

**FACULDADE DE TECNOLOGIA DO IPIRANGA**

**CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

CÉSAR GOMES RECHE

**Cinético: Avaliador de exercícios físicos**

SÃO PAULO

2017



**FACULDADE DE TECNOLOGIA DO IPIRANGA**

**CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

CÉSAR GOMES RECHE

**Cinético: Avaliador de exercícios físicos**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia do Ipiranga, como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas

Orientadora: Profa. Dra. Andréia C. G. Machion

SÃO PAULO

2017

|  |
| --- |
| RECHE, César Gomes  Cinético: Avaliador de exercícios físicos/César Gomes Reche; orientadora: Profa. Dra. Andréia Cristina Grisólio Machion – São Paulo, 2017  todo f.  Monografia (Graduação) – Faculdade de Tecnologia do Ipiranga  1-Sistemas 2-Desenvolvimento – RECHE, César Gomes.  Trad II – FATEC Ipiranga  CDU: \_\_\_\_\_\_\_\_2017 |

CÉSAR GOMES RECHE

**Cinético: Avaliador de movimentos físicos**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia do Ipiranga, como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Data de aprovação:

Banca examinadora:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Me. Carlos Alberto Piña Aragão

Presidente da Banca

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Me. Rodrigo Bossini Tavares Moreira

Professor Convidado

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Profa. Dra. Andréia Cristina Grisólio Machion

Professor Orientador

SÃO PAULO

2017

**AGRADECIMENTOS**

*Primeiramente, agradeço aos meus familiares que por consequência de seus esforços, direta ou indiretamente pude ter acesso à informação.*

*Agradeço a todos os professores da FATEC Ipiranga que, de certa forma, contribuíram para o meu desenvolvimento técnico, intelectual, político e acadêmico, mas principalmente aos professores Antônio Nunes Guardado, Andréia Machion, Ana Cláudia Tiessi, Carlos Aragão, Carlos Menezes, Edson Saraiva, Márcio Fontoura, Marcio Rodrigues, Norton Glaser, Rita Félix, Rodrigo Bossini, Sergio Rota e Simone Telles, que sem o apoio técnico ou emocional deles, certamente este trabalho não teria ficado com a mesma qualidade.*

*Agradeço também a todos os colegas por ter-me dado apoio sempre que possível, especialmente aos colegas: Luiz Henrique Ferreira, por ter estado sempre disposto a ajudar não só a mim, mas a todos os outros colegas; e Róger Miloni Santucci, por ter sido meu parceiro em muitos dos trabalhos realizados, e juntamente a mim, ter-se doado física e mentalmente para que pudéssemos finalizar todas as tarefas propostas por nossos professores. Por último, mas não menos importante, agradeço aos meus colegas de profissão Luiz Guilherme Pinto e Vitor Gouvea, que me incentivaram e ajudaram a finalizar este trabalho.*

**RESUMO**

A prática de exercícios físicos é algo comum na vida de um ser humano. Pessoas praticam exercícios por saúde, competição ou lazer. Muitos exercícios contêm movimentos que exigem uma maneira correta de serem executados, como uma forma eficaz de sua aplicação, e para isso é comum pessoas visitarem academias ou locais para treinamento, sendo instruídas por um profissional especialista. O problema ocorre quando a carga de alunos é muito grande para poucos instrutores. Para o instrutor, é muito difícil dar atenção aos problemas específicos de diversos alunos. Este problema se estende ao aluno, pois não tem a atenção necessária de seu instrutor, ficando assim mais sujeito a cometer erros. Através de um dispositivo sensor de movimentos e um software avaliador, este projeto visa diminuir tanto a carga sobre o instrutor, quanto a dependência da supervisão de um profissional da área para o aluno praticante de exercícios físicos. A metodologia adotada é indutiva, o tipo de pesquisa é documental de abordagem qualitativa. A tecnologia de apoio utilizada é o dispositivo Kinect v2.0, da Microsoft.

Palavras-chave: Kinect. Sensor de movimento. Exercícios físicos. Movimentos físicos. Avaliação.

**ABSTRACT**

The practice of physical exercises is a common thing in the life of the human being. People practice exercises for Health, Competition or Entertainment. Many exercises contain movements that needs a correct method of execution, as an effective manner of their application, and that makes people to go to the gym or training places, being supervised by a specialist. The problem occurs when there are many practicers for few instructors. To the instructor, it is hard to pay attention to an individual’s problem. This problem extends to the practicer, which can’t have the necessary attention from his instructor, making it more suitable for commiting execution errors for an exercice. Through a movement sensor device and a measurer software, this project aims to decrease both the charge over the instructor and a professional’s supervision dependency for the practicer. The adopted method for this project is inductive, the research type is documentary, with a qualitative approach. The supporting technology utilized is the Microsoft’s Kinect v2.0 device.

Keywords: Kinect. Movement sensor. Physical exercices. Physical movements. Measuring.

**LISTA DE FIGURAS**

[Figura 11 - Diagrama de Classes – Realizar exercício 38](#_Toc506948046)

[Figura 12 - Diagrama de Sequência – Realizar exercício 39](#_Toc506948047)

**LISTA DE ABREVIATUAS E SIGLAS**

API *Application Programming Interface*

CV *Computer Vision*

DAO *Data Access Object*

GDI *Graphics Device Interface*

GPU *Graphics Processing Unit*

K4W Kinect for Windows SDK

SDK S*oftware Development Kit*

SGBD Sistema Gerenciador de Banco de Dados

SQL S*tructured Query language*

**SUMÁRIO**

[1 INTRODUÇÃO 10](#_Toc506953039)

[**1.1** **Objetivos** 11](#_Toc506953040)

[**1.2** **Justificativa** 11](#_Toc506953041)

[**1.3** **Metodologia e Tecnologias** 11](#_Toc506953042)

[**1.4** **Organização do Trabalho** 12](#_Toc506953043)

[2 REVISÃO BIBLIOGRÁFICA 13](#_Toc506953044)

[**2.1** **Tecnologias** 13](#_Toc506953045)

[**2.1.1** **Kinect** 13](#_Toc506953046)

[**2.1.2** **Windows SDK** 14](#_Toc506953047)

[**2.1.3** **Direct3D 9** 14](#_Toc506953048)

[**2.1.4** **SQLite** 14](#_Toc506953049)

[**2.2** **Trabalhos correlatos usando o Kinect** 15](#_Toc506953050)

[3 REQUISITOS DO SISTEMA DE SOFTWARE 16](#_Toc506953051)

[**3.1** **Identificação dos requisitos** 16](#_Toc506953052)

[**3.1.1** **Requisitos Funcionais** 16](#_Toc506953053)

[**3.1.2** **Requisitos Não-Funcionais** 19](#_Toc506953054)

[**3.1.3** **Regras de negócio** 21](#_Toc506953055)

[**3.2** **Modelagem dos requisitos funcionais** 23](#_Toc506953056)

[**3.2.1** **Atores** 23](#_Toc506953057)

[**3.2.2** **Diagrama de Caso de Uso** 24](#_Toc506953058)

[Na Figura 2 é apresentado o Diagrama de Caso de Uso para requisitos do sistema. 24](#_Toc506953059)

[**3.2.3** **Especificação do Caso de Uso** 25](#_Toc506953060)

[4 DESENVOLVIMENTO DO PROJETO 34](#_Toc506953061)

[**4.1** **Análise** 34](#_Toc506953062)

[**4.1.1** **Diagrama de Classes (Visão de negócio)** 35](#_Toc506953063)

[**4.2** **Projeto** 36](#_Toc506953064)

[**4.2.1** **Arquitetura do Sistema** 36](#_Toc506953065)

[**4.2.2** **Realização dos Casos de Uso** 38](#_Toc506953066)

[***4.2.2.1*** ***Realização de Caso de uso CSU004 – Realizar exercício*** 38](#_Toc506953067)

[***4.2.2.2*** ***Realização de Caso de uso CSU006 – Gravar exercício*** 40](#_Toc506953068)

[***4.2.2.3*** ***Realização de Caso de uso CSU008 – Efetuar Login*** 42](#_Toc506953069)

[**4.2.3** **Diagrama de Atividades** 44](#_Toc506953070)

[**4.2.4** **Diagrama de Estados** 46](#_Toc506953071)

[**4.2.5** **Operações importantes** 47](#_Toc506953072)

[Nos diagramas de classe de negócio (Figura todo) e de projeto (Figura todo), o método evaluate da entidade Action são os responsáveis por definir se os gestos executados pelo usuário estão de acordo com os gestos esperados pelas ações do exercício. 47](#_Toc506953073)

[O método evaluate é responsável por executar a operação matemática associada aos gestos esperados pela ação, calcular a acurácia e acordo com a porcentagem de gestos executados corretamente, e atualizar o resultado da ação de acordo com a porcentagem obtida. O método retorna verdadeiro se todos os gestos foram avaliados, falso do contrário. 47](#_Toc506953074)

[Há 11 operações pré-definidas no Cinético. Cada operação usa o BodyPoint (bp) associado à Gesture e com exceção da operação de “Posição fixa”, o refPoint (ref) também é usado. Também podem ser atribuídos três parâmetros (px, py e pz) ao se criar um gesto para causar um deslocamento nos valores das coordenadas que estão sendo comparadas. 47](#_Toc506953075)

[Todas as funções adicionam uma folga de espaço para melhor experiência do usuário, dado que no mundo real é praticamente impossível manter-se na mesma posição exatamente. Esta folga tem como valor padrão de 0,6 cm podendo ser alteradas nas configurações gerais do aplicativo. 47](#_Toc506953076)

[O sistema de coordenadas segue o mesmo do Kinect, sendo: 47](#_Toc506953077)

[Esquerda: -x, Direita: +x, Cima: +y, Baixo: -y, Frente: -z, Traz: + 47](#_Toc506953078)

[As operações de comparação pré-programadas do Cinético são: 47](#_Toc506953079)

[**Operação** 47](#_Toc506953080)

[**Operação executada** 47](#_Toc506953081)

[Posição fixa 47](#_Toc506953082)

[se bp.x = px, bp.y = py, bp.z = pz 47](#_Toc506953083)

[À frente 47](#_Toc506953084)

[bp.z < ref.z + pz 47](#_Toc506953085)

[Atrás 47](#_Toc506953086)

[bp.position.z > ref.position.z + pz 47](#_Toc506953087)

[À direita 47](#_Toc506953088)

[bp.position.x > ref.position.x + px 47](#_Toc506953089)

[À esquerda 47](#_Toc506953090)

[bp.position.x < ref.position.x + px 47](#_Toc506953091)

[Acima 48](#_Toc506953092)

[bp.position.y > ref.position.y + py 48](#_Toc506953093)

[Abaixo 48](#_Toc506953094)

[bp.position.y < ref.position.y + py 48](#_Toc506953095)

[À mesma posição horizontal 48](#_Toc506953096)

[bp.position.x = ref.position.x 48](#_Toc506953097)

[À mesma altura 48](#_Toc506953098)

[bp.position.y = ref.position.y 48](#_Toc506953099)

[À mesma profundidade 48](#_Toc506953100)

[bp.position.z = ref.position.z 48](#_Toc506953101)

[Produto escalar 48](#_Toc506953102)

[O produto escalar faz a seguinte operação: dot(bp.lookingToVec,ref.lookingToVec); útil para verificar se dois pontos do corop estão “olhando” para o mesmo sentido ou sentidos opostos. 48](#_Toc506953103)

[Uma função pode ser extendida ao criar duas Gestures usando o mesmo BodyPoint para uma mesma ação. Por exemplo: para se criar a ação “não mover lateralmente mais que 3 cm a mão direita em relação ao centro do corpo”, pode-se utilizar a seguinte abordagem: 48](#_Toc506953104)

[ **Primeira Gesture:** 48](#_Toc506953105)

[BodyPoint: Mão direita 48](#_Toc506953106)

[Operation: Menor que 48](#_Toc506953107)

[RefPoint: Espinha 48](#_Toc506953108)

[Param: 3 cm 48](#_Toc506953109)

[ **Segunda Gesture:** 48](#_Toc506953110)

[BodyPoint: Mão direita 48](#_Toc506953111)

[Operation: Maior que 48](#_Toc506953112)

[RefPoint: Espinha 48](#_Toc506953113)

[Param: -3 cm 48](#_Toc506953114)

[5 RESULTADOS OBTIDOS 49](#_Toc506953115)

[6 CONSIDERAÇÕES FINAIS 54](#_Toc506953116)

[REFERÊNCIAS 56](#_Toc506953117)

[GLOSSÁRIO 57](#_Toc506953118)

[APÊNDICE A – PLANO E EXECUÇÃO DE TESTES 60](#_Toc506953119)

[Plano de Testes 60](#_Toc506953120)

[Roteiro de Testes 61](#_Toc506953121)

1. **INTRODUÇÃO**

Há muitos motivos pelos quais uma pessoa pratica exercícios físicos. Pode ser por lazer, estética, recuperação de lesões... Há uma vasta, quase que de tamanho inimaginável, lista de exercícios físicos que podem ser executados por um ser humano.

Em muitas atividades físicas, a forma de execução de um exercício pode ser ignorada e só visar a eficácia do mesmo; em outras, é essencial uma movimentação tecnicamente correta, aliando postura e velocidade, para se atingir um melhor resultado.

Os praticantes, cujos exercícios exigem a execução de técnicas de forma mais precisa, normalmente treinam em locais específicos e juntos a um profissional especialista, que analisa e avalia a execução dos movimentos pelo praticante. Reunir esses dois itens nem sempre é possível, seja por falta de tempo para ir a um local de treinamento, ou concorrência da atenção do profissional com outros praticantes. Uma vez que o praticante não tem como se autoavaliar, a detecção da má execução do exercício é dificultada.

Para o profissional, pela preocupação de atender a todos os praticantes, muitas vezes a atenção necessária a algum aluno especifico não é atendida.

Tendo em vista estes pontos, este projeto visa auxiliar na resolução desses problemas basicamente de duas formas: a primeira forma é servindo como uma ferramenta de ajuda para o profissional avaliar e corrigir os movimentos que porventura necessitem de melhorias; a segunda forma é permitindo a autoavaliação do praticante na ausência de algum profissional que o auxilie. Um relatório com o resultado da execução do exercício deverá ser mostrado na própria ferramenta, proporcionando ao usuário analisar e detectar pontos a serem melhorados.

Há muitos outros projetos e estudos similares que usam sensores de movimentos para ajudar pessoas a recuperarem a capacidade motora. Entretanto, estes projetos visam atingir áreas específicas como, por exemplo, a área de fisioterapia: recuperar pacientes com problemas patológicos nos ligamentos dos joelhos (MACHADO, 2013). Este trabalho se diferencia um pouco por avaliar exercícios de forma geral, em um escopo mais amplo, tentando atender a diversas áreas, não fazendo distinção sobre para qual área específica será aplicado tal exercício.

* 1. **Objetivos**

O intuito deste projeto é desenvolver um sistema computacional para avaliar movimentos executados em exercícios físicos por um usuário através de um dispositivo sensor de movimentos conectado ao computador e um software avaliador.

* 1. **Justificativa**

Exercícios físicos, embora importantes, são desgastantes. Um praticante não acostumado pode se sentir cansado em poucos minutos. Apesar disso, é interessante que a pessoa termine seu exercício com a execução correta passada pelo profissional, do contrário, isso pode prejudicar, por exemplo, um tratamento de fisioterapia, o ganho físico em um exercício para trabalho muscular ou a potência de um golpe de artes marciais. Para a área de artes marciais, há movimentos complexos, com muitos detalhes, e de fácil esquecimento para o praticante. A presença de um professor é muito importante neste caso, mas nem sempre há um professor disponível.

Este projeto visa desenvolver uma ferramenta que poderá cobrir grande parte dos problemas nas áreas mencionadas. Tanto o praticante quanto o profissional da área poderão utilizar a ferramenta, que funcionará como um avaliador de movimentos, e mostrará em um monitor qual movimento deve ser executado e a forma correta de sua execução.

* 1. **Metodologia e Tecnologias**

A metodologia adotada é indutiva, o tipo de pesquisa é documental de abordagem qualitativa.

O dispositivo sensor a ser utilizado será o Microsoft Kinect juntamente com seu *Software Development Kit (SDK),* o Kinect for Windows SDK (K4W)*,* e a documentação oficiais da Microsoft. Trabalhos, livros, artigos e bibliotecas externos sobre identificação de movimentos também serão analisados buscando alternativas de implementação para melhoria sobre a precisão e limitação das funcionalidades que o Kinect e o K4W proporcionam.

