

Implementação de um Sistema de Quiz no Aplicativo WebLectures

G. Araújo

R. Azevedo

Technical Report - IC-PFG-19-01 - Relatório Técnico
June - 2019 - Junho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Implementação de um Sistema de Quiz no Aplicativo WebLectures

Guilherme Ianhes Martins de Araújo¹, Rodolfo Jardim de Azevedo²

¹Aluno de graduação do Instituto de Computação Universidade Estadual de Campinas (UNICAMP),Campinas-SP, Brasil. g155616@dac.unicamp.br

²Professor Doutor do Instituto de Computação Universidade Estadual de Campinas (UNICAMP),Campinas-SP, Brasil.rodolfo@ic.unicamp.br

Resumo: Este trabalho é um relatório técnico de um projeto final de graduação, visando o detalhamento da construção de um sistema de quizzes para o aplicativo WebLectures. O objetivo deste projeto foi integrar a metodologia Peer Instruction (FAGEN *et al*,2000) de ensino ao aplicativo, possibilitando ao professor usuário criar questões de múltipla escolha a serem respondidas por seus alunos em uma página WEB, tanto durante a aula quanto posteriormente. O aplicativo, então, integraria os resultados dessas questões a aula do professor através do *whiteboard*. As principais ferramentas utilizadas neste projeto foram: Android Studio para criação do código e testes (Debug com simulador), MPAndroidChart para criação dos gráficos de barras para o resultado das questões e Firebase, plataforma na qual o Banco de Dados foi implementado. Este projeto contemplou a criação (reserva de identificador único, criação da estrutura, interações com o banco de dados, controle de concorrência, etc.) e finalização (busca no banco de dados, geração dos resultados em gráfico de barras) de quizzes criados pelo aplicativo WebLectures.

Palavras-Chave: Peer Instruction, Quiz, aplicativo Android, Banco de Dados, metodologia de ensino.

SUMÁRIO

| | |
|---|----|
| 1. INTRODUÇÃO | 3 |
| 2. JUSTIFICATIVA | 4 |
| 3. OBJETIVOS | 5 |
| 4. DESENVOLVIMENTO DO TRABALHO | 6 |
| 5. FERRAMENTAS UTILIZADAS | 8 |
| 5.1 Android Studio | 8 |
| 5.2 MPAndroidChart | 8 |
| 5.3 Firebase e Firestore | 8 |
| 6. RESULTADO DA IMPLEMENTAÇÃO EM DETALHES | 9 |
| 6.1 Criação de um Quiz | 9 |
| 6.1.1 Botão de Criação do Quiz | 9 |
| 6.1.2 Janela “Criar Quiz” | 10 |
| 6.1.3 Reserva de um Identificador Único | 10 |
| 6.2 Finalização de um Quiz | 12 |
| 6.2.1 Botão de Finalização do Quiz | 12 |
| 6.2.2 Janela “Finalizar Quiz” | 12 |
| 6.2.3 Tratamento dos Resultados | 13 |
| 7. LIMITAÇÕES DA SOLUÇÃO | 15 |
| 8. CONCLUSÃO | 16 |
| 9. REFERÊNCIAS BIBLIOGRÁFICAS | 17 |

1. INTRODUÇÃO

O WebLectures consiste em um aplicativo de uso acadêmico voltado para a gravação e exposição de aulas em geral de forma que o conteúdo desta seja facilmente publicado futuramente em páginas na web. A gravação das aulas se dá, resumidamente, na captura sincronizada: da voz do professor durante a exposição do conteúdo, do slide utilizado pelo professor e de quaisquer anotações feitas pelo professor no próprio slide ou em um quadro branco virtual. Nota-se que a gravação dispensa a captura de vídeo o que torna o resultado final da exportação da aula significativamente mais acessível.

