

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 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	9
Figura 2 - Tela de login	17
Figura 3 - Tela de seleção da conta Google.....	17
Figura 4 - Tela para informar a senha da conta Google	18
Figura 5 - Tela para escolher o tipo de conta	19
Figura 6 - Tela de dados do perfil do usuário.....	19
Figura 7 – Tela de pesquisa de crianças	20
Figura 8 – Tela de cadastro/edição de dados gerais (Passo 01).....	21
Figura 9 – Tela de cadastro/edição de endereço (Passo 02)	21
Figura 10 – Tela de cadastro/edição de contato (Passo 03).....	21
Figura 11 – Tela de cadastro/edição de medicamentos (Passo 04)	22
Figura 12 – Tela de cadastro/edição de alergias (Passo 05)	22
Figura 13 - Tela de vínculo de um familiar para uma criança.....	23
Figura 14 - Corpo do e-mail referente ao vínculo realizado.....	23
Figura 15 - Tela de diários (visão creche)	24
Figura 16 - Tela de diários (visão familiar).....	24
Figura 17 - Tela de cadastro e lista de eventos (visão creche)	25
Figura 18 - Tela de eventos (visão familiar).....	26
Figura 19 - Visão geral do ciclo de vida do Scrum	27
Figura 20 - Visão geral da arquitetura	28
Figura 33 - Modelo de dados	29
Figura 34- Funcionamento do sistema	31

SUMÁRIO

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

4.8	SERVIDOR DE HOSPEDAGEM	15
4.8.1	Heroku.....	15
4.9	BIBLIOTECAS EXTERNAS	15
4.9.1	Api Swagger	15
4.9.2	Api Google Maps	16
5	DESCRIÇÃO DA SOLUÇÃO.....	17
5.1	LOGIN	17
5.2	TIPO DE CONTA.....	18
5.3	PERFIL.....	19
5.4	CRIANÇAS.....	20
5.5	FAMILIARES	22
5.6	DIÁRIO	23
5.7	EVENTOS.....	25
5.8	AVISOS	26
6	ABORDAGEM DE DESENVOLVIMENTO	27
7	ARQUITETURA DO SISTEMA	28
7.1	MODELAGEM FUNCIONAL.....	29
7.1.1	Product Backlog	29
7.1.2	User Stories	29
7.1.3	Sprint Backlog	29
7.2	MODELAGEM DE DADOS.....	29
8	FUNCIONAMENTO DO SISTEMA.....	31
9	VALIDAÇÃO.....	32
9.1	ESTRATÉGIA	32

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 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.

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 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 Alergia

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.

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 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

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

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

A solução que foi desenvolvida nesse projeto visou sanar 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

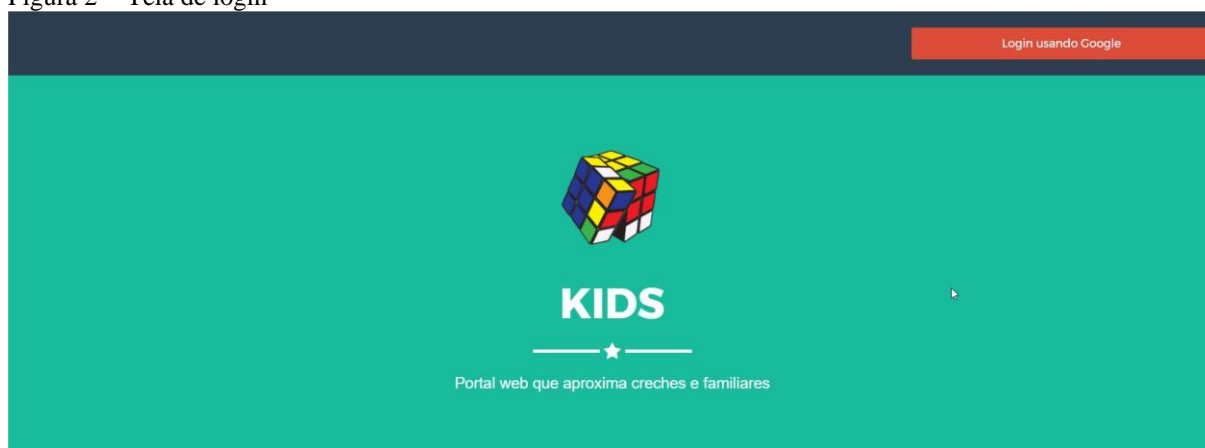


Figura 3 - Tela de seleção da conta Google




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...

