

# MINISTÉRIO DA EDUCAÇÃO

### UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

#### CAMPUS CORNÉLIO PROCÓPIO

**DEPARTAMENTO DE ANÁLISE E DESENVOLVIMENTO**

**DE SISTEMA**



**LEONARDO TEIXEIRA VIRGILIO**

**SISTEMA DE ROTAS DE ÔNIBUS UTILIZANDO O ALGORTIMO DE MELHOR CAMINHO**

**TRABALHO DE DIPLOMAÇÃO**

**CORNÉLIO PROCÓPIO**

**2014**

**LEONARDO TEIXEIRA VIRGILIO**

**SISTEMA DE ROTAS DE ÔNIBUS UTILIZANDO O ALGORITMO DE MELHOR CAMINHO**

Trabalho de conclusão de curso de graduação, apresentado à disciplina Trabalho de Diplomação, do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Coordenação de Informática – COINF – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Danilo Sipoli Sanches

**CORNÉLIO PROCÓPIO**

**2014**

**LEONARDO TEIXEIRA VIRGILIO**

**SISTEMA DE ROTAS DE ÔNIBUS UTILIZANDO O ALGORITMO DE MELHOR CAMINHO**

Trabalho de conclusão de curso de graduação, apresentado à disciplina Trabalho de Diplomação, do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Coordenação de Informática – COINF – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Tecnólogo.

Cornélio Procópio, 20 de novembro de 2014

BANCA EXAMINADORA

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

Prof.

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

Prof.

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

Prof.

“Quanto mais aumenta nosso conhecimento, mais evidente fica nossa ignorância.”

(KENNEDY, John F.)

**DEDICATÓRIA**

Aos meus familiares, Luciana (mãe), Lafaeti (pai) e Dominique (irmã), também dedico a Felipe Israel Marinho por ceder seu periférico, ao meu orientador pela paciência e por acreditar em mim, Prof. Dr. Eduardo, também gostaria de dedicar aos meus melhores amigos, dos quais sabem eles quem são, juntamente para as pessoas demasiadamente importante para mim e que me acompanharam neste trajeto de enriquecimento profissional e sobre tudo pessoal, pessoas estas que teria o prazer de estar por perto sempre. Obrigado.

**RESUMO**

Pedro, Daniel Longhi Fernandes. Sistema para rotas ônibus utilizando o algoritmo de Dijkstra (Melhor Caminho). 38 páginas. Tema de Trabalho de Diplomação (Tecnologia em Análise e Desenvolvimento de Sistemas). Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014. Este trabalho propõe o desenvolvimento de um sistema do qual seja capaz de reconhecer e mapear o caminho com o menor custo, bem como inserir, alterar e excluir os caminhos armazenados bem como desenvolver um aplicativo para a plataforma Android que irá apresentar os dados do caminho com menor custo. Neste trabalho será utilizada a linguagem de programação Java, PHP com IDE Netbeans, Eclipse, banco de dados MySQL e Scrum como metodologia para gerenciamento de cronograma.

Palavras-chave: Algoritmo de Dijkstra, Android, Algoritmo Melhor Caminho.

.

***ABSTRACT***

Pedro, Daniel Longhi Fernandes. System of bus routes using Dijkstra's algorithm (the Best Way). 38 pages. Graduation's work theme (Technology Analysis and Development of Systems). Federal Technological University of Paraná. Cornélio Procópio,2014. This work proposes the development of a system which can be able to recognize and map the course with a lower cost, as well as inserting, changing and deleting stored routes, and also the development of an application for the Adroid Platform that will present the data of a route with the lowest cost. In this work will be used Java programming language, PHD with Netbeans IDE, Eclipse, MySQL database and Scrum as methodology for schedule management.

Keywords: Dijkstra's algorithm, Android, Algorithm the Best Way.

**LISTA DE FIGURAS**

[FIGURA 1 - PSEUDOCÓDIGO DA IMPLEMENTAÇÃO...........................................................................19](#_Toc390011014)

[FIGURA 2 - 60 SEGUNDOS DE CAPTAÇÃO.........................................................................................19](#_Toc390011014)

[FIGURA 3 - 30 SEGUNDOS DE CAPTAÇÃO..........................................................................................19](#_Toc390011015)

[FIGURA 4 - 30 SEGUNDOS DE CAPTAÇÃO..........................................................................................20](#_Toc390011014)

[FIGURA 5 - 60 SEGUNDOS DE CAPTAÇÃO...........................................................................................20](#_Toc390011014)

[FIGURA 6 - 30 SEGUNDOS DE CAPTAÇÃO.........................................................................................21](#_Toc390011014)

[FIGURA 7 - 60 SEGUNDOS DE CAPTAÇÃO...........................................................................................22](#_Toc390011014)

