

## **EASY LIFE TOOLS: Leitor RFID/NFC em Arduino para Aplicação Mobile em um Ambiente Escolar**

André Gustavo Fernandes Junior  
Redes de Computadores – Fatec Bauru  
andre.fernandes19@fatec.sp.gov.br

Leonardo Dias Leal  
Redes de Computadores – Fatec Bauru  
leonardo.leal01@fatec.sp.gov.br

Renan Gabriel Correia Himeno  
Redes de Computadores – Fatec Bauru  
renan.himeno@fatec.sp.gov.br

Anderson Francisco Talon  
Redes de Computadores – Fatec Bauru  
anderson.talon@fatec.sp.gov.br

### **RESUMO**

O presente artigo tem como objetivo o estudo das tecnologias RFID e NFC de forma a utilizar essas tecnologias em conjunto com o mobile para conseguir demonstrar sua versatilidade e aplicar esse tipo de recurso na solução de um problema para facilitar as interações humanas. A aplicação tem foco em plataformas mobile, sendo um software para funcionar como ponto eletrônico para os estudantes, o app foi testado a fim de verificar a integração de tecnologias novas no mercado e que possuem um grande potencial de crescimento especialmente na automação e na parte de redes. Além da parte do desenvolvimento também houve todo suporte de comunicação com o servidor em nuvem, abordando toda a configuração necessária para operar. Entre as técnicas/tecnologias utilizadas, podemos destacar Arduino, RFID, NFC e SCRUM. Com a versão beta pronta foi possível realizar testes na aplicação para avaliar melhor o conjunto de tecnologias empregadas.

**PALAVRAS-CHAVE:** Aplicativo Mobile, Computação em nuvem, RFID, Banco de Dados.

### **ABSTRACT**

This article aims to study RFID and NFC technologies in order to use these technologies in conjunction with mobile to be able to demonstrate their versatility and apply this type of resource in solving a problem to facilitate human interactions. The application is focused on mobile platforms, being a software to function as an electronic point for students, the app was tested in order to verify the integration of new technologies in the market and which have a great potential for growth, especially in automation and part of networks. In addition to the development part, there was also all communication support with the cloud server, covering all the necessary configuration to operate. Among the techniques / technologies used, we can highlight Arduino, RFID, NFC and SCRUM. With the beta version ready, it was possible to carry out tests in the application to better evaluate the set of technologies employed.

**KEYWORDS:** Mobile Application, Cloud Computing, RFID, Database.

## 1 INTRODUÇÃO

Hoje o Brasil enfrenta diversos problemas em diferentes áreas, como saúde, educação, economia, além de não ter medidas sustentáveis de forma satisfatória, pois ainda se gasta muito papel e tinta, que além de ocupar espaço físico ainda torna as coisas mais burocráticas.

Considerando que a tendência é deixar tudo universal e levando em consideração que há em média dois celulares por pessoa, é correto afirmar que cada vez mais pessoas utilizam dispositivos móveis no lugar do desktop. (AGNI, 2015).

O Easy Life Tools é um projeto com objetivo de solucionar problemas pertinentes e facilitar processos.

Em seu aspecto geral o Easy Life Tools é um aplicativo mobile, dividido em módulos (partes), sendo cada um deles destinado a solucionar um problema de uma área diferente, sendo essas partes separadas por saúde, educação, economia, achados e perdidos, agronomia, meio ambiente, e as pessoas também poderão desenvolver módulos extras que serão verificados e disponibilizados de graça.

Neste artigo será explorado exclusivamente o módulo de educação, sendo abordados todos os processos desde a parte de planejamento até a implantação.

Para ajudar nesse processo serão utilizadas tecnologias baseadas em RFID e NFC, que utilizam comunicação sem fio, pela proximidade necessária se torna um método muito difícil de fraudar (ALECRIM, 2012)

No módulo de Educação as chamadas serão feitas pelo próprio aplicativo através do recurso de NFC do celular ou cartão RFID para quem não possui aparelho telefônico com recurso NFC disponível, além de apresentar na sua interface dados como, nome do professor, matéria, horário de aula e faltas em tempo real, caso o aluno não chegue no prazo estipulado a falta já será computada, e para retirá-la o professor deve fazer um relatório sobre motivo do atraso, isso vale também para o professor que se atrasar. Um e-mail é enviado para o aluno confirmando sua presença na aula. E este módulo também visa automatizar o processo de retirada de livros da biblioteca

O objetivo da pesquisa é compreender o processo de desenvolvimento mobile, descobrir as possibilidades por trás da tecnologia NFC, entendendo como funciona a transmissão de dados e como usar esses recursos em conjunto com uma aplicação mobile, além de utilizar conceitos de redes de computadores que dará suporte ao aplicativo abordando a comunicação com o servidor, banco de dados, processamento e armazenamento em nuvem, além de segurança na transmissão dos dados.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Sistema *Radio Frequency Identification* (RFID)

Os sistemas RFID são muito ligados a cartões inteligentes, o procedimento técnico foi retirado da engenharia de rádio e radar, devido suas inúmeras vantagens essa tecnologia está sendo cada vez mais usada, além de estar conquistando novos mercados de massa, por exemplo, existem várias empresas de transporte público usando cartões como bilhete de passagem, tendo uso também na questão do pedágio no modelo sem parar (FINKENZELLER,2010).

