

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS PATO BRANCO
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

JÓRDAN LUIZ BISATO

**SOFTWARE PARA ACOMPANHAMENTO E PREDIÇÃO DE DADOS
ANTROPOMÉTRICOS INFANTIS**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2016**

JÓRDAN LUIZ BISATO

**SOFTWARE PARA ACOMPANHAMENTO E PREDIÇÃO DE DADOS
ANTROPOMÉTRICOS INFANTIS**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientadora: Prof. Rúbia Eliza de Oliveira Schultz Ascari

**PATO BRANCO
2016**

ATA Nº: 281

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO **JÓRDAN LUIZ BISATO**.

Às 16:43 hrs do dia 23 de junho de 2016, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Rúbia E. O. Schultz Ascari (Orientadora), Vinicius Pegorini (Convidado) e Kathya Silvia Collazos Linares (Convidada), para avaliar o Trabalho de Diplomação do aluno Jórdan Luiz Bisato, matrícula 1374567, sob o título **Software para Acompanhamento e Predição de Dados Antropométricos Infantis**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 17:05 hrs foi encerrada a sessão.

Profa. Rúbia E. O. Schultz Ascari, M.Sc.
Orientadora

Prof. Vinicius Pegorini, M.Sc.
Convidado

Profa. Kathya Silvia Collazos Linares, Dr.
Convidada

Profa. Eliane Maria de Bortoli Fávero, M.Sc
Coordenadora do Trabalho de Diplomação

Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

AGRADECIMENTOS

Agradeço a todos os professores que estiveram presentes durante minha graduação passando um pouco de seu conhecimento para todos, mas especialmente a professora Beatriz e ao professor Omero por me fazer sentir gosto pela programação e também a professora Rúbia que me orientou e auxiliou neste projeto, além de passar seus conhecimentos em Delphi. Deixo um agradecimento também a todos os meus colegas de curso, que me ajudaram a chegar até aqui, tanto na forma de auxílio durante as dúvidas, como também pelo ambiente que eles proporcionavam nas aulas.

RESUMO

BISATO, Jórdan Luiz. Software para acompanhamento e predição de dados antropométricos infantis. 2015. 40f. Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

Desde seu nascimento, as crianças recebem acompanhamento médico a fim de verificar seu desenvolvimento e possibilitar a detecção prévia de problemas e possíveis doenças. Além disso, a criança precisa se desenvolver bem nos primeiros meses para não ter problemas futuramente. Para saber se a criança está se desenvolvendo corretamente deve-se saber, dentre outras informações, o seu peso e altura, para comparar com os gráficos de referência do *National Center for Health Statistics* (NCHS) e da Organização Mundial da Saúde (OMS), que são gráficos que contêm uma linha média de crescimento saudável para crianças. Com o acompanhamento é possível prevenir problemas antes que eles aconteçam, e, para auxiliar nisso, esse trabalho apresenta um software desenvolvido para permitir o registro dos dados antropométricos infantis, e com base nesse histórico da criança, gerar uma predição de como estará o desenvolvimento da criança em determinado período de tempo no futuro. Para ser possível esta predição o software emprega um algoritmo de programação genética responsável por realizar um processo de regressão simbólica. Este algoritmo usará como base os dados cadastrados do sistema e irá gerar as predições. O software desenvolvido pode ser utilizado para identificar precocemente possíveis problemas de desnutrição, obesidade e outros aspectos relacionados ao desenvolvimento físico, servindo de auxílio na tomada de decisão por parte de especialistas da área da saúde.

Palavras-chave: Desenvolvimento infantil; Antropometria; Regressão Simbólica. Programação Genética.

ABSTRACT

BISATO, Jórdan Luiz. Software to Monitoring and prediction of children's anthropometric data. 2015. 40f. Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco 2015.

Since its birth, children receive medical care in order to verify their development and enable early detection of problems and possible diseases. In addition, the child needs to develop well in the first months to avoid problems in the future. To find out if the child is developing properly should know, among other information, your weight and height, to compare with the reference charts of the National Center for Health Statistics (NCHS) and the World Health Organization (WHO), this graphs contains a midline healthy growth for children. The monitoring can prevent problems before they happen, and to assist in this, this paper presents a software developed to allow the registration of children's anthropometric data, and based on that child's history, generating a prediction of how the development will be the child certain time in the future. To be able to predict this software employs a genetic programming algorithm which will be responsible for performing a symbolic regression process. This algorithm is based on the system registration data and will generate predictions. The software developed can be used to identify early possible malnutrition problems, obesity and other aspects related to the physical, serving as aid in decision making by health care experts.

Keywords: Child development; anthropometry; Symbolic regression. Genetic Programming.

LISTA DE FIGURAS

Figura 1 - Gráfico de peso x idade da caderneta de saúde.....	6
Figura 2 - Exemplo de curvas geradas por regressão simbólica.....	9
Figura 3 - Diagrama de Casos de Uso.....	16
Figura 4 - DER mostrando as tabelas do sistema.....	18
Figura 5 - Tela de login do sistema.....	18
Figura 6 - Tela principal do sistema.....	19
Figura 7 - Cadastro de Responsáveis.....	19
Figura 8 - Cadastro de Crianças.....	20
Figura 9 - Cadastro de Consultas.....	20
Figura 10 - Cadastro de Predições.....	21
Figura 11 - Cadastro de Predições.....	21
Figura 12 - Gráfico de um exemplo de predição gerada.....	22
Figura 13 - Resultado da predição.....	23

LISTA DE QUADROS

Quadro 1 - Sequencia usada pelo algoritmo de PG.	8
Quadro 2 – Caso de uso Gerar previsão.....	17
Quadro 3 - Caso de uso Manter cadastro de crianças.....	17
Quadro 4 - Caso de uso Manter cadastro de consultas.....	17
Quadro 5 - Caso de uso Manter cadastro de responsáveis.....	17
Quadro 6 - Código da stored procedure SP_INS_CRIANCA.....	23
Quadro 7 - Código do generator SP_INS_CRIANCA.....	23
Quadro 8 - Código da stored procedure SP_INS_CRIANCA.....	24

LISTAGENS DE CÓDIGO

Listagem 1 - Código da Geração de Fitness.....	24
Listagem 2 - Código da inicialização da população.....	25
Listagem 3 - Código da função usada para organizar equação (Parte 1).....	26
Listagem 4 - Código da função de organizar equação (Parte 2)	27
Listagem 5 - Código para identificar o Fim da Equação	29

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmos Genéticos
EQM	Erro Quadrático Médio
IDE	<i>Interface Development Environment</i>
IMC	Índice de Massa Corporal
MATLAB	<i>Matrix Laboratory</i>
NCHS	<i>National Center for Health Statistics</i>
OMS	Organização Mundial da Saúde
PG	Programação Genética
WHO	<i>World Health Organization</i>

SUMÁRIO

1 INTRODUÇÃO	1
1.1 CONSIDERAÇÕES INICIAIS	1
1.2 OBJETIVOS	2
1.2.1 Objetivo Geral	2
1.2.2 Objetivos Específicos	2
1.3 JUSTIFICATIVA	2
1.4 ESTRUTURA DO TRABALHO	4
2 REFERENCIAL TEÓRICO	5
2.1 ANTROPOMETRIA	5
2.2 PROGRAMAÇÃO GENÉTICA	7
2.3 REGRESSÃO SIMBÓLICA	8
3 MATERIAIS E MÉTODO	10
3.1 MATERIAIS	10
3.1.1 Visual Paradigm	10
3.1.2 Firebird Maestro	11
3.1.3 Matlab.....	11
3.1.4 Delphi	12
3.1.5 Firebird	12
3.1.6 IBExpert.....	13
3.2 MÉTODO.....	13
4 RESULTADOS	15
4.1 ESCOPO DO SISTEMA	15
4.2 MODELAGEM DO SISTEMA	15
4.3 APRESENTAÇÃO DO SISTEMA	18
4.4 IMPLEMENTAÇÃO DO SISTEMA	23
4.5 DISCUSSÕES.....	29
5 CONCLUSÃO	31
REFERÊNCIAS	32

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais com uma visão geral do trabalho, os objetivos, a justificativa e a estrutura do trabalho. Os objetivos explicitam as finalidades principais. A justificativa centra-se no tipo de sistema desenvolvido e nas tecnologias utilizadas.

1.1 CONSIDERAÇÕES INICIAIS

O acompanhamento médico é feito desde os primeiros dias de vida da criança, pois é nesta fase que o ser humano é mais vulnerável a doenças em geral. Por isso devem ser realizadas consultas frequentes nos primeiros meses de vida, porque com base no diagnóstico do pediatra é que são descobertos problemas e possíveis doenças que podem ser tratadas precocemente.

Além disso, a criança precisa se desenvolver bem nos primeiros meses para não ter problemas futuramente. Para saber se a criança está se desenvolvendo corretamente registra-se o peso, a altura, circunferência cefálica da criança entre outros dados, para comparar com os gráficos de referência do *National Center for Health Statistics* (NCHS) e da Organização Mundial da Saúde (OMS), que são gráficos que contém uma linha média de crescimento saudável para crianças.

Com o acompanhamento é possível prevenir problemas antes que eles aconteçam, e, para auxiliar nisso, neste trabalho será modelado e desenvolvido um software para acompanhamento e predição de dados antropométricos infantis, como peso e altura, que auxiliará a predizer como estará o desenvolvimento da criança em determinado período de tempo no futuro.

