

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 ATUALIZADO

KIDS

Aplicação web de interação entre creches e familiares

Porto Alegre
2017

LUCIANO ORTIZ SILVA

RELATÓRIO DE PROJETO ATUALIZADO

KIDS

Aplicação web de interação entre creches e familiares

Relatório de Trabalho 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

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	8
Figura 2 - Descrição da solução	15
Figura 3 - Visão geral do ciclo de vida do Scrum	16
Figura 4 - Visão geral da arquitetura	17
Figura 5 - Product Backlog.....	18
Figura 6 - User Stories.....	19
Figura 7 - Sprint 01.....	23
Figura 8 - Sprint 02.....	24
Figura 9 - Sprint 03.....	24
Figura 10 - Sprint 04.....	25
Figura 11- Sprint 05.....	26
Figura 12- Sprint 06.....	26
Figura 13 - Sprint 07.....	27
Figura 14 - Sprint 08.....	27
Figura 15 - Sprint 09.....	27
Figura 16 - Visão geral das regras de negócio	28
Figura 17 - Modelo de dados	29
Figura 18 - Tela de login	30
Figura 19 - Tela de cadastro de uma nova conta	30
Figura 20 - Tela de home - visão creche	31
Figura 21 - Tela de wizard - cadastro crianças - passo 1.....	31
Figura 22 - Tela de wizard - cadastro crianças - passo 2.....	32
Figura 23 - Tela de wizard - cadastro crianças - passo 3.....	32
Figura 24 - Tela de wizard - cadastro crianças - passo 4.....	33
Figura 25 - Tela de wizard - cadastro crianças - passo 5.....	33
Figura 26 - Tela para vincular novos usuários a crianças da creche	34
Figura 27 - Tela de diário escolar - visão creche.....	34
Figura 28 - Tela de diário escolar - visão familiar	35
Figura 29 - Tela de avisos - visão creche	35
Figura 30 - Tela de home/avisos - visão familiar	36
Figura 31- Funcionamento do sistema	37
Figura 32 - Cronograma de TCC 1.....	39
Figura 33 - Cronograma de TCC 2.....	39

SUMÁRIO

1.	APRESENTAÇÃO GERAL DO PROJETO	6
2.	DEFINIÇÃO DO PROBLEMA	7
3.	OBJETIVOS	9
3.1	OBJETIVO GERAL	9
3.2	OBJETIVOS ESPECÍFICOS	9
4.	ANÁLISE DE TECNOLOGIAS/FERRAMENTAS	10
4.1	LINGUAGENS DE PROGRAMAÇÃO.....	10
4.2	FRAMEWORKS.....	11
4.3	BANCO DE DADOS.....	11
4.3.1	POSTGRESQL.....	11
4.4	AMBIENTE DE DESENVOLVIMENTO	12
4.4.1	ECLIPSE	12
4.4.2	VISUAL STUDIO CODE.....	12
4.5	ARTEFATOS DE VERSIONAMENTO	12
4.5.1	GIT	12
4.5.2	GITHUB.....	13
4.6	ARTEFATO DE MODELAGEM.....	13
4.7	INTEGRAÇÃO CONTÍNUA	13
4.8	SERVIDOR DE HOSPEDAGEM	14
5.	DESCRIÇÃO DA SOLUÇÃO.....	15
6.	ABORDAGEM DE DESENVOLVIMENTO	16
7.	ARQUITETURA DO SISTEMA	17
7.1	MODELAGEM FUNCIONAL.....	18
7.2	MODELAGEM DE PROCESSO DE NEGÓCIO	28
7.3	MODELAGEM DE DADOS.....	29
7.4	MODELAGEM DE INTERFACE GRÁFICA DO USUÁRIO	30
8.	FUNCIONAMENTO DO SISTEMA.....	37
9.	VALIDAÇÃO.....	38
9.1	ESTRATÉGIA	38
10.	CRONOGRAMA	39
	REFERÊNCIAS	40
	APÊNDICE A.....	46

1. APRESENTAÇÃO GERAL DO PROJETO

A cada ano, inicia-se mais um ano letivo em nosso país. É mais um ano, onde serão inseridos pela sua primeira vez, diversas crianças em quaisquer instituições de ensino infantil (creches), tanto no âmbito público, quanto no privado. Tratando-se de educação, é a base inicial do futuro desse país.

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, 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 trajes 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? Evacuou?

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.

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

custo: gratuito

- **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!

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

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

- **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.

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

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

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
check-in	X			
check-out	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 estão 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:

- a) Manter a creche informada sobre alguma alergia que alguma criança possa conter.
- b) Informar aos familiares detalhes sobre um determinado evento a ser realizado na creche.
- c) Manter um quadro de cardápio completo e atualizado por parte da creche. Disponibilizando o mesmo para todos os familiares;
- d) Uma forma de familiares de comunicarem com a creche, mantendo um registro dessa comunicação.
- e) Deixar claro e visível para a creche, dados de medicamentos de uso da criança;
- f) Possibilitar a creche disponibilizar diversas imagens dos eventos que a mesma realizar.
- g) Gerar avisos de sobre alguma informação importante.
- h) Manter dados das crianças devidamente atualizados.
- i) Disponibilizar para os familiares, informações sobre algumas atividades diárias da criança.

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

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

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.

Esses recursos estão disponíveis no tempo de desenvolvimento para o desenvolvimento de aplicativos de alta confiança, mas são compilados em JavaScript simples que almeja os ambientes ECMAScript 3 (ou mais recentes). Será utilizado nesse projeto, por se tratar de uma linguagem que já vem integrada com o framework Angular JS.

4.2 FRAMEWORKS

4.2.1 ANGULAR JS 4

AngularJS é um framework JavaScript open-source, mantido pela Google, cujo objetivo é auxiliar o desenvolvimento de aplicações web. O AngularJS auxilia na construção de aplicações web, onde utiliza-se uma técnica conhecida como SPAs (do inglês, Single Page Applications). Por esta razão, o Angular JS versão 4 será utilizado no desenvolvimento do sistema proposto.

4.2.2 BOOTSTRAP

O Bootstrap é um conjunto de ferramentas de código aberto para desenvolvimento com HTML, CSS e JS. Podemos prototipar rapidamente de nossas idéias 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

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 POSTGRESQL

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. 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. A utilização deste DB neste trabalho visa a integração do sistema proposto com banco de dados de modo a abranger a persistência de dados da aplicação.

4.4 AMBIENTE DE DESENVOLVIMENTO

4.4.1 ECLIPSE

O eclipse é uma ferramenta para realizar o desenvolvimento de código fonte.

O eclipse iniciou quando os líderes da indústria Borland, IBM, MERANT, QNX Software Systems, Red Hat, SuSE, TogetherSoft e Webgain formaram o Conselho de Governadores do eclipse.org em novembro de 2001. No final de 2003, este consórcio inicial cresceu para mais de 80 membros.

Em 2 de fevereiro de 2004, o Conselho de Administração da Eclipse anunciou a reorganização do Eclipse em uma corporação sem fins lucrativos. Originalmente um consórcio que se formou quando a IBM lançou a Eclipse em Open Source, o Eclipse tornou-se um órgão independente que impulsionará a evolução da plataforma para beneficiar os provedores de ofertas de desenvolvimento de software e usuários finais. Toda a tecnologia e o código fonte fornecidos e desenvolvidos por esta comunidade em rápido crescimento são disponibilizados sem royalties através da Eclipse licença pública

Os desenvolvedores estratégicos e os consumidores estratégicos fundadores foram a Ericsson, HP, IBM, Intel, MontaVista Software, QNX, SAP, e Serena Software.

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

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 pelo fato de possuir fácil integração com o framework AngularJS 4.

4.5 ARTEFATOS DE VERSIONAMENTO

4.5.1 GIT

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.”

4.5.2 GITHUB

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

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.

4.6.2 BIZAGI

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.

