

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

**Automação e registro de visitas médicas utilizando
dispositivos mobile e blockchain**

Yuri Marques Strack

CENTRO TÉCNICO CIENTÍFICO - CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Ciência da Computação

Rio de Janeiro, novembro de 2021



Yuri Marques Strack

**Automação e registro de visitas médicas utilizando
dispositivos mobile e blockchain**

Projeto Final, apresentado ao programa **Ciência da Computação** da PUC-Rio como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Markus Endler
Departamento de Informática

Rio de Janeiro Novembro de 2021

Agradecimentos

Dedico essa dissertação, primeiramente, aos meus pais e irmã por me apoiarem e incentivarem a alcançar meus objetivos, servindo de base para tudo em minha vida.

Aos meus avós, tios e tias que torceram por esse momento, contribuíram com a minha formação e acreditaram no meu potencial.

Aos mentores do projeto Apple Developer Academy | PUC-Rio por todos ensinamentos, companheirismo e troca de experiências que me enriqueceram tanto como pessoal como profissional.

Aos amigos feitos durante o curso, que, com toda certeza, tornaram a jornada até a formação mais tranquila e prazerosa. Em especial ao Mateus Levi por todo apoio e pela ajuda.

Por fim, ao Markus Endler e Antonio Lyda por aceitarem me orientar e, também, por toda ajuda ao longo do tempo que me dediquei a este trabalho.

Resumo

Marques Strack, Yuri; Endler, Markus. **Automação e registro de visitas médicas utilizando dispositivos mobile e blockchain.** Rio de Janeiro, 2021. Número de páginas **42**. Projeto Final de Graduação - Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

A documentação do uso de materiais hospitalares durante o atendimento clínico é um processo de extrema importância para o Hospital, sendo de responsabilidade do médico a criação do registro. Mas problemas nos registros, ainda, são recorrentes e geram perdas monetárias para os prestadores do serviço, e existe espaço para a introdução de tecnologias que podem melhorar a qualidade do processo de documentação. Neste trabalho é apresentado uma solução mobile iOS que, junto ao uso de tags NFC e blockchain, busca automatizar e minimizar os erros de documentação.

Palavras-chave

Saúde, visita médica, aplicativo, iOS, NFC, Blockchain

Abstract

Marques Strack, Yuri; Endler, Markus. **Automation and registration of medical visits using mobile devices and blockchain.** Rio de Janeiro, 2021. Número de páginas **42**. Projeto Final de Graduação - Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Documenting the use of hospital materials during clinical care is an extremely important process for the Hospital, and the doctor is responsible for creating the record. Problems in records are still recurrent and generate monetary losses for service providers. There is room for the introduction of technologies that can improve the quality of the documentation process. This work presents a mobile iOS solution that, together with the use of NFC tags and blockchain, seeks to automate and minimize documentation errors.

Keywords

Health, clinical visit, application, iOS, NFC, Blockchain, iOS

Sumário

Introdução	1
Situação Atual	3
Objetivos do trabalho	5
Atividades realizadas	7
Conceituação do projeto	7
Estudos de tecnologia	10
Protótipos para aprendizado e demonstração	11
Método de desenvolvimento	14
Projeto e especificações do sistema	17
Banco de dados	17
Tags NFC	18
Aplicativo Mobile	20
Blockchain	22
API	23
Integração	24
Implementação e avaliação	26
Testes da API e Smart Contract	27
Testes com celular	28
Considerações finais	30
Referências bibliográficas	32

Lista de Figuras

Figura 1: Diagrama de contexto	8
Figura 2: Componentes do sistema	9
Figura 3: Modelo de dados	10
Figura 4: Início da jornada do usuário	11
Figura 5: Fim da jornada do usuário	12
Figura 6: Protótipo de alta fidelidade	13
Figura 7: Método CBL	14
Figura 8: Tabela CBL desenvolvida para o projeto	16
Figura 9: Dashboard do banco de dados	18
Figura 10: Conjunto de tags NFC	18
Figura 11: Dados armazenados nas tags	19
Figura 12: Diagrama de pacotes do App	20
Figura 13: Principais funções do blockchain	23
Figura 14: Diagrama de sequência	25
Figura 15: Interface do Ganache	28
Figura 16: Exemplo de alerta de erro	29
Figura 17: Método de priorização MoSCoW	31

1. Introdução

Auditoria hospitalar é uma atividade de avaliação dos processos e atividades médicas realizados em instituições de saúde. A auditoria é conduzida principalmente por auditores médicos e enfermeiros, com o objetivo de levantar dados para analisar a qualidade e os custos atrelados ao serviço médico prestado [1][2]. A análise é feita em cima dos registros gerados pelos profissionais de saúde sobre o atendimento dado ao paciente e é papel do auditor garantir a autenticidade e conformidade do documento [3]. Dessa forma, busca-se melhorar a qualidade das atividades médicas e ajudar na administração dos custos.

Uma das modalidades de auditoria hospitalar é a auditoria de contas, realizada antes dos registros do atendimento serem enviados para as operadoras quando um paciente recebe alta, a fim de verificar se houve alguma inconformidade que possa acarretar em glosas hospitalares [1]. Esse tipo de glosa é definida como o não pagamento, por parte das operadoras, de materiais, máquinas e procedimentos, levando as instituições de saúde a perderem receita e possivelmente receberem implicações legais [2].

Um estudo, feito por Ricardo de Oliveira e Elisabeth de Oliveira, aponta que a maioria das inconformidades encontradas, que podem gerar glosas, estão relacionadas ao registro irregular do procedimento (27.5%) e a falta de documentos que comprovem a atividade (21.6%). A pesquisa analisou 251 prontuários, sendo aplicadas glosas em 146 deles (58.2%). No período estudado, o valor perdido dado às glosas foi de R\$46.890,78, o que corresponde a 2.06% do faturamento de internações. Todavia, em análise nos centros cirúrgicos o valor glosado pode chegar a 50% do faturamento [2].

Nesse contexto, fica evidente a importância dos profissionais de saúde em gerar registros médicos de qualidade e confiança. O uso de tecnologia na área de saúde, tal como sensores, dispositivos de comunicação sem fio e blockchain, pode ajudar na redução das inconformidades presentes nos documentos. Isso porque, grande parte das inconformidades em registros vem de simples erros de preenchimento das informações ou falta do mesmo, tanto em meio manual quanto digital [6].

Dessa forma, existe a oportunidade de usar tecnologia de comunicação sem fio por aproximação para automatizar o processo de documentação da utilização de maquinários, materiais e a prestação de serviços, sem a

necessidade de entrada manual dos dados. Já o uso de blockchain entraria para trazer confiança aos auditores e operadoras, devido a inviolabilidade e autenticidade que é característica dessa tecnologia, pois ela é formada por blocos de informação armazenados em um ledger distribuído que atua como um livro-razão, onde a escrita nele depende do consenso entre todos os nós da rede [16]. Assim, médicos e enfermeiros teriam um menor esforço para cumprir adequadamente o registro das atividades prestadas aos pacientes durante seus atendimentos, além de beneficiar os auditores durante as reuniões de auditoria - assim minimizando futuras glosas.

Esta dissertação foi dividida da seguinte maneira: o capítulo a seguir apresenta o contexto atual de registro e identificação de informações médicas, bem como o projeto busca diferenciar-se do que já existe; no terceiro serão elucidados os objetivos e a proposta do projeto; já no quarto serão exibidos as atividades realizadas para alcançar o objetivo de construção do sistema; no quinto, as especificações de cada parte do sistema. No penúltimo, uma análise crítica das decisões e testes realizados; por fim, as considerações finais e trabalhos futuros.

2. Situação Atual

A tecnologia na área de saúde avança cada vez mais e atualmente, o uso de dispositivos inteligentes e a digitalização de dados médicos já é realidade. As redes de sensores têm se tornado cada vez mais populares devido à disseminação dos celulares e acessórios aptos a captar dados médicos para monitoramento do paciente [7]. Além disso, hospitais têm adotado sistemas eletrônicos para armazenamento do histórico clínico do paciente, como o prontuário eletrônico [13][17], e para gerência administrativa, armazenando dados dos insumos usados para pagamentos.

Com a modernização das atividades nos hospitais, é preciso entender como o processo de auditoria de contas é afetado e funciona. Dentro da atividade de auditoria os médicos e enfermeiros que participam são os auditores e possuem como papel repassar todos os registros dos insumos utilizados no tratamento de um paciente, analisando a autenticidade e conformidade com o que foi de fato realizado [3]. A origem dos registros utilizados no processo pode variar de hospital para hospital; aqueles que adotam medidas eletrônicas usam os dados vindo do sistema de gerenciamento administrativo (caso exista) ou do prontuário eletrônico [3]. Para os que não adotam medidas eletrônicas, ainda são usados registros anotados à mão. Após análise, o registro do atendimento pode ser enviado às operadoras para pagamento do serviço [1].

Dentre as informações que um registro deve conter estão os dados do paciente, do profissional de saúde prestador do serviço, materiais e/ou equipamentos utilizados e horário da realização da visita médica [4]. Foi feita uma análise de como ocorre a identificação de cada um dos pontos acima no mercado:

- Pacientes internados são identificados através do uso de pulseiras [8][9] que contêm uma etiqueta de identificação, impressa ou escrita, com nome completo do paciente, data de nascimento e número de identificação do prontuário. Além da pulseira, também é utilizado uma placa fixada ao lado do leito do paciente, com as mesmas informações contidas na etiqueta de identificação e também número do leito, riscos assistenciais e data de admissão no centro de internação [9][10].

- Profissionais de saúde são identificados através do nome e CRM. Em ambiente hospitalar, utilizando crachás com foto, nome, sobrenome e cargo.

- Maquinários são identificados pelo código de barras, número de série e nome. Existem hospitais que possuem um sistema de gerenciamento de equipamentos, armazenando neles também a frequência de falhas e o custo de uso, de reparo e de inatividade [11].

