

UNIVERSIDADE FEDERAL DO PARANÁ

JEAN CARLO DA SILVA
JEFFERSON DIEGO FIDENCIO
THIAGO FERREIRA VAZ

FAMILYST - ORGANIZADOR DE EVENTOS E ATIVIDADES FAMILIARES

CURITIBA

2016

JEAN CARLO DA SILVA
JEFFERSON DIEGO FIDENCIO
THIAGO FERREIRA VAZ

FAMILYST – ORGANIZADOR DE EVENTOS E ATIVIDADES FAMILIARES

Trabalho apresentado como requisito parcial à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, Setor de Educação profissional e Tecnológica, da Universidade Federal do Paraná.

Orientador: Professor Dr. Alexander Robert Kutzke.

CURITIBA

2016

RESUMO

Nos dias atuais, a tecnologia vem alcançando cada vez mais espaço no mercado e no dia a dia dos membros da sociedade, conectando-se com as diversas áreas e agilizando o acesso à informação, auxiliando na saúde, lazer, segurança, educação, entre outros. Com o intuito de aliar a tecnologia do armazenamento de dados em nuvem e os avanços dos dispositivos móveis à comunicação interpessoal entre membros de uma família, este projeto de conclusão de curso descreve o desenvolvimento de um aplicativo *mobile* Android e de um servidor com serviço RESTful, para organizar atividades, imagens, vídeos e notícias, visando unir todos os membros de uma família em uma só plataforma. Este documento compreende a descrição dos conceitos nos quais o projeto se baseia e das tecnologias utilizadas no desenvolvimento do sistema, plano de atividades, divisão de responsabilidades, documentação e descrição das funcionalidades do sistema desenvolvido.

Palavras-Chave: Organização, aplicativo, dispositivos móveis, família, armazenamento em nuvem.

ABSTRACT

Nowadays, technology is reaching more and more space in the Market and in the daily life of society members, connecting with the various areas and streamlining access to information, assisting in health, recreation, security, education, among others. In order to combine cloud data storage technology and the advancement of mobile devices with interpersonal communication between members of a Family, this project describes the development of an Android OS Application, to organize activities, images, videos and news, looking to unite all the members of a family in a single platform. This document includes the description of the concepts on which the project is based and the technologies used in the system's development, activity plan, division of responsibilities, documentation and description of the functionalities of the developed system.

Key words: Organization, application, mobile devices, family, cloud data storage.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| FIGURA 1 – MENU DE ITENS DO APLICATIVO 'COZI – <i>FAMILY ORGANIZER</i> ' ... | 20 |
| FIGURA 2 – CRIAÇÃO DE EVENTOS NA REDE SOCIAL 'FACEBOOK' | 22 |
| FIGURA 3 – FASES DO PROCESSO RUP | 23 |
| FIGURA 4 – GERENCIAMENTO DE BANCOS DE DADOS E GERENCIAMENTO DE SERVIDORES NO NETBEANS..... | 30 |
| FIGURA 5 – BREAKPOINT PARA DEBUG DE CÓDIGO FONTE EM TEMPO DE EXECUÇÃO NO NETBEANS | 31 |
| FIGURA 6 – SMART CODE COMPLETION DO NETBEANS..... | 31 |
| FIGURA 7 – TAG DO XML COM REFERÊNCIA AO “MYSQL-CONNECTOR-JAVA” | 32 |
| FIGURA 8 – ADMINISTRAÇÃO DO BANCO DE DADOS E TABELAS NO MYSQL WORKBENCH | 33 |
| FIGURA 9 – DIAGRAMA DE BANCO DE DADOS DESENVOLVIDO NO MYSQL WORKBENCH | 33 |
| FIGURA 10 – EXECUÇÃO DE UMA REQUISIÇÃO <i>POST</i> NOS PADRÕES REST . | 34 |
| FIGURA 11 – ARQUIVO “POM.XML” | 35 |
| FIGURA 12 – ANOTAÇÕES JAX-RS | 37 |
| FIGURA 13 – ANOTAÇÕES JAX-B..... | 37 |
| FIGURA 14 – DESCRIÇÃO REST | 39 |
| FIGURA 15 – IMPLIMENTAÇÃO NO NETBEANS DOS VERBOS HTTP UTILIZADOS | 39 |
| FIGURA 16 – EXEMPLO DE HIERARQUIA DE RECURSOS | 40 |
| FIGURA 17 – USO DOS <i>STATUS CODES</i> | 40 |
| FIGURA 18 – EXEMPLO DA UTILIZAÇÃO DOS TIPOS DE <i>HEADERS</i> | 40 |

| | |
|---|----|
| FIGURA 19 – ARQUITETURA MVC | 41 |
| FIGURA 20 – ARQUITETURA MVC NO FAMILYST | 42 |
| FIGURA 21 – IMPLEMENTAÇÃO DAO PATTERN | 43 |
| FIGURA 22 – TELA DE LOGIN | 53 |
| FIGURA 23 – TELA DE RECUPERAÇÃO DE SENHA | 54 |
| FIGURA 24 – TELA DE CADASTRO DE USUÁRIO | 55 |
| FIGURA 25 – TELA DE MENU | 56 |
| FIGURA 26 – TELA DE NOTÍCIAS | 57 |
| FIGURA 27 – TELA DE DETALHES DA NOTÍCIA | 58 |
| FIGURA 28 – TELA PARA ADICIONAR NOTÍCIA | 59 |
| FIGURA 29 – TELA DE EVENTOS | 60 |
| FIGURA 30 – TELA DE DETALHES DO EVENTO | 61 |
| FIGURA 31 – TELA CRIAR EVENTOS | 61 |
| FIGURA 32 – TELA DE MEMBROS | 62 |
| FIGURA 33 – TELA DE CADASTRO DE MEMBRO NA FAMÍLIA | 63 |
| FIGURA 34 – TELA DE ÁLBUNS | 64 |
| FIGURA 35 – TELA DE CADASTRO DE ÁLBUM | 65 |
| FIGURA 36 – TELA DE FOTOS | 65 |
| FIGURA 37 – TELA DE GALERIAS | 66 |
| FIGURA 38 – TELA DE CADASTRO DE VÍDEO | 67 |

LISTA DE TABELAS

| | |
|--|-----|
| TABELA 1 – REQUISITOS FUNCIONAIS | 50 |
| TABELA 2 – REQUISITOS NÃO FUNCIONAIS | 52 |
| TABELA 3 – API REST | 112 |

LISTA DE ABREVIATURAS

ANATEL – Agência Nacional de Telecomunicações

API – *Application Programming Interface*

DAO – *Data Access Object*

GPS – Sistema de Posicionamento Global

HTTP – *Hypertext Transfer Protocol*

IBGE – Instituto Brasileiro de Geografia e Estatística

IDE – *Integrated Development Environment*

J2SE – *JAVA 2 Standard Edition*

Java SE – *Java Standart Edition*

JAX-B – *Java Architecture for XML Binding*

JAX-RS – *Java API for RESTful Services*

JSON – *JavaScript Object Notation*

JSP – *Java Server Pages*

MVC – *Model, View and Controller*

PNAD – Pesquisa Nacional por Amostra de Domicílios

POM – *Project Object Model*

REST – *Representational State Transfer*

RUP – *Rational Unified Process*

WBS – *Work Breakdown Structure*

SUMÁRIO

| | |
|---|-----------|
| RESUMO | 3 |
| ABSTRACT | 4 |
| LISTA DE ILUSTRAÇÕES | 5 |
| LISTA DE TABELAS | 7 |
| LISTA DE ABREVIATURAS | 8 |
| | |
| 1. INTRODUÇÃO | 13 |
| 1.1 DESCRIÇÃO DO PROBLEMA | 14 |
| 1.2 JUSTIFICATIVA | 14 |
| 1.3 OBJETIVO GERAL | 15 |
| 1.4 OBJETIVOS ESPECÍFICOS..... | 16 |
| | |
| 2. FUNDAMENTAÇÃO TEÓRICA..... | 17 |
| 2.1 O CONCEITO DE SMARTPHONE..... | 17 |
| 2.2 O RELACIONAMENTO FAMILIAR E A INTERNET..... | 18 |
| 2.3 ORGANIZADORES FAMILIARES EXISTENTES NO MERCADO..... | 18 |
| | |
| 3. METODOLOGIA DO TRABALHO | 23 |
| 3.1 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE..... | 23 |
| 3.1.1 CONCEPÇÃO | 24 |
| 3.1.2 ELABORAÇÃO..... | 24 |
| 3.1.3 CONSTRUÇÃO..... | 25 |
| 3.1.4 TRANSIÇÃO..... | 25 |
| 3.2 PLANO DE ATIVIDADES | 25 |
| 3.3 PLANO DE RISCOS | 27 |
| 3.4 TECNOLOGIAS APLICADAS NO DESENVOLVIMENTO | 29 |
| 3.4.1 JUSTINMIND PROTOTYPER | 29 |
| 3.4.2 ANDROID STUDIO | 29 |
| 3.4.3 NETBEANS | 29 |
| 3.4.3.1 SMART CODE COMPLETION | 31 |

| | |
|--|-----------|
| 3.4.4 MYSQL SERVER | 32 |
| 3.4.5 MYSQL WORKBENCH..... | 32 |
| 3.4.6 POSTMAN | 33 |
| 3.4.7 APACHE MAVEN..... | 35 |
| 3.4.8 APACHE TOMCAT | 35 |
| 3.4.9 JERSEY..... | 36 |
| 3.4.10 JAX-RS | 36 |
| 3.4.11 JAX-B..... | 37 |
| 3.4.12 JSON | 38 |
| 3.4.13 REST | 38 |
| 3.4.14 MVC..... | 41 |
| 3.4.15 DAO PATTERN | 42 |
| 3.4.16 WBS TOOL..... | 43 |
| 3.4.17 ASTAH PROFFESIONAL | 44 |
| 3.4.18 GITHUB | 44 |
| 3.4.20 GOOGLE DRIVE..... | 45 |
| 3.4.20 PACOTE OFFICE..... | 45 |
| 3.4.21 LINGUAGENS DE PROGRAMAÇÃO | 45 |
| 3.4.21.1 JAVA | 46 |
| 3.5 RESPONSABILIDADES | 46 |
| 3.6 REQUISITOS..... | 47 |
| 3.6.1 LEVANTAMENTO DE REQUISITOS..... | 47 |
| 3.6.2 REQUISITOS FUNCIONAIS..... | 49 |
| 3.6.3 REQUISITOS NÃO FUNCIONAIS..... | 51 |
| 4. APRESENTAÇÃO DO SOFTWARE | 53 |
| 4.1 LOGIN (TELA INICIAL) | 53 |
| 4.2 RECUPERAÇÃO DE SENHA..... | 54 |
| 4.3 CADASTRO..... | 55 |
| 4.3 MENU | 56 |
| 4.4 NOTÍCIAS | 57 |
| 4.5 EVENTOS..... | 60 |
| 4.6 FAMÍLIA | 62 |
| 4.7 ÁLBUNS (FOTOS) | 64 |
| 4.8 GALERIAS (VÍDEOS) | 66 |

| | |
|---|------------|
| 5. CONSIDERAÇÕES FINAIS..... | 68 |
| 5.1 DESAFIOS..... | 68 |
| 5.2 IMPLEMENTAÇÕES FUTURAS..... | 69 |
| 5.3 CONCLUSÃO..... | 69 |
| | |
| REFERÊNCIAS..... | 70 |
| | |
| APÊNDICE 1 – QUESTIONÁRIO PARA USUÁRIOS..... | 72 |
| | |
| APÊNDICE 2 – DIAGRAMA DE CASOS DE USO..... | 73 |
| | |
| APÊNDICE 3 – ESPECIFICAÇÃO DE CASOS DE USO..... | 74 |
| UC01 – EFETUAR LOGIN..... | 74 |
| UC02 – EFETUAR CADASTRO..... | 76 |
| UC03 – VISUALIZAR NOTÍCIAS..... | 78 |
| UC04 – COMENTAR NOTÍCIA..... | 80 |
| UC05 – PUBLICAR NOTÍCIA..... | 82 |
| UC06 – VISUALIZAR ÁLBUNS..... | 84 |
| UC07 – CADASTRAR ÁLBUM..... | 86 |
| UC08 – VISUALIZAR VÍDEOS DA GALERIA..... | 88 |
| UC09 – CADASTRAR VÍDEO..... | 90 |
| UC10 – VISUALIZAR EVENTOS..... | 92 |
| UC11 – CADASTRAR EVENTO..... | 94 |
| | |
| APÊNDICE 4 – DIAGRAMA ENTIDADE RELACIONAMENTO..... | 96 |
| | |
| APÊNDICE 5 – DIAGRAMA DE CLASSES DE IMPLEMENTAÇÃO..... | 97 |
| | |
| APÊNDICE 6 – DIAGRAMAS DE SEQUÊNCIA..... | 98 |
| | |
| APÊNDICE 7 – DIAGRAMA WBS..... | 101 |
| PARTE 1 – CONCEPÇÃO..... | 101 |
| PARTE 2 – ELABORAÇÃO..... | 102 |
| PARTE 3 - CONTRUÇÃO..... | 103 |
| PARTE 4 – TRANSIÇÃO..... | 104 |
| PARTE 5 – GERÊNCIA DO PROJETO..... | 105 |

APÊNDICE 8 – GRÁFICO DE GANTT 106

APÊNDICE 8 – CÓDIGOS REST 107

1. INTRODUÇÃO

Atualmente, ter a tecnologia como ferramenta de comunicação é algo fundamental para o ser humano. Fatores como tempo e distância têm sido facilmente vencidos por aplicativos e serviços online em constante evolução, capazes de conectar membros de uma família de maneira fácil e relativamente barata.

Essa evolução estimula famílias a utilizarem o ambiente virtual para diversas atividades, como reuniões planejadas através de redes sociais, grupos em aplicativos de mensagens instantâneas, galerias de fotos compartilhadas, entre outros. Por isso, faz-se necessário pensar em inclusão, para que todos os membros de uma família, estando perto ou longe, independente de faixa etária, possam utilizar o mesmo ambiente.

Segundo o IBGE a expectativa de vida dos brasileiros em 2014 era de 75,2 anos (IBGE, 2014). Este fator corrobora para o aumento do percentual de idosos com acesso à internet, que subiu de 5,7% para 12,6% de 2008 a 2013. Porém, a faixa etária predominante de usuários da internet é a composta pelos adolescentes, chegando a aproximadamente 75% do total. Grande parte das aplicações móveis que existem atualmente contam com diversas funcionalidades e as interfaces não são criadas visando o uso dos idosos, ou seja, todos estes fatores contribuem para que as pessoas da terceira idade ainda enfrentem barreiras na utilização de qualquer ferramenta tecnológica.

Outro fator importante é o fato de que os smartphones são os aparelhos mais utilizados para o acesso à internet, segundo a pesquisa do IBGE, PNAD (IBGE, 2014). Os dados indicam que 29,6 milhões das 36,8 milhões de casas conectadas no Brasil possuem um telefone móvel para o acesso à rede, representando 80,4% do total.

Arrazoamos então que é necessário que todos os usuários da internet tenham direito a utilização dos recursos tecnológicos de modo igualitário, independente das limitações nas quais possam estar acometidos, para que tenham uma qualidade de vida cada vez melhor. Assim, o objetivo deste projeto é desenvolver um aplicativo

para dispositivos móveis, chamado Familyst, com interfaces gráficas simples e intuitivas, para que todos os membros de uma família possam ter acesso às atividades, postagem de fotos e vídeos, publicação de notícias e outras informações inerentes ao seu núcleo familiar.

1.1 DESCRIÇÃO DO PROBLEMA

Com base nos fatos anteriormente apresentados, destaca-se um problema principal: os recursos tecnológicos não são criados pensando na utilização facilitada por qualquer pessoa. Este problema se torna mais acentuado tendo em vista o ambiente familiar, que geralmente é composto por um número considerável de pessoas com idade avançada ou com pouca experiência com tecnologia. Nas diversas redes sociais disponíveis atualmente nota-se a ausência de algo voltado à família, com design simples e intuitivo, que facilite a utilização, principalmente pelos idosos.

Ao analisar-se as aplicações existentes, como por exemplo, a rede social Facebook, com 1,6 bilhão de usuários atualmente, pode ser utilizada para organização de eventos, criação de galerias de fotos compartilhadas e exposição de notícias, porém, além de não possuir o objetivo de ser um ambiente totalmente voltado a membros de uma família, pode oferecer dificuldades para encontrar essas informações, tornando essas atividades desgastantes e cansativas. Também há a possibilidade de grupos de mensagens instantâneas em aplicativos com o WhatsApp, que por sua vez, tem como objetivo apenas a conversação, e não o armazenamento de informações, planejamento de atividades e afins.

1.2 JUSTIFICATIVA

Nos dias de hoje, parece impossível não usufruir da tecnologia para realizar as atividades do cotidiano. Com o avanço tecnológico, presente nos aparelhos e que permitem cada vez mais recursos, nos vemos dependentes de computadores ou

dispositivos móveis no dia-a-dia, seja para trabalho, lazer ou simplesmente para se comunicar com pessoas.

Tendo como destaque os dispositivos móveis, que hoje já existem em maior quantidade do que seres humanos no Brasil, com aproximadamente 1,3 linhas telefônicas para cada pessoa (ANATEL, 2014), estes dispositivos já possuem, em alguns casos, funcionalidades mais sofisticadas que os próprios computadores. Aliados ao benefício da mobilidade, os chamados *smartphones* possuem *hardware* poderoso, contando com câmeras em altas resoluções, vasta memória de armazenamento e processadores de até oito núcleos.

Por isso, faz-se necessário voltar às atenções para estes dispositivos e às mudanças tecnológicas por eles possibilitadas. Assim, propõem-se desenvolver um aplicativo para que famílias possam usufruir desse avanço tecnológico, comunicando-se de maneira rápida e facilitada, organizando eventos e armazenando arquivos, em dispositivos com sistema operacional Android. O presente projeto pretende alcançar todas as faixas etárias, permitindo também a inclusão digital de pessoas de idade avançada, por meio de interfaces gráficas planejadas de modo a suprir suas necessidades e limitações.

1.3 OBJETIVO GERAL

Desenvolver uma aplicação para dispositivos móveis que ofereça serviços de organização de eventos, gerenciamento de arquivos de fotos, vídeos e documentos e comunicação entre os membros da família por meio de postagem de notícias. Visando a integração de todos os membros de uma família, sem limitação de idade atingindo até a mais alta faixa etária, todos os elementos visuais serão adequados proporcionando o uso facilitado do Familyst.

1.4 OBJETIVOS ESPECÍFICOS

Para atingir o objetivo geral do projeto, foram definidos alguns objetivos específicos:

- Desenvolver a aplicação Familyst REST, utilizando a tecnologia Java SE para o *back-end* da solução.
- Desenvolver o projeto utilizando o RUP como modelo de processo.
- Trabalhar em equipe, tornando possível a obtenção de conhecimento e experiência por todos os membros da equipe.
- Aplicar no desenvolvimento da aplicação conceitos de modelagem e programação aprendidos durante o curso de Tecnologia em Análise e Desenvolvimento de Sistemas.
- Desenvolver a aplicação mobile Familyst, utilizando a linguagem Java Android.
- Criar os layouts da aplicação *front-end* com base nos princípios do Material Design – padrões definidos pelo Google para unificar o visual do Android em diversas plataformas (Tecmundo, 2014).

2. FUNDAMENTAÇÃO TEÓRICA

Entende-se a funcionalidade de um aplicativo voltado para uma maior interação e aproximação dos membros de uma família, através de um sistema com o qual se faz possível o compartilhamento de notícias e publicação de imagens e vídeos num grupo fechado à família. Os smartphones ampliam e dão vida à essa possibilidade de modificar a relação familiar, principalmente ao se tratar de parentes distantes, pois gera integração e comunicação que antes, muitas vezes, se limitava a ligações por conta da distância.

2.1 O CONCEITO DE SMARTPHONE

Os *smartphones*, ou literalmente traduzidos “celulares inteligentes”, ainda não possuem uma definição clara e exata difundida entre a comunidade acadêmica. No entanto, essa terminologia é frequentemente utilizada para definir telefones celulares de alta tecnologia.

De acordo com André Lemos (2007, p.25):

O que chamamos de telefone celular é um *Dispositivo* (um artefato, uma tecnologia de comunicação); *Híbrido*, já que congrega funções de telefone, computador, máquina fotográfica, câmera de vídeo, processador de texto, GPS, entre outras; *Móvel*, isto é, portátil e conectado em mobilidade funcionando por redes sem fio digitais, ou seja, de *Conexão*; e *Multirredes*, já que pode empregar diversas redes, como *Bluetooth* e infravermelho (para conexões de curto alcance, entre outros dispositivos), internet (Wi-Fi e Wi-Max) e redes de satélites para uso como dispositivo GPS.

De fato, o poder de processamento desses aparelhos é um ponto importante na sua descrição, mas ainda há outras características a serem ressaltadas, como por exemplo, o fato de possuírem sistemas operacionais de terceiros e a capacidade de rodar softwares de terceiros.

2.2 O RELACIONAMENTO FAMILIAR E A INTERNET

Desde que a internet passou a ser parte importante na vida das pessoas, o conceito de relacionamento tem sido modificado e até mesmo reconstruído dinamicamente nas relações familiares. Cada vez mais, a fronteira entre o mundo real e o virtual se tornam invisíveis, o que pode gerar uma opinião controversa no sentido de unir ou afastar os membros da família.

Uma pesquisa realizada pela revista infantil norte-americana Highlights entrevistou crianças de 6 a 12 anos e levantou que 62% das crianças reclamam que os pais estão distraídos demais para ouvi-los, e o grande causador das distrações são os celulares. Segundo a revista, os smartphones fazem parte do grupo de aparelhos que causam mais distanciamento entre pais e filhos no ambiente familiar (Highlights, 2014)¹.

Por outro lado, quando se trata de comunicação com parentes distantes, a tecnologia tem se mostrado ferramenta fundamental. Até os anos de 1990, haviam duas opções de comunicação – telefone e carta. A primeira opção possuía a mesma velocidade e eficiência de entrega de dados que a internet oferece atualmente, porém, a telefonia era algo extremamente caro na época, e ainda é, caso trate-se de uma ligação de longa distância. Caso a carta fosse adotada como opção, seria necessário ter paciência para aguardar a resposta da pessoa tida como remetente.

A partir do surgimento das novas tecnologias, é inegável que, quando corretamente utilizadas, favorecem a comunicação, ajudando membros de famílias a manterem contato de forma eficaz e barata.

2.3 ORGANIZADORES FAMILIARES EXISTENTES NO MERCADO

Durante o período de levantamento de requisitos do projeto foram realizadas algumas comparações com aplicativos disponíveis no mercado, que apresentam

¹ Disponível em: <https://www.highlights.com/about-us/press-room/national-survey-reveals-62-kids-think-parents-are-too-distracted-listen>

propostas semelhantes ao sistema proposto. Nota-se certa carência no mercado em tal área, pois apenas dois serviços foram encontrados e que puderam ser comparados. Serão expostas a seguir as características de tais aplicativos.

- (1) Cozi – Family Organizer: disponível para todas as plataformas de dispositivos móveis (Android, iOS e Windows Phone). Tem a proposta de simular os tradicionais calendários compartilhados, geralmente na porta da geladeira das residências, listas de compra e de afazeres, caixa de receitas, entre outros. Porém, está disponível apenas em inglês e possui algumas funcionalidades liberadas mediante pagamento de mensalidade.

