



**GOVERNO DO
ESTADO DO
TOCANTINS**

108 Sul, Alameda 11, Lote 03, Casa Postal 173, Palmas - Tocantins - Cep: 77020-122 - Tel: +55 63 3218-2939 - www.unitins.br

**UNIVERSIDADE ESTADUAL DO TOCANTINS
SISTEMAS DE INFORMAÇÃO**

ALEXANDRA MESQUITA DO NASCIMENTO VALADARES

**UTILIZAÇÃO DE ALGORITMO GENÉTICO PARA DESENVOLVIMENTO DE UM
SISTEMA DE ESCALA DE TRABALHO**

**PALMAS - TO
2019**

ALEXANDRA MESQUITA DO NASCIMENTO VALADARES

UTILIZAÇÃO DE ALGORITMO GENÉTICO PARA DESENVOLVIMENTO DE UM SISTEMA DE ESCALA DE TRABALHO

Trabalho de Conclusão de Curso de Sistemas de Informação da Universidade Estadual do Tocantins, apresentado como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação.

**PALMAS - TO
2019**

Dados Internacionais de Catalogação na Publicação (CIP)

**Biblioteca da Universidade Estadual do Tocantins
Câmpus Graciosa – Palmas-TO**

- V136u Valadares, Alexandra Mesquita do Nascimento.
Uso de algoritmo genético para desenvolvimento de escala de trabalho na Segunda Igreja Batista em Palmas (SIBAPA) / Alexandra Mesquita do Nascimento Valadares. – Palmas, TO, 2019.
80f. il.col.
Inclui CD-ROM.
- Monografia (TCC) – Universidade Estadual do Tocantins, Câmpus Graciosa, Curso de Sistemas de Informação, 2019.
Orientador (a): Profº. Alex Coelho.
1. Sistemas de informação. 2. Algoritmo genético. 3. Inteligência artificial. 4. Escala – Equipe de trabalho. I. Título.

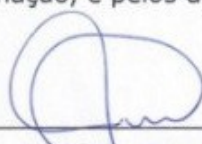
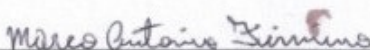
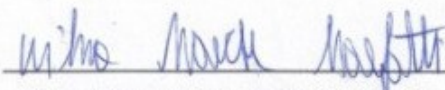
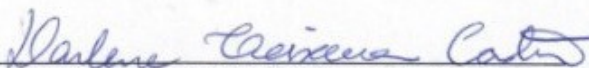
CDD 006.337

Bibliotecária: Cacilda Martins Madureira CRB-2 / 0561

Todos os Direitos Reservados – A reprodução parcial, de qualquer forma ou por qualquer meio deste documento é autorizado desde que citada à fonte. A violação dos direitos do autor (Lei nº 9.610/98) é crime estabelecido pelo artigo 184 do código penal.

**UNIVERSIDADE ESTADUAL DO TOCANTINS
PRÓ-REITORIA DE GRADUAÇÃO****ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO**

Aos dezoito dias do mês de junho, reuniu-se a banca examinadora do trabalho apresentado como Trabalho de Conclusão de Curso em **Sistemas de Informação** de **ALEXANDRA MESQUITA DO NASCIMENTO VALADARES**, intitulada: **UTILIZAÇÃO DE ALGORITMO GENÉTICO PARA DESENVOLVIMENTO DE UM SISTEMA DE ESCALA DE TRABALHO**. Compuseram a banca examinadora os professores **Me. Alex Coelho**, **Me. Marco Antônio Firmino** e **Me. Silvano Maneck Malfatti**. Após a exposição oral, o (a) candidato(a) foi arguido(a) pelos componentes da banca que reuniram-se reservadamente, e decidiram, **APROVADO**, com o conceito 8,5 ao trabalho. Para constar, redigi a presente Ata, que aprovada por todos os presentes, vai assinada por mim, Coordenadora de TCC do curso de Sistemas da Informação, e pelos demais membros da banca.

**Me. Alex Coelho****Me. Marco Antônio Firmino****Me. Silvano Maneck Malfatti****Coordenação de TCC/Sistemas da Informação**

Agradecimentos

Agradeço primeiramente a Deus, por ter me proporcionado a oportunidade de realizar este curso.

Ao João Neto, meu esposo, obrigada por ter aguentado todos os meus estresses, lamentos; obrigada pelo apoio em todas as situações, por se manter ao meu lado nos momentos de choro e desespero. Obrigada por me deixar tentar e errar várias vezes e, quando via que não ia para frente, dava seu ponto de vista, sentava ao meu lado e me ajudava a solucionar os problemas que para mim eram insolucionáveis.

Aos meus pais Marilene L. Mesquita e João Ivaldo Pereira do Nascimento, que mesmo a distância me apoiou e incentivou em todas as situações, me ligando e perguntando como estava e dizendo que eu conseguiria, pois, o Senhor está comigo o tempo todo.

Aos meus filhos Luccas Emannuel e Caleb, sou grata pela paciência a mim dedicada, onde por muitas vezes, abriram mão do tempo que era para ser de vocês para que eu pudesse cumprir atividades referentes ao curso.

Ah, aos meus professores, meu muito obrigada a cada um de vocês, pois me fizeram entender que eu era capaz de solucionar os problemas propostos em sala; obrigada pelos incentivos, palavras de motivação, por sempre me ouvirem e contribuírem da melhor forma para que eu realizasse os meus objetivos.

Ao professor Douglas Chagas, sou grata pois, encontrou em meu coração um carinho de filha; o mesmo com sua grande generosidade, trata seus alunos de tal forma, ajudando a quem precisa, chamando atenção na hora certa e onde estava sempre a postos a responder e-mails tirando as dúvidas de seus educandos.

Aos professores Marco Antônio Firmino e Silvano Malfatti, obrigada por verem e o meu potencial, me mostrado que sou capaz de muito mais; obrigada pelos esforços exigidos e apoio nos momentos difíceis, sou grata por nos acompanhar desde o início.

A professora Arlenes Delabary Spada, obrigada por sempre me ouvir e me apoiar nos momentos em que eu, por muitas vezes, pensei em desistir, por compartilhar comigo nos momentos difíceis desse trajeto.

Ao meu Orientador Alex Coelho, que escutava meus lamentos e mesmo assim, me apoiou e incentivou a continuar e a jamais desistir; sentava ao meu lado, me orientava e

extraia de mim as soluções nas quais ele tinha o conhecimento que eu tinha, porém que não sabia como expressá-las.

Não posso esquecer do professor Yzaack e da professora Darlene, que também foram peças fundamentais nesse processo de graduação.

Aos meus colegas de trabalho, que por muitas vezes dedicaram um pouco do seu tempo para me ajudar e tirar dúvidas no processo de construção do meu projeto.

Aos amigos (as) e familiares que estiveram junto, me apoiando e incentivando para que eu não desistisse e pudesse continuar em frente.

Agradeço ainda aos guerreiros que lutaram comigo em momentos nos quais a ajuda era mútua; sei que todos contribuíram para o progresso e bom empenho um do outro. Alguns formarão um semestre após o meu, outros ficaram pelo caminho e têm também aqueles que ainda vão demorar um pouco mais; porém, garanto que a amizade construída até aqui não vai acabar. Aos que ficam: sejam fortes e corajosos, pois o Senhor está com vocês.

Com perseverança, não se conformando com o fracasso, indo sempre em frente para superá-los, acreditando que o Senhor estará agindo, reafirmo o que diz o profeta Samuel: “Até aqui nos ajudou o Senhor e por isso estamos alegres.” 1º Samuel 7:12

RESUMO

Foi apresentada a Líder do Ministério Infantil da Segunda Igreja Batista em Palmas, uma proposta de desenvolvimento, no qual foi concebido um projeto para posterior desenvolvimento, com a aprovação do mesmo. Ainda foram, realizados estudos para aprimoramento e escolha de qual técnica, método e/ou metodologia a ser utilizada para a geração das escalas. Através do estudo de modelos de desenvolvimento de software, dos métodos ágeis e seus braços, foi criada uma estrutura para o desenvolvimento do sistema de Escala Dinâmica utilizando técnicas de inteligência artificial, para auxiliar na redução de custos e sobrecarga de serviço do ministério infantil da Segunda Igreja Batista em Palmas. Deste modo foi desenvolvido o sistema em *JSF*, utilizando o *framework Primefaces*, de modo responsivo, no qual com a utilização de técnica de algoritmo genético é possível a obtenção de escalas de trabalho de modo geral ou mesmo individualizado à cada equipe de trabalho cadastrada, além de ser utilizado o *Jasperreport* para relatórios das escalas geradas, facilitando assim a visualização das mesmas de acordo com cada data dos eventos cadastrados.

Palavras-chaves: equipe, escala, algoritmo genético, inteligência artificial.

ABSTRACT

It was presented the Leader of the Children's Ministry of the Second Baptist Church in Palmas, a development proposal, in which a project was designed for further development, with the approval of it. Studies were also carried out to improve and select which technique, method and / or methodology to be used for the generation of the scales. Through the study of software development models, agile methods and their arms, a structure was created for the development of the Dynamic Scale system using artificial intelligence techniques to help reduce cost and service overhead of the children's ministry of the Second Baptist Church in Palmas. In this way the system was developed in JSF, using the framework Primefaces, in a responsive way, in which with the use of genetic algorithm technique it is possible to obtain scales of work in a general way or even individualized to each work team registered, besides if Jasperreport is used to report the scales generated, thus facilitating their visualization according to each date of the events registered.

Keywords: team, scale, genetic algorithm, artificial intelligence.

Lista de Figuras

Figura 1: Metodologia SCRUM.....	20
Figura 2: Modelo de Incremento.....	20
Figura 3: Práticas do XP.....	22
Figura 4: Método – Família Christal.....	24
Figura 5: Modelo da Classe Usando o ManagedBean.....	26
Figura 6: Tela Responsiva.....	29
Figura 7: Possibilidades de Classificação dos Algoritmos Genéticos.....	31
Figura 8: Modelo de Implementação de um AG.....	32
Figura 9: Modelo de uma Rede Neural/Neurônio Artificial.....	34
Figura 10: Passos para implementação de PO.....	35
Figura 11: Algoritmo de Seleção Direta.....	36
Figura 13: Desenho da Metodologia Do Projeto.....	41
Figura 14: Arquitetura do sistema.....	43
Figura 15: Diagrama de Casos de Uso do Sistema.....	44
Figura 16: Tela de Gerenciamento do Líder de Equipe.....	45
Figura 17: Diagrama de Classes do sistema.....	46
Figura 18: Diagrama de Banco de dados do sistema.....	47
Figura 19: Cria a População Inicial.....	49
Figura 20: Roleta para gerar os Pais.....	49
Figura 21: Cruzamento de ponto duplo para gerar os Filhos.....	50
Figura 22: Relatório de Escala.....	54
Figura 23: Design do Relatório de escala em PDF.....	55
Figura 24: JRXML do Relatório de escala em PDF.....	55
Figura 25: Select Relatório de escala em PDF.....	56
Figura 26: Relatório de escala em PDF.....	56
Figura 27: Gráfico de Desempenho da Geração de Escalas.....	57

Lista de Tabelas

Tabela 1. Definições de Inteligência Artificial, organizadas em quadros e categorias.....	30
---	----

Lista de Quadros

<i>Quadro 1: Código que Cria a População Inicial.....</i>	<i>50</i>
<i>Quadro 2: Código que Gera os Indivíduo pais.....</i>	<i>51</i>
<i>Quadro 3: Código faz o cruzamento dos pais.....</i>	<i>52</i>
<i>Quadro 4: Código que cadastra as Escalas aos seus respectivos Eventos na Base de Dados.....</i>	<i>52</i>
<i>Quadro 5: Código que Troca os membros de Escala.....</i>	<i>53</i>

Sumário

1 INTRODUÇÃO	15
2 REVISÃO DE LITERATURA	16
2.1 DESENVOLVIMENTO DE SOFTWARE	16
2.1.1 MÉTODOS ÁGEIS	17
2.1.1.1 SCRUM	19
2.1.1.2 XP	21
2.1.1.3 FAMÍLIA CRYSTAL	23
2.2 DESENVOLVIMENTO WEB	24
2.2.1 JSF/MAPEAMENTO OBJETO-RELACIONAL	25
2.2.1.1 INTERAÇÃO HUMANO COMPUTADOR (HCI)	27
2.2.1.1.1 UTILIZAÇÃO DE TECNOLOGIAS PARA HCI	28
2.2.1.2 RESPONSABILIDADE	28
2.3 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL	29
2.3.1 ALGORITMOS GENÉTICOS	31
2.3.2 REDES NEURAIS	33
2.4 PESQUISA OPERACIONAL	34
2.4.1 TEORIA DOS JOGOS	36
2.4.2 ALGORITMOS DE SELEÇÃO/RANDON	36
3 METODOLOGIA	38
4 RESULTADOS E DISCUSSÕES	43
5 CONCLUSÃO	58
6 REFERÊNCIAS	60
7 APÊNDICES	65
7.1 APÊNDICE 1	65
7.2 APÊNDICE 2	67
7.2.1 ATIVIDADES	67
7.2.2 SEQUÊNCIA	67
7.3 APÊNDICE 3	68
7.4 APÊNDICE 4	69
7.4.1 DETALHAMENTO DO CASO DE USO	69
7.4.1.1 CASO DE USO: AUTENTICAÇÃO	69
7.4.1.2 CASO DE USO: GERENCIAMENTO DE MEMBROS	70
7.4.1.3 CASO DE USO: GERENCIAR ESCALA	71
7.4.1.4 CASO DE USO: GERENCIAR EQUIPE	72
7.4.1.5 CASO DE USO: GERENCIAR CALENDÁRIO	73
7.5 APÊNDICE 5	74
7.5.1 Visão de Negócio	74
7.5.1.1 Objetivo	74
7.5.1.2 Situação atual	74
7.5.1.2.1 Documentação de levantamentos	74
7.5.1.3 Situação proposta	75
7.5.2 Requisitos	75
7.5.2.1 Requisitos Funcionais	75
7.5.2.2 Requisitos Não Funcionais	75
7.5.3 Diagrama de Casos de Uso	75
7.5.3.1 Detalhamento do caso de Uso	76
7.5.4 Diagrama de Atividades	76

7.5.5 Diagrama de Classes.....	77
7.5.6 Diagrama de Sequência.....	77
7.5.7 Projeto de Banco de Dados.....	78

1 INTRODUÇÃO

Com o passar dos anos a tecnologia apresenta evoluções gradativas, à medida que a cada dia a humanidade vem tornando-se mais dependente de soluções, no qual os usuários buscam agilizar suas atividades diárias com o acesso à internet, troca de mensagens via *e-mail* e redes sociais. Mas, além destas atividades, a tecnologia é utilizada ainda proporciona a perspectiva de criação de sistemas, nos quais podem auxiliar empresas no controle de suas atividades diárias.

Tendo em vista tantas aplicações à tecnologia, diariamente, observou-se no caso aplicado, que, o ministério/departamento infantil da Segunda Igreja Batista de Palmas - TO (SIBAPA) possui um número elevado de voluntários e utilizava planilhas eletrônicas para realização de suas escalas.

Diante deste fato, este trabalho propôs-se a finalidade quanto ao desenvolvimento de um sistema de escalas com o intuito de auxiliar na organização das equipes do Ministério Infantil da Segunda Igreja Batista de Palmas - TO (SIBAPA), visando reduzir a sobrecarga de voluntários com a organização e reorganização de escalas de trabalho, de forma a satisfazer toda a equipe e contribuir com a congregação. Com a criação deste sistema o ministério infantil pode usufruir de um sistema de escalas automatizado, deixando as planilhas eletrônicas, além de economizar algo em torno de \$100,00(cem dólares) que seriam necessários para aquisição de licenças de softwares de gerenciamento anualmente.

Neste contexto o sistema conta com cadastro de equipes, na qual informações sobre disponibilidade de horário devem ser passado por todos os voluntários e líderes de suas respectivas equipes. Os membros no caso, são consultados via interação com um banco de dados já existente na comunidade, além das consultas de escala que serão feitas por todos os membros. As escalas serão geradas de forma aleatória após a inclusão dos dados necessários, como a conclusão dos cadastros e informação da disponibilidade de todos os membros de cada equipe, utilizando-se de técnicas de inteligência artificial, para isso foram aplicados algoritmos genéticos.

Com isto, o tema escolhido com o propósito de contribuir com as atividades do da comunidade cristã, mas de modo mais amplo apresentar e acrescentar o aprendizado no desenvolvimento de *software* utilizando conceito e metodologias de relevantes

envolvendo o estudo de modelos Ágeis da Engenharia de *Software*. Ainda, realiza-se considerações quanto ao desenvolvimento *WEB*, bem como a utilização de algoritmo genético na construção de um sistema que automatizado para a geração de escalas.

Assim, inicialmente, o trabalho leva em consideração a revisão de literatura, no qual são considerados os pontos relevantes para absorção do conhecimento necessário para o desenvolvimento do presente trabalho. Posteriormente é apresentada metodologia, materiais e métodos utilizados no desenvolvimento do projeto proposto. Por fim, são levantados nos resultados os principais pontos trabalhados e de maior interesse na discussão, bem como as conclusões que puderam ser obtidas do trabalho e suas perspectivas.

2 REVISÃO DE LITERATURA

Este tópico tem por finalidade mostrar aspectos relevantes, considerados por autores usados como base para o desenvolvimento de *software* e tão logo suporte a este trabalho. Assim, serão apresentados os principais pontos inerente às Engenharia de *Software* e Desenvolvimento *web*, como é destacado a seguir.

2.1 DESENVOLVIMENTO DE SOFTWARE

De acordo Sommerville (2007) um processo de *software* é um conjunto de atividades que levará a produção de um produto de *software*. Estas atividades podem envolver o desenvolvimento propriamente dito, utilizando uma linguagem de programação como *Java*, *C* ou *C++*, por exemplo, o desenvolvimento de um novo *software* pode ser feito baseando-se no uso de ampliações e/ou modificações de sistemas já existentes e de configuração e interação com softwares comerciais ou componentes dos Sistemas. Cita ainda que o *software* é um conjunto de programas separados, arquivos de configuração, documentação e sites.

Assim, Sommerville (2007) ainda mencionar que:

Um modelo de processo de *software* é uma representação abstrata de um processo de *software*. Cada modelo de processo representa um processo sob determinada perspectiva e, dessa forma, fornece somente informações parciais sobre esse processo (SOMMERVILLE, 2007).

Os modelos de desenvolvimento podem ser divididos em três tipos principais, no

qual se verificam o desenvolvimento em Cascata, Iterativo e Incremental, dentro de uma perspectiva clássica (MAGALHÃES, 2012). Sommerville (2007), leva em consideração alguns modelos de processo de desenvolvimento de Software:

- **Modelo cascata** - que considera as atividades fundamentais e as representa separadamente como fases do processo;
- **Desenvolvimento evolucionário** - realiza uma abordagem que intercala as atividades de desenvolvimento, especificação e validação do projeto de software;
- **Engenharia de software baseada em componentes** - tem o processo de desenvolvimento focado na interação de componentes baseando-se em um número significativo de componentes reutilizáveis ao invés de desenvolvê-los a partir do zero.

