

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
CAMPUS AVANÇADO DE NATAL
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

JOSSANA TARCILA WANDERLEY DA COSTA

COMANDANDO EM CASA: APLICAÇÃO GAMIFICADA DE REALIDADE
ALTERNATIVA PARA ESTIMULAR CRIANÇAS E ADOLESCENTES NA
REALIZAÇÃO DE TAREFAS DOMÉSTICAS

NATAL
2019

JOSSANA TARCILA WANDERLEY DA COSTA

COMANDANDO EM CASA: APLICAÇÃO GAMIFICADA DE REALIDADE
ALTERNATIVA PARA ESTIMULAR CRIANÇAS E ADOLESCENTES NA
REALIZAÇÃO DE TAREFAS DOMÉSTICAS

Monografia apresentada à Universidade do Estado do Rio Grande do Norte – UERN – como requisito obrigatório para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Dr. Alberto Signoretti

NATAL

2019

C837c Costa, Jossana Tarcila Wanderley da
COMANDANDO EM CASA: APLICAÇÃO GAMIFICADA
DE REALIDADE ALTERNATIVA PARA ESTIMULAR
CRIANÇAS E ADOLESCENTES NA REALIZAÇÃO DE
TAREFAS DOMÉSTICAS. / Jossana Tarcila Wanderley da
Costa. - Natal-RN, 2019.
57p.

Orientador(a): Prof. Dr. Alberto Signoretti.
Monografia (Graduação em Ciência da Computação).
Universidade do Estado do Rio Grande do Norte.

1. Ciência da Computação. I. Signoretti, Alberto. II.
Universidade do Estado do Rio Grande do Norte. III.
Título.

JOSSANA TARCILA WANDERLEY DA COSTA

COMANDANDO EM CASA: APLICAÇÃO GAMIFICADA DE REALIDADE
ALTERNATIVA PARA ESTIMULAR CRIANÇAS E ADOLESCENTES NA
REALIZAÇÃO DE TAREFAS DOMÉSTICAS

Monografia apresentada à Universidade do Estado do Rio Grande do Norte – UERN – como requisito obrigatório para obtenção do título de Bacharel em Ciência da Computação.

Aprovado em ____/____/____

Banca Examinadora

Dr. Alberto Signoretti

UERN

Prof. Me. André Gustavo Pereira da Silva

UERN

Prof. Me. Raul Benites Paradedda

UERN

A minha família, amigos, professores e a Deus por me ajudarem na jornada acadêmica.

AGRADECIMENTOS

À Deus por me dar forças para continuar e me abençoar sempre. Ao meu orientador e aos meus professores que me ensinaram muito durante o curso, sou grata pela orientação e todos os conhecimentos passados. Aos meus pais e irmãos pela confiança depositada em mim e todos os investimentos que foram necessários para realização deste sonho. Aos colegas do curso pelas ajudas, apoio, críticas e sugestões durante o curso. À universidade pela oportunidade e apoio.

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema do tipo aplicação web para o apoio a famílias que têm em suas casas crianças e adolescentes pouco participativas na realização de tarefas domésticas. Esta pesquisa mostra os desafios e possíveis soluções para resolução do problema enfrentado por essas famílias, assim como a descrição de como foi construída a aplicação que utilizou o uso de técnicas de jogos chamado 'gamificação', usando o método de ganho de pontos ao realizar tarefas pré-definidas por usuários. A aplicação foi desenvolvida de forma responsiva, onde se adequa a diversas plataformas com conexão à internet, como computadores, *tablets* e *smartphones*, para ser bem utilizada por qualquer usuário. Após o referencial teórico, a aplicação final é mostrada, assim como os resultados dos testes feitos e aplicados em famílias reais.

Palavras-chave: Aplicação em Tarefas Domésticas. Jogos Digitais. Gamificação. Sistema Web. Crianças e adolescentes.

ABSTRACT

This work presents the development of a web application to support families who have children and adolescents with little participation in housework. This research shows the challenges and possible solutions to solve the problem faced by these families, as well as the description of how the application was built using the techniques of games called 'gamification', using the points gain method when performing tasks pre-defined by users. The application was developed in a responsive way, where it is suitable for several platforms with internet connection, such as computers, tablets and smartphones, to be well used by any user. After the theoretical reference, the final application is presented, as well as the results of the tests made and applied in real families.

Keywords: Application in Housework. Digital games. Gamification. Web system. Children and Adolescents.

LISTA DE FIGURAS

1 - Sequência do modelo Cascata	17
2 - Site responsivo em diversas plataformas.....	21
3 - Estrutura de dados do banco MongoDB	23
4 - Estrutura da arquitetura do MongoDB.....	24
5 - Exemplo de diagrama de casos de uso.	25
6 - Diagrama de Caso de Uso da aplicação.....	30
7 - Diagrama de fluxo de páginas.....	31
8 - Diagrama de atividades da autenticação de usuário.....	32
9 - Separação de arquivos do sistema desenvolvido.	34
10 - Tela inicial na plataforma Desktop - Login de acesso.....	37
11 - Tela de Cadastro de Administrador na plataforma Desktop.....	37
12 - Tela Home da Aplicação na plataforma desktop.....	38
13 - Tela de Menu da Aplicação na plataforma desktop	38
14 - Tela de Perfil e Tela de Chat em Smartphone.	39
15 - Telas de tarefas e de ranking na plataforma Mobile.	40
16 - Tela de participantes e tela de menu administrativo na plataforma mobile	41
17 - Respostas da família 01 dos testes.	46
18 - Respostas da família 02 dos testes.	47
19 - Respostas da família 03 dos testes.	48

SUMÁRIO

1 INTRODUÇÃO	11
1.1 MOTIVAÇÃO	13
1.2 PRETENSÕES DE CONTRIBUIÇÃO	13
1.3 ORGANIZAÇÃO DA MONOGRAFIA	13
2 METODOLOGIA	15
2.1 METODOLOGIA DE PESQUISA BIBLIOGRÁFICA	15
2.2 FERRAMENTAS ESCOLHIDAS E SUAS DEFINIÇÕES	16
2.2.1 Definição do modelo de desenvolvimento escolhido	17
2.2.1.1 Análise e definição dos requisitos	17
2.2.1.2 Projeto do sistema.....	18
2.2.1.3 Implementação.....	18
2.2.2 Conceitos e definições de linguagens, <i>frameworks</i> e ferramentas utilizadas	18
2.2.2.1 Node.js	18
2.2.2.2 Express.js.....	19
2.2.2.3 EJS.....	20
2.2.2.4 JQuery.....	20
2.2.2.5 Bootstrap.....	20
2.2.2.6 Banco de dados NoSQL.....	21
2.2.2.7 Banco de Dados MongoDB.....	22
2.2.2.8 Linguagem de modelagem unificada (UML).....	24
2.2.2.9 Diagrama de casos de uso da UML	25
2.2.2.10 Diagrama de Atividades da UML.....	25
2.3 PROCESSO DE DESENVOLVIMENTO	26
2.3.1 Requisitos de interface de usuário	26
2.3.2 Requisitos de interface de hardware	26
2.3.3 Requisitos de interface de software	26
2.3.4 Requisitos de interface de comunicação	27
2.3.5 Requisitos do modo de operação	27
2.3.6 Requisitos de armazenamento.....	27
2.3.7 Requisitos de acesso.....	28

2.3.8 Requisitos de interface gráfica	28
2.3.9 Requisitos específicos.....	28
2.3.10 Requisitos de operações	29
2.3.11 Requisitos gerais.....	29
2.3.12 Casos de uso	29
2.4 FLUXO DE PÁGINAS DA APLICAÇÃO	30
2.4.1 Autenticação do usuário.....	31
2.4.2 Segurança de dados confidenciais.....	33
2.4.3 Arquitetura e organização do desenvolvimento	34
3 RESULTADOS E APLICAÇÃO FINAL.....	36
4 TESTES E AVALIAÇÕES.....	42
4.1 REQUISITOS TESTADOS	42
4.1.1 Teste do Banco de Dados.....	42
4.1.2 Teste Funcional	42
4.1.3 Teste da Interface do Usuário	43
4.1.4 Teste de Segurança e de Controle de Acesso	43
4.1.5 Teste de Instalação	43
4.1.6 Resultados dos testes	43
4.1.7 Testes com usuários reais	44
4.1.8 Conclusões dos testes reais	48
5 CONCLUSÃO	50
5.1 TRABALHOS FUTUROS	50
REFERÊNCIAS.....	52

1 INTRODUÇÃO

A participação de crianças e adolescentes nas tarefas domésticas e sua importância vêm se tornando alvo de muitos estudos e pesquisas por sociólogos, psicólogos e por terapeutas ocupacionais.

Um exemplo é a tese de doutorado de Adriana de França Drummond cujo título é “Participação de crianças e de adolescentes nas tarefas domésticas” (DRUMMOND, 2014), onde mostra a importância, as consequências e o processo de aprendizagem na participação de crianças e adolescentes na realização de tarefas domésticas, segundo pesquisas, testes e a partir do ponto de vista de especialistas.

O objetivo desses estudos é compreender que resultados e influências o envolvimento das crianças e adolescentes pode ter na realização dessas tarefas. Algumas dessas pesquisas apontam que essa participação nas tarefas domésticas afeta diretamente o desenvolvimento da criança e do adolescente.

Um dos motivos é o fato de o ambiente doméstico provocar uma gama de desafios e oportunidades para que a criança possa trabalhar sua independência e inclusão social, visto que a aprendizagem das tarefas domésticas envolve objetivação, tomada de decisões e resolução de problemas, estimula também a cooperação, independência e cuidado com o outro (DUNN et al, 2009b).

Como os estudos mencionados mostram, essa participação em tarefas domésticas possibilita vantagens, se forem aplicadas da maneira correta, porém há uma grande dificuldade para colocar essa ideia em prática. Uma pesquisa realizada pela “AVG Technologies com famílias de todo o mundo mostrou que 66% das crianças entre 3 e 5 anos de idade conseguia usar jogos de computador, 47% sabia como usar um *smartphone*, mas apenas 14% era capaz de amarrar os sapatos sozinha” (EXAME, 2011).

As crianças em especial adoram andar com vassouras, baldes e panos, imitando os adultos da casa, até a hora de terem de fazer tudo isso a sério. Estimular a participação em tarefas que necessitam de algum tipo de esforço físico em crianças e adolescentes que hoje convivem em meio a uma era de facilidades e divertimentos tecnológicos torna essa aplicação ainda mais difícil.

Alguns autores dizem que o uso da tecnologia por crianças e adolescentes

traz desvantagens como atraso no desenvolvimento, afirmando que o uso de tecnologia restringe os movimentos físicos, o que pode resultar em atraso no desenvolvimento motor. A movimentação reforça a capacidade de atenção e aprendizado (RATEY, 2008).

Existem outros problemas relacionados ao uso excessivo da tecnologia, um deles é o aumento nas taxas de obesidade que pode ser gerada pela facilidade do uso dessas tecnologias, na falta de necessidade de movimento para utilizá-las, tornando assim seus usuários mais sedentários, outro problema que o uso excessivo da tecnologia pode gerar é a agressividade.

Outros autores defendem o uso da tecnologia, mostrando que ela pode trazer benefícios como a facilidade em se estabelecer comunicação, a propagação de ideias, o acesso à informação educacional, cultural e artístico. Dessa forma é visto que existem vantagens e desvantagens.

Os jogos digitais são um exemplo de tecnologia que é muito usado por crianças e adolescentes atualmente, como a pesquisa da AVG Technologies mostra que 66% das crianças entre 3 e 5 anos de idade fazem uso de jogos digitais (SANCHES; CARDELINO; RAMOS, 2014). E como meio tecnológico tem seu lado positivo e negativo, dependendo da forma como aplicado e usado.

A gamificação, que vem do inglês “*gamification*”, pode ser defendida como sendo uma estratégia apoiada na aplicação de elementos de jogos para atividades não definidas como jogos, sendo utilizada para influenciar e causar mudanças no comportamento de indivíduos e grupos (BUNCHBALL INC., 2010).

