

Implementação de um Sistema Multi-plataforma para Gerenciamento de Atividades Complementares em Cursos Superiores utilizando Banco de Dados *noSQL*

Patrick Gonçalves de Lima ¹, Luiz Henrique Correia ², Carlos Alexandre Gouvea da Silva ³

Resumo

Atividades Complementares (AC) são uma exigência do Ministério da Educação e têm como objetivo colaborar para o crescimento social e profissional dos estudantes de cursos do ensino superior, exceto nos cursos para tecnólogos. A gestão dessas atividades, muitas das vezes, é custosa devido ao fluxo de etapas de registro, catalogação, verificação, contagem e validação das atividades, sendo necessário o uso de ferramentas auxiliares como planilhas eletrônicas ou sistemas computacionais. Desta forma, Instituições de Ensino Superior (IES) elaboram mecanismos e soluções para gerenciar as etapas de gestão das AC, incluindo por exemplo, a entrega física dos comprovantes de estudantes, o preenchimento de formulários manuscritos, realização de entregas digitais por correio eletrônico ou em plataformas de ensino a distância. Os problemas observados nesses processos causam transtornos para estudantes devido a perda ou rasura de comprovantes de AC já entregues, falta de informação das AC homologadas pela IES, demora na confirmação de recebimento ou homologação das AC, necessidade de comparecimento na IES, e ainda acesso não facilitado à lista de atividades aceitas como AC. Um dos grandes desafios na gerencia dessas AC é manter esse processo simples e transparente para as IES e os estudantes, de maneira que o acompanhamento seja realizado em tempo real e totalmente pela Internet, sem desperdício de material com impressões, com visibilidade da totalidade de AC já entregues e homologadas, e o acesso facilitado das regras de AC aceitas nos cursos. O presente artigo apresenta uma solução computacional que mitiga os problemas e solucione os desafios da gestão de AC a partir de um sistema multi-plataforma que pode ser utilizado por qualquer IES utilizando ambiente em nuvem (*Cloud*) e banco de dados não-relacional (*noSQL*). Resultados mostraram a implementação do sistema *Web* e aplicativo móvel, nos quais o controle e inserção das AC no sistema permeiam análises na visão dos coordenadores de curso, estudantes e da instituição.

Palavras-chave: Atividades complementares, Computação na Educação, Javascript, *noSQL*, Sistema *Web*.

¹Centro Universitário UNIFACEAR, Araucária-PR, E-mail: patrickgoncalveslima@gmail.com

²Centro Universitário UNIFACEAR, Araucária-PR, E-mail: luizhenriquecorreia1997@gmail.com

³Centro Universitário UNIFACEAR, Araucária-PR, E-mail: carlos.gouvea@unifacear.edu.br

Abstract

Complementary Activities (CA) are a requirement of the Ministry of Education and aim to collaborate for the social and professional growth of students of higher education courses, except in courses for technologists. The management of these activities is often costly due to the flow of registration, cataloging, verification, counting and validation of activities, requiring the use of auxiliary tools such as spreadsheets or computer systems. In this way, Higher Education Institutions (IES) develop mechanisms and solutions to manage the steps of CA management, including, for example, physical delivery of student certificates, completing handwritten forms, performing digital deliveries by electronic mail or platforms for distance learning. Problems observed in these processes cause issues for students due to loss or erasure of CA certificates already delivered, lack of information about CA homologated by the IES, delay in confirmation of receipt or approval of CAs, need for attendance at IES, and non-access list of activities accepted as CA. One of the great challenges in the management of these CAs is to keep this process simple and transparent for IES and students, so that the monitoring is carried out in real time and totally on the Internet, without wasting material with impressions, with visibility of the total CA already delivered and homologated, and facilitated access of the AC rules accepted in the courses. The present article presents a computational solution that mitigates the problems and solves the challenges of CA management from a multi-platform system that can be used by any IES using a cloud environment and non-relational Data Base (noSQL). Results showed the implementation of this web system and a mobile application, in which the control and insertion of the AC in the system permeate analyzes in the view of course coordinators, students and the institutions.

Keywords: Complementary Activities, Computing in Education, Javascript, noSQL, Web System.

1 Introdução

No Brasil, a transição realizada pelos jovens do ensino médio para o ensino superior, assim com em outras transições, modifica a forma de como o jovem se adaptará as mudanças na forma de como ele irá aprender e se adaptará aos futuros desafios de um curso de graduação (OLIVEIRA; SANTOS; DIAS, 2016). Em uma parcela dessas transições do ensino médio, muito jovens optam pelo ensino técnico que permite uma formação mais rápida, flexiva e mais direcionada às necessidades do mercado de trabalho (SILVA et al., 2016a).

Realizar um curso de graduação proporciona ao estudante grande conhecimento pessoal, mas principalmente experiência para avançar na carreira com a preparação técnica e científica para o mercado de trabalho, onde é possível aplicar de forma prática o conhecimento adquirida, além de trabalhar exercícios essenciais como de cidadania, relacionamento, responsabilidade e comprometimento. A graduação é essencial, mas não exclusiva, para o estudante ter ideia precisa de um ramo em índices teóricos e práticos, demonstrando ao longo dos anos tudo o que é adequado aprender até completar a sua formação (SPARTA; GOMES, 2005).

Os cursos de graduação compõem o ensino superior, onde os mesmos são fiscalizados e regulamentados pelo MEC (Ministério da Educação). Dentre as regulamentações impostas está o estudo de determinados grupos de matérias que a instituição deve obrigar para atingir a integralização do curso por parte dos estudantes. Cada curso possui um regimento diferente e cada Instituição de Ensino Superior (IES) define a sua disponibilidade de ofertas para suprir essas exigências. Dentre essas exigências podem ser citadas as Atividades Complementares (AC).

As AC visam enriquecer e flexibilizar o currículo pleno em cursos de nível superior, incluindo os de nível tecnológico (BRASIL, 2008), propiciando o aproveitamento de conhecimentos adquiridos pelo estudante através de estudos e práticas independentes a

grade curricular, com regras pré-definidas. Segundo o Ministério da Educação, “As Atividades Complementares têm a finalidade de enriquecer o processo de ensino-aprendizagem, privilegiando a complementação da formação social e profissional” (MEC, 2018). As exigências da quantidade de horas obrigatórias e optativas, assim como sua categorização do que se deve realizar pode variar de curso para curso, ou até mesmo de instituição para instituição.

Para o estudante o processo de validação dessas atividades se torna confuso e moroso, visto que o controle é feito unilateralmente pelas IES, pois as entregas são aleatórias, periódicas ou pontuais durante o ano. E mesmo a entrega de atividades sendo uma obrigatoriedade para os estudantes, as AC geram uma carga administrativa intensa às IES para o cumprimento desse quesito. O processo se torna muito custoso, tanto de esforço quanto de tempo devido a quantidade de alunos e comprovantes a serem analisados. Assim, o uso de tecnologias de informação se torna de grande auxílio na proposta de soluções práticas para resolver ou mitigar os impactos negativos dessa problemática.

O uso de sistemas computacionais são apresentados como alternativas para propor soluções aos desafios existentes. A utilização de sistemas adaptados como o Moodle proposto por (SILVA; VALLIM-FILHO, 2012)(SILVA; VALLIM-FILHO, 2013) não permite a contabilização das AC realizadas pelos estudantes. Já em (SILVA; CONCEIÇÃO; SILVA, 2016) e também (RIZZI et al., 2017) sistemas de gestão acadêmica de AC são desenvolvidos para ambiente *web* em servidor exclusivamente local o que não permite o acesso através da Internet de outros locais. Esses dois sistemas também se mostram ineficientes devido ao fato de não permitirem o envio de arquivos anexos, como por exemplos cópias digitais de comprovantes. Outra dificuldade é a ausência de sistemas específicos para aplicativos móveis, sendo possível apenas o uso de aplicações *web* sem responsividade, o que consequentemente diminui a qualidade de usabilidade dos usuá-

rios. Estes sistemas ainda possuem limitações quanto a geração de certificados de AC realizadas ou sugestão de AC ao estudantes por parte de professores ou coordenadores de cursos. Em (PAULA, 2014) é proposto um sistema *web* que permite a emissão de certificados online, porém o sistema não foi adaptado para dispositivos móveis e não realiza a contagem total das horas de AC realizadas. Um sistema com responsividade em ambiente móvel chamado *Cronos* é apresentado por (VANUCHI et al., 2016) no qual o estudante pode visualizar relatório das AC por grupos de atividades, verificar os comprovantes validados e não validados, identificar a totalidade de AC realizadas e gerar certificados de AC realizadas. Embora o *Cronos* apresente funcionalidades importantes não permite a sugestão de eventos aos estudantes e o banco de dados ainda é relacional.