Google

Luciano Ortiz

 lucianoortizsilva@gmail.com

Digite sua senha

.....

Outras maneiras de fazer login

PRÓXIMA

Português (Brasil) ▾

[Ajuda](#) [Privacidade](#) [Termos](#)

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.

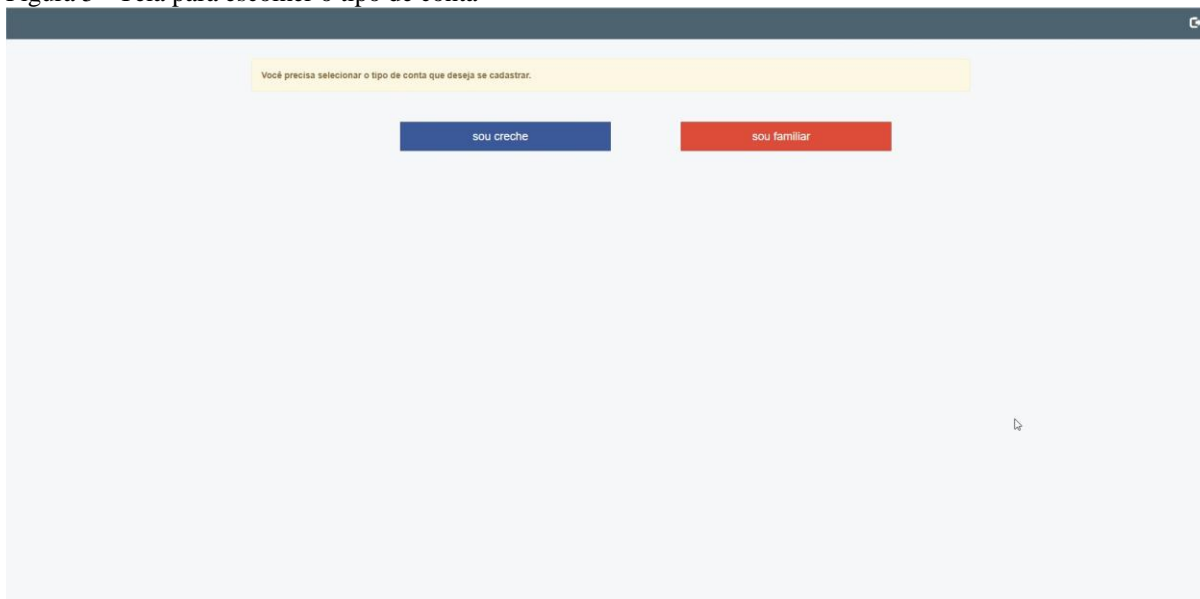
Para isso foi desenvolvido uma interface gráfica que disponibiliza duas opções de criação de conta, um botão para o tipo *familiar* e um outro botão para o tipo *creche*.

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 uma barra superior escura com um ícone de menu. Abaixo, uma mensagem amarela indica: "Você precisa selecionar o tipo de conta que deseja se cadastrar." Duas botões são exibidos: "sou creche" em azul e "sou familiar" em vermelho.

