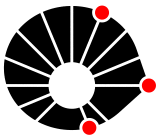

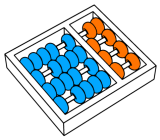

Adequando aplicativos móveis de saúde à Lei Geral de Proteção de Dados: Um caso prático

I. E. Ribeiro *J. F. Borin*

Relatório Técnico - IC-PFG-20-38
Projeto Final de Graduação
2020 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Adequando aplicativos móveis de saúde à Lei Geral de Proteção de Dados: Um caso prático

Ignácio Espinoso Ribeiro[1], Juliana Freitag Borin[2]

[1] Instituto de Computação Universidade Estadual de Campinas (UNICAMP), Caixa Postal 6176
13083-970 Campinas-SP, Brasil

2 Instituto de Computação Universidade Estadual de Campinas (UNICAMP), Caixa Postal 6176
13083-970 Campinas-SP, Brasil

Resumo. Com o início da vigência da Lei Geral de Proteção de Dados (LGPD) no Brasil, diversas organizações iniciaram um processo de readequação de seus sistemas de informação. Neste contexto, sistemas que manipulam dados de saúde de pacientes contém informações extremamente sensíveis. Este projeto visa aprimorar o aplicativo móvel de saúde Controle de DuploJ a partir da adequação do sistema à LGPD, ao mesmo tempo em que expande suas funcionalidades para suportar outros casos de uso. Para tanto, foram removidos campos de dados não essenciais e implementado um mecanismo de autenticação, tanto via API externa, como por biometria. O produto final também permite adicionar e visualizar um cateter de tipo distinto do DuploJ.

Palavras-Chave: Lei Geral de Proteção de Dados, aplicativos móveis, e-health.

1. Introdução

A discussão sobre privacidade de dados no meio da tecnologia ganhou força nos últimos anos e, como resultado dessa discussão, surgiu a Lei Geral de Proteção de Dados (LGPD) [1]. A LGPD, em vigor desde agosto de 2020 discorre sobre como lidar com dados pessoais nos meios digitais por pessoa natural ou por pessoa jurídica, de direito público ou privado, com o objetivo de proteger os direitos fundamentais de liberdade e de privacidade. Dentro da LGPD, um dos tipos de dados mais delicados de se lidar são os dados de saúde, uma vez que podem conter informações desde data de nascimento, endereço e etnia, até o histórico médico completo de um paciente.

Paralelo a este contexto, recentemente foi projetado e desenvolvido o aplicativo Controle de DuploJ [2], para plataforma iOS da Apple, com o objetivo de realizar o controle e análise de dados de pacientes submetidos à inserção de cateter Duplo J. Por se tratar de um aplicativo móvel voltado para a área de saúde, o DuploJ manipula dados sensíveis de pacientes, inclusive nome, idade e sexo. Desse modo, além de espaço para melhorias e expansão de funcionalidades como autenticação de acesso mais robusta e suporte para outros tipos de cateteres, se faz necessária uma reestruturação do aplicativo e sua base de dados como um todo para maior adequação à LGPD. Tal reestruturação envolve alterações em uma base de código com arquitetura já estabelecida, além de migração/adequação dos dados já existentes, visando garantir a privacidade dos dados de pacientes.

2. Justificativa

Com a aproximação da vigência da Lei Geral de Proteção de Dados, muitos artigos surgiram dando enfoque nas adaptações necessárias para adequar aplicações, inclusive aplicativos de celulares, à LGPD. Além das orientações de jornais de grande circulação [3][4], blogs relacionados à tecnologia tornam claras, principalmente:

- a distinção entre dados pessoais e dados pessoais sensíveis;
- definições sobre o tratamento / custódia de dados; e
- responsabilidades de quem detém os dados.

Tendo em vista as informações disponibilizadas nestes meios, na lei em si, e em palestras de especialistas e profissionais atuantes na área, se nota que o sistema atual do App Controle de DuploJ é responsável por diversos dados pessoais sensíveis (como dados étnicos e de saúde), que acabam por ter mais requisitos para se adequar à Lei. Tais requisitos incluem:

- anonimização dos dados coletados, sempre que possível;
- maior controle do acesso aos dados.