Pensando nessas dificuldades e desafios na gestão de AC o objetivo desse trabalho é apresentar a proposta de um sistema multiplataforma (*web online* e aplicativo móvel), chamado de *Akademic*, para registro de AC pelos estudantes, adaptável a diversas IES e cursos superiores. Nossa principal contribuição é um sistema disponível para ser acessado com responsividade via *web* nos principais navegadores de Internet ou também através de um módulo para aplicativo móvel sem a necessidade de instalar no dispositivo do estudante. Disponível em ambiente *cloud* o *Akademic* utiliza um banco de dados não-relacional (noSQL), o que mostra a preocupação em utilizar tecnologias atuais e em rápido aumento de aplicações disponíveis com essas plataformas. O *Akademic* permite o acesso dos estudantes através de redes sociais, a sugestão de eventos, o cadastro de regras de AC pelos coordenadores, a contagem das AC em tempo real após a validação da AC e o envio de arquivos anexos.

Além dessa seção introdutória, a seção 2 apresenta uma descrição e objetivos sobre as AC, além da apresentação de trabalhos relacionados disponíveis na literatura. Na seção 3 é apresentado a metodologia e méto-

dos utilizados para desenvolvimento do sistema proposto. Os resultados da implementação e apontamentos da vantagens do sistema são mostrados na seção 4. E por fim, na seção 5 são apresentadas algumas conclusões e apontamentos de trabalhos futuros.

2 Atividades complementares

A complementação curricular se torna necessária em qualquer nível de ensino uma vez que certas lacunas são encontradas para a boa formação dos estudantes, transcendendo questões ligadas a temas que eventualmente não são abordados nos cursos, necessitando um propósito mais amplo, que envolve experiências que promovam uma vivência democrática e comportamental. Como abordado por (BEANE, 2003), “o mundo da escola começa e termina com as disciplinas do conhecimento e a elevada cultura acadêmica que as produz, as mantém e as sanciona”. Tal cenário sugere atenção para abordagem curricular por disciplinas, no sentido de manter os conteúdos ofertados na graduação em constante atualização. O que se pretende discutir é como suprir eventuais lacunas e oferecer aos estudantes uma formação atualizada com as novidades do mercado de trabalho utilizando o apoio de Atividades Complementares.

Atividades complementares são consideradas parte integrante do currículo, obrigatórias para estudantes de graduação, e tem como principal objetivo enriquecer o processo de ensino-aprendizagem, incentivando a participação do acadêmico em atividades que privilegiem a construção de comportamentos sociais, humanos, culturais e profissionais. As atividades extra-curriculares desenvolvidas em ambientes acadêmicos permite uma aproximação ainda maior entre os estudantes e os professores, oportunizando a exploração de aspectos não abordados na grade curricular dos cursos superiores e o conhecimento de novas realidades (OLIVEIRA; SANTOS; DIAS, 2016).

Conforme as Diretrizes Curriculares Nacionais (DCN), as “AC são componentes curriculares que possibilitam o reconhecimento

de habilidades, conhecimentos, competências e atitudes do aluno, inclusive adquiridos fora do ambiente acadêmico”. Com o intuito de ampliar a formação dos alunos de cursos superiores, o MEC através das suas atribuições, exige que estudantes de cursos superiores realizem até 20% da carga horária de seus cursos em Atividades Complementares (SILVA; VALLIM-FILHO, 2013; SILVA; LOPES; VALLIM-FILHO, 2014).

Cada instituição de ensino e curso tem sua própria distribuição de horas e método de aplicar as atividades, sendo que todas devem atender normativas específicas como: o Parecer nº 67 do CNE/CES, que estabelece um referencial para as Diretrizes Curriculares Nacionais dos Cursos de Graduação, e a Resolução CNE/CES nº 2/2007, que dispõe sobre a carga horária mínima e os procedimentos relativos à integralização e duração dos cursos de graduação, bacharelados, na modalidade presencial (MEC, 2018). O colegiado de cursos é responsável por disponibilizar quais atividades serão aceitas, a quantidade de horas que serão validadas e o período que poderão ser entregues. As atividades são classificadas em categorias e para cada categoria há um limite máximo de horas que podem ser computadas, não permitindo ao aluno realizar uma única categoria de atividade.

Dentre as atividades que podem ser aceitas como atividades complementares, podem ser incluídos: projetos de pesquisa, monitoria, iniciação científica, projetos de extensão, módulos temáticos, cursos, seminários, simpósios, congressos, conferências, publicação de trabalhos em periódicos e revistas, disciplinas oferecidas por outras instituições de ensino, cursos de formação complementar, atividades culturais como teatros, cinema, apresentações, entre outros.

2.1 Trabalhos de implementação relacionados

O avanço tecnológico tem propiciado ao longo dos anos, um fator predominante e decisivo para melhoria de procedimentos e processos inerentes a educação e instituições de en-

sino, em especial em métodos de ensino-aprendizagem através do uso de tecnologias como computadores, *tablets* e *smartphones* (SOUZA et al., 2017). Esse avanço permite que novas ferramentas e *softwares* educativos sejam desenvolvidos não apenas para criar ambientes propícios à aprendizagem (LIMA et al., 2016), mas também como elementos que otimizem e melhorem procedimentos administrativos e institucionais.

O uso de sistemas informatizados na gestão de AC vem sendo utilizado e estudado em diferentes cenários de aplicações. Em (SILVA; VALLIM-FILHO, 2012)(SILVA; VALLIM-FILHO, 2013) (T1) são apresentados resultados do uso do sistema Moodle na gestão e análise das AC dos alunos dos cursos de Sistemas de Informação e Ciência da Computação em uma instituição privada. Resultados mostraram que gestão computacional trouxe benefícios como a não necessidade de impressão de documentos, o armazenamento das informações em bancos de dados, interação *online* entre os alunos e a coordenação.

Em (SILVA; CONCEIÇÃO; SILVA, 2016) (T2) é descrito o desenvolvimento de um sistema de controle das AC com interatividade e de boa usabilidade. Os resultados mostraram o protótipo de um sistema *web* que permitia um aluno enviar uma solicitação de validação de AC ao coordenador de curso. Em (PAULA, 2014) (T3) é proposto um sistema baseado em *Framework JavaServer Faces* (JSF) em banco de dados relacional que permite gerenciar o controle de AC de alunos. Uma das vantagens é que o aluno pode registrar as horas realizadas, eventos promovidos pela instituição e gerar certificados de atividades. Atividades realizadas em outras instituições também podem ser registradas no sistema. A eficiência do sistema foi medida a partir da avaliação subjetiva dos próprios alunos.

Um estudo e um produto de *software* chamado SisAti (Sistema de Atividades Complementares) sobre o controle das AC em um curso de tecnologia da Universidade Federal do Mato Grosso do Sul é proposto por (COSTA, 2016) (T4). Assim como outros,

esse trabalho tem o intuito de propor e implantar um sistema informatizado para a realização desse controle. Em (RIZZI et al., 2017) (T5) é proposto um sistema *web* desenvolvido em banco relacional SQL (*Structured Query Language*), HTML5 (*Hypertext Markup Language*) e PHP (*Hypertext Preprocessor*) para gestão de AC. O ambiente foi desenvolvido com características para uma instituição de ensino federal pública específica e está limitada a execução em rede local, sendo necessário a implantação na Internet. Os autores tem como objetivo que os usuários tenham acesso às informações de AC de forma simples, dinâmica, segura, confiável e interativa, mas não apresenta detalhamento da percepção da qualidade vista por outros usuários em potencial.

