

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

LUCIANO HENRIQUE QUINTINO

**UM SISTEMA DE MONITORAMENTO PARA IDOSOS
INSTITUCIONALIZADOS UTILIZANDO O SENSOR KINECT**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2015

LUCIANO HENRIQUE QUINTINO

**UM SISTEMA DE MONITORAMENTO PARA IDOSOS
INSTITUCIONALIZADOS UTILIZANDO O SENSOR KINECT**

TRABALHO DE CONCLUSÃO DE CURSO apresentado como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Eduardo Filgueiras Damasceno

CORNÉLIO PROCÓPIO

2015

AGRADECIMENTOS

Agradeço primeiramente à Deus o qual me deu esta oportunidade e esteve ao meu lado por todo caminho me dando forças para superar as dificuldades.

A todos os professores e em especial ao meu orientador Prof. Dr. Eduardo Filgueiras Damasceno, pela sabedoria, paciência, dedicação e atenção que me guiou nesta trajetória.

A todos os colegas de curso e profissão com os quais convivi durante esta importante etapa de minha vida.

Agradeço principalmente a minha família e amigos por terem me apoiado e ficarem ao meu lado nas horas que eu mais precisava.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

RESUMO

QUINTINO, Luciano. UM SISTEMA DE MONITORAMENTO PARA IDOSOS INSTITUCIONALIZADOS UTILIZANDO O SENSOR KINECT. 54 f. TRABALHO DE CONCLUSÃO DE CURSO – CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

Este trabalho apresenta um sistema para monitorar idosos residentes em instituições de longa permanência utilizando visão computacional. O sistema consiste basicamente em monitorar e informar o cuidador que ocorreu uma situação de risco de queda, resguardando a privacidade do idoso. Será utilizada a tecnologia do dispositivo Kinect, seus recursos de rastreamento de esqueleto e imagem de profundidade.

Palavras-chave: Monitoramento, Alerta de quedas, Idoso institucionalizado, Sensor Kinect, Sistema supervisorio

ABSTRACT

QUINTINO, Luciano. A MONITORING SYSTEM FOR ELDERLY INSTITUTIONALIZED USING THE KINECT SENSOR. 54 f. TRABALHO DE CONCLUSÃO DE CURSO – CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

This paper presents a system for monitoring elderly residents in long term care facilities using computer vision. The system basically consists of monitor and inform the caregiver that there was a decrease of risk, maintaining the privacy of the elderly. Will be used the Kinect device technology, its skeleton tracking capabilities and depth of image.

Keywords: Monitoring, Warning falls, institutionalized elderly, Supervisory system

LISTA DE FIGURAS

FIGURA 1	– Curva de capacidade funcional ao longo do ciclo vital	10
FIGURA 2	– Utilização de extração de silhueta	15
FIGURA 3	– Utilização de dispositivos Near-Fied	16
FIGURA 4	– Hardware do Kinect	16
FIGURA 5	– As 20 articulações mapeadas pelo Kinect	17
FIGURA 6	– Espiral Win Win	20
FIGURA 7	– Arquitetura física do sistema	21
FIGURA 8	– Diagrama Arquitetura MVC	22
FIGURA 9	– Diagrama de caso de uso - Estação Remota	23
FIGURA 10	– Diagrama de caso de uso - Central de monitoramento	24
FIGURA 11	– UC0001 - Ajustar Câmera	24
FIGURA 12	– UC0002 - Alterar configurações	25
FIGURA 13	– UC0001 - Logar no sistema	25
FIGURA 14	– UC0002 - Consultar quartos	25
FIGURA 15	– UC0003 - Cadastrar quartos	26
FIGURA 16	– UC0004 - Editar quarto	26
FIGURA 17	– UC0005 - Deletar quarto	26
FIGURA 18	– UC0006 - Consultar usuários	27
FIGURA 19	– UC0007 - Cadastrar usuário	27
FIGURA 20	– UC0008 - Editar usuário	27
FIGURA 21	– UC0009 - Deletar usuário	28
FIGURA 22	– UC0010 - Visualizar relatórios	28
FIGURA 23	– UC0011 - Monitorar quartos	28
FIGURA 24	– UC0012 - Alterar dados cadastrais	29
FIGURA 25	– Fluxograma do monitor	29
FIGURA 26	– Modelo relacional do banco de dados	30
FIGURA 27	– Diagrama de classes - Central de Monitoramento	31
FIGURA 28	– Diagrama de classe - Estação Remota	32
FIGURA 29	– Diagrama de sequência - Estação Remota	33
FIGURA 30	– Diagrama de Sequencia - Monitoramento	34
FIGURA 31	– Tela de Login	35
FIGURA 32	– Tela Principal	35
FIGURA 33	– Tela de listagem de quartos	36
FIGURA 34	– Tela de cadastro de quartos	36
FIGURA 35	– Tela de exclusão de quarto	37
FIGURA 36	– Tela de monitoramento	37
FIGURA 37	– Relatório Analítico	38
FIGURA 38	– Relatório gráfico diário em barras	38
FIGURA 39	– Relatório gráfico mensal em linhas	39
FIGURA 40	– Relatório gráfico anual em barras	39
FIGURA 41	– Tela dados do usuário	40
FIGURA 42	– Tela menu central de monitoramento	40

FIGURA 43	– Tela de configurações	40
FIGURA 44	– Armazenamento e verificação dos dados	41
FIGURA 45	– Trecho de código - Análise do esqueleto	42
FIGURA 44	– Queda lateral em frente ao Kinect	43
FIGURA 45	– Queda em movimentação na frente do Kinect	44
FIGURA 46	– Pessoa deitada capturada pelo Kinect	45
FIGURA 47	– Captura de duas pessoas pelo sensor	46
FIGURA 48	– Cronograma	49

LISTA DE TABELAS

TABELA 1	– Métodos para detecção de quedas	14
TABELA 2	– Resultados - Quedas em frente ao Kinect	44
TABELA 3	– Resultados - Queda em movimentação na frente do Kinect	45
TABELA 4	– Resultados - Quedas ao deitar e levar da cama	46
TABELA 5	– Resultados - Pessoa caindo ao prestar socorro	47

LISTA DE SIGLAS

IBGE	Instituto Brasileiro de Geografia e Estatística
UML	<i>Unified Modeling Language</i>
AVD	Atividades da vida diária
NFI	<i>Near Field Imaging</i>
NUI	<i>Natural User Interface</i>
RGB	<i>Red, Green, Blue</i>
VGA	<i>Video Graphics Array</i>
3D	<i>Tree Dimensions</i>
SDK	<i>Software Development Kit</i>
MVC	<i>Model, View, Controller</i>
ORM	<i>Object, Relational, Mapping</i>
WPF	<i>Windows Presentation Foundation</i>
HTML	<i>HyperText Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
USB	<i>Universal Serial Bus</i>
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVOS	12
1.1.1	Objetivo Geral	12
1.1.2	Objetivo Específicos	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	SENSORIAMENTO AMBIENTAL X SENSORIAMENTO CORPORAL	13
2.2	UM COMPARATIVO DE MÉTODOS EXISTENTES	13
2.3	SENSOR KINECT	16
3	DESENVOLVIMENTO DO PROJETO	18
3.1	TECNOLOGIAS E FERRAMENTAS UTILIZADAS	18
3.2	METODOLOGIA	19
3.3	ANÁLISE E IMPLEMENTAÇÃO	21
3.3.1	Arquitetura	21
3.3.2	Requisitos funcionais	22
3.3.3	Requisitos não funcionais	23
3.3.4	Diagrama de caso de uso	23
3.3.5	Descrição do caso de uso - Estação Remota	24
3.3.6	Descrição do caso de uso - Central de monitoramento	25
3.3.7	Diagrama de fluxo	29
3.3.8	Modelo relacional do banco de dados	30
3.3.9	Diagrama de classe - Central de Monitoramento	30
3.3.10	Diagrama de classe - Estação Remota	31
3.3.11	Diagrama de Sequência - Cadastro de Usuário	33
3.3.12	Diagrama de Sequência - Monitoramento	33
3.3.13	Telas do Sistema	35
3.3.14	Regras de Avaliação de Queda	41
4	RESULTADOS E DISCUSSÕES	43
4.1	VALIDAÇÃO	43
4.2	ANÁLISE DOS RESULTADOS	47
5	CRONOGRAMA	49
6	CONCLUSÃO	50
	REFERÊNCIAS	52

1 INTRODUÇÃO

Este documento tem por objetivo apresentar o Trabalho de Conclusão de Curso à Universidade Tecnológica Federal do Paraná (UTFPR) - *Câmpus* Cornélio Procópio. O tema apresentado refere-se ao desenvolvimento de uma ferramenta para alertar possíveis situações de riscos de quedas em idosos institucionalizados, contribuindo na prestação de auxílio e socorro do mesmo, utilizando o Sensor Microsoft Kinect.

Projeções da Nações Unidas (Fundo de População) apresentam que 1 em cada 9 pessoas no mundo tem 60 anos ou mais (BRASIL, 2014). Segundo este estudo, em 2050 haverá mais idosos do que crianças de 15 anos. No Brasil, de acordo com a pesquisa do IBGE (Instituto Brasileiro de Geografia e Estatística) pessoas com mais de 60 anos somam 23,5 milhões de brasileiros (BRASIL, 2014).

O envelhecimento, provoca o declínio da capacidade funcional, a Figura 1 apresenta as reduções dos níveis de força muscular, alterações da marcha e alterações no controle do equilíbrio estático (NOVO et al., 2011). Observamos que há um aumento na ocorrência de possíveis quedas nessa faixa etária.

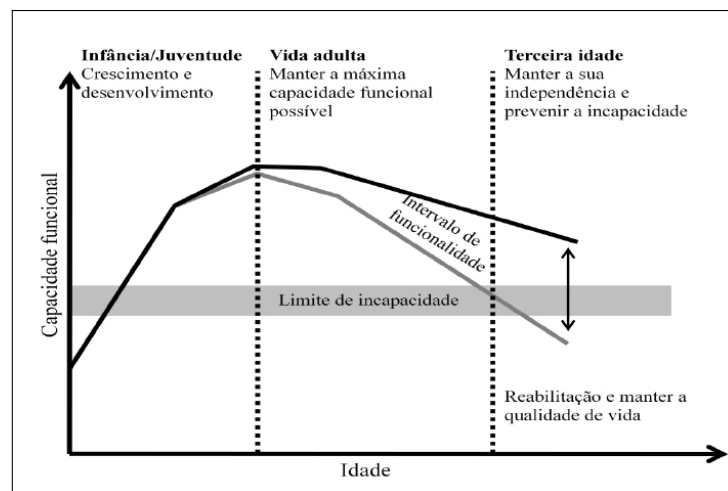


Figura 1: Curva de capacidade funcional ao longo do ciclo vital.

Fonte: (NOVO et al., 2011)

As quedas entre idosos constituem um dos principais problemas de saúde pública, devido à incidência, a complicações para a saúde e aos altos custos assistenciais (CARVALHO; COUTINHO, 2002). Alguns estudos realizados no Brasil e em outros países indicam que as quedas acontecem com mais frequência em idosos institucionalizados, que se encontram mais fragilizados e com diminuição da capacidade funcional (LOJUDICE et al., 2010). Em vários casos, a queda pode causar fraturas ou lesões graves. Esses ferimentos podem causar aos idosos a restrição nas suas atividades de vida diária e isolamento social (DUCA et al., 2013).

A assistência de uma pessoa 24 horas por dia não é acessível para todos. Em consequência disso, muitas vezes ao ocorrer um acidente, não tem ninguém no momento para socorrer a vítima (DAMASCENO et al., 2013).

A busca por novas tecnologias que otimizem os gastos com a saúde e proporcionem uma melhor qualidade de vida ao idoso é necessário. O monitoramento por meio de sensores ambientais representa uma alternativa, observando que o idoso institucionalizado normalmente permanece em um ambiente fechado, eliminando a necessidade de fixar equipamentos no corpo do paciente.

Este projeto propõe a construção de um sistema supervisório com a missão de informar ao cuidador que há uma situação de risco de queda, esta informação será exibida por um sinal de alerta em uma central de monitoramento, de modo que não haja invasão de privacidade do idoso.

Os sistemas supervisórios permitem que sejam monitoradas e rastreadas informações de um processo produtivo ou instalação física, coletando as informações com equipamentos de aquisição de dados e em seguida manipulando, analisando, armazenando e exibindo ao usuário (SILVA; SALVADOR, 2005).

Por se tratar de um sistema supervisório foi verificada a necessidade de dividir a aplicação em dois módulos. Um módulo será designado para as estações remotas, sendo responsável em manipular e analisar os dados adquiridos do sensor Kinect. Outro módulo será a estação central, responsável por recolher os dados fornecidos pelas estações remotas e agir em conformidade com os eventos detectados. Os registros dos eventos que ocorreram durante o monitoramento poderá ser analisados por meio de relatórios.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

Desenvolver um sistema supervisor para alertar possíveis riscos de quedas em idosos institucionalizados, resguardando sua privacidade.

1.1.2 OBJETIVO ESPECÍFICOS

- Pesquisar e compreender o monitoramento do idoso institucionalizado;
- Pesquisar e analisar abordagens de métodos para monitoramento existentes;
- Estudar as ferramentas necessárias para a desenvolvimento do sistema;
- Definir o escopo do sistema;
- Modelar o sistema utilizando linguagem de modelagem UML;
- Realizar a implementação do sistema respeitando a delimitação definida;
- Testar e validar o sistema;
- Documentar o desenvolvimento e os resultados obtidos.

Os objetivos específicos necessários para o desenvolvimento do sistema são apresentados na seção 3.3, tópicos requisitos funcionais e não funcionais.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 SENSORIAMENTO AMBIENTAL X SENSORIAMENTO CORPORAL

O monitoramento do ambiente requer equipar o local com uma infraestrutura para a coleta dos dados. Geralmente, para realizar a coleta utilizam-se as técnicas de visão computacional. Esse modo de monitoramento tem como principal vantagem a não fixação de equipamentos no corpo do indivíduo, contudo, necessita de uma área de cobertura do sistema de rastreamento (UGULINO et al., 2012).

O sensoriamento corporal utiliza dispositivos acoplado ao corpo do indivíduo, conhecido como *wearables*. Neste modo de sensoriamento é utilizado acelerômetros para coleta de dados. A principal vantagem desta abordagem é não necessitar de infraestrutura no ambiente, possibilitando ao indivíduo realizar a atividade em qualquer local (UGULINO et al., 2012).

No presente trabalho é adotado o sensoriamento ambiental, sem a necessidade de vestir equipamentos que podem causar desconforto para o idoso.

2.2 UM COMPARATIVO DE MÉTODOS EXISTENTES

Os sistemas de detecção de quedas, normalmente são semelhantes, seu principal objetivo é diferenciar entre eventos de queda e atividades da vida diária (AVD). As posições sentar-se ou ir da posição de pé para deitado, tem fortes semelhanças as quedas. Para realizar testes em um detector de queda, é necessário coletar dados de quedas e AVD, real ou simulado por voluntários jovens. Os dados são registrados por sensores, na forma de sinais de aceleração, imagens, sinais de pressão, etc. Os dados são processados e classificados de acordo com a lógica de detecção de queda (IGUAL et al., 2013).