Com isso, o aplicativo acaba por partilhar responsabilidade em conjunto com as informações já presentes no prontuário de pacientes, de forma que a redundância de dados deve ser evitada. Tendo em mente tais ponderações, além de aprimorar a autenticação para acesso aos dados, também se faz necessária uma filtragem dos dados atualmente presentes nos registros do aplicativo.

3. Modelo do Sistema

A arquitetura utilizada no desenvolvimento do App DuploJ, a MVVM-C (*Model-View-View Model - Coordinator*), uma versão modificada da arquitetura MVVM (*Model-View-View Model*) auxiliou na extensão das funcionalidades atuais e implementação das novas, já que o *Coordinator* possibilita um maior desacoplamento entre o modelo de dados ou lógica de negócio e as telas do aplicativo.

Relativo à integração do aplicativo com o banco de dados, o modelo do sistema se manteve inalterado, com apenas uma dependência nova adicionada relacionada à camada de autenticação (ver Figura 1), também fornecida pelo Firebase.

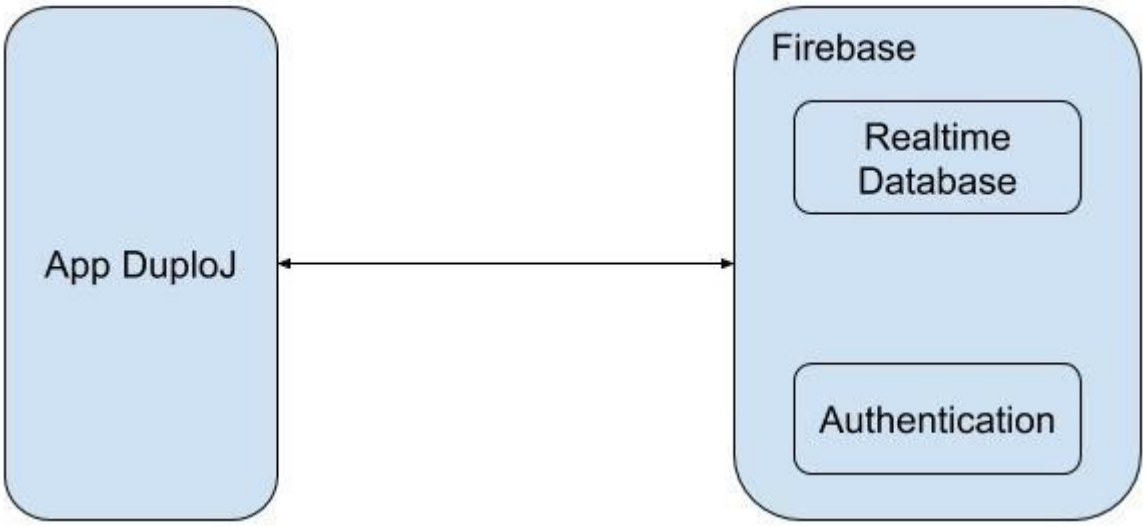

Figura 1 - Modelo do sistema atual do projeto.

4. Metodologia

Primeiramente, uma vez que o trabalho envolveu a alteração de código já em produção, isto é, já disponível ao usuário final, certas precauções foram necessárias. Isto incluiu a adoção do Git Flow, uma convenção que, ao utilizar a ferramenta de versionamento Git, separa as ramificações (*branches*) de versões do aplicativo, mitigando lançamentos e possíveis impactos em produção.

Tendo feito o estudo das adequações necessárias para a LGPD e com a arquitetura readequada, passou a ser necessário um contato frequente com usuários do aplicativo para avaliar as abordagens a serem tomadas, levantando novos requisitos.

Com os novos requisitos levantados, as entregas foram incrementais, seguindo a ordem:

- resolução de erros presentes no código do aplicativo;
- melhorias de impacto de experiência de usuário, mas com poucas alterações visuais e nos fluxos de uso;
- atualização do Banco de Dados, seguindo os novos requisitos;
- atualização da lógica de negócio atual para permitir a expansão aos novos casos de uso;
- modificações na interface de usuário para incorporar as novas funcionalidades;
- implementação das funcionalidades de autenticação pelo Firebase e por biometria.