Em linhas gerais o aplicativo depende de um tablet no qual o professor da disciplina possa carregar os slides de sua aula e escrever nesses suas notas de aula, exemplos, correções e explicações. Com o uso de um projetor e um telão, o professor consegue simultaneamente ministrar a aula para seus alunos na sala e gravar a aula para necessidades futuras. O WebLectures é, portanto, uma importante aplicação da tecnologia no aprimoramento do ensino, trazendo benefícios para alunos e professores com mais de mil gravações de aulas já realizadas.

Considerando os benefícios do uso do aplicativo WebLectures, seria interessante uma integração do aplicativo com a metodologia Peer Instruction, queconsiste na inclusão de questões criadas com a intenção de fazer o aluno participar ativamente da aula e encontrar suas dificuldades com o material (FAGEN *et al*,2000).Essa metodologia já apresentou melhorias nos resultados dos alunos quando aplicada a aulas de física na universidade de Harvard (CROUCH; MAZUR, 2001)como também quando aplicada a aulas de Sistemas de Informações na Universidade Estadual de Campinas (CACEFFO *et al*, 2018).

2.JUSTIFICATIVA

A implementação de um sistema de quizzes integrado ao aplicativo WebLectures pode trazer grandes benefícios à aplicação da metodologia Peer Instruction, visto que pode otimizar o tempo consumido em sala de aula devido à aplicação das questões. Ainda, o aplicativo seria uma importante ferramenta para a aplicação do Concept Inventory desenvolvido para a disciplina de Algoritmos e Programação de Computadores pela Universidade Estadual de Campinas (CACEFFO *et al*, 2016).

3. OBJETIVOS

O objetivo deste Projeto Final de graduação foi desenvolver um sistema para criação de quizzes no aplicativo WebLectures intuitivo para professores, funcional para a metodologia Peer Instruction, possibilitando que alunos em sala de aula ou em casa possam responder às questões aplicadas. Ainda, se fez fundamental que os resultados dos quizzes fossem apresentados aos professores e alunos de forma clara e concisa.

4. DESENVOLVIMENTO DO TRABALHO

Foi introduzido no canto superior esquerdo do aplicativo um botão com o rótulo “Quiz”, responsável por realizar todas as ações necessárias para criação, manutenção e finalização de uma questão feita pelo professor. Para isso, o seu funcionamento muda dependendo do estado do programa. Inicialmente, enquanto nenhum quiz estiver aberto, o botão abre uma janela de Criação de Quizzes. Nela, devem ser escolhidos um nome e um número de alternativas para a questão a ser criada. Vale lembrar que não é necessário informar o enunciado da questão ou o conteúdo de cada alternativa, estes devem estar previamente contidos no conjunto de slides da aula.

Assim que um quiz é criado, o aplicativo reserva em um banco de dados (Firebase) uma estrutura referente para esse. Essa estrutura contém campos para o nome do quiz, número de alternativas, número total de respostas e quantidade de respostas para cada alternativa possível. A reserva do quiz retorna para o usuário um identificador único (ID), o qual deve ser usado pelos alunos para efetuar a resposta em uma página web. Essa reserva foi feita com controle de concorrência, que será melhor explicado na sessão de Resultados da Implementação em Detalhes.

Com o fim da reserva dessa estrutura, o aplicativo emite um *toast* (breve mensagem padrão do sistema Android que aparece no canto inferior da tela) informando que: ou um quiz com certo número de alternativas foi reservado com sucesso através de certo ID, ou por algum problema de conexão com o servidor, o quiz não pode ser criado. Quando a reserva é efetuada com sucesso, a funcionalidade do botão “Quiz” muda. Agora, quando pressionado, abre uma nova janela de Finalização de Quizzes, contendo o título do quiz, seu ID e o link da página web que deve ser utilizada pelos alunos para registrarem suas respostas. Essa janela possui três opções: “voltar” que simplesmente fecha a janela e retorna para a apresentação da aula, sem nenhum efeito sobre o quiz; “anular quiz” que deleta do aplicativo a referência para o quiz no banco de dados, fazendo com que o botão “Quiz” volte a sua funcionalidade inicial de criação de quizzes; ou, por fim, “gerar resultados” que encerra o período de respostas, gera os resultados em um gráfico no *whiteboard*, também retornando o botão “Quiz” a sua funcionalidade inicial.

