



UNIVERSIDADE DO SUL DE SANTA CATARINA

LEONARDO J. C. MAY

TIAGO B. BOEING

BIBLIOTECA DOS CURSOS

Sistema para gerenciamento de atividades acadêmicas e
obtenção de código-fonte do GitHub de forma automática

Tubarão

2020

LEONARDO J. C. MAY

TIAGO B. BOEING

BIBLIOTECA DOS CURSOS

Sistema para gerenciamento de atividades acadêmicas e
obtenção de código-fonte do GitHub de forma automática

Trabalho de Conclusão de Curso
apresentado ao curso de Ciência da
Computação, da Universidade do Sul
de Santa Catarina, como requisito
parcial à obtenção do título de
Bacharel em Ciência da Computação.

Orientador: Prof. Clávison Martinelli Zapelini

Tubarão

2020

LEONARDO J. C. MAY

TIAGO B. BOEING

BIBLIOTECA DOS CURSOS

Sistema para gerenciamento de atividades acadêmicas e
obtenção de código-fonte do GitHub de forma automática

Este Trabalho de Conclusão de Curso foi
julgado adequado à obtenção do título de
Bacharel em Ciência da Computação e
aprovado em sua forma final pelo Curso de
Ciência da Computação, da Universidade
do Sul de Santa Catarina.

Tubarão, 15 de julho de 2020.

Professor e orientador Clávison Zapelini

Prof. Carlos Fernando Buss

Prof. Luciano Sávio

Prof. Márcia Cargnin Martins Giraldi

RESUMO

Este artigo aborda o desenvolvimento de uma ferramenta voltada ao gerenciamento de fluxo de trabalho de professores de cursos de tecnologia. Embora existam softwares que oferecem formas de realizar a correção de questões objetivas, poucos possuem como foco o auxílio através de rotinas automáticas de download e organização do código-fonte submetido pelos alunos, juntamente com a separação lógica de turmas e atividades. Atualmente este é um processo custoso para o professor, sendo necessário visitar diversas páginas e estabelecer um fluxo de trabalho para centralização das informações recebidas de diferentes fontes. A ferramenta desenvolvida neste artigo visa auxiliar no download das informações obtidas a partir do GitHub e centralizá-las em um único local, não sendo dessa forma necessário que o professor acesse cada perfil individualmente. Com a otimização desses processos muitas das vezes complexo, consequentemente será viável oferecer funcionalidades extras com o decorrer do tempo, como análise de similaridade de atividades, feedback por linha de código e/ou pontuação por requisitos atendidos.

Palavras-chave: Professor. GitHub. Educação.

ABSTRACT

This article discusses the development of a tool aimed at managing the workflow of teachers of technology courses. Although there is software that offers ways to correct objective questions, few focus on assistance through automatic download routines and organization of the source code submitted by students, along with the logical separation of classes and activities. This is currently a costly process for the teacher, and it is necessary to visit several pages and establish a workflow to centralize the information received from different sources. The tool developed in this article aims to assist in downloading the information obtained from GitHub and centralizing it in a single location, so it is not necessary to access each profile individually. With the optimization of these often-complex processes, consequently, it will be feasible to offer extra features over time, such as analysis of similarity of activities, feedback for a line of code, and/or punctuation for requirements met.

Keywords: Teacher. GitHub. Education.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 Caso de uso - criar conta | 25 |
| Figura 2 Representação MER no cenário de banco relacional | 27 |
| Figura 3 Linguagens de programação e banco de dados suportados..... | 29 |
| Figura 4 Diagrama de comunicação das tecnologias da solução..... | 30 |
| Figura 5 Mecanismo de execução e sincronização de uma atividade..... | 32 |
| Figura 6 Verificação de agendamentos para os próximos 15min..... | 33 |
| Figura 7 Disparo da função BibliotecaMessagesHandler | 33 |
| Figura 8 Ao centro BibliotecaMessagesHandler..... | 34 |
| Figura 9 Cada aluno representa uma mensagem | 34 |
| Figura 10 Primeira etapa do fluxo de sincronização..... | 35 |
| Figura 11 Tela de cadastro..... | 40 |
| Figura 12 Redirecionamento para autorizar utilização de conta do GitHub..... | 41 |
| Figura 13 Usuário deve autorizar utilização de sua conta | 41 |
| Figura 14 Página inicial da aplicação | 42 |
| Figura 15 Lista de alunos cadastrados..... | 43 |
| Figura 16 Cadastro de novo aluno | 44 |
| Figura 17 Criação de turma..... | 45 |
| Figura 18 Lista de turmas..... | 45 |
| Figura 19 Número indicador da quantidade de alunos e atividades..... | 46 |
| Figura 20 Adicionando aluno em uma turma..... | 46 |
| Figura 21 Criando atividade em uma turma | 47 |
| Figura 22 Cadastro de uma atividade | 48 |
| Figura 23 Lista de atividades de uma turma..... | 49 |
| Figura 24 Lista de materiais de uma atividade | 50 |
| Figura 25 Cenário em que o repositório não existe na conta do aluno | 50 |
| Figura 26 Comparação de fluxo de trabalho | 51 |

LISTA DE QUADROS

| | |
|--|----|
| Tabela 1 Múltipla escolha - recebimento de atividades práticas | 20 |
| Tabela 2 Período entre agosto de 2019 e junho de 2020 | 22 |
| Tabela 3 Caso de uso: cadastrar conta manual | 25 |
| Tabela 4 Caso de uso: cadastrar conta com o GitHub | 26 |
| Tabela 5 Caso de uso: cadastrar alunos | 26 |
| Tabela 6 Caso de uso: cadastrar turma | 26 |
| Tabela 7 Caso de uso: cadastrar atividade | 27 |
| Tabela 8 Caso de uso: cadastrar alunos na atividade..... | 27 |
| Tabela 9 Modelo de dados do Prisma | 39 |

SUMÁRIO

| | |
|--|----|
| 1. INTRODUÇÃO | 10 |
| 2. OBJETIVOS | 12 |
| 2.1. GERAL | 12 |
| 2.2. ESPECÍFICO | 12 |
| 3. RESULTADOS ESPERADOS | 13 |
| 4. JUSTIFICATIVA | 14 |
| 5. REVISÃO TEÓRICA RESUMIDA | 16 |
| 6. TRABALHOS CORRELATOS | 18 |
| 7. METODOLOGIA | 19 |
| 7.1. LEVANTAMENTO DE DADOS | 19 |
| 7.1.1. INTENÇÃO DE USO | 19 |
| 7.2. PLANO DE TRABALHO | 20 |
| 7.2.1. DEFINIÇÃO DE ETAPAS | 20 |
| 7.2.2. CRONOGRAMA | 21 |
| 8. REQUISITOS | 23 |
| 8.1. REQUISITOS FUNCIONAIS (RF) | 23 |
| 8.2. REQUISITOS NÃO FUNCIONAIS (RNF) | 23 |
| 9. DIAGRAMAS | 25 |
| 9.1. DIAGRAMA DE CASOS DE USO | 25 |
| 9.1.1. CRIAÇÃO DE CONTA | 25 |
| 9.1.2. CADASTRO DE ALUNO | 26 |
| 9.1.3. CADASTRO DE TURMA | 26 |
| 9.1.4. CADASTRAR ATIVIDADE | 26 |
| 9.2. DIAGRAMA MER | 27 |
| 10. DESENVOLVIMENTO | 28 |
| 10.1. CONTEXTUALIZAÇÃO | 28 |
| 10.2. TECNOLOGIAS UTILIZADAS | 30 |
| 10.3. ARQUITETURA DA SOLUÇÃO | 30 |
| 10.3.1. ARCHITECTURE PRINCIPLES | 31 |
| 10.3.2. EXECUÇÃO DE AGENDAMENTOS | 31 |
| 10.3.3. ESTRUTURA DOS DADOS | 36 |

| | |
|--|----|
| 11. FUNCIONALIDADES | 40 |
| 11.1. CRIAÇÃO DE CONTA | 40 |
| 11.1.1. VIA GITHUB | 40 |
| 11.2. PÁGINA INICIAL | 42 |
| 11.1. GERENCIAMENTO DE ALUNOS..... | 42 |
| 11.1.1. LISTA DE ALUNOS | 42 |
| 11.1.2. CADASTRAR DE ALUNO | 44 |
| 11.2. GERENCIAMENTO DE TURMAS | 44 |
| 11.2.1. ADICIONANDO ALUNOS..... | 46 |
| 11.3. GERENCIAMENTO DE ATIVIDADES | 47 |
| 11.3.1. CRIAR ATIVIDADE..... | 47 |
| 11.4. LISTAGEM DE MATERIAIS..... | 49 |
| 11.5. COMPARATIVO DE FLUXO DE TRABALHO | 51 |
| 12. CONSIDERAÇÕES FINAIS | 52 |
| 13. CONCLUSÃO | 53 |
| 14. REFERÊNCIAS BIBLIOGRÁFICAS..... | 54 |

1. INTRODUÇÃO

A tecnologia é uma área de extrema importância para qualquer país e através de suas inovações auxilia na impulsão de economias. É inegável sua contribuição para o mundo globalizado que conhecemos e a necessidade em praticamente qualquer local, especialmente aqueles países em desenvolvimento, grupo ao qual o Brasil pertence (INTERNATIONAL MONETARY FUND, 2017). (RODRIK, 2018) em artigo publicado no Fórum Econômico Mundial afirma que: “Os países em desenvolvimento devem atualizar seus sistemas educacionais e treinamento técnico, melhorar seu ambiente de negócios e suas redes de logística e transporte, a fim de fazer uso mais completo de novas tecnologias”.

Avanços tecnológicos demandam de mão-de-obra qualificada e consequentemente cresce a oferta de cursos na área, em paralelo com a quantidade de professores e alunos, fomentando oportunidades para ferramentas voltadas a auxiliar na resolução de possíveis desafios no cenário em que as práticas educacionais são mediadas por tecnologia ou fortemente relacionadas a mesma, como o caso dos cursos voltados a formar profissionais para atuação na área.

“Nos últimos anos, houve investimentos significativos nas tecnologias de aprendizagem digital em todos os setores. Notavelmente, o setor educacional tem sido mais adaptável na implementação de plataformas inteligentes de aprendizado, pois torna o aprendizado mais simples, fácil e eficaz para os alunos. O ensino superior está passando por melhorias significativas desde a expansão da tendência do *e-learning*. O *e-learning* ajuda os educadores a obter um maior grau de cobertura para comunicar a mensagem de forma consistente ao seu público-alvo. Assim, a crescente adoção de plataformas de aprendizado inteligente baseadas na Web e móveis no setor educacional está elevando o crescimento do mercado de educação inteligente e aprendizado.” (FORTUNE BUSINESS INSIGHTS, 2018)

A tecnologia aplicada às práticas educacionais tem criado um mercado que constantemente cresce a fatia de participação no mundo de software. Ainda segundo estudo publicado em 2018 pela (FORTUNE BUSINESS INSIGHTS, 2018) e que utiliza dados de 2015 à 2017, a América do Norte tem se mostrado a região dominante no mercado global de ensino e aprendizagem inteligente, gerando em 2018 uma receita de US\$ 59,42 bilhões e devendo chegar a US\$ 240,44 bilhões até 2026. O estudo cita

ainda que o avanço desse setor se deve principalmente ao crescimento econômico estável e aos desenvolvimentos tecnológicos recentes.

Da perspectiva de tecnologia educacional para crianças, publicação recente e ainda positiva de abril de 2020, considerando o cenário COVID-19 segundo (LI e LALANI, 2020) mesmo antes da pandemia já havia adoção e crescimento em tecnologia educacional, com investimentos globais atingindo US\$ 18,66 bilhões em 2019 e o mercado geral de educação on-line se projeta para atingir US\$ 350 bilhões em 2025. Foi constatado que o uso desde o COVID-19 aumentou.

2. OBJETIVOS

Este trabalho aborda a jornada de desenvolvimento de uma ferramenta voltada auxiliar na facilitação do fluxo de trabalho de um professor no processo de recepção e/ou obtenção de atividades avaliativas em que o aluno gera um código-fonte, além de gerenciar turmas e alunos. Atualmente esta jornada tende a exigir a utilização de recursos mais tradicionais como e-mail, diferentes softwares de EAD, em algumas ocasiões não preparados para tal cenário sendo menos especialistas, mídias removíveis ou ferramentas de armazenamento em nuvem (*cloud*). Para isto é necessário investir um tempo significativo no gerenciamento das informações e estabelecimento de um fluxo de trabalho, posteriormente com estas soluções é possível iniciar a atividade de correção.

2.1. GERAL

Ferramenta para auxiliar o fluxo de trabalho de professores ao realizar download de repositórios do GitHub, gerenciar turmas e alunos.