O uso de jogos digitais disponíveis em diversas plataformas mostra-se bastantes favorável a experiências de gamificação em crianças e adolescentes. No caso das crianças brasileiras, um levantamento apontou que 97% das crianças entre 6 e 9 anos usam a internet e 54% têm perfil no Facebook (SANCHES; CARDELINO; RAMOS, 2014).

O que se tem estudado e aplicado por diversos desenvolvedores de games é a criação jogos com objetivos de engajar pessoas, resolver problemas e melhorar o aprendizado, motivando ações e comportamentos em ambientes fora do contexto de jogos. O uso dessas ferramentas de entretenimento, como *smartphones* e computadores pode ser uma forma de motivação para a realização de tarefas domésticas, tornando essa prática mais atraente para quem a realiza.

É possível ver que as informações mencionadas aqui apresentam dados que

mostram que o projeto do jogo de realidade alternativa desenvolvido tem um propósito que poderá trazer benefícios sociais e educacionais para esses jovens, como a aquisição de responsabilidade, independência e cooperação.

1.1 MOTIVAÇÃO

Ao vivenciar com crianças e adolescentes em seu meio familiar, a autora deste trabalho percebeu várias vezes que primos, sobrinhos, entre outros estão cada vez mais conectados ao celular, em jogos e redes sociais, e menos participativos em tarefas da casa.

Isto a motivou a desenvolver a aplicação para que pudesse unir esses dois extremos, o uso dos jogos com a necessidade de realização de tarefas, para que um jogo motivasse essas crianças e adolescentes a mudarem o cenário de auxílio nas tarefas da casa.

1.2 PRETENSÕES DE CONTRIBUIÇÃO

O trabalho desenvolvido pretende contribuir na melhora do estímulo humano em crianças e adolescentes no momento da realização de tarefas domésticas. Assim como pretende contribuir para que famílias tenham mais controle e participação dessas tarefas.

1.3 ORGANIZAÇÃO DA MONOGRAFIA

Esta monografia está estruturada da seguinte forma: no segundo Capítulo serão apresentados:

No Capítulo 2.1, a metodologia de pesquisa utilizada para buscas de informações necessárias para conhecimento do tema, assim como das linguagens e ferramentas usadas.

No Capítulo 2.2, será apresentada qual metodologia de desenvolvimento foi usada, assim como quais linguagens, frameworks e ferramentas foram escolhidas para desenvolvimento da aplicação e o porquê de cada uma. Será descrito também alguns conceitos e definições importantes das ferramentas escolhidas para o desenvolvimento da aplicação proposta, para que fique claro para o leitor o que é e para que serve cada linguagem, framework e ferramenta escolhida.

Logo após, no Capítulo 2.3, serão descritos como foi realizado todo o desenvolvimento da aplicação, utilizando as linguagens, frameworks e ferramentas descritas no tópico anterior.

O resultado final do desenvolvimento deste projeto, a aplicação final, telas, usabilidade e configurações, juntamente com a descrição do funcionamento da aplicação serão apresentados no Capítulo 3.

No quarto Capítulo serão apresentados a metodologia de teste usada e os resultados dos testes para validação de funcionamento e de usabilidade do sistema desenvolvido.

Por fim, no Capítulo 5 serão apresentadas as conclusões do trabalho, assim como ideias para trabalhos futuros. Em seguida serão listadas as referências utilizadas na confecção desta pesquisa.

2 METODOLOGIA

Neste Capítulo serão apresentadas a metodologia usada na pesquisa, as ferramentas utilizadas, o processo de desenvolvimento do sistema e o fluxo de páginas da aplicação.

2.1 METODOLOGIA DE PESQUISA BIBLIOGRÁFICA

Este trabalho foi desenvolvido com pesquisa bibliográfica, em artigos publicados em periódicos, livros digitais, artigos online, trabalhos de conclusão de curso e teses. Inicialmente foi feito um levantamento de informações sobre os comportamentos e a evolução de crianças e adolescentes, também foi estudado a realização de tarefas domésticas por elas, seus pontos positivos e negativos.

A pesquisa também foi feita com um levantamento bibliográfico de informações sobre jogos digitais em livros, artigos, trabalhos de conclusão de curso e cursos online, também de como fazer sua utilização e seus diversos tipos de aplicação e pontos favoráveis ou não de cada um deles.

O objetivo era obter um maior entendimento do assunto de jogos digitais, principalmente quais critérios classificam algo a um jogo digital e quais os impactos e necessidades do seu uso, e para entender a necessidade e importância de desenvolver um jogo e aplicar ao público escolhido, podendo assim ver se o desenvolvimento deste trabalho terá resultados positivos na resolução dos problemas abordados neste tema.

Foram pesquisadas também quais plataformas disponíveis e mais acessíveis para o público alvo escolhido e as formas e estruturas de desenvolvimento da aplicação, como linguagens, frameworks, banco de dados, etc.

Estas pesquisas também foram feitas em buscas pela internet, cursos online e opiniões de professores e profissionais especializados em desenvolvimento de aplicações. Para escolha foram analisados os requisitos como desempenho, acessibilidade, facilidade de aplicação, facilidade de desenvolvimento, memória necessária para uso, compatibilidade das ferramentas, entre outros. Dessa forma foram escolhidas as plataformas, ferramentas e modelo de desenvolvimento, descritos no tópico seguinte.

2.2 FERRAMENTAS ESCOLHIDAS E SUAS DEFINIÇÕES

O sistema foi desenvolvido para plataformas *web*, para que tenha um acesso mais fácil e prático ao jogo pelos usuários. Dessa forma o sistema pode ser usado tanto em plataformas *desktops* quanto em *mobiles*, através de um navegador web.

A linguagem de programação utilizada foi a linguagem Javascript, por ser uma linguagem bastante versátil, também é “atualmente a principal linguagem de programação *cliente-side* na *web*” (DIAS, J; VIEIRA, R; SILVA, R, 2013). Ela possibilita a utilização da linguagem tanto do lado do cliente como do lado do servidor.

A linguagem Javascript foi escolhida por também possibilitar o desenvolvimento de aplicações web, assim como também aplicações mobile com o auxílio de frameworks como o Express, dessa forma é possível desenvolver a aplicação para duas plataformas diferentes com um único código e uma mesma linguagem.

Para utilização da linguagem Javascript do lado do servidor será utilizado o interpretador de código Node.js, que apesar de ser uma tecnologia desenvolvida no ano de 2009, “está sendo utilizado na construção de muitas aplicações Web” (BONFIM; LIANG, 2014).

Segundo Schroeder e Santos (2014) o desempenho do Node.js é melhor e mais ágil em relação a outras linguagens e frameworks. Para utilização junto com o Node.js será usado o framework Express.js, por possibilitar criar aplicações web do Node.js de modo simples e flexível e fornecendo um conjunto robusto de recursos para aplicativos web e móvel (EXPRESS, 2018).

A engine de view EJS foi outro framework escolhido para ser usado no desenvolvimento da aplicação por possibilitar o transporte de dados do *back-end* para o *front-end*, como por exemplo, um nome de usuário que foi buscado no banco de dados (*back-end*) e é mostrado ao usuário para visualização dessa informação em sua tela (*front-end*).

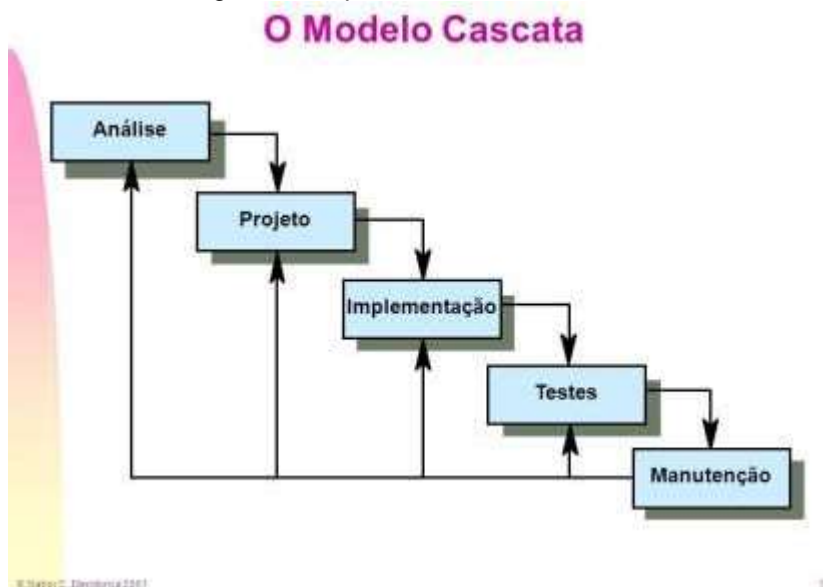
O banco de dados utilizado foi o orientado a documentos MongoDB, por ser um modelo de bancos não relacional, que possibilita menos complexidade de uso. Foi escolhido também por ser um banco que tem um ótimo desempenho (POLITOWSKI; MARAN, 2014).

2.2.1 Definição do modelo de desenvolvimento escolhido

Com o objetivo de aumentar a produtividade e diminuir os riscos de erros no desenvolvimento do sistema, foram utilizadas técnicas e conhecimentos de Engenharia de Software. O modelo de desenvolvimento do sistema que foi usado é o modelo Cascata, por ser um modelo que possui elementos sequenciais, como mostra a Figura 1, onde esses elementos sequenciais orientam o desenvolvimento, permitindo que todas as funções sejam estruturadas corretamente, desde o estudo de viabilidade para desenvolvê-los até a implementação.

Nesse modelo, o processo de desenvolvimento deve ser conduzido de forma disciplinada, com atividades claramente definidas, determinada a partir de um planejamento e sujeitas a gerenciamento durante a realização (LEITE, J. 2000).

Figura 1: Sequência do modelo Cascata



Fonte: Nabor e Mendonça (2001).

2.2.1.1 Análise e definição dos requisitos

Nesta etapa, foram estabelecidos os requisitos da aplicação desenvolvida, o que consiste nos serviços que devem ser fornecidos, limitações e objetivos do software. Esta etapa foi feita seguindo as necessidades de funções necessárias percebidas no cenário familiar e de amigos da autora, como o controle de horários e a lista de tarefas que devem ser realizadas.

Assim que estabelecidos, os requisitos devem ser definidos de forma

condizente com os objetivos propostos para que sejam úteis na etapa seguinte. Esta etapa inclui também a documentação e o estudo da facilidade e da viabilidade do projeto para determinar o processo de início de desenvolvimento do projeto do sistema; pode ser vista como uma concepção de um produto de software e também como o início do seu ciclo de vida.

2.2.1.2 Projeto do sistema

O projeto do sistema “Comandando em Casa” é um processo de vários passos que se centraliza em quatro atributos diferentes do sistema: estrutura de dados, arquitetura do software, detalhes procedais e caracterização das interfaces. O processo de projeto representa os requisitos de uma forma que permita a codificação do produto (é uma prévia etapa de codificação). Da mesma maneira que a análise dos requisitos, o projeto é documentado e transforma-se em uma parte do software.

2.2.1.3 Implementação

Esta é a etapa em que é criada a aplicação, onde é feita a codificação do sistema. Como o projeto possui um nível de detalhe elevado, a etapa de codificação pode ser implementada automaticamente, utilizando as linguagens e frameworks escolhidos durante o processo de projeto do sistema. A princípio, foi incluído um teste unitário dos módulos nesta etapa; nesse caso, as unidades de código produzidas foram testadas individualmente antes de passar a etapa de integração e teste global. Como será descrito no tópico de testes deste trabalho.

2.2.2 Conceitos e definições de linguagens, *frameworks* e ferramentas utilizadas

Aqui serão abordados alguns conceitos e definições de linguagens, frameworks, ferramentas, banco de dados e métodos utilizados no sistema que são relevantes e necessários para o entendimento do sistema proposto.

2.2.2.1 Node.js

Node.js é um interpretador de código da linguagem de programação

JavaScript que tem o seu código aberto, a função do interpretador Node.js é mudar o Javascript do lado do cliente para o lado dos servidores. Tem como objetivo principal ajudar programadores no desenvolvimento de aplicações de alta escalabilidade (como um servidor web), com códigos capazes de manipular dezenas de milhares de conexões simultâneas, numa única máquina física.