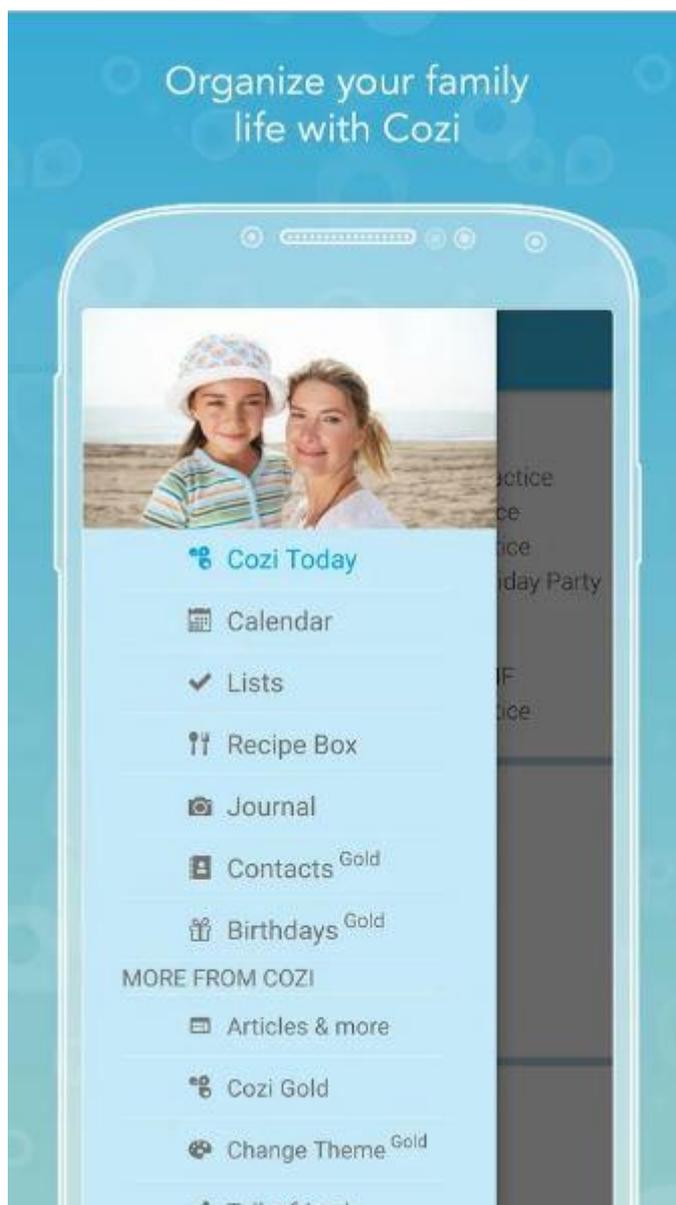
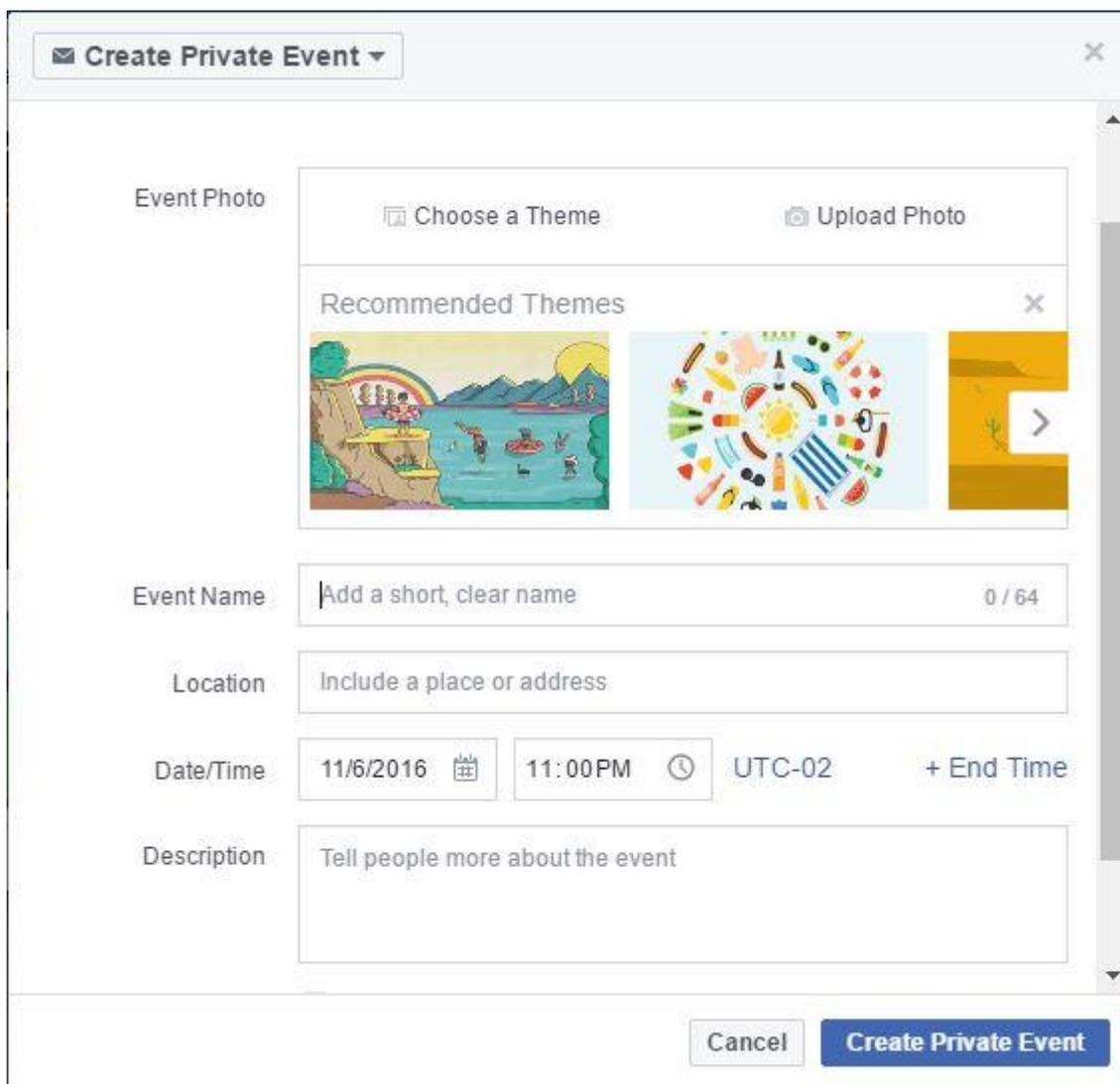


FIGURA 1 – MENU DE ITENS DO APLICATIVO 'COZI – FAMILY ORGANIZER'
FONTE: < https://play.google.com/store/apps/details?id=com.cozi.androidfree&hl=pt_BR>

- (2) Facebook: O Facebook é uma rede social mundialmente conhecida, e esta oferece uma diversidade de funções voltadas para a interação social, porém é bastante ampla, uma vez que necessita da criação de um perfil mais elaborado e de um conhecimento sobre os mecanismos disponibilizados pela rede.

Uma das funções do Facebook é a criação de eventos privados ou públicos, permitindo a realização de convites, observações relacionadas ao evento e interação entre pessoas convidadas. Esta função também está contida no sistema Familyst, porém encontra-se num formato simplificado e enxuto, a fim de propiciar uma facilitação de uso para todas as faixas-etárias.



The image shows a screenshot of the Facebook 'Create Private Event' dialog box. The window title is 'Create Private Event' with a close button (X) in the top right corner. The main content area is divided into several sections:

- Event Photo:** Contains two buttons: 'Choose a Theme' and 'Upload Photo'.
- Recommended Themes:** A carousel of three theme images. The first shows a landscape with a rainbow and people on a boat. The second is a colorful circular collage of various items. The third is a solid yellow background with a white arrow pointing right.
- Event Name:** A text input field with the placeholder text 'Add a short, clear name' and a character count '0 / 64' on the right.
- Location:** A text input field with the placeholder text 'Include a place or address'.
- Date/Time:** A row of three input fields: the first shows '11/6/2016' with a calendar icon, the second shows '11:00PM' with a clock icon, and the third shows 'UTC-02'. To the right of these is a '+ End Time' link.
- Description:** A large text area with the placeholder text 'Tell people more about the event'.

At the bottom right of the dialog, there are two buttons: a 'Cancel' button and a blue 'Create Private Event' button.

FIGURA 2 – CRIAÇÃO DE EVENTOS NA REDE SOCIAL 'FACEBOOK'
FONTE: <www.facebook.com>

3. METODOLOGIA DO TRABALHO

Neste capítulo serão apresentados os processos de softwares que são essenciais para elaboração, condução e desenvolvimento do projeto. Também constam neste tópico os planos de atividades, os planos de riscos identificados e seus requisitos.

3.1 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE

Para o desenvolvimento do projeto, foi utilizado o processo de engenharia chamado RUP. Trata-se de um processo no qual o projeto é dividido em quatro partes de execução: concepção, elaboração, construção e transição. Nessas quatro fases, são tratadas questões relacionadas ao planejamento, análise, levantamento de requisitos, implementação, teste e implantação de um sistema. Cada fase do processo possui papel fundamental para que o objetivo seja cumprido, no caso, o desenvolvimento da aplicação Familyst. Abaixo encontram-se as definições de cada uma dessas fases, explicando seus principais objetivos, e trazendo-os para o ambiente do projeto Familyst, bem como a Figura 3, representando o ciclo de desenvolvimento utilizando o processo RUP.



FIGURA 3 – FASES DO PROCESSO RUP

FONTE: < <http://www.infoescola.com/wp-content/uploads/2010/03>

3.1.1 CONCEPÇÃO

Esta fase do RUP é caracterizada por abranger as tarefas de comunicação com o cliente e planejamento. Nela, devem ser definidos os moldes para desenvolvimento do projeto, justificando a necessidade do desenvolvimento deste. Um plano de projeto é realizado, para análise dos possíveis riscos, estimativas de custo e definições de prazos, para estabelecer as prioridades, caracterizá-las como requisitos e, posteriormente, analisá-los.

No projeto Familyst, a ideia de desenvolver uma aplicação voltada para o ambiente familiar foi definida através da conclusão de que aplicações nesse contexto eram praticamente inexistentes no cenário atual da tecnologia, e os sistemas com características semelhantes não possuíam o caráter de inclusão de faixas etárias, dado ao projeto desenvolvido. Sendo assim, a justificativa principal foi unir membros de uma família em um ambiente em que todas as pessoas, independente de idade, possuíssem fácil acesso a todas as informações. Os primeiros requisitos foram levantados, para que se determinasse uma visão geral do que seria produzido, e foram definidos alguns casos de usos para determinar os principais riscos para o projeto.

3.1.2 ELABORAÇÃO

A fase de Elaboração é caracterizada por definir a arquitetura para o sistema. É nessa fase onde um modelo genérico de processo é definido. Sendo assim, tem-se como objetivo analisar de forma mais detalhada os requisitos do sistema, por meio da criação de diagramas, como o Diagrama de Casos de Uso, Diagrama Entidade-Relacionamento e Diagrama de Classes. Nessa etapa também são feitas indagações sobre os custos do projeto.

No projeto do Familyst a grande maioria das funcionalidades do aplicativo foi definida nessa etapa. Os diagramas iniciais foram desenvolvidos e, para manter a organização do projeto, um cronograma de atividades foi desenvolvido, com metas semanais a serem cumpridas por cada membro da equipe.

3.1.3 CONSTRUÇÃO

Nessa fase, desenvolvem-se os componentes do software. De maneira geral, o principal objetivo é a construção do sistema, tendo como foco o desenvolvimento dos componentes e recursos do sistema, definidos nas etapas anteriores do processo. É nessa parte em que grande parte da codificação ocorre.

Neste projeto, a construção do sistema foi feita em iterações semanais, com o desenvolvimento em paralelo das funcionalidades do aplicativo Android e do servidor REST.

3.1.4 TRANSIÇÃO

A fase de transição abrange a entrega do software, ou seja, a implementação do sistema e o uso pelos usuários finais. Nessa fase é essencial que haja o *feedback* dos usuários para que pequenas modificações sejam feitas a cada entrega. O treinamento dos usuários e a realização de testes também fazem parte deste processo.

O processo de transição do projeto Familyst deu-se por meio da apresentação do sistema ao professor orientador, bem como sua validação do sistema.

3.2 PLANO DE ATIVIDADES

O plano de atividades para o desenvolvimento do sistema, que também pode ser visto no gráfico de Gantt, disponível no apêndice 7, foi dividido segundo quatro etapas do RUP além de uma adicional: Concepção, Elaboração, Construção, Transição e Gerenciamento do Projeto. As quatro primeiras etapas referem-se ao desenvolvimento do sistema, levantamento de requisitos, modelagem do banco de dados, desenvolvimento do protótipo de telas do aplicativo e a programação, enquanto o gerenciamento do sistema, criação de diagramas e as reuniões de

alinhamento com o orientador para definir o andamento do projeto, compõem a quinta etapa do plano.

A etapa de concepção teve como principais atividades o levantamento dos requisitos do sistema, tanto do servidor REST quanto do aplicativo Android, a definição dos casos de uso, para definição das funcionalidades cruciais do sistema proposto. Nessa etapa também foi definida uma lista de riscos, contendo fatores que podem ocorrer durante o desenvolvimento do sistema, suas consequências, responsável, probabilidade, impacto e classificação.

Subsequentemente, iniciou-se a etapa de elaboração, que teve como foco tanto a parte visual do aplicativo Android, quanto à modelagem do banco de dados. Em sua primeira reunião, foi criado um protótipo de alto nível das telas do aplicativo, bem como um esboço do Diagrama de Entidade e Relacionamento, a ser discutido e melhorado ao longo do desenvolvimento do projeto e de acordo com as necessidades.

A terceira etapa, de construção, foi a mais longa do processo de desenvolvimento. Nessa parte ocorreu o desenvolvimento geral do sistema. A construção do aplicativo Android e do servidor REST ocorreram concomitantemente. Junto a elas, ocorreu também o desenvolvimento das especificações dos casos de uso, a finalização dos diagramas de Entidade e Relacionamento, Classes, Sequência e a fundamentação teórica do projeto.

A quarta etapa, de Transição, foi composta pela integração dos elementos e funcionalidades das aplicações, bem como o refinamento da parte gráfica, visando sempre a inclusão de usuários de todas as faixas etárias. Também ocorreram testes finais no projeto e a finalização da documentação e entrega do sistema.

A quinta etapa ocorreu durante todo o desenvolvimento do projeto e visava manter a organização do código, da documentação, e dos cronogramas definidos para o andamento deste. Foi composta pelas reuniões com o orientador da equipe, bem como o desenvolvimento dos documentos de análise, como o WBS, Gantt e Plano de Riscos.

3.3 PLANO DE RISCOS

Para gerenciar os riscos durante o desenvolvimento do projeto, delimitá-los e controlá-los, foi desenvolvido uma tabela apontando-os.

A tabela abaixo apresenta os detalhes e informações destes riscos, em especial as consequências, a pessoa responsável e sua ação, a probabilidade de acontecer tal risco e o seu impacto no todo.

TABELA 1 – PLANO DE RISCOS

| Nº | Condição | Data Limite | Consequência | Ação | Responsável | Probabilidade | Impacto | Classificação |
|-----------|--|------------------------|---|---|------------------------------|----------------------|----------------|----------------------|
| 1 | Mudança nos requisitos | 24/10/2016 | Atraso nas atividades definidas no cronograma | Identificar mudanças de maneira prévia | Jean | Alta | Moderado | 6 |
| 2 | Dificuldade no uso de novas tecnologias | 08/08/2016 | Atraso nas atividades, necessidade de aprendizado, mudança de tecnologia. | Considerar tempo e conhecimento da equipe para definir as tecnologias | Jefferson | Alta | Alto | 7 |
| 3 | Falhas de Comunicação | 15/11/2016 | Atraso no cronograma, realização da mesma atividade por dois membros da equipe. | Realizar iterações semanais e comunicar-se, se possível diariamente, pelos grupos em redes sociais. | Thiago | Moderada | Moderado | 5 |
| 4 | Ausência/Saída de membro da equipe | Último dia do projeto. | Atraso nas atividades, redistribuição das responsabilidades | | Toda a equipe/ Orientador | Baixa | Alto | 5 |
| 5 | Defeito em máquina que possui o software instalado | 15/11/2016 | Perda de todo o material, necessidade de refazê-lo | Realizar backups em locais separados, para restauração. | Toda a equipe | Baixa | Alto | 5 |

3.4 TECNOLOGIAS APLICADAS NO DESENVOLVIMENTO

A seguir, serão descritas as ferramentas e linguagens utilizadas nas etapas do desenvolvimento do Familyst.

3.4.1 JUSTINMIND PROTOTYPER

O Justinmind Prototyper² é um software para prototipação de aplicativos, principalmente voltados para os dispositivos móveis, de todas as plataformas. Apesar de ter-se utilizado a versão gratuita de 30 dias, a ferramenta foi considerada bastante completa, com diversos recursos gráficos e de gestos, o que permitiu obter-se uma melhor visão do aplicativo Familyst em funcionamento.

3.4.2 ANDROID STUDIO

O Android Studio³ é uma IDE para desenvolvimento de aplicativos Android baseado na IDE IntelliJ IDEA, e fornece ferramentas de desenvolvimento integradas para desenvolvimento e depuração.

3.4.3 NETBEANS

O NetBeans⁴ é uma IDE que suporta diversas linguagens de programação, dentre elas, Java, utilizada no desenvolvimento das aplicações *web* do sistema. Segundo o site da ferramenta, a IDE foi desenvolvida pela empresa Sun Microsystems e iniciou como um projeto de código aberto, tornando-se subsidiária

² Disponível em: <https://www.justinmind.com>

³ Disponível em: <https://developer.android.com/index.html>

⁴ Disponível em: <https://netbeans.org>

da empresa Oracle, em 2010. O NetBeans é amplamente utilizado no desenvolvimento de sistemas web atualmente.

Utiliza-se, neste trabalho, o NetBeans para trabalhar com toda a parte REST do sistema. A IDE fornece ferramentas para debug em tempo de execução, gerenciamento de servidor, gerenciamento da conexão com banco de dados e inserção inteligente de código para auxílio na programação do código fonte do serviço desenvolvido.

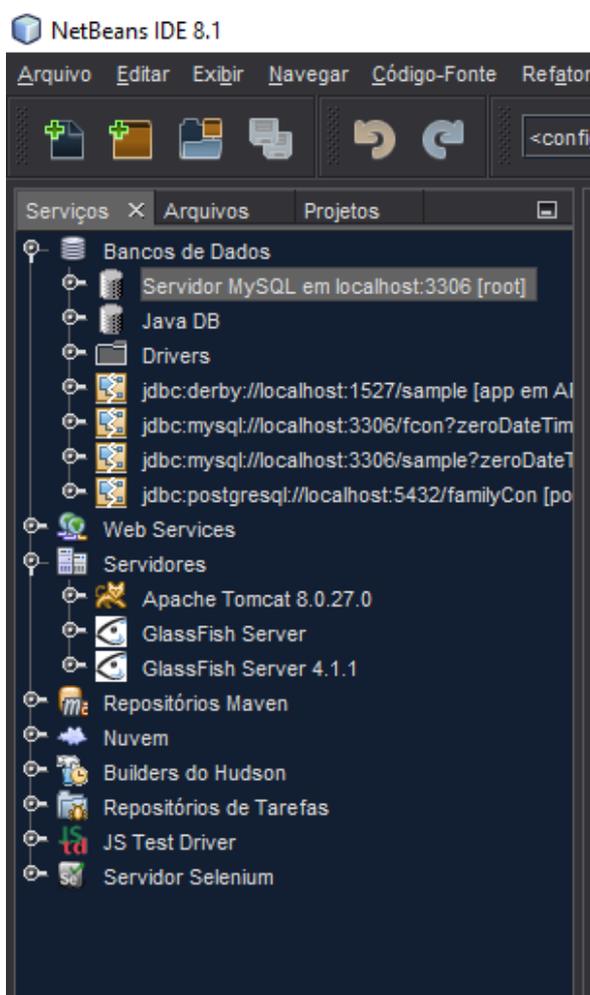


FIGURA 4 – GERENCIAMENTO DE BANCOS DE DADOS E GERENCIAMENTO DE SERVIDORES NO NETBEANS
FONTE: OS AUTORES

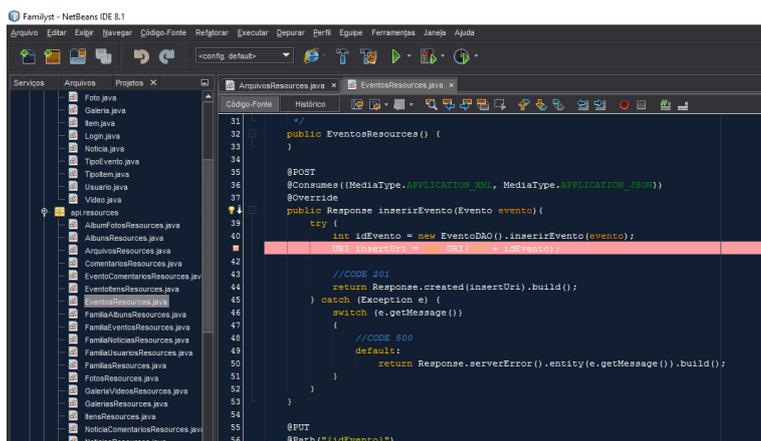


FIGURA 5 – BREAKPOINT PARA DEBUG DE CÓDIGO FONTE EM TEMPO DE EXECUÇÃO NO NETBEANS
FONTE: OS AUTORES

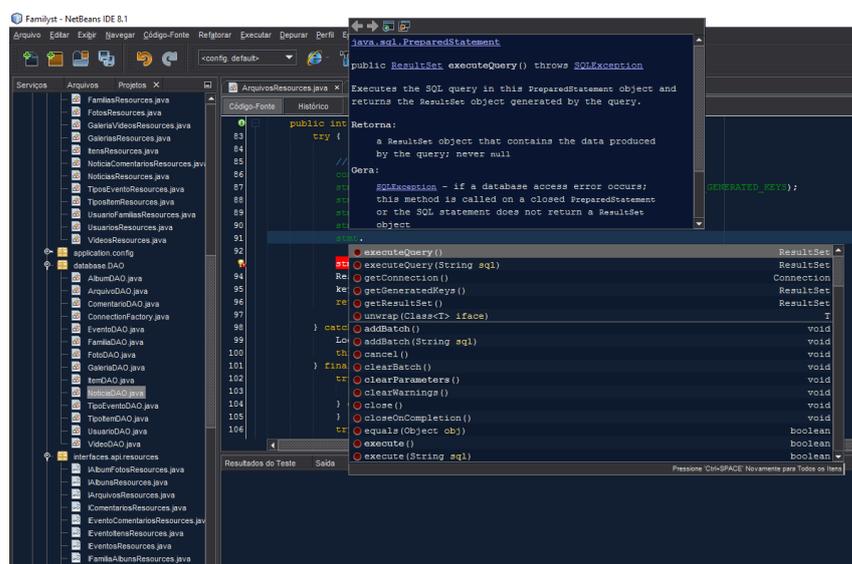


FIGURA 6 – SMART CODE COMPLETION DO NETBEANS
FONTE: OS AUTORES

3.4.3.1 SMART CODE COMPLETION

Tecnologia disponível na IDE NetBeans⁵ que sugere fragmentos de códigos para completar um código que teve seu desenvolvimento iniciado ou mesmo códigos a serem desenvolvidos.

⁵ Informações e dicas em: https://netbeans.org/kb/73/java/editor-codereference_pt_BR.html

3.4.4 MYSQL SERVER

O MySQL Server⁶ é um sistema de gerenciamento de bancos de dados relacional, ou seja, que divide seus dados em tabelas separadas e com relacionamentos entre si. É uma ferramenta *open source* que serve tanto ao gerenciamento de pequenos bancos de dados quanto de bancos gigantescos.