SOMMERVILLE (2007) relata que:

Estes três modelos genéricos de processos são amplamente usados na engenharia de *software*. Eles não são mutuamente exclusivos e frequentemente usados em conjunto, especialmente para o desenvolvimento de sistemas de grande porte. (..) os subsistemas contidos em um sistema maior podem ser envolvidos usando diferentes abordagens.

O processo de desenvolvimento de *software* são atividades realizadas com a finalidade de construir e/ou acrescentar novas funcionalidades a um produto de *Software* podendo utilizar um dos modelos genéricos para o seu desenvolvimento e/ou aprimoramento do *software*, trazendo confiabilidade, qualidade, segurança aos usuários do software. Isso, considerando que tais modelos são um norte para definir uma linha de raciocínio para o desenvolvimento do *software*.

Entretanto, um ponto relevante a se verificar no desenvolvimento de *software* foi o aparecimento dos métodos ágeis, como forma alternativa aos processos clássicos na engenharia de *software*, sendo que a seguir são feitas considerações sobre o tema.

2.1.1 MÉTODOS ÁGEIS

Para BERNARDO (2015) os métodos ágeis nasceram nos braços do desenvolvimento de software sendo uma alternativa à gestão tradicional de projetos, mas hoje podem ser aplicados a qualquer tipo de projeto (inclusive os que não se remetem ao software).

SOMMERVILLE (2007) menciona que:

Nas décadas de 1980 e início de 1990, havia uma visão geral de que a melhor maneira de obter o melhor software era por meio de um cuidadoso planejamento de projeto, garantia de qualidade formalizada, uso de métodos de análise e projeto apropriado por ferramentas CASE e controlados por um rigoroso processo de desenvolvimento de software.

De acordo com Sommerville (2007) a abordagem dos métodos ágeis surge como alternativa para a comunidade de engenharia de *software*, que estava preocupada com sistemas de *software* grande e de longa vida, que geralmente eram desenvolvidos por um grande número de desenvolvedores.

Sommerville (2007) cita também que devido a isto, esta abordagem de desenvolvimento baseada em planos foi aplicada em sistemas de pequenas e médias empresas, em que o tempo gasto no planejamento, era maior que o utilizado no desenvolvimento do mesmo e em testes.

Deste modo, nota-se que o tipo de metodologia de trabalho a ser adotado para seu planejamento, está intrinsecamente ligado ao seu tamanho, no qual os modelos clássicos de desenvolvimento de *software* demonstram ser mais relevantes para o desenvolvimento de sistemas grandes, porém para casos de sistemas pequenos, podem comprometer muito tempo no planejamento.

Com isto Sommerville (2007) informa que a insatisfação com este tipo de abordagem levou alguns desenvolvedores da década de 1990 a propor novos métodos de desenvolvimento de software, mais ágeis, que permitiam que os desenvolvedores se preocupassem mais com o desenvolvimento do *software* em vez de seu projeto e documentação, contando geralmente com abordagens interativas para especificar e entregar o *software*. Deste modo, foram criados principalmente para ajudar no desenvolvimento de software, em que os requisitos mudam rapidamente durante o processo de desenvolvimento.

Bernardo (2015) menciona que em fevereiro de 2001, em Utah, marcava o surgimento e propagação do paradigma dos métodos ágeis. Essa reunião desencadeou o que conhecemos hoje como manifesto ágil. Descreve ainda que no manifesto ágil surgiram princípios e valores onde cada método ágil existente carrega consigo os valores e princípios arraigados no manifesto ágil, que são apresentados abaixo:

- **Indivíduos e interação entre eles** mais que processos e ferramentas;
- **Software em funcionamento** mais que documentação abrangente;
- **Colaboração com o cliente** mais que negociação de contratos;
- **Responder a mudanças** mais que seguir um plano.

É mais importante que haja uma interação entre os desenvolvedores do que ficar muito tempo apegado a processo e/ou escolhas de ferramentas; no qual o funcionamento do software é o principal objetivo pois traz mais proximidade do cliente e estimula-o a colaborar com maiores informações do sistema desejado tendo por finalidade responder às solicitações do cliente ao invés de ficar muito tempo com negociações e/ou seguindo um determinado tipo de plano para o projeto.

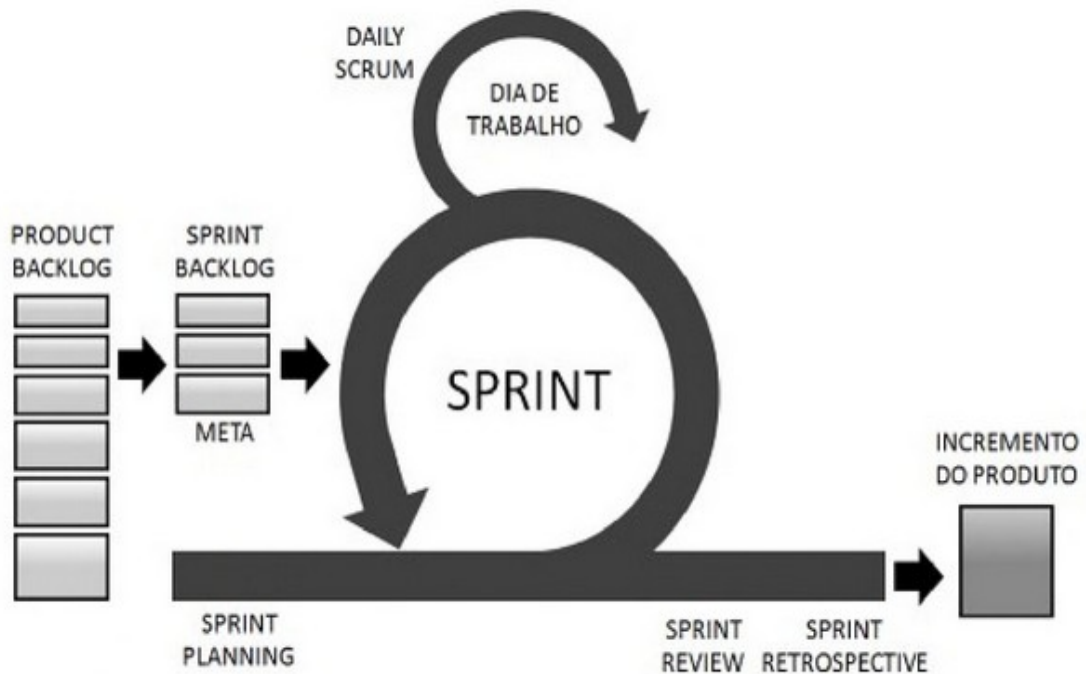
Com isto, percebe-se que as metodologias ágeis tentam satisfazer o cliente em detrimento suas necessidades de funcionamento. Neste caso, é mais importante um diálogo com o cliente que a documentação, pois, a documentação é menos importante, e que o software funcionando tem mais valor que a documentação completa e detalhada, e que a adaptação a mudanças tem mais importância que seguir um plano inicial e dentro dos Métodos Ágeis podem ser encontrados outros modelos de desenvolvimento, um deles é o SCRUM utilizado pela gestão de projetos de softwares seguindo um script para sequência do desenvolvimento. Sendo que estaremos apresentando mais sobre o tema na sequência.

2.1.1.1 SCRUM

Segundo a Comunidade de Desenvolvimento Ágil do Brasil (2013/2014) o Scrum é uma metodologia ágil para gestão e planejamento de projetos de software, no qual os projetos são divididos em ciclos conhecidos como Script.

Para Schimiguel (2014) o Scrum é usada para estabelecer conjuntos de regras e práticas de gestão para conseguir o sucesso de um projeto. Com o foco no trabalho em equipe, ocorre uma melhora na comunicação e maximiza o apoio de todos, fazendo com que todos do time se esforcem e se sintam bem com que estão fazendo e isso acaba gerando mais para frente um aumento de produtividade.

Figura 1: Metodologia SCRUM



Fonte: Scrum: Gestão ágil para projetos de sucesso (2013/2014)

Com base na **Figura 1** percebe-se que o Scrum segue um cronograma de reuniões e planejamentos com duração de 8 horas, encontros diários de 15 minutos com todos os participantes em pé, além de utilizar uma equipe de 3 a 9 pessoas para realização do processo, no qual é feito o incremento como na **Figura 2**.

Figura 2: Modelo de Incremento.



Fonte: Autor, 2017.

Bernardo (2015) diz que, muitas pessoas começam a conhecer os métodos ágeis via Scrum, e acabam se confundindo e associando de forma errônea como se o Scrum

fosse um método ágil, sendo que é apenas um subconjunto dos métodos ágeis assim como o Kanban, XP, ASD ou FDD.

Com base nas citações de Bernardo (2015) e Schimiguel (2014) verificar-se que independente da forma como é conhecido, se como um método ágil ou um subconjunto do mesmo, o Scrum é uma das metodologias de aplicação mais usada e conhecida entre os desenvolvedores.

Bernardo (2015) afirma que:

O fundador do Scrum (Schwaber e Sutherland, 1995) o descreveu como uma estrutura de processos que tem sido usada para gerenciar o desenvolvimento de produtos complexos desde o início da década de 1990.

Bernardo (2015) cita que o Scrum não é um processo ou técnica para construção de produtos, mas pelo contrário, é um framework que pode ser utilizado para a entrega de diversos processos e técnicas. Além de frequentemente ser usado para gerenciar softwares e o desenvolvimento de produtos complexos, através da utilização de técnicas iterativas e incrementais.

Ainda nos métodos ágeis destaca-se o XP, um modelo de desenvolvimento de software que envolve o cliente durante o processo de desenvolvimento, falaremos mais sobre o tema a seguir.

2.1.1.2 XP

Para Sommerville (2007) o XP é o mais conhecido e usado dos métodos ágeis, recebendo o nome por Beck pelo fato de sua abordagem ter sido desenvolvida pelo avanço de reconhecida boa prática, como o do desenvolvimento interativo e o envolvimento do cliente em níveis externos, além de o fato de seus requisitos serem expressos em forma de cenários implementados como uma série de tarefas.

Segundo Schimiguel (2014):

O XP foi criado por Kent Beck quando trabalhava na Chrysler Comprehensive Sistema de Compensação (C3) no projeto da folha de pagamento. Quando Beck tornou-se C3 líder do projeto em março de 1996, ele começou a refinar o método desenvolvimento usado no projeto e logo em seguida ele escreveu um livro. E em outubro de 1999, a Extreme Programming foi publicada. Em fevereiro de 2000, a Daimler-Benz adquiriu a Chrysler e acabou cancelando o projeto C3.

Segundo a Comunidade de Desenvolvimento Ágil do Brasil o Extreme Programming (XP) (2013/2014) é uma metodologia de desenvolvimento de software que nasceu nos Estados Unidos no final da década de 90. Para criar sistemas de melhor qualidade que são produzidos em menos tempo e de forma mais econômica que o habitual.

Ainda, Sommerville (2007) fala que, quando o XP está em processo de desenvolvimento os clientes são envolvidos na especificação e padronização dos requisitos do sistema, onde os mesmos não são especificados como lista de funções necessárias do sistema. Em vez disso, o cliente é parte da equipe de desenvolvimento e ambos discutem juntos os cenários e desenvolvem um cartão de histórias. E depois deste cartão pronto a equipe de desenvolvimento os dividirá em tarefas e estimará o esforço e os recursos necessários para a implementação.

Com isto o XP é um método de aplicação onde o cliente interage com os envolvidos na aplicação facilitando e reduzindo conflitos de interpretação em ambos os lados do desenvolvimento, como mostra a **Figura 3**.

Figura 3: Práticas do XP



Fonte: DevMedia- Métodos Ágeis – Parte 02, Adaptado

O XP tem como principal característica lidar com equipes pequenas propondo a programação em pares, porém não obrigatória. Como Apresentada na **Figura 3** os desenvolvedores realizam o planejamento, os testes do software junto com o cliente, estando em reuniões semanais com os mesmos, fazendo o compartilhamento dos códigos, liberando pequenas versões funcionais para o cliente realizando um ciclo de vida desenvolvedor → cliente → desenvolvedor → cliente.

Por fim, Sommerville (2007) também diz que, a XP exige uma abordagem externa para o desenvolvimento iterativo, onde novas versões de software podem ser geradas várias vezes por dia e seus incrementos entregues aos clientes em aproximadamente a cada duas semanas.

Diz ainda que quando um programador executa um sistema para criar uma nova versão, ele deve executar todos os testes automatizados existentes, bem como para os da nova funcionalidade, pois a mesma só será aceita se todos os testes tiverem sucesso.

Também pode ser observado no tema a seguir o desenvolvimento do software é visto de acordo com sua complexidade e identificado pela cor.

2.1.1.3 FAMÍLIA CRYSTAL

Segundo a Equipe DevMedia (2017) a Família Crystal tem seus membros identificados por cores que indicam a intensidade do método e quanto mais escura maior a complexidade do projeto.

Algumas características comuns à família Crystal, tais como o desenvolvimento incremental com ciclos de no máximo quatro meses, ênfase maior na comunicação e cooperação das pessoas, não limitação de quaisquer práticas de desenvolvimento, ferramentas ou produtos de trabalho, bem como a incorporação de objetivos para reduzir produtos de trabalho intermediários e desenvolvê-los como projetos evoluídos.

De acordo com o mencionado, os graus de complexidade dos membros da Família Chrystal são identificados por cores assim como os cristais e é representado de acordo com a **Figura 4**.

Figura 4: Método – Família Christal

	Clear	Yellow	Orange	Red	Maroon
Life (L)	L6	L20	L40	L80	L200
Essential Money (E)	E6	E20	E40	E80	E200
Discretionary Money (D)	D6	D20	D40	D80	D200
Comfort (C)	C6	C20	C40	C80	C200
	1-6	7-20	21-40	41-80	81-200

Fonte: Mishra- Mix and Match: Agile Flavours

Para Leandro (2016):

Crystal é uma família de metodologias de desenvolvimento de software e, como os cristais, possui diferentes cores e rigidez, referindo-se ao tamanho e ao nível crítico do projeto.

Ainda, Leandro (2016) diz que os métodos Crystal são focados nos talentos e nas habilidades das pessoas, permitindo que o processo de desenvolvimento seja moldado conforme as características específicas da equipe, mesclando a sua cultura de trabalho com a proposta de desenvolvimento ágil. Além disso, utiliza dois parâmetros para se adequar ao projeto de software, sendo a primeira o número de pessoas e a segunda a métrica no nível crítico.

2.2 DESENVOLVIMENTO WEB

Para Goulart (2013) o Desenvolvimento Web consiste no fato de compreender todos os aspectos do desenvolvimento de um site ou aplicação web, no qual constam desde se planejamento até seu desenvolvimento em uma linguagem apropriada.

Miletto e Bertagnolli (2014) mencionam que ao se tratar de uma aplicação web, deve-se levar em consideração aspectos, nos quais incluem o reconhecimento dos requisitos exigidos pela aplicação como:

- ❖ Quais tecnologias podem ser utilizadas;
- ❖ Como é realizado o processo de desenvolvimento do sistema;
- ❖ Quais os perfis do usuário e suas tarefas na interface gráfica da aplicação;
- ❖ Como armazenar os dados.

Logo, pode ser observado que no desenvolvimento web, existe a composição de várias etapas que vão de um planejamento do site, o desenvolvimento do protótipo, planejamento das páginas, o conteúdo das mesmas com foco na página principal, interação das ferramentas, planejamento e desenvolvimento do design do site, o desenvolvimento do código, os testes e o lançamento do site e/ou aplicativo web.

Cordeiro (2012) cita que:

Com a popularização das aplicações web, muitas plataformas ganharam popularidade, como foi o caso do PHP e do ASP, e não demorou para que o Java também suportasse de alguma maneira o desenvolvimento web. A infraestrutura para desenvolvimento Web em Java é baseada em Servlets. Hoje chamamos de infraestrutura, mas, nos primórdios, elas eram usadas diretamente para desenvolver aplicações. Uma Servlet é uma classe Java que fica no servidor e é invocada com base em algum padrão de URL.

Pode ser observado que no desenvolvimento web é comum o uso objeto relacional e o mapeamento do projeto além do uso do JSF que são assuntos do próximo tópico.

2.2.1 JSF/MAPEAMENTO OBJETO-RELACIONAL

De acordo com Cordeiro (2012), o que causou maior estranheza no primeiro contato com o JSF foi o fato de ele ser um framework baseado em componentes e não em ações, como eram os principais frameworks da época suas telas seriam escritas em um arquivo XHTML utilizando Tags do próprio JSF.

Para Coelho (2013) o JSF é um framework que tem um comportamento "*component-based*". Um framework component-based tem por característica principal que a página irá buscar a informação no ManagedBean que tem um tipo de escopo, ideal para cada situação. É possível utilizar um ManagedBean por CDI que utiliza o pacote javax.inject ou por ManagedBean encontrado no pacote javax.faces.bean. Existe diferença entre cada contexto que for utilizado e ambos com vantagens e desvantagens. Portanto, é importante conhecer os escopos para conseguir usá-los corretamente.

Ainda, segundo Cordeiro (2012):

Quando desenvolvemos um projeto usando o JSF, a construção da tela se torna um pouco diferente de outros frameworks, como o Struts, VRaptor e Spring MVC, nos quais costumamos usar apenas elementos HTML para criar os formulários, tabelas e demais características da visualização das informações. No JSF, praticamente

toda a tela é criada a partir de Taglibs próprias dele, que possuem o papel de renderizar o HTML adequado para o funcionamento do framework.

Na **Figura 5** é apresentado o modelo de uma classe onde são chamados alguns importes de mapeamento do JSF no qual um destes é o @ManagedBean.

Figura 5: Modelo da Classe Usando o ManagedBean.

```
// imports omitidos
@ManagedBean
@SessionScoped
public class UsuarioLoginMB {

    private Usuario usuario;
    private String localeUsuario = "pt_BR";

    public void logout() throws ServletException, IOException {
        HttpSession session = // busca a session

        session.invalidate();

        // efetua redirect
    }

    public boolean isAdministrador(){
        return usuario.getPerfil().equals(Perfil.ADMINISTRADOR);
    }

    // gets e sets omitidos
}
```

Fonte: Coelho (2013).

De acordo com Bernardi (2008) os objetos formam unidades que encapsulam atributos e operações. Os bancos de dados relacionais representam de forma bastante eficiente os atributos, mas são limitados na representação das operações. Para realizar o mapeamento de um objeto em uma tabela relacional, os atributos desse objeto devem ser representados através de colunas na tabela. Porém, isso é uma regra que não pode ser seguida ao pé da letra, pelo fato de existirem outras considerações a serem feitas como (tipos dos dados, tamanho dos campos, entre outras). Estas considerações podem fazer com que um atributo seja mapeado em várias colunas, ou então que vários atributos

sejam mapeados em apenas uma coluna.

Coelho (2012) cita que:

Para evitar toda essa verbosidade do JDBC, surgiu a JPA, uma ferramenta muito poderosa que pode aumentar consideravelmente o tempo de desenvolvimento da equipe, além de facilitar muito a implementação do código que manipula o banco de dados. Quando corretamente aplicado, a JPA se torna uma ferramenta que ajuda em todas as funções de um *CRUD* (*Create, Read, Update and Delete*), além de contribuir com diversas otimizações de performance e consistência de dados. É nesse exato momento que podemos entender por que as siglas *ORM* são tão faladas. **Object Relational Mapping (ORM)** quer dizer basicamente: “transformar as informações de um banco de dados, que estão no modelo relacional para classes Java, no paradigma Orientado a Objetos de um modo fácil”.

