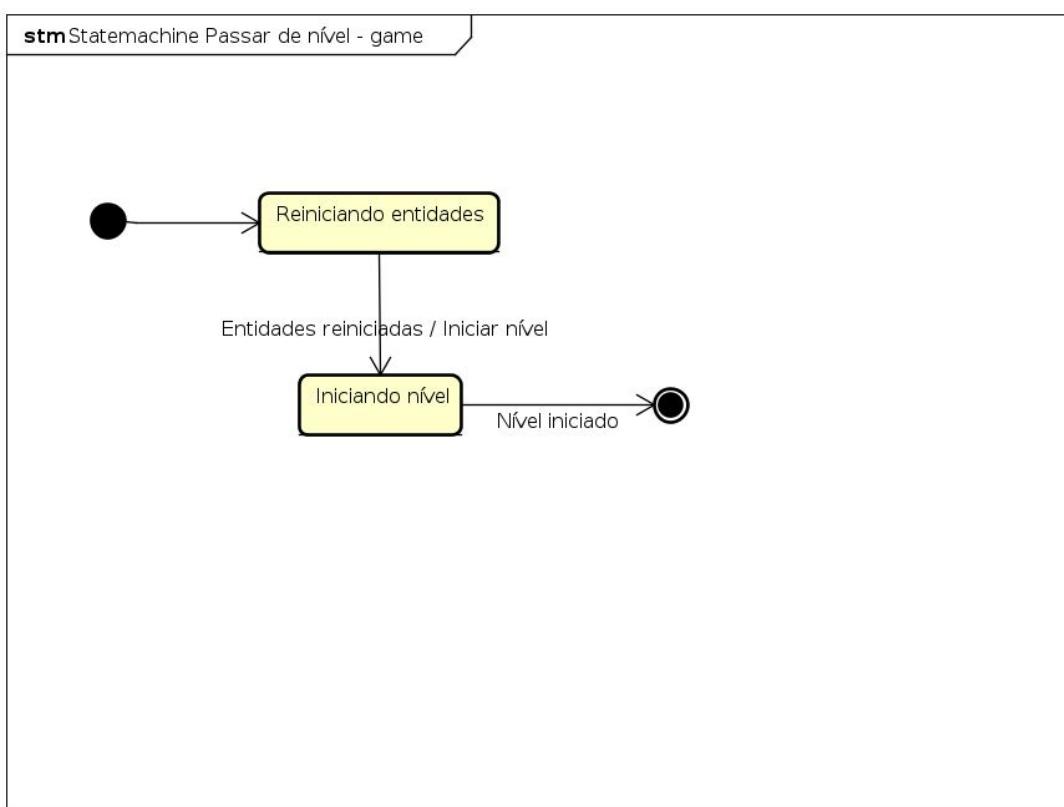


Figura 86. Diagrama Estados - Passar de nível - game

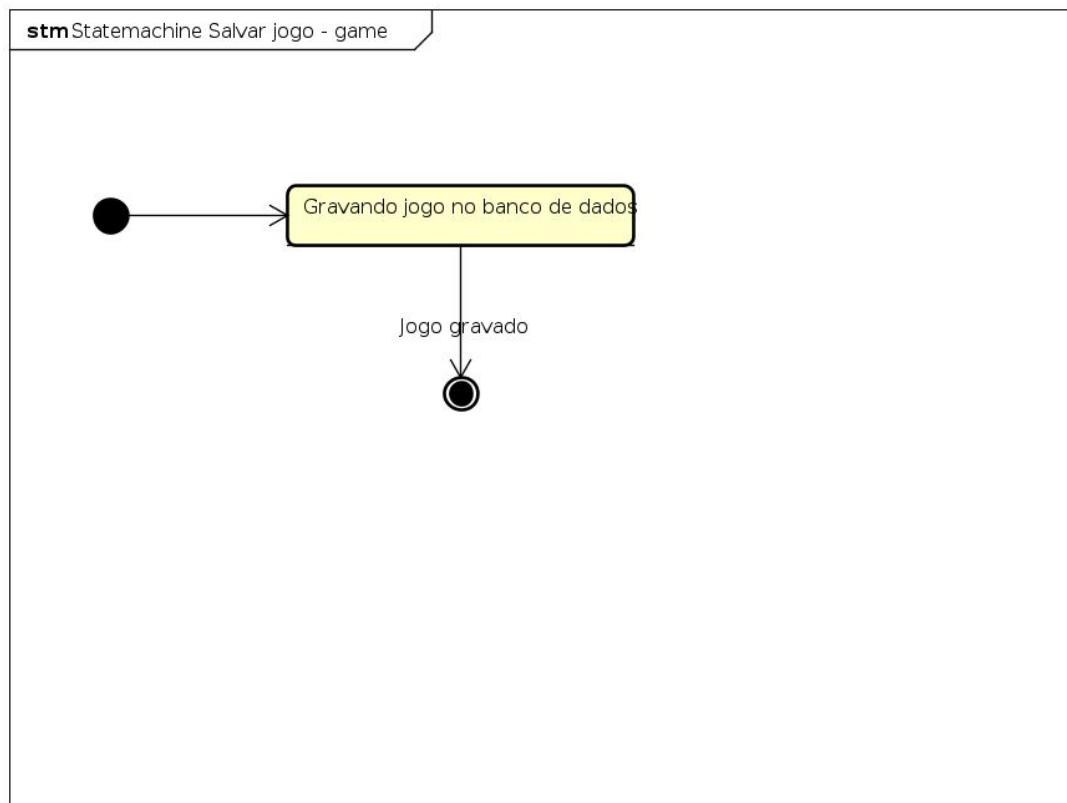