O projeto busca trazer uma abordagem diferente para algumas das tecnologias mencionadas e aprimorar a forma como a identificação em registros funciona. Serão usadas tags (NFC) de identificação para os pacientes, máquinas e ambientes hospitalares, que hoje são identificados por etiquetas e crachás. O celular será o ativo do processo, responsável por entrar em contato com as tags e através de um aplicativo montar o registro que será armazenado nos sistemas eletrônicos, sem a necessidade de preenchimento manual de dados pelos profissionais.

3. Objetivos do trabalho

A ideia é interfacear a funcionalidade de criação de registros de uso de materiais durante uma visita médica presente nos sistemas hospitalares por meio de uma aplicação mobile que interaja com dispositivos de comunicação sem fio. A escolha do celular é motivada pelo fato dele ser um aparelho que conta com a disponibilidade de diversos sensores, processamento e comunicação sem fio velozes, e facilidade em transporte para diversos locais. Sendo assim, máquinas e ambientes hospitalares não precisam possuir funcionalidade de troca de dados com dispositivos inteligentes e o profissional de saúde não fica limitado ao espaço físico onde o paciente/máquina/material está localizado.

A plataforma escolhida para o desenvolvimento foi o iOS; como estamos lidando com hardware específico de cada plataforma é preciso utilizar código nativo para ter acesso às funções, o que em código híbrido demandaria uma combinação de plugins e código específico da plataforma [14]. Além disso, o uso de sistemas iOS serve como ponto de partida para validação do projeto, uma vez que, com o produto em mãos é possível realizar testes e gerar métricas que indiquem a necessidade de evolução ou migração para outra plataforma ou tecnologia.

A dinâmica do App consiste em o usuário (médicos e enfermeiros) entrar em contato com tags NFC para identificação do contexto da visita médica e criação do registro da mesma. Pacientes, máquinas e ambientes hospitalares são identificados pela aplicação através das tags, já o médico que realizará o atendimento é identificado por meio do aparelho que roda a aplicação.

A formação do registro da visita médica é complexa e desafiadora, pois engloba muitas informações [4]. O escopo da aplicação tem como foco registrar que um paciente realizou o exame X, usando o aparelho hospitalar Y com o médico Z.

Com os registros criados, eles serão armazenados em um banco de dados convencional - mas é preciso atribuir confiança e imutabilidade a eles, visto que qualquer inconformidade pode acarretar nas glosas hospitalares. Para isso, o App mobile fará requisições a um backend, responsável pelo armazenamento dos dados e pela rotina de criação dos registros em uma rede blockchain.

O objetivo do trabalho é utilizar a comunicação sem fio NFC entre tags e celulares para automatizar o processo de registro, contribuindo com a diminuição

de erros de preenchimento e minimizando o problema da ausência de documentação de atendimento médico que acarretam em glosas, além de apoiar os profissionais de saúde ao reduzir os esforços com a geração de registros. Ademais, o projeto proposto tem como objetivo pessoal o aprimoramento dos conhecimentos na área de desenvolvimento de aplicativos para celular, uso de tecnologia blockchain e a realização de forma autônoma de um projeto de engenharia de software desde a fase de projeto até o desenvolvimento de um produto. Com essas propostas, é esperado ao final do trabalho uma POC (Proof of Concept) do sistema, mas sem a testagem em um ambiente hospitalar real.

4. Atividades realizadas

O projeto foi dividido em três grandes partes: na primeira, foi necessária uma pesquisa e investigação de problemas existentes na saúde que poderiam ser amenizados através do uso de tecnologia. O objetivo era encontrar uma área para atuar no projeto e o resultado dessa fase já foi abordado no primeiro e segundo capítulo; na próxima fase, foi preparada toda a base conceitual da solução a fim de clarificar o que será desenvolvido; na última ocorreu a implementação, de fato, do sistema.

4.1 Conceituação do projeto

Com a problemática a ser atacada estabelecida, era preciso entender como a solução proposta funcionaria dentro de um contexto hospitalar. Para isso, foi elaborado uma descrição do fluxo de uso da aplicação:

*O **Hospital** é responsável por fornecer **Profissionais de Saúde, Materiais e Ambientes** para o atendimento do **Paciente**.*

*O **Paciente**, ao chegar no **Hospital**, passa pelo processo de reconhecimento ou cadastro de seus dados no sistema por um **Profissional de Saúde**. Após o primeiro contato, uma pulseira (com uma tag NFC) de identificação é dada ao **Paciente**.*

*No momento do atendimento, a preferência de atenção do **Profissional de Saúde** é sempre no **Paciente**, portanto, nesta etapa, a avaliação ocorre normalmente até o fim. Ao terminar o serviço, antes de dispensar o **Paciente**, o **Profissional de Saúde**, utiliza seu celular para entrar em contato com a tag de identificação presente na pulseira e, em seguida, com as tags de identificação dos **Materiais** utilizados durante a prestação do serviço e na sala (ambiente) hospitalar.*

*Assim, um registro da visita médica é gerado pela aplicação, através da associação do **Profissional de Saúde** ao uso de **Materiais** durante o atendimento do **Paciente** no **Ambiente Hospitalar**. O registro gerado é direcionado para o **Registrador** onde uma versão imutável e inviolável será armazenada.*

*O **Profissional de Saúde** é o único a interagir com o sistema (o usuário) e deve ter a aplicação mobile previamente instalada em seu celular de trabalho para uso após atendimento.*

Além disso, o **Profissional de Saúde** também pode participar de auditorias hospitalares, onde precisará do acesso a todos os registros das visitas para prestação de contas de uso dos **Materiais**. Para isso, o mesmo pode visualizar os registros e verificar sua autenticidade no **Registrador**, caso tenha o **cargo de Auditor**.

Por fim, quando o paciente for embora do **Hospital**, sua pulseira deve ser devolvida e seus dados apagados da mesma.

A partir do fluxo descrito, é possível levantar quais entidades fazem parte do sistema e quais dados são transferidos e devolvidos por ele. Uma maneira de visualizar o sistema, sem descrição textual longa, é construindo um Diagrama de Contexto [21], pois ele é capaz de representar a relação do produto com o ambiente sem precisar do detalhamento das tecnologias por trás do funcionamento.

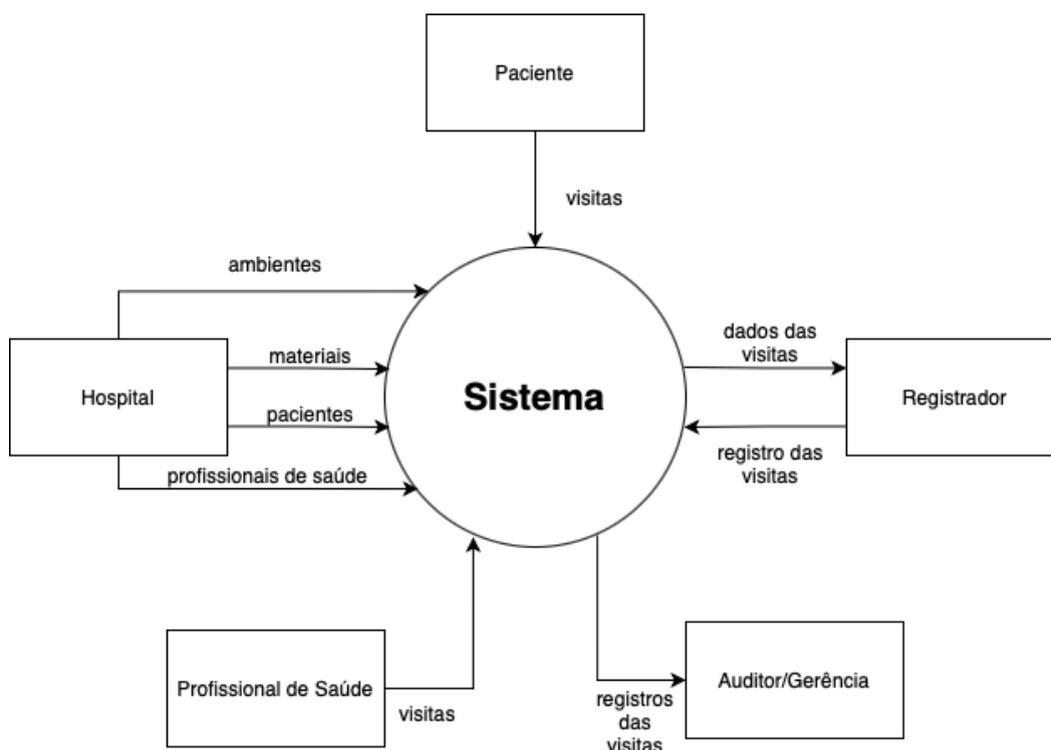


Figura 1: Diagrama de Contexto.

A próxima etapa, consiste no detalhamento e subdivisão do "Sistema" representado acima, já que ele ainda é uma caixa preta nesse estágio. Por se tratar de uma solução que envolve um aplicativo mobile, a primeira separação é bem clara: o App de celular. Também é preciso guardar os dados dos pacientes, profissionais de saúde, materiais, ambientes hospitalares e das visitas médicas

realizadas em algum lugar, por causa disso, a segunda separação é o banco de dados. Outro componente evidente é a parte da blockchain, que guardará a referência segura e imutável do registro do atendimento médico. Por fim, a última peça não é tão clara e sua criação surge dos estudos de tecnologia realizados e comentados na subseção 4.2, que é a API para permitir a comunicação entre os componentes sem precisar saber a maneira como foram implementados [24].

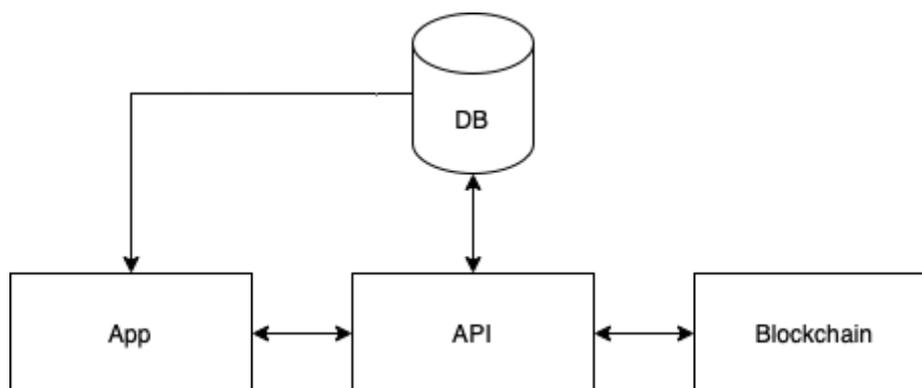


Figura 2: Componentes do sistema.

