HOLOS
ISSN 1807 - 1600

ARQUITETURA DE *SOFTWARE* VOLTADA PARA A AVALIAÇÃO CONTÍNUA DO PROCESSO DE ENSINO-APRENDIZAGEM

D. M. Cunha[1], A. M. Filgueira[2], G. L. Viana[3]
[1, 3]Instituto Federal do Rio Grande do Norte, [2]Universidade Federal do Rio Grande do Norte
dannilo.martins@ifrn.edu.br [1]

Submetido 11/11/2016 - Aceito 14/02/2017

DOI: 10.15628/holos.2017.5335

RESUMO

Este trabalho tem o objetivo de apresentar uma arquitetura de *software* que tenha a capacidade de possibilitar a construção de ferramentas que realizem a avaliação contínua do processo de ensino-aprendizagem utilizado por professores. Mais especificamente, essa avaliação visa realizar um mapeamento do entendimento dos alunos em relação aos assuntos abordados em sala de aula por cada professor. Além de apresentar a arquitetura propriamente dita, este trabalho tem também o propósito de mostrar os resultados obtidos a partir da validação de tal arquitetura através de três estudos de caso realizados no campus Lajes do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN).

PALAVRAS-CHAVE: Arquitetura, ensino, aprendizagem, ensino-aprendizagem, processo de ensino-aprendizagem.

SOFTWARE ARCHITECTURE FOR THE CONTINUOUS EVALUATION OF THE TEACHING-LEARNING PROCESS

ABSTRACT

This work aims to present a software architecture that has the capacity to enable the construction of tools that perform the continuous evaluation of the teaching-learning process used by teachers. More specifically, this evaluation aims at mapping the students' understanding of the subjects addressed in the classroom by each teacher. In addition to presenting the architecture itself, this work also aims to show the results obtained from the validation of such architecture through three case studies carried out at the Lajes campus of the Federal Institute of Education, Science and Technology of Rio Grande do Norte (IFRN).

KEYWORDS: Architecture, teaching, learning, teaching-learning, teaching-learning process.

HOLOS
ISSN 1807 - 1600

1 INTRODUÇÃO

1.1 Problemática

Após a ministração de um novo conteúdo em sala de aula, o professor normalmente se depara com a seguinte dúvida: os alunos realmente entenderam o novo assunto abordado durante a aula? Logo após o enfrentamento de tal pergunta, o professor recorre a uma técnica de mapeamento muito comum no meio acadêmico. Essa técnica corresponde a seguinte questão: quem entendeu (ou quem não entendeu) o novo conteúdo ministrado? Geralmente, o aluno responde ao professor com o silêncio! Esse tipo de *feedback* não corresponde à pergunta constituída e realizada pelo professor. O problema desse tipo de resposta está centrado no fato do professor não saber se continua o desenvolvido de sua disciplina ou se realiza uma revisão do novo assunto abordado. A partir disso, surge a necessidade de buscar novos mecanismos que possibilitem mapear o entendimento do aluno logo após a ministração de novos conteúdos em sala de aula.

1.2 Objetivos

O objetivo deste trabalho é construir uma arquitetura de *software* voltada para a avaliação contínua de alunos durante o processo de ensino-aprendizagem. Ademais, este trabalho apresentará os resultados obtidos através da realização de três estudos de caso realizados no campus Lajes do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN).

1.3 Resultados esperados

Após o término deste trabalho, espera-se que a arquitetura de software proposta seja aceita e utilizada pelo meio acadêmico a fim de possibilitar a construção de novas ferramentas voltadas para a avaliação contínua de alunos durante o processo de ensino-aprendizagem.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Arquitetura de *software*

Para alguns autores, a arquitetura de *software* pode ser vista como sendo um projeto de alto nível que permite aos engenheiros de *software* ganharem rápido entendimento sobre um sistema de *software* [1]. Para outros autores, a arquitetura de um sistema de *software* é a estrutura ou estruturas do sistema, que compreendem os elementos de *software*, as propriedades externamente visíveis desses elementos e o relacionamento entre eles [2]. Já para outros autores, a arquitetura de *software* é a representação que permite ao engenheiro de *software* (1) analisar a efetividade do projeto em satisfazer a seus requisitos declarados, (2) considerar alternativas arquiteturais em um estágio em que fazer modificações de projeto é ainda relativamente fácil, e (3) reduzir os riscos associados à construção do *software* [3]. A partir das definições expostas, pode-se observar com extrema clareza que a arquitetura de *software* não é o *software* operacional propriamente dito. Em um nível mais simplista, pode-se considerar a arquitetura de *software* como sendo um traçado inicial e abrangente de seus componentes.

HOLOS
ISSN 1807 - 1600

2.2 Estilos arquiteturais

Um estilo arquitetural é uma transformação imposta sobre o projeto de um sistema completo [5]. O objetivo é estabelecer uma estrutura para todos os componentes do sistema [5]. No caso em que uma arquitetura existente deve ser submetida a reengenharia, a imposição de um estilo arquitetural resultará em modificações fundamentais na estrutura do *software*, inclusive uma reatribuição da funcionalidade dos componentes [5].

De acordo com [4], [6] e [7], apesar de milhões de sistemas baseados em computador terem sido criados ao longo dos últimos 50 anos, a grande maioria pode ser categorizada em um dos relativamente poucos estilos arquiteturais apresentados na lista a seguir:

- Arquitetura centrada nos dados.
- Arquitetura de fluxo de dados.
- Arquitetura de chamada e retorno.
- Arquitetura orientada a objetos.
- Arquitetura em camadas.