2.2. ESPECÍFICO

- Identificar padrões no fluxo de trabalho realizado (durante correções de avaliações práticas) por professores de cursos de tecnologia;
- Definir as ferramentas para desenvolvimento da aplicação;
- Definir técnicas para integração de rotinas com as APIs envolvidas durante o fluxo de desenvolvimento;
- Desenvolver a aplicação;

3. RESULTADOS ESPERADOS

Espera-se obter a adoção da ferramenta por parte de professores, sendo eles o público-alvo e através da utilização gerar impactos positivos quanto ao processo de recepção das atividades avaliativas práticas, já que o maior benefício se encontra em centralizar os materiais oriundos de diversas fontes (e-mail, sistemas EAD, *pendrive*, armazenamento na nuvem).

Progressivamente se espera o incentivo à adoção e inclusão das ferramentas de controle de versões (versionamento de código) no currículo acadêmico de cursos de tecnologia, sendo estas indispensáveis para futuros profissionais que adentrarão ao mercado de trabalho. Para tal ação o professor é o protagonista e grande influenciador no processo de evangelização.

Outro resultado esperado é a validação da experiência de usabilidade da ferramenta bem como os recursos dessa primeira versão, sendo estes limitados a um escopo devido a disponibilidade de tempo.

4. JUSTIFICATIVA

Com a constante evolução das tecnologias, a presença significativa no cotidiano e a necessidade por profissionais para suprir às demandas do setor, instituições de ensino ampliam ofertas por especializações e consequentemente há aumento de matrículas em cursos da área de tecnologia, independente de modalidade (G1, 2018).

“Até 2022, funções em volta de tecnologia, como Analista de dados, Cientista de dados, Desenvolvedor de softwares e aplicativos, Especialista em e-commerce e Analista em redes sociais integram a lista de profissões em destaque.” (UBM, 2019)

Em cursos voltados a área de tecnologia, o aluno terá contato com disciplinas de desenvolvimento de software (a exemplo de graduações como Análise e Desenvolvimento de Sistemas, Sistemas de Informação, Ciência da Computação e afins), sendo comum que em determinado momento um professor forneça atividades práticas em que um código-fonte deve ser entregue por parte do aluno.

Com o exponencial crescimento da área tecnológica, especialmente no Brasil, levando-se em consideração os últimos anos é possível constatar que tais números fomentam oportunidades para aplicações diversas, inclusive aquelas que têm por objetivo facilitar o fluxo de trabalho de professores. Outro ponto que justifica investimentos é que segundo (YOUNG DIGITAL PLANET, 2015) STEM é a sigla em inglês de ciência, tecnologia, engenharia e matemática – quatro disciplinas básicas essenciais para o desenvolvimento da inovação tecnológica atual e futura.

“Como a humanidade continua a desenvolver novas tecnologias, ampliar o conhecimento e explorar novas áreas, o número de novas profissões em que são necessárias as habilidades STEM cresce diariamente.” (YOUNG DIGITAL PLANET, 2015)

Segundo artigo de (PAULA e MOREIRA, 2014) sobre práticas pedagógicas e formação de professores a mais influente perspectiva de avaliação escolar ainda em vigor pode ser denominada como modelo psicométrico, as limitações desse modelo, chamado de condutista visam avaliar as capacidades e conhecimentos predefinidos dos estudantes. O artigo fala ainda da fragilidade em limitar o ensino apenas aos instrumentos convencionais.

[...] restringir a avaliação a provas e testes equivale a exigir que os estudantes usem o lápis e o papel como únicos recursos mediacionais e a escrita como único modo de comunicação para expressar conhecimentos cuja construção e utilização depende de outros recursos mediacionais e de múltiplos modos de comunicação. (PAULA; MOREIRA, 2014)

[...] Quando consideramos ambientes educacionais que restringem as ações dos estudantes a escutar o que professor diz, a ler informações que os livros apresentam e a reproduzir os "conhecimentos" supostamente adquiridos por meio da realização de exercícios de lápis e papel, existe compatibilidade entre os recursos mediacionais usados e os instrumentos de avaliação restritos às provas e testes. Todavia, trata-se de uma compatibilidade que limita a diversificação das atividades de ensino e aprendizagem, ampliando as chances de os estudantes desenvolverem uma relação alienada e alienante com a atividade escolar. (PAULA e MOREIRA, 2014)

Com base no exposto, a ferramenta visa auxiliar nos processos necessários para a organização desses códigos-fontes, oriundos de diferentes atividades e turmas. Atualmente o trabalho costuma ser manual, sendo necessário visitar links individualmente e verificar todas essas fontes de dados. As tarefas envolvidas podem ser otimizadas através de uma ferramenta que centralize o recebimento das atividades.

É importante ter ciência do potencial das ferramentas voltadas ao controle de versões (versionamento de código) e que elas sejam abordadas durante o processo de ensino em universidades e cursos profissionalizantes voltados a área de tecnologia, sendo seu entendimento fundamental para atuação em desenvolvimento de software.

O Git foi utilizado como base para o desenvolvimento por ser a ferramenta de versionamento de código mais popular de todos os tempos e amplamente utilizada no mercado de trabalho. Já o GitHub o site de armazenamento Git mais popular de todos, gratuito e acessível para qualquer pessoa com a simples configuração de uma conta. (KADIVAR, 2018)

5. REVISÃO TEÓRICA RESUMIDA

Conforme (KENSKI, 2003) as tecnologias voltadas a mediação da aprendizagem, requerem um amplo conhecimento de suas especificidades para que possam ser utilizadas adequadamente em projetos sistemáticos de educação. A autora afirma ainda que é necessária a utilização ampla das capacidades humanas (muito além da cognição) em processos diferenciados de aprendizagem.

Segundo (PAULA e MOREIRA, 2014) na atualidade as tecnologias têm oferecido novos desafios. Os autores citam que as possibilidades de acesso à informação, interação e comunicação tem trazido novas formas de aprendizagem. O artigo conclui ainda que “[...] as atuais tecnologias digitais de comunicação e informação possibilitam o alcance de novas aprendizagens, que encaminham as pessoas para novos avanços, socialmente válidos, no atual estágio de desenvolvimento da humanidade”, embora (LÉVY, 1994) tenha citado ainda à época que “O preparo dos docentes brasileiros para a utilização de mídias e objetos digitais como materiais didático-pedagógicos ainda é insipiente”, a situação no Brasil praticamente permanece a mesma.

Já (MARTINES, MEDEIROS, *et al.*, 2018) citam que “O avanço da informática proporciona aos educadores grandes variedades de meios e recursos no sentido de auxiliá-los em seu trabalho. O surgimento das tecnologias e métodos de ensino propõe a utilização de instrumentos mais eficazes no que se refere ao ensino em sala de aula”, sendo assim podemos utilizar o ponto de vista de que a popularização dos sites dedicados a oferecer armazenamento de código-fonte gratuitamente através do protocolo Git, como o GitHub, tem aberto precedentes para que ferramentas como a do objetivo deste artigo surjam e visem auxiliar nestas questões ainda em aberto e que são mapeadas com o decorrer da utilização das mesmas.

O processo avaliativo sempre fez parte da aprendizagem e do ambiente educacional desde o início da vida escolar, sendo uma forma de aferir os conhecimentos de um indivíduo e se necessário adaptar a metodologia adotada para que ele atinja as habilidades necessárias. Conforme (GASPAR) “Sendo a avaliação uma das etapas da atividade escolar, é necessário que esteja sintonizada com a finalidade do processo ensino e aprendizagem e como possibilidade de perceber nos sujeitos escolares suas fragilidades, seus avanços e desta forma, mediar o processo

de apropriação do conhecimento [...]”. Embora não seja exclusividade ao mundo dos cursos de tecnologia as avaliações mediadas por atividades práticas são instrumentos que proporcionam maior interação para o aluno e alinhamento com as atividades do mercado de trabalho, sendo estas de extrema importância para a preparação do mesmo durante o processo de formação.

6. TRABALHOS CORRELATOS

O trabalho correlato que mais assemelha com os objetivos do artigo é o GitHub Classroom, acessível pelo endereço <https://classroom.github.com/> . A aplicação é uma ferramenta desenvolvida pela equipe do GitHub e atualmente permite a utilização do recurso “organização” que o site possui para gerenciar múltiplos repositórios (neste cenário cada aluno representa um repositório). Primeiramente é necessário que o professor crie uma organização ou vincule uma já existente. Logo após é necessário convidar os alunos para participarem da organização, sendo que este processo acaba não sendo instantâneo já que cada aluno precisará confirmar através de um link enviado ao e-mail. Feito isto será possível criar uma nova “atribuição” — forma com que as atividades são tratadas na plataforma. Há dois tipos possíveis de atribuições:

- Trabalho individual: cada participante trabalha individualmente em seu próprio repositório.
- Trabalho de grupo: os participantes formam equipes que trabalham juntas em um repositório compartilhado.

Independentemente do tipo escolhido, o professor poderá fornecer uma base de código inicial, isto é, um projeto com o ambiente já configurado ou uma *template* HTML por exemplo, no caso de um projeto web. Para projetos em grupo é possível definir o máximo de usuários por equipe e a quantidade máxima de equipes. *Deadline*, isto é, prazo para envio também pode ser configurada.

A aplicação para salas de aula do GitHub deixa alguns problemas em aberto, como a necessidade de convidar os usuários para organização, o que acaba demandando de uma ação secundária por parte dos usuários (professor que organizam turmas), como instalar um software no computador para conseguir baixar (*clone*) os repositórios, não é possível realizar agendamentos, as informações estarão presentes apenas no computador em que o software foi instalado e a organização será fixa com cada repositório (aluno) representando uma subpasta no sistema.

7. METODOLOGIA

A metodologia adotada no artigo foi de utilizar dados coletados através de pesquisas bibliográficas, análise de ferramentas correlatas e obtenção de feedback com profissionais atuantes na área de educação em cursos de tecnologia. O público-alvo dessa proposta são professores que buscam uma forma de gerenciar turmas e o processo de correção de atividades avaliativas práticas, podendo a solução ser utilizada por demais usuários com necessidades semelhantes, embora o desenvolvimento não seja voltado a estes.

7.1. LEVANTAMENTO DE DADOS

7.1.1. INTENÇÃO DE USO

No ano de 2019 durante validação da proposta do artigo, foi realizada a coleta de informações referente a intenção de uso e enviada a pessoas próximas dos autores, sendo professores ou que tenham vínculo com a área de educação. O link de participação para preenchimento de um formulário Google Forms foi enviado a partir de 06 de outubro de 2019. Entre os destinatários, professores das universidades UNISUL - Universidade do Sul de Santa Catarina, FURB - Universidade Regional de Blumenau e colaboradores do Sesi e do SENAI - Litoral Sul que atuam nos serviços de educação, consultoria em educação, serviços tecnológicos e saúde.

Resultados do levantamento

No total cinco professores participaram respondendo ao levantamento de dados. 80% possuem professor como a atividade profissional principal, sendo que 80% exercem outra profissão de forma paralela, todas na área de TI. Dos 5, 4 responderam positivamente quanto a utilização de alguma tecnologia em sala de aula. Quanto ao recebimento das atividades avaliativas práticas:

| Alternativa | Respostas |
|---|-----------|
| Armazenamento cloud (Google Drive e outros) | 4 de 5 |
| Via e-mail | 3 de 5 |
| Ferramenta da universidade | 3 de 5 |
| Ferramentas Git | 2 de 5 |
| Pendrive | 1 de 5 |

Tabela 1 Múltipla escolha - recebimento de atividades práticas

100% considera que essas tecnologias agregam valor para o processo de ensino; 60% costuma fornecer feedback individual por código; 60% não acredita que as ferramentas atuais são suficientes e 100% estaria disposto a utilizar outras ferramentas.

7.2. PLANO DE TRABALHO

7.2.1. DEFINIÇÃO DE ETAPAS

Algumas etapas foram definidas durante o processo de desenvolvimento do artigo para atingimento dos objetivos.

Análise de fluxo

Análise da jornada do usuário de trabalhos correlatos e outras ferramentas disponibilizadas no mercado. Comportamentos e/ou necessidades exclusivas foram analisadas individualmente a fim de determinar se estão ou não contempladas pelos objetivos do artigo.

Pesquisa bibliográfica

Com o refinamento das necessidades da ferramenta e o fluxo previamente estabelecido, foi realizado um estudo relacionado às tecnologias adequadas para auxiliar na resolução do problema, como integrações, APIs e técnicas.

Desenvolvimento do modelo

O modelo foi elaborado baseado em um *mockup* inicial, diagrama de classes e arquitetura da solução.

Elaborar fluxo de desenvolvimento

Foram criados os projetos, fluxo de desenvolvimento, *CI/CD* e outras etapas.

| | | | | | | | | | | | | |
|---|--|--|--|--|--|--|--|--|--|--|--|--|
| (Leonardo May) | | | | | | | | | | | | |
| Análise dos resultados obtidos (Leonardo May, Tiago Boeing) | | | | | | | | | | | | |

Tabela 2 Período entre agosto de 2019 e junho de 2020

8. REQUISITOS

8.1. REQUISITOS FUNCIONAIS (RF)

Os requisitos funcionais (RF) mapeados são os seguintes:

- um professor deve ser capaz de criar turmas, as quais representarão salas de aula com seus respectivos alunos;
- os alunos serão representados por contas do GitHub, não sendo necessário nenhuma ação por parte do mesmo;
- o professor deve realizar o cadastro dos alunos informando nome, sobrenome e nome de usuário do GitHub;
- permitir ao professor associar alunos a determinada turma;
- cada turma pode ou não conter atividades cadastradas por determinado professor; onde cada atividade contém:
 - título;
 - descrição textual a ser fornecida pelo professor;
 - data para sincronização do material.
- permitir ao professor visualizar os materiais sincronizados de cada atividade na data informada;
- permitir identificar sincronizações que falharam, a exemplo de o sistema tentar sincronizar um determinado repositório e o mesmo não existir na conta do usuário.