Para realizar essa predição será utilizado um algoritmo de programação genética como o apresentado por Ascari, Borsoi e Fávero (2013), que será responsável por realizar um processo de regressão simbólica. Este algoritmo utilizará como base os dados cadastrados no sistema para gerar as predições.

1.2 OBJETIVOS

O objetivo geral corresponde ao desenvolvimento de um software para acompanhamento do desenvolvimento infantil e os objetivos específicos complementam o objetivo geral, tanto em termos das tecnologias utilizadas como da finalidade do projeto.

1.2.1 Objetivo Geral

Desenvolvimento de um software para acompanhamento e predição de dados antropométricos infantis com base em um algoritmo de programação genética.

1.2.2 Objetivos Específicos

- Desenvolver um software que permita armazenar os dados antropométricos de crianças em seus primeiros anos de vida, a fim de poder acompanhar seu crescimento.
- Fornecer uma ferramenta que possa servir como auxílio na tomada de decisão por parte de especialistas da área da saúde.
- Aprimorar os conhecimentos nas ferramentas Delphi, MySQL e Visual Paradigm.
- Conhecer e utilizar um algoritmo de programação genética a fim de retratar tendências de evolução do desenvolvimento infantil, com base no histórico de dados antropométricos.

1.3 JUSTIFICATIVA

A Antropometria é a ciência que estuda e avalia as medidas de tamanho (altura, diâmetros e comprimentos ósseos, espessuras e dobras cutâneas, circunferência cefálica), peso e alguns índices que avaliam o risco de desenvolver doenças relacionadas ao desenvolvimento físico (FERNANDES, 1999;

VASCONCELOS, 2000). É amplamente aplicada na avaliação do estado nutricional de crianças e adolescentes e suas aplicações têm se expandido, incluindo a capacidade de selecionar indivíduos que se beneficiarão de determinada intervenção, identificar problemas socioeconômicos e avaliar a resposta às intervenções.

Para a Organização Mundial de Saúde (*World Health Organization (WHO)*), o Ministério da Saúde e a Sociedade Brasileira de Pediatria, o acompanhamento do crescimento deve ser atividade de rotina no atendimento infantil, devido ao reconhecimento da influência que as condições de vida exercem sobre o crescimento (ZEFERINO, 2003).

Nos primeiros meses de vida, uma criança deve fazer visitas frequentes a um médico pediatra para registrar e acompanhar o seu desenvolvimento. Nessas consultas são realizadas as medições de peso, altura, índice cefálico e demais dados antropométricos considerados importantes pelo médico. Esses dados são registrados na carteira de saúde da criança, e em alguns casos, é mantido também um histórico pelo médico que realizou a medição. Para os pais, se a caderneta for perdida, perde-se também o histórico de crescimento da criança, e conseqüentemente informações importantes para avaliar a evolução do desenvolvimento de seu filho.

Assim, viu-se oportuno desenvolver um software para auxiliar no registro dos dados antropométricos que são diagnosticados a cada consulta, sendo possível que o médico responsável ou mesmo os pais da criança façam o registro, mês a mês, ou a cada consulta. Tendo esses dados históricos, mesmo que existam intervalos temporais distintos, é possível realizar um processo chamado de regressão simbólica para identificar uma função que represente uma curva o mais próximo possível dos dados registrados.

Um algoritmo de programação genética pode gerar automaticamente uma equação matemática baseada em regressão simbólica, que represente o mais próximo possível a curva de crescimento infantil (referentes a peso e altura) registrada historicamente. Sendo possível, assim, realizar a predição de como será o seu desenvolvimento nos meses seguintes, permitindo prever possíveis problemas como obesidade, desnutrição, entre outros.

Assim, foi modelado e implementado um software que utiliza um algoritmo de programação genética para criar uma predição de dados antropométricos infantis,

utilizando os dados cadastrados pelo usuário no sistema. Os resultados do processo serão demonstrados em gráficos.

Este software auxiliará os pais e profissionais da área médica a prever possíveis problemas com o desenvolvimento de crianças, dando oportunidade de prevenir problemas antes mesmo de acontecerem, ao invés de tratá-lo somente após o aparecimento dos sintomas.

1.4 ESTRUTURA DO TRABALHO

O texto está organizado em capítulos. O primeiro capítulo apresenta a ideia do sistema, incluindo os objetivos e a justificativa.

O capítulo 2 contém o referencial teórico que fundamenta a proposta conceitual do sistema desenvolvido. O referencial teórico se concentra em antropometria, programação genética e regressão simbólica.

No capítulo 3 estão os materiais e o método utilizados. Os materiais se referem ao que é necessário para modelar e implementar o sistema, incluindo as tecnologias, as ferramentas e os ambientes de desenvolvimento utilizados. O método se refere aos procedimentos utilizados no ciclo de vida do sistema, abrangendo da definição dos requisitos à implementação do sistema.

O capítulo 4 contém o sistema desenvolvido, com exemplos e documentos de modelagem produzidos e as telas de codificação do sistema. Neste capítulo também está incluído o uso das tecnologias empregadas para desenvolver o sistema.

No capítulo 5 está a conclusão com as considerações finais. Por fim estão as referências bibliográficas.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico do trabalho, com conceitos, dados sobre antropometria e sua importância no desenvolvimento infantil.

2.1 ANTROPOMETRIA

O crescimento físico do ser humano ocorre segundo uma sequência característica, que está ligada a aspectos biológicos do desenvolvimento. Monitorar as medidas antropométricas, durante o processo de crescimento, permite a qualificação das variações morfológicas decorrentes desse processo, fornecendo dados para o diagnóstico de possíveis deficiências (LOPES, 2014).

Em crianças e adolescentes é comum o uso dos indicadores antropométricos de peso/idade, altura/idade, peso/altura e Índice de Massa Corporal (IMC) segundo idade e sexo (ARAÚJO e CAMPOS, 2008).

No Brasil, a partir de 2005, toda criança recebe ao nascer uma caderneta de saúde que contém informações úteis aos pais sobre temas diversos relacionados ao desenvolvimento da criança.

Além disso, há os gráficos antropométricos desenvolvidos com base nas referências da NCHS e OMS, para acompanhamento do crescimento da criança, levando em consideração a evolução de seu peso, altura, IMC e perímetro cefálico, como mostrado na Figura 1 abaixo.

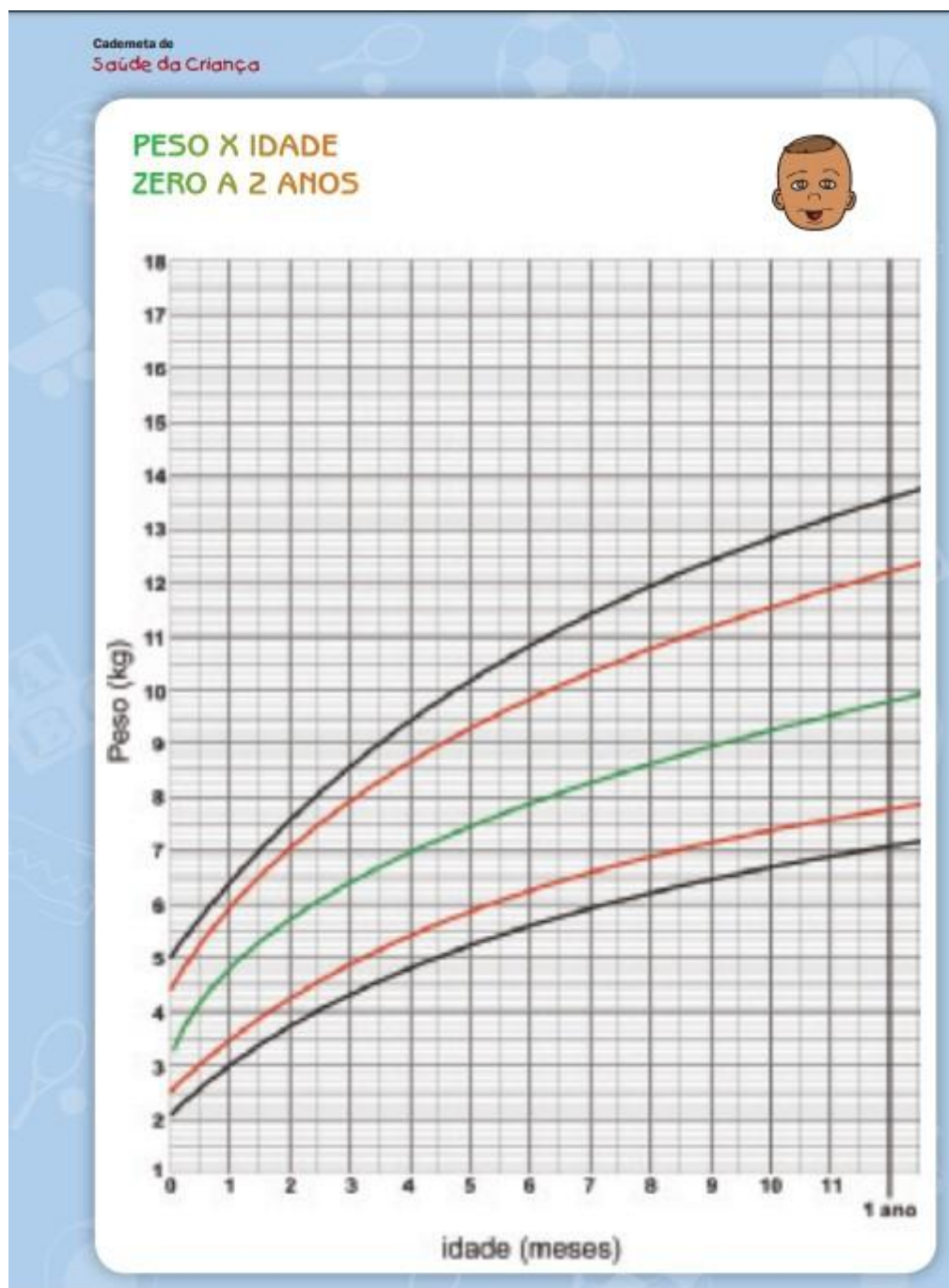


Figura 1 - Gráfico de peso x idade da caderneta de saúde
Fonte: Caderneta de saúde da criança(2013)

Os valores são válidos para crianças nascidas a termo (com 37 semanas ou mais de gestação). A avaliação do estado nutricional, usando os dados antropométricos, tem o objetivo de quantificar e qualificar essas medidas que são comparadas com valores de referência e assim, determinar se os valores encontrados estão ou não dentro dos intervalos de normalidade (ARAÚJO e CAMPOS, 2008).