Um sistema de RFID é feito com uma etiqueta eletrônica que serve tanto para transmitir quanto para receber o sinal, os dados de identificação ficam na etiqueta e são transferidos por radiofrequência (DUARTE; INIGUEZ; DESSENS, 2020).

O RFID é uma tecnologia muito utilizada, pois possui aplicabilidade em diversas áreas sendo usada para localização de pessoas e equipamentos, controle de acesso e histórico automatizado de informações como data e horário (NASSAR; VIEIRA, 2019).

As aplicações do RFID determinam que classe será usada, a classe ativa inclui uma fonte de energia seja por baterias ou fonte de alimentação, e a classe passiva não precisa de fonte de alimentação pois a energia será recebida de forma sem fio (OLIVEIRA, 2017).

A Figura 1 apresenta um modelo de composição básica de um sistema com RFID, sendo um emissor e o outro o receptor, o dispositivo de leitura estará conectado a um controlador que enviará os dados para serem processados em um computador, que neste projeto é um servidor em nuvem.

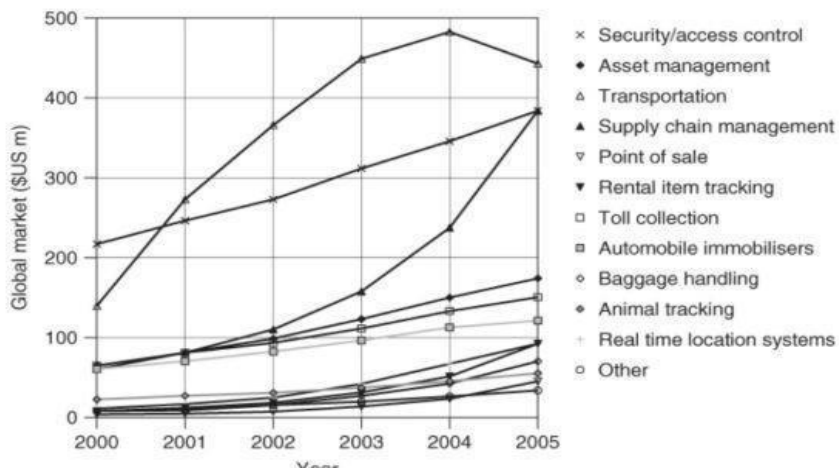
Figura 1 - Composição básica de um sistema RFID



Fonte: Oliveira, Pereira (2019).

A Figura 2 apresenta o crescimento do mercado global de sistemas RFID, sendo representado o uso para cada área.

Figura 2 - O crescimento estimado do mercado global de sistemas RFID entre 2000 e 2005 em milhões de US\$



Fonte: Krebs & Finkenzeller (2010).

O RFID pode operar em diferentes frequências, podendo ir de 135 kHz de onda longa até 5,8 GHz na faixa de micro-ondas, o alcance do sistema é pequeno, estando na média de 1 centímetro, mas pode chegar até no máximo 15 metros (FINKENZELLER, 2010).

## **2.2 Near Field Communication (NFC)**

É uma tecnologia que permite a troca de informações sem cabos, utilizando-se da aproximação física, é uma variante do RFID, mas tem campo de reconhecimento limitado, seu alcance máximo é de 10 centímetros, devido a aproximação necessária, essa comunicação se torna muito mais segura, possui várias formas de uso, em sua maioria ligadas a dispositivos móveis, com o NFC é possível fazer transações bancárias, recarregar o celular, transmitir dados entre dispositivos e até mesmo passar por catracas de ônibus e metrô apenas aproximando o celular (BRITO, 2012).

## **2.3 Arduino**

O Arduino é um microcontrolador que pode ser conectado ao computador via IDE usando um cabo usb, utiliza linguagem baseada em C/C++, pode ser usado para fazer robôs, controladores MIDI, videogames, dá para utilizar em conjunto com sensores (EVANS; NOBLE; HOCHENBAUM, 2013).

## **2.4 Cloud Computing**

A computação em nuvem é um modelo de computação que vem crescendo com o passar dos anos, permite ao usuário acessar serviços de qualquer lugar utilizando a internet (PEDROSA; NOGUEIRA, 2011).

O termo *Cloud Computing* surgiu em uma palestra da Google em 2006, sobre como a empresa gerenciava seus data centers, surgindo na necessidade de construir uma infraestrutura complexa, onde usuários não precisassem instalar, configurar e atualizar os softwares, sendo uma maneira inteligente de maximizar e flexibilizar os recursos computacionais (RUSCHEL; ZANOTTO; MOTA, 2010).

O conceito de nuvem pode ser representado pela internet, uma infraestrutura formada por hardwares, softwares, interfaces, redes de comunicação, dispositivos de controle e armazenamento (PEDROSA; NOGUEIRA, 2011).