5.3 PERFIL

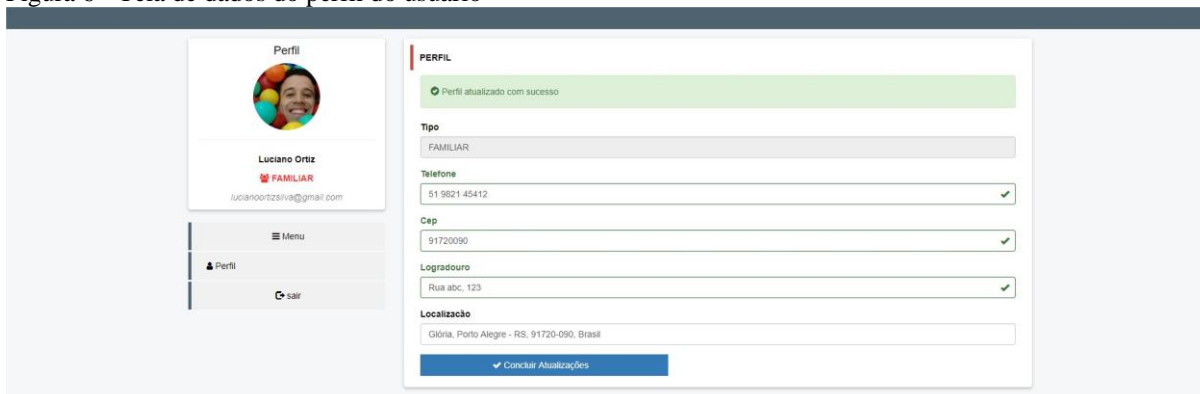
Trata-se de uma área onde o usuário poderá manter seus dados principais de cadastro devidamente atualizados, afim de serem utilizados em diversos pontos do portal kids.

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 mostra o perfil do usuário "Luciano Ortiz" com o e-mail "lucianoortizsiva@gmail.com". O menu lateral contém "Menu", "Perfil" e "sair". A seção "PERFIL" exibe uma mensagem de sucesso: "Perfil atualizado com sucesso". Os campos de formulário são: "Tipo" (FAMILIAR), "Telefone" (51 9821 45412), "Cep" (91720090), "Logradouro" (Rua abc, 123) e "Localização" (Glória, Porto Alegre - RS, 91720-090, Brasil). Um botão azul "Concluir Atualizações" está na base.

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

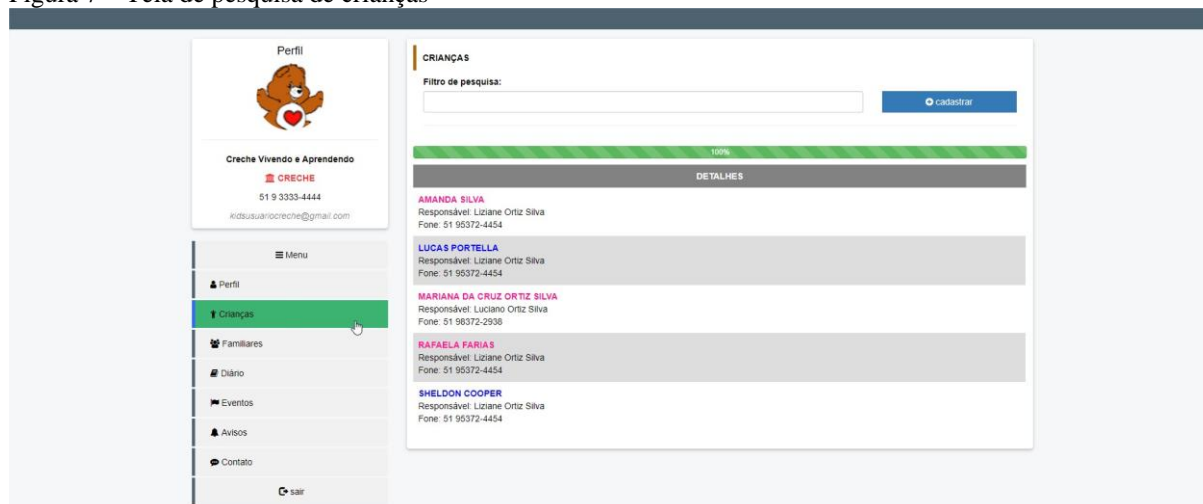


Figura 8 – Tela de cadastro/edição de dados gerais (Passo 01)

