

Pedro Henrique Araújo Sobral

Desenvolvimento de um sistema de resposta para uso em sala de aula

Juazeiro, Bahia, Brasil

2017

Pedro Henrique Araújo Sobral

Desenvolvimento de um sistema de resposta para uso em sala de aula

Trabalho de conclusão de curso apresentado à Universidade Federal do Vale do São Francisco, Campus Juazeiro – BA, como requisito para a obtenção do título de Engenheiro de Computação.

Orientador: Dr. Max Santana Rolemberg Farias

Juazeiro, Bahia, Brasil

2017

Pedro Henrique Araújo Sobral

Desenvolvimento de um sistema de resposta para uso em sala de aula

Trabalho de conclusão de curso apresentado à Universidade Federal do Vale do São Francisco, Campus Juazeiro – BA, como requisito para a obtenção do título de Engenheiro de Computação.

Trabalho _____. Juazeiro, Bahia, Brasil, 25 de Janeiro de 2017:

Orientador (CCOMP/UNIVASF)
Dr. Max Santana Rolemberg Farias

Professor (CCOMP/UNIVASF)
Dr. Brauliro Gonçalves Leal

Professor (CCIVIL/UNIVASF)
Me. João Carlos Sedraz Silva

Juazeiro, Bahia, Brasil
2017

*“Diga-me eu esquecerei,
ensina-me e eu poderei lembrar,
envolva-me e eu aprenderei.”
(Benjamin Franklin)*

Resumo

Sistemas de resposta em sala de aula permitem ao professor um retorno em tempo real sobre o entendimento de toda uma classe sobre um determinado tópico de estudo. Essa informação é valiosa porque permite ao educador, por exemplo, realizar uma avaliação formativa, orientando-o na prática pedagógica. No entanto, o custo associado à aquisição dos sistemas de resposta podem ser proibitivos ao uso. É importante destacar essa tecnologia apenas como um meio para colaborar no processo de ensino e aprendizagem, e só faz sentido quando associada com práticas pedagógicas de ensino como o aprendizado ativo. Dessa forma, nesse trabalho será desenvolvido um sistema de resposta em sala de aula, que possibilite aos professores e aos estudantes uma ferramenta de software livre, que permita usar *smartphones* como sistema de resposta.

Palavras-chave: Aprendizado Ativo. Instrução pelos Colegas. Sistemas de Resposta em Sala de Aula. Aplicativos Híbridos para Celular.

Abstract

Classroom response systems (CRS) or just clickers allow the teacher a real-time feedback on the understanding of a whole class on a topic of study. This information is valuable because it allows the educator, for example, conduct a formative assessment, guiding it in pedagogical practice. However, it is important to highlight this technology only as a means to assist in the process of teaching and learning, and only makes sense when combined with pedagogical practices of teaching and active learning. Moreover, the cost associated with clickers can be prohibitive to use. Thus, this work will develop a CRS that allows teachers and students a free software that will give them the opportunity to use smartphones as clickers so that it can be mainly used for educational practices of active learning as Peer Instruction.

Keywords: Active Learning. Peer Instruction. Classroom Response Systems. Hybrid Mobile Applications.

Lista de ilustrações

Figura 1 – Diagrama do processo de implementação do método IpC	13
Figura 2 – Exemplo de uma questão conceitual	14
Figura 3 – Exemplo de <i>clickers</i>	15
Figura 4 – Preço do <i>i>clicker 2</i>	16
Figura 5 – Preço de algumas sistemas de resposta	18
Figura 6 – Engenharia de Software - uma tecnologia em camadas	22
Figura 7 – Requisitos iniciais para o sistema em forma de histórias de usuário . . .	29
Figura 8 – HelloIonicPage: classe responsável por exibir e controlar a página . . .	30
Figura 9 – HelloIonicPage: elementos visuais da página	30
Figura 10 – Página em <i>Ionic</i> resultado das Figuras 8 e 9	31
Figura 11 – Exemplo de componentes no <i>Ionic</i>	32
Figura 12 – Interface de um serviço	33
Figura 13 – Exemplo registro de <i>hooks</i> no <i>FeathersJS</i>	34
Figura 14 – Exemplo eventos no <i>FeathersJS</i>	34
Figura 15 – Como o ciclo de uma requisição funciona no <i>FeathersJS</i>	35
Figura 16 – Aba <i>Questões</i>	36
Figura 17 – Página para a realizar a frequência dos estudantes	37
Figura 18 – Página para a realizar a frequência dos estudantes	38
Figura 19 – Página para a realizar a frequência dos estudantes	38
Figura 20 – Formulário <i>Nova Questão</i>	39
Figura 21 – Aba <i>Turmas</i>	40
Figura 22 – Adicionar estudantes em uma turma	41
Figura 23 – Exemplo de arquivo CSV válido para importar uma lista de estudantes	42
Figura 24 – Aba <i>Questões</i>	42
Figura 25 – Adicionar estudantes em uma turma	43
Figura 26 – Adicionar estudantes em uma turma	44
Figura 27 – Adicionar estudantes em uma turma	44
Figura 28 – Adicionar estudantes em uma turma	45
Figura 29 – Tela inicial da aplicação móvel do estudante	46
Figura 30 – Tela de identificação dos estudantes	47
Figura 31 – Telas para os estudantes realizarem a frequência	48
Figura 32 – Diferentes tipos de questões no aplicativo	49

Lista de abreviaturas e siglas

CSS	Cascading Style Sheets
EUA	Estados Unidos da América
HTML	HyperText Markup Language
IpC	Instrução pelos Colegas
MIT	Massachusetts Institute of Technology
USP	Universidade de São Paulo

Sumário

1	INTRODUÇÃO	10
1.1	Objetivo Geral	10
1.2	Objetivos Específicos	11
1.3	Organização do texto	11
2	REVISÃO DE LITERATURA	12
2.1	Aprendizado Ativo	12
2.1.1	Instrução pelos Colegas (IpC)	13
2.2	Sistemas de Resposta em Sala de Aula	14
2.2.1	Alternativas Disponíveis	17
2.3	Benefícios dos Sistemas de Resposta em Sala de Aula	17
2.3.1	Benefícios para a sala de aula	17
2.3.2	Benefícios para a aprendizagem	19
2.3.3	Benefícios para avaliação	20
2.3.4	Desafios para usar <i>clickers</i>	21
2.3.5	Considerações finais	21
3	ENGENHARIA DE SOFTWARE	22
3.1	Processos de Software	22
3.1.1	Processo Unificado	23
3.1.2	Métodos Ágeis	24
3.2	Metodologias e Ferramentas	24
3.2.1	Objetivos específicos	24
3.3	Planejamento	25
3.4	Elicitação de Requisitos	26
3.4.1	Análise de competidores	26
3.4.2	Requisitos gerados a partir da análise de competidores	27
3.5	Implementação	28
3.5.1	Plataforma	28
3.5.2	Arquitetura	28
3.5.3	Estrutura do Framework: <i>Ionic</i>	28
3.5.4	Estrutura do Framework: <i>FeathersJS</i>	31
4	RESPOSTA EM SALA DE AULA	36
4.1	Aplicação web para o Professor	36
4.1.1	Frequência dos alunos	37

4.1.2	Cadastro de Questões	38
4.1.3	Turmas	38
4.1.4	Sessão de questões: Aba Ao Vivo	40
4.2	Aplicação para dispositivos móveis, que será utilizado como <i>clickers</i>	46
4.2.1	Identificação dos estudantes	46
4.2.2	Responder frequência	47
4.2.3	Responder questões	47

REFERÊNCIAS	50
------------------------------	-----------

APÊNDICES	55
------------------	-----------

APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ES- CLARECIDO	56
--	-----------

1 Introdução

Por que ainda se justifica o modelo de ensino em que se baseia na disseminação de informações, que nunca foi tão fácil achar (internet, livros, etc)? O ensino de hoje deveria estar focado para uma nova visão em que o papel do professor seja de intermediar a aprendizagem (ARAÚJO; MAZUR, 2013).

Sistemas de respostas para uso em sala de aula são sistemas que possibilitam que todos os alunos respondam a questões apresentadas no projetor. Geralmente um gráfico de barras é apresentado logo em seguida que os estudantes submetem as suas soluções utilizando algum dispositivo remoto. As respostas são anônimas para os seus colegas, no entanto o professor pode identificar cada estudante individualmente pela identificação única do dispositivo, permitindo assim uma análise individual (KAY; LESAGE, 2009).

Não existe na literatura um consenso sobre a nomenclatura para referenciar sistema de resposta para uso em sala de aula. Pode-se encontrar termos como “*student response systems*” (sistemas de resposta do estudante), “*audience response system*” (sistema de resposta a audiência), “*personal response system*” (sistema de resposta pessoal), “*classroom response system*” (sistema de resposta em sala de aula), “*electronic feedback system*” (sistema eletrônico de retorno), ou principalmente como “*clickers*” (HUNSU; ADESOPE; BAYLY, 2016). Nesse trabalho foi utilizado o termo *clickers* por ser o mais frequente na literatura internacional e também por ter sido encontrado na literatura brasileira para representar as tecnologias de sistemas de resposta para uso em sala de aula (MATTOS, 2015; ARAÚJO; MAZUR, 2013).

O resultado imediato disponibilizado por *clickers* pode ser usado juntamente com métodos que promovem a interação social voltada para a aprendizagem como a Instrução pelos Colegas (IpC), que tem alcançado sucesso internacionalmente (ARAÚJO; MAZUR, 2013).

1.1 Objetivo Geral

Desenvolver um sistema de resposta para uso sala de aula, que possibilite aos professores e aos estudantes uma ferramenta de software livre, que permita usar *smartphones* como *clickers* para que o mesmo possa ser usado principalmente com práticas pedagógicas de aprendizado ativo como o IpC.

1.2 Objetivos Específicos

- Realizar levantamento de requisitos sobre os sistemas de resposta em sala de aula;
- Especificar e implementar uma aplicação para dispositivos móveis, que será utilizado como *clickers*;
- Especificar e implementar uma aplicação web para o professor administrar as questões e gerar relatórios;
- Especificar e implementar um sistema servidor, para receber e enviar dados para os clientes: dispositivos móveis dos alunos e navegador web do professor.

1.3 Organização do texto

Além desta introdução, este trabalho está dividido em mais dois capítulos.

Revisão de literatura: Esse capítulo discute um contexto pedagógico para o uso de sistemas de resposta em sala de aula. Inicialmente, é apresentado o conceito de aprendizado ativo e um método de implementação: Instrução pelos Colegas. Em seguida, são apontados os sistemas de resposta em sala de aula como uma ferramenta para ajudar o professor a mediar um aprendizado significativo em sala de aula. O capítulo é encerrado apresentando os benefícios e desafios de uso dessa tecnologia.

Material e Métodos: Nesse capítulo serão descritas as fases de especificação e projeto do sistema. Ainda nesse capítulo são apresentadas as ferramentas e a arquitetura do sistema, dando uma visão geral do que foi desenvolvido.

2 Revisão de Literatura

Neste capítulo, inicialmente, é apresentado o conceito de aprendizado ativo e uma alternativa para a implementação dessa metodologia de ensino. Em seguida, é apontado o clicker como uma ferramenta para ajudar o professor na promoção de um aprendizado significativo em sala de aula. O capítulo é encerrado com informações sobre os benefícios e desafios de uso de dessa tecnologia.

2.1 Aprendizado Ativo

Em resumo, o aprendizado ativo é um conjunto de práticas pedagógicas que além de envolver os estudantes no fazer, os faça pensar no que estão fazendo ([CHARLES; JAMES, 1991](#), p. 19).

Os princípios do aprendizado ativo são dois: introduzir atividades nas salas de aula tradicionais e promover o envolvimento dos estudantes. O primeiro é a forma mais simples do aprendizado ativo. Um exemplo seria fazer pequenas pausas na aula e colocar os estudantes para revisar as notas de aula com um colega. No entanto, é preciso que tais atividades promovam o engajamento dos estudantes no processo de ensino e aprendizagem ([PRINCE, 2004](#), p. 3).

Em salas de aula que utilizam o aprendizado ativo pode-se ter um aumento de até 6% na média final nas provas dos alunos e que aquelas salas de aula puramente expositivas têm 55% mais chance de reprovarem os alunos do que aquelas com aprendizado ativo. Esse aumento que pode parecer pouco (0,3 na média final) colocaria a média daqueles estudantes que desistem do curso bem próximo daqueles que permanecem, podendo-se dessa forma aumentar a taxa de retenção dos estudantes ([FREEMAN et al., 2014](#), p. 4).

O estudo de meta-análise de [Freeman et al. \(2014\)](#) envolveu mais de 200 artigos, que comparavam as performances dos alunos em salas de aula com pelo menos algum elemento de aprendizado ativo com as tradicionais aulas expositivas. Além de mostrarem evidências de que o aprendizado ativo pode melhorar o aprendizado dos estudantes de graduação, principalmente nas áreas de ciência, tecnologia, engenharia e matemática, [Freeman et al.](#) propõem aos futuros pesquisadores testarem não mais a eficiência dos métodos de aprendizado ativo frente as tradicionais aulas expositivas (“primeira geração de pesquisas”), mas sim qual o tipo de aprendizado ativo é mais apropriado para cada área do conhecimento (“segunda geração de pesquisas”).

Apesar dos indícios dos benefícios da Aprendizagem Ativa, no contexto brasileiro, tornar o aluno um agente ativo no processo de ensino e aprendizagem não é uma tarefa fácil. São muitas as adversidades de infraestrutura e institucionais encontradas de modo a propiciar o desenvolvimento dessa metodologia. Seja por salas com numero excessivo

de alunos, estes desinteressados, professores mal pagos, ou com a pressão de produzir cientificamente (ARAÚJO; MAZUR, 2013).

Por outro lado, muitas são as iniciativas encontradas na literatura mostrando resultados satisfatórios que podem ajudar o professor nesse processo (CROUCH; MAZUR, 2001; GOK, 2013; BARROS et al., 2004). Na próxima seção será apresentado o método ativo de ensino *Peer Instruction* ou *Instrução pelos Colegas* (IpC).