Este projeto utilizará o *SDK* oficial do Windows para desenvolvimento da interface gráfica em ambiente 2D para navegação entre telas e menus, e o *SDK* oficial do Microsoft Direct3D 9 para desenvolvimento da interface gráfica em ambiente 3D e para facilitar o uso de técnicas de computação gráfica.

Como apoio para persistência de dados, será utilizada a biblioteca SQLite, que emula as funções de um Sistema Gerenciador de Banco de Dados (SGBD) clássico, usando a linguagem *SQL* para tal.

* 1. **Organização do Trabalho**

Este trabalho é composto por seis capítulos. O primeiro capítulo contextualiza o projeto, identificando e proporcionando uma solução para os problemas encontrados. O segundo capítulo explica conceitos indispensáveis para o início do uso da ferramenta Kinect, sua tecnologia e técnicas utilizadas para implementação de funcionalidades para a aplicação, juntamente com uma introdução ao Direct3D 9. O terceiro capítulo mostra o levantamento dos requisitos e diagramas para o sistema de software, dando uma visão geral do negócio. O quarto capítulo mostra pontos importantes sobre o desenvolvimento do projeto, como análise do negócio e elaboração dos diagramas na visão puramente do negócio, oferecendo uma ideia geral do sistema, e na visão de projeto, mostrando a arquitetura física e de software, dando uma ideia de como o sistema está implementado. O quinto capítulo mostra os resultados finais obtidos com o projeto. O sexto capítulo é reservado para considerações finais.

1. **REVISÃO BIBLIOGRÁFICA**
   1. **Tecnologias**
      1. ***Kinect***

O Kinect é um dispositivo sensor de movimentos lançado oficialmente pela Microsoft em 2010. Está atualmente em sua versão 2.0 (v2), lançada oficialmente em 2013, a qual será usada neste projeto juntamente com o *Software Development Kit (SDK)* oficial. O Kinect será usado juntamente com o sistema operacional Windows, para o qual o SDK foi desenvolvido.

O Kinect v2 é composto por duas câmeras: *Depth Camera*, para detecção de profundidade e uma câmera em cores de 1080p. Conta também com quatro microfones para captura de som (MICROSOFT, 2013). A *Depth Camera* possibilita a captura de uma *depth Image.*

Uma *depth image* é uma foto em tons de cinza representando a distância entre o usuário e a câmera do Kinect. O Kinect usa luz infravermelha para detectar objetos a sua frente (BORENSTEIN, 2012, p. 1). Dentre muitas outras funcionalidades, facilitam a Remoção de fundo e a Geração de esqueleto*.*

O Kinect v2 aliado ao K4W, proporcionam diversas funcionalidades, como:

* Geração de esqueleto (*Body Tracking*);
* Leitura facial (*Face tracking*);
* Identificação e remoção de fundo;
* Gravação e identificação de gestos.

Neste projeto será utilizado a Geração de esqueleto, que oferece a posição e orientação de pontos do corpo do usuário no espaço em relação a câmera do sensor.Com essas informações, é possível desenhar um esqueleto, para fazer uma simulação, representando graficamente o corpo do usuário que está interagindo com o sensor. A lista de pontos do corpo do usuário é atualizada dinamicamente pelo K4W, e seu estado antigo é comparado com o novo para identificar movimentação oriunda do usuário.

Entretanto, o Kinect possui algumas limitações, como a frequência de atualização de informações do Kinect ser limitada a 30 Hz, proporcionando um limite de atualização a 30 quadros por segundo. Este comportamento implica na velocidade de atualização da imagem do esqueleto gerado. Outra limitação do *SDK*, pertencente à versão 2.0 do K4W é a detecção dos pontos da mão do usuário, onde não é feita a detecção dos cinco dedos da mão, mas apenas de três. Há ainda o fato de o Kinect perder precisão à medida que o usuário se aproxima da câmera, limitando seu uso apenas para locais espaçosos.

* + 1. ***Windows SDK***

O SDK do Windows oferece uma vasta gama de ferramentas, bibliotecas, APIs (*Application Programming Interface*) e aplicativos para o desenvolvedor poder usufruir das capacidades do sistema, incluindo fazer uso de seu sistema de controles comuns e de sua API gráfica (*GDI*). Toda interface gráfica em ambiente 2D apresentada pelo Cinético é baseada nestes componentes.

* + 1. ***Direct3D 9***

O Direct3D 9 é um conjunto de *APIs* para acesso quase direto (há uma camada de abstração) com a placa de vídeo (*GPU*) do sistema (MICROSOFT, 2017). Lançada em dezembro de 2002, é ainda muito utilizada pela sua facilidade de uso em relação a suas versões sucessoras e pela bela qualidade gráfica que ainda proporciona.

* + 1. ***SQLite***

SQLite é uma biblioteca para persistência de dados usando a linguagem *SQL*. O diferencial do SQLite, além de sua leveza e praticidade, é que não usa conexões de rede para se comunicar com a aplicação. SQLite é incluído como uma biblioteca e sem a necessidade de um driver intermediador, oferece acesso a sua *API* (SQLITE, 2017).

* 1. **Trabalhos correlatos usando o Kinect**

Há um grande acervo de projetos que já utilizaram e comprovaram a eficácia do Kinect. Dentre eles, é possível citar alguns projetos que fazem parte de um escopo similar ao deste projeto:

* Sistema para fisioterapia (CAMPOS, 2013), onde o autor consegue demonstrar que é possível usar o Kinect para acompanhar e auxiliar na recuperação física de um paciente;
* Kinect para reabilitação (KOSTRIUBA et al, 2014), onde os autores criaram um jogo para ajudar na reabilitação de movimentos físicos, inclusive recebendo resposta de mais de 90% dos usuários apoiando a ideia de realizar a fisioterapia através de jogos;
* Sistema interativo baseado em gestos para utilização de comandos no computador (ALMEIDA, 2014), onde foi criado um sistema para executar determinado comando através de gestos pré-definidos do usuário;
* Proposta de sistema baseado na plataforma Kinect para suporte a reabilitação de pacientes com patologias ligamentares nos joelhos (MACHADO, 2013), onde é feito um estudo pelo autor da possibilidade de se usar Kinect para reabilitação de pacientes com problemas no joelho. O autor consegue ao final de seu trabalho uma resposta positiva.

1. **REQUISITOS DO SISTEMA DE SOFTWARE**

Este capítulo tem como objetivo especificar os requisitos funcionais, não funcionais e as regras de negócio, bem como apresentar o protótipo das telas do sistema.

* 1. **Identificação dos requisitos** 
     1. ***Requisitos Funcionais***

**[RF001] – Capturar posição do usuário**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: A aplicação terá conhecimento do usuário que estiver em frente ao sensor de movimento.

**[RF002] – Criar corpo**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: Permite a detecção de Pontos do corpo do usuário que estiver em frente ao sensor.

**[RF003] – Analisar movimentação corporal**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: Um movimento executado será detectado e classificado como um tipo de movimento: linear ou curvilíneo. Também serão analisados posicionamento e velocidade.

**[RF004] – Analisar ação**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: Gestos de uma ação feitos pelo usuário serão identificados e classificados.

**[RF005] – Avaliar acurácia da ação**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: A acurácia do usuário ao executar uma determinada ação será avaliada. De acordo com determinada taxa de porcentagem, a ação poderá ser avaliada como aceita ou rejeitada.

**[RF006] – Analisar exercício**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: O progresso de um exercício será analisado de forma geral.

**[RF007] – Avaliar acurácia do exercício**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: A acurácia média do usuário ao executar todas as ações do exercício será avaliada. De acordo com determinada taxa de porcentagem, o exercício poderá ser avaliado como aceito ou rejeitado.

**[RF008] – Informar resultado do exercício**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: Será mostrado um relatório sobre o desempenho do usuário sobre a execução do exercício.

**[RF09] – Realizar exercício**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: O usuário poderá executar um exercício selecionado de uma lista de exercícios associada ao seu perfil.

**[RF010] – Cancelar exercício**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | 🞎 | Essencial | ◼ | Importante | 🞎 | Desejável |

**Descrição**: O usuário poderá cancelar a realização de um exercício que já está iniciado.

**[RF011] – Manter exercício**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: O usuário poderá criar, alterar, consultar o remover um exercício de uma lista de exercícios associado ao seu perfil.

**[RF012] – Gravar novo exercício**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | 🞎 | Essencial | 🞎 | Importante | ◼ | Desejável |

**Descrição**: O usuário poderá gravar movimentos utilizando a própria movimentação corporal para criação de um novo exercício.

**[RF013] – Manter perfil**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: O usuário poderá criar, alterar, ou excluir seu perfil, que será usado para identificá-lo no sistema.

**[RF014] – Efetuar login**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: O usuário poderá entrar no sistema através de um nome de usuário e, opcionalmente, senha.

* + 1. ***Requisitos Não-Funcionais***

**[RNF001] – Manual do usuário**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | 🞎 | Essencial | ◼ | Importante | 🞎 | Desejável |

**Descrição**: Deve-se prover um manual de instruções digital para usabilidade e comandos que a aplicação usa.

**[RNF002] – Linguagem de apresentação**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: A aplicação deve ter suporte às linguagens Português (do Brasil) e Inglês.

**[RNF003] – Usabilidade**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: O sistema deve prover uma interface baseada nos seguintes critérios ergonômicos visando facilidade de aprendizagem e utilização:

* Orientação a convite: deixar claro o que o usuário pode fazer;
* Ações mínimas: Diminuir a necessidade de acessos desnecessários a outras telas;
* Legibilidade: Contraste de fundo e tamanho da fonte não deve dificultar a leitura pelo usuário;
* Coerência Interna: Padronizar tamanho de fonte, tamanho dos campos, cores entre as telas etc.

**[RNF004] – Apresentação da Interface Gráfica**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | 🞎 | Essencial | ◼ | Importante | 🞎 | Desejável |

**Descrição**: Para os componentes das telas, o sistema deve manter uma coerência entre tamanho, fonte e cor para componentes que compartilhem de um mesmo grupo de funções. Para *viewports,* o sistema deve ser capaz de suportar diversas resoluções suportadas pelo monitor e a placa de vídeo, com *aspect ratios* de 4:3, 5:4, 16:9 e outros padrões adotados. A resolução mínima permitida é de 480x320.

**[RNF005] – Linguagem de programação**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: A linguagem a ser utilizada deverá ser C++.

**[RNF006] – Sistema Operacional**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: O sistema operacional a ser utilizado deve ser Windows 8 ou 10.

* + 1. ***Regras de negócio***

**[RN001] – Visibilidade de Exercício**

**Descrição:** Um usuário só poderá ver e executar os exercícios que o mesmo criou ou quando o exercício esteja marcado como público. Um exercício pode ser marcado como público na sua criação ou edição, possibilitando que qualquer usuário o veja e execute.

**[RN002] – Criação de Exercício**

**Descrição:** Um exercício, para ser executado, deve conter pelo menos uma ação.

**[RN003] – Criação de ação**

**Descrição:** Uma ação, para ser criada, deve conter pelo menos um gesto.

**[RN004] – Tipo de Gesto**

**Descrição:** Um gesto pode ser do tipo Simples (posicionamento com movimentação prévia livre) ou de Movimento (necessita de um deslocamento previamente definido).

**[RN005] – Tipo de movimento**

**Descrição:** Um movimento pode ser de dois tipos: Linear - O movimento de um ponto do corpo deve seguir em uma reta, salvo por uma pequena folga de erro de 5 cm; Curvilíneo (ou suave) – O movimento deve passar por pontos previamente definidos de forma suave, fazendo curva quando necessário.

**[RN006] – Avaliação de ação**

**Descrição:** Uma ação pode ser avaliada em: *Missed* (Insatisfatória), *Bad* (Ruim), *Good* (Boa) ou *Excellent* (Excelente). Uma ação é avaliada mediante a porcentagem de gestos acertados:

* Missed: abaixo de 50% (cinquenta por cento);
* Bad: abaixo de 75% (oitenta por cento);
* Good: abaixo de 95% (noventa e cinto por cento);
* Excellent: acima ou igual de 95% (noventa e cinco por cento).

Uma ação será considera *Accepted* (Aceita) se for avaliada em, pelo menos, *Good*. Do contrário, a ação será considerada *Rejected* (Rejeitada).

**[RN007] – Avaliação de exercício**

**Descrição:** Um exercício pode ser avaliado em: Missed (Insatisfatório), Bad (Ruim), Good (Bom) ou Excellent (Excelent). Um exercício será avaliado de acordo com a média do acerto de suas ações:

* Missed: abaixo de 50% (cinquenta por cento);
* Bad: abaixo de 75% (oitenta por cento);
* Good: abaixo de 95% (noventa e cinto por cento);
* Excellent: acima ou igual de 95% (noventa e cinco por cento).

Um exercício será considerado *Accepted* (Aceito) se for avaliado em, pelo menos, *Good*. Do contrário, o exercício será considerado *Rejected* (Rejeitado).

* 1. **Modelagem dos requisitos funcionais**

Neste item são descritos os requisitos a serem atendidos funcionalmente pelo sistema de uma forma simples, possibilitando a compreensão do comportamento do sistema pela perspectiva do usuário. São descritos os atores e o Diagrama de Caso de Uso.

* + 1. ***Atores***

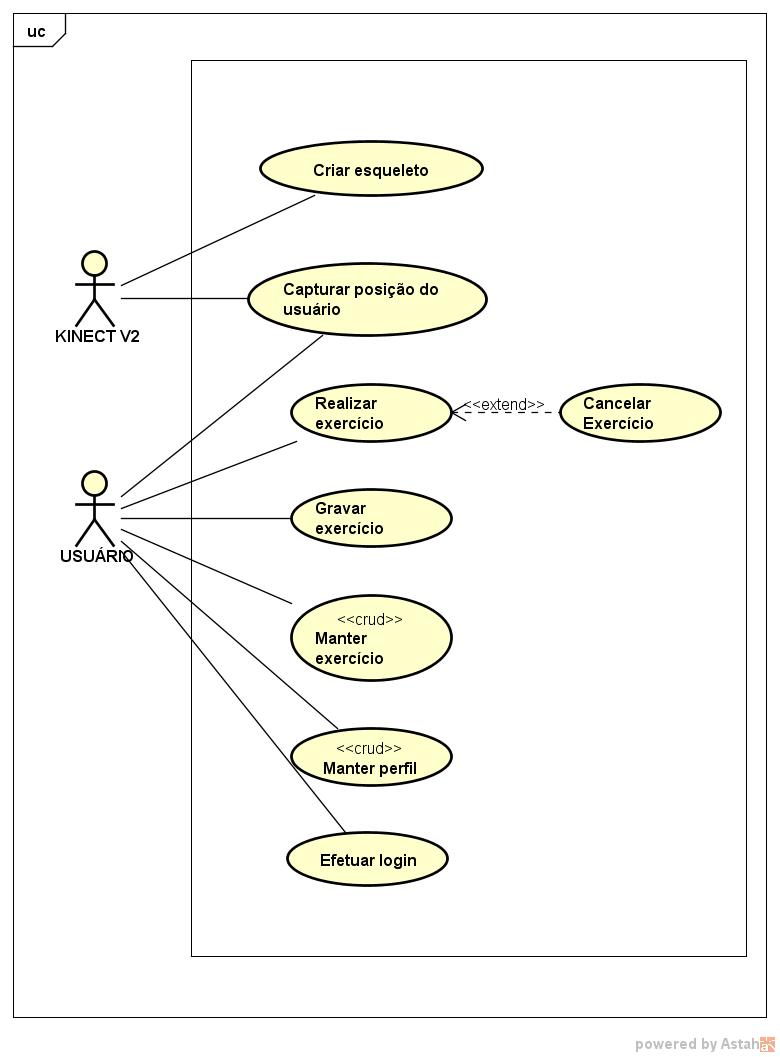
**USUÁRIO:** Representa a pessoa física que irá utilizar o sistema.

**KINECT V2:** Representa o sensor de movimentos juntamente com o SDK original da Microsoft e suas bibliotecas associadas

* + 1. ***Diagrama de Caso de Uso***

Na Figura 2 é apresentado o Diagrama de Caso de Uso para requisitos do sistema.