Com base no estudo abordado em Cordeiro (2012), vê-se que deverá ter uma forma de indicar que os objetos de uma classe *Xy* terão registros inseridos em uma tabela de mesmo nome *xy* e a este tipo de definição chama-se mapeamento, pelo fato de ser feita uma ligação entre modelos diferentes, o relacional e o orientado a objetos sendo conhecido como Mapeamento Objeto Relacional.

2.2.1.1 INTERAÇÃO HUMANO COMPUTADOR (HCI)

A ergonomia de interface com o usuário e a usabilidade que eles proporcionam, são diferenciais que cada vez mais atraem o interesse do setor público e empresarial. Com o objetivo de trazer conteúdo útil para o desenvolvimento de interfaces simples, intuitivas e fáceis de usar (Cybis, Betiol e Faust, 2010).

Interação Humano Computador (HCI ou IHC), é a comunicação que existe entre o homem e o computador através de uma interface interativa que em determinados momentos o usuário responde a uma caixa de diálogo que aparece na tela para realização de determinadas atividades como, por exemplo, ao realizar a instalação de alguma ferramenta e/ou aplicativo, o sistema operacional envia uma mensagem perguntando se o solicitante realmente deseja instalar o app ou se for deletar um arquivo aparecerá a mensagem perguntando se realmente deseja apagar o documento.

Para Freitas (2015.1), a HCI é o modo como os seres humanos relacionam com produtos ligados a computação e as profissões que decorrem da IHC são: Designers de interação, Arquitetos de Informação, Web Designers, Engenheiros de Usabilidade.

Morais e Loper (2014) fala que:

Em todo momento o homem está em contato com diversos tipos de interfaces, como exemplo: liquidificador, ventilador, controle remoto, fechaduras, torneiras, televisores, entre muitas outras. Todos os objetos citados têm algo em comum, suas características são sempre aperfeiçoadas ou remodeladas, pode ser por uma causa simplesmente comercial para aumentar as vendas, ou, em um processo de melhoria de adaptação ao ser humano. Para o campo da computação existe o aspecto comercial, mas também, a importância da adaptabilidade está ligada diretamente à produtividade, e isso motiva significativamente o estudo das interfaces.

De acordo com os autores citados acima, a IHC é utilizada para que o homem tenha uma interação com o computador respondendo as caixas de diálogo que o sistema lança na tela e também com a utilização dos menus do próprio software, dando sequência à IHC o próximo tema fala sobre algumas tecnologias para utilizá-la.

2.2.1.1.1 UTILIZAÇÃO DE TECNOLOGIAS PARA HCI

De acordo com estudo realizado no livro de Ergonomia e Usabilidade dos autores Walter Cybis, Adriana Holtz Betiol e Richard Faust (2010) as tecnologias de utilização do HCI são diversas como, por exemplo, os computadores de mão que são considerados “computadores portáteis”, “computadores sem fio”, “computadores de bolso”, “dispositivos móveis”, dentre outros. Isso ainda permite outras características como a mobilidade ao usuário, conexão sem fio, além de apresentar informações como entradas de dados, a TV digital o que traz a possibilidade de transmissão de múltiplas informações em um mesmo canal. O látex, Word, Google play, Excel, são ferramentas com tecnologias e aplicações que tem essa interação do homem com o computador.

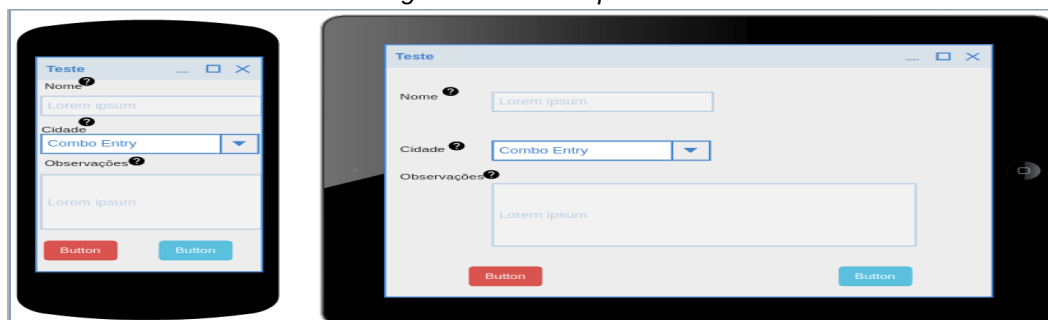
2.2.1.1.2 RESPONSABILIDADE

De acordo com Lopes (2013) o design responsivo nada mais é que relembrar que a Web é uma mídia flexível e adaptável. Logo, são páginas que se adaptam a todo tipo de dispositivo, onde se pode abrir a mesma página no celular, no tablet e no Desktop, e ela se adapta para melhorar a experiência do usuário. Assim, a chave pro design responsivo é fazer um design flexível e adaptável, que se ajuste às características do navegador, do dispositivo e do contexto do usuário. Também fala que o Web Design Responsivo é a

chave para essa nova Web (Zemel, 2012).

É pensar em páginas que se adaptem a todo tipo de dispositivo e contexto de uso. É sair das limitações de um browser Desktop e seu tamanho previsível, e pensar em páginas com flexibilidade que suportem todo tamanho de tela, qualquer tipo de resolução, interfaces com touch ou mouse como na **Figura 6** apresentada a seguir.

Figura 6: Tela Responsiva



Fonte: Autor, 2018.

Pode ser observado na figura acima que independente do dispositivo que foi utilizado o site não perdeu seus dados, mas foi adaptado de forma que o usuário pudesse utilizá-lo sem que fosse alterado parte da página facilitando o manuseio do usuário.

Entretanto, um outro assunto bem relevante e de muita importância é a utilização dos recursos do próprio computador para prever e solucionar possíveis problemas tanto computacionais quanto da humanidade, este assunto será abordado a seguir.

2.3 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

Começou após a segunda guerra mundial e seu próprio nome foi dado em 1956. Abrange uma enorme variedade de subcampos, como: aprendizado e percepção e até mesmo tarefas como jogo de xadrez, demonstrações de teoremas matemáticos, criação de poesias e diagnósticos de doenças. Sistematiza e automatiza atividades intelectuais, têm um potencial relevante para qualquer tipo de tarefa intelectual humana (Russell e Norvig, 2004).

Tabela 1. Definições de Inteligência Artificial, organizadas em quadros e categorias.

Sistemas que pensam como seres humanos	Sistemas que pensam racionalmente
"O novo e Interessante esforço para fazer os computadores pensares... máquinas com mentes, no sentido total e literal."(Haugeland,1985)	"O estudo das faculdades mentais pelo uso de modelos computacionais."(Charniak e McDemott, 1985)
"Automatização de atividades que associamos ao pensamento Humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado..."(Bellman, 1978)	"O estudo das computações que tornam possível perceber, raciocinar e agir."(Winston,1992)
Sistemas que atuam como seres humanos	Sistemas que atuam racionalmente
"A parte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas."(Kurzweil, 1990)	"A Inteligencia Computacional é o estudo do projeto de agentes inteligentes."(Poole et al., 1998)

Fonte: Russell e Norvig, 2004.

Para Lustosa e Alvarenga (2004) a Inteligência Artificial (IA), tenta entender o comportamento de entidades inteligentes. Ao contrário da filosofia e da psicologia, que estão mais preocupadas com o estudo da inteligência dentro de um contexto de relações humanas, a IA é mais focada em como essas entidades podem ser criadas e utilizadas para determinados fins.

Machado (2006) menciona que:

A investigação na área da Inteligência Artificial (IA) tem-se centrado no estudo e modelação de características usualmente associadas ao lado esquerdo do cérebro humano: lógica, cálculo, pensamento racional, etc. No entanto, a procura de IA com propriedades semelhantes às humanas estará necessariamente incompleta, a menos que tenha em consideração o facto de a cognição incluir características tipicamente associadas ao lado direito do cérebro, tais como: criatividade, juízo estético, capacidade artística, musical, etc. Este "lado" da inteligência humana é menos susceptível de formalização e, conseqüentemente, mais difícil de explorar, modelar e testar. Contudo, é amplamente reconhecido que estes aspectos têm importância para o comportamento humano.

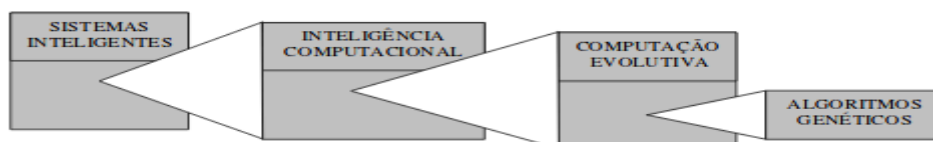
Contudo a IA tem como objetivo, simular no computador pequenos comportamentos, como um pinga de solda, por exemplo. De modo geral a Inteligência Artificial é uma máquina capaz de criar qualquer coisa, e pode ser implementada de acordo com a necessidade do desenvolvedor, pois temos várias técnicas para efetivá-la, e uma delas são os Algoritmos Genéticos que serão objetivo da leitura a seguir.

2.3.1 ALGORITMOS GENÉTICOS

Algoritmos genéticos (GA) é uma técnica de busca extremamente eficiente no seu objetivo de varrer o espaço de soluções e encontrar soluções próximas da ótima, quase sem precisar da interface humana. O problema dos GA é que eles não são tão bons em termos de tempo de processamento, com tudo são mais adequados em problemas especialmente difíceis (Linden, 2008).

Vargas (2000) menciona que os algoritmos genéticos fazem parte dos sistemas computacionais inspirados em modelos de processos naturais, mais especificamente da computação evolutiva como apresentada na **Figura 7**.

Figura 7: Possibilidades de Classificação dos Algoritmos Genéticos



Fonte: Vargas(2000)

Algoritmo Genético é uma técnica de Inteligência Artificial, criada com o intuito de imitar determinados processos observados na evolução natural das espécies. Tendo como objetivo solucionar problemas do mundo real de forma otimizada, sendo aplicável a diversas áreas (Rosa e Luz, 2009).

A técnica de busca está inspirada no processo de evolução dos seres vivos, com métodos generalizados de busca e otimização que simulam os processos naturais de evolução. Mais especificamente Algoritmo Genético é um procedimento iterativo que mantém uma população de estruturas (chamadas indivíduos, cromossomos ou strings), que representam possíveis soluções de um determinado problema. Durante o processo evolutivo, em cada geração, os indivíduos da população atual são avaliados de acordo com o valor de sua aptidão para a solução do problema (Mercado, 2001).

Os algoritmos genéticos são usados para resolver problemas de otimização e aprendizado de máquina nos quais simulam processos naturais de sobrevivência e reprodução das populações, essenciais em sua evolução; tendo como ideia básica de funcionamento dos algoritmos genéticos a de tratar possíveis soluções do problema como

"indivíduos" de uma "população", que irá "evoluir" a cada iteração ou "geração".

O algoritmo genético encontra-se na computação evolutiva, pois utiliza técnicas de busca baseadas na teoria da evolução, nos quais as variáveis são representadas como genes e cromossomos e pode ser implementada com um simples uso de strings, de bits ou caracteres para representar os cromossomos de acordo com a **Figura 8**.

Figura 8: Modelo de Implementação de um AG



Fonte : do Autor, 2018.

Russell e Norvig (2004) falam que o Algoritmo Genético é uma variante de busca em feixe estocástico, na qual os estados sucessores são gerados pela combinação de dois estados pai, em vez de serem gerados pela modificação de um único estado.

Os Algoritmos Genéticos (AG) possuem seus critérios de desenvolvimento e implementação, pois, para seu início deve-se identificar a função objetivo que geralmente é a indicação da solução do problema proposto, em seguida a forma de como os dados serão representados (se de forma binária (0,1), ou de forma real com inteiros), para realizar a seleção que pode ser feita com a roleta por exemplo que gera os pais de acordo com o valor de aptidão de cada cromossomo selecionado ou o torneio que é a escolha dos cromossomos da população atual de forma aleatória, para depois realizar o cruzamento que pode ser feito de ponto(os) único(é escolhido um ponto aleatório e a partir daí é feita a troca do material genético dos pais gerando dois novos cromossomos), duplo(é escolhido dois pontos de cruzamento partir daí é feita a troca incrementada do material genético dos pais gerando dois novos cromossomos) ou aleatórios (é o cruzamento de dois pontos aleatórios utilizando como base uma máscara que contém cromossomos binários onde pega-se o valor 1 do material genético de um dos pais e 0 do material do outro).

Assim como temos os Algoritmos Genético como técnica da IA, uma outra técnica de igual importância são as Redes Neurais que serão apresentadas no próximo tópico.

2.3.2 REDES NEURAIS

As Redes Neurais têm seu estudo motivado pelo reconhecimento do cérebro humano, realizando o processamento de informações de forma totalmente diferente que um computador convencional, no qual o cérebro é complexo, não-linear e paralelo. Possui a capacidade de organizar seus componentes estruturais conhecidos por neurônios, de forma a realizar seus processamentos muito mais rapidamente que o mais rápido computador digital existente hoje (Haykin, 2001).

Para Tafner(1998) as redes neurais artificiais representam um paradigma metodológico no campo da inteligência artificial, ou seja, no desenvolvimento de sistemas computacionais capazes de imitar tarefas intelectuais complexas, tais como resolução de problemas, o reconhecimento e classificação de padrões, processos indutivos e dedutivos. E assim como no cérebro, as redes neurais são organizadas de acordo com um número de elementos individuais, que se interconectam uns aos outros formando redes com a capacidade de armazenar e transmitir informações.

Para Barreto (2002) pode-se dizer que:

Redes neurais artificiais consistem em um modo de abordar a solução de problemas de inteligência artificial. Neste caso, em lugar de tentar programar um computador digital de modo a fazê-lo imitar um comportamento inteligente (saber jogar xadrez, compreender e manter um diálogo, traduzir línguas estrangeiras, resolver problemas de matemática tais como se encontram nos primeiros anos dos cursos de engenharia, etc.) procura-se construir um computador que tenha circuitos modelando os circuitos cerebrais e espera-se ver um comportamento inteligente emergindo, aprendendo novas tarefas, errando, fazendo generalizações e descobertas, e frequentemente ultrapassando seu professor. Da mesma forma, estes circuitos neurais artificiais poderão se auto organizar, quando apresentados a ambientes diversos, criando suas próprias representações internas e apresentar comportamentos imprevisíveis.

Através das afirmações apresentadas percebe-se que o cérebro é considerado um computador humano e busca-se uma inteligência de forma a torna seu processamento igual ou até melhor que o cérebro humano. Com isso desenvolve-se o estudo de estruturas cerebrais e algoritmos de treinamento aperfeiçoados.

Pensou-se no uso das redes neurais artificiais para realização do comparativo das escalas que ficarão armazenadas no banco de dados para dentre elas verificar qual foi a melhor escala de cada equipe.

Com base no que foi apresentado anteriormente sobre Redes Neurais elas são como um neurônio artificial que possui entradas, análise das entradas e saída como apresentada na **Figura 9** que é uma adaptação do modelo apresentado por Haykin(2001).

Figura 9: Modelo de uma Rede Neural/Neurônio Artificial



Fonte: Haykin(2001), Adaptado

Na figura apresentada as entradas são associadas a um peso W no qual é realizado um somatório de todas as entradas multiplicadas por seu peso calcula uma função de ativação, obtêm a saída do neurônio, e da mesma forma que as Redes Neurais usam métodos matemáticos para realizar seus cálculos na camada oculta o próximo tópico também trabalha métodos matemático, veremos isso a seguir.

2.4 PESQUISA OPERACIONAL

Para Cardoso (2011), o termo pesquisa operacional (PO) tem origem de uma área do conhecimento que busca desenvolver métodos científicos de sistemas complexos com o intuito de prever e comparar estratégias ou decisões alternativas, com o objetivo de dar suporte à definição e determinação de ações.

Pesquisa Operacional, conhecida como PO, pode ser notada tanto na administração quanto na matemática e pode ser definida de várias formas onde três destas não pode ficar de fora: o desejo de otimização e orientação às aplicações e o uso de métodos matemático (Colin, 2007).

Cardoso (2011) menciona que:

Resumidamente podemos dizer que o objetivo principal da PO é determinar a programação otimizada de atividades ou recursos, fornecendo um conjunto de

procedimentos e métodos quantitativos para tratar de forma sistematizada problemas que envolvam a utilização de recursos escassos.

A **Figura 10** demonstra os passos para a implementação de uma pesquisa operacional, em que se inicia pela definição do problema que será trabalhado, formulação que é a estruturação e as informações dos dados disponíveis, depois modelagem que significa apresentar o sistema de forma simples. Em seguida a solução que é encontrada através de métodos matemáticos, a avaliação que através de testes validar o modelo, a decisão que é a escolha da solução encontrada com o aprimoramento da mesma e por fim implementa-se a solução.

Figura 10: Passos para implementação de PO.



Fonte: Cardoso (2011).

Para Hillier e Lieberman (2013) a pesquisa operacional envolve pesquisa sobre operações e com isto a PO é aplicada a problemas que compreendem a condução e coordenação de operações em uma organização.

Desta forma a pesquisa operacional pode ser usada em diversas áreas funcionais como em estratégias (de dietas alimentares e investimento em subsídio) e finanças (orçamentos e seleção de projetos para atingir metas de lucros), por exemplo utilizando métodos estatísticos e probabilísticos para descobrir a melhor forma de realizar uma seleção e/ou um investimento buscando minimizar gastos.

2.4.1 TEORIA DOS JOGOS

Para os autores Sartini, Garbugio, Bortolossi, Santos e Barreto (2004) a *teoria dos jogos* é uma teoria matemática criada para que fenômenos possam ser observados quando mais de um “agente de decisão” interagem entre si, usadas para o estudo de assuntos como eleições, evolução genética, etc. Ainda não atingiu o patamar de conhecimento técnico escrito de como as decisões são feitas e como a economia funciona e é mais estudada nos aspectos matemáticos puro. Podendo ser definida como a teoria dos modelos matemáticos que estuda a escolha de decisões ótimas sob condições de conflito.

De acordo com Colin (2007) a *teoria dos jogos* é uma teoria matemática utilizada por jogadores que trata um meio de como formular suas estratégias considerando como estas são influenciadas. Criada na primeira década do século XX e tem evoluído bastante no ponto de vista teórico, mas, aparentemente não muito em suas aplicações.

Com isto percebe-se que a teoria dos jogos é um fundamento matemático utilizado para análise de determinado assunto e feita uma análise do mesmo formulando uma estratégia para sua resolução, onde será aplicada a melhor proposta.

2.4.2 ALGORITMOS DE SELEÇÃO/RANDON

Segundo Toledo (2011) o algoritmo conhecido por *seleção direta* baseia-se em passar sempre o menor valor do vetor para a primeira posição (ou o maior dependendo da ordem requerida), depois o de segundo menor valor para a segunda posição, e assim é feito sucessivamente com os (n-1) elementos restantes, até os últimos dois elementos, como apresentada na implementação feita em C na **Figura 11**.

Figura 11: Algoritmo de Seleção Direta

```
void selecaoDireta(int *vetor, int tamanho)
{
    int i, j, menor, aux;
    for(i = 0; i < tamanho - 1; ++i)
    {
        menor = i;
        for(j = i + 1; j < tamanho; ++j)
        {
            if(vetor[j] < vetor[menor])
                menor = j;
        }
        aux = vetor[i];
        vetor[i] = vetor[menor];
        vetor[menor] = aux;
    }
}
```

Fonte: Toledo, 2011.