8.2. REQUISITOS NÃO FUNCIONAIS (RNF)

- o *front-end* é implantado na web utilizando o serviço Hosting do Firebase;
- o empacotamento do *front-end* da aplicação é realizado utilizando o framework *Angular*;
- o *back-end* deve ser implantado utilizando AWS ECR, AWS ECS e AWS ELB para balanceamento de carga ou ferramentas similares;
- o desenvolvimento *back-end* utiliza a biblioteca Prisma para auxiliar na elaboração da API *GraphQL* em *NodeJS*;
- a base de dados utiliza o banco não relacional orientado a documentos *MongoDB (NoSQL)*;

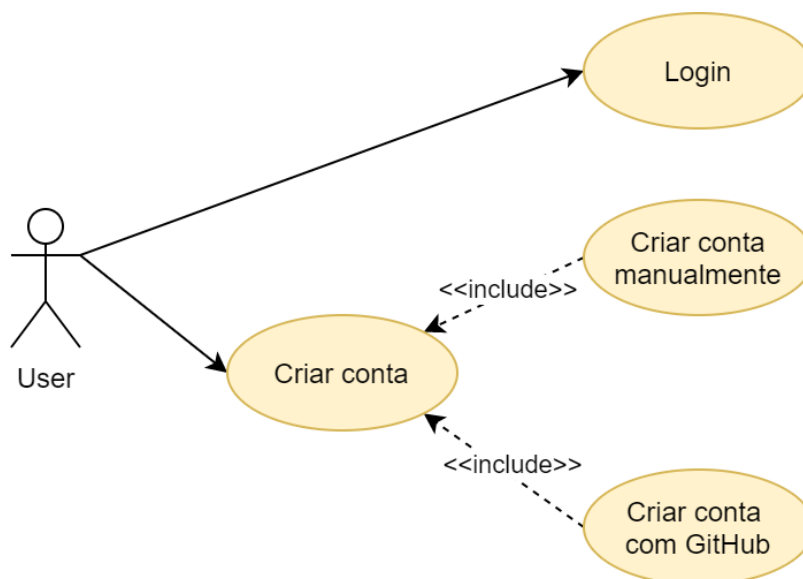
- a interface suportará os navegadores web mais utilizados no mercado e compatível com as versões mais recentes dos mesmos.

9. DIAGRAMAS

9.1. DIAGRAMA DE CASOS DE USO

9.1.1. CRIAÇÃO DE CONTA

Figura 1 Caso de uso - criar conta



Fonte: do próprio autor (2020)

| Cadastrar conta manualmente | |
|------------------------------------|--|
| Descrição | O ator deve informar o seu nome e sobrenome, e-mail, nome do usuário e senha para realizar o cadastro. |
| Atores | Usuário. |
| Pré-Condições | É obrigatório informar o nome e sobrenome, e-mail, nome do usuário e senha para ser efetuado o cadastro. |
| Cenário Principal | 1. Informar o nome e sobrenome, e-mail, nome do usuário e senha. |

Tabela 3 Caso de uso: cadastrar conta manual

| Cadastrar Conta com GitHub | |
|-----------------------------------|---|
| Descrição | O ator deve clicar no botão “Criar conta com GitHub”. |
| Atores | Usuário. |
| Pré-Condições | Será necessário iniciar uma sessão no site do GitHub e autorizar a utilização de sua conta. |

| | |
|-------------------|--|
| Cenário Principal | <ol style="list-style-type: none"> 1. Iniciar sessão com GitHub caso ainda não estiver logado 2. Autorizar utilização da conta |
|-------------------|--|

Tabela 4 Caso de uso: cadastrar conta com o GitHub

9.1.2. CADASTRO DE ALUNO

| | |
|-------------------|--|
| Descrição | O ator deve navegar na lista de alunos e clicar no botão “Adicionar aluno”. |
| Atores | Usuário. |
| Pré-Condições | É obrigatório navegar para a tela alunos. Preencher nome, sobrenome, provedor e nome de usuário no provedor. |
| Cenário Principal | <ol style="list-style-type: none"> 1. Navegar para lista de alunos 2. Clicar no botão “Adicionar aluno” 3. Preencher os respectivos campos 4. Salvar aluno |

Tabela 5 Caso de uso: cadastrar alunos

9.1.3. CADASTRO DE TURMA

| | |
|-------------------|--|
| Descrição | O ator deve informar o nome da turma para realizar o cadastro. |
| Atores | Usuário. |
| Pré-Condições | É obrigatório acessar a lista de turmas, clicar em “Criar turma” e informar o nome da turma. |
| Cenário Principal | <ol style="list-style-type: none"> 1. Navegar para lista de turmas 2. Clicar em “Criar turma” 3. Preencher os campos 4. Salvar turma |

Tabela 6 Caso de uso: cadastrar turma

9.1.4. CADASTRAR ATIVIDADE

| | |
|-------------------|---|
| Descrição | O ator deve informar o nome da atividade, nome do repositório a ser criado e data de sincronização. |
| Atores | Usuário. |
| Pré-Condições | Necessário acessar uma turma e clicar em “Criar atividade”. É obrigatório informar o título, data de sincronização e repositório a ser criado. |
| Cenário Principal | <ol style="list-style-type: none"> 1. Navegar para lista de turmas 2. Clicar em “Criar atividade” 3. Preencher os campos |

| | |
|--|---------------------|
| | 4. Salvar atividade |
|--|---------------------|

Tabela 7 Caso de uso: cadastrar atividade

Cadastrar aluno na atividade

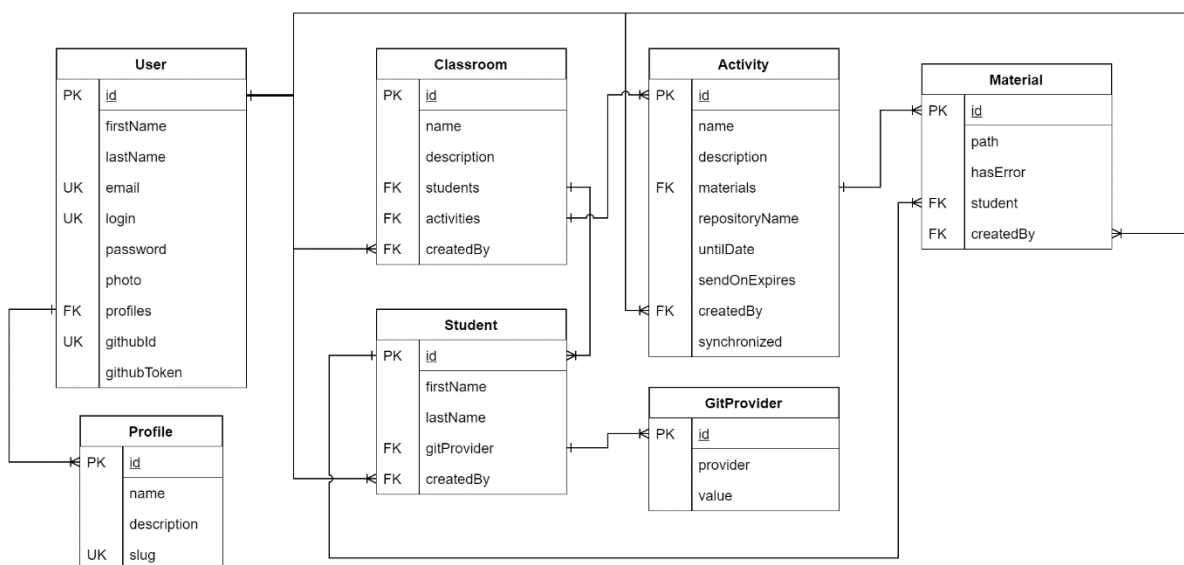
| | |
|-------------------|---|
| Descrição | O ator deve informar o nome do aluno que deseja adicionar na turma. |
| Atores | Usuário. |
| Pré-Condições | Necessário acessar uma turma, buscar um aluno e adicionar. |
| Cenário Principal | <ol style="list-style-type: none"> 1. Acessar a turma 2. Clicar em “Adicionar alunos” 3. Buscar aluno por nome 4. Clicar em “Adicionar aluno” |

Tabela 8 Caso de uso: cadastrar alunos na atividade

9.2. DIAGRAMA MER

Embora o banco de dados utilizado seja o MongoDB e se trate de um banco não relacional (NoSQL) orientado a documentos, há relacionamento entre os registros, mesmo que lógico ou embutido (comum em bancos do tipo). Na ferramenta foi utilizado o relacionamento lógico, referenciando o *ObjectId* e utilizando *aggregation* ou *Fragment* do GraphQL para obter as informações necessárias (JOIN em bancos SQL). É possível ilustrar os “relacionamentos” da mesma forma que em um banco relacional, visto que a estrutura das entidades não sofreu alterações.

Figura 2 Representação MER no cenário de banco relacional



Fonte: do próprio autor (2020)

10. DESENVOLVIMENTO

Para o desenvolvimento da solução foram utilizadas diferentes tecnologias para *front-end*, *back-end*, *serverless* e infraestrutura, algumas exclusivas do provedor de serviços *cloud* aplicado no artigo, no caso a Amazon Web Services.

O desenvolvimento local do *back-end* foi realizado através da utilização do *Docker*. O código é compilado com as alterações mais recentes dentro do próprio contêiner e fica disponível para utilização na sequência.

Para o desenvolvimento do *front-end* foi utilizada a biblioteca de componentes Angular chamada *Nebular* que segue as especificações do *Eva Design System*. O tema base é o *ngx-admin* que foi construído utilizando os componentes *Nebular*, sendo todos da mesma empresa, *Akveo*. Os repositórios são *open source*, em junho de 2020 o *ngx-admin* continha mais de 20 mil *stars* no GitHub (*stars* são utilizadas para medir a popularidade de um repositório). Todas as tecnologias utilizadas possuem a licença MIT, que é uma das mais permissivas.

10.1. CONTEXTUALIZAÇÃO

A computação em nuvem atualmente tem se tornado uma escolha quase que unânime para a construção de novas soluções, é importante ressaltar que “a computação em nuvem é uma nova maneira de entregar os recursos de computação, não uma nova tecnologia” (DUPRÉ e HAEBERLEN, 2012). “O principal objetivo da computação em nuvem é fazer um melhor uso dos recursos distribuídos, combiná-los para obter maior rendimento e ser capaz de resolver problemas de computação em larga escala. A computação em nuvem lida com virtualização, escalabilidade, interoperabilidade, qualidade de serviço e entrega modelos da nuvem, nomeadamente privados, públicos e híbridos”. (JADEJA e MODI, 2012)

Para atendimento aos princípios e objetivos previamente alavancados foram mapeadas tecnologias e técnicas para otimização da força de trabalho, como utilização de metodologias ágeis na divisão e organização de tarefas.

Uma API GraphQL através da utilização da biblioteca *Prisma* foi desenvolvida visando disponibilizar maior responsabilidade ao *front-end* e evitar problemas como *over fetching* e *under fetching*, comum em APIs baseadas apenas em REST.

GraphQL e REST são arquiteturas de *design* de APIs, ambos utilizam maneiras diferentes de enviar dados através do protocolo HTTP. Em REST é comum lidar com casos em que o retorno de determinadas requisições contenha mais informações que o cliente realmente necessita, neste cenário temos o *over fetching*. No caso em que a resposta da solicitação contenha menos informações é comum ser necessário realizar requisições adicionais para completar estes dados, isto se chama *under fetching*. Através do envio de uma única requisição os resolvedores do GraphQL são capazes de coletar as informações necessárias.

Prisma e representação a nível conceitual

Segundo documentação oficial, o Prisma preenche a lacuna entre o banco de dados e os resolvedores do GraphQL. Ele substitui os *ORMs* tradicionais e facilita a implementação de servidores GraphQL resilientes e prontos para produção. A Figura 3 Linguagens de programação e banco de dados suportados exibe as abstrações já implementadas pelo Prisma para comunicação com diversas tecnologias de banco de dados utilizando as linguagens de programação mais populares no mercado. Para algumas delas o suporte é parcial e se encontra em desenvolvimento.

Figura 3 Linguagens de programação e banco de dados suportados



Fonte: Prisma (2019)

A Figura 4 Diagrama de comunicação das tecnologias da solução representa a nível conceitual o funcionamento da solução desenvolvida. O *front-end (client-side)* é uma página estática que realiza chamadas de API através do GraphQL, que repassa as requisições para um interceptador responsável pela execução das regras de segurança e posteriormente a consulta à base de dados e retorna ao *front-end*.