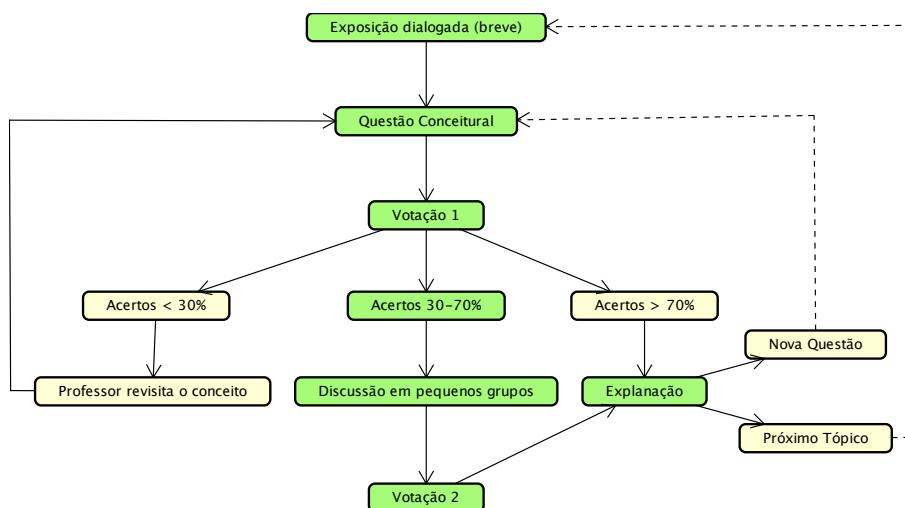
2.1.1 Instrução pelos Colegas (IpC)

O IpC foi desenvolvido pelo Professor Eriz Mazur da Universidade de Harvard (EUA) na década de 1990. O objetivo do IpC é fazer com que todos os estudantes se engajem em discussões com o vizinho de opinião diferente sobre um determinado conceito e fazer com que cada estudante tente explicar o conceito um para o outro (MAZUR, 2009).

No método IpC, geralmente, o professor começa fazendo uma breve exposição dialogada do conteúdo (15min). Depois é colocado para os estudantes uma questão conceitual, que é desenvolvida de modo a avaliar o entendimento dos estudantes sobre um tópico. A Figura 1 resume o processo de implementação do método IpC na sala de aula.

Um exemplo de uma questão conceitual de introdução a física é mostrada na Figura 2. Os estudantes respondem individualmente a questão (1-2min), geralmente, utilizando *clickers*, que são pequenos dispositivos transmissores como os da Figura 3. Em seguida, dependendo do percentual de alunos que acertem a questão, o professor pode revisar o assunto (acerto < 30%), fazer uma breve explanação da questão e ir para um próximo tópico ou nova questão (acerto > 70%), ou o que se deseja do método, um percentual de acerto entre 30% e 70% em que, nessa situação, o professor estimula que os

Figura 1 – Diagrama do processo de implementação do método IpC

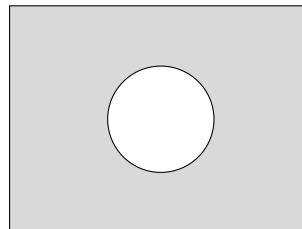


Fonte – Adaptado de (ARAÚJO; MAZUR, 2013)

Figura 2 – Exemplo de uma questão conceitual

Considere uma placa de metal retangular com um furo circular no centro. Se a placa for uniformemente aquecida, o diâmetro do buraco:

1. aumenta;
2. permanece o mesmo;
3. diminui;



Fonte – Adaptado de (WATKINS; MAZUR, 2013)

alunos encontrem um parceiro que respondeu de forma diferente e que tentem explicar um para o outro o porquê de estar correto. Com esse processo, os alunos se auto instruem, o que justifica o nome do método “Instrução pelos Colegas” (MAZUR, 2009; CROUCH; MAZUR, 2001).

Uma das etapas do IpC na sala de aula é a votação, em que os estudantes indicam as suas respostas. Existem pelo menos quatro maneiras (CROUCH et al., 2007) do professor obter um *feedback* da votação dos estudantes:

Levantar as mãos: a forma mais simples é pedir para os alunos levantarem as mãos para cada alternativa, mas, dentre as várias limitações desse método, os estudantes podem se influenciar pela resposta dos outros.

Cartões coloridos: uma segunda alternativa seria o uso de cartões coloridos (*flashcards*) que dificultaria os alunos ver a resposta dos outros e facilitaria a contagem pelo professor, porém uma limitação desse método, assim como o anterior é a dificuldade de alguma forma guardar os resultados.

Folha de respostas: outra alternativa, no entanto não seria possível ter um resultado imediato das respostas.

Sistemas de resposta em sala de aula: exemplo os *clickers* da Figura 3, ou *smartphones*, e sistemas web possibilitam aos estudantes enviarem imediatamente as respostas ao computador do professor, de forma anônima aos colegas de sala e visualizar graficamente os resultados.

2.2 Sistemas de Resposta em Sala de Aula

Sistemas de respostas em sala de aula são tecnologias que permitem ao professor realizar questionamentos a toda classe e, assim, obter o resultados das respostas em tempo

Figura 3 – Exemplo de *clickers*

Fonte – (a) iclicker.com (b) turningtechnologies.com

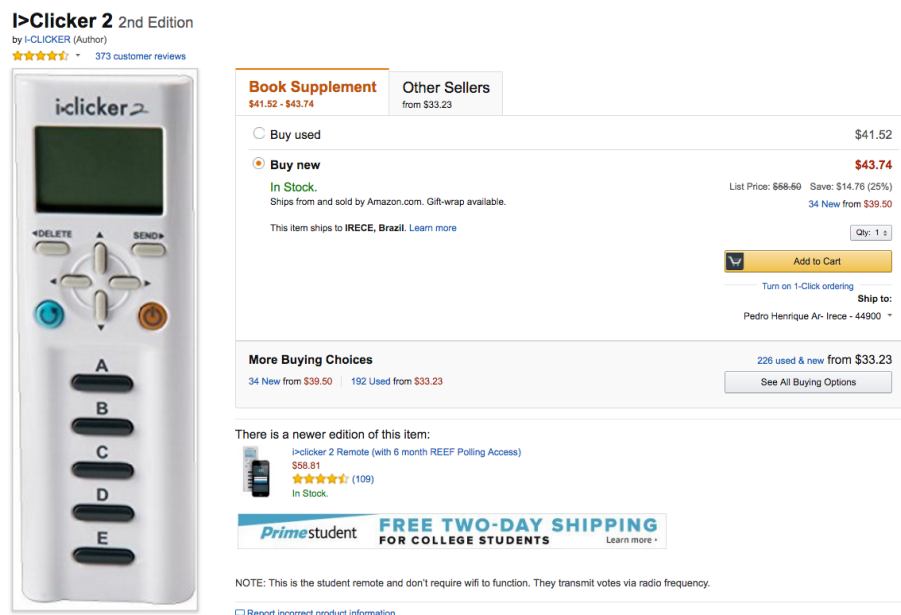
real (KAY; LESAGE, 2009). Geralmente, as questões são no formato de verdadeiro ou falso e questões de múltipla escolha. Tais sistemas também podem ser usados para saber a impressão dos alunos sobre um determinado tema (FIES; MARSHALL, 2006).

Inicialmente introduzido em 1966 na Universidade de Stanford (EUA), os sistemas de resposta não funcionavam direito, eram difíceis de usar e caros (KAY; LESAGE, 2009). Dispositivos de coleta de respostas como os da Figura 3 que usam infravermelho, com preços mais atrativos, começaram a ser extensivamente usados a partir de 2003 e hoje existe pelo menos uma disciplina em cada universidade dos Estados Unidos em que se faz uso de tais sistemas no processo de ensino e aprendizagem (ABRAHAMSON; BRADY, 2014).

Entretanto, ainda que o custo individual de um aparelho seja atrativo para um estudante, a realidade da universidade pública brasileira é diferente em relação a dos Estados Unidos. Assim como não existe obrigatoriedade do aluno comprar um livro-texto no Brasil, também não existe mecanismo que os faça comprar *clickers* (KORTEMAYER; DIAS; CRUZ, 2011). Dessa forma, os *clickers* teriam que ser comprados pela universidade, e o custo total não seria tão atrativo.

Por exemplo, a Universidade de São Paulo (USP), em uma tentativa pioneira de Kortemeyer, Dias e Cruz (2011) de oferecer oportunidades para a avaliação formativa usando *clickers* como os da Figura 3a em duas turmas de física, com 80 alunos cada. Com cada *i>clicker 2* custando \$43,74 (Figura 4) o valor total apenas dos 160 *clickers* seria de R\$22.116,34¹, necessitando para compor o sistema, ainda, o aparelho que recebe as

¹ Considerando o valor do dólar comercial em 12. Ago. 2016 de R\$3,1602. Disponível em:

Figura 4 – Preço do *i>clicker 2*

Fonte – amazon.com pesquisando *i>clicker 2*. Acesso em 12. Ago. 2016

respostas, software que é instalado no computador do professor, treinamento e suporte, ou seja, o valor final pode ser ainda maior. Esse elevado valor de aquisição dos *clickers*, é uma das principais barreiras na adoção dessa ferramenta no processo de ensino, que ainda tem a resistência de alguns professores em usar novas tecnologias como ferramentas de ensino (MORATELLI; DEJARNETTE, 2014; BLASCO-ARCAS et al., 2013; STRASSER, 2010; KORTEMEYER; DIAS; CRUZ, 2011; KAY; LESAGE, 2009).

Todavia, uma maneira de tornar tal tecnologia mais acessível é usar os próprios celulares dos estudantes como *clickers* (STOWELL, 2015; MORRELL; JOYCE, 2015; ARAUJO; MAZUR, 2013). Além de mais acessível, usando os próprios celulares, os estudantes podem ver as questões e os resultados da classe nos próprios aparelhos. Para o professor, uma variedade maior de questões podem ser exploradas, como uma questão aberta (STOWELL, 2015).

Saliente-se ainda que, embora possa haver um pequeno aumento no número de questões sem respostas, as respostas dos estudantes são comparáveis quando se usa *clickers* (MORRELL; JOYCE, 2015; STOWELL, 2015).

Ainda que o uso do celular como *clicker* possa aumentar o nível de distrações, são muitos os benefícios de usar essa abordagem, no entanto, é preciso estar atento aos pequenos casos de estudantes que não têm a tecnologia adequada, ou que não queiram usar os seus dispositivos (MORRELL; JOYCE, 2015; STOWELL, 2015).

<http://www4.bcb.gov.br/pec/taxas/port/ptaxnpesq.asp?id=txcotacao>

2.2.1 Alternativas Disponíveis

Em um estudo para desenvolver uma estratégia de ensino utilizando *clickers* como recurso didático, [Mattos \(2015\)](#), utilizou o *Poll Everywhere*, que foi escolhido principalmente por ser compatível com as principais plataformas móveis (Android, iOS e Windows Phone). O *Poll Everywhere* oferece uma versão gratuita com no máximo 40 respostas por votação com algumas limitações. Na versão paga, o estudante pode pagar \$14/ano ou o professor pode pagar \$349/semestre (vide [Figura 5a](#)).

Outra solução é o *Socrative*, que também é compatível com todas as plataformas, e na sua versão gratuita permite até 50 alunos por votação, questões de múltipla escolha, de verdadeira e falso, resposta curta, *quizzes*, etc ([SOCRATIVE, 2016](#)). Alguns relatos do uso do *Socrative* em sala de aula para promover a interatividade são encontrados em ([KAYA; BALTA, 2016](#); [TRINDADE, 2014](#)). Uma opção interessante do *Socrative* é o *Exit Ticket*, em que ao final da aula os estudantes respondem a três perguntas. A primeira funciona como uma autoavaliação, perguntando o quanto o estudante entendeu da aula, já as outras duas como identificação, que são o nome do aluno e uma pergunta que o professor coloca no quadro, ou seja, teoricamente, apenas os alunos presentes vão ser capazes de responder corretamente, servindo então como um controle de frequência. Na versão paga o professor pode pagar \$49,99/ano (vide [Figura 5b](#)).

Por último, o aplicativo *TopHat*, usado diariamente por mais 500 instituições de ensino pelo mundo ([TOPHAT, 2016](#)). Para o professor, *TopHat* permite incorporar uma variedade de questões, uma revisão interativa da aula e para provas ([NEILSON et al., 2016](#); [LANTZ; STAWISKI, 2014](#)). A inscrição mais popular no *TopHat* são de \$36/ano (vide [Figura 5c](#)).

2.3 Benefícios dos Sistemas de Resposta em Sala de Aula

Os benefícios do uso de sistemas de resposta em sala de aula podem ser divididos em benefícios para a sala de aula, para a aprendizagem e para avaliação ([KAY; LESAGE, 2009](#)). As subseções a seguir discutem cada tópico.

2.3.1 Benefícios para a sala de aula

2.3.1.1 Frequência Escolar

Não apenas uma ferramenta que também pode facilitar o controle de frequência ([STRASSER, 2010](#)), os sistemas de resposta em sala de aula têm sido utilizados com sucesso para aumentar a frequência escolar ([FOTARIS et al., 2016](#); [VELASCO; ÇAVDAR, 2013](#); [PUENTE; SWAGTEN, 2012](#); [MAYER et al., 2009](#); [CALDWELL, 2007](#)). O fato dos estudantes terem tido um retorno imediato das atividades que faziam interativamente

Figura 5 – Preço de algumas sistemas de resposta(a) Disponível em: polleverywhere.com/plans/higher-ed: em 16 ago. 2016**Higher education plans**

Over 100,000 Educators use Poll Everywhere in their classrooms and lecture halls

	Higher-Ed Free	Per Student	Per Instructor	Institution-wide
	Free	\$14 year per student	\$349 semester	Custom
	Sign Up	Sign Up	Sign Up	Contact Us
Responses per poll <input type="checkbox"/> Explain	40	Varies	400	Custom
Choose your username	-	✓	✓	✓

(b) Disponível em: socrative.com/pricing: em 16 ago. 2016

K-12 EDUCATORS

HIGHER ED/CORPORATE

SOCRATIVE

Free

GET FREE

★ SOCRATIVE PRO

\$49⁹⁹

ANNUAL PRICING

GET PRO

(c) Disponível em: tophat.com/pricing em: 16 ago. 2016



Top Hat Lecture

Students purchase a subscription to access the real-time platform
[Learn more about Top Hat Lecture](#)

1 Term

\$24

4 Month access to your courses on Top Hat

Get Started

Annual

\$36

Most Popular!