4.7 INTEGRAÇÃO CONTÍNUA

4.7.1 TRAVIS

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

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 fácil integração o a ferramenta Travis.

4.9 BIBLIOTECAS EXTERNAS

4.9.1 API SWAGGER

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.

4.9.2 API GOOGLE MAPS

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.

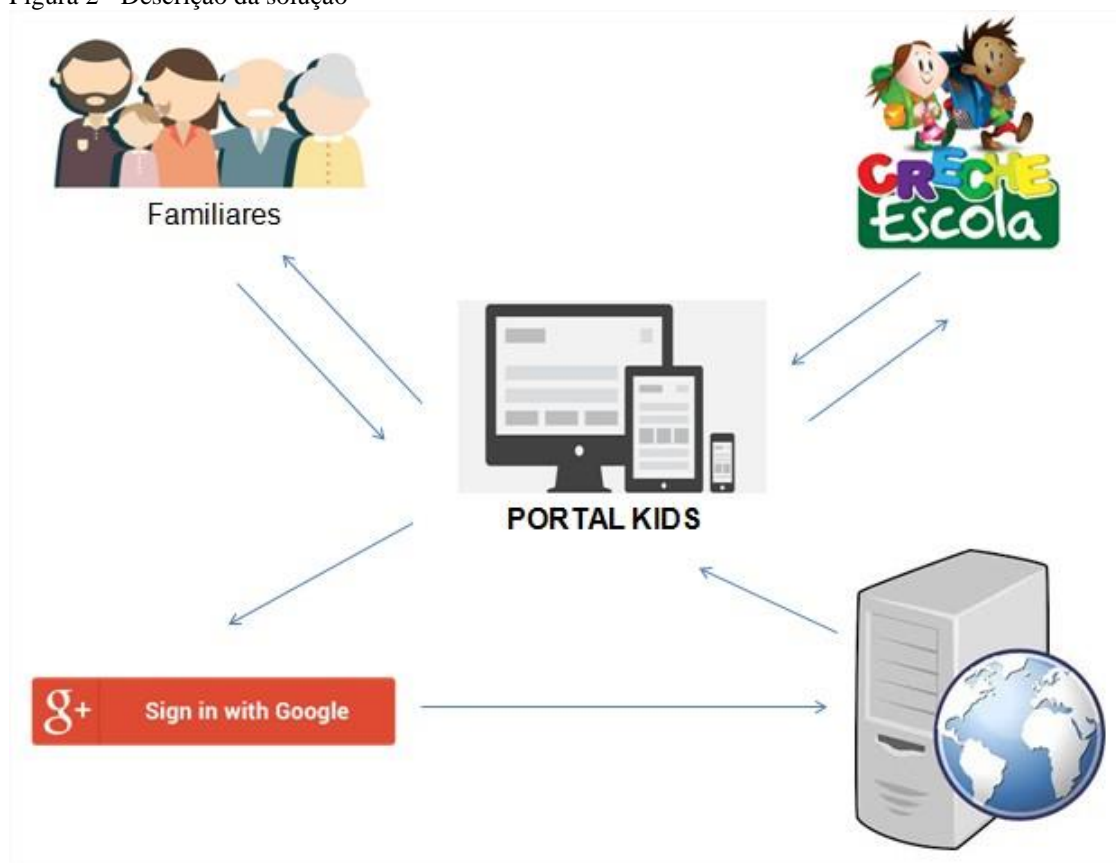
5. DESCRIÇÃO DA SOLUÇÃO

O sistema foi organizado da seguinte forma:

A instituição de ensino, que demonstra interesse pelo portal, realiza um simples cadastro para iniciar sua utilização. Onde a principal informação se dará pelo *e-mail google*, o qual será seu login, assim como sua ponte de ligação entre seus alunos, e os responsáveis por eles.

Já aqueles familiares que desejam acessar dados de suas crianças, basta solicitar seu vínculo junto a instituição de ensino. Após um familiar ser vinculado a uma creche, basta o mesmo efetuar o login, e sair navegando o portal.

Figura 2 - Descrição da solução



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 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 (James Brum), afim de organizar uma melhor forma para preencher um determinado sprint por um período de 15 dias.

Passado o fim de 15 dias de desenvolvimento, as funcionalidades estão sendo entregues para a Project Owner Liziane Ortiz, onde a mesma tem a oportunidade para "dar o aceite", sugerir melhorias, e sugerir novas funcionalidades para o projeto.

Ao final de cada sprint, foi realizado 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.

Figura 3 - Visão geral do ciclo de vida do Scrum



Fonte: Criado pelo autor do projeto

7. ARQUITETURA DO SISTEMA

Na prática, foi desenvolvido 1 projeto web, que roda em um servidor HTTP. Esse projeto conta com artefatos de html, css, Javascript, desenvolvidos através do framework AngularJS 4. Ou seja, apenas a parte de visualização para os usuários. Por outro lado, também foi construído um outro projeto, onde ficam as regras de negócios, assim como acesso a base de dados. Nesse projeto foi desenvolvido utilizando o framework Spring Boot. Para realizar a comunicação entre os projetos, foram criados web services, para que através da view, sejam consumidos os serviços necessários para a utilização do portal.

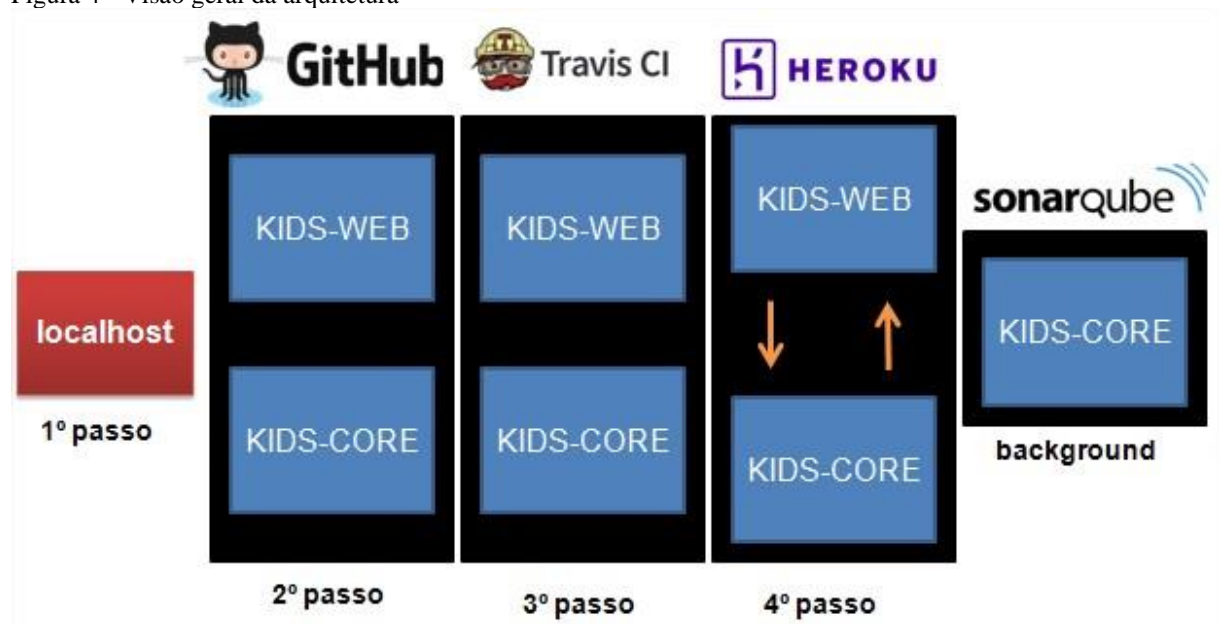
No dia a dia do desenvolvimento, o projeto seguiu o seguinte processo:

A cada commit realizado no repositório github, automaticamente é feita uma chamada para uma ferramenta realizar uma integração continua. Essa ferramenta chama-se *Travis*. Quando o Travis é chamado, é realizado toda a construção dos artefatos necessários para que a aplicação rode com sucesso, além disso são executados automaticamente todos os testes unitários desenvolvidos. Ao final da execução também é realizada uma chamada para a ferramenta chamada Sonar, que é responsável por avaliar a qualidade do código comitado.