O estilo arquitetural Arquitetura centrada nos dados possui um depósito de dados, que fica no centro dessa arquitetura e dá acesso a outros componentes que modificam os dados contidos no depósito. Em alguns casos, o repositório de dados é passivo, ou seja, o *software*-cliente tem acesso aos dados independentemente de quaisquer alterações nos dados ou das ações dos outros *softwares*-clientes. Uma variante dessa abordagem transforma o repositório em um "quadro-negro", que envia notificações ao *software*-cliente quando dados de seu interesse sofrem modificações. A figura 1 ilustra um estilo arquitetural comum centrado nos dados.

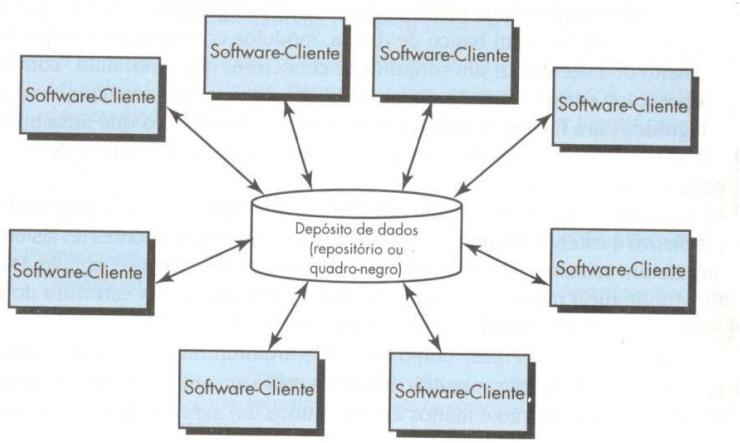

Figura 1: Estilo arquitetural centrado nos dados.

O estilo arquitetural Arquitetura de fluxo de dados é aplicado quando dados de entrada devem ser transformados, por meio de uma série de componentes computacionais ou manipulativos, em dados de saída. Neste caso, uma estrutura tubo e filtro tem um conjunto de componentes, denominados de filtros, conectados por tubos que transmitem dados de um componente para o próximo. Cada filtro é projetado para esperar entrada de dados de uma certa forma e produzir dados de saída de um modo específico. No entanto, o filtro não exige conhecimento do trabalho dos filtros vizinhos. Se o fluxo de dados se degenera em uma única linha de transformações, é chamado sequencial por lotes. Essa estrutura aceita um lote de dados

HOLOS ISSN 1807 - 1600

e então aplica uma série de componentes sequenciais para transformá-lo. A figura 2 ilustra um estilo arquitetural típico de fluxo de dados.

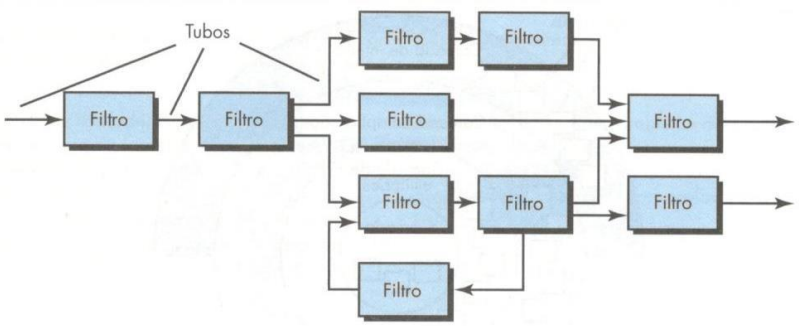

Figura 2: Estilo arquitetural de fluxo de dados.

O estilo arquitetural Arquitetura de chamada e retorno permite ao projetista de *software* conseguir uma estrutura de programa relativamente fácil de alterar e ampliar. Dois subestilos existem nessa categoria: arquiteturas de programa principal/subprograma; e arquiteturas de chamada de procedimentos remotos. O primeiro subestilo decompõe a função em uma hierarquia de controle em que um programa principal aciona um certo número de componentes de programa, que, por sua vez, invocam outros componentes. E o segundo subestilo distribui os componentes entre vários computadores em uma rede. A figura 3 ilustra um estilo arquitetural comum de chamada e retorno.

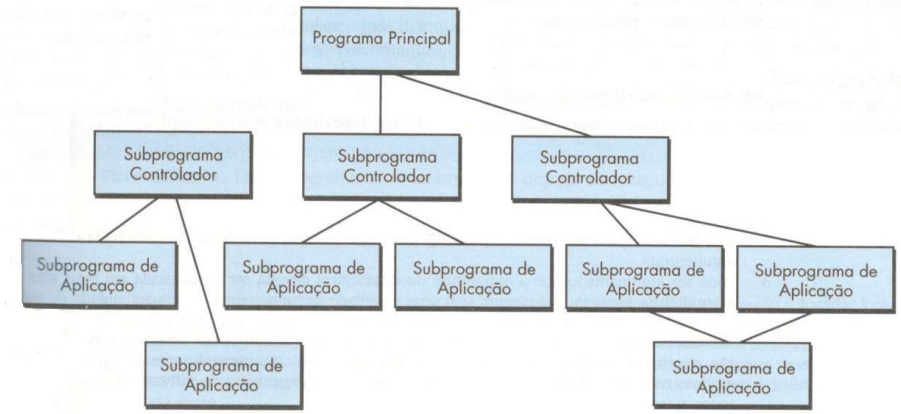

Figura 3: Estilo arquitetural de chamada e retorno.