A obtenção dos dados antropométricos é realizada por um método barato, não-invasivo, de fácil aplicação, de boa aceitação por parte da população e extremamente útil para rastrear obesidade, subnutrição e outros agravos nutricionais (WHO, 2002). Basicamente a criança é pesada em balança específica para registrar o seu ganho de peso, e medida para registrar sua altura e circunferência cefálica no momento da consulta. Assim, cada criança tem um registro básico de peso, altura e demais informações consideradas necessárias para cada mês ou consulta realizada.

2.2 PROGRAMAÇÃO GENÉTICA

A representação de programas em árvores e a definição de operadores genéticos foram apresentadas inicialmente por Cramer (1985). Seguindo esta linha de raciocínio, e tomando como base os trabalhos de Holland (1975) em Algoritmos Genéticos (AG), Koza (1989; 1992) introduziu o conceito de programação genética (PG) utilizando-a como meio de conduzir a pesquisa por programas no espaço de soluções.

A Programação Genética se baseia na combinação de ideias da teoria da evolução (seleção natural), genética (reprodução, cruzamento e mutação), inteligência artificial (busca heurística) e teoria de compiladores (representação de programas como árvores sintáticas) (KOZA, 1992).

Conceitualmente, a Programação Genética representa uma abordagem para a geração automática de programas de computador. Neste trabalho a PG será empregada para gerar automaticamente expressões simbólicas que serão utilizadas na predição de dados antropométricos infantis (peso e altura).

O algoritmo de programação genética desenvolvido nesse trabalho foi baseado no apresentado em Ascari, Borsoi e Fávero (2013) e considera como entrada o peso ou a altura registrada da criança e a diferença temporal entre esses registros. Então, para um conjunto de medições registradas em “n” instantes de tempo, o peso ou a altura poderá predito para o instante “tn+1”, com base na melhor solução apresentada pelo algoritmo de PG. A melhor solução neste caso será uma expressão cujos valores de saída resultem no menor erro de aproximação (modificação do Erro Quadrático Médio)(EQM). O método considera todos os pontos

registrados pela avaliação destinados para treinamento, assumindo que os pontos finais têm maior influência.

Basicamente, o algoritmo de PG gera uma população de equações matemáticas (indivíduos), avalia cada indivíduo com base na equação que ele representa, usa métodos de seleção para definir quais equações sobreviverão, aplica operadores genéticos para modificar equações existentes e gerar novas equações, a fim de convergir para uma boa solução. O algoritmo de PG proposto é baseado nos passos definidos por Suttasupa et al. (2011), conforme o Quadro 1 apresentado em Ascari, Borsoi e Fávero (2013).

1. Criar uma população randômica inicial. O conjunto de funções (como +, -, * e /) torna-se os nós internos da árvore. O conjunto terminal (as variáveis livres) torna-se nós externos (folha).
2. Evoluir a população de acordo com a função de aptidão. A aptidão indica quão bem cada indivíduo resolve o problema.
3. Criar a próxima geração da população pela evolução da sua estrutura da seguinte forma:
 - 3.1 Selecionar os melhores indivíduos da população e copiá-los para a próxima geração usando seleção por torneio (escolha aleatória ou determinística).
 - 3.2 Criar uma nova árvore pelo método crossover.
 - 3.3 Criar nova árvore pelo método de mutação.
4. Repetir os passos 2 e 3 até a solução ser encontrada ou a geração máxima seja alcançada.

Quadro 1 - Sequencia usada pelo algoritmo de PG.

Fonte: Ascari, Borsoi e Fávero (2013).

2.3 REGRESSÃO SIMBÓLICA

O algoritmo de PG proposto está focado na indução de expressões matemáticas baseadas em dados de estudo, ou seja, executa o processo chamado de regressão simbólica (KOZA, 1992).

O objetivo principal da regressão simbólica é encontrar uma função que se aproxime ao máximo de outra função desconhecida por meio de um determinado conjunto de dados. No algoritmo desenvolvido, o algoritmo de PG busca por expressões matemáticas que apresentem uma boa aproximação da curva de evolução do desenvolvimento da criança em estudo.

A Figura 1 apresenta um gráfico onde pode-se visualizar alguns pontos considerados reais (destacados como pequenos círculos), de peso de uma determinada criança, registrados em instantes de tempo diferentes, a curva gerada com base nesses registros (destacada na cor vermelha), e os pontos que representam outra curva (destacados na cor azul), gerada por um processo de

regressão simbólica, tentando se aproximar da curva existente. A equação utilizada para gerar essa segunda curva é apresentada na parte inferior da figura.

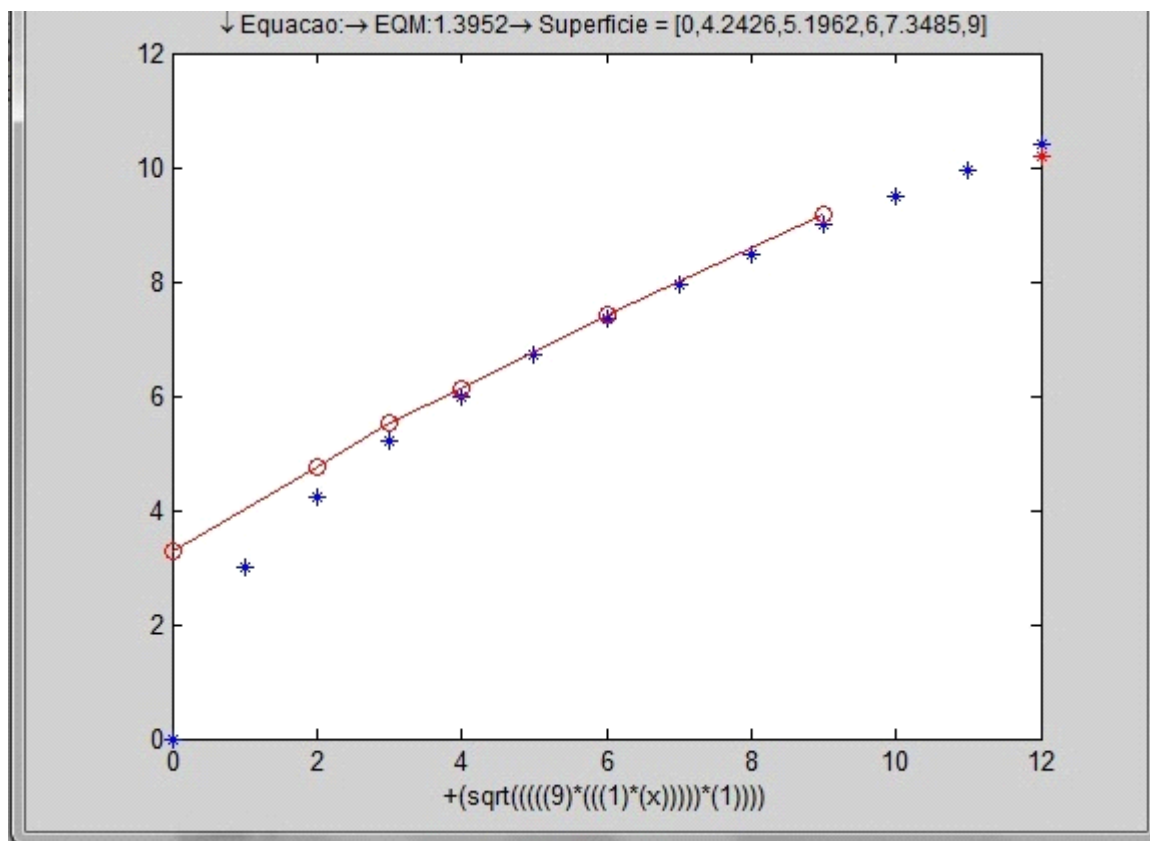


Figura 2 - Exemplo de curvas geradas por regressão simbólica.
 Fonte: Ascari, Borsoi e Fávero (2013).

Assim, tendo uma expressão que permita gerar um resultado próximo à tendência apresentada pelo conjunto de dados de treinamento, pode-se considerar que é possível prever dados futuros, baseados nessa tendência.