As características essenciais dessa arquitetura segundo Sousa *et al* (2011) são:

- a) *Self-service* sob demanda: O usuário utilizar o recurso quando, como e onde quiser, sem precisar interagir com terceiros para acessar o serviço;
- b) Amplo acesso: Recursos são disponibilizados por meio da rede e acessados por qualquer dispositivo devidamente conectado;
- c) *Pooling* de recursos: Os recursos computacionais do provedor são organizados em um *pool* para servir múltiplos usuários;
- d) Elasticidade rápida: Recursos podem ser adquiridos de forma rápida e elástica, podendo ser esse processo automático, caso haja a necessidade de escalar com o aumento da demanda;
- e) Serviço medido: Sistemas em nuvem automaticamente controlam e otimizam o uso de recursos por meio de uma capacidade de medição.

## 2.5 Scrum

O Scrum surgiu em 1990, mas ficou mais conhecido na década seguinte, pode ser implementado em uma grande variedade de projetos tendo ganhado destaque em projetos de marketing e na área de TI, os benefícios do Scrum são:

- a) Entregas frequentes de retorno aos investimentos do cliente;
- b) Redução dos riscos na realização do projeto;
- c) Produto com melhor qualidade;
- d) Redução do desperdício;
- e) Maior produtividade.

Sendo assim o Scrum permite realizar projetos em menos tempo e com maior certeza de satisfazer o cliente, tornando o projeto mais aberto a alterações se necessárias (SABBAGH, 2014). No Scrum as etapas ficam mais claras além de se conseguir entregar algo funcional a cada ciclo (COHN, 2015).

## 3 MATERIAIS E MÉTODOS

Cada “parte” do aplicativo funcionará de forma isolada, assim o usuário pode escolher o que melhor atende a sua necessidade. O cadastro de usuário será único e servirá para todos os módulos.

Com o objetivo de utilizar o recurso NFC com o aplicativo, e compreender as etapas do desenvolvimento de um projeto real, no primeiro momento se fez necessário realizar um planejamento, para saber como seria o decorrer do projeto. Foi utilizado o Scrum para conseguir manter um ritmo de trabalho. Para realizar esse planejamento e implementar a metodologia ágil, foi usada a plataforma Trello, pois seu uso é gratuito e ele gera alertas nos e-mails registrados caso exista algum tipo de atraso.

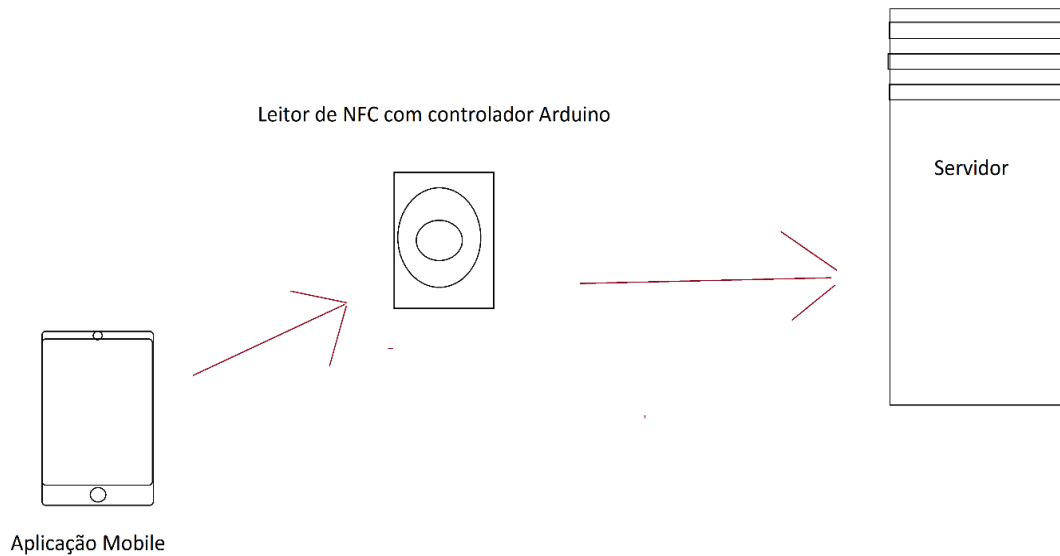
A ideia é utilizar o celular do usuário como se fosse uma tag RFID, com o mesmo princípio de pontos eletrônicos, o usuário devidamente registrado no app terá sua presença na aula confirmada por meio da aproximação com o sensor feito em Arduino. Os dados serão enviados pela internet até o servidor em nuvem onde serão processados, e guardados em um banco de dados MySQL, para criar o banco foi feita uma API em Laravel utilizando a linguagem PHP, nessa API além de ajudar na criação das tabelas, também é possível fazer alterações complexas mais facilmente, sendo possível criar dados de teste para realização dos testes.

Para dar suporte a tudo isso foi necessário um servidor que vai hospedar o banco de dados, prover serviço de e-mail que servirá para enviar notificações de presença via e-mail e processar os dados que serão enviados pelo Arduino, o servidor funcionará em nuvem, serão feitas configurações de segurança para impedir que de alguma forma os dados sejam danificados.