O estilo arquitetural Arquitetura orientada a objetos apresenta componentes que encapsulam os dados e as operações que devem ser aplicadas para manipular esses mesmos dados. A comunicação e a coordenação entre componentes são obtidas por meio de passagens de mensagens.

E, por fim, o estilo arquitetural Arquitetura em camadas possui um certo número de camadas distintas. Cada uma dessas camadas realiza operações que tornam progressivamente mais próximas do conjunto de instruções de máquina. Na camada mais exterior, os componentes servem as operações da interface com o usuário. Na camada seguinte, os componentes realizam

HOLOS
ISSN 1807 - 1600

a interface com o sistema operacional. Nas camadas intermediárias tem-se serviços utilitários e funções do *software* de aplicação. A figura 4 ilustra um estilo arquitetural típico em camadas.

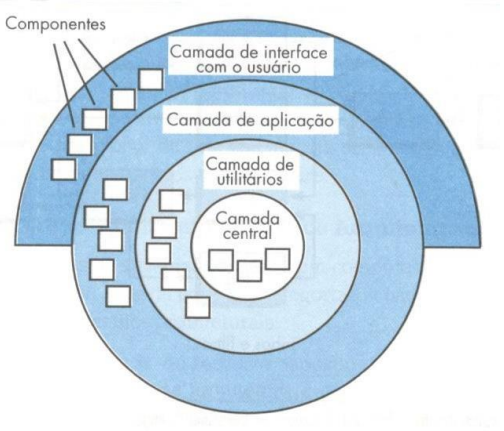

Figura 4: Estilo arquitetural em camadas.

Os estilos arquiteturais supramencionados fazem parte de um pequeno subconjunto dos estilos que estão disponíveis para os arquitetos de *software*. Depois que a engenharia de requisitos descobre as características funcionais e não funcionais do sistema a ser construído, o estilo arquitetural ou combinação de estilos pode ser selecionado. No caso deste trabalho, o estilo arquitetural que melhor se aplica é o estilo arquitetural em camadas, visto que esse estilo arquitetural possibilita dividir um sistema em módulos. Isso significa dizer que os desenvolvedores poderão trabalhar em um mesmo sistema simultaneamente, aumentando a velocidade com que o sistema poderá ser construído. Além disso, as partes do sistema poderão sofrer alterações de maneira independente, sem afetar as demais partes.

3 TRABALHOS RELACIONADOS

3.1 Aumento da interação durante as aulas

Em [12], os pesquisadores descobriram que a interação efetiva entre professores e alunos pode ter efeitos positivos sobre o desempenho acadêmico do aluno. Maior rendimento acadêmico foi relatado em palestras com interação entre o professor e os alunos, ao contrário de palestras convencionais, nas quais os alunos apenas ouviram o professor.

Simoni propôs usar *tablets* para promover a participação ativa dos alunos [13]. O professor explicou o conteúdo da palestra usando um *tablet*, compartilhando notas manuscritas, sublinhando partes importantes e enviando diagramas de circuito complexos para os *tablets* dos alunos. Como resultado, o professor passou menos tempo em diagramas de circuito e os alunos foram capazes de acompanhar a palestra com mais facilidade.

Liu *et al*. apresentou outro caso que demonstrou interação efetiva entre o professor e os alunos [14]. O sistema *EduClick* foi utilizado para implementar entrevistas ou pesquisas durante a palestra. Este sistema permite que o professor verifique as ideias e opiniões de toda a classe. Isso incentiva os alunos a participarem ativamente durante a palestra, possibilitando uma interação efetiva com o professor.

HOLOS ISSN 1807 - 1600

Embora os estudos supracitados tenham provado sucesso em aumentar as interações professor-aluno, vale ressaltar que esses métodos foram aplicados em classes menores. Portanto, os resultados têm aplicabilidade limitada para palestras maiores.

3.2 Interações durante palestras em larga escala

Clickers são ferramentas conhecidas como CRS já sendo utilizadas em palestras em muitas universidades. O estudo de Caldwell [15] descreveu como os *clickers* foram utilizados para melhorar as interações professor-aluno em palestras de grande escala e concluiu que o uso de um *clicker* teve um impacto positivo na aprendizagem do aluno e na criação de uma atmosfera ativa.

Martyn descreveu o uso de *clickers* em uma configuração baseada em discussão e comparou esses resultados com aqueles utilizados pelos *clickers* em classes de grande escala [16]. *Clickers* foram encontrados para ser úteis na promoção da participação ativa dos alunos em classes de grande escala, mesmo que nenhum efeito mensurável de aprendizagem tenha sido encontrado [17]. Douglas considerou o método benéfico em termos de realização, compreensão e participação nas aulas e interação entre o professor e os alunos, quando comparado às aulas nas quais os dispositivos não foram utilizados [18].

Os pesquisadores também propuseram utilizar um novo dispositivo, "*Thanks teacher Thanks student*" (TT) [9], cujas características permitem que o professor verifique a compreensão dos alunos através de *pop quizzes*. Os alunos respondem "verdadeiro" ou "falso" às perguntas apresentadas pelo sistema. Em uma grande palestra, o equipamento TT foi mostrado para ser altamente valioso para o professor, fornecendo funções básicas para a interação e criação de participação voluntária na palestra. No entanto, os métodos *clicker* e TT exigem o *design* ou a compra de novos dispositivos.

