

RECONSTRUINDO UM SOFTWARE DE AUXÍLIO AO PROGRAMA DE BOLSAS DE ESTUDOS

PEDRO HENRIQUE GOMES BARBOSA¹, RODRIGO MAIA BARRETO²,
CRISTIANE YAE MI IMAMURA³

¹ Graduando em Engenharia de Software, Centro Universitário das Faculdades Associadas de Ensino - UNIFAE, Campus São João da Boa Vista, pedro-gomes17@hotmail.com

² Graduando em Engenharia de Software, Centro Universitário das Faculdades Associadas de Ensino - UNIFAE, Campus São João da Boa Vista, rodrigobarretto@gmail.com

³ Mestre em Ciências pela USP - São Carlos. Professora dos cursos de Engenharia de Computação e Engenharia de Software, Centro Universitário das Faculdades Associadas de Ensino - UNIFAE, Campus São João da Boa Vista, cristiane@fae.br

Área de conhecimento (Tabela CNPq): 1.03.03.02-2 Engenharia de Software

RESUMO: Este trabalho tem por objetivo descrever o processo de evolução do software, de uma instituição de ensino superior, que permite aos alunos realizar a solicitação de bolsas de estudo e, para a instituição, oferece uma organização dos documentos recebidos, bem como o cálculo dos índices a serem analisados. A reengenharia realizada sobre o software original, chamado de Produto Mínimo Viável – ou *Minimum Viable Product* (MVP), foi conduzida para construir uma nova versão com custos mais baixos para manutenção e com melhor layout tanto para as funções a ser desempenhadas pelos alunos quanto para aquelas que seriam realizadas pelos administradores do sistema. Na reconstrução, foi utilizada a metodologia Scrum. Após testes de caixa preta e beta realizados, a nova versão proposta foi implantada e está em uso para a solicitação de bolsas a serem concedidas no segundo semestre em 2020.

PALAVRAS-CHAVE: reengenharia de software; scrum; node.js; react;

REBUILDING A SOFTWARE TO SUPPORT THE SCHOLARSHIP ASSIGNMENT PROCESS

ABSTRACT: The aim of the present work is to describe the improvement of software that was built in an institution of higher learning to allow the student to request scholarships. The original software, called Minimum Viable Product - or Minimum Viable Product (MVP), organizes the documents received, as well as offers the calculation of the indices to be analyzed. The software reengineering on MVP enables to build a new version with lower maintenance costs and with a better layout. The Scrum methodology was used for the software rebuilding. The proposed new version was implemented and it is in use for the application for scholarships to be awarded in the second half of 2020.

KEYWORDS: software reengineering; scrum; node.js; react;

INTRODUÇÃO

Apesar do crescimento das matrículas nos últimos dez anos no Brasil, frequentam curso superior atualmente apenas 25,2% dos jovens brasileiros na faixa etária entre 18 e 24 anos. Em 2018, o percentual de estudantes que frequentavam instituições públicas no ensino superior era de 25,8%, logo, no ensino superior de graduação a rede privada atendeu a maioria dos estudantes (IBGE, 2019).

As ações de assistência estudantil, na instituição de ensino em questão, possui uma longa tradição na adoção de políticas de inclusão de jovens, com maior necessidade financeira, em um programa de bolsas de estudos. Esse programa tem por finalidade garantir o acesso à educação superior através da concessão de bolsas de estudo para estudantes de baixa renda.

Portanto, para sanar os problemas enfrentados pela instituição no período da solicitação das bolsas foi criado inicialmente em 2019 um software, o Produto Mínimo Viável – ou *Minimum Viable Product* (MVP). Antes disso, o processo acontecia de forma mecânica com filas de alunos e inúmeros documentos sendo arquivados em espaços físicos.

O MVP, durante 6 meses, cumpria seu objetivo de anexar os arquivos exigidos e, com base nos dados fornecidos pelos alunos no momento de seu cadastro, realizar o cálculo da porcentagem de bolsa concedida de forma autônoma. Entretanto, os alunos possuíam dificuldades quanto ao preenchimento das informações solicitadas, e a implementação do MVP onerava a sua manutenção, por usar uma plataforma paga. Desta forma, este trabalho tem por objetivo descrever o processo de reengenharia de software necessário para fornecer uma nova versão da aplicação que sanou os problemas encontrados e que fizeram com que o MVP fosse substituído já em 2020 por uma atualização.