O Diagrama de Contexto mostrado na figura 1, junto à descrição do fluxo de uso apresentada anteriormente, revela as entidades que fazem parte da solução que, agora, também precisam ser detalhadas. O conjunto das entidades representa o modelo de dados da aplicação e para cada uma das informações que fazem parte do sistema, foram adicionados seus respectivos atributos e a maneira como eles se relacionam, conforme a figura 3.

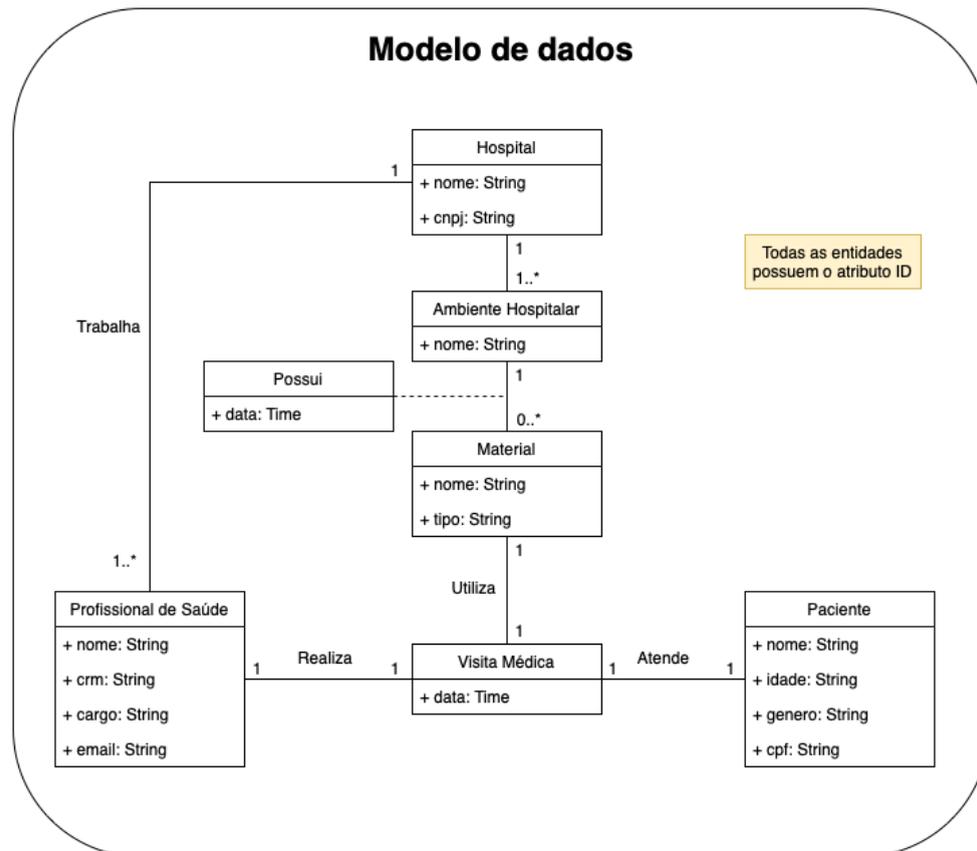


Figura 3: Modelo de dados.

4.2 Estudos de tecnologia

Durante o projeto, foi preciso relembrar algumas tecnologias e também aprender novas. Isso porque, apesar de ter conhecimento prévio na área de desenvolvimento iOS, o sistema proposto requer a utilização de comunicação NFC e uma forma de interação entre o aplicativo no celular e o local de armazenamento dos dados, tanto o banco de dados convencional como a rede blockchain.

Para a comunicação sem fio NFC, a Apple disponibiliza o framework Core NFC [22] que permite detectar, ler e escrever nas tags usando o formato NDEF (NFC Data Exchange Format). O uso dessa biblioteca já facilita o desenvolvimento, uma vez que ela faz o intermédio entre o hardware e o software, deixando para o desenvolvedor a tarefa única de manipulação dos dados. O desafio nesse ponto é a integração dele com o framework usado para desenvolver a interface. Assim, foi necessária a leitura da documentação oficial da Apple e também de artigos sobre o assunto.

O outro ponto de estudo foi a maneira como o aplicativo iria se comunicar com a blockchain. Existe uma biblioteca bastante conhecida e recomendada em diversas publicações chamada web3.js [23]. Somado a isso, não existe nenhum framework oficial da Apple que faça essa comunicação. Por isso, surgiu a necessidade da criação de uma API em Javascript que recebesse os dados vindos do escaneamento feito no aplicativo e enviasse para a blockchain, e vice-versa.

4.3 Protótipos para aprendizado e demonstração

Dado o fato do projeto possuir uma interface de interação do usuário (profissional de saúde) com o sistema, foi necessária uma etapa extra após a conceituação e antes da implementação em código do produto. Nela, diversos protótipos de interface foram elaborados, começando por um wireframe em papel até a evolução para um protótipo de alta fidelidade.

Com o objetivo de facilitar o desenvolvimento das telas, uma jornada do usuário [25] foi elaborada previamente. O intuito deste artefato é compreender todas as fases de interação entre o produto e o usuário final. Dessa forma, são mapeados todos os passos de contato do usuário durante o uso da aplicação.

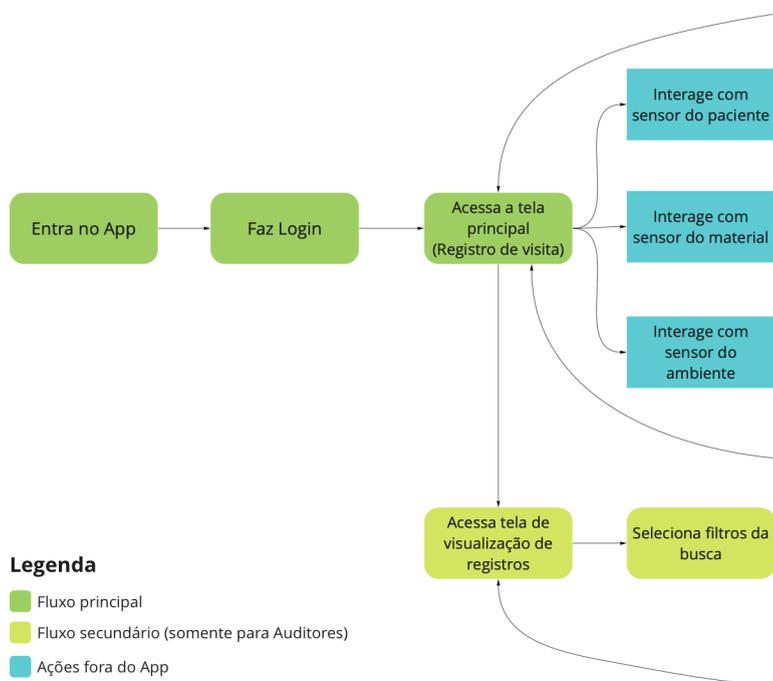


Figura 4: Início da jornada do usuário.

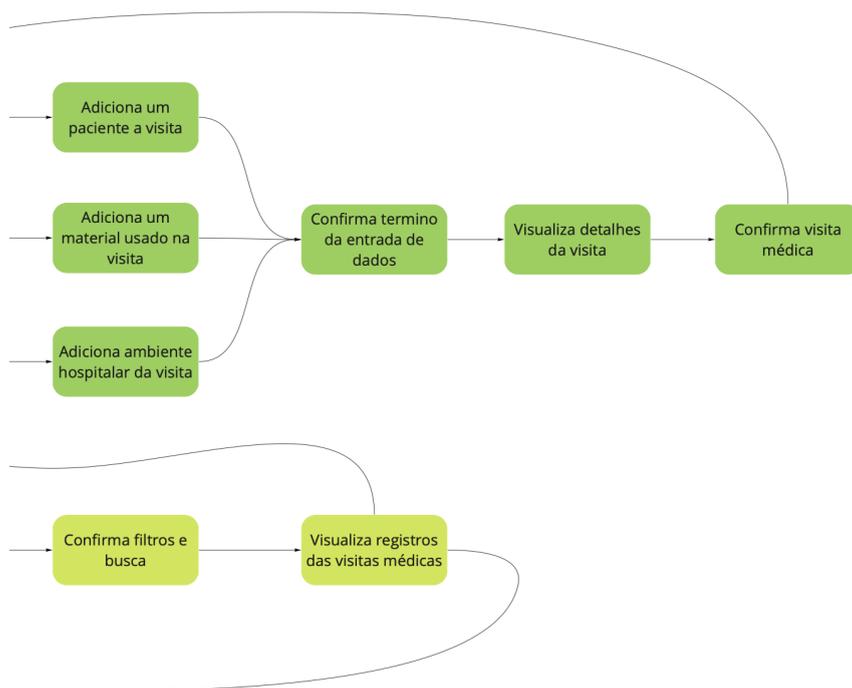


Figura 5: Fim da jornada do usuário.

Em seguida, já tendo os caminhos possíveis de interação com as telas, o protótipo de alta fidelidade foi construído usando a ferramenta Figma. Para sua criação, foram adotadas algumas das diretrizes expostas no Human Interface Guideline da Apple [26], conjunto de documentos que possuem recomendações para melhorar a experiência do usuário por meio de interfaces intuitivas, didáticas e consistentes.