A Figura 3 representa o fluxo de transmissão de informações, com o celular enviando o sinal de requisição que é enviado ao Arduino que compara as informações existentes no servidor e traz o retorno com as informações.

O Arduino escolhido foi do tipo Uno e nele foi instalado um módulo NFC/RFID, o Arduino envia todos os dados pela rede até o servidor em nuvem.

Figura 3 – Etapas do fluxo de dados do projeto.



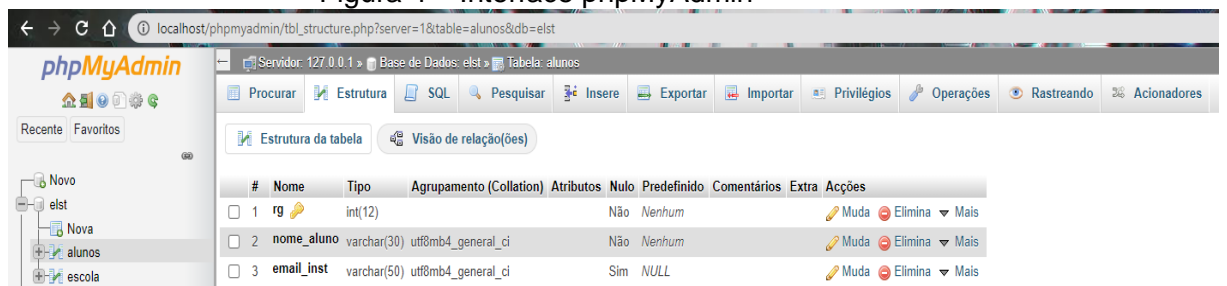
Fonte: Elaborada pelos autores (2020).

### 3.1 Desenvolvimento Mobile

Esta primeira etapa diz respeito a criação do aplicativo, sendo necessário criar a telas, campos e variáveis que irão receber os dados inicialmente. Para desenvolver o app foi escolhida uma tecnologia nova, mas que tem estado cada vez mais presente no mercado que é o React-Native, com isso é possível criar qualquer aplicação utilizando JavaScript, outro ponto forte é que os códigos são convertidos para elementos nativos de cada sistema, podendo ser compilado para Android e iOS sem ser necessário fazer alterações. Ainda nesta etapa foi feito o banco de dados e todos os relacionamentos necessários. O Web Service foi criado utilizando a linguagem PHP como base e serve para fazer a comunicação entre os diferentes dispositivos.

A Figura 4 apresenta a Interface utilizada para criação do banco de dados MySQL.

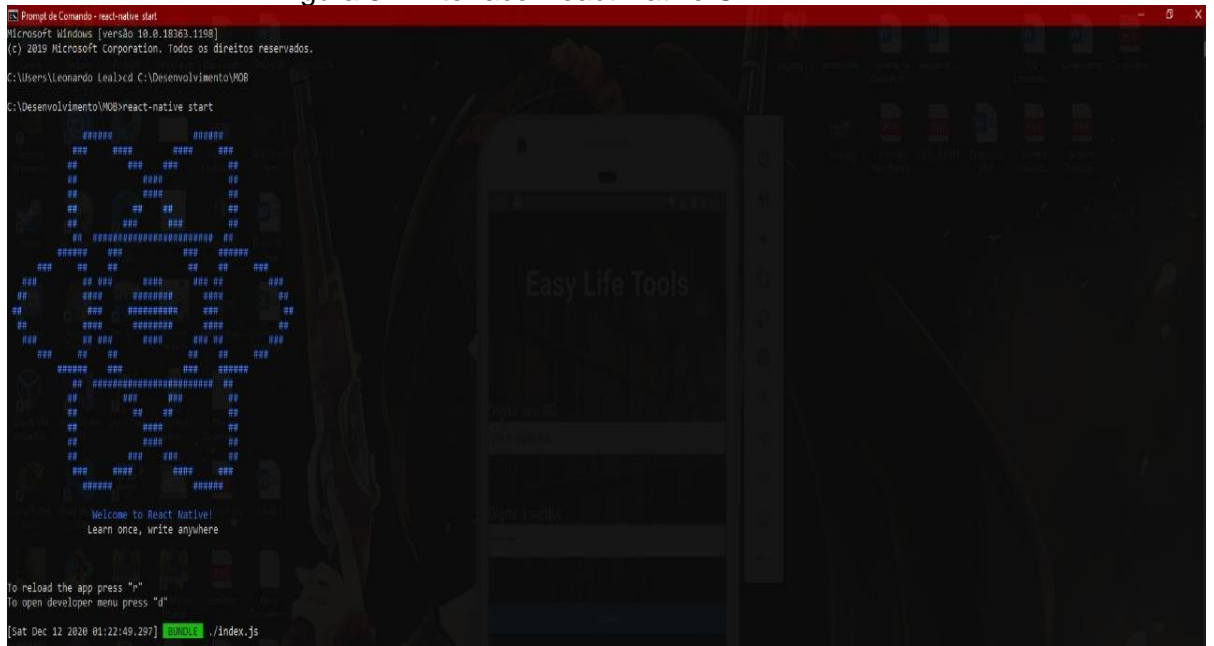
Figura 4 – Interface phpMyAdmin



Fonte: Elaborada pelos autores (2020)