A Tabela 1 apresenta uma comparação entre as principais características dos trabalhos relacionados (T1, T2, T3 e T5) e o Akademic, proposto neste artigo (P*). Devido a falta de informações do trabalho T4, não foi possível incluí-lo na tabela para comparação.

Tabela 1: Comparação entre os trabalhos relacionados e o proposto neste artigo.

Característica	T1	T2	T3	T5	P*
Web online	✓				✓
Web local		✓		✓	
Aplicativo móvel					✓
Necessita Internet	✓		✓		✓
Cloud Computing					✓
Banco noSQL					✓
Ambiente responsivo					✓
Envio de AC	✓	✓	✓	✓	✓
Envio Anexos	✓		✓		✓
Contagem das horas		✓		✓	✓
Cadastro regras		✓	✓		✓
Sugestão de Eventos			✓		✓
Qualquer IES					✓
Acesso por Rede Social					✓
Gera certificados			✓		

Observa-se na tabela anterior que o Akademic apresenta melhorias ao comparar com os demais projetos como: tratando-se de um sistema desenvolvido na plataforma Firebase o mesmo já se encontra disponível na Internet para consulta, sendo o seu endereço *web* disponível apenas aos desenvolvedores por enquanto, e utiliza o conceito de *cloud computing*. O uso de banco de dados não-relacional

(*noSQL*) a partir dos conceitos de coleção de dados tem ganhado grande visibilidade entre os desenvolvedores, permitindo que sistemas possuam melhor desempenho e tempo de resposta no acesso as informações (HAN et al., 2011). A possibilidade de qualquer IES utilizar esse sistema parte do princípio que a proposta desse projeto pode ser oportunizada como um produto de negócio rentável futuramente.

3 Desenvolvimento da aplicação

O Akademic foi desenvolvido para auxiliar que estudantes de cursos de graduação possam realizar o acompanhamento de suas AC a partir da gestão de um coordenador de curso ou pela própria IES. Assim, uma das principais atividades antes do início do desenvolvimento foi realizar a coleta e identificação das necessidades dos usuários do sistema Akademic, o que gerou a seguinte lista de requisitos funcionais (RF) e não-funcionais (RNF) do sistema.

- **RF01** – Gerenciar Tipos de AC (regras). O sistema deve permitir ao controlador gerenciar o que é ou não permitido, assim como as definições de carga horária permitida para cada AC.
- **RF02** – Validar Atividade Complementar. O sistema deve permitir a validação de uma AC (aprovar ou reprovar) inserida por um estudante.
- **RF03** – Gerenciar Eventos. O sistema deve permitir a inclusão de sugestões de eventos para possíveis AC, sendo essa possibilidade para qualquer tipo de usuário.
- **RF04** – Gerenciar Atividade Complementar. O sistema deve permitir a inclusão, edição, exclusão e a busca de tipo de AC.
- **RF05** – Gerenciar Perfil. O sistema deve permitir a inclusão, edição, a exclusão e a busca de usuários, sendo o mesmo gerenciado pelo Administrador e com permissões de edição baixas para o usuário.

- **RF06** – Gerenciar Cursos. Baseado na lista atualizada fornecida pelo MEC, alimentar o sistema com os cursos que serão associados a instituições, estudantes e controladores.
 - **RF07** – Gerenciar Controladores. O sistema deve permitir a inclusão, edição, a exclusão e a busca de controladores, que serão os usuários com permissão para criar eventos, validar atividades e criar as regras de horas do curso.
 - **RF08** – Gerenciar Instituições. O sistema deve permitir a inclusão, edição, a exclusão e a busca das instituições baseada no cadastro atualizado do MEC.
 - **RNF01** – Responsividade. Feito para qualquer dispositivo: *desktop*, *tablet* ou *mobile*.
 - **RNF02** – *App-like*. Através da tecnologia SPA (*Single Page Application*) disponibilizada pelo *framework Angular*, permitir o usuário ter a experiência de um *App* nativo.
 - **RNF03** – Engajamento. Através de *push-notifications* e a funcionalidade *Firebase Cloud Messaging (FCM)* o usuário ser constantemente notificado e engajado ao sistema
 - **RNF04** – Fluidez. Através de técnicas conhecidas de otimização: performar as imagens para *SVG (Scalable Vector Graphics)* o qual renderiza via memória no navegador, minificar *CSS* e *Javascript*, diminuindo o tamanho de requisições, etc.
 - **RNF05** – Segurança. Usando *tokens JWT (JSON Web Tokens)* para cada requisição *HTTP* o sistema não permitirá acessos à não ser que o usuário esteja autenticado e adequado ao seu perfil.
 - **RNF06** – Visual uniforme. Dentro de conceitos heurísticos e de *design*, criar uma aplicação consistente e uniforme visualmente para atendimento requisitos de usabilidade (SILVA et al., 2016).
- Originalmente, estes requisitos foram extraídos a partir da observação das dificuldades encontradas no processo de gestão de AC na IES onde foi realizado o desenvolvimento desse sistema, e também no *feedback* a partir da percepção de estudantes dos últimos períodos do curso de Sistemas de Informação e coordenadores de cursos. Entretanto, sendo as AC regimentado pelo MEC, as exigências de realização e validação seguem padrões similares na grande maioria das IES com cursos superiores. Isso pode permitir que a maioria dos requisitos podem ser similares também às demais IES.
- Os quatro perfis de usuários possuem particularidades específicas em relação aos níveis de acesso e permissão do sistema. Essas características são apresentadas a seguir:
- **Administrador:** Possui maior nível de privilégio no sistema e possui como principal ação o gerenciamento das IES que utilizam o sistema.
 - **IES:** Pode realizar o cadastramento dos controladores que são associados aos cursos que o mesmo irá gerenciar dentro da IES. A IES pode ainda registrar eventos de cunho coletivo que seja de interesse a todos na IES gerenciada.
 - **Controlador:** Possui privilégios para gerenciamento do próprio perfil de seus dados acadêmicos e do curso ao qual ficará responsável. O controlador, pode sugerir e gerenciar os eventos que os alunos do curso podem realizar. A principal ação desse usuário será a validação das AC enviadas pelos estudantes, aprovando ou não, e criar as regras de categorias e quantidade de atividades aceitas dentro do curso.
 - **Estudante:** Pode realizar a gerência do próprio perfil acadêmico cadastrado, associando assim o seu cadastro a um curso

específico. Esse usuário também pode sugerir eventos dentro do curso. A principal ação do estudante é realizar a gerencia das próprias AC, submetendo-as para validação do controlador.

O sistema proposto possui como principais usuários o administrador do sistema (com privilégios máximos de acesso), a IES, o controlador (geralmente o coordenador de curso) e o estudante. A interação desses usuários com o sistema é mostrada no diagrama de caso de uso da Figura 1.

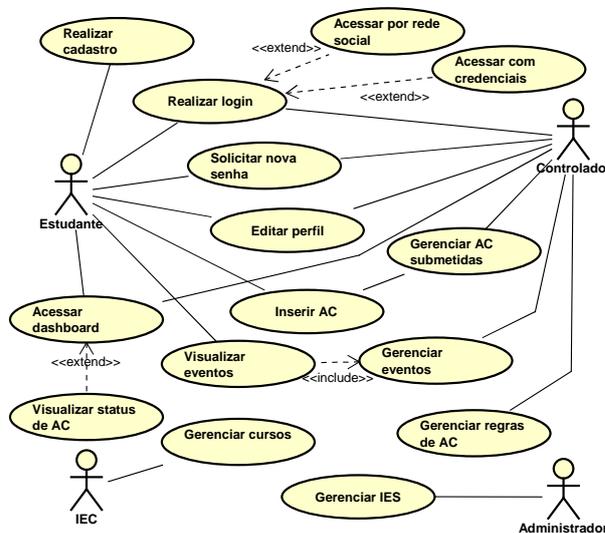


Figura 1: Diagrama de Casos de Uso.