Quando o Travis termina de ser executado, caso não ocorrer nenhuma falha, então é realizado o deploy no servidor chamado Heroku, para que seja disponibilizada uma nova versão a disposição do usuário final.

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

Figura 4 - Visão geral da arquitetura



Fonte: Criado pelo autor do projeto.

7.1 MODELAGEM FUNCIONAL

Para a realização deste projeto, foi utilizado como modelagem funcional, o Product Backlog, as User Stories, e o Sprint Backlog.

7.1.2 PRODUCT BACKLOG

No Product Backlog, foi definido juntamente com o Product Owner, quais funcionalidades teriam prioridades na etapa de desenvolvimento.

Segue abaixo uma tabela ilustrativa, exibindo um checklist do product backlog gerado.

Figura 5 - Product Backlog

Product Backlog		
	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	Vinculo 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
*PB = Product Backlog		

Fonte: Criado pelo autor do projeto.

7.1.3 User Stories

Através das Users Stories, foram detalhadas as necessidades dos usuários. Segue abaixo uma tabela ilustrativa, exibindo todo o conteúdo das Users Stories:

Figura 6 - User Stories

Product Backlog - Detalhado por User Stories
PB-01 Login com conta google
US-01 - Realizar login com conta google
Como um usuário interessado em utilizar o portal
Eu quero realizar meu login usando minha conta google
Para que seja possível me cadastrar no sistema
PB-02 Controle de usuários
US-02 - Definir tipo de conta do 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 - Cadastrar perfil do usuário
Como um usuário do tipo creche ou familiar
Eu quero cadastrar minhas informações de perfil
Para que sejam mantidos meus dados atualizados
PB-03 Módulo de crianças
US-04 - Pesquisar, cadastrar, e editar crianças
Como um usuário do tipo creche
Eu quero cadastrar crianças na minha instituição
Para que eu possa manter e dirigir o controle sobre os seus dados pessoais
US-05 - Pesquisar e editar crianças
Como um usuário do tipo creche ou familiar
Eu quero cadastrar / editar medicamentos para as crianças
Para que lembrar do seu uso diário

PB-04 - Controle de medicamentos e alergias das crianças
US-06 - Cadastrar e editar 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 - Cadastrar e editar alergias das crianças
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
PB-05 - Vinculo de familiares com as crianças cadastradas na creche
US-08 - Vincular familiares com as crianças
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
PB-06 - Módulo de diário escolar
US-09 - Cadastrar, editar 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 - Visualizar diário escolar
Como um usuário do tipo familiar
Eu quero visualizar o diário escolar das crianças que possuo vinculo
Para que possa acompanhar suas atividades diariamente
PB-07 - Módulo de eventos
US-11 - Cadastrar / cancelar eventos
Como um usuário do tipo creche
Eu quero cadastrar e cancelar eventos
Para que notificar os familiares para que se programem a vir na data informada

US-12 - Aceitar / Rejeitar 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
PB-08 - Módulo de avisos
US-13 - Cadastrar avisos
Como um usuário do tipo creche
Eu quero cadastrar avisos
Para que aos familiares sejam notificados de algo importante
US-14 - Visualizar avisos
Como um usuário do tipo familiar
Eu quero visualizar os avisos deixados pela creche
Para que eu possa tomar alguma ação quando necessário
PB-09 - Módulo de contato entre familiar x creche
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
PB-10 - Módulo de avaliação escolar
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 vínculo
Para que eu possa acompanhar seu desenvolvimento escolar

PB-11 - Módulo de cardápio	
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	
PB-12 - Módulo de galeria de fotos	
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.

7.1.4 Sprint Backlog

Após o Product Backlog ter sido detalhado através de diversas User Stories, foram planejados inicialmente 10 sprints, contendo em cada sprint um backlog de users Stories. Abaixo segue detalhes sobre as sprints já realizadas, e sobre as próximas sprints planejadas.

➤ **Sprint 01:**

Figura 7 - Sprint 01

Sprint 01	
US-01	Realizar login com conta google
US-02	Definir tipo de conta do usuário
US-03	Cadastrar perfil do usuário

Essa Sprint teve início no dia 16/06/2017, e término no dia 30/06/2017.

Sprint Review:

Foram realizados os aceites de:

- Login com conta google;
- Escolha do tipo de usuário;
- Cadastro de perfil;

Foram solicitadas as seguintes melhorias:

- Descrever na capa do sistema o que ele faz;
- Breve texto sobre o tipo de usuário;
- Adicionar telefone no 'perfil reduzido';
- Destacar o 'tipo de usuário';

Foram acordadas com o cliente que as melhorias citadas acima, serão entregues no decorrer do projeto.

Retrospectiva:

Pontos positivos:

- Entrega em dia dos requisitos funcionais;
- Desenvolvimento do front-end com a última versão do Angular;
- Aplicação hospedada no Heroku;
- Adição do Sonar;
- Criação de um repositório somente para artefatos de documentação;

Pontos negativos:

- Sem internet para o desenvolvimento durante uns 3/4 dias;
- Remodelagem da estrutura do front-end, durante umas 5/6 vezes;
- Bastante horas utilizadas p\ subir o Angular 4 no Heroku (até achar o cmd correto);
- Sonar ainda não configurado por completo;
- Pouco teste unitário;

➤ **Sprint 02:**

Figura 8 - Sprint 02

Sprint 02	
US-04	Pesquisar, cadastrar, e editar crianças
US-05	Pesquisar e editar crianças
US-06	Cadastrar e editar medicamentos
US-07	Cadastrar e editar alergias das crianças

Essa Sprint teve início no dia 16/07/2017, e término no dia 30/07/2017.

Sprint Review:

Foram realizados os seguintes:

- Wizard para cadastro de crianças;

- Pesquisa de crianças;

Foram solicitadas as seguintes melhorias:

- Adicionar número de telefone na área de perfil resumido;

- Cadastrar/editar a foto da criança;

Foram acordadas com o cliente que as melhorias citadas acima, serão entregues no decorrer do projeto.

Retrospectiva:

Pontos positivos:

- Aumento na cobertura de testes unitários;

- Refactory de código fonte / lado servidor;

- Criação de um repositório somente para artefatos de DB, para agilizar os testes;

Pontos negativos:

- Sem internet para o desenvolvimento durante uns 4 dias;

- Remoção do campo "foto da criança" (problema técnico);

- Os dois sistemas distribuídos não conseguem mais se comunicar no servidor Heroku;

➤ **Sprint 03:**

Figura 9 - Sprint 03

Sprint 03	
US-08	Cadastrar, editar diário escolar

Essa Sprint teve início no dia 01/08/2017, e término no dia 12/08/2017.

Sprint Review:

Foram realizados os seguintes:

- Vínculo de crianças com seus familiares;

Melhoria de adicionar número de telefone na área de perfil resumido;
 Melhoria de destacar o 'tipo de usuário';
 Melhoria de ajuste dos textos em todas as telas onde não está responsivo;

Retrospectiva:

Pontos positivos:

Entrega de algumas melhorias solicitadas anteriormente pelo cliente;
 Entrega 3 dias antes do prazo;

Pontos negativos:

Os dois sistemas distribuídos ainda permanecem sem se comunicar no servidor Heroku.

➤ **Sprint 04:**

Figura 10 - Sprint 04

Sprint 04	
US-09	Cadastrar, editar diário escolar
US-10	Visualizar diário escolar

Essa Sprint teve início no dia 16/08/2017, e término no dia 29/08/2017.