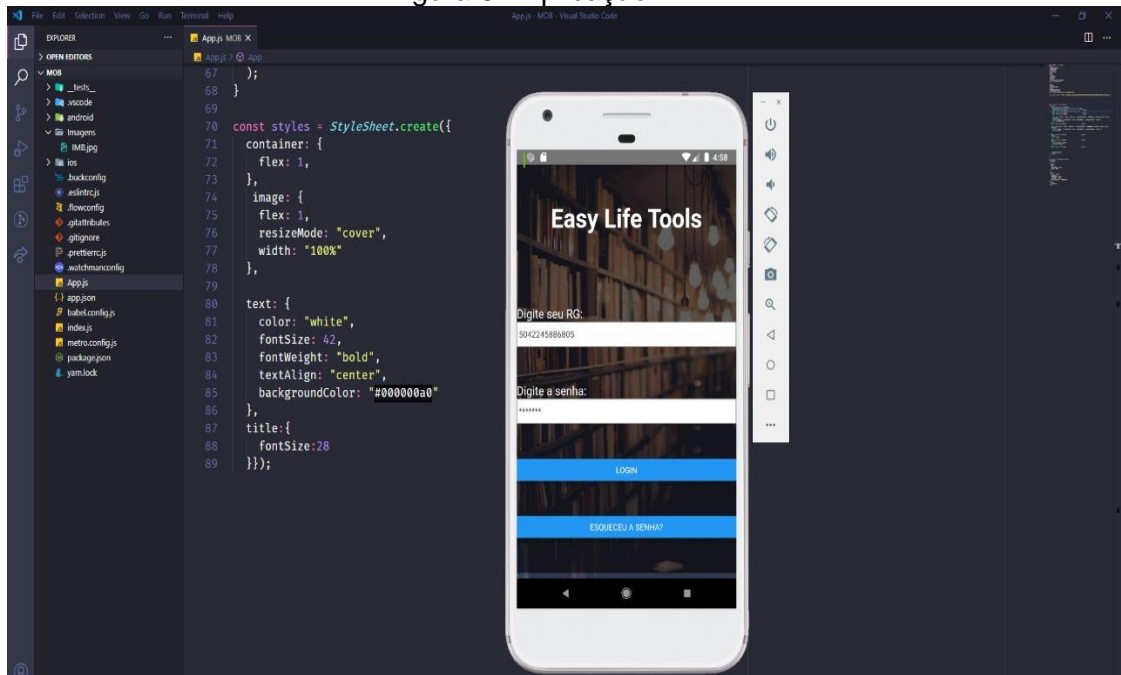
A Figura 5 apresenta a Interface React-Native CLI, na qual foi utilizada para desenvolver o aplicativo como mostra a Aigura 6.

Figura 5 – Interface React Native CLI



Fonte: Elaborada pelos autores (2020)

Figura 6 - Aplicação

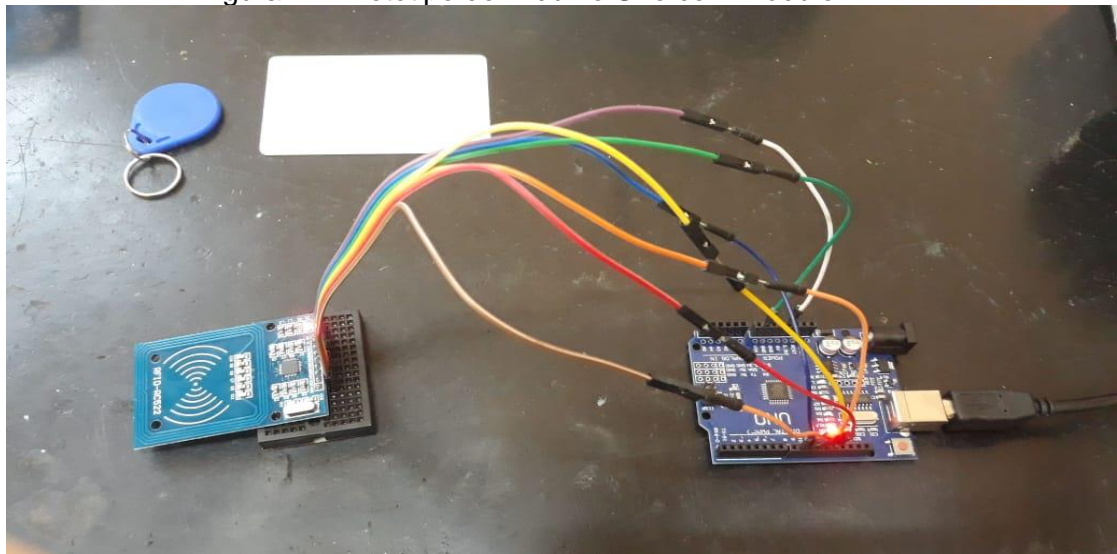


Fonte: Elaborada pelos autores (2020)

### 3.2 Configuração dos dispositivos físicos

Já com Arduino Uno R3 a linguagem para configurá-lo foi em C, com a pinagem feita, foi escolhida a frequência de operação do dispositivo e configurado seu acesso à rede. O Arduino escolhido foi o Arduino Uno R3 por ter um preço aceitável, os módulos NFC/RFID foram acrescentados e configurados junto ao Arduino como mostrado na Figura 7.