Perfil

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

Menu

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

CRIANÇAS

1º DADOS GERAIS

Matricula*
0001

Nome*
Mariana Da Cruz Ortiz Silva

Data de Nascimento*
29/08/2011

Sexo*
FEMININO

Voltar Próximo

(*) preenchimento obrigatório

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

Perfil

Creche Vivendo e Aprendendo
CRECHE
51 9 3333-4444
kidsusuariocreche@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

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

Perfil

Creche Vivendo e Aprendendo
CRECHE
51 9 3333-4444
kidsusuariocreche@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 96372-2938

Fone Outros
51 3232-2322

Anterior Próximo

(*) preenchimento obrigatório

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

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

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 possua 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

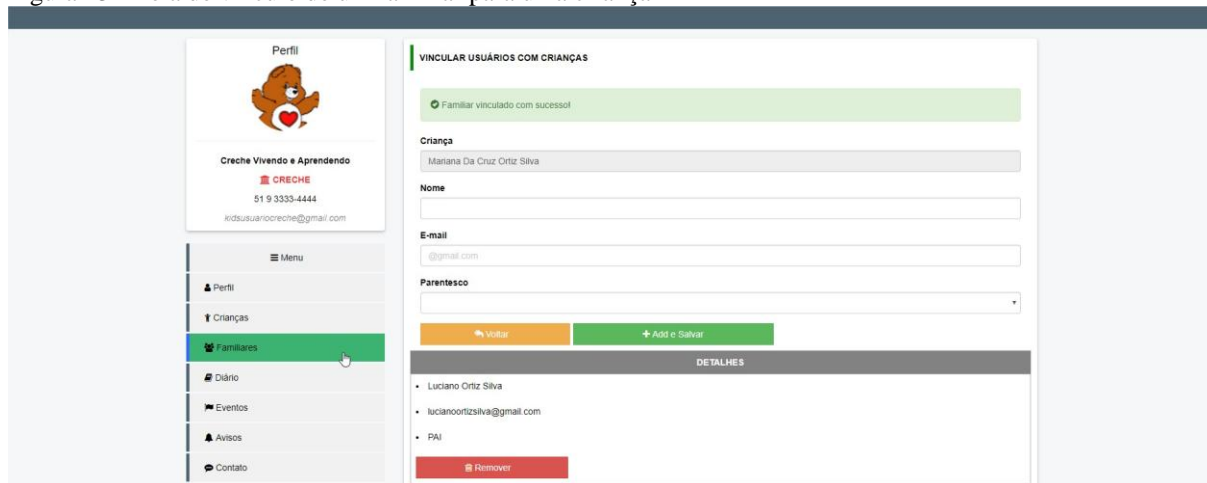


Figura 14 - Corpo do e-mail referente ao vínculo realizado



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 que correspondentes a esses diários:

Figura 15 - Tela de diários (visão creche)

Figura 16 - Tela de diários (visão familiar)

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;