Neste projeto utilizou-se um conector para realizar a conexão ao MySQL chamado “mysql-connector-java”, inserido no projeto através de uma definição de dependência do Apache Maven⁷.

```
16
17     <dependencies>
18         <dependency>
19             <groupId>com.sun.jersey</groupId>
20             <artifactId>jersey-servlet</artifactId>
21             <version>1.10-b02</version>
22         </dependency>
23         <dependency>
24             <groupId>mysql</groupId>
25             <artifactId>mysql-connector-java</artifactId>
26             <version>5.1.39</version>
27         </dependency>
28         <dependency>
29             <groupId>com.sun.jersey</groupId>
30             <artifactId>jersey-json</artifactId>
31             <version>1.18.1</version>
32         </dependency>
33     </dependencies>
```

FIGURA 7 – TAG DO XML COM REFERÊNCIA AO “MYSQL-CONNECTOR-JAVA”
FONTE: OS AUTORES

3.4.5 MYSQL WORKBENCH

O MySQL Workbench⁸ é uma ferramenta para modelagem e administração de bancos de dados. Apresenta funcionalidades de desenvolvimento SQL,

⁶ Disponível em: <https://www.mysql.com>

⁷ Informações sobre o Apache Maven no tópico 3.4.7 deste documento

⁸ Disponível em: <http://www.mysql.com/products/workbench>

administração de usuários e ferramentas para configurações de servidor. Também é possível o desenvolvimento de diagrama de banco de dados e, por meio deste, a geração automática do banco. O caminho inverso do citado também é possível.

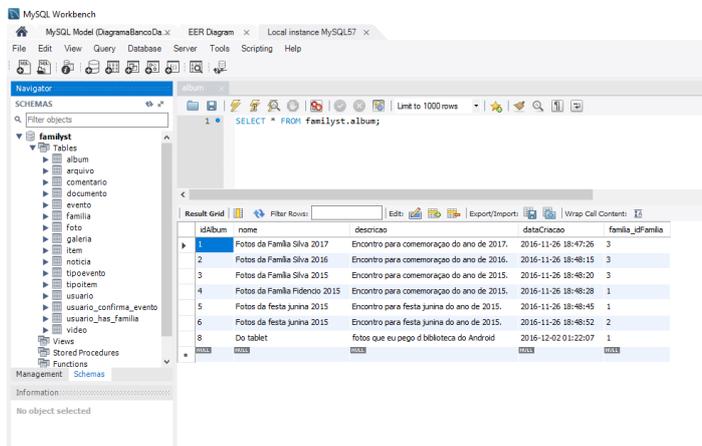


FIGURA 8 – ADMINISTRAÇÃO DO BANCO DE DADOS E TABELAS NO MYSQL WORKBENCH
FONTE: OS AUTORES

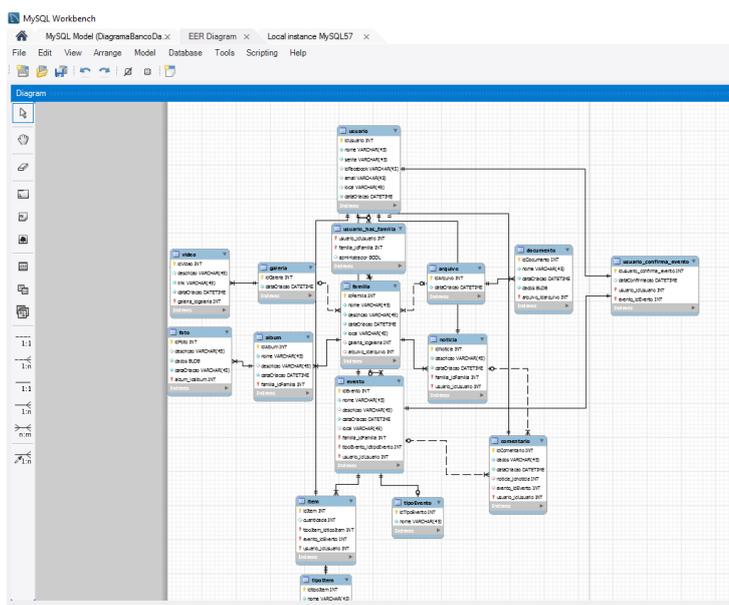


FIGURA 9 – DIAGRAMA DE BANCO DE DADOS DESENVOLVIDO NO MYSQL WORKBENCH
FONTE: OS AUTORES

3.4.6 POSTMAN

O Postman⁹ é uma extensão do Google Chrome¹⁰ que fornece uma série de ferramentas para simular um cliente de serviço web. Neste projeto foi simulado e testado todas as requisições ao nosso serviço REST através da ferramenta. O Postman fornece uma interface amigável e simples para execução de requisições HTTP com métodos como GET, POST, PUT e DELETE, incluindo *headers* adicionais. Pode-se também visualizar as respostas do servidor através do mesmo.

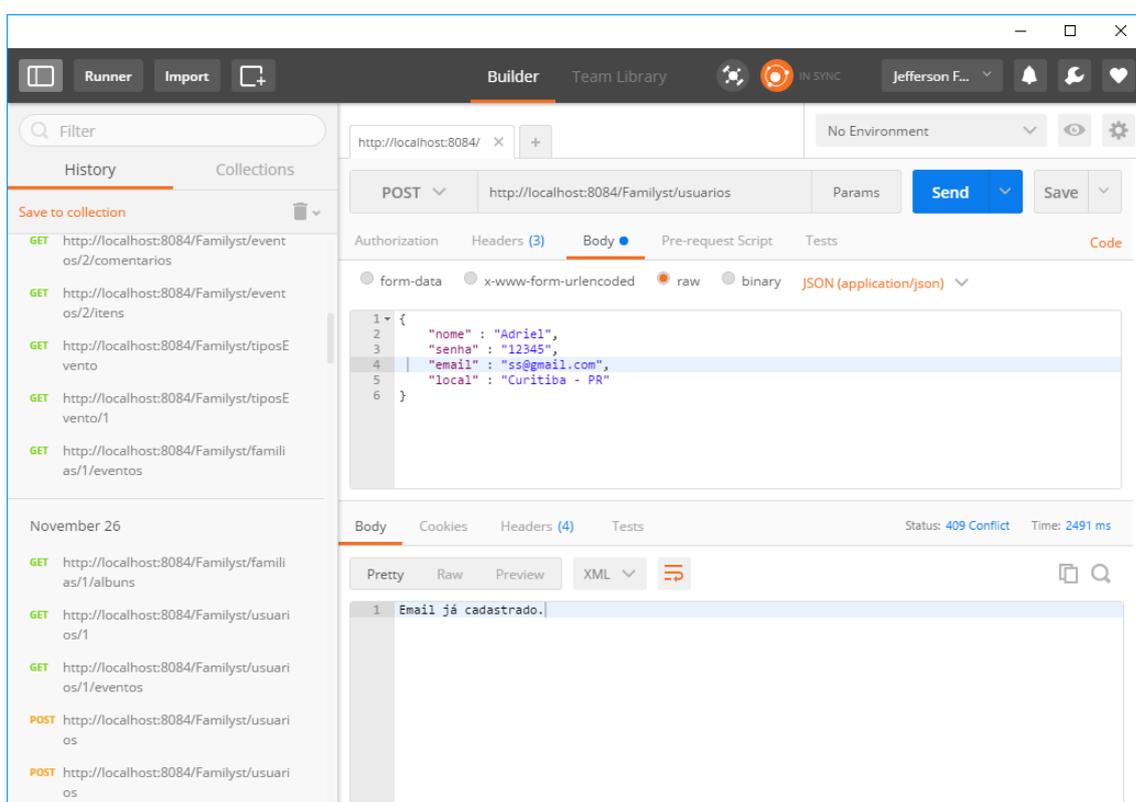


FIGURA 10 – EXECUÇÃO DE UMA REQUISIÇÃO *POST* NOS PADRÕES REST
FONTE: OS AUTORES

⁹ Disponível em: <https://www.getpostman.com>

¹⁰ Disponível em: <https://www.google.com/chrome>

3.4.7 APACHE MAVEN

O Apache Maven¹¹ é uma ferramenta para construção e gerenciamento de dependências de quaisquer projetos desenvolvidos em JAVA. Ela facilita a importação de plug-ins e APIs necessárias ao projeto.

No sistema do Familyst foi editado um arquivo de extensão XML (pom.xml) fornecido pela própria ferramenta a fim de indicar quais APIs são necessárias ao funcionamento do projeto. Dessa forma, o Maven se encarrega de baixar as versões informadas das APIs e as instalar corretamente no projeto em referência.

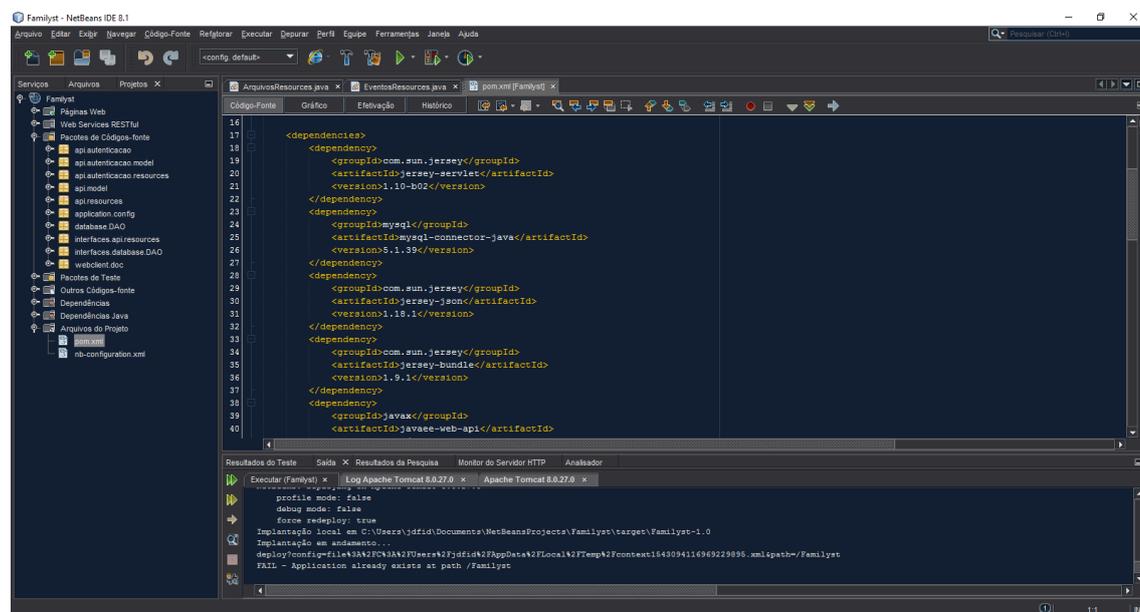


FIGURA 11 – ARQUIVO “POM.XML”
FONTE: OS AUTORES

3.4.8 APACHE TOMCAT

O software Tomcat¹², desenvolvido pela Fundação Apache, permite a execução de aplicações para web. Sua principal característica técnica é estar

¹¹ Disponível em: <https://maven.apache.org>

¹² Disponível em: <https://jersey.java.net>

centrada na linguagem de programação Java, mais especificamente nas tecnologias de Servlets e de JSPs.

O Tomcat é escrito em Java e, por isso, necessita que a versão do Java J2SE esteja instalada no mesmo computador onde ele será executado. No entanto, não basta ter a versão Runtime de Java instalada, pois o Tomcat necessita compilar (e não apenas executar) programas escritos em Java.

3.4.9 JERSEY

Jersey¹³ é uma ferramenta utilizada na implementação de *webservices* RESTful, em um container Java Servlet, permitindo a implementação das interfaces com classes predefinidas para identificar recursos do servidor. O Jersey também provê uma biblioteca, uma espécie de guia de acesso para o cliente se comunicar com o servidor. O Jersey é uma implementação referencial do JAX-RS e é responsável por interceptar e interpretar as requisições feitas em Java ao servidor.

3.4.10 JAX-RS

O JAX-RS¹⁴ é uma abreviação de Java API for RESTful Web Services e fornece ferramentas para desenvolvimento, exposição e acesso a aplicações web desenhadas e implementadas de acordo com os princípios da arquitetura REST. O JAX-RS permite que recursos sejam expostos através de anotações nos códigos fontes.

¹³ Disponível em: <https://jersey.java.net>

¹⁴ Disponível em: <https://jax-rs-spec.java.net>

```

@Path("/albums/{idAlbum}/fotos")
public class AlbumFotosResources implements IAlbumFotosResources {

    @GET
    @Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
    @Override
    public Response buscarFotosAlbum(@PathParam("idAlbum") Integer idAlbum) {
        try {
            List<Foto> fotos = new FotoDAO().listarFotosAlbum(idAlbum);
            GenericEntity<List<Foto>> fotosResponse = new GenericEntity<List<Foto>>(fotos);
            return Response.status(Response.Status.OK).entity(fotosResponse).build();
        } catch (Exception e) {
            switch (e.getMessage())
            {
                //CODE 404
                case "Album nao encontrado.":
                    return Response.status(Response.Status.NOT_FOUND).build();
                //CODE 500
                default:
                    return Response.serverError().entity(e.getMessage()).build();
            }
        }
    }
}

```

FIGURA 12 – ANOTAÇÕES JAX-RS
 FONTE: OS AUTORES

3.4.11 JAX-B

O intuito do JAX-B¹⁵ é facilitar a serialização de objetos Java. O JAX-RS utiliza o JAX-B para serializar conteúdos de requisições através de anotações do JAX-B, como @XmlRootElement e @XmlElement.

```

@XmlRootElement
public class Arquivo {

    public Arquivo(int idArquivo, Date dataCriacao) {
        this.idArquivo = idArquivo;
        this.dataCriacao = dataCriacao;
    }

    public Arquivo(){
        this.idArquivo = 0;
        this.dataCriacao = null;
    }

    @XmlElement
    private final int idArquivo;

    @XmlElement
    private final Date dataCriacao;
}

```

FIGURA 13 – ANOTAÇÕES JAX-B
 FONTE: OS AUTORES

¹⁵ Disponível em: <https://jax-rs-spec.java.net>

3.4.12 JSON

O JSON¹⁶ é um formato leve para troca de dados entre sistemas. É uma notação de fácil entendimento, tanto para o ser humano (leitura) quanto para o computador (geração e decodificação). JSON é independente de qualquer linguagem, mas possui convenções que são familiares aos programadores que utilizam linguagem C, Java, Perl, Python, entre outras.

3.4.13 REST

Segundo os padrões da arquitetura REST¹⁷ para *web services*, foram implementadas diversas funcionalidades, entre elas:

- Uso de verbos HTTP (GET, POST, PUT e DELETE)
- Nomeação e padronização das URLs dos recursos disponíveis no servidor web
 - Nomes dos recursos são substantivos e sempre no plural. Ex.: /álbuns/
 - Identificadores dos recursos padronizado. Ex.: /famílias/15
 - Hierarquia dos recursos. Ex.: famílias/15/álbuns/2/fotos
- Utilização dos códigos de resposta HTTP (mundialmente conhecidos) para indicar o estado de uma requisição
 - 200 OK – Sucesso na requisição
 - 201 CREATED – Recurso criado com sucesso
 - 204 NO CONTENT – Sucesso na requisição mas não há nada no *body* da resposta
 - 400 BAD REQUEST – Erro na requisição; requisição mal formatada
 - 401 UNAUTHORIZED – Usuário não autenticado

¹⁶ Disponível em: <http://www.json.org>

¹⁷ Disponível em: <http://www.restapitutorial.com>

- 403 FORBIDDEN – Usuário sem permissão de acesso ao recurso
- 404 NOT FOUND – Recurso solicitado não encontrado
- 405 METHOD NOT ALLOWED – Recurso existe mas não atende ao verbo HTTP requisitado
- 409 CONFLICT – Tentativa de criação ou edição de recurso já existente
- 500 INTERNAL SERVER ERROR – Erro no servidor
- Opção de conexão via JSON ou XML: permite o envio dos dados (resposta da requisição) no formato JSON ou XML, desde que informado no *header* “content-type”.

Representational State Transfer ou somente **REST**, é cada vez mais usado como alternativa ao “já antigo” SOAP onde que a principal crítica a este é a burocracia, algo que REST possui em uma escala muito menor.

REST é baseado no design do protocolo HTTP, que já possui diversos mecanismos embutidos para representar recursos como código de status, representação de tipos de conteúdo, cabeçalhos, etc.

O principal nesta arquitetura são as URLs do sistema e os *resources* (*resource* é um recurso, entidade). Ele aproveita os métodos HTTP para se comunicar, que são:

- **GET**: Solicita a representação de um determinado recurso. É definido como um método seguro e não deve ser usado para disparar uma ação (remover um usuário, por exemplo);
- **POST**: As informações enviadas no corpo (*body*) da requisição são utilizadas para criar um novo recurso. Também é responsável por fazer processamentos que não são diretamente relacionados a um recurso.
- **DELETE**: Remove um recurso. Deve retornar o status 204 caso não exista nenhum recurso para a URI especificada.
- **PUT**: Atualiza um recurso na URI especificada. Caso o recurso não exista, ele pode criar um. A principal diferença entre POST e PUT é que o primeiro pode lidar não somente com recursos, mas também pode fazer processamento de informações.

FIGURA 14 – DESCRIÇÃO REST

FONTE: [HTTP://WWW.MATERA.COM/BR/2012/10/22/COMO-FUNCIONA-UM-WEBSERVICE-REST/](http://www.matera.com/br/2012/10/22/como-funciona-um-webservice-rest/)

```
import javax.ws.rs.DELETE;
import javax.ws.rs.PUT;
import javax.ws.rs.Path;
import javax.ws.rs.GET;
```

FIGURA 15 – IMPLIMENTAÇÃO NO NETBEANS DOS VERBOS HTTP UTILIZADOS

FONTE: OS AUTORES

```
@Path("familias/{idFamilia}/usuarios")
public class FamiliaUsuariosResources implements IFamiliaUsuariosResources {
```

FIGURA 16 – EXEMPLO DE HIERARQUIA DE RECURSOS
FONTE: OS AUTORES

| Familias: Authorization: Return 401 - Unauthorized | |
|--|--|
| GET /familias | 200 - OK {List} 500 - Internal Server Error |
| GET /familias{idFamilia} | 200 - OK {familia} 404 - Not Found 500 - Internal Server Error |
| POST /familias | 201 - Created HEADER {resourcebasepath}/idFamilia 409 - Conflict 500 - Internal Server Error |
| PUT /familias{idFamilia} | 200 - OK 404 - Not Found 204 - No Content 500 - Internal Server Error |
| DELETE /familias{idFamilia} | 200 - OK 404 - Not Found 500 - Internal Server Error |

FIGURA 17 – USO DOS *STATUS CODES*
FONTE: OS AUTORES

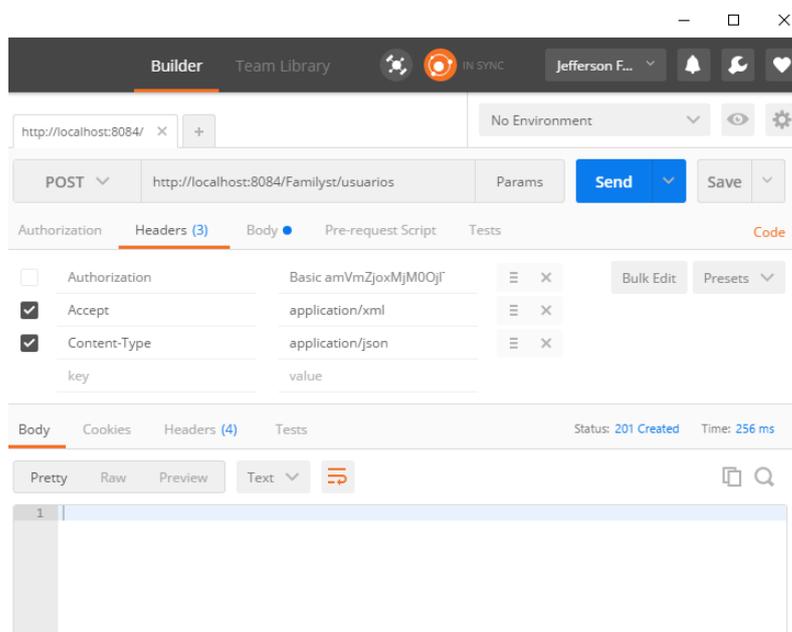


FIGURA 18 – EXEMPLO DA UTILIZAÇÃO DOS TIPOS DE *HEADERS*
FONTE: OS AUTORES

3.4.14 MVC

O padrão arquitetural MVC é uma forma de quebrar uma aplicação, ou até mesmo um pedaço da interface de uma aplicação, em três partes: o modelo, a visão e o controlador. A visão, camada *View*, é a camada de interface com usuário. A camada de modelo (*Model*) contém os elementos de dados da aplicação e a camada de controle, *Controller*, é a que recebe os comandos do usuário e as aplica realizando algum processamento, refletindo nos modelos. Esta última faz também a exibição dos dados da camada de modelo ao usuário.

No serviço REST do Familyst utiliza-se parte do padrão MVC, com algumas customizações adequadas ao projeto. A camada *Model* está separada em um *package api.model*. Já a camada controller está em *packages de resources*, pois no padrão REST todas as requisições são para recursos. Portanto, as classes de *resources* são as classes controladoras, que recebem as requisições com dados vindos da camada *View* (*Client Android*) e fazem o processamento adequado, retornando dados da camada *Model*, porém o projeto não apresenta camada *View*, uma vez considerado que esta seria o projeto cliente Android.

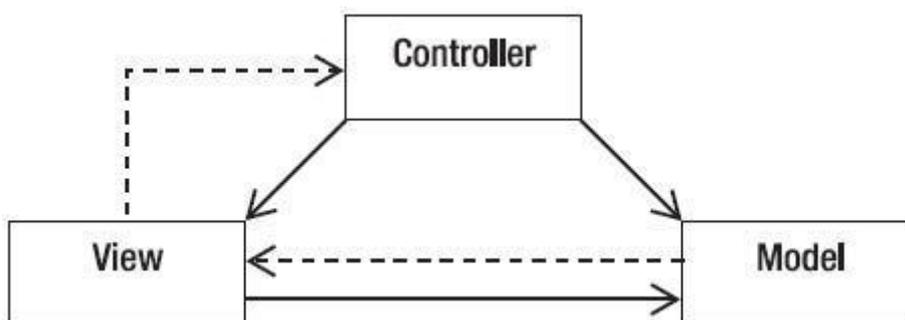


FIGURA 19 – ARQUITETURA MVC

FONTE:

MODELVIEWCONTROLLERDIAGRAM.PNG

[HTTPS://STATIC.DZONE.COM/DZ1/DZ-FILES/321PX-](https://static.dzone.com/dz1/dz-files/321px-)

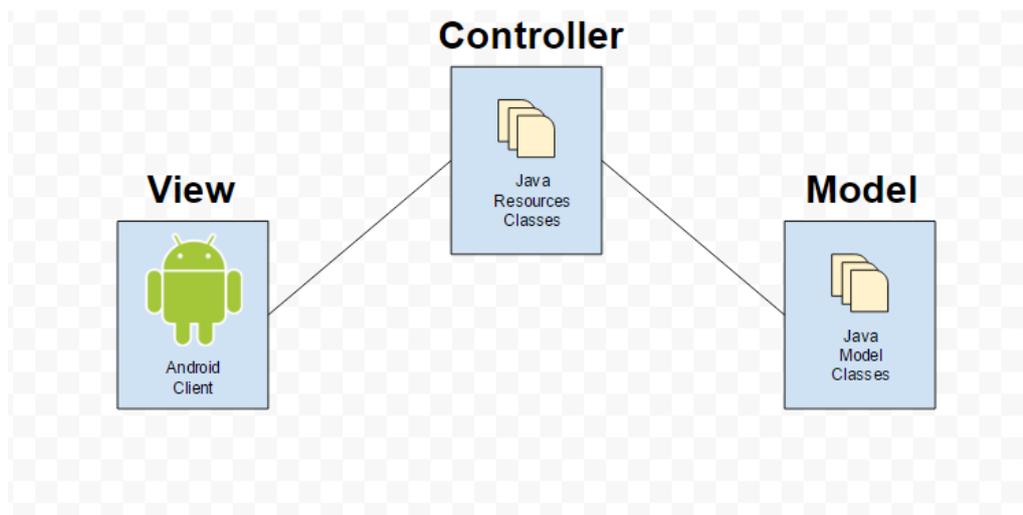


FIGURA 20 – ARQUITETURA MVC NO FAMILYST
FONTE: OS AUTORES

3.4.15 DAO PATTERN

É um padrão utilizado para separar a camada baixa de acesso aos dados da camada alta de negócios.

