

Abordagem Socialmente Responsável ao Desenvolvimento de Software: um estudo exploratório

Michel Saucedo Marques, Amanda Meincke Melo

Universidade Federal do Pampa (UNIPAMPA) – Campus Alegrete
Avenida Tiarajú, 810 – 97.546-550 – Alegrete – RS – Brasil
{michelmarques.aluno, amandamelo}@unipampa.edu.br

Abstract. *With software pervasive in people's daily lives, it is essential to consider ethical aspects in its development. However, few studies were identified that, when addressing ethical aspects, explained quality, processes, methods and tools. In addition, it is essential to highlight the importance of people in ethical decision-making. This work therefore presents the application of a socially responsible approach to software development. By applying it to high school students, it generated software with social responsibility.*

Resumo. *Com o software pervasivo no cotidiano das pessoas, é indispensável a consideração de aspectos éticos em seu desenvolvimento. Contudo, poucos foram os trabalhos identificados que, ao abordarem aspectos éticos, explicitaram qualidade, processos, métodos e ferramentas. Além disso, é fundamental destacar a importância das pessoas nas tomadas de decisões éticas. Este trabalho apresenta, portanto, a aplicação de uma abordagem socialmente responsável ao desenvolvimento de software. Ao aplicá-la junto a estudantes de ensino médio, gerou um software com responsabilidade social.*

1. Introdução

Uma grande variedade de tarefas, das mais simples às mais complexas, é apoiada por software. Engenheiros de software, portanto, têm muitas oportunidades de fazer o bem ou causar danos. De acordo com o Código de Ética e Prática Profissional de Engenharia de Software ACM/IEEE 2015, o profissional de Engenharia de Software deve agir de forma ética, ou seja, deve colocar o bem público em primeiro lugar. Para isso, são necessárias práticas que enfatizem princípios éticos e de responsabilidade social na produção de aplicações.

Responsabilidade social pode ser entendida como a busca por agir corretamente ou de forma a contribuir ao desenvolvimento social [Ribeiro 2002]. Enquanto para algumas pessoas possa representar a ideia de responsabilidade ou obrigação legal, para outras indica um comportamento no sentido ético [Benedicto *et al.* 2008].

Em uma revisão de literatura em bases como ACM Digital Library, IEEE Xplore e Scopus, identificaram-se poucos estudos que explicitam o desenvolvimento socialmente responsável de software contemplando todas as camadas de Engenharia de Software mencionadas por Pressman e Maxim (2016): Foco na Qualidade, Processos, Métodos e Ferramentas. Ademais, ao analisar o papel das pessoas nesses trabalhos e considerar que a responsabilidade ética é uma atribuição de pessoas, essa responsabilidade precisa ser evidenciada em processos, métodos e ferramentas de Engenharia de Software, promovendo a qualidade do software que atenda às suas necessidades e tragam impacto positivo em suas vidas [Marques 2022].

Este trabalho apresenta a aplicação de uma abordagem ao desenvolvimento socialmente responsável de software em um estudo exploratório. O texto está organizado como segue: a Seção 2 discute trabalhos relacionados; a Seção 3 apresenta a metodologia adotada; a Seção 4 sumariza os resultados; a Seção 5 realiza algumas considerações finais.

2. Trabalhos Relacionados

A fim de encontrar contribuições ao desenvolvimento socialmente responsável de software, realizou-se uma adaptação das diretrizes para revisões sistemáticas propostas por Kitchenham (2004), com a realização das seguintes atividades: Definição da questão de pesquisa, pesquisa em bases de busca, definição dos critérios de inclusão e exclusão, e, por fim, a leitura completa dos artigos e extração de dados [Marques 2022].

A questão de pesquisa formulada foi: *Como a Engenharia de Software tem abordado o desenvolvimento de software de modo socialmente responsável?*

Para realizar a busca, selecionaram-se bases que indexam trabalhos na área de Computação, como: ACM Digital Library, IEEE Xplore e Scopus. A seguinte *string* foi adaptada para as diferentes bases: ("Software Engineering" OR "Software Development") AND ("Socially Responsible" OR "Socially Conscious" OR "Ethical" OR "Ethic") e aplicada no dia 10 de dezembro de 2021. Observa-se que, na base ACM Digital Library, restringiu-se a busca ao título, resumo e palavras-chave.