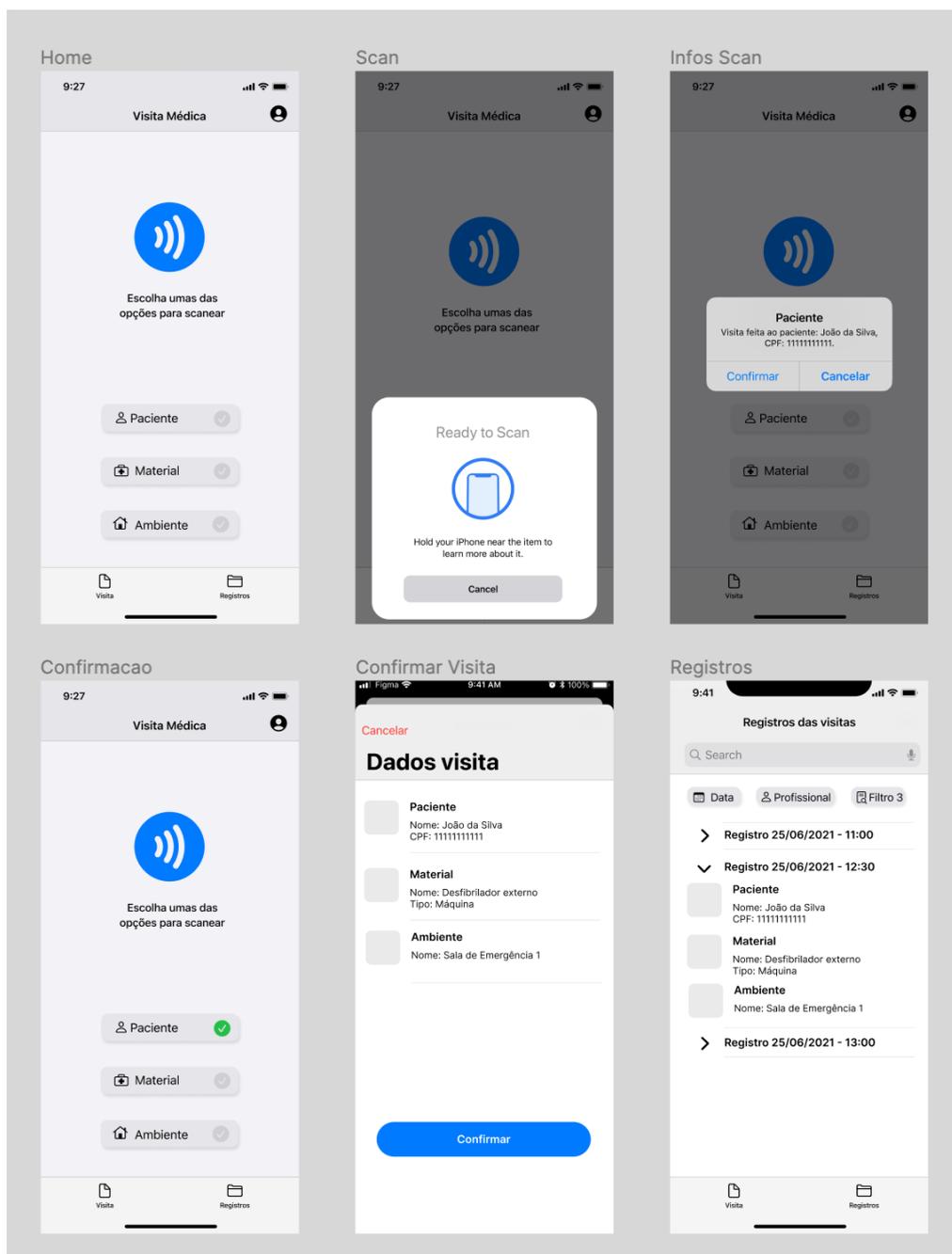


Figura 6: Protótipo de alta fidelidade

4.4 Método de desenvolvimento

Para desenvolvimento do projeto foi adotado o método Challenge Based Learning (CBL) [19]. Essa metodologia foi criada pela Apple como resultado de um projeto chamado "Apple Classroom of Tomorrow-Today", que possuía o objetivo de destacar os fundamentos do ambiente de aprendizado do século XXI. As descobertas desse projeto, com destaque ao aprendizado experimental,

inspiraram o CBL - que se destaca como um framework moderno e intuitivo, tanto para o aprendizado quanto para resolução de problemas.

O método do CBL é caracterizado por três fases: Engage, Investigate e Act. No engage, o aprendiz busca explorar um tema escolhido, buscando bibliografias, fazendo perguntas iniciais e, de forma geral, imergindo-se no assunto com o intuito de definir um problema central a ser solucionado.

Tendo definido o problema a ser abordado, inicia-se o investigate, no qual o aprendiz investiga opções de resolução desse problema. Essa investigação passa tanto por uma abordagem técnica - frameworks a serem usados, o estado da arte tecnológica para aquele problema - quanto de design - em termos de como pode ser construída. Nessa fase, ainda é produzido um planejamento de ações com datas e objetivos a serem cumpridos até o fim da implementação da solução.

O CBL então se conclui pela fase Act, na qual o aprendiz, de fato, traz à realidade sua solução baseando-se nos aprendizados das fases anteriores. É nessa fase que ocorre a implementação do código e a interação do público alvo com a solução construída.



Figura 7: Método CBL

No projeto, em si, a fase de engage foi o momento de busca por problemáticas dentro do "grande tema" saúde, que perdurou até a escolha do problema a ser atacado e envolveu pesquisa fria e conversas com os orientadores. No segundo momento, o investgate, realizei um estudo mais aprofundado em relação às tecnologias que gostaria de adotar na solução, bem como a preparação dos artefatos de conceituação do projeto, como definição do escopo de atuação, diagramas de fluxo de dados, prototipação e casos de uso. Por fim, no act, implementei a solução que divide-se em três partes distintas - aplicativo mobile, API e Smart Contract (programa que é executado na rede Ethereum de blockchain, associado a um endereço da rede e onde as funções e dados ficam armazenados [20]).

No decorrer de todas as fases, diversos questionamentos foram levantados e todo aprendizado advindo das perguntas foi registrado em uma tabela com três colunas: na primeira, fica a pergunta a ser respondida; na segunda, as atividades realizadas para responder e as fontes das informações, e na terceira, o que foi aprendido.

Guiding Questions (Questões Guia)	Guiding Activities (Atividades Guia)	O que Aprendemos?
O que é uma visita médica?	<ul style="list-style-type: none"> • Conversa com orientador (Antonio lyda) 	Visita médica é o momento em que o médico se encontra com o paciente, realizando um exame ou uma nova prescrição.
Qual a importância desse registro para rede hospitalar?	<ul style="list-style-type: none"> • Conversa com orientador (Antonio lyda) • Pesquisa sobre auditoria médica <ul style="list-style-type: none"> ◦ A importância dos registros cirúrgicos para auditoria médica e para o uso de inteligência artificial (IPC) no dia 01/03 • Leitura do artigo: <ul style="list-style-type: none"> ◦ A importância dos registros de enfermagem para auditoria no dia 01/03 ◦ https://cqh.org.br/ojs-2.4.8/index.php/ras/article/view/206/325 (acesso: 29 mar. 2021) • Pesquisa sobre fraudes em registros de exames médicos 	<ul style="list-style-type: none"> • O que tem acontecido muito nos registros de cirurgia é a falta de registros de materiais, que por se tratar de valores altos é onde o olhar do auditor foca, mas na verdade também temos problemas relacionados a descrição de fios cirúrgicos e a prescrição de medicamentos. • O CRM do Paraná publicou um PARECER No 2623/2017 que orienta como devem ser os registros: "A descrição de um procedimento cirúrgico deve conter os dados de identificação do paciente, o nome do procedimento cirúrgico, o diagnóstico, o nome do cirurgião e de seus auxiliares, a hora do início e do fim da operação e, principalmente, a descrição sumária dos tempos cirúrgicos. • É o profissional auditor médico quem analisa as descrições cirúrgicas por conta dos pagamentos que devem ser feitos referentes aos honorários médicos e uso de materiais, medicamentos e equipamentos. • Ferreira (2008), por seu turno, salienta que o registro de enfermagem fornece "a informação da assistência prestada ao cliente e a equipe multidisciplinar, servindo de instrumento para auditoria, evitando as glosas hospitalares"

Figura 8: Tabela CBL desenvolvida para o projeto

5. Projeto e especificações do sistema

O trabalho implementado é um sistema que automatiza o processo de formação de registros de materiais hospitalares durante um atendimento médico e garante a autenticidade e validade dos mesmos, a fim de reduzir a sobrecarga de trabalho dos profissionais e minimizar os problemas associados à auditoria, como as glosas. Nesta seção, serão explicitadas as características técnicas do sistema desenvolvido, dividida em partes que representam cada componente do produto final: banco de dados, tags NFC, App, blockchain e API.

5.1 Banco de dados

O banco de dados adotado para o projeto foi o Firestore do Firebase. Um banco NoSQL, em nuvem, adequado para desenvolvimento em dispositivos móveis e servidores. O Firebase disponibiliza o serviço de autenticação, usando senha, necessário para identificação do usuário profissional de saúde que está utilizando o aplicativo. Ambos os serviços possuem um SDK (Software Development Kit) [34] para o iOS, facilitando o processo de implementação da solução.

Dentro do contexto Firebase, os diferentes tipos de dados são chamados de coleções e o dado, em si, de documento. Uma coleção contém vários documentos. Seguindo o modelo de dados da figura 3, o banco tem 5 coleções - paciente, material, ambiente, profissional de saúde e visita médica. Quanto à entidade hospital, não é necessário criar uma coleção para a mesma pois os dados do banco são referentes a um único hospital. Quanto às propriedades dos documentos, elas são iguais à figura apresentada. Para os testes do projeto, foram criados diferentes documentos com dados mockados.