5. Desenvolvimento

Com os requisitos em mãos, após realizar testes de usabilidade foi dada prioridade a melhorias de curto tempo de desenvolvimento, mas de impacto na experiência do usuário final. Na sequência, foram realizadas mudanças de caráter mais profundo nas funcionalidades e características do aplicativo.

5.1 Correção de erros e melhorias de qualidade de vida

A versão inicial do projeto tinha uma falha que, ao realizar o primeiro *login*, não era possível acessar as demais telas do aplicativo, de forma que alguém utilizando pela primeira vez não conseguia, por exemplo, visualizar a lista de pacientes ou as estatísticas. A solução acabou por revelar algumas pendências na arquitetura, indicando que nem todas as dependências entre componentes estavam sendo corretamente inicializadas. A correção dessa falha foi essencial para que, posteriormente, fosse implementada a autenticação através do Firebase e pela biometria.

Em conjunto com a falha mencionada, também se revelou uma necessidade do usuário em encontrar pacientes específicos na lista. Isto ocorre pois, com uma lista crescente de pacientes, o processo manual de busca acaba por ser exaustivo, se fazendo necessária a implementação de um mecanismo de busca textual. Tal mecanismo foi implementado, facilitando a busca de pacientes em específico e podendo ser observado na parte superior da Figura 2.

5.2 Atualização do Banco de Dados

Dentro do trabalho técnico de adequação à LGPD, se fez necessária uma adaptação do banco de dados atual e, para tal, foi realizada uma alteração no modelo de dados, de forma que certos campos fossem removidos para garantir uma maior anonimização dos dados. Os campos removidos foram:

- *Initials* - Como tal campo continha dados das iniciais do paciente, poderia comprometer o anonimato. Além disso, criava também uma redundância da informação, uma vez que tal dado pode ser obtido com auxílio do número do prontuário diretamente na base de dados do hospital.
- *InsertionPhysician / RemovalPhysician* - Neste campo, que armazenava o nome do médico que inseriu/removeu o cateter, o anonimato do médico era diretamente posto em risco, uma vez que era possível estabelecer diretamente entre médico e pacientes.
- *Physician* - Este campo armazenava o nome do médico responsável pela retirada do cateter e, similarmente, consistia em uma potencial brecha de anonimato.

Para suportar essas atualizações, também foram necessárias modificações diretamente no aplicativo, de forma que o fluxo de cadastro de pacientes e a tela de sumário não inclui mais estes campos.

Complementarmente, o banco de dados foi modificado para dar suporte a diferentes cateteres, de forma que inicialmente já foi incluso o DuploJ de longa permanência. Para refletir tais mudanças, o aplicativo também foi atualizado para poder inserir o novo campo, além de agora fornecer uma nova visualização da lista de pacientes, podendo ser notada na parte inferior da Figura 2.