[FIGURA 8 - 30 SEGUNDOS DE CAPTAÇÃO...........................................................................................23](#_Toc390011014)

[FIGURA 9 - 30 SEGUNDOS DE CAPTAÇÃO...........................................................................................23](#_Toc390011014)

[FIGURA 10 - 60 SEGUNDOS DE CAPTAÇÃO.........................................................................................24](#_Toc390011014)

[FIGURA 11 - 30 SEGUNDOS ................................................................................................................24](#_Toc390011014)

[FIGURA 12 - MÉTODO SCRUM................................................................................................................33](#_Toc390011014)

**LISTA DE TABELAS**

[Tabela de Funcionalidades do Sistema.....................................................................................2](#_Toc390011014)5

[Cronograma..........................................................................................................................................34](#_Toc390011015)

**LISTA DE ABREVIATURAS**

|  |  |
| --- | --- |
| API | Application Programming Interface |
| C# | C Sharp |
| C++  CRUD | C mais mais  Create, Read, Update, Delete |
| DB4O | DataBase for Objects |
| IDE | IntegratedDevelopmentEnvironment |
| MSDN  SQL | Microsoft Developer Network  Structured Query Language |
| NoSql | Not Only Structured Query Language |
| UTFPR  TD  DB | Universidade Tecnologica Federal do Paraná  Trabalho de Diplomação  Database |
| BIN | Binário |

FPS Frames per Seconds

OPS Objetos por Segundos

SUMÁRIO

[1 INTRODUÇÃO 12](#_Toc390011014)

[2 JUSTIFICATIVA 14](#_Toc390011015)

[3 OBJETIVOS 15](#_Toc390011016)

[3.1 Objetivo Geral.....................................................................................................................15](#_Toc390011017)

[3.2 Objetivo Específico............................................................................................................15](#_Toc390011018)

[4 VISÃO GERAL DO TRABALHO DE CONCLUSÃO 18](#_Toc390011019)

[4.1 QT...............................................................................................................................................18](#_Toc390011020)

[4.2 OPS............................................................................................................................................22](#_Toc390011021)

[5 ESCOPO DA APLICAÇÃO 25](#_Toc390011022)

[5.1 Definição de escopo..........................................................................................................25](#_Toc390011023)

[6 DETALHES E LIMITAÇÕES DA APLICAÇÃO 27](#_Toc390011024)

[6.1 Detalhes da aplicação....................................................................................................27](#_Toc390011025)

[6.2 Limitações de Hardware e SoftwarE........................................................................27](#_Toc390011026)

[6.2.1 Hardware.............................................................................................................27](#_Toc390011027)

[6.2.2 Software..............................................................................................................27](#_Toc390011028)

[7 LINGUAGENS E FERRAMENTAS UTILIZADAS 28](#_Toc390011029)

[7.1 Linguagens...........................................................................................................................28](#_Toc390011030)

[7.1.1 Linguagem de programação C#..................................................................28](#_Toc390011031)

[7.2 Ferramentas.......................................................................................................................29](#_Toc390011032)

[7.2.1 Visual Studio......................................................................................................29](#_Toc390011033)

[7.2.2 DB4O.......................................................................................................................29](#_Toc390011034)

[7.2.3 PostgreSQL........................................................................................................30](#_Toc390011035)

[7.2.4 Disco ......................... ...........................................................................................30](#_Toc390011036)

[7.2.5 Microsoft Kinect .............................................................................................31](#_Toc390011037)

[8 METODOLOGIA DE DESENVOLVIMENTO 32](#_Toc390011038)

[9 CRONOGRAMA 34](#_Toc390011039)

[10 REFERÊNCIAS 35](#_Toc390011040)

[11 REFERÊNCIAS 37](#_Toc390011040)

# 1 INTRODUÇÃO

Computador, uma máquina capaz de fazer vários cálculos e nos dar uma resposta satisfatória em relação ao nosso problema; calma, mesmo sendo fantástico seu funcionamento e a maneira como tudo ocorre, anos para se chegar a maneira que é hoje. Ao longo dos anos nem tudo era a maravilha com que vemos hoje, os seus primórdios não tínhamos sequer um que se alocasse em nosso quarto, quanto mais essas interfaces com lindas transições.

Com a evolução da sociedade temos também a evolução destas tecnologias, que afinal nada mais são do que utensílios necessários para nos auxiliar em nosso dia-a-dia; destas tecnologias temos convívio com enumeras delas. São elas desde coisas simples como um pequeno despertador podendo até mesmo chegar a, por exemplo: um computador e seus *softwares* e *hardwares,* do qual, diga-se de passagem, é indispensável nos dias de hoje.

