

**Serviço Nacional de Aprendizagem Comercial do Rio Grande do Sul**  
**Faculdade Senac Porto Alegre**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

LUCIANO ORTIZ SILVA

RELATÓRIO DE PROJETO FINAL

KIDS

Aplicação web de interação entre creches e familiares

Porto Alegre  
2017

LUCIANO ORTIZ SILVA

## RELATÓRIO DE PROJETO FINAL

### KIDS

Aplicação web de interação entre creches e familiares

Relatório de Projeto Final apresentado como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas pela Faculdade Senac Porto Alegre.

Orientador: Prof. Me. James Brum

Porto Alegre  
2017

**LUCIANO ORTIZ SILVA**

**RELATÓRIO DE PROJETO FINAL**

**KIDS**

Aplicação web de interação entre creches e familiares

Relatório de Projeto Final apresentado como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas pela Faculdade Senac Porto Alegre.

Orientador: Prof. Me. James Brum

Aprovado pela banca examinadora em dezembro de 2017

---

Prof. Me. James Brum  
Orientador - Faculdade Senac Porto Alegre

---

Prof. **aguardando definição** - Faculdade Senac Porto Alegre

---

Prof. **aguardando definição** - Faculdade Senac Porto Alegre

Porto Alegre  
2017

## AGRADECIMENTOS

Ao meu orientador, James, pela constância, pelo profissionalismo e, em especial, por me mostrar o melhor caminho a seguir nas diversas vezes em que eu insistia em me desviar dele.

Aos professores Ries e Aline, que me ajudaram a resolver problemas técnicos durante o projeto.

A minha irmã e P.O. do projeto Liziane Ortiz, pela paciência em me atender aos finais de semana mesmo cansada e sem vontade alguma.

Ao Antonio Dias, por ter me tirado da zona de conforto tecnológica. Enquanto eu queria desenvolver somente com tecnologias que eu já conhecia, ele me incentivava a utilizar somente tecnologias super atuais, tais como o Angular na versão 4 e o grande Spring Boot.

Ao Joni Marques, por sempre me incentivar em desenvolver algo para se tornar um milionário, e abirmos uma empresa juntos ao final da faculdade. Além de me tirar da bad quando conseguia ingresso + churrasco + bebida (tudo de graça) no estádio do Sport Club Internacional. (Esse Joni conhece todo mundo kkk). "Cuidado, ele é muito comercial".

Ao Cristiano Serafim, por ter me levado de arrasto para a faculdade nos diversos dias em que eu pensei em trancar e parar de estudar, ele aparecia do nada na minha mesa, oferecendo carona, me livrando do terrível T7 lotado as 18h hahahaha.

Ao incrível Willian Marques, o que dizer desse rapaz ? Uns dizem que ele é doente, outros acham que ele veio de outro planeta. Prefiro acreditar que ele entra na vida das pessoas em forma de "terapia", para sermos testados sobre o que é certo ou errado nesse planeta. Não estão entendendo nada nesse trecho ??? Nem eu, essa é a vibe por trás do incrível Willian Marques, o famoso *Proxy*.

A empresa Neogrid, por sempre me liberar em horários de serviço para resolver alguma questão a respeito da faculdade. Além de pagar 50% da mesma.

## **RESUMO**

Esse projeto está sendo conduzido juntamente com a P.O (Product Owner) Liziane Ortiz (cuidadora de crianças), que busca corrigir diversos problemas citados pela mesma, que ocorrem dentro de uma creche de pequeno porte chamada "Vivendo e Aprendendo", situada na zona norte da cidade de Porto alegre.

Assim, esse projeto visa promover a integração entre familiares e creches, permitindo uma interatividade permanente entre a creche e a família, disponibilizando dados para que os pais tomem conhecimento da rotina escolar de seus filhos com mais agilidade. Para isso, oferece, uma plataforma web que disponibiliza diversas atividades, tais como: diário escolar, avaliação de rendimento, galeria de fotos, avisos, eventos, envio de sugestões, dados das crianças, atualização de dados cadastrais, diminuindo custos com agendas físicas e impressões.

**PALAVRAS-CHAVE:** creche, familiares

## LISTA DE ILUSTRAÇÕES

Figura 1 - Comparação de projetos .....	12
Figura 2 - Tela de login .....	20
Figura 3 - Tela de seleção da conta Google.....	20
Figura 4 - Tela para informar a senha da conta Google .....	21
Figura 5 - Tela para escolher o tipo de conta .....	22
Figura 6 - Tela de dados do perfil do usuário.....	22
Figura 7 – Tela de pesquisa de crianças .....	23
Figura 8 – Tela de cadastro/edição de dados gerais (Passo 01).....	24
Figura 9 – Tela de cadastro/edição de endereço (Passo 02) .....	24
Figura 10 – Tela de cadastro/edição de contato (Passo 03).....	25
Figura 11 – Tela de cadastro/edição de medicamentos (Passo 04) .....	25
Figura 12 – Tela de cadastro/edição de alergias (Passo 05) .....	26
Figura 13 - Tela de vínculo de um familiar para uma criança.....	27
Figura 14 - Corpo do e-mail referente ao vínculo realizado.....	27
Figura 15 - Tela de diários (visão creche) .....	28
Figura 16 - Tela de diários (visão familiar).....	28
Figura 17 - Tela de cadastro e lista de eventos (visão creche) .....	29
Figura 18 - Tela de eventos (visão familiar).....	30
Figura 19 - Tela de Avisos - (visão da creche).....	31
Figura 20- Tela de Avisos - (visão da familiar).....	31
Figura 21 - Tela de Interação - (visão da creche) .....	32
Figura 22 - Tela de Interação - (visão familiar).....	32
Figura 23 - Tela de Avaliações - (visão creche).....	33
Figura 24 - Tela de Avaliações - (visão familiar).....	33
Figura 25 - Tela de Cardápio - (visão creche).....	34
Figura 26 - Tela de cardápio / lista - (visão familiar).....	35
Figura 27 - Tela da Galeria de Imagens - (visão Creche).....	35
Figura 28 - Tela da Galeria de Imagens - (visão familiar) .....	36
Figura 29 - Visão geral da arquitetura .....	39
Figura 30 - Modelo de Entidade e Relacionamento .....	55

## LISTA DE TABELAS

Tabela 1 - Product Backlog .....	40
Tabela 2 - User Stories do Product Backlog - 01 .....	40
Tabela 3 - User Stories do Product Backlog - 02 .....	41
Tabela 4- User Stories do Product Backlog - 03 .....	41
Tabela 5- User Stories do Product Backlog - 04 .....	41
Tabela 6- User Stories do Product Backlog - 05 .....	41
Tabela 7- User Stories do Product Backlog - 06 .....	42
Tabela 8- User Stories do Product Backlog - 07 .....	42
Tabela 9- User Stories do Product Backlog - 08 .....	42
Tabela 10- User Stories do Product Backlog - 09 .....	42
Tabela 11- User Stories do Product Backlog - 10 .....	43
Tabela 12- User Stories do Product Backlog - 11 .....	43
Tabela 13- User Stories do Product Backlog - 12 .....	43
Tabela 14 - Sprint 01 - Planejamento .....	44
Tabela 15 - Sprint 01 - Resultado .....	44
Tabela 16 - Sprint 02 - Planejamento .....	45
Tabela 17 - Sprint 02 - Resultado .....	46
Tabela 18 - Sprint 03 - Planejamento .....	47
Tabela 19 - Sprint 03 - Resultado .....	47
Tabela 20 - Sprint 04 - Planejamento .....	48
Tabela 21 - Sprint 04 - Resultado .....	48
Tabela 22 - Sprint 05 - Planejamento .....	49
Tabela 23 - Sprint 05 - Resultado .....	49
Tabela 24 - Sprint 06 - Planejamento .....	51
Tabela 25 - Sprint 06 - Resultado .....	51
Tabela 26 - Sprint 07 - Planejamento .....	52
Tabela 27 - Sprint 07 - Resultado .....	53
Tabela 28 - Sprint 08 - Planejamento .....	54
Tabela 29 - Sprint 08 - Resultado .....	54
Tabela 30 - Permissões por papel de usuário .....	56
Tabela 31- Permissões por papel de usuário (continuação) .....	57

## SUMÁRIO

<b>1</b>	<b>APRESENTAÇÃO GERAL DO PROJETO .....</b>	<b>10</b>
<b>2</b>	<b>DEFINIÇÃO DO PROBLEMA .....</b>	<b>11</b>
<b>3</b>	<b>OBJETIVOS .....</b>	<b>13</b>
3.1	OBJETIVO GERAL .....	13
3.2	OBJETIVOS ESPECÍFICOS .....	13
3.2.1	Alergias.....	13
3.2.2	Familiares.....	13
3.2.3	Cardápio.....	13
3.2.4	Interação .....	13
3.2.5	Medicamentos .....	13
3.2.6	Galeria de Imagens.....	13
3.2.7	Avisos.....	13
3.2.8	Dados .....	13
3.2.9	Diário Escolar .....	13
3.2.10	Eventos .....	13
<b>4</b>	<b>ANÁLISE DE TECNOLOGIAS / FERRAMENTAS .....</b>	<b>14</b>
4.1	LINGUAGENS DE PROGRAMAÇÃO.....	14
4.1.1	Java <sup>1</sup> .....	14
4.1.2	Typescript <sup>2</sup> .....	14
4.2	FRAMEWORKS.....	15
4.2.1	Angular <sup>3</sup> .....	15
4.2.2	Bootstrap <sup>4</sup> .....	15
4.2.3	Spring boot <sup>5</sup> .....	15
4.3	BANCO DE DADOS.....	15
4.3.1	Postgres <sup>6</sup> .....	15
4.4	AMBIENTE DE DESENVOLVIMENTO .....	16
4.4.1	Eclipse <sup>7</sup> .....	16
4.4.2	Visual studio code <sup>8</sup> .....	16
4.5	ARTEFATOS DE VERSIONAMENTO .....	17
4.5.1	Git <sup>9</sup> .....	17
4.5.2	Github <sup>10</sup> .....	17
4.6	ARTEFATO DE MODELAGEM.....	17
4.6.1	Dezign <sup>11</sup> .....	17
4.6.2	Bizagi <sup>12</sup> .....	18
4.7	INTEGRAÇÃO CONTÍNUA .....	18



4.7.1	<b>Travis</b> <sup>13</sup>	18
4.8	SERVIDOR DE HOSPEDAGEM	18
4.8.1	<b>Heroku</b> <sup>14</sup>	18
4.9	FERRAMENTA DE QA	19
4.9.1	<b>Sonar</b> <sup>15</sup>	19
4.10	BIBLIOTECAS EXTERNAS	19
4.10.1	<b>Api swagger 2</b> <sup>16</sup>	19
4.10.2	<b>Api google maps</b> <sup>17</sup>	19
5	<b>DESCRIÇÃO DA SOLUÇÃO</b>	20
5.1	LOGIN	20
5.2	TIPO DE CONTA	21
5.3	PERFIL	22
5.4	CRIANÇAS	23
5.5	FAMILIARES	26
5.6	DIÁRIO	27
5.7	EVENTOS	28
5.8	AVISOS	30
5.9	INTERAÇÃO	31
5.10	AVALIAÇÃO	33
5.11	CARDÁPIO	34
5.12	GALERIA DE FOTOS	35
6	<b>ABORDAGEM DE DESENVOLVIMENTO</b>	37
7	<b>ARQUITETURA DO SISTEMA</b>	38
7.1	PROCESSO DE DESENVOLVIMENTO	39
7.2	MODELAGEM FUNCIONAL	40
7.2.1	<b>Product backlog</b>	40
7.2.2	<b>User stories</b>	40
7.2.3	<b>Sprints</b>	44
7.3	MODELAGEM DE DADOS	55
8	<b>FUNCIONAMENTO DO SISTEMA</b>	56
8.1	DADOS PARA VERIFICAR O FUNCIONAMENTO DO SISTEMA	57
9	<b>VALIDAÇÃO</b>	58
9.1	ESTRATÉGIA	58
9.2	CONSOLIDAÇÃO DOS DADOS COLETADOS	59

## 1 APRESENTAÇÃO GERAL DO PROJETO

O atendimento de crianças de 0 a 3 anos em creches está assegurado desde a Constituição Federal de 1988 e também pelo Plano Nacional de Educação (PNE). Porém, esse atendimento ainda é um grande desafio em todo o Brasil. De acordo com o atual PNE, em vigor desde 2014 e até 2024, o País precisa atender 50% das crianças dessa faixa etária até o final da vigência do plano. Essa meta também estava prevista no PNE anterior (2001-2010), porém, em 2011, apenas 24,5% das crianças de 0 a 3 anos estavam matriculadas em creches.

De acordo com os dados da Pnad, desde 2005 há um crescimento constante na porcentagem de crianças dessa faixa etária na Educação Infantil, atingindo 29,6% em 2014. Em números absolutos, isso significa cerca de 3,5 milhões de crianças matriculadas em creches em todo o País. (Estatão, Edu, 2016).

Uma alternativa prática que vem sendo investida cada vez mais no mercado digital, é o desenvolvimento de aplicações na área da educação que, se utilizados corretamente, aproximam a relação escola x alunos e escola x responsáveis por esses alunos.

Já existem muitos sistemas/aplicativos com diferentes modos de auxiliar nessa comunicação entre os envolvidos, alguns deles são o [bebealerta.com.br](http://bebealerta.com.br), e o aplicativo diário escolar. Apesar de existir diversas plataformas nesse ramo, ainda há espaço para uma gama de melhorias que venham satisfazer as necessidades desses usuários.

Dessa forma, esse projeto propõe desenvolver uma aplicação que atenda o máximo de necessidades centralizadas em apenas um só portal. Uma aplicação que seja de fácil acesso, sem burocracias ou ter que lembrar diferentes códigos e senhas para diversos cadastros em diferentes instituições de ensino.

A ideia é de se trabalhar com dois papéis diferentes dentro da aplicação, um papel é o da creche, onde os usuários poderão registrar diversos conteúdos referentes as crianças matriculadas em sua instituição de ensino, dentro desses conteúdos podemos citar: imagens, eventos, avisos, diário escolar, cardápio, avaliações, entre outros.

Já no papel de um familiar, os usuários poderão visualizar todo o conteúdo fornecido pela instituição de ensino, assim como manter atualizado tanto seus dados, quanto informações da criança matriculada, também poderão tirar dúvidas, enviar sugestões, entre outros.

## 2 DEFINIÇÃO DO PROBLEMA

A seguir serão descritos diversos problemas citados pela P.O do projeto, Liziane Ortiz (cuidadora de crianças). Problemas esses que ocorrem dentro de uma creche de pequeno porte chamada "Vivendo e Aprendendo", situada na zona norte da cidade de Porto alegre.

Na necessidade de algumas crianças receberem algum medicamento durante o dia, ocorrem momentos em que o funcionário extravie a sua anotação realizada no papel, referente ao intervalo ou a dosagem correta. Isso acaba gerando transtornos para os pais, onde os mesmos precisam ser contatados via ligação de celular ou via whats app, somente para informarem esses dados.

Um dos problemas que ocorrem com o cuidado de crianças é sua possível alergia a algum alimento ou medicação, que pode, em alguns casos, causar sérios problemas de saúde e emergência naquele ambiente.

Ocorre praticamente todas as vezes quando há um evento na escolinha, a ausências de alguns alunos, e o não cumprimento de trajas específicos para aquele evento. O que na maioria das vezes é relatado por não haver um local que deixe os detalhes do evento por escrito, sendo assim os pais acabam deixando para perguntar "depois", o que acaba não ocorrendo. Gerando esses pequenos transtornos.

Uma parte dos familiares reclamam que seus filhos não consomem alimentos o suficiente, e tem aqueles que reclamam que os alimentos são servidos em demasia ou de "má qualidade" (quando doces ou afins).

Qualquer discussão entre familiares e a creche, não há nada de escrito. O que acaba gerando as mesmas cobranças por mais de uma vez.

Sempre quando há qualquer tipo de evento na creche, ou até mesmo no dia a dia, quando chega a hora dos familiares buscarem suas crianças, eles acabam pedindo que enviem fotos de seus filhos no evento. Esses pais acabam deixando o whats app e alguns deixam o e-mail para aguardar o recebimento dessas fotos.

Quando chega ao final do dia, e os familiares buscam suas crianças, 90% querem um "relatório" de perguntas frequentes como: comeu bastante ou pouco? Dormiu bem? Evacuuou?

Dependendo da turma (idade das crianças), periodicamente a creche realiza uma avaliação de cada aluno matriculado. Que nada mais é do que uma descrição do desenvolvimento de cada criança. O problema acontece quando algumas dessas avaliações são rasuradas por acidentes como: derramar café por cima, alimentos, água, e etc. Isso gera todo um retrabalho para reescrever uma avaliação inteira.

Após análise realizada com projetos que atendem algumas dessas demandas, segue abaixo algumas descrições sobre esses projetos correlacionados:

- **Bebê Alerta:**

O aplicativo tem como objetivo principal promover a segurança das crianças que estão sujeitas ao esquecimento dos pais, em seus veículos.

Para isso, o Bebê Alerta tem um sistema de notificação integrado ao check-in diário que o pai deve realizar quando leva e busca o filho na creche ou na escolinha.

Ele avisa ao pai quando este check-in não é realizado, alertando-o para a possibilidade de esquecimento ou mudança de rotina.

Seu custo é gratuito.

Fonte: <http://www.bebealerta.com.br/>

- **Diário Escola:**

O aplicativo Diário Escola para Educação Infantil é a maneira mais simples, prática e segura de você acompanhar o dia a dia de seu filho na Escolinha ou na Creche. Esqueça as ultrapassadas cadernetas de papel, você pode utilizar seu celular e ter acesso ao diário de seu filho em tempo real e aonde estiver!

Seu custo é gratuito (para familiares), porém há custos R\$ para as escolas infantis.

Fonte: <https://diarioescola.com.br/>

- **Aplicativo Escolar:**

O Aplicativo Escolar oferece 3 papéis, são eles:

Os pais e responsáveis poderão acompanhar em tempo real a vida acadêmica do aluno. Consultar notas, chamadas, receber avisos de falta e atraso, tudo diretamente em seu smartphone.

O aluno baixa o material didático digital, realiza provas e testes online, consulta a agenda escolar, verifica datas de provas e testes, visualiza o resultado de provas, histórico de chamada e muito mais.

Os professores utilizam o Aplicativo Escolar como uma ferramenta que vai auxiliar em suas atividades diárias na escola, como realizar chamadas, lançar notas, consultar a grade de aulas e muito mais.

Seu custo é pago, com planos diferentes para cada tipo de usuário.

Fonte: <https://www.aplicativoescolar.com/>

Segue abaixo uma tabela que apresenta um comparativo entre as funcionalidades dos projetos citados acima:

Figura 1 - Comparação de projetos

Comparação de projetos correlacionados				
descrição	Bebê alerta	Diário Escolar	Aplicativo Escolar	Kids
medicamentos		X		X
alergias		X		X
eventos		X		X
cardápio			X	X
login rede social				X
galeria de fotos				X
avisos	X	X	X	X
avaliações			X	X
contato c\ creche	X	X	X	X
diário	X	X		X
app android	X	X	X	
app IOS	X	X	X	
sistema web		X		X
sistema responsivo				X

Fonte: construída pelo autor do projeto.