Ao todo, identificaram-se 132 trabalhos: 25 na ACM Digital Library, 100 na IEEE Xplore e 7 na Scopus. Esses tiveram seus títulos, palavras-chave e resumos lidos aplicando os critérios de inclusão e exclusão definidos [Marques 2022]. Então, selecionaram-se 36 trabalhos, sendo 7 da ACM Digital Library e 29 da IEEE Xplore. Estes foram, então, lidos em sua totalidade, resultando em 21 trabalhos que abordam qualidade de software, processo, método ou ferramentas de Engenharia de Software.

Nesses trabalhos, nota-se diversidade quanto aos aspectos de qualidade que abordam. Há um alto índice de atributos de qualidade de produto, com amplo destaque para confiabilidade, que é mencionada em mais da metade dos estudos. Já a privacidade se configura nos trabalhos que abordam a utilização de dados do usuário e têm forte preocupação com as novas leis de gestão de dados. Segurança, no sentido de *safety*, está presente nos casos de desenvolvimento de sistemas autônomos, onde a preocupação com a vida deve ser priorizada. Tem-se presente também requisitos éticos e legais no trabalho de Mustaffa *et al.* (2020), que relaciona requisitos éticos com a questão religiosa.

É possível notar, na camada de processos, que a Engenharia de Requisitos e a Abordagem Ágil estão relacionadas ao desenvolvimento socialmente responsável de software, visto que nos trabalhos identificados são os mais mencionados, enquanto que, no que tange aos métodos, tem-se uma diversidade de opções que se devem ao fato de que estão presentes em todas as fases do ciclo de desenvolvimento. Com relação às ferramentas, o *checklist* se destaca como forma de avaliação com base em listas e requisitos legais. Nessa perspectiva, tem-se o trabalho de Borg *et al.* (2021), que faz uma análise crítica da lista de avaliação de inteligência artificial confiável, proposta pela União Europeia, e de Vakkuri *et al.* (2020), que propõem a utilização de anotações em cartões para cada decisão ética tomada durante o processo de desenvolvimento relacionando essa decisão com um requisito.

Apenas os trabalhos de Dekhtyar *et al.* (2020) e Shneiderman (2020) abordam de

forma explícita todas as camadas da Engenharia de Software. Enquanto o trabalho de Dekhtyar *et al.* (2020) visa o desenvolvimento de software socialmente responsável e o define como um software que tem como objetivo ajudar a comunidade local, Shneiderman (2020) propõe uma forma de gestão de sistemas de inteligência artificial visando alinhar princípios éticos com as práticas de desenvolvimento.

Aspectos éticos devem ser considerados em todas as camadas da Engenharia de Software, sendo relevante abordá-los de, modo intencional, no desenvolvimento de software. Além disso, deve-se considerar que a responsabilidade ética no desenvolvimento de software é uma atribuição das pessoas. Portanto, processos, métodos e ferramentas devem fomentar e instigar as pessoas participantes do processo de desenvolvimento a pensar de maneira ética a cada tomada de decisão e, assim, promover a qualidade do software que atenda às suas necessidades e tragam impacto positivo em suas vidas [Marques 2022].

Desse modo, pode-se caracterizar o desenvolvimento socialmente responsável de software como o desenvolvimento de software onde os aspectos éticos devem ser considerados por suas partes interessadas, na definição de seus atributos de qualidade, assim como em seus processos, métodos e ferramentas, de forma que o software produzido atenda as necessidades implícitas e explícitas dos usuários, de forma ética e responsável [Marques 2022].

3. Metodologia

Neste estudo, adotou-se uma abordagem qualitativa de pesquisa [Terence e Filho 2006], de caráter exploratório, utilizando de observação, da interpretação e da descrição, de forma a abordar os aspectos éticos em uma experiência de desenvolvimento socialmente responsável de software. As camadas de Engenharia de Software, descritas em Pressmann e Maxim (2016), serviram de referência para organizar e explicitar que a abordagem proposta integra aspectos éticos às camadas de Engenharia de Software (Tabela 1), caracterizando, assim, o desenvolvimento socialmente responsável de software. Além disso, o papel das partes interessadas nas tomadas de decisão ética foi valorizado em todo o processo de desenvolvimento.