Essa tecnologia possui um modelo inovador, que é sua arquitetura, que é totalmente *non-blocking thread* (não bloqueante), ou seja, essa tecnologia usa ao máximo e de forma eficiente o poder de processamento dos servidores, principalmente em sistemas que produzem uma alta carga de processamento. Sistemas que utilizam o Node.js não sofrem de *dead-locks*, porque ele trabalha apenas em *single-thread* (única *thread* por processo). Desenvolver sistemas nesse paradigma é simples e prático.

Um estudo de testes de serviços web de alto desempenho feito por Ricardo Schroeder e Fernando dos Santos (2014) que gerou um artigo intitulado “Arquitetura e testes de serviços web de alto desempenho com Node.js e MongoDB” em 2014, mostram que a dupla Node.JS e o bando de dados não relacional MongoDB realmente é um servidor de alto desempenho, tendo uma performance em alguns aspectos seis vezes superior quando comparado as outras arquiteturas abordadas no testes, como o PHP/Apache, Node.js/PostgreSQL, Java/Netty/PostgreSQL, entre outras.

O teste de desempenho tem o objetivo duplo de entender como o sistema responde a carregamento (isto é, número de usuários, número de transações ou volume global de dados), e coletar métricas que vão levar a modificações de projeto. Este teste levou em consideração os seguintes aspectos: consumo de memória, consumo de CPU (*Central Process Unit*), requisições por tempo, número de requisições, tempo médio de resposta e vazão do servidor.

2.2.2.2 Express.js

O Express.js é um framework que visa facilitar o desenvolvimento de aplicações Web com o Node.js. O Express.js permite criar servidores web e receber requisições HTTP de maneira simples (EXPRESS, 2018). Ele também permite a criação de um conjunto de diretórios com uma estrutura padrão, além de organizar as rotas dos arquivos para as *views* da aplicação, como foi feito no desenvolvimento

do presente projeto. Geralmente os projetos que utilizam-se do Express também aderem a algum framework de templates como Jade ou EJS (*Embedded JavaScript*). No caso da aplicação de realidade alternativa que foi desenvolvida, foi utilizado o framework EJS, em conjunto com o Express.js.

2.2.2.3 EJS

O nome EJS vem do termo *Embedded JavaScript*, que significa Embutir Javascript, e é uma linguagem de templates, que é usada para gerar página HTML e XML do lado do cliente que era originalmente parte do JavaScriptMVC, que agora foi substituído pelo DoneJS. O EJS é considerado uma *engine de view*, ou seja, um mecanismo de vistas para transportar dados do *back-end* para o *front-end*. Basicamente ele faz a transferência de dados como por exemplo dados recuperados do banco de dados para uma página de visualização HTML, do lado do cliente, para que esses dados recuperados sejam colocados de forma visível para o usuário da aplicação.

2.2.2.4 JQuery

O jQuery é uma biblioteca JavaScript rápida, pequena e rica em recursos. Ele torna as coisas como passagem e manipulação de documentos HTML, manipulação de eventos, animação muito mais simples, com uma API fácil de usar que funciona em vários navegadores (JQUERY, 2018). Com uma combinação de versatilidade e extensibilidade, essa ferramenta mudou a maneira como milhões de pessoas escrevem JavaScript.

2.2.2.5 Bootstrap

Para ajudar no desenvolvimento da aplicação, de forma que ela se tornasse uma aplicação mais acessível de acordo com a plataforma que o usuário estaria utilizando com um único código, foi utilizado o framework Bootstrap. Desenvolvido pela equipe do Twitter, o Bootstrap é um framework *opensource* compatível com HTML5 e CSS3 que foi criado para auxiliar no desenvolvimento de web sites responsivos (vide Figura 2).

Este framework é bastante utilizado no desenvolvimento front-end e foi escolhido para ser usado na aplicação por possibilitar a responsividade da mesma,

ou seja, cada usuário terá uma visão e uma acessibilidade de funções da aplicação diferentes, conforme seu dispositivo de acesso, sem que precise ser desenvolvido um código diferente para cada um.

Dessa forma, pôde-se economizar bastante tempo no processo de desenvolvimento. Uma coisa importante para ter em mente são as diferenças na usabilidade entre dispositivos móveis e Desktops.

Um dos grandes defensores da usabilidade, Jakob Nielsen, criador de dez heurísticas indispensáveis para quem quer projetar uma boa interface e criar uma ótima experiência de uso, teve grande influência no trabalho de Lopes (2013), que afirma:

As diferenças são tão brutais que precisamos de design diferente para atacar esses públicos. Isso pode ser feito de várias maneiras: sites diferentes para mobile e Desktop; servidor otimizando a página; ou design responsivo e adaptação do design no cliente.

O Bootstrap foi usado também por ter uma “documentação simples e bastante ampla, ter inúmeros componentes à disposição, por conseguir manter um padrão de layout e por funcionar em diversos navegadores atuais, como o Google Chrome, Opera, Safari, Firefox, entre outros”. (COSTA, F., 2014).

Figura 2: Site responsivo em diversas plataformas



Fonte: Cammino (2018).

2.2.2.6 Banco de dados NoSQL

O termo NoSQL foi primeiramente utilizado em 1998 como o nome de um banco de dados não relacional de código aberto. O autor do termo, Carlo Strozzi,

alega que o movimento NoSQL "é completamente distinto do modelo relacional e, portanto, deveria ser mais apropriadamente chamado "NoREL" ou algo que produzisse o mesmo efeito" (DEV MEDIA, 2018b).

O NoSQL surgiu no momento em que a grande quantidade de dados gerados se tornou um problema, principalmente, quando era em um espaço de tempo (relativamente) curto, isso se torna cada vez mais presente e real com a crescente popularização da internet com diversos novos dados surgindo.

Assim sendo, trata-los foi se tornando cada vez mais complexo e sua manutenção cada vez mais cara. Como consequência dessa grande quantidade de dados, sistemas desenvolvidos para a manipulação desses dados gerados necessitavam de um grande poder de processamento de forma eficiente e escalável.

Além da alta taxa de geração dos dados, outro fator que influenciou a criação de sistemas NoSQL foi o suporte a tipo de dados complexos, semiestruturados ou não estruturados. Além do volume de geração desses dados ser grande, outro fator predominante é relacionada a dificuldade de modelagem de tais tipos de dados. Esses tipos de dados estão hoje presentes em inúmeros domínios de aplicações, tais como Web 2.0, redes sociais, redes de sensores, entre outros (VIEIRA et al, 2012).

O NoSQL veio para suprir essas novas necessidades que o mundo digital trazia, já que bancos de dados relacionais escalam, mas quanto maior o tamanho, mais caro se torna essa escalabilidade, pode ser pelo custo de novas máquinas, ou pelo aumento de especialistas nos bancos de dados utilizados. Já bancos de dados NoSQL que não são relacionais, permitem uma escalabilidade com menor custo e menos trabalho, pois não exigem máquinas extremamente poderosas e sua facilidade de manutenção permite que um número menor de profissionais seja necessário.

Os bancos de dados NoSQL permitem outros subtipos de bancos que se diferenciam quanto ao modelo de dados utilizados, como os bancos de dados que trabalham no esquema chave/valor, os bancos baseados em coluna, baseados em grafos e os bancos de dados baseado em documentos, que é o tipo que foi utilizado para desenvolver a aplicação do presente projeto.

2.2.2.7 Banco de Dados MongoDB

O MongoDB é um banco de dados NoSQL e orientado a documentos, possui

código aberto e é multiplataforma, este banco foi escrito na linguagem C++. O MongoDB usa documentos semelhantes a JSON com esquemas, que por sua vez, suporta estruturas como *arrays* e *embedded objects* assim como JSON (vide Figura 3).

Este banco permite que usuários realizem modificações de apenas um atributo em um documento, sem a necessidade de interação com o restante da estrutura. No MongoDB, documentos podem ser armazenados no que são chamados *collections* que são coleções de documentos (vide Figura 4), onde serão efetuadas operações de busca (*queries*) e indexações (*indexing*). *Queries* são expressadas na sintaxe JSON e enviadas ao MongoDB como objetos BSON (*Binary JSON*) pelo driver de conexão ao banco.

No tópico referente ao interpretador de códigos Node.js, foi mencionado um teste de desempenho que mostrou que junto com o Node.js o MongoDB demonstrava ter um alto desempenho, melhor que outras 'duplas' de serviços em vários aspectos.

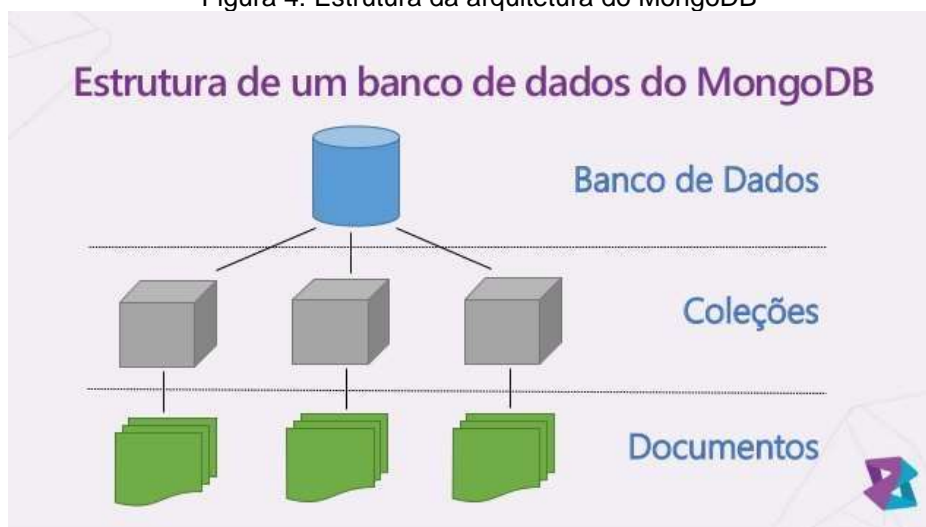
Em outro estudo de comparação de desempenho, feito por Cristiano Politowski e Vinícius Maran (2014), publicado logo depois como um artigo que foi intitulado "Comparação de Performance entre PostgreSQL e MongoDB", mostra que utilizando os dois bancos de dados na configuração padrão, sem ajustes de otimização e baseado no ambiente de testes proposto, foi possível concluir que o MongoDB obteve melhores resultados.

Figura 3: Estrutura de dados do banco MongoDB



Fonte: Elaborado pela autora.

Figura 4: Estrutura da arquitetura do MongoDB



Fonte: NetCoders (2018).

2.2.2.8 Linguagem de modelagem unificada (UML)

UML (*Unified Modeling Language*), ou seja, linguagem de modelagem unificada, é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de orientação a objetos. De acordo com Guedes (2004), nos últimos anos essa linguagem tornou-se padrão de modelagem de software adotada internacionalmente pela indústria de Engenharia de Software.

Vale ressaltar que a UML não é uma linguagem de programação, mas sim uma linguagem de modelagem cujo objetivo é auxiliar os engenheiros de software a definir as características destes, tais como seus requisitos, sua estrutura lógica, seu comportamento, a dinâmica de seus processos e até mesmo as necessidades físicas em relação ao equipamento sobre o qual o sistema deverá ser implantado. Ou seja, UML ajuda os engenheiros na visualização do projeto através de diagramas que mostram a comunicação entre os objetos.

Existem diversos diagramas na linguagem UML, onde é possível ter diferentes visões do sistema modelado. Cada diagrama analisa o sistema sob uma determinada ótica, pois cada um enfoca uma determinada característica. Além disso, “a utilização desses diagramas permite que falhas sejam descobertas mais rapidamente, diminuindo assim a possibilidade da ocorrência de erros futuros” (GUEDES, 2004). Os diagramas escolhidos para esse projeto foram: Diagrama de Caso de Uso e Diagrama de Atividades.