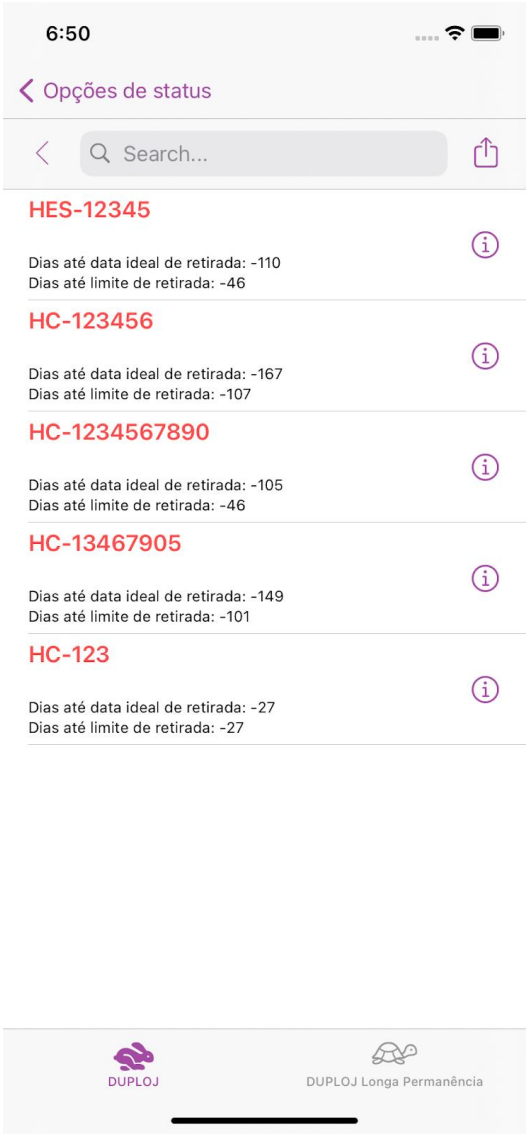

Figura 2 - Lista de Pacientes após as alterações.

5.3 Atualizações de lógica de negócio

Uma pendência na versão inicial do aplicativo era uma falta de checagem em certos dados fornecidos ao preencher a ficha de um novo paciente. Anteriormente, o uso de caracteres ou dados inválidos (e.g. caracteres não numéricos para idade do paciente, ou uso de uma data futura como data de inserção do cateter) era restrito apenas por meio do tipo de input

presente no teclado do *smartphone*, mas não havendo alguma lógica processando tal entrada de dado. Assim, por exemplo, ao tentar inserir a idade de um paciente, aparecia um teclado numérico, mas era possível usar outros tipos de caracteres ao colar algum dado anteriormente presente na área de transferência. Fazer uma validação das entradas do usuário é de grande importância para garantir a validade dos dados do estudo. Com isso, a lógica de negócio foi atualizada para os seguintes campos:

- *Prontuary*: Representa o número do prontuário do paciente. Agora o este valor deve poder sofrer um *cast* para *Int*, de forma que *Strings* normais são invalidadas.
- *Insertion/removal date:* Representa, respectivamente, as datas de inserção e remoção. Agora é impossibilitado o uso de datas inválidas, como datas futuras para inserção ou remoção.
- *YearsOld:* Representa a idade do paciente. Foi definido um valor máximo (150) e também foi implementada uma lógica formal para garantir que a idade será sempre um inteiro positivo.
- *PatientPhoneNumber:* Armazena o número de telefone. Na nova versão do aplicativo, o número de telefone de um paciente é deletado ao inserir a data de remoção do cateter.

5.4 Autenticação

A nova versão do aplicativo inclui autenticação dos usuários. A autenticação inicial foi implementada através do *Firebase Authentication* [5]. Para a nova experiência de uso da tela inicial (Figura 3), os usuários cadastrados no *TestFlight*, plataforma da Apple para *beta testing* de aplicativos, foram registrados no Firebase. No primeiro acesso, a senha deverá ser reconfigurada requisitando uma alteração de senha (ver Figura 4). Tal requisição desencadeia o envio de uma mensagem para o e-mail cadastrado. A principal vantagem dessa abordagem é que a senha pessoal de cada usuário permanece desconhecida para outras pessoas, mas sem tirar as permissões do administrador sobre as contas cadastradas no Firebase

Após realizar o primeiro acesso com o Firebase, o usuário visualiza uma requisição de permissão (Figura 5) para utilizar as ferramentas de biometria do iOS, *Face / Touch ID*, e assim facilitar os próximos acessos ao aplicativo. Caso o aplicativo seja desinstalado ou o usuário não consiga utilizar a biometria (Figura 6), pode voltar a utilizar a senha tradicional para acesso aos dados.

Para cadastro dos usuários, foi utilizado o console do Firebase para registrar os e-mails com acesso ao aplicativo no *TestFlight*. Ao realizar o primeiro acesso, o usuário pode requisitar uma nova senha no aplicativo, recebendo no e-mail cadastrado um *link* para redefinir sua senha.

Figura 3 -Nova tela inicial do aplicativo.