Quando os resultados de um quiz são gerados, o aplicativo retorna a mensagem “Resultados do Quiz <ID> recuperados com sucesso”, que indica que não

houve nenhum problema de conexão com o banco de dados. Logo em seguida, o aplicativo desenha um gráfico de barras em uma nova transparência do *whiteboard* com as respostas dos alunos. O gráfico foi elaborado da forma mais simples possível, visto que sua função é esclarecer ao professor e a classe quais foram os erros mais comuns entre os alunos ou se os alunos apresentam alguma dificuldade com o conteúdo. Por fim, após a geração dos resultados, o professor pode utilizar novamente o botão “Quiz” para criar outra questão.

5. FERRAMENTAS UTILIZADAS

5.1 Android Studio

O Android Studio é um IDE para desenvolvimento de aplicativos Android baseado no IntelliJ IDEA. Neste projeto todo o código foi escrito dentro deste ambiente assim como todos os testes foram realizados através das suas interfaces de debug. Uma grande contribuição do Android Studio para o projeto foi a facilidade proporcionada pela possibilidade de se testar os aplicativo em diversos dispositivos Android conectados ao computador pelo USB.

5.2 MPAndroidChart

MPAndroidChart é uma biblioteca open source para a criação e visualização de gráficos para aplicativos Android. Essa ferramenta foi utilizada somente para organização dos dados e criação dos gráficos de barras com as respostas dos quizzes. Esta biblioteca foi escolhida para esse projeto, pois apesar de não muito intuitiva, a alta customização dos gráficos facilitou a clareza das respostas dos quizzes.

5.3 Firebase e Firestore

Firestore é um banco de dados não relacional da Firebase, flexível e escalonável para desenvolvimento de dispositivos móveis, webs e servidores. O funcionamento se da, resumidamente, através de documentos (conjuntos de strings, números, etc) e coleções desses documentos que são armazenados no banco de dados, no qual existe a possibilidade de consulta, leitura e escrita entre diferentes sistemas para o mesmo banco de dados.

O Firestore foi escolhido para este projeto por algumas razões. A primeira delas foi sua compatibilidade com aplicativos Android. A API do Firestore para o Android é muito bem documentada e oferece diversas ferramentas de acesso assíncrono ao banco de dados. Ainda, o Firestore oferece suporte tanto para aplicativos mobile (Android no caso deste projeto) quanto web, o que se encaixou muito bem, visto que as questões são criadas pelo dispositivo Android do professor enquanto as respostas devem ser enviadas pelos alunos em qualquer dispositivo através de uma página web. Por fim, o Firestore oferece algumas possibilidades para controle de concorrência, que são fundamentais para o projeto, visto que vários professores podem criar várias questões ao mesmo tempo, o que possibilitaria duplicações dos identificadores únicos.

6. RESULTADO DA IMPLEMENTAÇÃO EM DETALHES

6.1 Criação de um Quiz

6.1.1 Botão de Criação do Quiz

O botão “Quiz” (destacado na Figura 1) é responsável pela criação de um quiz. Foi tomada a decisão de colocá-lo junto ao botão exportar, visto que o menu semicircular se dedica a funções majoritariamente relacionadas aos slides e ao “whiteboard”. Quando o botão é pressionado enquanto não exista um quiz ativo, o aplicativo chama uma função chamada *showQuizDialog()*, a qual abre uma nova janela do tipo *AlertDialog* para a criação do quiz (Figura 2).