Figura 1 - Diagrama de Caso de Uso



Fonte: Autor (2017)

* + 1. ***Especificação do Caso de Uso***

|  |  |
| --- | --- |
| **CSU001 – Capturar posição do usuário** | |
| Sumário: | Capturar posição do usuário em frente ao sensor |
| Ator Primário: | KINECT V2 |
| Ator Secundário: | USUÁRIO |
| Casos de Uso Associados: | |
| **Pré-condição:**   1. O usuário deve estar autenticado no sistema. 2. O exercício deve ter sido iniciado pelo usuário. | |
| **Fluxo Principal:**  **1 –** O Kinect ficará “observando” mudanças de movimentação.  **2 –** O usuário se mantém em frente ao Kinect executando ações necessárias para o exercício.  **3 –** O Kinect automaticamente captura a imagem do usuário, servindo de entrada para o SDK.  **4 –** O SDK calcula a posição do usuário, de acordo com a entrada provida do Kinect.  **5 –** O caso de uso termina. | |
| **Fluxo de exceção (4):** Corpo não identificado   1. Se o corpo não puder ter sido identificado, o sistema mantém informações anteriores sobre a posição dos pontos do corpo do usuário. 2. O caso de uso termina. | |
| **Requisitos:** RF001 | |
| **Regras de Negócio:** | |
| **Interface:** | |

|  |  |
| --- | --- |
| **CSU002 – Criar corpo** | |
| Sumário: | Identificar partes do corpo do usuário |
| Ator Primário: | KINECT V2 |
| Ator Secundário: |  |
| Casos de Uso Associados: CSU001 | |
| **Pré-condição:**   1. O corpo do usuário deve ter sido identificado (CSU001). | |
| **Fluxo Principal:**  **1 –** O SDK identifica pontos do corpo (extremidades ou pontos de articulação) do usuário com base na imagem gerada pelo Kinect.  **2 –** O caso de uso termina. | |
| **Requisitos:** RF002 | |
| **Regras de Negócio:** | |
| **Interface:** | |

|  |  |
| --- | --- |
| **CSU003 – Realizar exercício** | |
| Sumário: | Escolher e realizar um exercício da lista de exercícios |
| Ator Primário: | USUÁRIO |
| Ator Secundário: |  |
| Casos de Uso Associados: | |
| **Pré-condição:**   1. O usuário deve estar autenticado no sistema. | |
| **Fluxo Principal:**  **1 –** O usuário escolhe e inicia um exercício na lista de exercícios.  **2 –** O sistema redireciona o usuário para a tela de Realização de Exercícios.  **3 –** O usuário realiza posições e movimentos de acordo com a ação atual.  **4 –** O sistema avalia a ação do usuário e, se concluída, segue para a próxima ação.  **5 –** Após todas as ações serem executadas, o exercício termina.  **6 –** O sistema avalia o exercício de acordo com a avaliação das ações do mesmo.  **7 –** O resultado do exercício é mostrado ao usuário.  **8 –** O caso de uso termina. | |
| **Fluxo alternativo (3): Cancelar exercício**   1. O usuário cancela o exercício sendo executado. 2. O caso de uso termina. | |
| **Requisitos:** RF003, RF004, RF005, RF006, RF007, RF008, RF009, RF010 | |
| **Regras de Negócio:** RN001, RN004, RN005, RN006, RN007 | |
| **Interface:** Figura 2 | |

Figura 2 - Protótipo de tela: Realização de Exercício

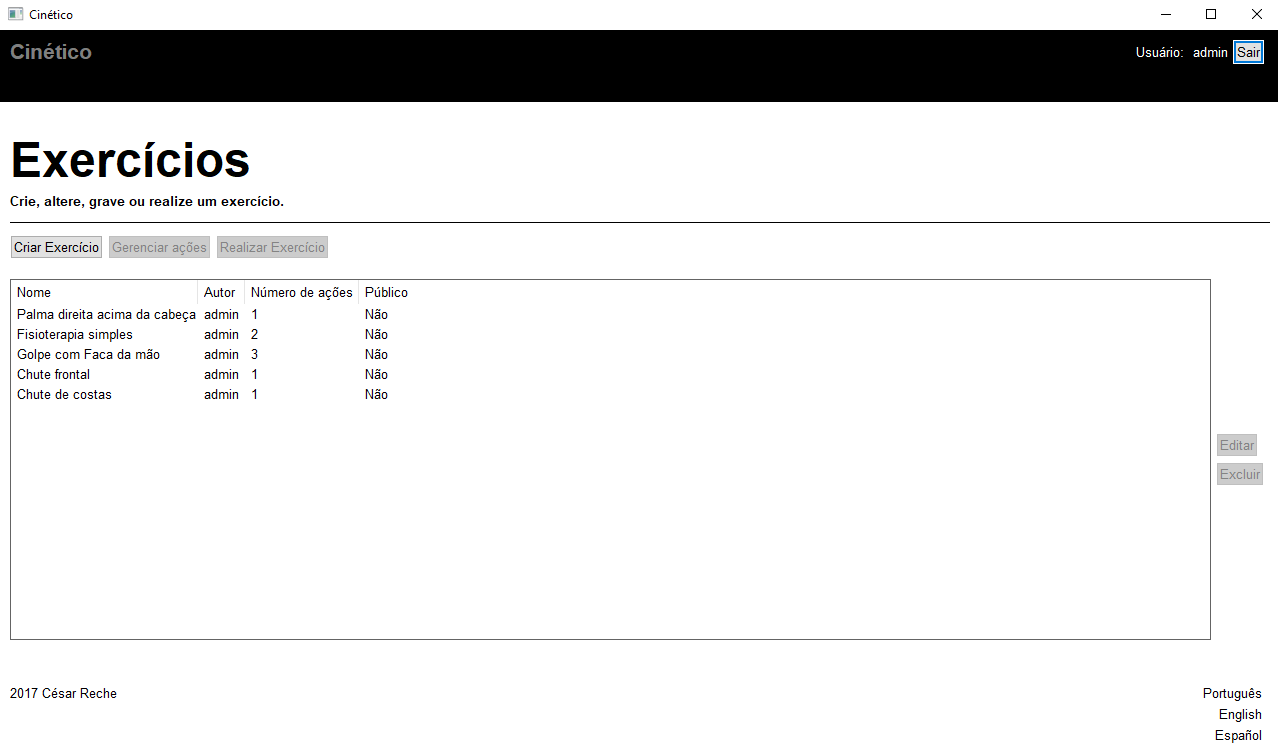
****

Fonte: Autor (2017)

|  |  |
| --- | --- |
| **CSU004 – Cancelar exercício** | |
| Sumário: | Cancelar a realização de um exercício que já está iniciado. |
| Ator Primário: | USUÁRIO |
| Ator Secundário: |  |
| Casos de Uso Associados: CSU003 | |
| **Pré-condição:**   1. O usuário deve estar autenticado no sistema. 2. O usuário deve estar executando um exercício | |
| **Fluxo Principal:**  **1 –** O usuário seleciona a opção de “Cancelar exercício”.  **2 –** O sistema encerra a realização de exercício.  **3 –** O sistema informa o resultado do exercício até o momento.  **4 –** O caso de uso termina. | |
| **Requisitos:** RF010 | |
| **Regras de Negócio:** | |
| **Interface:** | |

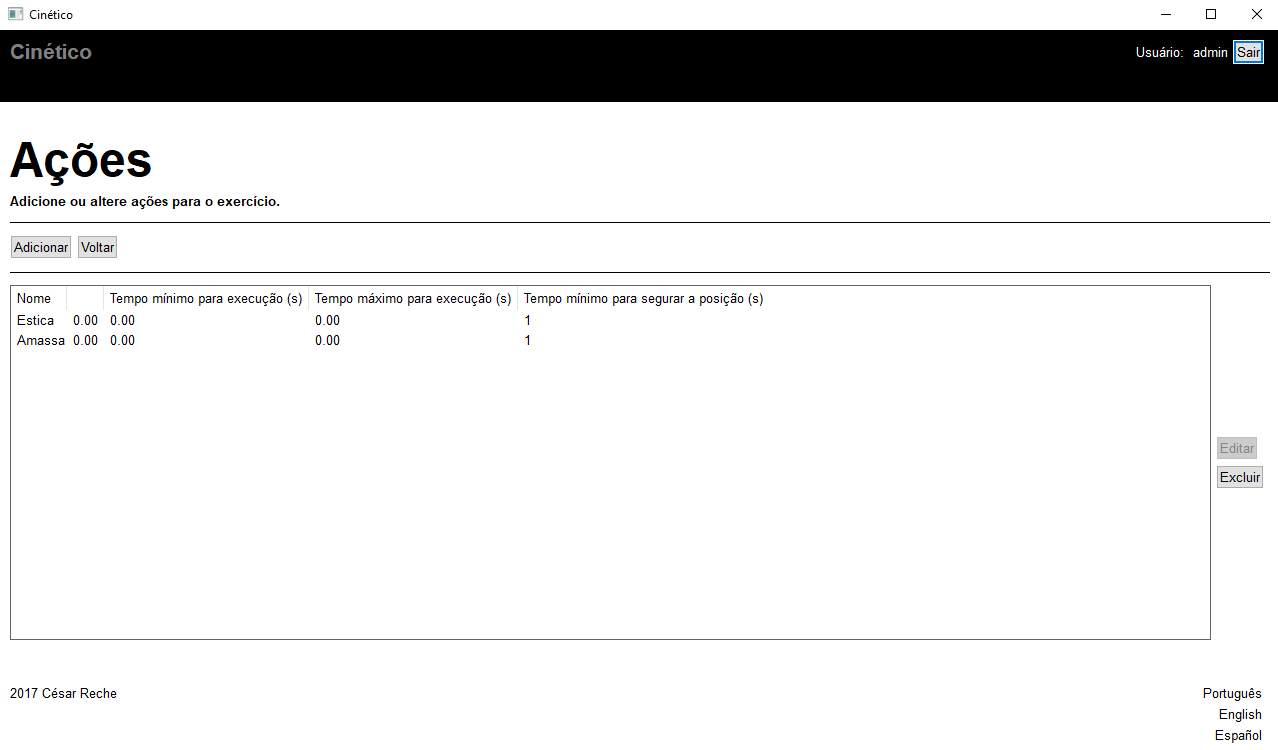
|  |  |
| --- | --- |
| **CSU005 – Manter exercício** | |
| Sumário: | Criar, atualizar, consultar e remover exercício. |
| Ator Primário: | Usuário |
| Ator Secundário: |  |
| Casos de Uso Associados: | |
| **Pré-condição:**   1. O usuário deve estar autenticado no sistema. | |
| **Fluxo Principal:**  **1 –** O usuário seleciona a opção para criar exercício.  **2 –** O sistema redireciona o usuário para a tela de Gerenciamento de Exercício.  **3 –** O usuário insere o nome do exercício e marca se é público.  **4** – O usuário salva o exercício.  **5 –** O usuário aciona a opção para adicionar Ações ao exercício.  **6 –** O sistema disponibiliza os seguintes campos para edição:  Nome;  Tempo mínimo para execução da ação;  Tempo máximo para execução da ação;  Tempo mínimo para segurar a posição;  Campos adicionais para adicionar Gestos.  **7 –** O usuário seleciona a opção de salvar ação.  **8 –** O sistema valida os campos.  **9 –** Se preferir, o usuário pode adicionar mais ações. Nesse caso, o usuário volta ao passo 5 deste caso de uso.  **10 –** O usuário seleciona a opção de salvar ação.  **11 –** O sistema valida os campos.  **12 –** O caso de uso termina. | |
| **Fluxo Alternativo (1): Edição de exercício**  O usuário seleciona a opção de edição de exercício.  O sistema redireciona o usuário para a tela de Gerenciamento de Exercício.  O usuário pode editar quaisquer campos, inclusive o nome do exercício.  O caso de uso termina. | |
| **Fluxo Alternativo (1): Remoção de exercício**  O usuário seleciona a opção para remover o exercício.  Se o exercício for público e o próprio usuário não for o criador, é mostrada uma mensagem de erro ao usuário: “Somente o criador do exercício pode remover um exercício público”. Do contrário, o exercício é removido da lista de exercícios.  O caso de uso termina. | |
| **Fluxo Alternativo (1): Busca de exercícios**  O usuário usa o campo de filtro para busca.  O sistema atualiza a lista de exercícios somente com resultados encontrados.  O caso de uso termina. | |
| **Fluxo de Exceção (9): Nome de exercício já existe**  O sistema mostra a mensagem: “Exercício com este nome já existe. Por favor, escolha outro nome.”.  O sistema redireciona o usuário ao passo 3 do Fluxo Principal. | |
| **Requisitos:** RF011 | |
| **Regras de Negócio:** RN002, RN003, RN004, RN005 | |
| **Interface:** Figura 3; Figura 4; Figura 5 | |

Figura 3 - Protótipo de tela: Exercícios



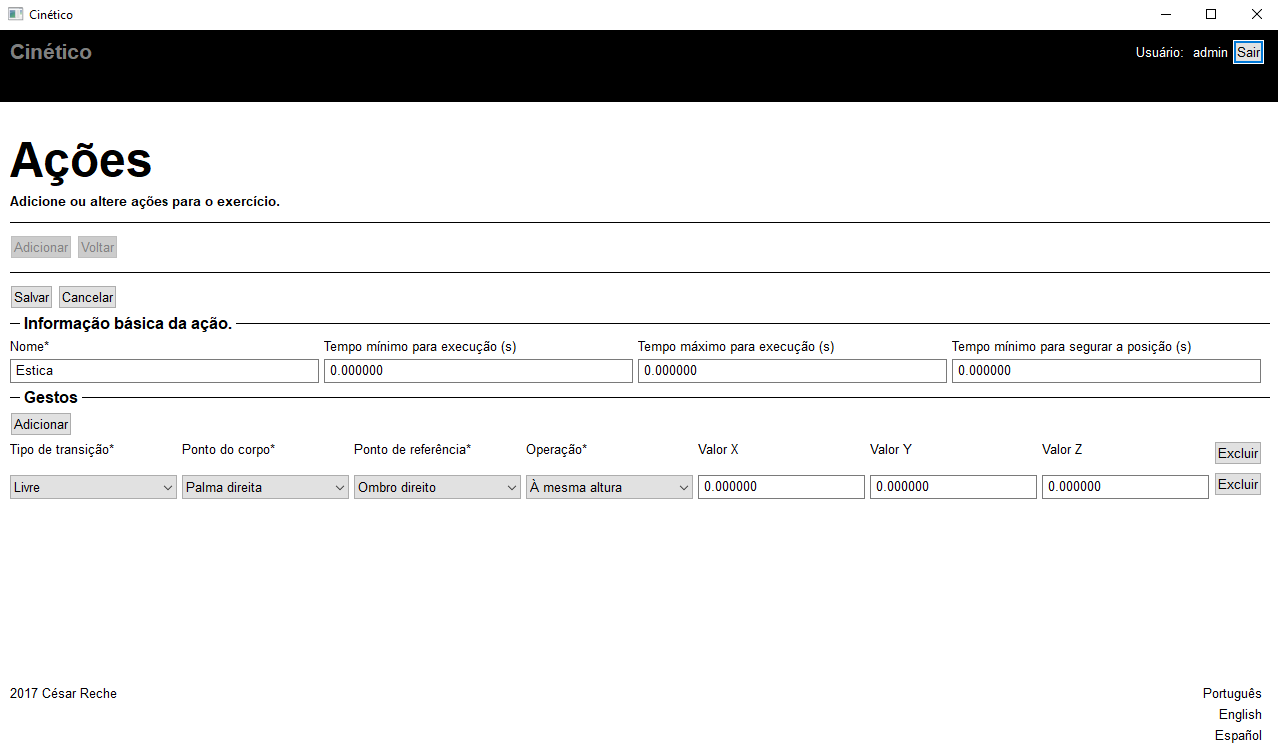
Fonte: Autor (2017)

Figura 4 - Protótipo de tela: Ações



Fonte: Autor (2017)

Figura 5 - Protótipo de tela: Ações (campos)