Seguindo a arquitetura é possível ter n *front-ends* realizando consultas simultâneas à API.

Figura 4 Diagrama de comunicação das tecnologias da solução



Fonte: Prisma (2019)

10.2. TECNOLOGIAS UTILIZADAS

Como ferramenta de metodologias ágeis foi utilizado o ClickUp, contendo todo o cronograma, divisão de tarefas e entregas. Quanto as ferramentas tecnológicas, para o desenvolvimento *front-end* foi utilizado Angular 8, HTML e CSS, *back-end* Node.JS, GraphQL e Prisma, os serviços da Amazon Web Services, sendo AWS Lambda, AWS SQS (para utilização do protocolo AMQP), AWS EventBridge, AWS CloudWatch, AWS Route 53 e AWS S3, Docker para contêineres, ferramentas de CI/CD do GitLab, MongoDB para banco de dados e Firebase (hosting) para hospedagem de páginas estáticas.

10.3. ARQUITETURA DA SOLUÇÃO

Para a definição da arquitetura de determinadas soluções é comum que princípios sejam alavancados, permitindo maior entendimento e alinhamento dos objetivos do negócio bem como as implementações tecnológicas a ser desenvolvidas.

Esta seção especifica o trabalho através da descrição de casos de uso, diagramas, requisitos funcionais e não funcionais.

10.3.1. ARCHITECTURE PRINCIPLES

Segundo artigo publicado pela organização The Open Group¹ e baseado no trabalho realizado pela Força Aérea dos EUA no estabelecimento de seus Princípios da Força Aérea para Gerenciamento de Informações (29 de junho de 1998), os princípios de arquitetura (em português) podem ser definidos como “regras e diretrizes gerais subjacentes ao uso e implantação de todos os recursos e ativos de TI. Eles refletem um nível de consenso entre os vários elementos da empresa e formam a base para a tomada de decisões futuras de TI. Cada princípio de arquitetura deve estar claramente relacionado aos objetivos de negócios e aos principais fatores de arquitetura”. (THE OPEN GROUP, 1999)

Levando em consideração tal definição, foram elaborados alguns princípios para a seção de sistemas e tecnologias atuais que vão de acordo com a solução e serviram como base durante a jornada de desenvolvimento:

- As ferramentas tecnológicas envolvidas no processo de desenvolvimento devem prover escalabilidade, sejam destinadas a base de dados, API ou *front-end*;
- Os recursos desenvolvidos devem estar de acordo com especificações de segurança da computação em nuvem;
- As APIs devem prover uma camada de segurança, garantindo que somente usuários com os privilégios necessários consigam acessar e/ou manipular os dados;
- Os recursos (*endpoints*) destinados a disponibilização pública devem adotar o inglês como único idioma;
- Utilizar contêineres sempre que possível ou hospedagem através de aplicações *serverless* para prover alta disponibilidade durante atualizações.

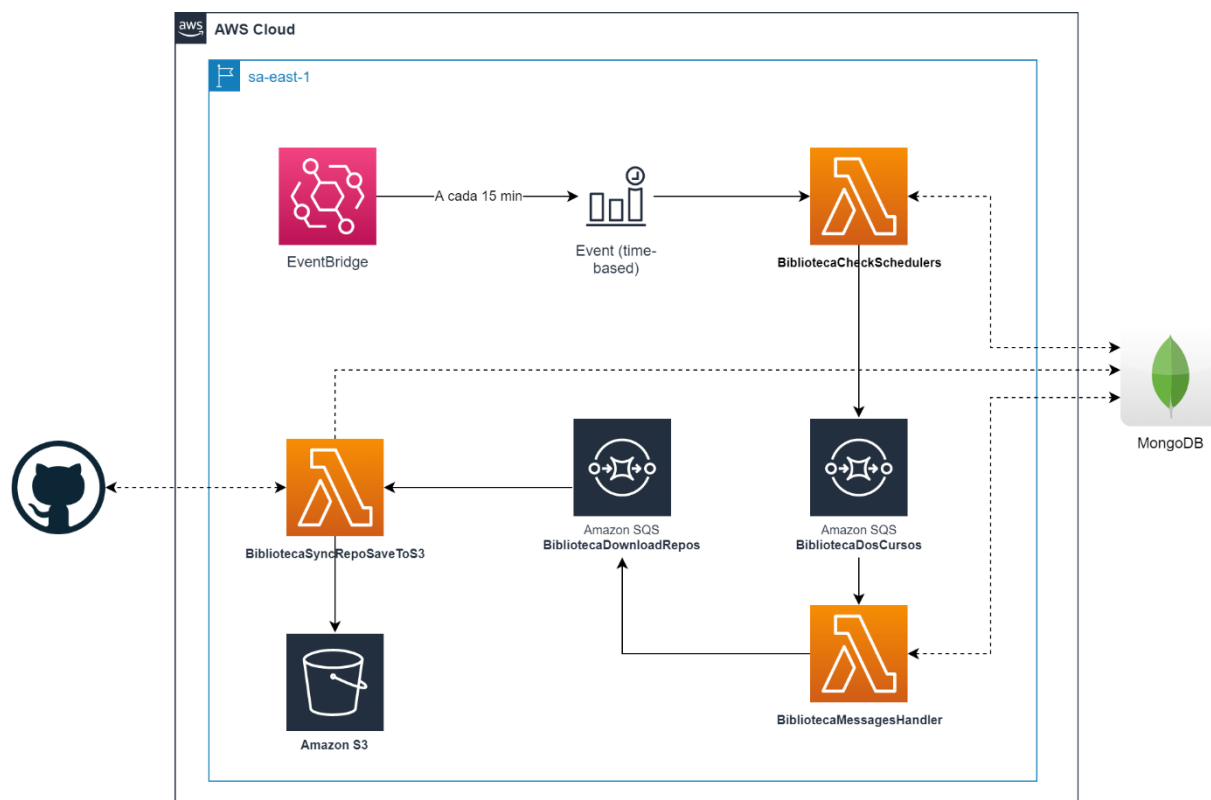
10.3.2. EXECUÇÃO DE AGENDAMENTOS

Quando uma atividade é criada por um professor o campo *untilDate* no tipo *Activity* é utilizado como base para que a função *Lambda* realize a verificação na base.

¹ The Open Group é um consórcio formado por empresas da indústria de informática para estabelecer padrões abertos para a infraestrutura de computação que surgiu em 1996 com a fusão do consórcio X/Open com a Open Software Foundation. — (THE OPEN GROUP, 1999)

Todas as datas são tratadas como UTC, que é a maneira com que foram armazenadas na base de dados. Nos casos em que é necessário conversão para determinado fuso-horário foi utilizada a biblioteca *moment-timezone*.

Figura 5 Mecanismo de execução e sincronização de uma atividade



Fonte: do próprio autor (2020)

Na sequência cada tópico contém a explicação de um fragmento do diagrama.

EventBridge

Existe um agendamento baseado em tarefas *cron* para que a cada 15 minutos o EventBridge dispare um evento de acionamento da AWS Lambda, neste caso a função *BibliotecaCheckSchedulers*.

Lambda: *BibliotecaCheckSchedulers*

Esta função contém a lógica necessária para verificar se existem atividades programadas para sincronização nos próximos 15 minutos, coletando os registros

correspondentes e enviando uma mensagem para o AWS SQS com um *delay* agendado — caso existir, conforme **Figura 6 Verificação de agendamentos para os próximos 15min**.

Figura 6 Verificação de agendamentos para os próximos 15min



Fonte: do próprio autor (2020)

O cálculo do *delay* em que a mensagem será enviada (tempo que levará para chegar até a fila principal) é realizado da seguinte forma:

- **Delay = Hora do agendamento – Hora atual**

A mensagem é enviada para a fila BibliotecaDosCursos com algumas informações como o identificador da atividade.

SQS: BibliotecaDosCursos

Figura 7 Disparo da função BibliotecaMessagesHandler



Fonte: do próprio autor (2020)

Esta é uma fila do tipo padrão do SQS. Responsável por receber as mensagens dos agendamentos com um *delay* programado. Conforme cada mensagem chega na fila a mesma dispara uma *trigger* para iniciar a execução da função Lambda BibliotecaMessagesHandler.

Lambda: BibliotecaMessagesHandler

Figura 8 Ao centro BibliotecaMessagesHandler



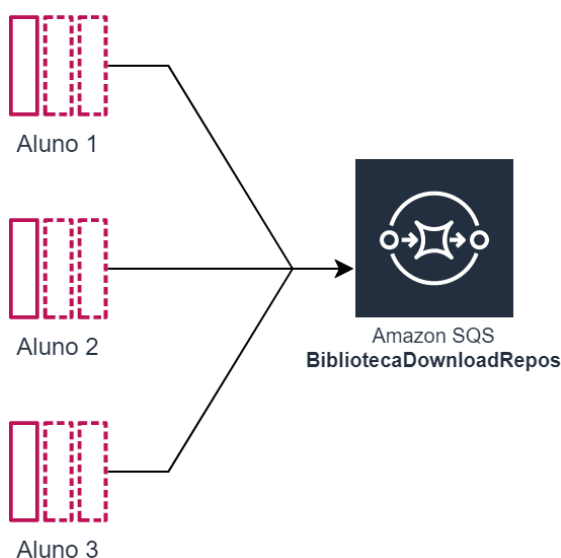
Fonte: do próprio autor (2020)

A função recebe as mensagens da fila BibliotecaDosCursos contendo o identificador da atividade a ser sincronizada, com esta informação a mesma conecta na base MongoDB e obtém mais detalhes para enviar até a fila responsável por disparar a sincronização. A etapa de conexão com a base MongoDB ocorre fora da infraestrutura da AWS. São obtidas as seguintes informações da base:

- Dados completos da atividade, incluindo nome do repositório a ser pesquisado;
- Turma a que a atividade pertence;
- Alunos da turma com nome de usuário no GitHub.

A **Figura 8 Ao centro BibliotecaMessagesHandler** deve ser analisada do centro para as extremidades, sendo que a função realiza a verificação na base e envia a mensagem para o SQS. Cada aluno da atividade será uma mensagem disparada para o SQS conforme **Figura 9 Cada aluno representa uma mensagem**.

Figura 9 Cada aluno representa uma mensagem



Fonte: do próprio autor (2020)

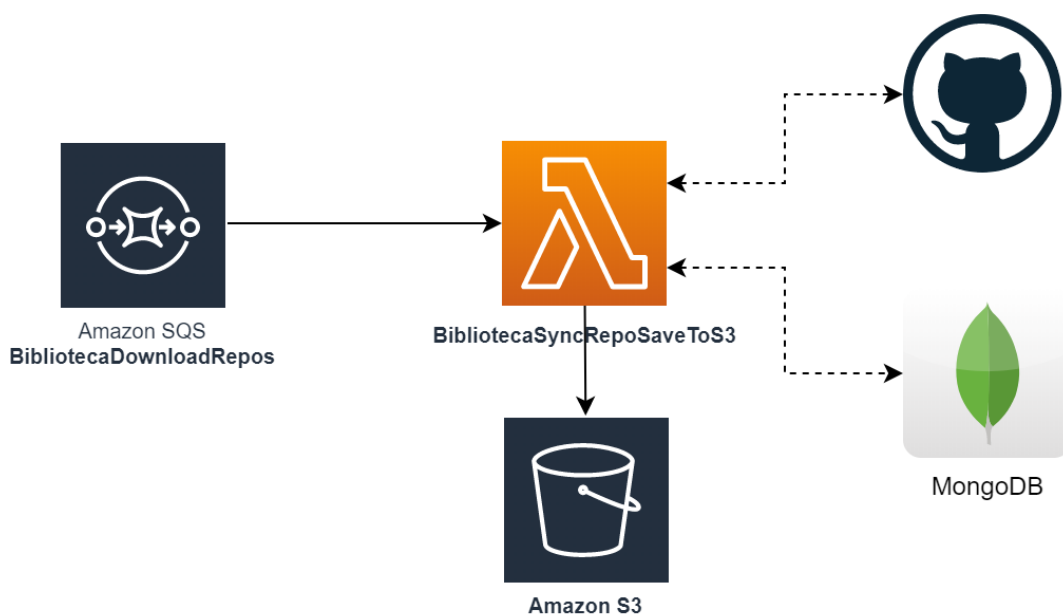
SQS: BibliotecaDownloadRepos

Fila do tipo padrão do AWS SQS. Recebe cada mensagem e dispara uma função Lambda para que a sincronização seja iniciada. Múltiplas mensagens representam múltiplas *invocation* do AWS Lambda, sendo assim é utilizado da simultaneidade da mesma.

Lambda: BibliotecaSyncRepoSaveToS3

É a primeira etapa do fluxo de sincronização, recebe os eventos da fila BibliotecaDownloadRepos com todas as informações referentes a atividade (dados do aluno, turma pai da atividade e nome do repositório a ser pesquisado) já coletadas anteriormente pela função BibliotecaMessagesHandler.