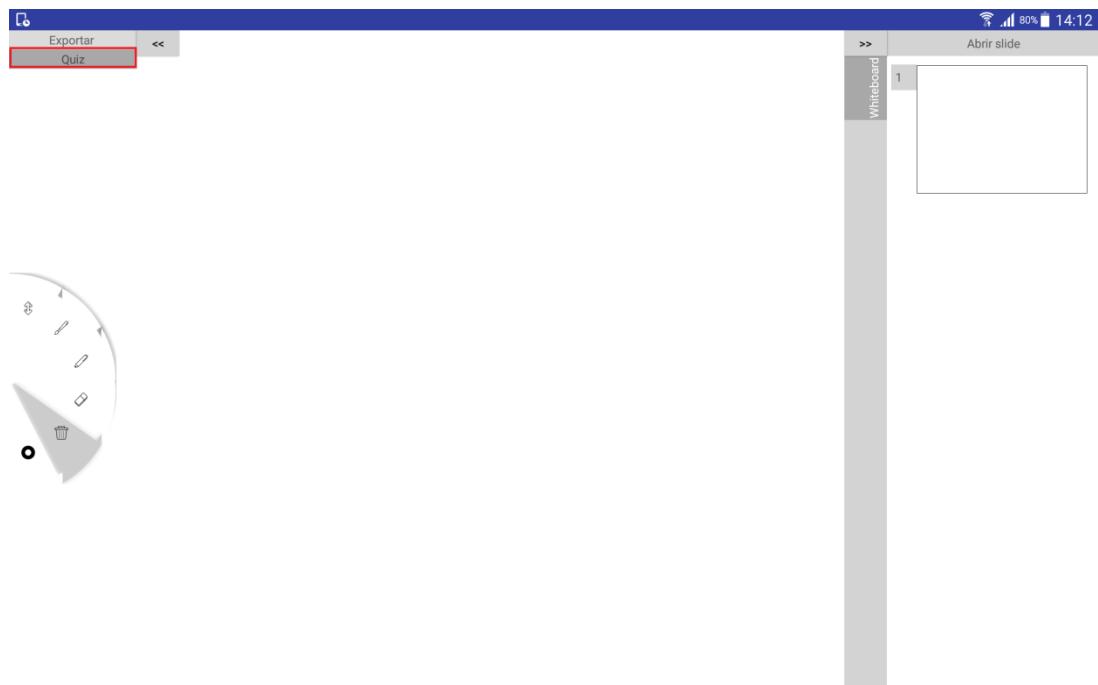


Figura 1: tela inicial do aplicativo WebLectures com destaque ao botão “Quiz”.

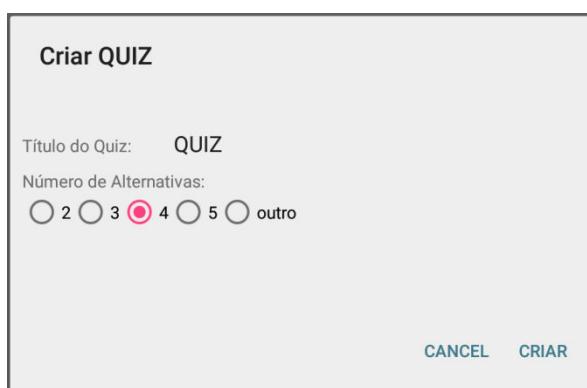


Figura 2: *AlertDialog* para a criação de um quiz.

6.1.2 Janela “Criar Quiz”

O menu de criação de um quiz requer o preenchimento de dois campos pelo usuário. O primeiro delas é um campo de texto editável que espera a entrada do título do quiz a ser criado. Este campo é automaticamente preenchido com a palavra “QUIZ”, caso o usuário não opte por escolher um nome específico. Em segundo lugar, o usuário deve escolher o número de alternativas para a sua questão através de *RadioButtons* com as opções “2”, “3”, “4”, “5” e “outro”.