Tabela 1. Camadas de Engenharia de Software neste trabalho [Marques 2022]

Foco na Qualidade	Processos	Métodos	Ferramentas
<ul style="list-style-type: none"> – ISO 25010 (2011) – Requisitos Éticos/Legais 	<ul style="list-style-type: none"> – Engenharia de Requisitos – Desenvolvimento incremental 	<ul style="list-style-type: none"> – <i>Brainstorming</i> – Oficinas Semio-participativas – <i>Mockups</i> – Validação de Software – Modelagem UML – Avaliação Heurística de Usabilidade – Avaliação SUS 	<ul style="list-style-type: none"> – Termos de Consentimento Livre e Esclarecido (TCLE) – Mapa de <i>Stakeholders</i> – Quadro de Avaliação – Escada Semiótica – <i>Checklist</i> – Figma – Google (Drive, Docs e Forms) – Astah UML – Trello – Lápis, papel e caneta

O processo adotado no estudo desenvolvido está organizado em três etapas

(Figura 1): Organização da Pesquisa, Engenharia de Requisitos e Desenvolvimento Incremental.

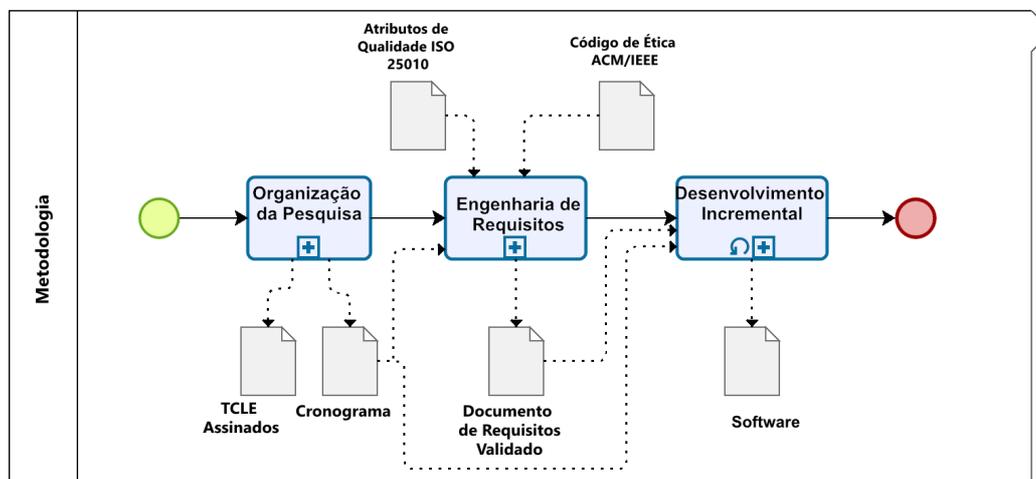


Figura 1. Metodologia [Marques 2022]

A etapa “Organização da Pesquisa” teve como objetivo a verificação da viabilidade, organização e planejamento da pesquisa. Teve início na concepção e na definição do público-alvo. Buscou-se um grupo de pessoas para a realização de Engenharia de Requisitos e desenvolvimento participativo: estudantes do ensino médio da Escola Estadual de Ensino Médio Waldemar Borges de Alegrete/RS.

Como primeiro passo, envolvendo uma professora da escola e o pesquisador, ocorreu a leitura do Termo de Consentimento Livre e Esclarecido (TCLE). Nesse momento, a professora foi informada dos objetivos da pesquisa, do caráter voluntário de sua colaboração, das vantagens e desvantagens ao participar, sendo convidada a assiná-lo. Na sequência, foi realizada uma entrevista, buscando-se informações sobre infraestrutura e disponibilidade de tempo das turmas.

A atividade de definição de diretrizes teve como objetivo a discussão entre pesquisador e professora de aspectos importantes no contato com os alunos e definição de que forma e quantos encontros seriam viáveis de serem realizados. Nessa etapa foi realizada a validação do TCLE para os Alunos e Responsáveis. Após isso, foi possível produzir um cronograma de atividades com a turma.

Já a etapa “Engenharia de Requisitos” teve como objetivo definir o software desenvolvido de forma colaborativa, levando sempre em consideração diretrizes éticas [ACM/IEEE 2015], visto que tanto o software desenvolvido quanto o processo de desenvolvimento deveriam estar eticamente alinhados. Iniciou com a apresentação da proposta de trabalho e sobre responsabilidade social e ética para a turma. Então, foi verificado se a turma tinha interesse na participação da pesquisa, sendo realizada a leitura e a distribuição dos TCLE aos alunos.