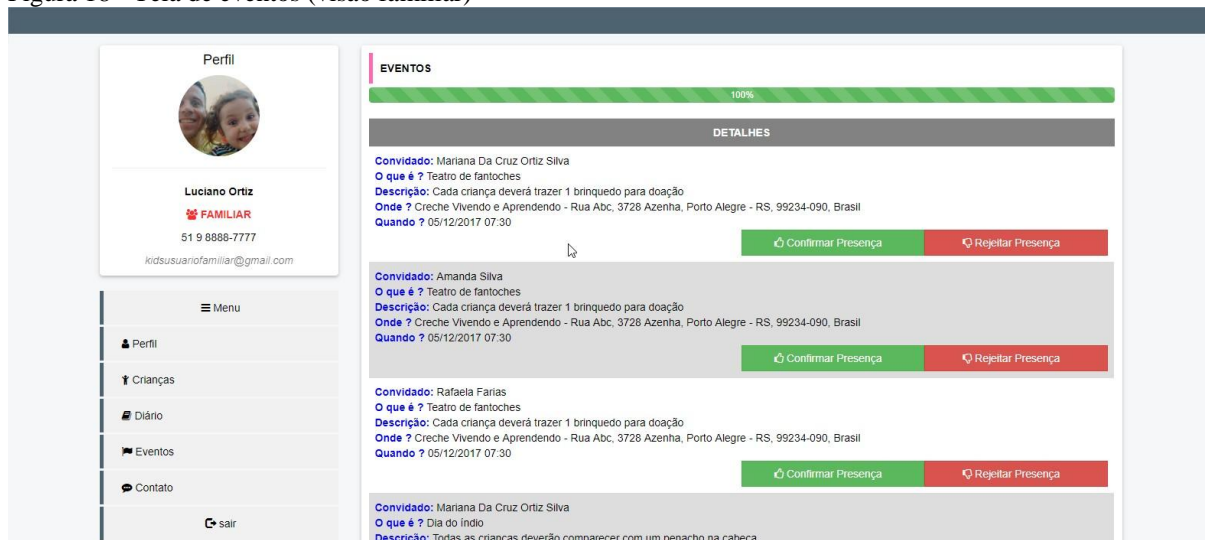
Para atender esses requisitos, foi desenvolvida uma interface gráfica para o usuário do perfil creche cadastrar seus eventos, e outra interface gráfica para que o perfil do usuário familiar possa receber esses convites, e aceitar ou rejeitar a presença nos mesmos.

A seguir são mostradas as telas que correspondentes a esses eventos:

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

The screenshot displays the 'Cadastro e lista de eventos' interface for a 'creche' user. The interface is divided into a left sidebar and a main content area. The sidebar features a profile card for 'Creche Vivendo e Aprendendo' and a menu with options like Perfil, Crianças, Familiares, Diário, Eventos (highlighted), Avisos, and Contato. The main content area contains a form for creating an event with fields for 'Nome*', 'Data de Realização*' (with a date picker), and 'Descrição*'. Below the form is a green progress bar at 100% and a 'Salvar' button. A section titled 'DETALHES' shows a list of events: 'Evento: Dia do vampiro' with date '29/12/2018 21:00' and status 'Previsto', and 'Evento: Dia das bruxas' with date '24/08/2018 14:00'. There are radio buttons for 'Previsto', 'Realizado', and 'Cancelado', and an 'Atualizar' button.

Figura 18 - Tela de eventos (visão familiar)



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;

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 19 - 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.