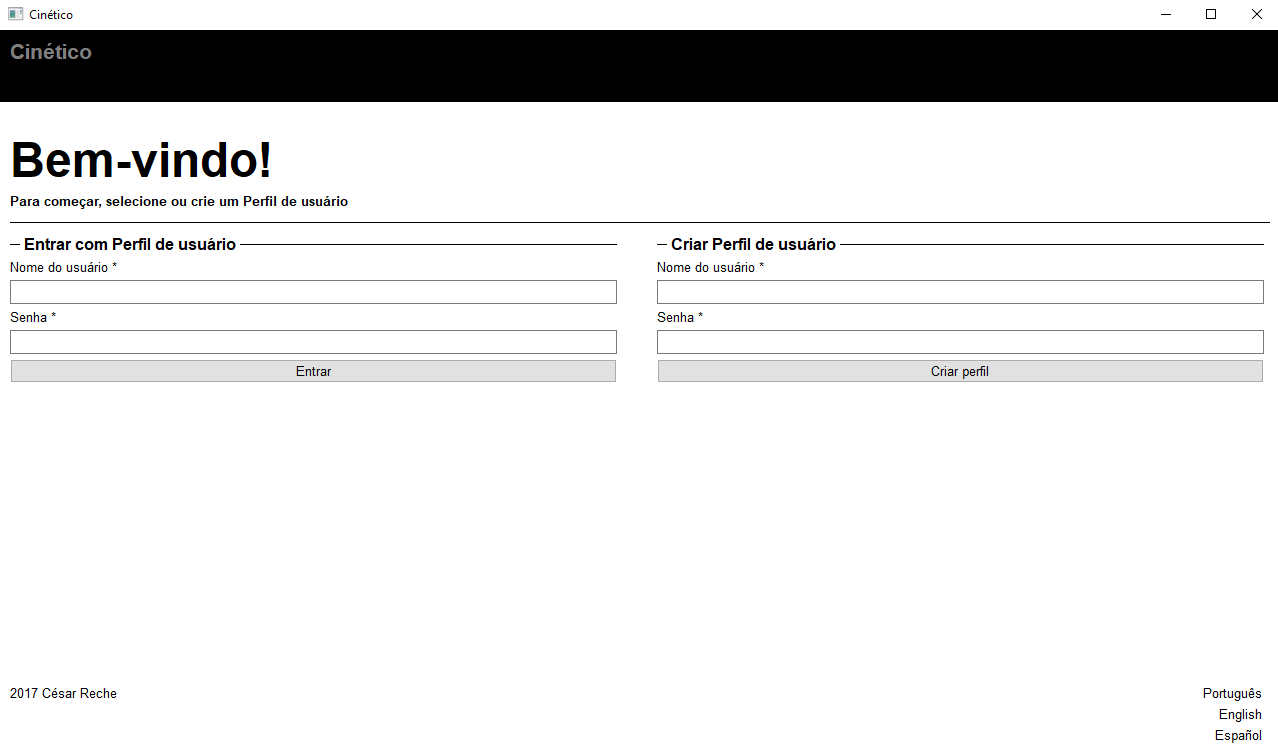


Fonte: Autor (2017)

|  |  |
| --- | --- |
| **CSU006 – Gravar exercício** | |
| Sumário: | Gravar exercício através de movimentação |
| Ator Primário: | USUÁRIO |
| Ator Secundário: KINECT2.0 | |
| Casos de Uso Associados: | |
| **Pré-condição:**   1. O usuário deve estar autenticado no sistema | |
| **Fluxo Principal:**  **1 –** O usuário seleciona a opção de “Gravar exercício”.  **2 –** O sistema redireciona o usuário a tela de “Realizar exercício” em um modo especial de gravação.  **3 –** O usuário executa movimento em frente ao Kinect.  **4 –** O sistema grava os movimentos realizados pelo usuário.  **5** – O usuário seleciona a opção de “Parar gravação”.  **6 –** O sistema interpreta pausas e separa os movimentos.  **7 –** O sistema cria um exercício com base nos movimentos capturados.  **8 –** O usuário informa um nome para o exercício e marca se é público.  **9 –** O sistema salva o exercício na lista de exercícios.  **10 –** O caso de uso termina. | |
| **Fluxo de Exceção (8): Nome de exercício já existe**   1. O sistema mostra a mensagem: “Exercício com este nome já existe. Por favor, escolha outro nome.”. 2. O sistema redireciona o usuário ao passo 3 do Fluxo Principal. | |
| **Pós-condições:** | |
| **Requisitos:** RF012 | |
| **Regras de Negócio:** | |
| **Interface:** | |

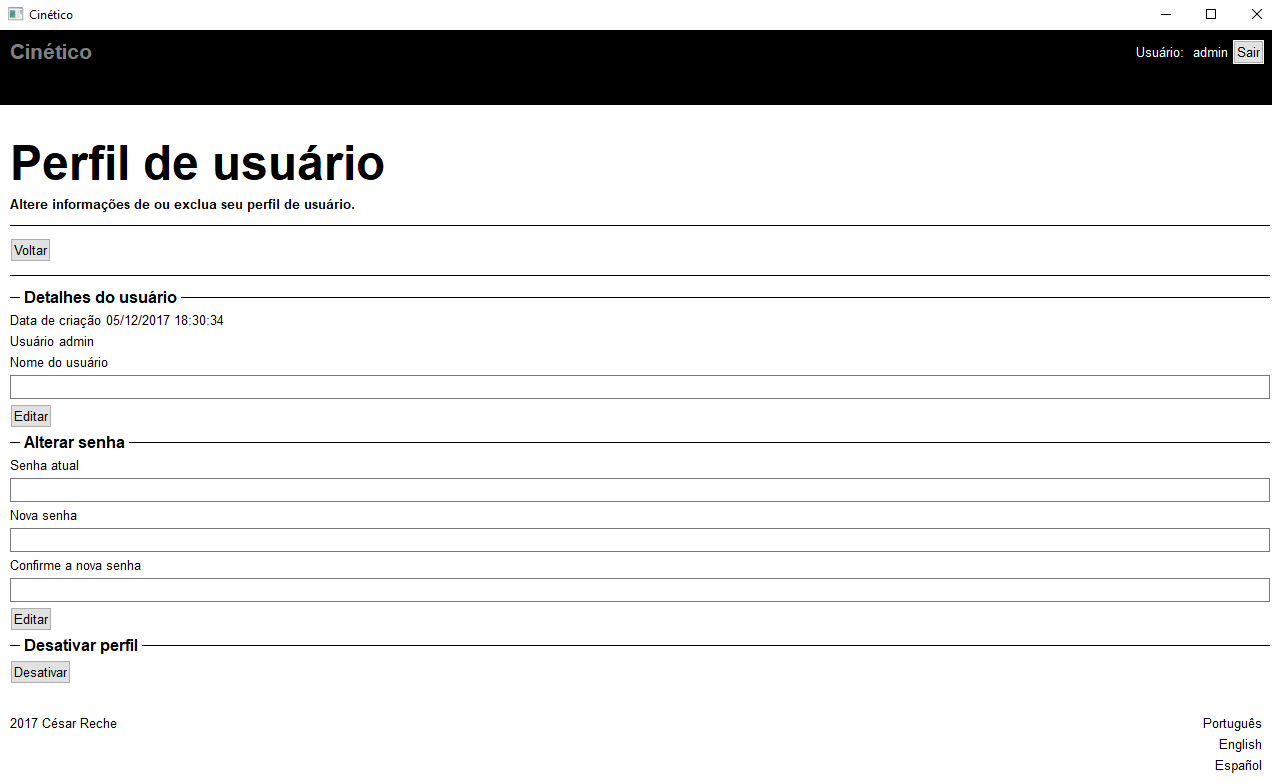
|  |  |
| --- | --- |
| **CSU007 – Manter perfil** | |
| Sumário: | Criar, alterar ou excluir perfil de usuário. |
| Ator Primário: | Usuário |
| Ator Secundário: | |
| Casos de Uso Associados: | |
| **Pré-condição:** | |
| **Fluxo Principal:**  **1 –** Na tela principal, o usuário insere o nome e, opcionalmente a senha de usuário e seleciona a opção de “Criar perfil”.  **2 –** O sistema valida nome de usuário e senha.  **3 –** O perfil é criado e o caso de uso termina. | |
| **Fluxo Alternativo (1): Alterar senha**   1. O usuário escolhe “Alterar senha”. 2. O usuário informa a senha atual e a nova senha. 3. O sistema compara ambas as senhas. Se estiverem iguais, a senha é alterada e o sistema retorna a mensagem “Senha alterada com sucesso” para o usuário; se estiverem diferentes, o sistema alerta o usuário com a mensagem: “Senhas não conferem.”. 4. O caso de uso termina. | |
| **Fluxo Alternativo (1): Excluir perfil**   1. O usuário escolhe a opção de “Excluir perfil”. 2. O sistema pede uma confirmação ao usuário com as opções “Sim” e “Não”. 3. Se o usuário selecionar sim, o sistema excluir o perfil do usuário e todos os exercícios relacionados a ele; se o usuário selecionar não, nada é feito. 4. O caso de uso termina. | |
| **Fluxo de Exceção (2): Usuário já existe**   1. O sistema mostra a mensagem: “Usuário já existe”. 2. O sistema redireciona o usuário para o passo 1 do Fluxo Principal. | |
| **Pós-condições:** | |
| **Requisitos:** RF013 | |
| **Regras de Negócio:** | |
| **Interface:** Figura 6; Figura 7 | |

Figura 6 - Protótipo de tela: Login



Fonte: Autor (2017)

Figura 7 - Protótipo de tela: Perfil de usuário



Fonte: Autor (2017)

|  |  |
| --- | --- |
| **CSU008 – Efetuar login** | |
| Sumário: | Entrar no sistema |
| Ator Primário: | USUÁRIO |
| Ator Secundário: | |
| Casos de Uso Associados: | |
| **Pré-condição:** Usuário cadastrado no sistema | |
| **Fluxo Principal:**  **1 –** O usuário insere o nome usado na criação de perfil de usuário.  **2 –** O sistema valida o login.  **3 –** O sistema redireciona o usuário para a tela de Seleção de Exercícios.  **4 –** O caso de uso termina. | |
| **Fluxo alternativo (1): Senha necessária**  **1 –** O usuário insere a senha.  **2 –** O sistema valida o login.  **3 –** O sistema redireciona o usuário para a tela de Seleção de Exercícios.  **4 –** O caso de uso termina. | |
| **Fluxo de Exceção (2): Senha necessária**  **1 –** O sistema alerta o usuário sobre a necessidade da senha mostrando a mensagem “Digite a senha de usuário.”.  **2 –** O sistema retorna ao passo 1 do Fluxo Principal. | |
| **Fluxo de Exceção (2): Usuário inválido**  **1 –** O sistema mostra a mensagem “Usuário inválido.” ao usuário.  **2 –** O sistema retorna ao passo 1 do Fluxo Principal. | |
| **Fluxo de Exceção (2): Usuário inválido**  **1 –** O sistema mostra a mensagem “Senha inválida.” ao usuário.  **2 –** O sistema retorna ao passo 1 do Fluxo Principal. | |
| **Pós-condições:**   1. Usuário autenticado no sistema. | |
| **Requisitos:** RF014 | |
| **Regras de Negócio:** | |
| **Interface:** Figura 6 | |

1. **DESENVOLVIMENTO DO PROJETO**

Este capítulo tem como objetivo analisar, detalhar e propor uma solução geral do sistema, sob o ponto de vista de negócio, de acordo com os requisitos levantados e validados no capítulo 3.

* 1. **Análise**

Com base nos requisitos funcionais, é possível identificar seis entidades que devem estar presentes no sistema para que todos os requisitos funcionais sejam atendidos. São elas: Gesture, Action, Exercise, Body, BodyPoint e UserProfile.

* BodyPoint: representa um ponto do corpo, podendo ser uma parte comum do corpo (cabeça, por exemplo), ou articulação (joelho, por exemplo);
* Body representa o corpo do usuário, que contém múltiplos BodyPoints;
* Gesture: representa um gesto com uma orientação bem definida já, ou para ser, realizado pelo usuário utilizando alguma parte do corpo. Sempre estará associada a um BodyPoint, que representa a parte do corpo que está sendo movida, e quase sempre a um Ponto de referência (refPoint), que poderá ser um outro BodyPoint ou o próprio sensor do Kinect. Também carregará uma operação pré-programada que será usada para definir como esse gesto deverá ser executado.

Por exemplo: Posicionar Mão direita acima da Cabeça. Nesta Gesture, “Mão direita” é o BodyPoint, “Cabeça” é o refPoint e “acima” é uma função pré-definida simples que executa uma operação matemática de comparação.

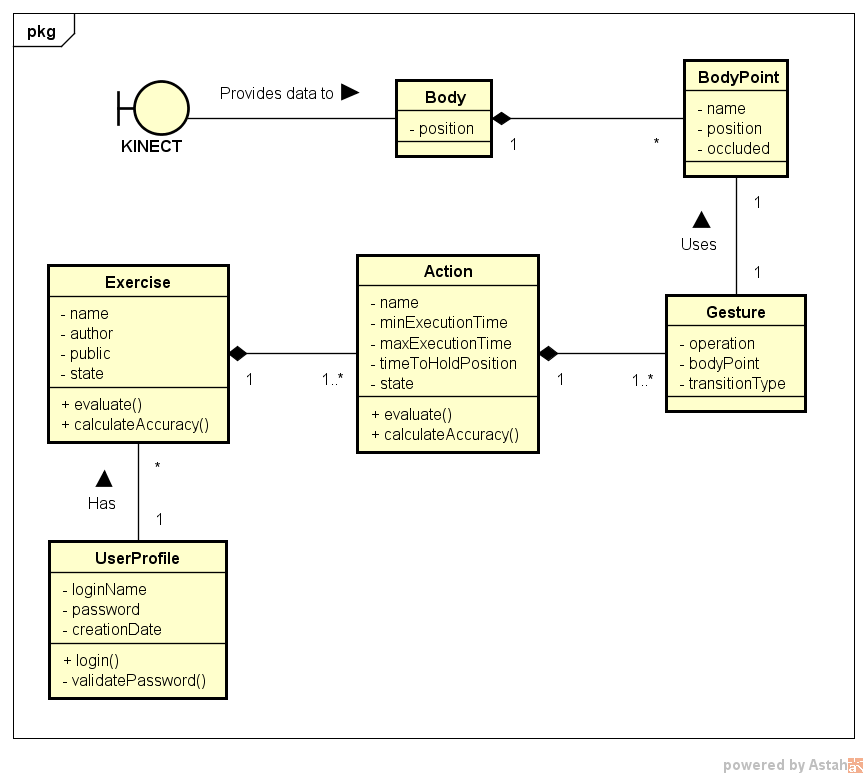
* Action: representa uma ação já, ou para ser, executada pelo usuário. É definida como um conjunto de Gestures sendo executados ao mesmo tempo.
* Exercise: representa um exercício já, ou para ser, realizado pelo usuário. É definido como um conjunto de Actions.
* UserProfile: identifica um usuário do sistema.

O item 4.1.1 deste capítulo demonstra visualmente a relação entre estas entidades.

* + 1. ***Diagrama de Classes (Visão de negócio)***

Na Figura 8 são apresentadas as classes mais importantes para o desenvolvimento do negócio. Não há detalhamento, pois o intuito é demonstrar uma visão geral das classes necessárias para atingir os requisitos de negócio.

Figura 8 - Diagrama de Classes (Visão de negócio)



Fonte: Autor (2017)

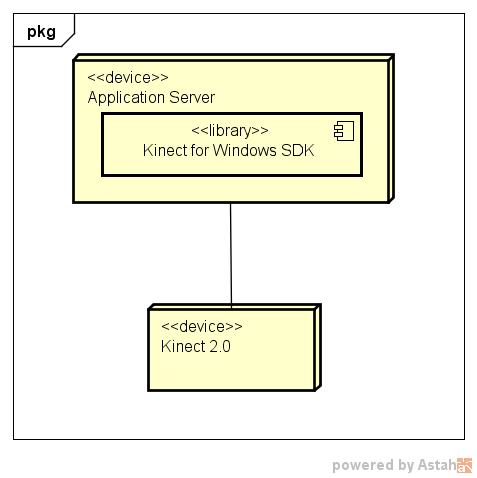
* 1. **Projeto**

Neste item são apresentados os diagramas mais importantes que representam uma visão mais íntima do funcionamento do sistema. Há um detalhamento maior bem similar à implementação final do sistema.

* + 1. ***Arquitetura do Sistema***

A infraestrutura necessária para o funcionamento do Kinect é conceitualmente simples, a Figura 9 mostra um servidor de aplicação que representa o hardware do sistema conectado ao dispositivo Kinect, e que contém o sistema de software Cinético, que inclui o *SDK* do Kinect.