Como segunda atividade, foi realizado um *Brainstorming*, ou tempestade de ideias – técnica de dinâmica em grupo, que tem como objetivo explorar a criatividade da equipe, na forma de uma ou várias reuniões onde são levantadas e exploradas ideias sem

críticas ou julgamentos [Carvalho e Chiossi 2001]. Tinha-se como objetivo levantar possíveis temas para o software que seria desenvolvido, levando em consideração que esse software deveria apresentar uma contribuição social.

Na definição de atributos de qualidade, foi apresentada aos *stakeholders* as diretrizes de qualidade da ISO 25012 e as diretrizes éticas da ACM/IEEE 2015, para que assim fossem selecionadas e relacionadas na forma de um documento para servir como base para validações e tomada de decisões éticas durante todo o desenvolvimento. Foi produzida como resultado dessa atividade uma lista de requisitos éticos a partir das diretrizes éticas.

Então, foram conduzidas duas Oficinas Semio-participativas (OSp) [Baranauskas 2013], que são práticas que visam abordar em um mesmo ambiente diferentes visões de mundo e pluralidade de experiências em um processo de *co-design*. A parte inicial das OSp tem como objetivo a clarificação do problema, buscando as percepções de todos os envolvidos no processo de *design* e vai se aprimorando ao longo do tempo. Na primeira oficina realizada, foram produzidos o Mapa de *Stakeholders* e o Quadro de Avaliação, enquanto que na segunda oficina foi validada a Escada Semiótica produzida. Enquanto o Mapa de *Stakeholders* e o Quadro de Avaliação organizam, respectivamente, as partes interessadas e problemas/questões assim como soluções/ideias relacionados a um problema focal, a Escada Semiótica explicita requisitos em diferentes níveis de abstração, desde aspectos técnicos até sociais.

Ainda na etapa de “Engenharia de Requisitos”, fez-se uso da técnica *Mockups*, que adota o *design* iterativo em ciclos de prototipação e avaliação da interface de usuário [Muller *et al.* 1997], de modo a clarificar ideias e validar os requisitos com a turma. Essa técnica foi aplicada em três iterações: na primeira e na segunda, ocorreu a prototipação em papel pela turma com o uso de lápis, papel e caneta, enquanto que na terceira foi produzido um protótipo pelo pesquisador com o auxílio da ferramenta Figma (<https://www.figma.com/>). Esse protótipo foi verificado em uma sessão de Avaliação Heurística de Usabilidade [Nielsen 1994], que é um método de inspeção de interface com o objetivo de avaliar o *design* proposto, explicitando problemas de usabilidade.

Após a aplicação da Avaliação Heurística de Usabilidade e correção dos problemas de usabilidade encontrados, iniciou-se a especificação de requisitos, onde foi produzido um documento de requisitos, relacionando-os aos aspectos éticos para orientar a etapa de “Desenvolvimento Incremental”. Com o documento de requisitos concluído, conduziu-se sua validação com o objetivo de verificar se os requisitos especificados representavam o que os *stakeholders* solicitaram. Também foi aplicado um *checklist* visando a priorização desses requisitos.

Nessa etapa, ferramentas da Google (drive.google.com) – Docs, Form e Drive – foram utilizadas na produção e no armazenamento de documentos e formulários. Já a ferramenta Astah UML (astah.net) foi adotada na produção do diagrama de Casos de Uso incorporado ao documento de requisitos.

Finalmente, o “Desenvolvimento Incremental” envolveu uma adaptação do desenvolvimento incremental proposto por Sommerville (2011). No entanto, apenas as atividades de implementação, testes e validação foram incrementais, visto que a especificação foi realizada de forma completa na etapa de “Engenharia de Requisitos”. Nesta etapa, a ferramenta Trello (trello.com) auxiliou no gerenciamento das atividades.

Como primeiro passo no processo de desenvolvimento, foram definidas as tecnologias que seriam usadas e como a solução seria implementada. Optou-se pelo

desenvolvimento de duas aplicações independentes e desacopladas, cada uma com uma responsabilidade bem definida: uma aplicação desenvolvida em Node, responsável pela parte de servir dados; e uma aplicação em React Native, responsável pelo lado do cliente. A comunicação entre elas se dá seguindo os padrões da arquitetura cliente-servidor. No entanto, na implementação da API (do inglês, *Application Program Interface*), foi aplicado o padrão *Model-View-Controller* (MVC) e, na aplicação *mobile*, foi implementada a arquitetura em camadas, devido à divisão de componentes por sua funcionalidade, possibilitando o reúso.