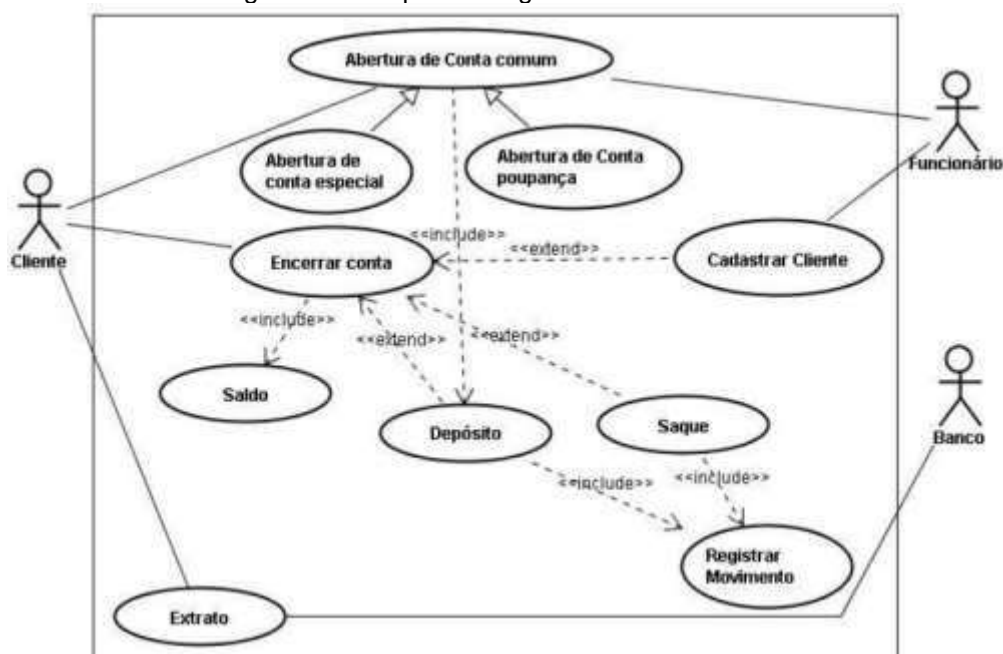
2.2.2.9 Diagrama de casos de uso da UML

O Diagrama de casos de uso tem o objetivo de auxiliar a comunicação entre os analistas e o cliente. O cliente deve ver no diagrama de Casos de Uso as principais funcionalidades de seu sistema.

O diagrama de casos de uso é representado por atores e casos de uso e relacionamentos entre estes elementos. Estes relacionamentos podem ser associações entre atores e casos de uso, generalizações entre os atores, generalizações, *extends* e *includes* entre os casos de uso (GUEDES, 2004).

Um relacionamento include de um caso de uso A para um caso de uso B indica que B é essencial para o comportamento de A. Um relacionamento extend de um caso de uso B para um caso de uso A indica que o caso de uso B pode ser acrescentado para descrever o comportamento de A (não é essencial). A extensão é inserida em um ponto de extensão do caso de uso A (TONIN; CITTOLIN; SOUZA; 2015). Veja na Figura 5, um exemplo de um diagrama de casos de uso.

Figura 5: Exemplo de diagrama de casos de uso.



Fonte: Elaborado pela autora.

2.2.2.10 Diagrama de Atividades da UML

O diagrama de atividades é um dos diagramas definidos pela linguagem UML,

ele é utilizado para modelar o aspecto comportamental de processos e representa os fluxos conduzidos por processamentos. É basicamente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra. Comumente isso envolve a modelagem das etapas sequenciais em um processo computacional, como por exemplo, o processo de autenticação de um usuário de um sistema, ou o processo de um cadastro em um aplicativo, depende muito da necessidade do sistema.

2.3 PROCESSO DE DESENVOLVIMENTO

Seguindo o roteiro do modelo de processo de desenvolvimento de sistemas escolhido que foi o modelo Cascata, para aplicar ao desenvolvimento dessa aplicação, foram descritos primeiramente os requisitos do sistema. Os requisitos foram descritos da seguinte maneira:

2.3.1 Requisitos de interface de usuário

A interface com o usuário foi implementada em HTML e CSS, para que seja facilmente acessada através de qualquer *browser*. O projeto possui duas visões diferentes de interface com o usuário definidas: administrador da casa e participante da casa.

2.3.2 Requisitos de interface de hardware

A interface de hardware se dará através do uso do teclado e do mouse quando o sistema for utilizado em um computador. Quando o sistema for utilizado por plataformas *mobile*, a interface de *hardware* a ser utilizada deverá ser o *touchscreen* do aparelho.

2.3.3 Requisitos de interface de software

A interface do software será desenvolvida em linguagem Javascript, que também servirá para fazer a comunicação entre a interface com o usuário e o servidor onde estará o banco de dados. Qualquer modificação feita através da interface gráfica será enviada ao banco de dados com ajuda de um framework Express.jsl.

2.3.4 Requisitos de interface de comunicação

Toda a interface de comunicação do sistema será baseada na Web. Um servidor Web estará rodando no servidor principal e será a interface de comunicação entre os terminais e o banco de dados central.

Como todo o processamento estará centralizado no servidor, as restrições de memória do sistema estarão todas ligadas à limitação de memória do servidor central. Os terminais apenas acessarão esse servidor e, portanto, só precisam apresentar uma configuração básica. O servidor deverá ter uma memória suficiente para armazenar os dados do sistema e tratar um número de requisições correspondente ao necessário para o bom funcionamento do sistema proposto.

2.3.5 Requisitos do modo de operação

A operabilidade do sistema estará dividida em dois modos de acesso, restritos pela autenticação inicial do usuário. Tais modos de operação são:

- **Modo administrador:** O administrador terá acesso total ao sistema. Portanto, estará apto a cadastrar, remover, ver ou atualizar informações dos demais usuários; cadastrar, remover, ver ou alterar informações das tarefas para realização; bem como autenticar ou não se as tarefas foram realmente realizadas; trocar pontos dos demais participantes. O usuário administrador também estará apto a participar da realização das tarefas da casa e por isso terá acesso, a escolher e cadastrar tarefas realizadas e participar do ranking dos usuários, assim como ver o histórico de operações do jogo. Para acessar funções específicas do modo administrador, o usuário deverá utilizar a *login* e senha informados ao fazer o cadastro na aplicação;

- **Modo participante:** Os usuários do modo participante simplesmente têm acesso a ver as tarefas disponíveis para realizar, assim como cadastrar uma tarefa realizada pelo mesmo, também têm acesso ver quais são os usuários cadastrados no jogo, assim como o ranking dos participantes, suas informações pessoais de perfil e linha do tempo de operações do jogo.

2.3.6 Requisitos de armazenamento

O sistema deve ser capaz de armazenar dados de usuários, tarefas realizadas durante o jogo. Portanto, as rotinas de acesso ao Banco de Dados estão

relacionadas às seguintes situações:

- O Banco de Dados deve ser capaz de cadastrar, remover, buscar ou atualizar informações referentes aos usuários do sistema;
- Também deve ser capaz de cadastrar, remover, buscar e atualizar informações relativas às tarefas;
- Também deve ser capaz de cadastrar, buscar e atualizar informações relativas as operações realizadas durante o jogo;

2.3.7 Requisitos de acesso

Apenas usuários cadastrados no sistema e participantes cadastrados pelo administrador terão acesso ao sistema. Antes de acessar qualquer página, o usuário será obrigado a autenticar-se através de seu *login* e senha.

2.3.8 Requisitos de interface gráfica

O sistema deve apresentar uma interface gráfica amigável, descritiva e visualmente desenhada para atrair e possibilitar o uso por crianças e usuários leigos em jogos digitais e tecnologia.

2.3.9 Requisitos específicos

- **Cadastrar usuário:** O cadastro de usuário administrador deve ser feito fora da área restrita de usuários, e pede um nome, um email válido, um *login* e uma senha, de no mínimo 5 (cinco) caracteres. Esse cadastro serve para dar aos usuários administrador acesso às informações do sistema, bem como a permissão para efetuar alterações. O cadastro de usuário participante deve ser feito dentro da área restrita de usuário administrador, por um administrador já autenticado no sistema, e pede somente um nome, um *login* e uma senha, de no mínimo 5 (cinco) caracteres.
- **Remover usuário:** Apenas o usuário do tipo administrador poderá remover os usuários participantes, ele deverá está autenticado dentro do sistema para ter acesso a essa função.
- **Cadastrar tarefas disponíveis:** Cadastrar uma tarefa é uma opção que está disponível apenas para o usuário do tipo administrador, e ele precisará

fornecer informações como nome da tarefa, descrição, imagem que define a tarefa e os pontos que serão atrelados a essa tarefa. Além do mais, haverá cadastro da data, horário e *login* do usuário que efetuou o cadastro da tarefa, porém esses dados não serão fornecidos pelo usuário, o sistema irá fazer esse cadastro automaticamente quando o cadastro for finalizado pelo usuário administrador.

- **Remover tarefa:** Remover uma tarefa que foi cadastrada no banco de dados é uma opção que está disponível apenas para o usuário do tipo administrador, e ele só precisará escolher qual tarefa deseja remover na opção de visualização de tarefas, sejam elas disponíveis para realizar ou tarefas pendentes.

2.3.10 Requisitos de operações

O sistema deverá fazer o cadastro no banco de dados automaticamente de algumas informações de operações realizadas pelos usuários no sistema, para que depois essas informações sejam recuperadas do banco de dados e fiquem visíveis aos demais usuários cadastrados, para que todos os usuários acompanhem em tempo real o que está acontecendo no jogo.

2.3.11 Requisitos gerais

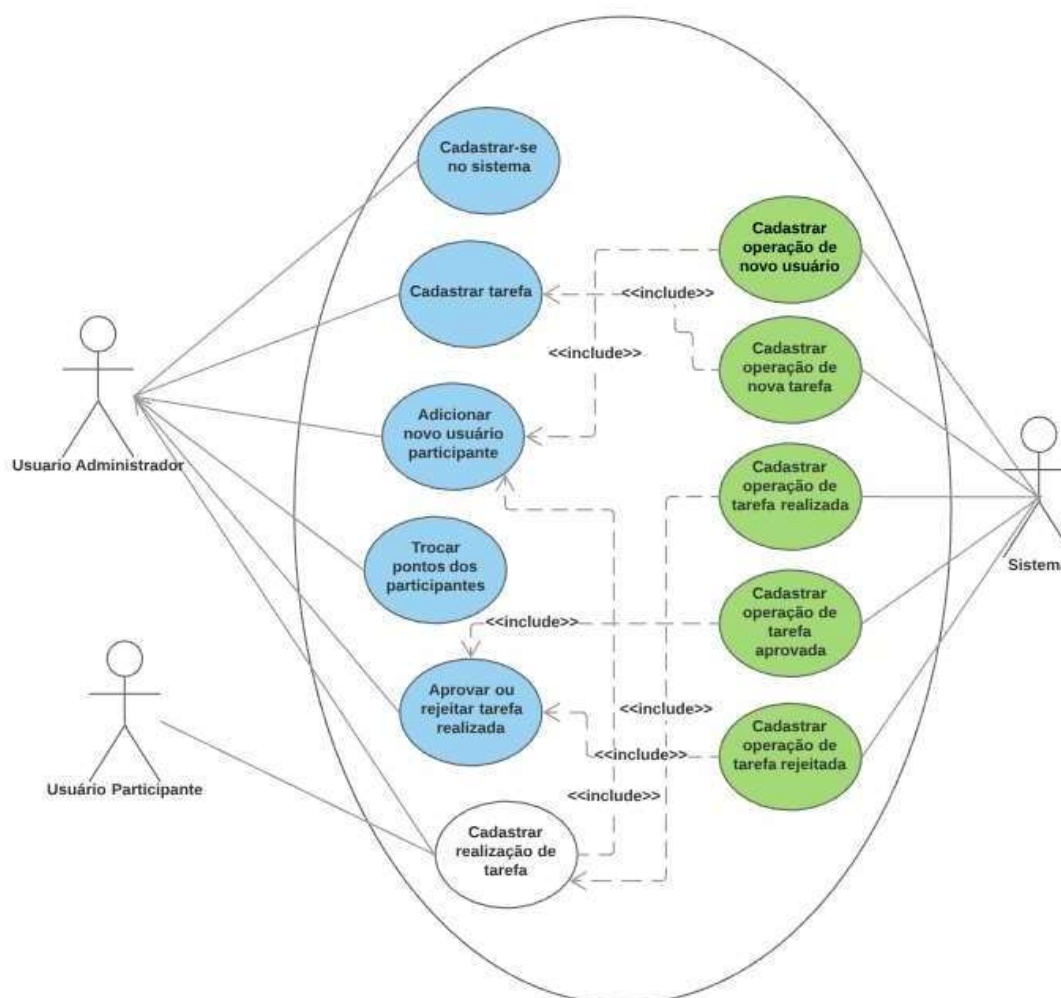
Alguns outros requisitos são definidos de forma geral ao jogo e podem ser descritos da seguinte maneira:

- O jogo dá a possibilidade de ter apenas um usuário do tipo administrador ligado a uma equipe de participantes. Dessa forma a equipe é formada através dele.
- Cada usuário está associado a um único administrador, que é o que fez seu cadastro, ou no caso de o usuário ser do tipo administrador, a ele mesmo. Dessa forma um grupo de casa será formado automaticamente.
- Todos os usuários só terão acesso a informações da sua equipe, como na linha do tempo de operações do jogo, participantes da casa, ranking da casa, etc.