Figura 9 - Diagrama de Implantação



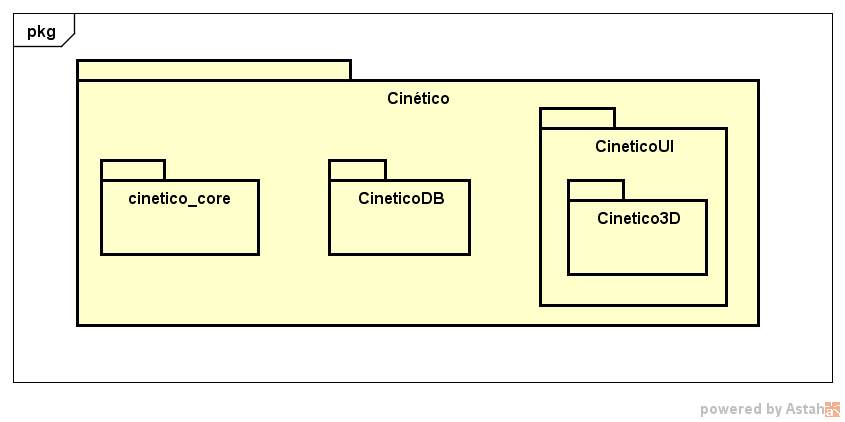
Fonte: autor (2017)

Para realização do software Cinético, foi necessário desenvolver um conjunto de operações relacionadas ao negócio, outro conjunto relacionado à interface gráfica para que seja possível uma apresentação visual ao usuário e uma camada contendo operações para gravar e recuperar dados. Esses conjuntos de operações são subsistemas que têm estruturas e operam de forma bem diferente comparados entre si. Então, o Cinético é separado em três módulos:

* cinetico\_core – faz o gerenciamento de todas as operações e recursos relacionados ao negócio;
* CineticoUI – faz o gerenciamento de toda a interface gráfica. Contém um subsistema chamado Cinetico3D para a apresentação de gráficos 3D;
* CineticoDB – faz todo o gerenciamento da persistência de dados e oferece acesso às classes DAO.

A Figura 10 mostra esta separação em módulos do Cinético.

Figura 10 - Estrutura Cinético



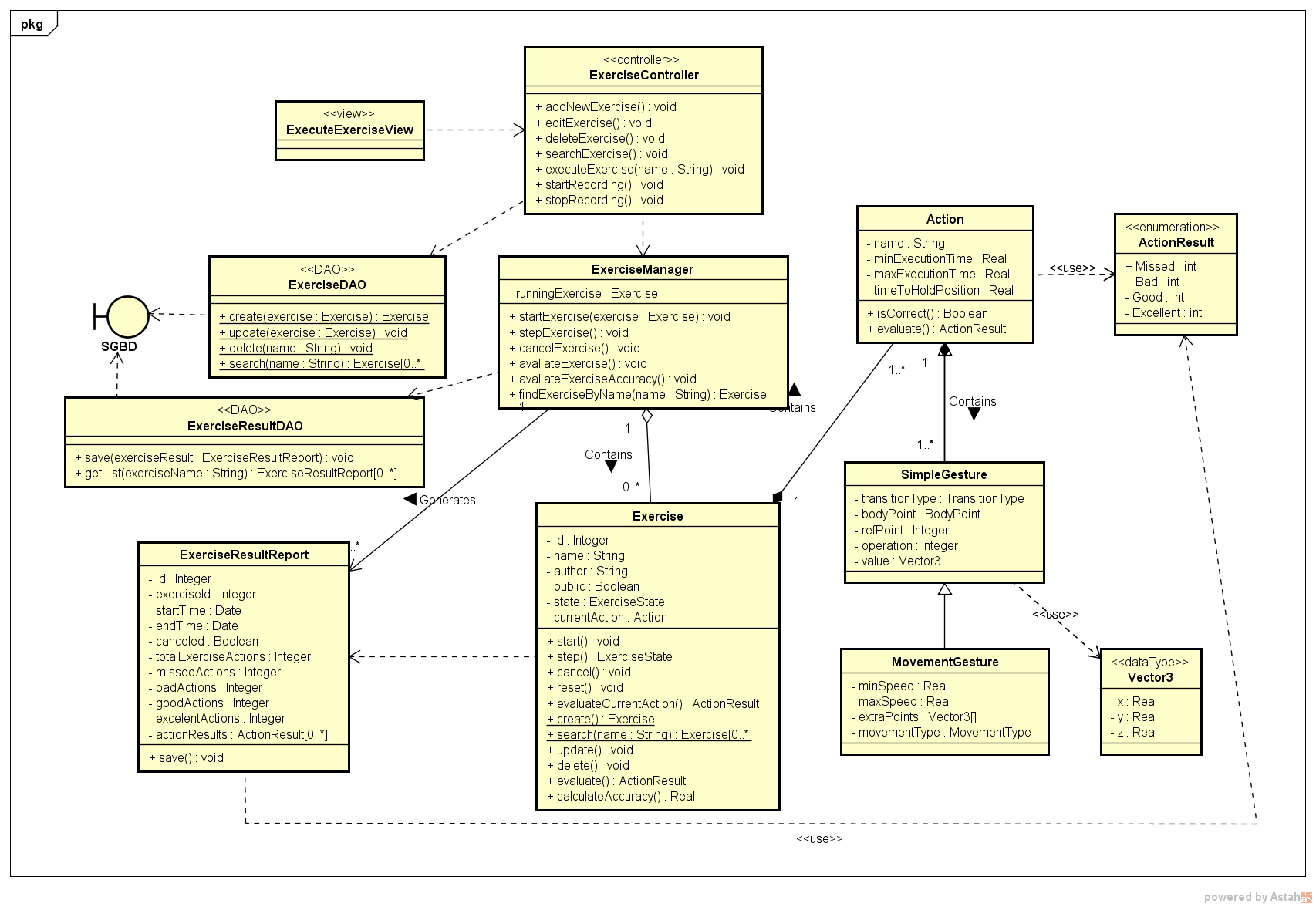
Fonte: Autor (2017)

* + 1. ***Realização dos Casos de Uso***

Neste item serão apresentados os diagramas de Classe de Projeto e de Sequência que compõe a realização de um Caso de Uso.

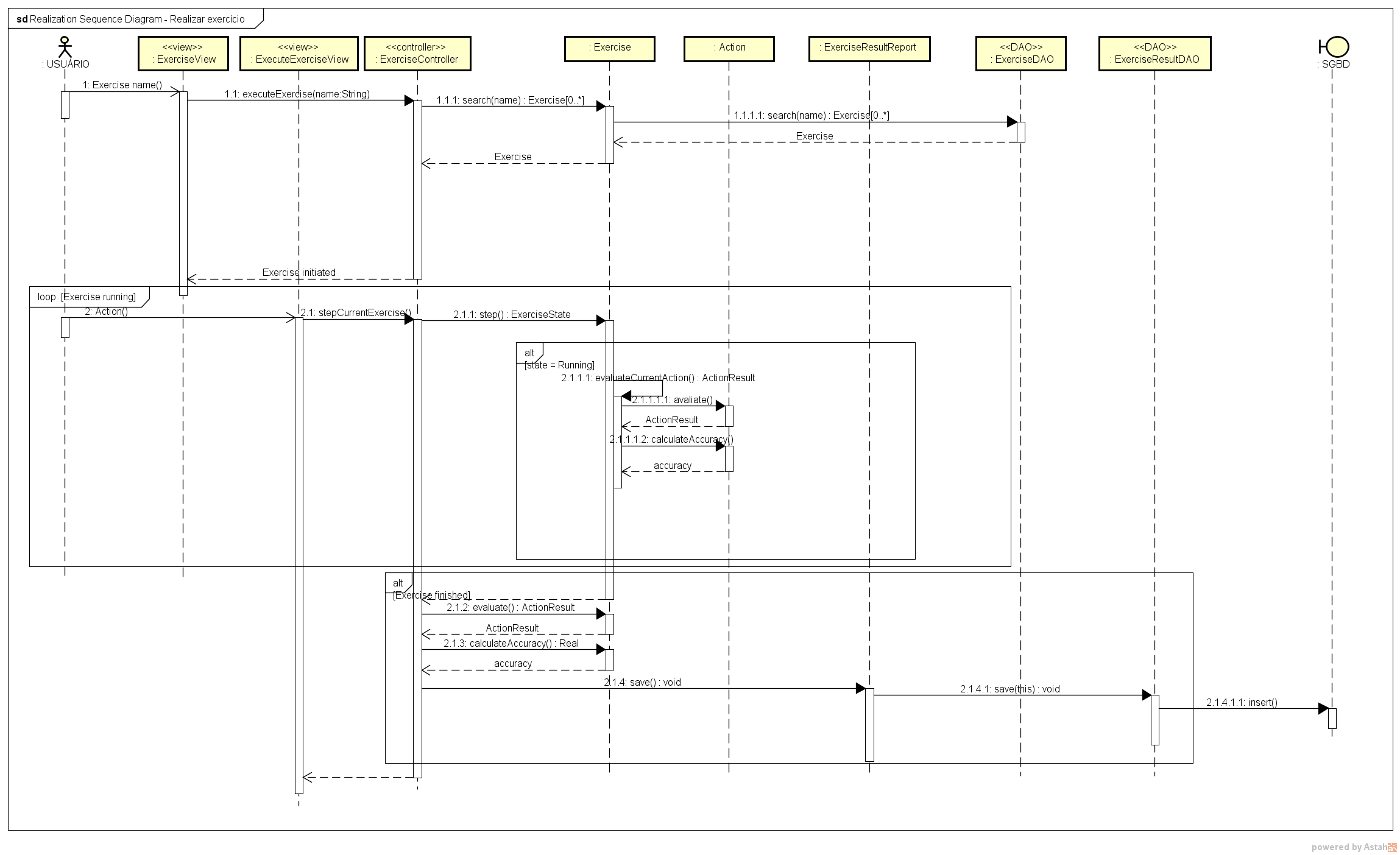
* + - 1. ***Realização de Caso de uso CSU004 – Realizar exercício***

Figura 11 - Diagrama de Classes – Realizar exercício

******

Fonte: Autor (2017)

Figura 12 - Diagrama de Sequência – Realizar exercício

******

Fonte: Autor (2017)

* + - 1. ***Realização de Caso de uso CSU006 – Gravar exercício***

Figura 13 - Diagrama de Classes – Gravar exercício

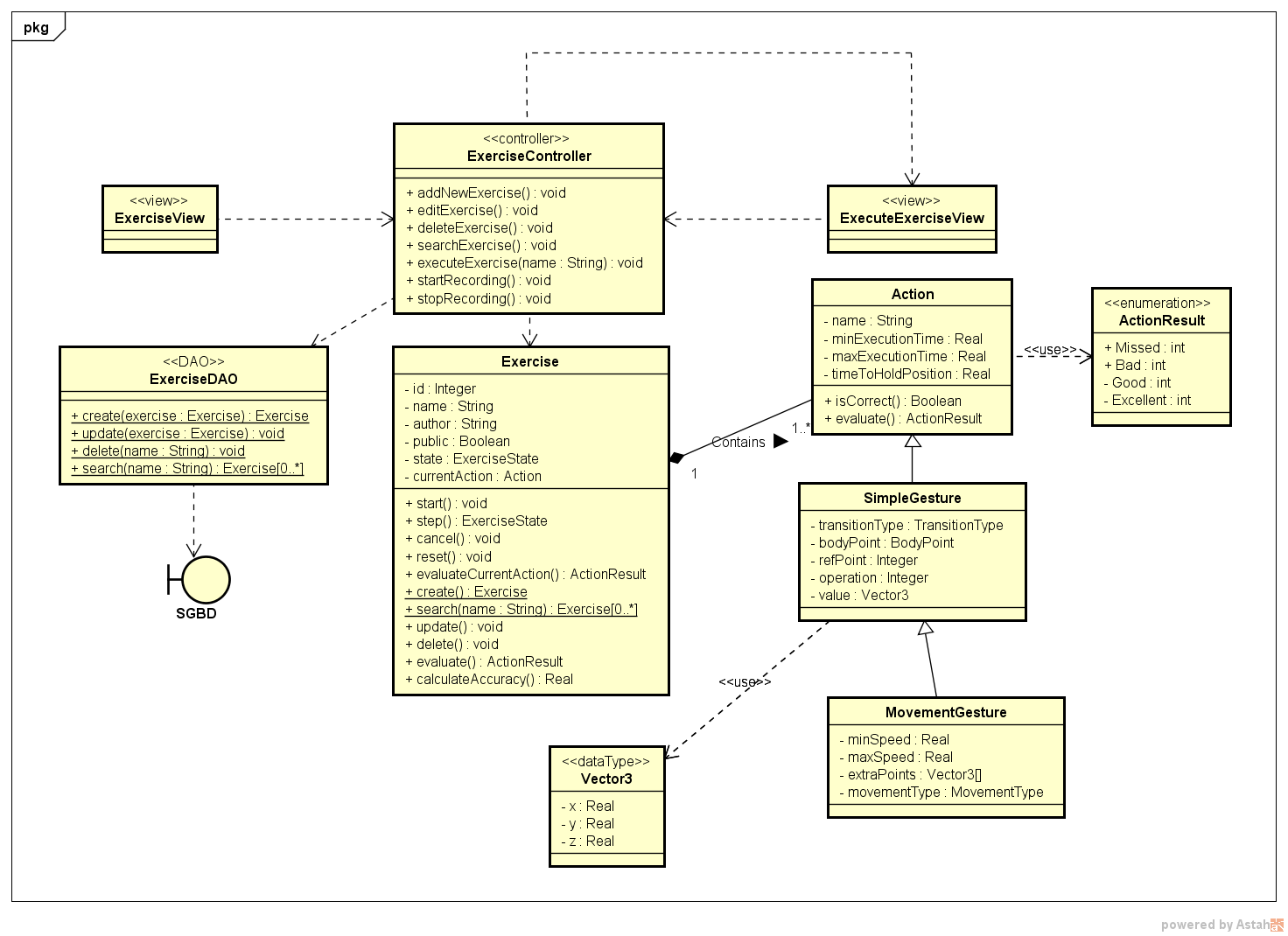
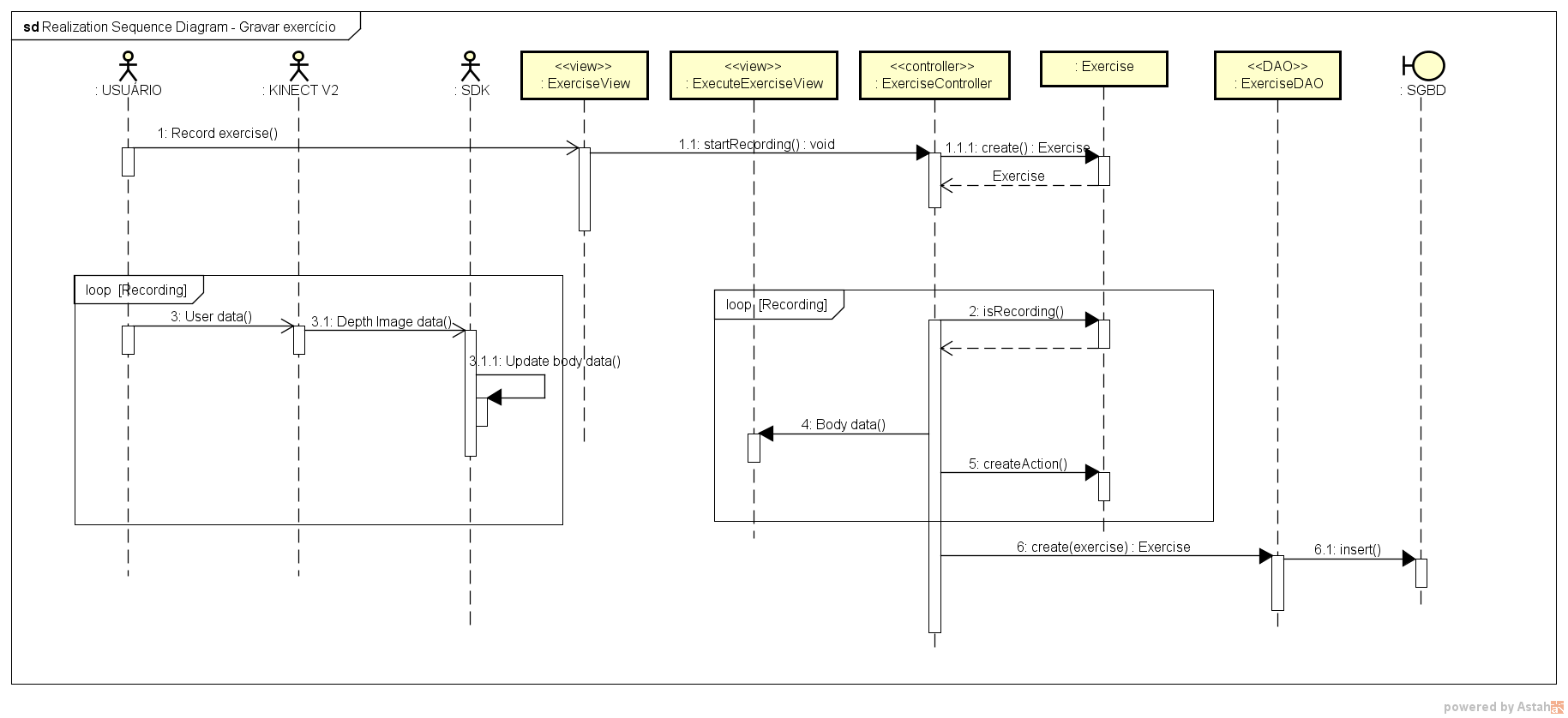
****** Fonte: Autor (2017)