O processo de desenvolvimento foi composto por 2 ciclos de desenvolvimento, priorizando os casos de uso de maior prioridade, seguido da implantação e atualização da documentação final. Em cada um desses ciclos foram realizadas as atividades de projeto, implementação, teste e validação. No primeiro ciclo de desenvolvimento, foi definida a implementação dos seguintes Casos de Uso: autenticar-se, manter itens para doação e manter informações dos ônibus. Já o segundo ciclo de desenvolvimento incremental, objetivou-se inicialmente corrigir os problemas pontuados durante a primeira avaliação e após implementação dos Casos de Uso que faltavam. A validação do software foi realizada com a aplicação do questionário SUS (do inglês, *System Usability Scale*).

Realizados os dois ciclos de desenvolvimento, deu-se início à atividade de implantação do software. Esta teve como objetivo produzir artefatos para auxiliar os usuários na utilização de software no seu ambiente e a produção dos termos de uso. Por fim, foi realizada a atividade de Documentação Final, com foco na revisão e na atualização dos documentos produzidos e a publicação do código desenvolvido em um repositório: Aplicativo¹ e aplicação de *backend*².

4. Resultados e Discussões

Como resultado da etapa “Organização da Pesquisa”, destaca-se o cronograma de atividades. A Tabela 2 apresenta esse cronograma atualizado.

Tabela 2. Cronograma de participação da turma [Marques 2022]

Data	Atividade
1º/04/2022	Apresentação da proposta de trabalho, entrega de TCLE aos alunos, apresentação sobre ética e responsabilidade social
11/04/2022	<i>Brainstorming</i> , relação ética e qualidade
18/04/2022	1ª Oficina Semio-Participativa
02/05/2022	2ª Oficina Semio-Participativa
09/05/2022	Prototipação em papel, validação documento de requisitos
16/05/2022	Prototipação em papel e consolidação
10/06/2022	Validação 2º <i>mock-up</i> , priorização de requisitos
27/06/2022	Validação primeira versão aplicativo
18/07/2022	Validação final

1 <https://github.com/michelmaarques/ComuniApp>

2 <https://github.com/michelmaarques/ComuniApp-Back-end>

O principal resultado da etapa “Engenharia de Requisitos” é o documento de requisitos, que expressa as funcionalidades do sistema e os requisitos não funcionais, destacando-se os requisitos éticos. A Figura 2 apresenta diagrama de Casos de Uso.