3.1 Arquitetura do Sistema

Aplicativos móveis geralmente precisam de serviços de *back-end* para trabalhar com a interface do usuário do *front-end*. Isso significa que deve haver códigos e servidores de *back-end* para trabalhar com aplicativos móveis (CHENG; CHENG, 2018). Para facilitar a construção, definição e manipulação dos dados foi utilizado o *Cloud Firestore*, disponível no Firebase como mostrado na Figura 2.

O Firebase (GOOGLE, 2019) é um serviço em nuvem para impulsionar os *back-ends* dos aplicativos. O Firebase pode fornecer suporte para armazenamento de dados e autenticação dos usuários. Depois de integrar os aplicativos para dispositivos móveis ao Firebase, não

é preciso escrever código de *back-end* ou gerenciar a infraestrutura. Os desenvolvedores *front-end* podem usar o código JavaScript para interagir com o Firebase (CHENG; CHENG, 2018).

O serviço Firebase consiste de ferramentas para o projeto, como: o *Authentication* que permite a autenticação e gerenciamento de usuários a partir de uma variedade de provedores sem código do lado do servidor, de maneira a facilitar a implementação de *login* centralizado para o padrão comum (usuário e senha) e também o acesso com provedores diretos de redes sociais (*Google+*, *Twitter* e *Facebook*). O Firebase ainda fornece o *Cloud Storage* com um serviço de armazenamento de objetos simples e econômico que visa facilitar a centralização de documentos anexáveis, sendo este útil no processo de inclusão de AC onde o usuário pode anexar comprovantes. E por fim, viabiliza o *Hosting* a partir de uma hospedagem de conteúdo *web* em nível de produção, onde é possível a implantação do *software* para se tornar acessível via Internet.

As *Cloud Functions* executam o código *back-end* automaticamente em resposta a eventos acionados pelos recursos do *Firebase Cloud Messaging* pelas solicitações HTTP (*Hyper Text Transfer Protocol*) ou através de mensagens XMPP (*eXtensible Messaging and Presence Protocol*). Neste trabalho utilizamos apenas chamadas HTTP. O código é armazenado na nuvem e é executado em um ambiente gerenciado, onde não é necessário gerenciar e dimensionar servidores próprios. Em outras palavras, entra no conceito de *serverless*, onde uma aplicação incorpora de forma significativa aplicativos e serviços de terceiros hospedados na nuvem, para gerenciar a lógica e o estado do lado do servidor, como o Firebase. As arquiteturas sem servidor (*serverless*) são projetos de aplicativos que incorporam serviços “*Back-end as a Service*” (BaaS) de terceiros e/ou que incluem execução de código personalizado em contêineres efêmeros gerenciados em uma plataforma “*Functions as a Service*” (FaaS) (SEWAK; SINGH, 2018; AGARWAL, 2018).

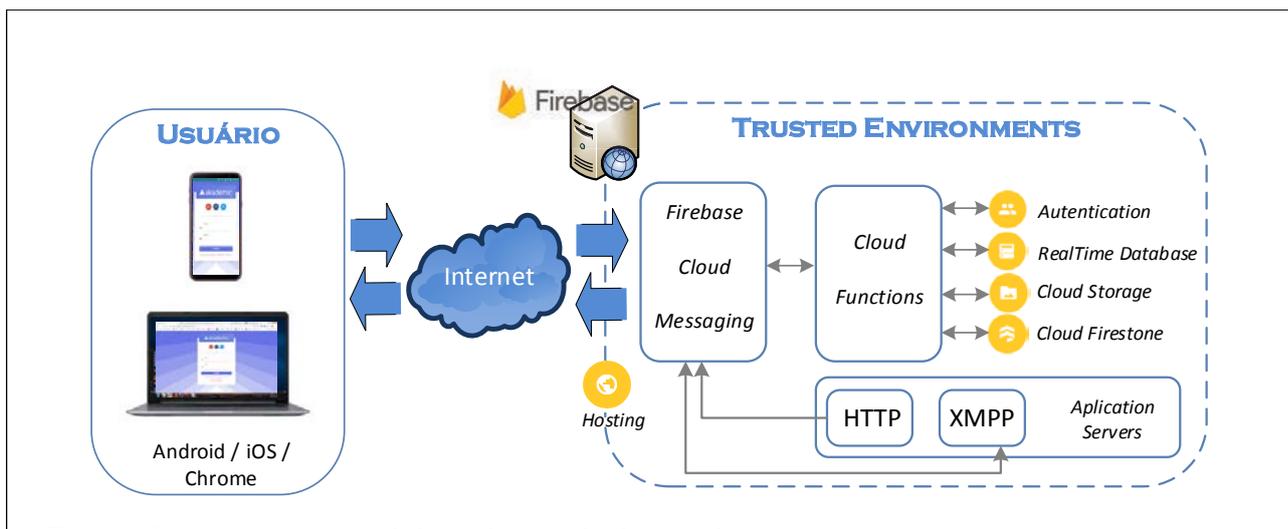


Figura 2: Arquitetura geral do ambiente de desenvolvimento. Adaptado de (ALBERTENGO et al., 2019)

O sistema proposto está implementado utilizando o *Framework Angular* que é caracteristicamente baseado em Javascript e serve para preparar sua tradução do código para ambiente da *web*. O Javascript é uma linguagem para programação *client-side* em navegadores *web* e foi concebida para ser uma linguagem *script* com orientação a objetos (OO, *object oriented*) baseada em protótipos, tipagem fraca e dinâmica, e funções de primeira classe (AGARWAL, 2018). O *Angular* foi utilizado, pois oferece um padrão simples para a construção das interfaces de usuário para aplicações *web* de alto desempenho. *Angular* utiliza tecnologia SPA (*Single Page Application*) que é uma aplicação que consiste de uma única página *web* com o objetivo de fornecer uma experiência do usuário similar a de um aplicativo *desktop*. O *Angular* é enriquecido com HTML, CSS (*Cascading Style Sheets*) e Javascript, linguagens padrões e universais da *web*.

3.2 Modelo de dados

Pela flexibilidade e alto desempenho, as informações registradas no sistema são armazenadas em um banco de dados não-relacional em nuvem que armazena todos os dados para o fácil acesso. Descaracterizando-se de um banco de dados relacional, onde os dados são armazenados em tabelas separadas e com relacionamentos definidos por restrições de chave,

ou seja uma coleção de dados relacionados logicamente, em um banco de dados não-relacional, um registro é armazenado em modelos de dados noSQL. Os principais modelos de dados noSQL destacam-se em (LÓSCIO; OLIVEIRA; PONTES, 2011):

- Chave-valor (*key-value*): método simples que permite visualizar o banco de dados como uma grande tabela *hash*, no qual um conjunto de chaves é associada a um único valor (*string* ou binário);
- Orientado a colunas: mais complexo que o método chave-valor, no qual os dados são indexados por linha, coluna e *timestamp* (tripla), no qual as linhas e colunas são identificadas por chaves e o *timestamp* diferencia as múltiplas versões de um mesmo registro;
- Orientado a documentos: método que armazena coleções de documentos, no qual cada documento é um objeto com um identificador único e um conjunto de campos;
- Orientado a grafos: método baseado em três componentes, no qual os “nós” são os vértices do grafo, os “relacionamentos” são as arestas e as “propriedades” são os atributos dos mesmos nós e relacionamentos.

Desta forma, foi utilizado para implementação do banco de dados não-relacional o modelo orientado à documentos disponível no Firebase (*RealTime Database*), no qual esses documentos estão no formato JSON (*JavaScript Object Notation*) onde os registros são armazenados. JSON é um formato de texto para a serialização de dados estruturados, no qual pode representar quatro tipos de dados (*strings*, números, booleanos e nulos) e dois tipos de estrutura (objetos e matrizes) (BRAY, 2017). Para cada conjunto de dados (como por exemplo de um estudante) seus atributos (nome, curso, idade, código matrícula, outros) e dados relacionados (AC realizadas, total de AC, outros) são armazenados em um único documento.