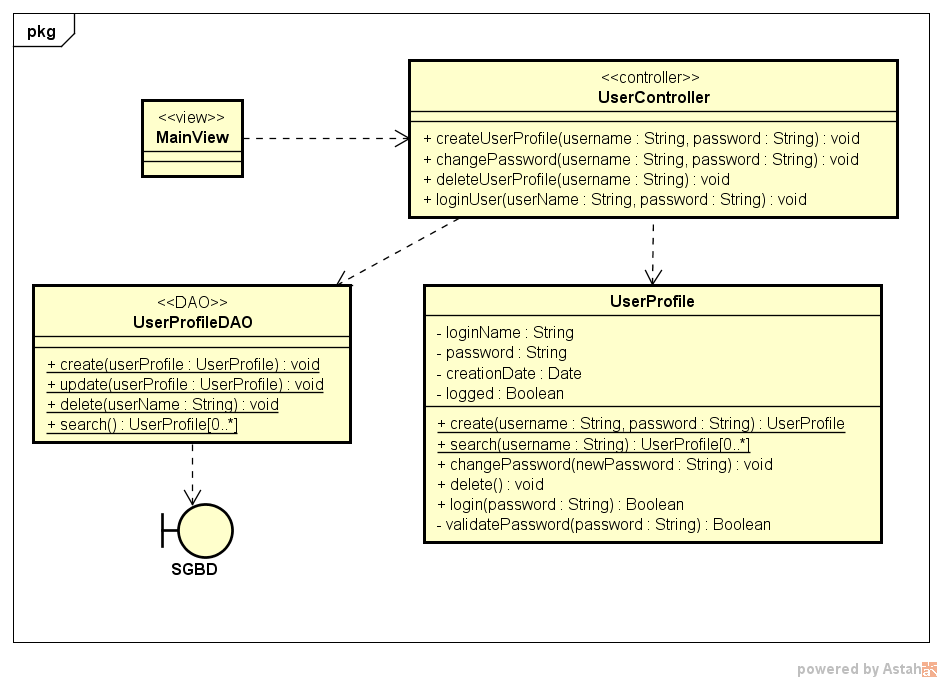
Figura 14 - Diagrama de Sequência – Gravar exercício



Fonte: Autor (2017)

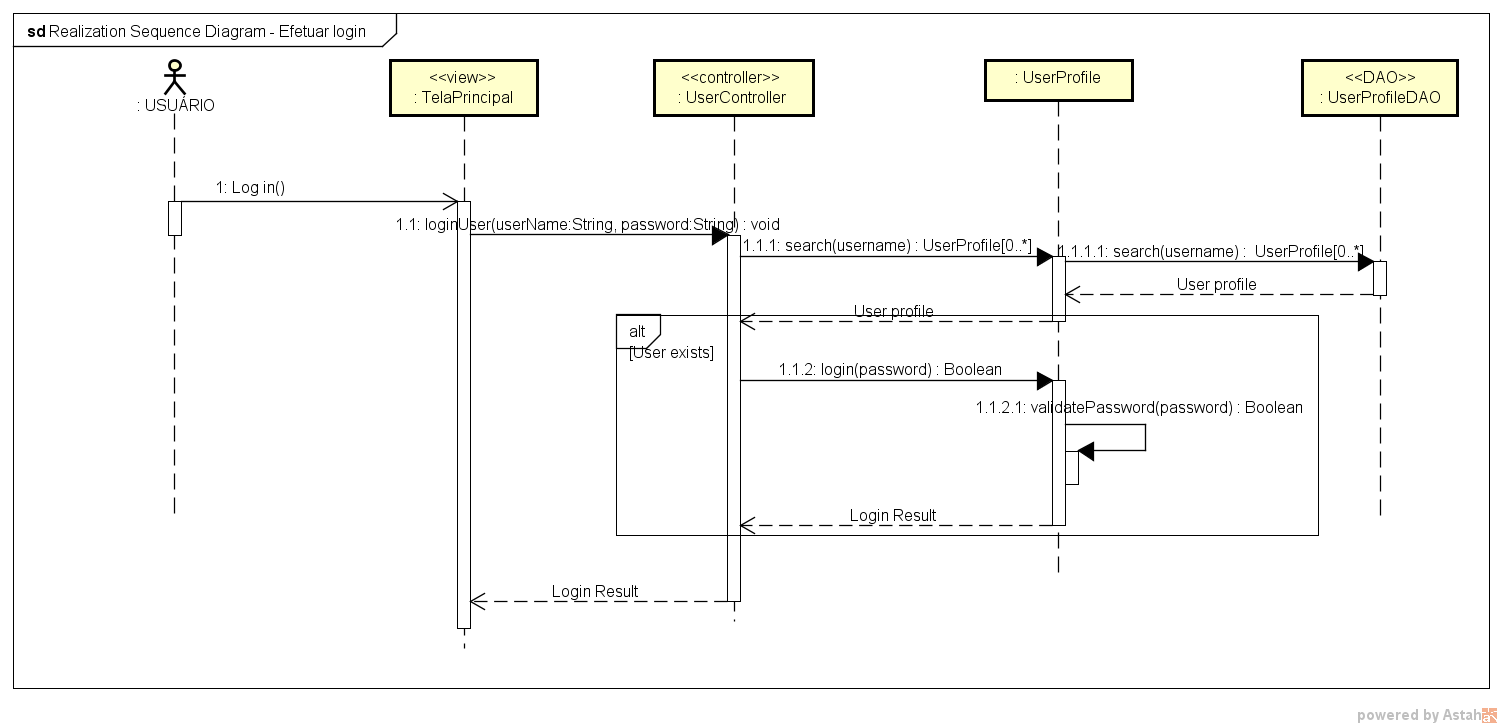
* + - 1. ***Realização de Caso de uso CSU008 – Efetuar Login***

Figura 15 - Diagrama de Classes – Efetuar Login

******

Fonte: Autor (2017)

Figura 16 - Diagrama de Sequência – Efetuar login



Fonte: Autor (2017)

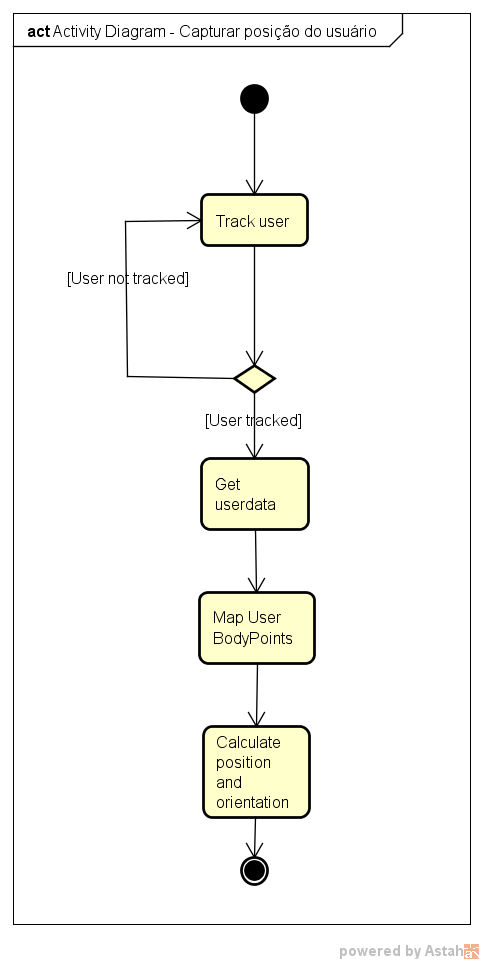
* + 1. ***Diagrama de Atividades***

Os Diagramas de Atividades representam o detalhamento de tarefas e o fluxo de uma atividade para outra de um sistema.

* + - 1. ***Diagrama de Atividades – Capturar posição do usuário***

A Figura 17 mostra o fluxo das atividades para a captura de posição do usuário.

Figura 17 - Diagrama de Atividades – Capturar posição do usuário

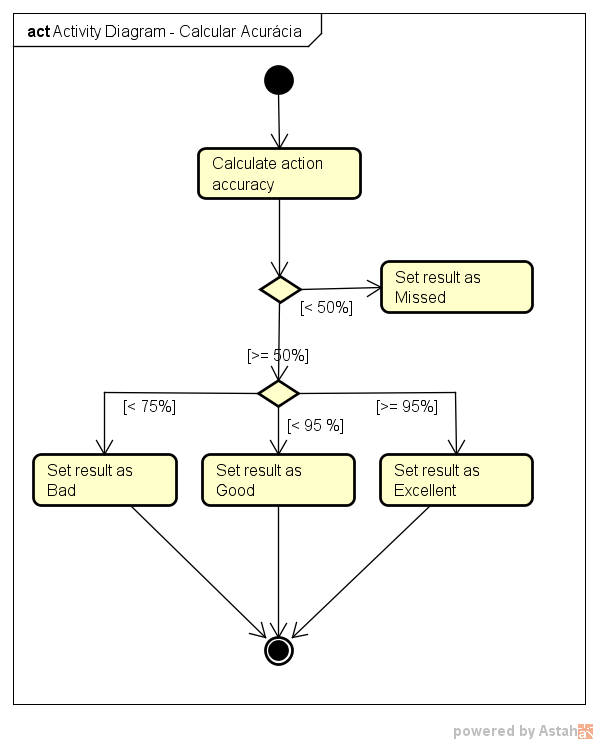
******

Fonte: Autor (2017)

* + - 1. ***Diagrama de Atividades – Calcular Acurácia***

A Figura 18 mostra o fluxo das atividades para o cálculo de acurácia de uma ação.

Figura 18 - Diagrama de Atividades – Calcular Acurácia

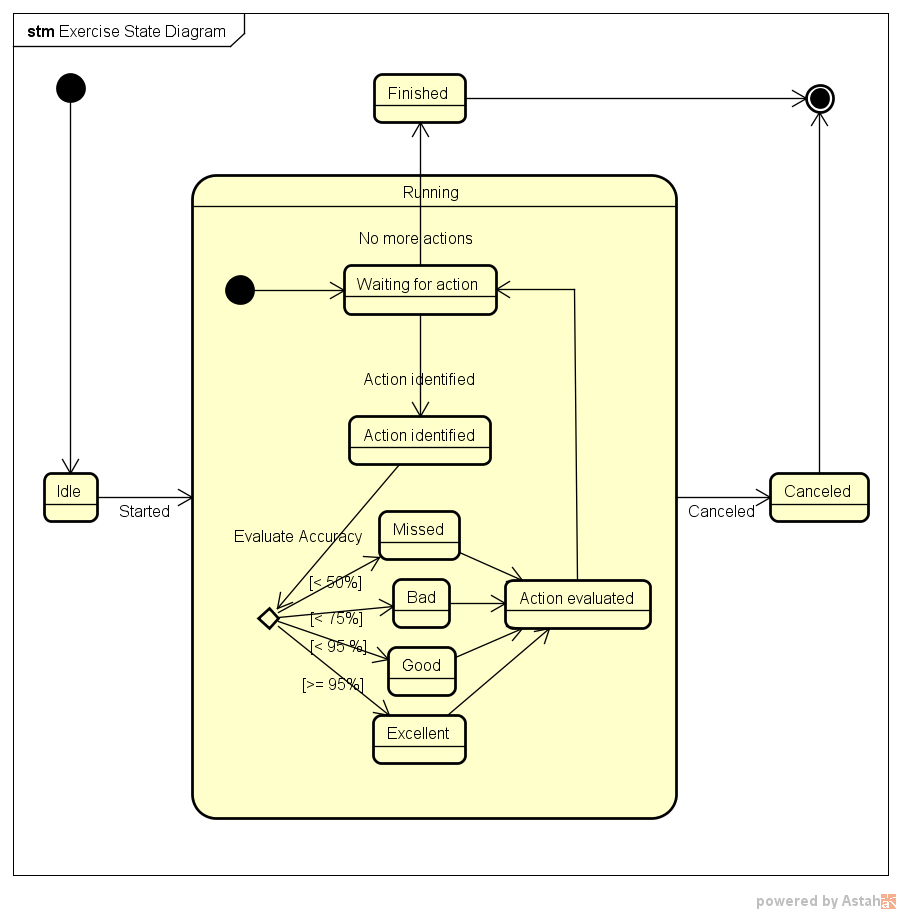
******

Fonte: Autor (2017)

* + 1. ***Diagrama de Estados***

O Diagrama de Estados especifica as sequências de estados pelas quais o objeto pode passar durante seu ciclo de vida em resposta a eventos. A Figura 19 mostra o ciclo de vida de um Exercício.

Figura 19 - Diagrama de Estados – Exercício



Fonte: Autor (2017)

* + 1. ***Operações importantes***

Nos diagramas de classe de negócio (Figura todo) e de projeto (Figura todo), o método evaluate da entidade Action são os responsáveis por definir se os gestos executados pelo usuário estão de acordo com os gestos esperados pelas ações do exercício.

O método evaluate é responsável por executar a operação matemática associada aos gestos esperados pela ação, calcular a acurácia e acordo com a porcentagem de gestos executados corretamente, e atualizar o resultado da ação de acordo com a porcentagem obtida. O método retorna verdadeiro se todos os gestos foram avaliados, falso do contrário.

Há 11 operações pré-definidas no Cinético. Cada operação usa o BodyPoint (bp) associado à Gesture e com exceção da operação de “Posição fixa”, o refPoint (ref) também é usado. Também podem ser atribuídos três parâmetros (px, py e pz) ao se criar um gesto para causar um deslocamento nos valores das coordenadas que estão sendo comparadas.

Todas as funções adicionam uma folga de espaço para melhor experiência do usuário, dado que no mundo real é praticamente impossível manter-se na mesma posição exatamente. Esta folga tem como valor padrão de 0,6 cm podendo ser alteradas nas configurações gerais do aplicativo.

O sistema de coordenadas segue o mesmo do Kinect, sendo:

Esquerda: -x, Direita: +x, Cima: +y, Baixo: -y, Frente: -z, Traz: +

As operações de comparação pré-programadas do Cinético são:

|  |  |
| --- | --- |
| **Operação** | **Operação executada** |
| Posição fixa | se bp.x = px, bp.y = py, bp.z = pz |
| À frente | bp.z < ref.z + pz |
| Atrás | bp.position.z > ref.position.z + pz |
| À direita | bp.position.x > ref.position.x + px |
| À esquerda | bp.position.x < ref.position.x + px |
| Acima | bp.position.y > ref.position.y + py |
| Abaixo | bp.position.y < ref.position.y + py |
| À mesma posição horizontal | bp.position.x = ref.position.x |
| À mesma altura | bp.position.y = ref.position.y |
| À mesma profundidade | bp.position.z = ref.position.z |
| Produto escalar | O produto escalar faz a seguinte operação: dot(bp.lookingToVec,ref.lookingToVec); útil para verificar se dois pontos do corop estão “olhando” para o mesmo sentido ou sentidos opostos. |

Uma função pode ser extendida ao criar duas Gestures usando o mesmo BodyPoint para uma mesma ação. Por exemplo: para se criar a ação “não mover lateralmente mais que 3 cm a mão direita em relação ao centro do corpo”, pode-se utilizar a seguinte abordagem:

* **Primeira Gesture:**

BodyPoint: Mão direita

Operation: Menor que

RefPoint: Espinha

Param: 3 cm

* **Segunda Gesture:**

BodyPoint: Mão direita

Operation: Maior que

RefPoint: Espinha

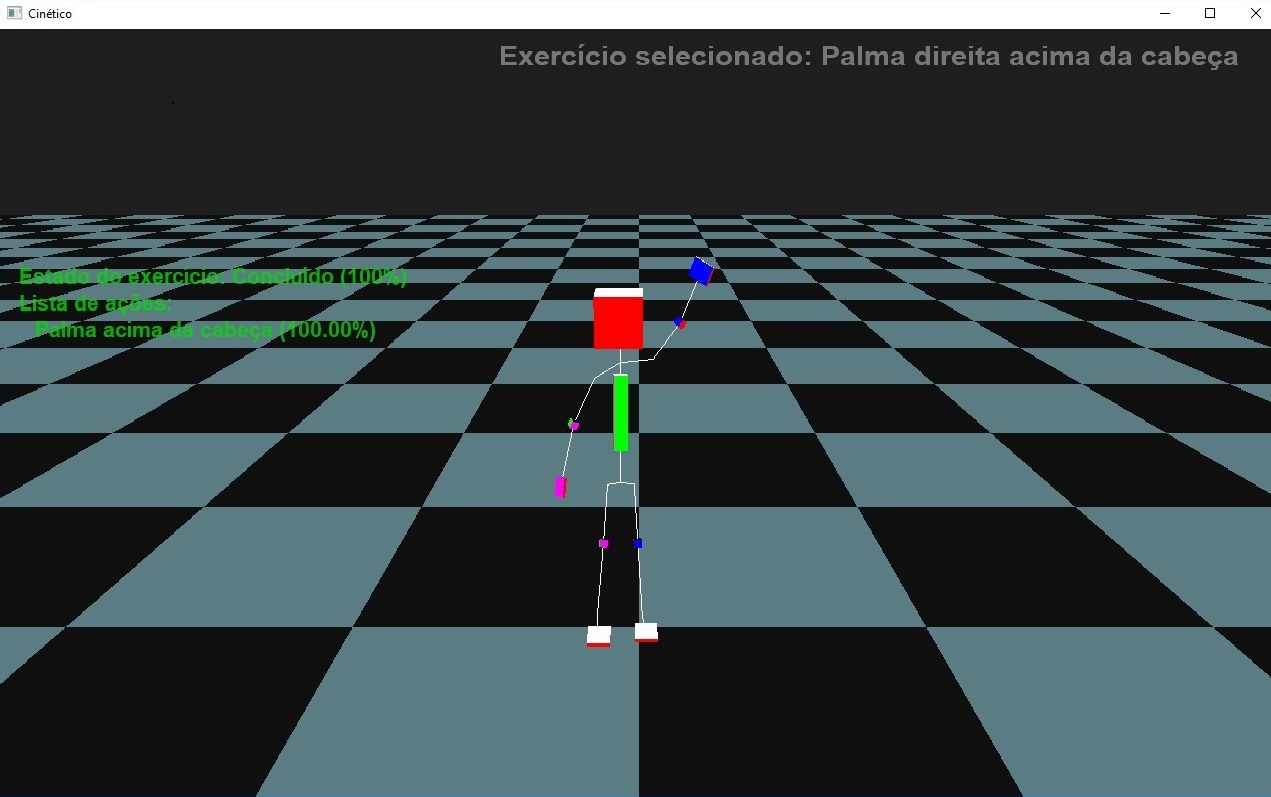
Param: -3 cm

1. **RESULTADOS OBTIDOS**

O Microsoft Kinect v2 se mostrou uma ferramenta poderosa, mostrando uma boa precisão enquanto o usuário estiver em uma posição moderadamente afastada do sensor. Com o apoio do sensor, o Cinético mostrou uma boa capacidade de documentar e avaliar exercícios de simples a moderadamente complexos.

Pode-se observar na Figura 20 um movimento simples, sendo a palma direita (espelhado na figura) acima da cabeça.

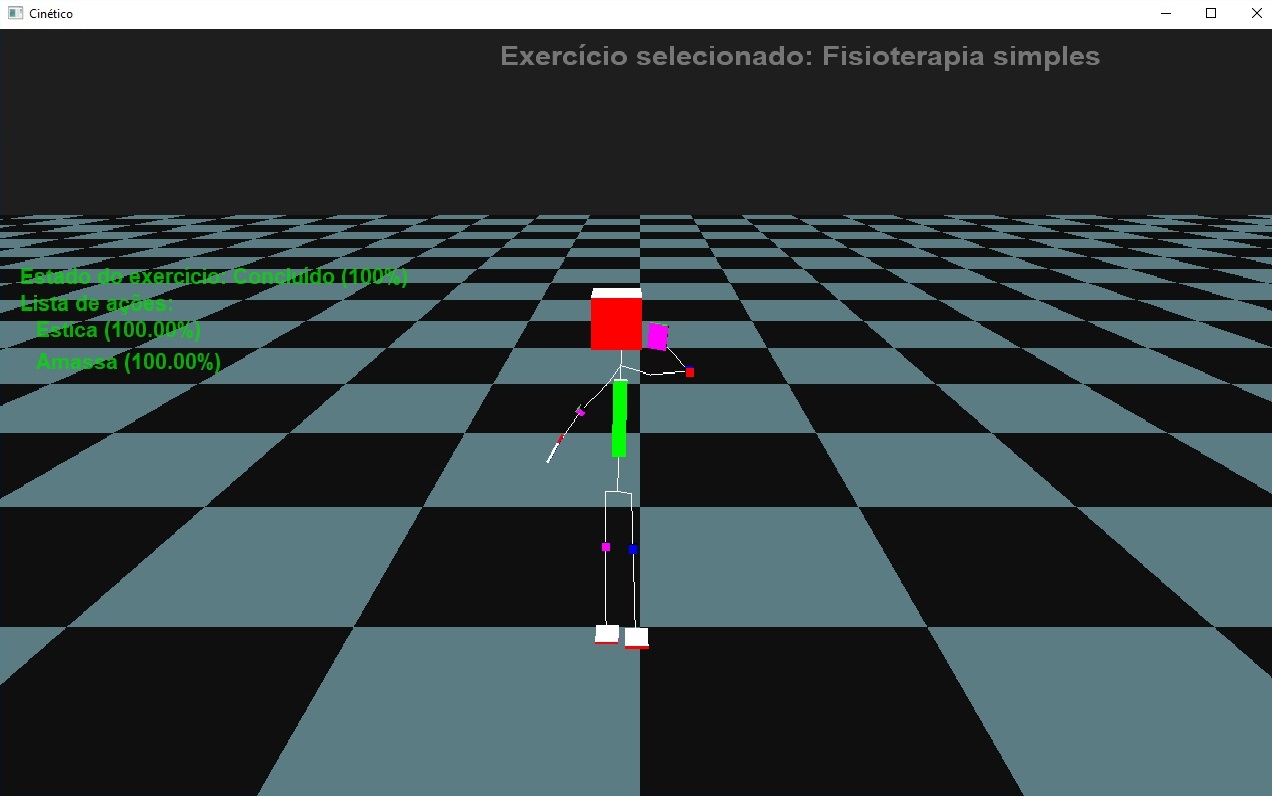
Figura 20 - Exercício Palma direita acima da cabeça

****

Fonte: Autor (2017)

Na Figura 21 temos um exercício um pouco mais elaborado, possuindo duas ações. Denominado “Fisioterapia simples”, o usuário deve esticar totalmente o braço, mantendo-o na mesma linha horizontal com ombro, e flexioná-lo até o pulso estiver na mesma linha vertical com ombro.

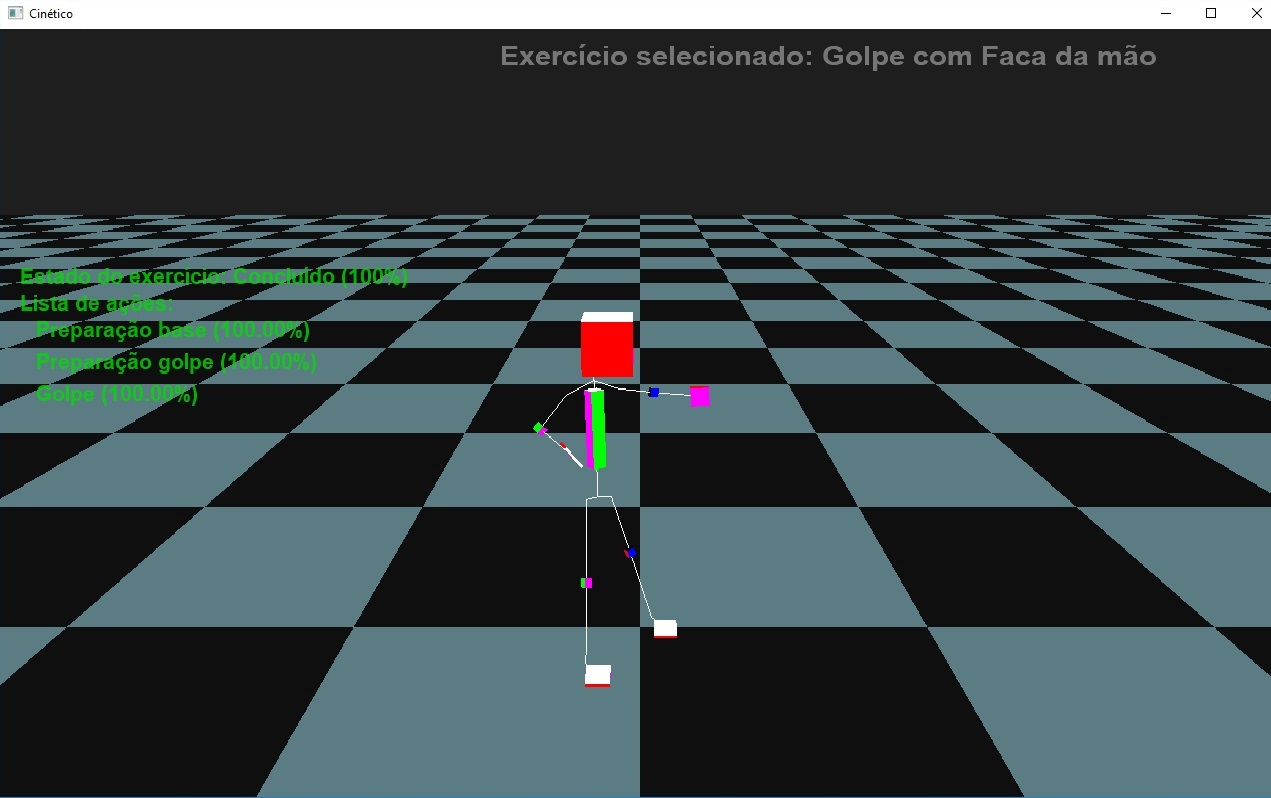
Figura 21 - Exercício Fisioterapia simples



Fonte: Autor (2017)

O Cinético também se mostrou capaz de avaliar exercícios mais elaborados. Na Figura 22 está demonstrado um golpe de artes marciais utilizando a “faca” da mão. O exercício possui três ações, onde a primeira deve-se manter os dois pés na mesma linha horizontal, a segunda deve-se elevar a mão direita à frente da região torácica e finalmente esticar a mão tal que fique na mesma linha horizontal do ombro.

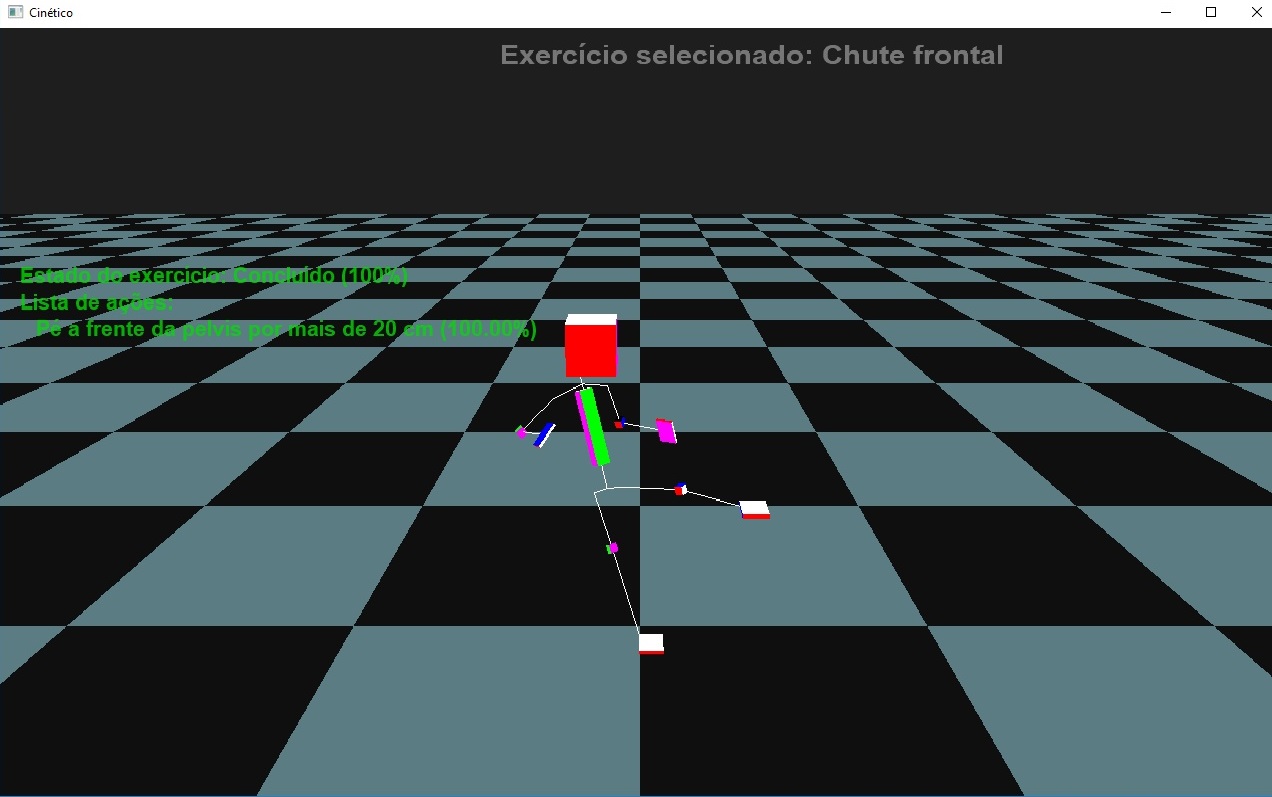
Figura 22 - Exercício Golpe com Faca da mão



Fonte: Autor (2017)

Agora, na Figura 23 pode-se ver que os pés também são bem capturados e processados. Pode-se ver um chute frontal sendo detectado.

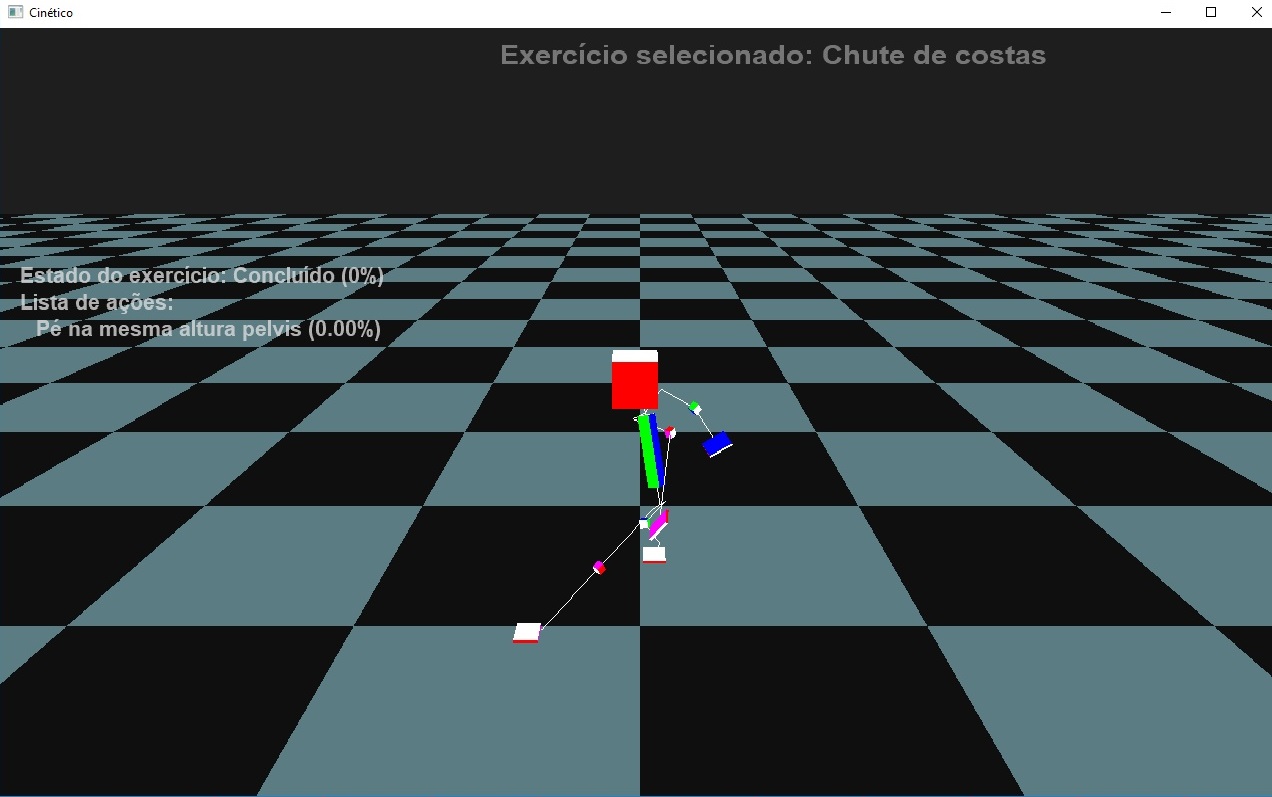
Figura 23 - Exercício Chute frontal



Fonte: Autor (2017)

O Kinect mostra-se bem eficaz quando se trata de movimentos mais simples, mas para movimentos mais complexos onde é necessário virar de costas, por exemplo, o dispositivo já não se mostrou eficaz, impossibilitando a avaliação de um exercício pelo Cinético. Pode-se perceber na Figura 24 que o *avatar* fica distorcido.