A pesquisa realiza por Igual et al. (2013) apresenta um levantamento bibliográfico, que identificar métodos usados para a detecção de quedas em sistemas que utilizam câmera, apresentada na tabela 1.

Tabela 1: Métodos para detecção de quedas

	1º fase	2º fase	3º fase	4º fase	5º fase
Anderson et al. (2009)	Extração da silhueta de cada câmera. Então, uma representação 3D do corpo é construída.	Extração do centróide (centro geométrico), altura, principal orientação do corpo e similaridade da principal orientação com o plano normal do chão.	Estado humano deduzido usando lógica <i>fuzzy</i> (3 estados: vertical, no deitado ou entre).	Informação em sequências de estado são reduzidas por sumarização linguística para produzir sentenças legíveis por humanos.	Detecção de queda por uma segunda camada de lógica <i>fuzzy</i> , tomando entradas de um único resumo: média de estado, duração, velocidade, oscilação, etc.
Liu et al. (2010)	Identificação do corpo humano usando técnica de diferenciação de <i>frame</i> .	Processamento de imagem: filtro médio para deixar a imagem mais clara, limiar para obter uma imagem binária, rotulando componentes conectados.	Extração de características e redução de efeitos de atividades de membros superiores.	Classificador de <i>K-Nearest Neighbor</i> para classificação de posturas humanas.	Fluxo de detecção de evento de queda: a decisão de um incidente de queda é determinado pela transição de evento e diferença de tempo entre eventos.
Rimminen et al. (2010)	Estimar a posição de sujeito usando observações dos sensores de imagem <i>near-field</i> .	<i>Tracking</i> (filtro de Kalman) e <i>multi-target tracking</i> (Algoritmo de associação de dados de <i>Rao-Blackwellized Monte Carlo</i>).	Extração de características relacionadas ao nível NFI.	Modelagem de evolução de estado como uma <i>two-state Markov chain</i> (queda, levantar).	Estimação de pose usando filtragem <i>Bayesian</i> . Ela combina o modelo prévio com informação das características.
Tzeng et al. (2010)	Suspeita de queda: limiar do sinal de pressão do solo.	Se a pressão do solo exceder um dado limiar: Captura de Imagem.	Remoção de fundo através de uma filtragem de imagem e expansão (operações morfológicas).	Extração de características de imagem	Combinação do sinal de pressão do solo e características de imagem para sinalizar na queda.
Diraco et al. (2010)	Calibragem de câmera	Modelagem de fundo usando mistura de métodos <i>Gaussians</i> .	Detecção de regiões de movimento (segmentação <i>Bayesian</i>) e refinamento de <i>blobs</i> segmentados (operações morfológicas e componentes conectados).	Suspeita de queda: A distância do centróide ao solo plano é menor que um valor pré-fixado.	Confirmação de queda se uma situação imutável persiste por pelo menos 4 segundos.

Fonte: (IGUAL et al., 2013)

Verificamos na Tabela 1 diversas técnicas utilizadas na detecção de quedas, como: extração de silhueta e lógica *fuzzy* (ANDERSON et al., 2009), classificador de *K-Nearest Neighbor* (LIU et al., 2010), utilização de sensores de imagem *Near-Field* e *Tracking* (RIMMINEN et al., 2010), limiar (TZENG et al., 2010), métodos *Gaussians*, segmentação *Bayesian* e refinamento de *blogs* segmentados (DIRACO et al., 2010).

A Figura 2 mostra a utilização de extração de silhueta.

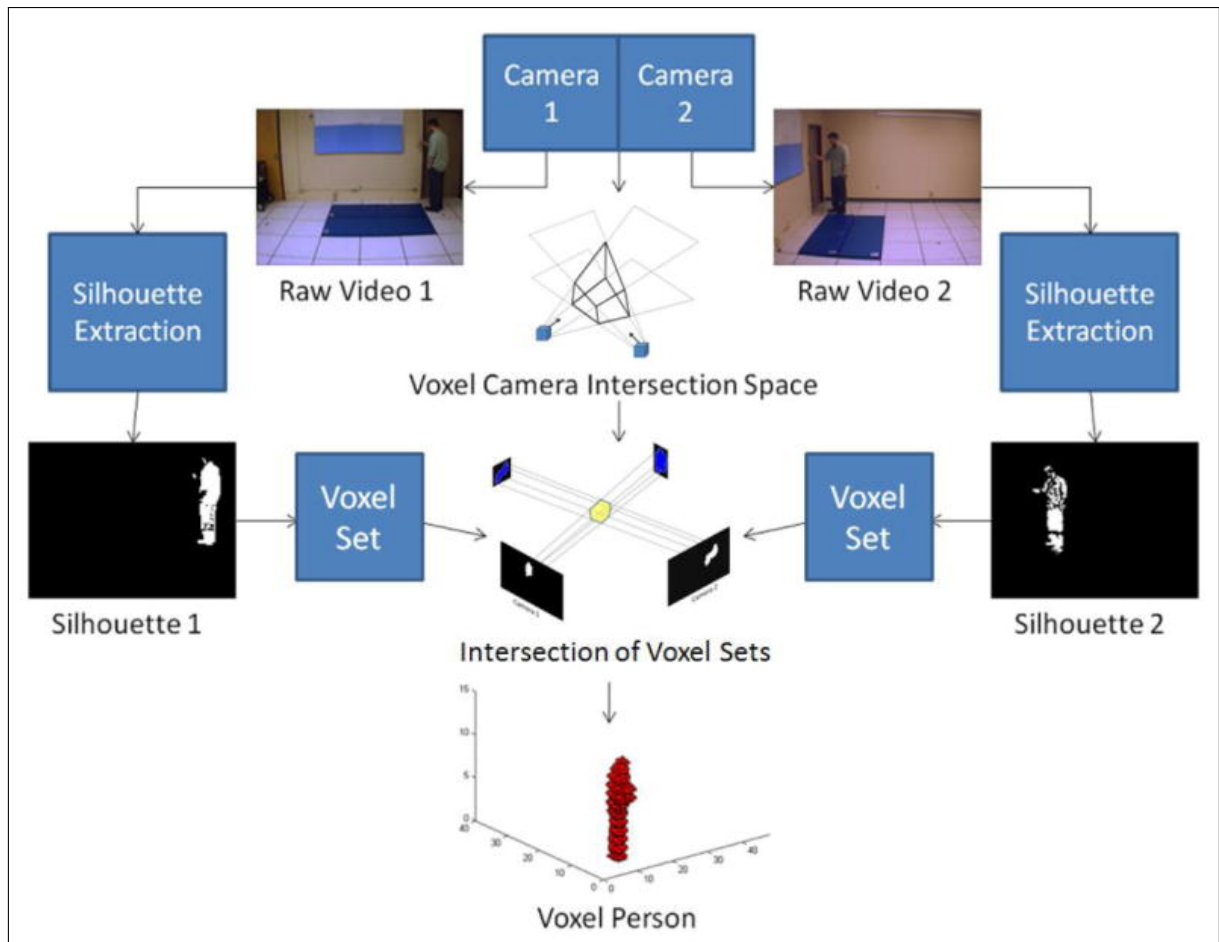


Figura 2: Utilização de extração de silhueta.

Fonte: (ANDERSON et al., 2009)

A utilização de dispositivos *Near-Field* é mostrada na Figura 3.

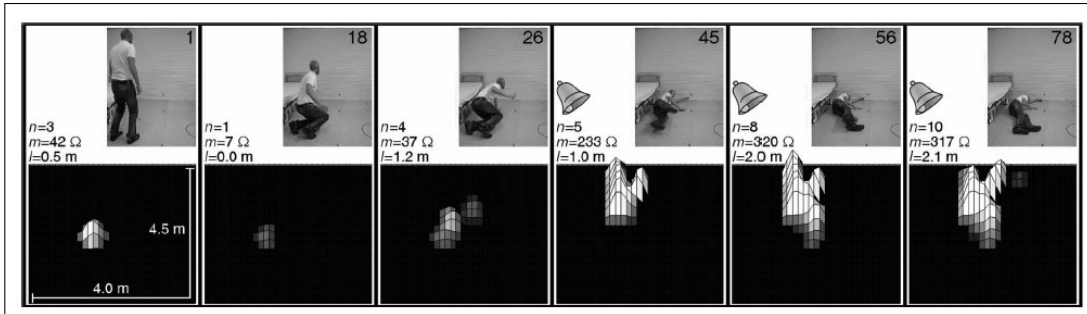


Figura 3: Utilização de dispositivos Near-Fied.

Fonte: (RIMMINEN et al., 2010)

2.3 SENSOR KINECT

O Kinect foi idealizado pelo brasileiro Alex Kipman, inicialmente sob o codinome de projeto natal, fazendo referência à cidade brasileira Natal (CARDOSO; SCHMIDT, 2012). Foi lançado no fim do ano de 2010 como acessório do console de jogos Xbox 360, da Microsoft. (GALVEIA, 2014). O conceito de Kinect é a base de sistemas que usam *Natural User Interface* (NUI), no qual é possível controlar dispositivos com gestos corporais e comando de voz. (ISLAM et al., 2013).

A Figura 4 apresenta a estrutura física do Kinect, que dispõe de uma câmara RGB, um emissor e receptor de infravermelhos (*Depth Sensor*), um conjunto de microfones e um motor na base que permite movimentar sua estrutura (GALVEIA, 2014).



Figura 4: Hardware do Kinect.

Fonte: (FERNANDEZ, 2012)

A câmera RGB (*Red, Green, Blue*) é uma câmera de 8-bits com resolução VGA de 640x480 pixels, no qual opera em 30 quadros por segundo, com 8 bits por canal (NETO et al., 2014).

O sensor de profundidade 3D consiste em um projetor laser de infravermelho e uma câmera infravermelha. O projetor e a câmera cria um mapa de profundidade, no qual provê a informação de distância entre um objeto e a câmera (HAN et al., 2013).

O Kinect possui um processamento interno que mapeia 20 articulações do usuário formando um esqueleto, conforme apresentando na Figura 5. O esqueleto mapeado pelo dispositivo é composto das coordenadas tridimensionais, X, Y e Z, de cada articulação, permitindo a representação e manipulação 3D do mesmo (CARDOSO; SCHMIDT, 2012).

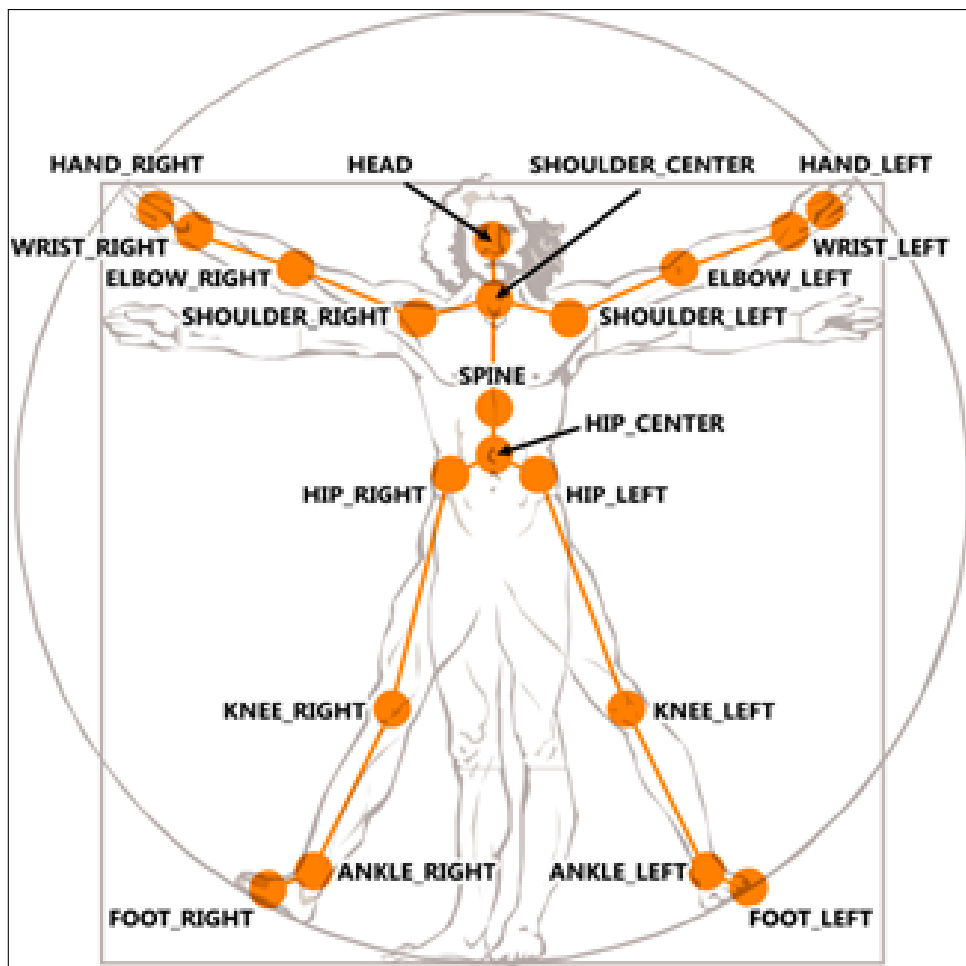


Figura 5: As 20 articulações mapeadas pelo Kinect.

Fonte: (MSDN, 2012)

3 DESENVOLVIMENTO DO PROJETO

3.1 TECNOLOGIAS E FERRAMENTAS UTILIZADAS

C#: Linguagem de desenvolvimento escolhida por ser uma linguagem orientada a objetos, que possui o SDK para desenvolvimento do Kinect fornecido pela Microsoft. Além da linguagem possuir um ambiente de desenvolvimento integrado para Windows, o Visual Studio.

Kinect for Windows SDK: Biblioteca que permite aos desenvolvedores criar aplicativos que suportam gestos e reconhecimento de voz, utilizando a tecnologia de sensores Kinect em computadores com o Windows (MICROSOFT, 2014). Escolhida por ser a biblioteca oficial fornecida pela Microsoft.

ASP NET MVC: É um *framework* da Microsoft para desenvolvimentos de sites seguindo a arquitetura MVC. Permite a separação de responsabilidade e um desenvolvimento ágil (MICROSOFT, 2015a). Escolhido por possuir uma versão para a linguagem C# e ser recomendado pela Microsoft para desenvolvimento web.

SignalR: É uma biblioteca para desenvolvedores ASP.NET que torna o desenvolvimento de funcionalidades web em tempo real fácil. Permite a comunicação bidirecional entre o servidor e o cliente. Suporta Web Sockets e outras técnicas compatíveis para navegadores mais antigos e possui gerenciamento de conexão (MICROSOFT, 2015b). Escolhida por possuir integração com ASP NET MVC, além de suportar navegadores antigos. Essa tecnologia foi utilizada para a transmissão dos dados em tempo real entre as estações remotas e a central de monitoramento.

Entity Framework: É um framework do tipo ORM (*Object/Relational Mapping*) que permite aos desenvolvedores trabalhar com dados relacionais como objetos de domínio específico, eliminando a necessidade de maior parte dos códigos de acesso de dados que os desenvolvedores geralmente precisam escrever (MSDN, 2015b). Escolhida por possuir uma fácil integração com o banco de dados SQL Server.

WPF: Fornece aos desenvolvedores um modelo de programação unificado para criar aplicativos modernos no Windows (MSDN, 2015a). Tal tecnologia foi adotada por ser ideal para trabalhar com *canvas* e figuras geométricas.