De acordo com Feofiloff (2013) o algoritmo de seleção é tal como o de inserção que faz cerca de $n^2/2$ comparações entre os elementos do vetor no pior caso onde o algoritmo de seleção copia um elemento de um vetor de um lugar para outro. Quantas vezes isso ocorrer no melhor caso? Diferentemente do algoritmo de inserção, o de seleção também faz cerca de $n^2/2$ no melhor caso que leva o algoritmo a executar o menor número possível de comparações.

Figura 12: Algoritmo de Seleção.

```
void  
selecao (int n, int v[])  
{  
    int i, j, min, x;  
    for (i = 0; i < n-1; ++i) {  
        min = i;  
        for (j = i+1; j < n; ++j)  
            if (v[j] < v[min]) min = j;  
        x = v[i]; v[i] = v[min]; v[min] = x;  
    }  
}
```

Fonte: Feofiloff, (2013).

Na **Figura 12** observa-se que nos algoritmos de seleção são usadas condições para que sejam realizadas funções de modo que se a condição for positiva faz uma ação se negativa faz outra.

Após a análise dos métodos de desenvolvimento e das formas de seleção, decidiu-se trabalhar com a metodologia *ágil Scrum* utilizando alguns diagramas da *UML* e usufruindo da forma de seleção dos Algoritmos genéticos para o desenvolvimento do sistema de escala de trabalho das equipes do departamento infantil da SIBAPA.

3 METODOLOGIA

Para a realização do presente trabalho, fora realizado uma análise quanto as possibilidades de modos, operando quanto a sua perspectiva de aplicação, que fossem adequados ao projeto.

Com isso, percebe-se que metodologia é a estruturação da realização das atividades, de forma descritiva especificando cada uma delas em uma estrutura sequencial para chegar a um determinado fim ou objetivo. Tendo em vista isso, esta seção mostrará a metodologia que foi utilizada durante o desenvolvimento do projeto.

Várias são as modalidades de pesquisa que se podem praticar o que implica coerência epistemológica, metodológica e tecnológica, para seu adequado desenvolvimento (SEVERINO, 2007. pg. 118).

Assim, parece ser mais apropriada a metodologia quando os dados e sua interpretação derivam de fontes documentais de eventos passados. Se o estudo é do presente, pode-se utilizar o estudo de caso, levantamento ou pesquisa experimental. (BOAVENTURA, 2011. pg. 55-56).

Para Lakatos (2003) a pesquisa bibliográfica compreende oito fases distintas que são:

- Escolha do tema-> é o assunto no qual se deseja prover ou desenvolver, algo que ainda não possui solução, realizar o levantamento de fatos internos ou externos de algo;
- Elaboração do plano de trabalho-> pode proceder do fichamento ou ocorre depois de iniciada a coleta de dados bibliográficos, quando mais se dispõe de mais subsídios para elaboração do plano definitivo;
- Identificação-> é a fase de reconhecimento do assunto pertinente ao tema em estudo; o primeiro passo seria a procura de catálogos onde se encontram as relações das obras, o segundo passo, tendo em mãos o livro ou periódico, seria o levantamento, pelo Sumário ou Índice, dos assuntos nele abordados o último passo teria em vista a verificação da bibliografia ao final do livro ou do artigo, se houver, constituída, em geral, pela indexação de artigos de livros, teses, folhetos, periódicos, relatórios, comunicações e outros documentos sobre o mesmo tema;

- **Localização**-> tendo realizado o levantamento bibliográfico, com a identificação das obras que interessam, passa-se à localização das fichas bibliográficas nos arquivos das bibliotecas públicas, nas de faculdades oficiais ou particulares e outras instituições;
- **Compilação**-> é a reunião de todo o material contido em livros, revistas publicações avulsas ou trabalhos mimeografados;
- **Fichamento**-> á medida que o pesquisador tem em mãos as fontes de referência, deve transcrever os dados em fichas, com no máximo de exatidão e cuidado;
- **Análise e Interpretação**-> A primeira fase da análise e da interpretação é a crítica do material bibliográfico, sendo considerado, um Juízo de valor sobre determinado material científico, divide-se em crítica externa e interna, a segunda, terceira e quarta fases, respectivamente, decomposição dos elementos essenciais e sua classificação, generalização e análise crítica, correspondem às três da análise de texto. Finalmente, a interpretação exige a comprovação ou refutação das hipóteses. Ambas só podem ocorrer com base nos dados coletados.
- **Redação**-> A redação da pesquisa bibliográfica varia de acordo com o tipo de trabalho científico que se deseja apresentar. Pode ser uma monografia, uma dissertação ou uma tese.

Com isto, percebe-se que a metodologia bibliográfica se baseia em componentes de muita importância, pois, sem um tema e uma elaboração do plano de trabalho não se tem como decorrer as demais fases como, por exemplo, a identificação do assunto e levantamento do material bibliográfico para embasamento do projeto.

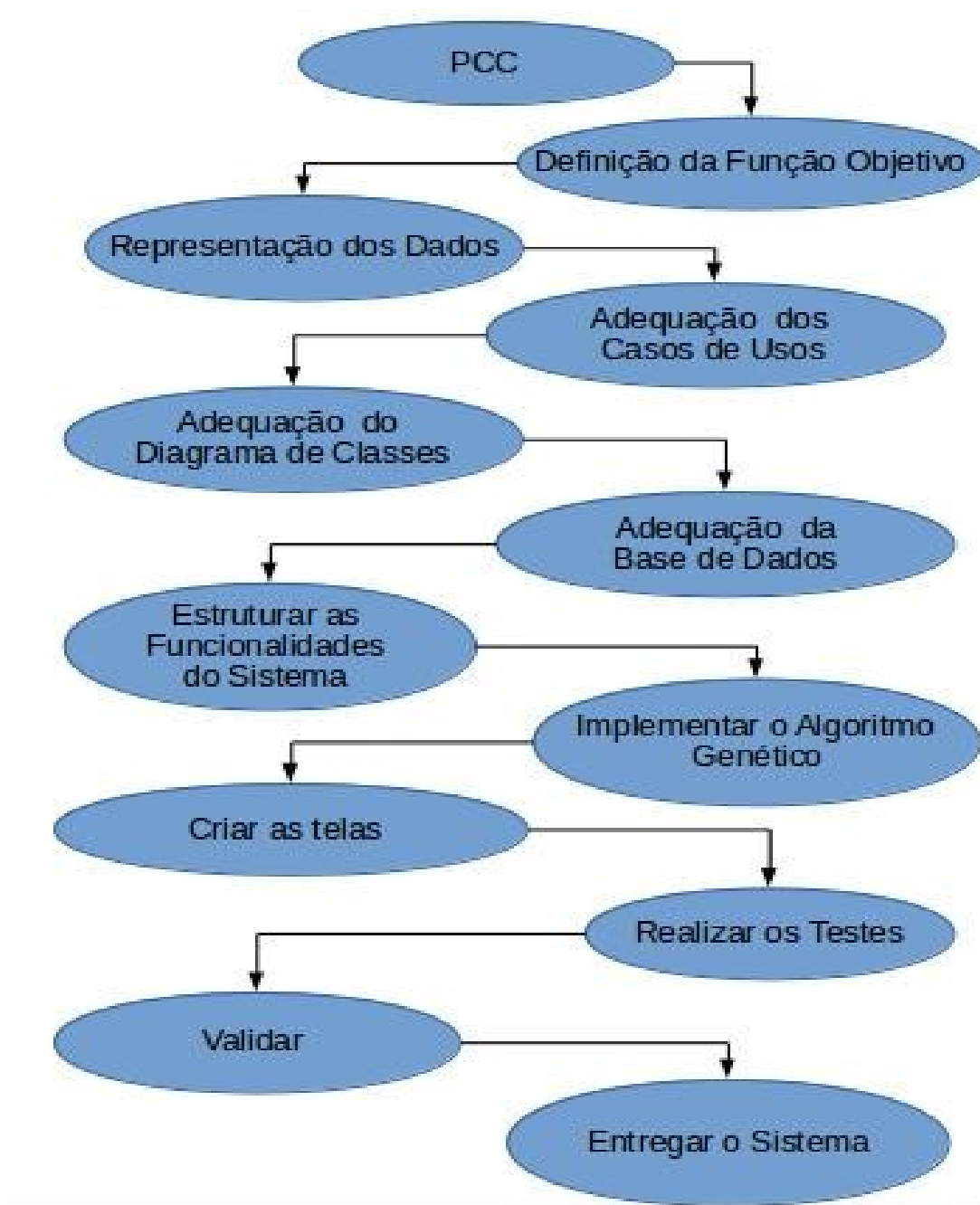
Metodologia é a via ou realização de algo em que se deve determinar como e onde a pesquisa será realizada, se será quantitativo ou qualitativo, se coleta de dados será realizada via formulário, questionário ou entrevista, por e-mail ou pessoalmente.

Também fora realizada, a escolha do melhor algoritmo para o desenvolvimento da função de gerar escala, dentre os diversos temas estudados no referencial teórico, foi escolhido o algoritmo genético, pois o problema trata-se de um típico para resolução com o algoritmo genético.

Assim, o presente trabalho foi realizado na segunda Igreja Batista em Palmas (SIBAPA) com o intuito de contemplar aproximadamente 180 voluntários a partir do início da sua utilização.

Para representar de maneira mais clara e objetiva metodologia utilizada, com os passos, fora criado o Diagrama abaixo, como apresentado na **Figura 13**, que poderá ser observado os passos que concretizaram o projeto onde:

Figura 13: Desenho da Metodologia Do Projeto



Fonte: do Autor, 2019.

Inicialmente, fora realizada pesquisa sobre o processo de desenvolvimento de Software, modelos de desenvolvimento Ágeis, pois, os métodos pesquisados foram utilizados para o desenvolvimento do software (sistema).

Após todo o estudo, o seguimento do projeto derivou-se da realização e análise da ata de reuniões, no qual foram feitas análises sobre o problema a ser solucionado e

através desta análise identificados os requisitos do sistema sendo eles funcionais e não funcionais, após a identificação dos requisitos os mesmos apresentados ao líder do Ministério Infantil da SIBAPA, sendo recebida sua aprovação e/ou sugestão para o sistema.

Depois da análise de requisitos, foi feito a análise para encontrar os casos de uso e fazer o detalhamento dos mesmos, onde os casos de uso foram utilizados para realização do diagrama de casos de uso, em seguida, o diagrama de classes que o sistema possui, e será utilizado para a realização da prototipação do mesmo. A Prototipação demonstrar a representação das telas que o usuário do sistema terá acesso.

Quanto ao banco de dados, foi necessária a realização da consulta na tabela de membros do banco de dados já existente para poder ser feita a realização do preenchimento do banco construído para o sistema, no qual possuiu dados do usuário cadastrados como voluntário, uma tabela de líderes, uma tabela de escalas, uma de calendário programado pelos líderes das equipes, uma de equipes, para posterior consulta.

Com base nos conhecimentos da IA o sistema utilizou o algoritmo genético auxiliado por um algoritmo de seleção para gerar a escala de forma aleatória. Deste modo, puderam ser feitas prioridades e realização de uma busca com a melhor estratégia para realizar a escala de acordo com a disponibilidade do voluntário e as datas cadastradas na tabela calendário do banco de dados.

Foram utilizadas para a implementação e conclusão do projeto, o *jasperreport* para configuração do *layout* de *download* das escalas; o *spring security* para configurar a autenticação do usuário do sistema, o *primefaces*, *JSF* e *JPA* para implementação, desenvolvimento e configurações necessárias do projeto; fazendo com que o mesmo deixe de se ser um projeto e se torne o sistema que se tornou.

Foi utilizado um *Laptop* com: 4 GB Memória *Ram DDR4*, 120 GB *SSD*, Processador *Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz*, localizado na Universidade Estadual do Tocantins.

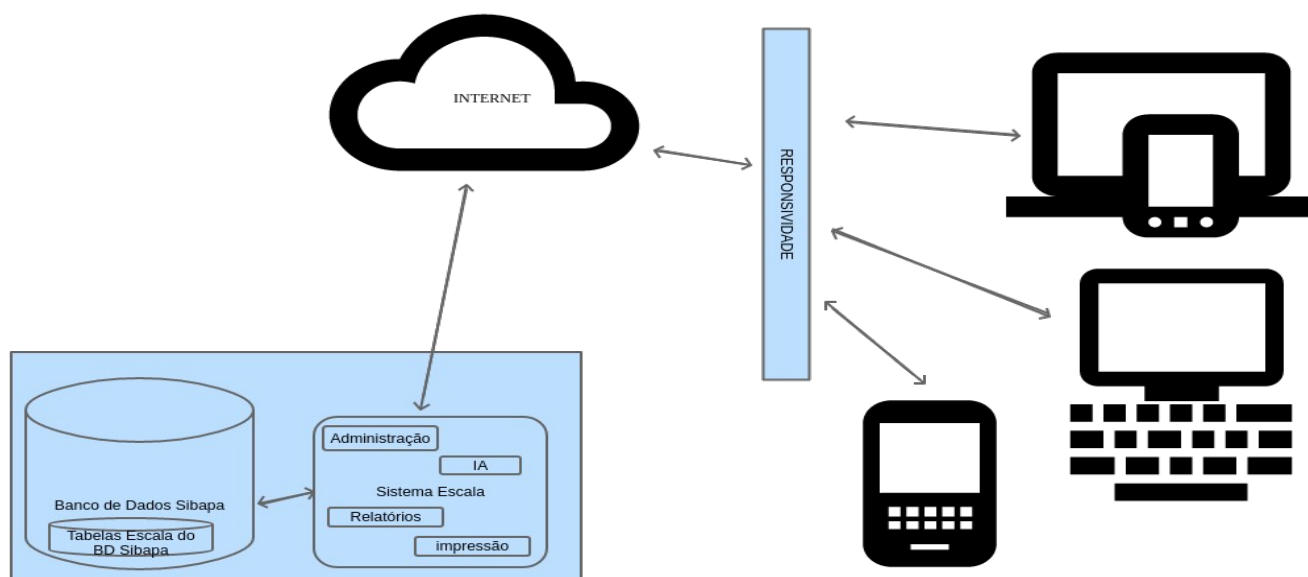
4 RESULTADOS E DISCUSSÕES

Neste tópico serão apresentados os resultados das atividades realizadas durante o projeto de conclusão, tendo maior ênfase no projeto de software.

Foi realizada uma reunião de cinco minutos com o líder do ministério, onde a mesma passou suas necessidades, com isso fora criada a ata de reunião. Dias depois, fora apresentado a mesma, o levantamento dos requisitos e diagramas necessários para seu desenvolvimento com a aprovação do mesmo. A referida ata de reunião, os diagramas de atividades, sequências, análise de requisitos, projeto do software, detalhamento dos casos de uso, foram descritos no projeto apresentado anteriormente e fazem parte do Apêndice do trabalho.

Com isso a proposta inicial é descrita conforme a arquitetura do sistema apresentada a seguir.

Figura 14: Arquitetura do sistema



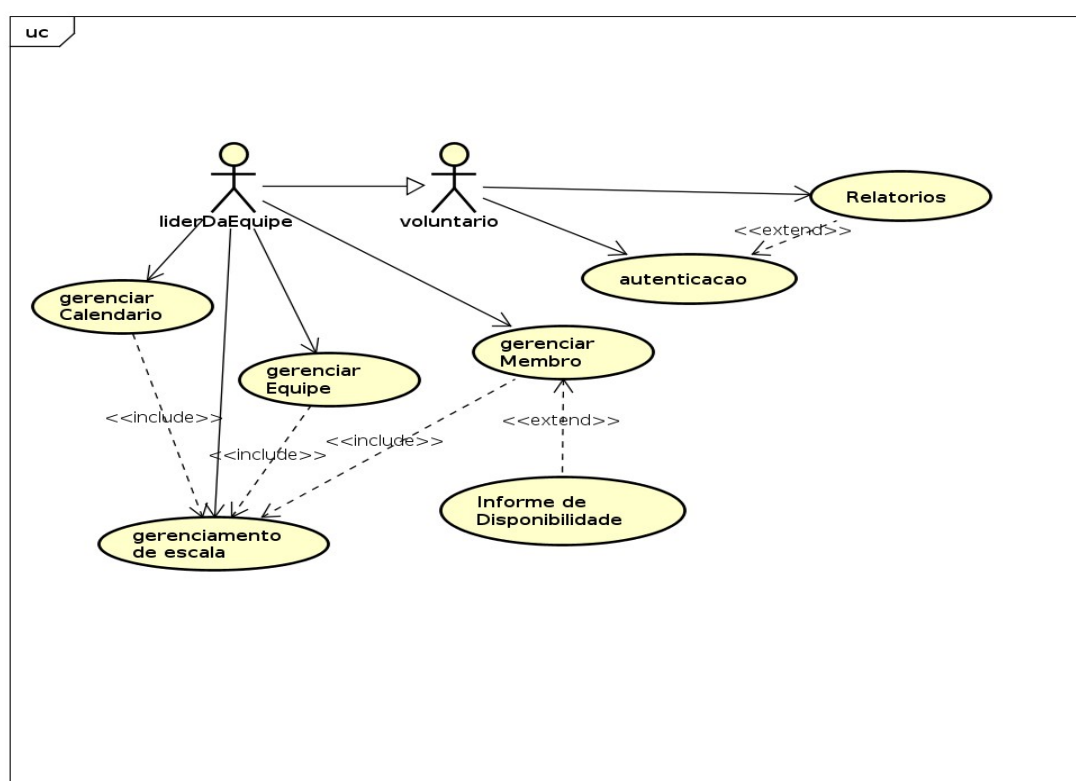
Fonte: do Autor, 2019.

De acordo com a arquitetura demonstrada na **Figura 14**, o sistema foi integrado ao banco de dados da igreja para consulta de membros, poderá ser acessado via computadores e dispositivos móveis. O sistema, consulta a tabela de membros da igreja com suas interfaces e interações, como a consulta dos dados que são utilizados na

implementação da tabela de voluntários, para poder dar continuidade das demais funções/atividades do sistema. O Sistema pode ser acessado pelos diversos tipos de aparelhos como computadores de mesa, *notebook*, celulares, *netbooks* e *tablets* facilitando o acesso de cada usuário.

Foram realizadas as verificações e devidas correções aos diagramas de casos de uso do sistema como apresentado na **Figura 15**.

Figura 15: Diagrama de Casos de Uso do Sistema



Fonte: do Autor, 2019.

No diagrama apresentado na **Figura 15**, são discriminadas as funcionalidades a serem realizadas pelo usuário, mostrando os atores do sistema, que são: usuário comum chamado de Voluntário e o usuário Administrador chamado de Líder da Equipe. Apresentam também as permissões de acesso de cada um dos tipos de atores. O ator voluntário, tem permissão de autenticação e consulta de escala e o ator líder de equipe tem as permissões do voluntário e as permissões de gerenciamento de membro (realiza consultas ao banco de dados e selecionará o membro para incluí-lo na equipe), de equipe (inclusão, consulta, deleção, alteração), escala e calendário.

Em seguida baseada na prototipação do sistema, são apresentadas as imagens das telas do sistema de acordo com a **Figura 16** que posteriormente foram utilizadas como as telas reais do sistema.