Ainda sim, sendo o computador uma ferramenta muito necessária, em algumas áreas ou tipos de trabalhos; seriam impossíveis sem computadores, nele podemos aprender, ensinar; em poucas palavras, nos informar do que ocorre no mundo todo, não há limites, não há barreiras, contudo temos que ter bom senso de uso e manipulação deste poderoso objeto.

Segundo Costa e Oliveira (2004), o desenvolvimento da informática se iniciou a priori com o emprego de computadores de grande porte, que eram instalado em salas especiais e operado por profissionais altamente qualificados. A partir de então os computadores são utilizados de forma mais intensa, nesse período intensificou-se os avanços e a internet entra no campo acadêmico contribuindo para o desenvolvimento do conhecimento científico.

Nesta frenética evolução temos o aparecimento de periféricos aplicados nas mais diversas áreas. Dentre elas, a área de Jogos e Entretenimento Digital tem tido um grande crescimento na sociedade. Esta área tem ultrapassado as fronteiras apenas dos “joguinhos” e está cada vez mais sendo aplicado para outras áreas como educação, saúde ou afins. Nesse contexto, os periféricos contribuem para interação homem-máquina em jogos digitais. Por exemplo, o Kinect – Uma plataforma desenvolvida pela Microsoft, da qual possui sensores que captam a movimentação do esqueleto humano ali identificado - tem sido comercializado para jogos, mas sua *API* de desenvolvimento nos possibilita o uso imensurável deste periférico para outros fins.

# 2 JUSTIFICATIVA

Hoje em dia com o Sistema de Seleção Unificada (SISU), sistema informatizado gerenciado pelo Ministério da Educação no qual instituições públicas de ensino superior oferecem vagas para candidatos participantes do Exame Nacional de Ensino Médio (Enem). Está cada vez mais comum a migração de estudantes para diversos lugares do Brasil. Muitos alunos estudam em universidades que ficam localizadas a mais de mil quilômetros de distância de sua residência. Essa distância, de certa forma é um grande problema que o estudante enfrenta ao sair de casa, pois na maioria dos casos eles dependem do transporte coletivo para deslocar-se de sua cidade natal para a cidade em que irá estudar.

A partir deste problema, foi realizado entrevistas com alunos da Universidade Tecnológica Federal do Paraná do Campus Cornélio Procópio (UTFPR-CP) e muitos deles citaram a dificuldade em encontrar rotas de ônibus para chegar ou sair de Cornélio Procópio, pois há poucas empresas que oferecem esse serviço para destinos com grandes distâncias em Cornélio Procópio.

Pensando nessa necessidade de que o estudante tem em planejar toda a logística de sua viagem foi que surgiu a ideia do aplicativo para a plataforma Android que propõem minimizar esse problema.

Um pequeno exemplo para demonstrar o problema:

Se um estudante precisar ir para Presidente Prudente – SP, cerca de duzentos quilômetros de distância, ele precisa de ir para Londrina – PR, Assis – SP ou Ourinhos – SP, a cidade de Cornélio Procópio não possui uma rota/linha do ônibus que faz esse caminho diretamente.

# 3 OBJETIVOS

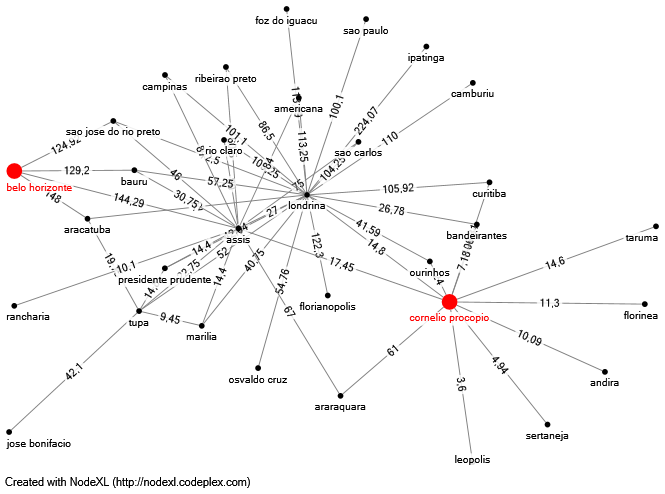
# 3.1 Objetivo Geral

O objetivo geral do trabalho é desenvolver um sistema para estudantes ou passageiros em geral que dependem do transporte coletivo para se locomoverem entre uma cidade e outra.

O sistema funcionará da seguinte maneira:

Uma pessoa que precisa viajar de Cornélio Procópio – PR para a cidade de Belo Horizonte – MG a distância entre as duas cidades é de cerca 1034 KM (Google Maps) dependendo do caminho, essa pessoa leva o tempo de mais o menos 24 horas de viajem e gasta cerca de R$ 300,00 com passagens. Com o aplicativo, esse viajante poderá pesquisar caminhos mais baratos ou que demorem menos tempo para chegar em seu destino.