Neste modelo orientado a documento, os dados são otimizados para desenvolvimento intuitivo e com escalabilidade horizontal, pois os documentos são representações diretas do item ao que está se manipulando dentro do banco de dados não-relacional.

Inicialmente uma solução com banco de dados relacional foi cogitada, no qual outros bancos de dados poderiam facilmente suprir as necessidades de implementação desse sistema. Contudo, pensando em uma solução de gestão de AC que pode ser utilizada por diversas IES, a longo prazo pode apresentar desafios devido a quantidade de dados armazenados e também a concorrência de múltiplos acessos no sistema. Desta forma, o sistema pode representar desvantagens em relação ao tempo de acesso à dados e alta latência na busca de dados quanto utilizado um banco de dados relacional. Neste caso, banco de dados noSQL representam uma alternativa interessante no desenvolvimento do nosso sistema.

Como uma representação gráfica de como está estruturado o banco de dados e seus relacionamentos, a seguir indicamos o diagrama de modelo de dados. Salientando que a diagramação de dados não está seguindo os princípios do UML (*Unified Modeling Language*), isso porque ainda não há oficialmente uma representação para banco de dados *noSQL*. As pastas são consideradas *collections* que possuem seus *documents*, conforme organização

estruturada pelo *Cloud Firestore* e exemplificado na Figura 3.

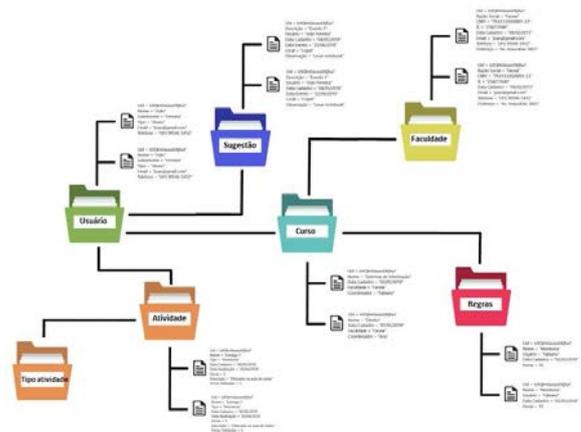


Figura 3: Diagrama de entidade de dados não-relacional.

A Figura 4 exemplifica relação entre coleções de dados, onde a *collection* Atividades possui um *document* que identifica todas as propriedades AC incluída em formato JSON. Se alguma requisição for solicitada o valor por completo será recuperado, sem a necessidade de “*joins*”, tornando a consulta mais rápida e precisa.

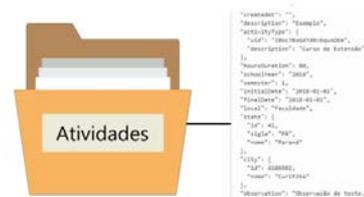


Figura 4: Diagrama ampliado de coleção e filho (JSON).

3.3 Desenvolvimento ágil

Os passos metodológicos utilizados para o desenvolvimento dessa aplicação foram baseados no manifesto ágil (BECK et al., 2001) e fundamentado no ciclo de vida de desenvolvimento de sistemas incremental com objetivo de maximizar as funcionalidades mais importantes. Neste sentido utilizamos como base de desenvolvimento a metodologia ágil Scrum que é

amplamente utilizada por equipes de desenvolvimento de *software* com objetivo de maximizar as tarefas a serem realizadas e a produtividade da equipe, sempre com foco em qualidade dos produtos de sistemas (SILVA et al., 2016b).

Ao longo de dez meses (março a novembro de 2018) foram realizados onze *Sprints* de desenvolvimento. Um *Sprint* dentro da metodologia Scrum representa um período onde os desenvolvedores implementam partes cruciais do sistema definidas entre todos os envolvidos e que representam alto valor de negócio. Geralmente os *Sprints* acontecem de duas a quatro semanas (SILVA et al., 2016b). No desenvolvimento do Akademic foi utilizado *Sprints* de 18 dias em média, considerando sábados e domingos. Ao final de cada *Sprint* foi avaliado entre os envolvidos o que foi desenvolvido, reavaliado as possíveis modificações no projeto e definido o que seria desenvolvido no próximo *Sprint*. Esse processo se repetiu até a entrega final do sistema proposto. A visão do desempenho de desenvolvimento é apresentado em um *Burndown Chart* na Figura 5.

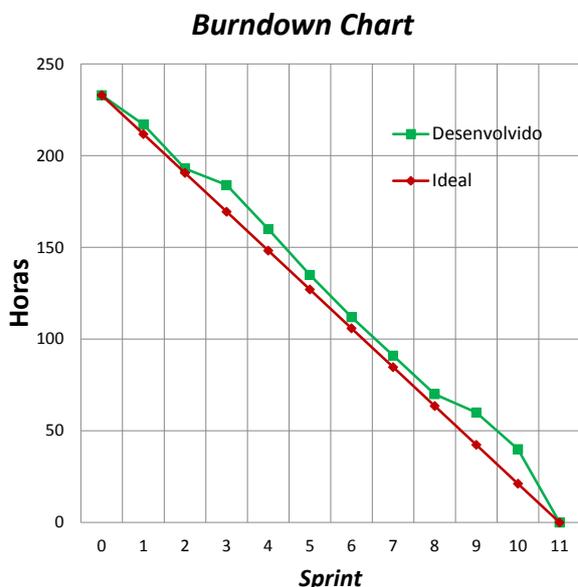


Figura 5: *Burndown Chart* de desenvolvimento.

Ficou evidenciado que durante o desenvolvimento do projeto nem todas as atividades planejadas foram finalizadas dentro dos

Sprints específicos 1 à 8, isso ocorreu devido a necessidade de ambientação dos desenvolvedores com algumas tecnologias. A incorreta previsão da necessidade de horas para execução dos *Sprints*, falta de tempo e também por problemas de especificação de requisitos foram dificuldades encontradas durante todo o desenvolvimento. O *Sprint* 11 foi mais longo com 23 dias e possuía mais atividades, pois este *Sprint* final foi utilizado para corrigir as falhas e erros no sistema encontrados. O uso do Scrum permitiu a priorização de atividades de maior valor como produto, foco nas entregas contínuas ao final de cada *Sprint* e em tarefas essenciais para obtenção do resultado final planejado.

4 Resultados

Em seguida é apresentado as principais telas e visões do sistema Akademic implementado.

4.1 Sistema Akademic

Inicialmente é apresentado na Figura 6 a tela inicial da aplicação. Como o Akademic foi desenvolvido utilizando a plataforma *Angular* de aplicação *web* de código fonte aberto, permitiu que o mesmo sistema pudesse ser acessado em diferente plataformas sem a necessidade de adaptação do código fonte. Assim, a tela responsiva pode se adaptar tanto a ambientes de *desktop* como também ser utilizado no próprio *smartphone* do usuário como um aplicativo nativo mas sem a necessidade de ser instalado, apenas criando um atalho da aplicação.

A tela de acesso tem o objetivo de efetuar e validar se um usuário possui acesso ao sistema, assim como fornecer provedores de redes sociais para o acesso ao sistema. Nesta mesma tela inicial, é possível ser redirecionado ao cadastro de um novo usuário assim como a possibilidade de reiniciar a senha de um usuário cadastrado.