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e o método utilizados para a análise e o desenvolvimento do sistema. Os materiais se referem às ferramentas e às tecnologias, incluindo linguagem de programação, banco de dados, interface de desenvolvimento e aplicativo para análise e modelagem. O método se refere aos procedimentos utilizados no ciclo de vida do sistema, abrangendo da definição dos requisitos à implementação do sistema.

3.1 MATERIAIS

As ferramentas e as tecnologias utilizadas para as atividades de modelagem, implementação e execução do software foram:

- a) Visual Paradigm para modelagem do diagrama de caso de uso.
- b) Firebird Maestro para modelagem do diagrama de entidades e relacionamentos.
- c) Matlab para estudo do algoritmo de programação genética utilizado como base.
- d) Delphi XE3 como IDE (*Interface Development Environment*) de desenvolvimento e linguagem de programação.
- e) Firebird como banco de dados.
- f) IBExpert como IDE para gerenciar o banco de dados Firebird.

3.1.1 Visual Paradigm

Conforme apresentado no site de um de seus fornecedores (SOFTWARE, 2015), Visual Paradigm para UML é uma ferramenta para desenvolvimento de aplicativos que utiliza modelagem UML, ideal para Engenheiros de Software, Analistas de Sistemas e Arquitetos de Sistemas que estão interessados em criação de sistemas em larga escala e necessitam de confiabilidade e estabilidade no desenvolvimento orientado a objetos.

Ainda conforme apresentado em Software (2015), o Visual Paradigm proporciona:

- Navegação intuitiva entre a escrita do código e a sua visualização;
- Gerador de relatórios em PDF/HTML;
- Documentação automática Ad-hoc;
- Ambiente visualmente superior de modelagem;
- Sofisticado diagramador automático de lay-out;
- Sincronização de código fonte *Real time* ou *On Demand*, e outras opções.

3.1.2 Firebird Maestro

Firebird Maestro é uma solução para a administração do servidor Firebird e desenvolvimento. Os principais recursos incluem suporte para todas as versões do Firebird, designer de banco de dados, ferramentas de gerenciamento de dados com visualização, edição, agrupamento, entre outras. Editor SQL acessível com destaque de sintaxe, formatador SQL e multi-threading, construtor de consulta visual, dados de exportação/importação para os formatos mais populares como Excel, HTML e XML. Acesso facilitado para os recursos de segurança do Firebird (SOFTWARE, 2015).

O aplicativo também fornece um conjunto de ferramentas para editar e executar scripts SQL e construir diagramas visuais para dados numéricos.

3.1.3 Matlab

MATLAB (*Matrix Laboratory*) é um software de alto nível voltado para cálculo numérico, análise de dados, cálculo com matrizes, construção de gráficos e implementação de algoritmos. Os comandos utilizados são próximos da forma que se escreve expressões algébricas (PETELETRICA, 2015).

O MATLAB é desenvolvido pela MatWorks e, apesar de ser um software proprietário, é muito utilizado em instituições de pesquisa, universidades e empresas, principalmente pelo grande número de toolboxes disponíveis, como as específicas para trabalhar com inteligência artificial.

3.1.4 Delphi

Delphi é uma IDE que possibilita o desenvolvimento de sistemas com interface visual, incluindo sistemas gerenciais onde se consegue de maneira prática e fácil produzir softwares desktop com qualidade e em pouco tempo. Foi produzido pela Borland Software Corporation e a atual empresa proprietária é a Embarcadero.

É um ambiente de desenvolvimento de aplicações baseado em programação orientada a objetos, visual e muito fácil de utilizar. É bastante utilizado para desenvolvimento de aplicações multicamadas e Cliente/Servidor. Permite acesso aos bancos de dados mais conhecidos do mercado.

3.1.5 Firebird

O Firebird é derivado do código do Borland InterBase 6.0. Ele tem o código aberto e não possui licença dupla, portanto pode ser utilizado em qualquer tipo de aplicação, seja ela comercial ou não, gratuitamente.

O servidor Firebird é disponibilizado em quatro variações: SuperServer, Classic, SuperClassic e Embedded. Para iniciar seu uso, sugere-se o uso da variação SuperServer.

A versão Classic é indicada para máquinas com mais de um processador e em algumas situações específicas. O SuperServer compartilha o cache entre as conexões com o banco, e utiliza threads para gerenciar cada conexão. A versão Classic inicia um processo independente do servidor para cada conexão estabelecida. O SuperClassic usa threads em um único processo do servidor, com cache independente para cada conexão. A versão embedded consiste em um servidor Firebird completo composto por apenas alguns arquivos, o que o torna muito fácil de ser distribuído, pois não há necessidade de instalação. Embedded torna-se ideal para o uso em catálogos em CDROM, versões de avaliação de utilitários ou aplicações standalone (CANTU, 2010, p. 1).

No Windows, o Firebird pode ser executado como um serviço ou no modo de aplicação, que permite ao usuário iniciar ou parar o servidor por meio de um utilitário instalado no painel de controle do computador.

Conforme descrito em Cantu (2010), o Firebird vem com uma série de utilitários de linha de comando que permitem criar bancos de dados, recuperar estatísticas, executar comandos e scripts SQL, efetuar backups e restores, entre outros.

3.1.6 IBExpert

O IBExpert é um software de gerenciamento de bancos de dados Interbase e Firebird, possui várias ferramentas que permitem total controle sobre os servidores e bases de dados gerenciados.

Este programa foi criado pela HK-Software e conta com atualizações frequentes, sua principal função é gerenciar bancos de dados relacionais utilizando da linguagem SQL.

3.2 MÉTODO

As principais atividades realizadas para o desenvolvimento deste trabalho foram:

a) Levantamento de Requisitos

Os requisitos foram levantados e definidos tendo como base o interesse em desenvolver um software para auxiliar médicos, pais e responsáveis a acompanhar o desenvolvimento antropométrico de crianças. O levantamento de requisitos foi realizado pelo autor deste trabalho a partir das informações que teriam necessidade de se armazenar a fim de permitir a criação de um histórico dos dados antropométricos e com base nisso permitir gerar predições sobre o estado futuro das crianças.

b) Planejamento

Definição de recursos e estrutura do software a ser desenvolvido.

Estudo sobre o desenvolvimento de algoritmos de programação genética e processos de regressão simbólica.

Estudo sobre o desenvolvimento de algoritmos no software Malab.

Estudo do código do algoritmo apresentado em Ascari, Borsoi e Fávero (2013).

Definição dos requisitos técnicos para o funcionamento (dados essenciais para armazenamento, migração do algoritmo existente em Matlab para o ambiente de desenvolvimento Delphi, etc.).

Definição e modelagem dos requisitos do sistema por meio de casos de uso e diagrama de entidades e relacionamentos.

c) Desenvolvimento

Modelagem do software e implementação das funcionalidades utilizando o ambiente de desenvolvimento Delphi, visando apresentar o uso das tecnologias no desenvolvimento do software.

d) Testes

Os testes de navegabilidade, funcionalidade e compatibilidade foram realizados pelo autor deste trabalho e pela professora orientadora.

4 RESULTADOS

Neste capítulo são apresentados os resultados da realização do trabalho, que apresenta a modelagem e a implementação de um software para acompanhamento e predição de dados antropométricos infantis, visando mostrar o uso das tecnologias utilizadas no desenvolvimento do sistema.

4.1 ESCOPO DO SISTEMA

O sistema é um aplicativo de apoio a tomada de decisão, que tem como objetivo principal registrar os dados antropométricos de crianças em intervalos de tempo diversos, e com base nesses dados gerar uma predição de seu desenvolvimento futuro.

O sistema permitirá:

- a) Cadastrar responsáveis.
- b) Cadastrar crianças relacionadas a responsáveis.
- c) Cadastrar consultas e dados antropométricos identificados na data.
- d) Gerar predições com base no histórico de dados antropométricos registrados no sistema para determinada criança.
- f) Pesquisar crianças, consultas e responsáveis.
- g) Visualizar na forma de relatórios e gráficos os dados antropométricos registrados para cada criança.

4.2 MODELAGEM DO SISTEMA

A modelagem do sistema foi feita com base na UML (*Unified Modeling Language*). Booch, Rumbaugh e Jacobson (2006, p. 14) afirmam que “a UML é uma linguagem-padrão para a elaboração da estrutura de projetos de *software*. Ela poderá ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam o uso de sistemas complexos de *software*”.

Os principais requisitos definidos para o sistema são:

- 1) O sistema deverá permitir que o usuário:
 - a) realize o cadastro de seus pacientes (crianças) e responsáveis.
 - b) realize o cadastro das consultas realizadas para determinado paciente.
 - c) consulte os dados cadastrados por meio de relatórios e gráficos.
 - d) gere uma predição dos dados antropométricos futuros da criança, com base no histórico registrado a cada consulta e salve os dados gerados.
- 2) O sistema deverá fornecer relatórios de:
 - a) pacientes (crianças).
 - b) responsáveis.
 - c) consultas.

A Figura 2 apresenta o diagrama de casos de uso definido para o sistema.