Figura 16: Tela de Gerenciamento do Líder de Equipe

Sistema Escala Trilhar

Alexandra

Menu Admin

- Cadastro de Voluntari
- Cadastro de Equipe
- Cadastro de Turma
- Cadastro de Eventos
- Gerar Escala
- Gerar Escala Geral
- Troca Escala

Menu Cliente

- Lista Escala Turma
- Lista Turma_Voluntari

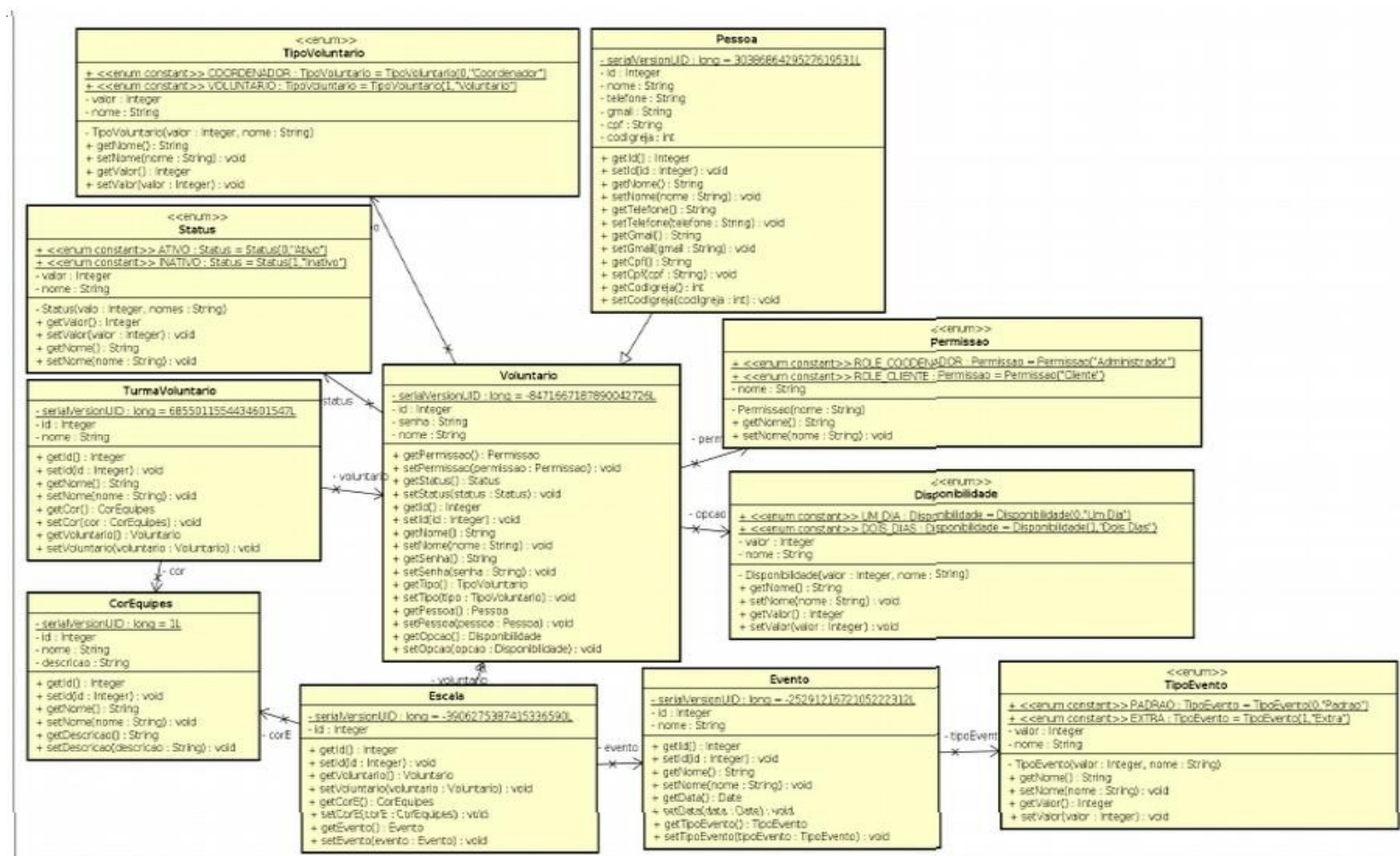
Gerar Escala

Gerar

Data	Nome
19/06/2019 00:00:00	Célia
19/06/2019 00:00:00	Isabela Dias
19/06/2019 00:00:00	Leidiane Chaves
19/06/2019 00:00:00	Adriana Jugton
19/06/2019 00:00:00	Suely Costa
19/06/2019 00:00:00	Deybianne
19/06/2019 00:00:00	Evelyn
19/06/2019 00:00:00	Geovanna Carvalho

Fonte: do Autor, 2019.

Figura 17: Diagrama de Classes do sistema

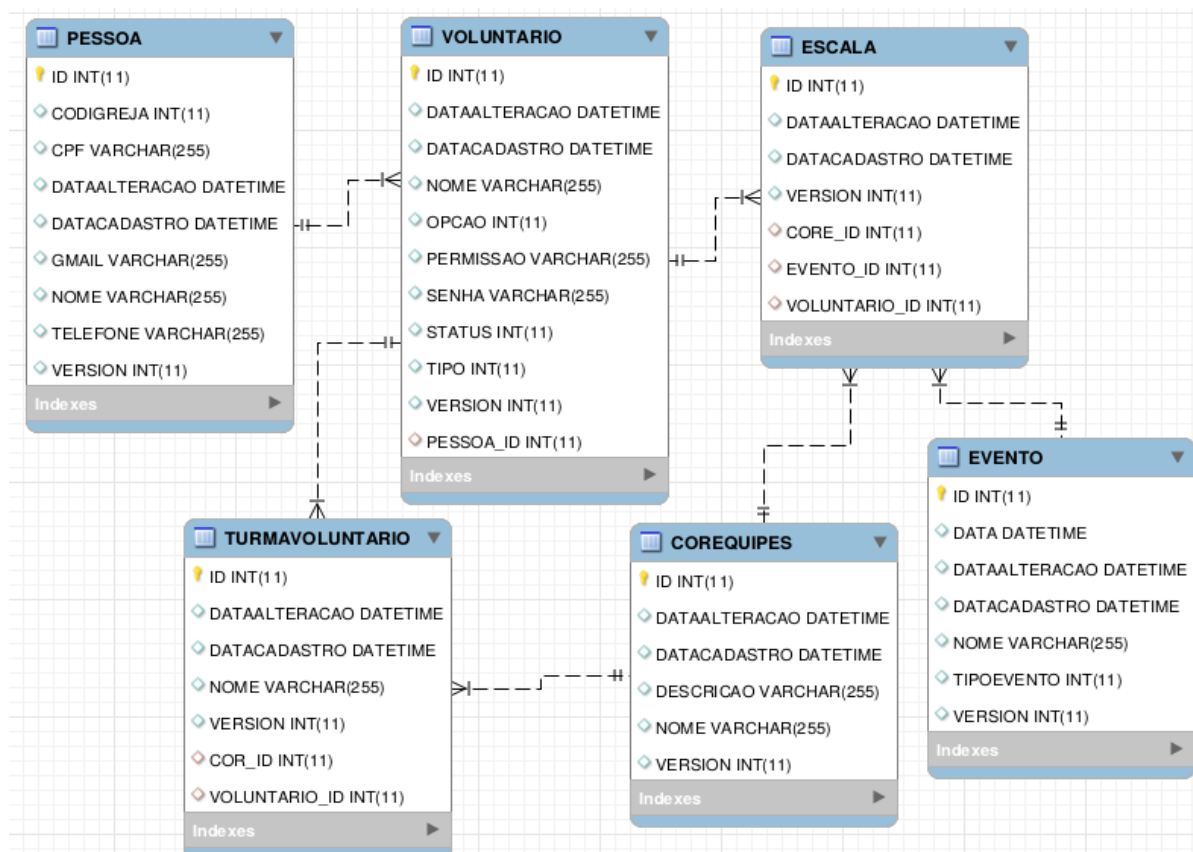


Fonte: do Autor, 2019

Como apresentado na **Figura 17** o sistema tem um voluntário que também pode ser coordenador. Depois de os voluntários serem inseridos serão montadas as equipes com o voluntário e a cor da equipe que foi cadastrada no sistema, em seguida cadastrados os eventos com suas datas e após estes cadastros foi realizada a geração da escala de trabalho com o uso do algoritmo genético, pois foi o escolhido e estará inserido no método de gerar escala para cada departamento/ministério. O AG (algoritmo genético) foi escolhido devido sua forma de criar as populações fazendo com que as escalas sejam definidas da melhor forma possível separando os indivíduos de acordo com suas disponibilidades.

Foi feito ainda a criação do protótipo do Banco de dados conforme demonstração na imagem da **Figura 18**.

Figura 18: Diagrama de Banco de dados do sistema



Fonte: do Autor, 2019.

Com base no diagrama de banco de dados foram criadas as tabelas que serão

utilizadas para armazenamento dos dados do sistema. Estas tabelas são armazenadas na base de dados da própria instituição para que possam ter maior facilidade de acesso com isto o sistema vai acessar.

A instituição disponibilizou a estrutura da tabela pessoa da base de dados na qual o sistema tem acesso para consulta de pessoa e cadastro de voluntário através da pessoa cadastrada.

Foi disponibilizada um acesso de consulta da tabela já existente, no qual o sistema consulta somente os dados necessários para cadastro do voluntário como índice, nome, telefone, e-mail.

Uma das dificuldades encontradas na estruturação da base de dados foi descobrir qual a melhor forma de integração já que o banco de dados a ser trabalhado já existe na instituição. Ou seja, já está em utilização devendo inicialmente possuir muito cuidado na integração, principalmente ao ser consultado e persistir dados.

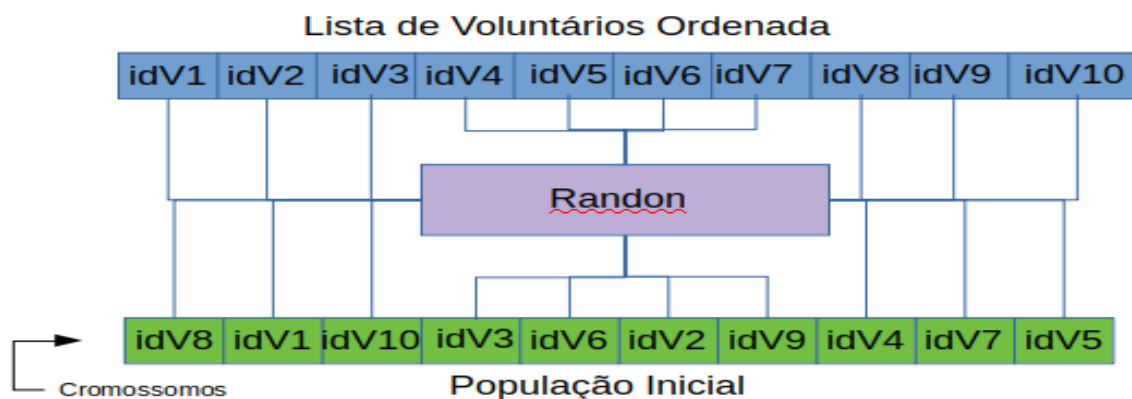
Porém, por se tratar do sistema já existente na comunidade e o sistema que estava sendo desenvolvido, houve alguns conflitos na comunicação com responsável do banco de dados. Assim, foi criado um tipo na tabela que disponibiliza ao sistema de escala e acessado por uma *View* de consulta, permitindo a visualização somente dos membros que continham este tipo criado impossibilitando o cadastro de voluntários que não tivessem este tipo. Com isto, não foi realizado o acesso ao banco de dados da comunidade, mas foi criado um banco de dados com uma tabela preenchida chamada PESSOA, com os dados dos membros para que o sistema consulte e realize o cadastro dos voluntários.

Superado o problema quanto aos dados, todo o trabalho foi concentrado no desenvolvimento do AG para geração da escala. Foram encontradas algumas dificuldades para definir a função objetivo. A referida função no fato é a forma na qual seria a resolução do problema proposto e definir como seriam representados os dados para a implementação. Com isso, foi decidido que a representação seria com os objetos ao invés de usar binário (0,1). Usou-se os objetos das classes implementadas (Voluntario, TurmaVoluntario, Evento e CorEquipe) e definiu-se que a função objetivo é gerar os indivíduos que são as escalas de cada equipe (Turma). Para isso foi usado como operador de seleção a roleta, já que existe a população inicial e os valores de aptidão atrelados a cada indivíduo da população para selecionar os pais da nova população

(geração) /(indivíduos). Em seguida foi feito o cruzamento por ponto duplo.

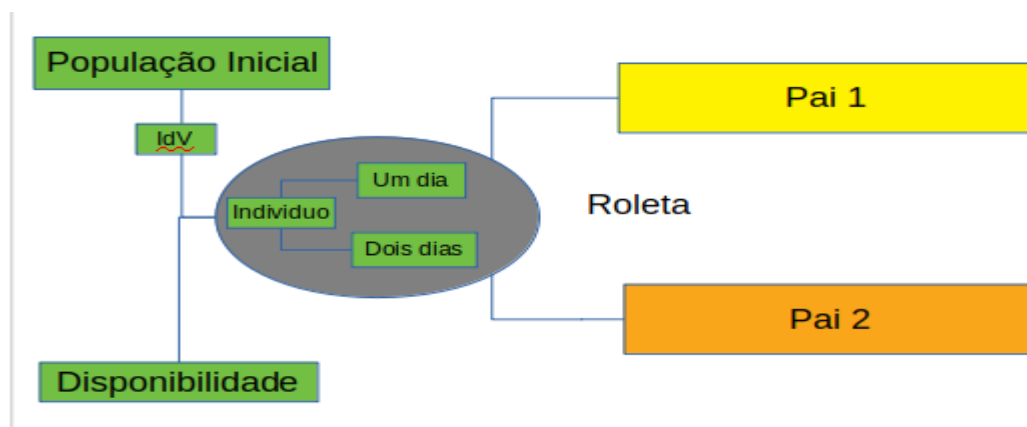
No desenvolvimento observou-se a necessidade de através dos dados existentes criar uma população inicial usando como representação o objeto da classe (TurmaVoluntario) e realizando as verificações de o voluntário pertence a equipe selecionada e sua disponibilidade. Após criada a população inicial, foram gerados os pais da nova população e em seguida feito o cruzamento de ponto duplo gerando dois novos indivíduos que são duas escalas, como é apresentado nas imagens do fluxograma das **Figuras 19, 20 e 21** e no trecho de código demonstrado a seguir.

Figura 19: Cria a População Inicial



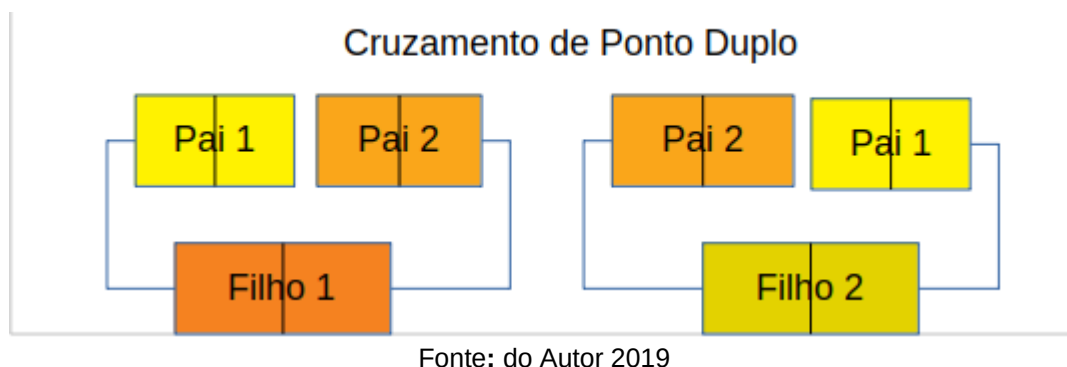
Fonte: do Autor 2019

Figura 20: Roleta para gerar os Pais



Fonte: do Autor 2019

Figura 21: Cruzamento de ponto duplo para gerar os Filhos



As **Figuras 19, 20 e 21** estão representando a forma de como os dados foram executados e nos **Quadros** a seguir os trechos de código do desenvolvimento.

Quadro 1: Código que Cria a População Inicial

```

1 //For de verificação para não repetição dos campos sorteados
2 for (int i=0; i<listaVoluntario.size(); i++) {
3     intervaloSorteio.add(i);
4 }
5 //Gera população inicial
6 for (int i=0; i<listaVoluntario.size(); i++) {
7     Random rand = new Random();
8     int posicaoSorteada = rand.nextInt(intervaloSorteio.size());
9     int conteudoSorteado = intervaloSorteio.get(posicaoSorteada);
10    listPopulacaoI.add(listaVoluntario.get(conteudoSorteado));
11    for (int j = 0; j<intervaloSorteio.size(); j++) {
12        if (conteudoSorteado==intervaloSorteio.get(j)) {
13            intervaloSorteio.remove(j);
14            break;
15        }
16    }
17 }
    
```

Fonte: do Autor, 2019

Na **Quadro 1** é apresentada as linhas de código que geram a população inicial do algoritmo, que das linhas 2 a 4 é feito um for para evitar que algum dado seja sorteado mais de uma vez adicionando a uma variável do tipo voluntário todos os voluntários que serão trabalhados no momento das gerações. Na linha 8 pega a posição sorteada e na 9 o conteúdo da posição sorteada em seguida na linha 10 é adicionada a uma outra variável também do tipo voluntário o dado pego na variável conteúdo. Em seguida das linhas 11 a 15 um for com uma condição *if* é feito para retirar do intervalo o objeto adicionado à lista

de população e isso é feito até que não tenha nenhum objeto na lista do intervalo.

Quadro 2: Código que Gera os Indivíduo pais

```

1 //For de verificação para não repetição dos campos sorteados
2 for (int i=0; i<listaVoluntario.size(); i++) {
3     intervaloSorteio.add(i);
4 }
5 //Gera população inicial
6 for (int k=0; i<listaVoluntario.size(); k++) {
7     Random rand = new Random();
8     //verifica a lista para adicionar na equipe A (pai1)
9     if (intervaloSorteio.isEmpty()) {
10         break;
11     } else {
12         int posicaoSorteadaA = rand.nextInt(intervaloSorteio.size());
13         int conteudoSorteadoA=intervaloSorteio.get(posicaoSorteadaA);
14         equipeA.add(listPopulacaoI.get(conteudoSorteadoA));
15         for (int j = 0; j<intervaloSorteio.size(); j++) {
16             if (conteudoSorteadoA==intervaloSorteio.get(j)) {
17                 intervaloSorteio.remove(j);
18                 break;
19             }
20         }
21     }
22     //vrifica a lista para adicionar na equipe B (pai2)
23     if (intervaloSorteio.isEmpty()) {
24         break;
25     } else {
26         int posicaoSorteadaB= rand.nextInt(intervaloSorteio.size());
27         int
28         conteudoSorteadoB=intervaloSorteio.get(posicaoSorteadaB);
29         equipeB.add(listPopulacaoI.get(conteudoSorteadoB));
30         for (int j = 0; j<intervaloSorteio.size(); j++) {
31             if (conteudoSorteadoB==intervaloSorteio.get(j)) {
32                 intervaloSorteio.remove(j);
33                 break;
34             }
35         }
36     }

```

Fonte: do Autor, 2019

Na **Quadro 2** são apresentados os comandos que criam os pais, no qual é representada a roleta. Com isso nas linhas 9-11 e 23-25 são feitas verificações para quando o intervalo não tiver mais dado, seja encerrada a execução, nas linhas 12 e 26 é pega a posição sorteada e nas 13 e 28, o conteúdo da posição sorteada em seguida nas linhas 14 e 29 é adicionada a uma outra variável também do tipo voluntário o dado pego na variável conteúdo, gerando assim os pais dos novos indivíduos as escalas.