12 Month access to your courses on Top Hat

Get Started

Lifetime

\$72

Lifetime access* to your courses on Top Hat

Get Started

*Access to Top Hat for as long as you're in school

na sala de aula, pode ter contribuído para motivar os alunos para irem para as aulas (PUENTE; SWAGTEN, 2012). Outro fator para o aumento da frequência é quando as questões dos *clickers* contribuem na nota final, em que o peso pode variar de 5% a 15% que o resultado parece ser o mesmo (CALDWELL, 2007).

2.3.1.2 Atenção

Sabe-se hoje que a maioria das pessoas não conseguem se concentrar ininterruptamente por mais de 20 min em uma única atividade (CALDWELL, 2007; D'INVERNO; DAVIS; WHITE, 2003). Com aulas que duram entre 2h-3h, os professores podem quebrar as aulas em miniaulas com questões ao final delas usando os *clickers* (HUNSU; ADE-SOPE; BAYLY, 2016). Estudo de ondas cerebrais mostraram que a atenção dos estudantes aumentam durante as atividades de votação, além de reduzir a ansiedade (SUN, 2014). Acrescenta-se também que quando os *clickers* são adotados, os estudantes relatam a necessidade de estudar antes da aula para participarem e prestarem atenção, dessa forma participam mais da aula (TERRION; ACETI, 2012).

2.3.1.3 Privacidade e Participação

Os estudantes podem responder as questões sem se preocupar com o julgamento de seus colegas de classe ou do professor. Nesse ambiente seguro o estudante não tem nenhuma razão para sentir medo de responder errado (SCHMIDT, 2011). A falta de privacidade pode coibir a completa honestidade na votação (CALDWELL, 2007), e a presença da mesma da a oportunidade para os estudantes responderem e tomarem posições polêmicas em questões sensíveis (RANA; DWIVEDI, 2016).

2.3.1.4 Engajamento

Inúmeros estudos indicam estudantes mais interessados ou engajados no material apresentado quando sistemas de resposta são usados (KAYA; BALTA, 2016; RANA; DWIVEDI, 2016; HORNE, 2015; MATTOS, 2015; MORATELLI; DEJARNETTE, 2014; KULATUNGA; RAMEEZDEEN, 2014; BLOOD; GULCHAK, 2013; TERRION; ACETI, 2012; CALDWELL, 2007).

2.3.2 Benefícios para a aprendizagem

2.3.2.1 Interação e Discussão

A interação na sala de aula é importante por três motivos: primeiro pelo que foi apresentado na subseção 2.3.1.2, segundo que permite clarificar pontos obscuros e terceiro que permite uma monitoração do entendimento da classe e velocidade do que é apresentado (D'INVERNO; DAVIS; WHITE, 2003). Nesse sentido, estudos indicam aumento da

interação na sala de aula (MATTOS, 2015; BARRAGUÉS; MORAIS; GUIASOLA, 2011; TITMAN; LANCASTER, 2011; MAYER et al., 2009; CALDWELL, 2007).

2.3.2.2 *Contingent Teaching*

Contingent Teaching nada mais é do que a possibilidade de modificar o curso do que está sendo ensinado com base no *feedback* dos estudantes (ARNESEN et al., 2013; CALDWELL, 2007). Por exemplo, se pouco mais da metade da classe responde corretamente a uma questão, conclui-se uma falta de compreensão geral do tópico apresentado, e assim o professor pode tentar explicar o conceito de maneira diferente (TERRION; ACETI, 2012; STRASSER, 2010).

2.3.2.3 Melhora no aprendizado

Lantz e Stawiski (2014) verificaram um aumento significativo (15%) no grupo de participantes que tiveram aulas com *clickers* em relação ao sem o uso em testes que ocorriam dois dias depois as aulas, e também permitiu aos estudantes corrigir eventuais equívocos sobre o material de aula. Outros estudos apontam melhora na performance dos estudantes (SUN, 2014; CALDWELL, 2007). Entretanto, em um recente estudo de meta-análise, em que se comparou classes que usaram e não usaram sistemas de resposta em sala de aula, indicou um significativo impacto positivo nas variáveis cognitivas (retenção, transferencia de conhecimento e sucesso), e nas variáveis não-cognitivas (engajamento e participação, frequência, interesse, percepção de qualidade) (HUNSU; ADESOPE; BAYLY, 2016).

2.3.3 Benefícios para avaliação

2.3.3.1 *Feedback*

Como destacado na subseção 2.1.1, dentre as quatro formas disponíveis para o professor obter um *feedback* ou retorno do entendimento dos estudantes sobre determinado tópico, se disponível, os *clickers* são os mais eficazes (CROUCH et al., 2007). Os *clickers* permitem um retorno imediato, preciso, privacidade (vide subseção 2.3.1.3) e são divertidos de usar (RANA; DWIVEDI, 2016; BLOOD; GULCHAK, 2013; CALDWELL, 2007).

2.3.3.2 Avaliação Formativa

Avaliar o estudante permanentemente, e não apenas utilizando avaliações pontuais, não apenas classificar os alunos em os que sabem e os que não sabem, não apenas verificar e sim avaliar, dedicar mais tempo nos erros (UNIVESPTV, 2013). Esses são alguns pontos da avaliação formativa, que para o professor orienta a prática pedagógica, e para o aluno permite uma autoavaliação sobre a aprendizagem (KAY; LESAGE, 2009). Os artigos

a seguir indicam que o uso de sistemas de resposta em sala de aula ajudam a fornecer avaliações formativas eficazes (KORTEMEYER, 2016; THAMPY; AHMAD, 2014; KAY; LESAGE, 2009; FIES; MARSHALL, 2006).

2.3.4 Desafios para usar *clickers*

O uso dos *clickers* também trazem alguns desafios, principalmente tecnológicos ou estruturais e desafios centrado nos estudantes (CUBRIC; JEFFERIES, 2015; KAY; LESAGE, 2009).

2.3.4.1 Tecnológicos/Estruturais

Stowell (2015) recomenda que as salas de aula tenham acesso confiável a rede Wi-Fi, por isso o professor deve estar preparado quando a internet não estiver disponível (STRASSER, 2010). Como já abordado nesse trabalho (vide seção 2.2), nem todos os estudantes vão ter a tecnologia adequada ou vontade de usar (MORRELL; JOYCE, 2015; STOWELL, 2015). Dessa forma, Velasco e Çavdar (2013) aconselham os *clickers* como uma ferramenta opcional para os estudantes.

2.3.4.2 Desafio para os estudantes

Alguns estudantes podem ter resistência ao novo, principalmente se já acomodados a passividade em sala de aula (TERRION; ACETI, 2012; KAY; LESAGE, 2009). Além disso, quando usados apenas para controle de frequência, os estudantes podem ter uma percepção negativa da tecnologia (TERRION; ACETI, 2012; CALDWELL, 2007).

2.3.5 Considerações finais

Por outro lado, apesar das seções anteriores serem animadoras quanto ao uso dos *clickers* é importante destacar a tecnologia apenas como um meio para colaborar no processo de ensino e aprendizagem, e só faz sentido quando associada com práticas pedagógicas de ensino como o aprendizado ativo (vide seção 2.1) (TERRION; ACETI, 2012; MORAN; MASETO; BEHRENS, 2006). O leitor interessado sobre o uso das novas tecnologias em sala de aula, poderá buscar nas obras indicadas em (BRASIL, 2014).

Han (2014) concluiu que o ensino e aprendizado com o auxílio de *clickers* ainda é um fenômeno complexo e relacional. Segundo o mesmo autor, estudos mais rigorosos devem ser conduzidos para se entender por exemplo a relação entre o uso de *clickers* e o aprendizado, pois o benefícios da tecnologia podem variar pelo histórico do estudante, motivação e metacognição.

3 Engenharia de Software

3.1 Processos de Software

Engenharia de Software pode ser definida como:

1. the systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software [...] 2. the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software (SYSTEMS..., 2010).

A engenharia de software deve ter foco na qualidade, que apoia as outras camadas dessa tecnologia, que são as camadas de processo, métodos e ferramentas [Figura 6](#). A camada de processo define um conjunto de atividades ou um arcabouço que tem como finalidade garantir a efetiva utilização da tecnologia engenharia de software, que dessa forma leva à produção de um software. Os detalhes de como fazer o software pertencem a camada de métodos. Os métodos da engenharia de software incluem tarefas de planejamento e estimativa de software, análise de requisitos, modelagem de projeto, codificação, testes e manutenção. As ferramentas de engenharia de software auxiliam as camadas de processo e métodos, com ferramentas automatizadas, que por sua vez, quando integradas, é estabelecido um suporte ao desenvolvimento de software chamado CASE - *Computer Aided Software Engineering* ([PRESSMAN, 2009](#); [SOMMERVILLE, 2006](#)).

Entre o conjunto de atividades definidas pela camada de processo, quatro são fundamentais, a saber, especificação de software, projeto e implementação de software, validação de software e evolução de software. Especificação de software ou engenharia de requisitos é uma fase importante e crítica do processo de engenharia de software. Importante porque é uma análise de requisitos bem feita que possibilitará atender as demandas dos usuários. Crítica porque um sistema mal especificado, pode até ser bem

Figura 6 – Engenharia de Software - uma tecnologia em camadas



Fonte – ([PRESSMAN, 2009](#))

projetado e construído, mas não vai atender as necessidades dos usuários. Em seguida, na fase de projeto e implementação os requisitos são projetados e programados, tendo como resultado um sistema executável. Depois, o software deve ser verificado para mostrar que atende às demandas dos usuários (validação do software). Finalmente, na fase de evolução de software, o mesmo é modificado devido às mudanças de requisitos e às necessidades dos usuários.

Quanto ao modelos de engenharia de software temos os modelos prescritivos e os ágeis. Os modelos prescritivos definem um conjunto de atividades explícitas para serem desenvolvidos no processo de engenharia de software. O Processo Unificado de desenvolvimento tem o maior destaque e reconhecimento dentre os modelos prescritivos.

3.1.1 Processo Unificado

O Processo Unificado define quatro atividades no processo de desenvolvimento de software. As fases são descritas como:

Concepção: Por meio da comunicação com os clientes, um conjunto preliminar de casos de uso UML e da arquitetura do sistema é elaborada.

Elaboração: Nessa fase, os casos de uso UML são expandidos e refinados, bem como a definição da arquitetura do sistema. Ao final dessa fase, um conjunto de casos de uso UML descrevem os requisitos dos sistema. Uma descrição da arquitetura do sistema e um plano de desenvolvimento do software também são esperados.

Construção: Os casos de uso UML desenvolvidos na fase de elaboração são então desenvolvidos e testados. O software deve estar funcionando e documentado na conclusão dessa fase.

Transição: transferência do software aos usuários finais.

Fluxos de trabalhos ou *workflows* ocorrem durante todas as fases no processo de desenvolvimento. São seis *workflows* principais e três de apoio:

1. *Modelagem de negócios:* processos de negócios modelados por casos de uso.
2. *Requisitos:* atores são identificados e casos de uso são desenvolvidos.
3. *Análise e projeto:* vários modelos são desenvolvidos como de arquitetura e componentes.
4. *Implementação:* componentes são implementados.
5. *Teste:* processo iterativo em conjunto com a implementação.

6. *Implantação*: software é distribuído aos usuários finais.
7. *Gerenciamento de configuração e mudança*: apoio as mudanças do sistema.
8. *Gerenciamento de projeto*: apoio de desenvolvimento do sistema.
9. *Meio ambiente*: apoio a equipe de desenvolvimento com ferramentas.

3.1.2 Métodos Ágeis

Contrapondo-se aos modelos prescritivos em que propoem especificar por completo os requisitos do sistema e só então projetar, construir e testar o sistema, surgiu os métodos ágeis, que têm como filosofia o manifesto ágil. Esse manifesto afirma:

Estamos descobrindo melhores maneiras de desenvolver softwares, fazendo-o e ajudando outros a fazê-lo. Através desse trabalho, valorizamos mais:

- Indivíduos e interações do que processos e ferramentas
- Software em funcionamento do que documentação abrangente
- Colaboração do cliente do que negociação de contrato
- Resposta a mudanças do que seguir um plano

Ou seja, embora itens à direita sejam importantes, valorizamos mais os que estão à esquerda.

Abordagens ágeis incluem *Extreme Programming* e o *Scrum*. Eles propõem diferentes processos para que tenha-se um desenvolvimento e entrega incremental do sistema, tendo em comum princípios baseados no manifesto ágil.

3.1.2.1 Scrum

3.2 Metodologias e Ferramentas

A metodologia escolhida nesse projeto levou em consideração as necessidades de um trabalho de conclusão de curso no curto prazo e os recursos limitados. Dessa forma, uma abordagem baseada em metodologias ágeis foi utilizada para modelagem, documentação e codificação do software.

3.2.1 Objetivos específicos

Para cada objetivo específico deste trabalho, as seguintes técnicas e ferramentas foram utilizadas:

3.2.1.1 Realizar levantamento de requisitos sobre os sistemas de resposta em sala de aula

O processo de elicitação de requisitos do sistema utilizou a análise de competidores que consistiu basicamente em buscar em alguns sistemas de resposta existentes, referências positivas e negativas para definição do modelo a ser proposto.

3.2.1.2 Especificar e implementar uma aplicação para dispositivos móveis, que será utilizado como *clickers*

Com os requisitos da primeira etapa, tivemos os requisitos iniciais para a aplicação móvel. A linguagem de programação usada na fase de implementação do aplicativo foi JavaScript, tendo como auxílio o *framework Ionic*.

O *Ionic* é um *framework* de código aberto para o desenvolvimento de aplicativos híbridos e desktop utilizando tecnologias web como HTML, CSS e JavaScript otimizados para dispositivos móveis, com código fonte sobre a licença MIT.

3.2.1.3 Especificar e implementar uma aplicação web para o professor administrar as questões e gerar relatórios

Com os requisitos da primeira etapa, tivemos os requisitos iniciais para a aplicação desktop do professor. O *Ionic* também foi utilizado na aplicação desktop do professor.