Esta situação é um problema comum que estudantes sofrem ao ir morar longe de casa e acontece frequentemente quando a cidade de origem e destino são cidades pequenas, pois há poucas rotas de ônibus e o tempo que se gasta é elevado para chegarem ao seu destino.



# 3.2 Objetivo Específico

A identificação e informações dos dados obtidos nos movimentos serão gravadas em bancos de dados para que nenhumas das informações sejam perdidas ou que possíveis simulações futuras possam ser executadas.

O módulo de obtenção destas informações que foram gravadas poderá servir para simulação e estudo de futuros exercícios, podendo envolver áreas da saúde, entretenimento, análise e pesquisa através desde projeto; para uma pesquisa mais aprofundada e com valores agregados também fizemos três tipos de gravação, sendo elas em métodos e processos diferentes, como: Banco Orientado a Objetos (BD4o), armazenamento em um Banco de Dados Relacional (Postgres) e armazenamento em disco através de conversão do objeto em arquivos binários (BIN), dos quais todas essas formas de dados armazenados garantam a integridade dos dados e que seja passível de resgate para reprodução dos movimentos captados.

Foram realizados testes semelhantes a todos os tipos de armazenamento, fortalecendo então nossa expectativa e confirmando no quesito desempenho para *insert* dos dados que foram sendo obtidos no momento da captação dos mesmos, na implementação do código houve apenas a modificação nas partes pertinentes à especificação e implementação específica de cada forma de armazenamento, abaixo:

1 - Inicio Aplicação

2 - Reconhecimento esqueleto

3 - Inicio da gravação

4 - Posterior reprodução

Figura 1 - pseudocódigo da implementação

Detalhamento dos passos:

1 – Fase de instanciação e inicialização das dependências necessárias para início do programa, bem com suas bibliotecas.

2 – O reconhecimento do esqueleto a frente do Microsoft Kinect é inicializado, para início da gravação se tem a necessidade de no mínimo um (01) esqueleto identificado podendo ter no máximo dois (02) esqueletos.

3 – Em meio ao código de reconhecimento de cada movimento e captação de cada frame é inserido o código especificado nas *APIs* dos banco de dados, e adição do código de armazenamento em disco, já que armazenamento em disco não tem uma especificação a ser seguida.

4 – Após para a gravação, os dados estão prontos para recuperação e estudo dos movimentos realizados, entretanto ainda para iniciar a reprodução é realizado a programação para recuperação desses dados, tendo em vista que sua forma de armazenamento é a captação do objeto, conversão em base64 e para melhor inserção no banco de dados.

# 4 VISÃO GERAL DO TRABALHO DE CONCLUSÃO

Com a implementação do proposto acima, após as análises dos dados obtidos através da aplicação pudemos então elaborar gráficos para poder mensurar e apresentar em uma forma concreta e concisa os testes obtidos, mensurando então o desempenho de cada tipo de armazenamento proposto, contudo, também elaboramos gráficos com duas diferentes premissas, sendo eles, quantidade total de objetos armazenados no banco e posteriormente dividindo a quantidade de objetos (esqueletos) armazenados no banco pelo tempo de captação, nos dando então os OPS (Objetos por Segundos).

Os dados foram coletados com as mesmas pessoas e no mesmo local, para que não houvesse possível discrepância nas inserções por diferentes locais ou diferentes pessoas.

# 4.1 QT

Relação de quantidade de objetos armazenados durante o tempo especificado na legenda de seus respectivos gráficos;

, onde QT = Quantidade Total e QOA = Quantidade Objetos Armazenados por Segundos.

Figura 2 - 60 segundos de captação

Figura 3 - 30 segundos de captação

Figura 4 - 30 segundos de captação

Figura 5 - 60 segundos de captação

Figura 6 - 30 segundos de captação

# 4.2 OPS

Relação de quantidade de objetos armazenados e feito a divisão pelo tempo especificado na legenda de seus respectivos gráficos.

, onde OPS representa Objetos por Segundos; QT, representa Quantidade Total; T representa do Tempo e P o número de pessoas, chegando em um valor estimado de quantos objetos por segundo foram armazenados.

Figura 7 - 60 segundos de captação

Figura 8 - 30 segundos de captação

Figura 9 - 30 segundos de captação

Figura 10 - 60 segundos de captação

Figura 11 - 30 segundos de captação

# 5 ESCOPO DA APLICAÇÃO

# 5.1 Definição de escopo

Nesta seção será descritos algumas das funcionalidades e prioridades do sistema a ser desenvolvido.

Alta: Codificação da funcionalidade completa.