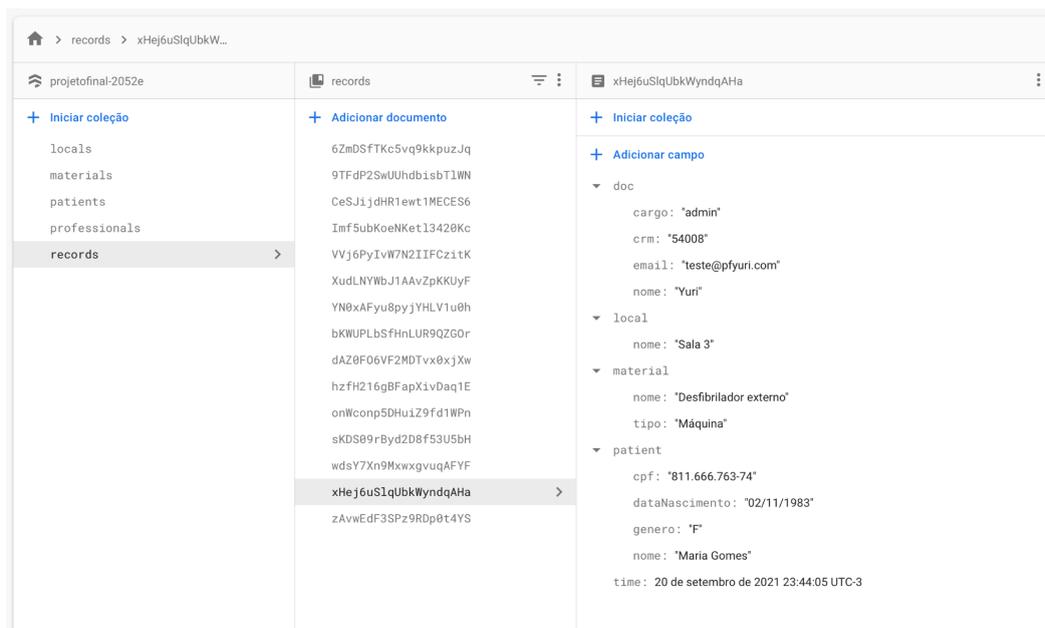


Figura 9: Dashboard do banco de dados

5.2 Tags NFC

A menor e mais simples unidade do sistema. As tags NFC são responsáveis por armazenar dados de certas entidades, dentre as quais estão: o paciente, o material e o ambiente hospitalar. Para o projeto, foi adquirido um conjunto de 10 tags, protocolo ISO 14443 [28] e tipo A, que são compatíveis com o framework de leitura e escrita disponível nos sistema iOS.



Figura 10: Conjunto de tags NFC.

As tags aceitam um conjunto vasto de formatos de armazenamento de dados, sendo o mais comum (dentre os possíveis) as URLs que direcionam à páginas com as informações necessárias para o domínio da aplicação. Todavia, o formato escolhido para o sistema foi outro, devido a dois motivos:

1. O objetivo da tag é identificar uma das entidades, porém todas as informações daquela entidade já estão guardadas no banco de dados e não precisam ser escritas por completo na tag, tanto pela capacidade de armazenamento da mesma quanto pela redundância desnecessária dos dados. Dessa forma, é preciso fazer apenas a relação entre a tag e o dado armazenado no banco via ID.
2. É preciso saber que tipo de dado está sendo lido na tag escaneada (paciente, material ou ambiente) para acessar a coleção correta onde o ID se encontra. Então, além da identificação, também é necessário dizer qual a entidade aquela tag está associada.

Dessa forma, o formato adotado foi a de texto simples. Apesar da não existência de uma formatação na escrita, o conteúdo do texto é um JSON com id e tipo. O campo "id" contém a identificação do dado no banco de dados, o que atende ao primeiro motivo citado anteriormente, e o campo "type" contém o tipo do dado sendo escaneado, atendendo o segundo motivo.

A escolha facilita todo o processo de desenvolvimento do projeto, uma vez que mantém unificada a maneira de troca de dados do sistema como um todo (JSON), permite o uso de enumeração (Enum) para identificar os tipos (campo type) e facilita a query de responsabilidade do App.



Figura 11: Dados armazenados nas tags

5.3 Aplicativo Mobile

O App para iOS foi desenvolvido na linguagem Swift utilizando o Xcode [28], um ambiente de desenvolvimento de software para o ecossistema Apple. As duas principais bibliotecas utilizadas foram o SwiftUI [29] (para interface

visual) e o Core NFC. No início do desenvolvimento do projeto, foi criado um repositório Git local para o versionamento de código.

A função do aplicativo é permitir que o profissional de saúde possa escanear as tags de identificação resgatando e exibindo os dados associados a cada uma delas, para formar um registro do uso dos materiais durante o atendimento. Enviar o registro para a API e, também, visualizar todos os registros criados, para uso no processo de auditoria. Com essas funcionalidades, é possível reduzir a sobrecarga de trabalho do médico e diminuir as inconformidades encontradas nos registros, como falhas de preenchimento ou ausência de informações.

O padrão de arquitetura a utilizado foi o MVVM (Model View ViewModel) com a adoção de alguns conceitos de Clean Architecture, permitindo modularizar e separar o App em três camadas bem definidas: model layer, data layer e scene layer.

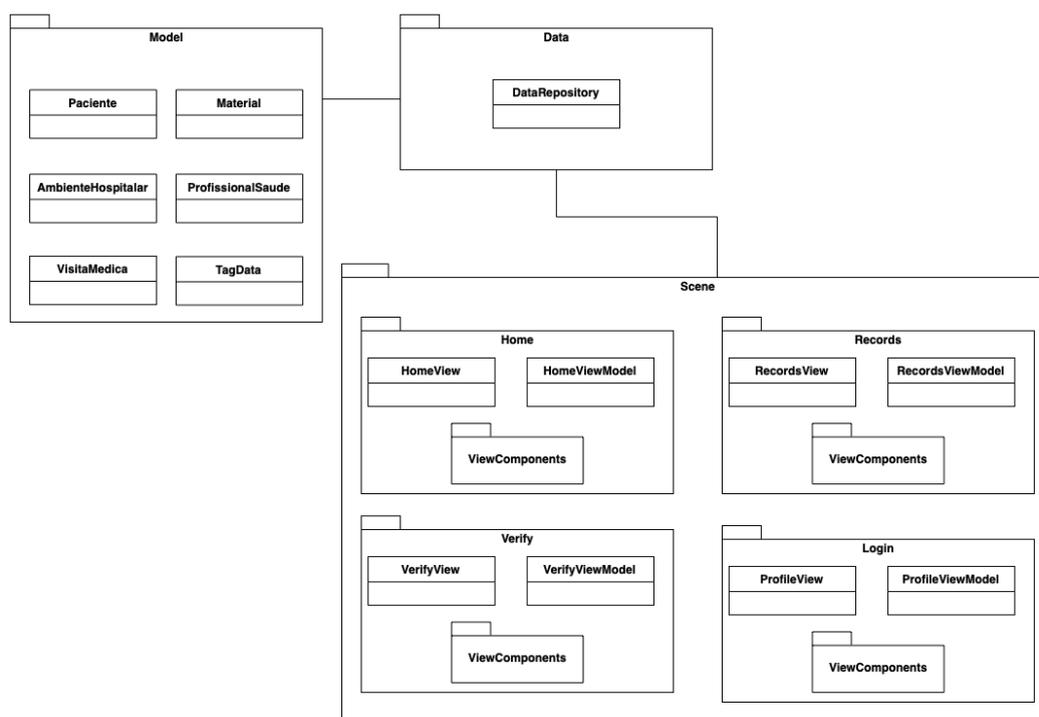


Figura 12: Diagrama de pacotes do App.

A camada de modelo ("Model", na figura 12) é responsável por conter todas as entidades associadas ao contexto da aplicação. Elas não são implementadas como classes no sistema, mas sim em structs com as propriedades apresentadas no modelo de dados da figura 3.

Já a camada de dados ("Data", na figura 12) possui uma única classe denominada *DataRepository* que é a responsável por manipular todos os dados

da aplicação. Dessa forma, é ela que recebe as informações escaneadas, faz as requisições ao banco de dados, devolve os dados do *Paciente*, *Material* ou *AmbienteHospitalar*, monta o registro da visita (*VisitaMedica*) e envia para a API. A classe também possui a referência ao profissional médico autenticado no aplicativo (*ProfissionalSaude*).

Por último, a camada de apresentação ("Scene", na figura 12) contém as cinco telas que formam o aplicativo. Cada uma delas é composta por três elementos: duas classes essenciais - a View e a ViewModel - e uma pasta chamada *ViewComponents* que abriga possíveis componentes de tela menores que compõem a View principal.

A primeira cena a ser apresentada, conforme a jornada mostrada na figura 4, é a de Login. Nela o médico precisa entrar com os dados de email e senha para se autenticar no aplicativo e ter acesso às funcionalidades. Toda a parte de autenticação é feita usando o Firebase Authentication [30]. Essa etapa é importante para que, durante a formação do registro do atendimento, as informações do médico autenticado no sistema possam ser utilizadas, evitando um possível escaneamento extra dos dados do profissional.

Com o login realizado, a segunda tela apresentada é a Home. Sua principal funcionalidade é permitir que o usuário escolha de qual das opções (paciente, material ou ambiente hospitalar) irá ler os dados e escaneá-los. Para isso, existem três botões: um para cada possibilidade, que, ao serem clicados, iniciam uma sessão de leitura NFC - sendo de responsabilidade do médico aproximar o aparelho a tag nesse momento. O App faz a leitura dos dados no formato apresentado na subseção anterior, usando o campo "type" e "id" para fazer o pedido das informações ao Firebase. Em caso de sucesso, o aplicativo mostra as informações associadas à tag NFC na tela, pedindo ao médico a confirmação dos dados e, logo após, sinalizando que a etapa está completa. Já em caso de falha uma mensagem apropriada de erro é exibida. Além disso, a tela ainda permite ao usuário acessar a cena Profile para ver seus dados ou fazer logout.

Após o escaneamento das três informações que compõem o registro médico, a cena de verificação (Verify) é apresentada, tomando todo o espaço em tela. Nela os dados do registro que será criado são apresentados para que o médico possa validar as informações. Ao apertar o botão de confirmação, a visita médica é enviada para a API.