### **3 OBJETIVOS**

Neste projeto, os objetivos foram organizados em objetivo geral e objetivos específicos, conforme segue abaixo.

#### **3.1 OBJETIVO GERAL**

O objetivo geral desse projeto, tem como fornecer um único portal de acesso, que facilite a comunicação entre instituições de ensino infantil, e familiares que possuem filhos em idade escolar que estiverem matriculados na mesma. Após essa conexão ter sido feita, os familiares poderão acompanhar diariamente as atividades de suas crianças.

#### **3.2 OBJETIVOS ESPECÍFICOS**

Os objetivos específicos deste projeto são:

##### **3.2.1 Alergias**

Manter a creche informada sobre alguma alergia que alguma criança possa conter.

##### **3.2.2 Familiares**

Informar aos familiares detalhes sobre um determinado evento a ser realizado na creche.

##### **3.2.3 Cardápio**

Manter um quadro de cardápio completo e atualizado por parte da creche.  
Disponibilizando o mesmo para todos os familiares;

##### **3.2.4 Interação**

Uma forma de familiares de comunicarem com a creche, mantendo um registro dessa comunicação.

##### **3.2.5 Medicamentos**

Deixar claro e visível para a creche, dados de medicamentos de uso da criança;

##### **3.2.6 Galeria de Imagens**

Possibilitar a creche disponibilizar diversas imagens dos eventos que a mesma realizar.

##### **3.2.7 Avisos**

Gerar avisos de sobre alguma informação importante.

##### **3.2.8 Dados**

Manter dados das crianças devidamente atualizados.

##### **3.2.9 Diário Escolar**

Disponibilizar para os familiares, informações sobre algumas atividades diárias da criança.

##### **3.2.10 Eventos**

Possibilitar a creche enviar convites para as crianças, referente a eventos realizados na creche.

## 4 ANÁLISE DE TECNOLOGIAS / FERRAMENTAS

Neste projeto a grande maioria das ferramentas utilizadas serão gratuitas, tanto na parte de desenvolvimento de software como na modelagem. A seguir, são apresentadas as tecnologias e ferramentas utilizadas no desenvolvimento do projeto.

### 4.1 LINGUAGENS DE PROGRAMAÇÃO

#### 4.1.1 Java <sup>1</sup>

Java é a base de praticamente todos os tipos de aplicativos em rede, e é o padrão global para desenvolvimento e fornecimento de aplicativos incorporados para jogos, conteúdo na Web e software corporativo. Com mais de 9 milhões de desenvolvedores em todo o mundo, o Java permite desenvolver e implementar aplicativos e serviços incríveis de maneira eficiente.

A Plataforma Java, permite que você desenvolva aplicativos seguros, portáteis e de alto desempenho para a maior variedade possível de plataformas de computação. Disponibilizando aplicativos em ambientes heterogêneos, as empresas podem aumentar a produtividade do usuário final, a comunicação e a colaboração, além de reduzir drasticamente o custo de propriedade de aplicativos tanto de empresas quanto de clientes.

Essa linguagem de programação é utilizada na implementação do sistema proposto por trata-se de uma linguagem que permite a implementação de um código flexível a partir dos conceitos de programação orientada a objetos.

#### 4.1.2 Typescript <sup>2</sup>

O TypeScript é uma linguagem de programação que compila para código JavaScript simples, e é executado em qualquer navegador, no Node.js ou em qualquer mecanismo de JavaScript que ofereça suporte ao ECMAScript 3 (ou mais recente).

Os tipos permitem que os desenvolvedores de JavaScript usem ferramentas e práticas de desenvolvimento altamente produtivas, como verificação estática e refatoração de código ao desenvolver aplicativos de JavaScript.

Os tipos são opcionais e a inferência de tipos permite que algumas anotações de tipo façam uma grande diferença na verificação estática do seu código. Os tipos permitem definir interfaces entre componentes de software e obter informações sobre o comportamento das bibliotecas de JavaScript existentes.

O TypeScript oferece suporte para os recursos de JavaScript mais recentes e em evolução, como funções assíncronas e decoradores, para ajudar a construir componentes robustos.

Será utilizado nesse projeto, por se tratar de uma linguagem que já vem integrada com o framework Angular JS.

---

<sup>1</sup> (Java. Informações sobre a Tecnologia Java)

<sup>2</sup> (TypeScript. Why TypeScript ?)

## 4.2 FRAMEWORKS

### 4.2.1 Angular <sup>3</sup>

Angular é um framework para desenvolvimento front-end com HTML, CSS e TypeScript que, no final, é compilado para JavaScript. Com ele, você constrói aplicações juntando trechos de código HTML, que são chamados de templates — com suas tags customizadas e classes que são marcadas com `@component` para gerenciar o conteúdo que será exibido no template. O conjunto desses dois conceitos forma um componente em Angular.

Foi escolhido para usar nesse projeto pois permite a reutilização de seus componentes.

### 4.2.2 Bootstrap <sup>4</sup>

O Bootstrap é um conjunto de ferramentas de código aberto para desenvolvimento com HTML, CSS e JS. Podemos prototipar rapidamente de nossas ideias ou criar um aplicativo completo com sistema de grade responsivo, extensos componentes pré-construídos e plugins poderosos criados no jQuery.

O termo “Bootstrap” em inglês significa “inicialização”, isto é, foca no auxílio como ponto de partida.

Por essas razões, o Bootstrap é utilizado no desenvolvimento deste projeto, especialmente na etapa de criação do layout que será utilizado em todas as páginas desse projeto.

### 4.2.3 Spring boot <sup>5</sup>

O Spring Boot é um framework que facilita a criação de aplicativos baseados em projetos de produção autônomos, que você pode "simplesmente executar". O springboot tem uma visão de opinião da plataforma Spring e de bibliotecas de terceiros para que possamos começar com um mínimo de confusão. A maioria dos aplicativos Spring Boot precisam de pouca configuração Spring.

Foi escolhido para esse projeto, pois tem um ganho muito grande na produtividade para se trabalhar.

## 4.3 BANCO DE DADOS

### 4.3.1 Postgres <sup>6</sup>

O PostgreSQL é um poderoso sistema de banco de dados relacional objeto-fonte aberto. Possui mais de 15 anos de desenvolvimento ativo e uma arquitetura comprovada que lhe valeu uma forte reputação de confiabilidade, integridade de dados e correção. Ele é executado em todos os principais sistemas operacionais, incluindo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, macOS, Solaris, Tru64) e Windows.

---

<sup>3</sup> (Guedes, 2017)

<sup>4</sup> (Bootstrap. Desenvolvendo com Bootstrap 3)

<sup>5</sup> (Spring Boot. Project)

<sup>6</sup> (Postgres. The Postgres Global Development Group, 1996-2017)

É totalmente compatível com ACID, tem suporte total para chaves estrangeiras, visualizações, disparadores e procedimentos armazenados (em vários idiomas). Inclui a maioria dos tipos de dados SQL. Ele também suporta o armazenamento de objetos grandes binários, incluindo imagens, sons ou vídeos. Possui interfaces de programação nativas para C / C ++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre outros, e documentação excepcional.

Um banco de dados de classe empresarial, o PostgreSQL possui recursos sofisticados, como o Controle de Concorrência de Multi-Versão (MVCC), recuperação de ponto-a-tempo, espaços de tabelas, replicação assíncrona, transações aninhadas (savepoints), backups em linha / hotups, um planejador / otimizador de consultas sofisticados, registrando a tolerância a falhas. Ele suporta conjuntos de caracteres internacionais, codificações de caracteres multibyte, Unicode, e é compatível com localidade para classificação, sensibilidade a maiúsculas e formatação. É altamente escalável, tanto na grande quantidade de dados que pode gerir quanto na quantidade de usuários simultâneos que pode acomodar.

Foi escolhido para se utilizar nesse projeto, pelo fato de possuir suporte pelo servidor Heroku.

## 4.4 AMBIENTE DE DESENVOLVIMENTO

### 4.4.1 Eclipse <sup>7</sup>

O Eclipse é uma plataforma de desenvolvimento de software livre extensível, baseada em Java. Por si só, é simplesmente uma estrutura e um conjunto de serviços para desenvolvimento de aplicativos de componentes de plug-in. Felizmente, o Eclipse vem com um conjunto padrão de plug-ins, incluindo as amplamente conhecidas Ferramentas de Desenvolvimento Java (JDT).

Embora a maioria dos usuários esteja satisfeita em usar o Eclipse como um ambiente de desenvolvimento integrado (IDE) Java, a plataforma ambiciona mais. O Eclipse também inclui o Plug-in Development Environment (PDE), que é de interesse principalmente daqueles que desejam estender o Eclipse, visto que ele permite desenvolver ferramentas que se integram perfeitamente ao ambiente do Eclipse. Como tudo no Eclipse é plug-in, todos os desenvolvedores de ferramentas têm um campo de ação nivelado para oferecer extensões ao Eclipse e fornecer um IDE unificado e consistente aos usuários.

Será utilizado nesse projeto pelo fato de possuir diversos plugins e uma fácil integração com a linguagem de programação Java.

### 4.4.2 Visual studio code <sup>8</sup>

O Visual Studio Code, é uma ferramenta para realizar o desenvolvimento de código fonte. Os recursos que o Visual Studio Code inclui fora da caixa são apenas o começo. As extensões de código VS permitem que você adicione linguagens, depuradores e ferramentas à sua instalação para suportar o fluxo de trabalho de desenvolvimento. O rico modelo de extensibilidade do Código permite que os autores de extensão se conectem diretamente na UI do Código VS e contribuam com a funcionalidade através das mesmas API usadas pelo Código.

Foi escolhido para esse projeto pelo fato de possuir fácil integração com o framework Angular 4.

<sup>7</sup> (Eclipse. O que é o eclipse ?)

<sup>8</sup> (Visual Studio Code. Apresentando o editor multiplataforma da Microsoft)



## 4.5 ARTEFATOS DE VERSIONAMENTO

### 4.5.1 Git<sup>9</sup>

O *Git* é um sistema de controle de versão distribuído e um sistema de gerenciamento de código fonte, com ênfase em velocidade (GIT, 2017). O *Git* foi inicialmente projetado e desenvolvido por Linus Torvalds para o desenvolvimento do *kernel Linux*, mas foi adotado por muitos outros projetos. Cada diretório de trabalho do *Git* é um repositório que possui um histórico completo e habilidade total de acompanhamento das revisões, sendo totalmente independente de acesso a uma rede ou a um servidor central, como destaca Robert (2009): “Um aspecto que realmente define *Git* e o diferencia é a sua velocidade.”

Foi escolhido para ser usado nesse projeto pelo fato de ser gratuito, além de ser uma das ferramentas de versionamento que vem se tornando uma grande tendência em diversos projetos pelo mundo.

### 4.5.2 Github<sup>10</sup>

O *GitHub* é um serviço de *Web Hosting* Compartilhado para projetos que usam o controle de versionamento Git. É escrito em *Ruby on Rails* pelos desenvolvedores da *Logical Awesome* (GITHUB, 200-).

Foi escolhido para esse projeto, porque o Github disponibiliza opções para criação e gerenciamento de sprints, criação de issues (tarefas), além não possuir custo algum dependendo do tamanho de armazenamento de dados do projeto.

## 4.6 ARTEFATO DE MODELAGEM

### 4.6.1 DeZign<sup>11</sup>

É uma ferramenta intuitiva de design de banco de dados para desenvolvedores e DBAs que podem ajudá-lo a modelar, criar e manter bancos de dados. O DeZign for Databases usa diagramas de relacionamento de entidade (ERDs) para projetar graficamente bancos de dados e gerar automaticamente os bancos de dados SQL e desktop mais populares. O DeZign for Databases oferece um ambiente sofisticado de modelagem de dados visuais para desenvolvimento de aplicativos de banco de dados que facilita muito o processo de desenvolvimento de banco de dados. A ferramenta reduz falhas no desenvolvimento de banco de dados e melhora sua produtividade. Você pode visualizar estruturas de banco de dados para entender seu banco de dados, criar novos bancos de dados ou criar sistemas de pesquisa reversa para modificar, documentar, analisar e otimizar.

Foi utilizada nesse projeto pelo fato de possuir uma versão de 30 dias gratuita, e possuir integração com o banco de dados Postgres.

<sup>9</sup> (GIT. Getting Started - About Version Control.)

<sup>10</sup> (GITHUB. Getting your project on GitHub.2016.)

<sup>11</sup> (DeZign. DeZign for Databases.)

#### 4.6.2 Bizagi <sup>12</sup>

É um software gratuito de notação e modelagem de processos de negócio (BPMN), com ferramentas totalmente baseadas em notação BPMN capazes de oferecer simplicidade na hora de modelar processos. Desenvolvido para sistemas operacionais Windows, o Bizagi está atualmente na versão 3.1 e também disponível em português.

Foi escolhido para se utilizar no projeto, pois possui versões gratuitas e online.

### 4.7 INTEGRAÇÃO CONTÍNUA

#### 4.7.1 Travis <sup>13</sup>

O Travis CI é um serviço de Integração Contínua na nuvem que pode ser conectado a repositórios no GitHub. Ele é gratuito para repositórios públicos e pago para repositórios privados.

É um serviço excelente, amplamente usado em projetos no GitHub. Possui suporte a diversas linguagens de programação. Como o Travis funciona? Toda vez que dermos um push no nosso repositório do GitHub, o Travis irá executar os seguintes passos:

Criar uma máquina virtual em branco;

Pegar o código-fonte no GitHub;

Rodar os testes;

Fazer o deploy da aplicação;

Notificar o usuário (via e-mail, por exemplo);

Foi escolhido pelo fato de trabalhar juntamente com o github e o servidor Heroku, além de ser gratuito.

### 4.8 SERVIDOR DE HOSPEDAGEM

#### 4.8.1 Heroku <sup>14</sup>

O Heroku é uma plataforma em nuvem como um serviço (PaaS) que suporta várias linguagens de programação que são usadas como um modelo de implantação de aplicativos web. O Heroku foi uma das primeiras plataformas da nuvem, está em desenvolvimento desde junho de 2007, oferecendo suporte à linguagem de programação Ruby, Java, Node.js, Scala, Clojure, Python, PHP entre outras. Por esta razão, Heroku é dito ser uma plataforma de poliglota, pois permite ao desenvolvedor criar, executar e dimensionar aplicativos de forma semelhante em todos os idiomas.

Foi escolhido pelo fato de ser gratuito e possuir fácil integração com a ferramenta Travis.

---

<sup>12</sup> (Lima, 2016)

<sup>13</sup> (Travis. Configurando Travis CI., 2014)

<sup>14</sup> (Heroku. Introdução.)

## 4.9 FERRAMENTA DE QA

### 4.9.1 **Sonar** <sup>15</sup>

O Sonar fornece o que é provavelmente o melhor analisador de código estático que você pode encontrar no mercado para PL / SQL. Com base em nosso próprio front-end do compilador PL / SQL, ele usa as técnicas mais avançadas (correspondência de padrões, análise de fluxo de dados) para analisar código e encontrar cheiros de código, bugs e vulnerabilidades de segurança. Quanto a qualquer produto que desenvolvamos no Sonar, foi construído com os seguintes princípios: profundidade, precisão e velocidade.

O SonarPLSQL possui uma ótima cobertura de padrões de qualidade bem estabelecidos. O recurso SonarPLSQL está disponível no Eclipse para desenvolvedores (SonarLint), bem como em toda a cadeia de desenvolvimento para revisão de código automatizada com o SonarClube on-site ou o SonarCloud on-line.

Foi escolhido pelo fato de possuir uma versão online e gratuita.

## 4.10 BIBLIOTECAS EXTERNAS

### 4.10.1 **Api swagger 2** <sup>16</sup>

O Swagger é uma especificação que cria o contrato RESTful para nossas API's, detalhando todos os seus recursos e operações em um formato humano e legível por máquina para fácil desenvolvimento, descoberta e integração.

Foi escolhido pelo fato de facilitar os testes das API's criadas, além de manter o código documentado.

### 4.10.2 **Api google maps** <sup>17</sup>

É uma API que oferece geocodificação e geocodificação reversa de endereços.

A geocodificação é o processo de converter endereços (como nome da rua e número) em coordenadas geográficas (latitude e longitude) que permitem a inserção de marcadores de local em um mapa ou o posicionamento do mapa.

A geocodificação inversa é o processo de conversão de coordenadas geográficas em um endereço legível. O serviço de geocodificação inversa da Google Maps Geocoding API também permite que você encontre o endereço de um determinado ID de local.

Foi escolhida essa API na intenção de ajudar o usuário final, completando um trecho do endereço após informar um CEP válido.

---

<sup>15</sup> (Sonar. Introdução.)

<sup>16</sup> (Swagger 2. Spring Boot RESTful API Documentation With Swagger 2)

<sup>17</sup> (Maps JavaScript API. Serviço Geocoding)

## 5 DESCRIÇÃO DA SOLUÇÃO

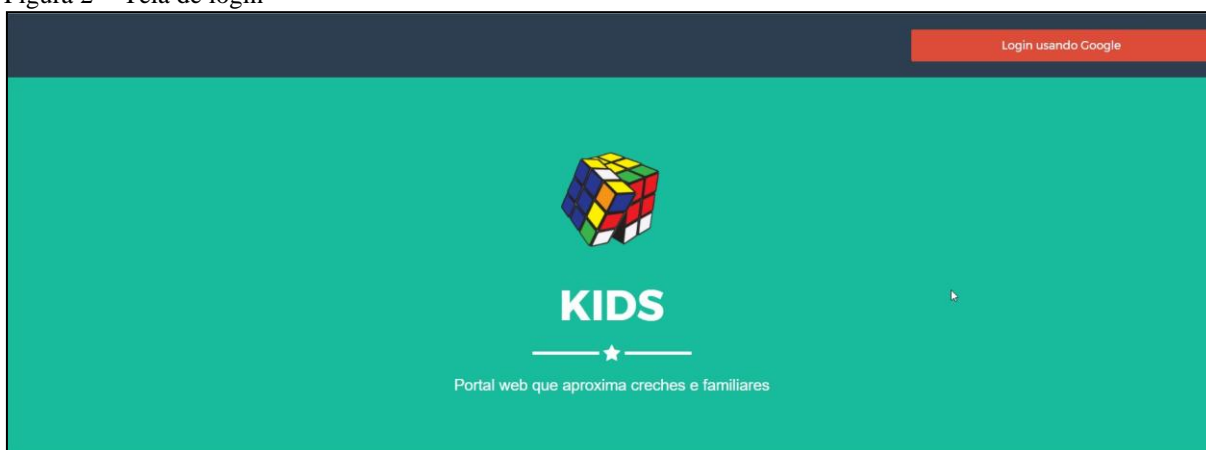
A solução que foi desenvolvida nesse projeto visou sanar todos os problemas que foram apontados pelo P.O do projeto. Para isso, foi desenvolvido uma aplicação web responsiva, contemplando as seguintes funcionalidades:

### 5.1 LOGIN

O login desenvolvido nesse portal, é realizado através de autenticação de uma conta Google, ou seja, o usuário deve informar seu e-mail e senha válidos da sua conta Google para obter acesso ao portal kids.