HTML5: Linguagem de marcação de hipertexto em documentos para web, com a característica de interligar conteúdos web (SILVA, 2008). Foi utilizada no desenvolvimento da central de monitoramento.

CSS3: Linguagem usada para a apresentação e formatação de um documento HTML (SERRA et al., 2012). Foi utilizada para melhorar a interface da central de monitoramento.

JavaScript: Linguagem de *script* utilizada em páginas web para validar formulários, detectar objetos e adicionar uma série de outras funcionalidades interativas (SILVA, 2010). Utilizada no desenvolvimento da central de monitoramento.

Bootstrap: Biblioteca HTML, CSS e Javascript para desenvolvimento web. Utilizado no desenvolvimento da central de monitoramento, possibilitando a construção de uma interface adaptável a todos os tamanhos de telas.

UML: Linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos (MIRA; MARQUEZ, 2000). Escolhida por ser a linguagem padrão utilizada na área de engenharia de software.

Microsoft Sql Server: Sistema de gerenciamento e análise de banco de dados para soluções de comércio eletrônico, linha de negócios e *data warehouse* (MSDN, 2014). Escolhido por ser um sistema que tem fácil acesso com a linguagem C# e possuir integração com o Visual Studio.

Microsoft Visual Studio: Pacote de produtos, ferramentas e tecnologias da Microsoft para desenvolvimento de software (MSDN, 2012). Escolhido por possuir integração com SQL Server e ser o principal ambiente de desenvolvimento da linguagem C#.

Astah: Ferramenta para construção de diagramas, seguindo o padrão da UML. Escolhida por ser uma das ferramentas mais completas do mercado e utiliza no ensino acadêmico.

3.2 METODOLOGIA

A metodologia adotada para o desenvolvimento da aplicação foi a Espiral *Win Win*, apresentada na Figura 6. Este modelo define um conjunto de atividades de negociação que se

iniciam a cada ciclo da espiral (BOEHM; EGYED, 1998). Escolhida por tornar as estimativas mais realísticas e ser mais versátil para lidar com mudanças.

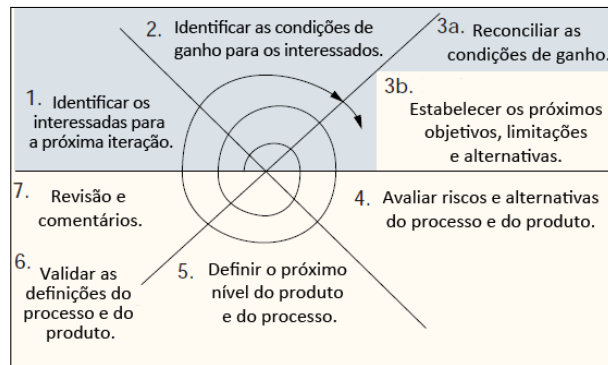


Figura 6: Espiral Win Win.

Fonte: (BOEHM et al., 1998)

As seguintes etapas foram definidas ao longo do projeto:

Levantamento bibliográfico – Esta etapa consistiu no estudo do idoso institucionalizado, bem como métodos de monitoramento existentes.

Estudo do dispositivo Kinect – Após o estudo inicial, foi feito o estudo sobre o dispositivo Kinect e suas bibliotecas para desenvolvimento.

Modelagem do projeto Estação Remota – A partir das informações levantadas, foi concentradas as informações necessárias para iniciar a modelagem do banco de dados e dos diagramas do sistema.

Implementação do Estação Remota – Com base nos diagramas descritos foi iniciado a implementação do sistema, bem como definição de ferramentas e tecnologias a serem usadas.

Modelagem do projeto Estação Central de Monitoramento – Após o término da implementação inicial da estação remota, foi iniciado a modelagem da central de monitoramento e um estudo sobre tecnologias para transmissão de dados em tempo real.

Implementação do projeto Central de Monitoramento – Nesta fase foi feita a implementação da central de monitoramento e da integração entre os sistemas, bem como definição das tecnologias a serem utilizadas.

Testes e validação do sistema – Com os dois sistemas implementados, foi realizado testes para verificar as funcionalidades e a comunicação entre os sistemas.

Coleta de dados – Com a validação do sistema completa, foi iniciado a coleta de dados para verificar a capacidade do sensor Kinect em capturar dados em diversas posições.

Análise dos dados – Após a coleta dos dados, foi realizado a análise dos dados e uma discussão sobre a capacidade do sensor Kinect.

O esforço para a realização de cada ciclo pode ser visualizado na seção Cronograma.

3.3 ANÁLISE E IMPLEMENTAÇÃO

3.3.1 ARQUITETURA

O sistema é composto por várias estações remotas, as quais são compostas por um computador e um sensor Kinect. O sistema utiliza a câmera RGB e o sensor de profundidade do Kinect para detectar e rastrear movimentos dos idosos no quarto. Os dados são capturados pelo Kinect na resolução de 640x480 pixels, com uma taxa de 30 quadros por segundo, esses dados são enviados para o computador via conexão USB, que são analisados e então enviados a central de monitoramento via conexão de rede.

A central de monitoramento é composta por um servidor web e um banco de dados, o qual é utilizado para armazenar os dados recebidos das estações remotas. O cuidador acessará a central de monitoramento por meio de um navegador web. A Figura 7 ilustra a arquitetura física do sistema.

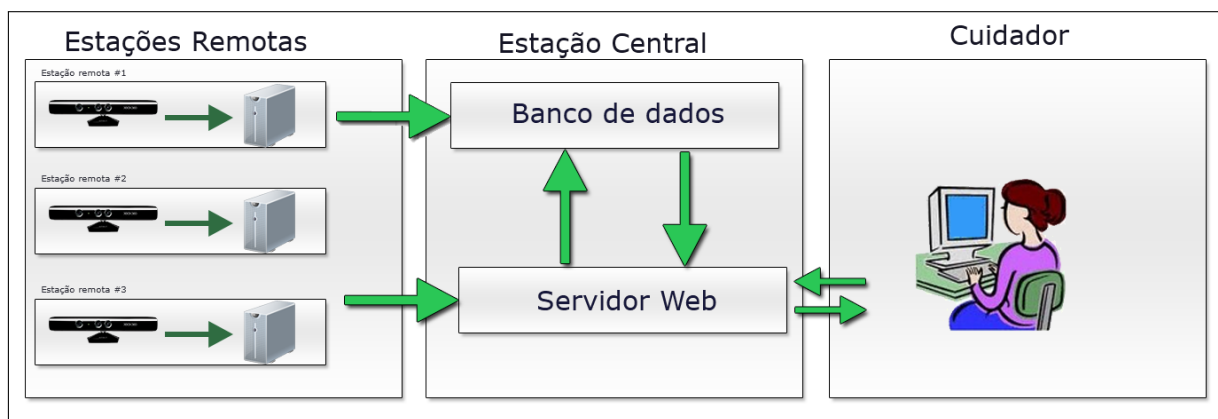


Figura 7: Arquitetura física do sistema.

Para a implementação da central de monitoramento foi utilizado o padrão de projeto MVC (*Model-View-Controller*). O padrão MVC sugere uma arquitetura de software dividida em camada, separando as regras de negócio da lógica de apresentação, viabilizando o desenvolvimento de um código organizado, e posteriormente, a manutenção do sistema sem dificuldade (SILVA, 2012). Esse padrão é adotado pelo *framework* ASP NET MVC da Microsoft, utilizado no desenvolvimento da central de monitoramento.

A camada *model* é responsável pelo gerenciamento da informação, como regras de negócio e acesso a banco de dados.

A camada *view* apresenta os dados do *model* para o usuário, bem como envia os dados fornecidos pelo usuário para o *controller*.

A camada *controller* gerencia o fluxo entre a *view* e o *model*, e é responsável pela validação e filtragem da entrada de dados.

A Figura 8 representa o diagrama da arquitetura MVC.

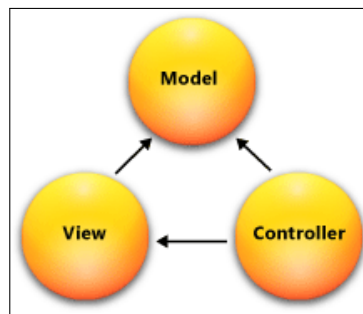


Figura 8: Diagrama Arquitetura MVC.

Fonte: (MICROSOFT, 2015a)

3.3.2 REQUISITOS FUNCIONAIS

- RF1** O módulo de estação remota deve fazer a aquisição dos quadros capturado pelo Sensor Kinect;
- RF2** O módulo de estação remota deve identificar o corpo humano com o recurso de mapeamento das articulações fornecido pelo Kinect para a construção do esqueleto;
- RF3** O módulo de estação remota deve extrair as características da imagem utilizando os recursos do Kinect de profundidade de imagem, gesto, poses e rastreamento do esqueleto;
- RF4** O módulo de estação remota deve identificar riscos de quedas com base nos dados capturados pelo Kinect;
- RF5** O módulo de estação remota deve enviar ocorrências dos eventos monitorados para a central de monitoramento;
- R6** O módulo de estação remota deve enviar o número de pessoas que estão na área de monitoramento do quarto para a central;

- RF7** Os módulos devem ser integrados de forma que as estações remotas e a estação central consiga-se comunicar pela rede;
- RF8** A estação central de monitoramento deve identificar os eventos transmitidos pelas estações remotas e exibir uma mensagem de alerta conforme o evento ocorrido;
- RF9** A estação central de monitoramento deve identificar o número de pessoas transmitidas pelas estações remotas e exibir na central de monitoramento.;
- RF10** O sistema deve gravar os registros dos eventos ocorridos, indicando o horário e quarto em que ocorreu o evento.

3.3.3 REQUISITOS NÃO FUNCIONAIS

- RNF1** O sistema deve ser desenvolvido para uma plataforma Web, por atender um maior número de dispositivos;
- RNF2** O sistema deve ser desenvolvido utilizando a linguagem C#;
- RNF3** A biblioteca do Kinect utilizada para desenvolvimento deve ser da Microsoft;
- RNF4** O sistema deve utilizar banco de dados SQL Server;
- RNF5** A central de monitoramento deverá ter a interface agradável e objetiva.

3.3.4 DIAGRAMA DE CASO DE USO

As Figuras 9 e 10 mostram os diagramas de caso de uso, que designa as funcionalidades do sistema.

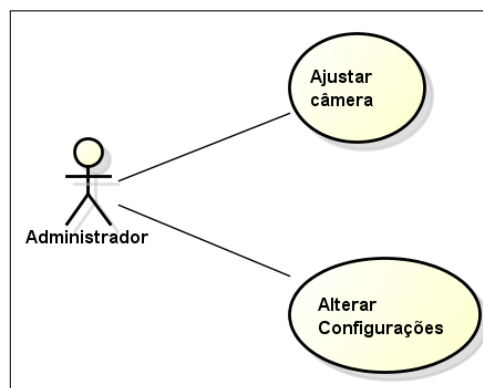


Figura 9: Diagrama de caso de uso - Estação remota.

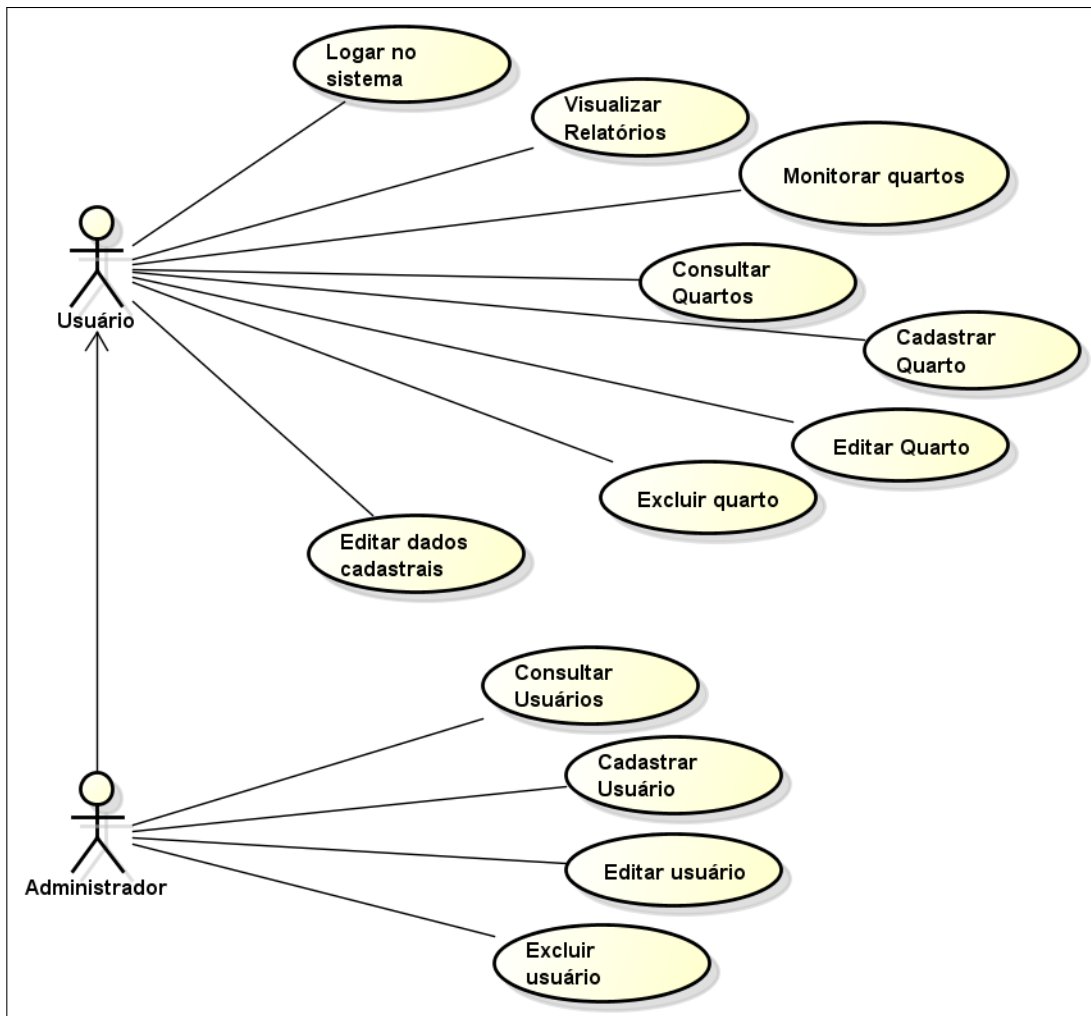


Figura 10: Diagrama de caso de uso - Central de monitoramento.

3.3.5 DESCRIÇÃO DO CASO DE USO - ESTAÇÃO REMOTA

Identificação: UC0001	
Nome: Ajustar câmera	
Atores: Administrador	
Pré-condições:	
Pós-condições:	
Sequência típica de eventos	
Ator	Sistema
1. Acessa a opção de câmera	3. Verifica se câmera está ligada
2. Altera ângulo da câmera	4. Seta novo ângulo na câmera
Sequência alternativa	
3. Câmera desligada	
1. Exibe imagem de câmera desligada	

Figura 11: UC0001 - Ajustar câmera.