Figura 7 – Protótipo do Arduino Uno com módulo



Fonte: Elaborada pelos autores (2020)

### 3.3 Suporte de Comunicação

Nessa parte a conexão NFC do celular é alinhada com a do leitor, o servidor e seus serviços são configurados. Para o serviço em nuvem foi escolhido o pacote grátis da Azure, todos os serviços irão rodar em nuvem como indica a Figura 8, não sendo necessário manter um servidor físico, todas as configurações extras no Web Service foram feitas e tudo foi devidamente direcionado.

Figura 8 – Serviços do Azure

Serviços do Azure

- [Criar um recurso](#)
- [Máquinas virtuais](#)
- [Assinaturas](#)
- [Serviços de Aplicativos](#)
- [Contas de armazename...](#)
- [Bancos de dados SQL](#)
- [Servidores do Banco de...](#)
- [Azure Cosmos DB](#)
- [Serviços do Kubernetes](#)
- [Mais serviços](#)

Recursos recentes

| Nome                               | Tipo              | Última visualização |
|------------------------------------|-------------------|---------------------|
| <a href="#">SERVIDOR-ELT</a>       | Máquina virtual   | um dia atrás        |
| <a href="#">SERVIDOR-ELT_group</a> | Grupo de recursos | 3 semanas atrás     |

Navegar

- [Assinaturas](#)
- [Grupos de recursos](#)
- [Todos os recursos](#)
- [Painel](#)

Ferramentas

- [Microsoft Learn](#)  
Saiba mais sobre o Azure com o treinamento online gratuito da Microsoft
- [Azure Monitor](#)  
Monitorar seus aplicativos e sua infraestrutura
- [Central de Segurança](#)  
Proteja seus aplicativos e sua infraestrutura
- [Gerenciamento de Custos](#)  
Analisar e otimizar gratuitamente seus gastos com a nuvem

Fonte: Elaborada pelos autores (2020)

### 3.4 Testes

Os testes de software estabelecem os engenheiros de sistema pensar de maneira diferente do que estão frequentemente acostumados, eles precisam elaborar casos de testes para descobrir defeitos no software (PRESSMAN, 2006).



Na fase dos testes foram feitos vários testes sendo eles:

- a) Caixa Branca – Neste teste o aplicativo foi avaliado junto com o código fonte para avaliar problemas visíveis na parte da programação.
- b) Caixa Preta – O teste foi feito sem olhar o código fonte e serviu para avaliar o projeto com a visão que um usuário iria ter.
- c) Vulnerabilidades – Foram analisados riscos à segurança em cada um dos pontos principais, sendo o app, banco de dados e servidor.
- d) Teste de conexão – Foi avaliada a conectividade, se está havendo conectividade em locais diferentes.
- e) Análise de requisitos – Foi avaliado se o projeto está atendendo os requisitos pela qual foi proposto

Cada teste tem plano de duração para uma semana, foram testadas cada parte do projeto e assim gerar dados estatísticos sobre seu funcionamento.

Na Figura 9 foi utilizado o recurso *faker* na API em Laravel, que é utilizado para criar dados de testes fictícios para testar o banco de dados.

Figura 9 – Recurso Faker

```
$factory->define(App\User::class, function (Faker $faker) {
    return [
        'nome' => $faker->nome,
        'email' => $faker->unique()->Emailinst,
        'email_verified_at' => now(),
        'password' => '$1234',
        'remember_token' => Str::random(10),
    ];
});
```

Fonte: Elaborada pelos autores (2020)

### 3.5 Descrição do processo

Na primeira etapa foi preparado o ambiente de desenvolvimento para Android pois o método de compilação para iOS é mais complexo, para facilitar o desenvolvimento foi utilizado também o GIT para fazer os envios a um repositório gratuito na internet. Para os testes foi criado um emulador de Android no Android Studio e o JDK exportado, utilizando o React-Native CLI foi dado início ao desenvolvimento, o banco de dados foi criado com uma API em Laravel usando o Artisan, foram criadas as tabelas e estabelecidos os relacionamentos. Por conta da quantidade de detalhes foi possível montar um beta do aplicativo.

O servidor em nuvem utilizou o Windows Server 2016, serviços para email, banco de dados e *active directory* foram instalados. Para o servidor em nuvem foi usado o Azure na forma grátis.