MATERIAL E MÉTODOS

O processo de reengenharia de software é um trabalho de reforma, que pode envolver atividades como análise de inventário, reestruturação dos documentos, engenharia reversa, reestruturação do código, reestruturação dos dados e engenharia direta (PRESSMAN, 2011). Nesse trabalho, os autores possuíam toda a documentação do MVP, como o documento dos requisitos, projeto, descrição da arquitetura, código fonte, portanto não foi necessário aplicar uma ferramenta para engenharia reversa. A reengenharia de software ocorreu mais na aplicação da engenharia direta, usando as informações para alterar e reconstruir o sistema existente para melhorar sua qualidade geral.

O novo software desenvolvido a partir do MVP precisava garantir as mesmas funcionalidades, ou seja, os mesmos requisitos funcionais. Segundo Sommerville (2007), requisitos funcionais são descrições de serviços oferecidos pelo sistema e suas restrições operacionais, sendo que esses requisitos refletem necessidades de clientes de um sistema.

Assim, foram considerados todos os requisitos levantados para a construção do MVP, que na época foram obtidos através dos levantamentos por meio de entrevistas, que é uma técnica para elicitación de requisitos e, segundo Marconi e Lakatos (2002), é um encontro entre duas pessoas para que uma delas obtenha informações a respeito de um assunto. Na Tabela 1 podem ser encontrados os requisitos do MVP que também são os requisitos da nova versão proposta na reengenharia.

Tabela 1 – Requisitos necessários para o desenvolvimento do software.

Requisitos Gerais	<ul style="list-style-type: none"> • Cadastro de usuários com tipos diferentes de permissões; • Autenticação dos usuários para acesso a diferentes funcionalidades; • Calcular e exibir os índices para auxiliar o serviço social a definir o percentual de bolsa a ser concedido; • Alertar ao aluno possíveis inconsistências nas informações do cadastro; • Gerar relatório a ser exportado em planilha contendo o nome do aluno, RA, curso, semestre cursando, percentual de bolsa, valor do desconto concedido;
Requisitos dos administradores	<ul style="list-style-type: none"> • O sistema deve permitir que o administrador cadastre, inative, altere ou visualize os dados da entidade, dos usuários, dos cursos, dos períodos letivos de cada curso, do valor da mensalidade, dos arquivos anexados pelos alunos e os índices calculados, possibilitando também que o administrador solicite mais informações aos alunos.
Requisitos dos alunos	<ul style="list-style-type: none"> • O sistema deve conceder que o mesmo crie seu cadastro com login e senha para acessá-lo; • Cadastro, alteração ou visualização das informações solicitadas pela entidade para o pleito da bolsa de estudos, permitindo que os alunos anexem arquivos dos documentos solicitados.

Fonte: Elaborado pelos Autores.

Alem dos requisitos, foi aproveitado o modelo do banco de dados do MVP e sobre algumas das 21 tabelas modeladas foram realizadas pequenas alterações fundamentais necessárias, como a ajustes nos relacionamentos. Esse modelo relacional continua sendo gerenciado pelo SGBD MySQL. Essas

tabelas são utilizadas para armazenar e relacionar todos os dados dos alunos para possibilitar o cálculo da porcentagem da bolsa a partir do ano de 2020, não sendo importado nenhum dado presente no banco de dados original de 2019.

Como mencionado, o MVP ficou em uso e a partir da experiência dos usuários foi possível definir as mudanças e melhorias a serem realizadas na reconstrução do sistema, algumas delas em função do layout não deixar claro que informações eram necessárias e como deveriam ser preenchidas.

Para a instituição de ensino, soluções de software que possibilitem o menor custo são mais interessantes. O MVP foi concebido utilizando um framework SCRIPTCASE (2020), que é uma plataforma paga de desenvolvimento rápido para aplicações web, que utiliza a linguagem PHP (um acrônimo recursivo para PHP: *Hypertext Preprocessor*) como a linguagem de programação base. Este programa cria o front-end e quase a totalidade do back-end, sendo o restante concluído com pouco esforço, mas obrigaria a instituição a pagar pela plataforma para gerar novas versões.