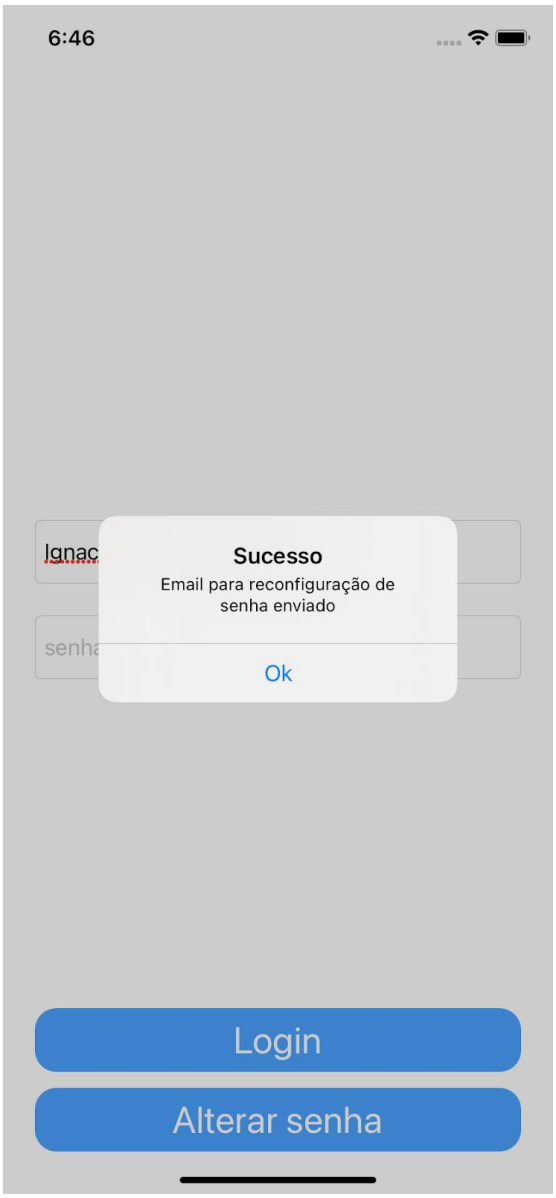

Figura 4 -Tela inicial ao requisitar nova senha.

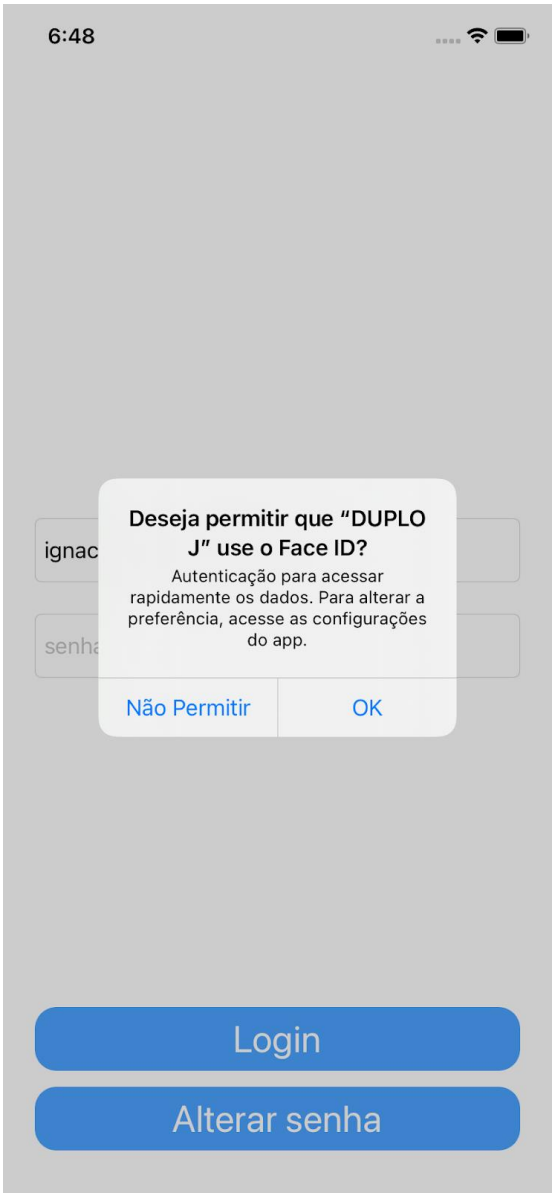

Figura 5 - Requisição de permissão para biometria

Figura 6 - Biometria não reconhecida

6. Limitações e proposta de nova arquitetura

Apesar de terem sido implementados avanços significativos para conformidade com a lei, em especial no que se dispõe ao controle de acesso aos dados, ainda existe espaço para melhoria dentro da aplicação.