Para isso foi desenvolvido uma interface gráfica que disponibiliza um botão de acesso para realizar a autenticação com uma conta Google. Segue abaixo imagens do processo de autenticação:

Figura 2 - Tela de login



Fonte: construída pelo autor do projeto.

Figura 3 - Tela de seleção da conta Google



Fonte: <https://accounts.google.com/ServiceLogin/signinchooser>


Figura 4 - Tela para informar a senha da conta Google

Fazer login nas Contas do Google - Google Chrome

Seguro | [https://accounts.google.com/signin/v2/challenge/pwd?client\\_id=359998324820...](https://accounts.google.com/signin/v2/challenge/pwd?client_id=359998324820...)

Google

Luciano Ortiz

 lucianoortizsilva@gmail.com

Digite sua senha

.....

Outras maneiras de fazer login

PRÓXIMA

Português (Brasil) ▾

Ajuda Privacidade Termos

Fonte: <https://accounts.google.com/signin/v2/challenge/pwd>

## 5.2 TIPO DE CONTA

Trata-se de definir o tipo de conta que o usuário será habilitado no portal.

Fazendo uma analogia com a rede social Facebook, quando criamos uma conta devemos escolher o seu tipo, pessoa física ou pessoa jurídica. Aqui no portal kids também devemos escolher o tipo de conta ao acessar pela primeira vez a aplicação.

O portal kids oferece duas opções de tipo de conta. Uma opção é o tipo de conta **familiar**, que compreende a qualquer pessoa física que deseja obter acesso a atividades de uma ou mais crianças que estão registradas em alguma creche cadastrada no portal. Uma outra opção de conta é a conta do tipo **creche**, que compreende a qualquer pessoa jurídica que deseja disponibilizar informações sobre a creche e atividades realizadas pelas crianças que estão matriculadas na mesma.

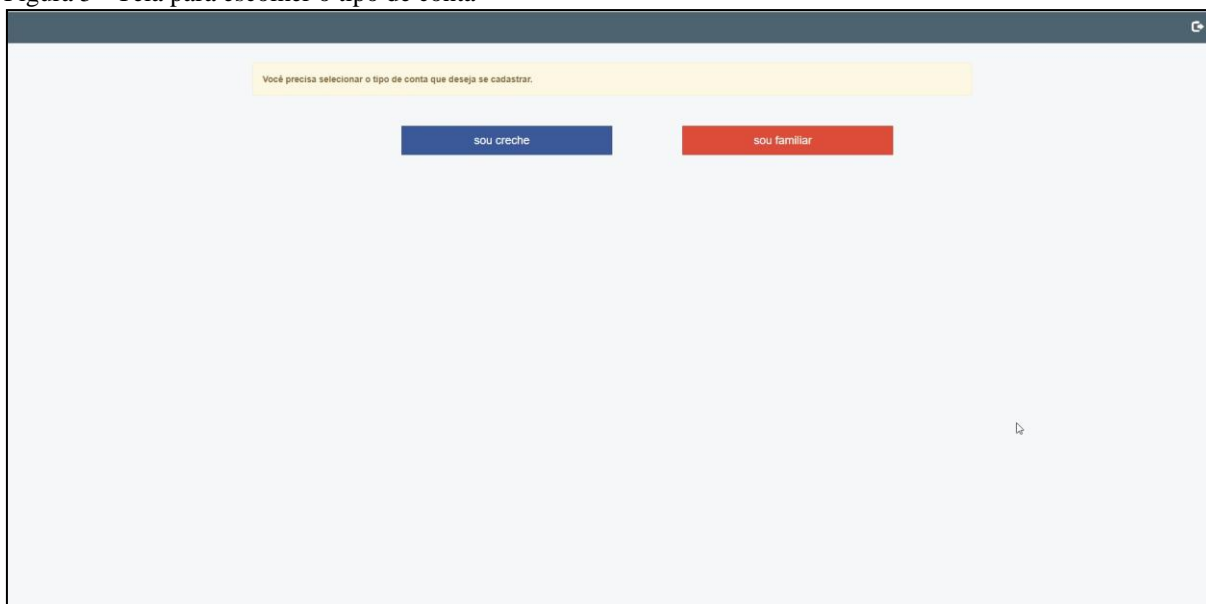
Para isso foi desenvolvido uma interface gráfica que disponibiliza duas opções de criação de conta, um botão para o tipo **familiar** e um outro botão para o tipo **creche**.

Ao clicar no botão do tipo **familiar**, uma conta é criada automaticamente, ou seja, o usuário já terá visibilidade a todas atividades das crianças das quais o mesmo possui algum vínculo familiar. Caso o familiar ainda não possua vínculos com alguma criança (*os vínculos são cadastrados por uma creche*), o mesmo permanecerá inativo no sistema, até que alguma creche realize o registro de seu vínculo.

Ao clicar no botão do tipo **creche**, uma conta é criada automaticamente, ou seja, a creche poderá registrar quaisquer tipos de dados competentes a mesma.

Segue abaixo imagem da interface para a escolha do tipo de conta a ser cadastrada:

Figura 5 - Tela para escolher o tipo de conta

A interface apresenta um fundo cinza claro. No topo, uma barra escura contém um ícone de menu. Abaixo, uma mensagem amarela indica: "Você precisa selecionar o tipo de conta que deseja se cadastrar.". Duas botões retangulares estão centralizados: um azul com o texto "sou creche" e um vermelho com o texto "sou familiar".

Fonte: construída pelo autor do projeto.

### 5.3 PERFIL

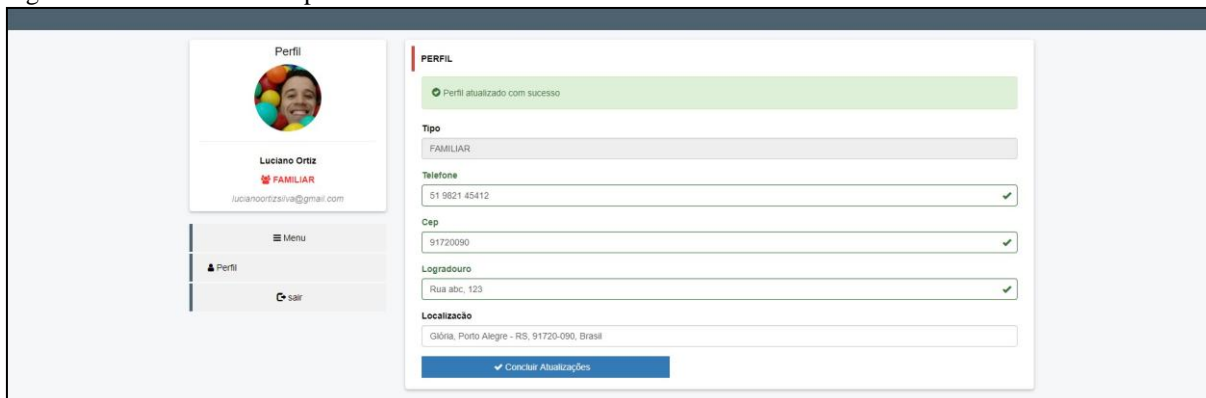
Trata-se de uma área onde o usuário poderá manter seus dados principais de cadastro devidamente atualizados, afim de serem utilizados em diversos pontos do portal kids.

Nessa área conta com dados de telefone e endereço.

Nos dados que compreendem ao endereço, o usuário poderá informar primeiramente o seu CEP, assim o restante de seu endereço será preenchido de forma automática.

Para atender essa funcionalidade, foi desenvolvida uma interface gráfica como segue abaixo:

Figura 6 - Tela de dados do perfil do usuário

A interface é dividida em duas colunas. A coluna da esquerda, sob o título "Perfil", mostra uma foto de perfil, o nome "Luciano Ortiz", o status "FAMILIAR" com um ícone de família, e o e-mail "lucianoortizsiva@gmail.com". Abaixo há um menu com opções "Menu", "Perfil" (destacada) e "sair". A coluna da direita, sob o título "PERFIL", exibe uma mensagem verde de sucesso: "Perfil atualizado com sucesso". Seguem campos de formulário para "Tipo" (FAMILIAR), "Telefone" (51 9821 45412), "Cep" (91720090), "Logradouro" (Rua abc, 123) e "Localização" (Glória, Porto Alegre - RS, 91720-090, Brasil). Cada campo possui um ícone de confirmação verde à direita. No final, há um botão azul com o texto "Concluir Atualizações".

Fonte: construída pelo autor do projeto.

## 5.4 CRIANÇAS

Nessa funcionalidade tanto um usuário de perfil creche, quanto um usuário do perfil familiar, obterão acesso para visualizar e editar dados a respeito das crianças.

Cabe ao usuário do perfil creche as atribuições de cadastrar as crianças matriculadas na mesma. Nesse cadastro são solicitados dados gerais da criança, assim como dados para contato, e endereço. Haverá casos em que a criança necessite de medicação durante seu horário escolar, e/ou possua certas alergias que requerem cuidados especiais, sendo necessário o cadastro desse tipo de informação no momento do cadastro ou edição dos dados da criança.

Cabe ao usuário do perfil familiar as atribuições de manter os dados da criança devidamente atualizados.

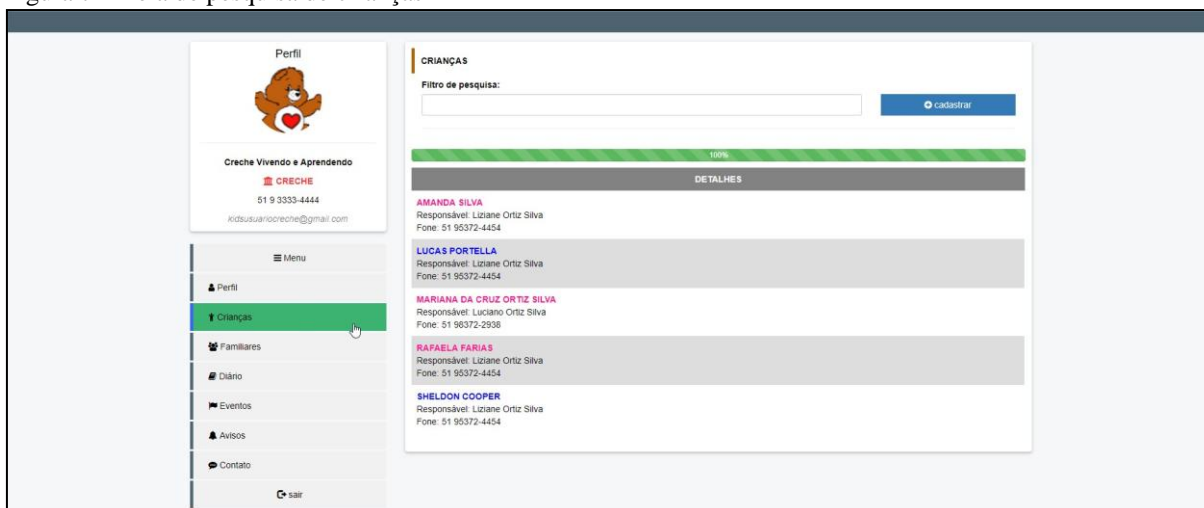
- **Regras de negócio aplicada:**

- ✓ O cadastro de novas crianças em uma creche está atribuído apenas aos usuários do perfil da creche em questão, ficando proibido o cadastro de novas crianças por usuário do perfil familiar;
- ✓ Os usuários do perfil creche, poderão visualizar/editar apenas dados de crianças matriculadas em sua creche;
- ✓ As crianças cadastradas na creche, deverão possuir códigos de matrícula distintos;
- ✓ Os usuários do perfil familiar, apenas poderão visualizar / editar dados de crianças nas quais o mesmo possui vínculo (*o vínculo se dá através da funcionalidade de familiares*);

Para atender esses requisitos, foram desenvolvidas algumas interfaces gráficas no estilo de wizard (passo a passo), para auxiliar o cadastro dessas informações em pequenas partes.

A seguir são mostradas as telas que correspondem a pesquisa e cadastro/edição de crianças no portal:

Figura 7 – Tela de pesquisa de crianças



Fonte: construída pelo autor do projeto.

Figura 8 – Tela de cadastro/edição de dados gerais (Passo 01)

Perfil

Creche Vivendo e Aprendendo  
CRECHE  
51 9 3333-4444  
kidsusuario@creche@gmail.com

Menu

- Perfil
- Crianças
- Familiares
- Diário
- Eventos
- Avisos
- Contato
- Sair

CRIANÇAS

1º DADOS GERAIS

Matricula\*  
0001

Nome\*  
Mariana De Cruz Ortiz Silva

Data de Nascimento\*  
29/08/2011

Sexo\*  
FEMININO

Voltar Próximo

(\*) preenchimento obrigatório

Fonte: construída pelo autor do projeto.

Figura 9 – Tela de cadastro/edição de endereço (Passo 02)

Perfil

Creche Vivendo e Aprendendo  
CRECHE  
51 9 3333-4444  
kidsusuario@creche@gmail.com

Menu

- Perfil
- Crianças
- Familiares
- Diário
- Eventos
- Avisos
- Contato
- Sair

CRIANÇAS

2º ENDEREÇO

Cep\*  
91720090

Logradouro\*  
Rua Engenheiro Homero Carlos Simon 9999

Localização\*  
Restinga Nova, Porto Alegre - RS, 91790-330, Brasil

Anterior Próximo

(\*) preenchimento obrigatório

Fonte: construída pelo autor do projeto.



Figura 10 – Tela de cadastro/edição de contato (Passo 03)

Perfil

Creche Vivendo e Aprendendo  
CRECHE  
51 9 3333-4444  
kidsusuariofamiliar@gmail.com

Menu

- Perfil
- Crianças
- Familiares
- Diário
- Eventos
- Avisos
- Contato
- sair

**CRIANÇAS**

3º CONTATO

Responsável Legal\*  
Luciano Ortiz Silva

E-mail\*  
kidsusuariofamiliar@gmail.com

Fone Principal\*  
51 98372-2938

Fone Outros  
51 3232-2322

Anterior Próximo

(\*) preenchimento obrigatório

Fonte: construída pelo autor do projeto.

Figura 11 – Tela de cadastro/edição de medicamentos (Passo 04)

Perfil

Creche Vivendo e Aprendendo  
CRECHE  
51 9 3333-4444  
kidsusuariofamiliar@gmail.com

Menu

- Perfil
- Crianças
- Familiares
- Diário
- Eventos
- Avisos
- Contato
- sair

**CRIANÇAS**

4º MEDICAMENTOS

Medicamento  
Digite o nome do medicamento...

Dosagem  
exemplo: 1 comprimido de 250 mg

Intervalo de horas  
exemplo: a cada 6h

Data Final  
dd/mm/aaaa

Adicionar

**DETALHES**

- melhoral infantil
- 1 comprimido / 250gr
- a cada 12h
- até o dia 11/12/2017

Remover

Anterior Próximo

Fonte: construída pelo autor do projeto.

Figura 12 – Tela de cadastro/edição de alergias (Passo 05)

Fonte: construída pelo autor do projeto.

## 5.5 FAMILIARES

Essa funcionalidade é onde ocorre a etapa de vínculos de uma ou mais crianças, com um ou mais usuários do perfil familiar.

Esse vínculo ocorre quando um usuário do tipo creche acessa o menu de familiares, seleciona uma criança, e informa dados básicos, tais como o nome, e-mail e grau de parentesco do usuário familiar.

Após esse cadastro, o usuário familiar receberá permissão para acessar quaisquer atividades relacionadas a criança no qual foi vinculado.

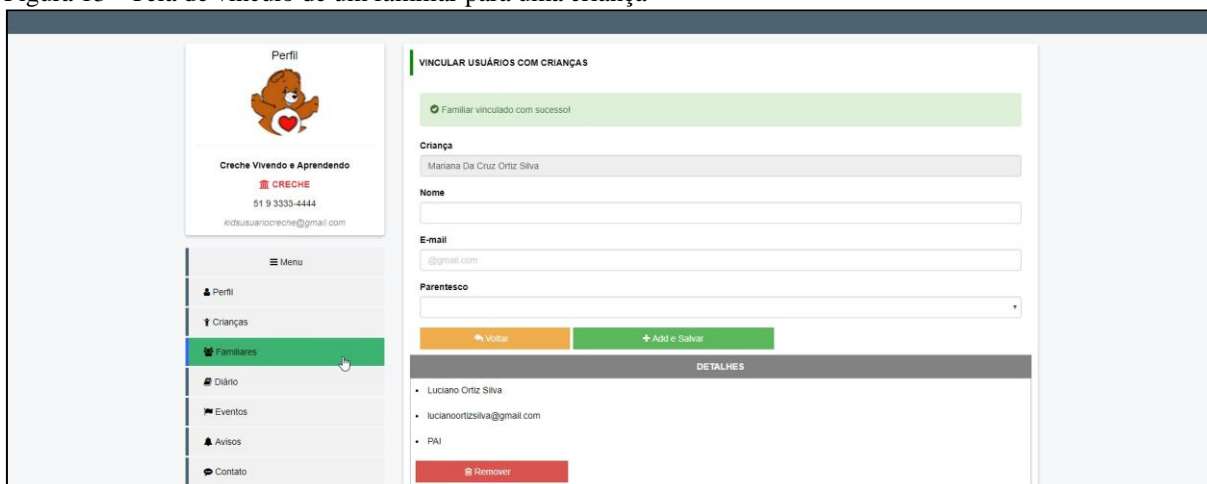
- **Regras de negócio aplicada:**

- ✓ O menu de familiares será visível apenas para os usuários do perfil do tipo creche;
- ✓ Atualmente o sistema deverá aceitar apenas cadastros de usuários que possuam conta do Google;
- ✓ Após a creche vincular alguma criança com algum usuário do tipo familiar, esse usuário familiar deverá receber um e-mail, informado que o mesmo acaba de receber permissão para acessar as atividades da criança em questão no portal;
- ✓ O usuário do tipo creche poderá remover a permissão de acesso de quaisquer familiares que possuam vínculo há alguma criança de sua creche;

Para atender esses requisitos, foi desenvolvida uma interface gráfica, assim como um corpo de e-mail.

A seguir são mostradas as telas que correspondem a ao vínculo de familiares com uma ou mais crianças de uma determinada creche, e o e-mail correspondente a notificação de vínculo ocorrida:

Figura 13 - Tela de vínculo de um familiar para uma criança



Fonte: construída pelo autor do projeto.

Figura 14 - Corpo do e-mail referente ao vínculo realizado



Fonte: construída pelo autor do projeto.

## 5.6 DIÁRIO

Essa funcionalidade trata de avaliar algumas atividades realizadas por uma criança dentro do intervalo de 24h horas.

As atividades avaliadas são: Café da manhã, Almoço, Café da tarde, Janta, Sono e Evacuação da criança.

O método de avaliação é dados através da intensidade do fato ocorrido, tais como: Pouco, Normal ou Muito.