Identificação: UC0002	
Nome: Alterar configurações	
Atores: Administrador	
Pré-condições:	
Pós-condições: Configuração altera com sucesso	
Sequência típica de eventos	
Ator	Sistema
1. Acessa a opção de configurações	2. Carrega informações de configurações
3. Altera configurações	5. Verifica configurações alteradas
4. Clica no botão salvar	6. Efetiva alteração
Sequência alternativa	
5a. Informações preenchidas incorretamente	
1. Exibe mensagem de erro	

Figura 12: UC0002 - Alterar configurações.

3.3.6 DESCRIÇÃO DO CASO DE USO - CENTRAL DE MONITORAMENTO

Identificação: UC0001	
Nome: Logar no sistema	
Atores: Usuário, Administrador	
Pré-condições:	
Pós-condições: Login efetuado com sucesso.	
Sequência típica de eventos	
Ator	Sistema
1. Acessa a página de login	4. Valida informações
2. Preenche informações do formulário	5. Efetua o login
3. Clica no botão entrar	
Sequência alternativa	
4a. Informações preenchidas incorretamente	
1. Exibe mensagem de erro	

Figura 13: UC0001 - Logar no sistema.

Identificação: UC0002	
Nome: Consultar quartos	
Atores: Usuário, Administrador	
Pré-condições: Estar logado	
Pós-condições:	
Sequência típica de eventos	
Ator	Sistema
1. Entra na listagem de quartos	2. Verifica os quartos cadastrados
	3. Exibe a listagem de quartos

Figura 14: UC0002 - Consultar quartos.

Identificação: UC0003	
Nome: Cadastrar quarto	
Atores: Usuário, Administrador	
Pré-condições: Estar logado	
Pós-condições: Quarto cadastrado com sucesso	
Sequência típica de eventos	
Ator	Sistema
1. Entra na listagem de quartos	5. Valida informações
2. Clica no link de cadastrar quarto	6. Realiza o cadastro
3. Preenche informações do formulário	
4. Clica no botão gravar	
Sequência alternativa	
5a. Informações preenchidas incorretamente	
2. Exibe mensagem de erro e coloca o foco no primeiro campo com problema	

Figura 15: UC0003 - Cadastrar quartos.

Identificação: UC0004	
Nome: Editar quarto	
Atores: Usuário, Administrador	
Pré-condições: Estar logado	
Pós-condições: Informações do quarto alterada com sucesso	
Sequência típica de eventos	
Ator	Sistema
1. Entra na listagem de quartos	4. Valida informações alteradas
2. Seleciona o quarto que deseja alterar	5. Efetiva as alterações
3. Clica no botão de editar	
4. Alterar as informações desejadas	
Sequência alternativa	
4a. Informações preenchidas incorretamente	
3. Exibe mensagem de erro	

Figura 16: UC0004 - Editar quarto.

Identificação: UC0005	
Nome: Deletar quarto	
Atores: Usuário, Administrador	
Pré-condições: Estar logado	
Pós-condições: Quarto excluído da base de dados	
Sequência típica de eventos	
Ator	Sistema
1. Entra na listagem de quartos	4. Verifica dependências
2. Seleciona o quarto que deseja deletar	5. Realiza a exclusão
5. Clica no botão de deletar	
6. Confirma a exclusão	
Sequência alternativa	
4a. Quarto referenciado em outra tabela	
3. Exibe mensagem de erro	

Figura 17: UC0005 - Deletar quarto.

Identificação: UC0006	
Nome: Consultar quartos	
Atores: Administrador	
Pré-condições: Estar logado e ser administrador	
Pós-condições:	
Sequência típica de eventos	
Ator	Sistema
1. Entra na listagem de quartos	2. Verifica os quartos cadastrados
	3. Exibe a listagem de quartos

Figura 18: UC0006 - Consultar usuários.

Identificação: UC0007	
Nome: Cadastrar usuários	
Atores: Administrador	
Pré-condições: Estar logado e ser administrador	
Pós-condições: Usuário cadastrado com sucesso	
Sequência típica de eventos	
Ator	Sistema
1. Entra na listagem de usuários	5. Valida informações
2. Clica no link de cadastrar usuários	6. Realiza o cadastro
3. Preenche informações do formulário	
4. Clica no botão gravar	
Sequência alternativa	
5a. Informações preenchidas incorretamente	
4. Exibe mensagem de erro e coloca o foco no primeiro campo com problema	

Figura 19: UC0007 - Cadastrar usuário.

Identificação: UC0008	
Nome: Editar usuário	
Atores: Administrador	
Pré-condições: Estar logado e ser administrador	
Pós-condições: Informações do usuário alterada com sucesso	
Sequência típica de eventos	
Ator	Sistema
1. Entra na listagem de usuários	4. Valida informações alteradas
2. Seleciona o usuário que deseja alterar	5. Efetiva as alterações
3. Clica no botão de editar	
4. Alterar as informações desejadas	
Sequência alternativa	
4a. Informações preenchidas incorretamente	
5. Exibe mensagem de erro	

Figura 20: UC0008 - Editar usuário.

Identificação: UC0009	
Nome: Deletar usuário	
Atores: Administrador	
Pré-condições: Estar logado e ser administrador	
Pós-condições: Usuário excluído da base de dados	
Sequência típica de eventos	
Ator	Sistema
1. Entra na listagem de usuários	4. Verifica dependências
2. Seleciona o usuário que deseja deletar	5. Realiza a exclusão
3. Clica no botão de deletar	
4. Confirma a exclusão	
Sequência alternativa	
4a. Usuário referenciado em outra tabela	
2. Exibe mensagem de erro	

Figura 21: UC0009 - Deletar usuário.

Identificação: UC0010	
Nome: Visualizar relatórios	
Atores: Usuário, Administrador	
Pré-condições: Estar logado	
Pós-condições:	
Sequência típica de eventos	
Ator	Sistema
1. Seleciona o relatório desejado	3. Verifica os dados armazenados
2. Seleciona os filtros desejados	4. Exibe relatório
Sequência alternativa	
4a. Dados não encontrados.	
1. Exibe mensagem de erro	

Figura 22: UC0010 - Visualizar relatórios.

Identificação: UC0011	
Nome: Monitorar quartos	
Atores: Usuário, Administrador	
Pré-condições: Estar logado	
Pós-condições:	
Sequência típica de eventos	
Ator	Sistema
1. Entra na área de monitoramento	2. Verifica os quartos cadastrados
	3. Exibe quartos em monitoramento

Figura 23: UC0011 - Monitorar quartos.

Identificação: UC0012	
Nome: Alterar dados cadastrais	
Atores: Usuário, Administrador	
Pré-condições: Estar logado	
Pós-condições: Dados alterados com sucesso	
Sequência típica de eventos	
Ator	Sistema
1. Acessa o menu meus dados	4. Verifica os dados alterados
2. Altera as informações desejadas	5. Efetiva as alterações
3. Clica no botão gravar	
Sequência alternativa	
4a. Informações preenchidas incorretamente	
1. Exibe mensagem de erro	

Figura 24: UC0012 - Alterar dados cadastrais.

3.3.7 DIAGRAMA DE FLUXO

A Figura 25 ilustra o fluxo do módulo da estação remota, responsável por capturar as imagens. Caso detectado uma situação de risco, um sinal de alerta é enviado central.

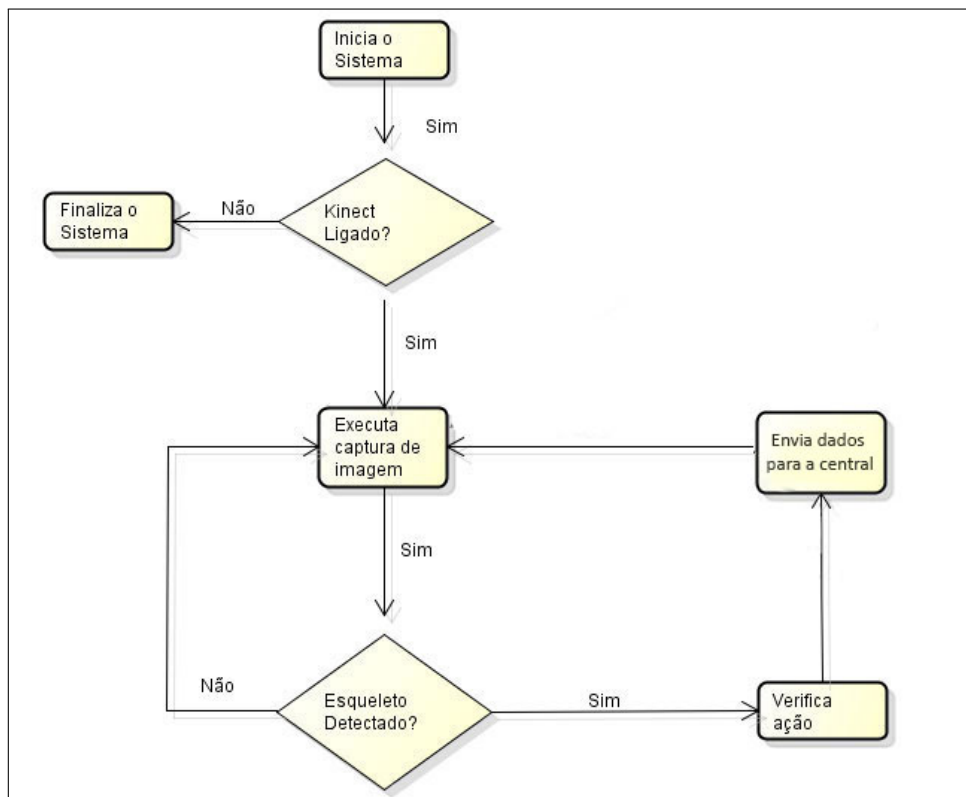


Figura 25: Fluxograma do monitor.

3.3.8 MODELO RELACIONAL DO BANCO DE DADOS

A Figura 26 mostra a visão do modelo relacional do banco de dados, detalhando os atributos de cada entidade. As tabelas *Usuario* e *Permissao* são responsáveis por armazenar os dados dos usuários do sistema e sua permissão de acesso. Na tabela *Queda* fica armazenado os dados obtidos das estações remotas, todas as quedas pertencem a um quarto, representado pela tabela *Quarto* na qual é armazenado dados referentes ao quarto.

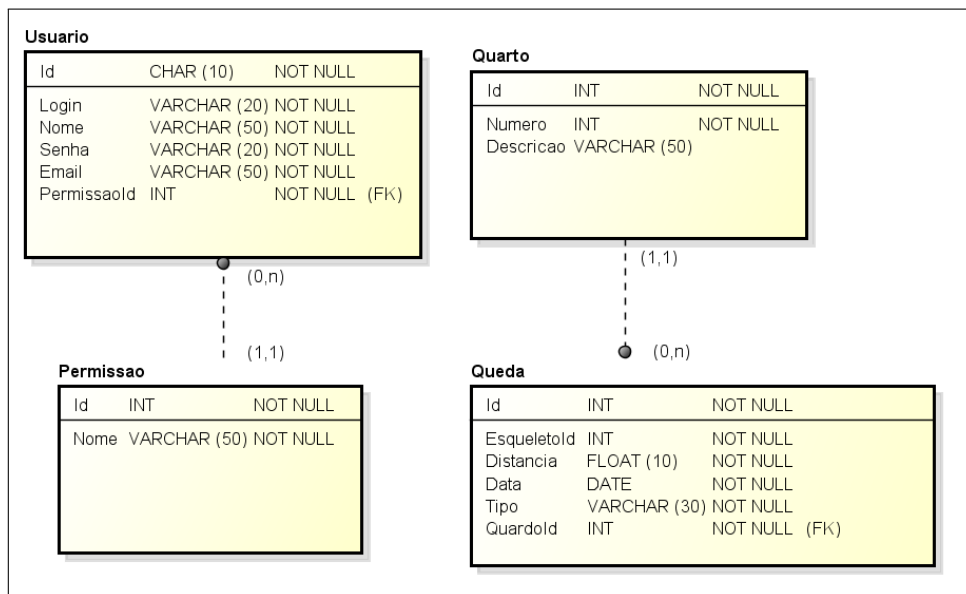


Figura 26: Modelo relacional do banco de dados.

3.3.9 DIAGRAMA DE CLASSE - CENTRAL DE MONITORAMENTO

A Figura 27 representa o diagrama de classe da central de monitoramento, separado por pacotes. O pacote de domínio representa as entidades do banco de dados, as classes para acesso a banco de dados são representados no pacote *Infra*. O pacote *Hub* contém as classes responsáveis por fazer a comunicação em tempo real. Para o acesso ao banco de dados foi utilizado a biblioteca *Entity framework*, presentanda pela classe *Contexto* do pacote *Infra* e para comunicação em tempo real a biblioteca *SignalR*, presentado pela classe *Hub* do pacote *Hub*.

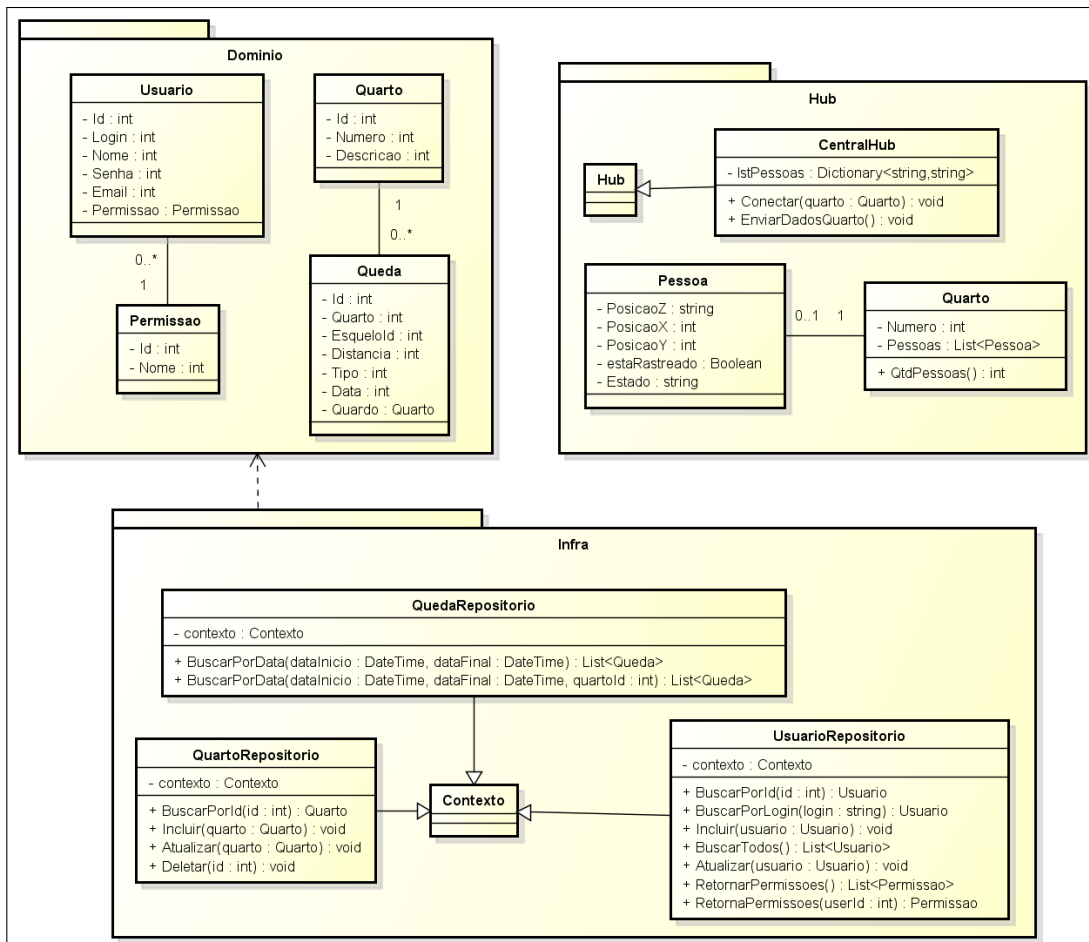


Figura 27: Diagrama de classe - Central de Monitoramento.