Com a rápida disseminação de *smartphones* e SNS, o uso educacional de aplicações móveis e ferramentas SNS foi recentemente proposto. Judd e Graves descreveram um método em que um SNS foi utilizado para enviar uma solicitação de processamento para um mecanismo de visualização computacional [19]. "*Poll Everywhere*" [20] e "*Socrative*" [21] são aplicações móveis conhecidas e que também foram desenvolvidas para melhorar a interação em sala de aula. A *Poll Everywhere* é um sistema de resposta de público baseado em texto que permite que os instrutores transformem telefones celulares básicos em cliques de sala de aula, e a *Socrative* implementa as funções de um *clicker*, fazendo com que os alunos respondam via *smartphones*, permitindo que o instrutor obtenha um *feedback* dos alunos de forma rápida e eficaz. A desvantagem de "*Socrative*" é que somente um número limitado de estudantes pode ser servido através do programa.

3.3 Sistema de resposta usando *twitter*

Em salas de aula com mais de 80 (oitenta) alunos, a interação eficaz entre os alunos e os seus professores tem se mostrado difícil [9]. Consequentemente, de acordo com [10] e [11], pôde-se observar os seguintes problemas:

- Dificuldade em avaliar o entendimento absorvido pelos estudantes.
- Disponibilidade de pouco tempo para resolver questões dos estudantes.

HOLOS ISSN 1807 - 1600

- Desperdício de tempo com ensino individual para os estudantes dispersos.
- Utilização frequente dos *smartphones* pelos estudantes durante a aula.

Esses problemas têm um impacto negativo tanto em termos de desempenho acadêmico quanto em termos de baixo nível de concentração dos estudantes. Por isso, torna-se difícil para o professor alcançar os objetivos da aula. [8]

De acordo com [8], o sistema de resposta em questão propõe um método que utiliza o serviço de rede social denominado *Twitter* em grandes salas de aula a fim de melhorar o desempenho acadêmico dos alunos e de aumentar os seus níveis de concentração. O método fundamentado em *smartphones* propõe melhorar os testes baseados em papel que, por sua vez, são menos eficazes em relação ao *feedback* da compreensão dos alunos sobre a aula. Ademais, o sistema baseado no *Twitter* ajuda os estudantes a aumentarem os seus níveis de concentração, pois o sistema impede que esses estudantes utilizem os *smartphones* para fins que não estejam relacionados com a aprendizagem durante a aula.

Em comparação com outras aplicações de *software* existentes, o *Twitter* é aceito sem resistência, não depende de qualquer plataforma de *smartphone* particular, suporta um número ilimitado de estudantes durante as aulas e as respostas de cada aluno podem ser classificadas em relação ao tempo. Se os alunos são obrigados a responder as perguntas do *pop quiz* usando seus *smartphones*, então eles têm que estar prontos para responder a qualquer momento e devem manter o foco durante as aulas. Além disso, o método proposto evita o problema dos alunos responderem as questões e, depois, mudarem as suas respostas. A ordem em que as respostas dos alunos são recebidas é a ordem em que podem ser verificadas rapidamente. Dessa forma, os professores podem analisar seu conteúdo e usar os resultados para a próxima aula.

Durante o período deste estudo, os professores postaram uma média de cinco perguntas do *pop quiz* em cada aula teórica. Um exemplo de pergunta do questionário foi "Qual é a função utilizada para inicializar todos os pixels de um LCD em NXT?". Perguntas desse tipo aparecem em momentos inesperados, entre os *slides* de aula projetados em uma tela grande. Os *pop quizzes* cobrem o material essencial das aulas para ajudar os alunos a melhorarem as suas notas nos exames. De todos os estudantes que enviam a resposta correta, apenas a alguns são atribuídos pontos, respeitando a ordem de chegada das respostas. A velocidade de resposta também incentiva os alunos a se concentrarem na aula, em vez de usar os *smartphones* para outros fins. Esses questionários também impedem a cópia de respostas de colegas próximos.

As vantagens e as potenciais desvantagens que estão associadas ao sistema de resposta baseado no *Twitter* podem ser visualizadas nas listagens a seguir:

- Vantagens:
    - Uso reduzido de *smartphones* por parte dos alunos para fins não relacionados com a aprendizagem.
    - Melhora a participação dos alunos durante a aula, visto que eles têm de responder ao *pop quiz*, se quiserem obter boas notas.
    - Melhora a compreensão da palestra e, assim, melhora o desempenho acadêmico por causa dos maiores níveis de concentração estudantil.
    - Entrega de *feedback* para o professor em relação ao entendimento do aluno a respeito da aula.
- Desvantagens:

HOLOS ISSN 1807 - 1600

- A experiência dos alunos em relação ao *Twitter* pode se configurar como sendo um problema.
- Certos alunos que se sentem resistentes ao uso de dispositivos inteligentes não estão em conformidade com o sistema de resposta em questão.
- A análise do *feedback* não é automatizada e deve ser realizada pelo próprio professor, demandando um tempo específico de sua parte para a realização de tal atividade.