Sprint Review:

Foram realizados os aceites de:

Visualizar diários das crianças vinculadas (familiar);
 Cadastrar e editar diário dos alunos (creche);
 Melhoria de descrever na capa do sistema o que ele faz melhoria;

Retrospectiva:

Pontos positivos:

Entrega de melhorias na capa do portal kids;
 Entrega 2 dias antes do prazo estimado;

Pontos negativos:

Os dois sistemas distribuídos ainda permanecem sem se comunicar no servidor Heroku;

Problemas para trabalhar na geração de ids para o banco de dados, utilizando a opção de sequence por tabela do hibernate 5;

➤ Sprint 05:

Figura 11- Sprint 05

Sprint 05	
US-11	Cadastrar / cancelar eventos
US-12	Aceitar / Rejeitar eventos
US-13	Cadastrar avisos
US-14	Visualizar avisos

Essa Sprint teve início no dia 01/09/2017, e término no dia 10/09/2017.

Sprint Review:

Foram realizados os aceites de:

- Cadastrar e cancelar eventos (creche);
- Aceitar ou Rejeitar presença nos eventos (familiar);
- Cadastrar avisos (creche);
- Visualizar avisos (familiar);

Obs.: Nesse sprint, 2 User Stories que estavam planejadas para o sprint 06, foram adiantadas para esse sprint.

Retrospectiva:

Pontos positivos:

- Foram adiantadas tarefas de outros sprints;
- Entrega 5 dias antes do prazo;
- Com internet durante todo sprint;
- Os sistemas distribuídos dentro do Heroku, voltaram a funcionar.

Pontos negativos:

- Redução na cobertura de testes unitários;

➤ Sprint 06:

Figura 12- Sprint 06

Sprint 06	
US-15	Reclamar, elogiar, sugerir
US-16	Feedback da reclamação, elogio ou sugestão

Essa Sprint tem previsão para início no dia 16/09/2017, e término no dia 30/09/2017.

➤ **Sprint 07:**

Figura 13 - Sprint 07

Sprint 07	
US-17	Cadastrar / editar avaliação
US-18	Visualizar avaliação

Essa Sprint tem previsão para início no dia 01/10/2017, e término no dia 15/10/2017.

➤ **Sprint 08:**

Figura 14 - Sprint 08

Sprint 08	
US-19	Cadastrar cardápio
US-20	Visualizar cardápio

Essa Sprint tem previsão para início no dia 16/10/2017, e término no dia 31/10/2017.

➤ **Sprint 09:**

Figura 15 - Sprint 09

Sprint 09	
US-21	Cadastrar galeria de fotos
US-22	Visualizar galeria de fotos

Essa Sprint tem previsão para início no dia 01/11/2017, e término no dia 15/11/2017.

7.2 MODELAGEM DE PROCESSO DE NEGÓCIO

Para criar uma abstração de como irá funcionar o negócio, segue abaixo as regras de negócio implementadas no sistema:

Figura 16 - Visão geral das regras de negócio

RN	Familiar editar / visualizar dados de crianças
1	Um familiar poderá editar quaisquer dados de crianças, porém apenas as crianças na quais o mesmo possui vínculo no sistema. Não podendo se quer visualizar dados de outra criança na qual não tenha vínculo habilitado no sistema.
RN	Creche editar / visualizar dados de crianças
2	Uma creche poderá editar quaisquer dados de crianças, porém apenas as crianças na quais tenham sido cadastradas pela creche logada. Não podendo se quer visualizar dados de outra criança na qual não tenha sido cadastrada.
RN	Vínculo de familiares - email
3	No momento o sistema permitirá apenas cadastros para pessoas que possuam login com conta google
RN	Vínculo de familiares - somente usuário familiar
4	O sistema não deverá permitir vínculo de familiares que já possuam seu e-mail associados a uma creche no sistema.
RN	Cadastro automático - creche
5	Quando o usuário entrar pela primeira vez no sistema e ainda não possuir cadastro. Se desejar se cadastrar como um usuário do tipo creche, basta o usuário escolher o tipo de conta creche, e cadastrar-se automaticamente no sistema.
RN	Cadastro automático - familiar
6	Quando o usuário entrar pela primeira vez no sistema e ainda não possuir cadastro. Se desejar se cadastrar como um usuário do tipo familiar, basta o usuário escolher o tipo de conta familiar, e cadastrar-se automaticamente no sistema. Porém o mesmo permanecerá inativo, até que seja vinculado há alguma creche.

Fonte: Criado pelo autor do projeto.

7.4 MODELAGEM DE INTERFACE GRÁFICA DO USUÁRIO

A seguir serão exibidas as interfaces gráficas tanto na visão de um usuário familiar, quanto na visão de um usuário do tipo creche.