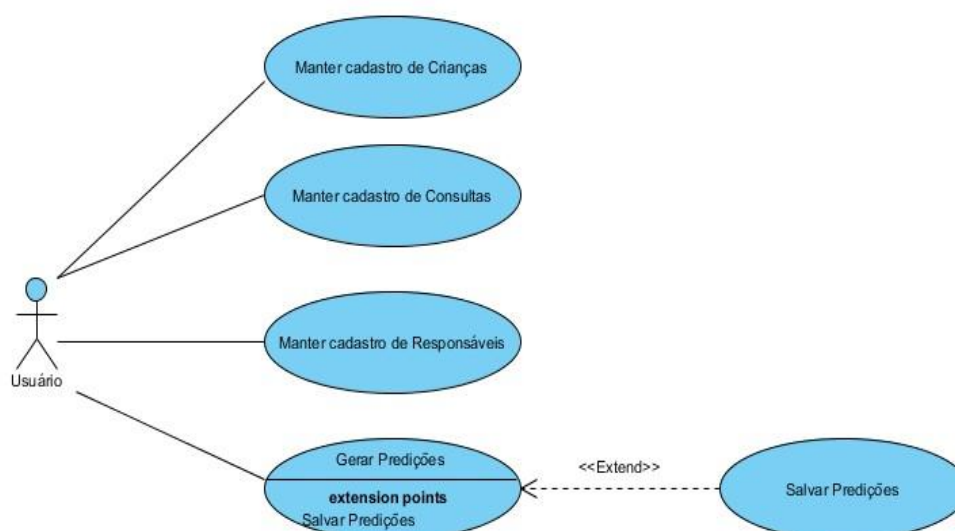


Figura 3 - Diagrama de Casos de Uso

Os quadros 2, 3, 4 e 5 apresentados a seguir representam as descrições dos casos de uso apresentados na Figura 2.

Identificador do requisito: Gerar Predição.

Descrição: Permite que o usuário gere uma predição do peso ou altura da criança.

Evento Iniciador: Tela inicial.

Atores: Usuário.

Pré-condição: Ter mais de uma consulta cadastrada para a criança.

Sequência de Eventos:

1 – Usuário seleciona uma criança e clica em gerar predição.

2 – Sistema gera a predição com base nos dados cadastrados das consultas da criança.

3 – Sistema retorna os pesos preditos até o 12º mês de idade da criança.

Pós-Condição: É gerado um gráfico comparando os dados cadastrados com os preditos.
Inclusões: Salvar predição.

Quadro 2 – Caso de uso Gerar predição

Identificador do requisito: Manter cadastro de crianças.
Descrição: Permite que o usuário cadastre crianças no sistema, informando nome, data de nascimento, sexo e o responsável.
Evento Iniciador: Tela inicial.
Atores: Usuário.
Pré-condição: Ter um responsável cadastrado.
Sequência de Eventos:
 1 – Usuário abre a tela de Cadastro de Crianças.
 2 – Usuário informa os dados da criança.
 3 – Sistema salva a criança no banco de dados e informa se obteve sucesso.
Pós-Condição: Nenhuma.
Inclusões: Nenhuma.

Quadro 3 - Caso de uso Manter cadastro de crianças

Identificador do requisito: Manter cadastro de consultas.
Descrição: Permite que o usuário cadastre consultas no sistema, informando a criança, o peso, a altura e a data da consulta.
Evento Iniciador: Tela inicial.
Atores: Usuário.
Pré-condição: Ter uma criança cadastrada.
Sequência de Eventos:
 1 – Usuário abre a tela de Cadastro de Consultas.
 2 – Usuário insere as informações da consulta.
 3 – Sistema salva a consulta no banco de dados e informa se obteve sucesso.
Pós-Condição: Nenhuma.
Inclusões: Nenhuma.

Quadro 4 - Caso de uso Manter cadastro de consultas

Identificador do requisito: Manter cadastro de responsáveis.
Descrição: Permite que o usuário cadastre responsáveis (pais da criança) no sistema, informando nome, CPF, email, telefone, estado e cidade.
Evento Iniciador: Tela inicial.
Atores: Usuário.
Pré-condição: Nenhuma.
Sequência de Eventos:
 1 – Usuário abre a tela de Cadastro de Responsáveis.
 2 – Usuário insere os dados do responsável.
 3 – Sistema salva o responsável no banco de dados e informa se obteve sucesso.
Pós-Condição: Nenhuma.
Inclusões: Nenhuma.

Quadro 5 - Caso de uso Manter cadastro de responsáveis

A Figura 3 apresenta o Diagrama de entidades e relacionamentos (DER) criado para o sistema.

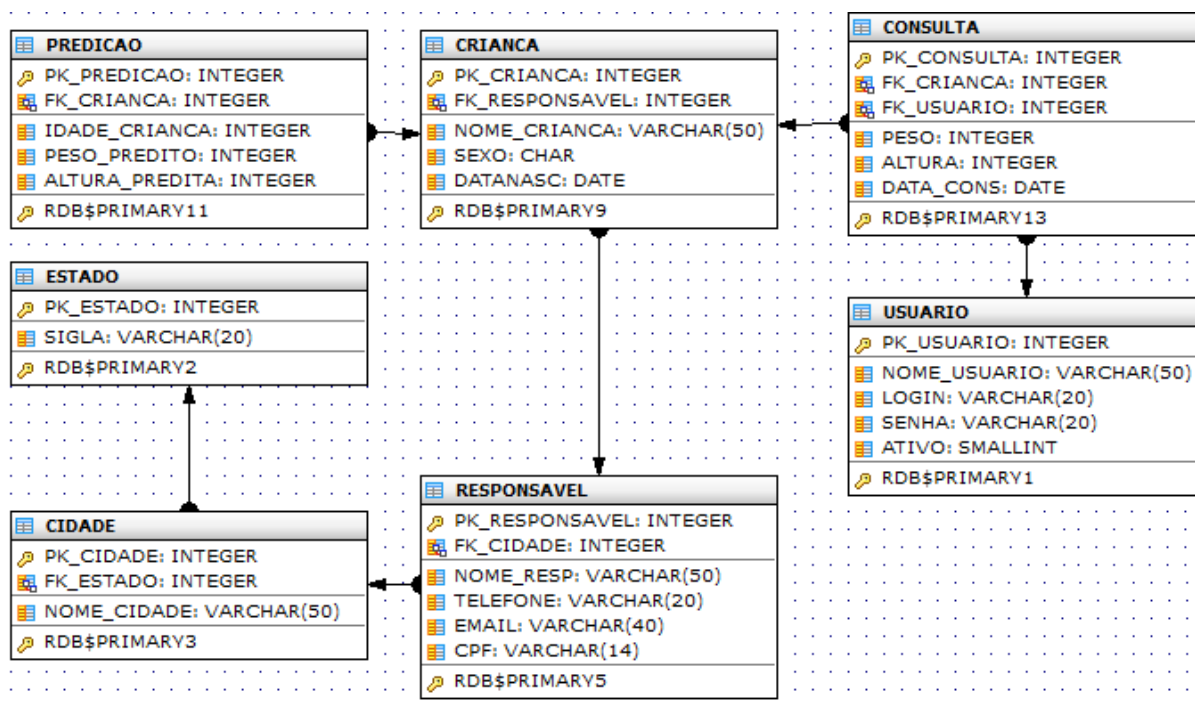


Figura 4 - DER mostrando as tabelas do sistema.

4.3 APRESENTAÇÃO DO SISTEMA

A Figura 4 apresenta a tela de login do sistema, que é a primeira a ser apresentada ao usuário.



Figura 5 - Tela de login do sistema

A Figura 5 apresenta a tela inicial do sistema, a partir da qual é possível acessar as funcionalidades disponíveis para registro e acompanhamento dos dados antropométricos infantis.

Os cadastros e funcionalidades do sistema são acessados por meio do menu disponível na parte superior da tela inicial.



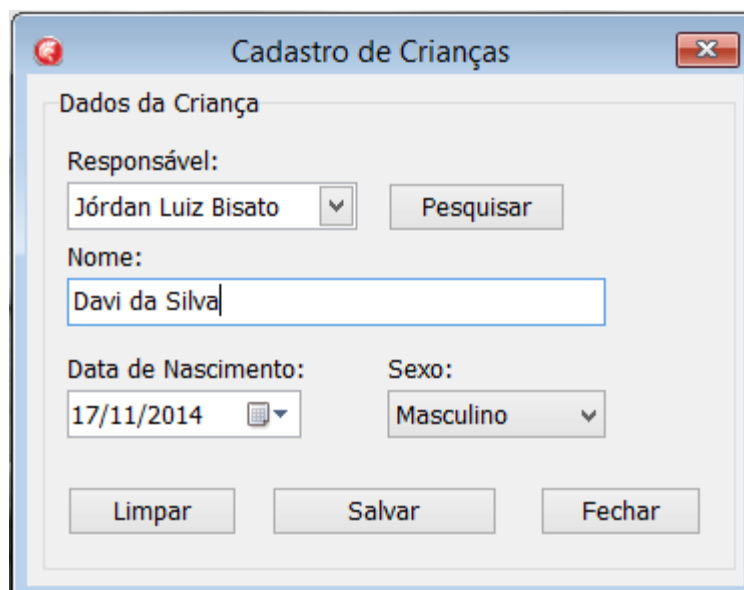
Figura 6 - Tela principal do sistema

A Figura 6 apresenta o cadastro do responsável pela criança, ou seja, pai, mãe ou responsável legal.

A screenshot of a web form titled 'Cadastro de Responsáveis'. The form is titled 'Dados do Responsável' and contains the following fields: 'Nome:' (text input), 'CPF:' (text input with a mask of ___-__-____-__), 'Email:' (text input), 'Telefone:' (text input with a mask of (____) ____-____), 'Estado:' (dropdown menu), and 'Cidade:' (dropdown menu). At the bottom of the form, there are three buttons: 'Limpar', 'Salvar', and 'Fechar'.