4 METODOLOGIA

Neste trabalho, três estudos de caso foram realizados a fim de validar a arquitetura de *software* proposta. O propósito principal da realização desses estudos de caso foi coletar dados que tenham a capacidade de convencer o leitor de que a arquitetura construída realmente funciona. Diante disso, pode-se dizer que a metodologia utilizada neste trabalho possuiu caráter exploratório. A fim de aplicar uma metodologia exploratória, a disciplina-alvo escolhida foi a disciplina de matemática 1 e o assunto ministrado em sala de aula correspondeu às funções exponenciais. O procedimento utilizado nos três estudos de caso consistiu em seis etapas:

1. Ministrar um novo conteúdo para os alunos.
2. Disponibilizar uma questão que aborde o novo assunto ministrado.
3. Resolver (por parte dos alunos) a questão disponibilizada pelo professor.
4. Gerar o desempenho da turma em relação à questão respondida.
5. Analisar o desempenho da turma que foi gerado automaticamente.
6. E decidir se:
    - Continua com o desenvolvimento da disciplina.
    - Realiza uma revisão do novo conteúdo ministrado.

A primeira etapa foi realizada ministrando o novo conteúdo para os alunos através da utilização de um quadro branco, pincéis e apagador.

A segunda, a terceira e a quarta etapas foram realizadas através da utilização de um protótipo implementado com base na arquitetura de *software* proposta.

E, por fim, a quinta e a sexta etapas foram realizadas através de uma reflexão por parte do próprio professor.

A fim de possibilitar a realização do procedimento apresentado, foram utilizados os equipamentos e os *softwares* mostrados nas listagens exibidas a seguir:

- Equipamentos:
    - *Smartphones* (dos próprios alunos).
    - *Notebook* (do próprio professor).
- *Softwares*:
    - Navegador *web*.
    - Protótipo implementado com base na arquitetura de *software* proposta.

Vale ressaltar que o protótipo utilizado foi disponibilizado na *Internet* e, diante disso, a rede sem fio local foi também utilizada, servindo-se de grande valia para a realização dos estudos de caso concretizados com sucesso.

HOLOS
ISSN 1807 - 1600

5 ARQUITETURA DE *SOFTWARE* PROPOSTA

No tópico 2 deste artigo, foi dito que o propósito principal deste trabalho era construir uma arquitetura de *software* que possibilite o desenvolvimento de ferramentas voltadas para a realização de avaliações contínuas do processo de ensino-aprendizagem utilizado em sala de aula pelo professor.

A arquitetura proposta foi desenvolvida com base no estilo arquitetural em camadas, pois tal estilo possibilita dividir um sistema em módulos. Isso significa dizer que os desenvolvedores poderão trabalhar em um mesmo sistema simultaneamente, aumentando a velocidade com que o sistema poderá ser construído. Ademais, as partes do sistema poderão sofrer alterações de maneira independente, sem afetar as demais partes. No caso específico deste trabalho, a arquitetura em questão foi construída com seis camadas muito bem definidas. A figura 5 mostra uma possível disposição dessas seis camadas através de um diagrama de pacotes do *Unified Modeling Language* (UML).

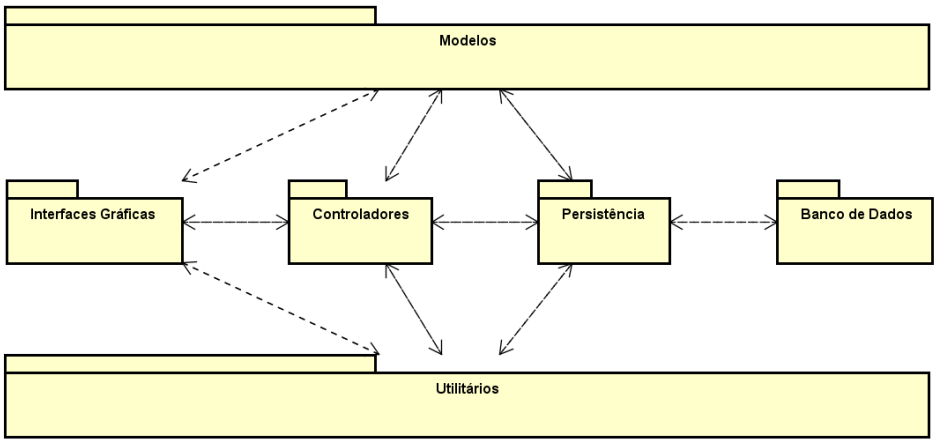

Figura 5: diagrama de pacotes da UML da arquitetura proposta.

De acordo com a figura 5, a arquitetura proposta é constituída pela camada de interfaces gráficas, pela camada de controladores, pela camada de persistência, pela camada de banco de dados, pela camada de modelos e pela camada de utilitários.

A camada de interfaces gráficas é responsável pelo depósito das interfaces visuais através das quais os usuários poderão interagir com o sistema. Essa camada é necessária, pois ela pode armazenar as interfaces gráficas de diferentes módulos. Neste caso, a camada em questão pode fornecer suporte para módulos *desktops*, para módulos *Web*, para módulos de dispositivos móveis, para módulos de realidade virtual e aumentada, entre outros. Esta camada possui uma ligação direta com outras três camadas: a camada de controladores, a camada de modelos e a camada de utilitários. Isso significa dizer que a camada de interfaces gráficas tem a capacidade de trocar informações com essas outras três camadas.