Figura 10 Primeira etapa do fluxo de sincronização



Fonte: do próprio autor (2020)

Os valores JSON devem ser enviados como texto (String) no atributo da mensagem no AWS SQS. Para isso utilizamos *stringify*.

Com as informações *repositoryName* e *value* (do *GitProvider*) da mensagem a função realiza a montagem da URL necessária para obter dados de um repositório no GitHub. Levamos em consideração que os dados são referentes a execução da função apenas uma vez, representando o material de apenas um aluno. A função é executada de forma simultânea para n alunos, onde n é a quantidade de alunos na turma.

Obtendo o *branch* principal é possível realizar a sincronização. A API do GitHub retornará um arquivo do tipo ZIP com todo o conteúdo do repositório e *branch* em questão. Sendo assim o arquivo é armazenado no AWS S3 e um registro é cadastrado na base de dados na *collection Material* contendo o diretório onde se encontra. Se ocorrer um erro no processo de *download* do repositório o material ainda assim será criado, porém com o valor da propriedade *hasError* marcado como verdadeiro, dessa forma é possível que o *front-end* apresente um status de sincronização para o usuário.

No AWS S3 a estrutura de pastas segue o padrão:

- `activities/identificador/materials/identificador/arquivo.zip`

Onde o identificador representa o índice dos respectivos índices na base de dados. Cada arquivo adicionado ao AWS S3 dispara uma *trigger* para que uma função Lambda (não ilustrada no diagrama) realize a descompactação do conteúdo (.zip) e crie uma subpasta. Esta é uma funcionalidade adicionada para futuras implementações.

10.3.3. ESTRUTURA DOS DADOS

A modelagem dos dados quando utilizado *GraphQL* + *Prisma* é dividida em duas etapas (dois arquivos), um deles o *schema.graphql* e outro o *prisma.datamodel*

ANOTAÇÕES

Anotações ou *annotations* são representadas pelo caractere @ e normalmente utilizadas para indicar trechos de um código que devem ser mapeados para uma função de geração de fontes. No caso do *Prisma* as anotações determinam propriedades de uma entidade que devem seguir alguns padrões de dados previamente definidos pela biblioteca, como identificador, valores que necessitam ser únicos, data de criação ou atualização do registro.

Dentre todas as *annotations* do *Prisma* foram utilizadas as mais comuns, sendo:

- **@id** mapeia para objeto do tipo identificador de acordo com o banco de dados utilizado. No caso do MongoDB será utilizado *ObjectId* ;
- **@unique** cria índice identificando para o campo marcado com essa anotação para garantir que seja uma chave única na base de dados;
- **@relation** identifica se tratar de um campo que possui relacionamento com outro *type*, o Prisma passará a armazenar o *ObjectId* do registro relacionado;
- **@default** aplica um valor padrão para o campo caso o mesmo não seja preenchido;
- **@createdAt** preenche o campo com a data e hora da criação do registro. Esta informação não é alterada quando o registro for atualizado;
- **@updatedAt** preenche o campo com a data de criação do registro no momento da inserção e atualiza com a data da modificação conforme sofre alterações.

MODELO DE DADOS

Por não ter sido utilizado um banco relacional, o **modelo de dados** diferente dos tradicionais, sendo definido com base em uma sintaxe própria (do Prisma, em um arquivo `prisma.datamodel`). Campos do tipo data (mesmo aqueles preenchidos pelo usuário) são armazenados na base de dados com o fuso horário UTC, facilitando a conversão para qualquer região, a exemplo do Brasil que se encontra em UTC-3.

Definição

```
type Profile {
  id: ID! @id
  createdAt: DateTime! @createdAt
  updatedAt: DateTime @updatedAt

  name: String!
  slug: String! @unique
  description: String
}
```

```
type User {
  id: ID! @id
  createdAt: DateTime! @createdAt
  updatedAt: DateTime @updatedAt

  firstName: String!
  lastName: String
  email: String! @unique
  login: String! @unique
  password: String
  photo: String
  profiles: [Profile!]! @relation(link: INLINE)
```

```

githubId: String @unique
githubToken: String
}

```

```

type Classroom {
  id: ID! @id
  createdAt: DateTime! @createdAt
  updatedAt: DateTime @updatedAt

  name: String!
  description: String
  students: [Student!]! @relation(link: INLINE)
  activities: [Activity!] @relation(link: INLINE)
  createdBy: User! @relation(link: INLINE)
}

```

```

type Student {
  id: ID! @id
  createdAt: DateTime! @createdAt
  updatedAt: DateTime @updatedAt

  firstName: String!
  lastName: String!
  gitProvider: [GitProvider!] @unique @relation(link: INLINE)
  createdBy: User! @relation(link: INLINE)
}

```

```

type GitProvider {
  id: ID! @id
  createdAt: DateTime! @createdAt
  updatedAt: DateTime @updatedAt

  provider: GitProviderType! @default(value: GITHUB)
  value: String
}

```

```

enum GitProviderType {
  GITHUB
  GITLAB
}

```

```

type Activity {
  id: ID! @id
  createdAt: DateTime! @createdAt
  updatedAt: DateTime @updatedAt

  name: String!
  description: String
  materials: [Material!] @relation(link: INLINE)
  repositoryName: String!
  untilDate: DateTime
  sendOnExpires: Boolean! @default(value: false)
  createdBy: User! @relation(link: INLINE)
  synchronized: Boolean! @default(value: false)
}

```

```

type Material {
  id: ID! @id
  createdAt: DateTime! @createdAt
  updatedAt: DateTime @updatedAt

  path: String
  hasError: Boolean! @default(value: false)
  student: Student! @relation(link: INLINE)
  createdBy: User! @relation(link: INLINE)
}

```

| |
|---|
| } |
|---|

Tabela 9 Modelo de dados do Prisma

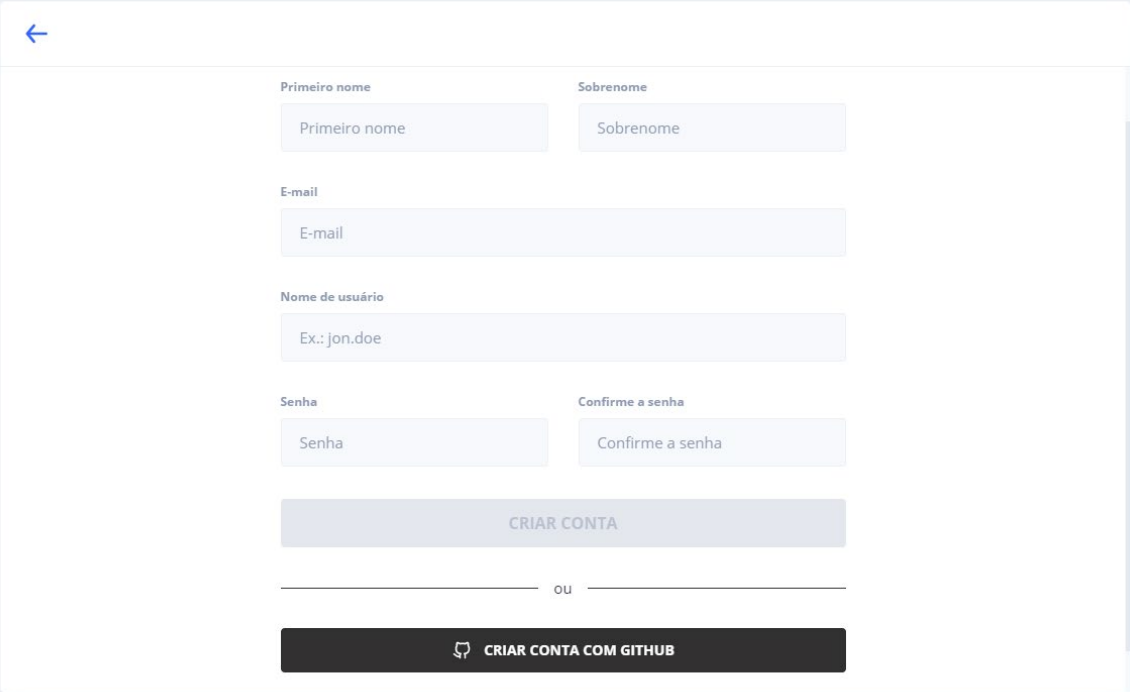
11. FUNCIONALIDADES

11.1. CRIAÇÃO DE CONTA

A criação de uma conta é o primeiro passo, sendo obrigatório para conseguir acessar e utilizar a ferramenta. Um professor pode realizar a criação de duas formas:

- Preencher o cadastro manualmente;
- Criar conta utilizando uma já existente no GitHub.

Figura 11 Tela de cadastro



A imagem mostra a interface de criação de conta de um sistema. No topo, há uma seta azul para voltar. O formulário contém os seguintes campos:

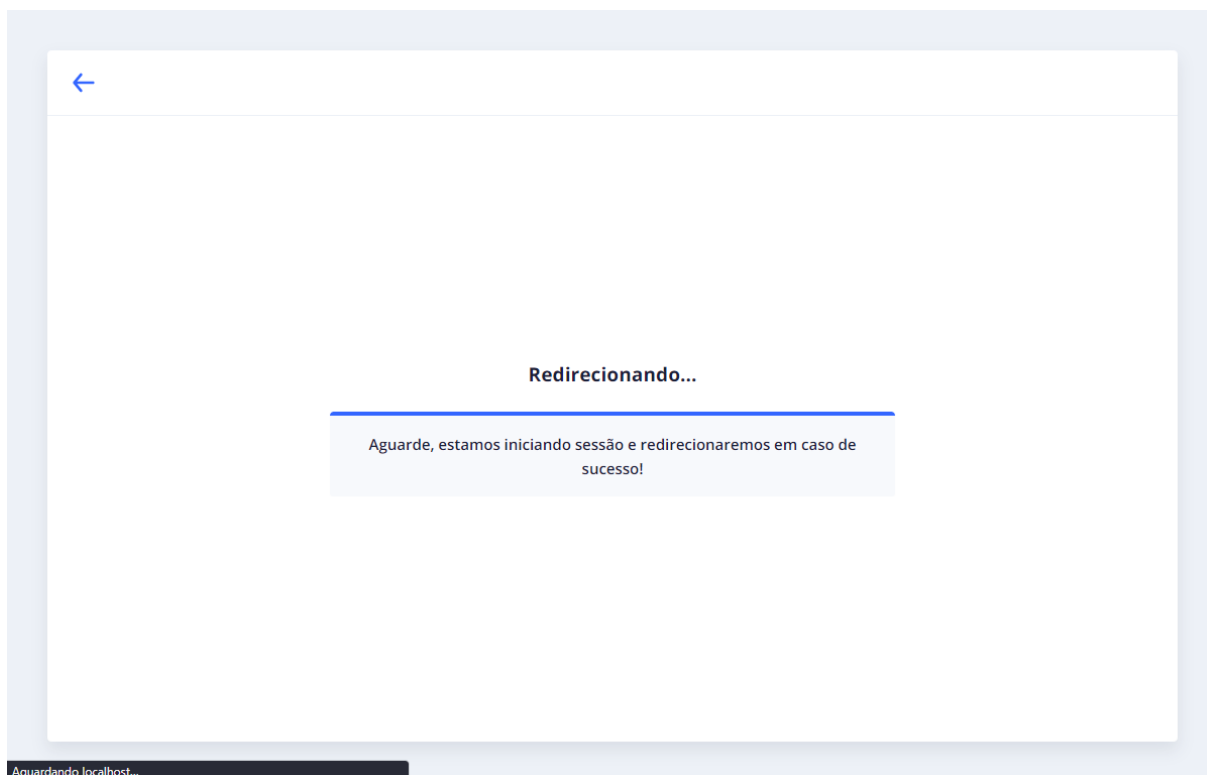
- Primeiro nome** e **Sobrenome**: Campos separados para o primeiro e sobrenome.
- E-mail**: Campo único para o endereço de e-mail.
- Nome de usuário**: Campo único com o exemplo "Ex.: jon.doe".
- Senha** e **Confirme a senha**: Campos separados para a senha e sua confirmação.

Abaixo dos campos, há um botão cinza com o texto "CRIAR CONTA". Logo abaixo, há uma linha com "ou" no centro. No final, há um botão preto com o ícone do GitHub e o texto "CRIAR CONTA COM GITHUB".