Figura 7 - Cadastro de Responsáveis

A Figura 7 apresenta o cadastro da criança, onde é feito o relacionamento com o responsável, e informado seu nome, data de nascimento e sexo.



Cadastro de Crianças

Dados da Criança

Responsável:
Jórdan Luiz Bisato

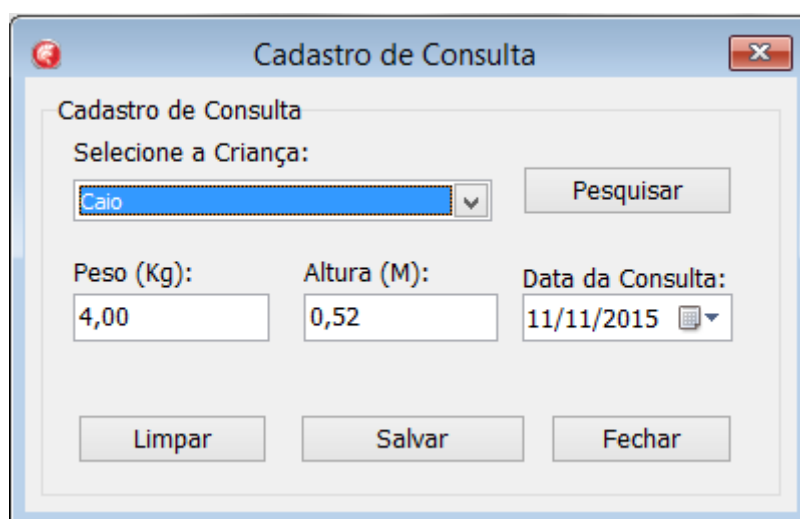
Nome:
Davi da Silva

Data de Nascimento: 17/11/2014

Sexo: Masculino

Figura 8 - Cadastro de Crianças

A Figura 8 apresenta o cadastro de consultas, onde são registrados o peso e altura da criança em determinada data.



Cadastro de Consulta

Cadastro de Consulta

Selecione a Criança:
Caio

Peso (Kg): 4,00

Altura (M): 0,52

Data da Consulta: 11/11/2015

Figura 9 - Cadastro de Consultas

A Figura 9 apresenta o cadastro de predições, onde é possível selecionar uma determinada criança, visualizar o histórico de consultas registrados para ela e então gerar uma predição.

Cadastro de Predições

Pesquisa de Criança

Criança: Lara B

Pesquisar

Consultas

ID	Nome da Criança	Peso	Altura	Data da Consulta	on
17	Lara B	2,540 KG	0,75 M	10/11/2015	
21	Lara B	4,570 KG	0,90 M	10/12/2015	
22	Lara B	5,180 KG	1,05 M	11/01/2016	
23	Lara B	5,820 KG	1,20 M	10/02/2016	

Funções

Gerar predição de peso Gerar predição de altura

Gerar Predição

Figura 10 - Cadastro de Predições

A Figura 10 apresenta a tela de pesquisa de consultas. Há telas neste padrão também para pesquisar crianças e responsáveis.

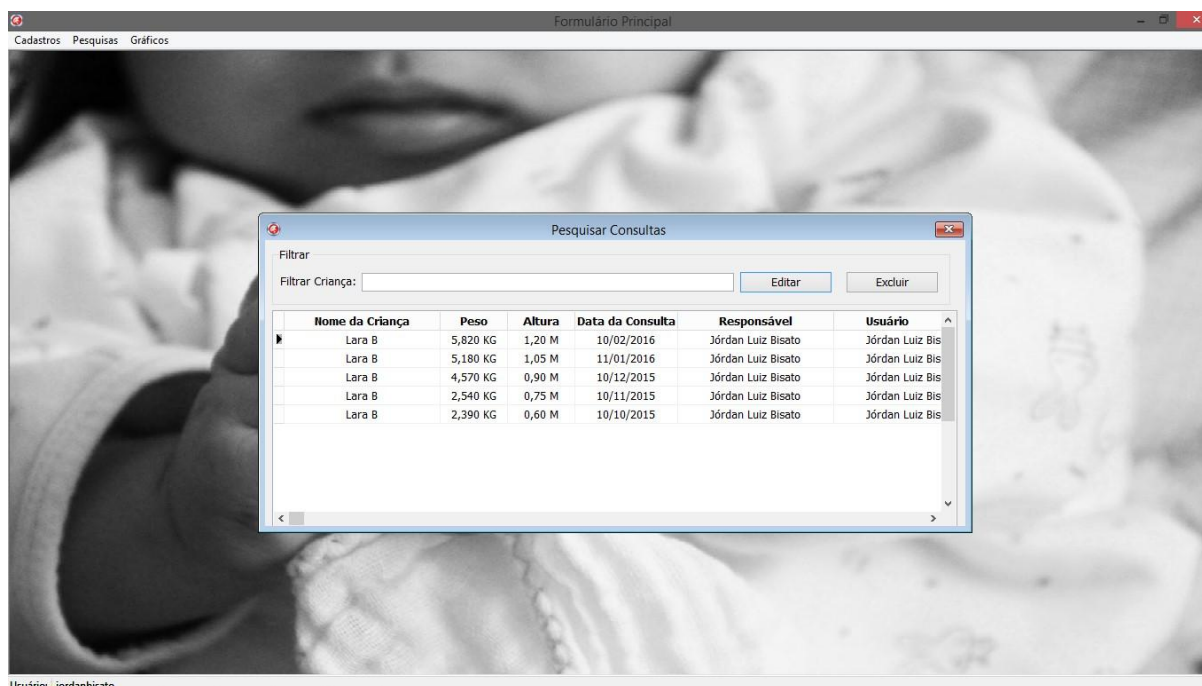


Figura 11 - Cadastro de Predições

A Figura 11 apresenta o gráfico de uma predição de peso gerada para uma criança, com base no peso registrado em 4 consultas. Com base nesse histórico

registrado, o algoritmo de programação genética identificou uma equação matemática que permitiu gerar uma curva próxima à correspondente aos dados do histórico. Com base nesta equação é gerado uma predição de como seria a tendência de evolução do peso da criança nos meses seguintes. Por padrão o sistema gera uma predição considerando meses de 1 a 12, e esses dados são apresentados em um gráfico, em paralelo com os dados do histórico da criança. Em geral, após a criança completar um ano de idade, as consultas pediátricas são realizadas com uma frequência menor, como por exemplo a cada três meses.

Nesse exemplo de predição, a equação matemática apresentada pelo algoritmo de programação genética corresponde a uma curva pouco próxima dos dados de treinamento, que não é uma situação ideal. Contudo, foram realizados vários testes no algoritmo, variando o tamanho de cromossomos, profundidade das equações, número de gerações, entre outras características, e ainda assim, não foram obtidos resultados muitos satisfatórios, ou seja, o algoritmo ainda precisa ser melhor analisado e ajustado, para que possa gerar equações mais adequadas.

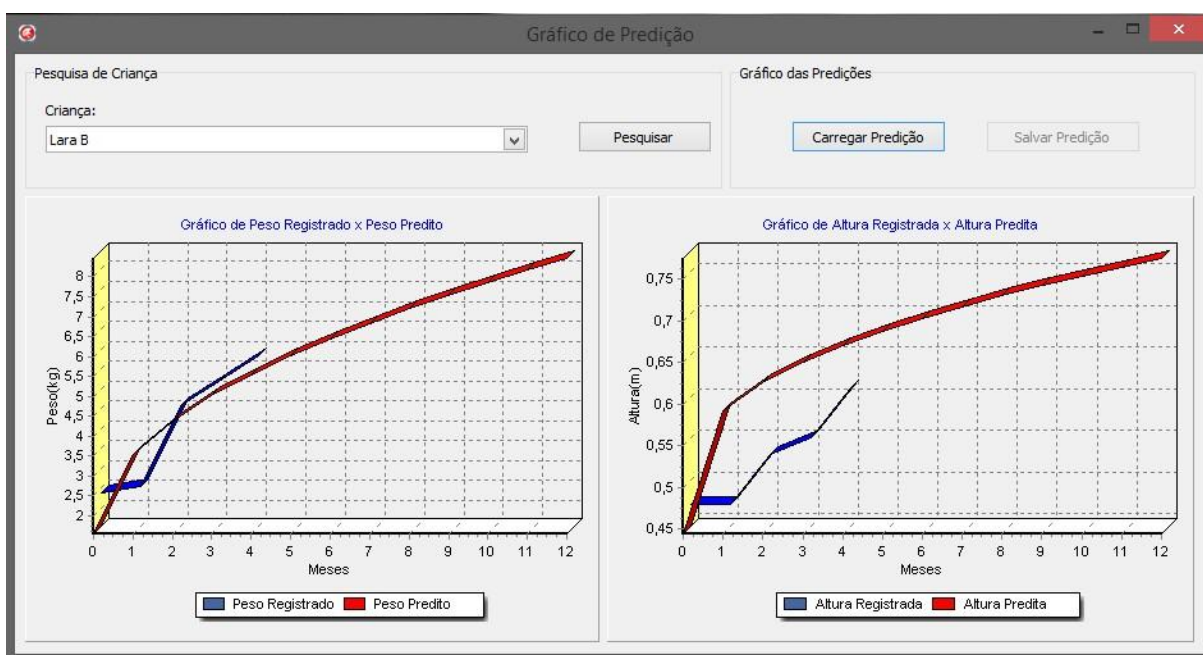


Figura 12 - Gráfico de um exemplo de predição gerada.