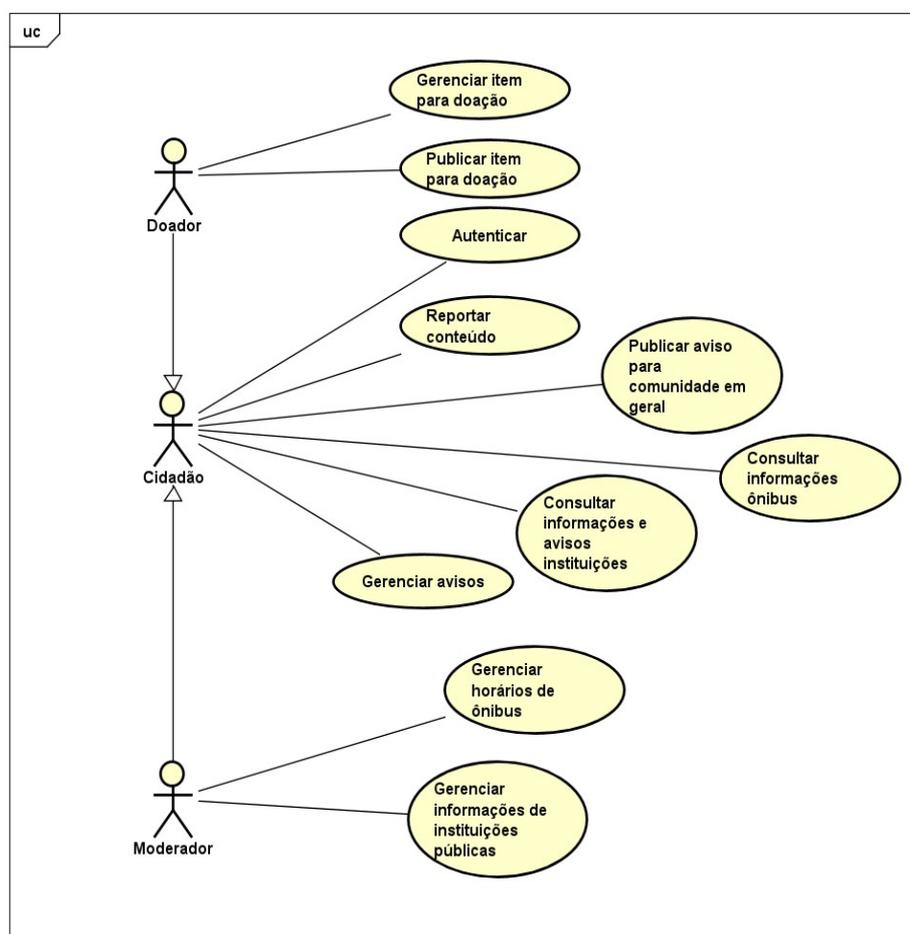


Figura 2. Diagrama de Casos de Uso [Marques 2022]

Assim, uma vez autenticado, um cidadão deveria conseguir auxiliar outro com a doação de algum item ou publicação de avisos. Além disso, poderia consultar informações sobre ônibus e instituições, como escolas, postos de saúde e centros de assistência social. Ao moderador caberia gerenciar os horários de ônibus e informações sobre instituições públicas.

Entre os requisitos éticos mapeados, tem-se: requisitos relacionados a não utilização de software mantido ou obtido de forma ilegal ou aética; tem-se também a preocupação quanto à qualidade e adequação do processo de desenvolvimento; a confidencialidade e privacidade dos dados; a necessidade de considerar em todos os julgamentos técnicos os valores humanos; a acessibilidade quanto ao uso da aplicação por pessoas com deficiência, alocação de recursos, desvantagem econômica ou outros fatores que possam diminuir o acesso aos benefícios do software.

Ainda durante a etapa de “Engenharia de Requisitos”, foi possível notar a importância das discussões sobre a ética no desenvolvimento e sua relação com os

requisitos funcionais, visto que questões sobre acessibilidade e usabilidade surgiram diretamente por parte dos alunos, assim como o requisito funcional para reportar conteúdo inapropriado.

A Figura 3, a seguir, apresenta telas da aplicação, produzidas na etapa de Desenvolvimento Incremental.