2.3.12 Casos de uso

Para visualizar melhor o papel de cada ator, ou seja, a função que cada tipo de usuário pode ter acesso no sistema do jogo desenvolvido foi criado um diagrama de caso de uso, que demonstra em um diagrama como ficaria alguns requisitos descritos nos tópicos anteriores, veja o resultado desse diagrama na Figura 6.

Figura 6: Diagrama de Caso de Uso da aplicação
BASIC USE CASE DIAGRAM



Fonte: Elaborado pela autora.

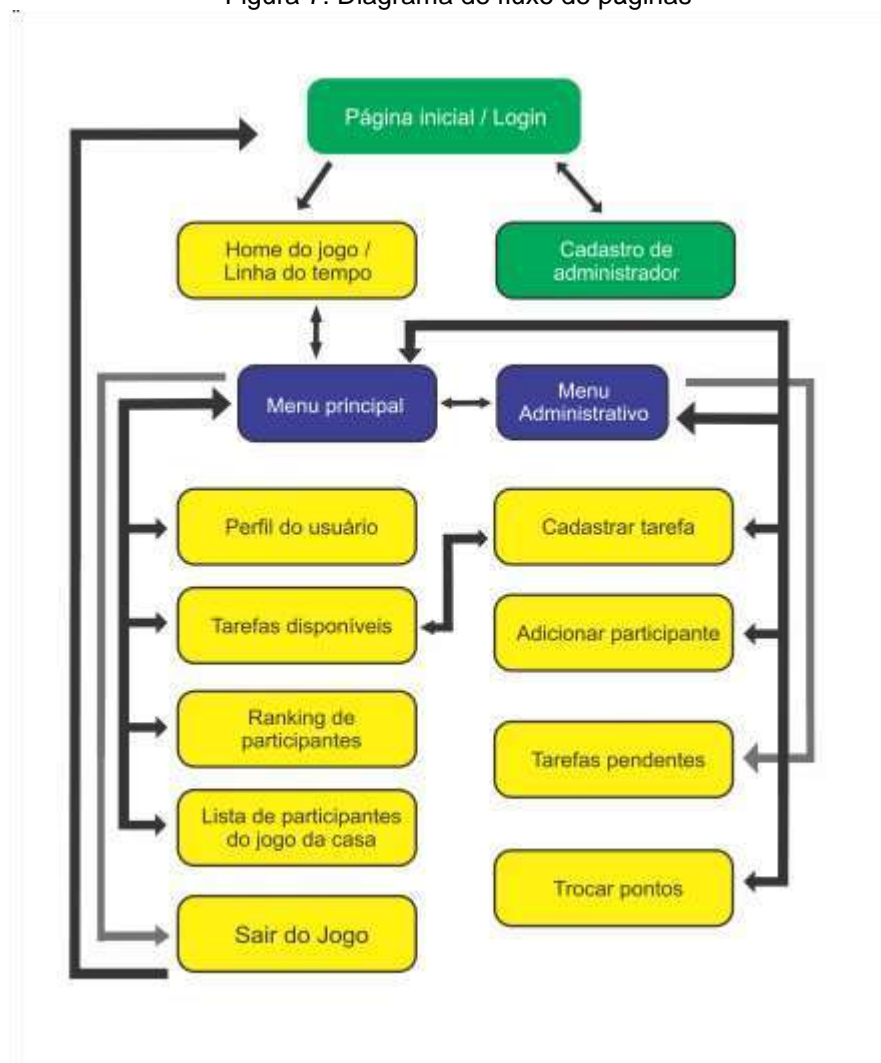
Na Figura 6, os atores são os usuários, do tipo administrador e do tipo participante, e o sistema, que também realiza funções de acordo com outras funções realizadas. Todas as funções que são desempenhadas pelo sistema necessitam obrigatoriamente da realização de outras funções para serem realizadas, e são descritas no diagrama pela propriedade “<<include>>” da linguagem UML.

2.4 FLUXO DE PÁGINAS DA APLICAÇÃO

A aplicação foi dividida em diversas páginas para mostrar aos usuários informações com finalidades diferentes. Essas páginas de visualizações estão ligadas umas a outras, onde cada página poderá encaminhar o usuário para uma ou diversas outras páginas. A página principal da aplicação é o menu principal, onde o usuário tem a opção de escolher qual página quer ver, ou que tipo de informação ele

quer ter acesso e a qual função da aplicação ele quer executar. Para demonstrar melhor o fluxo de comunicação dessas páginas entre si, foi desenhado um diagrama de fluxo de páginas, exibido na Figura 7, onde cada página redireciona a visualização do usuário para outra, conforme solicitado pelo usuário do sistema.

Figura 7: Diagrama de fluxo de páginas



Fonte: Elaborado pela autora.

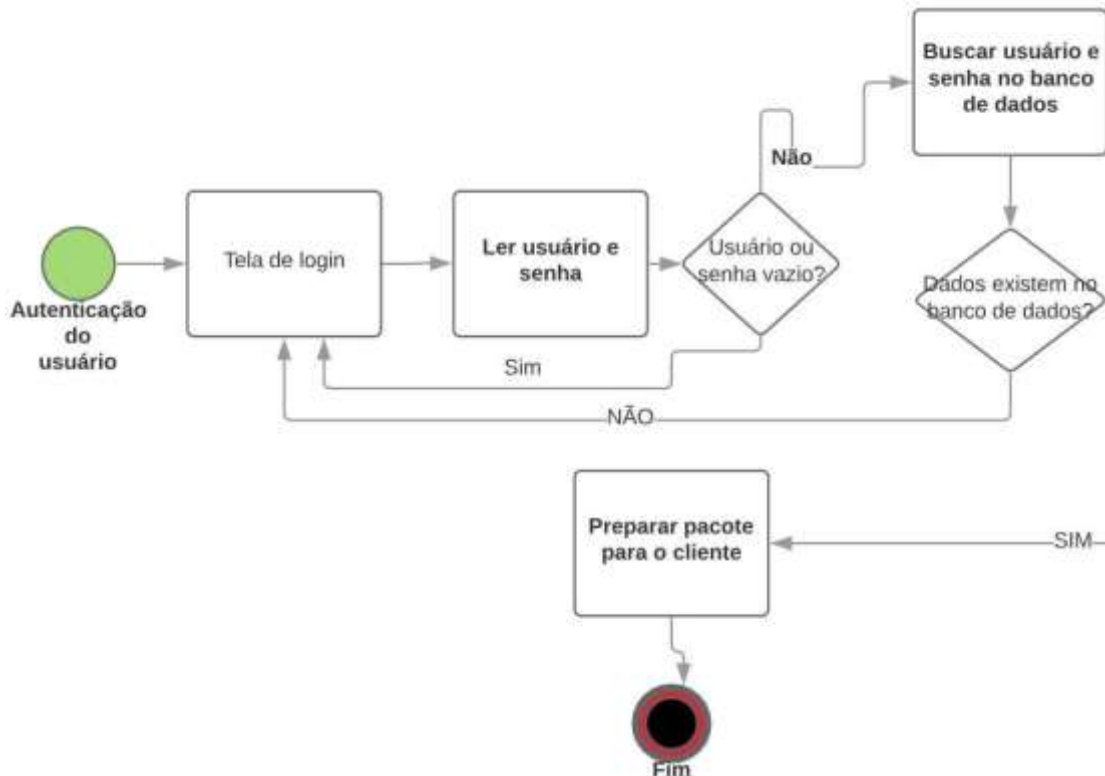
2.4.1 Autenticação do usuário

Para que o sistema fosse restrito apenas a usuários cadastrados no mesmo, foi implementado um método de verificação de informações que são essenciais para que o usuário seja identificado no sistema, informações que são o usuário e a senha, definidos no momento do cadastro do usuário.

Dessa forma foi implementada e seguida uma sequência lógica de verificação dessas informações. Para entender melhor essa sequência e como ela

funciona foi desenhado um diagrama de atividades (utilizado na linguagem UML) desse processo. Veja na Figura 8.

Figura 8: Diagrama de atividades da autenticação de usuário.



Fonte: Elaborado pela autora.

Para garantir que somente usuários que foram autenticados corretamente, ou seja que tenham efetuado *login* com nome de usuário e senhas cadastradas na aplicação, tenham acesso ao sistema foi implementado um método de Sessão Web. Em um site ou sistema web, a sessão é importante quando se quer mais segurança na página ou quando se quer ter um controle de usuário.

Alguns programadores também utilizam-se deste recurso para guardar informações da navegação em uma aplicação web do usuário que está utilizando o sistema, como os itens de um carrinho de compras de um site de vendas online. As variáveis de sessão armazenam informações sobre um único usuário e estão disponíveis para todas as páginas em um único aplicativo.

Para utilizar o método de criação de sessões na aplicação desenvolvida nesse trabalho, foi utilizado o módulo *Session* presente no framework Express. Ele utiliza toda a lógica de criação de sessão, que foi demonstrado no parágrafo anterior, para ser utilizado em conjunto com o interpretador Node.js.

No desenvolvimento da aplicação, a criação da variável de sessão do usuário

é feita assim que o usuário é autenticado no site. Seguindo o modelo da Figura 8, quando o usuário e senha que foram inseridas são encontradas no banco de dados da aplicação, é criada uma variável no navegador do usuário, que define que ele está liberado para acessar páginas protegidas por sessões. Nesta aplicação, com exceção da página de *login* e cadastro, todas as outras páginas foram protegidas para apenas usuários que tenham uma sessão aberta, ou seja, liberada, possam acessar.

2.4.2 Segurança de dados confidenciais

Visando proteger dados cadastrais dos usuários, como as senhas, para que somente cada usuário tenha acesso a sua informação, foi utilizado a criptografia de dados, que diz respeito a conceitos e técnicas usadas para codificar uma informação, de forma que somente seu destinatário e o emissor da mensagem possam acessá-la, com o objetivo de que terceiros, que não podem ter acesso a essas informações não interceptem e nem entendam a mensagem, nesse caso a senha.

Atualmente, as técnicas de criptografia mais utilizadas envolvem o conceito das chaves criptográficas, que são um conjunto de bits, baseados em um algoritmo capaz de interpretar a informação, ou seja, capaz de codificar e decodificar essa informação. Neste caso se a chave do receptor não for compatível com a do emissor, a informação então não será extraída para a original.

Existem termos que são 'chave de 64 bits' e 'chave de 128 bits' e são usados para expressar o tamanho da chave, assim, quanto mais bits forem utilizados, mais segura será essa criptografia, e essa quantidade de bits está diretamente relacionada a quantidade de chaves que poderão ser geradas para decodificar essa informação, ou seja a quantidade de tentativas de gerar chaves diferentes para decodificar uma informação.

Para garantir a criptografia das senhas dos usuários cadastrados no sistema, foi utilizado o módulo Crypto do Node.js, que contém diversos algoritmos de criptografia, mas que para o presente trabalho, foi utilizado o algoritmo MD5, que gera uma hash de 32 posições, ou seja, independente do tamanho da senha cadastrada, o algoritmo irá gerar uma chave totalmente diferente com 32 *caracteres*.