Caso o usuário queira criar uma questão com mais de cinco alternativas, deve escolher a opção “outro”. Quando selecionada, esta opção torna visível um terceiro campo do tipo texto editável (Figura 3), o qual espera que o usuário digite o número de alternativas desejado. Se qualquer um dos números de dois a cinco for selecionado, o terceiro campo se tornará novamente indisponível. Dois detalhes importantes desse terceiro campo são que: o aplicativo sugere uma opção padrão de 6 alternativas e o número máximo de alternativas foi limitado a 26 devido ao número de letras no alfabeto.

Por fim, após preenchidos os campos descritos nos últimos parágrafos, a opção “CRIAR” iniciará um processo no aplicativo para a reserva de um identificador único e criação de um quiz com as opções escolhidas, enquanto a opção “CANCEL” cancelará a criação do quiz retornando para o *whiteboard* ou os slides da aula.

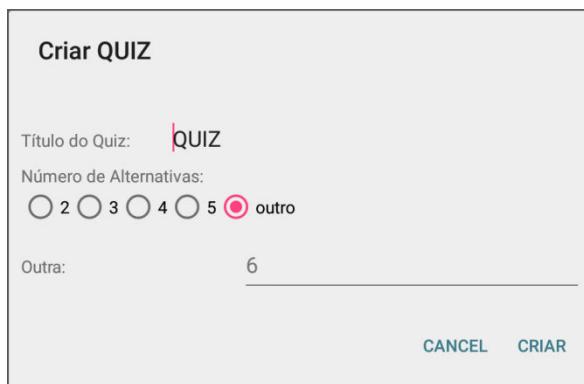


Figura 3: *AlertDialog* para a criação de um quiz quando a opção “outro” é selecionada.

6.1.3 Reserva de um Identificador Único

Assim que o usuário seleciona a opção “CRIAR” da janela de criação de quizzes, o aplicativo inicia uma função chamada *getNewQuizIDwT()* responsável por reservar um identificador único no banco de dados (Firestore) para o novo quiz (a sigla “wT” no código abrevia *withTransaction*, que significa “com Transação” indica que

esta versão da função de reserva de Ids foi implementada com controle de concorrência).

A primeira etapa para a reserva do Id é encontrar a referência para o banco de dados e em seguida para o documento que controla o Id atual:

```
final FirebaseFirestore db = FirebaseFirestore.getInstance();
final DocumentReference mDR = db.document("IDref/currentID");
```

Os identificadores são reservados sequencialmente, o que permite ao banco de dados armazenar somente o próximo número a ser reservado. Em seguida, o aplicativo se utiliza de uma função da API do Firestore chamada *db.runTransaction()*, implementada para atender as necessidades de controle de concorrência, justamente o caso deste aplicativo, visto que é fundamental que o identificador seja único. Essa função, resumidamente, permite ao desenvolvedor executar atomicamente no banco de dados qualquer quantidade de operações de leitura, seguidas de qualquer quantidade de operações de escrita. Note que esse “envelope de operações” impede que dois usuários acessem o Id atual simultaneamente, fazendo com que ambos tenham o mesmo identificador.

Dentro da função de transação, o aplicativo realiza uma leitura do valor do Id no documento de referência para reserva de Ids e em seguida atualiza esse valor para o seu incremento (operação de escrita). Operações com o banco de dados são realizadas de forma assíncrona, ou seja, o aplicativo não pode ficar travado enquanto a comunicação com o servidor não é concluída. Assim, foi feito uso da função *addOnCompleteListener()*, que recebe duas *tasks* a serem possivelmente executadas ao fim da transação. Uma *task* será executada somente se a transação foi realizada com sucesso, enquanto a outra será executada quando ocorrerem erros na comunicação com banco de dados. No último caso o aplicativo escreve na tela uma mensagem do tipo *Toast* informando a falha de comunicação com o servidor.