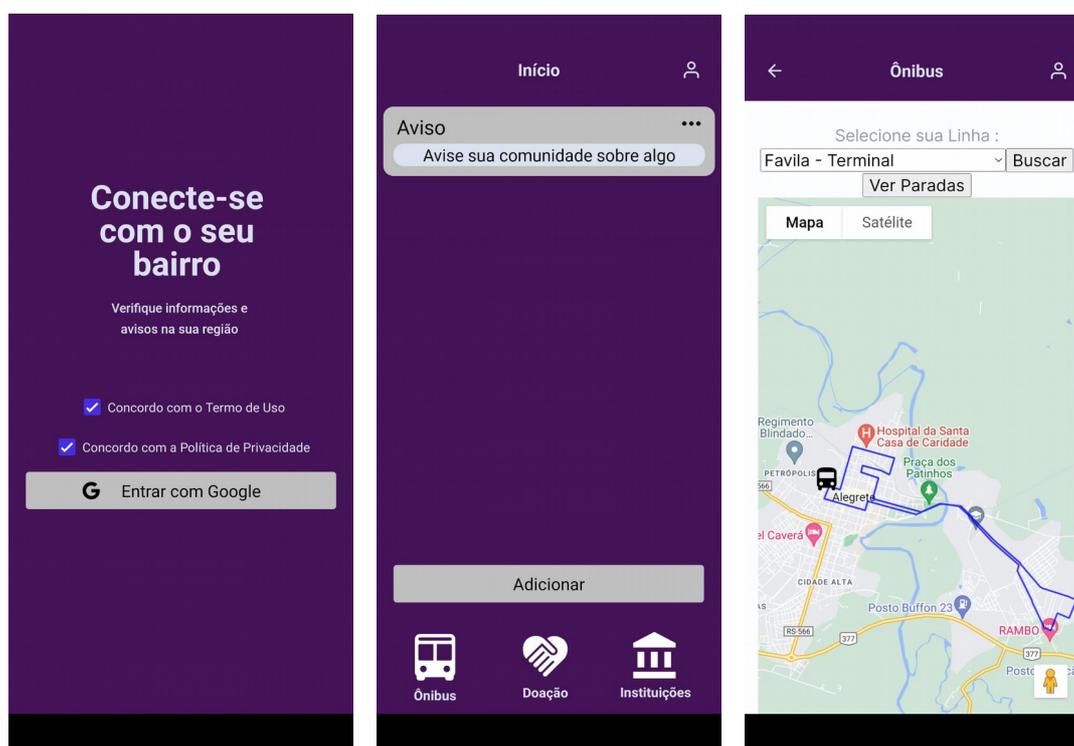


Figura 3. Telas do aplicativo ComuniApp desenvolvidas no primeiro ciclo do Desenvolvimento Iterativo [Marques 2022].

Na Figura 3, tem-se as telas desenvolvidas no primeiro ciclo de desenvolvimento iterativo: tela de autenticação, tela de início tela da funcionalidade relacionada aos ônibus. Conforme definido durante a Engenharia de Requisitos, o aplicativo deve oferecer um mecanismo de autenticação – nesta versão do *app*, isso é realizado com a conta Google depois que o usuário concorda com os Termos de Uso e Política de Privacidade. Na tela de início, além de visualizar e publicar avisos, um usuário autenticado pode acessar outras funcionalidades a partir do menu que destaca as opções Ônibus, Doações e Instituições. Na tela relacionada aos ônibus, é possível selecionar a linha que deseja visualizar, então o ônibus aparecerá no mapa em tempo real com o seu respectivo trajeto. Esta última funcionalidade usa como fonte de dados a aplicação desenvolvida e disponibilizada pela empresa de transporte coletivo da cidade. O aplicativo desenvolvido neste trabalho torna visual as informações disponibilizadas.

O software desenvolvido, portanto, é socialmente responsável em seu objetivo, pois tem alto potencial de impacto no dia a dia de seus usuários. Com o ComuniApp, uma pessoa pode auxiliar outra com a doação de algum item; seus usuários, que dependem de ônibus para se deslocar, podem monitorar a localização de determinada linha; pode-se publicar avisos que ficarão visíveis para os demais usuários; além disso,

o acesso à informações de instituições como escolas, postos de saúde e centros de assistência social é facilitado.

5. Considerações Finais

Aspectos éticos estiveram presentes durante todo o processo de desenvolvimento de software em tela neste trabalho, visto que ocorreram diversos momentos onde foi necessário tomar decisões que remetem ao impacto social do software em perspectiva. Destaca-se a participação dos alunos durante todas as etapas desse processo e seu envolvimento nas tomadas de decisão, desde o início do processo até chegar no resultado final. A ideia de produzir um software socialmente responsável resultou em um produto que reflete muita consciência social de suas partes interessadas, onde valores humanos estão presentes especialmente nas funcionalidades relacionadas às doações e aos avisos.

No produto, a ética e a responsabilidade social são refletidas em seus requisitos, como, por exemplo, o *reporte* de conteúdo inapropriado, que surgiu durante uma das últimas tarefas da etapa de Engenharia de Requisitos – fruto de uma conversa sobre os aspectos éticos relacionados ao uso da aplicação. Já a visualização da rota do ônibus no mapa pode ser considerada de grande impacto social, visto que as pessoas que moram em bairros mais distantes e dependem dos poucos ônibus que têm na cidade podem se beneficiar de forma a evitar atrasos ao ir para a parada. Além de observar aspectos éticos no produto, é importante destacar que o comportamento ético do usuário é de suma importância para que o aplicativo alcance seu propósito.

Entre as principais contribuições deste trabalho estão: o produto gerado, constituído de duas aplicações com potencial significativo de impacto social; e as reflexões sobre ética na Engenharia de Software, considerando atributos de qualidade, processo, métodos, ferramentas e pessoas. Quanto às limitações deste trabalho, destaca-se que, embora o *app* possa ser utilizado, não está suficientemente maduro para ser amplamente difundido junto à população em geral, devido à necessidade de atribuição de papel de moderador para as publicações e atualização de dados das instituições. Pode-se citar, ainda, que a promoção da acessibilidade não foi bem sistematizada no desenvolvimento do aplicativo, mas deve ser considerada para a publicação de uma versão que alcance mais usuários, além dos atuais.