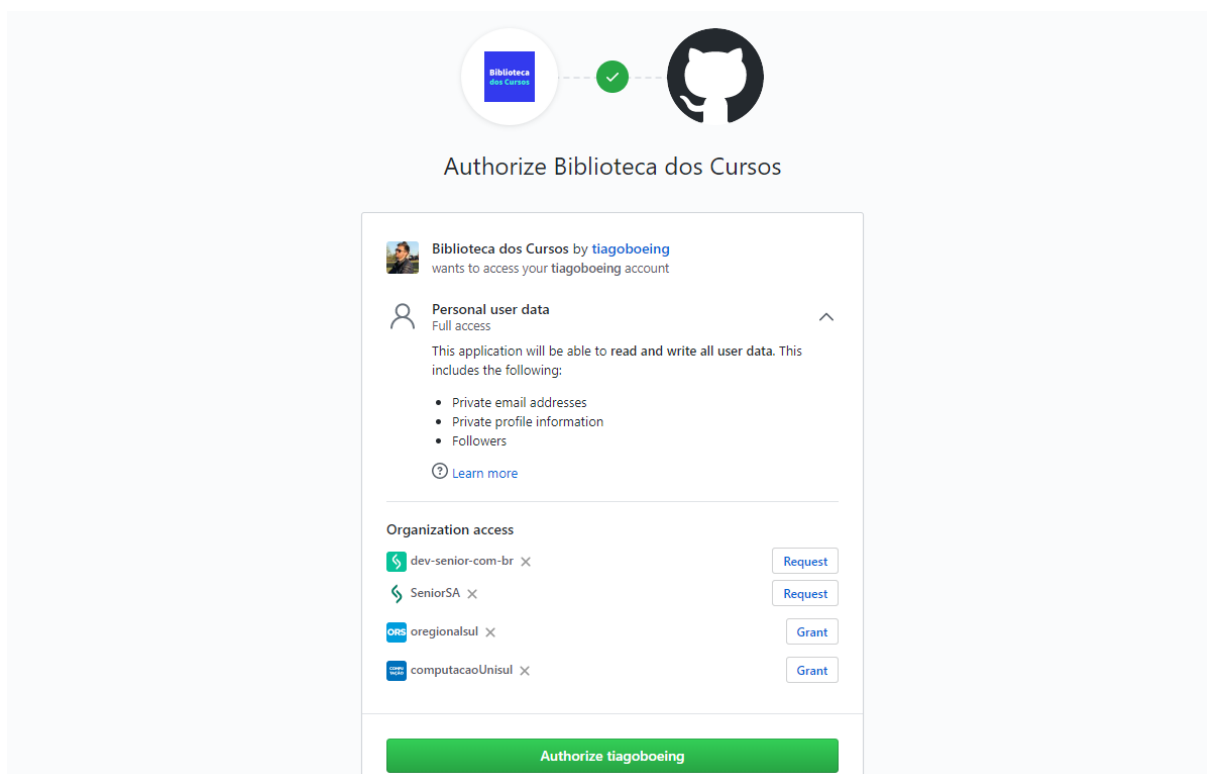
Fonte: do próprio autor (2020)

11.1.1. VIA GITHUB

Ao clicar em "Criar conta com GitHub" o usuário será redirecionado para a página do site que solicita permissões para uma aplicação terceira.

Figura 12 Redirecionamento para autorizar utilização de conta do GitHub

Fonte: do próprio autor (2020)

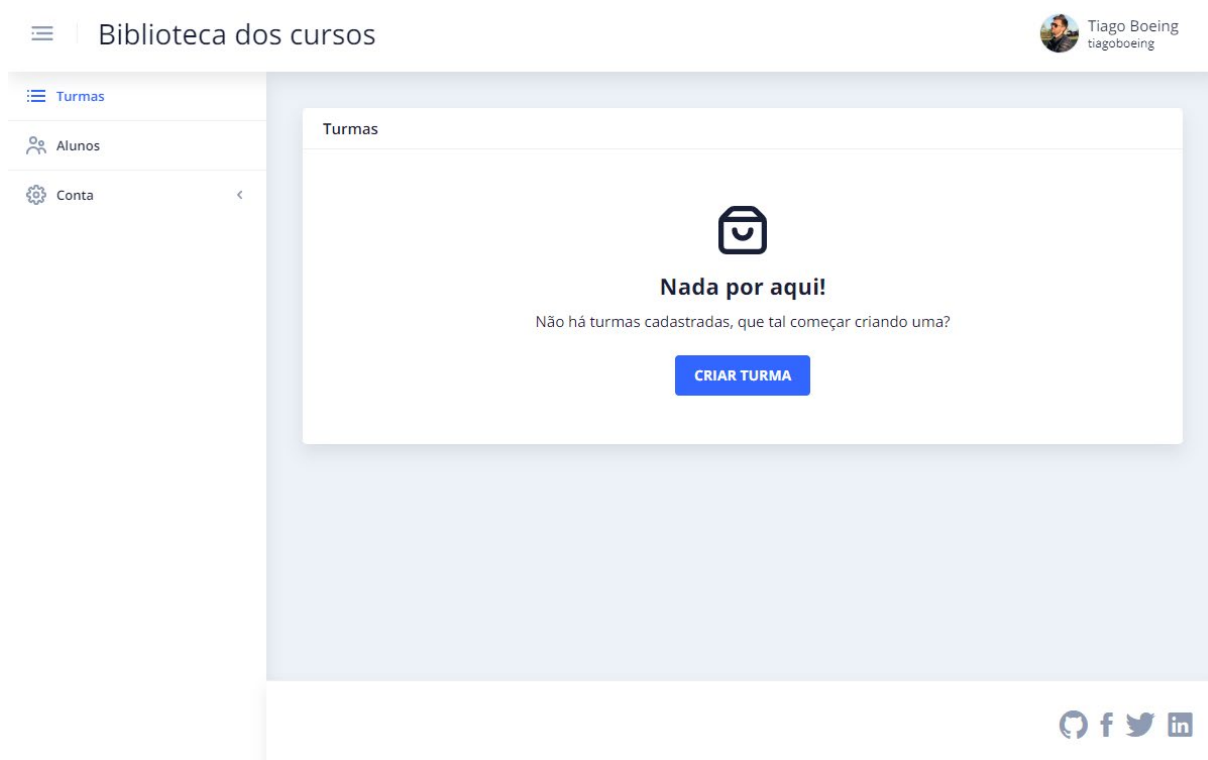
Figura 13 Usuário deve autorizar utilização de sua conta

Fonte: do próprio autor (2020)

Posteriormente ao redirecionamento será exibida uma página do próprio GitHub que solicita que o usuário autorize a aplicação terceira a utilizar sua conta. Tal página fornece informações quanto a privacidade e informa que a aplicação terceira terá acesso ao e-mail, perfil e seguidores. Ao clicar em autorizar será criado o usuário automaticamente e ocorrerá o redirecionamento já logado para a página inicial.

11.2. PÁGINA INICIAL

Figura 14 Página inicial da aplicação



Fonte: do próprio autor (2020)

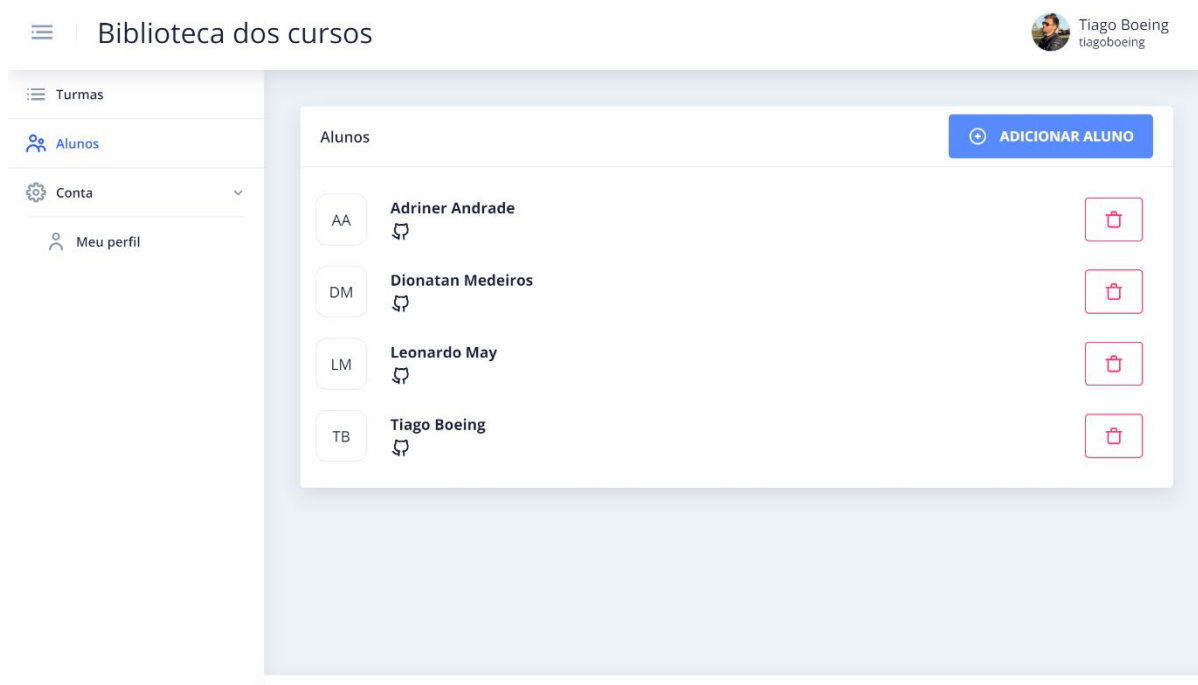
11.1. GERENCIAMENTO DE ALUNOS

11.1.1. LISTA DE ALUNOS

A funcionalidade de alunos não possui dependência de outras, a lista é a representação de todos os alunos ao qual determinado usuário (professor) leciona ou

lecionou. A partir desses registros é possível relacioná-los posteriormente na tela de turmas.

Figura 15 Lista de alunos cadastrados



Fonte: do próprio autor (2020)

A listagem apresenta os alunos cadastrados na conta com a possibilidade de exclusão e seus respectivos usuários no GitHub (fornecidos durante etapa de cadastro).

11.1.2. CADASTRAR DE ALUNO

Figura 16 Cadastro de novo aluno

A imagem mostra a interface de usuário para o cadastro de um novo aluno. No topo, há uma barra de navegação com o título 'Biblioteca dos cursos' e o perfil do usuário 'Tiago Boeing tiagoboeing'. À esquerda, um menu lateral contém opções como 'Turmas', 'Alunos', 'Conta' e 'Meu perfil'. O formulário principal, intitulado 'Adicionar aluno', possui campos para 'Primeiro nome' e 'Sobrenome'. Abaixo, há uma seção 'Provedores Git' com o texto 'Serão utilizados para sincronizar materiais.' e campos para selecionar o 'Provedor' (com 'GITHUB' selecionado) e o 'Nome de usuário'. Botões de ação incluem 'ADICIONAR ALUNO' no topo direito, 'ADICIONAR PROVEDOR' e 'SALVAR ALUNO' no rodapé do formulário. À direita do formulário, há uma barra lateral com quatro ícones de lixeira.

Fonte: do próprio autor (2020)

Durante a etapa de cadastro existem os seguintes campos:

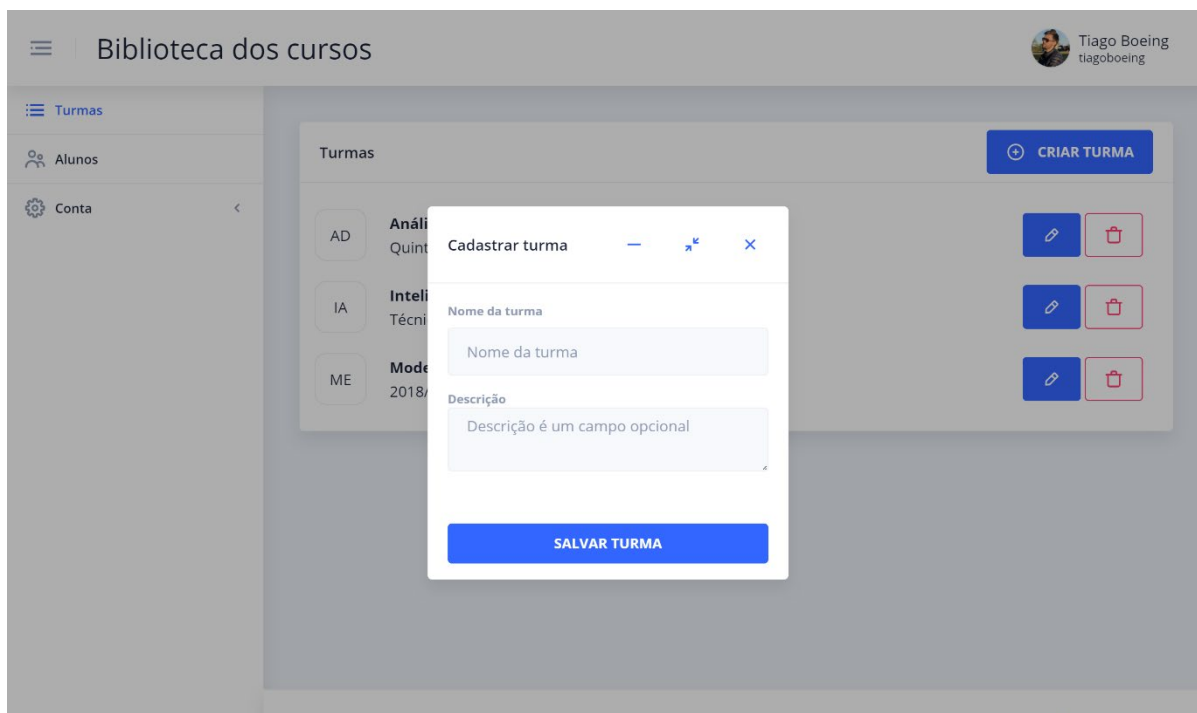
- Primeiro nome (obrigatório);
- Sobrenome (obrigatório);
- Provedor Git; onde:
 - Necessário selecionar o provedor, atualmente apenas GitHub disponível;
 - Informar o nome de usuário do aluno (no GitHub);

11.2. GERENCIAMENTO DE TURMAS

Para o gerenciamento de turmas estão disponíveis as ações de criar, editar e excluir. Uma turma contém atualmente:

- Nome (obrigatório);
- Descrição (opcional).