Figura 18 - Tela de login

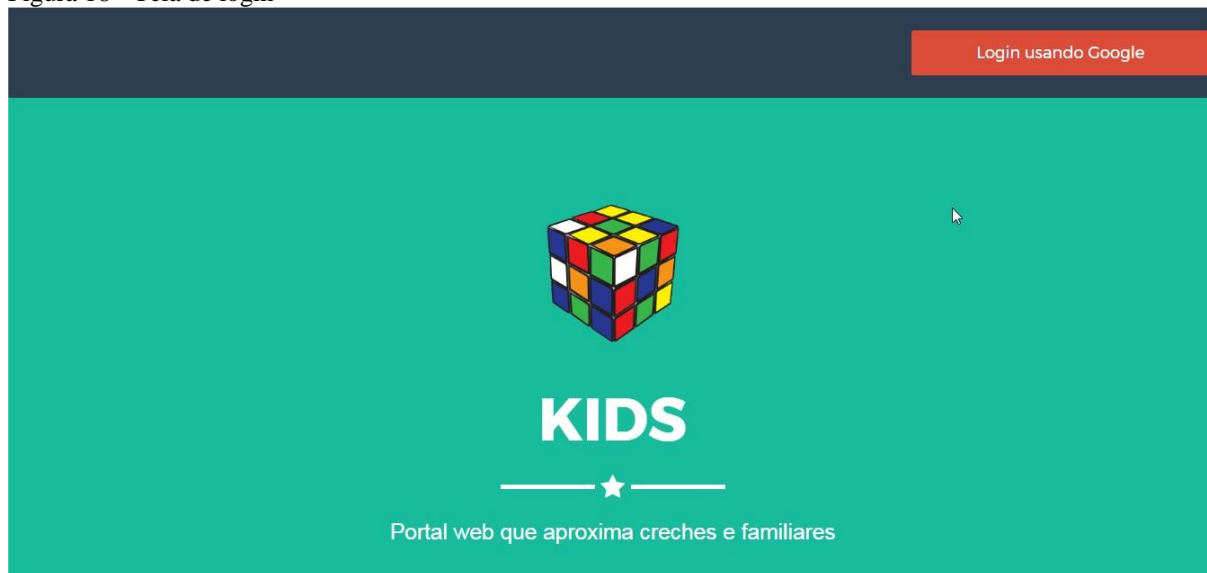


Figura 19 - Tela de cadastro de uma nova conta

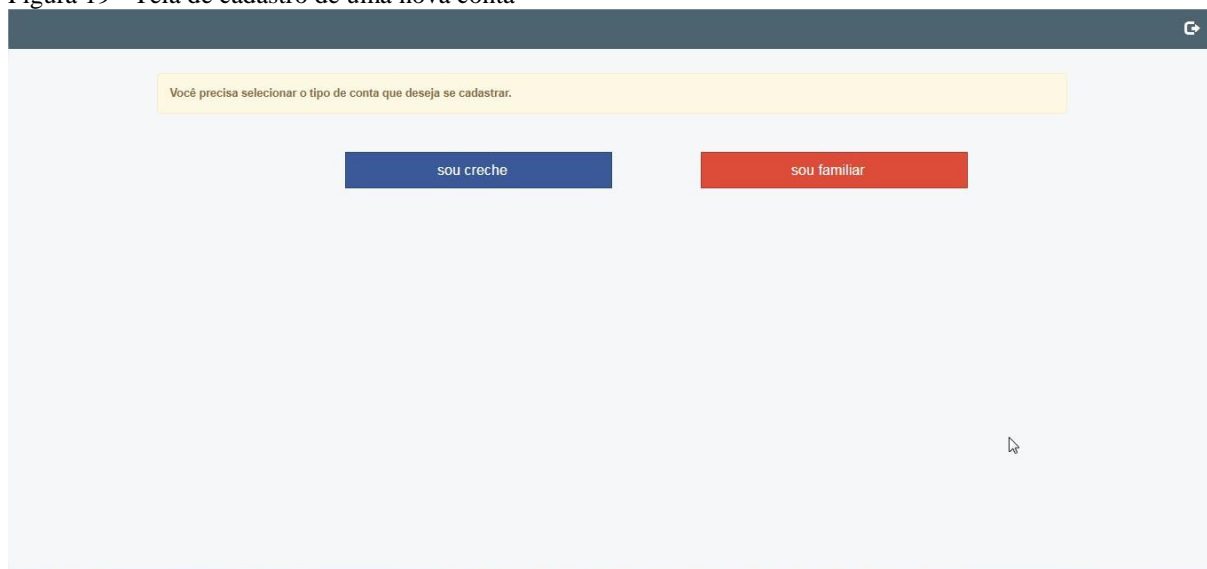


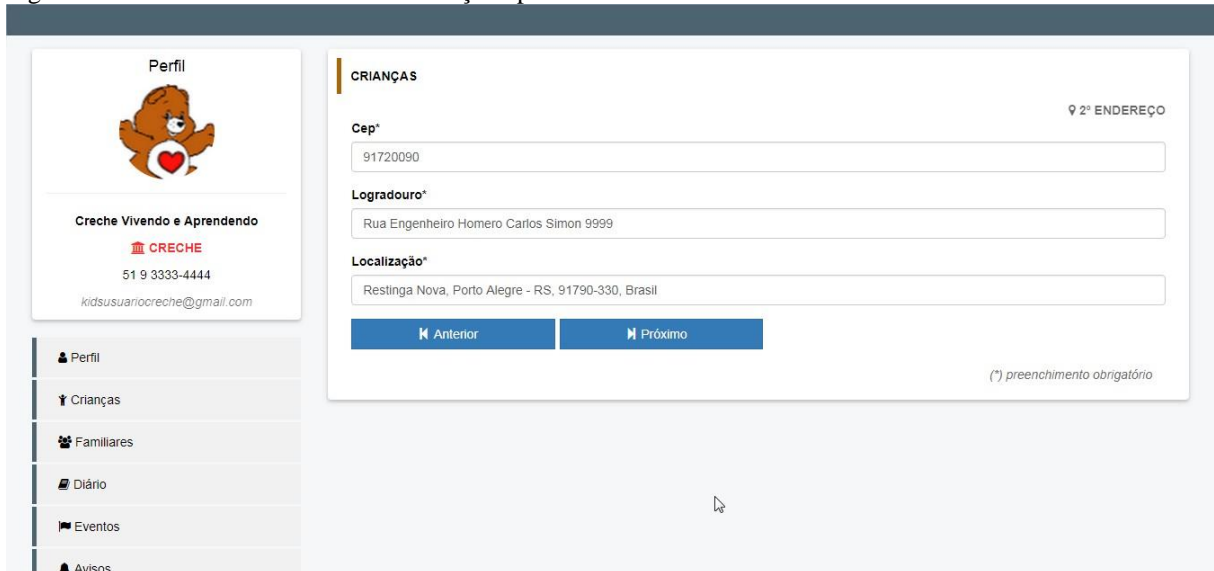
Figura 20 - Tela de home - visão creche

The screenshot shows a web application interface for a daycare. On the left is a sidebar menu with icons and labels for 'Perfil', 'Crianças', 'Familiares', 'Diário', 'Eventos', and 'Avisos'. The main content area has a header with a 'Perfil' section containing a cartoon bear logo, the name 'Creche Vivendo e Aprendendo', the logo 'CRECHE', the phone number '51 9 3333-4444', and the email 'kidsusuariocreche@gmail.com'. Below this is a large white box with the text 'Olá Creche Vivendo e Aprendendo'.

Figura 21 - Tela de wizard - cadastro crianças - passo 1

The screenshot shows the 'CRIANÇAS' registration wizard. The sidebar menu is the same as in Figure 20. The main content area has a header 'CRIANÇAS' and a sub-header '1º DADOS GERAIS'. Below this are four form fields: 'Matricula*' with the value '0001', 'Nome*' with the value 'Mariana Da Cruz Ortiz Silva', 'Data de Nascimento*' with the value '29/08/2011', and 'Sexo*' with a dropdown menu showing 'FEMININO'. At the bottom of the form are two buttons: 'Voltar' (orange) and 'Próximo' (blue). A footnote at the bottom right states '(*) preenchimento obrigatório'.

Figura 22 - Tela de wizard - cadastro crianças - passo 2



The screenshot shows a web application interface for child registration. On the left is a sidebar with a 'Perfil' section containing a logo of a bear holding a heart, the text 'Creche Vivendo e Aprendendo', 'CRECHE', the phone number '51 9 3333-4444', and the email 'kidsusuariocreche@gmail.com'. Below this are menu items: Perfil, Crianças, Familiares, Diário, Eventos, and Avisos. The main area is titled 'CRIANÇAS' and contains three input fields: 'Cep*' with the value '91720090', 'Logradouro*' with 'Rua Engenheiro Homero Carlos Simon 9999', and 'Localização*' with 'Restinga Nova, Porto Alegre - RS, 91790-330, Brasil'. At the bottom of the form are two blue buttons: 'Anterior' and 'Próximo'. A note at the bottom right states '(*) preenchimento obrigatório'.

Perfil

Creche Vivendo e Aprendendo

CRECHE

51 9 3333-4444

kidsusuariocreche@gmail.com

Perfil

Crianças

Familiares

Diário

Eventos

Avisos

CRIANÇAS

Cep* 91720090

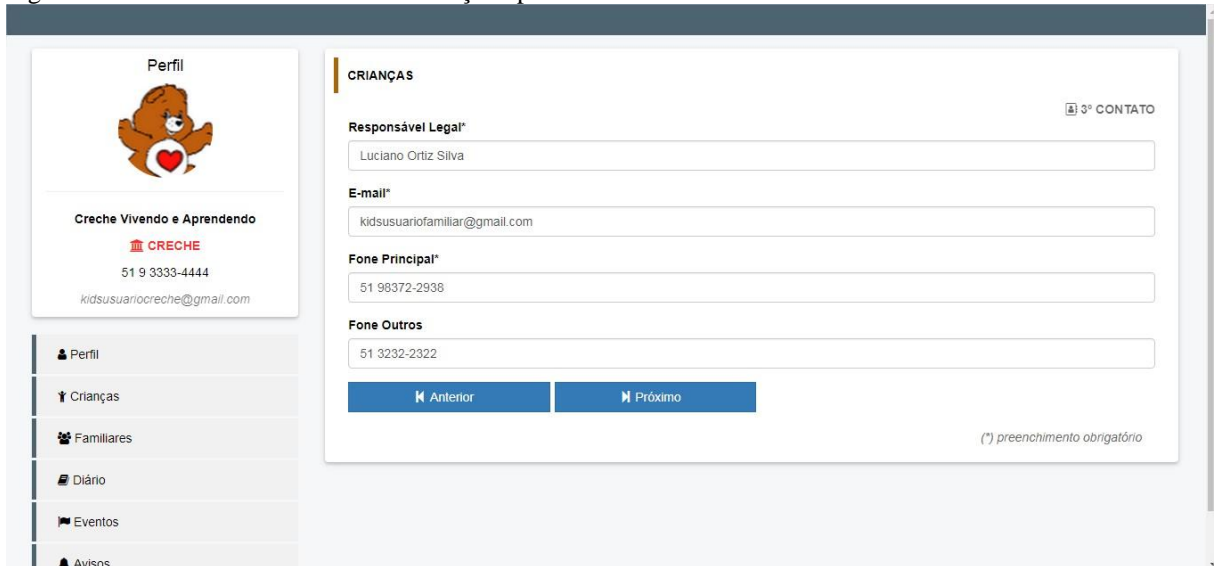
Logradouro* Rua Engenheiro Homero Carlos Simon 9999