3.2.1.4 Especificar e implementar um sistema servidor, para receber e enviar dados para os clientes: dispositivos móveis dos alunos e navegador web do professor

O sistema servidor foi desenvolvido utilizando tecnologias como *Node.js*, *MongoDB*, e o *framework FeathersJS*. Tais tecnologias foram utilizadas por permitir o fácil desenvolvimento de aplicações web de tempo real entre o servidor e os seus clientes.

A teoria mais aprofundada sobre os métodos e ferramentas citadas serão descritas nas próximas seções.

3.3 Planejamento

A fase inicial do projeto foi a de planejamento para a definição dos requisitos do sistema. Nessa etapa, utilizamos a análise de competidores para elucidar os requisitos que foram descritos em histórias de usuário.

3.4 Elicitação de Requisitos

3.4.1 Análise de competidores

A análise de competidores é uma técnica oriunda engenharia da usabilidade que consiste em avaliar produtos concorrentes em busca de pontos positivos e negativos. Tal técnica é útil no levantamento de requisitos de um novo sistema, identificação de pontos fortes e fracos os produtos, reutilização de design, dentre outros.

Avaliar produtos concorrentes é valioso, porque oferece a oportunidade de novos produtos evitarem problemas existentes dos competidores, explorar os pontos fracos, além da reutilização dos pontos positivos.

Nesse sentido, a análise de competidores foi utilizada neste trabalho para elicitar requisitos e boas práticas de design de interfaces. Os resultados obtidos foram utilizados no processo de desenvolvimento do software.

3.4.1.1 *Socrative*

Socrative é um sistema de resposta específico para usar em salas de aula. O sistema pode ser acessado pelo site ou nos aplicativos para *iOS* e *Android*. No *Socrative*, apenas o professor precisa fazer um cadastro no site (questões demográficas são solicitadas). Existe uma versão gratuita e paga do aplicativo.

Na conta do professor, é possível criar questionários de múltipla escolha, verdadeira e falso e de questões abertas. Quando o professor cria uma conta, é gerado um código de identificação para que os alunos possam entrar na sala virtual. Na interface do estudante, é necessário colocar o código de identificação do professor.

3.4.1.2 *PollEverywhere*

O *PollEverywhere* é um sistema de resposta mais genérico, possibilitando fazer votações em shows e apresentações diversas. Possibilita integração com ferramentas de apresentação como o *PowerPoint*. Outra característica é a possibilidade dos usuários votarem por SMS. Além dos tipos de questões básicas, o *PollEverywhere* permite criar nuvem de palavras e questões com imagens clicáveis.

A conta do usuário é associado com uma URL, em que é usada para os participantes da votação entrarem e votarem.

3.4.1.3 *TopHat*

TopHat é outra solução voltada para a educação, contando com seis tipos de questões. Adicionalmente o *TopHat* permite ao professor fazer a chamada dos estudantes, isso porque o professor pode gerar um código aleatório no *TopHat* para que os estudantes

Caraterística	<i>PollEverywhere</i>	<i>TopHat</i>	<i>Socrative</i>
Open-Source	Não	Não	Não
Integração com LMS	Blackboard	Excel	MasteryConnect
Formatos	JSON, RSS, CSV	Não	Não
Read-only API	Sim	Não	Não
Integração com PowerPoint	Possibilita	Não	Não
Métodos de votação	SMS, web	SMS, Web	Internet
Tempo-real	Sim	Sim	Sim
Acesso ao sistema	URL	Código de Acesso	Código de Acesso
Tipos de questões	5	7	3
Mínimo de passos para votação	2	3	4
Anonimato	Possibilita	Possibilita	Possibilita
Contagem regressiva	Possibilita	Possibilita	Não
Download CSV	Possibilita	Não	Não
Relatórios por estudante	Possibilita	Possibilita	Não

Tabela 1 – Análise de Competidores

presentes possam enviar o código e marcar presença. O produto também disponibiliza uma sala de discussão e a possibilidade de criar slides dentro do aplicativo.

3.4.2 Requisitos gerados a partir da análise de competidores

A partir das informações coletadas na análise de competidores, foram extraídos um conjunto de requisitos iniciais para o sistema. Os requisitos funcionais gerados pela análise dos competidores foram então descritos como histórias de usuário.

Integração com sistemas LMS : O sistema deve permitir integração com sistemas LMS (preferencialmente Moodle);

Todas as plataformas: é muito importante que o sistema seja capaz de funcionar em smartphones, tables e computadores independentemente do sistema operacional.

Questões abertas, verdadeiro/falso e de múltipla-escolha: O sistema deve fornecer pelo menos esses três tipos básico de questões;

Modo de votação: O sistema deve permitir votação anônima ou requisitar a identificação;

Customização das questões: O sistema deve permitir inserção de equações matemáticas (\LaTeX), imagens e texto como opção das questões;

Controle da votação: Opções básicas como ativar ou desativar a votação e limpar uma votação em andamento;

Controle de frequência: o sistema gera um código aleatório ou uma questão trivial, em que o professor pode solicitar que os estudantes respondam, contando como controle de frequência. Os dados devem ser facilmente exportados para CSV.

Tempo-real: No momento da votação, o professor pode escolher entre apresentar o resultado em tempo-real, quando todos votarem, ou quando determinado;

Banco de questões: As questões elaboradas pelo professor podem ser armazenadas em um banco de questões que o sistema deve manter;

Facilidade do uso e de criação de votação: O sistema não deve oferecer dificuldades de uso e de criação de questões;

Código de acesso: O sistema deve gerar um código de acesso único para identificar o ambiente do professor, usado para que os alunos respondam.

3.4.2.1 Histórias de Usuário

As histórias de usuário ([Figura 7](#)), formaram a lista de tarefas (*Product Backlog*). Esse foi um conjunto inicial de tarefas do projeto, ou os requisitos principais no momento em que foi criado. Lembrando que o *Product Backlog* nunca é completo, no scrum ele é um artefato vivo que vai sendo modificado.

3.5 Implementação

3.5.1 Plataforma

3.5.2 Arquitetura

3.5.3 Estrutura do Framework: *Ionic*

Ionic é um *framework open-source* para o desenvolvimento de aplicativos híbridos utilizando tecnologias web como HTML, CSS e JavaScript otimizadas para dispositivos móveis, com código fonte sobre a licença MIT ([DRIFTY, 2016](#)). Uma das principais vantagens do desenvolvimento de aplicativos híbridos é que com apenas um código base é possível criar aplicativos para várias plataformas como *iOS*, *Android* e *Windows Phone*, Desktop, que aliás foi uma das razões que fez o Moodle usar o *Ionic* como *framework* para o desenvolvimento do *Moodle Mobile 2* ([MOODLE, 2015](#)).

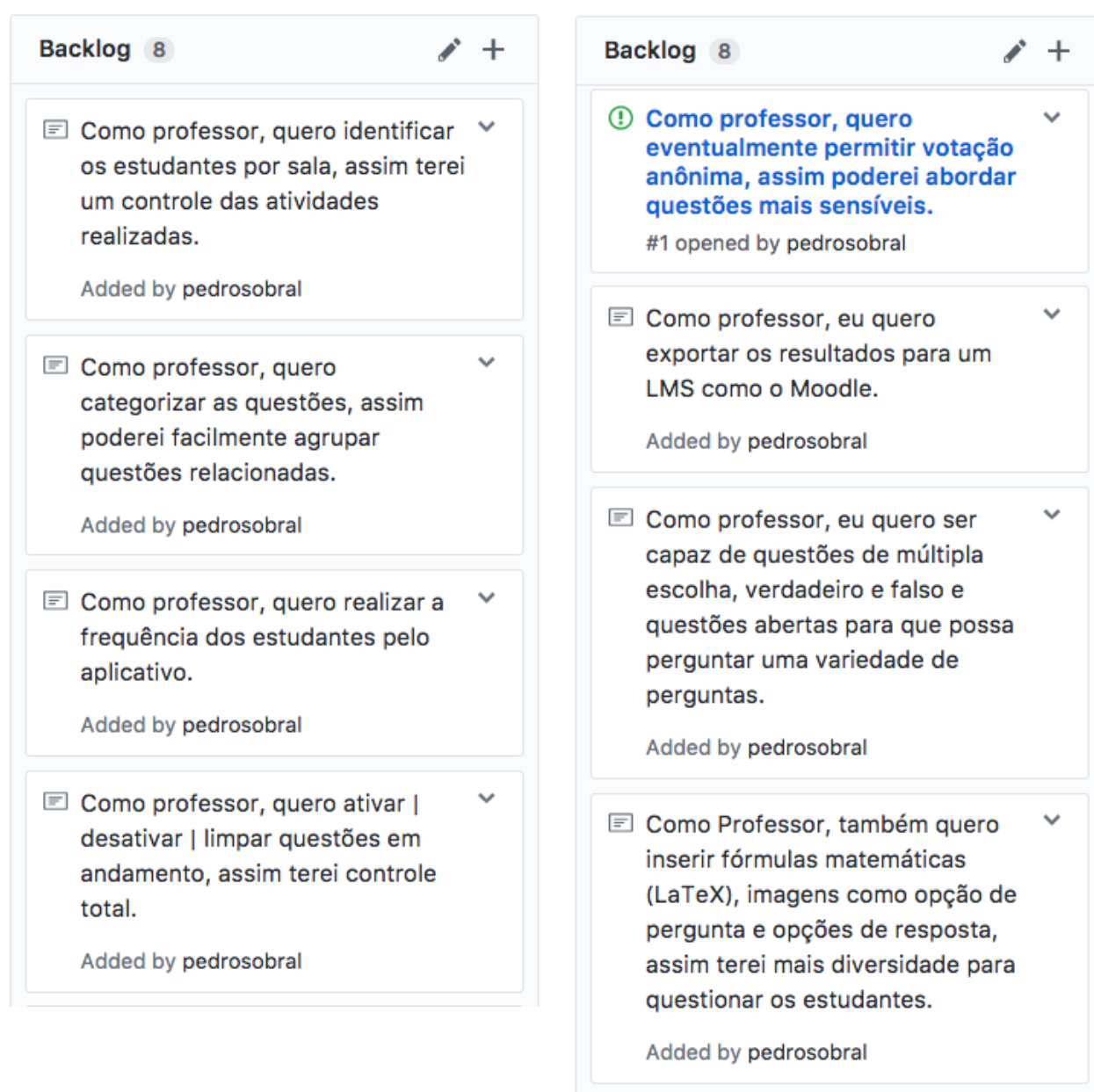


Figura 7 – Requisitos iniciais para o sistema em forma de histórias de usuário

3.5.3.1 Pages

Um aplicativo desenvolvido no *Ionic* é composto por um conjunto de pages ou páginas. Cada página é composta por alguns arquivos. Um arquivo é responsável pelo elemento visual da página, desenvolvido em HTML. Existe o arquivo de estilos da página, desenvolvido em SCSS. O arquivo principal é o responsável por controlar a página, desenvolvido em TypeScript.

As Figuras 8 e 9 são um exemplo básico de uma página de um aplicativo desenvolvido em *Ionic*. O resultado dessa página é mostrado na Figura 10. Observe que com apenas um código base, os elementos visuais da página são diferentes dependendo da plataforma (*iOS*, *Android* e *Windows*).

```
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'page-hello-ionic',
5    templateUrl: 'hello-ionic.html'
6  })
7  export class HelloIonicPage {
8    constructor() {}
9  }
```

Figura 8 – HelloIonicPage: classe responsável por exibir e controlar a página

```
1  <ion-header>
2    <ion-navbar>
3      <button ion-button menuToggle>
4        <ion-icon name='menu'></ion-icon>
5      </button>
6      <ion-title>Hello Ionic</ion-title>
7    </ion-navbar>
8  </ion-header>
9
10 <ion-content padding>
11
12   <h3>Welcome to your first Ionic app!</h3>
13
14   <p>
15     This starter project is our way of helping you get a functional
16     app running in record time.
17   </p>
18   <p>
19     Follow along on the tutorial section of the Ionic docs!
20   </p>
21   <button ion-button color='primary' menuToggle>Toggle Menu</button>
22 </p>
23
24 </ion-content>
```

Figura 9 – HelloIonicPage: elementos visuais da página

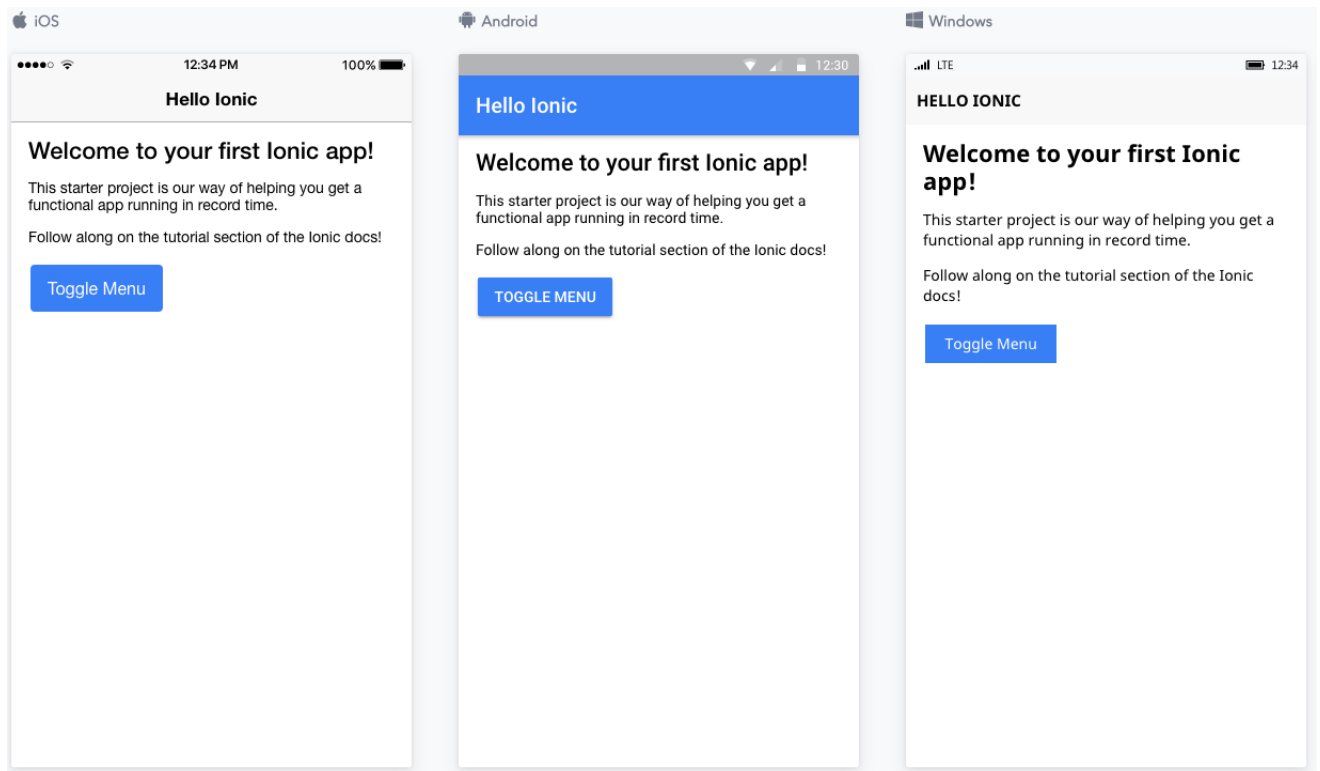


Figura 10 – Página em *Ionic* resultado das Figuras 8 e 9

3.5.3.2 Components

Os elementos visuais e também o comportamento desses elementos em uma página são normalmente construídos por meio dos *components* ou componentes. Os componentes permitem criar facilmente a interface do aplicativo. Exemplo de componentes são botões, *modals*, *popup* e *cards*. Um aspecto interessante é que os componentes se adaptam visualmente a cada plataforma, como já mostramos na [Figura 10](#). Além do aspecto visual eles também se comportam de maneira diferente dependendo da plataforma. Por comportamento, entende-se, por exemplo, os efeitos visuais de cada componente e também efeitos de transição entre as páginas. A [Figura 11](#) ilustra alguns componentes disponíveis no *Ionic*.