A camada de controladores é responsável pelo armazenamento de arquivos cuja principal função é administrar o fluxo das transações que ocorrem no sistema. Como consequência, essa camada possui a capacidade de tratar questões relacionadas com a segurança. Por exemplo, caso

HOLOS ISSN 1807 - 1600

algum usuário malicioso realize uma injeção SQL através de alguma interface gráfica, esta camada poderá impedir que a injeção SQL seja executada com sucesso. Isso certamente evitará danos sérios no banco de dados e, consequentemente, no sistema. Esta camada possui uma ligação direta com outras quatro camadas: a camada de interfaces gráficas, a camada de persistência, a camada de modelos e a camada de utilitários.

A camada de persistência é responsável pelo depósito de arquivos cuja finalidade maior é estabelecer a conexão com o banco de dados e realizar as devidas consultas e alterações relacionadas com os dados armazenados. Essa camada é essencial, visto que há a possibilidade de realizar interações do sistema com bancos de dados diferentes, caso seja necessário. Dessa forma, o sistema possui flexibilidade suficiente para trabalhar com bancos de dados relacionais, orientados a objetos, entre outros. Esta camada possui uma ligação direta com outras quatro camadas: a camada de controladores, a camada de banco de dados, a camada de modelos e a camada de utilitários.

A camada de banco de dados é responsável pelo armazenamento do banco de dados propriamente dito. Isso significa dizer que os dados relacionados com o sistema em questão estão depositados nesta camada. Vale ressaltar que há a possibilidade de armazenar bancos de dados distintos, conforme seja necessário. No caso deste trabalho, o banco de dados utilizado pertence à família dos bancos de dados relacionais. Esta camada possui uma ligação direta apenas com a camada de persistência. Em outras palavras, esta camada só pode trocar informações com a camada de persistência supramencionada.

A camada de modelos é responsável pelo depósito de arquivos que representam as entidades relacionadas com a proposta do sistema abordado. Por exemplo, há um arquivo que representa as informações de uma entidade denominada professor. Isso significa dizer que os dados relacionados com cada professor são guardados em uma instância correspondente. Isso permite que a aplicação realize as devidas transferências de dados e os devidos processamentos dessas informações de forma encapsulada. Esta camada possui uma ligação direta com outras três camadas: a camada de interfaces gráficas, a camada de controladores e a camada de persistência.

E, por fim, a camada de utilitários é responsável pelo armazenamento de arquivos que possuem a capacidade de realizar processamentos que sejam úteis para a aplicação. Por exemplo, caso seja necessário realizar o cálculo da média de acertos de uma determinada questão aplicada em uma turma específica, essa camada será acionada para que o cálculo desejado seja realizado com sucesso. Esta camada possui uma ligação direta com outras três camadas: a camada de interfaces gráficas, a camada de controladores e a camada de persistência.

6 RESULTADOS OBTIDOS

Após a construção da arquitetura explanada, um protótipo foi desenvolvido a fim de validar a arquitetura em questão e realizar a coleta de dados através de três estudos de caso. Em seguida, o protótipo foi posto em execução em três turmas do 1º ano do curso de informática do campus Lajes do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

HOLOS
ISSN 1807 - 1600

(IFRN). A disciplina-alvo foi a disciplina de matemática 1 e o conteúdo abordado pelo professor correspondente a tal disciplina durante a sua aula dizia respeito às funções exponenciais.

Os dados coletados podem ser visualizados nas figuras 6, 7 e 8. Os dados da figura 6 dizem respeito à turma do 1º ano do curso de informática do turno da manhã. Os dados da figura 7 dizem respeito à turma 1 do 1º ano do curso de informática do turno da tarde. E os dados da figura 8 dizem respeito à turma 2 do 1º ano do curso de informática do turno da tarde.

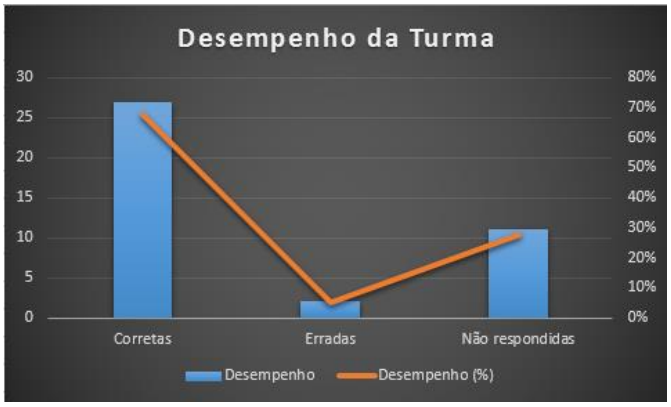

Figura 6: dados coletados da turma do 1º ano de informática do turno da manhã.

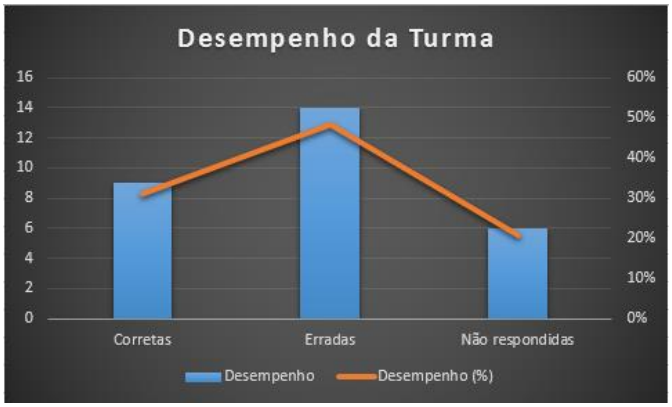

Figura 7: dados coletados da turma 1 do 1º ano de informática do turno da tarde.