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

Anterior Próximo

(*) preenchimento obrigatório

Figura 23 - Tela de wizard - cadastro crianças - passo 3



The screenshot shows the next step in the child registration wizard. The sidebar is identical to the previous screen. The main area is titled 'CRIANÇAS' and contains three input fields: 'Responsável Legal*' with the value 'Luciano Ortiz Silva', 'E-mail*' with 'kidsusuariofamiliar@gmail.com', and 'Fone Principal*' with '51 98372-2938'. Below these is a 'Fone Outros' field with the value '51 3232-2322'. At the bottom of the form are two blue buttons: 'Anterior' and 'Próximo'. A note at the bottom right states '(*) preenchimento obrigatório'.

Perfil

Creche Vivendo e Aprendendo

CRECHE

51 9 3333-4444

kidsusuariocreche@gmail.com

Perfil

Crianças

Familiares

Diário

Eventos

Avisos

CRIANÇAS

Responsável Legal* Luciano Ortiz Silva

E-mail* kidsusuariofamiliar@gmail.com

Fone Principal* 51 98372-2938


Fone Outros 51 3232-2322

Anterior Próximo

(*) preenchimento obrigatório

Figura 24 - Tela de wizard - cadastro crianças - passo 4

Perfil



Creche Vivendo e Aprendendo

CRECHE

51 9 3333-4444

kidsusuariocreche@gmail.com

Perfil

Crianças

Familiares

Diário

Eventos

Avisos

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

Figura 25 - Tela de wizard - cadastro crianças - passo 5

Perfil



Creche Vivendo e Aprendendo

CRECHE

51 9 3333-4444

kidsusuariocreche@gmail.com

Perfil

Crianças

Familiares

Diário

Eventos

Avisos

CRIANÇAS

5º ALERGIA

Descrição

Adicionar

DETALHES

- lactose
- glúten


Remover

Remover

Anterior

Salvar

Figura 26 - Tela para vincular novos usuários a crianças da creche



Perfil

Creche Vivendo e Aprendendo

CRECHE

51 9 3333-4444

kidsusuariocreche@gmail.com

Perfil

Crianças

Familiares

Diário

Eventos

Avisos

VINCULAR USUÁRIOS COM CRIANÇAS

Criança

Mariana Da Cruz Ortiz Silva

Nome

E-mail

@gmail.com


Parentesco

[Voltar](#) [+ Add e Salvar](#)

DETALHES

- Luciano Ortiz - Test
- kidsusuariofamiliar@gmail.com
- PAI

Figura 27 - Tela de diário escolar - visão creche



Perfil

Creche Vivendo e Aprendendo

CRECHE

51 9 3333-4444

kidsusuariocreche@gmail.com

Perfil

Crianças

Familiares

Diário

Eventos

Avisos

DIÁRIO

Diário atualizado com sucesso!

Café da manhã

Almoço

Café da tarde

Janta

Sono

Evacuação

☕ CAFÉ DA MANHÃ

Amanda Silva	<input type="radio"/> Pouco	<input type="radio"/> Normal	<input type="radio"/> Muito
Lucas Portella	<input type="radio"/> Pouco	<input type="radio"/> Normal	<input type="radio"/> Muito
Mariana Da Cruz Ortiz Silva	<input checked="" type="radio"/> Pouco	<input type="radio"/> Normal	<input type="radio"/> Muito
Rafaela Farias	<input type="radio"/> Pouco	<input type="radio"/> Normal	<input type="radio"/> Muito
Sheldon Cooper	<input type="radio"/> Pouco	<input type="radio"/> Normal	<input type="radio"/> Muito

[Atualizar Café da Manhã](#)