Seguindo este padrão, houveram implementações de interfaces das classes DAO, implementações concretas DAO e classes de modelo. Implementou-se ainda uma classe que gera a conexão com o banco de dados na qual as classes DAO se utilizam para realizar operações no banco. Todas operações no banco de dados ocorrem por meio das chamadas às classes DAO, que implementam as interfaces DAO.

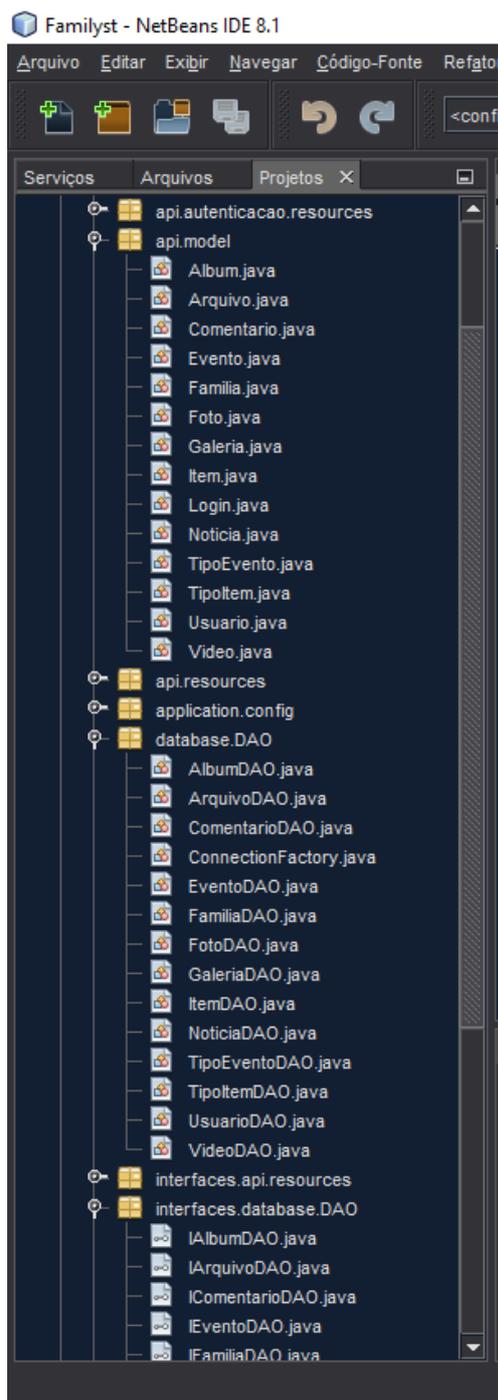


FIGURA 21 – IMPLEMENTACAO DAO PATTERN
FONTE: OS AUTORES

3.4.16 WBS TOOL

O WBS Tool¹⁸ é uma ferramenta disponibilizada gratuitamente, na plataforma *web*, para criação de diagramas WBS. A ferramenta conta com a possibilidade de armazenamento em nuvem, que pode ser privado caso o usuário tenha realizado o cadastro no site.

3.4.17 ASTAH PROFFESIONAL

O Astah¹⁹ é uma ferramenta, desenvolvida em Java, de modelagem UML. Nele é possível modelar diversos tipos de diagramas. Em resumo, é possível construir os diagramas de estruturas, diagramas de comportamentos e diagramas de interações.

Neste projeto utilizou-se o Astah para a modelagem dos seguintes diagramas (presentes nos apêndices 2, 5 e 6, respectivamente, deste documento):

- Diagrama de Casos de Uso (comportamento)
- Diagrama de Classes (estrutura)
- Diagramas de Sequência (interação)

3.4.18 GITHUB

Conhecido como a “a fábrica social de software”, o GitHub²⁰ é um serviço de *Web Hosting* Compartilhado disponível para o desenvolvimento em conjunto.

O GitHub neste projeto foi utilizado como repositório de códigos-fonte, mantendo o versionamento do desenvolvimento do sistema.

¹⁸ Disponível em: <http://www.wbstool.com>

¹⁹ Disponível em: <http://astah.net>

²⁰ Disponível em: <https://github.com>

3.4.20 GOOGLE DRIVE

Um dos serviços de hospedagem em nuvem de arquivos mais utilizados atualmente, o Google Drive²¹ permite o *upload* de arquivos para que o usuário tenha acesso em qualquer computador ou dispositivo compatível com este serviço.

Utilizou-se o Google Drive para armazenar toda a documentação do Familyst, bem como seus diversos diagramas, arquivos de análise e acompanhamento do desenvolvimento e a documentação final.

3.4.20 PACOTE OFFICE

Trata-se de uma série de ferramentas criadas pela Microsoft²², com o intuito de auxiliar na produtividade. O Word, por exemplo, é utilizado para criação e edição de documentos de texto. O PowerPoint, para a criação de slides para a apresentação e o Excel para desenvolvimento de planilhas.

3.4.21 LINGUAGENS DE PROGRAMAÇÃO

O desenvolvimento dos dois sistemas, web e mobile, ocorreram utilizando a linguagem de programação Java, tendo em vista que esta é a linguagem nativa para o desenvolvimento de aplicações Android. Para o *web service* REST, decidiu-se utilizar essa linguagem levando em consideração a experiência dos desenvolvedores na equipe, que já haviam utilizado a linguagem, e também pelo fato de ser a linguagem utilizada em grande parte das disciplinas ofertadas no curso. Isso tornou mais fácil a comunicação entre toda a equipe.

²¹ Disponível em: <https://www.google.com/intl/pt-BR/drive>

²² Disponível em: <https://products.office.com/pt-br/home>

3.4.21.1 JAVA

Java²³ é uma linguagem de programação amplamente utilizada, pois é uma linguagem gratuita, de alto desempenho e disponível para diversas plataformas (Java, 2013). A linguagem executa em uma máquina virtual chamada JVM responsável por converter os códigos de byte em linguagem de máquina. Essa máquina virtual é responsável pela aplicação Java ser capaz de ser executada em qualquer sistema operacional, sem a necessidade de adaptação do código fonte.

Há ainda a especificação Java SE²⁴ que traz suporte à tecnologia Java para a web, proporcionando a criação de Servlets (classes especiais que estendem a funcionalidade de um servidor, o que permite tratar as requisições feitas através do protocolo HTTP). No sistema REST do Familyst, todas as páginas possuem código Java e a comunicação com o banco de dados é realizada utilizando o conceito de Servlets.

3.5 RESPONSABILIDADES

Inicialmente houve a divisão do projeto em três partes: o aplicativo Android, o servidor REST e a documentação. Foi acordado entre os membros da equipe que todos estariam em constante visão e organização dos itens do projeto, porém realizou-se a divisão de responsabilidades, ocorrida da seguinte forma:

- Jean: Aplicativo Android – Criação dos protótipos de tela, definição do design do aplicativo, programação da parte navegacional e integração com o servidor.

²³ Disponível em: <https://www.oracle.com/br/java/index.html>

²⁴ Disponível em: <http://www.oracle.com/technetwork/java/javase/overview/index.html>

- Jefferson: Servidor REST – Definição de arquitetura, modelagem, criação e gerenciamento do banco de dados e integração com o aplicativo Android.
- Thiago: Documentação – Criação dos diagramas e descrição do software desenvolvido.

Também havia responsabilidades referentes a toda a equipe, como manter-se atualizado sobre o desenvolvimento da documentação do sistema, bem como possuir todos os ambientes atualizados e em funcionamento em suas máquinas.

3.6 REQUISITOS

Nesta seção será descrito o processo de levantamento de requisitos bem como a descrição dos requisitos funcionais e não-funcionais do sistema.

3.6.1 LEVANTAMENTO DE REQUISITOS

No início do projeto foi realizado o levantamento de requisitos através de uma pesquisa realizada por meio de uma entrevista com 26 pessoas, entre jovens e adultos, em geral na faixa dos 20~50 anos, no qual foram direcionadas nove perguntas, quatro questões binárias “Sim/Não”, uma de múltipla escolha e mais quatro do tipo discursiva (campo aberto para digitação). Para conhecimento das perguntas, estão disponíveis no apêndice 1 deste documento.

Em relação às questões de resposta binária, tivemos as seguintes conclusões:

- 100% dos entrevistados se reúnem com suas famílias em eventos.
- Quase 90% dos entrevistados utilizam um serviço online ou um aplicativo local dedicado para armazenagem e gerenciamento de fotos, vídeos e documentos.

- Dos 90% do tópico acima, mais de 50% dos entrevistados utilizam o Facebook para armazenar e gerenciar fotos.
- 85% dos entrevistados utilizaria um novo serviço eletrônico social para guardar fotos e documentos familiares

Nas questões discursivas tivemos alguns fatos relevantes levantados, sendo os principais problemas daqueles que utilizam um serviço já existente: falta de privacidade, espaço de armazenamento limitado e difícil organização das fotos/documentos (layout do sistema).

3.6.2 REQUISITOS FUNCIONAIS

Com base no levantamento de requisitos foram definidos no início do projeto os requisitos funcionais, ou seja, as principais funcionalidades essenciais do sistema a ser desenvolvido. Na tabela abaixo segue cada requisito com suas descrições e nível de prioridade.

| Código | Requisito | Descrição | Prioridade |
|---------------|--------------------------|---|-------------------|
| RF1 | Cadastro de Usuário | O sistema deve permitir apenas o cadastro de usuários que possuem convite enviado via e-mail. | Alta |
| RF2 | Gerenciamento de Perfil | Um usuário pode editar e desativar seu perfil quando lhe for conveniente. | Alta |
| RF3 | Convite de Usuário | Um usuário pode enviar convites para que outros usuários também tenham acesso ao sistema. | Alta |
| RF4 | Cadastro de Família | Um usuário pode cadastrar uma família | Alta |
| RF5 | Gerenciamento de Família | Um usuário pode ativar e desativar uma família quando achar conveniente, bem como convidar e | Alta |

| | | | |
|-----|--------------------------------|---|------|
| | | remover usuários. | |
| RF6 | Cadastro de Evento | O sistema deve permitir o cadastro de eventos. | Alta |
| RF7 | Gerenciamento de Eventos | Um usuário pode editar um evento (local, itens e convite de usuários) | Alta |
| RF8 | Postar Atualizações – Notícias | O sistema deve permitir que os usuários postem notícias e atualizações. | Alta |

TABELA 1 – REQUISITOS FUNCIONAIS
 FONTE: OS AUTORES

3.6.3 REQUISITOS NÃO FUNCIONAIS

A tabela a seguir revela os requisitos relacionados ao uso do sistema, não necessariamente direcionados aos usuários, mas sim voltados à melhoria do sistema em si, seja na usabilidade, segurança e até desempenho. Abaixo apresenta-se tais requisitos, com suas descrições e nível de prioridade.

| Código | Requisito | Descrição | Prioridade |
|---------------|--|--|-------------------|
| RNF1 | Disponibilidade do Servidor | O servidor deve estar disponível e em pleno funcionamento durante 100% do tempo. | Média |
| RNF2 | Criptografia de Senhas | Todas as senhas de usuários devem possuir criptografia. | Alta |
| RNF3 | Linguagem de programação e arquitetura | A linguagem utilizada para o desenvolvimento de todas as ferramentas será JAVA, utilizando também a arquitetura MVC. | Alta |
| RNF4 | Desempenho | O desempenho e a qualidade do sistema deverão ser garantidos através da realização de testes unitários. | Alta |
| RNF5 | Testabilidade | O sistema deve ter sua qualidade assegurada por meio da realização | Alta |

| | | | |
|--|--|-----------|--|
| | | de testes | |
|--|--|-----------|--|

TABELA 2 – REQUISITOS NÃO FUNCIONAIS
FONTE: OS AUTORES

4. APRESENTAÇÃO DO SOFTWARE

Neste tópico encontram-se especificadas as funcionalidades de cada interface pertencentes ao sistema Familyst.

4.1 LOGIN (TELA INICIAL)

Ao abrir o Familyst a primeira tela exibida é a tela de login. Esta tela tem a funcionalidade de permitir o acesso do usuário ao sistema por meio do login, caso o usuário já tenha realizado um cadastro. Se for o primeiro acesso, o usuário deverá utilizar a opção “Não possui login? Cadastre-se!” e realizar o seu cadastro. Nesta tela há também a opção de recuperação de senha, caso o usuário tenha esquecido a mesma, através da opção “Esqueceu sua senha?”.

Familyst

Familyst

Email

Senha

Esqueceu a senha?

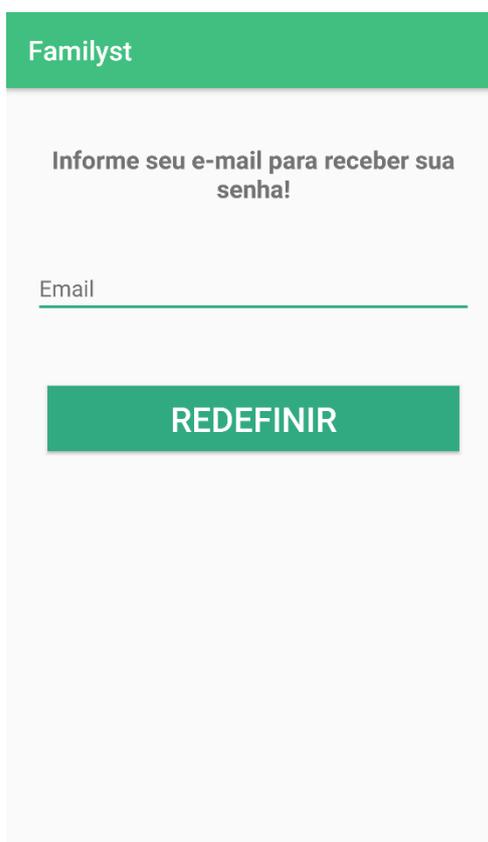
ENTRAR

Não possui login? Cadastre-se!

FIGURA 22 – TELA DE LOGIN
FONTE: OS AUTORES

4.2 RECUPERAÇÃO DE SENHA

Uma vez feito o cadastro no Familyst, caso o usuário venha a esquecer sua senha, haverá uma opção de recuperação de senha na tela de login. Nesta tela será solicitado o e-mail de cadastro do usuário (confirmação do e-mail), para o qual será enviada a senha de acesso.

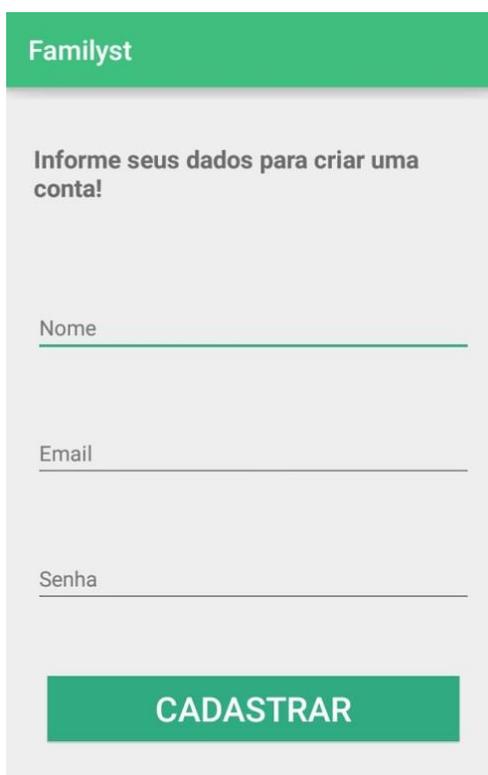


A imagem mostra a interface de usuário para a recuperação de senha no aplicativo Familyst. No topo, há uma barra verde com o nome 'Familyst' em branco. Abaixo, o texto 'Informe seu e-mail para receber sua senha!' é exibido em uma fonte preta. Segue-se um campo de entrada rotulado 'Email' com uma linha verde de base. Na base da tela, um botão verde com o texto 'REDEFINIR' em branco está centralizado.

FIGURA 23 – TELA DE RECUPERAÇÃO DE SENHA
FONTE: OS AUTORES

4.3 CADASTRO

A tela de cadastro no sistema Familyst solicita ao usuário que ele informe seus dados nos campos respectivos (nome, e-mail e senha). Em seguida, o usuário deverá clicar em “CADASTRAR” a fim de que seja efetuado o cadastro com sucesso.



A imagem mostra a interface de usuário para o cadastro no sistema Familyst. No topo, há uma barra verde com o nome "Familyst" em branco. Abaixo, o texto "Informe seus dados para criar uma conta!" é exibido em negrito. Seguem três campos de entrada: "Nome", "Email" e "Senha", cada um com uma linha de texto verde. No rodapé, um botão verde com o texto "CADASTRAR" em branco está centralizado.

FIGURA 24 – TELA DE CADASTRO DE USUÁRIO
FONTE: OS AUTORES

4.3 MENU

Feito o login, o usuário é redirecionado para a tela de Notícias. Porém, através do Menu é possível acessar todos os recursos do Familyst (Eventos, Famílias, Álbuns, Fotos, Galerias, Vídeos, Membros da Família, etc.). Para acessar o Menu basta somente deslizar o dedo da esquerda para a direita na parte esquerda da tela, conforme a imagem abaixo. O Menu pode ser acessado em qualquer tela do sistema, que não seja de cadastro.

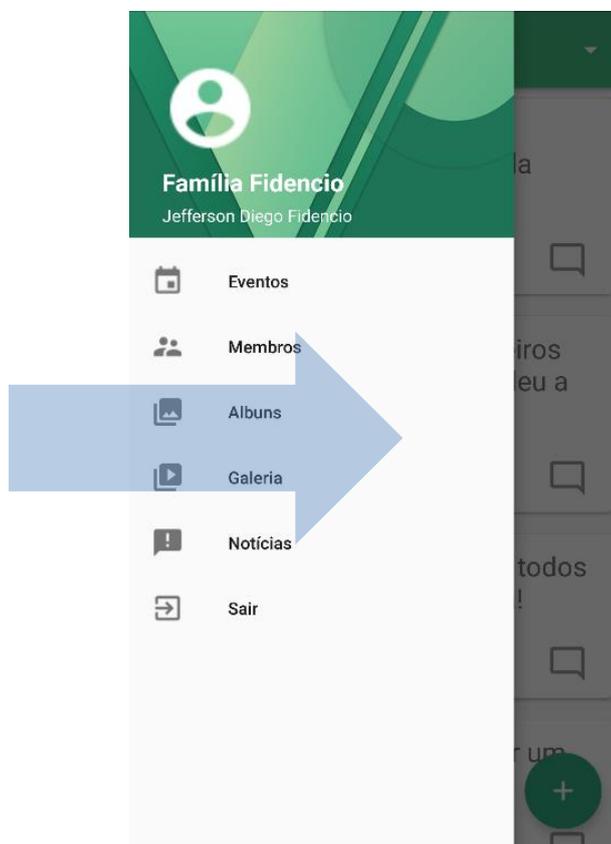


FIGURA 25 – TELA DE MENU
FONTE: OS AUTORES

4.4 NOTÍCIAS

A tela de notícias é exibida após o usuário efetuar o login. Nesta tela aparecerão tópicos de publicações, os quais poderão ser visualizados e comentados pelos demais membros da família.

Caso o usuário pertença a mais de uma família, no canto superior direito há uma flecha que dá a opção de selecionar as famílias das quais o usuário é membro. As notícias exibidas são referentes somente à família selecionada.

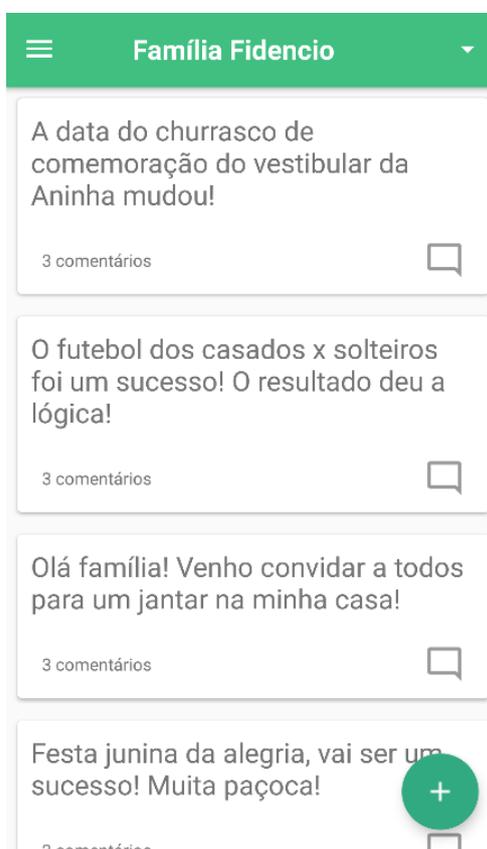


FIGURA 26 – TELA DE NOTÍCIAS
FONTE: OS AUTORES

Clicando em alguma notícia o usuário irá para a tela de detalhes da notícia, onde aparecerá os seus respectivos comentários. Na parte inferior da tela o usuário poderá inserir um novo comentário relacionado à notícia.

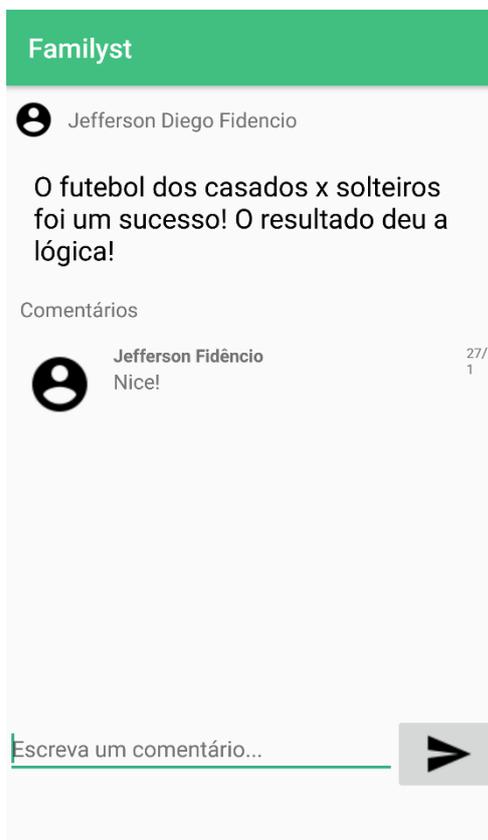
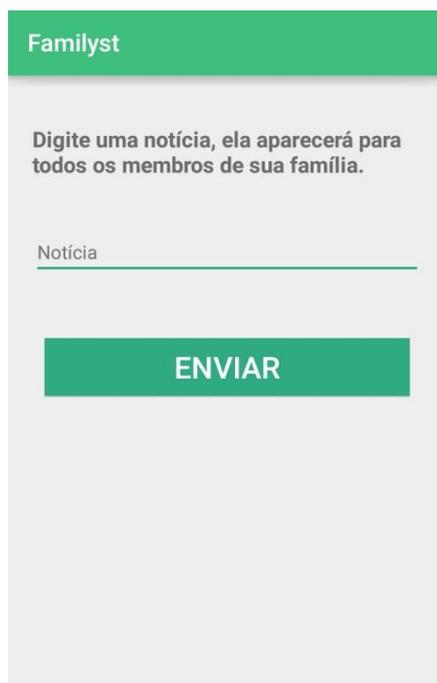


FIGURA 27 – TELA DE DETALHES DA NOTÍCIA
FONTE: OS AUTORES

Na tela principal de notícias, também é possível inserir uma nova publicação. O usuário deverá clicar no ícone “+” localizado no canto inferior direito da tela e então será levado à tela de adicionar notícia. Após escrever a notícia no respectivo campo, basta somente clicar no botão “ENVIAR” para finalizar a publicação. Essa notícia será exibida somente para a família já selecionada na tela das notícias.