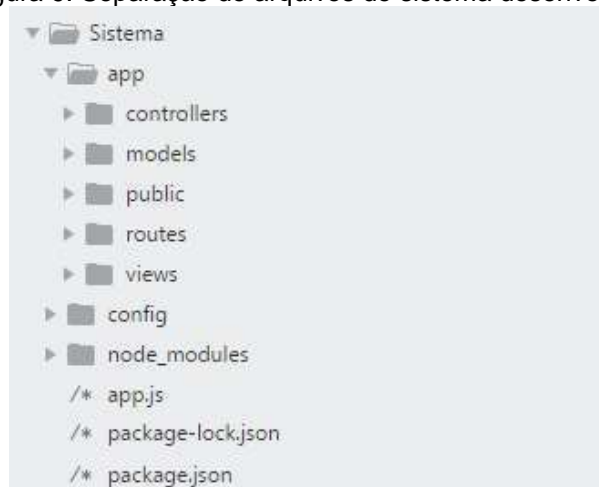
2.4.3 Arquitetura e organização do desenvolvimento

No desenvolvimento do sistema, todos os arquivos foram organizados de forma a ajudar a manutenção e desenvolvimento da aplicação. Para ajudar na organização desses arquivos foi utilizado um padrão de arquitetura de software chamado MVC, que é uma abreviação das palavras *model*, *view* e *controller*, que em português significam respectivamente, modelo, visão e controlador. Esse é um padrão de arquitetura de software onde é separada a representação da informação da interação do usuário com o software.

São separados componentes maiores possibilitando a reutilização de código e desenvolvimento paralelo de maneira eficiente. Assim alterações feitas, por exemplo, no *layout* não afetam o controle de dados, e estes também poderão ser alterados sem que precise mudar ou mexer no visual da aplicação.

Esse modelo de padrão de arquitetura define que sejam separados arquivos de visualização, de modelagem e arquivos que façam o controle de informações. Sendo assim a aplicação teve toda codificação separada adequadamente para realizar tipos diferentes de funcionalidades, veja a Figura 9. Foram separados também os direcionamentos de rotas do sistema, cada rota sendo encaminhada a um determinado controlador que fazia o tratamento adequado de informações passadas por essas rotas e a chamada da *view* de cada página para mostrar ao usuário.

Figura 9: Separação de arquivos do sistema desenvolvido.



Fonte: Elaborado pela autora.

No caso do desenvolvimento dessa aplicação, todos estes arquivos de

visualizações, de controle e de modelagem de dados ficaram separados, mas conectados uns aos outros para que toda comunicação entre eles pudesse acontecer.

Um exemplo dessa comunicação é a autenticação do usuário, descrita nos tópicos anteriores e desenhado na Figura 8, onde a tela de *login* é um arquivo de visualização que envia informações para um arquivo de controle dessa visualização que faz a verificação da existência e consistência dos dados que foram recebidos e caso tudo esteja correto, esse controlador faz a comunicação com um arquivo de modelagem de dados para fazer uma busca no banco de dados com as informações inicialmente passadas pela *view* para o controlador.

3 RESULTADOS E APLICAÇÃO FINAL

A ideia do jogo “Comandando em Casa” é que as crianças ganhem pontos ao realizarem tarefas domésticas pré-definidas pelos administradores da casa. Dessa forma esses pontos podem ser trocados por algo que seja combinado com o administrador da casa, fora do jogo. Ou seja, por dinheiro, uma saída ao shopping, etc.

Esses pontos são cadastrados no momento em que o administrador cadastra uma tarefa a ser feita. E os participantes ganham esses pontos quando realizam as tarefas e essas tarefas são aprovadas pelo administrador do grupo (administrador da casa e do jogo).

Depois de todo processo seguido pelo modelo de desenvolvimento Cascata, como definição de requisitos, análise de requisitos, definição de métodos e ferramentas utilizadas, codificação e testes do sistema, a aplicação está pronta para ser utilizada e atender a funcionalidades descritas inicialmente.

No desenvolvimento da aplicação, toda interface gráfica, desenvolvida em html e css foi feita utilizando elementos visuais como imagens, ícones e cores diversas para atrair a atenção dos usuários público alvo, como crianças e adolescentes e referenciar informações do sistema.

Na sequência estão algumas figuras e descrições de como ficou o resultado final deste projeto, nas plataformas *desktops* e *mobile*, respectivamente.

Figura 10: Tela inicial na plataforma Desktop - Login de acesso.



Fonte: Elaborado pela autora.

Como foi descrito no requisito de modo de operação, a aplicação é restrita apenas a usuários cadastrados, com isso a tela inicial apresentada na Figura 10 é utilizada para apresentar as opções de acesso aos usuários, sejam eles cadastrados ou não. Esta tela oferece a opção de *login* para usuários já cadastrados e o link de redirecionamento para página de cadastro para usuários que ainda não estão cadastrados no sistema, neste caso o responsável pelo grupo que vai jogar é quem precisa se cadastrar por esse meio.

Figura 11: Tela de Cadastro de Administrador na plataforma Desktop



Fonte: Elaborado pela autora.

Na Figura 11 está a tela de cadastro de usuário inicial, que é o usuário que será responsável por criar e administrar o grupo para jogar. Nesta tela, para se

cadastrar o usuário precisa preencher os campos obrigatórios para ser um usuário do sistema.

Figura 12: Tela Home da Aplicação na plataforma desktop.



Fonte: Elaborado pela autora.

A tela home é direcionada a usuários que se autenticaram através de um *login* de acesso na página inicial, onde todos os participantes do grupo tem acesso a todas as operações realizadas pelo grupo dentro do jogo, como informações de usuários adicionados no grupo, cadastro de novas tarefas disponíveis para serem feitas, informações de tarefas concluídas por usuários e etc. A cada nova operação realizada dentro do jogo por usuários, serão apresentadas nesta tela para que todos do grupo possam acompanhar o que está acontecendo no jogo.

Figura 13: Tela de Menu da Aplicação na plataforma desktop



Fonte: Elaborado pela autora.

Em todas as telas do jogo, está o botão de opção de visualização de 'menu' do jogo, que é a tela onde estão todas as opções disponíveis para fazer operações no jogo, veja na Figura 13. Nesta tela de menu é possível escolher entre ver todas as tarefas, ver informações do seu perfil, entrar no chat do grupo, ver a lista de participantes do grupo, ver o ranking do jogo, etc.

As figuras anteriores mostram como ficaram algumas telas da aplicação na plataforma *desktop*, que são dispositivos como computadores e notebooks que têm acesso a essa visão de tela. A seguir serão mostradas como ficaram mais algumas telas da aplicação, desta vez nas plataformas *mobile*, que são acessados por dispositivos como *tablets* e *smartphones*.

Figura 14: Tela de Perfil e Tela de Chat em Smartphone.



Fonte: Elaborado pela autora.

Na Figura 14 está a tela de perfil que está disponível para os dois modos de usuários do sistema, administrador e jogador, e nessa tela é apresentado todas as informações do usuário que está *logado* no sistema, como nome, nome de usuário e pontos obtidos no jogo para serem trocados.

Na Figura 14 está também a tela de chat do jogo e é nesta área que fica toda conversa feita pelos usuários do grupo. Nesta área qualquer usuário do grupo pode escrever mensagens a todo o restante do grupo e ter acesso a toda conversa do mesmo.

Figura 15: Telas de tarefas e de ranking na plataforma Mobile.



Fonte: Elaborada pela autora.

Nas figuras 14 e 15 estão duas áreas do jogo, que são respectivamente, a área de tarefas disponíveis para realização e a área de *ranking* do jogo. Estas duas áreas podem ser acessadas através do menu principal do jogo, mostrado na Figura 13

Na área de tarefas do jogo, todos os usuários do grupo têm acesso e poderão ver todas as tarefas cadastradas pelo administrador do grupo, que estão disponíveis para realização, ou seja, são todas as tarefas que ainda não foram executadas. Esta área também mostra as opções de realização da tarefa assim como de exclusão da tarefa (esta opção apenas administradores do grupo conseguem visualizar)

Na área de *ranking* do jogo é possível visualizar a colocação e quantidade de pontos de todos os jogadores do grupo, em ordem decrescente de pontos, onde quem tem mais pontos fica no topo do *ranking*, mostrando assim quem mais ganhou

com as realizações das tarefas do grupo.

Figura 16: Tela de participantes e tela de menu administrativo na plataforma mobile



Fonte: Elaborado pela autora.

Na Figura 16 estão as telas das áreas “participantes” do jogo e “menu administrativo”, respectivamente.

Na tela “participantes” estão todos os usuários cadastrados no grupo que competirão entre si para ganhar liderar o *ranking* de pontos do jogo. Nesta área, todos podem ver informações como nome, nome de usuário e pontuação atual de cada jogador do grupo. É possível também escolher a opção de trocar os pontos de algum participante escolhido, porém esta opção é visível apenas para usuário administrador.

Na tela de Menu administrativo somente usuários que são administradores dos grupos tem acesso a essa área, portanto só eles podem ver informações desta tela. Neste menu tem opções restritas aos comandantes dos grupos, como adição de novas tarefas para o grupo realizarem, adição de novos participantes ao jogo, tarefas pendentes para serem verificadas e aprovadas, assim como a troca de pontos dos participantes do grupo.

4 TESTES E AVALIAÇÕES

Este tópico tem como objetivo identificar e definir os requisitos e componentes que foram testados.

4.1 REQUISITOS TESTADOS

Abaixo, estão listados os requisitos e componentes da aplicação considerados críticos e que, portanto, foram testados.

4.1.1 Teste do Banco de Dados

- Verificar se o sistema está apto a cadastrar usuários administradores (iniciais);
- Verificar se o sistema está apto a cadastrar participantes através do perfil administrador;
- Verificar se o sistema está apto a cadastrar ou remover tarefas;
- Verificar se o sistema distingue tarefas que estão disponíveis para realizar, tarefas pendentes de aprovação e tarefas executadas e aprovadas;
- Verificar se o sistema guarda a hora do sistema e o nome do usuário para todas as operações realizadas;
- Verificar se o sistema guarda todos os pontos ganhos ao serem aprovadas as tarefas realizadas aos usuários.
- Verificar se o sistema guarda todas as conversas do grupo escritas no chat.
- Verificar se o sistema retira os pontos dos usuários quando é feita a troca de pontos por administradores.
- Verificar se os dados do sistema estão sempre atualizados (se há consistência).

4.1.2 Teste Funcional

- Verificar o acesso dos usuários através de *login* e senha;
- Verificar se cada usuário pode efetuar as operações, através do sistema, a ele atribuídas;
- Verificar se as informações estão sendo apresentadas no formato correto.

4.1.3 Teste da Interface do Usuário

- Verificar se há redundância de acesso a um mesmo formulário (se há duas formas de chegar ao mesmo formulário), quando necessário;
- Verificar se os menus são o mais simplificado possível;
- Verificar se a disposição dos dados na tela tem boa visibilidade;
- Verificar a gramática e a ortografia do português apresentado na interface gráfica.

4.1.4 Teste de Segurança e de Controle de Acesso

- Verificar se cada usuário apenas tem acesso às informações a ele necessárias;
- Verificar se apenas o administrador do sistema tem acesso às operações de cadastro de participantes, cadastro de novas tarefas, remoção de tarefas, aprovação e reprovação de tarefas e troca de pontos.
- Verificar se todas as operações apenas podem ser efetuadas a partir da autenticação de usuário previamente solicitada.

4.1.5 Teste de Instalação

- Verificar se as configurações do servidor são suficientes para a instalação do sistema;
- Verificar se o sistema foi devidamente instalado no servidor principal;
- Verificar se os terminais conseguem acessar o sistema através do *browser* e se este funciona da forma esperada.