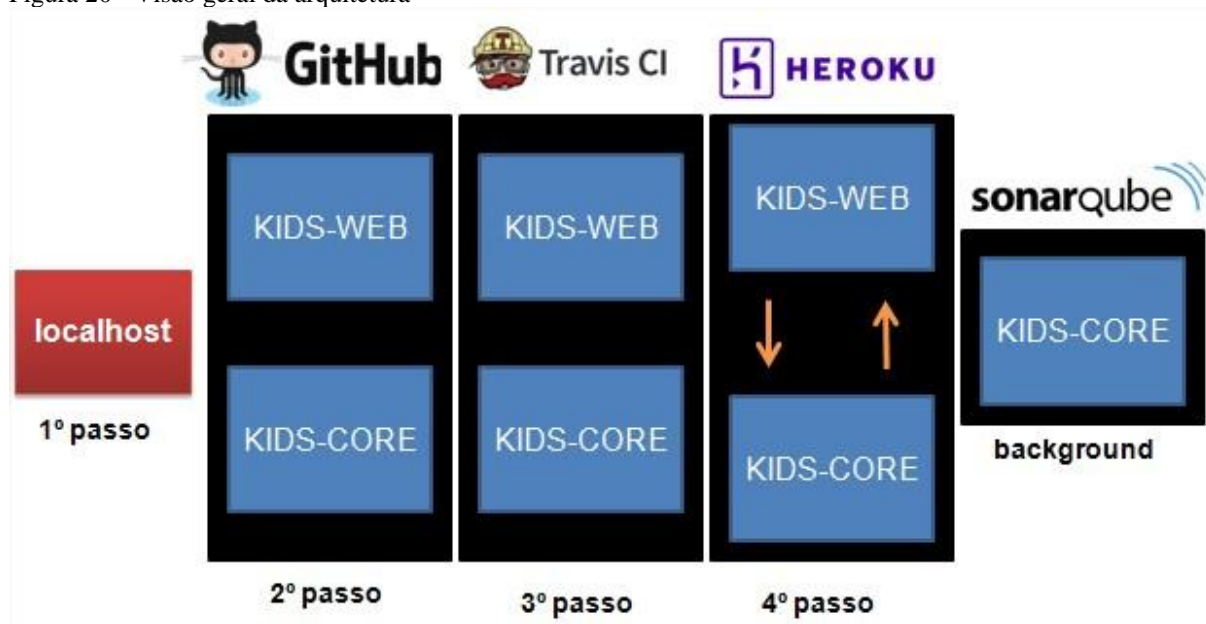
No dia a dia do desenvolvimento, o projeto seguiu o seguinte processo:

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 realizada 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 20 - Visão geral da arquitetura



7.1 MODELAGEM FUNCIONAL

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

7.1.1 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.