A figura 12 demonstra os resultados obtidos da predição durante o *debug* do programa.

```

Res := Resultados(equacao, X, 13);
Res (3072, 3720, 4368, 5016, 5664, 6312, 6960, 7608, 8256, 8904, 9552, 10200, 10848)
// [1] 3720
// [2] 4368
// [3] 5016
// [4] 5664
Res [5] 6312
end;
// [6] 6960
// [7] 7608
funct ... pulacao: ArrInteiro): Integer; overload;
var ler [9] 8904
ler [10] 9552
[11] 10200
[12] 10848
Code History

```

Figura 13 - Resultado da predição.

4.4 IMPLEMENTAÇÃO DO SISTEMA

As opções de inclusão, consulta, alteração e exclusão dos dados da tabela foram feitas com base em stored procedures. O código apresentado no Quadro 3 apresenta um exemplo de stored procedure usada para inclusão de registros na tabela CRIANCA.

```

CREATE OR ALTER PROCEDURE SP_INS_CRIANCA (
    nome_crianca varchar(50),
    sexo char(1),
    datanasc date,
    fk_responsavel integer)
as
BEGIN
    INSERT INTO CRIANCA (NOME_CRIANCA, SEXO, DATANASC, FK_RESPONSAVEL)
    VALUES (:NOME_CRIANCA, :SEXO, :DATANASC, :FK_RESPONSAVEL);
END

```

Quadro 6 - Código da stored procedure SP_INS_CRIANCA

A opção de auto-incremento no banco de dados Firebird não é automática, então é necessário criar um generator, conforme apresentado no Quadro 4, e uma trigger como apresentada no Quadro 5. A trigger é executada no evento *beforeInsert* da tabela desejada, que fará uso do generator para incrementar o código da chave primária da tabela.

```

CREATE SEQUENCE GEN_CRIANCA_ID;
ALTER SEQUENCE GEN_CRIANCA_ID RESTART WITH 1;

```

Quadro 7 - Código do generator SP_INS_CRIANCA

```
CREATE OR ALTER trigger crianca_bi for crianca
active before insert position 0
as
begin
  if (new.pk_crianca is null) then
    new.pk_crianca = gen_id(gen_crianca_id,1);
  end
end
```

Quadro 8 - Código da stored procedure SP_INS_CRIANCA

A seguir são apresentadas algumas listagens de código, exemplificando funcionalidades implementadas no sistema de predição de dados antropométricos desenvolvido.

A Listagem 1 apresenta o trecho de código responsável pela geração do *fitness* (que é a capacidade de um indivíduo em resolver um problema), onde a função recebe os pesos preditos e faz a verificação para saber se não houve nenhum retorno com "-1" (que indica que houve erro na predição de algum mês) ou se não há predições iguais para meses diferentes. Após a verificação é feito o cálculo do *fitness* utilizando o erro de todos os pontos.

```
Predicao := Resultados(equacao, X, nRow);
// Calcula o erro do valor com base no resultado da equação
SetLength(vet, nRow);
for k := 0 to nRow - 1 do
  begin
    // verifica se o erro anterior é igual ao atual, o que identifica que foram geradas
    // Predicoes iguais para meses diferentes( pode acontecer em equacoes com 1 coeficiente)
    if (Predicao[k] = -1) or ((k > 1) and (Predicao[k] = Predicao[k - 1])) or
      ((k > 1) and ((Y[k] > Y[k - 1]) and ((Predicao[k] < Predicao[k - 1]))))
      or ((Y[k] < (Y[k - 1])) and ((Predicao[k]) > (Predicao[k - 1]))))
    then
      begin
        fit := -1;
        break;
      end;
    // Guarda o valor do erro
    vet[k] := Power((Abs(Y[k] - Predicao[k])), 2);
    // Soma o erro de todos os pontos (k)
    Erro := Erro + vet[k];
  end;

if fit = 0 then // se verdadeiro, a equação é válida
  begin
    // Divide o erro de todos os pontos, gerando o EQM e soma com o erro do ultimo ponto,
    // para que nos pontos finais a curva fique mais proxima do desejado
    fit := Sqrt(Erro + vet[nRow - 1]) / nRow;
  end
else
  begin
    // Se trouxe valores iguais para todos os pontos está errado
    fit := 999999;
  end;
end;
```

Listagem 1 - Código da Geração de Fitness

A Listagem 2 apresenta o código utilizado para a inicialização da população,

onde é passado para a função *Grow* a profundidade atual do Ramo, o índice, um array vazio e a profundidade máxima do ramo, o retorno desta função é um membro da população, que é formado por números entre 10 e 10, gerados aleatoriamente. Após isso o membro é colocado na matriz de população.

```
nTerminais := Length(vetTerminais);
SetLength(vetPop, 1, 0);
I := 0;
SetLength(Ramo, 0);
while I < nCromossomos - 1 do
begin
  //É passado para a função Grow a Profundidade atual do Ramo, o índice,
  //Um array vazio e a Profundidade maxima do Ramo
  Ramo := Grow(0, 1, Ramo, nProfundidade);
  c := Length(Ramo);

  //Feita verificação pra evitar Ramos com
  //tamanho menor que 6, para evitar predições com poucas funções e terminais
  while c < 6 do
  begin
    //Sempre zera o Ramo antes de criar outro.
    SetLength(Ramo, 0);
    Ramo := Grow(0, 1, Ramo, nProfundidade);
    c := Length(Ramo);
  end;

  tam := Length(vetPop[0]);

  //Verifica se o tamanho do Ramo não é maior que a matriz de população
  if c > tam then
  begin
    SetLength(vetPop, I + 1, c);
  end
  else
  begin
    SetLength(vetPop, I + 1, tam);
  end;

  //Adiciona o ramo na população
  for J := 0 to c - 1 do
  begin
    vetPop[I][J] := Ramo[J];
  end;

  //Incrementa o índice da população, após a inserção
  I := I + 1;
end;
```

Listagem 2 - Código da inicialização da população.

Na listagem 3, tem-se a primeira parte da função que organiza a equação, colocando os parênteses de forma que as funções unárias e binárias tenham o número correto de terminais em cada uma, evitando desta forma que uma equação esteja inadequada por erro de sintaxe.

Como podem-se observar no código abaixo, o objetivo é separar a equação em partes, utilizando a função FimEquação para descobrir em qual índice da equação as funções binárias e unárias tem o número de terminais suficientes, separando o primeiro termo até este ponto.

```

len := Tamanho(vetEquacao);
I := 1;
l := 0;
Membro := False;

while I <= len do
begin
  //Verifica se há alguma função dentro da equação
  for k := 0 to Length(vetFuncoes) - 1 do
  begin
    if vetEquacao[0][I - 1] = vetFuncoes[k] then
    begin
      Membro := true;
      break;
    end;
  end;

  if (len = I) and (I <> 1) then
  begin
    break;
  end;

  if Membro then
  begin
    Membro := False;
    sfuncao := IntToStr(vetEquacao[0][I - 1]);
    //Verifica qual o índice do fim da equação
    idx := FimEquacao(vetEquacao, I + 1, 1);

    //Evita o bug do loop infinito em certas situações
    if (idx = 0) and ((Length(sValor)) > 150) then
    begin
      break;
    end;

    //Define o tamanho da variavel temporaria que irá receber a equacao até o índice
    if (idx - (I + 1) + 1) >= 0 then
    begin
      SetLength(tpEquacao, 1, (idx - (I + 1) + 1));
    end
    else
    begin
      SetLength(tpEquacao, 1, 0);
      break;
    end;

    for k := I + 1 to idx do
    begin
      tpEquacao[0][1] := vetEquacao[0][k - 1];
      l := l + 1;
    end;
    l := 0;

    //É organizada a 1ª parte da equação, utilizando recursividade da função Organiza
    Termol := Organiza(tpEquacao);
    J := idx;
  end;
end;

```

Listagem 3 - Código da função usada para organizar equação (Parte 1)

A listagem de número 4 apresenta a segunda parte da função Organiza, onde após organizar o primeiro termo, é necessário trabalhar com a outra parte da equação após o índice. Para o segundo Termo o processo de organizar é muito semelhante ao realizado para o primeiro.

```

for k := 0 to Length(vetFuncUnarias) - 1 do
begin
  if (StrToInt(sfuncao) = vetFuncUnarias[k]) then
  begin
    Membro := true;
    break;
  end;
end;

if Membro then
begin
  //Se for uma função unaria, a string recebe o 1º termo organizado da equação
  Termol := Concat(sfuncao, Concat('(', Concat(Termol, ')')));
  sValor := Concat(sValor, Termol);
end
else
begin
  //Caso contrario, obtenha o 2º termo da equação,
  //começando após o indice de fim do 1º, para organizar
  idx := FimEquacao(vetEquacao, J + 1, 1);
  if ((idx - (J + 1)) + 1) >= 0 then
  begin
    SetLength(tpVetEq, 1, (idx - (J + 1)) + 1);
  end
  else
  begin
    SetLength(tpVetEq, 1, 0);
  end;

  for k := J + 1 to idx do
  begin
    tpVetEq[0][1] := vetEquacao[0][k - 1];
    l := l + 1;
  end;
  l := 0;

  Termo2 := Concat(Organiza(tpVetEq, ')');
  Termo2 := Concat('(', Concat(Concat(Concat(Concat('(', Termol), ')'),
    sfuncao), Concat(Concat('(', Termo2), ')')));
  sValor := Concat(sValor, Termo2);
end;

if idx = 0 then
begin
  if J = 0 then
  begin
    I := len;
  end
  else
  begin
    I := J;
  end;
end
else
begin
  I := idx;
end;
end
else
begin
  //Se não houver mais nenhuma função dentro da equação, concatena a equação na string
  sValor := Concat(sValor, IntToStr(vetEquacao[0][I - 1]));
end;
I := I + 1;
end;
Result := sValor;