4.1.1 Visão estudante

Ao acessar a plataforma, o perfil do usuário é identificado e o sistema disponibiliza as seções

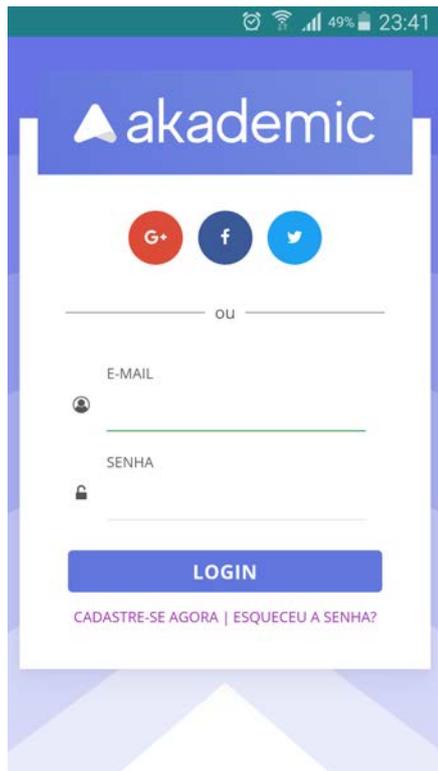


Figura 6: Tela inicial da aplicação como aplicativo nativo no *smartphone*.

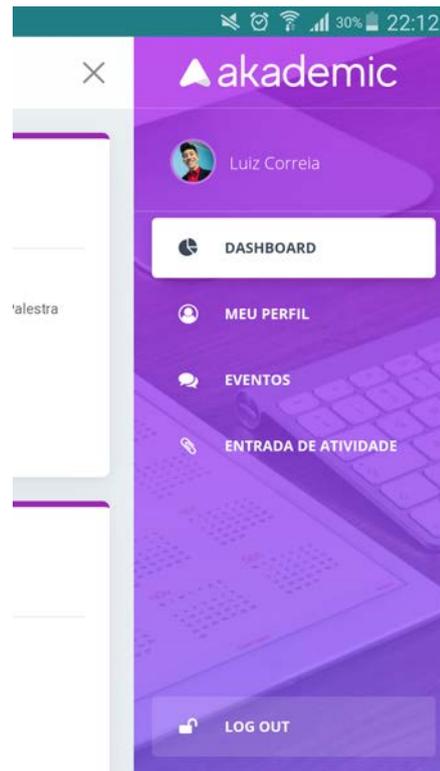


Figura 7: Visão inicial usuário estudante.

permitidas a este perfil. Na Figura 7 é mostrado a visão inicial para um usuário do tipo estudante. Como já informado anteriormente, esse usuário pode em “Meu Perfil” editar seus dados pessoais acadêmicos, em “Eventos” inserir eventos e em “Entrada de Atividade” realizar o envio de AC para validação.

A opção *Dashboard* está relacionada ao estado atual das atividades já enviadas para validação do controlador. A Figura 8 apresenta o *Dashboard* do estudante onde é apresentado as atividades enviadas para validação em relação aos grupos de atividades no curso do estudante. A mesma figura ainda apresenta, dentre as atividades enviadas para validação qual a relação das que estão pendentes ou já validadas. Ainda nesta figura, abaixo dos gráficos, é possível verificar a lista de atividades pendentes de validação que ficam a disposição do estudante para exclusão ou edição. Por regra de negócio, após a validação da atividade pelo controlador, a atividade fica bloqueada de qualquer ação do estudante.

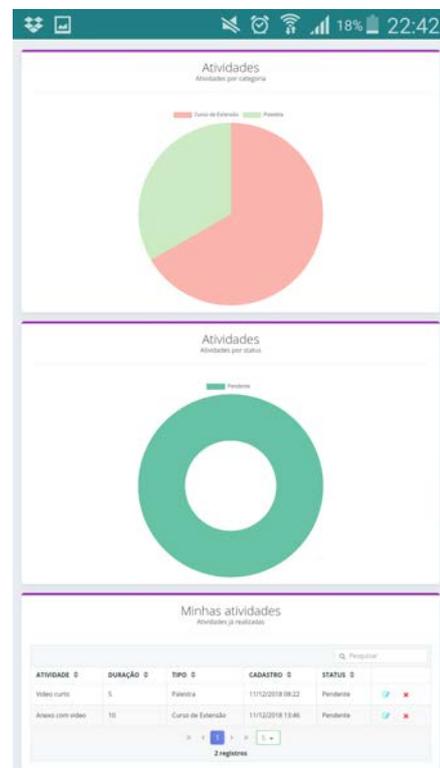


Figura 8: *Dashboard* do estudante referente às AC enviadas e validadas.

Para o envio de uma atividade complementar o estudante deve acessar a opção “Entrada de Atividade”, como mostrado na Figura 9 mostrado usando a versão responsiva para um *browser web* para *smartphone*. Nesta opção o estudante deve anexar um comprovante que confirme a atividade realizada, podendo ser incluído desde imagens, vídeos curtos ou arquivos texto como PDF. Em seguida o estudante deve informar “Descrição” da atividade realizada, “Tipo de atividade” dentre aqueles grupos cadastrados pelo controlador, “Carga horária”, “Ano letivo” e “Semestre” quando o estudante realizou a atividade, “Datas” inicial e final, “Local” da realização, “Estado” e “Cidade”. No campo “Observações” o estudante pode incluir informações importantes da atividade que permite auxiliar na validação da AC.



Figura 9: Página em *browser web* para submissão de AC pelo estudante.

Após enviado a atividade para validação, a mesma entra no estado de “Pendente” até que o controlador valide a atividade enviada. A Figura 10 apresenta o estado de uma atividade enviada pelo estudante mas não validada

ainda pelo controlador. Neste estado a atividade pode ser modificada e excluída pelo aluno sem a necessidade de solicitação de permissão.

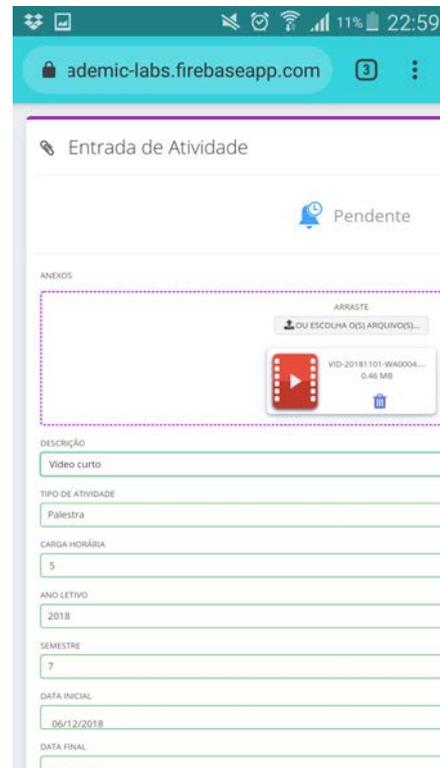


Figura 10: Tela com estado atual de atividade “Pendente” de validação.

Na opção “Eventos” um ambiente colaborativo permite que o estudante compartilhe atividades que podem ser realizadas por todos os alunos do curso ao qual ele está associado. Este cenário é mostrado na Figura 11, e mostra um ambiente colaborativo amplamente discutido e desenvolvido em outras aplicações que envolvam o uso de sistemas de apoio, colaboração e socialização entre usuários (SILVA et al., 2016c).

4.1.2 Visão controlador

O controlador do sistema possui condições de gerenciar a validação das atividades encaminhadas pelos estudantes. A Figura 12 mostra as opções de ações permitidas a esse perfil, onde no “Dashboard” é apresentado a tela inicial do controlador, em “Meu Perfil” o mesmo pode gerenciar os dados pessoais acadêmicos,

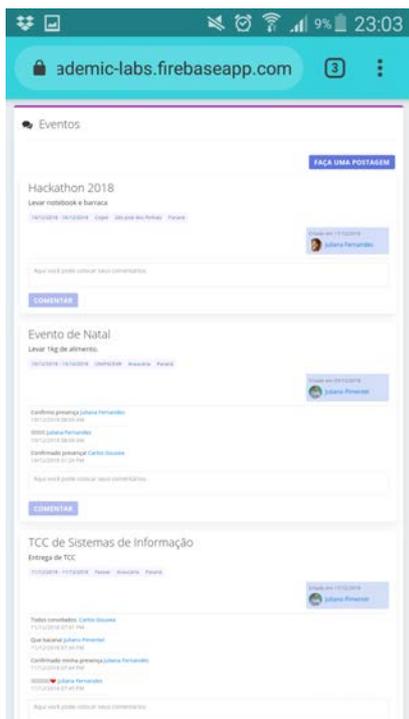


Figura 11: Tela com visão dos eventos postados pelos usuários.