Figura 87. Diagrama Estados - Salvar jogo - Game

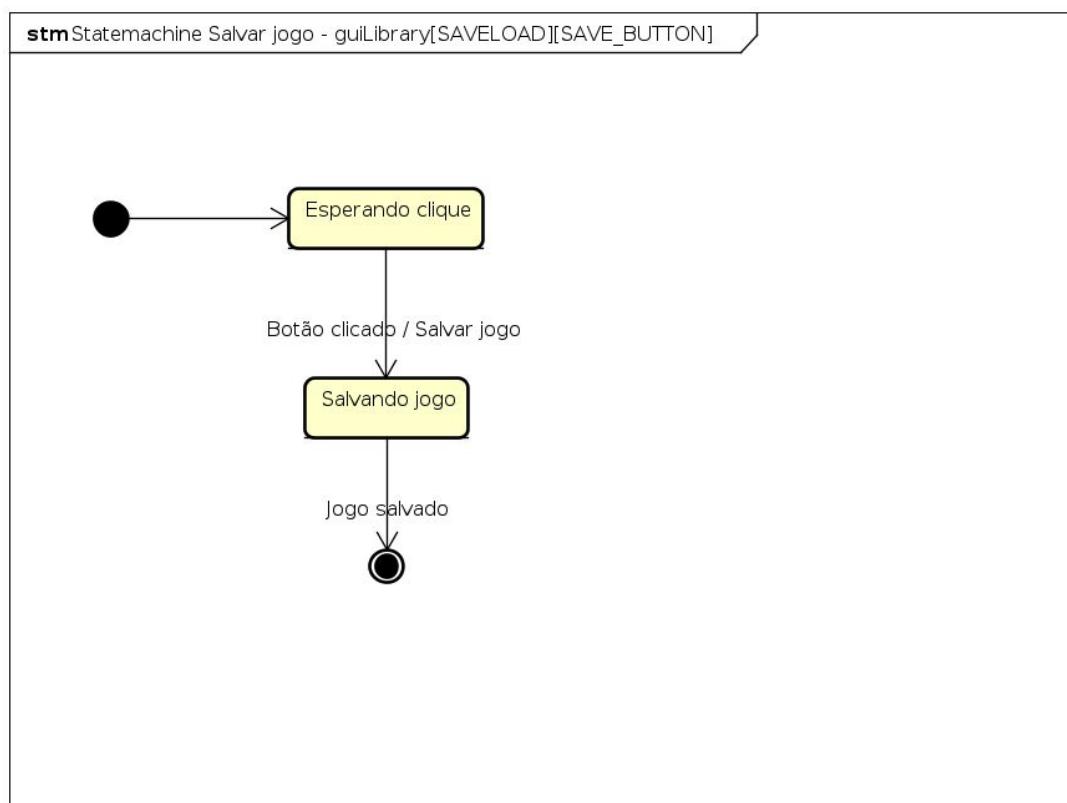


Figura 88. Diagrama Estados - Salvar jogo -
guiLibrary[SAVELOAD][SAVE_BUTTON]