- **Regras de negócio aplicada:**

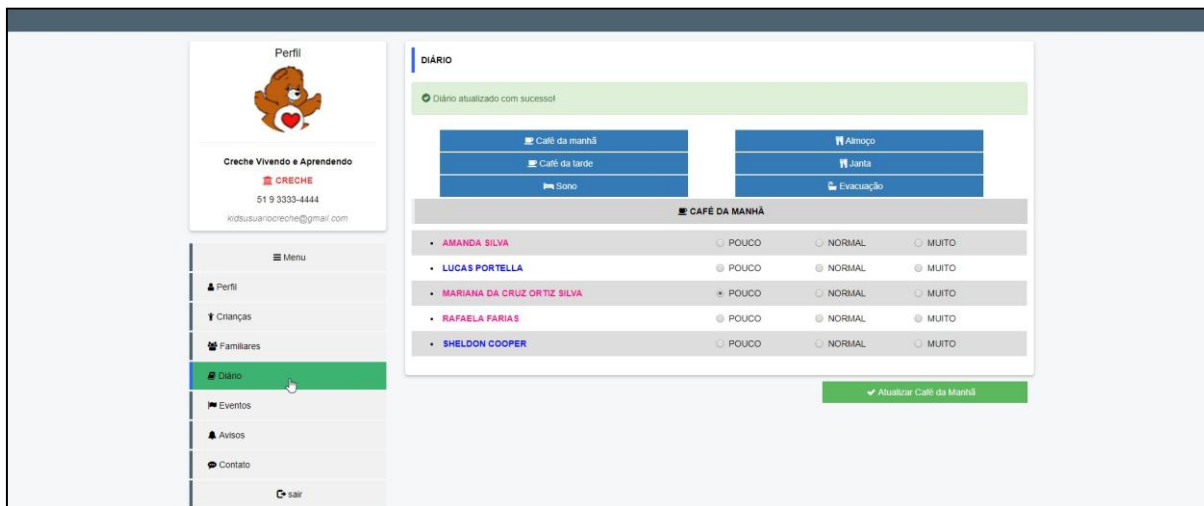
- ✓ Apenas os usuários do perfil creche são capazes de avaliar uma criança;
- ✓ Os usuários do perfil familiar apenas poderão visualizar os diários das crianças nos quais o mesmo possui vínculo;
- ✓ Será possível avaliar crianças apenas ao dia corrente, não podendo realizar avaliações retroativas;

- ✓ Os usuários do tipo familiar apenas poderão visualizar os diários do dia corrente, não sendo possível visualizar diários retroativos;

Para atender esses requisitos, foi desenvolvida uma interface gráfica para o usuário do perfil creche inserir a avaliação para cada criança, e outra interface gráfica para que o perfil do usuário familiar visualizar as avaliações cadastradas.

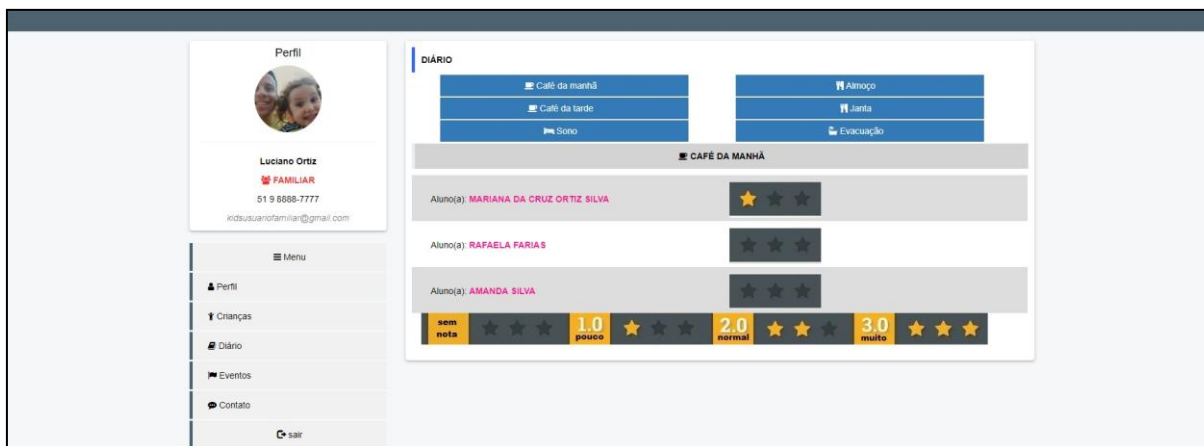
A seguir são mostradas as telas correspondentes a esses diários:

Figura 15 - Tela de diários (visão creche)



Fonte: construída pelo autor do projeto.

Figura 16 - Tela de diários (visão familiar)



Fonte: construída pelo autor do projeto.

## 5.7 EVENTOS

Essa funcionalidade trata de controlar eventos gerados por uma determinada creche.

Cabe a creche realizar o cadastro de evento em que a mesma irá realizar, informando dados como: nome do evento, a data e hora em que irá ocorrer, e a descrição desse evento.

Cabe ao familiar aceitar ou rejeitar a presença da criança a um determinado evento que tenha sido convidado.

- **Regras de negócio aplicada:**

- ✓ O cadastro de eventos se dará somente através do usuário com perfil do tipo creche;
- ✓ Não será permitido a criação de eventos com data/hora igual ou inferior a data atual do cadastro;
- ✓ O usuário do perfil creche poderá atualizar o status do evento;
- ✓ Não será possível atualizar um evento para o status de previsto quando a data/hora do evento for menor ou igual a data/hora corrente;
- ✓ A cada evento criado por uma creche, todos os usuários do tipo familiar que possuírem vínculos com crianças dessa creche, receberão um convite para que suas crianças confirmem ou rejeitem a presença nesse evento;
- ✓ A notificação que um familiar receberá, se dará através do familiar acessar o menu de eventos, após o mesmo autenticar-se no portal;

Para atender esses requisitos, foi desenvolvida uma interface gráfica para o usuário do perfil creche cadastrar seus eventos, e outra interface gráfica para que o perfil do usuário familiar possa receber esses convites, e aceitar ou rejeitar a presença nos mesmos.

A seguir são mostradas as telas correspondentes a esses eventos:

Figura 17 - Tela de cadastro e lista de eventos (visão creche)

A imagem mostra a interface de usuário de uma aplicação web, acessada no navegador em <https://kids-web.herokuapp.com/kids/home/evento>. A interface é dividida em duas seções principais: uma barra lateral esquerda com o menu e uma área principal de conteúdo.

**Barra Lateral (Menu):**

- Perfil: Exibe o nome "Creche Vivendo e Aprendendo", o logotipo "CRECHE" e o contato "51 9 3333-4444" e "kidsusuario@creche@gmail.com".
- Crianças
- Familiares
- Diário
- Eventos** (destacado em verde)
- Avisos
- Contato

**Área Principal (Eventos):**

**Formulário de Cadastro:**

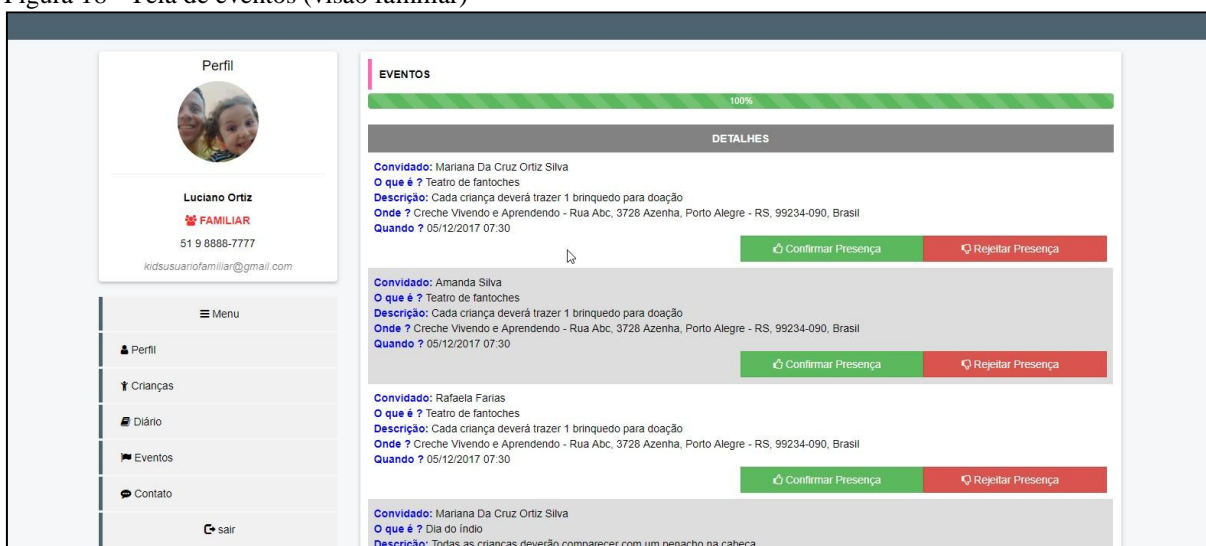
- Nome\***: Campo de texto obrigatório.
- Data de Realização\***: Campo de data no formato dd/mm/aaaa.
- Descrição\***: Campo de texto maior para detalhes do evento.
- Botão **Salvar** (verde) para confirmar o cadastro.
- Indicador de progresso: 100% (verde).
- Nota: (\*) preenchimento obrigatório.

**Detalhes dos Eventos:**

- Evento:** Dia do vampiro
- Data:** 29/12/2018 21:00
- Status: ☒ Previsto, ☐ Realizado, ☐ Cancelado.
- Botão **Atualizar** (verde) para modificar o status.
- Evento listado abaixo: Dia das bruxas, 31/10/2018 21:00.

Fonte: construída pelo autor do projeto.

Figura 18 - Tela de eventos (visão familiar)



Fonte: construída pelo autor do projeto.

## 5.8 AVISOS

Essa é uma funcionalidade que serve para as creches notificarem aos familiares cadastrados no portal, sobre algum fato urgente / importante referente a algo relacionado a creche.

Cabe a creche realizar o cadastro desse aviso, informando uma breve descrição sobre o fato, o tipo desse aviso (a nível de informação, ou para informar algum cancelamento), e uma data de expiração, que corresponde até qual data o usuário do tipo familiar irá visualizar esse aviso.

Cabe ao usuário do perfil familiar visualizar os avisos lançados pela creche. Esses avisos estarão visíveis para o usuário familiar no momento do login no portal, sendo exibidos em forma de lista no dashboard da aplicação.

- **Regras de negócio aplicada:**

- ✓ O cadastro de novos avisos é de competência apenas dos usuários do perfil creche;
- ✓ O usuário do tipo familiar irá visualizar os avisos gerados apenas das creches onde o mesmo possuir vínculos com as crianças matriculadas;
- ✓ O usuário familiar irá visualizar esses avisos até a data de expiração cadastradas pela creche;
- ✓ O usuário do perfil familiar irá visualizar esses avisos apenas no momento do login no portal, não havendo links para acessar essas informações novamente;
- ✓ Na visão de usuário do tipo familiar, deverá ser destacado de forma diferente quando um aviso for do tipo a nível de informação, ou a nível de cancelamento;

A seguir são mostradas as telas correspondentes ao processo de avisos:

Figura 19 - Tela de Avisos - (visão da creche)

Perfil

Creche Vivendo e Aprendendo

CRECHE

51 9 3333-4444

kidsusuariocreche@gmail.com

Perfil

Crianças

Familiares

Diário

Eventos

**Avisos**

sair

AVISOS

Descrição\*

Tipo\*

Data de expiração\*

dd/mm/aaaa

Salvar

(\*) preenchimento obrigatório

AVISOS
Não haverá aula no dia 12/12/12. Motivo: Limpeza da caixa de água
remover
O Evento de final de ano previsto para 29/12/2018 está Cancelado!
remover

https://kids-web.herokuapp.com/kids/home/aviso

Fonte: construída pelo autor do projeto.

Figura 20- Tela de Avisos - (visão da familiar)

Perfil

Luciano Ortiz

FAMILIAR

51 9 8888-7777

kidsusuariofamiliar@gmail.com

Perfil

Crianças

Diário

Eventos

**Avisos**

sair

Olá Luciano Ortiz

AVISOS

**Aviso informativo**

Creche Vivendo e Aprendendo, informa:  
Não haverá aula no dia 12/12/12. Motivo: Limpeza da caixa de água

**Aviso Cancelamento**

Creche Vivendo e Aprendendo, informa:  
O Evento de final de ano previsto para 29/12/2018 está Cancelado!

**Aviso informativo**

Creche Vivendo e Aprendendo, informa:  
O aluno(a) Rafaela Farias, possui presença confirmada no evento: Dia do índio. Data: 19-04-2018 14:30

Fonte: construída pelo autor do projeto.

## 5.9 INTERAÇÃO

Essa funcionalidade é onde permite ocorrer uma comunicação, a partir de uma ação do usuário com papel de familiar.

Esse usuário do perfil familiar, pode enviar uma mensagem para alguma creche, sinalizando o tipo de ocorrência, que se divide em três tipos: Reclamação, Sugestão ou Elogio.

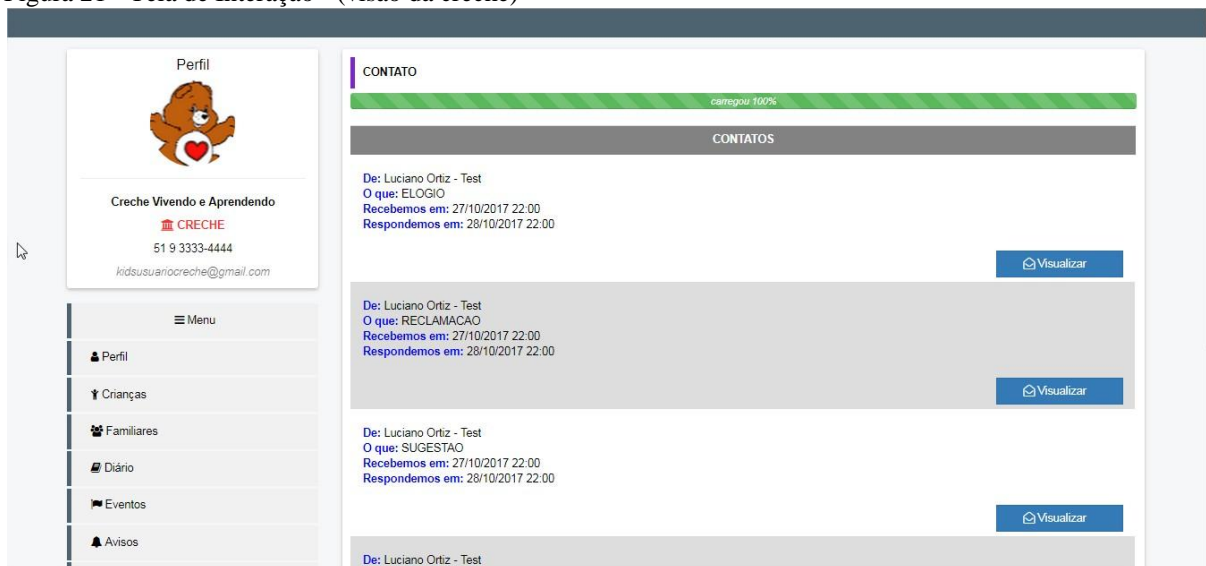
Já o usuário com o perfil de creche, poderá visualizar as mensagens recebidas de familiares, assim como disponibilizar um feedback para cada familiar.

- **Regras de negócio aplicada:**

- ✓ Cada familiar poderá enviar mensagens somente para as creches nas quais possuem alguma criança matriculada, que o mesmo tenha algum vínculo.

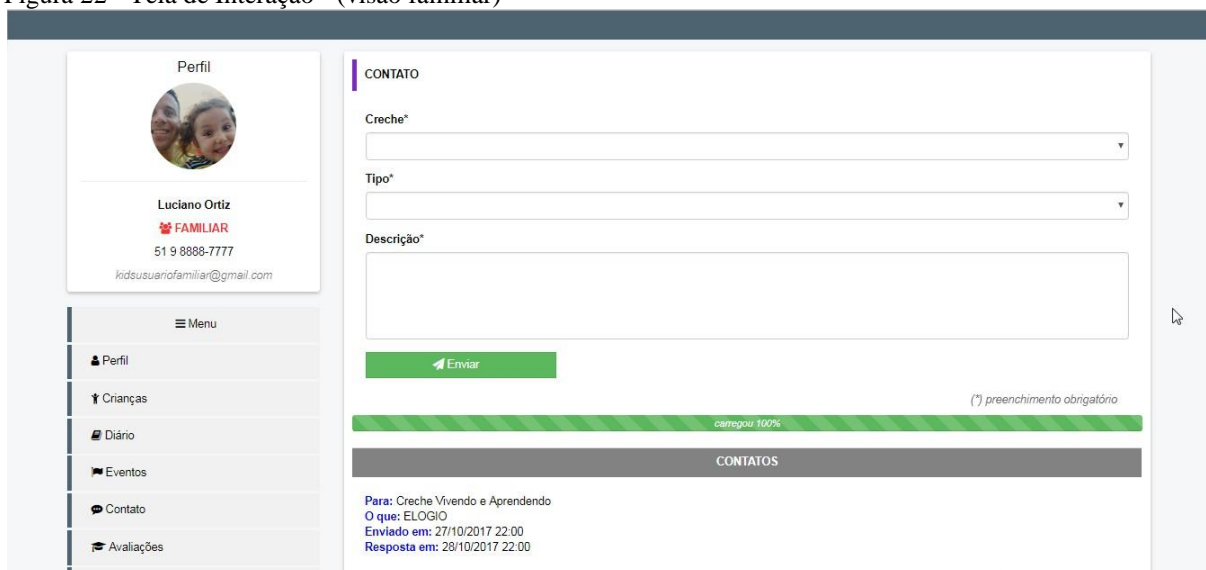
A seguir são mostradas as telas correspondentes ao processo de interação:

Figura 21 - Tela de Interação - (visão da creche)



Fonte: construída pelo autor do projeto.

Figura 22 - Tela de Interação - (visão familiar)



Fonte: construída pelo autor do projeto.



## 5.10 AVALIAÇÃO

Essa funcionalidade permite que as creches realizem uma ou mais avaliações, de suas crianças matriculadas, de uma maneira descritiva. Afim de disponibilizar aos familiares dessas crianças, quaisquer informações sobre os rendimentos escolares dos alunos.

- **Regras de negócio aplicada:**

- ✓ Cada familiar poderá visualizar as avaliações somente das crianças nas quais possuem vínculo de familiar.
- ✓ Uma creche não poderá remover avaliações já realizadas, apenas retificá-las.

A seguir são mostradas as telas correspondentes ao processo de avaliações:

Figura 23 - Tela de Avaliações - (visão creche)

**Perfil**

Creche Vivendo e Aprendendo  
**CRECHE**  
 51 9 3333-4444  
 kidsusuariocreche@gmail.com

**Menu**

- Perfil
- Crianças
- Familiares
- Diário
- Eventos
- Avisos

**AVALIAÇÕES**

Filtro de pesquisa:

carregou 100%

**DETALHES**

<b>Amanda Silva</b> Responsável: Liziane Ortiz Silva Fone: 51 95372-4454	<a href="#">avaliações</a>
<b>Lucas Portella</b> Responsável: Liziane Ortiz Silva Fone: 51 95372-4454	<a href="#">avaliações</a>
<b>Mariana Da Cruz Ortiz Silva</b> Responsável: Luciano Ortiz Silva Fone: 51 95372-2938	<a href="#">avaliações</a>
<b>Rafaela Farias</b> Responsável: Liziane Ortiz Silva Fone: 51 95372-4454	<a href="#">avaliações</a>

Fonte: construída pelo autor do projeto.