Quando a transação é completada com sucesso, o *task* a ser executado cria uma estrutura de dados do tipo *HashMap* a ser armazenada no banco de dados. Esta contém um campo “Nome” do tipo *String*, um campo “nAlt” (número de alternativas) do tipo *Int*, um campo “nRes” (número de respostas) do tipo *Int*, e N campos “a_i” do tipo *Int*, para cada uma das N alternativas do quiz. Por fim escreve na tela uma

mensagem do tipo *Toast* informando o usuário que “um quiz com <N> alternativas e ID <X> foi criado com sucesso”.

A partir desse ponto, o aplicativo agora tem uma referência para a estrutura de um quiz no banco de dados.

6.2 Finalização de um Quiz

6.2.1 Botão de Finalização do Quiz

Após os alunos terem respondido o quiz em aberto pela página web, o mesmo botão utilizado para criar o quiz pode ser utilizado para finalizar o mesmo. Quando pressionado com um quiz aberto, esse botão chamará a função *showQuizResultDialog()* que abrirá uma janela do tipo *AlertDialog* com o nome de Finalizar Quiz (Figura 4).

6.2.2 Janela “Finalizar Quiz”

A primeira informação disposta na janela “Finalizar Quiz” é o nome da pergunta escolhido pelo professor ao lado do identificador único em destaque. Abaixo dessa linha tem-se o link da página web que deve ser utilizada pelos alunos para responder ao quiz. Essas informações estão dispostas em grande destaque, visto que devem ser visíveis para todos os alunos na sala de aula, e não somente ao professor.

A janela “Finalizar Quiz” traz três opções ao usuário. A primeira e mais simples dela é a opção de “voltar” ao *whiteboard* ou aos slides da aula, caso o professor deseje utilizá-los antes de finalizar realmente o quiz. Essa opção não altera os resultados do quiz de nenhuma forma. A segunda opção, “ANULAR QUIZ”, apaga a referência do quiz do aplicativo, fazendo com que o botão “QUIZ” do menu volte a função inicial de criar um quiz. É importante ressaltar que, uma vez selecionada, essa opção descarta tudo envolvendo o quiz anulado, inclusive as respostas dos alunos. Por fim, a opção “GERAR RESULTADOS” finaliza o quiz, executando a função *GenerateResults()* e posteriormente apagando a referência do quiz do aplicativo.

Uma observação importante é: mesmo depois de finalizado, um quiz pode ser respondido pelos alunos que estão assistindo a aula pela web, pois apesar do aplicativo perder a referência do quiz, o banco de dados mantém a sua estrutura, fazendo com que a página web ainda possa acessá-lo através do Id.

6.2.3 Tratamento dos Resultados

Usando a referência do quiz, o aplicativo faz uma busca por sua estrutura no banco de dados. Essa busca é, mais uma vez, assíncrona, fazendo necessário o uso da função `addOnSuccessListener()` que recebe um *task* a ser executado assim que a estrutura é recuperada com sucesso. Esse *task* extrai as informações necessárias do documento recuperado, formatando os dados e passando-os para a função `generateGraph()`.

A função `generateGraph()` faz uso da biblioteca MPAndroidChart mencionada na sessão de Ferramentas Utilizadas. O primeiro passo, como as respostas seriam apresentadas em um gráfico de barras, foi criar um *BarDataSet* que recebe os conjuntos de dados para dispô-los em um gráfico de barras. Logo após são feitos vários ajustes com o tamanho das fontes, as cores das barras, as legendas e a formatação dos valores de referência. As linhas de grades são desativadas e o tamanho do gráfico é ajustado à resolução do dispositivo. Por fim inicia-se a renderização do gráfico.

Quando o gráfico é renderizado, uma janela com a mensagem "Resultados do quiz <X> recuperados com sucesso" aparecerá na tela, e em seguida um gráfico de barras contendo as quantidades de respostas dos alunos por alternativa é fixado à última página do *whiteboard* (Figura 5).