Média: Codificação da funcionalidade completa, porém dá-se prioridade as funcionalidades qualificadas como Alta.

Baixa: Codificação após o desenvolvimento das funcionalidades de prioridades Altas e Médias, podendo no caso se ocorrer atraso no cronograma, descartar a funcionalidade.

|  |  |  |
| --- | --- | --- |
| IDENTIFICAÇÃO | FUNCIONALIDADE | PRIORIDADE |
| Func\_1 | Reconhecimento esqueleto humano | Alta |
| Func\_2 | Captação do Esqueleto | Alta |
| Func\_3 | Análise e obtenção estatística | Média |
| Func\_4 | Elaboração de ambientes diferentes | Média |
| Func\_5 | Armazenamento dos movimentos | Alta |
| Func\_6 | Reprodução dos dados | Média |
| Func\_7 | Transmissão por rede | Baixa |

Tabela 1 – Tabela de Funcionalidades do Sistema.

**Func\_1** – Reconhecimento esqueleto humano: Ser capaz de identificar qualquer esqueleto, bem como sua reprodução no sistema.

**Func\_2** – Capitação de coordenadas: Ser capaz de capturar o Esqueleto Humano.

**Func\_3** – Após todos os testes terem sido realizados, fazer uso desses dados para gerar informações estatísticas de qual tecnologia é mais viável para o armazenamento.

**Func\_4** – Criação de diferentes ambientes, sendo eles adequação a programação para as diferentes formas de armazenamento, bem como métodos de armazenamento.

**Func\_5** – Armazenamento dos movimentos: Armazenamento dos movimentos realizado pelo esqueleto identificado para posterior reprodução do mesmo.

**Func\_6** – Reprodução dos dados: Os dados deverão ser obtidos e a partir dos mesmos serão interpretados e reproduzidos, a fim de simular o movimento feito.

**Func\_7** – Transmissão dos dados: Após esses dados armazenados, podemos reproduzi-los ou disponibilizá-los a outros usuários, será utilizado uma forma de codificação que pessoas da rede poderão obter esses dados gravados, bem como poder gravar no servidor ligado à rede.

# 6 DETALHES E LIMITAÇÕES DA APLICAÇÃO

# 6.1 Detalhes da aplicação

A aplicação final será uma versão de teste, que poderá ser incrementada para uso profissional ou para fins acadêmicos, podendo ela ser comercializada ou então continuada para estudo e pesquisa. O Microsoft Kinect tem como especificação e em cima disso fora implementado com captação em uma resolução de 640x480 pixels em 30*fps*, no entanto estes *fps* sofre variação de acordo com o computador instalado, bem como em resoluções maiores ou menos, podendo afetar na comparações, caso sejam testados em configurações diferentes.

# 6.2 Limitações de Hardware e Software

As restrições têm como base o sistema e o computador onde foi desenvolvida e testada a aplicação, podendo então ter uma margem de tolerância para mais ou para menos nos requisitos necessários ou nos testes obtidos em outros computadores de *hardwares* diferentes.

# 6.2.1 Hardware

* Processador Core 2 Duo 2.4 GHz.
* 4 GigaBytes de Memoria RAM.
* Placa de video Nvidia 320M.
* 1 Microsoft Kinect.

**6.2.2 Software**

* Microsoft Windows 7 x64 ou superior.
* Microsoft .Net Framework 4.0 ou superior.

# 7 LINGUAGENS E FERRAMENTAS UTILIZADAS

Descrição das ferramentas e tecnologias utilizadas para a construção da aplicação serão apresentadas abaixo:

# 7.1 Linguagens

# 7.1.1 Linguagem de programação C#

Linguagem de programação idealizada pela Microsoft, como parte da plataforma.Net, é uma linguagem orientada a objetos que foi baseada na linguagem C++, no entanto possui influências de outras linguagem, como Java e Object Pascal. A linguagem foi criada pelo Anders Hejsberg e sua equipe em meados de 2000, do qual seu primeiro nome foi Cool e posteriormente renomeada em C#.

Segundo (MSDN ; 2014) C # (pronuncia-se "C sharp") é uma linguagem de programação que é projetada para a construção de uma variedade de aplicações que rodam no NET Framework.. C # é simples, poderoso, de tipo seguro, e orientada a objetos. As muitas inovações em C# permitem o desenvolvimento rápido de aplicações, mantendo a expressividade e elegância das linguagens C-estilo.

Visual C# é uma implementação da linguagem C# pela Microsoft. Visual Studio suporta Visual C# com um editor completo de código, compilador, modelos de projeto, os designers, assistentes de código, um depurador poderoso e fácil de usar, e outras ferramentas. O NET Framework. Fornece acesso a muitos serviços do sistema operacional e outros, classes bem desenhados úteis que acelerar o ciclo de desenvolvimento de forma significativa.