Quadro 3: Código faz o cruzamento dos pais

```

1 //Cruzamento
2 int tamanhoA = equipeA.size();
3 int tamanhoB = equipeB.size();
4 int metadeA = (int) tamanhoA/2;
5 int metadeB = (int) tamanhoB/2;
6 for (int i=0; i<metadeA; i++) {
7     escalaA.add(equipeA.get(i));
8 }
9 for (int i=metadeA; i<tamanhoA; i++) {
10     escalaB.add(equipeA.get(i));
11 }
12 for (int i=0; i<metadeB; i++) {
13     escalaB.add(equipeB.get(i));
14 }
15 for (int i=metadeB; i<tamanhoB; i++) {
16     escalaA.add(equipeB.get(i));
17 }

```

Fonte: do Autor, 2019

Na **Quadro 3** é feito o cruzamento dos pais apresentados da **Tabela 2** para o qual metade do pai1 e metade do pai2 é inserido em uma lista e a outra metade de ambos os pais 1 e 2 é inserida em outra.

Foi decidido que, já que foram geradas duas escalas (listas) foi pego as datas dos eventos padrões e feita a intercalação das escalas nos mesmos, pelo fato no qual nos eventos extras todos voluntários são escalados.

Quadro 4: Código que cadastra as Escalas aos seus respectivos Eventos na Base de Dados

```

1 //grava dados na base
2 int tamanhoEvanto = listaEvento.size();
3 int metadeTamanhoEv = tamanhoEvanto/2;
4 for (int i = 0; i < listaEvento.size(); i++) {
5     if (listaEvento.get(i).getTipoEvento().getValor().equals(0)) {
6         for(int j = 0; j < equipeA.size(); j++) {
7             getEntity().setCorE(cor);
8             getEntity().setEvento(listaEvento.get(i));
9             getEntity().setVoluntario(equipeA.get(j));
10            insert(actionEvent);
11        }
12        i++;
13        for(int j = 0; j < equipeB.size(); j++) {
14            getEntity().setCorE(cor);
15            getEntity().setEvento(listaEvento.get(i));
16            getEntity().setVoluntario(equipeB.get(j));
17            insert(actionEvent);
18        }
19    }else{
20        for(int j = 0; j < listPopulacaoI.size(); j++)
21            getEntity().setCorE(cor);
22            getEntity().setEvento(listaEvento.get(i));

```

```

23         getEntity().setVoluntario(listPopulacaoI.get(j));
24         insert(actionEvent);
25     }
26 }
27 }
```

Fonte: do Autor, 2019

Os códigos inseridos na **Quadro 4** são a conclusão do método geraEscala() no qual é feita a inclusão das escalas em seus respectivos eventos de forma que se o evento for um evento do tipo "extra", todos os membros da equipe são escalados. Caso contrário apenas metade da equipe trabalhará e mesmo que tenha um evento extra em meio a dois padrões (ou comuns) a equipe a trabalhar depois do evento extra será a que folgou no evento anterior, deixando assim uma escala justa e sem preferencialismo na hora da seleção.

Quadro 5: Código que Troca os membros de Escala

```

1 public void getTrocaEscala(ActionEvent actionEvent) throws
2 OptimisticLockException, ValidationException, ApplicationException{
3     EscalaRepository repository = new
4 EscalaRepository(JPAFactory.getEntityManager());
5     Voluntario voluntario = getEscala().getVoluntario();
6     getEscala().setVoluntario(getEntity().getVoluntario());
7     getEntity().setVoluntario(voluntario);
8     update(actionEvent);
9     setEntity(getEscala());
10    update(actionEvent);
11 }
```

Fonte: do Autor, 2019

No **Quadro 5**, o sistema recebe da tela os dados dos voluntários e os eventos nos quais cada um está escalado, nas linhas 3 e 4 é criada uma variável auxiliar para gravar os dados de um dos voluntários enquanto na linha 6 a variável que teve o dado armazenado na auxiliar recebe os dados do outro voluntário, na linha 7 a entity a variável auxiliar é salva na linha 8, e na linha 9 é setada na entity os dados da outra variável e salva na linha 10, concluindo assim a troca de escala dos membros inseridos na tela.

Os dados apresentados anteriormente na geração de escala são dados para a geração por equipe onde é selecionada a equipe que será feita a escala e em seguida o sistema seleciona os membros da mesma e inicia o processo a partir da criação da

população inicial, dando sequência ao processo até o momento que é salvo o último dado na base de dados.

No caso da escala geral, o sistema também realiza fazendo o mesmo processo da escala por equipe, porém como não é informada a equipe, acaba deixando o campo equipe da tabela escala nulo/vazio.

Para os testes iniciais de operação do sistema foi usado o cadastro com dados reais da Equipe Laranja, com os dados dos voluntários da equipe fornecidos pela sua líder, os eventos também com datas reais, fornecidos por uma das coordenadoras da equipe de referência, com isso, pode-se constatar que os dados para testes são verídicos.

Visto que os testes iniciais com a equipe Laranja foram bem-sucedidos, foi feito teste mais amplo com as equipes Amarela, Branco e Rosa para ampliar e averiguar a veracidade do sistema. Os dados foram informados por cada líder de equipe.

Figura 22: Relatório de Escala

Relatório Escala	
Laranja	
Selecionar	
Data	Nome
29/05/2019 00:00:00	Alexandra
22/05/2019 00:00:00	Debora Carvalho
22/05/2019 00:00:00	Debora Nunes
29/05/2019 00:00:00	Gisele
22/05/2019 00:00:00	Laura
22/05/2019 00:00:00	Luciene
29/05/2019 00:00:00	Mauro
22/05/2019 00:00:00	Mikaela
29/05/2019 00:00:00	Mikaele
29/05/2019 00:00:00	Silvia

Fonte: do Autor, 2019

Como apresentado na **Figura 22**, uma das telas do sistema é a tela de listar a escala com a seleção a equipe, após isso, viu-se a necessidade de geração de relatório *PDF*, que foi implementado com *JasperReport*. Foi realizada a configuração de conexão com o banco e em seguida configurado o layout do *PDF* e do *select* de consulta, para carregar os dados da escala selecionada. Deixando assim implementados os relatórios para gerar o *PDF* da escala, caso o membro da escala queira tê-la impressa. Tal relatório encontra-se nas **Figura 23, 24, 25 e Figura 26**.

Figura 23: Design do Relatório de escala em PDF

Relatório de Escala			Escala Trilhar
"Equipe: " \$F{NOME}			
Column Header			
Data Escala	\$F{DATA}		
Voluntario(a)	\$F{COLUMN_3}		
Group Header 1			
new java.util.Date()		"Page "+\$V{ PAGE_NUMBER }+" of" " +	

Fonte: do Autor, 2019

Figura 24: JRXML do Relatório de escala em PDF

```

32 <group name="Group1">
33   <groupExpression><![CDATA[$F{DATA}]]></groupExpression>
34   <groupHeader>
35     <band height="33">
36       <staticText>
37         <reportElement mode="Opaque" x="0" y="0" width="100" height="32" forecolor="#666666" backcolor="#E6E6E6" />
38         <textElement>
39           <font size="12"/>
40         </textElement>
41         <text><![CDATA[Data Escala]]></text>
42       </staticText>
43       <textField>
44         <reportElement mode="Opaque" x="100" y="0" width="455" height="32" forecolor="#006699" backcolor="#E6E6E6" />
45         <textElement>
46           <font size="24" isBold="true"/>
47         </textElement>
48         <textFieldExpression><![CDATA[$F{DATA}]]></textFieldExpression>
49       </textField>
50       <line>
51         <reportElement x="-20" y="32" width="595" height="1" forecolor="#666666" uuid="6fc389c6-32d0-46cb-a8ea-61" />
52       </line>
53     </band>
54   </groupHeader>
55   <groupFooter>
56     <band/>
57   </groupFooter>
58 </group>

```

Fonte: do Autor, 2019

Houve um pouco de dificuldade na configuração do relatório, pelo fato de ser uma lista e a forma na qual o layout estava sendo configurado era para campos simples, que não recebiam listas até que, percebeu-se esta falha e encontrou-se a forma adequada para configuração, que era usando o campo “group name” que atribui a determinado tipo de campo, um grupo tornando assim possível a impressão de uma lista no PDF como apresentado na **Figura 25**.

Figura 25: Select Relatório de escala em PDF

```

6 <queryString language="SQL">
7 <![CDATA[select `escalaTrilhar`.`COREQUIPES`.`NOME`,
8 `escalaTrilhar`.`EVENTO`.`DATA`,
9 `escalaTrilhar`.`PESSOA`.`NOME`
10 FROM `escalaTrilhar`.`COREQUIPES`,
11 `escalaTrilhar`.`EVENTO`,
12 `escalaTrilhar`.`PESSOA`,
13 `escalaTrilhar`.`VOLUNTARIO`,
14 `escalaTrilhar`.`ESCALA`
15 WHERE `escalaTrilhar`.`COREQUIPES`.`ID`=`escalaTrilhar`.`ESCALA`.`CORE_ID`
16 AND `escalaTrilhar`.`EVENTO`.`ID` = `escalaTrilhar`.`ESCALA`.`EVENTO_ID`
17 AND `escalaTrilhar`.`VOLUNTARIO`.`ID` = `escalaTrilhar`.`ESCALA`.`VOLUNTARIO_ID`
18 AND `escalaTrilhar`.`PESSOA`.`ID` = `escalaTrilhar`.`VOLUNTARIO`.`PESSOA_ID`]]>
19 </queryString>

```

Fonte: do Autor, 2019

Figura 26: Relatório de escala em PDF

Relatório de Escala		Equipe: Laranja	Escala Trilhar
Data Escala	29/05/19 00:00		
Voluntario(a)	Alexandra		
Voluntario(a)	Silvia		
Data Escala	22/05/19 00:00		
Voluntario(a)	Debora Nunes		
Voluntario(a)	Debora Carvalho		
Data Escala	29/05/19 00:00		
Voluntario(a)	Gisele		
Data Escala	22/05/19 00:00		
Voluntario(a)	Laura		
Voluntario(a)	Luciene		
Data Escala	29/05/19 00:00		
Voluntario(a)	Mikaele		
Data Escala	22/05/19 00:00		
Voluntario(a)	Mikaela		
Terça-feira 21 Maio		Page 1 of 21	

Fonte: do Autor, 2019

No teste a escala geral foi testada e demorou cerca de 3,99 segundos para sua conclusão. Enquanto para gerar uma escala de equipes demora cerca de 3,75 segundos para sua conclusão. Com isso na **Figura 27** pode ser observado o gráfico criado com o intuito de demonstrar o tempo de geração de cada escala gerada com o algoritmo e é identificado uma pequena distância em sua geração e que ambas demoram pouco menos de 4 segundos para seu término.

Figura 27: Gráfico de Desempenho da Geração de Escalas



Fonte: do Autor, 2019.

O sistema está em funcionamento com a aprovação dos líderes de equipe, no qual os mesmos já fazem uso; e como pode ser observado na figura acima, as escalas possuem uma pequena diferença em sua execução. Isso se deve pelo fato de que, as amostras serem de quatro equipes apenas, com o aumento da quantidade, a diferença será proporcional a das amostras.

5 CONCLUSÃO

Para a realização do objetivo proposto, percebeu-se que seria necessária uma abordagem bem abrangente em todos os modelos de desenvolvimento de *software* e um bom conhecimento teórico para poder desempenhar uma implementação estruturada e bem-sucedida. Percebe-se também, que não se deve iniciar um projeto de imediato na linha de comando, pois é necessário primeiro entender o problema proposto pelo cliente, através de métodos de análise.

Observa-se que para o desenvolvimento de um sistema *web* é aconselhável o uso das práticas de desenvolvimento de *software*, pois, independente de ser ou não para a *web*, é um sistema que necessita da escolha de uma metodologia de desenvolvimento e uma boa documentação para ser implementado.

Para o desenvolvimento do sistema proposto no trabalho, o ponto crítico demonstrou ser necessária boa comunicação com os desenvolvedores e responsáveis pelo banco de dados já existente, no qual são necessárias consultas a tabelas populadas, gerando dados que foram recebidos pelo sistema de gerenciamento de escalas.

Já a utilização da *UML* foi essencial para nortear os passos do projeto com a criação dos diagramas de classe que demonstra como serão as classes do sistema, diagrama de atividades, exemplificando as atividades dos usuários, o diagrama de sequência no qual apresenta a sequência de cada atividade, e o diagrama de casos de uso que irá mostrar os casos de uso do sistema.

Quanto a *IA*, seu estudo foi relevante para a escolha do melhor algoritmo da criação da melhor escala para montá-las, sendo assim escolhido o algoritmo genético que cria sempre a melhor população, como foi demonstrado no trabalho. Com o levantamento da função objetivo e criação do algoritmo genético, fora observado, através de testes de execução de escala em pequeno e longo período de tempo, com dados reais fornecidos por líderes de equipes, uma boa execução e um bom desempenho com a geração da escala, o que a torna eficiente e satisfatória, já que ela cumpre com o objetivo proposto pelo usuário.

Para a aplicação e obtenção dos resultados, será necessário de no mínimo uma máquina com 4 GB Memória *Ram DDR4*, 120 GB *SSD*, Processador *Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz*.

Assim, o trabalho foi concluído com sucesso, sendo que podem ser trabalhados novos conceitos como:

- ☐ Fornecer um aplicativo para dispositivos móveis, tanto para *Android* quanto para *IOS*;
- ☐ Possibilitar que outras equipes possam utilizar o sistema;
- ☐ Fornecer a solução para outras instituições;

Tais situações e novas propostas podem contribuir ainda para trabalhos futuros como:

- ☐ O desenvolvimento de uma *API* para implementação de Aplicativos *Android* e *IOS*.
- ☐ Uma adequação para as necessidades de demais equipes da comunidade Sibapa.
- ☐ Criação de *Framework* para uso da escala

6 REFERÊNCIAS

- BOAVENTURA, Edivaldo M.. **Metodologia da Pesquisa:** Monografia Dissertação Tese. São Paulo: Atlas, 2011. 160 p.
- Colin, E., Carlos., Pesquisa Operacional: 170 Aplicações em estatística, finanças, logística, produção, marketing e vendas. Rio de Janeiro: Arte & Ideia LCT, 2007.
- Cordeiro, Gilliard., Aplicações Java para web com JSF e JPA. Casa do Código, 2012. 346 p.
- Cybis, Walter., Betiol, A. H., Faust, Richard., Ergonomia e Usabilidade: Conhecimentos, Métodos e Aplicações. 2º ed. São Paulo Novatec, 2010. 422 p.
- Haykin, Simon. **Redes Neurais:** Princípios e Práticas. 2. ed. São Paulo: Bookman, 2001. 906 p.
- Haykin, Simon., Redes Neurais: Princípios e Práticas. 2º ed. Porto Alegre: Bookman, 2001. 900 p.
- Lopes, Sérgio., A Web Mobile: Programe para um Mundo de Muitos Dispositivos. Casa do Código. 2013. 223 p.
- LAKATOS, Eva Maria, MARCONI. Marina de Andrade. **Fundamentos de Metodologia Científica.** 5. ed. - São Paulo : Atlas 2003.
- Oliveira, Hébert Coelho., JSF Eficaz - As melhores práticas para o desenvolvedor Java. Casa do Código: 2013. 181 p.
- Oliveira, Hébert Coelho., JPA Eficaz - As melhores práticas de persistência de dados em Java. Casa do Código: 2012. 346 p.
- Russell, Start J., Norvig, Peter., Inteligência Artificial. 2º ed. Rio de Janeiro: Elsevier 2004. 1021 p.
- SEVERINO, Antônio Joaquim. **Metodologia do Trabalho Científico.** 23. ed. São Paulo: Cortez, 2007. 304 p.
- SOMMERVILLE, Lan. **Engenharia de Software.** 8. ed. São Paulo: Pearson Addison Wesley, 2007. 552 p.
- Zemel, Tércio., Web Design Responsivo: Páginas Adaptáveis para todos os Dispositivos. Casa do Código. 2012. 151 p.
- AGIL, Comunidade Desenvolvimento. **Desenvolvimento Ágil de Software,** 2013/2014 disponível em: < , <http://www.desenvolvimentoagil.com.br/scrum/> > acessado

em 10 de novembro de 2017.

AGIL, Comunidade Desenvolvimento. **Desenvolvimento Ágil de Software**, 2013/2014 disponível em: < , <http://www.desenvolvimentoagil.com.br/xp/> > acessado em 10 de novembro de 2017.

Bernardi, D. Alencar., Técnicas de mapeamento objeto relacional - Revista SQL Magazine 40. Disponível em: < <https://www.devmedia.com.br/tecnicas-de-mapeamento-objeto-relacional-revista-sql-magazine-40/6980> >. Acessado em 22 de fevereiro de 2018.

BERNARDO, Kleber. **Cultura Ágil: Métodos Ágeis**, 2015 disponível em: <<https://www.culturaagil.com.br/o-que-sao-metodos-ageis/>> acessado em: 02 de outubro de 2017.

Barreto, Jorge M., Introdução às Redes Neurais Artificiais (2002). Disponível em: < <http://www.inf.ufsc.br/~j.barreto/tutoriais/Survey.pdf> > Acessado em 10 de fevereiro de 2018.

Cardoso, Andréa., Fundamentos da Pesquisa Operacional (2011). Disponível em: < <https://docplayer.com.br/5281170-Fundamentos-da-pesquisa-operacional.html> >. Acessado em: 10 de fevereiro de 2018.

Feofiloff, Paulo., Ordenação Algoritmos Elementares (2016). Disponível em: < <https://www.ime.usp.br/~pf/algoritmos/aulas/ordena.html#best-case-selection> >. Acessado em: 05 de fevereiro de 2018.

Freitas, Daniele., Interação Homem~Computador. 2015.1. Disponível em: < <https://docente.ifrn.edu.br/daniellefreitas/disciplinas/ihc/aulas/Aula2-design%20de%20interacao.pdf> >. Acessado em 16 de fevereiro de 2018.

GOULART, Talita, Desenvolvimento Web(2013). Disponível em <https://developer.mozilla.org/pt-BR/docs/desenvolvimento_web>. Acessado em 05 de fevereiro de 2018.

Hiller, F., S., Lieberman, Introdução a Pesquisa Operacional. 9º ed. MC Graw Hill bookman, 2013. 1004 p. Disponível em: < https://books.google.com.br/books?id=-A88a0-KxQ0C&printsec=frontcover&dq=Pesquisa+operacional&hl=pt-BR&sa=X&ved=0ahUKEwixkOK-u_jiAhVpDrkGHw4CXcQ6wEIKjAA#v=onepage&q=Pesquisa%20operacional&f=false >. Acessado em 10 de fevereiro de 2018.