Primeiramente, a solução como um todo ainda não foi integrada com os sistemas existentes na universidade, de forma que elementos como banco de dados para o aplicativo e autenticação podem ser suportados por sistemas já implementados pelos respectivos departamentos responsáveis na Unicamp. No caso do banco de dados, a comunicação com o aplicativo pode ser feita através de uma API que reenforce as regras de negócio já estabelecidas. Para a Unicamp, tal API ainda pode ser hospedada pelos sistemas da Faculdade de Ciências Médicas (FCM), de forma que o modelo do sistema pode passar a ser representado pela Figura 7. Já para a autenticação de usuários da comunidade da Unicamp, a autenticação pode ser feita através do mecanismo de autenticação centralizada (*OAuth*) disponibilizado pela CCUEC, resultando na arquitetura representada na Figura 8. Estudos preliminares do uso de *OAuth* no ambiente iOS revelam que já existem bibliotecas [6] capazes de darem o suporte necessário para facilitar a implementação dessa funcionalidade. As vantagens de usar tal mecanismo incluem:

- reutilização de uma autenticação robusta já utilizada em diversos sistemas da universidade;
- usuários da Unicamp podem utilizar o sistema sem ter a necessidade de criar outra senha;
- maiores registros de cada usuário para situações, por exemplo, de auditoria; e

Adicionalmente, pode ser estudada a implementação de uma encriptação do banco de dados (caso migre do Firebase), ou então a criptografia na comunicação com o mesmo.

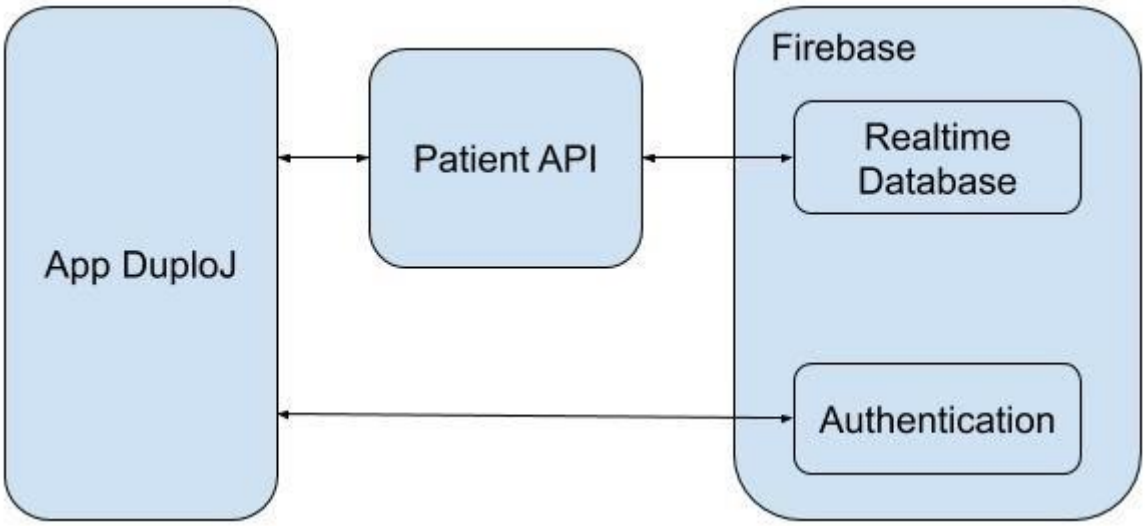

Figura 7 - Modelo do sistema ao adicionar uma API.