3.5.4 Estrutura do Framework: *FeathersJS*

FeathersJS é um *framework* de código livre de desenvolvimento rápido para aplicações web em tempo-real escritas em JavaScript. Disponibiliza uma arquitetura simples mas poderosa para a construção de aplicações utilizando padrões de programação orientada a aspectos e serviços. Os principais componentes do *FeathersJS* são os *services*, *hooks* e *events* que são detalhados nas próximas seções.

Figura 11 – Exemplo de componentes no *Ionic*

3.5.4.1 Services

Services ou serviços são a camada principal do *FeathersJS*. Um serviço é simplesmente uma instância de uma classe JavaScript que implementa métodos básicos para criação, consulta, atualização e destruição de dados. Esse conjunto de operações ou funcionalidades é conhecido como CRUD (*Create, Read, Update, Delete*).

Os serviços no *FeathersJS* expõem uma interface uniforme de acesso, permitindo assim fornecer uma única API tanto para chamadas HTTP REST e *websockets*. Os verbos HTTP (GET, POST, PUT, PATCH e DELETE) têm a correspondência com os métodos de um serviço no *FeathersJS* listados na [Figura 12](#).

```
1 const meuServico = {
2   // GET /path
3   find(params, callback) {},
4   // GET /path/<id>
5   get(id, params, callback) {},
6   // POST /path
7   create(data, params, callback) {},
8   // PUT /path/<id>
9   update(id, data, params, callback) {},
10  // PATCH /path/<id>
11  patch(id, data, params, callback) {},
12  // DELETE /path/<id>
13  remove(id, params, callback) {}
14 }
```

Figura 12 – Interface de um serviço

3.5.4.2 Hooks

Hooks são tecnicamente *middleware* ou funções que têm acesso aos objetos de solicitação (**req**) e resposta (**res**). Dessa forma os *hooks* podem fazer mudanças nos objetos de solicitação e resposta. O *FeathersJS* permite registrar *hooks* antes (*before*), depois (*after*) ou em caso de erro (*error*) dos métodos de um serviço, como mostrado na [Figura 12](#).

Como eles têm acesso ao objetos de uma requisição (**req** e **res**), eles são usados para política de controle de acesso da aplicação, registro de eventos, enviar notificações, adicionar propriedades e muito mais.

Essa abordagem é conhecida como Programação Orientada a Aspectos (POA), que permite a separação de propriedades ortogonais (ou que não fazem parte da funcionalidade principal) dos componentes funcionais de uma forma natural e concisa.

Na [Figura 13](#) três *hooks* foram registrados para um serviço de questões (*questions*) (*ℓ. 1*), em que é adicionado a propriedade **createdAt** antes (*before*) da criação (*create*) de um objeto questão (*ℓ. 3 – 5*), e a propriedade **updatedAt** quando uma questão é modificada (*update* e *patch*), (*ℓ. 7 – 13*).

```
1  app.service('questions').hooks({
2    before: {
3      create(hook) {
4        hook.data.createdAt = new Date();
5      },
6
7      update(hook) {
8        hook.data.updatedAt = new Date();
9      },
10
11     patch(hook) {
12       hook.data.updatedAt = new Date();
13     }
14   }
15 });
```

Figura 13 – Exemplo registro de *hooks* no *FeathersJS*

3.5.4.3 Events

São os *events* ou eventos no *FeathersJS* permitem a criação de aplicações de tempo-real usando *WebSockets*.

No *FeathersJS*, os serviços enviam automaticamente eventos ou notificações **created**, **updated**, **patched**, **removed** quando algum dos respectivos métodos listados na [Figura 12](#) finalizam com sucesso. Os clientes da aplicação podem então ouvir a esses eventos e reagirem de acordo.

Na [Figura 14](#) o cliente obtém uma referência do serviço de votação (*ℓ.2*) e então passa a ouvir quando uma nova votação é criada (**created**) (*ℓ.5 – 7*). Nesse caso, os clientes de uma aplicação de votação, por exemplo, poderiam receber as questões da votação publicada por outro cliente e então responder.

```
1  // Retrieve the wrapped service object which will be an event emitter
2  const poll = app.service('poll');
3
4  // Listen 'created' event
5  poll.on('created', (poll) => {
6    console.log('New poll created', poll);
7  });
```

Figura 14 – Exemplo eventos no *FeathersJS*

3.5.4.4 Visão geral

A [Figura 15](#) ilustra como funciona o ciclo de uma requisição entre clientes e uma aplicação baseada no *FeathersJS*.

O cliente faz uma requisição para um serviço, que antes de chegar no serviço passa pela camada de *before hooks*, o método requisitado pode completar com sucesso indo para

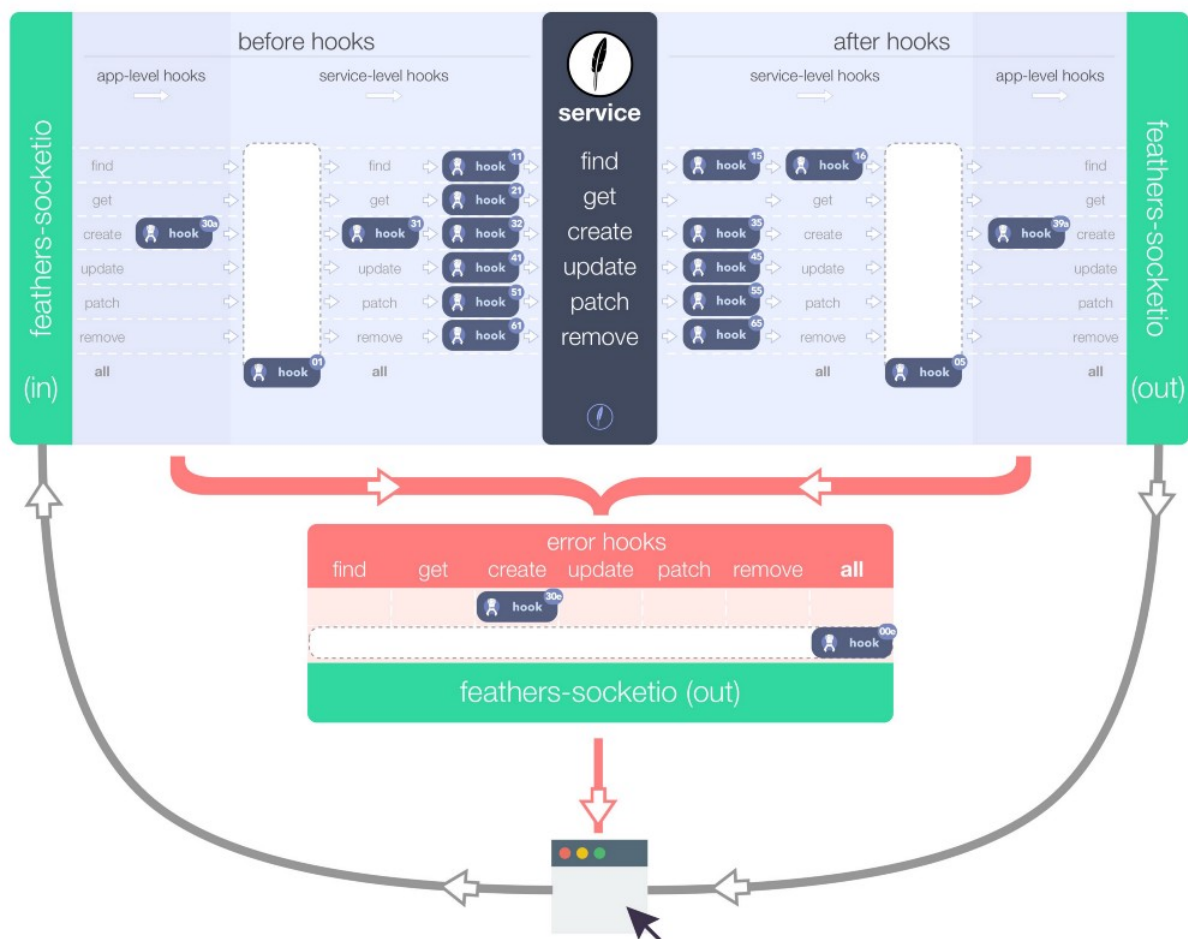


Figura 15 – Como o ciclo de uma requisição funciona no *FeathersJS*

a *after hooks* e enviando um evento para os clientes conectados. Qualquer erro no processo é enviado para a camada *error hooks* que também pode notificar os clientes da aplicação.

A utilização do *FeathersJS* simplificou muito o processo de construção do software por disponibilizar facilidades para o programador e implementar uma API de tempo-real via serviços de forma nativa. Além disso, a interface dos serviços [Figura 12](#) torna fácil a integração com qualquer banco de dados. Nesse sentido, o *FeathersJS* suporta alguns ORM que permitem uma integração com uma variedade de banco de dados por meio de uma interface única.

4 Resposta em Sala de Aula

As principais tecnologias, ferramentas e métodos foram descritos nos capítulos anteriores. Como resultado final foi desenvolvido um sistema de resposta em sala de aula denominado *Resposta em Sala de Aula*.

4.1 Aplicação web para o Professor

A Figura 16 ilustra a tela inicial da aplicação web para o professor. Nela é possível perceber as três abas principais do sistema. A primeira *Questões* que está selecionada, permite ao professor gerir um banco de questões criadas, iniciar uma sessão com questões selecionadas e categorizar as questões para uma melhor organização das mesmas.

Figura 16 – Aba *Questões*



Fonte – do autor (2017)

4.1.1 Frequência dos alunos

Um dos requisitos do sistema era permitir ao professor realizar a frequência dos alunos pelo aplicativo. O sistema gera um código aleatório de quatro dígitos e permite aos estudantes submeterem esse código pelo aplicativo (*ver* FIGURA XX). O professor pode ativar a frequência clicando no botão *Frequência* no canto superior direito da aba *Questões* (*ver* Figura 16). Em seguida, o sistema muda para a aba *Ao Vivo*, tendo como resultado a Figura 17. Nela é possível perceber o código de aleatório de quatro dígitos que foi gerado e alguns botões de ação no canto inferior direito, permitindo ao professor encerrar a frequência por exemplo.







Figura 17 – Página para a realizar a frequência dos estudantes



Fonte – do autor (2017)

Posteriormente, o professor pode acessar a lista de frequência por meio do aba *Turmas*, clicando para cada turma a opção de *atividades*, Figura 18. A Figura 19 exibe a lista de frequência em detalhes.

Figura 18 – Página para a realizar a frequência dos estudantes

← Atividades					
Data	Frequência	Presentes	Ausentes		
27 de Junho de 2017 Ontem às 20:38	0%	0	3		
27 de Junho de 2017 Ontem às 09:08	100%	2	0		
27 de Junho de 2017 Ontem às 08:58	50%	1	1		

Fonte – do autor (2017)

Figura 19 – Página para a realizar a frequência dos estudantes

← Lista de Frequência

Engenharia de Software ◦ 27 de Junho de 2017 ◦ Ontem às 09:08
100% ◦ 2 presentes ◦ 0 ausentes

Nome	Situação
Luiz ID: 102	<input checked="" type="button" value="PRESENTE"/> <input type="button" value="AUSENTE"/>
Pedro ID: 101	<input checked="" type="button" value="PRESENTE"/> <input type="button" value="AUSENTE"/>

Fonte – do autor (2017)

4.1.2 Cadastro de Questões

A [Figura 20](#) exibe o formulário para o cadastro de uma nova questão. A aplicação permite a criação de questões de múltipla escolha, verdadeiro e falso e questão aberta. Em questões do tipo múltipla escolha o professor deve indicar uma alternativa como correta.

4.1.3 Turmas

Na aba de *Turmas* ([Figura 21](#)), o professor faz o cadastro das turmas para permitir acesso aos estudantes. Foi criado o conceito de *turmas públicas* e *turmas privadas*. Uma turma pública é aquele em que não existe uma lista de alunos cadastrados, ou seja, qualquer aluno com o código de acesso da turma vai poder acessar e participar das atividades apenas digitando o código de acesso.

Figura 20 – Formulário *Nova Questão*

Nova Questão CANCELAR CONCLUÍDO

- Digite a questão
- Escolha uma opção como correta
- Adicione pelo menos duas alternativas

Questão

O projeto de arquitetura de um sistema é um processo que visa criar uma...

Descrição

Julgue o item quanto à engenharia de software e à linguagem de modelagem unificada (UML).