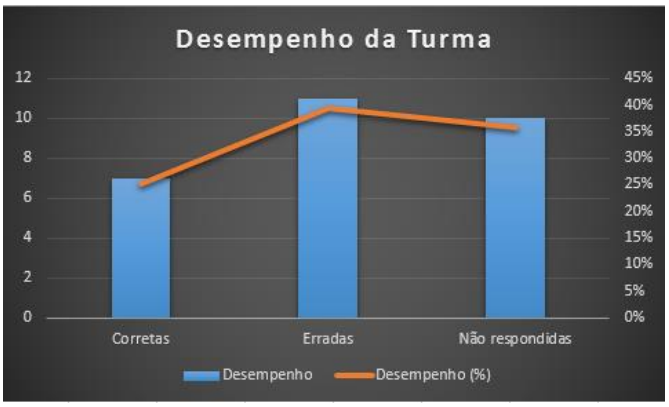

Figura 8: dados coletados da turma 2 do 1º ano de informática do turno da tarde.

Com base nos dados das figuras 6, 7 e 8, o professor terá condições de saber o número de questões corretas, o número de questões erradas e o número de questões não respondidas. A partir desses valores, o professor terá, no mínimo, a capacidade de decidir se continua com o

HOLOS ISSN 1807 - 1600

desenvolvimento de sua disciplina ou se realiza uma revisão do novo conteúdo abordado em sala de aula.

Por exemplo, vamos supor que o professor de matemática 1 definiu que se 60% da turma conseguir acertar as questões disponibilizadas através do protótipo desenvolvido, então ele continuará com o desenvolvimento de sua disciplina. Porém, se menos de 60% da turma não conseguir acertar as questões disponibilizadas, então o professor realizará uma revisão do novo assunto abordado em sala de aula.

Tomando como referência os dados coletados que estão sendo apresentados nas figuras 6, 7 e 8, o professor perceberá que:

- A turma do 1º ano de informática do turno da manhã:
  - Obteve um número de questões corretas igual a 67.5%.
  - Obteve um número de questões erradas igual a 5%.
  - E obteve um número de questões não respondidas igual a 27.5%.
- A turma 1 do 1º ano de informática do turno da tarde:
  - Obteve um número de questões corretas igual a 31.04%.
  - Obteve um número de questões erradas igual a 48.27%.
  - E obteve um número de questões não respondidas igual a 20.69%.
- E a turma 2 do 1º ano de informática do turno da tarde:
  - Obteve um número de questões corretas igual a 25%.
  - Obteve um número de questões erradas igual a 39.29%.
  - E obteve um número de questões não respondidas igual a 35.71%.

Diante desses dados, o professor de matemática 1 observou que a turma do 1º ano de informática do turno da manhã conseguiu ultrapassar a porcentagem mínima de 60% predefinida em relação ao número de questões corretas. Nesse caso, isso significa dizer que o professor poderia dar sequência ao desenvolvimento de sua disciplina. Ainda de acordo com os dados obtidos, o professor de matemática 1 observou também que as duas turmas do 1º ano de informática do turno da tarde não conseguiram atingir a porcentagem mínima de 60% predefinida em relação ao número de questões corretas. Nesse último caso, isso significa dizer que o professor deveria realizar uma revisão sobre o novo conteúdo abordado em sala de aula.

Independente da turma avaliada, sabe-se que há a possibilidade de alguns alunos terem "chutado" as respostas de cada questão da avaliação. No entanto, foi observado pelo professor que a maioria dos alunos das três turmas recorreram aos seus cadernos e tentaram de fato resolverem a avaliação construída e disponibilizada após a ministração do conteúdo exposto. Além disso, é interessante atribuir uma atenção especial para os possíveis significados do número de questões não respondidas, pois esse valor poderá representar o número de alunos que estavam em sala de aula e não responderam de fato a avaliação, assim como poderá representar o número de alunos evadidos.

Por exemplo, vamos supor que o professor realize uma avaliação por aula durante um bimestre inteiro. Após as aplicações das avaliações, o professor poderá observar que o número de questões não respondidas permaneceu praticamente constante. Isso poderá representar o número de alunos evadidos de uma turma em particular. De posse dessa informação, o professor poderá reportar às coordenações de cursos correspondentes que provavelmente o Instituto está

HOLOS ISSN 1807 - 1600

perdendo alunos. A partir disso, a Instituição poderá promover ações específicas a fim de identificar os motivos dessas evasões e buscar alternativas para tratar tal problema.

Ademais, como outro exemplo, o professor terá dados suficientes para realizar uma comparação entre os desempenhos das turmas. Mas, afinal, qual seria um dos valores dessa informação? Nesse caso, o professor poderá identificar, com maior facilidade e agilidade, a turma que tem o melhor desempenho em sua disciplina e, dessa forma, selecionar um grupo seleto de bolsistas para auxiliar no desenvolvimento de algum projeto de pesquisa/extensão de seu interesse.

7 RESULTADOS ESPERADOS *VS* RESULTADOS OBTIDOS

Após a coleta, a análise e a divulgação dos dados obtidos, o professor de matemática 1 não só aceitou que a arquitetura proposta fosse utilizada para a construção de novas ferramentas voltadas para a avaliação contínua de alunos durante o processo de ensino-aprendizagem, como também apresentou grande interesse em gerar uma plataforma de ensino que englobasse interesses dos professores, dos alunos e dos coordenadores de cursos. Em outras palavras, as expectativas iniciais do trabalho proposto foram ultrapassadas, tendo em vista o seu grande potencial voltado para o processo de ensino-aprendizagem.