Figura 12: Tela com opções disponíveis a usuários do tipo Controlador.

em “Eventos” - assim como para os estudantes - é possível sugerir atividades coletivas, em

“Tipos de Atividade” são indicados os grupos de atividades e em “Regras” é configurado a quantidade de horas por grupo de atividades.

Na Figura 13 *Dashboard* apresentado inicialmente indica a evolução das atividades entregues ao controlador no último semestre letivo, sempre mostrando a evolução mensal. Abaixo do gráfico de evolução, no gráfico de barras, é indicado a relação dos estudantes que entregaram atividades e o número respectivo de atividades entregues. O objetivo desses gráficos é fazer sempre o acompanhamento dos estudantes com objetivo de orientá-los e realizar políticas de incentivo a participação de eventos dentro do curso que possam ser contabilizados como AC.

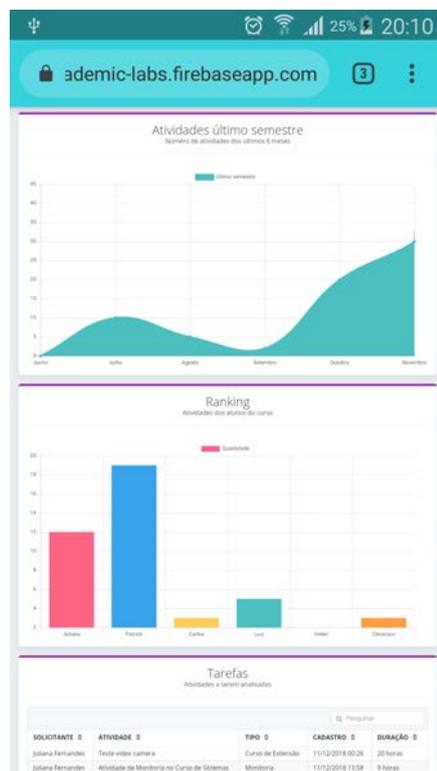


Figura 13: Tela inicial com gráficos de acompanhamento para o Controlador.

Abaixo dos gráficos mostrados anteriormente, uma lista de atividades pendentes para aprovação fica visível ao controlador para que sejam validadas. Ao clicar sobre uma atividade, as informações submetidas pelo estudante são apresentadas, como mostrado na Figura 14. Nesta tela, o controlador verifica se

a atividade atende aos critérios do curso, analisando quantidade de horas, grupo da atividade, período de realização, descrição e o arquivo anexo que confirma a realização da AC.

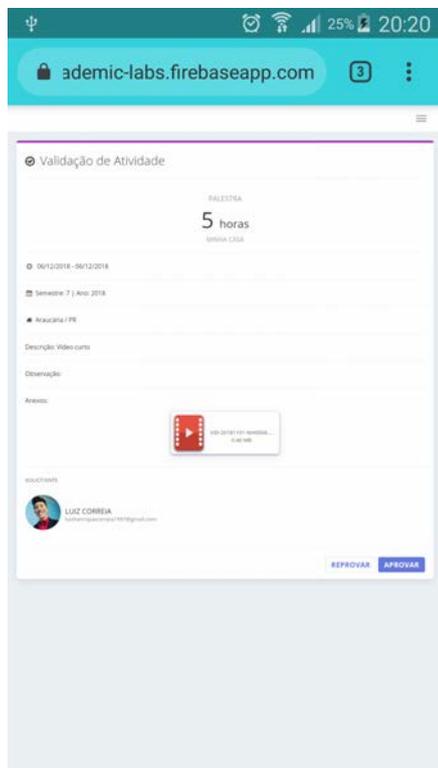


Figura 14: Tela de validação da AC submetida pelo estudante.

Nestas condições, o controlador pode “Reprovar” ou “Aprovar” a AC. No primeiro caso, para reprovação é necessário preencher um “Feedback” informando porque a atividade foi recusada. Ao verificar que a atividade foi recusada, o estudante deve realizar um novo envio considerando as orientações ou razões de reprovação informados pelo controlador. No segundo caso de aprovação, a quantidade de horas é somada ao restante que o estudante já possui dentro do grupo indicado na submissão.

Na opção “Tipo de atividade” é cadastrado o tipo de atividade onde será criada a regra para contabilização das atividades enviadas pelos estudantes. Neste tópico é comum o cadastramento de tipos como: Extensão, Pesquisa e Iniciação Científica, Monitoria, Disciplinas ou Trabalho de Ação Social. Contudo, esses grupos podem mudar de curso para curso. Em

“Regras”, como mostrado na Figura 15, o controlador define o número máximo de horas que cada tipo de atividade permite que o estudante realize. Esse procedimento é necessário de forma a evitar que o estudante monopolize a quantidade de horas realizadas em apenas um tipo de atividade, o que vai em desacordo com o regulamento do curso na maioria das vezes.

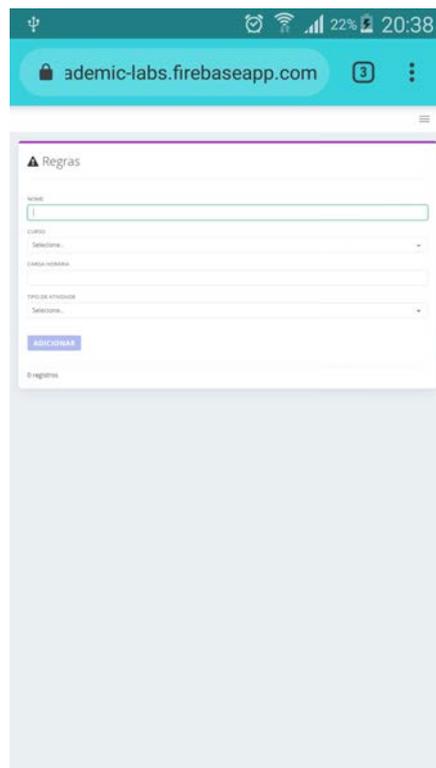


Figura 15: Tela de gerenciamento das regras de AC no curso ao qual o controlador é responsável.

4.1.3 Visão administrador e IES

As visões do administrador e IES possuem como principal objetivo a gerência das IES que utilizam o sistema e no gerenciamento dos controladores e cursos dentro da instituição. Até o momento da escrita desse artigo, essas duas visões ainda não haviam sido desenvolvidas, e estão na lista de funções a serem implementadas em uma *release* (versão) futura.

4.2 Contribuições do Sistema Akademic

Durante o desenvolvimento desse sistema diversas contribuições foram observadas, sendo muitas delas relacionadas as vantagens que os usuários possuirão ao utilizar as funcionalidades disponíveis. Essas vantagens associadas a cada tipo de usuário são listadas abaixo:

- Estudantes: Os jovens cada vez mais procuram facilitar as suas atividades através do uso de tecnologias embarcadas em *smartphones*. O acesso pode ser feito a partir de um cadastro pessoal ou a partir de uma conta de rede social. Com o Akademic, os estudantes tem a disposição mais um recurso interativo e responsivo a diversos tipos de equipamentos e ambientes conectados a Internet. Ainda, possuem um controle mais dinâmico das atividades realizadas, sem a necessidade do envio de documentos impressos ou por email. Acompanhamento da quantidade de atividades realizadas e o montante ainda pendente para entrega ou validação. Participação interativa de grupos de indicação e acompanhamento de atividades coletivas.
- Controladores: Aos coordenadores de cursos, a entrega de atividades é melhor gerenciada devido a facilidade de validação da atividade a partir de qualquer dispositivo conectado a Internet. O uso do sistema, evita uma massiva entrega de documentos e comprovantes para verificação, eliminando assim a possibilidade de acúmulo de papel, possibilidade de perda de comprovantes e necessidade de recontagem dentro de cada grupo de atividades. A partir das regras, os controladores definem limites de atividades, e mesmo que eles aceitem atividades que excedam um grupo, para o estudante apenas é contabilizado até o limite definido.
- IES: As IESs economizarão em diversos aspectos como: diminuição de pessoal dedicado ao recebimento das atividades e

economia de material como papéis. Outro aspecto importante é o cumprimento das normativas e regras estimuladas pelos órgãos de fiscalização do ensino superior no Brasil.