4.1.6 Resultados dos testes

Todos os requisitos de testes descritos acima foram testados diversas vezes durante o desenvolvimento e após sua conclusão, seguindo o roteiro do modelo de desenvolvimento, onde testes unitários dos módulos criados eram realizados, como por exemplo, o teste de banco de dados, onde era verificado se o sistema cadastrou no banco de dados um usuário fictício, criado no momento do teste. Durante o desenvolvimento os testes eram feitos e quando não aprovados, eram refeitos os códigos das respectivas funções a serem executadas, para que logo depois novos

testes fossem feitos até a validação do resultado.

Todos esses testes foram feitos pela desenvolvedora do sistema em protótipos e no sistema final e todos eles tiveram sucesso no final da aplicação. Portanto o sistema está totalmente apto a realizar todas as funcionalidades descritas no levantamento de requisitos mencionado no tópico de 2.3 do Capítulo 2.

4.1.7 Testes com usuários reais

Para verificar se a aplicação conseguiria cumprir com os objetivos descritos nesse trabalho, o sistema foi colocado em um servidor externo para possibilitar ser acessado de qualquer dispositivo com *browser* e internet. Dessa forma foi colocado para teste a aplicação com 3 (três) famílias diferentes com rotinas variadas em um período de 20 dias. Veja a descrição do perfil das famílias encontradas para fazer o teste da aplicação:

- (01) A primeira família é formada por uma mãe, um pai e dois filhos do gênero masculino, um com 8 anos de idade e o outro com 6 anos. Nesse caso, no período de teste o pai que trabalha em outra cidade por alguns períodos não estava presente para participar.
- (02) A segunda família é formada por uma mãe, um pai e um filho do gênero masculino, com 10 anos de idade.
- (03) A terceira família é formada por uma mãe e dois filhos, um do gênero feminino e outro do gênero masculino, uma com 10 anos e o outro com 13.

Para realização dos testes foi feita uma breve descrição do propósito da aplicação e uma breve explicação de como funcionava o sistema para que as famílias pudessem entender e começar a utilizá-los, também foram tiradas pequenas dúvidas em relação à utilização e como poderia aplicá-la no decorrer dos dias.

Após os 20 (vinte) dias de testes, foi levado até as três famílias um pequeno formulário com algumas questões para serem respondidas. Segue abaixo as perguntas elaboradas para avaliar a aplicação e até quatro sugestões pré-definidas de respostas:

1. Como você avaliava a necessidade de utilização dessa aplicação na sua família para o objetivo proposto? (Possibilidades de respostas pré-definidas: Não necessitávamos, Necessitávamos).

2. Qual ou quais plataformas foram usadas para utilizar a aplicação? (Possibilidades de respostas pré-definidas: *Smartphones/tablets*, Computadores, *Smartphones/tablets* e Computadores).
3. Qual nível de dificuldade em realizar as funções propostas na aplicação nos primeiros dias de uso pelo administrador? (Possibilidades de respostas pré-definidas: Muito difícil, difícil, médio, fácil).
4. Qual nível de dificuldade em realizar as funções propostas na aplicação nos primeiros dias de uso pelos participantes? (Possibilidades de respostas pré-definidas: Muito difícil, difícil, médio, fácil).
5. Como vocês avaliam a parte visual da aplicação? (interface do sistema) (Possibilidades de respostas pré-definidas: Ruim, Média, Boa).
6. Como vocês avaliam a dificuldade em visualizar informações na aplicação? Como textos, imagens, campos a serem preenchidos, botões, etc? (Possibilidades de respostas pré-definidas: Ruim, Média, Boa).
7. Qual nível de dificuldade em realizar as funções propostas na aplicação nos últimos dias de uso pelo administrador? (Possibilidades de respostas pré-definidas: Muito difícil, difícil, médio, fácil).
8. Qual nível de dificuldade em realizar as funções propostas na aplicação nos últimos dias de uso pelos participantes? (Possibilidades de respostas pré-definidas: Muito difícil, difícil, médio, fácil).
9. A aplicação foi utilizada pelo administrador e pelos participantes por pelo menos quanto tempo durante o período de teste (Possibilidades de respostas pré-definidas: Por até 10 dias (até de 50%), por mais de 10 dias (acima de 50%), por mais de 16 dias (80%)).
10. Em relação à quantidade de realização de tarefas no período de teste, como você avalia o desempenho? (Possibilidades de respostas pré-definidas: Piorou, nada mudou, melhorou).
11. Em relação à satisfação da realização de tarefas no período de teste, como você avalia o desempenho? (Possibilidades de respostas pré-definidas: Piorou, nada mudou, melhorou).
12. Ao administrador: Você pretende continuar utilizando a aplicação recorrentemente com sua família? (Possibilidades de respostas pré-definidas: Não, Talvez, Sim).

Com as perguntas elaboradas, foram solicitadas respostas às famílias para que fosse necessário ter retorno dos testes feitos, em relação à utilização, desempenho e resultados dos objetivos. Seguem abaixo tabelas com a relação de respostas das perguntas feitas às famílias.

Figura 17: Respostas da família 01 dos testes.

Família 01	
Pergunta	Resposta
1	Necessitávamos
2	Smartphones/tablets
3	Difícil
4	Difícil
5	Boa
6	Boa
7	Médio
8	Médio
9	Por mais de 10 dias (acima de 50%)
10	Nada mudou
11	Nada mudou
12	Talvez

Fonte: Elaborado pela autora.

Com as respostas obtidas no relatório da família 01, é possível ver que havia a necessidade de melhora na realização de tarefas domésticas na casa. É possível perceber também que inicialmente ao utilizar a aplicação, a família teve uma dificuldade, mas que com o passar do tempo de utilização melhorou, conforme foram utilizando e conhecendo a aplicação. Percebe-se também que a parte visual e disposição dos elementos da interface gráfica da aplicação teve um bom resultado.

A família não utilizou a aplicação em muito tempo durante os testes, em média entre 10 e 16 dias, mas por mais de 50% do período de teste e isso pode influenciar no resultado do objetivo proposto de melhorar a realização de tarefas domésticas por parte dos participantes da casa.

Na avaliação final os participantes revelaram não ter tido resultados diferentes com o uso da aplicação e sendo questionados quanto ao que pode ter dado errado, eles mencionaram o fato de um dos filhos, o de 6 anos não saber ler e por isso ter um pouco mais de dificuldade em utilizar a aplicação, precisando de auxílio do administrador do grupo.

Foi questionado também em que os pontos estavam sendo trocados, ou seja, que tipo de recompensas eles estavam tendo quanto a realização das tarefas. A

mãe, responsável pelo grupo, disse que estava trocando em dinheiro para ser gasto na semana, porém a mesma também constatou que mesmo com pontos baixos acabava dando dinheiro extra aos filhos. Nesse caso o intuito de incentivar as crianças a realizarem as tarefas pode ter um resultado negativo, já que fazendo ou não as operações do jogo eles continuam ganhando.

Figura 18: Respostas da família 02 dos testes.

Família 02	
Pergunta	Resposta
1	Necessitávamos
2	Smartphones/tablets
3	Médio
4	Médio
5	Boa
6	Boa
7	Fácil
8	Médio
9	Por mais de 10 dias (acima de 50%)
10	Melhorou
11	Nada mudou
12	Talvez

Fonte: Elaborado pela autora.

Com as respostas obtidas no relatório da família 02, é possível ver que também havia a necessidade de melhora na realização de tarefas domésticas na casa. É possível perceber também que inicialmente ao utilizar a aplicação, a família teve pouca dificuldade em utilizar a aplicação, e que com o passar do tempo de utilização melhorou, conforme foram utilizando e conhecendo a aplicação.

Esta família também utilizou a aplicação em média entre 10 e 16 dias do teste. Na avaliação final da aplicação, a família revelou ter tido bons resultados (melhora) na quantidade de tarefas realizadas pelos participantes, em relação ao período em que não utilizavam, mas que em satisfação (prazer) em realizar as tarefas, continuavam do mesmo jeito.

Em conversa com a família, a mesma revelou que o filho questionou não ter muitos participantes disputar o ranking e assim ficando um pouco desestimulado para competir, tendo apenas ele e a mãe, já que o pai não participava do jogo. Dessa forma é possível perceber também que o jogo fica cada vez mais atrativo conforme mais participantes para competir.

Figura 19: Respostas da família 03 dos testes.

Família 03	
Pergunta	Resposta
1	Necessitávamos
2	Smartphones/tablets e computadores
3	Médio
4	Médio
5	Boa
6	Boa
7	Fácil
8	Fácil
9	Por mais de 16 dias (acima de 80%)
10	Melhorou
11	Melhorou
12	Sim

Fonte: Elaborado pela autora.

Com as respostas obtidas no relatório da família 03, é possível ver que também havia a necessidade de melhora na realização de tarefas domésticas na casa. É possível perceber também que inicialmente ao utilizar a aplicação, a família teve pouca dificuldade em utilizar a aplicação, e que com o passar do tempo de utilização melhorou, conforme foram utilizando e conhecendo a aplicação.

Esta família também utilizou a aplicação em média mais de 16 dias, ou seja, mais de 80% do período do teste. Na avaliação final da aplicação, a família revelou ter tido bons resultados (melhora) na quantidade de tarefas realizadas pelos participantes, em relação ao período em que não utilizavam. Tiveram bons resultados também em relação a satisfação (prazer) em realizar as tarefas. Em conversa com a família, a mãe, responsável pelo grupo falou ter ajudado na casa e que pretende continuar utilizando a aplicação para incentivar os filhos a ajudá-la.

4.1.8 Conclusões dos testes reais

Com a análise das respostas do formulário e em conversa com as famílias que participaram do teste, foi possível perceber que em mais de 66,6% das famílias se satisfizeram com a aplicação e tiveram resultados positivos quanto aos objetivos propostos, de melhora na participação de crianças e adolescentes na realização de tarefas domésticas com a família.

Foi possível também perceber que os participantes mais velhos do grupo se atraía mais pelo jogo e participava mais, assim como também tiveram mais

facilidade de uso da aplicação.

Além disso, foi visto que para o resultado do objetivo de a aplicação ser melhor é preciso ter disciplina do administrador para garantir a troca dos pontos e as recompensas serem dadas de acordo com o que é realizado.

5 CONCLUSÃO

O desenvolvimento do presente projeto que foi feito com o estudo do problema, dos casos, das possíveis soluções e das ferramentas utilizadas possibilitou a criação da aplicação denominada “Comandando em Casa” que tinha como objetivo melhorar a participação de crianças e adolescentes na realização de tarefas domésticas com a família utilizando técnicas aplicadas em jogos, transformando assim a aplicação criada em um jogo de realidade alternativa.

Foi possível ver que a utilização de um modelo de desenvolvimento de softwares que fez seguir um cronograma e uma sequência de etapas, assim como a utilização de linguagens de modelagem como a UML, ajudaram no processo de desenvolvimento em relação ao tempo de criação e a redução de erros no sistema.

Na documentação deste projeto foi mostrado o resultado final da aplicação, como ficaram as telas, as opções de operações e todas as funções que a aplicação pode executar. Mostrando assim que no desenvolvimento da aplicação foi possível atender a todos os requisitos pré-determinados para o bom funcionamento do sistema e para um bom resultado dos objetivos.

O objetivo geral foi atingido em pelo menos 66,6% das famílias que utilizaram a aplicação em suas casas no período de teste disponível, onde foi visto que crianças e adolescentes ficaram mais participativos em casa na maioria dessas famílias. Porém, é preciso ressaltar que a pesquisa contou com a participação de apenas três famílias.

E dessa forma, não é possível afirmar com total certeza que essa aplicação poderá ajudar famílias a incentivarem e melhorarem a participação de crianças e adolescentes na realização de tarefas domésticas. Serão necessárias pesquisas com amostras maiores de dados.

Foi concluído também que para chegar a essa situação, as famílias têm que participar mais do jogo assim como cumprir as regras de recompensar os participantes que ganharam os pontos durante o jogo.

5.1 TRABALHOS FUTUROS

Esse trabalho permite identificar alguns temas para trabalhos futuros. São

eles:

- Desenvolver aplicativos *mobile* para sistemas como *Android* e *IOS* que possibilitem o acesso a essa aplicação por meio de dispositivos móveis como *smartphones* e *tablets* de uma forma mais fácil e direta ao usuário.
- Divulgar a aplicação para que mais pessoas com esses problemas possam usá-las e relatar dados mais específicos quanto ao seu uso e seus resultados.
- Criar a possibilidade de famílias (grupos) poderem disputar entre si, onde uma família possa competir com a outra, podendo assim possibilitar uma maior participação dos usuários do jogo e tentar resolver o problema que famílias pequenas enfrentam na aplicação.
- Fazer melhorias internas e possíveis correções de erros que aparecerem.