**7.2 Ferramentas**

**7.2.1 Visual Studio**

Segundo [Moura, Arnaldo V. e Ferber, Daniel F., 2005] O Visual Studio.net é um conjunto de ferramentas para construir aplicações para diversas plataformas, começando de aplicações desktop até aplicações coorporativas e aplicações para a internet. Ele inclui três dos compiladores mais populares: Visual Basic, Visual C/C++ e Visual C#. Além dos compiladores, o Visual Studio conta com uma ampla biblioteca de ferramentas de programação, possui facilidades tais como editor de texto integrado com o compilador, ajuda interativa e ferramentas de depuração e identificação de erros no código.

É uma IDE da qual reúne um conjunto de ferramentas Microsoft que viabiliza a construção de aplicações desktop e também para internet, com suporte as linguagens, VB, C, C++, C#, Asp.net, sharespoints, entre outras, é uma ferramenta ponderosa da qual e capaz de construir aplicações inteiras somente utilizado esta ferramenta para codificação .

**7.2.2 DB4O**

Segundo [DB4O, 2014] O db4o é um banco de objetos de código aberto que oferece vantagens exclusivas. Objetos são armazenados nativamente, eliminando a complexidade extra e a perda de performance com a conversão para outros formatos como SQL. O suporte nativo às funcionalidades de base dados orientada a objetos, como as aclamadas Queries Nativas e replicação orientada a objetos, aumenta a performance e os ganhos de desempenho de linguagens orientadas a objeto como o Java e as linguagens da plataforma .NET.

Banco de dados orientado a objetos da qual dá suporte nativo para aplicações Java e .NET, não há necessidade de códigos SQL para realização de CRUD, tem sua especificidade de armazenamento de objetos direto no banco, sem necessidade de definição dos campos ou o tipo de objeto que será inserido.

# 7.2.3 PostgreSQL

Segundo [PostgreSQL, 2014] PostgreSQL é um sistema de banco de dados objeto-relacional open source poderoso. Tem mais de 15 anos de desenvolvimento ativo e uma arquitetura comprovada que ela ganhou uma forte reputação de confiabilidade, integridade de dados e correção. Ele funciona em todos os principais sistemas operacionais, incluindo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), e Windows.É totalmente compatível com ACID, tem suporte completo para as chaves estrangeiras, joins, views, triggers, e procedimentos armazenados (em vários idiomas). Ele inclui mais SQL: 2008 tipos de dados, incluindo INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, e TIMESTAMP. Ele também suporta o armazenamento de grandes objetos binários, incluindo imagens, sons ou vídeo.Possui interfaces de programação nativas para C / C + +, Java,. Net, Perl, Python, Ruby, Tcl, ODBC, dentre outros.

# 7.2.4 Disco

Armazenamento em disco utilizando a classe de implementação baseada na própria da biblioteca de desenvolvimento Microsoft .NET, da qual fora necessário também a conversão do objeto captado para Base64, onde nos retorna uma cadeia de caracteres e então o mesmo é inserido em um arquivo binário e posteriormente gravado em uma pasta específica da aplicação, gerando assim uma sequência cronológica dos esqueletos obtidos.

# 7.2.5 Microsoft Kinect

Kinect e produto desenvolvido pela Microsoft utilizado em conjunto com o Xbox 360, com a finalidade de quebra de paradigma e inovação na questão de jogos, tornando a pratica de jogos mais dinâmica e divertida.

O Kinect possui um trio de inovações de hardware, são câmeras, das quais suas especificações são:

* Câmera de vídeo VGA colorida - Esta câmera de vídeo ajuda no reconhecimento facial e na detecção de outras características ao detectar três cores componentes: vermelho, verde e azul. A Microsoft chama isso de "câmera RGB" se referindo aos componentes de cor que ela detecta.
* Sensor de profundidade - Um projetor infravermelho e um sensor CMOS monocromático trabalham juntos para "ver" a sala em 3D sem levar em conta as condições de iluminação.
* Microfone multi-matriz - Este é uma série de quatro microfones que podem isolar as vozes dos jogadores do barulho da sala. Ela possibilita ao jogador estar a poucos metros de distância do microfone e ainda usar controles por voz.

# 8 METODOLOGIA DE DESENVOLVIMENTO

A metodologia que será utilizada para construção do software será a o processo Ágil Scrum, Segundo [Schwaber, Ken e Sutherland, Jeff, 2013] Scrum é uma estrutura processual (framework) para suportar o desenvolvimento e manutenção de produtos complexos. O Scrum consiste em Equipes do Scrum associadas a seus papéis, eventos, artefatos e regras. Cada componente dentro do framework serve a um propósito específico e é essencial para o uso e o sucesso do Scrum.

