



UMA INTERFACE ADMINISTRATIVA UTILIZANDO FIREBASE

Mateus Zanini¹, Tiago Possato², Tiago Heineck³

INTRODUÇÃO

Atualmente, grandes volumes de dados são gerados em um intervalo relativamente curto. Sua manipulação demanda de um alto poder de processamento, precisando ser eficiente e escalável. Neste contexto surgiu o NoSQL (*Not only SQL*), que com a proposta de oferecer melhor desempenho, supriu também a necessidade de suporte a tipos de dados complexos, semi-estruturados ou não-estruturados (Vieira et al., 2012).

Neste trabalho foi desenvolvida uma aplicação *web* de gerenciamento de usuários, que tem por objetivo servir como interface base para outros sistemas. Foram implementadas duas funcionalidades: (1) o gerenciamento de usuários (com três níveis de acesso) e (2) o gerenciamento de empresas. Avaliamos de maneira empírica o Firebase, uma aplicação da tecnologia de NoSQL disponível pela nuvem da Google(GOOGLE, 2016c). Uma interface gráfica foi desenvolvida para se adequar a qualquer tipo de dispositivo, independente do tamanho da tela. O objetivo principal foi avaliar de maneira experimental, utilizando um projeto do mundo real, a utilização desse novo paradigma de banco de dados.

¹ Aluno do IFC – Campus Videira. Curso de Bacharelado em Ciência da Computação.
E-mail: mateus.zanini.gl@gmail.com

² Aluno do IFC – Campus Videira. Curso de Bacharelado em Ciência da Computação.
E-mail: tiago.possato@yahoo.com.br

³ Professor orientador do IFC – Campus Videira. Curso de Bacharelado em Ciência da Computação.
E-mail: tiago.heineck@ifc.edu.br



PROCEDIMENTOS METODOLÓGICOS

O Firebase, é uma solução completa de *back-end* para desenvolvimento tanto *mobile* quanto *web*. É oferecido como um serviço pela Google, sendo hospedado e mantido em seus *datacenters* (AVRAM, 2016). Oferece diversos recursos, sendo que neste trabalho foram utilizados: autenticação, armazenamento em banco de dados e hospedagem.

Um banco de dados NoSQL, baseado em documentos, é utilizado para armazenar os dados como JSON (*JavaScript Object Notation*) e sincronizar com todos os clientes conectados (GOOGLE, 2016c). Quando os dados são alterados em um dispositivo eles são enviados para o Firebase e então para todos os dispositivos conectados. Caso existam dispositivos *offline* neste momento os mesmos serão atualizados com a última versão dos dados logo após a conexão com a Internet, dando a ideia de banco de dados em tempo real (Firebase Realtime Database).

O Firebase Realtime Database fornece uma linguagem de regras baseada em expressão com sintaxe semelhante ao *JavaScript*. As Firebase Realtime Database Rules são regras que determinam quem tem acesso de leitura e gravação ao banco de dados, como os dados são estruturados e quais índices existem. Essas regras são aplicadas automaticamente a cada requisição, que só será concluída se as regras permitirem. Combinando-as com os serviços de autenticação, é possível definir quem tem acesso a quais dados e proteger as informações contra o acesso não autorizado (GOOGLE, 2016a).

Conforme a Google (2016b) explica, a maioria dos aplicativos precisa saber a identidade do usuário, permitindo que ele salve seus dados de maneira segura para que consiga a mesma experiência em todos os seus dispositivos. Por isso, o Firebase implementa o recurso de autenticação, chamado Firebase Authentication. Ele fornece todas as ferramentas necessárias para autenticar um usuário, seja através de e-mail e senha (método utilizado na implementação deste

trabalho) ou provedores populares de identidade, como Google, Facebook, Twitter e GitHub.

Sendo assim, a aplicação desenvolvida foi hospedada no recurso de Hosting do Firebase, que é uma CDN (*Content Delivery Network*) distribuída globalmente para servir aplicações *web*. A distribuição deste conteúdo é feita de forma segura utilizando o protocolo HTTPS e fornece uma maneira rápida e fácil de implementar aplicativos *web* em nível de produção (GOOGLE, 2016d).

A modelagem dos dados para bancos NoSQL, especificamente para o Firebase, é um processo que requer quebras de paradigmas. Segundo Narayanan (2015), para construir aplicações verdadeiramente escalonáveis, a desnormalização é quase um requisito. Antes de iniciar seu desenvolvimento, é preciso observar como os dados serão estruturados, tanto para facilitar o emprego das regras do Firebase, quanto para otimizar as consultas realizadas. Dessa forma, uma estrutura de árvore bem projetada é crucial para a aplicação.

A modelagem de dados deste trabalho foi feita com base nos objetos usados na interface, tomando cuidado para evitar sobrecarga no sistema. A aplicação possui somente dois objetos, que estão relacionados: Usuários e Empresas. Assim, cada empresa possui um nó ‘usuarios’, contendo o uid (identificador único universal) de cada usuário que tem acesso àquela empresa. Da mesma forma, cada usuário possui um nó ‘empresas’ com o uid das empresas que ele tem acesso. Essa duplicação de dados permite otimização nas consultas. Conforme Narayanan (2015) defende, o espaço em disco é barato, mas o tempo de um usuário não.

Como a aplicação é baseada na *web*, foi utilizado o *kit* de desenvolvimento de *software* (SDK) para JavaScript do Firebase. No desenvolvimento *front-end* foram aplicados o Materialize, que é um *framework* responsivo baseado em Material Design, e o JQuery, uma biblioteca JavaScript que auxilia no desenvolvimento, tornando-o mais rápido e simples.