REFERÊNCIAS

ABRIL, EXAME. Tecnologia. **69% das crianças aprendem primeiro a usar um PC.** 2011. Disponível em: <https://exame.abril.com.br/tecnologia/69-das-criancas-aprendem-primeiro-a-usar-um-pc/>. Acesso em 20 out. 2018.

_____. **10 razões pelas quais aparelhos móveis devem ser proibidos para crianças menores de 12 anos.** 2014. Disponível em: <https://exame.abril.com.br/tecnologia/10-razoes-pelas-quais-os-aparelhos-moveis-devem-ser-proibidos-para-criancas-menores-de-12-anos/>. Acesso em 20 out. 2018.

BONFIM, F. L.; LIANG, M. Aplicações escaláveis com Mean Stack. 2014. 48 f. **Tese (Doutorado) Curso de Bacharelado em Ciência da Computação**, Universidade Federal do Paraná, Curitiba, 2014. Disponível em: <http://www.inf.ufpr.br/bmuller/TG/TG-FilipeMichael.pdf>. Acesso em 30 maio 2018.

BORBA, Â. M. O brincar como um modo de ser e estar no mundo. In: BEAUCHAMP, J et al (Org.). Ensino fundamental de nove anos: orientações para a inclusão da criança de seis anos de idade. Brasília: Ministério da Educação, Secretaria de Educação Básica, 2007.

BUNCHBALL INC. **Gamification 101: an introduction to the use of game dynamics to influence behavior.** 2010. Disponível em: <https://www.bunchball.com/sites/default/files/downloads/gamification101.pdf>. Acesso em 18 dez. 2017.

CAELUM. **Desenvolvimento WEB com HTML, CSS e JavaScript.** 2017. Disponível em: <https://www.caelum.com.br/apostila-html-css-javascript/>. Acesso em 15 jun. 2018.

CAMMINO. **Cammino Digital.** Disponível em: www.cammino.com.br. Acesso em 23 out. 2018.

CHOREWARS. **Chore Wars :: Claim Experience Points for Housework.** Disponível em: <http://www.chorewars.com>. Acesso em 9 mar. 2017.

CORDEIRO, A. C. B.; SAÉS, M. E. L. Jogos de realidade alternativa para a educação. **Revista Tecnológica da Fatec Americana**, v. 1, n. 1, p. 14, 2016.

CORRÊA, G. G. Aplicações empresarias adaptativas ao dispositivo. 2013. Monografia (Ciência da Computação) — **Instituto Municipal de Ensino Superior de Assis** – IMESA e Fundação Educacional do Município de Assis – FEMA.

COSTA, A. C. S.; MARCHIORI, P. Z. Gamificação, elementos de jogos e estratégia: uma matriz de referência. InCID: **Revista de Ciência da Informação e Documentação**, v. 6, n. 2, p. 44-65, 2015. Disponível em: <<https://www.revistas.usp.br/incid/article/download/89912/103928>>. Acesso em 18 dez. 2017.

COSTA, F. F. **Design responsivo para WEB com Bootstrap**. 2014. Disponível em: <<http://blog.unifimes.edu.br/fernando/wp-content/uploads/sites/2/2014/05/Design-responsivo-para-WEB-com-Bootstrap.pdf>>. Acesso em 14 jun. 2018.

CRUZ JUNIOR, G. **Da Realidade em Jogo ao Jogo na Realidade: Os Games como Baluartes da Mudança Social**. Disponível em: <https://www.academia.edu/6062405/Da_Realidade_em_Jogo_ao_Jogo_na_Realidade_Os_Games_como_Baluartes_da_Mudanca_Social>. Acesso em 8 mar. 2017.

DE PAIVA, N. M. N.; COSTA, J. S. **A influência da tecnologia na infância: desenvolvimento ou ameaça**. 2015. Disponível em: <<http://www.psicologia.pt/artigos/textos/A0839.pdf>>.

DEVMEDIA. **DevMedia | Plataforma para Programadores**. Disponível em: <www.devmedia.com.br>. Acesso em 23 out. 2018.

_____. **NoSQL Tutorial: Introdução aos bancos de dados NoSQL**. Disponível em: <<https://www.devmedia.com.br/introducao-aos-bancos-de-dados-nosql/26044>>. Acesso em 23 out. 2018.

DIAS, J. C.; VIEIRA, R. M. B.; SILVA, R. R. da. Desenvolvimento de um aplicativo integrado baseado em data scrapping – WINDOWS 8 APP. **Revista Eletrônica Científica de Ciência da Computação**, 2013.

DRUMMOND, A. de F. **Participação de crianças e de adolescentes nas tarefas domésticas**. Disponível em: <<http://www.bibliotecadigital.ufmg.br/dspace/handle/1843/BUOS-9JUHQY>>. Acesso em 9 mar. 2017.

DUNN, L. et al. **Factors associated with participation of children with and without ADHD in household tasks**. Phys Occup Ther Pediatr., v. 29, n. 3, p. 274-294, 2009.

EDUCAUSE. **7 Things you should know about... Alternate Reality Games**. Disponível em: <<https://net.educause.edu/ir/library/pdf/ELI7045.pdf>>. Acesso em 9 mar. 2017.

EXPRESS. **Express – framework de aplicativo da web Node.js**. Disponível em: <<https://expressjs.com/pt-br/>>. Acesso em 20 out. 2018.

GOODNOW, J. J. et al. **Would Ask Someone Else to Do This Task? Parents' and Children's Ideas About Household Work Requests**. Developmental Psychology, v. 27, n. 5, p. 817-828, sep. 1991.

GUDWIN, R. R. **Diagramas de Atividade e Diagramas de Estado**. Disponível em: <<http://www.dca.fee.unicamp.br/~gudwin/ftp/ea976/AtEst.pdf>>. Acesso em 16 jun. 2018.

GUEDES, G. **UML: uma abordagem prática**. São Paulo. Editora: Novatec, 2004.

IBGE. Acesso a internet e a televisão e posse de telefone móvel celular para uso pessoal. Disponível em: <<http://biblioteca.ibge.gov.br/visualizacao/livros/liv93373.pdf>>. Acesso em 6 abr. 2017.

JQUERY. **jQuery**, Disponível em: <<https://jquery.com/>>. Acesso em 23 out. 2018.

LEITE, J. C. **O Processo de Desenvolvimento de Software**, 2000. Disponível em: <<https://www.dimap.ufrn.br/~jair/ES/c2.html>>. Acesso em 22 maio 2018.

LESSA, R. O.; LESSA JUNIOR, E. O. **Modelos de Processos de Engenharia de Software**. Disponível em: <[8fa4dcb2-a-62cb3a1a-sites.google.com/site/20121engsoft/arquivos2012-1/ModelodeArtigo.pdf](https://sites.google.com/site/20121engsoft/arquivos2012-1/ModelodeArtigo.pdf)>. Acesso em 11 jun. 2018.

LIMA, A. G.; NETSCAPE; NETCENTER. **Javascript Aplicações interativas para Web. 2015.** Disponível em: <<http://pavesys.com.br/download/JavaScript.pdf>>. Acesso em 11 jun. 2018.

LOPES, So. **A web mobile: programe para um mundo de muitos dispositivos.** São Paulo: Casa do Código, 2013.

MCGONIGAL, Jane. **Reality is broken: Why games make us better and how they can change the world.** Penguin, 2011.

NETCODERS. **Utilizando o MongoDB em aplicações .NET.** Disponível em: <<https://medium.com/netcoders/utilizando-o-mongodb-em-aplica%C3%A7%C3%B5es-net-6a7c1cceeec>>. Acesso em 23 out. 2018.

OLIVEIRA, L. **Diagrama de Atividades - Linguagem UML.** 2010. Disponível em: <https://sistemas.riopomba.ifsudestemg.edu.br/dcc/materiais/2121423876_DiagramaDeAtividades.pdf>. Acesso em 16 jun. 2018.

PEREIRA, C. R. **Aplicações web real-time com Node.js.** [S.l.]: Casa do Código, 2013.

PEREIRA, F. S. et al. **Utilização de Banco de Dados NoSql em Ambientes Corporativos.** Disponível em: <<http://www.computacao.unitri.edu.br/erac/index.php/erac/article/view/163/230>>. Acesso em 2 jun. 2018.

PIRES, J. A. L.; GOMES, G. S.; BORTOLIN, F. **Apresentação ao HTML e ao CSS.** Goiânia: [s.n.], 2016. Disponível em: <http://gti.projetointegrador.com.br/~161M154200161/Modulo_1/Lucilia/AECLucilia.pdf>. Acesso em 14 jun. 2018.

POLITOWSKI, C.; MARAN, V. **Comparação de desempenho de PostgreSQL e MongoDB.** 2014. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/erbd/2014/003.pdf>>. Acesso em 1 jun. 2018.

RABIN, Steve. **Introdução ao desenvolvimento de games.** São Paulo: Cengage Learning, 2012.

REIS, R. S. **Uso ilimitado e frequente da tecnologia em crianças na faixa etária de 2 à 6 anos de idade**. 2015. 63 f. Monografia (Licenciatura em Pedagogia)— Universidade de Brasília, Brasília, 2015. Disponível em: <http://bdm.unb.br/bitstream/10483/12944/1/2015_RaquelSampaioReis.pdf>. Acesso em 29 mar. 2017.

SAMPAIO, M. C. **Material sobre UML**. 2007. Disponível em: <<http://www.dsc.ufcg.edu.br/~sampaio/cursos/2007.1/Graduacao/SI-II/Uml/>>. Acesso em 13 jun. 2018.

SANCHES, C.; CARDELINO, L.; RAMOS, T. Proteja nossas crianças e jovens. 2014. Disponível em: <http://www.crianca.mppr.mp.br/arquivos/File/publi/internet/avg_ebook.pdf>. Acesso em 7 mar. 2017.

SAVI, R.; ULBRICHT, V. R. Jogos digitais educacionais: benefícios e desafios. **Renote**, v. 6, n. 1, 2008.. Disponível em: <<http://www.seer.ufrgs.br/renote/article/viewFile/14405/8310>>. Acesso em 7 mar. 2017.

SCHROEDER, R.; SANTOS, F. dos. Arquitetura e testes de serviços web de alto desempenho com Node.JS e MongoDB. Universidade do Estado de Santa Catarina - UDESC, 2014. Disponível em: <http://www.ceavi.udesc.br/arquivos/id_submenu/787/ricardo_schroeder_versao_final_.pdf>. Acesso em 9 mar. 2017.

STATCOUNTER. Android overtakes Windows for first time. Disponível em: <<http://gs.statcounter.com/press/android-overtakes-windows-for-first-time>>. Acesso em 3 maio. 2017.

TANENBAUM, A. Redes de Computadores. 4. ed. [S.l.]: Editora Campus. 2003.

TOMAZINI, M.; LOPES, L. F. B. Web design responsivo - Bootstrap. **II Seminário empresarial e III Jornada de TI**, 2015.

TONIN, F. S.; CITTOLIN, G. F.; SOUZA, V. D. **Desenvolvimento de um sistema web de controle de acesso**. 2015. Monografia (Engenharia industrial elétrica/automação) — Universidade tecnológica federal do Paraná.

VIEIRA, M. R. et al. **Bancos de Dados NoSQL**: Conceitos, Ferramentas, Linguagens e Estudos de Casos no Contexto de Big Data. Simpósio Brasileiro de Bancos de Dados, 2012.

WAGNER, A. et al. **Compartilhar tarefas? Papéis e funções de pai e mãe na família contemporânea**. *Psic.: Teor. e Pesq.* [online]. 2005, vol.21, n.2, pp.181-186. Disponível em: <<http://www.scielo.br/pdf/ptp/v21n2/a08v21n2>>. Acesso em 9 mar. 2017.