Assim, para a nova versão do software foi definido o uso da linguagem de programação JavaScript que é uma das linguagens mais populares no mundo (TIOBE, 2020), e é uma linguagem interpretada e desenvolvida para que scripts possam ser executados no servidor e no lado usuário, não sendo necessário toda interação e processamento ser realizado no servidor (ROSA; DIAS, 2018).

Para desenvolvimento da lógica de negócio do sistema foi utilizada no back-end a plataforma de aplicação da linguagem JavaScript denominada Node.js, que é construída sobre o motor JavaScript do Chrome (V8), que tornou viável criar sistemas web complexos, utilizando a mesma linguagem em todos os ângulos do desenvolvimento, tais como: back-end, front-end, pré processadores e automatizadores de tarefas.

No front-end foi decidido usar o React JavaScript, que é “uma biblioteca de código aberto para criar interfaces de usuário. É mantido pelo Facebook, Instagram e uma comunidade de desenvolvedores individuais e outras empresas”. (ROSA; DIAS, 2018).

Para definição do novo layout foi realizada a prototipação das telas para auxiliar no desenvolvimento do front-end, com a ferramenta denominada FIGMA (2020), onde é possível desenhar e criar interações ou navegações a partir de botões e outros componentes que auxiliam e facilitam o desenvolvimento do programa.

No armazenamento dos arquivos foram disponibilizadas duas estratégias. Uma delas contou com uso do próprio disco da máquina e outra com o serviço Amazon s3, para fornecer uma alternativa para evitar o consumo total do espaço disponível do servidor local da instituição que hospeda a aplicação. O Amazon s3 é uma solução de serviço de armazenamento em nuvem (AWS, 2020), que pode garantir que sejam escalados os recursos necessários.

Em oposição ao MVP que se utilizou o modelo cascata para o desenvolvimento, para a reconstrução do software foram usadas algumas técnicas e práticas do Scrum, que é um modelo ágil voltado para a gestão de projetos que surgiu em uma indústria automobilística, e que pode ser adaptado para diferentes áreas na produção de softwares (WAZLAWICK, 2013). No *Scrum*, os projetos são divididos em ciclos denominados *Sprints*, e nestes ciclos a execução de sua funcionalidade é realizada em poucas semanas. Para esse projeto usou-se Planning Poker, para definição das Sprints e para acertar a velocidade do time e acompanhar os resultados foram executadas reuniões de Sprint Review.

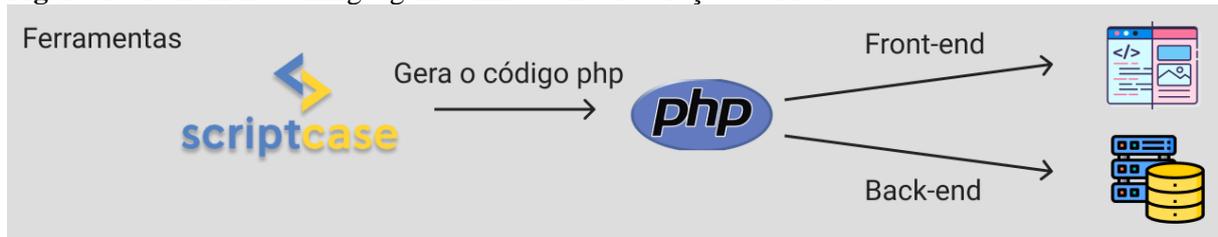
Para avaliar o software produzido na reengenharia foram empregados teste de caixa preta e teste beta. O teste de caixa preta foca nos requisitos funcionais do sistema e, executados posteriormente no processo de teste, avaliam se os resultados esperados pelos desenvolvedores são obtidos na execução do sistema frente a um conjunto de entrada dada. O sistema neste caso não considera o desempenho do componente interno e se porta como uma caixa preta (PRESSMAN, 2011).

De acordo com Sommerville (2007), o teste beta consiste em disponibilizar uma versão do software aos usuários finais com o intuito de que eles possam avaliar o software em um ambiente no qual não estejam presentes os desenvolvedores, e após isso apontar todos os problemas descobertos.