3.3.10 DIAGRAMA DE CLASSE - ESTAÇÃO REMOTA

A Figura 28 representa o diagrama de classe da estação remota. A classe Camera é a principal classe do sistema, composta por várias classes que auxiliam na inicialização da câmera, conexão com a central e análise dos quadros capturados pelo Kinect. A classe MovimenQueda é responsável por verificar o movimento do usuário. As classes EsqueletoUtil e ExtensãoSketonFrame são classes utilitárias para facilitar as operações no sistema. As configurações da estação remota são salvas em um arquivo do tipo XML (*eXtensible Markup Language*), as classes responsáveis por essas operações são a Config e ConfigXml.

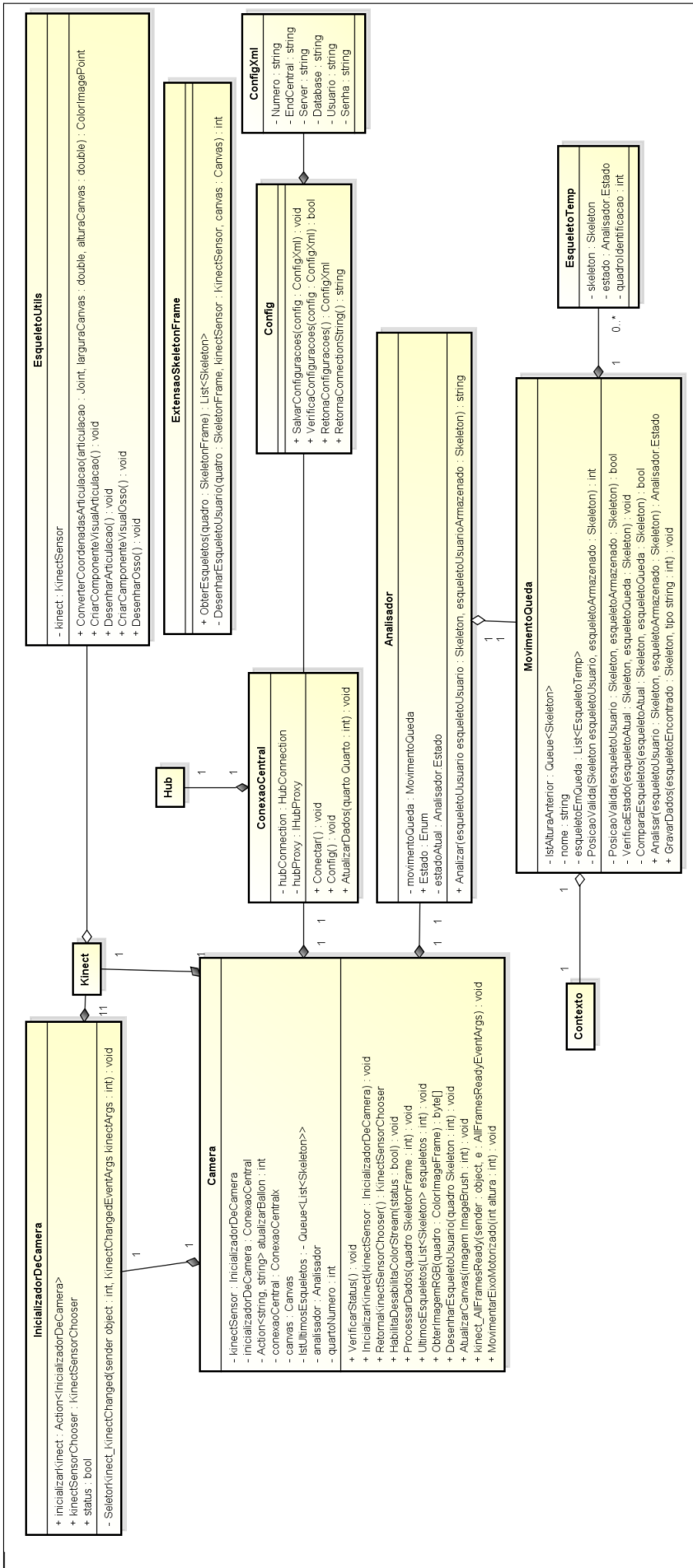


Figura 28: Diagrama de classe - Estação Remota.

3.3.11 DIAGRAMA DE SEQUÊNCIA - CADASTRO DE USUÁRIO

A Figura 29 representa o diagrama de sequência da central de monitoramento, caso de uso UC0007 cadastrar usuários, seguindo o padrão de arquitetura MVC.

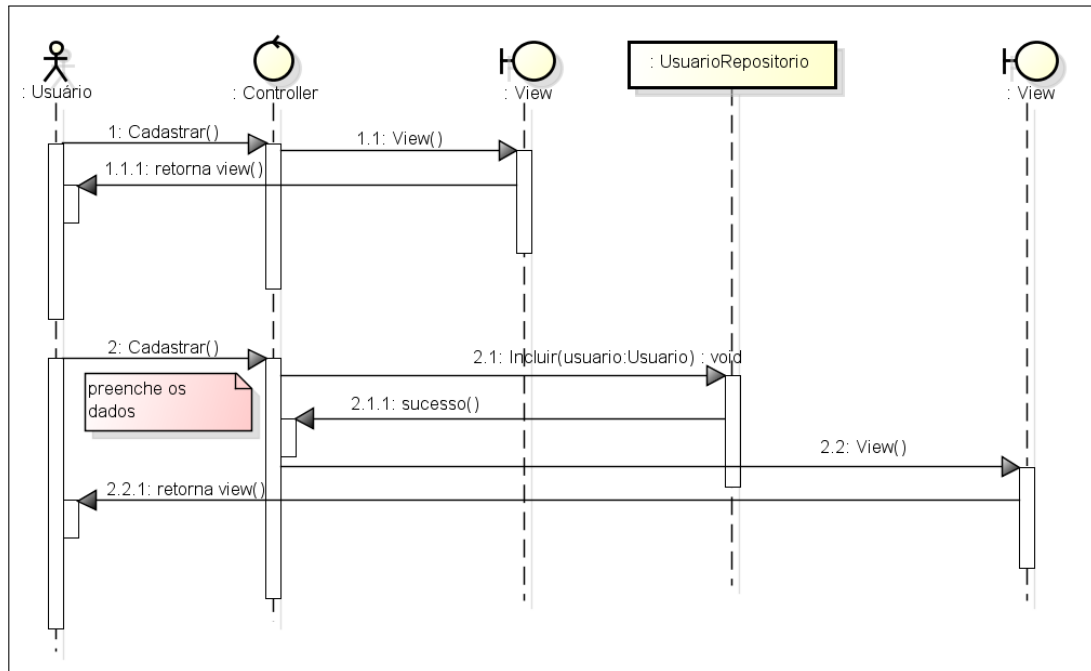


Figura 29: Diagrama de sequência - Estação Remota.

3.3.12 DIAGRAMA DE SEQUÊNCIA - MONITORAMENTO

A Figura 30 representa o diagrama de sequência ao ligar o monitor, bem como a interação entre as classes Analisador e MovimentoQueda responsável por analisar os quadros fornecidos pelo Kinect, e posteriormente, o envio de dados para a central por meio da classe ConexaoCentral.

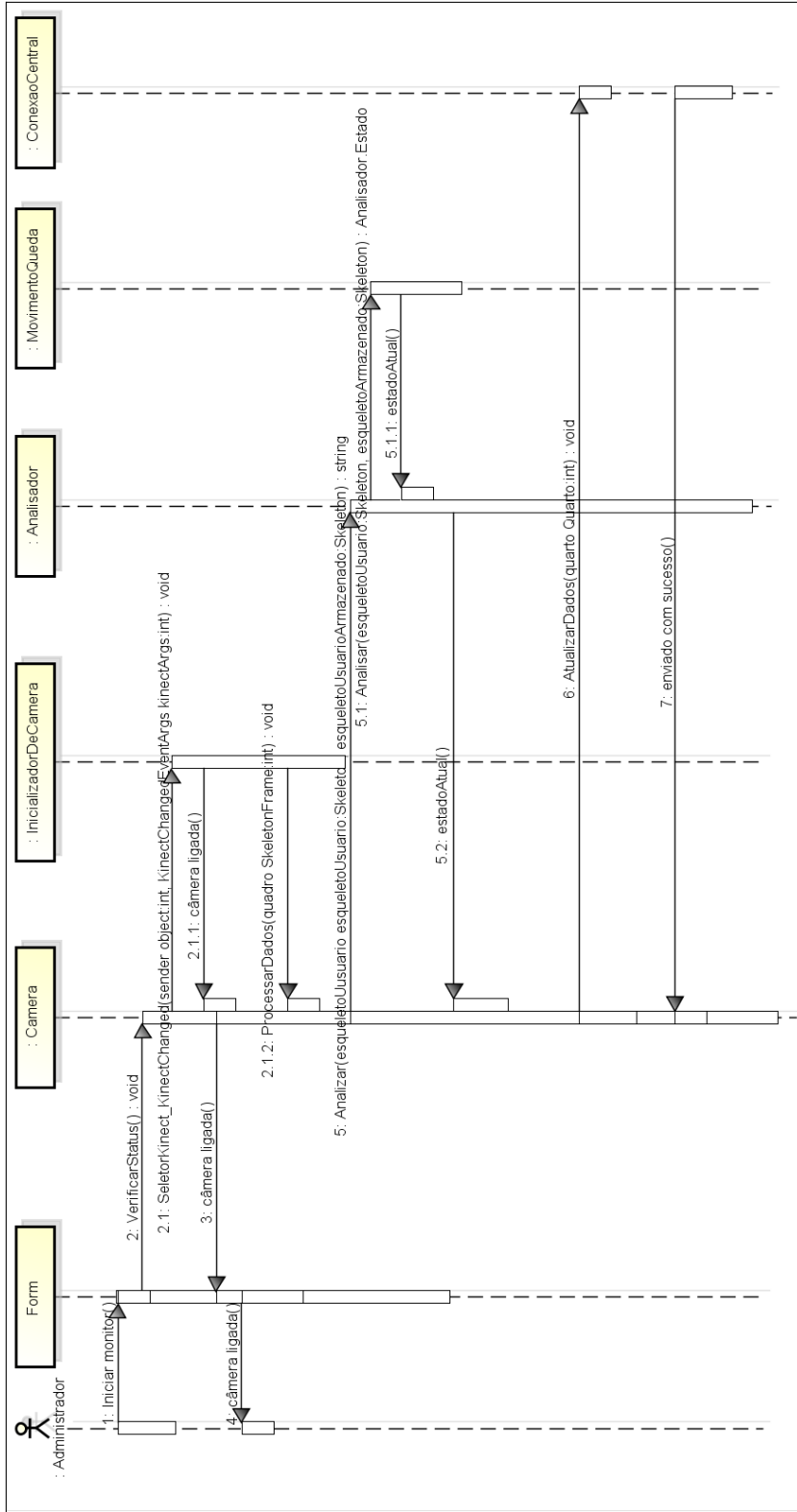
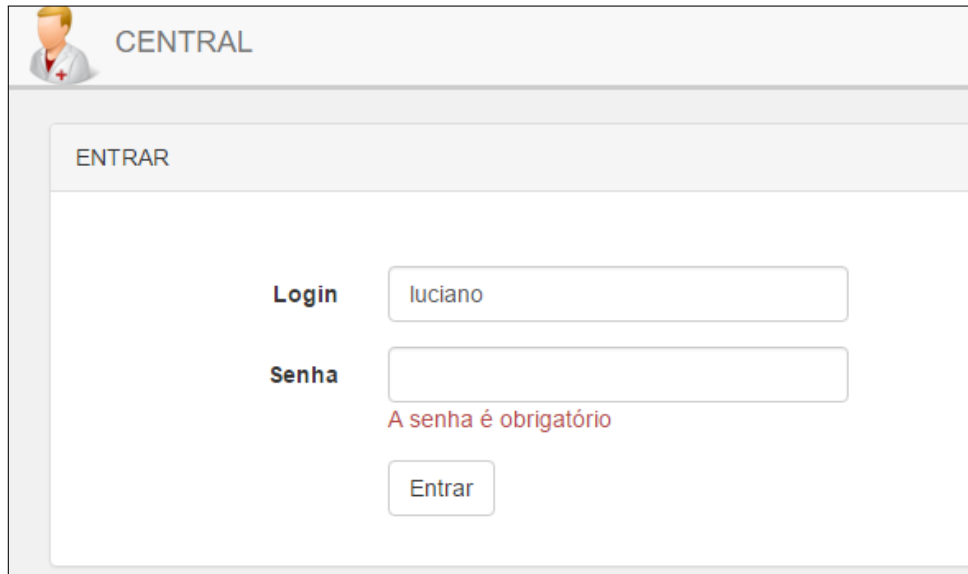


Figura 30: Diagrama de Sequência - Monitoramento.

3.3.13 TELAS DO SISTEMA

A Figura 31 mostra a tela login do sistema, no qual o cuidador poderá acessar o sistema de monitoramento. As mensagens de erros são exibidas em baixo do campo em que o erro ocorreu, os textos de erros são exibidos em cor vermelha para facilitar sua visualização. Esse padrão de mensagem de erro é utilizado em todo o sistema.



A imagem mostra a interface de login do sistema 'CENTRAL'. No topo, há um ícone de um médico e o nome 'CENTRAL'. Abaixo, o título 'ENTRAR' indica a seção de login. Há dois campos de entrada: 'Login' com o texto 'luciano' e 'Senha' que está vazio. Abaixo do campo de senha, há uma mensagem de erro em vermelho: 'A senha é obrigatório'. Um botão 'Entrar' está localizado abaixo dos campos.

Figura 31: Tela de Login.

A Figura 32 representa a tela inicial do sistema, após o usuário ter efetuado login.

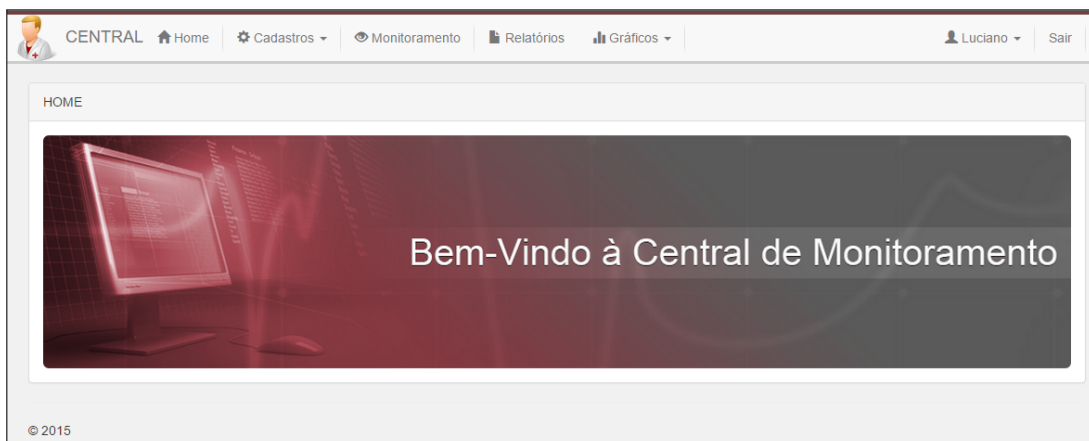


Figura 32: Tela principal da Central de Monitoramento.