Figura 28 - Tela de diário escolar - visão familiar

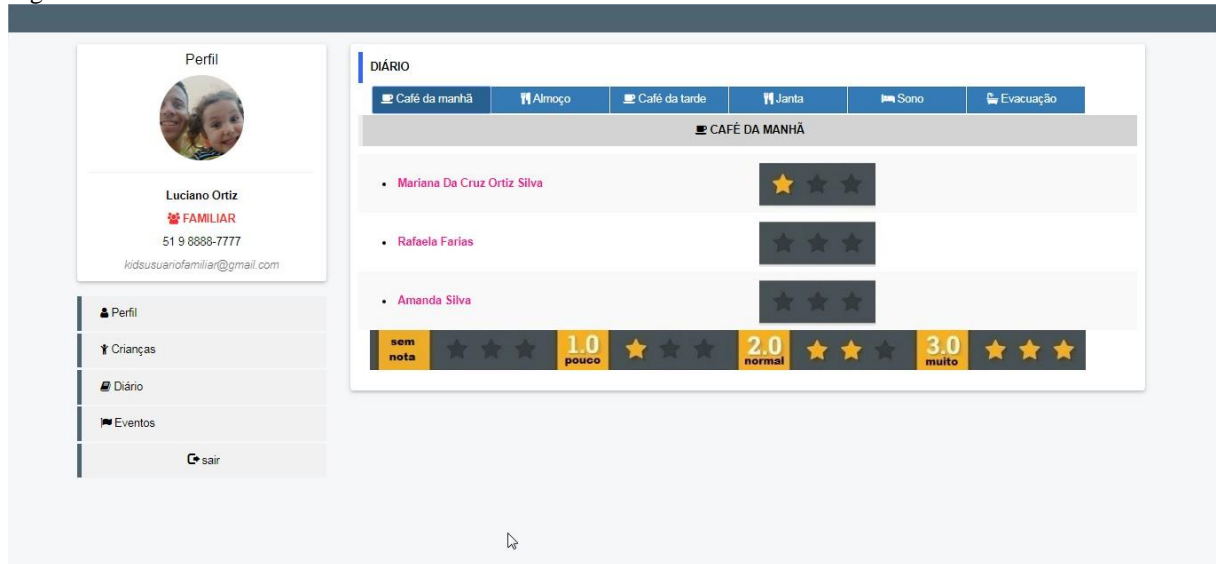


Figura 29 - Tela de avisos - visão creche

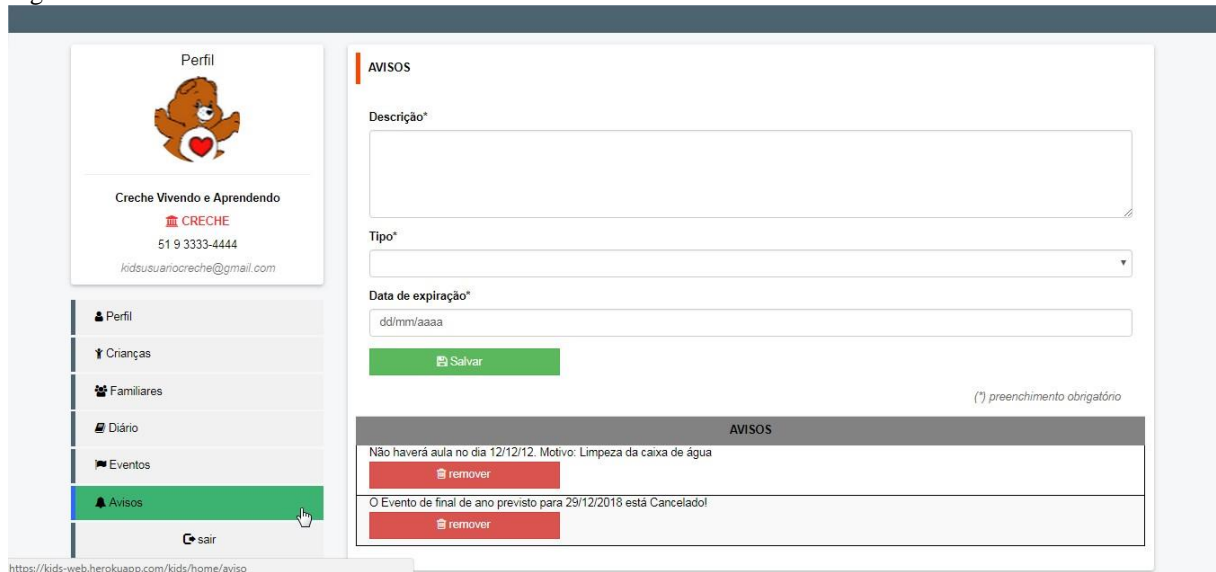
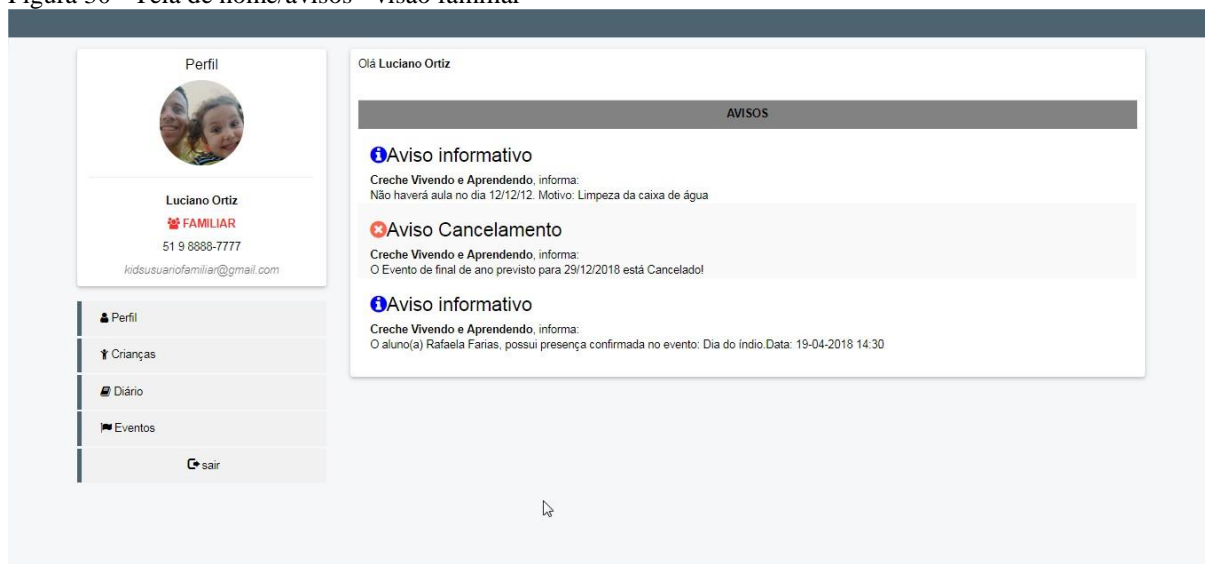


Figura 30 - Tela de home/aviso - visão familiar



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 as funções disponíveis para diferentes tipos de usuário:

- **Usuário Creche:**

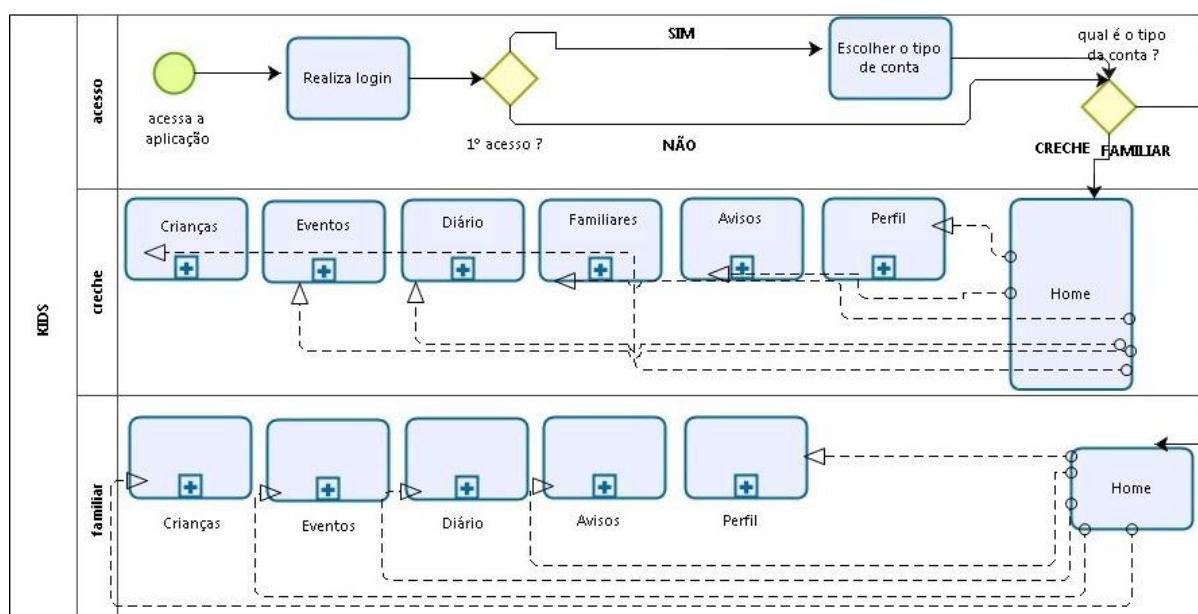
Atualizar perfil; pesquisar crianças; cadastrar/editar informações das crianças; cadastrar o diário escolar das crianças; criar/atualizar eventos; cadastrar/remover avisos; vincular usuários familiares, com crianças.

- **Usuário Familiar:**

Atualizar perfil; visualizar avisos; pesquisar crianças; editar informações das crianças; visualizar o diário escolar das crianças; aceitar ou rejeitar a presença em eventos (convite para a criança).

A seguir será demonstrado o funcionamento do sistema, utilizando um modelo de BPM.

Figura 31- Funcionamento do sistema



9. VALIDAÇÃO

O produto de software deste projeto será validado a partir de Testes Unitários realizados com bibliotecas em Java, tais como Junit e Mockito. Além disso o produto será validado utilizando a ISO/IEC 25000. Essa norma trata de questões de qualidade do produto de software.

Estão sendo feitas as sprints review antes das orientações, juntamente com o P.O do projeto, que acontece no final de cada sprint. Nessas revisões o P.O analisa se aquilo que foi proposto no sprint foi desenvolvido de acordo. Nessa etapa existe a possibilidade de ocorrer inserções de demandas e manutenções no planejamento da sprint posterior.

Esse processo de validação do projeto está verificando se os objetivos do sistema foram alcançados, seus pontos falhos e possíveis melhorias, com o intuito principal de aprimorar a qualidade do software em funcionamento.

9.1 ESTRATÉGIA

A estratégia eleita por este projeto é que permitirá comprovar que os objetivos do mesmo foram alcançados, através de uso pelo próprio cliente (Liziane Ortiz - PO).

10. CRONOGRAMA

Segue abaixo a o cronograma de atividades do trabalho de conclusão de curso para 1º e 2º semestre de 2017.