RESULTADOS E DISCUSSÕES

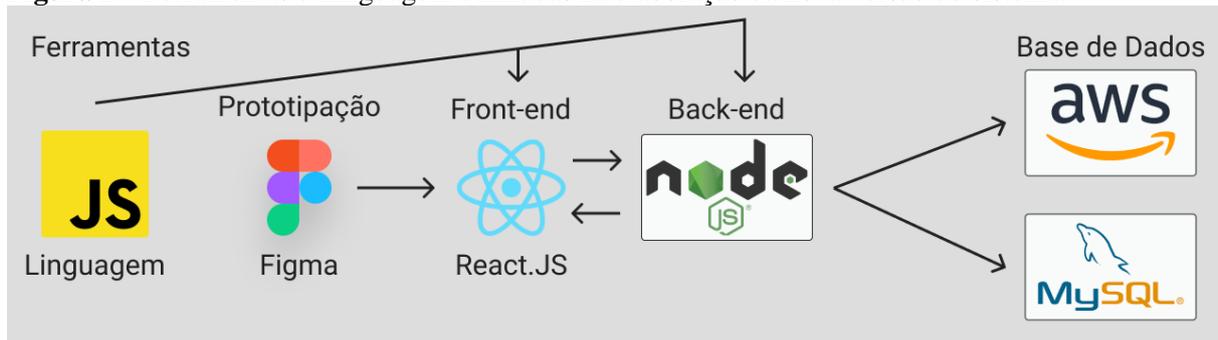
Para evidenciar as diferenças resultantes do MVP e da nova versão proposta, foram colocadas na Figura 1 as tecnologias e linguagens que se encontram no MVP, mas não na reconstrução. Na Figura 2, podem ser vistas as ferramentas e linguagens usadas na nova versão do software que também não estão presentes no MVP.

Figura 1 – Ferramenta e Linguagem utilizadas na elaboração do MVP.



Fonte: Elaborado pelos Autores.

Figura 2 – Ferramentas e Linguagem utilizadas na elaboração da nova versão do sistema.



Fonte: Elaborado pelos Autores.

Para reconstrução do software, foi primeiramente desenvolvido o back-end e para ele foram definidas nove Sprints, sendo que cada uma teve como prazo total de execução de uma semana. E o time se mostrou com a velocidade correta para o desenvolvimento das mesmas, sem ter tido a necessidade de ajustes. Na Tabela 2, podem ser vistos os detalhes da metodologia Scrum empregada à fase de desenvolvimento do back-end.

Tabela 2 – Detalhes do *Scrum* empregado o back-end

Equipe	Quantidade	Sprints	Tempo da Sprint	Tempo Total
Scrum Master	01			
Product Owner	01	09	01 semana	9 semanas
Development Team	02			

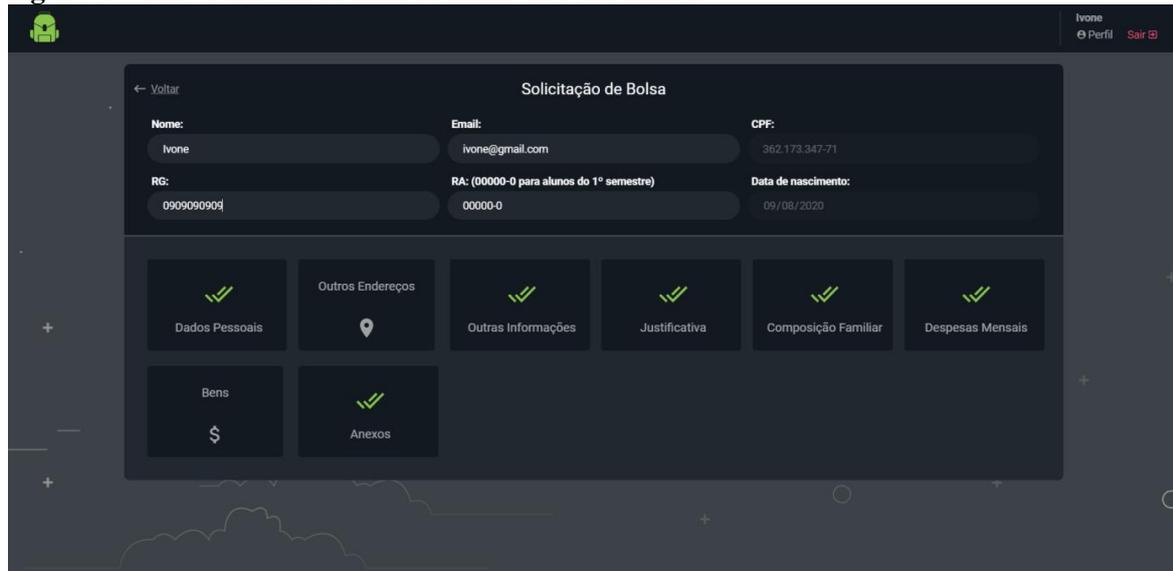
Fonte: Elaborado pelos Autores.