A Figura 33 mostra a tela de listagem de quartos cadastrados no sistema. Por meio dessa tela é possível acessar as opções para editar e deletar os quartos, possui um botão para cadastros de novos quartos. Esse padrão de tela também é utilizado para a tela de listagem de usuários.

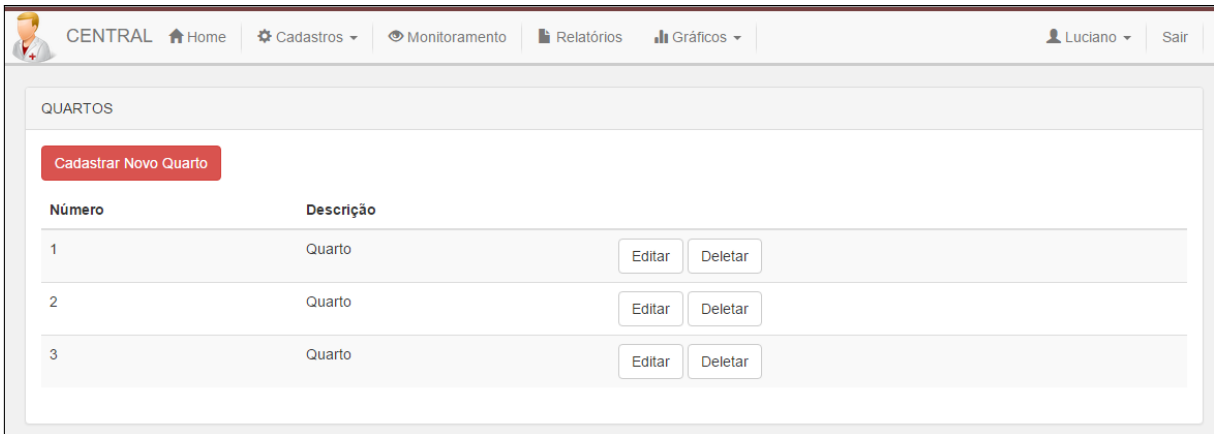


Figura 33: Tela de listagem de quartos.

A Figura 34 representa a tela de cadastro de quartos. Tal tela é usada para cadastrar os quartos que serão monitorados pelo sistema. Possui os campos número e descrição do quarto, a confirmação é feita por meio do botão gravar e o usuário poderá voltar para a tela de listagem de quartos por meio do link Voltar. Este padrão de tela também é utilizado para o cadastro de usuários.



Figura 34: Tela de cadastro de quartos.

A Figura 35 representa a tela de exclusão de quarto. Uma mensagem de aviso e os dados do quarto é exibido para o usuário. A confirmação é feita por meio do botão deletar, o usuário poderá retorna a listagem de quarto por meio do link Voltar. Tal padrão de tela também é utilizado para a exclusão de usuários.

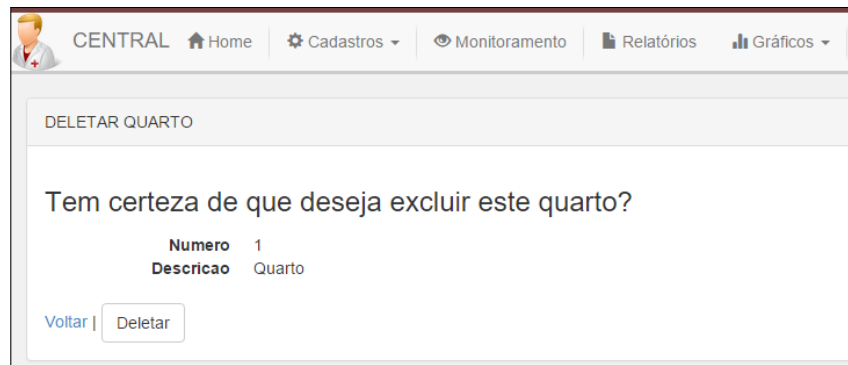


Figura 35: Tela de exclusão de quarto.

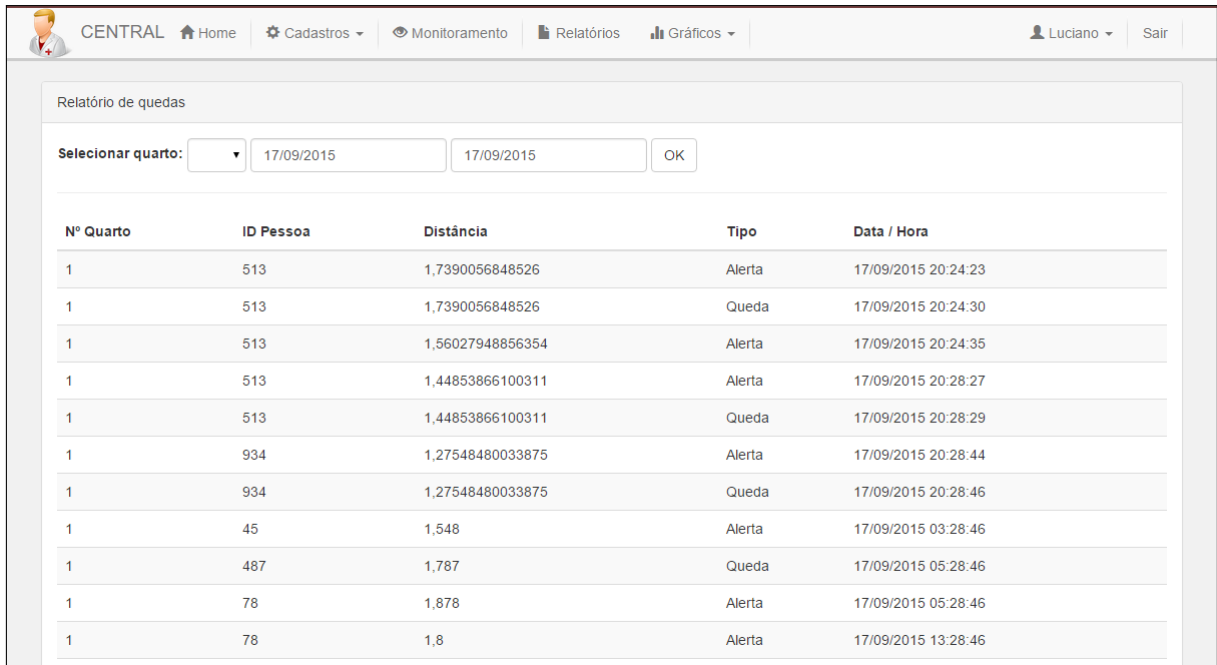
A Figura 36 mostra a tela monitoramento de quartos. Tal tela exibe em tempo real as informações dos pacientes que se encontram no quarto. Para cada quarto é exibido uma visão da posição dos pacientes em coordenadas cartesianas 2D, representados por um círculo na tela, distância em metros, situação do paciente (normal, alerta e queda) e a informação de rastreo. A informação rastreo indica se o Kinect conseguiu capturar o esqueleto do paciente. Além dos dados a central de monitoramento possui duas formas de alertar o cuidador, por cores e áudio. As cores alertam sobre o estado em que se encontra o quarto, sendo branco normal, amarelo alerta e vermelho queda. Sempre que uma queda é detectada um alerta sonoro é disparado na central de monitoramento. O alerta sonoro é essencial, visto que o cuidador às vezes tenha que sair da frente da tela do computador, desse modo é possível saber de uma queda no mesmo instante, possibilitando o socorro de forma rápida.

As telas de monitoramento são exibidas com base nos quartos cadastrados. Sempre que um cadastro é feito, uma tela referente ao quarto é exibida na tela de monitoramento.



Figura 36: Tela de monitoramento.

A Figura 37 representa a tela de relatório analítico do sistema. Por meio desse relatório o cuidador poderá verificar as quedas ocorridas em determinados momentos. A tela possui filtros por quarto e data, facilitando a busca de informações.



Nº Quarto	ID Pessoa	Distância	Tipo	Data / Hora
1	513	1,7390056848526	Alerta	17/09/2015 20:24:23
1	513	1,7390056848526	Queda	17/09/2015 20:24:30
1	513	1,56027948856354	Alerta	17/09/2015 20:24:35
1	513	1,44853866100311	Alerta	17/09/2015 20:28:27
1	513	1,44853866100311	Queda	17/09/2015 20:28:29
1	934	1,27548480033875	Alerta	17/09/2015 20:28:44
1	934	1,27548480033875	Queda	17/09/2015 20:28:46
1	45	1,548	Alerta	17/09/2015 03:28:46
1	487	1,787	Queda	17/09/2015 05:28:46
1	78	1,878	Alerta	17/09/2015 05:28:46
1	78	1,8	Alerta	17/09/2015 13:28:46

Figura 37: Relatório Analítico.

A Figura 38 representa a tela de relatório diário modo gráfico. Por meio desse gráfico o cuidador poderá acessar as informações referentes as quedas que ocorreu no dia, podendo visualizar os dados em formato de gráfico de barras ou linhas, além de um gráfico de pizza com o total de quedas ocorridas. A tela possui filtros por quarto, tipo de gráfico e data.

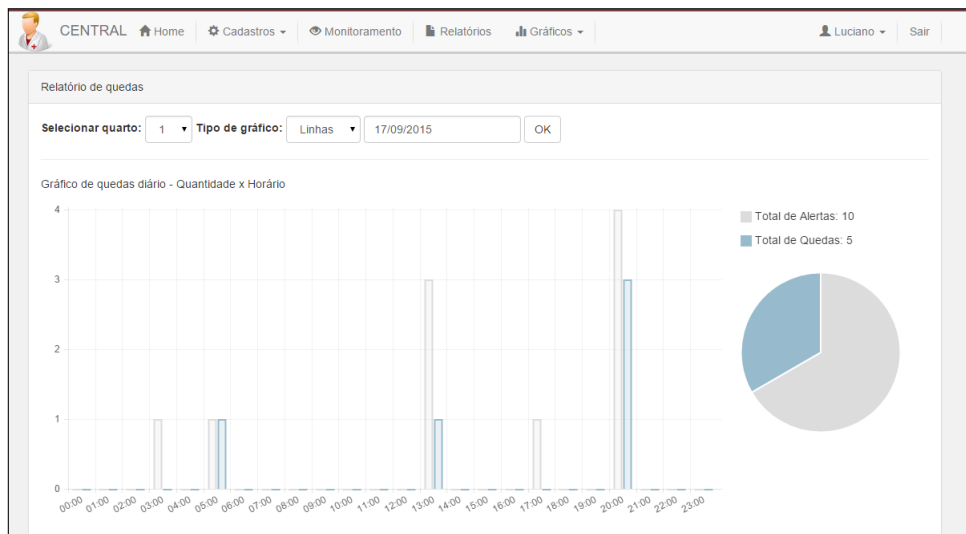


Figura 38: Relatório gráfico diário em barras

A Figura 39 representa a tela de relatório gráfico mensal em barras. Por meio desse gráfico o cuidador poderá acessar as informações referentes as quedas que ocorreu no mês, podendo visualizar os dados em formato de gráfico de barras ou linhas, além de um gráfico de pizza com o total de quedas ocorridas. A tela possui filtros por quarto, tipo de gráfico e mês.

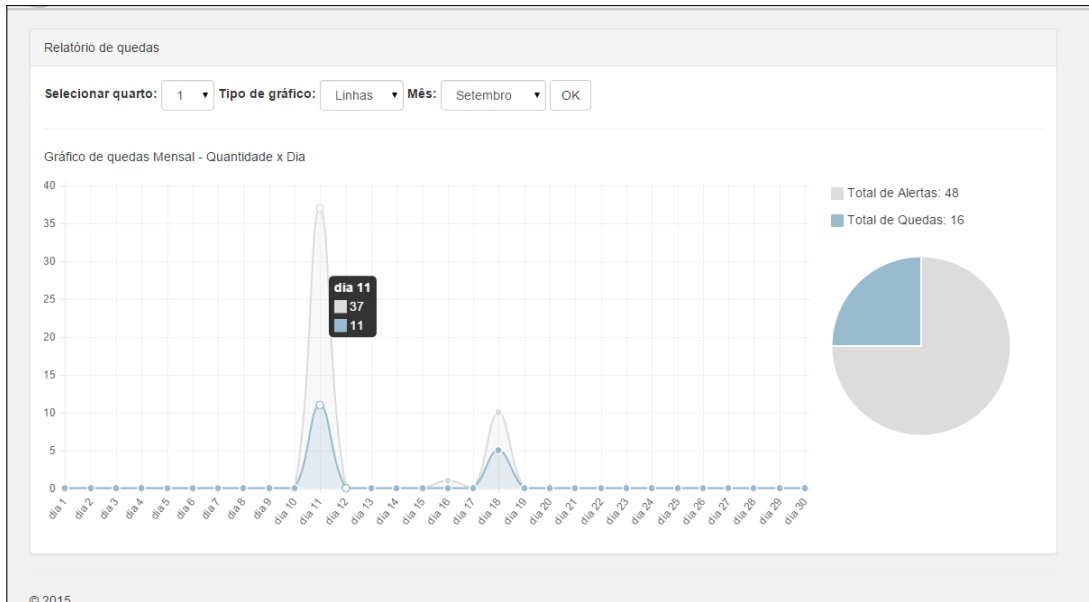


Figura 39: Relatório mensal em linhas.

A Figura 40 representa a tela de relatório gráfico anual em barras. Por meio desse gráfico o cuidador poderá acessar as informações referentes as quedas que ocorreu no ano, podendo visualizar os dados em formato de gráfico de barras ou linhas, além de um gráfico de pizza com o total de quedas ocorridas. A tela possui filtros por quarto e tipo de gráfico.