Múltipla escolha ☐

Verdadeiro/Falso ☒

Aberta ☐

Correta Verdadeiro ☐ x

Correta Falso ☒ x

ADICIONAR ALTERNATIVA

Fonte – do autor (2017)

Quando uma lista de estudantes é adicionada em uma turma (ver Figura 22) ela se torna uma turma privada. Quando os estudantes acessarem a turma eles terão que digitar além do código da turma, o código de identificação de cada um que foi cadastrado no sistema.

4.1.3.1 Lista de estudantes em uma turma

A Figura 22 ilustra o cadastro de estudantes de uma turma. O professor tem a opção de adicionar manualmente cada estudante, clicando na opção *Adicionar manualmente* no canto inferior direito, digitando o nome do estudante e o código de identificação (esse é o código que os estudantes terão que digitar além do código da sala para ter acesso).

A outra opção para adicionar a lista de estudantes é importar um arquivo CSV contendo o nome dos estudantes e código de identificação, clicando na opção *Importar Estudantes (CSV)* no canto inferior direito. Um exemplo de um arquivo CSV válido para importação é ilustrado na 23. A importação do arquivo CSV trata a primeira linha como o cabeçalho de coluna, e deve conter exclusivamente **name**, **id**. Nas outras linhas deve incluir primeiro o nome do estudante e depois o código de identificação.

Figura 21 – Aba *Turmas*

Turmas				
Nome	Acesso	Alunos	Atividades	Ativar
Engenharia de Software	ENGSOFT2017 (Acesso privado)	3	0	<input checked="" type="checkbox"/>
Tópicos Avançados ES	TAES2017 (Acesso privado)	14	0	<input type="checkbox"/>

+

Questões

Ao Vivo

Turmas

Fonte – do autor (2017)

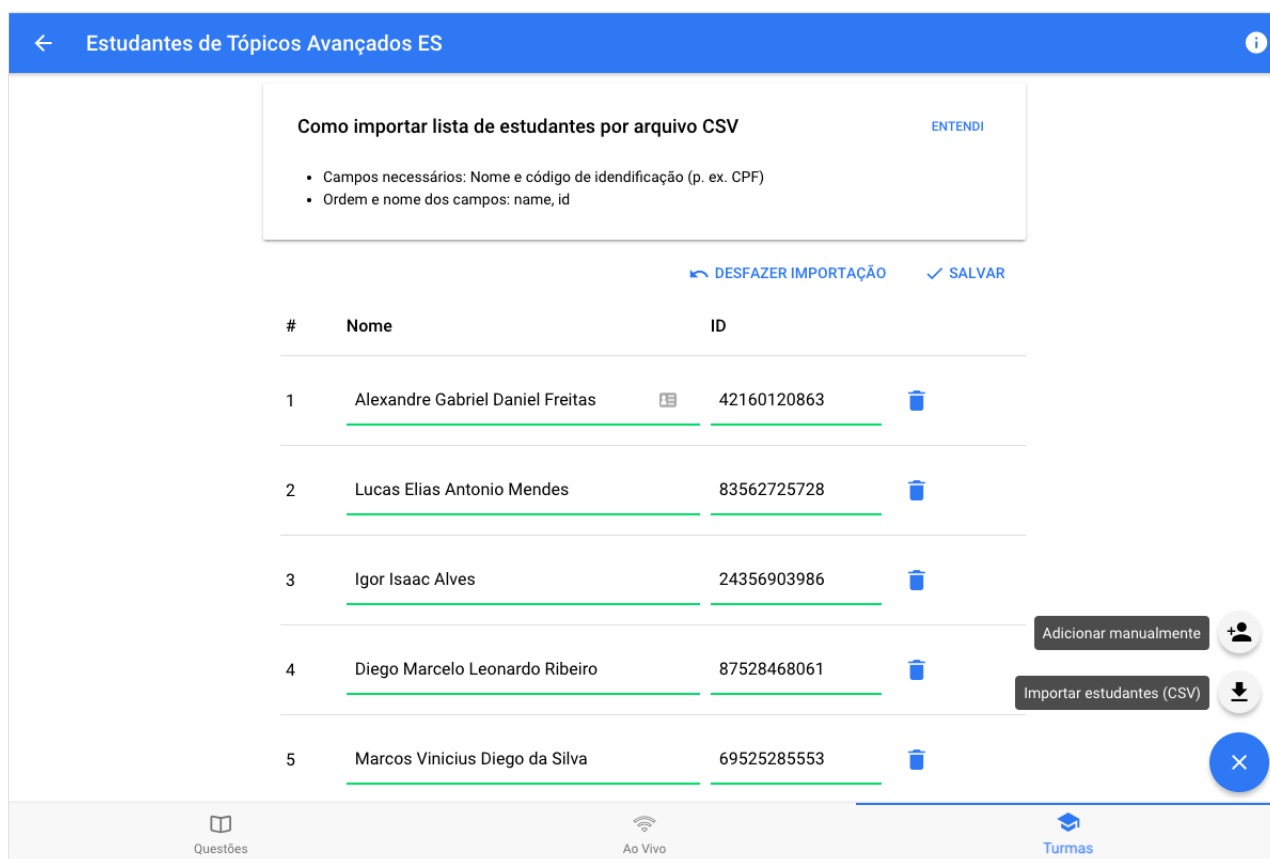
4.1.4 Sessão de questões: Aba Ao Vivo

No sistema tem-se o conceito de *Sessão de questões*, que é simplesmente um conjunto de questões selecionadas pelo professor para apresentar para os estudantes.

O professor pode iniciar uma sessão na aba *Questões*, selecionando uma ou mais questões que deseja apresentar para os estudantes e em seguida clicar no botão *INICIAR SESSÃO*. A Figura 24 destaca da aba *Questões* um exemplo de duas questões selecionadas e o botão *INICIAR SESSÃO* no canto superior direito. Quando o professor inicia uma sessão, o sistema redireciona para a aba *Ao Vivo* projetando a primeira questão selecionada (Figura 25)

A Figura 25 ilustra uma sessão iniciada de uma questão. Uma questão quando iniciada ainda não fica disponível para os estudantes responderem, que está indicado na parte superior da tela com o texto *VOTAÇÃO INDISPONÍVEL*. Nessa mesma parte, o código para acesso da sala fica visível para os estudantes, no caso *ENGSOFT2017*.

No canto inferior direito da Figura 25, o professor tem acesso a um conjunto de funcionalidades para controlar a sessão. A opção *Ativar votação*, quando selecionada permite ao professor disponibilizar para os alunos a questão que está sendo exibida para

Figura 22 – Adicionar estudantes em uma turma

Fonte – do autor (2017)

que eles possam responder no aplicativo (*ver ??*). As opções *Mostrar resposta* e *Mostrar resultado* permitem respectivamente indicar na tela a alternativa correta e exibir na tela como os estudantes responderam. No canto superior direito da tela um número é incrementado indicando a quantidade de alunos que já responderam a questão. A [Figura 26](#) ilustra quando todas essas opções estão selecionadas.

4.1.4.1 Sessões encerradas

Quando uma sessão é encerrada, clicando no botão *Encerrar sessão* no canto inferior direito exibido na [Figura 25](#), uma lista das sessões encerradas é mostrado, indicando o nome da turma, a quantidade de questões, o número de alunos participantes e a data da sessão, como exibido na [Figura 27](#).

Clicando em um item dessa lista, é possível ver como cada estudante respondeu as questões e um desempenho geral da turma por questão. A [Figura 28](#) exibe o desempenho de cada estudante por questão e a percentagem de acerto na sessão.

Figura 23 – Exemplo de arquivo CSV válido para importar uma lista de estudantes

```
1  name,id
2  Alexandre Gabriel Daniel Freitas,42160120863
3  Lucas Elias Antonio Mendes,83562725728
4  Igor Isaac Alves,24356903986
5  Diego Marcelo Leonardo Ribeiro,87528468061
6  Marcos Vinicius Diego da Silva,69525285553
7  Arthur Ryan Barbosa,45863716258
8  Elias Raul Calebe Cardoso,17505265300
9  Paulo Thiago Costa,55883747540
10 Carlos Eduardo Henry Cardoso,24051360318
11 Ricardo Giovanni Bryan Pereira,84740254433
12 Helena Ester Rayssa Costa,84617846859
13 Maria Bruna Cardoso,45713856925
14 Stefany Mariana Sophie Pereira,65113718408
15 Isabel Nina Gomes,16108646020
```

Fonte – do autor (2017)

Figura 24 – Aba *Questões*

2 questões seleccionadas [ADICIONAR MARCADOR](#) [▶ INICIAR SESSÃO](#)

☒ O projeto de arquitetura de um sistema é um processo que visa criar uma orga...
Verdadeiro/Falso • Julgue o item quanto à engenharia de software e à linguagem de modela... [✎](#) [🗑](#)
2017.1 **engenharia de software**

☒ Desde que empregada corretamente, a UML, por meio de diagramas, consegue..
Verdadeiro/Falso [✎](#) [🗑](#)
2017.1 **engenharia de software**

Fonte – do autor (2017)

Figura 25 – Adicionar estudantes em uma turma

The screenshot shows a live classroom interface. At the top, a blue bar says "Ao Vivo". Below it, the poll title is "VOTAÇÃO INDISPONÍVEL ENGSOFT2017" with a green icon and "0" students. The question text is: "O projeto de arquitetura de um sistema é um processo que visa criar uma organização de sistema que satisfaça, exclusivamente, os requisitos funcionais. É um processo cujas atividades são padronizadas e semelhantes para todos os tipos de sistemas que serão desenvolvidos, o que torna mais fácil o trabalho do arquiteto." Below the question are two options: "A) Verdadeiro" and "B) Falso". On the right side, there is a vertical toolbar with buttons: "Tela cheia", "Mostrar resposta", "Mostrar resultado", "Ativar votação", and "Encerrar sessão". At the bottom, there is a navigation bar with icons for "Questões", "Ao Vivo", and "Turmas".

Ao Vivo

VOTAÇÃO INDISPONÍVEL ENGSOFT2017 0

O projeto de arquitetura de um sistema é um processo que visa criar uma organização de sistema que satisfaça, exclusivamente, os requisitos funcionais. É um processo cujas atividades são padronizadas e semelhantes para todos os tipos de sistemas que serão desenvolvidos, o que torna mais fácil o trabalho do arquiteto.

A) Verdadeiro

B) Falso

Tela cheia

Mostrar resposta

Mostrar resultado

Ativar votação

Encerrar sessão

Questões

Ao Vivo

Turmas

Fonte – do autor (2017)

Figura 26 – Adicionar estudantes em uma turma

Ao Vivo

RESPONDA AGORA EM ENGSOFT2017

0% A) Verdadeiro

100% B) Falso

1

O projeto de arquitetura de um sistema é um processo que visa criar uma organização de sistema que satisfaça, exclusivamente, os requisitos funcionais. É um processo cujas atividades são padronizadas e semelhantes para todos os tipos de sistemas que serão desenvolvidos, o que torna mais fácil o trabalho do arquiteto.

Tela cheia
⛶

Esconder resposta
✓

Esconder resultado
📊

Pausar votação
▶

Encerrar sessão
⛔

✕

📖
Questões

📶
Ao Vivo

🎓
Turmas

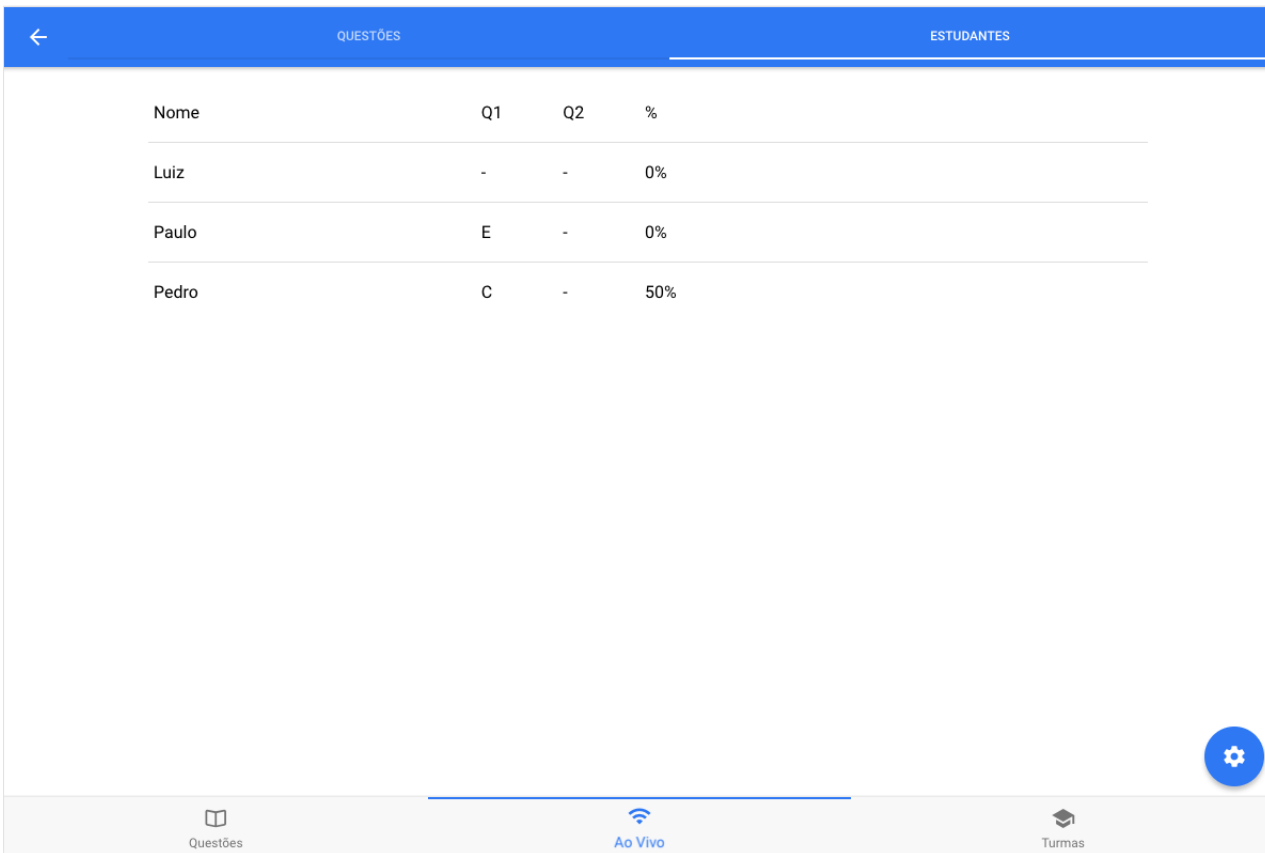
Fonte – do autor (2017)