```

Listagem 4 - Código da função de organizar equação (Parte 2)

A listagem 5 demonstra partes da função utilizada para verificar qual índice corresponde ao fim da equação, onde primeiramente a equação é transformada em string, após isto verifica se é uma função binária, e caso seja, o contador é acrescido em 2, pois são necessários 2 números para uma função binária, o mesmo ocorre com as funções unárias, a única diferença é que é necessário apenas 1 número para a função. No fim, quando o contador chega a 0, sabe-se que ali é o fim da equação, onde todas as funções binárias e unárias têm o número correto de terminais.

```

if concatenar = 1 then
begin
len := Tamanho(sEquacao);
for I := 0 to len - 1 do
begin
//Caso o número da equação seja negativo, substitui por uma letra
if (sEquacao[0][I] < 0) and (sEquacao[0][I] >= (-10)) then
begin
sNova := Concat(sNova, vetTemp[((sEquacao[0][I]) * (-1)) - 1]);
end
else if (sEquacao[0][I] >= 0) and (sEquacao[0][I] <= 10) then
begin
//Se não, apenas adiciona o número na string
sNova := Concat(sNova, IntToStr(sEquacao[0][I]));
end;
end;
end;
n := 1;
idx := 0;

len := Length(sNova);
for I := Inicio to len do
begin
Membro := False;
//Verifica se o caractere representa uma função binária ('*', '/', '^'...)
for k := 0 to Length(vetFuncBinarias) - 1 do
begin
if sNova[I] = IntToStr(vetFuncBinarias[k]) then
begin
Membro := true;
break;
end;
end;

if Membro then
begin
//Caso seja, necessitamos de dois números para esta função, portanto n := + 2
n := n + 2;
end
else
begin
//Verifica se o caractere representa uma função unária ('+', '-', 'Exp'...)
for k := 0 to Length(vetFuncUnarias) - 1 do
begin
if sNova[I] = IntToStr(vetFuncUnarias[k]) then
begin
Membro := true;
break;
end;
end;

if Membro then
begin
//Caso seja unária, precisamos de apenas 1 número pra esta função, então n := +1
n := n + 1;
end
end;

n := n - 1;
if (n = 0) then
begin
//Se n chegar a 0, quer dizer que todas as funções da equação estão completas
//Então setamos o Índice como o ultimo caractere válido,
//para informar que este índice é o fim da equação.
idx := I;
break;
end;
end;
end;

```

Listagem 5 - Código para identificar o Fim da Equação

4.5 DISCUSSÕES

O projeto apresentado foi um grande desafio para o autor, visto que o software criado envolve conhecimentos tanto da área de programação como da área

biológica de dados antropométricos, o que custou mais tempo e pesquisa para compreender todo o processo e implementá-lo. Uma das grandes dificuldades foi a necessidade da utilização de matrizes, pois a linguagem de programação usada (Delphi) não tem muitos recursos avançados para a manipulação deste tipo de dado, se comparada ao software Matlab.

A predição dos dados antropométricos também exigiu muitas horas de teste e revisão, pois apesar de tudo aparentar estar correto, as predições não apresentavam dados próximos aos cadastrados, e como muitas partes do cálculo são geradas de forma aleatória, dificultava ainda mais a análise dos testes e *debug* para a solução dos erros. Além disso, o programa demorava muito tempo para gerar a predição, visto que o código é extenso, tem funções recursivas, e muitas estruturas de repetição.

5 CONCLUSÃO

O projeto apresentou a modelagem e a implementação de um software para acompanhamento e predição de dados antropométricos infantis. Esse projeto possibilita que profissionais da área da saúde, especificamente pediatras, e/ou familiares acompanhem o crescimento das crianças, permitindo criar uma previsão do peso dos pacientes até o seu primeiro ano de idade, além do cadastro de consultas, crianças e responsáveis.

Este trabalho foi possível graças às funcionalidades presentes nas ferramentas utilizadas, como o Visual Paradigm na criação do diagrama de casos de uso, Firebird Maestro na criação do diagrama de entidades e relacionamentos, o MatLab para estudo do algoritmo de programação genética utilizado como base para o software, o banco de dados Firebird para armazenar os dados dos pacientes e consultas, o Delphi como interface de programação e compilador do código utilizado para implementação do programa.

Ao final deste projeto, é possível afirmar que o objetivo inicial foi alcançado, apesar dos problemas que ainda ocorrem, como a demora em realizar a predição e o fato de que nem todos os resultados obtidos são satisfatórios. O programa realiza com sucesso o seu objetivo que é registrar os dados antropométricos de crianças a cada consulta pediátrica realizada nos seus meses iniciais de vida, permitir a consulta desses dados, e criar uma predição de peso ou altura dessa criança com base nos dados cadastrados.

Considera-se que o emprego deste software para uso comercial ainda não é eficiente, sendo necessário mais ajustes no algoritmo, além dos que já foram efetuados, como as validações para filtragem de resultados e os ajustes no código para evitar processamento desnecessário, a fim de melhorar a performance e a porcentagem de predições satisfatórias.

REFERÊNCIAS

ARAÚJO, A. C. T., CAMPOS, J. A. D. B. **Subsídios para a avaliação do estado nutricional de crianças e adolescentes por meio de indicadores antropométricos**. Alim. Nutr, Araraquara. 2008; 19(2):219-25.

ASCARI, R. E. O. S., BORSOI, B. T. e FÁVERO, E. M. B. **Algoritmo evolutivo para predição de dados antropométricos infantis como apoio à tomada de decisão**. Journal of Health Informatics. 2013; São Paulo, v. 5, n. 1. ISSN 2175-4411.

BOOCH, J., RUMBAUGH, J, JACOBSON, I. **UML: Guia do Usuário**. 2a Edição. Campus, 2006.

CANTU, H. **Conheça o Firebird em 2 minutos**. 2010. Disponível em: <http://www.firebirdnews.org/docs/fb2min_ptbr.html>. Acesso em 13/06/2016.

CRAMER N. **A representation for the adaptive generation of simple sequential programs**. Proceedings of the International Conference on Genetic Algorithms and their Applications; 1985 Jul 24-26; Carnegie-Mellon University, Pittsburgh, (USA). p.183-7.

FERNANDES, J. F. A. **Prática da avaliação física**. Rio de Janeiro: Shape; 1999. 3.

HOLLAND, J. H. **Adaptation in natural and artificial systems**. Ann Arbor, MI: The University of Michigan Press; 1975.

KOZA, J. R. **Hierarchical genetic algorithms operating on populations of computer programs**. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence; 1989 Aug 24-26; Detroit (USA). p. 768-74.

KOZA, J. R. **Genetic programming: on the programming of computers by means of natural selection**. Massachusetts: MIT Press; 1992.

LOPES, M. C. L. Crescimento de recém-nascidos pré-termos tardios nos primeiros seis meses de idade corrigida em Cuiabá-MT. 2014. Tese de Doutorado em Ciências. Faculdade de Medicina da Universidade de São Paulo.

MINISTÉRIO DA SAÚDE - **Caderneta de saúde da criança**. 2013. Disponível em: http://bvsms.saude.gov.br/bvs/publicacoes/caderneta_saude_crianca_menino.pdf. Acesso em 29/06/2016.

PETELETRICA. **Tutorial Matlab Básico – 1ª parte**. Disponível em: <<http://www.peteletrica.eng.ufba.br/?p=991>>. Acesso em: 18/11/2015.

SOFTWARE. **Visual Paradigm**. Disponível em: <<http://www.software.com.br/p/visual-paradigm-para-uml?gclid=CMCOoefZmskCFVWakQodw6kO-Q>>. Acesso em: 18/11/2015.

SOFTWARE. **Firebird Maestro**. Disponível em: <http://pt.softoware.net/apps/download-firebird-maestro-for-windows.html>. Acesso em

07/06/2016.

SUTTASUPA, Y., RUNGRAZUNGSILP, S., PINYOPAN, S., WUNGCHUSUNTI, P., CHONGSTITVATANA, P. A. **Comparative study of linear encoding in genetic programming**. Proceedings of the Ninth International Conference on ICT and Knowledge; 2011 Nov. 24-25; Bangkok, Thailand. p.13-7.

VASCONCELOS F. A. G. **Avaliação nutricional de coletividades**. 3a. ed. Florianópolis: Editora UFSC; 2000.WHO – Whold Health Organization. How much physical activity needed to improve and maintain health [Internet]. 2002 [citado 2012 ago 13]. Disponível em: www.who.int/hpt/physactiv/p.a.how.much.html.

ZEFERINO AMB. **Acompanhamento do crescimento**. J Pediat (Rio J). 2003;79(Supl1): S23-S32.