7.1.2 User Stories

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

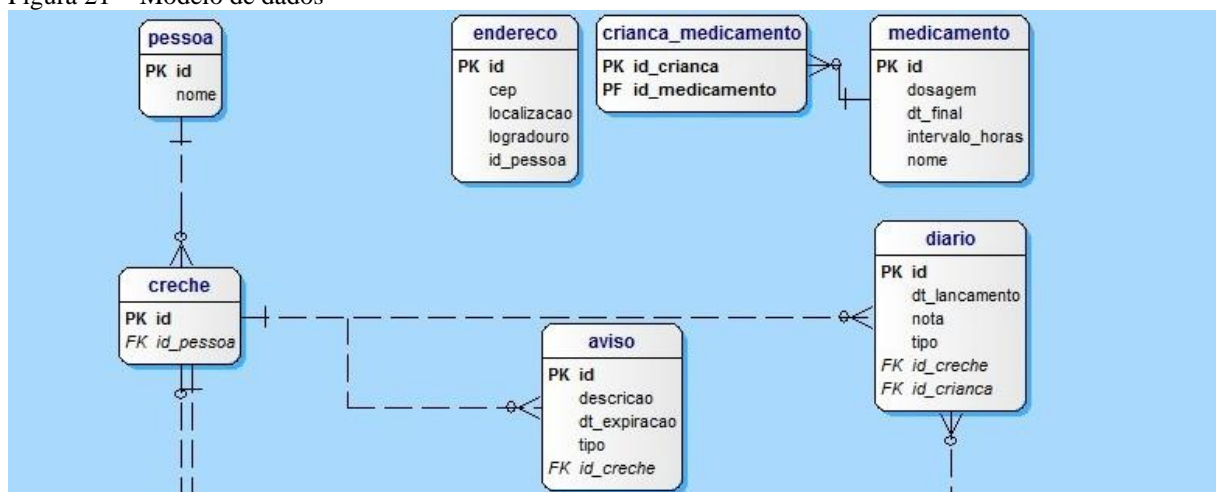
7.1.3 Sprint Backlog

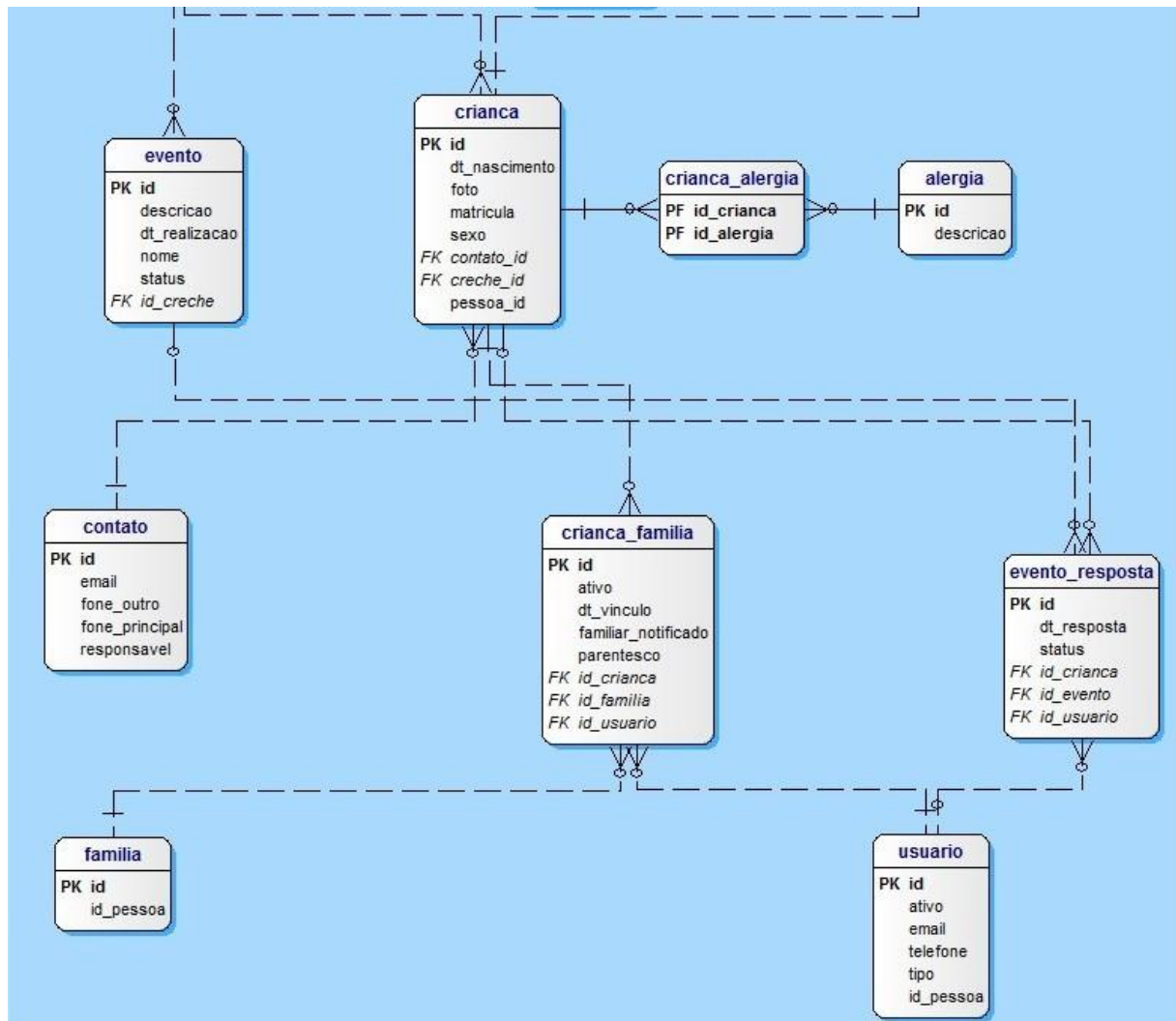
Após o Product Backlog ter sido detalhado através de diversas User Stories, foram planejados inicialmente 10 sprints, contendo em cada sprint um backlog de users Stories. Abaixo segue detalhes sobre as sprints já realizadas, e sobre as próximas sprints planejadas.

7.2 MODELAGEM DE DADOS

A seguir será exibido o modelo de dados da aplicação.