Figura 24 - Tela de Avaliações - (visão familiar)

**Perfil**

Luciano Ortiz  
**FAMILIAR**  
 51 9 8888-7777  
 kidsusuariofamiliar@gmail.com

**Menu**

- Perfil
- Crianças
- Diário
- Eventos
- Contato
- Avaliações

**CONTATO**

Creche\*

Tipo\*

Descrição\*

[Enviar](#)

(\*) preenchimento obrigatório

carregou 100%

**CONTATOS**

Para: Creche Vivendo e Aprendendo  
 O que: ELOGIO  
 Enviado em: 27/10/2017 22:00  
 Resposta em: 28/10/2017 22:00

## 5.11 CARDÁPIO

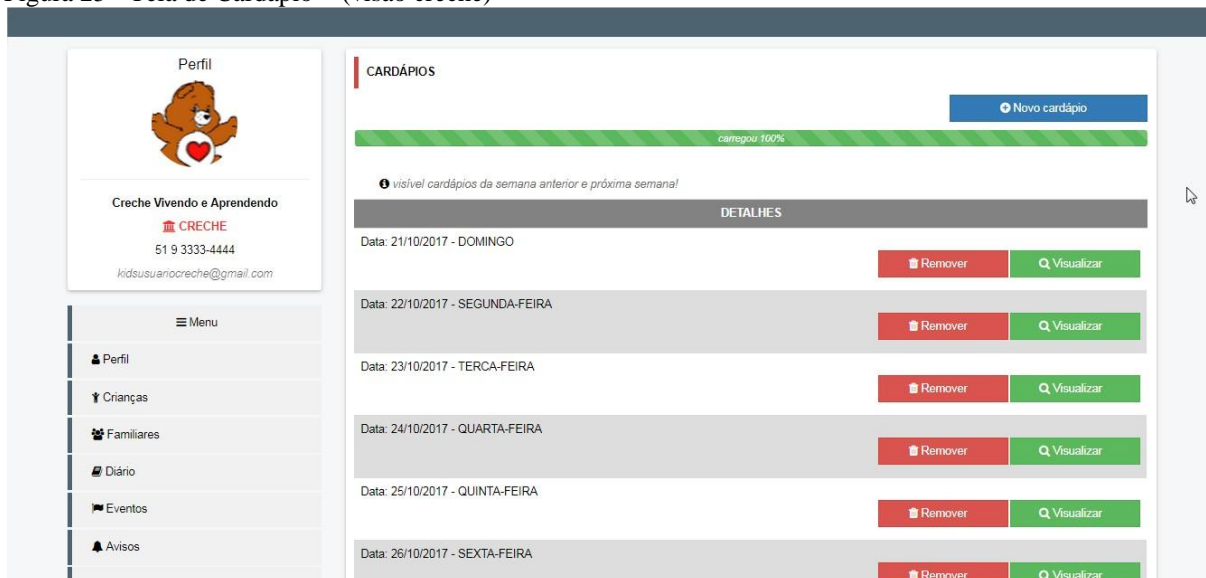
Essa funcionalidade consiste em que as creches disponibilizem de uma forma online, todos os itens da alimentação diária que será servida na creche. Afim de disponibilizar aos familiares informações sobre a dieta das crianças.

- **Regras de negócio aplicada:**

- ✓ Cada familiar poderá visualizar o cardápio apenas das creches onde possuem alguma criança matriculada.
- ✓ As informações de cardápios estarão disponíveis somente para o dia atual, ou uma semana anterior ao dia atual (dia atual - 7 dias), ou de 1 semana posterior ao dia atual (dia atual + 7 dias).
- ✓ Não será permitido mais de 1 cadastro de cardápio para um mesmo dia.
- ✓ Uma creche poderá remover um cardápio a qualquer momento.

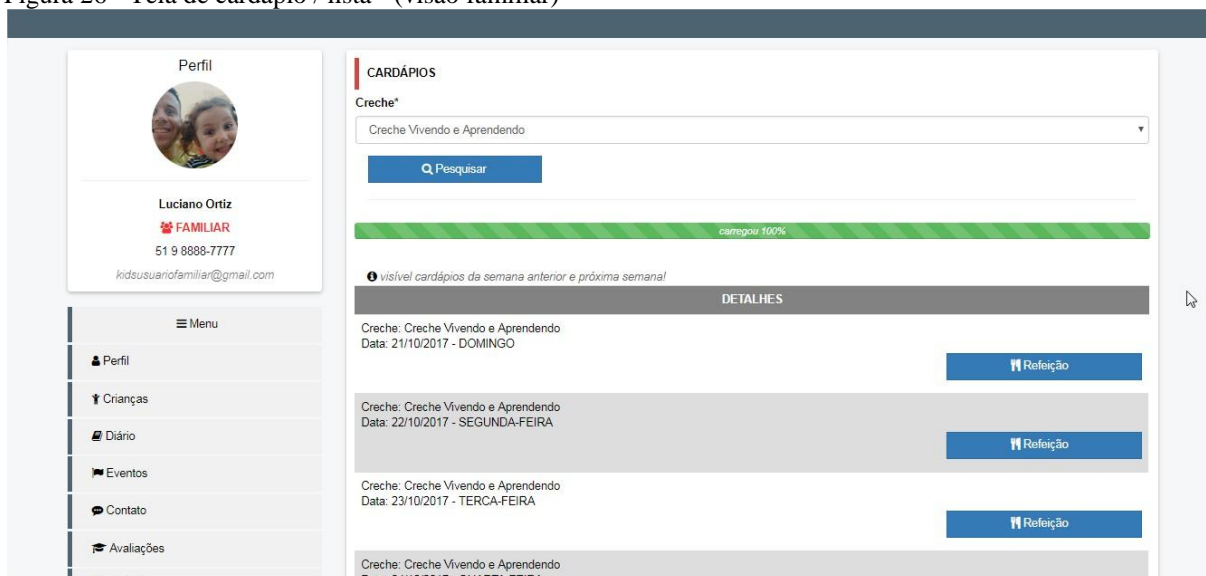
A seguir são mostradas as telas correspondentes ao cardápio:

Figura 25 - Tela de Cardápio - (visão creche)



Fonte: construída pelo autor do projeto.

Figura 26 - Tela de cardápio / lista - (visão familiar)



Fonte: construída pelo autor do projeto.

## 5.12 GALERIA DE FOTOS

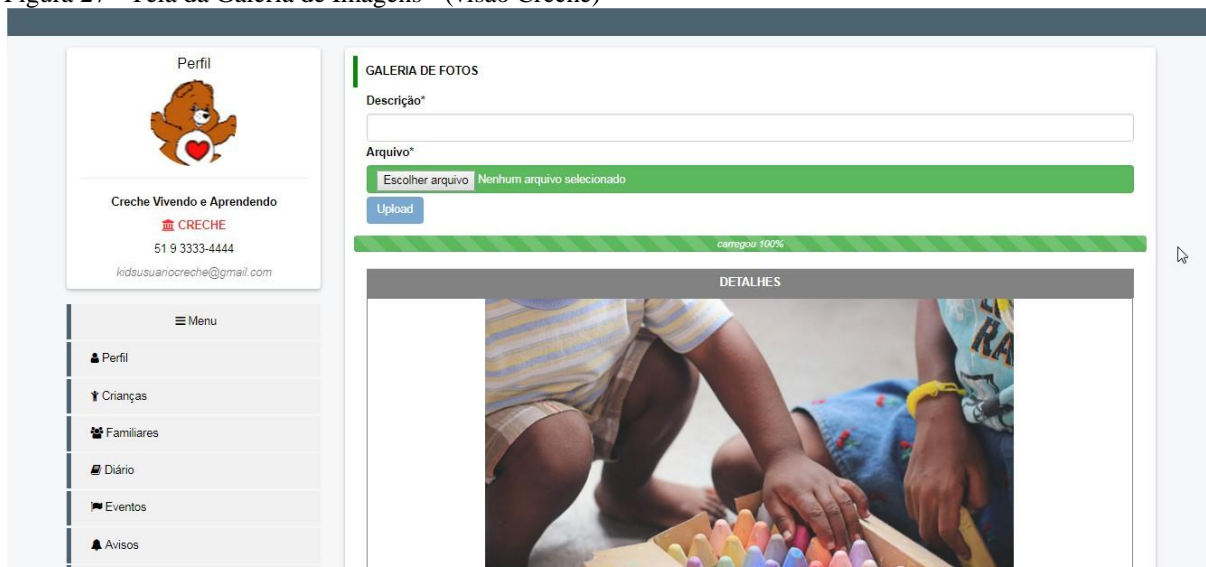
Essa funcionalidade permite que as creches realizem upload de fotos retiradas na creche, afim de disponibilizar aos familiares imagens das atividades realizadas na mesma.

- **Regras de negócio aplicada:**

- ✓ As creches poderão realizar upload apenas de imagens nos formatos de jpg, jpeg ou png.

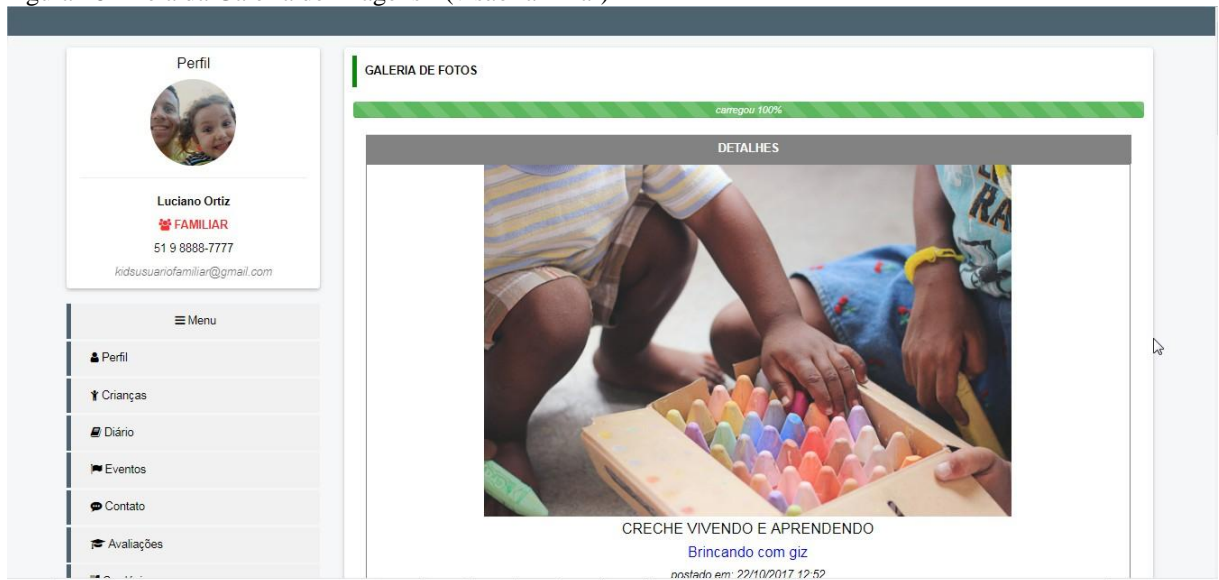
A seguir são mostradas as telas correspondentes a galeria de imagens:

Figura 27 - Tela da Galeria de Imagens - (visão Creche)



Fonte: construída pelo autor do projeto.

Figura 28 - Tela da Galeria de Imagens - (visão familiar)



## 6 ABORDAGEM DE DESENVOLVIMENTO

A estratégia de desenvolvimento do sistema que foi utilizada durante a construção deste projeto foi o Scrum – onde o principal aspecto da metodologia Scrum é a iteração. Por se tratar de uma prática em grupo, apenas alguns artefatos serão utilizados no presente projeto.

O que é Scrum ?

Scrum é um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível. Scrum é:

- Leve;
- Simples de entender;
- Extremamente difícil de dominar;

Scrum é um framework estrutural que está sendo usado para gerenciar o desenvolvimento de produtos complexos desde o início de 1990. Scrum não é um processo ou uma técnica para construir produtos; em vez disso, é um framework dentro do qual você pode empregar vários processos ou técnicas. O Scrum deixa claro a eficácia relativa das práticas de gerenciamento e desenvolvimento de produtos, de modo que você possa melhorá-las.

O framework Scrum consiste nos times do Scrum associadas a papéis, eventos, artefatos e regras. Cada componente dentro do framework serve a um propósito específico e é essencial para o uso e sucesso do Scrum.

As regras do Scrum integram os eventos, papéis e artefatos, administrando as relações e interações entre eles. As regras do Scrum são descritas ao longo deste documento. Estratégias específicas para o uso do framework Scrum variam e são descritas em outros documentos (SCHWABER, 2013 - pág. 3).

Como o Scrum foi aplicado nesse projeto?

O Scrum foi aplicado nesse projeto da seguinte maneira:

O Product Backlog foi criado no início do projeto, juntamente com a Project Owner Liziane Ortiz (funcionária da creche "Vivendo e Aprendendo"). Nesse momento foram levantadas diversas User Stories, e já organizadas por ordem de prioridade.

Logo após o Product Backlog ter sido definido, foram planejados todos os Sprints Backlog, juntamente com o professor orientador, afim de organizar uma melhor forma para preencher um determinado sprint por um período de 15 dias.

Passado o final de 15 dias de desenvolvimento, as funcionalidades foram entregues para a Liziane Ortiz (Project Owner), onde a mesma teve a oportunidade para "dar o aceite", sugerir melhorias, e sugerir novas funcionalidades para o projeto.

Ao final de cada sprint, foi realizada uma retrospectiva juntamente com o professor orientador, apontando os pontos positivos e as dificuldades encontradas durante o sprint, além de mostrar tudo o que foi entregue para o cliente.

## 7 ARQUITETURA DO SISTEMA

Afim de obter uma boa gestão deste projeto, foi construído uma organização no Github chamada **kids-tcc-senacrs**, que pode ser acessada através do link <https://github.com/kids-tcc-senacrs>. Dentro desta organização foram criados quatro (4) repositórios, conforme descritos logo abaixo:

- **kids-core:**

Esse repositório corresponde ao armazenamento de todos os artefatos que foram realizados no desenvolvimento do back-end.

Nesse projeto foi utilizado a linguagem de programação Java, juntamente com o framework Spring Boot.

O conteúdo desse projeto ficou responsável por suprir toda a demanda relativa as regras de negócio, disponibilizando diversas API's a serem consumidas pelo projeto responsável pelo front-end.

O mesmo encontra-se disponível em: <https://github.com/kids-tcc-senacrs/kids-core>

- **kids-web:**

Esse repositório corresponde ao armazenamento de todos os artefatos que foram realizados no desenvolvimento do front-end.

No desenvolvimento deste projeto, foi utilizado o framework Angular, na versão 4, onde implicitamente foram utilizados artefatos de TypeScript, CSS, e HTML.

O conteúdo deste projeto ficou responsável por suprir toda a demanda relativa as interfaces disponibilizadas para o usuários final, onde a exibição de registros nas telas, se deu através do consumo das API's disponibilizadas pelo projeto de back-end.

O mesmo encontra-se disponível em: <https://github.com/kids-tcc-senacrs/kids-web>

- **kids-db:**

Esse repositório corresponde ao armazenamento de todos os scripts SQL utilizados para realizar os testes da aplicação.

O mesmo encontra-se disponível em: <https://github.com/kids-tcc-senacrs/kids-db>

- **kids-doc:**

Esse repositório corresponde ao armazenamento dos artefatos utilizados na gestão do projeto. Podendo ser visualizado todos os registros das entregas realizadas deste projeto.

O mesmo encontra-se disponível em: <https://github.com/kids-tcc-senacrs/kids-doc>

Além destes quatro (4) repositórios criados, também foram utilizadas outras ferramentas externas, conforme descritas logo abaixo:

- **Travis:** Foi a ferramenta utilizada para realizar todo o processo de Integração Contínua da aplicação.

O mesmo encontra-se disponível em: <https://travis-ci.org/kids-tcc-senacrs>

- **Sonar:** Foi a ferramenta utilizada para avaliar a qualidade do código fonte gerado.

O mesmo encontra-se disponível em: <https://sonarcloud.io/dashboard?id=com.kids>

- **Heroku:** Foi utilizado como o servidor de hospedagem da aplicação.  
O mesmo encontra-se disponível em: <https://kids-web.herokuapp.com/kids/login>

## 7.1 PROCESSO DE DESENVOLVIMENTO

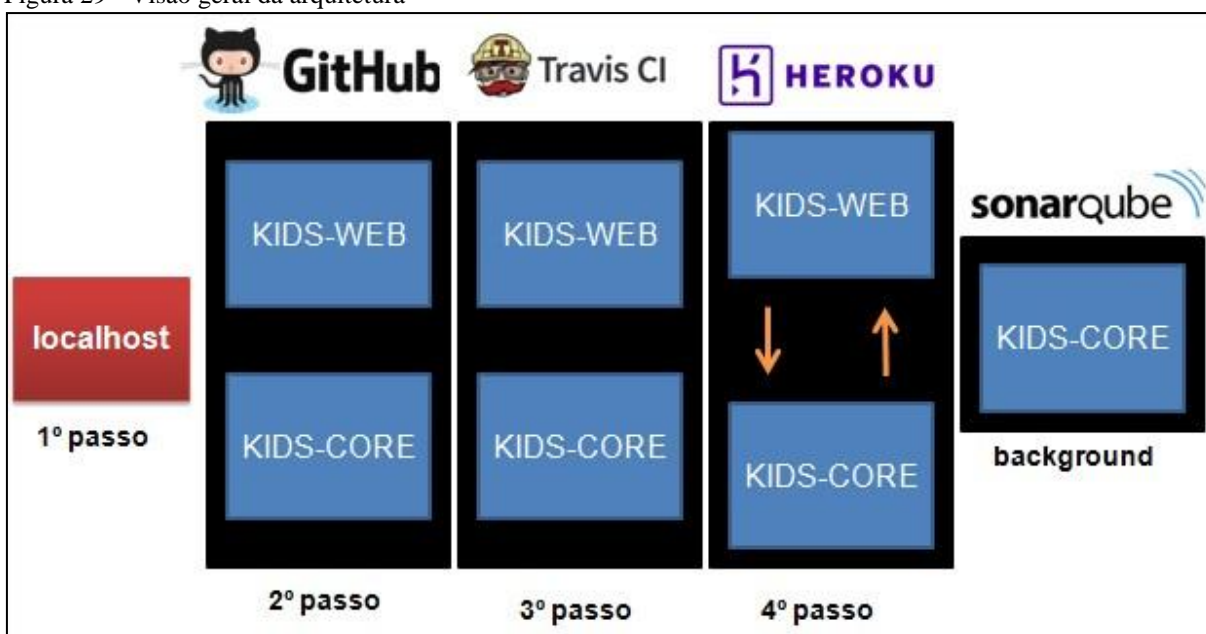
No dia a dia do desenvolvimento, o projeto seguiu o seguinte processo:

A cada commit que era realizado no repositório Github, automaticamente era feito uma chamada ao Travis, para iniciar o processo de integração continua. A cada chamada ao Travis, era realizado toda a construção dos artefatos necessários para que a aplicação rodasse com sucesso, além disso eram executados todos os testes unitários desenvolvidos, e ao final do build, realizava uma chamada ao Sonar para realizar a avaliação do código fonte.