O Scrum tem a seguinte hierarquia:

* ProductOwner – E o dono do produto, e o responsável por maximizar o valor do produto e o esforço despendido da sua equipe, ele também é o único responsável por organizar o BacklogProduct.
* BacklogProduct – Lista de definições que compõe o software, ou seja, é uma lista de atividades que dará origem ao software completo, sendo eles seus requisites funcionas e não funcionais do software proposto.
* Planning Meeting – Plano para cada Sprint, geralmente ocorre antes do início de cada Sprint, o ScrumMaster mostra a sua equipe o que será necessário desenvolver até o final deste Sprint, essas funcionalidades expostas aos integrantes da equipe ficam definidas no BacklogProduct.
* Daily Scrum – Reuniões diárias da que a equipe se reúne e compartilha o que foi feito no dia anterior, basicamente são realizados algumas questões, como: O que você fez no projeto desde a última Daily Scrum? O que você planeja fazer no projeto no período entre essa reunião e a próxima? O que está te impedindo de cumprir suas metas para esse projeto? O objetivo destas reuniões são para sincronização das tarefas e identificar problemas dos quais a equipe esta tendo e então procurar soluções rápidas.
* Sprint Review – Reunião antes do Planning Meeting, a equipe mostra ao ScrumMaster o que fora desenvolvido referente ao Sprint referido, geralmente nestas reuniões são integrantes também os usuários do sistema, para que então eles homologuem o que fora feito pela equipe de desenvolvimento.

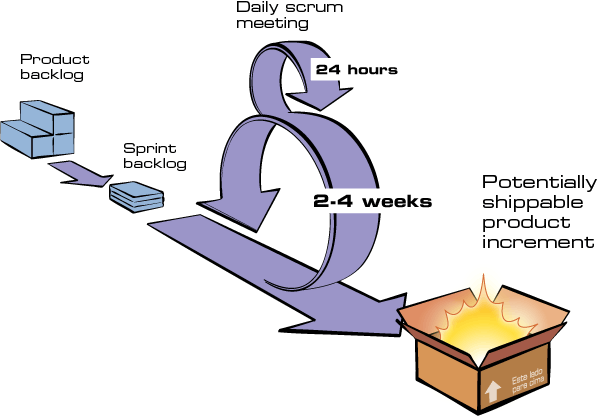


Figura 12 - Método Scrum

# 9 CRONOGRAMA

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ATIVIDADES** | **NOV** | **DEZ** | **JAN** | **FEV** | **MAR** | **ABR** | **MAI** | **JUN** |
| Requisitos |  |  |  |  |  |  |  |  |
| Análise |  |  |  |  |  |  |  |  |
| Projeto |  |  |  |  |  |  |  |  |
| Codificação |  |  |  |  |  |  |  |  |
| Teste |  |  |  |  |  |  |  |  |
| Reparos |  |  |  |  |  |  |  |  |
| Escrita Final TD |  |  |  |  |  |  |  |  |
| TCC |  |  |  |  |  |  |  |  |

Tabela 2 – Cronograma

# 10 REFERÊNCIAS

COSTA, José Wilson da; OLIVEIRA, Maria Auxiliadora. **Novas linguagens e** **novas tecnologias**: educação e sociabilidade. Petrópolis, RJ: Vozes, 2004.

MENDES, Oston. **Realidade Virtual: Xbox Kinect é utilizado para reabilitação na fisioterapia**. Nova Fisioterapia, São Paulo. Disponível em: <http://www.novafisio.com.br/artigos_jornalisticos/noticias/realidade-virtual-xbox-kinect-e-utilizado-para-reabilitacao-na-fisioterapia/> : Acesso em 23/11/2013, as 18:12.

NUNES, Douglas. **Mercado de jogos eletrônicos explode no Brasil**. Brasil Economico, 26/12/2013. Disponivel em: <http://economia.ig.com.br/empresas/2013-12-26/mercado-de-jogos-eletronicos-explode-no-brasil.html> : Acesso em 22/01/2014.

RIBEIRO, K. S. Q. **A atuação da fisioterapia na atenção primária à saúde**. Fisioterapia Brasil, v.3, n.5, p.311-318, 2002.

<http://msdn.microsoft.com/en-us/library/kx37x362.aspx> : Acesso em 05/01/2014, as 20:13.

RAMOS, Daniela. **Jogos eletrônicos, desejo e juízo moral**. Universidade Federal de Santa Catarina, 2008.

MOURA, Arnaldo V. e FERBER, Daniel F. **O primeiro programa em Visual Studio**, 11/03/2009, p.01. Disponível em: <http://www.ic.unicamp.br/~mc102/aulas/p_arnaldo/texto/cap01_introducao_visual_studio.pdf> : Acesso em 18/01/2014, as 09:02.