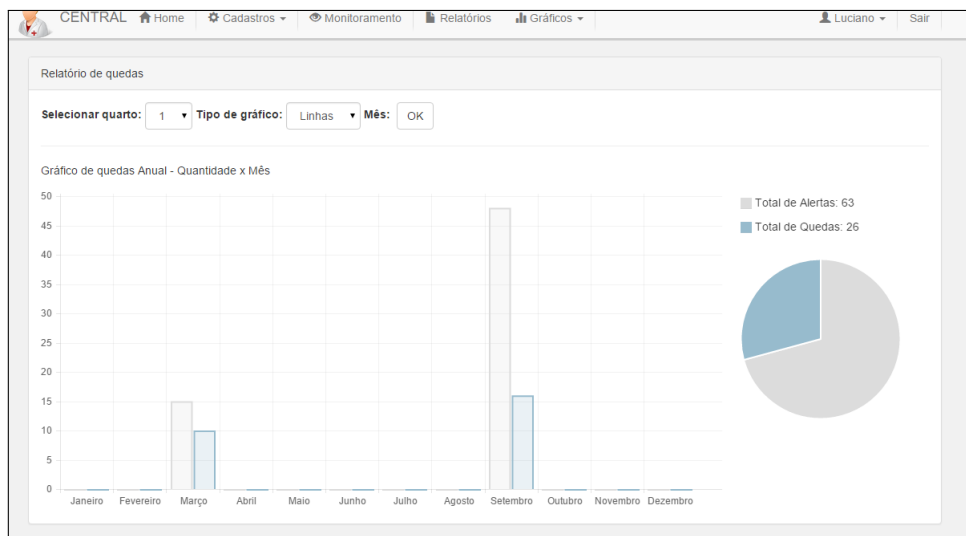


Figura 40: Relatório gráfico anual em barras

A Figura 41 representa a tela de dados do usuário. Por meio dessa tela o usuário poderá alterar seus dados cadastrais, com exceção do login.

Figura 41: Tela dados do usuário

A Figura 42 representa o menu de opções da estação remota. Tais opções são acessas por meio de um TrayIcon que fica acessível após a inicialização do sistema de monitoramento. A opção Câmera exibe a imagem que está sendo captura no momento e uma opção para ajustar o ângulo de captura do sensor Kinect. As configurações da estação são feita por meio da opção configurações e a opção sair, fecha o sistema.



Figura 42: Tela menu central de monitoramento

A Figura 42 representa a tela configurações, sendo possível configurar a qual quarto pertencem o monitor, endereço da central de monitoramento e os dados para acesso a banco de dados. O endereço para a central de monitoramento é necessário para o envio de informações em tempo real.

Figura 43: Tela de configurações

3.3.14 REGRAS DE AVALIAÇÃO DE QUEDA

Para a avaliação das quedas foram desenvolvidos algoritmos que verificam a movimentação de um esqueleto em um certo espaço de tempo. O primeiro passo foi armazenar os últimos 60 quadros fornecidos pelo kinect em um vetor e comparar a altura das articulações superiores dos esqueletos do momento atual em relação às articulações armazenadas nos momentos anteriores. Caso haja uma variação negativa (em relação ao chão) superior a 50% da altura do momento armazenado, o esqueleto é situado como estado de alerta e armazenado em uma novo vetor.

O segundo passo foi analisar os esqueletos que se encontram em estado de alerta, de modo a verificar se o movimento que ocorreu é uma queda ou um falso-positivo. Para tal, foi verificado durante 2 segundos se houve uma variação positiva superior a 10% da altura das suas articulações atual em relação à altura das suas articulações no momento em que entrou em estado de alerta. Caso não haja uma variação superior ao percentual determinado, o esqueleto analisado entra em estado de queda.

A Figura 44 exemplifica o armazenamento dos quadros em vetores e a verificação entre os dados do quadro atual e os dados dos quadros já armazenados em momentos anteriores.

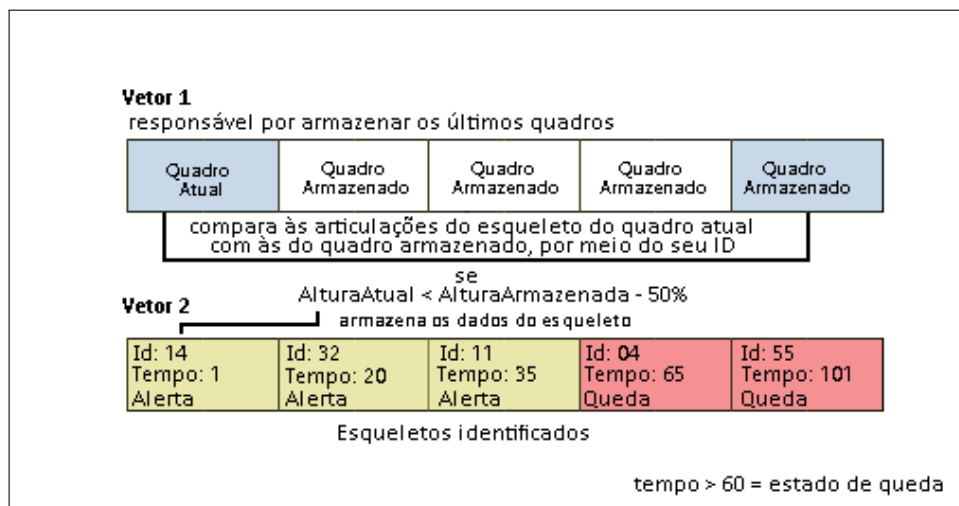


Figura 44: Armazenamento e verificação dos dados

A Figura 45 mostra um trecho de código responsável por verificar o estado do esqueleto.

```

public Analisador.Estado Analisar(Skeleton esqueletoUsuario, Skeleton esqueletoArmazenado)
{
    Analisador.Estado novoEstado = Analisador.Estado.NaoIdentificado;
    if (esqueletoUsuario != null && PosicaoValida(esqueletoUsuario, esqueletoArmazenado))
    {
        var esqueletoEncontrado = esqueletoEmQueda.Where(x => x.skeleton.TrackingId == esqueletoUsuario.TrackingId).FirstOrDefault();
        if (esqueletoEncontrado != null)
        {
            if (esqueletoEncontrado.quadroIdentificacao >= 60){
                if(esqueletoEncontrado.estado != Analisador.Estado.Identificado)
                {
                    esqueletoEncontrado.estado = Analisador.Estado.Identificado;
                    GravaDados(esqueletoEncontrado.skeleton, "Queda");
                }
                novoEstado = Analisador.Estado.Identificado;
            }
            else
            {
                novoEstado = Analisador.Estado.EmExecucao;
                if (esqueletoEncontrado.quadroIdentificacao == 1)
                {
                    GravaDados(esqueletoEncontrado.skeleton, "Alerta");
                    esqueletoEncontrado.quadroIdentificacao += 1;
                }
            }
        }
        else
        {
            novoEstado = Analisador.Estado.NaoIdentificado;
        }
        return novoEstado;
    }
}

```

Figura 45: Trecho de código - Análise do esqueleto

4 RESULTADOS E DISCUSSÕES

4.1 VALIDAÇÃO

Para verificar a eficiência do sistema foram feitas simulações em diferentes situações e ambientes, de forma a conseguir resultados mais precisos e possibilitar uma melhor análise dos dados.

A primeira simulação foi feita com um indivíduo posicionado em frente ao sensor Kinect de forma a simular quedas frontais, traseiras e laterais em ambas as direções. O Kinect foi posicionado a 2 metros do usuário e a uma altura de 1 metro em um ambiente claro. A Figura 44 exibe a simulação de uma queda lateral, os dados obtidos nessa simulação se encontram na Tabela 2.

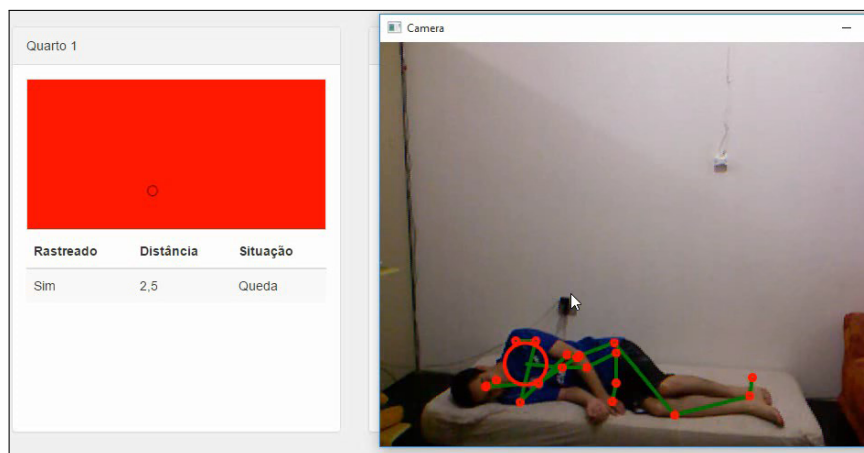


Figura 44: Queda lateral em frente ao Kinect

Tabela 2: Resultados - Quedas em frente ao Kinect

Queda	Esquerda	Direita	Frontal	Traseira
1	Detectada	Detectada	Detectada	Detectada
2	Detectada	Detectada	Detectada	Detectada
3	Detectada	Detectada	Detectada	Detectada
4	Detectada	Detectada	Detectada	Detectada
5	Detectada	Detectada	Detectada	Detectada
6	Detectada	Detectada	Detectada	Detectada
7	Detectada	Falha	Detectada	Detectada
8	Detectada	Detectada	Detectada	Detectada
9	Falha	Detectada	Falha	Detectada
10	Falha	Detectada	Falha	Falha

O segundo teste foi feito com uma pessoa em movimentação, simulando uma queda ao entrar em um quarto com a luz apagada. O Kinect foi posicionado a uma distância de três metros da porta e a uma altura de um metro e meio. A Figura 45 mostra o momento em que ocorre uma queda no escuro. Os dados referentes ao teste se encontram na Tabela 3.

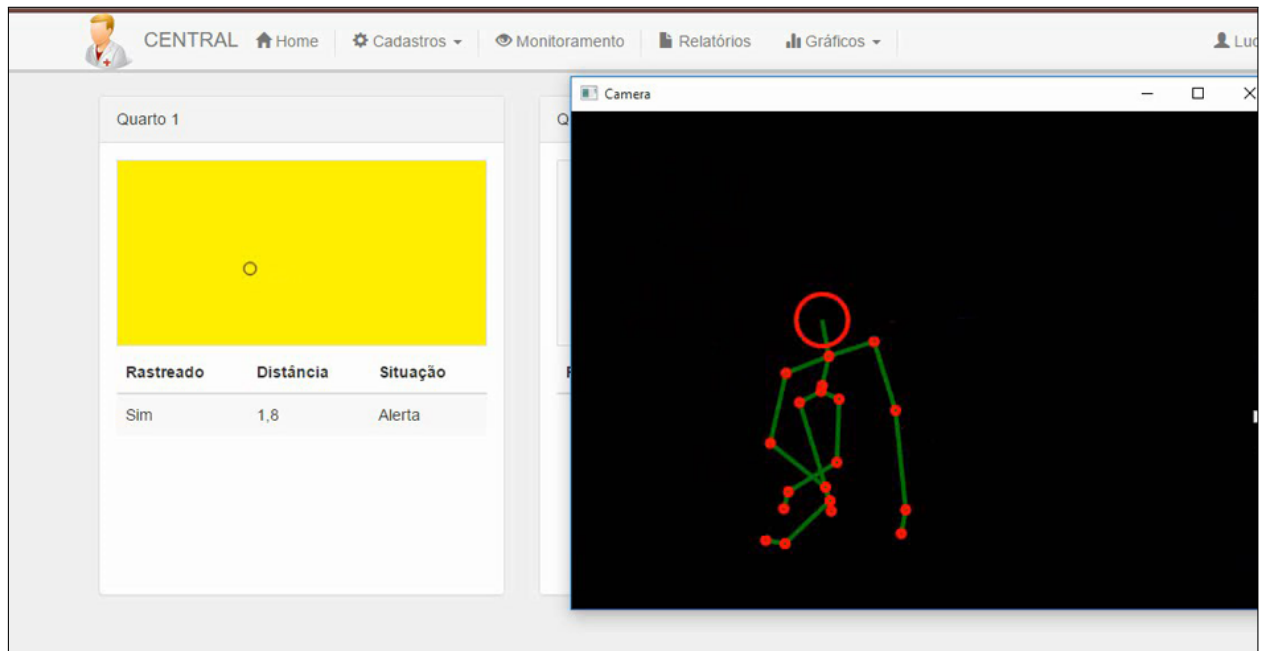
**Figura 45: Queda em movimentação na frente do Kinect**

Tabela 3: Resultados - Queda em movimentação na frente do Kinect

Queda	Resultado
1	Detectada
2	Detectada
3	Detectada
4	Detectada
5	Detectada
6	Detectada
7	Detectada
8	Detectada
9	Detectada
10	Falha

No terceiro teste foi simulado quedas ao se deitar e levantar da cama em um ambiente escuro. Nesse caso o Kinect foi posicionado a um metro de distância da cama e a uma altura de 2 metros para conseguir capturar a pessoa deitada. Os resultados obtidos são mostrados na Tabela 4. A Figura 46 exibe a captura de uma pessoa deitada.

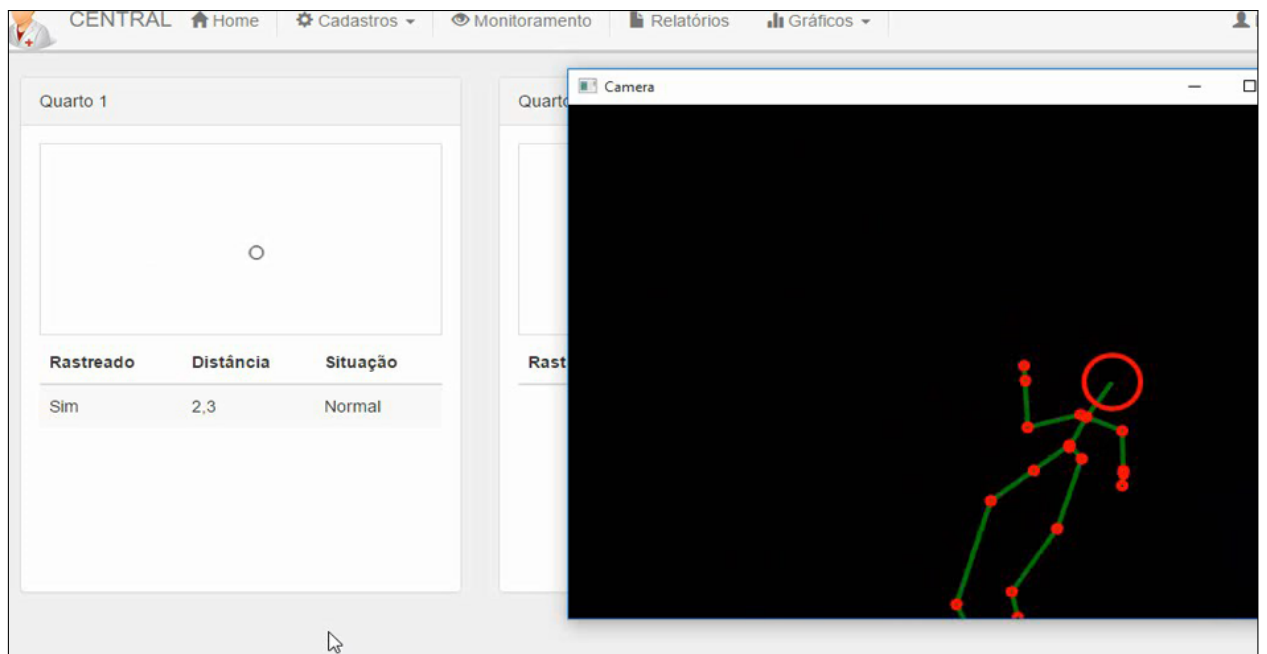
**Figura 46: Pessoa deitada capturada pelo Kinect**

Tabela 4: Resultados - Quedas ao deitar e levar da cama

Queda	Deitar	Levantar
1	Falha	Falha
2	Detectada	Falha
3	Falha	Falha
4	Falha	Detectada
5	Detectada	Falha
6	Falha	Falha
7	Falha	Falha
8	Detectada	Falha
9	Falha	Falha
10	Falha	Falha

Na última simulação foi criado uma situação em que uma pessoa está caída e uma segunda tenta fazer o socorro, porém a mesma também cai. O Kinect foi posicionado a uma distância de dois metros e meio, e a uma altura de um metro e meio. A Figura 47 mostra o momento em que a segunda pessoa tenta fazer o socorro. Os resultados são apresentados na Tabela 5.