Figura 17 Criação de turma



Fonte: do próprio autor (2020)

Figura 18 Lista de turmas



Fonte: do próprio autor (2020)

Ao clicar sobre determinado registro na listagem o usuário será encaminhado para os detalhes, onde será possível gerenciar alunos e atividades.

Figura 19 Número indicador da quantidade de alunos e atividades



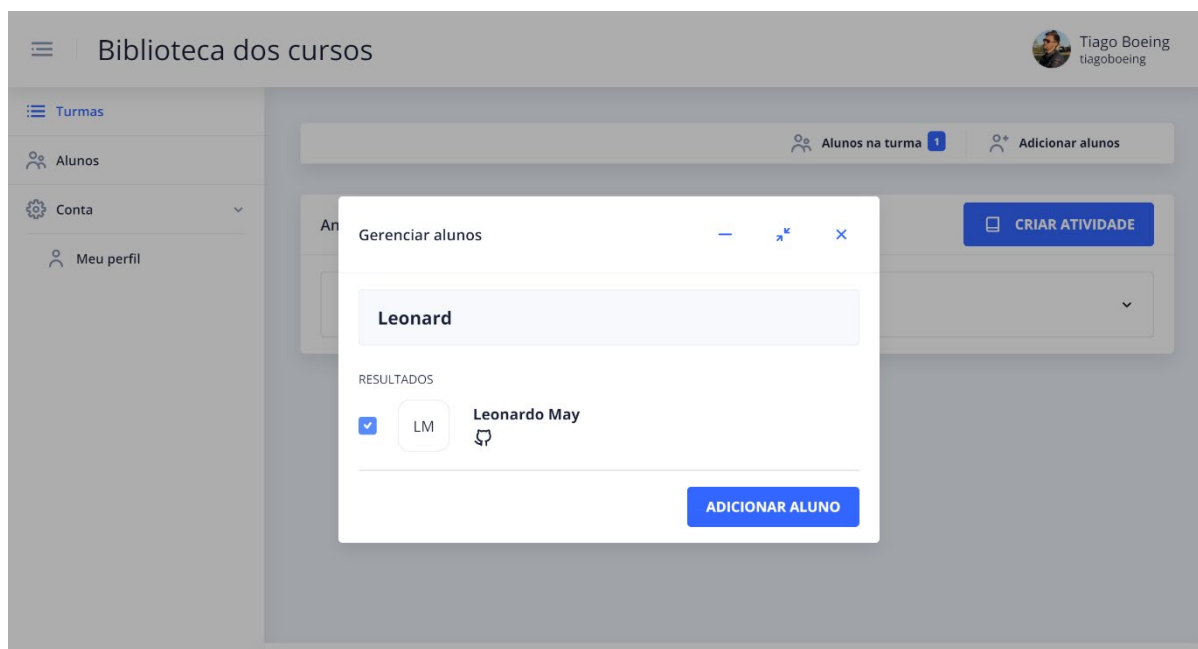
Fonte: do próprio autor (2020)

Caso a turma contenha alunos associados e atividades criadas será exibido um *badge* indicando a quantidade de cada um deles para facilitar a localização na listagem conforme **Figura 19 Número indicador da quantidade de alunos e atividades**.

11.2.1. ADICIONANDO ALUNOS

Para adicionar alunos em uma turma, basta acessar os detalhes e clicar no topo superior direito em “Adicionar alunos” — os alunos devem ter sido cadastrados previamente.

Figura 20 Adicionando aluno em uma turma

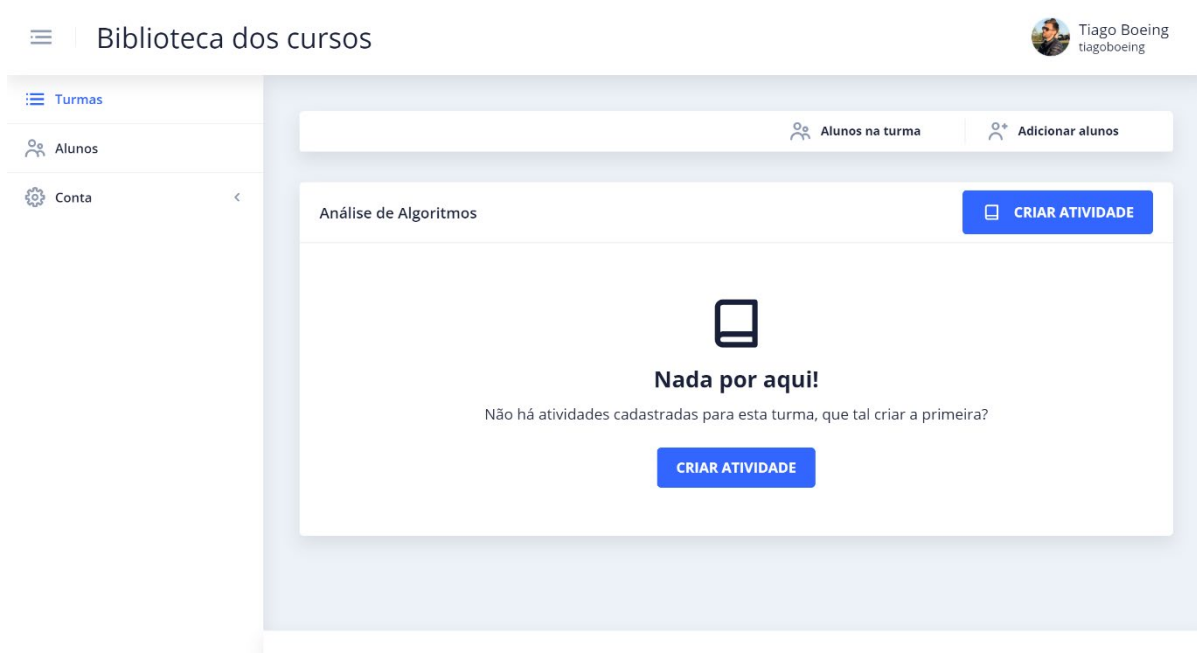


Fonte: do próprio autor (2020)

11.3. GERENCIAMENTO DE ATIVIDADES

As atividades são elementos compostos em turmas, sendo assim obrigatoriamente a atividade deve ser criada a partir dos detalhes de uma turma. Basta acessar a lista de turmas e na sequência clicar sobre um registro da listagem. Será exibido então o botão “Criar atividade”.

Figura 21 Criando atividade em uma turma



Fonte: do próprio autor (2020)

Na **Figura 21 Criando atividade em uma turma** estamos acessando a turma “Análise de Algoritmos” e verificando que até o momento não há atividades cadastradas.

11.3.1. CRIAR ATIVIDADE

Para criação de uma atividade estão disponíveis os seguintes campos:

- Nome da atividade (obrigatório);
- Descrição (opcional);

- Data de sincronização (obrigatório): indica a data e hora em que a aplicação deverá buscar os repositórios dos alunos no GitHub e iniciar a sincronização para que os materiais fiquem disponíveis ao professor;
- Repositório a ser criado (obrigatório): indica o nome do repositório que os alunos da turma devem criar em suas respectivas contas no GitHub. O repositório deve estar disponibilizado publicamente. Ao preencher o campo “Nome da atividade” será sugerido um nome de repositório de forma automática, porém o professor pode alterar para qualquer valor desejado.

Figura 22 Cadastro de uma atividade

Biblioteca dos cursos

Turmas

Alunos

Conta

Nova atividade

Nome da atividade

Trabalho 1

Descrição

Nesse pull request, o módulo de r

P

Data de sincronização

Sáb 27/06/2020 20:00

Defina uma data de sincronização dos materiais ou deixe em branco.

Repositório a ser criado

trabalho-1

O sistema buscará em cada perfil de aluno por este nome de repositório durante a sincronização.

SALVAR ATIVIDADE

Jun 6, 2020 Today

JUN 2020

| Su | Mo | Tu | We | Th | Fr | Sa |
|----|----|----|----|----|----|----|
| 31 | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 1 | 2 | 3 | 4 |

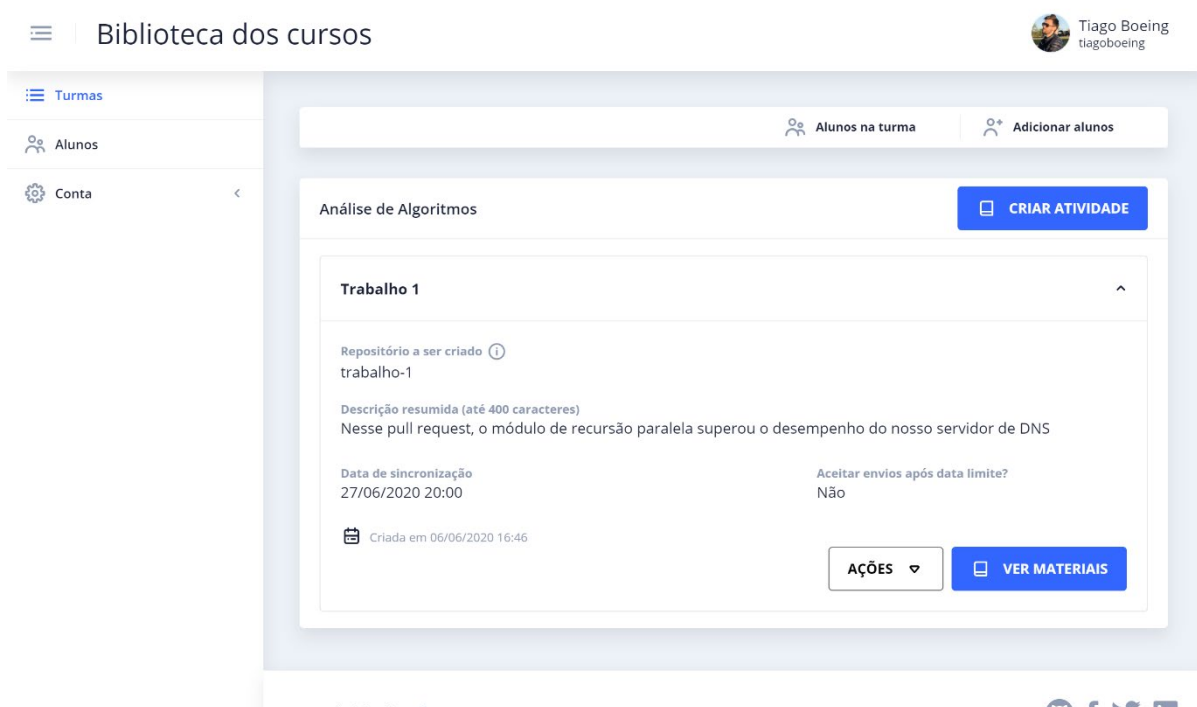
16 WORDS POWERED BY TINY

nosso servidor de DNS

tiago boeing tiagoboeing

Fonte: do próprio autor (2020)

Figura 23 Lista de atividades de uma turma

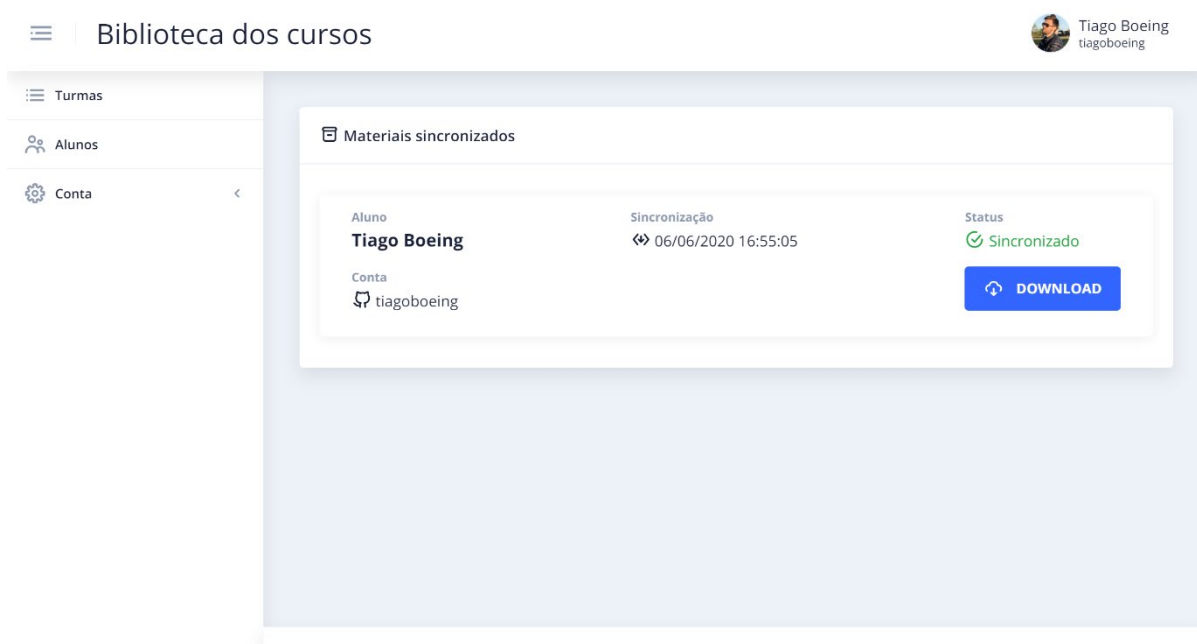


Fonte: do próprio autor (2020)

11.4. LISTAGEM DE MATERIAIS

Assim como as atividades são visualizadas a partir de uma turma, para visualizar a lista de materiais é necessário estar em uma atividade. Dentro da turma em que estiver trabalhando o professor selecionará uma atividade e clicará em “Ver materiais”. Todos os materiais sincronizados serão exibidos com o botão *download*. Ao clicar em *download* será disponibilizado o arquivo compactado (.zip) com o conteúdo do repositório na data da sincronização.