The image shows a mobile application interface for 'Familyst'. At the top, there is a green header bar with the text 'Familyst' in white. Below the header, the main content area has a light gray background. It contains the instruction 'Digite uma notícia, ela aparecerá para todos os membros de sua família.' in bold black text. Underneath this instruction is a text input field with the placeholder text 'Notícia' and a green underline. At the bottom of the input area, there is a prominent green button with the white text 'ENVIAR'.

FIGURA 28 – TELA PARA ADICIONAR NOTÍCIA
FONTE: OS AUTORES

4.5 EVENTOS

Ao selecionar a opção “Eventos” no menu do Familyst, aparecerá ao usuário a tela de eventos. Nesta tela estão listados os próximos eventos que ocorrerão para a família selecionada na caixa de listagem na parte superior da tela (igual tela de notícias). Nesta tela é possível visualizar os detalhes de cada evento, selecionando tal evento, bem como criar um novo evento.

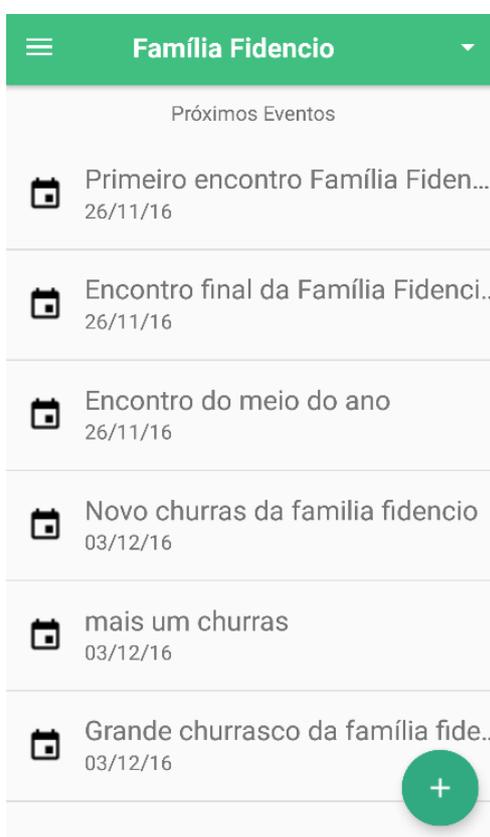


FIGURA 29 – TELA DE EVENTOS
FONTE: OS AUTORES (2016)

Ao selecionar um evento contido na lista, o usuário é direcionado à tela na qual se encontra os detalhes do evento e comentários feitos pelos demais membros da família. Nesta tela aparecerá o nome do usuário que criou o evento e detalhes como: descrição, local, uma lista com os itens necessários para o evento e a quantidade de cada item que será levado.

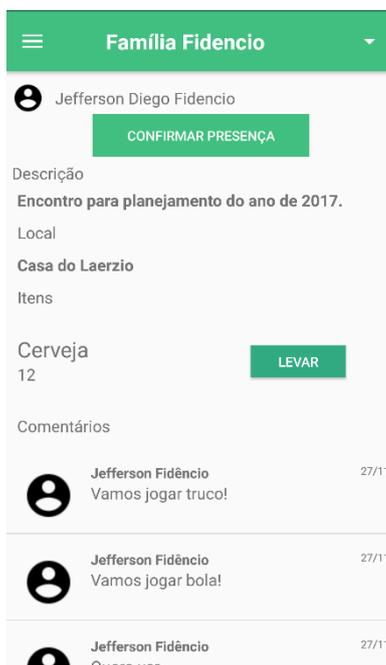


FIGURA 30 – TELA DE DETALHES DO EVENTO
 FONTE: OS AUTORES

Após clicar no ícone de adicionar evento, localizado no canto inferior direito da tela de eventos, aparecerá os campos a serem preenchidos pelo usuário. É solicitado apenas o nome, descrição e o local do evento.

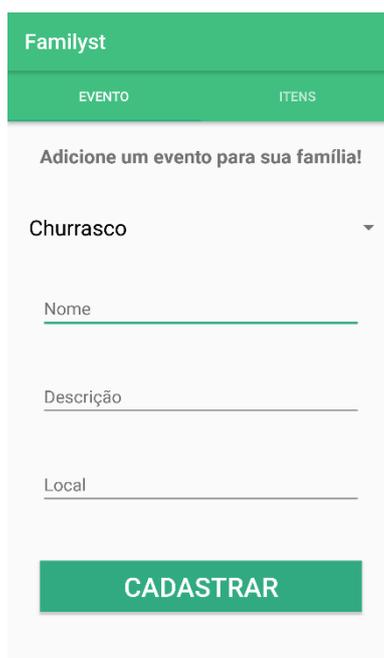


FIGURA 31 – TELA CRIAR EVENTOS
 FONTE: OS AUTORES (2016)

4.6 FAMÍLIA

A família é um grupo de pessoas (usuários do Familyst) para representar uma família real. Um usuário pode pertencer a mais de uma família, e, através destes grupos, marcar eventos, postar notícias, fotos e vídeos.

No menu do Familyst há a opção “Membros”, onde são mostrados todos os usuários pertencentes àquela família. Nesta tela encontram-se listados todos os membros da família a qual o usuário está adicionado. Semelhante à tela de notícias, na parte superior da tela há a opção do usuário selecionar a família para visualizar os respectivos membros.

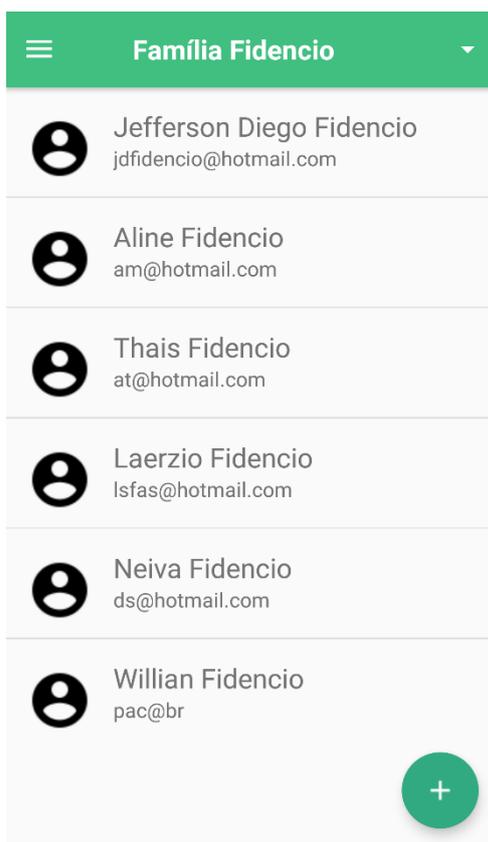


FIGURA 32 – TELA DE MEMBROS
FONTE: OS AUTORES (2016)

No canto inferior direito há o botão para inserir um usuário já cadastrado no Familyst como membro da família. Clicando neste botão, o usuário irá para a tela de cadastro de membro, bastando inserir somente o e-mail cadastrado da pessoa que será a nova integrante da família para concluir tal ação.



The image shows a mobile application interface for 'Familyst'. At the top, there is a green header bar with the text 'Familyst'. Below the header, the main content area is white and contains the following elements: a status bar at the very top showing system icons and '80%' battery; the text 'Informe o email do membro de sua família!' in bold; a text input field with the placeholder 'Email'; and a large green button with the text 'CADASTRAR' in white capital letters.

FIGURA 33 – TELA DE CADASTRO DE MEMBRO NA FAMILIA
FONTE: OS AUTORES

4.7 ÁLBUNS (FOTOS)

No menu do Familyst há uma opção chamada “Álbuns”. Os Álbuns são “pastas” criadas para armazenar diversas fotos, podendo ser criadas por qualquer usuário/membro da família. Cada álbum é criado para determinada família, e dentro dele, cada membro desta família poderá inserir fotos.

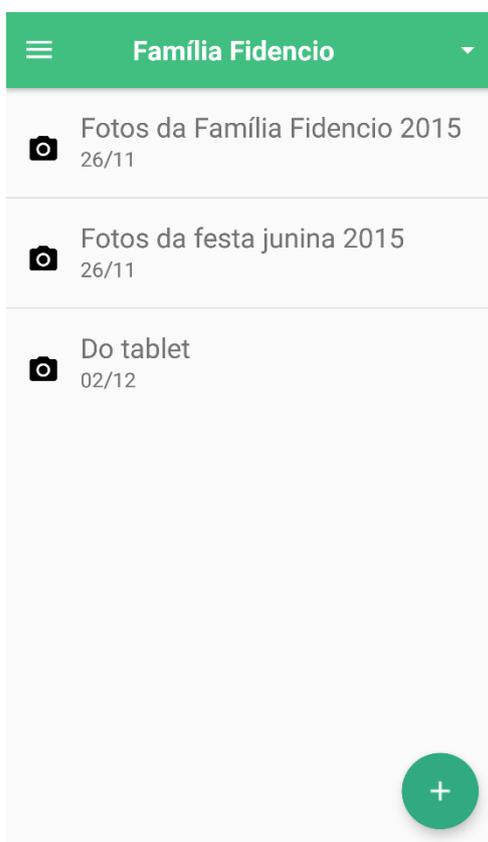


FIGURA 34 – TELA DE ÁLBUNS
FONTE: OS AUTORES

Para criar um novo álbum basta clicar no botão “+” no canto inferior direito da tela, e na tela que irá aparecer, informar o nome e uma breve descrição do novo álbum.



Familyst

Cadastro de Álbum

Nome

Descrição

CADASTRAR

FIGURA 35 – TELA DE CADASTRO DE ÁLBUM
FONTE: OS AUTORES

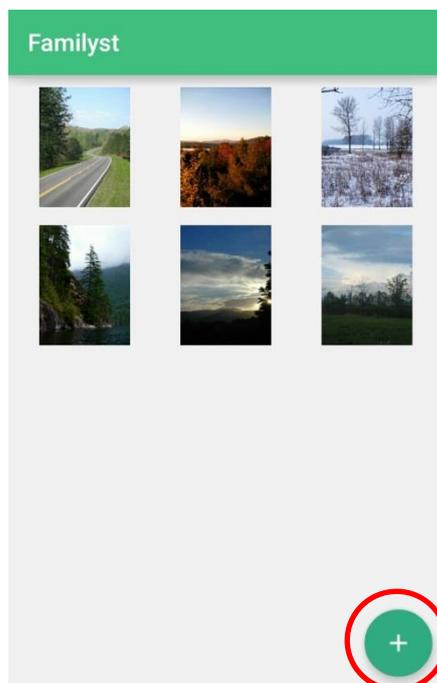


FIGURA 36 – TELA DE FOTOS
FONTE: OS AUTORES

4.8 GALERIAS (VÍDEOS)

A seção Galeria do Familyst é utilizada para armazenar vídeos, podendo ter vários vídeos inseridos por qualquer usuário/membro da família.

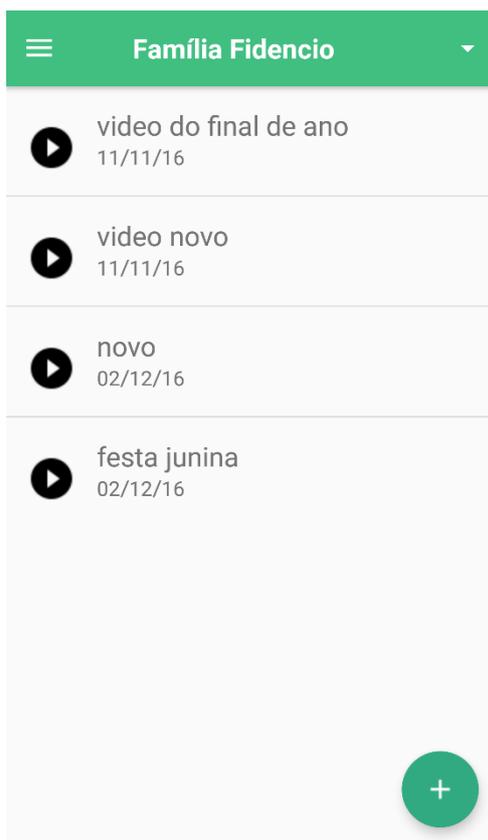


FIGURA 37 – TELA DE GALERIAS
FONTE: OS AUTORES

Para inserir um novo vídeo basta clicar no botão “+” no canto inferior direito da tela. Só poderão ser salvos vídeos que sejam *online*, ou seja, para inserir um vídeo em determinada galeria é necessário informar o nome do vídeo e o *link* do endereço do vídeo na tela de cadastro de vídeo.

Familyst

Cadastro de Video

Nome

Cole aqui o link do vídeo

CADASTRAR

FIGURA 38 – TELA DE CADASTRO DE VÍDEO
FONTE: OS AUTORES

5. CONSIDERAÇÕES FINAIS

Neste trabalho foram apresentados os detalhes do desenvolvimento da aplicação e dos serviços do Familyst. Em resumo, trata-se de uma solução composta por um servidor REST e um aplicativo Android, voltado a todas as faixas etárias, com o intuito de organizar eventos familiares e facilitar a comunicação entre membros de uma família. Durante o projeto procurou-se identificar e abranger todos os pontos que fariam com que o aplicativo fosse intuitivo para a maioria das pessoas.

5.1 DESAFIOS

Para que o projeto ocorresse foi necessário que os membros da equipe despendessem de tempo para dominar as ferramentas que seriam utilizadas no desenvolvimento, a fim de que o sistema fosse desenvolvido da melhor forma possível. De fato, o tempo foi um dos principais obstáculos superados pela equipe, que, a cada iteração com o orientador, procurava estabelecer datas anteriores à entrega do projeto, para garantir que, caso ocorresse algo que não foi planejado, a equipe tivesse tempo para fazer o necessário para o andamento do projeto.

Outro desafio foi desenvolver o ambiente permitindo a interação das pessoas de idade avançada e que, no geral, fosse de fácil aprendizado, sem complicações e dificuldades no uso do sistema. Isso se deu por conta da utilização do Material Design no desenvolvimento da aplicação em Android.

Por fim, destacou-se como maior desafio atender as necessidades do cliente, representado pelo professor orientador. Para que o prazo fosse cumprido, foi necessário alterar alguns requisitos ou modificar alguns detalhes, o que resultou em refatoração de código, porém sempre visando manter as funcionalidades principais do sistema, bem como a construção de um software com qualidade aceitável.

5.2 IMPLEMENTAÇÕES FUTURAS

Com relação ao trabalho futuro no Familyst, graças ao servidor REST, almeja-se o desenvolvimento de aplicativos para outras plataformas, como o sistema operacional da Apple (iOS) e até mesmo uma aplicação *web*, permitindo a integração entre as diversas plataformas.

Também visa-se realizar melhorias no layout do aplicativo, tornando-o cada vez mais intuitivo e inclusivo, acompanhando as evoluções das versões dos sistemas operacionais *mobiles*.

5.3 CONCLUSÃO

Ao fim deste projeto, alcançou-se o desenvolvimento do sistema planejado e concluiu-se que os desafios encontrados no decorrer foram um grande motivador para continuar seu desenvolvimento. A cada iteração com o orientador, foi possível observar a evolução, não só do projeto em si, mas de cada membro da equipe.

A experiência adquirida com as plataformas e linguagens utilizada foi de grande proveito à equipe, sempre interessada em sistemas que oferecem mobilidade e inclusão.

REFERÊNCIAS

GOMES, H. Smartphone passa PC e vira aparelho nº 1 para acessar a internet no Brasil <<http://g1.globo.com/tecnologia/noticia/2016/04/smartphone-passa-pc-e-vira-aparelho-n-1-para-acessar-internet-no-brasil.html>> Acesso em 08/11/2016.

ARAÚJO, R.R. Web e acessibilidade na terceira idade: a utilização de recursos tecnológicos para melhoria no acesso das informações na Internet <https://monografias.ufrn.br/jspui/bitstream/1/727/3/RayssaRA_Monografia.pdf> Acesso em 08/11/2016.

COUTINHO, Gustavo Leuzinger. A Era dos Smartphones: Um estudo Exploratório sobre o uso dos SmartPhones no Brasil <http://bdm.unb.br/bitstream/10483/9405/1/2014_GustavoLeuzingerCoutinho.pdf> Acesso em 07/11/2016.

FONTANA, Rafaela Mantovani. Planejamento de Projetos. Plano de Custos, Riscos e Comunicação. Universidade Federal do Paraná.

SISTEMA DE BIBLIOTECAS DA UFPR. Orientação para normalização de trabalhos acadêmicos. <<http://www.portal.ufpr.br/normalizacao.html>> Acesso em 13/11/2016.

IBM. Engenharia de Software RUP <<http://www.ibm.com/software/awdtools/rup/>> Acesso em 10/08/2016.

SALEH, Naíma. A tecnologia está afetando as relações familiares dentro da sua casa? <<http://revistacrescer.globo.com/Familia/Rotina/noticia/2014/12/tecnologia-esta-afetando-relacoes-familiares-dentro-da-sua-casa.html>> Acesso em 08/11/2016.

TOMAZELLI, Idiana. Acesso à internet avança entre idosos, pessoas sem instrução e de baixa renda. <<http://economia.estadao.com.br/noticias/geral,acesso-a-internet-avanca-entre-idosos-pessoas-sem-instrucao-e-de-baixa-renda,1678148>> Acesso em 08/11/2016.

OLIVEIRA, Nielmar. IBGE: expectativa de vida dos brasileiros aumentou mais de 40 anos em 11 décadas. <<http://agenciabrasil.ebc.com.br/geral/noticia/2016-08/ibge-expectativa-de-vida-dos-brasileiros-aumentou-mais-de-75-anos-em-11>> Acesso em 09/11/2016.

IGBE – Instituto Brasileiro de Geografia e Estatística. **Em 2014, esperança de vida ao nascer era de 75,2 anos.** IGBE, Brasília, 2014. Disponível em:<<http://saladeimprensa.ibge.gov.br/noticias?view=noticia&id=1&busca=1&idnoticia=3045>>. Acesso em 25/11/2016

IGBE – Instituto Brasileiro de Geografia e Estatística. PNAD – Pesquisa Nacional Por Amostra de Domicílios, <ano>. Acesso em <data>.

Tecmundo. **Material Design: um olhar aprofundado sobre o novo estilo visual da Google.**< <https://www.tecmundo.com.br/google/58278-material-design-olhar-aprofundado-novo-estilo-visual-google.htm>>. Tecmundo, 2014. Acesso em 26/11/2016.

APÊNDICE 1 – QUESTIONÁRIO PARA USUÁRIOS

1 - Você se reúne com sua família em eventos?

2 - Você utiliza algum serviço na internet, ou aplicativo, para guardar fotos e documentos da sua família?

3 - (Responda se SIM na pergunta 2) Qual serviço você utiliza?

4 - (Responda se SIM na pergunta 2) Tem algum pró ou contra sobre o serviço selecionado na questão 3?

5- (Responda se NÃO na pergunta 2) Como você guarda as fotos e documentos da sua família hoje?

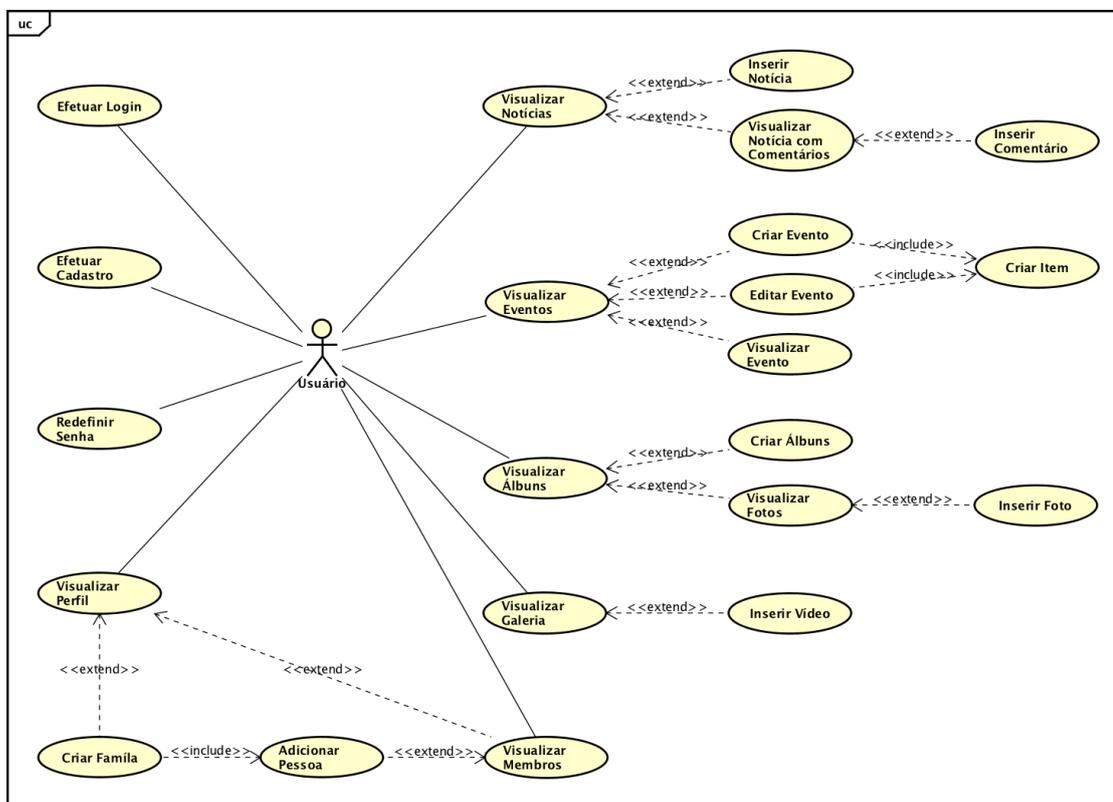
6 - Você utilizaria um novo serviço na internet, ou aplicativo, para guardar fotos e documentos da sua família, onde qualquer membro da família pudesse colaborar?

7 - (Responda se SIM na pergunta 1) Como vocês organizam o evento e dizem aos membros da família a respeito do mesmo?

8 - Você gostaria de visualizar uma árvore genealógica da sua família no seu celular?