ZHOU, Huiyu e HU, Huosheng. **A Survey - Human Movement Tracking and Stroke Rehabilitation**, University of Essex, Departamento de Ciência da Computação, Reino Unido. 08/12/2004.

Desenvolvimento Ágil.**Scrum**, Disponível em: [http://desenvolvimentoagil.com.br/scrum/](http://desenvolvimentoagil.com.br/scrum/.ht) : Acesso em 17/01/2014, as 17:52.

SCHWABER,Ken e SUTHERLAND, Jeff. **Guia do Scrum**, Scrum.org, 10/2011. Disponível em: <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum%20Guide%20-%20Portuguese%20BR.pdf>: Acesso em 19/01/2014, as 08:37**.**

# COLDEWEY, Devin. Kinect specs posted: 640×480 at 30FPS, two players maximum, 29/06/2010, Disponível em: <http://techcrunch.com/2010/06/29/kinect-specs-posted-640x480-at-30fps-two-players-maximum/> : Acesso em 20/04/2014.

<http://social.msdn.microsoft.com/Forums/en-US/1a17bf21-b504-4a7d-a24b-7477debd9a53/kinect-skeleton-data?forum=kinectsdknuiapi>, Acesso em: 17/04/2014.

<http://social.msdn.microsoft.com/Forums/en-US/d810e3ff-4b1c-48d2-a3af-62a14f265b4a/threads-trouble?forum=kinectsdk>, Acesso em: 26/04/2014.

# Cassio, [Drawing skeleton animation from file to screen](http://stackoverflow.com/questions/13940341/drawing-skeleton-animation-from-file-to-screen), 18/12/2012, <http://stackoverflow.com/questions/13940341/drawing-skeleton-animation-from-file-to-screen>, Acesso em 22/03/2014.

OKANO, Marcelo. **Análise dos melhores ORM (Object-Relational Mapping) para plataforma .NET**, Disponível em:<http://www.devmedia.com.br/analise-dos-melhores-orm-object-relational-mapping-para-plataforma-net/5548>, Acesso em: 05/05/2014.

# 11 APÊNDICE A

# 

Segue abaixo as seguintes diferenças de implementações para conexão com o banco de dados, armazenamento e recuperação destes dados:

PostgreSQL

//Comando para fazer a recuperação dos dados gravados e gerar uma lista;

cmd2 = new NpgsqlCommand("select DISTINCT nome from esqueleto", conn);

//Preenchimento desta lista no combobox;

listaDB.Items.Add(dr[0].ToString());

# 

//Inserção do objeto no banco de dados PostgreSQL

bf.Serialize(memorystream, skel);

byte[] yourBytesToDb = memorystream.ToArray();

var base64 = Convert.ToBase64String(yourBytesToDb);

cmd2 = new NpgsqlCommand("insert into esqueleto (nome,obj) values ('" + nome + "',' " + base64 + " ')", conn);

# //Buscar esqueletos para reprodução

cmd2 = new NpgsqlCommand("select obj from esqueleto where nome like '" + response + "'", conn);

DB4o

string User = "123";string Password = "123";

public IList<Skeleton> skeletonData;

int Porta = 8080;

IObjectServer server;

IObjectContainer client;

# 

//Configurar server;

server = Db4oClientServer.OpenServer(Db4oClientServer.NewServerConfiguration(),

databasenome, Porta); server.GrantAccess(User, Password);

//Configurar Cliente;

client = Db4oClientServer.OpenClient(Db4oClientServer.NewClientConfiguration(), "localhost", 8080, User, Password);

//Gravar esqueleto;

client.Store(skel);

//Recuperar esqueleto;

skeletonData = client.Query<Skeleton>();

Disco

//Gravacao;

this.serilizaObjeto(skel, ondeSalva + "\\" + gravacaoTeste + ".bin");

gravacaoTeste++;

//Reprodução;

this.DrawBonesAndJoints(skeletonData[Desenhar], dc);

public void serilizaObjeto(Skeleton skel, string Path) {

System.IO.Stream ms = File.OpenWrite(Path);

BinaryFormatter formatter = new BinaryFormatter();

formatter.Serialize(ms, (Skeleton)skel);

}

public void DeserializarObjeto(String filename){

BinaryFormatter formatter = new BinaryFormatter();

skeletonData = new List<Skeleton>();

for (int i = 0; i < TotalSkeletos; i++) {

FileStream fs = File.Open(filename + "\\" + i + ".bin", FileMode.Open);

Skeleton obj = (Skeleton)formatter.Deserialize(fs);

skeletonData.Add((obj));

}

}

# 

# 