## 4 APRESENTAÇÃO DOS RESULTADOS

No desenvolvimento foram testados o React-Native CLI e o Expo que é um facilitador, em meio ao desenvolvimento foi visto que o Expo era melhor para testar pois suas bibliotecas já tinham funções prontas e a geração e exportação do APK

também eram facilitados não sendo necessário fazer alguns processos de implementação, porém, ao verificar, não foi encontrada uma biblioteca compatível com o uso de NFC e outras funções nativas, o que impossibilitou o uso da ferramenta para o desenvolvimento, já o CLI por utilizar totalmente os recursos nativos permitiu o uso do NFC para testar. No início foi preciso emular com um SDK do Android Studio, um software de desenvolvimento mobile criado pela Google.

Com o banco de dados pronto e os devidos relacionamentos, foram inclusos dados fictícios, utilizando o *Faker* da API para gerar dados teste e assim testar o SELECT no banco.

Na segunda fase o Arduino foi configurado e gerado um PIN junto ao RFID, com esse PIN toda vez que o NFC fosse usado os dados seriam puxados do banco contabilizando assim a presença do aluno.

Quanto ao servidor, bastou transferir o banco de dados e configurar o IP para haver a comunicação por meio do direcionamento, o servidor operou normalmente e foi feito um teste de conexão, o aplicativo também foi direcionado para fazer o teste, foram criados campos de teste para ver se havia retorno, a comunicação estava funcionando normalmente.

Com todas as etapas prontas o APK foi gerado e exportado para um celular Samsung J7 Pro, o modelo é octa-core com 3 gb de memória RAM.

Nos testes de caixa branca foi verificado o código fonte para otimizar e tornar o aplicativo mais leve, pois estava com um tempo de 10 segundos para iniciar completamente.

Testes de caixa preta ajudaram na ordem dos elementos na tela e nos testes alguns problemas no retorno das tabelas foi corrigido, porém como não foi possível aplicar um tratamento nos campos para que retorne erro quando algum campo é preenchido de forma errada.

A segurança foi comprometida com a falta de tratamento nos campos de preenchimento na criação de conta, sendo vulnerável a um ataque que utiliza os campos de *input*, para corrigir momentaneamente a tela de criação de conta estará bloqueada no momento, para o beta serão usados dados criados direto no banco para os testes.

Quanto a conectividade foi vista que o aplicativo trava quando não há rede, e algumas vezes a conexão com o banco não acontece sendo necessário reiniciar a aplicação, esse problema deve ser corrigido nas *strings* de direcionamento.

O teste de requisitos foi feito e mesmo a versão beta já cumpriu com o proposto, e mesmo com os problemas e recursos faltando ele atendeu aos requisitos.

## 5 CONCLUSÃO

Após toda a etapa de testes chegou-se à conclusão de que o NFC é uma tecnologia que tem muito potencial, especialmente com controle de presença, não sendo necessário o cartão, mas apenas um celular conectado à rede, e para aplicações mais complexas que usam recursos nativos o Expo não é indicado e é preciso fazer uma boa tratativa a respeito das vulnerabilidades nos campos de preenchimento.

A conexão com o servidor precisa estar correta pois caso contrário a aplicação não irá conectar ao banco de dados o que ocasionará no travamento do aplicativo, além do fato de que o Arduino precisa estar configurado corretamente, pois pode haver erros na hora de buscar as tabelas e para conectar, diversas vezes a aplicação não conseguia localizar o sensor, e o sensor também deixou de se comunicar algumas vezes.

Foi visto que a linguagem PHP teve um bom desempenho junto ao banco MySQL, devido ao tempo de resposta ter estado dentro do esperado, considerando o tamanho da aplicação, não havendo bugs e houve maior facilidade de encontrar materiais relacionados a essa integração, o fator que mais limitou o projeto foram as vulnerabilidades, pois elas demandam muito tempo para fazer a tratativa, com isso foi preciso sacrificar a segurança para ganhar tempo e desempenho, mas em um cenário ideal deve haver um equilíbrio entre as especificações. A aplicação funcionou normalmente no aparelho selecionado para testar.

Conforme o apresentado na tabela 1, foram constatados 3 problemas principais, sendo eles:

Tabela 1 – Alunos e Escolas

| Falhas                               | Solução   |
|--------------------------------------|---|
| Lentidão                             | A solução encontrada foi otimizar o código        |
| Travamentos com falta de internet    | Manter um banco de dados local                    |
| Inputs aceitam qualquer tipo de dado | Tratar os inputs para aceitarem os dados corretos |

Fonte: Elaborada pelos autores (2020)

- a) Lentidão – Devido a quantidade de tecnologias empregadas houve gargalos, o aplicativo levava mais de 20 segundos para iniciar ou responder algumas ações, a solução encontrada foi otimizar o código e tentar usar ao máximo as tecnologias de forma separada, para que não houvesse uma integração extra custando desempenho;
- b) Travamentos – O aplicativo para de funcionar devido a falta de internet já que o banco de dados está na nuvem, a solução seria manter um servidor local para manter a operação do aplicativo, e os dados seriam atualizados no servidor em nuvem, com isso seria possível eliminar o problema;
- c) Inputs – Os inputs caracterizam uma falha na segurança pois campos que deveriam aceitar só números estavam aceitando qualquer caractere, isso poderia gerar aberturas para ataques e poderia haver uma entrada incorreta de dados gerando o travamento do aplicativo, a solução encontrada é o tratamento dos campos para dificultar que esses problemas citados ocorram.