9 - Deixe sugestões que você gostaria que o serviço oferecesse (funcionalidades)

APÊNDICE 2 – DIAGRAMA DE CASOS DE USO



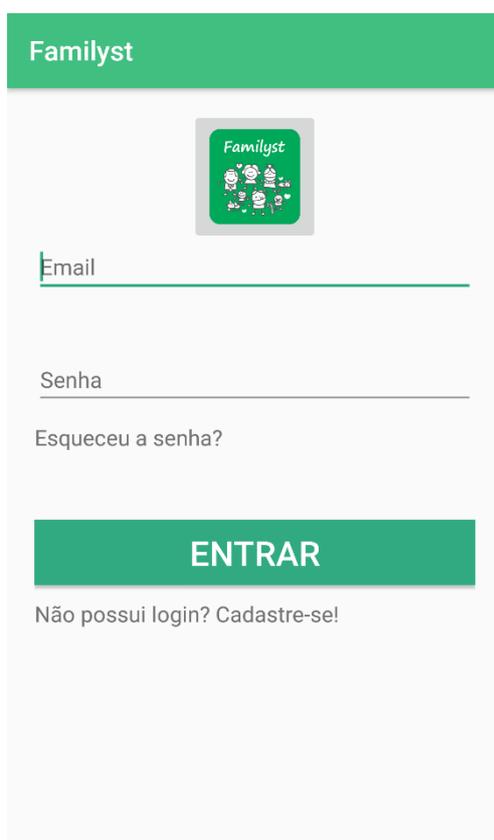
APÊNDICE 3 – ESPECIFICAÇÃO DE CASOS DE USO

UC01 – EFETUAR LOGIN

DESCRIÇÃO

Este caso de uso efetua o login do usuário no sistema.

DATA VIEW



Familyst

Familyst

Email

Senha

Esqueceu a senha?

ENTRAR

Não possui login? Cadastre-se!

DV1 – Tela de login

ATOR PRINCIPAL

Usuário.

PRÉ-CONDIÇÕES

Não há pré-condições para este caso de uso.

PÓS-CONDIÇÕES

Após o fim normal deste caso de uso o sistema deve:

1. Autenticar o login do usuário
2. Redirecionar o usuário para a página de notícias (UC03 – Visualizar Notícias)

FLUXO DE EVENTOS PRINCIPAL

1. O sistema apresenta a tela **(DV1)**
2. O usuário preenche o campo “E-mail” **(A1)**
3. O usuário preenche o campo “Senha” **(A2)**
4. O usuário pressiona o botão “ENTRAR”
5. O caso de uso é encerrado

FLUXOS ALTERNATIVOS

A1) Usuário não possui cadastro

1. O usuário clica em “Não possui login? Cadastre-se!”
2. O usuário é redirecionado para a página de cadastro. (UC02 – Efetuar Cadastro)

A2) Usuário esqueceu a senha

1. O usuário clica em “Esqueceu a senha?”
2. O sistema apresenta a tela de recuperação de senha
3. O usuário preenche o campo “E-mail”
4. O usuário pressiona o botão “ENVIAR”

FLUXOS DE EXCEÇÃO

Não há fluxo de exceção para este caso de uso.

REGRAS DE NEGÓCIO

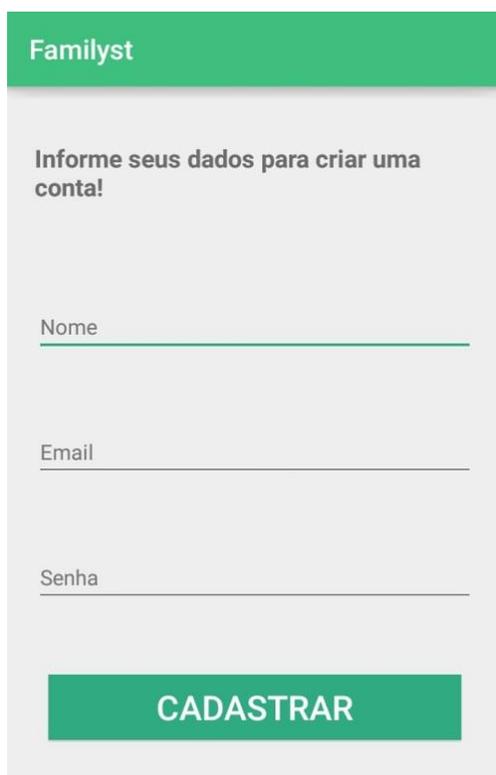
Não há regra de negócio para este caso de uso.

UC02 – EFETUAR CADASTRO

DESCRIÇÃO

Este caso de uso efetua o cadastro do usuário no sistema.

DATA VIEW



Familyst

Informe seus dados para criar uma conta!

Nome

Email

Senha

CADASTRAR

DV1 – Tela de cadastro

ATOR PRINCIPAL

Usuário.

PRÉ-CONDIÇÕES

O usuário deve ter executado o UC01 – Efetuar Login.

PÓS-CONDIÇÕES

Após o fim normal deste caso de uso o sistema deve:

1. Autenticar e armazenar o cadastro do usuário;
2. Redirecionar o usuário para a página de login (UC01 – Efetuar Login)

FLUXO DE EVENTOS PRINCIPAL

1. O sistema apresenta a tela **(DV1)**
2. O usuário preenche o campo “Nome”
3. O usuário preenche o campo “E-mail” **(E1)**
4. O usuário preenche o campo “Senha”
5. O usuário pressiona o botão “CADASTRAR”
6. O caso de uso é encerrado

FLUXOS ALTERNATIVOS

Não há fluxo alternativo para este caso de uso.

FLUXOS DE EXCEÇÃO

E1) E-mail já cadastrado.

1. O sistema exibe a mensagem “Já existe um cadastro com este e-mail. Informe outro e-mail. ”

REGRAS DE NEGÓCIO

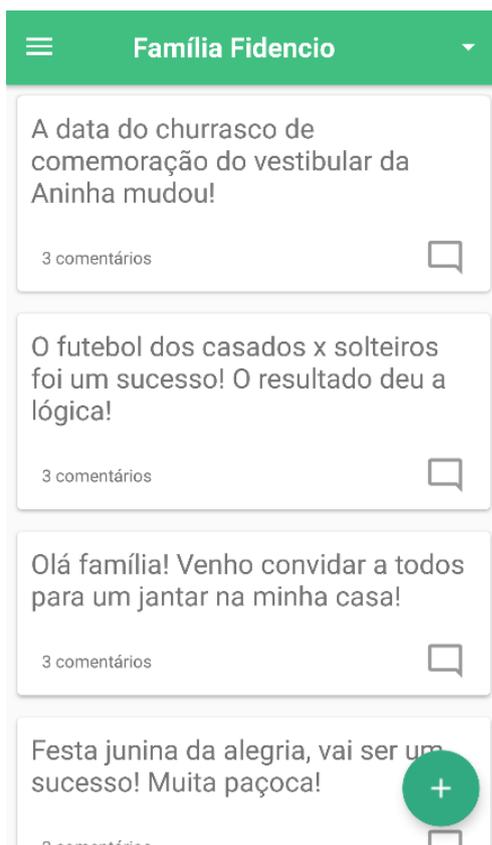
Não há regra de negócio para este caso de uso.

UC03 – VISUALIZAR NOTÍCIAS

DESCRIÇÃO

Este caso de uso descreve a tela para visualizar todas as notícias publicadas em determinada família.

DATA VIEW



DV1 – Tela de notícias

ATOR PRINCIPAL

Usuário.

PRÉ-CONDIÇÕES

O usuário deve ter executado o UC01 – Efetuar Login e estar logado no sistema.

PÓS-CONDIÇÕES

Não há pós-condições para este caso de uso.

FLUXO DE EVENTOS PRINCIPAL

1. O sistema apresenta a tela **(DV1)**
2. O usuário clica em uma notícia **(A1) (A2)**
3. [O caso de uso é encerrado]

FLUXOS ALTERNATIVOS

A1) Selecionar família

1. O usuário pressiona o ListBox de famílias
2. O usuário seleciona uma família
3. O sistema apresenta as notícias relacionadas à família selecionada pelo usuário

A2) Publicar nova notícia

1. O usuário pressiona o botão “+”
2. O usuário é redirecionado para a tela de publicação (UC05 – Publicar Notícia)

FLUXOS DE EXCEÇÃO

Não há fluxo de exceção para este caso de uso.

REGRAS DE NEGÓCIO

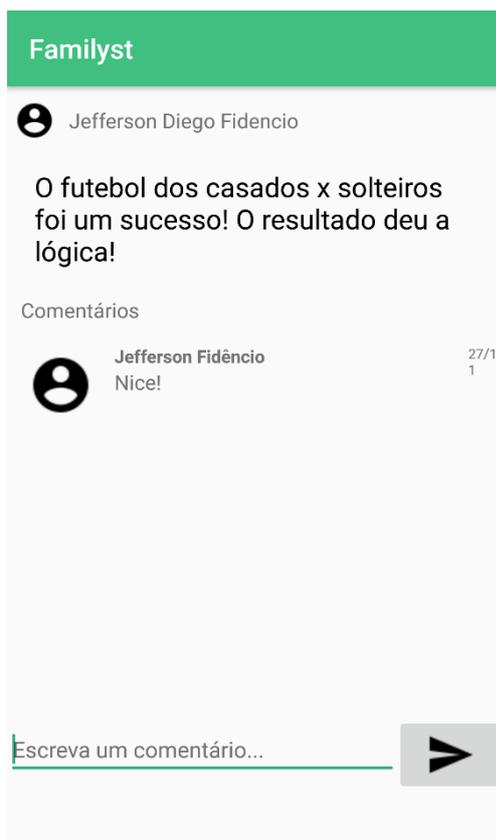
Não há regra de negócio para este caso de uso.

UC04 – COMENTAR NOTÍCIA

DESCRIÇÃO

Este caso de uso descreve a tela para visualizar determinada notícia e seus respectivos comentários.

DATA VIEW



DV1 – Tela de detalhes da notícia

ATOR PRINCIPAL

Usuário.

PRÉ-CONDIÇÕES

O usuário deve ter executado o UC03 – Visualizar Notícias e clicado em alguma notícia.

PÓS-CONDIÇÕES

Salvar o comentário da respectiva notícia.

FLUXO DE EVENTOS PRINCIPAL

1. O sistema apresenta a tela **(DV1)**
2. O usuário preenche o campo “Escreva um comentário...”
3. O usuário pressiona o botão de enviar

FLUXOS ALTERNATIVOS

Não há fluxo de alternativo para este caso de uso.

FLUXOS DE EXCEÇÃO

Não há fluxo de exceção para este caso de uso.

REGRAS DE NEGÓCIO

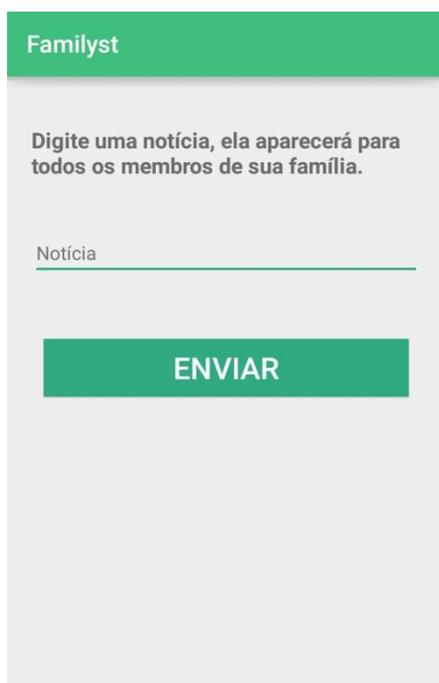
Não há regra de negócio para este caso de uso.

UC05 – PUBLICAR NOTÍCIA

DESCRIÇÃO

Este caso de uso descreve a tela para publicar notícia.

DATA VIEW



A imagem mostra a interface de usuário para publicar uma notícia no aplicativo 'Familyst'. No topo, há uma barra verde com o nome 'Familyst'. Abaixo, um texto instrutivo diz: 'Digite uma notícia, ela aparecerá para todos os membros de sua família.' Segue um campo de entrada com o rótulo 'Notícia' e uma linha verde de base. Abaixo do campo, há um botão verde com o texto 'ENVIAR' em branco.

DV1 – Tela de publicação

ATOR PRINCIPAL

Usuário.

PRÉ-CONDIÇÕES

O usuário deve ter executado o UC03 – Visualizar Notícias.

PÓS-CONDIÇÕES

O sistema deve publicar a notícia e voltar para a tela de notícias (UC03 – Visualizar Notícias).

FLUXO DE EVENTOS PRINCIPAL

1. O sistema apresenta a tela **(DV1)**
2. O usuário preenche o campo “Notícia”
3. O usuário pressiona o botão “ENVIAR”

4. O caso de uso é encerrado

FLUXOS ALTERNATIVOS

Não há fluxo alternativo para este caso de uso.

FLUXOS DE EXCEÇÃO

Não há fluxo de exceção para este caso de uso.

REGRAS DE NEGÓCIO

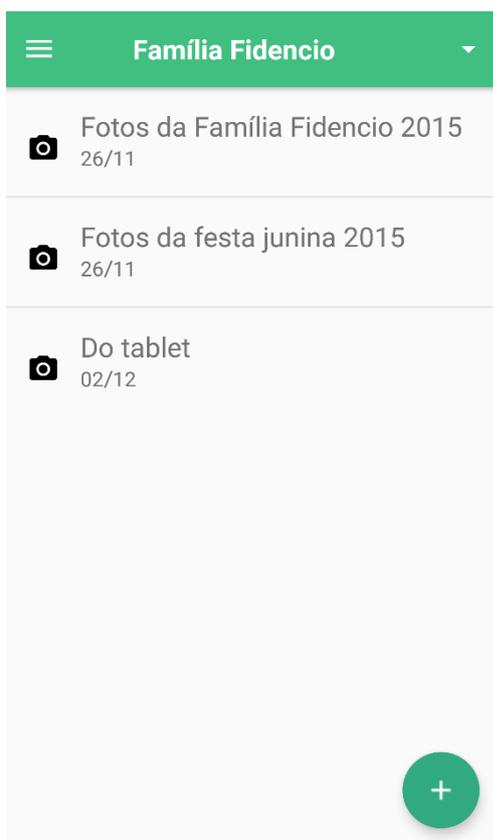
Não há regra de negócio para este caso de uso.

UC06 – VISUALIZAR ÁLBUNS

DESCRIÇÃO

Este caso de uso descreve a tela para visualizar todos os álbuns publicados em determinada família, assim como as fotos publicadas dentro do álbum.

DATA VIEW



DV1 – Tela de álbuns

ATOR PRINCIPAL

Usuário.

PRÉ-CONDIÇÕES

O usuário deve ter executado o UC01 – Efetuar Login e estar logado no sistema.

PÓS-CONDIÇÕES

Não há pós-condições para este caso de uso.

FLUXO DE EVENTOS PRINCIPAL

1. O sistema apresenta a tela **(DV1)**
2. O usuário clica em um álbum **(A1) (A2)**
3. [O caso de uso é encerrado]

FLUXOS ALTERNATIVOS

A1) Selecionar família

1. O usuário pressiona o ListBox de famílias
2. O usuário seleciona uma família
3. O sistema apresenta os álbuns relacionados à família selecionada pelo usuário

A2) Cadastrar álbum

1. O usuário pressiona o botão “+”
2. O usuário é redirecionado para a tela de cadastro de álbum (UC07 – Cadastrar Álbum)

FLUXOS DE EXCEÇÃO

Não há fluxo de exceção para este caso de uso.

REGRAS DE NEGÓCIO

Não há regra de negócio para este caso de uso.

UC07 – CADASTRAR ÁLBUM

DESCRIÇÃO

Este caso de uso descreve a tela para cadastrar um novo álbum.

DATA VIEW



Familyst

Cadastro de Álbum

Nome

Descrição

CADASTRAR

DV1 – Tela de cadastro de álbum

ATOR PRINCIPAL

Usuário.

PRÉ-CONDIÇÕES

O usuário deve ter executado o UC06 – Visualizar Álbuns.

PÓS-CONDIÇÕES

O sistema deve cadastrar o álbum e voltar para a tela de álbuns (UC06 – Visualizar Álbuns).

FLUXO DE EVENTOS PRINCIPAL

1. O sistema apresenta a tela **(DV1)**
2. O usuário preenche o campo “Nome”
3. O usuário preenche o campo “Descrição”
4. O usuário pressiona o botão “CADASTRAR”
5. O caso de uso é encerrado

FLUXOS ALTERNATIVOS

Não há fluxo alternativo para este caso de uso.

FLUXOS DE EXCEÇÃO

Não há fluxo de exceção para este caso de uso.

REGRAS DE NEGÓCIO

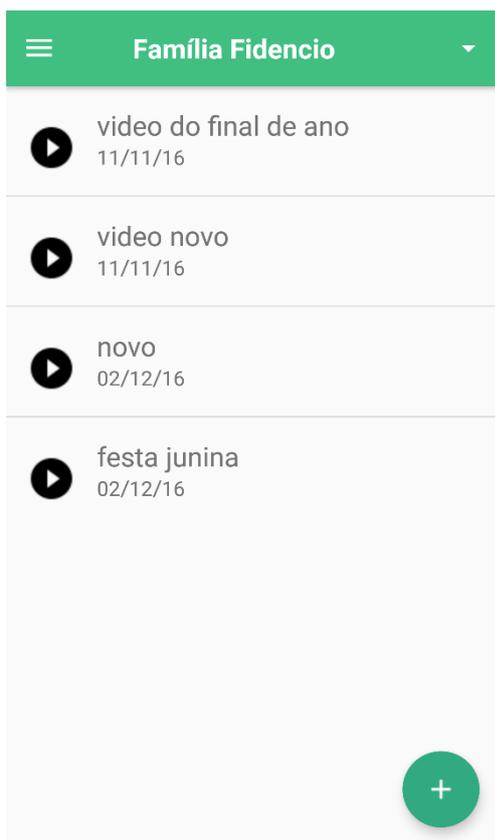
Não há regra de negócio para este caso de uso.

UC08 – VISUALIZAR VÍDEOS DA GALERIA

DESCRIÇÃO

Este caso de uso descreve a tela para visualizar todos os vídeos publicados em determinada família.

DATA VIEW



DV1 – Tela de galerias

ATOR PRINCIPAL

Usuário.

PRÉ-CONDIÇÕES

O usuário deve ter executado o UC01 – Efetuar Login e estar logado no sistema.

PÓS-CONDIÇÕES

Não há pós-condições para este caso de uso.

FLUXO DE EVENTOS PRINCIPAL

4. O sistema apresenta a tela **(DV1)**
5. O usuário clica em um vídeo **(A1) (A2)**
6. O usuário é redirecionado para o site onde o vídeo está hospedado
7. [O caso de uso é encerrado]

FLUXOS ALTERNATIVOS

A1) Selecionar família

4. O usuário pressiona o ListBox de famílias
5. O usuário seleciona uma família
6. O sistema apresenta os álbuns relacionados à família selecionada pelo usuário

A2) Cadastrar vídeo

3. O usuário pressiona o botão “+”
4. O usuário é redirecionado para a tela de cadastro de vídeo (UC09 – Cadastrar Vídeo)

FLUXOS DE EXCEÇÃO

Não há fluxo de exceção para este caso de uso.

REGRAS DE NEGÓCIO

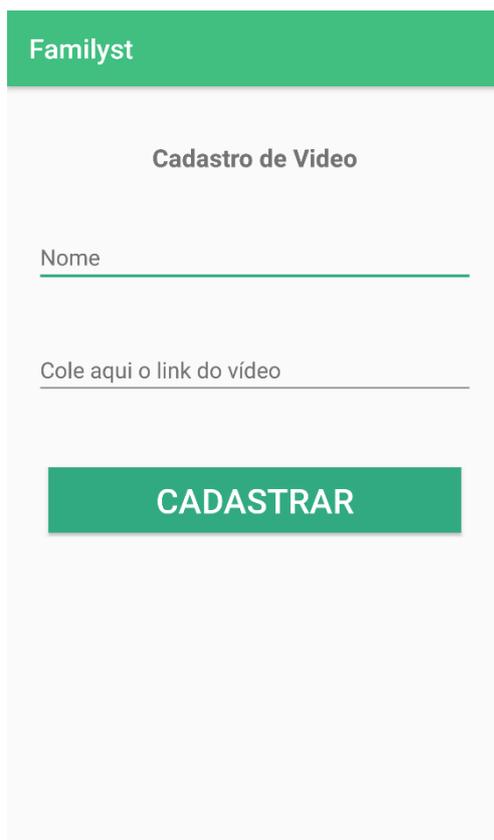
Não há regra de negócio para este caso de uso.

UC09 – CADASTRAR VÍDEO

DESCRIÇÃO

Este caso de uso descreve a tela para cadastrar um vídeo na galeria de uma determinada família.

DATA VIEW



Familyst

Cadastro de Video

Nome

Cole aqui o link do vídeo

CADASTRAR

DV1 – Tela de cadastro de vídeo

ATOR PRINCIPAL

Usuário.

PRÉ-CONDIÇÕES

O usuário deve ter executado o UC08 – Visualizar Vídeos da Galeria.

PÓS-CONDIÇÕES

O sistema deve armazenar as informações do vídeo cadastrado e voltar para a galeria (UC08 – Visualizar Vídeos da Galeria)

FLUXO DE EVENTOS PRINCIPAL

1. O sistema apresenta a tela **(DV1)**
2. O usuário preenche o campo “Nome”
3. O usuário preenche o campo “Cole aqui o link do vídeo”
4. O usuário pressiona o botão “CADASTRAR”
5. O caso de uso é encerrado

FLUXOS ALTERNATIVOS

Não há fluxo alternativo para este caso de uso.

FLUXOS DE EXCEÇÃO

Não há fluxo de exceção para este caso de uso.

REGRAS DE NEGÓCIO

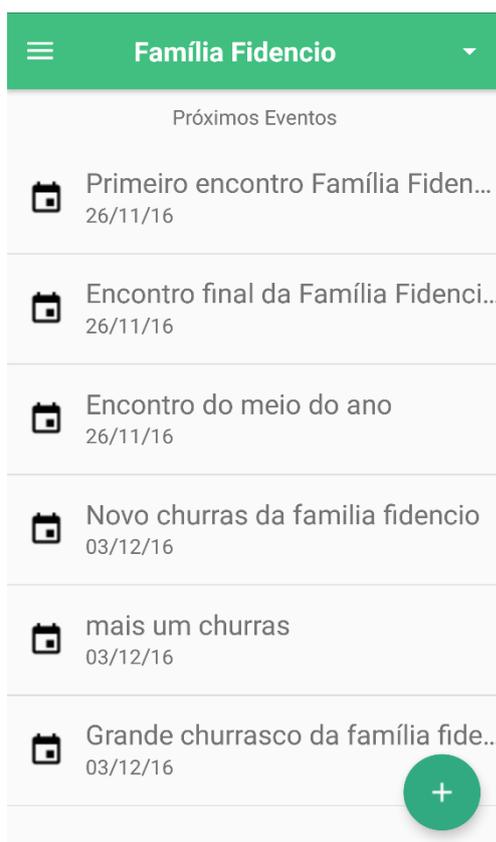
Não há regra de negócio para este caso de uso.

UC10 – VISUALIZAR EVENTOS

DESCRIÇÃO

Este caso de uso descreve a tela para visualizar todos os eventos criados para determinada família.

DATA VIEW



DV1 – Tela de eventos

ATOR PRINCIPAL

Usuário.

PRÉ-CONDIÇÕES

O usuário deve ter executado o UC01 – Efetuar Login e estar logado no sistema.

PÓS-CONDIÇÕES

Não há pós-condições para este caso de uso.