Figura 27 – Adicionar estudantes em uma turma

Sessões encerradas

Nome	#	Questões	Alunos	Última atividade
Engenharia de Software	ENGSOFT2017	☰ 2	👤 0	🕒 15 horas atrás
Engenharia de Software	ENGSOFT2017	☰ 2	👤 0	🕒 15 horas atrás
Engenharia de Software	ENGSOFT2017	☰ 2	👤 0	🕒 15 horas atrás
Engenharia de Software	ENGSOFT2017	☰ 1	👤 0	🕒 um dia atrás
Engenharia de Software	ENGSOFT2017	☰ 1	👤 0	🕒 um dia atrás
Engenharia de Software	ENGSOFT2017	☰ 2	👤 0	🕒 um dia atrás
Engenharia de Software	ENGSOFT2017	☰ 1	👤 0	🕒 um dia atrás

Fonte – do autor (2017)

Figura 28 – Adicionar estudantes em uma turma

The screenshot shows a mobile application interface for classroom management. At the top, there is a blue header bar with a back arrow on the left, a tab labeled 'QUESTÕES' in the center, and a tab labeled 'ESTUDANTES' on the right. Below the header, a table lists three students: Luiz, Paulo, and Pedro. The table has four columns: 'Nome', 'Q1', 'Q2', and '%'. Luiz has scores of '-' for Q1 and Q2, and 0% for the percentage. Paulo has a score of 'E' for Q1, '-' for Q2, and 0% for the percentage. Pedro has a score of 'C' for Q1, '-' for Q2, and 50% for the percentage. At the bottom of the screen, there is a grey navigation bar with three icons: a book icon labeled 'Questões', a Wi-Fi icon labeled 'Ao Vivo', and a graduation cap icon labeled 'Turmas'. A blue circular settings icon with a gear symbol is located in the bottom right corner of the screen.

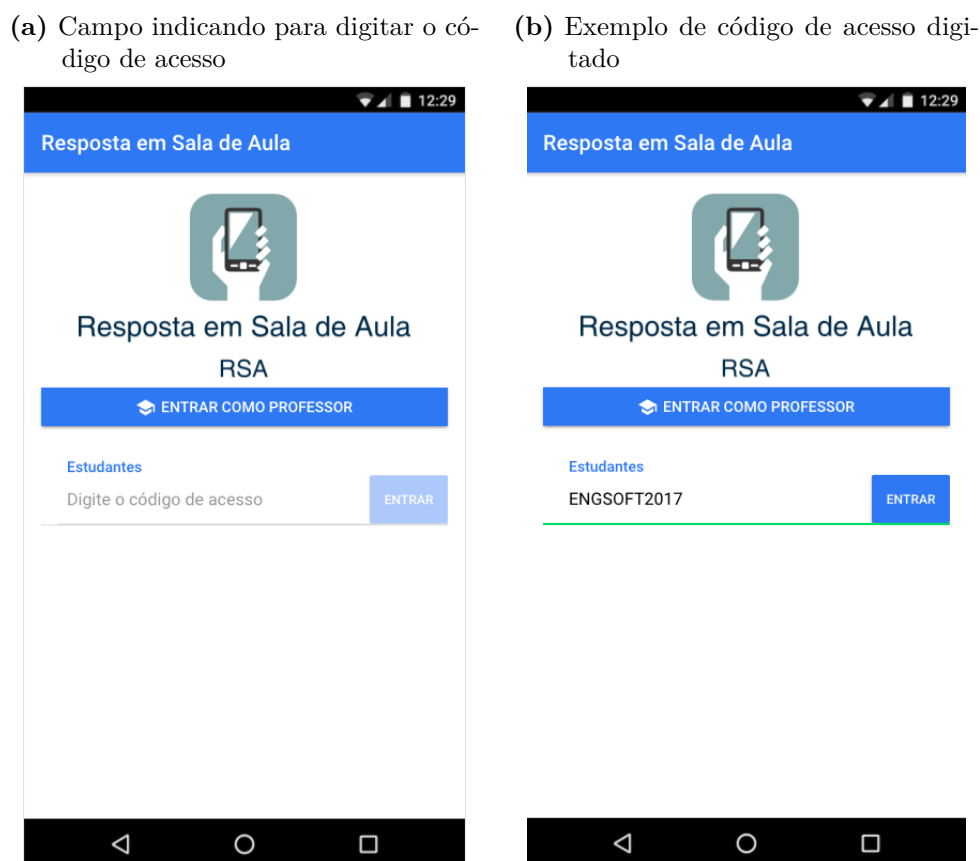
Nome	Q1	Q2	%
Luiz	-	-	0%
Paulo	E	-	0%
Pedro	C	-	50%

Fonte – do autor (2017)

4.2 Aplicação para dispositivos móveis, que será utilizado como *clickers*

A [Figura 29](#) exibe a tela inicial do aplicativo em que é possível perceber um campo para que os estudantes possam digitar o código de acesso da turma, conforme detalhou-se na [subseção 4.1.3](#).

Figura 29 – Tela inicial da aplicação móvel do estudante



Fonte – do autor (2017)

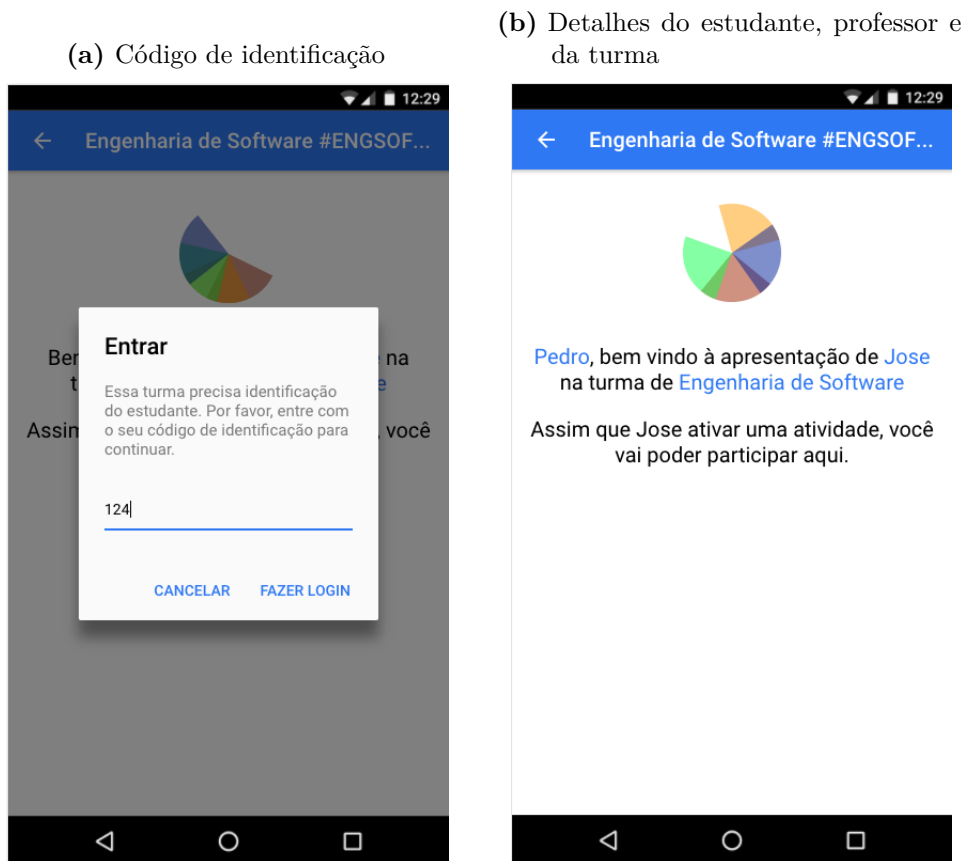
4.2.1 Identificação dos estudantes

Conforme apresentado na [subseção 4.1.3](#) tem-se o conceito de turmas públicas (que permitem acesso a turma apenas com o código da turma) e as turmas privadas que além do código da turma é necessário o código de identificação do estudante cadastrado previamente pelo professor.

A [Figura 30](#) ilustra o processo de identificação do estudante considerando que o acesso a turma é privado. Na [Figura 30a](#) o aplicativo exibe uma mensagem e um campo para que o estudante digite o código de identificação único do estudante. A [Figura 30b](#),

exibe a tela quando o estudante digita o código correto, detalhando o nome do estudante, do professor e da turma.

Figura 30 – Tela de identificação dos estudantes



Fonte – do autor (2017)

4.2.2 Responder frequência

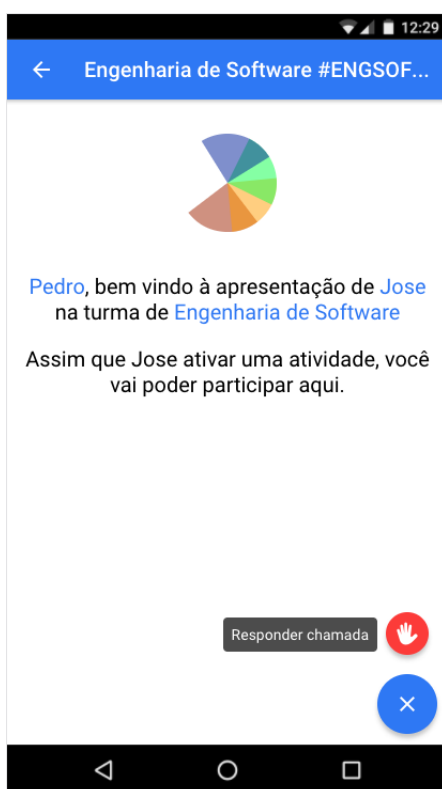
Na subseção 4.1.1 tem-se o processo para o professor realizar a frequência dos estudantes. Quando o professor habilita para os estudantes enviarem o código de quatro dígitos gerado, um botão aparece no aplicativo dos estudantes com a opção *Responder chamada* no canto inferior direito, como mostra na Figura 31a. Em seguida, quando o estudante clica nesse botão, o aplicativo exibe uma mensagem e um campo para que os estudantes digitem o código disponibilizado pelo professor (Figura 31b).

4.2.3 Responder questões

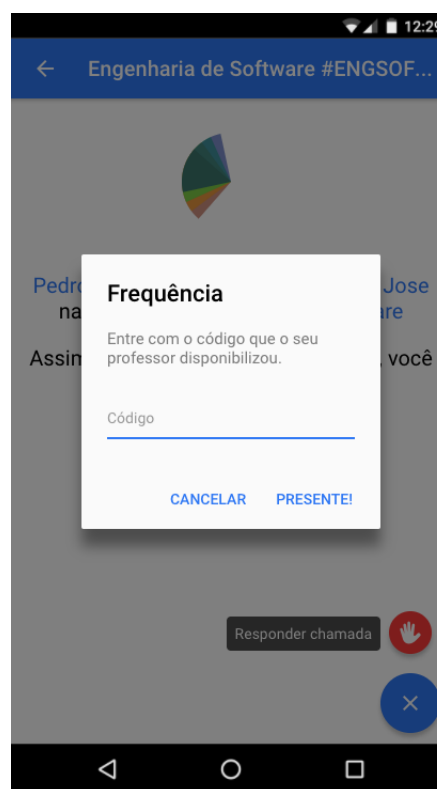
O sistema permite ao professor criar questões de múltipla escolha e de resposta livre. A Figura 32 ilustra como fica apresentado no aplicativo as questões. A Figura 32a exibe uma questão de verdadeiro e falso e a Figura 32b ilustra uma questão aberta.

Figura 31 – Telas para os estudantes realizarem a frequência

(a) Botão para responder a chamada



(b) Tela com campo para digitar código



Fonte – do autor (2017)

Figura 32 – Diferentes tipos de questões no aplicativo

(a) Questão do tipo múltipla escolha

The screenshot shows a mobile application interface with a blue header bar containing a back arrow and the text "Engenharia de Software #ENGSOFT...". The status bar at the top shows the time as 12:29. The main content area contains a paragraph of text: "O projeto de arquitetura de um sistema é um processo que visa criar uma organização de sistema que satisfaça, exclusivamente, os requisitos funcionais. É um processo cujas atividades são padronizadas e semelhantes para todos os tipos de sistemas que serão desenvolvidos, o que torna mais fácil o trabalho do arquiteto. Julgue o item quanto à engenharia de software e à linguagem de modelagem unificada (UML).". Below the text are two radio button options: "Verdadeiro" and "Falso". At the bottom of the content area is a blue button labeled "ENVIAR". The Android navigation bar is visible at the very bottom.

(b) Questão do tipo resposta curta

The screenshot shows a mobile application interface with a blue header bar containing a back arrow and the text "Engenharia de Software #ENGSOFT...". The status bar at the top shows the time as 12:29. The main content area contains the question "Qual a sua cor favorita?". Below the question is a text input field with the placeholder text "Resposta" and "Digite aqui a sua resposta". At the bottom of the content area is a blue button labeled "ENVIAR". The Android navigation bar is visible at the very bottom.