Por fim, existe a possibilidade de visualizar os registros dos atendimentos médicos armazenados. Para isso, existe a tela Records, que possui uma lista das visitas existentes e uma barra de pesquisa para encontrar a informação desejada com maior facilidade.

5.4 Blockchain

Uma das metas do projeto é solucionar o problema da falta de confiança das informações presentes no registro de um atendimento médico, enviadas para as operadoras responsáveis pelo pagamento do serviço prestado. O blockchain atua como uma forma de armazenamento imutável da prova digital de que o documento gerado é autêntico e foi validado por uma organização de confiança, no caso o hospital.

Os dados médicos são considerados sensíveis, por isso devem receber atenção quanto à segurança e caem sobre as implicações da Lei Geral de Proteção de Dados (LGPD) [7]. Além disso, guardar o registro completo da visita médica seria ineficiente e custoso dado o tempo e valor de uma transação no blockchain. Por isso, o uso do termo "prova digital" do documento, formado por dois produtos: o hash e a assinatura digital do mesmo.

O processo de cifragem (hashing) do registro é realizado para gerar uma identificação única e também assegurar a não legibilidade dos dados presentes nele, já que é um ato unilateral, ou seja, não é possível gerar o documento a partir do hash. O algoritmo de cifragem escolhido foi o MD5, que recebe como parâmetro o conteúdo do registro médico salvo no banco e devolve um valor hash de 128 bits (32 caracteres).

Quanto à assinatura digital, ela é utilizada como prova da fonte de origem e validade do registro do atendimento, pois, em uma analogia ao meio físico, o hospital que emite os documentos está rubricando-os. Existem alguns métodos para gerar assinaturas digitais, o escolhido para o projeto foi o JSON Web Token (JWT) . Para o processo, a organização (hospital) precisa de um par de chaves que a identifique. Sendo assim, um par foi criado em um site online de geração de chaves RSA para teste e uso exclusivo no trabalho.

A etapa de geração, tanto do hash quanto da assinatura digital, é feita pela API e está descrita na próxima subseção, a responsabilidade do blockchain é apenas de receber ambos e guardá-los.

A rede de blockchain utilizada no trabalho foi a Ethereum, devido ao conhecimento prévio e por ela ser aberta, o que permite que qualquer operadora

que receba o documento consiga também verificar sua validade. O programa construído, chamado de Smart Contract (Contrato inteligente), foi escrito na linguagem Solidity.

O contrato inteligente armazena os dados da prova digital em um dicionário, sendo a chave o hash da informação e valor a assinatura digital. Além disso, possui duas funções principais, a de adicionar um novo par chave-valor do registro no dicionário e a de resgatar as respectivas informações.

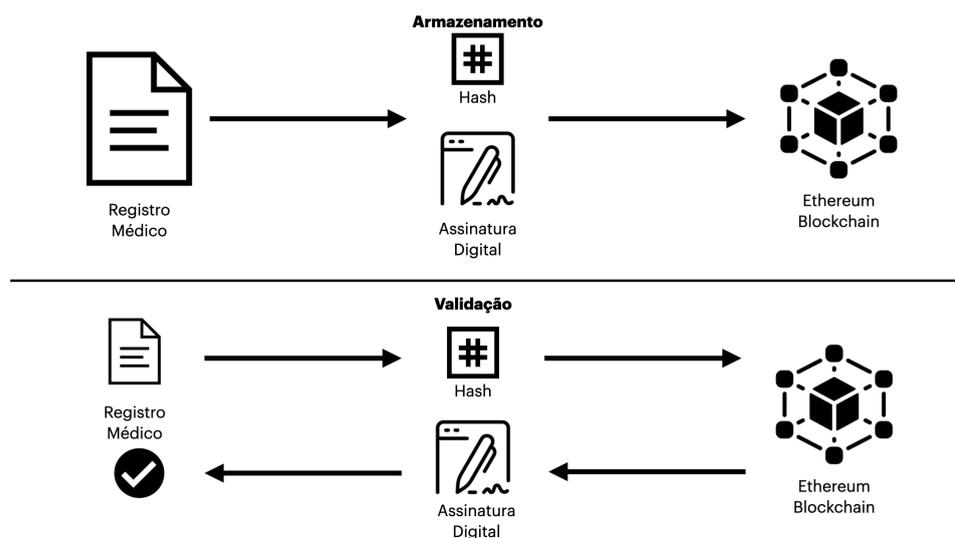


Figura 13: Principais funções do blockchain.

5.5 API

A API foi escrita em JavaScript utilizando NodeJS e usa as bibliotecas necessárias para desenvolver a comunicação HTTP e com a rede Ethereum de blockchain, como ExpressJS e web3js. O seu desenvolvimento aconteceu por último, para facilitar a integração com as outras duas partes.

O principal objetivo da API é intermediar a comunicação entre o App e a plataforma Ethereum, a fim de garantir a criação do registro médico no banco de dados e sua respectiva informação de validação, imutabilidade e autenticidade, descritas na subseção anterior, no contrato inteligente (Smart Contract). Por isso, ela não precisa receber todas as requisições feitas pelo aplicativo, como as de resgate dos dados do paciente, médico, material e ambiente hospitalar que são feitas direto do app para o banco de dados (conforme figura 2), ficando responsável apenas pelas direcionadas à criação do registro e verificação do mesmo.

Dado o contexto, a API possui dois endpoints de método POST [31]: `/api/records`, responsável por criar o registro médico no banco e na blockchain; e `/api/record/verify`, que faz a verificação do registro recebido.

O primeiro recebe, pelo corpo da requisição enviada pelo aplicativo, os dados da visita médica criada. Em seguida, conecta-se com o contrato na Ethereum, gera o hash do registro recebido utilizando o algoritmo MD5 [32], cria a assinatura digital (JWT) associada ao documento da visita utilizando a chave privada RSA256, e envia a forma "limpa" do dado para o banco e o par hash-assinatura para o Smart Contract.

O segundo também recebe o registro completo da visita médica e gera o hash do dado, porém, agora com a função pedindo ao contrato inteligente a assinatura digital do mesmo. Nesse ponto, pode não existir um hash igual ao gerado pela informação recebida, o que significa que o registro foi modificado (por isso não encontra o hash) ou o dado recebido está corrompido, retornando não sucesso na verificação. Caso o hash seja encontrado na blockchain, a assinatura digital (JWT) é resgatada e verificada com o chave pública RSA para provar a validade do dado, retornando sucesso ou falha dependendo do resultado da validação.

5.6 Integração

A última fase da implementação foi a integração de todas as partes, já que uma depende fortemente da outra e somente unidas o sistema funciona por completo. A troca de informação é feita usando o formato JSON e o fluxo principal do trabalho pode ser descrito da seguinte maneira: o App escaneia as informações necessárias para formar uma visita médica, para cada uma delas é feito a leitura da tag e recuperação dos dados presentes no Firebase; com tudo escaneado, o App gera o registro e envia para a API. Na etapa seguinte, a API cria o hash e assinatura digital do documento, armazenando-os no contrato inteligente na rede Ethereum e envia para o Firebase o registro também. Em um momento de auditoria, no futuro, o profissional de saúde também consegue verificar a validade do documento, enviando o registro para API, que pede para o contrato o token, validando-o. A imagem a seguir ilustra a sequência descrita.

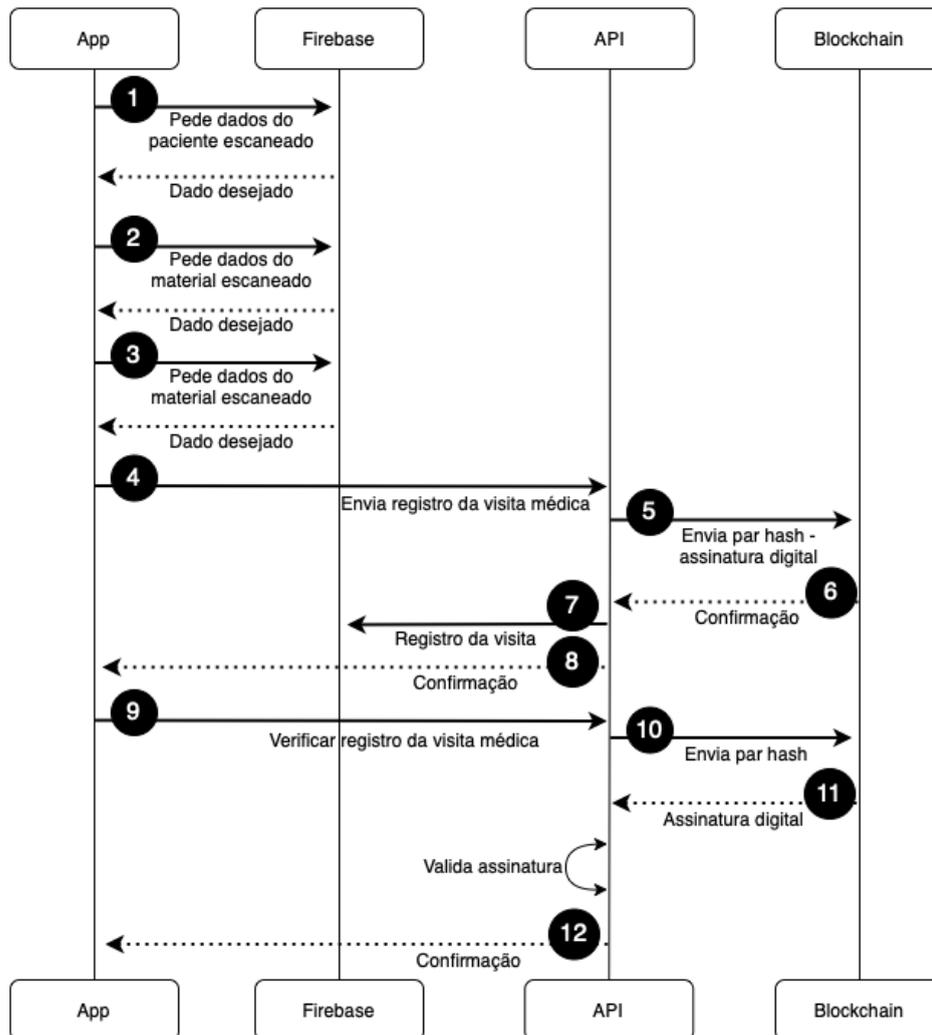


Figura 14: Diagrama de sequência

6. Implementação e avaliação

A maior parte do sistema, em termos de implementação, está contida no aplicativo iOS para celular. Nele foram adotados boas práticas de programação, como a adoção de princípios de SOLID e escolha de uma arquitetura que orna com os frameworks adotados, o que permitiu um código mais modularizado e de fácil entendimento. Quanto ao contrato inteligente e a API, ambos foram feitos em um único arquivo, visto a simplicidade de seus códigos, e desempenharem uma responsabilidade única.