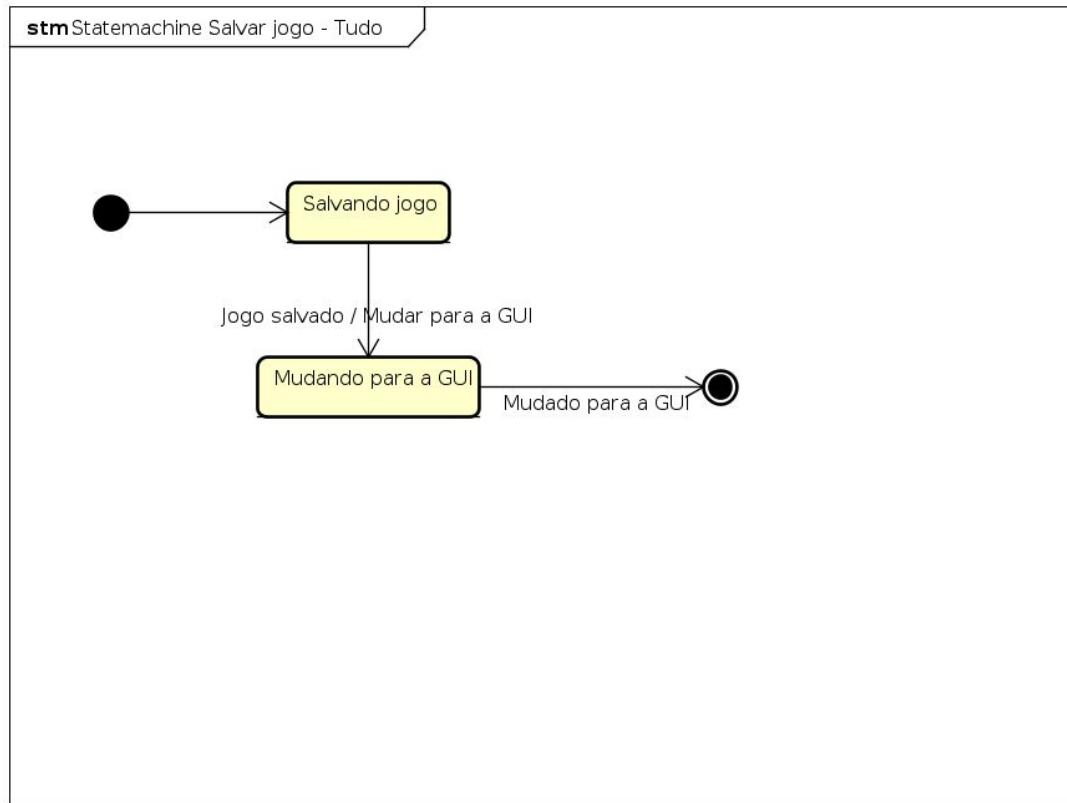


Figura 89. Diagrama Estados - Salvar jogo - Tudo

5 CONCLUSÕES

Como o grande parte da codificação base do jogo já estava feita a maior dificuldade do projeto foi fazer a engenharia reversa para documentar as classes e casos de usos do sistema.

Também foi necessário adequar o jogo à jogabilidade desejada pelo grupo, adicionando o que achávamos interessante estar disponível nele, também tendo que incluir isso nos casos de uso.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- ¹ ABOUT THE UNIFIED MODELING LANGUAGE SPECIFICATION VERSION 2.5. Disponível em: <<http://www.omg.org/spec/UML/2.5/>>. Acesso em 10 out. 2017
- ² Estratégia por turnos. Disponível em: <https://pt.wikipedia.org/wiki/Estrat%C3%A9gia_por_turnos>. Acesso em 15 out. 2017
- ³ BOOCH, G; RUMBAUGH, J e JACOBSON, I: UML, Guia do Usuário: tradução; Fábio Freitas da Silva, Rio de Janeiro, Campus, 2012.
- ⁴ Project Management Body of Knowledge, V1.0. Disponível em: <<http://www.laudelinobastos.com.br/analiseEC/PMBok3-Processos.pdf>>. Acesso em 15 out. 2017
- ⁵ Allegro game from zero. Disponível em: <https://wiki.allegro.cc/index.php?title=Allegro_game_from_zero> Acesso em 9 dez. 2017