RESULTADOS E DISCUSSÕES

O sistema desenvolvido atende à todos os requisitos propostos. É possível gerenciar usuários e empresas, visualizando os dados que cada usuário tem acesso com base em seu perfil, podendo ser um administrador, gerente ou operador. No gerenciamento de usuários é possível: (1) cadastrar usuários; (2) visualizar os usuários cadastrados; (3) ativar/inativar usuários e; (4) editar as informações dos usuários, como nome e e-mail, tipo de perfil e seu vínculo com empresas.

Na parte de empresas, funcionalidades semelhantes estão presentes: (1) cadastrar empresas; (2) visualizar as empresas cadastradas; (3) ativar/inativar empresas; (4) editar as informações cadastrais das empresas e; (5) visualizar quais usuários estão vinculados com determinada empresa.

Como trata-se de um sistema sem auto-cadastro, no momento que um novo usuário é criado, uma senha aleatória é gerada e um e-mail é enviado para ele contendo um *link* de redefinição de senha. Um processo parecido ocorre quando o usuário utiliza a funcionalidade “Esqueci minha senha”, disponível na tela de *login*. Neste caso apenas o *link* é enviado.

A aplicação está funcionando adequadamente em telas pequenas, como *smartphones*, sendo que os elementos se adaptam de forma responsiva ao seu tamanho. Para evitar sobrecarga de informações, alguns elementos menos importantes são ocultados quando o sistema é utilizado em telas menores, dando maior ênfase nas operações consideradas primárias.

Alguns problemas que podem ser encontrados ao utilizar o banco de dados NoSQL do Firebase devem ser mencionados, como (JAMIN, 2016):

- O uso de paginação nas consultas ao banco de dados não é possível pois o tamanho da matriz de retorno não é especificado. É possível limitar a quantidade de registros que serão retornados, porém não de forma ordenada;



- A migração de dados não é tão fácil quanto em bancos SQL, sendo necessário efetuar o download de um arquivo JSON com todos os dados do banco de dados;
- O Firebase não oferece uma forma de realizar consultas complexas comparadas com as que são encontradas em bancos SQL.

Com o uso de NoSQL é possível perceber uma maior facilidade e flexibilidade na modelagem dos dados se comparado aos bancos SQL tradicionais, podendo ser forjada conforme a estrutura dos objetos utilizados. De outro lado, implementar os relacionamentos na aplicação é complexo e deve ser feito com cuidado, pois o Firebase não implementa esse recurso nativamente, assim como banco de dados relacional. Para esta garantia, os relacionamentos devem ser programados de forma robusta, possuindo regras bem claras nas Firebase Rules.

CONSIDERAÇÕES FINAIS

O Firebase é uma boa opção para o desenvolvimento de aplicações *web*, pois oferece uma estrutura de *back-end* bastante completa, contendo um ambiente onde é possível gerenciar os usuários, armazenar os dados aplicando regras de segurança, além de hospedar aplicações. O processo de desenvolvimento deste trabalho permitiu perceber o poder dos bancos de dados NoSQL e de serviços de computação em nuvem para o desenvolvimento ágil.

Como futuros trabalhos, algumas melhorias e novas funcionalidades podem ser agregadas, tais como: (1) implementar filtros na exibição de usuários e empresas, facilitando a busca; (2) melhorar o gerenciamento de empresas, com *marketing/mala-direta*, informações sobre cobrança e; (3) tornar o vínculo de usuários à empresas mais rápido (atualmente esta edição ocorre em um usuário por vez). Testes de desempenho também se fazem necessários, como por exemplo, carregar o banco com grandes quantidades de dados, testes em conexões de Internet com taxas



FICE

6^ª FEIRA DE INICIAÇÃO
CIENTÍFICA E EXTENSÃO
05 e 06 de setembro

de transferência reduzida e alta latência e em dispositivos com baixa capacidade de processamento e memória.

REFERÊNCIAS

AVRAM, ABEL. **Google Firebase: back-end completo para aplicações web e mobile.** 2016. Disponível em: <<https://www.infoq.com/br/news/2016/07/google-firebase>>. Acesso em: 24 abril 2017.

GOOGLE. **Entender as Firebase Realtime Database Rules.** 2016a. Disponível em: <<https://firebase.google.com/docs/database/security>>. Acesso em: 24 abril 2017.

GOOGLE. **Firebase Authentication.** 2016b. Disponível em: <<https://firebase.google.com/docs/auth>>. Acesso em: 25 abril 2017.

GOOGLE. **Firebase Realtime Database.** 2016c. Disponível em: <<https://firebase.google.com/docs/database>>. Acesso em: 25 abril 2017.

GOOGLE. **Firebase Hosting.** 2016d. Disponível em: <<https://firebase.google.com/docs/hosting>>. Acesso em: 25 abril 2017.

JAMIN, Baptiste. **Reasons Not To Use Firebase.** 2016. Disponível em: <<https://crisp.im/blog/why-you-should-never-use-firebase-realtime-database>>. Acesso em: 24 abril 2017.

NARAYANAN, Anant. **Denormalizing Your Data is Normal.** 2015. Disponível em: <<https://firebase.googleblog.com/2013/04/denormalizing-your-data-is-normal.html>>. Acesso em: 25 abril 2017.

VIEIRA, Marcos Rodrigues et al. **Bancos de Dados NoSQL: conceitos, ferramentas, linguagens e estudos de casos no contexto de Big Data.** Simpósio Brasileiro de Bancos de Dados, 2012.