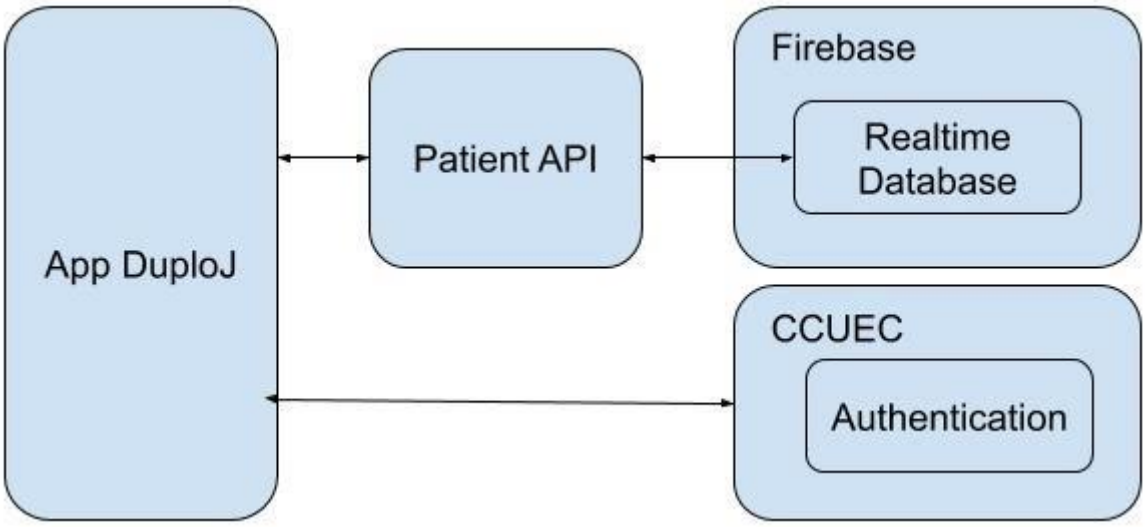

Figura 8 -Modelo do Sistema com API e OAuth.

7. Conclusão

Este projeto teve como objetivo adequar um aplicativo móvel voltado para área da saúde, o APP DuploJ, à Lei Geral de Proteção de Dados (LGPD), que entrou em vigor no Brasil em 2.020. Com base no estudo da LGPD e da arquitetura e código do aplicativo foi possível um avanço significativo na conformidade à LGPD. As alterações implementadas satisfazem aspectos essenciais da Lei, entre eles: i) anonimização, sempre que possível, dos dados dos pacientes e ii) remoção de dados dispensáveis para o usuário final. Adicionalmente, foram corrigidos erros presentes no aplicativo e adicionadas novas funcionalidades requisitadas pelos usuários.

Apesar dos avanços alcançados, ainda existem pontos que podem ser aprimorados, como descrito na seção anterior. Ademais, uma eventual consultoria direta de especialistas na área pode revelar ameaças imprevistas à conformidade da Lei, tanto do escopo do aplicativo como fora dele, uma vez que há uma interseção dos dados com os armazenados no prontuário dos hospitais.

Por outro lado, uma integração total com os sistemas da Unicamp ainda é vital para a manutenção do aplicativo em um maior prazo, uma vez que minimiza o risco ao vazamento ou exposição dos dados, ao mesmo tempo em que se integra melhor à Política de Privacidade estabelecida pela Unicamp.

Referências

[1] http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/Lei/L13709.htm
[2] E. C. Albizzati e J. F. Borin, Controle de Duplo J: Um aplicativo móvel para adição, controle e análise de dados de pacientes submetidos à inserção de cateter Duplo J. Repatório Técnico IC-PFG-20-09, Instituto de Computação, Unicamp, 2020. Disponível em: https://www.ic.unicamp.br/~reltech/PFG/2020/PFG-20-09.pdf
[3] https://www1.folha.uol.com.br/mpme/2020/10/empresas-buscam-se-adaptar-a-lei-que-protege-dados-de-clientes.shtml
[4] https://link.estadao.com.br/noticias/cultura-digital,apps-tentam-se-adaptar-a-lei-de-dados-e-e-por-isso-que-voce-esta-recebendo-notificacoes,70003399721
[5] https://firebase.google.com/docs/auth.
[6] https://github.com/openid/AppAuth-iOS
[7] https://www.unicamp.br/unicamp/noticias/2020/10/06/unicamp-aprova-politica-de-privacidade-de-dados