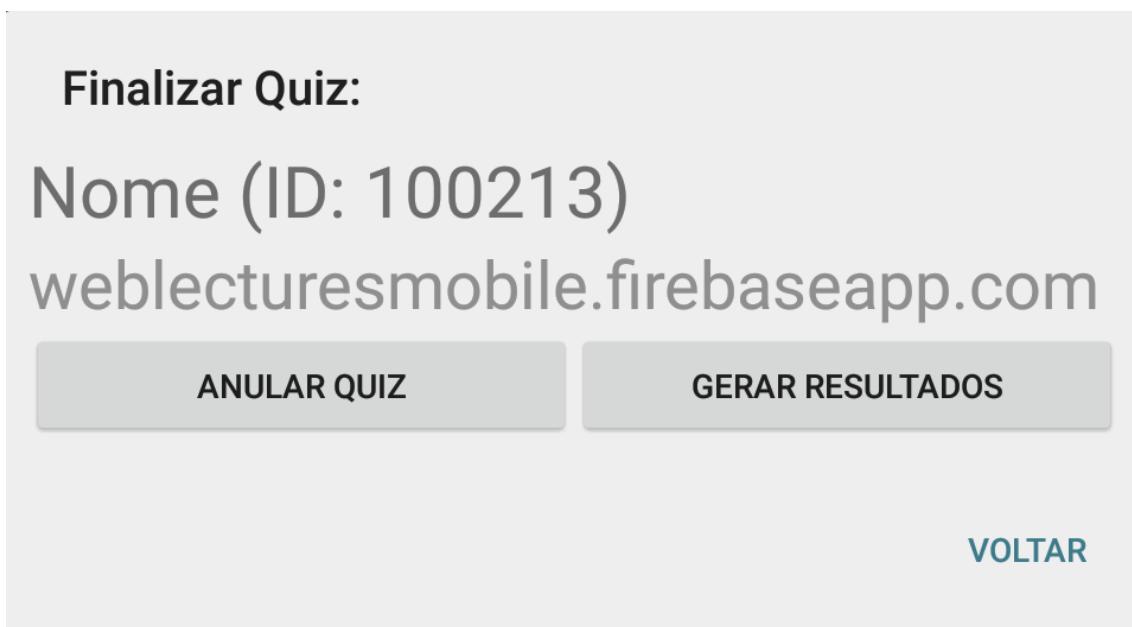


Figura 4: *AlertDialog* para a finalização de um quiz.