Uma vez que todas as Sprints tiveram sucesso, a fase de construção do front-end iniciou e seguiu a mesma ordem das sprints definidas para o back-end, ou seja, as telas foram construídas para promover as tarefas determinadas no back-end, para que fosse possível realizar o teste beta e o teste de caixa preta.

A versão de release para teste beta só foi liberada, uma vez que os testes de caixa preta não falhassem. Para o teste beta, participaram três usuários finais que são administradores do sistema e quatro alunos que utilizaram a versão do MVP no ano de 2019. Através do teste beta foi possível identificar e corrigir erros de layout e funcionalidades.

Na Figura 3 é demonstrado o resultado do desenvolvimento do front-end que teve como referência o protótipo, sendo este com uma interface simples, intuitiva e responsiva para ser acessada via dispositivo móvel ou por navegadores web, sendo então de fácil acesso e interação na solicitação da bolsa. O mesmo protótipo também serviu para a desenvolvimento do resultado final do front-end dos administradores, sendo que este possui acesso apenas através do desktop, uma vez que seria impraticável tornar responsivo, com uso do celular, o acesso à tabela de informações que o administrador analisa, por conter inúmeras informações.

Figura 3 – Tela do resultado final do Front-end dos alunos.



Fonte: Elaborado pelos Autores.

CONSIDERAÇÕES FINAIS

Ter um MVP e toda a sua documentação do software a ser desenvolvido possibilitou realizar a reengenharia com segurança, e analisar se a nova versão provinha as funcionalidades corretamente já implementadas na versão anterior. Portanto, os testes de caixa preta e beta puderam ocorrer de forma eficiente, comparando-se o conhecimento que o MVP havia proporcionado.

O uso de uma metodologia ágil, como Scrum, foi bastante interessante para promover as entregas para o usuário final administrador de forma antecipada no desenvolvimento da nova versão, ao contrário do que ocorreu no MVP com uso do Cascata.

A nova versão do software implementada como evolução do MVP está em uso pela instituição desde setembro deste ano. Deseja-se futuramente obter o feedback dos alunos que fizeram uso dela para requisitar bolsas nesse período, para avaliar a usabilidade do mesmo.

REFERÊNCIAS

AWS. Amazon S3-Armacenamento de objetos construído para armazenar e recuperar qualquer volume de dados de qualquer local. 2020. Disponível em <https://aws.amazon.com/pt/s3/?c=s&sec=srv>, acesso em 06 novembro 2020.

FIGMA. 2020. Disponível em: <https://www.figma.com/>, acesso em 3 maio 2020.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). *Síntese de Indicadores Sociais: uma análise das condições de vida da população brasileira 2019*. Rio de Janeiro, 2019. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv101678.pdf>, acesso em: 07 maio 2020.

MARCONI, M.A.; LAKATOS, E.M., *Técnicas de pesquisa*. Editora Atlas S.A. ed.5, 2002.

PRESSMAN, Roger. *Engenharia de Software: Uma abordagem profissional*. 7. ed. Porto Alegre: McGraw-Hill, 2011.

ROSA, M. M.; DIAS, I. A. *Desenvolvendo software com react, javascript, redux e mongodb: o caso da aplicação fast fix*. 4ª Jornada de integração e iniciação científica, v. 3, n. 1. Florianópolis, SC, junho de 2018.

SCRIPTCASE. 2020. Disponível em: <https://www.scriptcase.com.br/>, acesso em 03 maio 2020.

SOMMERVILLE, Ian. *Engenharia de Software*. 8ª ed. São Paulo: Pearson Addison-Wesley, 2007.

TIOBE. **TIOBE Index for November 2020 November Headline: Python is unstoppable and surpasses Java**. 2020. Disponível em: <https://www.tiobe.com/tiobe-index/> acesso em 06 novembro 2020.

WAZLAWICK, Raul Sidnei. *Engenharia de Software: Conceitos e práticas*. [S.l.]: Elsevier Editora Ltda, 2013. 360 p.