5 Conclusão

Neste artigo foi apresentado um sistema *web* multi-plataforma, chamado de Akademic, para dispositivos móveis ou não, que tem por objetivo auxiliar que estudantes, controladores, coordenadores de cursos e instituição de ensino superior gerenciem o processo de validação de atividades complementares em cursos de graduação. Além de permitir mobilidade pelo uso de dispositivos móveis conectados a Internet, o sistema permite o compartilhamento de atividades coletivas e também que esse mesmo sistema possa ser utilizado por diversas instituições sem a necessidade de sistema sob demanda.

Através de tecnologias de desenvolvimento de *software* sofisticados como o *Angular* e banco de dados não-relacional, o Akademic emprega outras tecnologias que permitem que a experiência dos usuários seja adequada no quesito de usabilidade e que apresenta soluções para os problemas dos usuários. Com o sistema implementado, podemos verificar que o Akademic apresenta funcionalidades primordiais que permita que estudantes possam gerenciar as suas atividades complementares de maneira mais simples e controlada. Aos controladores, a facilidade e agilidade na verificação e validação das AC dos estudantes se mostrou de grande ajuda, pois eliminou a necessidade de recebimento de solicitações que antes eram realizadas manualmente.

Como proposta de trabalhos futuros, os autores estimam a implementação de outras funcionalidades como: a possibilidade de geração de certificados *online*, a criação de grupos de atividades coletivas que sejam validadas por professores, a validação parcial de uma AC pelo controlador, entre outras funcionalidades que possam melhorar ainda mais a experiência dos usuários.

Referências

- AGARWAL, U. *Hands-on Full Stack Development with Angular 5 and Firebase: Build real-time, serverless, and progressive web applications with Angular and Firebase*. Birmingham, Reino Unido: Packt Publishing, 2018.
- ALBERTENGO, G. et al. On the performance of Web Services, Google Cloud Messaging and Firebase Cloud Messaging. *Digital Communications and Networks*, p. 1–9, 2019.
- BEANE, J. A. a essência de uma escola democrática. *Currículo sem fronteiras*, v. 3, n. 2, p. 91–110, 2003.
- BECK, K. et al. *Manifesto for agile software development*. Disponível em: <<http://www.manifestoagil.com.br/>>. Acesso em 10 de Junho de 2018.: [s.n.], 2001.
- BRASIL. *Ministério da Educação. Conselho Nacional de Educação e Câmara de Educação Superior. PARECER CNE/CES Nº:239/2008. Carga horária das atividades complementares nos cursos superiores de tecnologia*. [S.l.], 2008.
- BRAY, T. *RFC 8259 - The javascript object notation (JSON) data interchange format*. [S.l.], 2017.
- CHENG, F.; CHENG, F. *Build Mobile Apps with Ionic 4 and Firebase*. [S.l.]: Springer, 2018.
- COSTA, B. C. SisAti: um sistema para controle de atividades complementares. *Repositório Digital-Trabalhos de Conclusão de Curso (Graduação)*, n. 1, 2016.
- GOOGLE. *Firestore helps mobile and web app teams succeed*. 2019. Disponível em: <<https://firebase.google.com/>>.
- HAN, J. et al. Survey on NoSQL database. In: *2011 6th International Conference on Pervasive Computing and Applications*. Port Elizabeth, South Africa: IEEE, 2011. p. 363–366.
- LIMA, M. M. de et al. Analisando processos de desenvolvimento de software educativo a partir de uma revisão sistemática de literatura. *Revista de Informática Aplicada*, v. 11, n. 1, 2016.
- LÓSCIO, B. F.; OLIVEIRA, H. d.; PONTES, J. d. S. Nosql no desenvolvimento de aplicações web colaborativas. In: *VIII Simpósio Brasileiro de Sistemas Colaborativos*. [S.l.: s.n.], 2011. v. 10, n. 1, p. 1–11.
- MEC. *Perguntas frequentes sobre educação superior: Atividades Complementares*. Disponível em: <<http://portal.mec.gov.br/sesu-secretaria-de-educacao-superior/perguntas-frequentes>>>. Acesso em 18 de Novembro de 2018.: [s.n.], 2018.
- OLIVEIRA, C. T. de; SANTOS, A. S. dos; DIAS, A. C. G. Percepções de estudantes universitários sobre a realização de atividades extracurriculares na graduação. *Psicologia: Ciência e Profissão*, Conselho Federal de Psicologia, v. 36, n. 4, p. 864–876, 2016.
- PAULA, L. B. *Desenvolvimento de um software web para gestão de eventos acadêmicos e gerenciamento da carga horária de atividades complementares e estágio curricular em java utilizando os frameworks JSF e Primefaces*. Ituverava, SP: Faculdade Dr. Francisco Maeda, Curso de Bacharel em Sistemas de Informação, 2014.
- RIZZI, H. S. et al. Desenvolvimento de um sistema web para acesso e controle de atividades complementares em cursos superiores. *Revista Univap*, v. 22, n. 40, p. 678, 2017.
- SEWAK, M.; SINGH, S. Winning in the era of serverless computing and function as a service. In: *IEEE. 2018 3rd International Conference for Convergence in Technology (I2CT)*. Pune, India, 2018. p. 1–5.
- SILVA, A. M. et al. Adoção de técnicas fuzzy para análise de usabilidade de software. *Revista de Informática Aplicada*, v. 11, n. 1, 2016.

- SILVA, A. S. S.; CONCEIÇÃO, E. B.; SILVA, L. S. Desenvolvimento de um sistema de controle de atividade complementares em ruby on rails. 2016.
- SILVA, C. A. G. et al. Análise da influência da formação de ensino médio no desempenho acadêmico de estudantes de engenharia. *International Journal on Alive Engineering Education*, v. 3, n. 2, p. 65–78, 2016.
- SILVA, C. A. G. et al. A utilização do scrum como recurso educacional no processo de aprendizagem em engenharia de software. *International Journal on Alive Engineering Education*, v. 3, n. 2, p. 87–102, 2016.
- SILVA, D. M. A. et al. Abordagem utilizando o design science research para o desenvolvimento de sistema colaborativo assistivo. *Revista de Informática Aplicada*, v. 12, n. 1, 2016.
- SILVA, L. A.; LOPES, F. S.; VALLIM-FILHO, A. R. A. Uma proposta de modelagem analítica para o uso efetivo de atividades complementares no ensino superior. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. Dourados, MS: CBIE, 2014. v. 3, n. 1, p. 611.
- SILVA, L. A.; VALLIM-FILHO, A. R. A. Uma experiência na utilização do ambiente virtual de aprendizagem moodle no gerenciamento de atividades complementares. In: *Anais do Congresso Brasileiro de Recursos Digitais na Educação*. Sao Paulo, SP: Mackenzie, 2012.
- SILVA, L. A.; VALLIM-FILHO, A. R. A. Um processo de gestão e análise de atividades complementares com o uso do ambiente virtual de aprendizagem moodle. *Revista Trilha Digital*, v. 1, n. 1, 2013.
- SOUZA, H. V. L. de et al. Discussão sobre as abordagens associadas à aprendizagem autodirigida e sua relação com as tecnologias educacionais. *Revista de Informática Aplicada*, v. 13, n. 1, 2017.
- SPARTA, M.; GOMES, W. B. Importância atribuída ao ingresso na educação superior por alunos do ensino médio. *Revista brasileira de orientação profissional*, v. 6, n. 2, p. 45–53, 2005.
- VANUCHI, M. C. S. et al. Software para gerenciamento de atividades complementares. In: *III Semana de Tecnologia da Informação*. Paranavaí, PR: IFPR, 2016. p. 1–17.