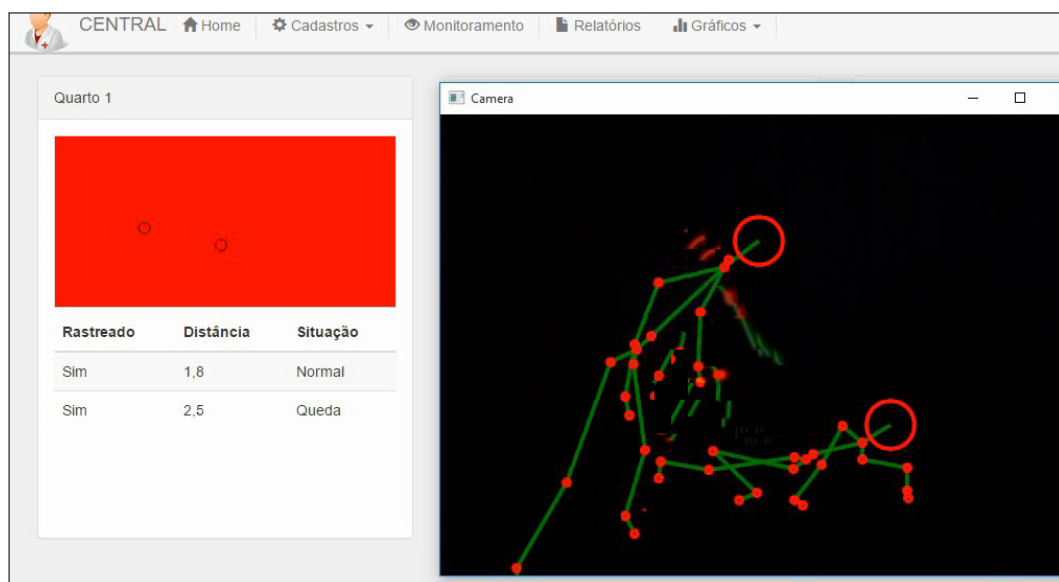
**Figura 47: Captura de duas pessoas pelo sensor**

Tabela 5: Resultados - Pessoa caindo ao prestar socorro

Queda	Resultado
1	Falha
2	Falha
3	Falha
4	Falha
5	Falha
6	Falha
7	Falha
8	Falha
9	Falha
10	Falha

4.2 ANÁLISE DOS RESULTADOS

Podemos observar com base nos dados das Tabelas 2, 3 e 4 que o Kinect se comportou melhor a uma altura aproximadamente entre um metro a um metro e meio. A melhor posição obtida foi a frontal, com a pessoa se movendo em direção as lentes do sensor, conforme a Tabela 3. A maioria das falhas ocorrem devido a uma perda aleatória dos pontos do esqueleto e a uma limitação de capacidade do Kinect, no qual o mesmo não consegue plotar corretamente o esqueleto em certas situações devido a diversos fatores externo, como ângulo, quantidade de objetos encontrado no campo de visão, posicionamento do esqueleto entre outros. Essas falhas ficaram evidentes nos dados das Tabelas 4 e 5 no qual o sensor foi submetido a um teste com algumas das situações mencionadas.

O Kinect apresentou uma falha ocasional que ocorre quando a pessoa rastreada pelo sensor sai do seu campo de visão, ocasionando na perda da referência do esqueleto e o mesmo acaba sendo plotado em um lugar aleatório, devido as regras definidas o sistema detecta uma sinal de alerta indevidamente, porém uma queda não é acionada, amenizando um pouco o problema.

Outro dado importante foi a capacidade do Kinect detectar o esqueleto no escuro, o mesmo não mostrou um diferença significante em relação a esse fator, conforme as Tabelas 2 e 3.

O Kinect também conseguiu capturar mais de uma pessoa ao mesmo tempo conforme a Figura 47, porém devido a suas limitações mencionadas o mesmo não conseguiu plotar duas

peças juntas e caídas ao mesmo tempo, ocasionando muitas vezes em perda do esqueleto ou esqueletos fantasmas.

Para a realização dos testes do presente trabalho foi utilizado a primeira versão do Sensor Kinect, acredita-se que nova versão do dispositivo lançado para o console Xbox One, apresentaria um algoritmo mais eficiente, diminuindo das limitações mencionadas. Os parâmetros utilizados foram: altura, distância, tempo e número de pessoas monitoradas (1 e 2 pessoas).

5 CRONOGRAMA

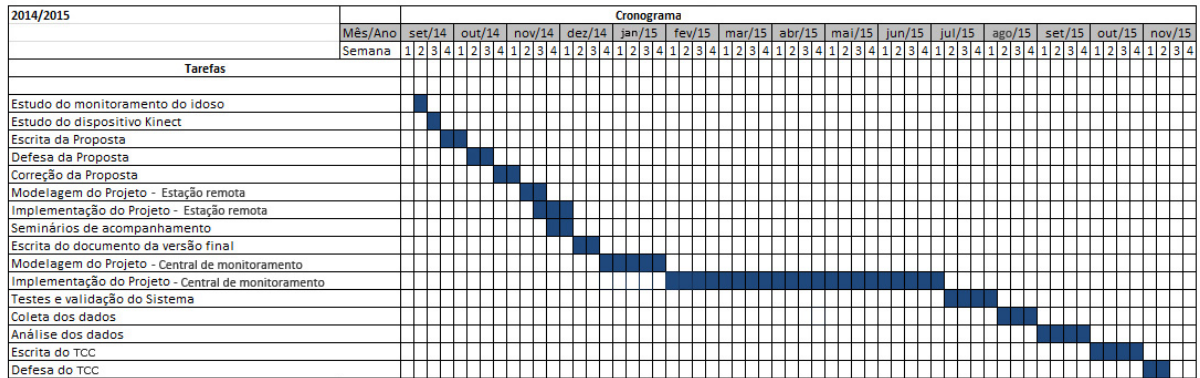


Figura 48: Cronograma

6 CONCLUSÃO

O propósito do presente trabalho foi desenvolver um sistema de monitoramento de idosos institucionalizado utilizando o sensor Kinect, com foco no pronto atendimento em caso de quedas ou acidentes corriqueiros de modo não invasivo ao idoso e muito menos intrusivo em sua privacidade. Concluímos que os resultados obtidos foram positivos, uma vez que o sistema desenvolvido foi capaz de capturar e analisar os dados para a verificação de quedas.

Os resultados mostraram que é possível utilizar o Kinect como dispositivo para monitoramento, obtendo bons resultados ao monitorar uma pessoa de modo frontal a câmera.

A metodologia aplicada no presente trabalho utilizou conceitos de engenharia de software. As linguagens de programação utilizadas foram C# e Javascript em conjunto com os outros *frameworks* como ASP.NET MVC e SignalR.

Este trabalho foi de grande importância, uma vez que foi possível contribuir para a prestação de auxílio e socorro do idoso institucionalizado, além de poder compreender o funcionamento do sensor Kinect e ter permitido a utilização do dispositivo para outros meios fora da área de entretenimento.

Embora o trabalho tenha demonstrado que é possível utilizar o Kinect como uma solução para monitoramento, existem algumas limitações do dispositivo, como ao plotar o esqueleto de duas pessoas que estão muito próximas uma da outra ou quando há objetos que podem interferir no rastreamento.

O presente trabalho apenas analisou algumas situações iniciais, ficando para trabalhos futuros, testes com parâmetros mais precisos, bem como testes de falso-positivo. Sendo assim, o mesmo pode ser aprimorado de várias formas, desenvolvendo regras mais eficientes e extraindo mais informações de padrões de quedas, além da utilização de uma versão mais aperfeiçoada do sensor, por exemplo a versão do Xbox One, que possui algoritmos e hardware mais avançado.

Para trabalhos futuros destaca-se a importância da utilização de mais de um sensor de monitoramento. Uma vez que os dados obtidos neste estudo foram decorrentes da utilização

de apenas um sensor. A utilização de vários sensores poderia melhorar quanto a ocorrência de falso-positivo e problemas que envolve o campo de visão do sensor. Outra abordagem que poderia ser interessante, é a utilização do dispositivo de áudio acoplado do Sensor Kinect, tal recurso poderia melhorar o algoritmo quanto as ocorrências de falso-positivo e possibilitar interação com a central de monitoramento por meio de comando por voz. No entanto tais implementações seriam mais complexas, sendo estas propostas adequadas para futuras pesquisas.

REFERÊNCIAS

- ANDERSON, D. et al. Linguistic summarization of video for fall detection using voxel person and fuzzy logic. **Comput Vis Image Underst**, v. 113, p. 80–89, 2009.
- BOEHM, B.; EGYED, A. Software requirements negotiation: Some lessons learned. In: **Proceedings of the 20th International Conference on Software Engineering**. Washington, DC, USA: IEEE Computer Society, 1998. (ICSE '98), p. 503–506. ISBN 0-8186-8368-6.
- BOEHM, B. et al. Using the winwin spiral model: A case study. **Case Study, Computer**, v. 31, p. 33–44, 1998.
- BRASIL, S. d. D. H. d. P. d. R. **Dados sobre o envelhecimento no Brasil**. [S.l.]: Secretaria de Direitos Humanos da Presidência da República, 2014.
- CARDOSO, G. S.; SCHMIDT, A. E. F. Biblioteca de funções para utilização do kinect em jogos eletrônicos e aplicações nui. SBGames, 2012.
- CARVALHO, A. d. M.; COUTINHO, E. d. S. F. Demência como fator de risco para fraturas graves em idosos. **Revista de Saúde Pública**, scielo, v. 36, p. 448 – 454, 08 2002. ISSN 0034-8910.
- DAMASCENO, E.; GOULART, E.; MORAES, R. Um sistema de monitoramento para idosos utilizando técnicas de visão computacional. **XXXIII Congresso da Sociedade Brasileira de Computação**, 2013.
- DIRACO, G.; LEONE, A.; SICILIANO, P. An active vision system for fall detection and posture recognition in elderly healthcare. **Conference & Exhibition: Design, Automation & Test in Europe**, Dresden: European Design and Automation Association, p. 1536–1541, 2010.
- DUCA, G. F. D.; ANTES, D. L.; HALLAL, P. C. Quedas e fraturas entre residentes de instituições de longa permanência para idosos. **Revista Brasileira de Epidemiologia**, scielo, v. 16, p. 68 – 76, 03 2013. ISSN 1415-790X.
- FERNANDEZ, D. **Installing and Using the Kinect Sensor (Beta 2 SDK)**. 2012. Disponível em: <<http://channel9.msdn.com/Series/KinectSDKQuickstarts/Understanding-Kinect-Hardware>>. Acesso em: 20 de novembro de 2014.
- GALVEIA, B. M. G. D. Um sistema de monitoramento para idosos utilizando técnicas de visão computacional. 2014.
- HAN, J. et al. Enhanced computer vision with microsoft kinect sensor: A review. IEEE, 2013.
- IGUAL, R.; MEDRANO, C.; PLAZA, I. Challenges, issues and trends in fall detection systems. **BioMedical Engineering OnLine**, v. 12, n. 1, p. 66, 2013. ISSN 1475-925X.
- ISLAM, M. et al. A novel approach for constructing emulator for microsoft kinect xbox 360 sensor in the .net platform. IEEE, 2013.

- LIU, C.; LEE, C.; LIN, P. A fall detection system using k-nearest neighbor classifier. **Expert Syst Appl**, v. 37, p. 7174–7181, 2010.
- LOJUDICE, D. C. et al. Quedas de Idosos Institucionalizados: Ocorrência e Fatores Associados. **Revista Brasileira de Geriatria e Gerontologia**, unati, v. 13, p. 403 – 412, 00 2010. ISSN 1809-9823.
- MICROSOFT. Kinect for windows sdk v1.8. 2014.
- MICROSOFT. **Learn About ASP.NET MVC**. 2015. Disponível em: <<http://www.asp.net/mvc>>. Acesso em: 08 de outubro de 2015.
- MICROSOFT. **Learn About ASP.NET SignalR**. 2015. Disponível em: <<http://www.asp.net/signalr>>. Acesso em: 08 de outubro de 2015.
- MIRA, N. V. de; MARQUEZ, U. M. L. Importância do diagnóstico e tratamento da fenilcetonúria. **Revista de Saúde Pública**, SciELO Brasil, v. 34, n. 1, p. 86–96, 2000.
- MSDN. **Kinect Sensor**. 2012. Disponível em: <<http://msdn.microsoft.com/en-us/library/hh438998.aspx>>. Acesso em: 24 de setembro de 2014.
- MSDN. **Microsoft SQL Server**. 2014. Disponível em: <<http://msdn.microsoft.com/pt-br/library/bb545450.aspx>>. Acesso em: 24 de setembro de 2014.
- MSDN. **Introdução ao WPF no Visual Studio 2015**. 2015. Disponível em: <<https://msdn.microsoft.com/pt-br/library/aa970268>>. Acesso em: 08 de outubro de 2015.
- MSDN. **Visão geral e breve análise do ADO.NET Entity Framework**. 2015. Disponível em: <<https://msdn.microsoft.com/pt-br/data/aa937709.aspx>>. Acesso em: 08 de outubro de 2015.
- NETO, J. D. et al. Incentivo ao desenvolvimento e pesquisa em robótica educacional utilizando o kinect sensor. COBENGE, 2014.
- NOVO, A. et al. Capacidade funcional e risco de queda - aptidão física, composição corporal e medo de cair em idosos institucionalizados. **III Seminário Contributos para a Saúde na População Sénior + Idade + Saúde**, 06 2011.
- RIMMINEN, H. et al. Detection of falls among the elderly by a floor sensor using the electric near field. **IEEE Trans Inf Technol Biomed**, v. 14, p. 1475–1476, 2010.
- SERRA, R. J. M. et al. Interfaces tácteis baseadas em html5/css3/javascript. 2012.
- SILVA, A. P. G. da; SALVADOR, M. O que são sistemas supervisórios? 2005.
- SILVA, M. S. **Criando sites com HTML: sites de alta qualidade com HTML e CSS**. [S.l.]: Novatec Editora, 2008.
- SILVA, M. S. Javascript–guia do programador. **São Paulo: Novatec Editora**, 2010.
- SILVA, V. M. da. Revisão sistemática da evolução mvc na base acm. 2012.
- TZENG, H.; CHEN, M.; CHEN, J. Design of fall detection system with floor pressure and infrared image. **Proceedings of the International Conference on System Science and Engineering**, Taipei: Institute of Electrical and Electronics Engineers, p. 131–135, 2010.

UGULINO, W. et al. Virtual caregiver: A system for supporting collaboration in elderly monitoring. In: **SBSC**. [S.l.]: IEEE Computer Society, 2012. p. 43–48. ISBN 978-1-4673-4696-2.