LOPES, Ben-Hur de Sousa, DIAS, Jaime William (2015). Utilizando os Diagramas da

UML(Linguagem Unificada de Modelagem) para desenvolver aplicação em JSF. Disponível em:< web.unipar.br/~seinpar/2015/_include/artigos/BenHur_de_Sousa_Lopes.pdf > Acessado em 10 de dezembro de 2017.

Lustosa, Volney Gadelha., Alvarenga, Prof. Dr. Rogério.,O Estado da Arte em Inteligência Artificial. 2004. Disponível em: < http://pead.ucpel.tche.br/revistas/index.php/colabora/article/viewFile/60/53 >. Acessado em 15 de fevereiro de 2018.

Linden, Ricardo, Algoritmo Genético Uma Importante Ferramenta da Inteligência Computacional 2º edição (2008). Disponível em: < https://books.google.com.br/books?id=it0kv6UsEMEC&printsec=frontcover&dq=Linden,+Ricardo,&hl=pt-BR&sa=X&ved=0ahUKEwjZqJCetvjAhXQH7kGHWK1BCsQ6wEIKjAA#v=onepage&q=Linden%2C%20Ricardo%2C&f=false>. Acessado em 05 de fevereiro de 2018.

Miletto, Evandro Manara., Bertagnolli, Silva de Castro, Desenvolvimento de Software II: Introdução ao desenvolvimento web com HTML, CSS, JAVASCRIPT e PHP(2014). Disponível em: < https://books.google.com.br/books?id=lcLFAwAAQBAJ&printsec=frontcover&hl=pt-BR&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false > Acessado em: 06 de fevereiro de 2018.

Mercado, N. B. Guzmán., Técnica de Busca Baseada em Algoritmo Genético para Localização de p-Mediana (2001). Disponível em: < https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/81702/189369.pdf?sequence=1&isAllowed=y >. Acessado em 09 de fevereiro de 2018.

Machado, F. J. P. Martins., Inteligência Artificial e Arte. 2006. Disponível em: < https://estudogeral.sib.uc.pt/handle/10316/9599 >. Acessado em 15 de fevereiro de 2018.

MAGALHÃES, G. **Protocolos TI:** Tutoriais, artigos e matérias sobre Desenvolvimento de Software, Os Modelo de Desenvolvimento de Software, 2012 disponível em: <http://protocoloti.blogspot.com.br/2012/03/os-modelos-de-desenvolvimento-de.html> acessado em: 30 de setembro 2017.

MARTINEZ, Marina (2015) **UML.** Disponível em: <http://www.infoescola.com/enenharia-de-software/uml/>. Acesso em: 10 de dezembro de 2017.

MTR, Leandro. **Metodologia ágil Crystal,** 2016 disponível em: <

<http://www.leandromtr.com/gestao/metodologia-agil-crystal/> > acessado em: 20 de novembro de 2017.

Rosa, T., de Oliveira., Luz, H. Souza., Conceitos Básicos de Algoritmos Genéticos: Teoria e Prática (2009). Disponível em: < <http://docplayer.com.br/3981222-Conceitos-basicos-de-algoritmos-geneticos-teoria-e-pratica.html> >. Acessado em 28 de janeiro de 2018.

SANTOS, Rogério Garcia. LUZ, Giulian Dalton. **Métodos Ágeis**: Crystal Processo de Desenvolvimento de Software, disponível em: < https://www.ime.usp.br/~gdaltonl/ageis/ageis_6pp.pdf > acessado em: 20 de novembro de 2017.

SANTOS, Jorge., Quintiliano, Leandro Lago., Barbosa, Lílian., Silva , Paulo Roberto H. da., Júnior, Wanderley Giralde . **Métodos Ágeis Parte 02**, disponível em: < <https://www.devmedia.com.br/metodos-ageis-parte-02/9443> > acessado em 10 de novembro de 2017.

SCHIMIGUEL, Juliano. **Agile Development**: XP e Scrum em uma Abordagem Comparativa, 2014 disponível em: < <https://www.devmedia.com.br/agile-development-xp-e-scrum-em-uma-abordagem-comparativa/30808> > acessado em: 20 de novembro de 2017.

SABBAGH, Rafael (2014) **Scrum**: Gestão ágil para projetos de sucesso. Disponível em:< https://books.google.com.br/books?id=pG-CCwAAQBAJ&printsec=frontcover&hl=pt-BR&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false >. Acessado em: 10 de dezembro de 2017.

SILVA, Aristófanos C. (2015) **Unified Modeling Language (UML)**. Disponível em: <http://www.deinf.ufma.br/~acmo/MOO_Obj.pdf>. Acesso em: 10 de dezembro de 2017.

Toledo, Rafael., Algoritmo de Ordenação #2= Seleção Direta (2011). Disponível em: < <http://www.rafaeltoledo.net/algoritmos-de-ordenacao-2-selecao-direta/> >. Acessado em: 08 de fevereiro de 2018.

Vargas, Patrícia Amâncio, Sistemas Classificados Para Redução de Perdas em Redes de Distribuição de Energia Elétrica(cap.3 Algoritmos Genéticos: Conceitos Básicos e Extensões Vinculadas ao Problema de Minimização de Perdas), (2000). Disponível em: < http://www.dca.fee.unicamp.br/~vonzuben/research/pvargas_mest.html >. Acessado em

04 de fevereiro de 2018.

VARGAS, Thânia C. S. (2007) **A História de UML e seus Diagramas**. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_721/artigo.tcc.pdf>. Acessado em: 10 de dezembro de 2017.

Tafner, M. Anderson, Redes Neurais Artificiais: Aprendizado e Plasticidade(1998). Disponível em: < <http://www.cerebromente.org.br/n05/tecnologia/nervoso.htm> , <http://www.cerebromente.org.br/n05/tecnologia/plasticidade2.html> >. Acessado em 09 de fevereiro de 2018.

7 APÊNDICES

7.1 APÊNDICE 1

Universidade Estadual do Tocantins

Prof. Orientador: Mestre Alex Coelho

Acadêmico (a): Alecxandra Mesquita do Nascimento Valadares

Pastora: Lídia Mury

Ata de Reunião do Sistema de Escala Ministério Infantil

Na (SIBAPA) Segunda Igreja Batista de Palmas - TO Com CNPJ: 01.618.985/0001-86 tendo sua sede Administrativa/Auditório Sibapa situada na, 208 Sul, Alamedropõe-se que seja feito um sistema que automatize a criação das escalas, pois esta, por ser feita de forma manual, acaba criando um preferencialismo entre os voluntários e uma divisão entre a equipe no todo, pois, a mesma é dividida em dois grupos/turma como, por exemplo, equipe A sendo dividida em A.1 e A.2, e tirando a comunhão entres a mesma.

Com base nas informações acima: o Sistema deverá ter uma tela de autenticação;

Deve ser informado o calendário semestral para que possa ser feita as escalas de acordo com o calendário.

Os membros das equipes deverão ser alimentados pelo banco de dados do sistema já existente (Nome, Telefone).

As equipes devem ser cadastradas com (Nome e Cor).

Cada equipe deverá ser dividida em 2 fazendo uma subdivisão da equipe toda, sendo seus membros distribuídos semestralmente de forma aleatória com o uso de um algoritmo de seleção.

Cada membro deverá ter a informação de sua disponibilidade para atuação em 1 ou 2 vezes por mês.

Deixar definido os líderes das equipes onde cada equipe possui dois (as), e na escala que um estiver o outro (a) não poderá estar, a não ser que, seja no dia em que todos os voluntários estiverem escalados.

Deve ser possível incluir uma pessoa uma ou duas vezes por mês na escala, dependendo da disponibilidade do membro.

Deverá haver a possibilidade de troca, onde um membro trocará com outro e ambos trabalharão duas escalas seguidas.

Em dias de confraternizações deve ser escalada toda a equipe.

Quando houver uma escala da equipe completa os voluntários que folgaram uma escala antes deverão ser os escalados.

Utilizando banco de dados utilizado pela instituição, desenvolvido em linguagem de programação orientada a objeto (JAVA) via WebService.

Deverá possuir o nome de domínio informado pelo gestor de comunicação da Congregação onde lá serão encontradas as escalas de todas as turmas e um acesso para a autenticação do líder de equipe caso este esteja acessando e/ou queira fazer alguma alteração ou inclusão.

Com isto seu desenvolvimento deverá ter um período de conclusão de quatro a oito meses.

Alexandra Mesquita do Nascimento Valadares

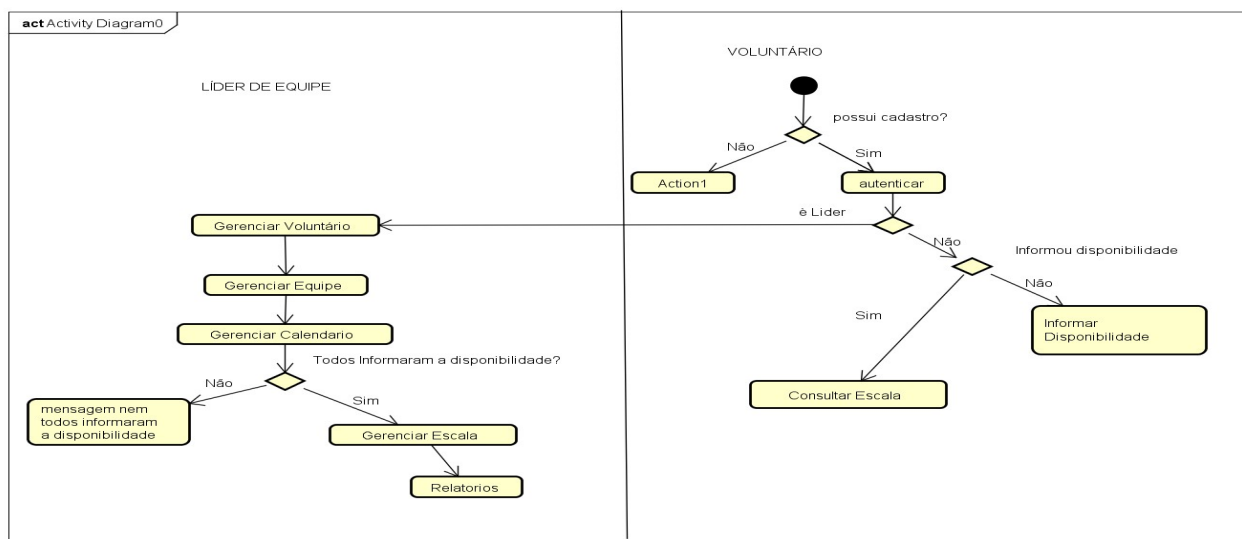
Lídia Mury

Palmas, _____ de _____ de 2017.

7.2 APÊNDICE 2

7.2.1 ATIVIDADES

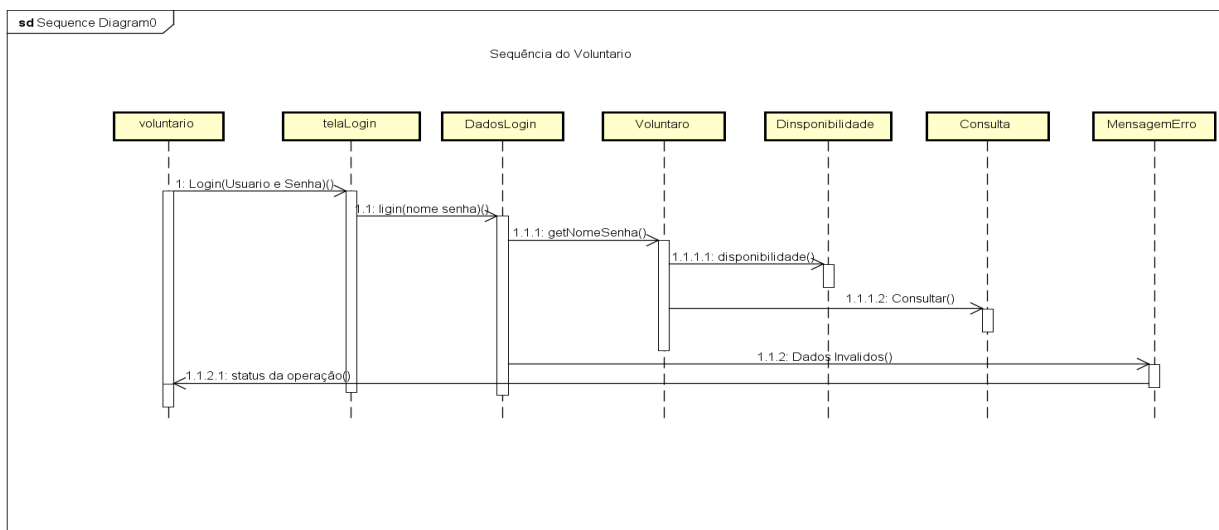
Diagrama de Atividade dos Casos de Uso



Fonte: do Autor, 2017.

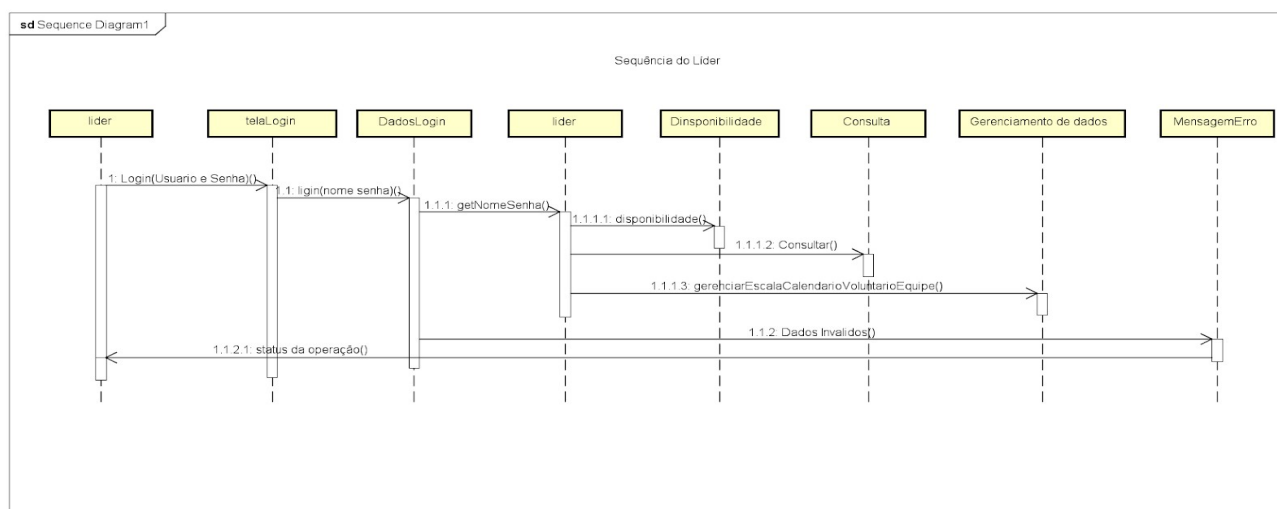
7.2.2 SEQUÊNCIA

Diagrama de Sequência das Atividades do Voluntário



Fonte: do Autor, 2017.

Diagrama de Sequência das Atividades do Líder de Equipe



Fonte: do Autor, 2017.

7.3 APÊNDICE 3

Análise de Requisitos Funcionais e não Funcionais

Requisitos Funcionais:

ID	Requisito Funcional	Descrição	Caso de Uso
RF001	Controle de acesso de usuários ao sistema	Receberá um conjunto de chaves de modo único que deverão ser comparados no banco... Usuários possíveis adm e Membro líder de equipe	Autenticação.
RF002	Gerenciar Membro	Buscar os membros no banco de dados via WebService	Gerenciar Membro
RF003	Gerenciar Escala da Equipe	Permitir Inclusão, exclusão, alteração, consulta e trocas de escalas entre membros; Impressão na tela a escala.	Gerenciar Escala da Equipe
RF004	Gerenciar Equipe	Permitir Inclusão, exclusão, alteração e consulta.	Gerenciar Equipe
RF005	Gerenciar Calendário	Inserir , exclusão alteração e consulta	Gerenciar Calendário

Requisitos não Funcionais:

ID	Descrição
RNF001	Banco de Dados Gratuito Mysql
RNF002	Linguagem (Java) e JSF
RNF003	Layout do sistema será feito em compatibilidade ao sistema já existente
RNF004	A autenticação deve ser feita via confirmação de senha e nome do usuário
RNF005	4 GB Memória <i>Ram DDR4</i> , 120 GB SSD, Processador <i>Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz</i>

7.4 APÊNDICE 4

7.4.1 DETALHAMENTO DO CASO DE USO

7.4.1.1 CASO DE USO: AUTENTICAÇÃO

I DESCRIÇÃO

I O caso de uso “Autenticação no Sistema” permitirá o acesso do usuário as suas atribuições de acordo com o tipo de usuário.

I ATORES

I Líderes de Equipe e voluntários (poderão atuar sobre este caso de uso)

I PRÉ-CONDIÇÕES

I Os líderes e voluntários deverão estar cadastrados no sistema

I FLUXO DE EVENTOS

◦ FLUXO PRINCIPAL

- O líder e/ou voluntário acessará o site.
- O líder e/ou voluntário informará seu nome e senha na página de autenticação do site.

I PÓS-CONDIÇÕES

I É permitido o acesso ao sistema do usuário cadastrado de acordo com suas atribuições e/ou permissões.

7.4.1.2 CASO DE USO: GERENCIAMENTO DE MEMBROS

☐ DESCRIÇÃO

I DESCRIÇÃO

O caso de uso “Autenticação no Sistema” permitirá o acesso do usuário as suas atribuições de acordo com o tipo de usuário.

I ATORES

Líderes de Equipe e voluntários (poderão atuar sobre este caso de uso)

I PRÉ-CONDIÇÕES

Os líderes e voluntários deverão estar cadastrados no sistema

I FLUXO DE EVENTOS

◦ FLUXO PRINCIPAL

- O líder e/ou voluntário acessará o site.
- O líder e/ou voluntário informará seu nome e senha na página de autenticação do site.

PÓS-CONDIÇÕES O caso de uso “Gerenciamento de Membros” descreve as operações realizadas para o registro na base de dados de Gerenciamento de membros nas equipes.

☐ ATORES

Os Líderes (por herança do Administrador também poderão atuar sobre este caso de uso)

☐ PRÉ-CONDIÇÕES

O líder deve ser cadastrado no sistema.

☐ FLUXO DE EVENTOS

- inicial do site.
- O líder consulta e seleciona o membro já cadastrado no banco de dados da Igreja.

- O líder consulta e seleciona a equipe na qual o membro participará.

☐ **PÓS-CONDIÇÕES**

Este caso de uso iniciará quando o líder realizar a operação correspondente ao gerenciamento de membros.

- **FLUXO PRINCIPAL**

- O líder informará o nome e senha no sistema.
- O líder escolherá a opção gerenciamento de membros na página inicial do site.
- O líder consulta e seleciona o membro já cadastrado no banco de dados da Igreja.
- O líder consulta e seleciona a equipe na qual o membro participará.

☐ **PÓS-CONDIÇÕES**

É registrado o cadastro de novos membros a equipe já existente.

7.4.1.3 CASO DE USO: GERENCIAR ESCALA

☐ **DESCRIÇÃO**

O caso de uso “Gerenciar Escala” permitirá o acesso da escala de acordo com as atribuições de Gerenciar Escala (Inserir, excluir, alterar).