Após o Travis executar todos os passos, caso não ocorresse nenhuma falha, então era realizado o deploy no servidor Heroku, para que fosse disponibilizado uma nova versão da aplicação disponível para o usuário final.

Segue abaixo um fluxograma exibindo o processo da arquitetura do sistema:

Figura 29 - Visão geral da arquitetura



Fonte: Criado pelo autor do projeto

## 7.2 MODELAGEM FUNCIONAL

Para a realização deste projeto, foi utilizado como modelagem funcional o Product Backlog, através dele foram criadas diversas User Stories, logo após as User Stories foram divididas inicialmente em dez (10) Sprints, porém no decorrer do desenvolvimento, foi possível adiantar algumas Users Stories que estavam planejadas para outros Sprints, com isso, totalizando no final do projeto o número de nove (8) Sprints realizadas.

### 7.2.1 Product backlog

No Product Backlog, foi definido juntamente com o Product Owner quais funcionalidades tiveram prioridades nas etapas de desenvolvimento.

Segue abaixo uma tabela exibindo o Product Backlog gerado. Nessa tabela contém um código (Exemplo: PB-00, que serve apenas como um localizador de Product Backlog na gestão do projeto), a descrição, e o nível de prioridade.

Tabela 1 - Product Backlog

CÓDIGO	DESCRIÇÃO	PRIORIDADE
PB - 01	Login com conta google	1
PB - 02	Controle de usuários	2
PB - 03	Módulo de crianças	3
PB - 04	Controle de medicamentos e alergias das crianças	4
PB - 05	Vínculo de familiares com as crianças cadastradas na creche	5
PB - 06	Módulo de diário escolar	6
PB - 07	Módulo de eventos	7
PB - 08	Módulo de avisos	8
PB - 09	Módulo de contato entre familiar x creche	9
PB - 10	Módulo de avaliação escolar	10
PB - 11	Módulo de cardápio	11
PB - 12	Módulo de galeria de fotos	12

Fonte: Criado pelo autor do projeto.

### 7.2.2 User stories

Através das Users Stories, foram detalhadas as necessidades dos usuários. Segue abaixo diversas tabelas exibindo todas as Users Stories desenvolvidas nesse projeto. Nessa tabela contém uma descrição e um código (Exemplo: US-00, que serve apenas como um localizador na gestão do projeto);

Tabela 2 - User Stories do Product Backlog - 01

<b>US-01 - Realizar login com conta google</b>
<p>Como um usuário interessado em utilizar o portal</p> <p>Eu quero realizar meu login usando minha conta google</p> <p>Para que seja possível me cadastrar no sistema</p>

Fonte: Criado pelo autor do projeto.



Tabela 3 - User Stories do Product Backlog - 02

**US-02 - Escolha do tipo de usuário**

**Como um usuário logado no portal pela primeira vez**  
**Eu quero escolher meu tipo de conta para se cadastrar (CRECHE/FAMILIAR)**  
**Para que eu possa obter minhas informações personalizadas**

**US-03 - Cadastro de perfil**

**Como um usuário do tipo creche ou familiar**  
**Eu quero cadastrar minhas informações de perfil**  
**Para que sejam mantidos meus dados atualizados no portal**

Fonte: Criado pelo autor do projeto.

Tabela 4- User Stories do Product Backlog - 03

**US-04 - Wizard para cadastro de crianças;**

**Como um usuário do tipo creche**  
**Eu quero pesquisar, cadastrar e editar crianças na minha instituição**  
**Para que eu possa manter e dirigir o controle sobre os seus dados pessoais**

**US-05 - Pesquisa e edição de dados gerais das crianças**

**Como um usuário do tipo familiar**  
**Eu quero pesquisar e editar crianças**  
**Para que eu possa manter e dirigir o controle sobre os seus dados pessoais**

Fonte: Criado pelo autor do projeto.

Tabela 5- User Stories do Product Backlog - 04

**US-06 - Cadastro e edição de medicamentos**

**Como um usuário do tipo creche ou familiar**  
**Eu quero cadastrar / editar medicamentos para as crianças**  
**Para que lembrar do seu uso diário**

**US-07 - Cadastro de edição de alergias**

**Como um usuário do tipo creche ou familiar**  
**Eu quero cadastrar / editar alergias para as crianças**  
**Para que lembrar de tomar todo o cuidado possível, preservando a criança dos riscos**

Fonte: Criado pelo autor do projeto.

Tabela 6- User Stories do Product Backlog - 05

**US-08 - Vinculo de crianças com seus familiares**

**Como um usuário do tipo creche**  
**Eu quero vincular usuários familiares com as crianças matriculadas na creche**  
**Para que seja possível compartilhar informações da creche há seus familiares**

Fonte: Criado pelo autor do projeto.

Tabela 7- User Stories do Product Backlog - 06

**US-09 - Cadastro e edição de diário escolar**

**Como um usuário do tipo creche**  
**Eu quero cadastrar/editar o diário escolar de cada criança matriculada**  
**Para que seus familiares possam acompanhar como foi o seu dia**

**US-10 - Visualização de diário escolar**

**Como um usuário do tipo familiar**  
**Eu quero visualizar o diário escolar das crianças que possuo vínculo**  
**Para que possa acompanhar suas atividades diariamente**

Fonte: Criado pelo autor do projeto.

Tabela 8- User Stories do Product Backlog - 07

**US-11 - Cadastro e atualização de eventos**

**Como um usuário do tipo creche**  
**Eu quero cadastrar e atualizar eventos**  
**Para que notificar os familiares para que se programem a vir na data informada**

**US-12 - Aceite e Rejeição de eventos**

**Como um usuário do tipo familiar**  
**Eu quero aceitar ou rejeitar a presença em eventos**  
**Para que a creche possa se organizar a respeito do número de participantes confirmados**

Fonte: Criado pelo autor do projeto.

Tabela 9- User Stories do Product Backlog - 08

**US-13 - Cadastro de avisos**

**Como um usuário do tipo creche**  
**Eu quero cadastrar avisos**  
**Para que aos familiares sejam notificados de algo importante**

**US-14 - Visualização de avisos**

**Como um usuário do tipo familiar**  
**Eu quero visualizar os avisos deixados pela creche**  
**Para que eu possa tomar alguma ação quando necessário**

Fonte: Criado pelo autor do projeto.

Tabela 10- User Stories do Product Backlog - 09

**US-15 - Reclamar, elogiar, sugerir**

**Como um usuário do tipo familiar**  
**Eu quero poder reclamar, elogiar ou sugerir algo para a creche**  
**Para que eu possa ajudar no desenvolvimento da instituição**

**US-16 - Feedback da reclamação, elogio ou sugestão**

**Como um usuário do tipo creche**  
**Eu quero poder dar um feedback para o familiar que iniciou uma interação**  
**Para que eu possa manter um atendimento com qualidade**

Fonte: Criado pelo autor do projeto.

Tabela 11- User Stories do Product Backlog - 10

**US-17 - Cadastrar / editar avaliação**

**Como um usuário do tipo creche**  
**Eu quero poder avaliar cada criança matriculada**  
**Para que eu possa disponibilizar seu desenvolvimento para seus familiares**

**US-18 - Visualizar avaliação**

**Como um usuário do tipo familiar**  
**Eu quero poder visualizar as avaliações das crianças que possuo vinculo**  
**Para que eu possa acompanhar seu desenvolvimento escolar**

Fonte: Criado pelo autor do projeto.

Tabela 12- User Stories do Product Backlog - 11

**US-19 - Cadastrar cardápio**

**Como um usuário do tipo creche**  
**Eu quero poder cadastrar as refeições servidas**  
**Para que eu possa disponibilizar seu desenvolvimento para seus familiares**

**US-20 - Visualizar cardápio**

**Como um usuário do familiar**  
**Eu quero poder visualizar as refeições servidas pela creche**  
**Para que eu possa acompanhar a dieta das crianças**

Fonte: Criado pelo autor do projeto.

Tabela 13- User Stories do Product Backlog - 12

**US-21 - Cadastrar galeria de fotos**

**Como um usuário do tipo creche**  
**Eu quero poder cadastrar um galeria de fotos**  
**Para que eu possa disponibilizar as imagens do dia a dia para seus familiares**

**US-22 - Visualizar galeria de fotos**

**Como um usuário do tipo familiar**  
**Eu quero poder visualizar uma galeria de fotos**  
**Para que eu possa acompanhar o dia a dia das crianças**

Fonte: Criado pelo autor do projeto.

### 7.2.3 Sprints

Nessa sessão serão descritas todas as atividades que envolveram os sprints, atividades tais como: Os planejamentos que foram feitos para cada sprint, as entregas que foram realizadas, as melhorias sugeridas pelo P.O, as não conformidades encontradas, os resultados do Sprint Review e os dados coletados nas retrospectivas.

#### 7.2.3.1 Sprint 01

Para apresentar o planejamento do Sprint 01, segue abaixo uma tabela explicativa, exibindo informações do período do Sprint, User Stories, além das tarefas criadas para atender a demanda:

Tabela 14 - Sprint 01 - Planejamento

SPRINT 01 - Planejamento	
Data inicial: 16/06/2017	Data final: 30/06/2017
USER STORIES	TAREFAS
US-01 - Realizar login com conta google;	- Criar tela de login; - Criar base de dados; - Criar API e serviço de usuário;
US-02 - Escolha do tipo de usuário;	- Criar tela de escolha do tipo de usuário; - Criar serviço para tipo de usuário;
US-03 - Cadastro de perfil;	- Criar tela de perfil; - Criar base de dados; - Criar serviço para perfil;

Fonte: Criado pelo autor do projeto.

Para apresentar os resultados gerados do Sprint 01, segue abaixo uma tabela explicativa, exibindo informações dos dados coletados no Sprint Review juntamente com o P.O, exibindo dados apresentados na retrospectiva juntamente com o professor orientador, e observações importantes:

Tabela 15 - Sprint 01 - Resultado

SPRINT 01 - Resultado
SPRINT REVIEW
Entregas Realizadas
- US-01 - Realizar login com conta google; - US-02 - Escolha do tipo de usuário; - US-03 - Cadastro de perfil;
Melhorias Solicitadas
[MELHORIA] - Adicionar telefone no 'perfil reduzido'; [MELHORIA] - Destacar o 'tipo de usuário' logado no portal';

RETROSPECTIVA	
Pontos Positivos	Pontos Negativos
<ul style="list-style-type: none"> <li>- Entrega em dia dos requisitos funcionais;</li> <li>-Desenvolvimento do front-end com a última versão do Angular;</li> <li>- Aplicação hospedada no Heroku;</li> <li>- Adição do Sonar para analisar a qualidade do código fonte;</li> <li>- Criação de um repositório somente para artefatos de documentação;</li> </ul>	<ul style="list-style-type: none"> <li>-Sem internet para o desenvolvimento durante uns 3/4 dias;</li> <li>-Remodelagem da estrutura do front-end, durante umas 5/6 vezes;</li> <li>-Bastante horas utilizadas p\ subir o Angular 4 no Heroku (até achar o cmd correto);</li> <li>-Sonar ainda não configurado por completo;</li> <li>Pouco teste unitário;</li> </ul>
OBSERVAÇÕES	
<p>No término do Sprint 01, ficou um saldo acumulado de 02 melhorias a serem entregues para o cliente, são elas:</p> <ul style="list-style-type: none"> <li>• Adicionar telefone no 'perfil reduzido';</li> <li>• Destacar o 'tipo de usuário' logado no portal';</li> </ul> <p>Foram acordadas com o P.O, que as melhorias citadas acima serão postergadas para o próximo Sprint, porém sem prioridades alguma sobre as User Stories.</p>	

Fonte: Criado pelo autor do projeto.

### 7.2.3.2 Sprint 02

Para apresentar o planejamento do Sprint 02, segue abaixo uma tabela explicativa, exibindo informações do período do Sprint, User Stories, além das tarefas criadas para atender a demanda:

Tabela 16 - Sprint 02 - Planejamento

SPRINT 02 - Planejamento	
Data inicial: 16/07/2017	Data final: 30/07/2017
USER STORIES	TAREFAS
US-04 - Wizard para cadastro de crianças;  US-05 - Pesquisa e edição de dados gerais das crianças;  US-06 - Cadastro e edição de medicamentos;  US-07 - Cadastro de edição de alergias;	<ul style="list-style-type: none"> <li>- Criar telas do wizard de criança;</li> <li>- Criar base de dados;</li> <li>- Criar API e serviço de criança;</li> </ul>

MELHORIAS
[MELHORIA] - Adicionar telefone no 'perfil reduzido'; [MELHORIA] - Destacar o 'tipo de usuário' logado no portal';

Fonte: Criado pelo autor do projeto.

Para apresentar os resultados gerados do Sprint 02, segue abaixo uma tabela explicativa, exibindo informações dos dados coletados no Sprint Review juntamente com o P.O, exibindo dados apresentados na retrospectiva juntamente com o professor orientador, e observações importantes:

Tabela 17 - Sprint 02 - Resultado

SPRINT 02 - Resultado	
SPRINT REVIEW	
Entregas Realizadas	
<ul style="list-style-type: none"> <li>- US-04 - Wizard para cadastro de crianças;</li> <li>- US-05 - Pesquisa e edição de dados gerais das crianças;</li> <li>- US-06 - Cadastro e edição de medicamentos;</li> <li>- US-07 - Cadastro e edição de alergias;</li> </ul> [MELHORIA] - Adicionar telefone no 'perfil reduzido'; [MELHORIA] - Destacar o 'tipo de usuário' logado no portal';	
Melhorias Solicitadas	
[MELHORIA] - Ajustar textos em todas as telas onde não está responsivo;	
RETROSPECTIVA	
Pontos Positivos	Pontos Negativos
<ul style="list-style-type: none"> <li>- Aumento na cobertura de testes unitários;</li> <li>- Refatoração de código fonte / lado servidor;</li> <li>- Criação de um repositório somente para artefatos de DB, para agilizar os testes;</li> </ul>	<ul style="list-style-type: none"> <li>- Sem internet para o desenvolvimento durante uns 4 dias;</li> <li>- Os dois sistemas distribuídos não conseguem mais se comunicar no servidor Heroku;</li> </ul>
OBSERVAÇÕES	
<p>No término do Sprint 02, ficou um saldo acumulado de 01 melhoria a ser entregue para o cliente:</p> <ul style="list-style-type: none"> <li>• Ajustar textos em todas as telas onde não está responsivo;</li> </ul> <p>Foram acordadas com o P.O, que a melhoria citada acima será postergada para o próximo Sprint, porém sem prioridade alguma sobre as User Stories.</p>	

Fonte: Criado pelo autor do projeto.

### 7.2.3.3 Sprint 03

Para apresentar o planejamento do Sprint 03, segue abaixo uma tabela explicativa, exibindo informações do período do Sprint, User Stories, além das tarefas criadas para atender a demanda:

Tabela 18 - Sprint 03 - Planejamento

SPRINT 03 - Planejamento	
Data inicial: 01/08/2017	Data final: 15/08/2017
USER STORIES	TAREFAS
- US-08 - Vinculo de crianças com seus familiares;	- Criar telas de vinculo dos familiares; - Criar base de dados; - Criar API e serviço de família;
MELHORIAS	
[MELHORIA] - Ajustar textos em todas as telas onde não está responsivo;	

Fonte: Criado pelo autor do projeto.

Para apresentar os resultados gerados do Sprint 03, segue abaixo uma tabela explicativa, exibindo informações dos dados coletados no Sprint Review juntamente com o P.O, exibindo dados apresentados na retrospectiva juntamente com o professor orientador, e observações importantes:

Tabela 19 - Sprint 03 - Resultado

SPRINT 03 - Resultado	
SPRINT REVIEW	
Entregas Realizadas	
- US-08 - Vinculo de crianças com seus familiares;	
Melhorias Solicitadas	
- Não foram solicitadas novas melhorias;	
RETROSPECTIVA	
Pontos Positivos	Pontos Negativos
- Entrega de melhorias solicitada anteriormente pelo cliente;	- Os dois sistemas distribuídos ainda permanecem sem se comunicar no servidor Heroku.

Fonte: Criado pelo autor do projeto.

### 7.2.3.4 Sprint 04

Para apresentar o planejamento do Sprint 04, segue abaixo uma tabela explicativa, exibindo informações do período do Sprint, User Stories, além das tarefas criadas para atender a demanda:

Tabela 20 - Sprint 04 - Planejamento

<b>SPRINT 04 - Planejamento</b>	
<b>Data inicial: 16/08/2017</b>	<b>Data final: 30/08/2017</b>
<b>USER STORIES</b>	<b>TAREFAS</b>
<ul style="list-style-type: none"> <li>- US-09 - Cadastro e edição de diário escolar;</li> <li>- US-10 - Visualização de diário escolar;</li> </ul>	<ul style="list-style-type: none"> <li>- Criar telas do diários escolar;</li> <li>- Criar base de dados;</li> <li>- Criar API e serviço de diário;</li> </ul>

Fonte: Criado pelo autor do projeto.

Para apresentar os resultados gerados do Sprint 04, segue abaixo uma tabela explicativa, exibindo informações dos dados coletados no Sprint Review juntamente com o P.O, exibindo dados apresentados na retrospectiva juntamente com o professor orientador, e observações importantes:

Tabela 21 - Sprint 04 - Resultado

<b>SPRINT 04 - Resultado</b>	
<b>SPRINT REVIEW</b>	
<b>Entregas Realizadas</b>	
<ul style="list-style-type: none"> <li>- US-09 - Cadastro e edição de diário escolar;</li> <li>- US-10 - Visualização de diário escolar;</li> </ul>	
<b>Melhorias Solicitadas</b>	
<ul style="list-style-type: none"> <li>- Não foram solicitadas novas melhorias;</li> </ul>	
<b>RETROSPECTIVA</b>	
<b>Pontos Positivos</b>	<b>Pontos Negativos</b>
<ul style="list-style-type: none"> <li>- Entrega realizada conforme o planejado;</li> </ul>	<ul style="list-style-type: none"> <li>- Os dois sistemas distribuídos ainda permanecem sem se comunicar no servidor Heroku.</li> </ul>

Fonte: Criado pelo autor do projeto.



### 7.2.3.5 Sprint 05

Para apresentar os resultados gerados do Sprint 05, segue abaixo uma tabela explicativa, exibindo informações do período do Sprint, dados coletados no Sprint Review, dados coletados na retrospectiva e observações importantes:

Tabela 22 - Sprint 05 - Planejamento

SPRINT 05 - Planejamento	
Data inicial: 01/09/2017	Data final: 15/09/2017
USER STORIES	TAREFAS
<ul style="list-style-type: none"> <li>- US-11 - Cadastro e atualização de eventos;</li> <li>- US-12 - Aceite e Rejeição de eventos;</li> </ul>	<ul style="list-style-type: none"> <li>- Criar telas de eventos escolar;</li> <li>- Criar base de dados;</li> <li>- Criar API e serviço de evento;</li> </ul>

Fonte: Criado pelo autor do projeto.