Figura 21 - Modelo de dados





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 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:**

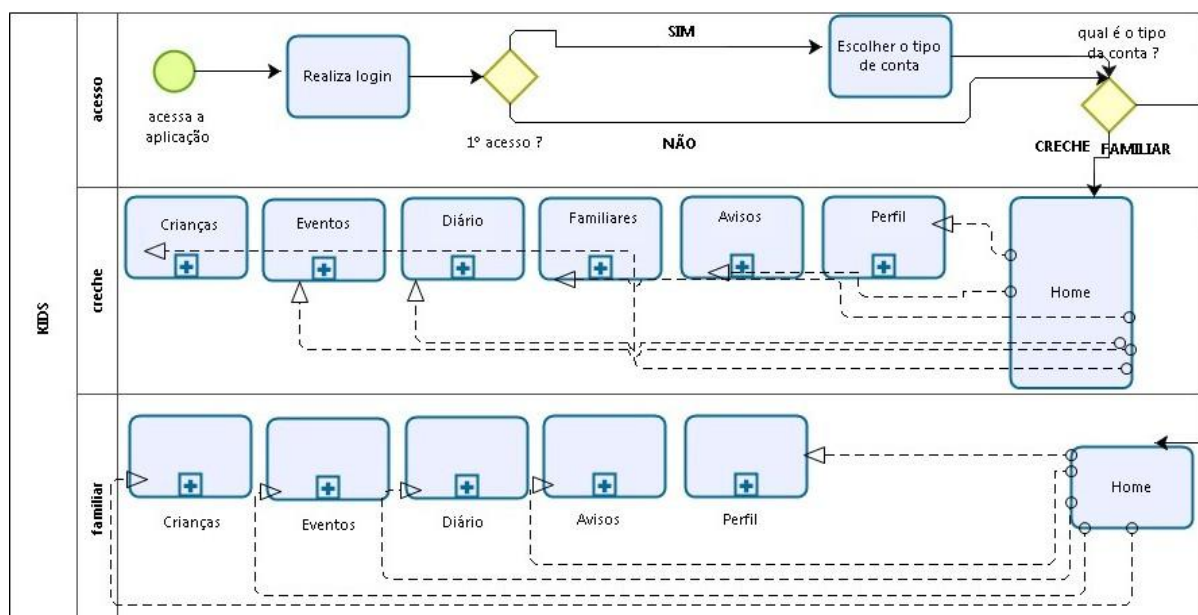
Atualizar perfil; pesquisar crianças; cadastrar/editar informações das crianças; cadastrar o diário escolar das crianças; criar/atualizar eventos; cadastrar/remover avisos; vincular usuários familiares, com crianças.

- **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 22- 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).

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. 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.

BIZAGI. Modele processos de negócio.

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

BOOTSTRAP. Desenvolvendo com Bootstrap 3.

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.

DEZIGN. Download do Dezipn.

Disponível em: <http://www.datanamic.com/download/download-dezipn.html>. Acessado em 03 setembro de 2017.

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 your project on GitHub. GIT.

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

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.

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

GOOGLE. Adicionando login na aplicação.

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

HEROKU. Deploy Angular 2 CLI app to Heroku.

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

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.

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.

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.

LIQUIBASE. FAQ. LIQUIBASE. 2016.

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

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/ch09lvl1sec77/connecting-to-heroku-postgres-from-psql. Acessado em 03 setembro de 2017.

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.

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.

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/gabrielfeitosa/ci-spring-boot>. Acessado em 25 junho 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 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.

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

POSTGRESQL. THE POSTGRESQL GLOBAL DEVELOPMENT GROUP.1996-2017.

Disponível em: <https://www.postgresql.org/about/>. Acesso em 12 março. 2016.

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. 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.

TYPESCRIPT. Why TypeScript ?

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

THE TWELVE FACTOR APP. Os doze fatores.

Disponível em: https://12factor.net/pt_br/. Acessado em 18 junho de 2017.

W3C. Template Rede Social.

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