☐ **ATORES**

Líderes de Equipe (poderão atuar sobre este caso de uso)

☐ **PRÉ-CONDIÇÕES**

Os líderes deverão estar cadastrados no sistema e membro da equipe da escala a qual quer gerenciar.

☐ **FLUXO DE EVENTOS**

Este caso de uso se inicia quando o líder realizar a operação

que corresponde a Gerenciar Escala.

- **FLUXO PRINCIPAL**

- O líder informará seu nome e senha na página de autenticação do site.
- O líder clicará no botão gerenciar escala.
- O líder poderá alterar, excluir e trocar voluntário de escala.

- **PÓS-CONDIÇÕES**

É permitida a troca de escala entre voluntários da mesma equipe

7.4.1.4 CASO DE USO: GERENCIAR EQUIPE

- **DESCRIÇÃO**

O caso de uso “Gerenciar Equipe” descreve as operações realizadas para o registro na base de dados de Gerenciar Equipe.

- **ATORES**

Os Líderes (por herança do Administrador também poderão atuar sobre este caso de uso).

- **PRÉ-CONDIÇÕES**

O líder deve ser cadastrado no sistema.

O líder deve ser membro da equipe.

- **FLUXO DE EVENTOS**

Este caso de uso iniciará quando o líder realizar a operação correspondente a gerenciar equipe.

- **FLUXO PRINCIPAL**

- O líder informará o nome e senha na página de autenticação do site.
- O líder clicará no botão gerenciar escala.
- O líder poderá alterar, excluir e Inserir uma equipe.

- **PÓS-CONDIÇÕES**

É permitido realizar alterações e cadastro de equipe.

7.4.1.5 CASO DE USO: GERENCIAR CALENDÁRIO

I DESCRIÇÃO

O caso de uso “Gerenciar Calendário” descreve as operações realizadas para o registro na base de dados de Gerenciar Calendário.

I ATORES

Os Líderes (por herança do Administrador também poderão atuar sobre este caso de uso)

I PRÉ-CONDIÇÕES

O líder deve ser cadastrado no sistema.

I FLUXO DE EVENTOS

Este caso de uso iniciará quando o líder realizar a operação correspondente a gerenciar calendário.

◦ FLUXO PRINCIPAL

- O líder informará o nome e senha na página de autenticação do site.
- O líder clicará no botão gerenciar calendário.
- O líder poderá informar os períodos nos quais haverá eventos diferentes dos comuns.

I PÓS-CONDIÇÕES

É permitido realizar alterações nos eventos.

Alexandra Mesquita do Nascimento Valadares

Lídia Mury

Palmas, _____ de _____ de 2017.

7.5 APÊNDICE 5

7.5.1 Visão de Negócio

Construção de um sistema de escalas com o intuito de auxiliar na organização das equipes do Ministério Infantil da Segunda Igreja Batista em Palmas - TO (SIBAPA), para reduzir a sobrecarga de voluntários com a organização e reorganização de escalas, de forma a satisfazer toda a equipe e contribuir com a congregação, com a criação deste sistema o ministério infantil poderá usufruir de um sistema de escalas automatizado, deixando um pouco as planilhas eletrônicas onde ao fim de cada semestre são excluídas e/ou alteradas por quem as manuseiam e estarão guardadas em um banco de dados para caso haja alguma necessidade, além de economizar um percentual de \$100,00 dólares que seria usado na compra de licenças.

7.5.1.1 Objetivo

Construir um sistema que automatize a criação de escalas no ministério infantil integrado ao sistema já existente de forma que reduza o uso de planilhas eletrônicas e os gastos com a contratação e/ou aquisição de um sistema para a criação das escalas.

7.5.1.2 Situação atual

Atualmente a instituição possui um sistema de escala realizado através do uso de planilha eletrônica que são printadas na tela e enviado para seus voluntários por uma rede social.

7.5.1.2.1 Documentação de levantamentos

Os documentos que foram levantados foram a Ata de Reuniões no qual se pode ter uma noção sobre o que o contratante deseja para o sistema.

7.5.1.3 Situação proposta

Construir um sistema web que realize de forma aleatória a escala de voluntários e possibilita que o mesmo tenha acesso a visualização de sua escala via sistema, no qual sua localização para acesso ao banco de dados da empresa o próprio banco da empresa.

7.5.2 Requisitos

7.5.2.1 Requisitos Funcionais

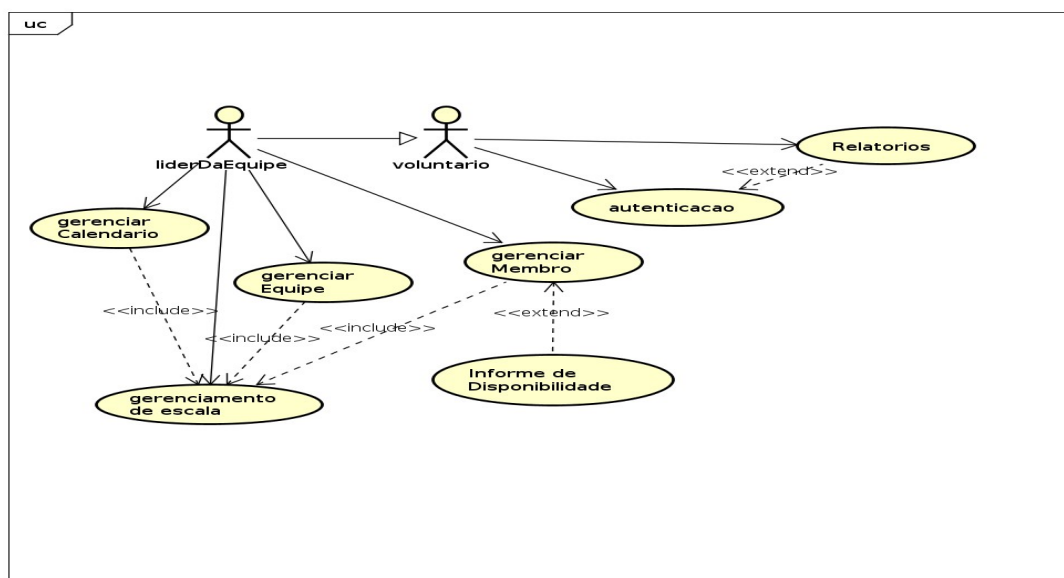
Gerenciar calendário, gerenciar escala, gerenciar líder e voluntário login (solicitar e-mail, senha), consulta.

7.5.2.2 Requisitos Não Funcionais

Gerar escala aleatoriamente de acordo com a disponibilidade de cada voluntário, alteração no banco de dados, busca dos dados do membro,

7.5.3 Diagrama de Casos de Uso

Diagrama de Casos de Uso do Sistema



Fonte: do Autor, 2019.

7.5.3.1 Detalhamento do caso de Uso

Autenticar: O usuário deve estar logado no sistema mediante cadastro previamente feito, a autenticação é feita por meio do respectivo nome e senha informada na hora do cadastro.

Gerenciar Membro/Voluntário: O líder poderá cadastrar os voluntários em suas respectivas equipes, e também poderá gerenciá-los, removendo-os ou alterando-os.

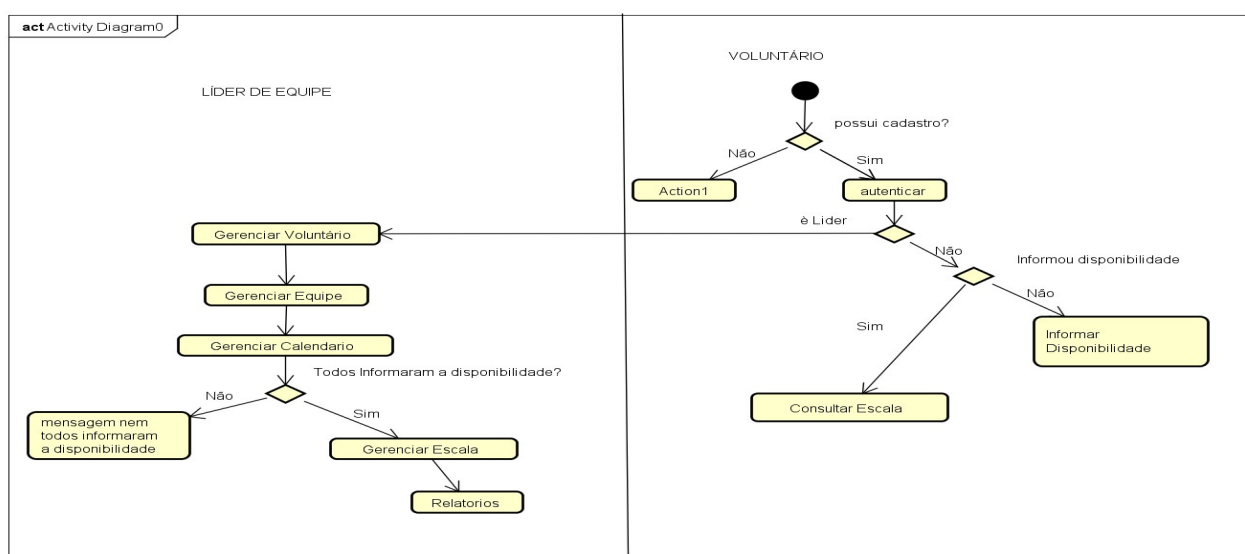
Gerenciar Escala: O usuário terá o acesso da escala de acordo com as atribuições

Gerenciar Equipe: O líder deve ser membro da equipe e está cadastrado no sistema para poder realizar este gerenciamento.

Gerenciamento de Calendário: O líder poderá informar os períodos nos quais haverá eventos diferentes dos comuns e possui permissão para realizar alterações nos eventos.

7.5.4 Diagrama de Atividades

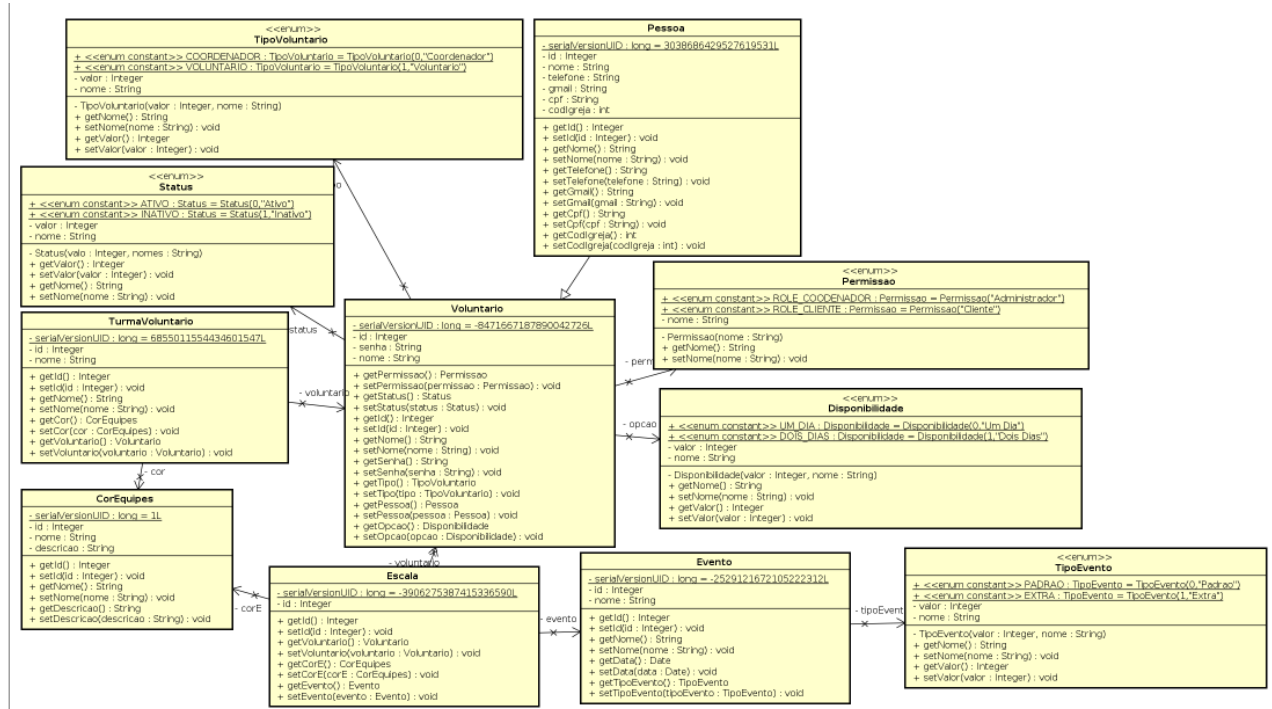
Diagrama de Atividade dos Casos de Uso



Fonte: do Autor, 2017.

7.5.5 Diagrama de Classes

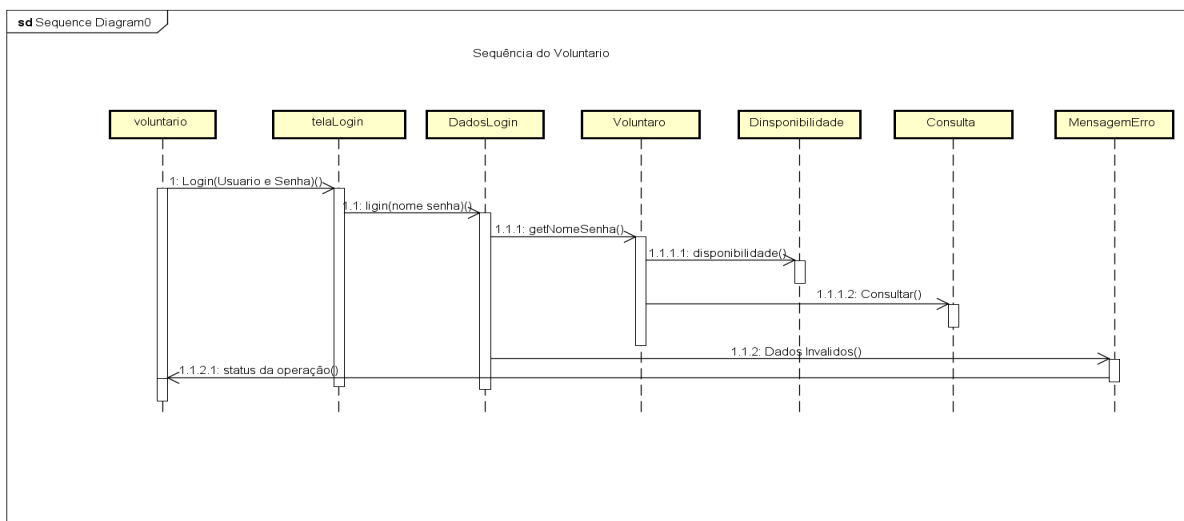
Diagrama de Classes do sistema



Fonte: do Autor, 2017.

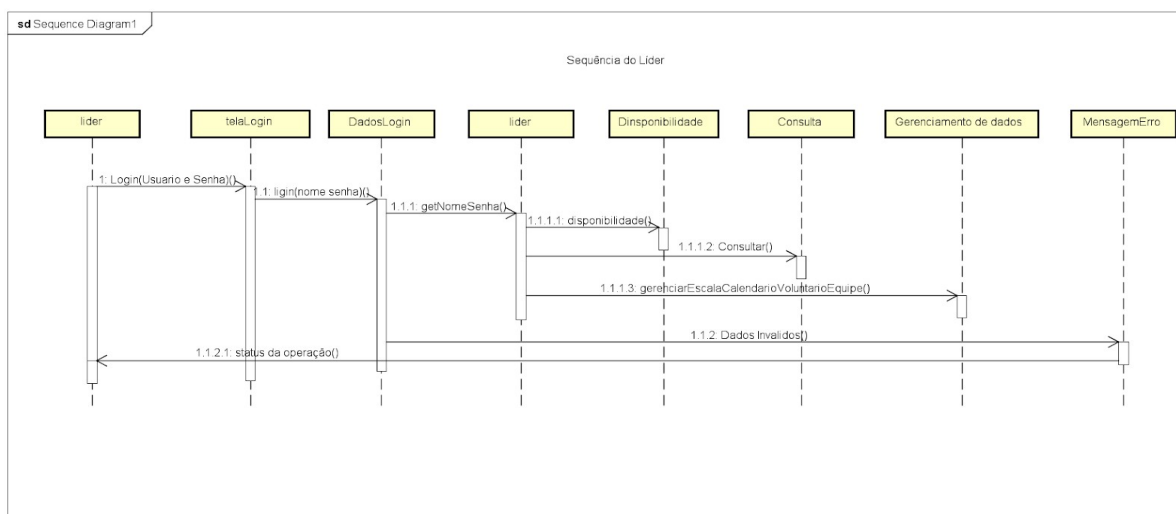
7.5.6 Diagrama de Sequência

Diagrama de Sequência das Atividades do Voluntário



Fonte: do Autor, 2017.

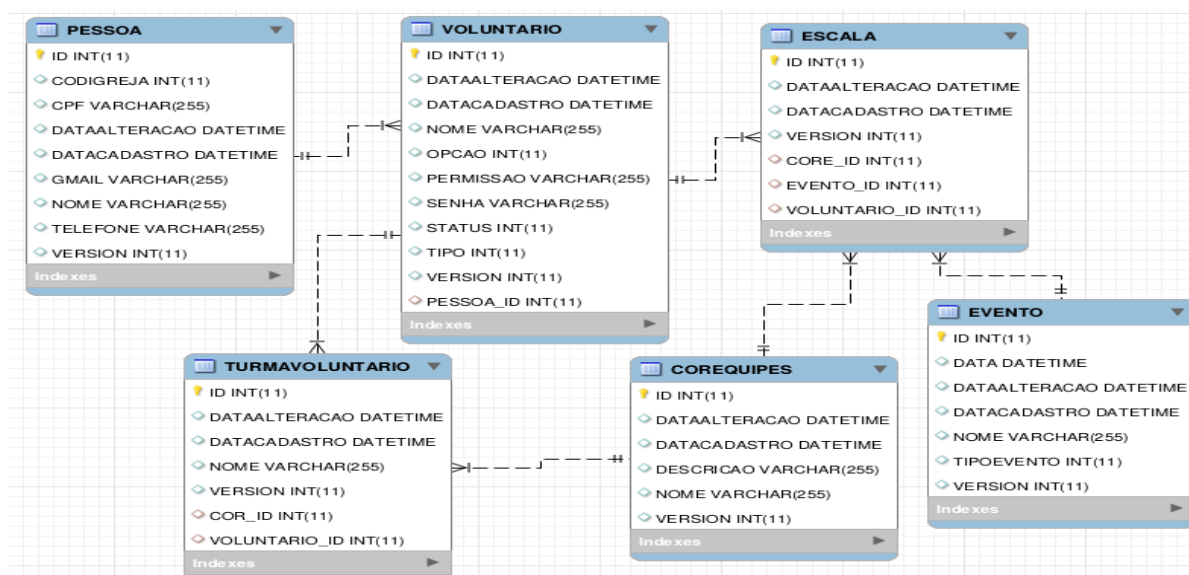
Diagrama de Sequência das Atividades do Líder de Equipe



Fonte: do Autor, 2017.

7.5.7 Projeto de Banco de Dados

Diagrama de Banco de dados do sistema



Fonte: do Autor, 2019.