Além disso, algumas decisões também foram tomadas, durante a implementação, que impactam pequenos detalhes do sistema. A primeira delas é a não existência da funcionalidade de CRUD através da interface do aplicativo, onde o usuário somente consegue criar um novo registro e ler os dados salvos. Essa decisão é motivada pelo fato dos dados, necessários para a criação de um registro da visita, virem direto do banco de dados por um pedido feito com a identificação presente na tag NFC. Dessa forma, o médico que está utilizando a aplicação assume a correte dos dados recebidos e não precisa ter a habilidade de edição do que está sendo criado, o que evita erros de preenchimento caso uma edição manual fosse necessária.

Outra questão foi o tempo do registro: ao criar o documento do atendimento, o médico precisa escanear cada uma das informações solicitadas (paciente, material utilizado e ambiente hospitalar da prestação do serviço), o que demanda tempo e requer chamadas assíncronas ao armazenador de dados. Somado a isso, o médico tem a opção de querer registrar o paciente e o ambiente hospitalar pré atendimento, e os materiais utilizados somente após a consulta. Sendo assim, fica a dúvida de qual data ligar ao registro - a de início ou do fim do processo. Para o trabalho, foi escolhido usar a data de fim do trabalho de escaneamento de todos os requisitos.

Outro detalhe não implementado foi a escrita dos dados nas tags NFC pelo próprio aplicativo desenvolvido. Por questões de escopo e da existência de diversos aplicativos utilitários com funcionalidades NFC, a escrita das informações nas tags foi feita através de um aplicativo de terceiros, denominado NFC Tools [35], disponível na App Store do iPhone. No entanto, faz todo sentido o desenvolvimento dessa habilidade em uma versão futura do aplicativo, permitindo o usuário escolher direto do banco o dado que será escrito na tag, com a formatação apresentada na subseção 5.1.

Durante o desenvolvimento, cada uma das unidades foram testadas, primeiro, isoladamente e, depois, integradas. As próximas subseções vão descrever cada um dos testes realizados.

6.1 Testes da API e Smart Contract

Os testes da API foram feitos utilizando o Insomnia [36], um programa open source que funciona como um cliente para a API. Nele é possível requisitar cada uma das chamadas implementadas e visualizar o status de resposta obtido. Para cada um dos endpoints (`/api/records` e `/api/record/verify`) foram testadas requisições com valores incorretos e corretos, analisando se a resposta do servidor estava de acordo com o enviado.

Nesse cenário, foi possível observar e corrigir algumas falhas, dentre elas, a formatação da informação de tempo dos registros. Em Swift, datas possuem formatação diferente da linguagem da API, colocando esse formato no Insomnia, a API, em javascript, manipulava de maneira errada o data pois estava em uma forma desconhecida para a linguagem. A correção adotada foi utilizar o formato de data em unix timestamp para a comunicação entre as diferentes linguagens.

Quanto ao teste do contrato, ele foi dividido em duas partes: de início foi criado um arquivo em javascript para teste unitário de cada uma das funções presentes nele; em seguida, com os testes todos passando, foi preciso subir uma rede de testes Ethereum para implantar o contrato em um bloco.

Para o segundo teste, foi utilizado o Ganache (figura 15) [37], um programa que permite iniciar uma rede ethereum local, dedicada a testes com interface simples e interativa. Nele é possível visualizar o contrato implantado, informações das transações e dados das contas utilizadas para interagir, como endereço, chave privada e balanço. No caso, apesar da disponibilidade de 10 contas, só foi usada uma conta que é a associada ao hospital que roda o sistema.

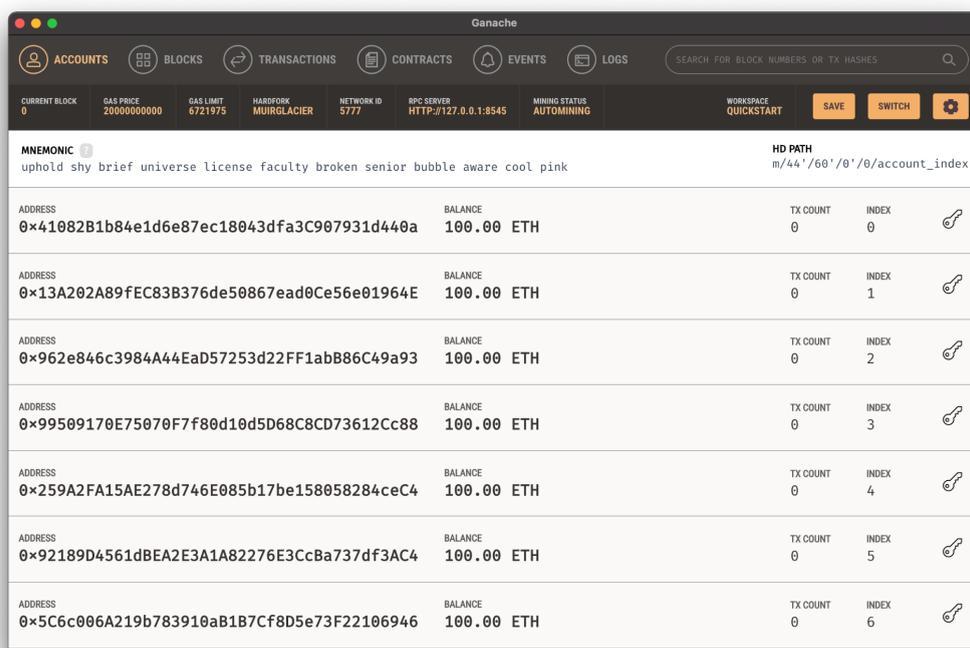


Figura 15: Interface do Ganache.

6.2 Testes com celular

Quanto aos testes do aplicativo, sempre foi utilizado um dispositivo físico, pois as suas funcionalidades dependem do hardware do aparelho, não sendo oferecidas por um simulador.

Os primeiros testes do app não utilizavam as tags NFC para obter os dados, mas sim QR Codes com comportamento similar, já que armazenavam a informação na mesma estrutura. Isso permitiu verificar, em um estágio inicial, a conexão com o banco de dados e validar se a construção do registro da visita médica estava sendo feito de maneira correta.

No momento seguinte, com a validação efetuada e as tags NFC adquiridas, o uso dos QR Codes foi abandonado e a troca para a nova tecnologia foi bem rápida, uma vez que toda a estrutura por trás já havia sido implementada.

Apesar da funcionalidade já ter sido testada, com a troca, novos testes foram feitos para garantir a continuidade do funcionamento. Na nova rodada de testagem, a leitura e criação estavam funcionando, mas a sessão de leitura NFC não terminava após o escaneamento ter acontecido. Também foi observado que em caso de erro da leitura nenhum aviso era mostrado na tela. Para resolução destas questões, foi preciso invalidar a sessão NFC programaticamente após

detectar que os dados já foram lidos. Somado a isso, foi implementado um alerta (figura 16) com a mensagem apropriada na interface, no caso de que um erro fosse detectado.

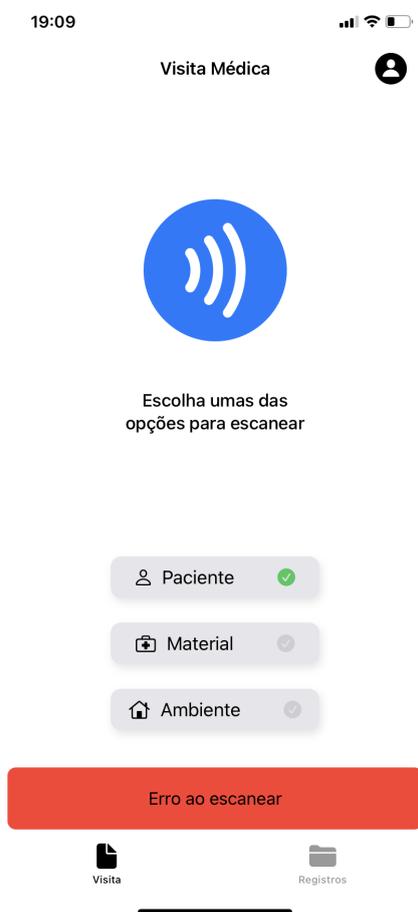


Figura 16: Exemplo de alerta de erro.

7. Considerações finais

A execução desse projeto permitiu colocar em prática diversos aprendizados adquiridos durante o curso, com destaque aos conceitos de programação orientada a objetos, modelagem de software, interação humano computador (IHC) e disciplinas de tópicos em computação que envolveram programação mobile e blockchain. Somado a isso, foram utilizados conhecimentos advindos da participação no programa de inovação tecnológica Apple Developer Academy PUC-Rio.

A confecção do trabalho foi enriquecedora, perpassando por pontos além do tecnológico, como pesquisa, conceituação e design. A escolha por desenvolver um projeto completo de engenharia de software proporcionou grande crescimento e confiança para futuros projetos.

O projeto contribui com uma possível abordagem para solucionar o problema das glosas hospitalares, melhorar o processo de auditoria e reduzir a sobrecarga de trabalho em cima de médicos que precisam registrar, muitas vezes à mão, o uso dos materiais durante o atendimento. Em relação à tecnologia, o trabalho colabora trazendo um aplicativo de celular que interage com a rede Ethereum de blockchain para armazenamento e verificação de documentos digitais, campo ainda pouco explorado.