FLUXO DE EVENTOS PRINCIPAL

1. O sistema apresenta a tela **(DV1)**
2. O usuário clica em um evento **(A1) (A2)**
3. [O caso de uso é encerrado]

FLUXOS ALTERNATIVOS

A1) Selecionar família

1. O usuário pressiona o ListBox de famílias
2. O usuário seleciona uma família
3. O sistema apresenta as notícias relacionadas à família selecionada pelo usuário

A2) Criar novo evento

1. O usuário pressiona o botão “+”
2. O usuário é redirecionado para a tela de publicação (UC11 – Cadastrar Evento)

FLUXOS DE EXCEÇÃO

Não há fluxo de exceção para este caso de uso.

REGRAS DE NEGÓCIO

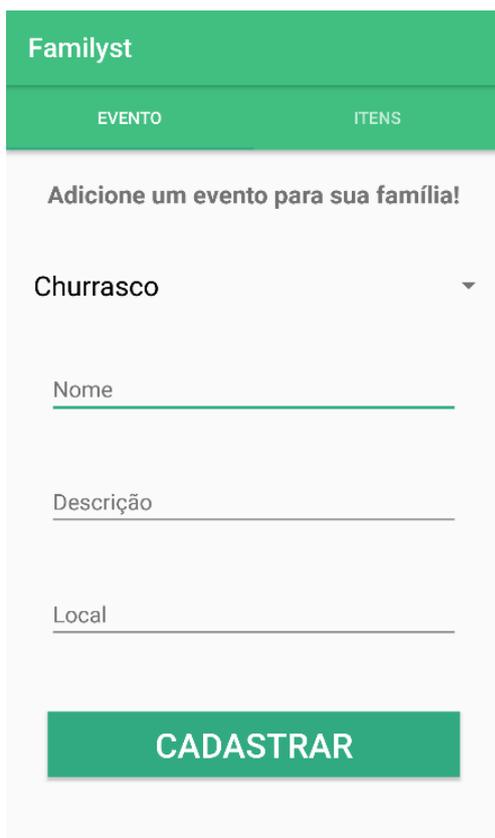
Não há regra de negócio para este caso de uso.

UC11 – CADASTRAR EVENTO

DESCRIÇÃO

Este caso de uso descreve a tela para visualizar todos os eventos criados para determinada família.

DATA VIEW



The screenshot displays the 'Familyst' application interface for event registration. At the top, a green header bar contains the app name 'Familyst' and two navigation tabs: 'EVENTO' and 'ITENS'. Below the header, a prompt reads 'Adicione um evento para sua família!'. A dropdown menu is open, showing 'Churrasco' as the selected event type. Below the dropdown are three text input fields labeled 'Nome', 'Descrição', and 'Local'. At the bottom of the screen is a prominent green button with the text 'CADASTRAR' in white capital letters.

DV1 – Tela de cadastro de evento

ATOR PRINCIPAL

Usuário.

PRÉ-CONDIÇÕES

O usuário deve ter executado o UC10 – Visualizar Eventos.

PÓS-CONDIÇÕES

O sistema deve publicar o evento e voltar para a tela de eventos (UC10 – Visualizar Eventos).

FLUXO DE EVENTOS PRINCIPAL

1. O sistema apresenta a tela **(DV1)**
2. O usuário preenche o campo “Nome”
3. O usuário preenche o campo “Descrição”
4. O usuário preenche o campo “Local”
5. O usuário pressiona o botão “CADASTRAR”
6. O caso de uso é encerrado

FLUXOS ALTERNATIVOS

Não há fluxo alternativo para este caso de uso.

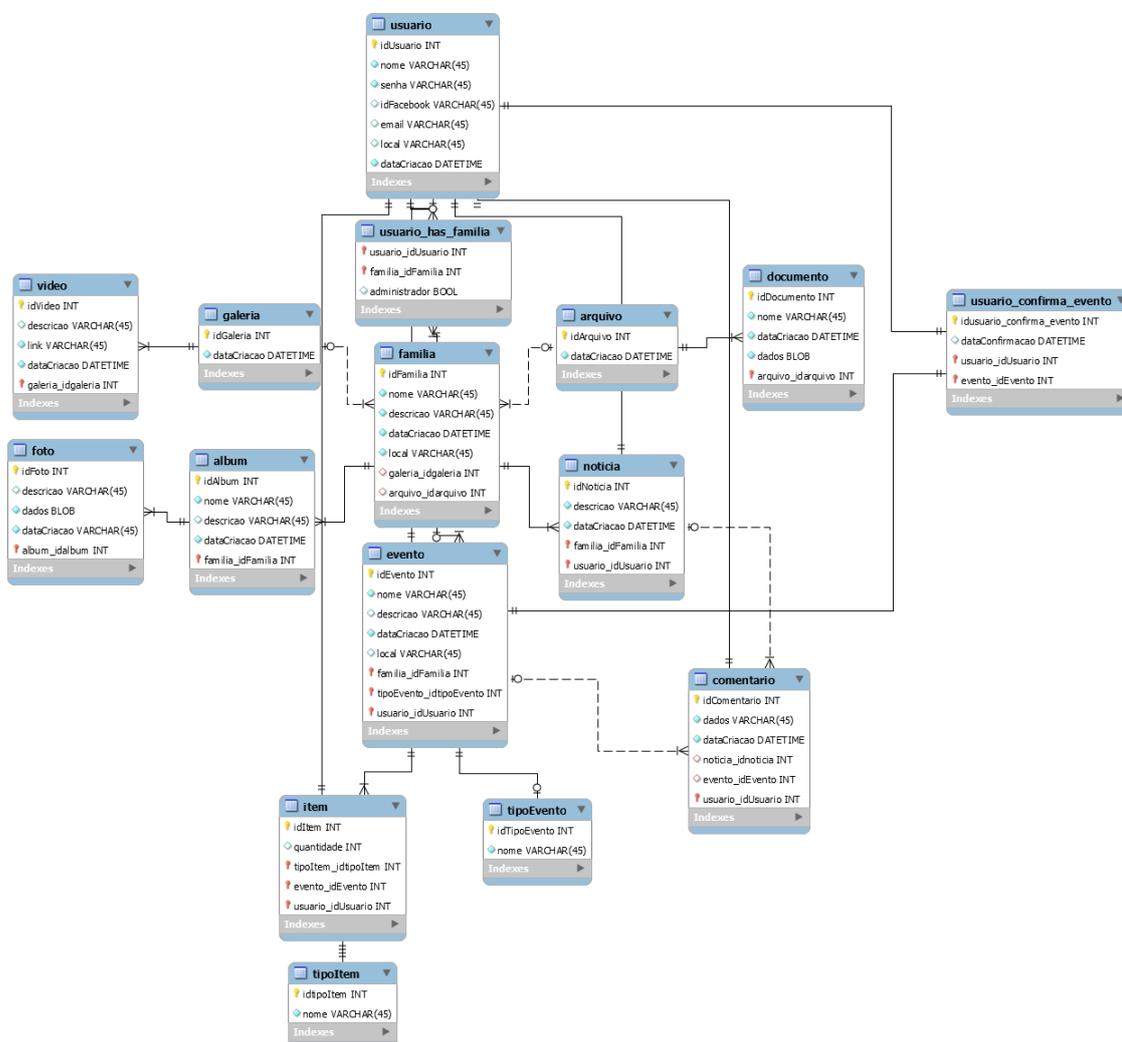
FLUXOS DE EXCEÇÃO

Não há fluxo de exceção para este caso de uso.

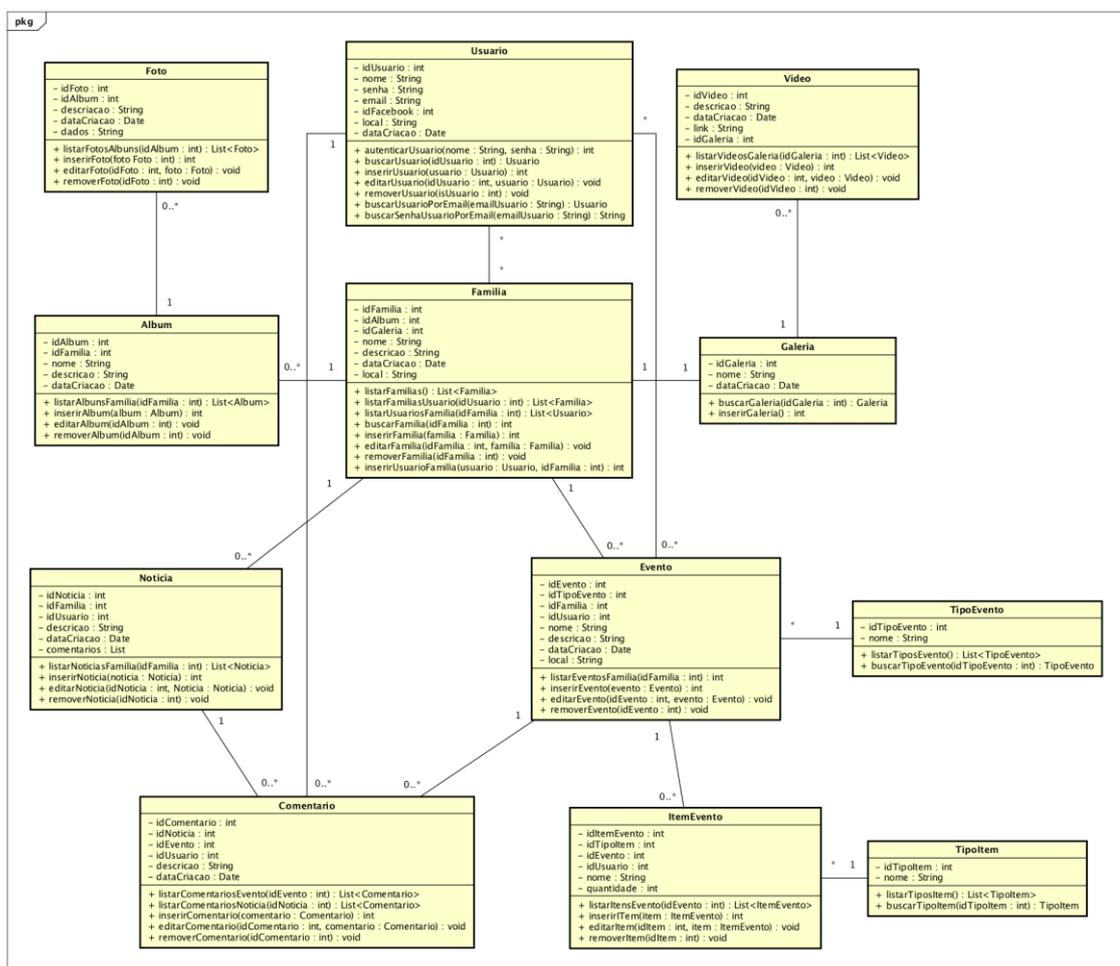
REGRAS DE NEGÓCIO

Não há regra de negócio para este caso de uso.

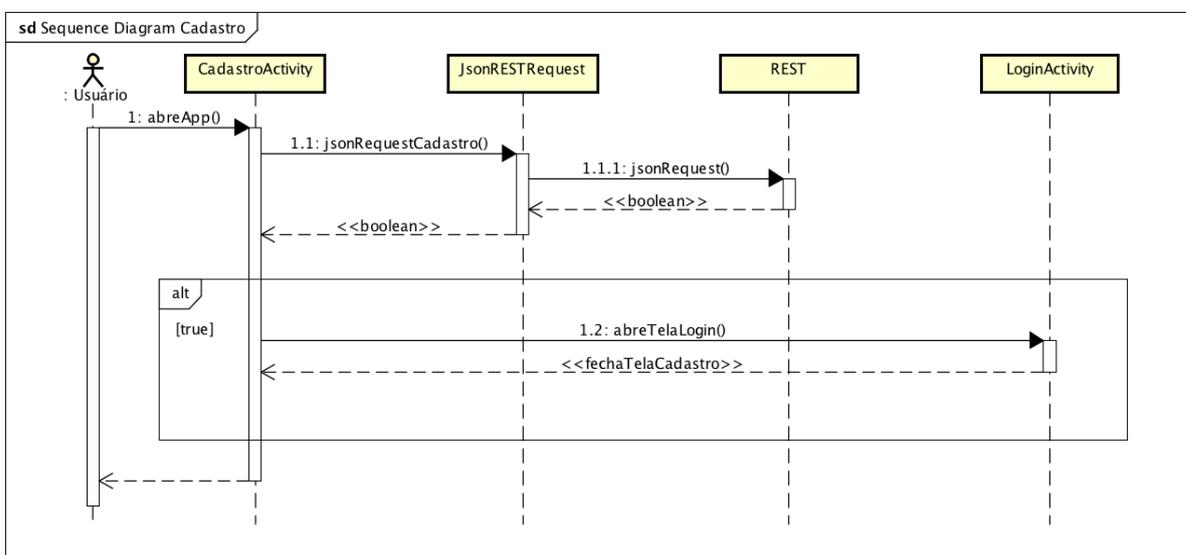
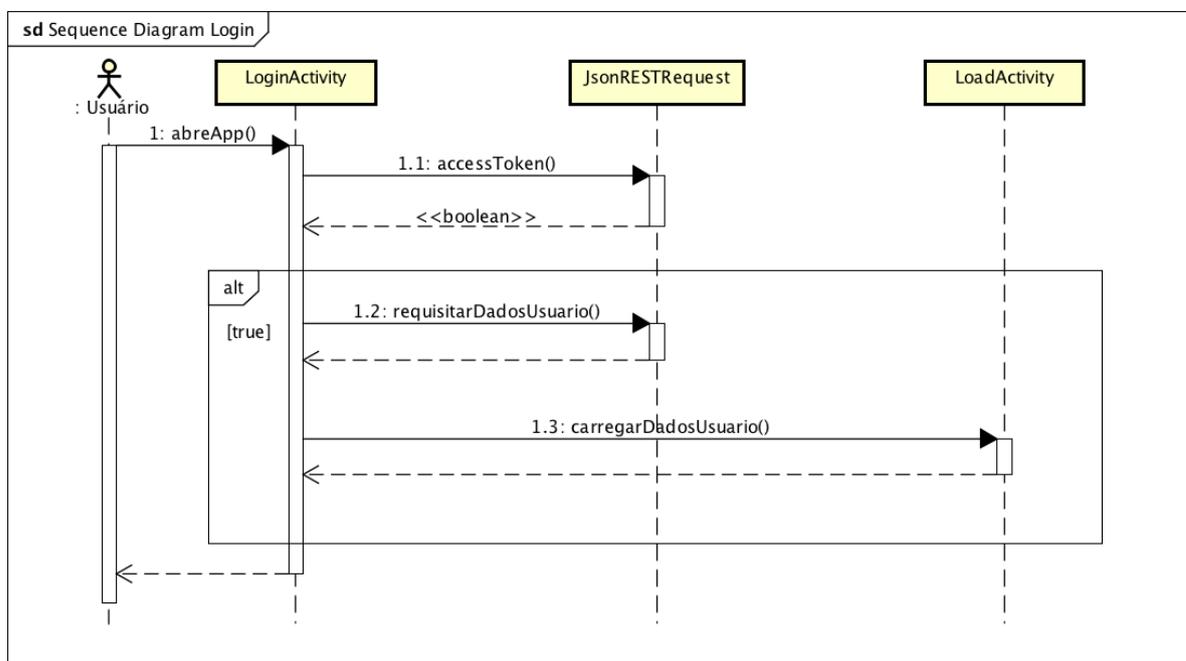
APÊNDICE 4 – DIAGRAMA ENTIDADE RELACIONAMENTO

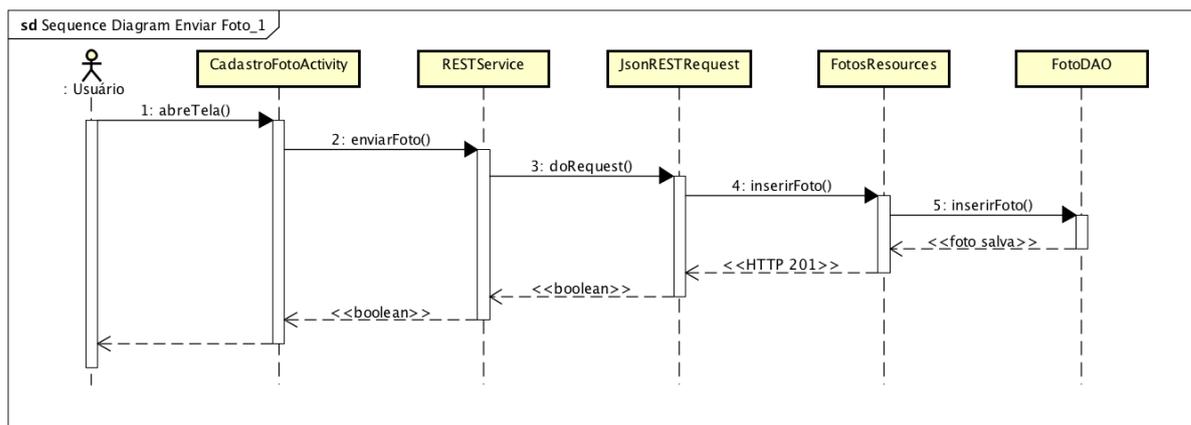
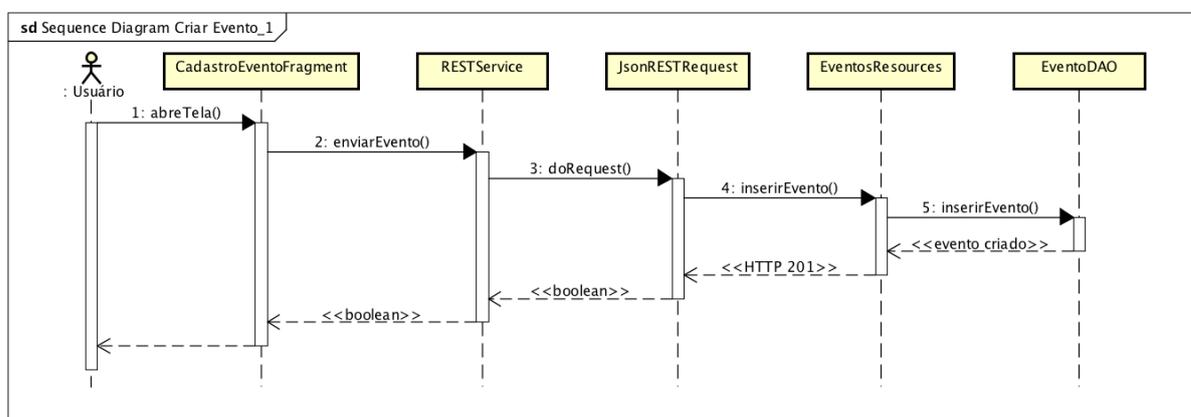
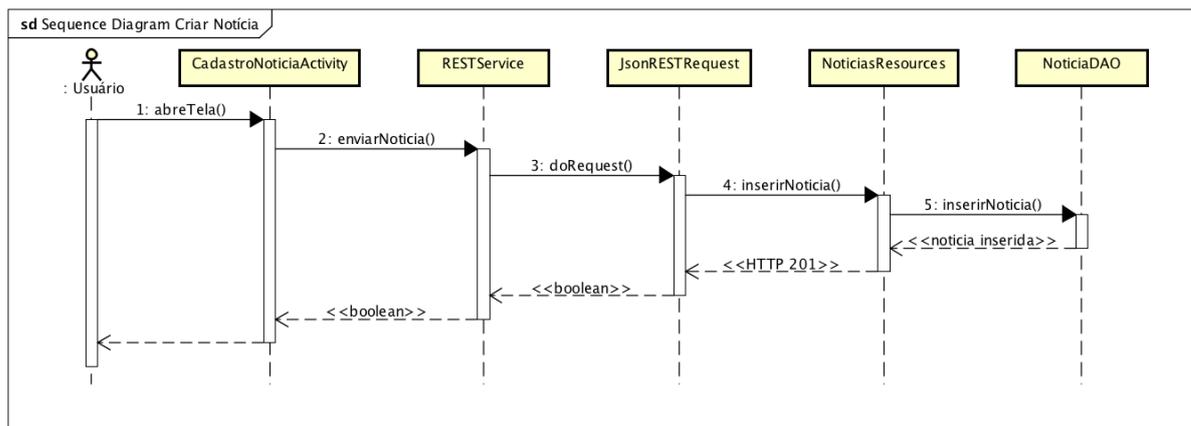


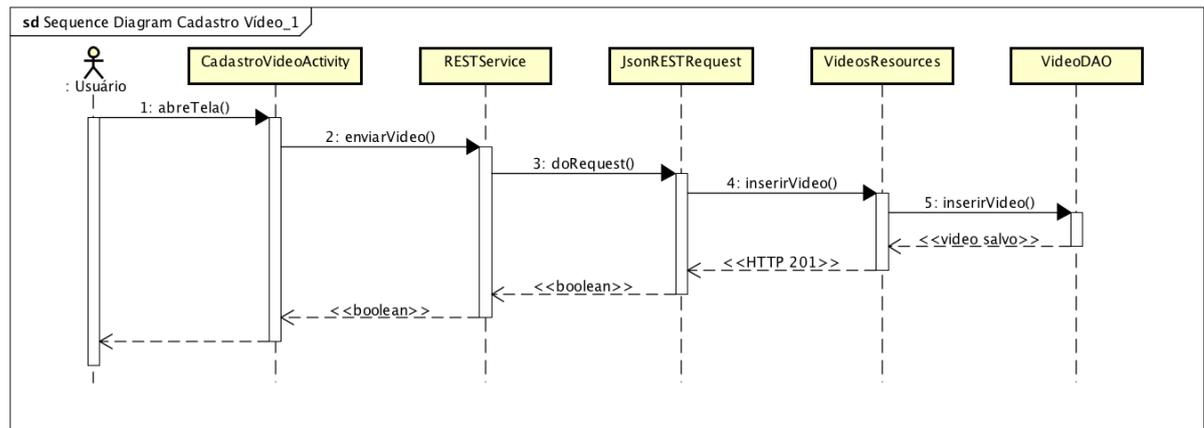
APÊNDICE 5 – DIAGRAMA DE CLASSES DE IMPLEMENTAÇÃO



APÊNDICE 6 – DIAGRAMAS DE SEQUÊNCIA







APÊNDICE 7 – DIAGRAMA WBS

O diagrama WBS foi dividido em cinco partes para melhor visualização:

PARTE 1 – CONCEPÇÃO

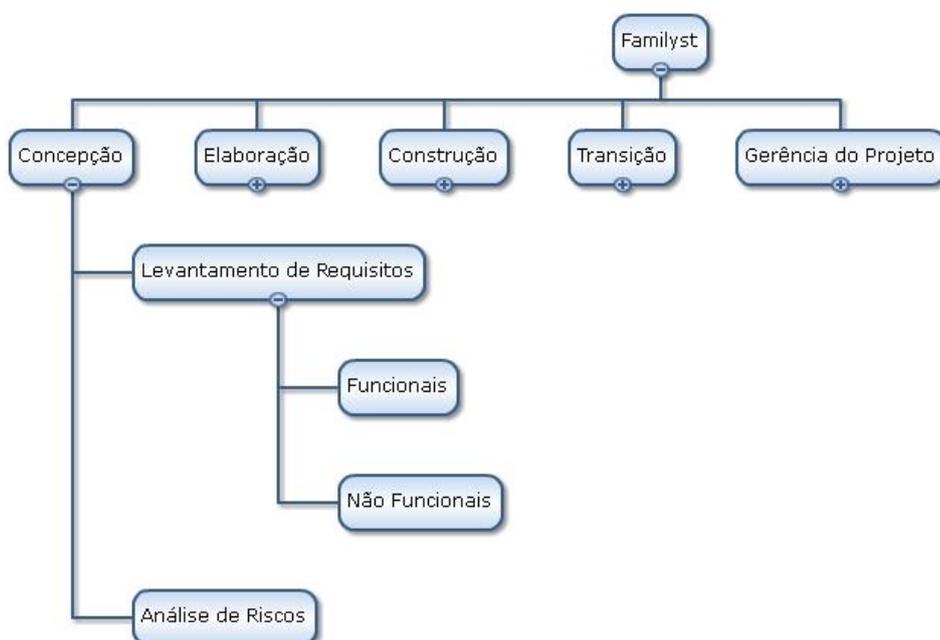


FIGURA x – DIAGRAMA WBS (CONCEPÇÃO)