Fonte – do autor (2017)

Referências

- ABRAHAMSON, L.; BRADY, C. A Brief History of Networked Classrooms to 2013:. **International Journal of Quality Assurance in Engineering and Technology Education**, v. 3, n. 3, p. 1–54, 2014. ISSN 2155-496X. Citado na página 15.
- ARAÚJO, I. S.; MAZUR, E. Instrução pelos colegas e ensino sob medida: uma proposta para o engajamento dos alunos no processo de ensino-aprendizagem de Física. **Caderno Brasileiro de Ensino de Física**, v. 30, n. 2, p. 362–384, 2013. ISSN 2175-7941. Citado 3 vezes nas páginas 10, 13 e 16.
- ARNESEN, K. et al. Experiences with use of various pedagogical methods utilizing a student response system - Motivation and learning outcome. **Electronic Journal of e-Learning**, v. 11, n. 3, p. 169–181, 2013. ISSN 14794403. Citado na página 20.
- BARRAGUÉS, J.; MORAIS, A.; GUIASOLA, J. Use of a classroom response system (CRS) for teaching mathematics in Engineering with large groups. **Education in a technological world: communicating current and emerging research and technological efforts**, p. 572–580, 2011. Citado na página 20.
- BARROS, J. A. D. et al. Engajamento interativo no curso de Física I da UFJF. **Revista Brasileira de Ensino de Física**, v. 26, n. 1, p. 63–69, 2004. Citado na página 13.
- BLASCO-ARCAS, L. et al. Using clickers in class. the role of interactivity, active collaborative learning and engagement in learning performance. **Computers and Education**, Elsevier Ltd, v. 62, p. 102–110, 2013. ISSN 03601315. Citado na página 16.
- BLOOD, E.; GULCHAK, D. Embedding “Clickers” Into Classroom Instruction: Benefits and Strategies. **Intervention in School & Clinic**, v. 48, n. 4, p. 246–253, 2013. ISSN 1053-4512. Citado 2 vezes nas páginas 19 e 20.
- BRASIL, P. **Portal do Professor disponibiliza lista de livros sobre novas tecnologias**. 2014. Disponível em: <<http://www.brasil.gov.br/educacao/2014/07/portal-do-professor-disponibiliza-lista-de-livros-sobre-novas-tecnologias#wrapper>>. Acesso em: 18 Ago. 2016. Citado na página 21.
- CALDWELL, J. E. Clickers in the Large Classroom: Current Research and Best-Practice Tips. **CBE - Life Sciences Education**, v. 6, p. 1–15, 2007. ISSN 0004-069X. Citado 4 vezes nas páginas 17, 19, 20 e 21.
- CHARLES, C.; JAMES, A. **Active Learning: Creating Excitement in the Classroom**. Ashe-eric. Washington, D.C: The George Washington University, School of Education and Human Development, 1991. 121 p. ISBN 1878380087. Citado na página 12.
- CROUCH, C. H.; MAZUR, E. Peer instruction: Ten years of experience and results. **American Journal of Physics**, v. 69, n. 9, p. 970–977, 2001. ISSN 00029505. Citado 2 vezes nas páginas 13 e 14.

CROUCH, C. H. et al. Peer Instruction: Engaging Students One-on-One, All At Once. **Research-Based Reform of University Physics**, p. 1–55, 2007. ISSN 10476938. Citado 2 vezes nas páginas 14 e 20.

CUBRIC, M.; JEFFERIES, A. The benefits and challenges of large-scale deployment of electronic voting systems: University student views from across different subject groups. **Computers and Education**, Elsevier Ltd, v. 87, p. 98–111, 2015. ISSN 03601315. Citado na página 21.

D'INVERNO, R. A.; DAVIS, H. C.; WHITE, S. Using a personal response system for promoting student interaction. **Teaching Mathematics and its Application**, v. 22, n. 4, p. 163–169, 2003. ISSN 02683679. Citado na página 19.

DRIFTY. **Ionic: Advanced HTML5 Hybrid Mobile App Framework**. 2016. Disponível em: <<http://ionicframework.com/>>. Acesso em: 20 Ago. 2016. Citado na página 28.

FIES, C.; MARSHALL, J. Classroom response systems: A review of the literature. **Journal of Science Education and Technology**, v. 15, n. 1, p. 101–109, 2006. ISSN 10590145. Citado 2 vezes nas páginas 15 e 21.

FOTARIS, P. et al. Climbing Up the Leaderboard : An Empirical Study of Applying Gamification Techniques to a Computer Programming Class. **Electronic Journal of E-Learning**, v. 14, n. 2, 2016. ISSN 14794403. Citado na página 17.

FREEMAN, S. et al. Active learning increases student performance in science, engineering, and mathematics. **Proceedings of the National Academy of Sciences of the United States of America**, v. 111, n. 23, p. 8410–5, 2014. ISSN 1091-6490. Citado na página 12.

GOK, T. A comparison of students' performance, skill and confidence with peer instruction and formal education. **Journal of Baltic Science Education**, v. 12, n. 6, p. 747–758, 2013. ISSN 16483898. Citado na página 13.

HAN, J. H. Closing the missing links and opening the relationships among the factors: A literature review on the use of clicker technology using the 3P model. **Educational Technology and Society**, v. 17, n. 4, p. 150–168, 2014. ISSN 14364522. Citado na página 21.

HORNE, A. **An Evaluation of an Electronic Student Response System in Improving Class-wide Behavior**. 56 p. Tese (Graduate Theses and Dissertations) — University of South Florida, 2015. Citado na página 19.

HUNSU, N. J.; ADESOPE, O.; BAYLY, D. J. A meta-analysis of the effects of audience response systems (clicker-based technologies) on cognition and affect. **Computers and Education**, Elsevier Ltd, v. 94, p. 102–119, 2016. ISSN 03601315. Citado 3 vezes nas páginas 10, 19 e 20.

KAY, R. H.; LESAGE, A. A strategic assessment of audience response systems used in higher education. **Australasian Journal of Educational Technology**, v. 25, n. 2, p. 235–249, 2009. ISSN 14495554. Citado 6 vezes nas páginas 10, 15, 16, 17, 20 e 21.

- KAYA, A.; BALTA, N. Taking Advantages of Technologies: Using the Socrative in English Language Teaching Classes. **International Journal of Social Sciences & Educational Studies**, v. 2, n. 3, p. 4–12, 2016. Citado 2 vezes nas páginas 17 e 19.
- KORTEMAYER, G. The Psychometric Properties of Classroom Response System Data: A Case Study. **Journal of Science Education and Technology**, Springer Netherlands, n. March, p. 1–14, 2016. ISSN 15731839. Citado na página 21.
- KORTEMAYER, G.; DIAS, E.; CRUZ, H. The effect of formative assessment in Brazilian university physics courses. **Revista Brasileira de Ensino de Física**, v. 33, n. n. 4, p. 4501, 2011. ISSN 01024744. Citado 2 vezes nas páginas 15 e 16.
- KULATUNGA, U.; RAMEEZDEEN, R. Use of Clickers to Improve Student Engagement in Learning: Observations from the Built Environment Discipline. **International Journal of Construction Education and Research**, v. 10, n. 1, p. 3–18, 2014. ISSN 1557-8771. Citado na página 19.
- LANTZ, M. E.; STAWISKI, A. Effectiveness of clickers: Effect of feedback and the timing of questions on learning. **Computers in Human Behavior**, Elsevier Ltd, v. 31, n. 1, p. 280–286, 2014. ISSN 07475632. Citado 2 vezes nas páginas 17 e 20.
- MATTOS, E. B. D. A. **Imunologia e Biotecnologia: o clicker como recurso didático estratégico**. 109 p. Tese (Dissertação de Mestrado) — Universidade Federal Fluminense, 2015. Citado 4 vezes nas páginas 10, 17, 19 e 20.
- MAYER, R. E. et al. Clickers in college classrooms: Fostering learning with questioning methods in large lecture classes. **Contemporary Educational Psychology**, Elsevier Inc., v. 34, n. 1, p. 51–57, 2009. ISSN 0361476X. Citado 2 vezes nas páginas 17 e 20.
- MAZUR, E. Farewell, Lecture? **Science**, v. 323, p. 50–51, 2009. ISSN 1095-9203. Citado 2 vezes nas páginas 13 e 14.
- MOODLE. **Moodle Mobile 2.0 spec**. 2015. Disponível em: <https://docs.moodle.org/dev/Moodle_Mobile_2.0_spec>. Acesso em: 20 Ago. 2016. Citado na página 28.
- MORAN, J. M.; MASETO, M. T.; BEHRENS, M. A. **Novas Tecnologias e mediação pedagógica**. 10. ed. Campinas, SP: Papirus, 2006. 173 p. ISBN 85-308-0594-1. Citado na página 21.
- MORATELLI, K.; DEJARNETTE, N. K. Clickers to the rescue. **The Reading Teacher**, v. 67, n. 8, p. 586–593, 2014. ISSN 00340561. Citado 2 vezes nas páginas 16 e 19.
- MORRELL, L. J.; JOYCE, D. A. Interactive lectures: Clickers or personal devices? **F1000Research**, v. 64, p. 1–12, 2015. ISSN 2046-1402. Citado 2 vezes nas páginas 16 e 21.
- NEILSON, M. et al. Students perceptions regarding the use of tophat as an interactive tool in meat science clall. **Meat Science**, 2016. Citado na página 17.
- PRESSMAN, R. S. **Software Engineering A Practitioner's Approach**. 7. ed. New York, NY 10020: McGraw-Hill, 2009. 930 p. ISSN 1098-6596. ISBN 978-0-07-337597-7. Citado na página 22.

PRINCE, M. Does active learning work? A review of the research. **Journal of Engineering education- Washington**, v. 93, n. July, p. 223–232, 2004. ISSN 1069-4730. Citado na página 12.

PUENTE, S. G.; SWAGTEN, H. Designing learning environments to teach interactive Quantum Physics. **European Journal of Engineering Education**, v. 37, n. June 2013, p. 37–41, 2012. ISSN 03043797. Citado 2 vezes nas páginas 17 e 19.

RANA, N. P.; DWIVEDI, Y. K. Using Clickers in a Large Business Class: Examining Use Behavior and Satisfaction. **Journal of Marketing Education**, v. 38, n. 1, p. 47–64, 2016. ISSN 02734753. Citado 2 vezes nas páginas 19 e 20.

SCHMIDT, B. Teaching engineering dynamics by use of peer instruction supported by an audience response system. **European Journal of Engineering Education**, v. 36, n. 5, p. 413–423, 2011. ISSN 0304-3797. Citado na página 19.

SOCRATIVE. 2016. Disponível em: <<http://socrative.com/pricing>>. Acesso em: 16 Ago. 2016. Citado na página 17.

SOMMERVILLE, I. **Software Engineering: (Update) (8th Edition) (International Computer Science)**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006. ISBN 0321313798. Citado na página 22.

STOWELL, J. R. Use of clickers vs. mobile devices for classroom polling. **Computers and Education**, Elsevier Ltd, v. 82, p. 329–334, 2015. ISSN 03601315. Citado 2 vezes nas páginas 16 e 21.

STRASSER, N. Who Wants to Pass Math? Using Clickers in Calculus. **Journal of College Teaching Learning**, v. 7, n. 3, p. 49–52, 2010. ISSN 15440389. Citado 4 vezes nas páginas 16, 17, 20 e 21.

SUN, J. C. Y. Influence of polling technologies on student engagement: An analysis of student motivation, academic performance, and brainwave data. **Computers and Education**, Elsevier Ltd, v. 72, p. 80–89, 2014. ISSN 03601315. Citado 2 vezes nas páginas 19 e 20.

SYSTEMS and software engineering – Vocabulary. **ISO/IEC/IEEE 24765:2010(E)**, p. 1–418, Dec 2010. Citado na página 22.

TERRION, J. L.; ACETI, V. Perceptions of the effects of clicker technology on student learning and engagement: A study of freshmen Chemistry students. **Research in Learning Technology**, v. 20, n. 2, 2012. ISSN 21567069. Citado 3 vezes nas páginas 19, 20 e 21.

THAMPY, H.; AHMAD, Z. How to. Use audience response systems. **Education for Primary Care**, v. 25, n. 5, p. 294–296, 2014. ISSN 14739879. Citado na página 21.

TITMAN, A.; LANCASTER, G. Personal response systems for teaching postgraduate statistics to small groups. **Journal of Statistics Education**, v. 19, n. 2, p. 1–20, 2011. ISSN 10691898. Citado na página 20.

TOPHAT. 2016. Disponível em: <<http://tophat.com/customers/>>. Acesso em: 16 Ago. 2016. Citado na página 17.

TRINDADE, J. Promoção da interatividade na sala de aula com Socrative: estudo de caso. **Indagatio Didactica**, v. 6, n. 1, 2014. Citado na página 17.

UNIVESPTV. **D-29 - Avaliação da Aprendizagem: Formativa ou Somativa?** 2013. Disponível em: <<https://youtu.be/G5VEkMf5DRk>>. Acesso em: 16 Ago. 2016. Citado na página 20.

VELASCO, M.; ÇAVDAR, G. Teaching Large Classes with Clickers: Results from a Teaching Experiment in Comparative Politics. **PS: Political Science & Politics**, v. 46, n. 04, p. 823–829, 2013. ISSN 1049-0965. Citado 2 vezes nas páginas 17 e 21.

WATKINS, B. J.; MAZUR, E. Retaining Students in Science, Technology, Engineering, and Mathematics (STEM) Majors. **Journal of College Science Teaching**, v. 42, n. 5, p. 36–41, 2013. ISSN 0047-231X. Citado na página 14.

Apêndices

APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Declaro que estou sendo convidado(a) a participar do teste de usabilidade de um protótipo de sistema da área de sistemas de resposta para uso em sala de aula, desenvolvido no contexto do Trabalho de Conclusão de Curso do curso de Engenharia de Computação da Universidade Federal do Vale do São Francisco intitulado: Desenvolvimento de um sistema de resposta para uso em sala de aula, realizado pelo aluno/pesquisador Pedro Henrique Araújo Sobral - pedrosobralxv@gmail.com, sendo orientado pelo Prof. Dr. Max Santana Rolemberg Farias - max.santana@univasf.edu.br, cujo objetivo é avaliar e melhorar a usabilidade do protótipo desenvolvido.

A minha participação no referido projeto será no sentido de auxiliar na identificação de pontos fortes e fracos no design de interface deste protótipo por meio da realização de pequenas tarefas no protótipo e ao final das tarefas o preenchimento de um questionário composto por 10 questões de múltipla escolha.

Recebi esclarecimentos sobre a pesquisa e estou ciente de que minha privacidade será respeitada, ou seja, meu nome será mantido em sigilo.

Eu autorizo a gravação de áudio e vídeo durante os testes de usabilidade e entendo que as gravações de áudio e vídeo serão utilizadas somente para os fins desta pesquisa e não serão divulgados fora do contexto desta pesquisa.

Fui informado(a) de que posso me recusar a participar do estudo, ou retirar meu consentimento a qualquer momento, sem precisar justificar.

É assegurada a assistência durante toda a pesquisa.

Manifesto meu livre consentimento em participar.