8 CONCLUSÕES

Após a construção da arquitetura tratada neste trabalho e de acordo com os resultados obtidos através da realização de três estudos de caso no campus Lajes do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN), pode-se observar que os resultados coletados mostraram que a aplicação da arquitetura proposta conseguiu “quebrar” o silêncio obtido dos alunos quando os professores perguntavam se o novo conteúdo abordado em sala de aula havia sido entendido ou não. Esse tipo de *feedback* abre uma vasta gama de possibilidades, tanto do ponto de vista da arquitetura quanto do ponto de vista dos resultados obtidos. Por exemplo, a partir da arquitetura em questão, há a possibilidade de realizar o desenvolvimento de uma plataforma de ensino-aprendizagem que abranja algumas necessidades educacionais específicas tanto do professor quanto do aluno. Um outro exemplo, agora a partir dos resultados obtidos, está na possibilidade do professor não só de decidir se continua ou não com desenvolvimento de sua disciplina, mas também de ser levado a uma autorreflexão sobre a própria metodologia utilizada em sala de aula. Portanto, a arquitetura supramencionada poderá ser o ponto de partida para a construção de trabalhos com grande potencial na área da educação.

9 REFERÊNCIAS

BASS, L., CLEMENTS, P. & KAZMAN, R. (2ª Edição). (2003). *Software Architecture in Practice*. Addison-Wesley.

BOSCH, J. (2000). *Design & Use of Software Architectures*. Addison-Wesley.

BRAUDE, E. (2005). *Projeto de Software: da programação à arquitetura: uma abordagem baseada em Java*. Porto Alegre: Bookman.

BUSCHMANN, F. (1996). *Pattern-Oriented Software*. Wiley.

CALDWELL, J. E. (2007). Clickers in the large classroom: Current research and best-practice tips. *CBE-Life Sci. Educ.*, volume (7), pp. 9–t20. doi: 10.1187/cbe.06-12-0205

COCA, D. M. & SLISKO, J. (2013). Software socrative and smartphones as tools for implementation of basic processes of active physics learning in classroom: An initial feasibility study with prospective teachers. *Eur. J. Phys. Educ.*, volume (4), pp. 17-24.

DOUGLAS, D. (2006). Clickers: A new teaching aid with exceptional promise. *Astron. Educ. Rev.*, volume (5), pp. 70–88. doi: 10.3847/AER2006005

HAKE, R. R. (1998). Interactive-engagement versus traditional methods: A six thousand-student survey of mechanics test data for introductory physics courses. *Amer. J. Phys.*, volume (66), pp. 64–74. doi: 10.1119/1.18809

HWANG, C. I. & LEE, S. H. (2011). An Analysis of Hindrance Factors of Students' Questioning in the University Lecture Class. *J. Educ. Studies*, volume (42), pp. 181–212.

JUDD, B. C. & GRAVES, C. A. (2012). Cellular STEM. *Proc. Int. Symp. Cluster*, *Cloud Grid Comput.*, pp. 799–804. doi: 10.1109/CCGrid.2012.111

KIM, E. K., CHO, J. E. & JUNG, E. C. (2009). The Study on Alternatives for Activating Communication Between Instructor and Students in Large Scale Lecture. *J. Korea Soc. Design Forum*, volume (22), pp. 225–234. doi: 10.21326/ksdt.2009.22.021

KIM, Y., JEONG, S., JI Y., LEE, S., KWON, K. H. & JEON, J. W. (2014). Smartphone Response System Using Twitter to Enable Effective Interaction and Improve Engagement in Large Classrooms. *IEEE Transactions on Education*, volume (58), pp. 98–103. doi: 10.1109/TE.2014.2329651

KOLB, L. (2011). Adventures with cell phones, *Educ. Leadership*, volume (68), pp. 39–43.

LEE, Y. S. (2012). The Influence of Smartphones on the Off-Task Behavior of College Students. *J. Yeolin Educ.*, volume (20), pp. 221–250.

LIU, T., LIANG, J., WANG, H., CHAN, T. & WEI, L. (2003). Embedding EduClick in classroom to enhance interaction. *Proc. ICCE*, pp. 117–125.

MARTYN, M. (2007). Clickers in the classroom: An active learning approach. *EDUCAUSE Quart.*, volume (30), pp. 71–74.

PRESSMAN, R. S. (6ª Edição). (2006). *Engenharia de Software*. São Paulo: McGraw-Hill.

ROZANSKI, N. & WOODS, E. (1ª Edição). (2005). *Software Systems Architecture: working with stakeholders using viewpoints and perspectives*. Addison-Wesley.

RUSH, B. R. et al. (2010). The effect of differing audience response system question types on student attention in the veterinary medical classroom. *J. Veterinary Med. Educ.*, volume (37), pp. 145–153. doi: 10.3138/jvme.37.2.145

SHAW, M. & GARLAN, D. (1996). *Software Architecture*. Prentice-Hall.

SIMONI, M. (2009). Using tablet PCs and interactive software in IC design courses to improve learning. *IEEE Trans. Educ.*, volume (54), pp. 216–221. doi: 10.1109/MSE.2009.5270839