Verificando melhor as conexões, foi visto que é possível trabalhar de maneira rápida e o acesso a recursos nativos foi facilitado com o React-Native e o Node no back-end, também é recomendado utilizar um aparelho real para testar pois o emulador apresenta alguns bugs e tem um tempo de resposta maior. Com a *cloud computing* não houve grandes problemas com o servidor, a única desvantagem é que conexão com a internet é indispensável, com isso o sistema não funciona sem conexão, a parte de backup foi facilitada por estar em nuvem, sendo preciso fazer as configurações, a aplicação funcionou corretamente nos testes e registrou corretamente além de trazer as pesquisas através das *strings* com o SELECT. React-Native e Node.js funcionam em harmonia no desenvolvimento, gerou maior rapidez e por ambos os frameworks usarem o JavaScript, a integração foi boa e gerou uma maior rapidez tanto no desenvolvimento quanto na otimização.

## REFERÊNCIAS

AGNI, Eduardo. **Mobile First** (ou boas razões para investir em Mobile). São Paulo. 2015

ALECRIM, Emerson. **O que é NFC? (Near Field Communication)?**. 2012. Disponível em: <https://www.infowester.com/nfc.php>. Acesso em: 05 maio. 2020.

BRITO, Edivaldo. **O que é NFC?**. 2012. Disponível em: <https://www.techtudo.com.br/artigos/noticia/2012/01/o-que-e-nfc.html> Acesso em: 05 maio. 2020.

COHN, Mike. **Sucedendo com agilidade**: desenvolvimento de software usando Scrum. Educação Pearson, 2010.

EVANS, Martins; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino em Ação**. 1. ed. São Paulo: Novatec Editora, 2013.

FINKENZELLER, Klaus. **RFID Handbook**: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication. 3. ed. Reino Unido: John Wiley and Sons, 2010.

LEON-DUARTE, Jaime A.; RE-INIGUEZ, Blanca M. De La; ROMERO-DESSENS, Luis F. Ventajas del uso de sistemas de trazabilidad electrónica en procesos de manufactura. **Inf. tecnol**, La Serena, v. 31, n. 1, p. 237-244, fev. 2020. Disponível em: [https://scielo.conicyt.cl/scielo.php?script=sci\\_arttext&pid=S0718-07642020000100237&lng=es&nrm=iso](https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-07642020000100237&lng=es&nrm=iso). Acesso em: 05 maio. 2020.

NASSAR, Victor; VIEIRA, Milton. A rastreabilidade da medicação do paciente no hospital com a tecnologia RFID. **Rev. cienc. tecnol**, Posadas, n. 32, p. 1-10, dez. 2019. Disponível em: [http://www.scielo.org.ar/scielo.php?script=sci\\_arttext&pid=S185175872019000200007&lng=es&nrm=iso](http://www.scielo.org.ar/scielo.php?script=sci_arttext&pid=S185175872019000200007&lng=es&nrm=iso). Acesso em: 05 maio. 2020.

OLIVEIRA, Sérgio de. **Internet das Coisas com ESP8266, Arduino e Raspberry Pi**. 1. ed. São Paulo: Novatec Editora, 2017.

PEDROSA, Paulo H.C.; NOGUEIRA, Tiago. Computação em Nuvem. **UNICAMP**, Campinas, p. 1-4, jun. 2011. Disponível em: <https://www.ic.unicamp.br/~ducatte/mo401/1s2011/T2/Artigos/G04-095352-120531-t2.pdf>. Acesso em: 05 jun. 2020.

PINHEIRO, Roberto Meireles. Pesquisa de mercado. Editora FGV, 2015.

PRESSMAN, R. S. **Engenharia de Software**. 6. ed. Rio de Janeiro: McGraw-Hill, 2006, 720p.

RUSCHEL, Henrique; ZANOTTO, Mariana Susan; MOTA, Wélton Costa da. Computação em Nuvem. **Pontifícia Universidade Católica do Paraná**, Curitiba, p. 1-15, abr. 2010. Disponível em:

<https://www.ppgia.pucpr.br/~jamhour/RSS/TCCRSS08B/Welton%20Costa%20da%20Mota%20-%20Artigo.pdf>. Acesso em: 05 jun. 2020.

SABBAGH, Rafael. **Scrum**: Gestão ágil para projetos de sucesso. 1. ed. São Paulo: Casa do Código, 2014.

SOUSA, Flávio R. C. et al. Gerenciamento de Dados em Nuvem: Conceitos, Sistemas e Desafios. **Universidade Federal do Ceará**, Ceará, p. 1-40, dez. 2011. Disponível em: [http://200.17.137.109:8081/novobsi/Members/josino/fundamentos-de-banco-de-dados/2012.1/Gerenciamento\\_Dados\\_Nuvem.pdf](http://200.17.137.109:8081/novobsi/Members/josino/fundamentos-de-banco-de-dados/2012.1/Gerenciamento_Dados_Nuvem.pdf). Acesso em: 05 jun. 2020.