Como trabalhos futuros, tem-se a possibilidade de escalar as aplicações para que sejam capazes de atender a cidade como um todo. Contudo, por mais que sua arquitetura tenha sido pensada de forma escalável, os custos de manutenção devem ser considerados. Como a parte de autenticação foi desenvolvida de forma a possibilitar que esta funcionalidade seja feita a partir de outras redes sociais, é algo que pode ser implementado futuramente. Ademais, na condução de um processo de evolução, aspectos éticos também devem ser observados, por exemplo, mediando as doações em mecanismo de comunicação próprio, evitando a exposição do dado de contato. Finalmente, pode ser desenvolvido um *checklist* que auxilie a contemplar aspectos éticos na definição de qualidade, processos, métodos e ferramentas no desenvolvimento de software com a participação das partes interessadas nas tomadas de decisão.

6. Agradecimentos

Aos estudantes e professores da EEEM Waldemar Borges, de Alegrete/RS, *co-designers* do produto apresentado neste artigo.

Referências

- ACM/IEEE (2015), Software Engineering 2014, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, A Volume of the Computing Curricula Series.
- Baranauskas, M. C. C. (2013) “O Modelo Semioparticipativo de Design”, Codesign de redes digitais: Tecnologia e educação a serviço da inclusão social, Baranauskas, M. C. C., Martins and M. C. Valente, J. A., Penso Editora.
- Benedicto, S. *et al.* (2008) “Surgimento e evolução da responsabilidade social: uma reflexão teórico-analítica.”, In: Encontro Nacional de Engenharia de Produção, Rio de Janeiro.
- Borg, M. *et al.* (2021), “Exploring the Assessment List for Trustworthy AI in the Context of Advanced Driver-Assistance Systems”, In SEthics 2021, pp. 5-12.
- Carvalho, A. M. B. R., Chiossi, T. C. S. (2001), Introdução à Engenharia de Software, Editora da Unicamp, 1st edition.
- Dekhlyar, A. *et al.* (2020), “From RE Cares to SE Cares: Software Engineering for Social Good, One Venue at a Time”. In ICSE-SEIS’20.
- ISO (2011). ISO/IEC 25010:2011 Systems and software engineering: Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. 1st edition.
- Kitchenham, B. (2004) Procedimentos para realizar revisões sistemáticas. Keele University, v. 33, p. 1-26, 2004.
- Marques, M. S. (2022) Uma abordagem socialmente responsável ao desenvolvimento de software, Unipampa.
- Muller, M. J., Haslwanter, J. H., Dayton, T. (1997) “Participatory Practices in the Software Lifecycle”, Handbook of Human-Computer Interaction, M. G. Helander, T. K. Landauer and P. V. Prabhu, Elsevier Science, 2nd rev. edition, p. 255-297.
- Nielsen, J. (1994), Usability Engineering, Morgan Kaufman, 1st editon.
- Mustafa, N. A., Yusoff, M. F., Rofie, M. K. *et al.* (2020), “Modeling the Structural of Intention and Attitude of Muslim Software Developer to Use Islamic Work Ethic in Developing Islamic Content Mobile Apps”, In ICIDM 2020, pp. 1-6.
- Pressmann, R. S., Maxim, B. R. (2016), Engenharia de software: uma abordagem profissional, AMGH, 8th edition.
- Ribeiro, M. S. (2002), “A evolução dos conceitos de responsabilidade social”, In IX Congresso Brasileiro de Custos.
- Shneiderman, B. (2020). Bridging the Gap Between Ethics and Practice: Guidelines for Reliable, Safe, and Trustworthy Human-centered AI Systems, In *ACM Transactions on Interactive Intelligent Systems*, vol. 10, n. 4, art. 26.
- Sommerville, I. (2011), Engenharia de Software, Pearson Education, 9th edition.
- Terence, A. C. F., Escrivão Filho, E. (2006), “Abordagem quantitativa, qualitativa e a utilização da pesquisa-ação nos estudos organizacionais”, In: XXVI ENEGEP.
- Vakkuri, V., Kemell, K. Abrahamsson, P. (2020), “ECCOLA – A Method for Implementing Ethically Aligned AI Systems”, In: SEAA 2020, pp. 195-204.