Figura 24 - Exercício Chute de costas



Fonte: Autor (2017)

1. **CONSIDERAÇÕES FINAIS**

O intuito deste trabalho foi de criar uma ferramenta poderosa que pudesse identificar e avaliar com um bom grau de precisão os movimentos executados em exercícios físicos, para poder ser aplicado nas áreas de Musculação, Fisioterapia, Artes Marciais, Dança, entre outras. Pode-se dizer que um *proof-of-concept* foi atingido com sucesso, pois foi possível identificar muitos gestos de complexidade simples a média. Entretanto, pela dificuldade do uso do Kinect, que não é uma ferramenta familiar, somado aos limites da tecnologia e sua documentação rasa, o escopo do projeto foi diminuído, limitando a avaliação de exercícios a apenas aqueles que podem ser construídos por um conjunto de gestos simples e estando de frente para o sensor. Para piorar a situação, em 2017 o Kinect v2 foi oficialmente descontinuado pela Microsoft[[1]](#footnote-1), diminuindo as esperanças de uma atualização e modernização de suas funcionalidades e melhoria na documentação do dispositivo e seu *SDK*.

Aliado a isto, ainda há um custo relativamente alto de implantação de todo o sistema, pois o Kinect v2 necessita de um adaptador para se conectar ao computador, que além da difícil obtenção por um preço “em conta”, necessita de um computador relativamente potente para funcionar bem. O dispositivo Kinect v2 tem um custo médio de R$ 300,00 e seu adaptador pelo mesmo preço. Há, no entanto, a opção de importar o adaptador, podendo obtê-lo por volta de U$ 30,00 até o momento da realização deste trabalho.

De qualquer forma, é pretendida a continuidade do projeto, e tornar o cinetico\_core uma biblioteca de uso geral, para poder ser usada nos mais variáveis tipos de software, como um controlador remoto ou em games, por exemplo. Há também a ambição de cada vez mais se desprender da geração do esqueleto provida pelo K4W e se aprofundar nos conceitos de *Computer Vision* (*CV*) e Inteligência artificial para criar uma implementação própria, usando o Kinect somente para a obtenção da *Depth Image*.

É previsto também uma forma intuitiva de se criar os movimentos (Caso de uso Gravar Exercício), tendo em vista a dificuldade que é definir um movimento via seleção e definição de valores por meio de campos usuais. Por fim, é previsto também, descontinuar o uso do Windows SDK para interface gráfica e começar a usar somente o ambiente 3D para navegação nas telas, criação e realização dos exercícios.

# REFERÊNCIAS

ALMEIDA, Filipe Barbosa de. **Sistema interativo baseado em gestos para utilização de comandos no computador**. Trabalho de Conclusão de Curso (Bacharel em Engenharia de Software) - Universidade de Brasília, Brasília, 2013.

BORENSTEIN, Greg. ***Making Things See***. Maker Media, Sebastopol, CA, 2012. p. 1.

CAMPOS, Guilherme Pires. **Sistema para ﬁsioterapia baseado na plataforma Kinect**. Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores) - Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2013.

KOSTRIUBA, Henrique et al. **Kinect para Reabilitação**. Trabalho de Conclusão de Curso (Tecnólogo em Jogos Digitais) - Faculdade de Tecnologia de Carapicuíba, São Paulo, 2014.

MACHADO, Ricardo Nagel. **Proposta de sistema baseado na plataforma Kinect para suporte a reabilitação de pacientes com patologias ligamentares nos joelhos**. Trabalho de Conclusão de Curso

(Bacharelado em Tecnologia da Informação) - Universidade Federal de Santa Catarina, Araranguá, 2013.

MICROSOFT, ***Body Tracking***. 2013. Disponível em: [<](https://msdn.microsoft.com/en-us/library/jj131025.aspx)https://msdn.microsoft.com/en-us/library/dn799273.aspx>. Acesso em: 12 jun 2017.

MICROSOFT, ***Direct3D Devices (Direct3D 9)*.** Disponível em: <https://msdn.microsoft.com/pt-br/library/windows/desktop/bb219680(v=vs.85).aspx> Acesso em: 04 dez 2017.

MICROSOFT, ***Hardware Kinect***. 2013. Disponível em: <<https://developer.microsoft.com/pt-br/windows/kinect/hardware>>. Acesso em: 11 jun 2017.

MICROSOFT, **Windows GDI,** Disponível em: <https://msdn.microsoft.com/pt-br/library/windows/desktop/dd145203(v=vs.85).aspx> Acesso em: 04 dez 2017.

SQLITE, ***About SQLite***. Disponível em:<https://www.sqlite.org/about.html> Acesso em: 05 dez 2017.

# GLOSSÁRIO

**A**

**Ação:** um conjunto de gestos, executados ao mesmo tempo.

**Action**: entidade interna do sistema que representa uma ação.

***Aspect ratio*:** Proporção entre as dimensões de uma tela.

***Avatar*:** Representação gráfica virtual para identificação de usuário.

**C**

***Computer Vision*:** Termo utilizado para designar a interação entre o computador e o mundo real, executando uma ação ou gerando uma saída de dados sem a interferência humana.

**Controle:** qualquer componente gráfico que desempenha uma função. Exemplos: botão; caixa de texto.

***DAO (Data Access Object):*** Padrão de projeto que visa separar as classes do sistema de software que contém código para persistência de dados das classes que representam o negócio em si.

**E**

**Exercício:** um conjunto de ações, executadas em ordem.

**Exercise:** entidade interna do sistema que representa um exercício.

**G**

**Gesto:** posicionamento ou movimentação de algum ponto do corpo.

**Gesture**: entidade interna do sistema que representa um gesto.

***GDI (Graphics Device Interface):*** Interface de abstração do Windows para acesso aos drivers da GPU. Usado par desenhar diretamente nas janelas das aplicações.

***GPU (Graphics Processing Unit):*** dispositivo processador dedicado a processamento de cálculos matemáticos voltados para computação gráfica.

**P**

**Persistência de dados:** Termo utilizado geralmente no contexto de Banco de Dados que detona a gravação e recuperação de dados.

**Ponto do corpo:** ponto do corpo identificável por um sensor de movimentos. Pode ser um arcabouço de ossos ou uma articulação. Exemplos: mão; joelho.

***Proof-of-concept*:** Realização de conceito ou ideia.

**S**

**SGBD (Sistema Gerenciador de Banco de Dados):** software responsável por manter uma estrutura de dados em disco ou memória RAM para gravação de dados, sendo possível fazer operações de inclusão, edição, busca e exclusão destes dados através de uma linguagem.

***SDK (Software Development Kit)*:** Conjunto de ferramentas, bibliotecas e aplicativos que permitem a criação de um software ou componente para determinado sistema de software.

***SQL (Structured Query Language):*** Linguagem padrão universal designada para fazer consultas em banco de dados do tipo relacional.

**V**

***Viewport*:** A área de uma tela que o usuário pode enxergar.

# APÊNDICE A – PLANO E EXECUÇÃO DE TESTES

Neste capítulo o objetivo é validar as funções do sistema, verificando se os requisitos foram implementados de forma. A abordagem utilizada para a validação do sistema, foi o esquema de priorização de requisitos funcionais, e o critério de aceitação foi a execução dos casos de testes para todos os fluxos básicos dos requisitos funcionais.

Plano de Testes

Neste trabalho, foi utilizado a técnica de priorização de requisitos, para definir a prioridade dos casos de testes. A tabela 1 demonstra o perfil operacional de uso, com base nos atores do sistema.

Tabela 1 - Perfil Operacional de uso

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ator | Perfil de uso (%) | | | | |
| CSU004 | CSU005 | CSU006 | CSU007 | CSU008 |
| USUÁRIO | 75 | 5 | 5 | 5 | 10 |

Fonte: Autor (2017)

Com base no perfil operacional de uso, foi elaborado a priorização dos requisitos, conforme tabela 2, que demonstra o coeficiente de impacto em caso de falha dos requisitos funcionais.

Tabela 2 – Probabilidade de falha x Impacto

|  |  |  |  |
| --- | --- | --- | --- |
| Caso de Uso | Probabilidade de falha | Impacto | Coeficiente |
| CSU004 | 3 | 3 | 9 |
| CSU005 | 2 | 2 | 4 |
| CSU006 | 3 | 2 | 6 |
| CSU007 | 1 | 2 | 3 |
| CSU008 | 1 | 1 | 1 |

Fonte: Autor (2017)

Com base no cálculo do coeficiente de impacto (calculado pela formula probabilidade de falha multiplicado pelo impacto), foi elaborada a tabela com a priorização dos requisitos, conforme demonstrado na tabela 3.

Tabela 3 – Prioridade dos Casos de Uso

|  |  |
| --- | --- |
| Caso de Uso | Prioridade |
| CSU004 – Realizar exercício | Alta |
| CSU005 – Manter exercício | Importante |
| CSU006 – Gravar exercício | Alta |
| CSU007 – Manter perfil | Baixa |
| CSU008 – Efetuar login | Baixa |

Fonte: Autor (2017)

Neste Plano de testes, foi elaborado casos de testes para todos os requisitos de prioridade Moderada, Importante e Alta.

Roteiro de Testes

Este item visa mostrar o resultado obtido a partir dos testes baseado na priorização dos casos de uso. Até o presente momento deste trabalho, os requisitos para o CSU006 – Gravar exercício não foram implementados, e não puderam ser testados.

|  |  |
| --- | --- |
| **ID** | CSU004\_RealizarExercício |
| **Objetivo** | Permite ao usuário criar e realizar um exercício |
| **Pré-Condição** | Usuário autenticado no sistema |
| **Data de realização** | 30/11/2017 |
| **Procedimentos** | **Resultados esperados** |
| Acessar a tela de exercícios | Tela de “Exercícios” exibida |
| Selecionar a opção “Gerenciamento de Ações” | Tela de “Gerenciamento de Ações” exibida |
| Selecionar a opção “Adicionar ação” | Campos para adição de ação exibidos |
| Selecionar a opção “Adicionar” na seção Gestos e criar um gesto. Repetir para outros 2 gestos diferentes. | Gestos criados |
| Acionar a opção “Voltar”. | Campos escondidos. |
| Selecionar a opção “Voltar” | Usuário redirecionado para a tela de “Exercícios” |
| Selecionar a opção “Realizar exercício” | Usuário redirecionado para a tela de “Realização de Exercício” |
| Testar as condições de cada ação escolhidas na criação do exercício | Sistema valida ação |
| **RESULTADO DO TESTE** | SUCESSO. |

|  |  |
| --- | --- |
| **ID** | CSU004CT02\_ExecutarExercícioZeroAções |
| **Objetivo** | Não permitir o usuário executar um exercício sem ações definidas |
| **Pré-Condição** | Usuário autenticado no sistema. |
| **Data de realização** | 08/06/2017 |
| **Procedimentos** | **Resultados esperados** |
| Acessar tela de Exercícios | Tela de "Exercícios" exibida |
| Selecionar um Exercício com 0 ações na lista |  |
| Executar o exercício | Mensagem exibida: "Um exercício deve conter pelo menos uma Ação" |
| Selecionar a opção "Meus chamados" |  |
| **RESULTADO DO TESTE** | SUCESSO. |

|  |  |
| --- | --- |
| **ID** | CSU005CT01\_CriarExercício |
| **Objetivo** | Permite ao usuário criar um exercício. |
| **Pré-Condição** | Usuário autenticado no sistema. |
| **Data de realização** | 08/06/2017 |
| **Procedimentos** | **Resultados esperados** |
| Acessar tela de Exercícios | Tela de "Exercícios" exibida |
| Selecionar a opção "Meus chamados" | Exibe tela com todos os chamados abertos pelo usuário |
| Ativar a opção "Criar exercício" | Redirecionamento para a tela de "Gerenciamento de exercícios" |
| Preencher campo nome e ativar a opção criar | Lista de ações exibida (inicialmente vazia) |
| Selecionar tipo de ação | Campos específicos para o tipo de ação exibidos |
| Preencher campos específicos para o tipo de ação e ativar a opção "Salvar" | Ação salva para o exercício atual |
| Para salvar mais ações, voltar ao passo 1.4; Para criar o exercício, ativar a opção "Salvar" | Exercício criado e um id atribuído a ele |
| RESULTADO DO TESTE | SUCESSO |

|  |  |
| --- | --- |
| **ID** | CSU005CT02\_EditarExercício |
| **Objetivo** | Permitir ao usuário editar um exercício criado. |
| **Pré-Condição** | Usuário autenticado no sistema;  Exercício criado. |
| **Data de realização** | 08/06/2017 |
| **Procedimentos** | **Resultados esperados** |
| Acessar tela de Exercícios | Exibe opções adicionais |
| Selecionar um Exercício existente |  |
| Ativar a opção de "Editar" | Redirecionamento para a tela de "Gerenciamento de exercícios" |
| Alterar campo nome | Campo nome alterado |
| Ativar opção "Editar ação" | Campos referentes à ação e ao tipo de ação exibidos |
| Alterar campos da ação e ativar a opção "Salvar" do escopo da Ação | Ação alterada |
| Ativar opção "Salvar" do escopo do Exercício | Exercício alterado |
| RESULTADO DO TESTE | SUCESSO |

|  |  |
| --- | --- |
| **ID** | CSU005CT03\_ExcluirExercício |
| **Objetivo** | Permite ao usuário excluir um exercício criado. |
| **Pré-Condição** | Usuário identificado no sistema. |
| **Data de realização** | 08/06/2017 |
| **Procedimentos** | **Resultados esperados** |
| Acessar tela de Exercícios | Tela de "Exercícios" exibida |
| Selecionar Exercício existente |  |
| Ativar a opção "Excluir" | Exercício excluído |
| RESULTADO DO TESTE | SUCESSO |

|  |  |
| --- | --- |
| **ID** | CSU005CT04\_ExercícioExistente |
| **Objetivo** | Bloquear a criação de um exercício de mesmo nome |
| **Pré-Condição** | O usuário deve estar identificado pelo sistema |
| **Data de realização** | 08/06/2017 |
| **Procedimentos** | **Resultados esperados** |
| Acessar tela de Exercícios | Tela de "Exercícios" exibida |
| Criar ou editar um Exercício | Redirecionamento para a tela de "Gerenciamento de exercícios" |
| Alterar campo nome do Exercício para um nome de outro exercício já existente no sistema | Mensagem exibida: "Já existe um exercício de mesmo nome" |
| RESULTADO DO TESTE | SUCESSO |

|  |  |
| --- | --- |
| **ID** | CSU004\_RealizarExercício |
| **Objetivo** | Permite ao usuário criar e realizar um exercício |
| **Pré-Condição** | Usuário autenticado no sistema |
| **Data de realização** | 30/11/2017 |
| **Procedimentos** | **Resultados esperados** |
| Acessar a tela de exercícios | Tela de “Exercícios” exibida |
| Selecionar a opção “Gerenciamento de Ações” | Tela de “Gerenciamento de Ações” exibida |
| Selecionar a opção “Adicionar ação” | Campos para adição de ação exibidos |
| Selecionar a opção “Adicionar” na seção Gestos e criar um gesto. Repetir para outros 2 gestos diferentes | Gestos criados |
| Acionar a opção “Voltar” | Usuário redirecionado para a tela de “Exercícios” |
| Selecionar a opção “Realizar exercício” | Usuário redirecionado para a tela de “Realização de Exercício” |
| Testar as condições de cada ação escolhida na criação do exercício | Sistema valida ação |
| RESULTADO DO TESTE | SUCESSO |

1. MICROSOFT, ***Body Tracking***. 2013. Disponível em: [<](https://msdn.microsoft.com/en-us/library/jj131025.aspx)https://support.xbox.com/en-US/xbox-on-windows/accessories/kinect-for-windows-v2-setup>. Acesso em: 20 fev 2018. [↑](#footnote-ref-1)