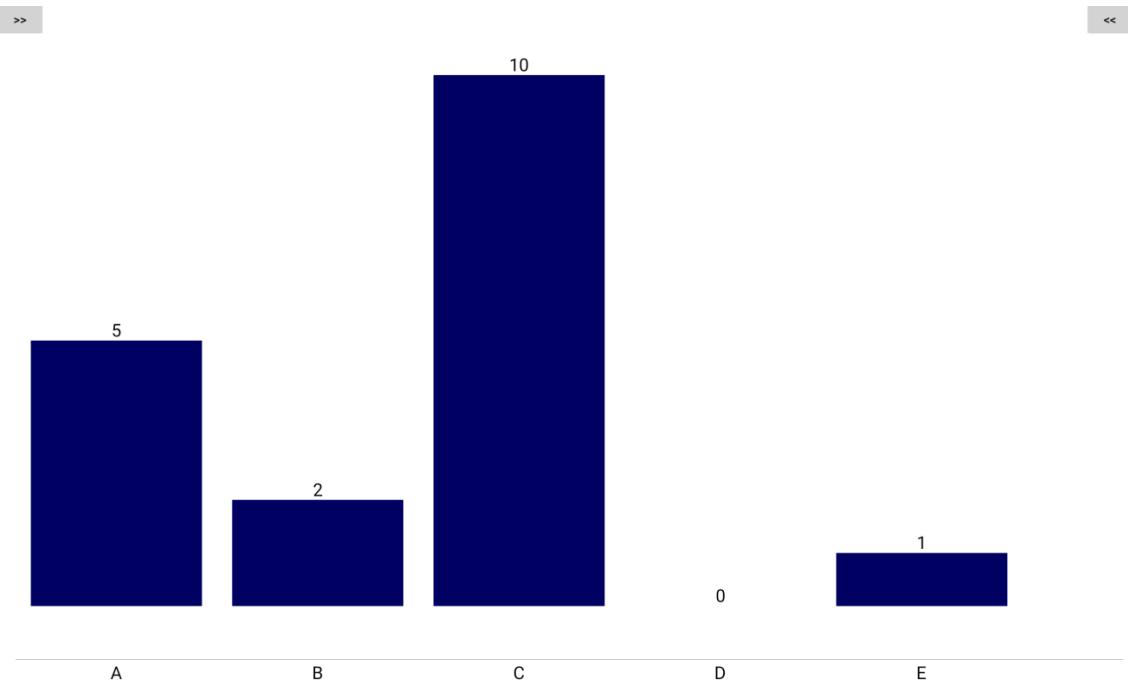


Figura 5: gráfico de barras contendo as quantidades de respostas dos alunos por alternativa.

7. LIMITAÇÕES DA SOLUÇÃO

A primeira limitação deste projeto a ser considerada é a distração causada pelos smartphones em sala de aula. Essa limitação já foi considerada em outros estudos e infelizmente não pode ser resolvida nos limites deste projeto. Uma possível solução seria uma rede própria da instituição que limite o acesso à internet, porém isso não impediria os alunos de usarem seus próprios pacotes de dados.

Outra possível limitação seria o limite de usuários. O banco de dados Firebase é gratuito até certo limite de operações diárias, sendo essas 50.000 operações de leitura, 20.000 de escrita. Considerando uma sala de 50 alunos, uma questão gastaria 51 operações de leitura e 52 operações de escrita. Dessa forma, seria possível criar em média 385 questões por dia, o que é suficiente para o uso atual do aplicativo. Ainda que o número de usuários cresça a ponto de que essa cota não seja suficiente, por 24 centavos de dólar diários (7 dólares e 20 centavos por mês), essa cota pode aumentar para 2300 questões diárias.

A última limitação encontrada seria a impossibilidade de se criar mais de um quiz simultaneamente. Essa decisão foi tomada em função de simplificar o uso do aplicativo, tanto quanto tornar mais organizada a exposição da aula futuramente na web.

8. CONCLUSÃO

Neste projeto foi desenvolvido um sistema de perguntas e respostas para o aplicativo WebLectures no qual professores, através do aplicativo em seus sistemas Android, podem criar questões de múltipla escolha a fim de que seus alunos às respondam em seus próprios dispositivos em uma página web. A implementação do sistema foi feita de forma clara e concisa de forma que a criação de um quiz é intuitiva tanto quanto os resultados são limpos e eficientes. Essa nova funcionalidade no aplicativo deve favorecer a aplicação em sala de aula dos modelos de Peer Instruction já discutidos (CROUCH; MAZUR, 2001) (CACEFFO *et al*, 2018).

9. REFERÊNCIAS BIBLIOGRÁFICAS

CACEFFO, R.; GAMA, G.; AZEVEDO, R. **Exploring Active Learning Approaches to Computer Science Classes.**SIGCSE, Baltimore, MD, p. 922-927, fev. 2018.

CACEFFO, R.; WOLFMAN, S.; BOOTH, K. S.; AZEVEDO, R. **Developing a Computer Science Concept Inventory for Introductory Programming.**SIGCSE, Memphis, TN, p. 364-369mar. 2016.

CROUCH, C. H.; MAZUR, E. **Peer Instruction: Ten years of experience and results.** American Journal of Physics, Melville, NY, v. 69, n. 9, p. 970-977, set. 2001.

FAGEN, A. P.; YANG, T.; CROUCH, C. H.; MAZUR, E. **Factors That Make Peer Instruction Work: A 700-User Survey.** AAPT Winter Meeting, Kissimmee, FL, 2000.