Vale ressaltar a dificuldade de trabalhar em uma solução voltada para o ambiente hospitalar, visto que cada hospital opera de uma maneira diferente, alguns em forma escrita, outros em sistemas tecnológicos próprios e ainda aqueles híbridos entre físico e digital.

Além disso, os registros médicos são bastante complexos e englobam diferentes informações. Para o projeto, foi delimitado até onde o sistema iria agir, focando em resolver a problemática enfrentada pelos profissionais de saúde e auditores detalhada anteriormente. A solução tem implementada todas as funcionalidades da coluna "Must Have" e a primeira da coluna "Should Have" do método MoSCoW (figura 17).



Figura 17: Método de priorização MoSCoW

Sendo assim, ainda existe a oportunidade de expandir a solução adicionando algumas das funcionalidades pensadas, mas que não foram priorizadas durante esse trabalho. Outra possibilidade é a realização de testes em ambiente real do aplicativo, para gerar métricas e fazer ajustes conforme a necessidade dos profissionais.

8. Referências bibliográficas

- [1] ZG Soluções. **Auditoria Hospitalar: o que é, tipos de auditoria e atuação profissional**, 10 set. 2019. Disponível em: <<https://zgsolucoes.com.br/blog/auditoria-hospitalar/>>. Acesso em: 21 mar. 2021.
- [2] OLIVEIRA, R. L. DE; OLIVEIRA, E. M. N. DE. Auditoria médica em hospital geral de médio porte: análise das glosas hospitalares. **Revista de Administração em Saúde**, v. 20, n. 78, 1 abr. 2020. Disponível em: <<https://cqh.org.br/ojs-2.4.8/index.php/ras/article/view/206/325>>. Acesso em: 29 mar. 2021.
- [3] COSTA, L. R. A importância dos registros de enfermagem para auditoria no centro cirúrgico. **Revista Científica Multidisciplinar Núcleo do Conhecimento**, v. 03, n. 02, p. 97–108, 23 mar. 2020. Disponível em: <<https://www.nucleodoconhecimento.com.br/saude/importancia-dos-registros>>. Acesso em: 01 mar. 2021.
- [4] **A importância dos registros cirúrgicos para auditoria médica e para o uso de inteligência artificial - Medical Fair Brasil**. Disponível em: <<https://www.medicalfair-brasil.com.br/pt/noticias/tecnologia/a-importancia-dos-registros-cirurgicos-para-auditoria-medica-e-para-o-uso-de-inteligencia-artificial>>. Acesso em: 01 mar. 2021.
- [5] LEITE, GRACE; SILVEIRA, ANDRÉ; SILVA ELAINE. **PEP - Prontuário Eletrônico do Paciente: vantagens para o atendimento**. Disponível em: <<https://www.pixeon.com/blog/pep-prontuario-eletronico-do-paciente/>>. Acesso em: 29 mar. 2021.
- [6] JUNIOR, LAERTE. **O que são glosas e como evitá-las? Dicas práticas para sua instituição**. Disponível em: <<https://www.pixeon.com/blog/como-evitar-glosas/>>. Acesso em: 30 mar. 2021.
- [7] SANTOS, A. et al. Aplicações em redes de sensores na área da saúde e gerenciamento de dados médicos: tecnologias em ascensão. In: MUCHALUAT-SAADE, D.; PITA, R. (Eds.). **Minicursos SBCAS 2020**. 1.

- ed. [s.l.] SBC, 2020. p. 137–186. Disponível em: <<https://sol.sbc.org.br/livros/index.php/sbc/catalog/download/47/213/434-1?inline=1>>. Acesso em: 1 abr. 2021.
- [8] LOPES, EMELINE; VIANA, EUGENIE. Protocolo de identificação segura. **EBSERH HOSPITAIS UNIVERSITÁRIOS FEDERAIS**. Disponível em: <<http://www2.ebserh.gov.br/documents/214336/1110036/PRO.NUSEP.006+-+PROTOCOLO+DE+IDENTIFICA%C3%87%C3%83O+SEGURA.pdf/c0d3a32b-b5b8-4bde-8389-355f17d290c3>>. Acesso em: 1 abr. 2021.
- [9] BRITO, M. DE F. P. **Avaliação do processo de identificação do paciente em serviços de saúde**. Doutorado em Enfermagem Fundamental—Ribeirão Preto: Universidade de São Paulo, 26 jan. 2016. Disponível em: <<https://teses.usp.br/teses/disponiveis/22/22132/tde-07012016-110042/publico/MARIADEFATIMAPAIVABRITO.pdf>>. Acesso em: 1 abr. 2021.
- [10] CPPAS. **Segurança do Paciente: identificação do usuário**. Disponível em: <<http://www.saude.df.gov.br/wp-conteudo/uploads/2018/04/4.-Seguranca-do-Paciente-identificacao-do-usuario.pdf>>. Acesso em 1 abr. 2021.
- [11] ALEXANDRE MALAFAIA. **Equipamentos Médicos Hospitalares [Guia Completo]**. Disponível em: <<https://www.mobiloc.com.br/blog/equipamentos-medicos-hospitalares/>>. Acesso em: 1 abr. 2021.
- [12] **Medicalchain**. Disponível em: <<https://medicalchain.com/en/>>. Acesso em: 1 abr. 2021.
- [13] **O que é um prontuário eletrônico e como ele funciona? Meu Consultório**, 27 jun. 2019. Disponível em: <<https://www.meuconsultorio.com/blog/o-que-e-prontuario-eletronico/>>. Acesso em: 4 abr. 2021
- [14] **Hybrid Apps: An overview of Advantages, Limitations & Consequences for your Testing Phases**. Disponível em: <<https://www2.stardust-testing.com/en/blog-en/hybrid-apps>>. Acesso em: 4 abr. 2021.

- [15] APPLE. **HealthKit | Apple Developer Documentation**. Disponível em: <<https://developer.apple.com/documentation/healthkit>>. Acesso em: 4 abr. 2021.
- [16] LEITE, VITOR. **Blockchain: uma explicação simples - Fala, Nubank**. Disponível em: <<https://blog.nubank.com.br/o-que-e-blockchain/>>. Acesso em: 11 abr. 2021.
- [17] MOURÃO, A. D. **Impactos da Implantação do Prontuário Eletrônico do Paciente sobre o Trabalho dos Profissionais de Saúde da Prefeitura Municipal de Belo Horizonte**. Disponível em: <https://www.aedb.br/seget/arquivos/artigos07/56_SEGET.pdf>. Acesso em: 11 abr. 2021.
- [18] ALCÂNTARA, A. **A Análise Essencial de Sistemas**. Disponível em: <<http://www.anisio.eti.br/index.php/sistemas-de-informacao-menuvertical/conceito-de-sistema/item/58-a-analise-essencial-de-sistemas>>. Acesso em: 26 jun. 2021.
- [19] **About | Challenge Based Learning**. , [s.d.]. Disponível em: <<https://www.challengebasedlearning.org/about/>>. Acesso em: 26 out. 2021
- [20] **Introduction to smart contracts**. Disponível em: <<https://ethereum.org>>. Acesso em: 27 out. 2021.
- [21] CAMARGO, R. **Quais os benefícios de criar um diagrama de contexto?** Text.homepage.institucional. Disponível em: <<https://robsoncamargo.com.br/blog/Quais-os-beneficios-de-criar-um-diagrama-de-contexto>>. Acesso em: 28 out. 2021.
- [22] **Core NFC | Apple Developer Documentation**. Disponível em: <<https://developer.apple.com/documentation/corenfc>>. Acesso em: 31 out. 2021.
- [23] **web3.js - Ethereum JavaScript API**. [s.l.] ChainSafe, 2021.

- [24] **O que é API?** Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>>. Acesso em: 31 out. 2021.
- [25] **Echos - Jornada do usuário em UX: o que é e sua importância.** Disponível em: <<https://escoladesignthinking.echos.cc/blog/2020/09/jornada-do-usuario-ux/>>. Acesso em: 31 out. 2021.
- [26] **Themes - iOS - Human Interface Guidelines - Apple Developer.** Disponível em: <<https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/>>. Acesso em: 1 nov. 2021.
- [27] **ISO/IEC 14443.** , 31 jul. 2020. (Nota técnica). Disponível em <https://en.wikipedia.org/wiki/ISO/IEC_14443>. Acesso em: 1 nov. 2021.
- [28] **Xcode | Apple Developer Documentation.** Disponível em: <<https://developer.apple.com/documentation/xcode>>. Acesso em: 2 nov. 2021.
- [29] **SwiftUI | Apple Developer Documentation.** Disponível em: <<https://developer.apple.com/documentation/swiftui/>>. Acesso em: 2 nov. 2021.
- [30] **Firebase Authentication | Firebase Documentation.** Disponível em: <<https://firebase.google.com/docs/auth?hl=pt-br>>. Acesso em: 6 nov. 2021.
- [31] **POST - HTTP | MDN.** Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods/POST>>. Acesso em: 7 nov. 2021.
- [32] **Message-Digest Algorithm 5 - an overview | ScienceDirect Topics.** Disponível em: <<https://www.sciencedirect.com/topics/computer-science/message-digest-algorithm-5>>. Acesso em: 7 nov. 2021.
- [33] **Stellnberger, Martin. Document Certification through the Blockchain.** Disponível em: <<https://www.martinstellnberger.co/document-certification-through-the-blockchain>>. Acesso em: 7 nov. 2021.
- [34] **SDK vs. API: What's the Difference? | IBM.** Disponível em: <<https://www.ibm.com/cloud/blog/sdk-vs-api>>. Acesso em: 9 nov. 2021.

- [35] **NFC Tools**. Disponível em: <<https://apps.apple.com/br/app/nfc-tools/id1252962749>>. Acesso em: 21 nov. 2021.
- [36] **The API Design Platform and API Client**. Disponível em: <<https://insomnia.rest/>>. Acesso em: 21 nov. 2021.
- [37] **Ganache**. Disponível em: <<https://trufflesuite.com/ganache>>. Acesso em: 21 nov. 2021.