Figura 32 - Cronograma de TCC 1

TRABALHO DE CONCLUSÃO 1			
Atividade	Descrição	data	status
Pesquisa	Busca por fontes e informações relacionadas ao projeto	18/03/2017	entregue
Planejamento	Product Backlog	15/05/2017	entregue
Tecnologias	Estudos das tecnologias	31/05/2017	entregue
Infra	Configuração de ambiente	15/06/2017	entregue
Sprint 01	User Stories 1,2 e 3	30/06/2017	entregue

Figura 33 - Cronograma de TCC 2

TRABALHO DE CONCLUSÃO 2			
Atividade	Descrição	data	status
Sprint 02	User Stories 4,5,6 e 7	30/07/2017	entregue
Sprint 03	User Story 8	12/08/2017	entregue
Sprint 04	User Stories 9 e 10	01/09/2017	entregue
Sprint 05	User Stories 11, 12, 13 e 14	10/09/2017	entregue
Sprint 06	User Stories 15 e 16	30/09/2017	previsto
Sprint 07	User Stories 17 e 18	15/10/2017	previsto
Sprint 08	User Stories 19 e 20	31/10/2017	previsto
Sprint 09	User Stories 21 e 22	15/11/2017	previsto

Fonte: criado pelo autor do projeto.

REFERÊNCIAS

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

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

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

ISO/IEC 25000, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE. Disponível em: <https://www.iso.org/standard/64764.html>. Acesso em 06 abril. 2016.

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.

SCRUM. Desenvolvimento Ágil. 2014.
Disponível em: <http://www.desenvolvimentoagil.com.br/scrum/>. Acesso em 06 abril. 2017.

GITHUB. Getting your project on GitHub. GITHUB. 2016.
Disponível em: <https://guides.github.com/introduction/getting-your-project-on-github/>. Acesso em 01 abril. 2016.

GIT. Getting your project on GitHub. GIT.
Disponível em: <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>. Acesso em 28 março. 2016.

LIQUIBASE. FAQ. LIQUIBASE. 2016.
Disponível em: <http://www.liquibase.org/faq.html>. Acesso em 29 março. 2016.

APACHE MAVEN PROJECT. What is Maven. MAVEN. 2002-2017.
Disponível em: <https://maven.apache.org/what-is-maven.html>. Acesso em 24 março. 2016.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. About. POSTGRESQL. 1996-2017.
Disponível em: <https://www.postgresql.org/about/>. Acesso em 12 março. 2016.

TRAVIS CI, GmbH. Building a Java project. TREVIS. 2011-2017.
Disponível em: <https://docs.travis-ci.com/user/languages/java>. Acesso em 10 março. 2016.

HIBERNATE, Idiomatic persistence for Java and relational databases. HIBERNATE.
Disponível em: <http://hibernate.org/orm/>. Acesso em 09 abril. 2016.

ANGULAR 2. Disponível em: <http://www.andrefelizardo.com.br/blog/o-que-e-angularjs/>. Acessado em: 22 mar. 2017.

BANCO DE DADOS Disponível em: <http://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>>. Acessado em 02 maio 2017.

BOOTSTRAP. Disponível em: <http://thiagonasc.com/desenvolvimento-web/desenvolvendo-com-bootstrap-3-um-framework-front-end-que-vale-a-pena>. Acessado em 04 junho de 2017.

TYPESCRIPT. Disponível em: <https://www.typescriptlang.org>. Acessado em 04 junho de 2017.

RESTFul. Disponível em: <http://blog.algaworks.com/como-criar-web-services-restful-com-spring-boot/>. Acessado em 04 junho de 2017.

SWAGGER 2 Disponível em: <https://dzone.com/articles/spring-boot-restful-api-documentation-with-swagger>. Acessado em 04 junho de 2017

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

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.

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

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

THE TWELVE FACTOR APP. Disponível em: https://12factor.net/pt_br/. Acessado em 18 junho de 2017.

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

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

SPRING BOOT. Injeção de Dependência. Disponível em: <https://github.com/gabrielfeitosa/ci-spring-boot>. Acessado em 25 junho de 2017.

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

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

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

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

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

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

Heroku Deployment. Realizando deploy de aplicação Angular. Disponível em: <https://docs.travis-ci.com/user/deployment/heroku/>. Acessado em 02 julho 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.

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

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

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

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

FONT AWESOME. Ícones. Disponível em: <http://fontawesome.io/>. Acessado em 16 julho de 2017

ANGULAR 2. Protocolo HTTP. Disponível em: <https://www.youtube.com/watch?v=L7xPwhwbcHE>. Acessado em 16 julho de 2017

ANGULAR 4. HTTP Client. Disponível em: <https://angular.io/guide/http>. Acessado em 16 julho de 2017.

ANGULAR 4. Entendendo RxJS Observable com Angular. Disponível em: <https://medium.com/tableless/entendendo-rxjs-observable-com-angular-6f607a9a6a00>. Acessado em 23 julho de 2017.

ANGULAR 4. HTTP Example with Observables. Disponível em: <https://codecraft.tv/courses/angular/http/http-with-observables/>. Acessado em 23 julho de 2017

ANGULAR 2. Save Navigation Operator In Angular. Disponível em: <http://www.syntaxsuccess.com/viewarticle/elvis-operator-in-angular-2.0>. Acessado em 23 julho de 2017.

ANGULAR 4. Routing. Disponível em: <https://www.youtube.com/watch?v=L6ipgij-AUw>. Acessado em 23 julho de 2017.

ANGULAR 4. Route Configuration. Disponível em: <https://codecraft.tv/courses/angular/routing/route-configuration/>. Acessado em 30 julho de 2017.

ANGULAR 2. Acesso HTTP a um servidor remoto. Disponível em: <http://kazale.com/curso-angular-2-aula-10-acesso-http-um-servidor-remoto/>. Acessado em 30 julho de 2017.

ANGULAR 4. Properly Set Environment Variables. Disponível em: <https://scotch.io/tutorials/properly-set-environment-variables-for-angular-apps-with-gulp-ng-config>. Acessado em 30 julho de 2017.

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

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. Acessado em 06 agosto de 2017.

ANGULAR. Deploy no Heroku. Disponível em: <https://www.youtube.com/watch?v=MUDFLxveBzI>. Acessado em 06 agosto 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

W3C. Templates. Disponível em: https://www.w3schools.com/w3css/tryw3css_templates_social.htm. Acessado em 13 agosto de 2017.

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

TRAVIS. Using SonarCloud with Travis CI. Disponível em: <https://docs.travis-ci.com/user/sonarcloud/>. Acessado em 13 agosto de 2017.

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

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

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

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 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.

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

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

NPM. How npm handles the "scripts" field. Disponível em: <https://docs.npmjs.com/misc/scripts>. Acessado em 27 agosto de 2017.

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.

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.

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. Acessado em 03 setembro de 2017.

DEZIGN. Download. Disponível em: <http://www.datanamic.com/download/download-dezign.html>. Acessado em 03 setembro 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.

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

BIZAGI. Modele processos de negócio. Disponível em: <http://www.techtudo.com.br/tudo-sobre/bizagi-modeler.html>. Acessado em 10 setembro de 2017.

APÊNDICE A

- Dados para acessar o sistema:

O sistema pode ser acessado por qualquer pessoa que possua uma conta google. Porém a nível de testes já há uma base preparada, com dados fakes, na intenção de melhorar a interação.

- **1º passo:** preparar a base de dados:
Chame no browser a seguinte URL: <https://kids-core.herokuapp.com/db/12345>
- **2º passo:** acesse a aplicação:
<https://kids-web.herokuapp.com/kids/login>
- **3º passo:** escolha um tipo de usuário e acesse o sistema:

Usuário creche:

Login: kidsusuariocreche@gmail.com

Senha: !@#kidsusuariocreche

Usuário familiar:

Login: kidsusuariofamiliar@gmail.com

Senha: !@#kidsusuariofamiliar

Obs: Talvez demore alguns minutos no primeiro acesso, mas depois segue normalmente. Motivo: Pelo fato de estar sendo utilizado a versão free do servidor Heroku, acontece algumas vezes de o servidor estar hibernando no momento da requisição.