Para apresentar os resultados gerados do Sprint 05, segue abaixo uma tabela explicativa, exibindo informações dos dados coletados no Sprint Review juntamente com o P.O, exibindo dados apresentados na retrospectiva juntamente com o professor orientador, e observações importantes:

Tabela 23 - Sprint 05 - Resultado

SPRINT 05 - Resultado
SPRINT REVIEW
Entregas Realizadas
<ul style="list-style-type: none"> <li>- US-11 - Cadastro e atualização de eventos;</li> <li>- US-12 - Aceite e Rejeição de eventos;</li> <li>- US-13 - Cadastro de avisos;</li> <li>- US-14 - Visualização de avisos;</li> </ul>
Melhorias Solicitadas
<p>[MELHORIA] - Enviar e-mail de notificação ao vincular usuário;</p> <p>[MELHORIA] - Trazer a lista de crianças ordenada pelo alfabeto;</p> <p>[MELHORIA] - Exibir mensagem padrão quando não há registros cadastrados;</p> <p>[MELHORIA] - Exibir menus somente para familiares ou creches ativos no sistema;</p> <p>[MELHORIA] - Exibir mensagem para família quando não existirem vínculos;</p> <p>[MELHORIA] - Barra de progresso. Adicionar nas telas que realizam buscas;</p> <p>[MELHORIA] -No diário escolar, deverá exibir mensagem padrão ao acessar, solicitando para selecionar algum tipo na tela;</p>
Não conformidades encontradas
<p>[ERRO]-Vincular familiares - Está exibindo mensagem de sucesso, mesmo quando há erros;</p>

[ERRO]-Crianças - O usuário familiar está atualizando a matrícula da criança na creche; [ERRO]-Eventos - Está exibindo as mensagens de erros duplicadas; [ERRO]-Creche nova - Está exibindo mensagem de crianças não vinculadas ao entrar na aplicação; [ERRO]-Avisos - Está permitindo cadastrar avisos com data menor que a data atual para expirar;	
RETROSPECTIVA	
Pontos Positivos	Pontos Negativos
<ul style="list-style-type: none"> <li>- Foram adiantadas escopos planejados para outros Sprints;</li> <li>- Com internet 100% durante todo Sprint;</li> <li>- Os sistemas distribuídos dentro do Heroku, voltaram a funcionar;</li> </ul>	<ul style="list-style-type: none"> <li>- Redução na cobertura de testes unitários;</li> <li>- Aparecimento de diversos bugs;</li> </ul>
OBSERVAÇÕES	
<p>No decorrer deste Sprint foi possível adiantar uma parte do escopo. Sendo assim, por determinação do P.O do projeto, foram adiantadas as seguintes User Stories:  US-13 - Cadastro de avisos e US-14 - Visualização de avisos.</p> <p>No término do Sprint 05, ficou um saldo acumulado de 07 melhorias e mais 05 correções de bugs, a serem entregues para o cliente:</p> <ul style="list-style-type: none"> <li>• [MELHORIA] - Enviar e-mail de notificação ao vincular usuário;</li> <li>• [MELHORIA] - Trazer a lista de crianças ordenada pelo alfabeto;</li> <li>• [MELHORIA] - Exibir mensagem padrão quando não há registros cadastrados;</li> <li>• [MELHORIA] - Exibir menus somente para familiares ou creches ativos no sistema;</li> <li>• [MELHORIA] - Exibir mensagem para família quando não existirem vínculos;</li> <li>• [MELHORIA] - Barra de progresso. Adicionar nas telas que realizam buscas;</li> <li>• [MELHORIA] -No diário escolar, deverá exibir mensagem padrão ao acessar, solicitando para selecionar algum tipo na tela;</li> <li>• [ERRO]-Vincular familiares - Está exibindo mensagem de sucesso, mesmo quando há erros;</li> <li>• [ERRO]-Crianças - O usuário familiar está atualizando a matrícula da criança na creche;</li> <li>• [ERRO]-Eventos - Está exibindo as mensagens de erros duplicadas;</li> <li>• [ERRO]-Creche nova - Está exibindo mensagem de crianças não vinculadas ao entrar na aplicação;</li> <li>• [ERRO]-Avisos - Está permitindo cadastrar avisos com data menor que a data atual para expirar;</li> </ul> <p>Foram acordadas com o P.O, que as melhorias citadas acima serão postergadas para o próximo Sprint, porém sem prioridade alguma sobre as User Stories. Exceto os bugs, que deverão ser planejados e executados no próximo Sprint.</p>	

Fonte: Criado pelo autor do projeto.

#### 7.2.3.6 Sprint 06

Para apresentar os resultados gerados do Sprint 06, segue abaixo uma tabela explicativa, exibindo informações do período do Sprint, dados coletados no Sprint Review, dados coletados na retrospectiva e observações importantes:

Tabela 24 - Sprint 06 - Planejamento

SPRINT 06 - Planejamento	
Data inicial: 16/09/2017	Data final: 30/09/2017
USER STORIES	TAREFAS
<del>-US-13 - Cadastro de avisos; [ANTECIPADO]</del> <del>-US-14 - Visualização de avisos; [ANTECIPADO]</del> -US-15 - Reclamar, elogiar, sugerir; -US-16 - Feedback da reclamação, elogio ou sugestão;	<del>-Criar telas de avisos; [ANTECIPADO]</del> <del>-Criar base de dados para avisos; [ANTECIPADO]</del> <del>-Criar API e serviço de avisos; [ANTECIPADO]</del> - Criar telas de comunicação; - Criar base de dados para comunicação; - Criar API e serviço de comunicação;
MELHORIAS E CORREÇÕES DE BUGS	
[MELHORIA] - Enviar e-mail de notificação ao vincular usuário; [MELHORIA] - Trazer a lista de crianças ordenada pelo alfabeto; [MELHORIA] - Exibir mensagem padrão quando não há registros cadastrados; [MELHORIA] - Exibir menus somente para familiares ou creches ativos no sistema; [MELHORIA] - Exibir mensagem para família quando não existirem vínculos; [MELHORIA] - Barra de progresso. Adicionar nas telas que realizam buscas; [MELHORIA] -No diário escolar, deverá exibir mensagem padrão ao acessar, solicitando para selecionar algum tipo na tela; [ERRO]-Vincular familiares - Está exibindo mensagem de sucesso, mesmo quando há erros; [ERRO]-Crianças - O usuário familiar está atualizando a matrícula da criança na creche; [ERRO]-Eventos - Está exibindo as mensagens de erros duplicadas; [ERRO]-Creche nova - Está exibindo mensagem de crianças não vinculadas ao entrar na aplicação; [ERRO]-Avisos - Está permitindo cadastrar avisos com data menor que a data atual para expirar;	

Fonte: Criado pelo autor do projeto.

Para apresentar os resultados gerados do Sprint 06, segue abaixo uma tabela explicativa, exibindo informações dos dados coletados no Sprint Review juntamente com o P.O, exibindo dados apresentados na retrospectiva juntamente com o professor orientador, e observações importantes:

Tabela 25 - Sprint 06 - Resultado

SPRINT 06 - Resultado
SPRINT REVIEW
Entregas Realizadas
-US-15 - Reclamar, elogiar, sugerir; -US-16 - Feedback da reclamação, elogio ou sugestão; [MELHORIA] - Enviar e-mail de notificação ao vincular usuário; [MELHORIA] - Trazer a lista de crianças ordenada pelo alfabeto; [MELHORIA] - Exibir mensagem padrão quando não há registros cadastrados; [MELHORIA] - Exibir menus somente para familiares ou creches ativos no sistema;

[MELHORIA] - Exibir mensagem para família quando não existirem vínculos; [MELHORIA] - Barra de progresso. Adicionar nas telas que realizam buscas; [MELHORIA] -No diário escolar, deverá exibir mensagem padrão ao acessar, solicitando para selecionar algum tipo na tela; [ERRO CORRIGIDO]-Vincular familiares - Está exibindo mensagem de sucesso, mesmo quando há erros; [ERRO CORRIGIDO]-Crianças - O usuário familiar está atualizando a matrícula da criança na creche; [ERRO CORRIGIDO]-Eventos - Está exibindo as mensagens de erros duplicadas; [ERRO CORRIGIDO]-Creche nova - Está exibindo mensagem de crianças não vinculadas ao entrar na aplicação; [ERRO CORRIGIDO]-Avisos - Está permitindo cadastrar avisos com data menor que a data atual para expirar;	
<b>Melhorias Solicitadas</b>	
- Não foram solicitadas novas melhorias;	
<b>RETROSPECTIVA</b>	
<b>Pontos Positivos</b>	<b>Pontos Negativos</b>
<ul style="list-style-type: none"> <li>- Correção de todas não conformidades encontradas no Sprint anterior;</li> <li>- Entrega de 07 melhorias sugeridas pelo P.O em Sprints anteriores;</li> <li>- Com internet 100% durante todo Sprint;</li> </ul>	<ul style="list-style-type: none"> <li>- Redução na cobertura de testes unitários;</li> </ul>
<b>OBSERVAÇÕES</b>	
No término do Sprint 06, foi constatado a entrega de todas as melhorias sugeridas durante o projeto, assim como a correção de todos os bugs encontrados.	

Fonte: Criado pelo autor do projeto.

### 7.2.3.7 Sprint 07

Para apresentar os resultados gerados do Sprint 07, segue abaixo uma tabela explicativa, exibindo informações do período do Sprint, dados coletados no Sprint Review, dados coletados na retrospectiva e observações importantes:

Tabela 26 - Sprint 07 - Planejamento

<b>SPRINT 07 - Planejamento</b>	
<b>Data inicial: 01/10/2017</b>	<b>Data final: 15/10/2017</b>
<b>USER STORIES</b>	<b>TAREFAS</b>

<ul style="list-style-type: none"> <li>- US-17 - Cadastrar / editar avaliação;</li> <li>- US-18 - Visualizar avaliação;</li> </ul>	<ul style="list-style-type: none"> <li>- Criar telas de avaliação;</li> <li>- Criar base de dados para avaliação;</li> <li>- Criar API e serviço de avaliação;</li> </ul>
--	---

Fonte: Criado pelo autor do projeto.

Para apresentar os resultados gerados do Sprint 07, segue abaixo uma tabela explicativa, exibindo informações dos dados coletados no Sprint Review juntamente com o P.O, exibindo dados apresentados na retrospectiva juntamente com o professor orientador, e observações importantes:

Tabela 27 - Sprint 07 - Resultado

SPRINT 07 - Resultado	
SPRINT REVIEW	
Entregas Realizadas	
<ul style="list-style-type: none"> <li>- US-17 - Cadastrar / editar avaliação;</li> <li>- US-18 - Visualizar avaliação;</li> <li>- US-19 - Cadastrar cardápio;</li> <li>- US-20 - Visualizar cardápio;</li> </ul>	
Melhorias Solicitadas	
<ul style="list-style-type: none"> <li>- Não foram solicitadas novas melhorias;</li> </ul>	
RETROSPECTIVA	
Pontos Positivos	Pontos Negativos
Antecipação de User Stories de Sprints posteriores;	- Com internet somente por 4 dias;
OBSERVAÇÕES	
<p>No decorrer deste Sprint foi possível adiantar uma parte do escopo. Sendo assim, por determinação do P.O do projeto, foram adiantadas as seguintes User Stories:</p> <p>US-19 - Cadastrar cardápio e US-20 - Visualizar cardápio.</p>	

Fonte: Criado pelo autor do projeto.

#### 7.2.3.8 Sprint 08

Para apresentar os resultados gerados do Sprint 08, segue abaixo uma tabela explicativa, exibindo informações do período do Sprint, dados coletados no Sprint Review, dados coletados na retrospectiva e observações importantes:

Tabela 28 - Sprint 08 - Planejamento

SPRINT 08 - Planejamento	
Data inicial: 16/10/2017	Data final: 30/10/2017
USER STORIES	TAREFAS
<del>- US-19 - Cadastrar cardápio; [ANTECIPADO]</del> <del>- US-20 - Visualizar cardápio; [ANTECIPADO]</del> - US-21 - Cadastrar galeria de fotos - US-22 - Visualizar galeria de fotos	<del>- Criar telas de cardápio; [ANTECIPADO]</del> <del>- Criar banco para cardápio; [ANTECIPADO]</del> <del>- Criar API e serviço de cardápio; [ANTECIPADO]</del> - Criar telas de galeria; - Criar base de dados para galeria; - Criar API e serviço de galeria;

Fonte: Criado pelo autor do projeto.

Tabela 29 - Sprint 08 - Resultado

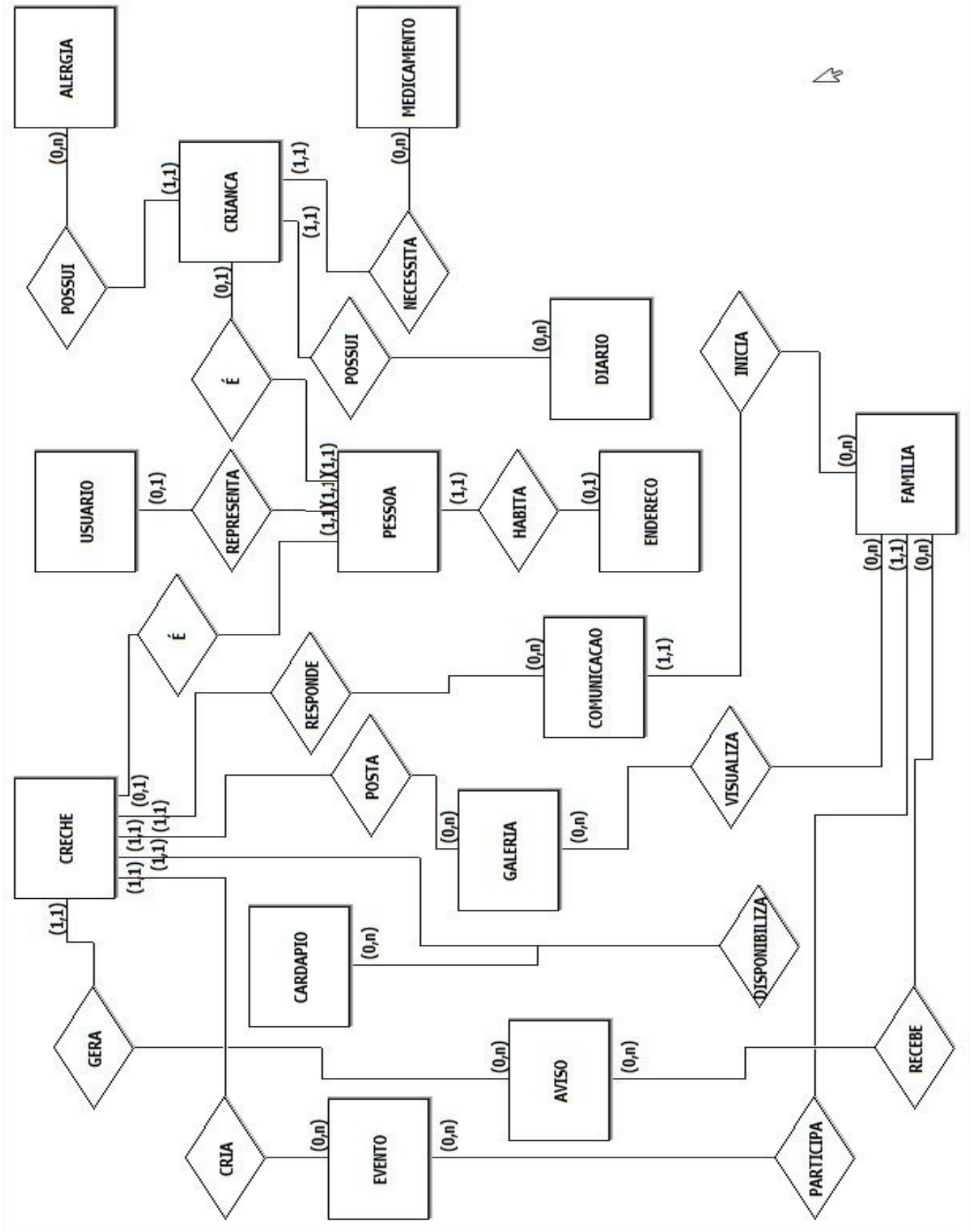
SPRINT 08 - Resultado	
SPRINT REVIEW	
Entregas Realizadas	
- US-21 - Cadastrar galeria de fotos - US-22 - Visualizar galeria de fotos	
Melhorias Solicitadas	
- Não foram solicitadas novas melhorias;	
RETROSPECTIVA	
Pontos Positivos	Pontos Negativos
- Finalização de todo o escopo do projeto, terminado antes do prazo inicial;	- Com internet 100% do tempo;

Fonte: Criado pelo autor do projeto.

### 7.3 MODELAGEM DE DADOS

Para exibir a modelagem de dados utilizada nessa projeto, segue abaixo uma figura ilustrativa representando o modelo de Entidade e Relacionamento do projeto:

Figura 30 - Modelo de Entidade e Relacionamento



## 8 FUNCIONAMENTO DO SISTEMA

O funcionamento do sistema está relacionado a dois papéis diferentes dentro da aplicação. Por um lado, existe um usuário que representa o papel de uma creche, por outro lado existe o papel de um familiar, que é qualquer pessoa que possua visão para acompanhar as atividades de uma ou mais crianças cadastradas em alguma creche da aplicação.

Quando algum usuário entra pela primeira na vez na aplicação, a primeira coisa a se fazer é definir o tipo de conta que esse usuário representará na aplicação, ou do tipo *familiar* ou irá representar uma *creche*.

Após a criação dessa conta ser realizada com sucesso, ficará à disposição as funções visíveis para cada tipo de usuário.

Segue abaixo uma tabela que identifica as ações que poderão ser acessadas por diferentes papéis logados na aplicação:

Tabela 30 - Permissões por papel de usuário

Permissões por papel de usuário			
Categoria	Ação	Creche	Familiar
Perfil	Atualizar dados do perfil	Sim	Sim
Crianças	Pesquisar Crianças	Sim	Sim
Crianças	Cadastrar novas crianças	Sim	Não
Crianças	Editar cadastro de crianças	Sim	Sim
Medicamentos	Cadastrar/editar/remover medicamento para crianças	Sim	Sim
Alergias	Cadastrar/editar/remover alergias para crianças	Sim	Sim
Familiares	Vincular crianças para usuários com papel de familiar	Sim	Não
Familiares	Remover vínculos de crianças com familiares	Sim	Não
Diários	Cadastrar e editar as atividades diárias	Sim	Não
Diários	Visualizar as atividades diárias	Sim	Sim
Eventos	Cadastrar, editar, visualizar eventos	Sim	Não
Eventos	Confirmar, Rejeitar presença	Não	Sim
Avisos	Cadastrar e remover	Sim	Não
Avisos	Visualizar	Sim	Sim
Interação	Iniciar uma interação	Não	Sim
Interação	Responder uma interação	Sim	Não
Interação	Visualizar Interações	Sim	Sim
Avaliações	Avaliar um ou mais alunos	Sim	Não
Avaliações	Visualizar avaliações	Sim	Sim

Fonte: Criado pelo autor do projeto.