PARTE 2 – ELABORAÇÃO

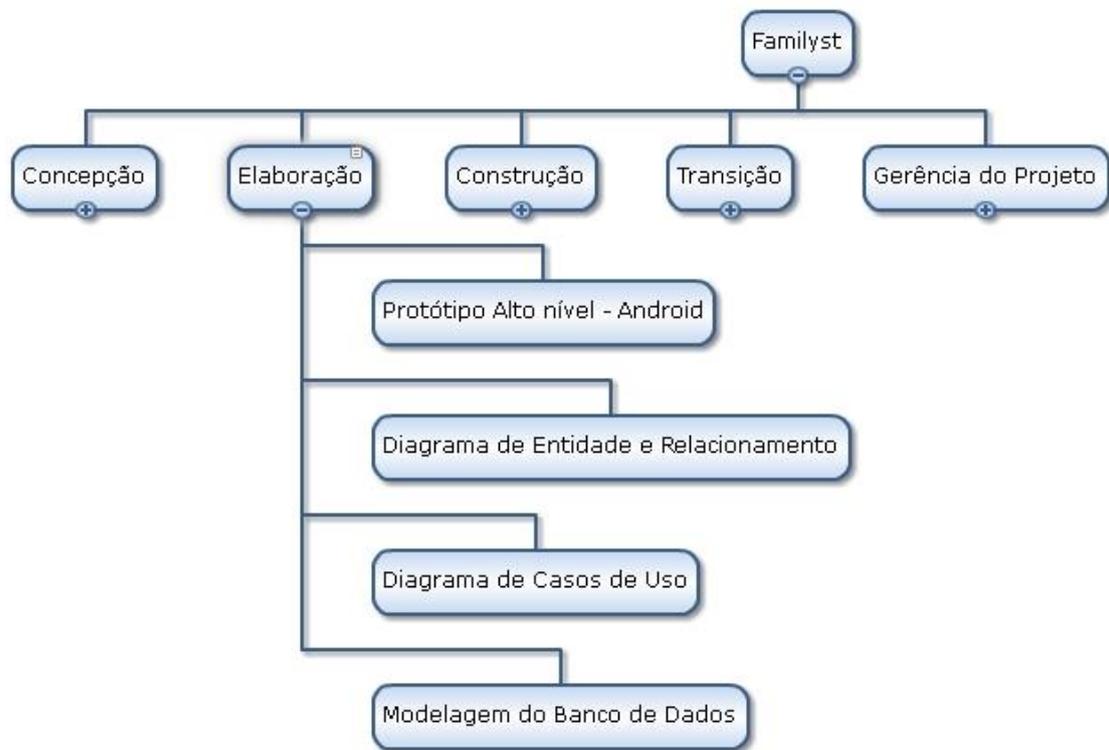


FIGURA X – DIAGRAMA WBS (ELABORAÇÃO)

PARTE 3 - CONTRUÇÃO

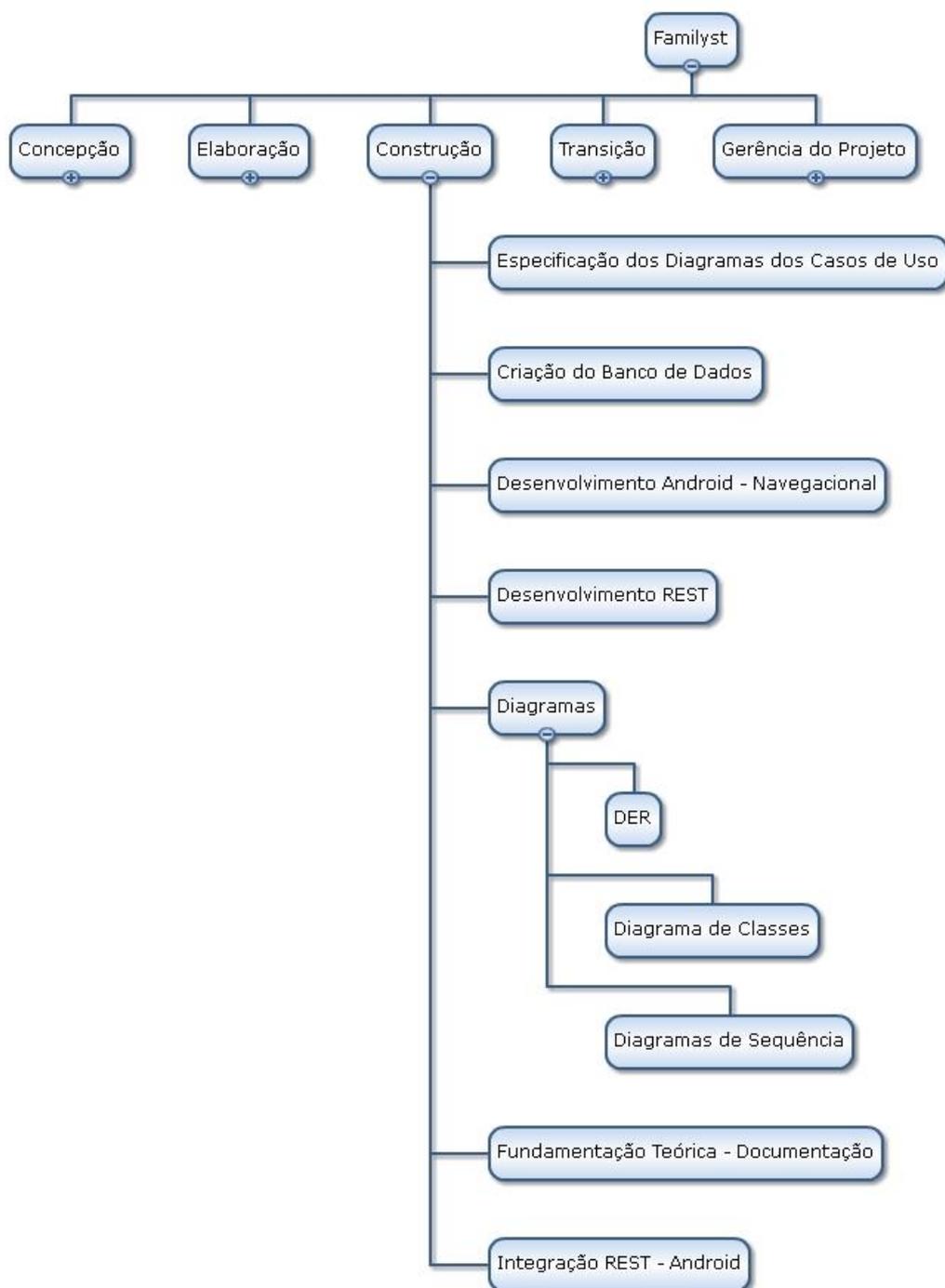


FIGURA X – DIAGRAMA WBS (CONSTRUÇÃO)

PARTE 4 – TRANSIÇÃO

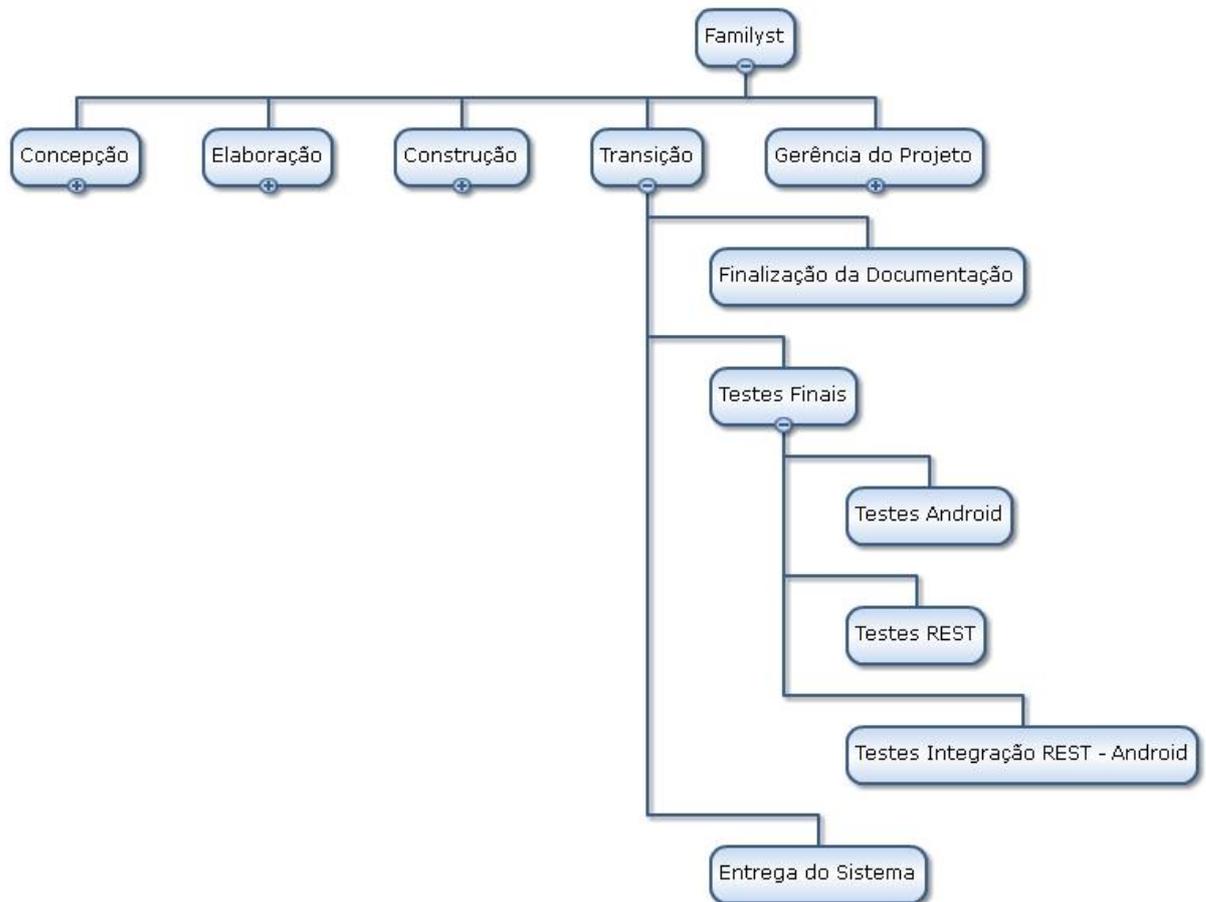


FIGURA X – DIAGRAMA WBS (TRANSIÇÃO)

PARTE 5 – GERÊNCIA DO PROJETO

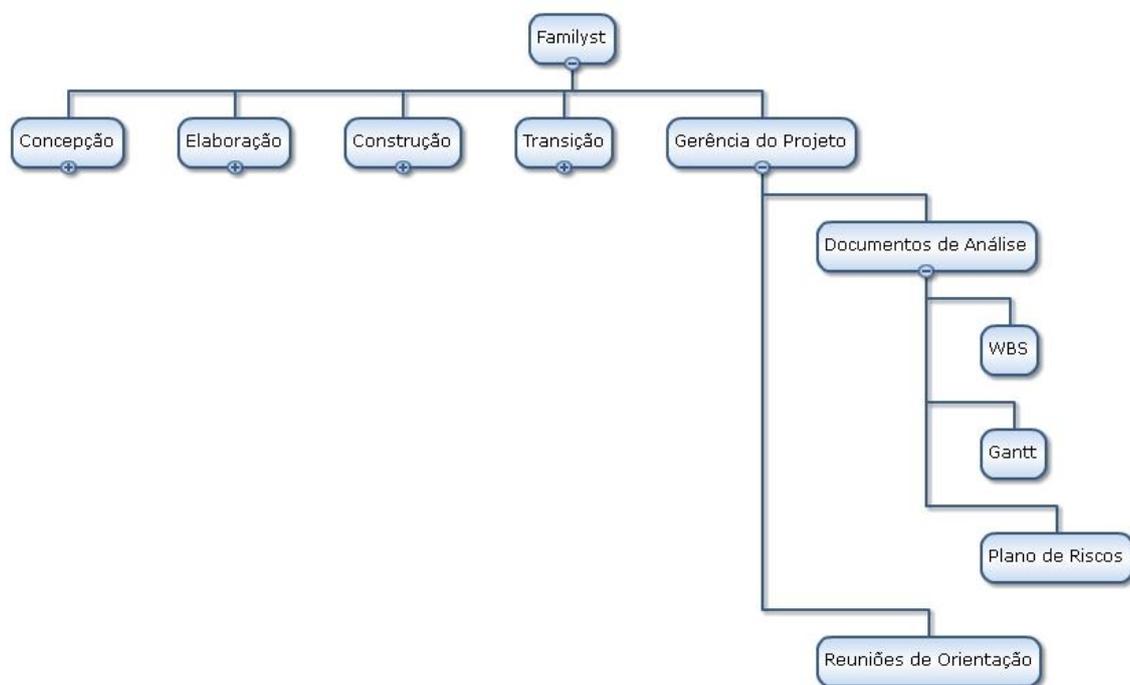
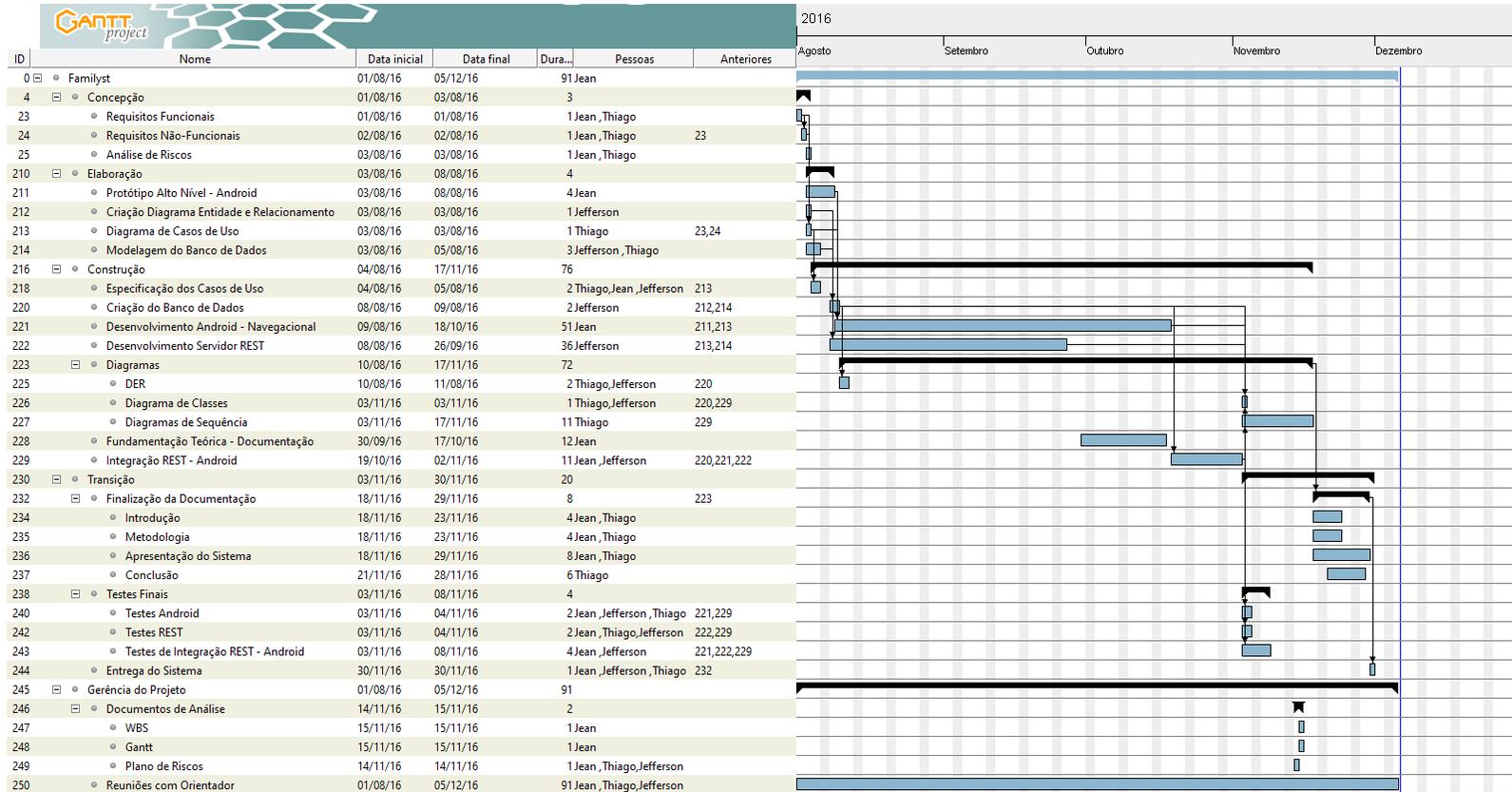


FIGURA X – DIAGRAMA WBS (GERÊNCIA DO PROJETO)

APÊNDICE 8 – GRÁFICO DE GANTT



APÊNDICE 8 – CÓDIGOS REST

Familias: Authorization: Return 401 - Unauthorized

| | | |
|--------------------------------|--|---|
| GET /familias | 200 - OK {List} | |
| | 500 - Internal Server Error | |
| GET /familias{idFamilia} | 200 - OK {familia} | |
| | 404 - Not Found | |
| | 500 - Internal Server Error | |
| POST /familias | 201 - Created HEADER {resourcebasepath}/idFamilia | Body (JSON - XML) |
| | 409 - Conflict | { "familia": { "nome": , "descricao": ,"local": }} |
| | 500 - Internal Server Error | |
| PUT /familias{idFamilia} | 200 - OK | Body (JSON - XML) |
| | 404 - Not Found | { "familia": { "nome": , "descricao": ,"local": }} |
| | 204 - No Content | |
| | 500 - Internal Server Error | |
| DELETE /familias{idFamilia} | 200 - OK | |
| | 404 - Not Found | |
| | 500 - Internal Server Error | |

AccessTokens:

| | | |
|--------------------|------------------------------------|--|
| POST /accesstokens | 200 - OK { idUsuario, accesstoken} | Body (JSON - XML) |
| | 404 - Not Found | { "accesstokenrequest": { "nome": ,"senha":}} |
| | 500 - Internal Server Error | |

Usuarios: Authorization: Return 401 - Unauthorized

| | | |
|-----------------------------|--|---|
| GET /usuario{idUsuario} | 200 - OK {usuario} | |
| | 404 - Not Found | |
| | 500 - Internal Server Error | |
| POST /usuarios | 201 - Created HEADER {resourcebasepath}/idUsuario | Body (JSON - XML) |
| | 409 - Conflict | { "usuario": { "nome": , "senha": ,"email": , "local":}} |
| | 500 - Internal Server Error | |
| PUT /usuarios{idUsuario} | 200 - OK | Body (JSON - XML) |
| | 404 - Not Found | { "usuario": { "nome": , "local": }} |
| | 204 - No Content | |
| | 500 - Internal Server Error | |
| DELETE /usuarios{idUsuario} | 200 - OK | |
| | 404 - Not Found | |
| | 500 - Internal Server Error | |

Galerias: Authorization: Return 401 - Unauthorized

| | |
|--------------------------|-----------------------------|
| GET /galerias{idGaleria} | 200 - OK {galeria} |
| | 404 - Not Found |
| | 500 - Internal Server Error |

Arquivos: Authorization: Return 401 - Unauthorized

| | |
|---------------------------|-----------------------------|
| GET /arquivos{idArquivos} | 200 - OK {arquivo} |
| | 404 - Not Found |
| | 500 - Internal Server Error |

Videos: Authorization: Return 401 - Unauthorized

| | | |
|------------------------|---|---|
| GET /videos{idGaleria} | 200 - OK {videos da galeria} | |
| | 404 - Not Found | |
| | 500 - Internal Server Error | |
| POST /videos | 201 - Created HEADER {resourcebasepath}/idVideo | Body (JSON - XML) |
| | 500 - Internal Server Error | { "video": { "descricao": , "link": , "idGaleria": }} |
| PUT /videos{idVideo} | 200 - OK | Body (JSON - XML) |
| | 404 - Not Found | { "video": { "descricao": , "link": }} |
| | 204 - No Content | |
| | 500 - Internal Server Error | |
| DELETE | 200 - OK | |

| | |
|------------------|-----------------------------|
| /videos{idVideo} | 404 - Not Found |
| | 500 - Internal Server Error |

TiposEvento: Authorization: Return 401 - Unauthorized

| | |
|------------------|-----------------------------|
| GET /tiposEvento | 200 - OK {tiposEvento} |
| | 500 - Internal Server Error |

TiposItem: Authorization: Return 401 - Unauthorized

| | |
|----------------|-----------------------------|
| GET /tiposItem | 200 - OK {tiposItem} |
| | 500 - Internal Server Error |

Eventos: Authorization: Return 401 - Unauthorized

| | | |
|-------------------------|---|---|
| GET /eventos{idFamilia} | 200 - OK {eventos} | |
| | 404 - Not Found | |
| | 500 - Internal Server Error | |
| POST /eventos | 201 - Created HEADER {resourcebasepath}/idEvento | Body (JSON - XML) |
| | 500 - Internal Server Error | { "evento": { "nome": , "descricao": ,"local": , "idFamilia": ,"idTipoEvento": , "idUsuario":}} |
| PUT /eventos{idEvento} | 200 - OK | Body (JSON - XML) |

| | | |
|---------------------------|-----------------------------|---|
| | 404 - Not Found | {"evento": { "nome": , "descricao": , "local": , "idTipoEvento": }} |
| | 204 - No Content | |
| | 500 - Internal Server Error | |
| DELETE /eventos{idEvento} | 200 - OK | |
| | 404 - Not Found | |
| | 500 - Internal Server Error | |

Itens: Authorization: Return 401 - Unauthorized

| | | |
|-----------------------|--|--|
| GET /itens{idEvento} | 200 - OK {itens} | |
| | 404 - Not Found | |
| | 500 - Internal Server Error | |
| POST /itens | 201 - Created HEADER {resourcebasepath}/idItem | Body (JSON - XML) |
| | 500 - Internal Server Error | {"item": { "quantidade": , "idTipoItem": , "idEvento": , "idUsuario": }} |
| PUT /itens{idItem} | 200 - OK | Body (JSON - XML) |
| | 404 - Not Found | {"item": { "quantidade": , "idTipoItem": , "idUsuario": }} |
| | 204 - No Content | |
| | 500 - Internal Server Error | |
| DELETE /itens{idItem} | 200 - OK | |

404 - Not Found
500 - Internal Server Error

Albums: Authorization: Return 401 - Unauthorized

| | | |
|-------------------------|--|--|
| GET /albuns{idFamilia} | 200 - OK {albuns} 404 - Not Found 500 - Internal Server Error | |
| POST /albuns | 201 - Created HEADER {resourcebasepath}/idItem 500 - Internal Server Error | Body (JSON - XML) { "album": { "nome": , "descricao": , "idFamilia": }} |
| PUT /albuns{idAlbum} | 200 - OK 404 - Not Found 204 - No Content 500 - Internal Server Error | Body (JSON - XML) { "album": { "nome": , "descricao": }} |
| DELETE /albuns{idAlbum} | 200 - OK 404 - Not Found 500 - Internal Server Error | |

TABELA 3 – API REST
FONTE: OS AUTORES