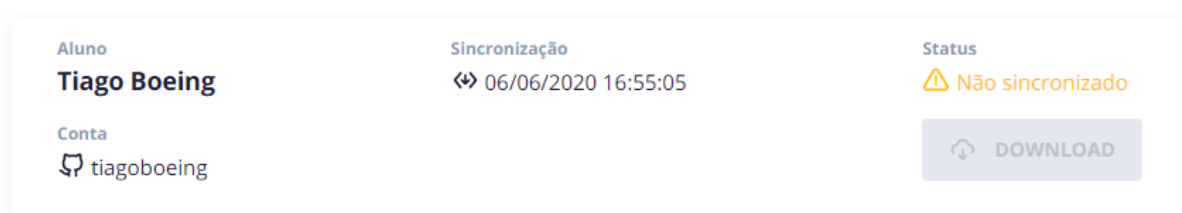
Figura 24 Lista de materiais de uma atividade



Fonte: do próprio autor (2020)

Quando a aplicação acessar o GitHub de um determinado aluno e o repositório não existir — solicitado pelo professor (preenchido na etapa de cadastro da atividade) — o item será cadastrado em materiais, porém não será possível realizar *download*, o botão ficará bloqueado. Ao passar o mouse sobre ele indicará que não foi possível sincronizar o material do aluno em questão.

Figura 25 Cenário em que o repositório não existe na conta do aluno



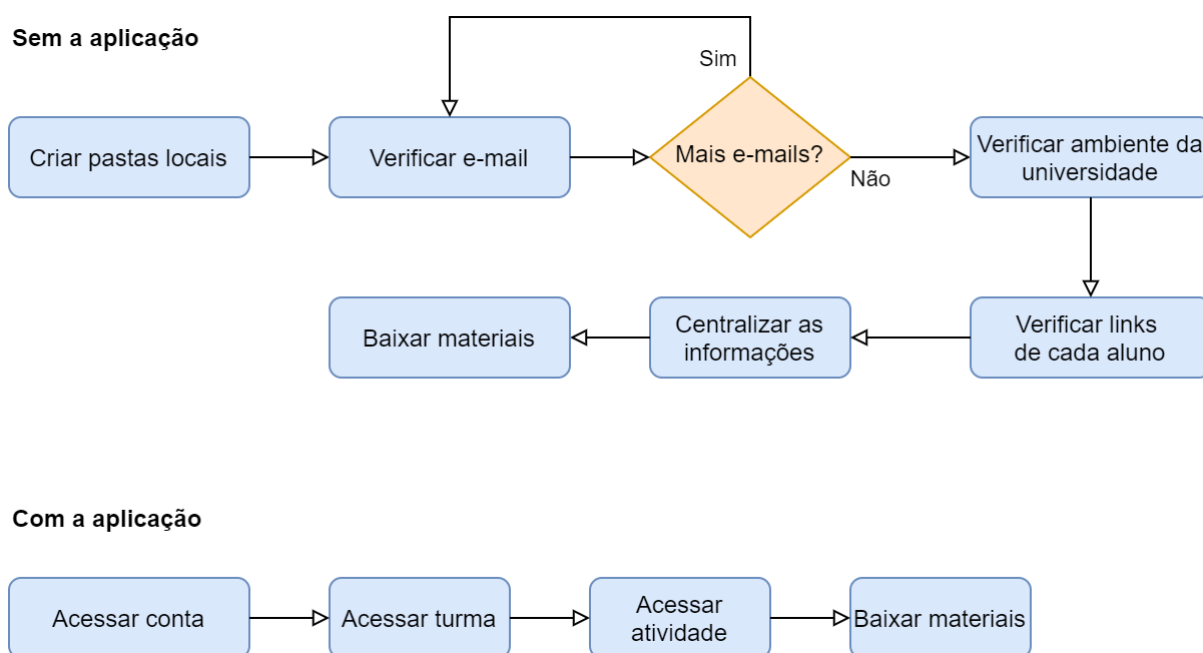
Fonte: do próprio autor (2020)

Caso o botão *download* estiver ativo basta clicar para baixar o arquivo com o conteúdo do repositório.

11.5. COMPARATIVO DE FLUXO DE TRABALHO

A Figura 26 Comparação de fluxo de trabalho ilustra um comparativo do fluxo de trabalho desempenhado na recepção de atividade avaliativas práticas sem a utilização da ferramenta, e com a utilização.

Figura 26 Comparação de fluxo de trabalho



Fonte: do próprio autor (2020)

12. CONSIDERAÇÕES FINAIS

Com a aplicação desenvolvida neste trabalho é possível citar implementações futuras com a finalidade de torna-la integrável (para obtenção de código) com outros sites que fornecem hospedagem de código-fonte através do protocolo Git, bem como novas formas de autenticação, visto que até o momento o GitHub é o único provedor de acesso externo — sendo a outra forma interna através da criação de uma conta manualmente.

Quanto a sincronização dos materiais há observações relacionadas a melhorias no tratamento de exceções e apresentação detalhada do status, a exemplo de um repositório não ter sido obtido devido a um erro de integração ou simplesmente não existir na conta do aluno, atualmente a aplicação tratará ambos os cenários da mesma forma. A possibilidade de cadastrar uma atividade e informar que ela é capaz de receber materiais sob demanda é outro ponto observado durante o desenvolvimento, para que dessa forma à medida que a aplicação detecte alterações no repositório do aluno realize a sincronização, sem que seja necessário prefixar uma data.

Outro ponto interessante e alavancado como implementação futura devido ao escopo e tempo limitado para o artigo é de possibilitar a visualização do conteúdo do repositório sincronizado, comentários em uma determinada linha de código, pontuação através da aplicação e gerenciamento (criação) dos repositórios de forma automática na conta de cada aluno, sendo a última melhoria fortemente dependente de uma configuração diferenciada no GitHub, abandonando o *OAuth Apps* e passando a utilizar *GitHub Apps*.

13. CONCLUSÃO

Com a finalização do desenvolvimento da ferramenta conclui-se que ela é altamente escalável sendo seguidos os padrões de desenvolvimento *cloud* e fornece funcionalidades para gerenciar alunos, turmas e criar atividades, sendo estas de extrema importância e que auxiliam o professor a gerir melhor seu fluxo de trabalho. As tecnologias quando voltadas ao meio acadêmico — sendo que este muitas das vezes carece de ferramentas específicas — criam um leque de novas possibilidades ao professor, podendo diversificar ainda mais sua metodologia e os recursos que possui à disposição na aplicação das atividades avaliativas.

Foi realizado um teste de sincronização de uma turma contendo 10 alunos e no cenário onde todos possuem o repositório criado em suas respectivas contas (considerando que todos enviaram a atividade). O tempo necessário para a sincronização da atividade completa (materiais de todos os alunos) foi de 2 segundos, sendo esta a latência obtida no momento do *download* do repositório no GitHub e cadastro da atividade na base de dados.

Por fim, para o futuro ficam diversos *gaps* de possíveis melhorias para evolução constante da aplicação, bem como sua disponibilização na internet e a reflexão da importância de o sistema educacional preparar os alunos através da utilização de tecnologias amplamente difundidas atualmente — não só no mercado de trabalho — e que elas sejam incluídas no meio com a maior frequência possível. Oferecer formas práticas, lúdicas e dinâmicas de explicar um conteúdo através da tecnologia é de extrema importância, fazendo com que o aluno se sinta envolvido e disponha de autonomia para cumprir com os objetivos atribuídos. Estas ferramentas quando voltadas para o ambiente educacional, especificamente na entrega de atividades avaliativas o familiarizam quanto a utilização prática e permitem a automatização dos processos relacionados centralizando materiais de diversos alunos e turmas para o professor.

14. REFERÊNCIAS BIBLIOGRÁFICAS

DUPRÉ, L.; HAEBERLEN, T. **Cloud Computing: Benefits, risks and recommendations for information security**. European Network and Information Security Agency. [S.l.], p. 50. 2012.

FORTUNE BUSINESS INSIGHTS. Smart Education and Learning Market Size, Share & Industry Analysis, By Component (Hardware, Software, and Services), By Learning Mode. **Fortune Business Insights**, 2018. Disponível em: <<https://www.fortunebusinessinsights.com/industry-reports/smart-education-and-learning-market-101942>>. Acesso em: 03 jun. 2020.

G1. Procura por curso de ensino superior na modalidade 'tecnológica' dobra em 10 anos. **G1**, 2018. Disponível em: <<https://g1.globo.com/educacao/guia-de-carreiras/noticia/2018/12/12/procura-por-curso-de-ensino-superior-na-modalidade-tecnologica-dobra-em-10-anos.ghtml>>. Acesso em: 12 out. 2019.

GASPAR, M. L. F. **O processo de avaliação da aprendizagem escolar na prática pedagógica**. [S.l.], p. 13.

INTERNATIONAL MONETARY FUND. **Seeking Sustainable Growth: Short-Term Recovery, Long-Term Challenges**. Washington, p. 304. 2017.

JADEJA, Y.; MODI, K. **Cloud computing - concepts, architecture and challenges**. U. V. Patel College of Engineering, Ganpat University. Kherva (Mahesana), p. 5. 2012. International Conference on Computing, Electronics and Electrical Technologies [ICCEET].

KADIVAR, N. Top 10 Version Control Systems. **Hackernoon**, 2018. Disponível em: <<https://hackernoon.com/top-10-version-control-systems-4d314cf7adea>>. Acesso em: 06 jun. 2020.

KENSKI, V. M. Learning mediated by the technology. **Revista Diálogo Educacional**, Curitiba, v. 4, n. 10, p. 10, dez. 2003.

LÉVY, P. **As Tecnologias da Inteligência: o Futuro do Pensamento na era da Informática**. 1. ed. [S.l.]: Instituto Piaget, 1994. 263 p. Acesso em: 10 dez. 2019.

LI, C.; LALANI, F. The COVID-19 pandemic has changed education forever. This is how. **World Economic Forum**, 2020. Disponível em: <<https://www.weforum.org/agenda/2020/04/coronavirus-education-global-covid19-online-digital-learning/>>. Acesso em: 03 jun. 2020.

MARTINES, R. D. S. et al. **O uso das TICs como recurso pedagógico em sala de aula**. Congresso Internacional de Educação e Tecnologias. [S.l.], p. 12. 2018.

PAULA, H. D. F. E.; MOREIRA, A. F. Atividade, ação mediada e avaliação escolar. **Educação em Revista**, Belo Horizonte, v. 30, n. 1, p. 17-36, mar. 2014. Disponível em: <<http://dx.doi.org/10.1590/s0102-46982014000100002>>. Acesso em: 10 dez. 2019.

RODRIK, D. **Will new technology in developing countries be a help or a hindrance?** World Economic Forum. [S.I.]. 2018.

THE OPEN GROUP. **Characteristics of Architecture Principles**. San Francisco. 1999.

UBM. Mercado de trabalho mostra alta procura nas áreas de Tecnologia, Gestão e Relacionamento. **G1**, 2019. Disponível em: <<https://g1.globo.com/rj/sul-do-rio-costa-verde/especial-publicitario/ubm/conhecimento-transforma/noticia/2019/10/11/mercado-de-trabalho-mostra-alta-procura-nas-areas-de-tecnologia-gestao-e-relacionamento.ghtml>>. Acesso em: 20 jul. 2020.

YOUNG DIGITAL PLANET. **The Book of Trends in Education 2.0**. Tradução de Danielle Mendes Sales. Edição Brasileira. ed. São Paulo: Editora Moderna, 2015. 355 p. Disponível em: <<https://smartlab.me/wp-content/uploads/2017/05/Educac%CC%A7a%CC%83o-no-se%CC%81culo-21-download-1.pdf>>. Acesso em: 06 jun. 2020.