Tabela 31- Permissões por papel de usuário (continuação)

Permissões por papel de usuário			
Categoria	Ação	Creche	Familiar
Cardápio	Cadastrar e remover alimentação	Sim	Não
Cardápio	Visualizar alimentação	Sim	Sim
Galeria	Adicionar e remover fotos	Sim	Não
Galeria	Visualizar fotos	Sim	Sim

## 8.1 DADOS PARA VERIFICAR O FUNCIONAMENTO DO SISTEMA

Afim de agilizar o acesso as diversas funcionalidades do sistema, o mesmo está disponível no servidor Heroku, através do endereço <https://kids-web.herokuapp.com/kids/login>. Além disso já existem dados pré prontos para preparar uma base de testes. Sendo assim, é possível ter acesso a esses dados seguindo os seguintes passos:

**1º Passo:** Acesse o endereço: <https://kids-core.herokuapp.com/db/12345>

Geralmente demora uns 5 segundos. No máximo um pouco mais de 1 minuto (acontece quando o servidor Heroku está hibernando).

Quando essa tarefa for concluída, ira ser exibido no navegador a mensagem: " [KIDS INFORMA] - Base de testes preparada com sucesso!."

**2º Passo:** Acesse o endereço: <https://kids-web.herokuapp.com/kids/login>, escolha um dos dois papéis abaixo para acessar o sistema:

- **Creche:**  
**Login:** kidsusuariocreche@gmail.com  
**Senha:** !@#kidsusuariocreche
- **Familiar:**  
**Login:** kidsusuariofamiliar@gmail.com  
**Senha:** !@#kidsusuariofamiliar

## 9 VALIDAÇÃO

Esse sistema foi validado em diversas etapas, umas das etapas aconteceram ao final de cada Sprint, ou seja, no Sprint Review, que foram realizados juntamente com o P.O do projeto. Nessa etapa foi possível obter a validação das entregas, assim como possibilitar a geração de melhorias para o projeto.

Além disso na parte técnica na qual se refere o código fonte, foram utilizados testes unitários, utilizando o Junit juntamente com o Mockito. Também no requisito de validação do código fonte, foi utilizado o Sonar, que é uma ferramenta online responsável por realizar validações sobre a qualidade do código.

E ao final de toda a implementação do projeto, foi aplicado mais uma validação, utilizando as Heurísticas de Nielsen<sup>18</sup>.

### 9.1 ESTRATÉGIA

Foi disponibilizado um ambiente online do projeto para que se pudesse avaliar se aquilo que foi entregue está de acordo com os objetivos desse projeto.

Para auxiliar nessa validação, foi disponibilizado uma pesquisa online, utilizando a ferramenta Google Docs, que permite a criação de um formulário de pesquisa, armazenando as respostas em uma planilha online.

Além disso, a pesquisa que foi aplicada, também visou seguir as seguintes Heurísticas de Nielsen:

- **Visibilidade do status do sistema:**

O sistema precisa manter os usuários informados sobre o que está acontecendo, fornecendo um feedback adequado dentro de um tempo razoável.

- **Compatibilidade do sistema com o mundo real:**

O sistema precisa falar a linguagem do usuário, com palavras, frase e conceitos familiares ao usuário, ao invés de termos orientados ao sistema. Seguir convenções do mundo real, fazendo com que a informação apareça numa ordem natural e lógica.

- **Controle do usuário e liberdade:**

Os usuários frequentemente escolhem, por engano, funções do sistema e precisam ter claras saídas de emergência para sair do estado indesejado sem ter que percorrer um extenso diálogo. Prover funções undo e redo.

- **Consistência e padrões:**

Usuários não precisam adivinhar que diferentes palavras, situações ou ações significam a mesma coisa.

---

<sup>18</sup> (Nielsen, 1993)

- **Prevenção de erros:**

Melhor que uma boa mensagem de erro é um projeto cuidadoso, o qual previne o erro antes dele acontecer.

- **Reconhecimento ao invés de lembrança:**

Tornar visíveis os objetos, as ações e as opções. O usuário não deve ter que lembrar informação de uma para outra parte do diálogo. Instruções para uso do sistema devem estar visíveis e facilmente recuperáveis quando necessário.

- **Flexibilidade e eficiência de uso:**

Usuários novatos se tornam peritos com o uso. Prover aceleradores de forma a aumentar a velocidade de interação. Permitir a usuários experientes “cortar caminho” em ações frequentes.

- **Estética e projeto minimalista:**

Diálogos não devem conter informação irrelevante ou raramente necessária. Qualquer unidade de informação extra no diálogo irá competir com unidades relevantes de informação e diminuir sua visibilidade relativa.

- **Ajudar os usuários a reconhecer, diagnosticar e corrigir erros:**

Mensagens de erro devem ser expressas em linguagem clara (sem códigos), indicando precisamente o problema e, construtivamente, sugerindo uma solução.

- **Help e documentação:**

Embora seja melhor um sistema que possa ser usada sem documentação, é necessário prover help e documentação. Essas informações devem ser fáceis de encontrar, focalizadas na tarefa do usuário e não muito extensas.

## 9.2 CONSOLIDAÇÃO DOS DADOS COLETADOS

**VOU COLOCAR AQUI AS RESPOSTAS DA PESQUISA...ATÉ O DIA 13/NOVEMBRO**

## REFERÊNCIAS

**ALMEIDA, Flávio.** MEAN: Full Stack Javascript para aplicações web com MongoDB, Express, Angular e Node. 1. ed. São Paulo. Casa do código, 2015.

Angular. **Deploy no Heroku.** Disponível em: <https://www.youtube.com/watch?v=MUdFLxveBzI>. Acessado em 06 agosto de 2017.

Angular. **Single Page Application.** Disponível em: <https://hipsters.tech/single-page-applications-hipsters-16/>. Acessado em 25 junho de 2017.

Angular. **Testing With AngularJS Part 1: Setting up Unit Testing With Karma.** Disponível em: <https://www.credera.com/blog/technology-insights/open-source-technology-insights/testing-angularjs-part-1-setting-unit-testing-karma/>. Acessado em 02 julho de 2017.

Angular e Heroku. **Host Your Angular 2 App on Heroku.** Disponível em: [https://medium.com/@ryanchenkie\\_40935/angular-cli-deployment-host-your-angular-2-app-on-heroku-3f266f13f352](https://medium.com/@ryanchenkie_40935/angular-cli-deployment-host-your-angular-2-app-on-heroku-3f266f13f352). Acessado em 06 agosto de 2017.

Angular 2. **APP LIFECYCLE.** Disponível em: <http://learnangular2.com/lifecycle/>. Acessado em 09 julho de 2017.

Angular 2. **Angular 2 Authentication Tutorial.** Disponível em: <https://auth0.com/blog/angular-2-authentication/>. Acessado em 13 agosto de 2017.

Angular 2. **O que é Angular JS ?** Disponível em: <http://www.andrefelizardo.com.br/blog/o-que-e-angularjs/>. Acessado em: 22 mar. 2017.

Angular 2. **Login com rede social.** Disponível em: <https://github.com/rudrakshpathak/angular2-google-login>. Acessado em 25 junho de 2017.

Angular 4. **Build.** Disponível em: <https://stackoverflow.com/questions/43030744/ng-build-prod-fails-after-upgrade-from-angular2-to-angular4>. Acessado em 06 agosto de 2017

Apache Maven Project. **What is Maven.** 2002-2017. Disponível em: <https://maven.apache.org/what-is-maven.html>. Acesso em 24 março. 2016.

Banco de Dados. **Conceitos Fundamentais de Banco de Dados.** Disponível em: <http://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>. Acessado em 02 maio 2017.

Befunky. **Editor de imagens.** Disponível em: <https://www.befunky.com/pt/criar/>. Acessado em 30 julho de 2017.

Bizagi. **The Digital Business Platform.**

Disponível em: <https://www.bizagi.com/pt/produtos/bpm-suite/modeler>. Acessado em 10 setembro de 2017.

Bootstrap. **Desenvolvendo com Bootstrap 3.** (s.d.). Acesso em 04 de 06 de 2017, disponível em Bootstrap: <http://thiagonasc.com/desenvolvimento-web/desenvolvendo-com-bootstrap-3-um-framework-front-end-que-vale-a-pena>

DeZign. **DeZign for Databases.** (s.d.). Acesso em 03 de 09 de 2017, disponível em DeZign: [http://download.cnet.com/DeZign-for-Databases/3000-2065\\_4-187946.html](http://download.cnet.com/DeZign-for-Databases/3000-2065_4-187946.html)

Eclipse. **O que é o eclipse ?** (s.d.). Acesso em 02 de 06 de 2017, disponível em IBM: <https://www.ibm.com/developerworks/br/library/os-eclipse-platform/>

Estadão, Edu. **Atendimento de crianças de 0 a 3 anos em creche aumenta, mas ainda em ritmo lento.** ESTADÃO EDUCAÇÃO. 2016.

Disponível em: <http://educacao.estadao.com.br/blogs/de-olho-na-educacao/atendimento-de-criancas-de-0-a-3-anos-em-creche-aumenta-mas-ainda-em-ritmo-lento>. Acesso em 06 abril. 2016.

Font Awesome. **Ícones.** Disponível em: <http://fontawesome.io/>. Acessado em 16 julho de 2017

Git. **Getting Started - About Version Control.** (s.d.). Acesso em 28 de 03 de 2016, disponível em GIT: <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

Github. **Deletar commit.** Disponível em: <https://www.youtube.com/watch?v=B5Ss4xNYWDY>. Acessado em 20 agosto de 2017.

Github. **Getting your project on GitHub.2016.** (s.d.). Acesso em 01 de 04 de 2016, disponível em GITHUB: <https://guides.github.com/introduction/getting-your-project-on-github>

Google. **Adicionando login na aplicação.** Disponível em: <https://hipsters.tech/single-page-applications-hipsters-16/>. Acessado em 25 junho de 2017.

Guedes, T. (2017). Angular. In: T. Guedes, *Crie Aplicações com Angular* (p. 1). São Paulo: Casa do Código.

Heroku. **Deploy Angular 2 CLI app to Heroku.**

Disponível em: <https://paucls.wordpress.com/2016/11/25/deploy-angular-2-cli-app-to-heroku/>. Acessado em 27 agosto de 2017.

Heroku. **Introdução.** (s.d.). Acesso em 04 de 07 de 2017, disponível em Imasters: <https://imasters.com.br/box/ferramenta/heroku/>

Heroku. **Realizando deploy de aplicação Angular.** Disponível em: <http://kazale.com/angular-2-heroku-deploy/>. Acessado em 02 julho de 2017.

Heroku. **Suporte para Node.** Disponível em: <https://devcenter.heroku.com/articles/nodejs-support#specifying-a-node-js-version>. Acessado em 27 agosto de 2017.

Heroku Deployment. **Realizando deploy de aplicação Angular.**

Disponível em: <https://docs.travis-ci.com/user/deployment/heroku/>. Acessado em 02 julho de 2017.

Hibernate. **Idiomatic persistence for Java and relational databases.**

Disponível em: <http://hibernate.org/orm/>. Acesso em 09 abril. 2016.

Hibernate. **Problemas com tipo Lob.**

Disponível em: <https://forum.hibernate.org/viewtopic.php?p=2452481>. Acessado em 27 agosto de 2017.

HTTPS. **Descrição dos status.** Disponível em: <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>. Acessado em 04 junho de 2017.

Nielsen. **Interação Humano Computador.**

Disponível em <http://www.lbd.dcc.ufmg.br/colecoes/wei/2003/0016.pdf>. Acessado em 29 de outubro de 2017.

Java. **Informações sobre a Tecnologia Java.** (s.d.). Acesso em 02 de 07 de 2017, disponível em Oracle: [https://www.java.com/pt\\_BR/about/](https://www.java.com/pt_BR/about/)

JPA. **From JPA to Hibernate's Legacy and Enhanced Identifier Generators.** Disponível em: <https://dzone.com/articles/jpa-hibernates-legacy-and>. Acessado em 03 setembro de 2017.

Karma. **Unit Testing JavaScript.**

Disponível em: <http://www.bradoncode.com/blog/2015/02/27/karma-tutorial/>. Acessado em 09 julho de 2017.

Lima, D. d. (22 de 11 de 2016). **Bizagi Modeler. Modeler processos de negócio.** Acesso em 10 de 09 de 2017, disponível em Tectudo: <http://www.techtudo.com.br/tudo-sobre/bizagi-modeler.html>

Maps JavaScript API. **Serviço Geocoding.** (s.d.). Acesso em 20 de 08 de 2017, disponível em Google: <https://developers.google.com/maps/documentation/javascript/geocoding?hl=pt-br>

NPM. **Build amazing things.** Disponível em: <https://www.npmjs.com/>. Acessado em 09 julho de 2017.

NPM. **How npm handles the "scripts" field.**

Disponível em: <https://docs.npmjs.com/misc/scripts>. Acessado em 27 agosto de 2017.

NPM. **Login com conta google.** Disponível em: <https://www.npmjs.com/package/angular2-google-login>. Acessado em 25 junho de 2017.

NPM JSON. **package json.** Disponível em: <https://docs.npmjs.com/files/package.json>. Acessado em 09 julho de 2017.

Postgres. **Connecting to Heroku Postgres from psql.**

Disponível em:

[https://www.packtpub.com/mapt/book/web\\_development/9781782177944/9/ch09lv11sec77/connecting-to-heroku-postgres-from-psql](https://www.packtpub.com/mapt/book/web_development/9781782177944/9/ch09lv11sec77/connecting-to-heroku-postgres-from-psql). Acessado em 03 setembro de 2017.

Postgres. **The Postgres Global Development Group**. (1996-2017). Acesso em 12 de 03 de 2017, disponível em Postgres: <https://www.postgresql.org/about/>

Rest. **Maturity Model**.

Disponível em: <https://martinfowler.com/articles/richardsonMaturityModel.html/>. Acessado em 18 junho de 2017.

Scrum. **Desenvolvimento Ágil**. 2014.

Disponível em: <http://www.desenvolvimentoagil.com.br/scrum/>. Acesso em 06 abril. 2017.

**SCHWABER Ken, SUTHERLAND Jeff**. Guia do Scrum. 2013

**SIERRA, Kathy; BATES, Bert**. SCJP: Certificação Sun para Programador Java. 1. ed. Rio de Janeiro. Alta Books, 2006.

Sonar. **Integração com Maven**. Disponível em: <https://dzone.com/articles/integration-jenkins-jacoco-and-sonarqube>. Acessado em 20 agosto de 2017.

Sonar. **Análise de Código utilizando SonarQube**.

Disponível em: <https://umovme.cc/2013/09/02/analise-de-codigo-utilizando-sonarqube/>. Acessado em 06 agosto de 2017.

Sonar. **Analysis Parameters**.

Disponível em: <https://docs.sonarqube.org/display/SONAR/Analysis+Parameters>. Acessado em 20 agosto de 2017

Spring Boot. **CORS support in Spring Framework**.

Disponível em: <https://spring.io/blog/2015/06/08/cors-support-in-spring-framework>. Acessado em 20 agosto de 2017.

Sonar. **Introdução**. (s.d.). Acesso em 08 de 10 de 2017, disponível em sonarsource: <https://www.sonarsource.com/products/codeanalyzers/sonarplsql.html>

Spring Boot. **Desenvolvendo no dia a dia**.

Disponível em: <https://cezbatistao.wordpress.com/2016/05/05/comecando-com-o-spring-boot-parte-2/>. Acessado em 04 junho de 2017.

Spring Boot. **Deploying Spring Boot Applications to Heroku**. Disponível em: <https://dzone.com/articles/spring-boot-heroku-and-cicd>. Acessado em 03 setembro de 2017.

Spring Boot. **Enabling Cross Origin Requests for a Restful Web Service**. Disponível em:

<https://dzone.com/articles/integration-jenkins-jacoco-and-sonarqube>. Acessado em 20 agosto de 2017.

Spring Boot. **Formatting Java Time with Spring Boot using JSON**. Disponível em:

<http://lewandowski.io/2016/02/formatting-java-time-with-spring-boot-using-json/>. Acessado em 20 agosto de 2017.

Spring Boot. **Injeção de Dependência**. Disponível em: <https://github.com/gabrielfeitosaci-spring-boot>. Acessado em 25 junho de 2017.

Spring Boot. **Project**. (s.d.). Acesso em 02 de 05 de 2017, disponível em [Spring Boot: https://projects.spring.io/spring-boot/](https://projects.spring.io/spring-boot/)

Spring Boot. **Propriedades**.

Disponível em: <https://docs.spring.io/spring-boot/docs/current/reference/html/common-application-properties.html>. Acessado em 20 agosto de 2017.

Spring Boot. **Spring From the Trenches: Parsing Date and Time Information From a Request Parameter**.

Disponível em: <https://www.petrikainulainen.net/programming/spring-framework/spring-from-the-trenches-parsing-date-and-time-information-from-a-request-parameter/>. Acessado em 20 agosto de 2017.

Spring Data. **Spring Data REST é Vida**.

Disponível em: <https://domineiospring.wordpress.com/category/springdata/>. Acessado em 04 junho de 2017.

Spring MVC. **Spring MVC - Caelum**. Disponível em: <https://www.caelum.com.br/apostila-java-web/spring-mvc/>. Acessado em 18 junho de 2017.

**SUTHERLAND**, Jeff. **SCRUM**. A arte de fazer o dobro do trabalho na metade do tempo. 1. ed. São Paulo. Grupo Leya, 2014.

Swagger. **Annotations**.

Disponível em: <https://github.com/swagger-api/swagger-core/wiki/annotations>. Acessado em 25 junho de 2017.

Swagger 2. **Spring Boot RESTful API Documentation With Swagger 2**. (s.d.). Acesso em 04 de 06 de 2017, disponível em DZONE: <https://dzone.com/articles/spring-boot-restful-api-documentation-with-swagger>

Travis. **Adicionando aplicação Angular**. Disponível em: <https://medium.com/from-the-couch/angular-2-with-travis-ci-922040e01937>. Acessado em 25 junho de 2017.

Travis. **Configurações**. Disponível em: <http://stefanteixeira.com.br/2014/09/01/configurando-travis-ci-para-um-projeto-com-ghostdriver-e-gradle/>. Acessado em 03 setembro de 2017.

Travis. **Configurando Travis CI**. (01 de 09 de 2014). Acesso em 14 de 06 de 2017, disponível em <http://stefanteixeira.com.br/2014/09/01/configurando-travis-ci-para-um-projeto-com-ghostdriver-e-gradle/>

Travis. **Using SonarCloud with Travis CI**.

Disponível em: <https://docs.travis-ci.com/user/sonarcloud/>. Acessado em 13 agosto de 2017.

Travis. **Using Travis Ci With Angular**.

Disponível em: <https://codeforgeek.com/2017/03/deploy-awesome-angular-app-heroku/>. Acessado em 02 julho de 2017.



Travis CI. **Building a Java project**. 2011-2017.

Disponível em: <https://docs.travis-ci.com/user/languages/java>. Acesso em 10 março. 2016.

The Twelve Factor App. **Os doze fatores**.

Disponível em: [https://12factor.net/pt\\_br/](https://12factor.net/pt_br/). Acessado em 18 junho de 2017.

TypeScript. **Why TypeScript ?** (s.d.). Acesso em 04 de 06 de 2017, disponível em <https://www.typescriptlang.org>

Visual Studio Code. **Apresentando o editor multiplataforma da Microsoft**. (s.d.). Acesso em 04 de 06 de 2017, disponível em Imasters: <https://imasters.com.br/desenvolvimento/visual-studio-code-apresentando-o-editor-multiplataforma-da-microsoft/?trace=1519021197&source=single>