

Uma Proposta de Ferramenta de Apoio ao Processo de Avaliação de Usabilidade em Linguagens Específicas de Domínio

Allan Pedroso¹, Ildevana Poltronieri², Maicon Bernardino¹

¹ Universidade Federal do Pampa (UNIPAMPA)
Laboratory of Empirical Studies in Software Engineering (LESSE)
Av. Tiarajú, 810, Ibirapuitã – Alegrete, RS – Brasil

²Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Av. Ipiranga, 6681, Partenon – Porto Alegre – RS – Brasil

{allannpedroso, ildevana}@gmail.com, bernardino@acm.org

Abstract. *DSLs have gained attention in the industry for being focused on a particular domain and used by users with no programming experience. It allows users with no prior knowledge of programming languages but familiarity with the problem domain to develop their systems. However, usability criteria must be taken into consideration for DSL to be utilized fully by users. The development of the Usa-DSL Process aims to meet these criteria which, seeks to mitigate existing gaps concerning the processes or guidelines that guide the DSL designer in evaluation usability. Although this process supports the evaluation of the usability of DSLs, automation is necessary. Therefore, this study aims to propose a Web tool that automates the Usa-DSL Process.*

Resumo. *As DSLs têm ganhado notoriedade na indústria por serem focadas em um domínio particular, e utilizadas por usuários sem experiência em programação. Isso permite que seus usuários tenham familiaridade com o domínio do problema, mesmo não possuindo conhecimento prévio em programação. Porém, para que a DSL seja usada pelos usuários, critérios de usabilidade devem ser estabelecidos. Para atender tais critérios, desenvolveu-se o Processo Usa-DSL, que busca mitigar as lacunas existentes em relação aos processos ou diretrizes que orientam o designer de DSL na avaliação de usabilidade. Embora esse processo suporte a avaliação de usabilidade das DSLs, compreende-se que sua automatização é necessária. Sendo assim, este estudo propõe uma ferramenta Web que automatize o Processo Usa-DSL.*

1. Introdução

Com a crescente adesão dos sistemas de informação nos mais diversos tipos de domínio, fez-se necessário a busca de alternativas para o seu desenvolvimento, dado o fator implicador da utilização das Linguagens de Propósito Geral (*General-purpose Programming Language - GPL*), a exemplo de linguagens como Java, Python, C++ e além de outras que são comumente utilizadas na implementação das aplicações que conhecemos. As GPL oferecem uma vasta gama de tipos de dados, controles e estruturas de abstração, porém, isso faz com que a curva de aprendizado se torne alta, o que tornam o seu uso mais difícil [Fowler 2010].

Por outro lado, as Linguagens Específicas de Domínio (*Domain-Specific Language* - DSL) tem ganhado notoriedade em razão de que suprem algumas das deficiências encontradas no uso das GPL, como ganhos em termos de eficiência na implementação de soluções de software em domínios específicos. Com as DSL é possível mitigar os problemas de abstração na conversão de modelos de domínio para conceitos computacionais, facilitando a comunicação dos programadores com seus clientes. Segundo [Fowler 2010], esta falta de comunicação se trata do maior gargalo do desenvolvimento de software, dado o desafio que se tem em aproximar a solução de software da base de conhecimento de certos domínios.

As DSL são linguagens voltadas para uso em um domínio em particular, a exemplo de uso em testes de desempenho [Bernardino et al. 2016], utilização em aplicações de roteamento de veículos [Hoffmann et al. 2019], no controle de robôs [Barisic et al. 2018b], no campo da medicina [Msosa 2019], entre outras. Dentre as vantagens da utilização das DSL, incluem-se a redução do custo de manutenibilidade somado a redução considerável da necessidade de experiência em programação por parte de seus usuários [Mernik et al. 2005]. Porém, segundo [Fowler 2010], para que essas vantagens sejam observadas e a linguagem seja de fato usada em sua plenitude, ela deve ter um foco claro do domínio que quer atingir. Vale ressaltar que o foco na descrição fiel ao domínio é o que mede o sucesso de uma linguagem.

Em relação a usabilidade das DSL é dito que com a adoção de DSL existe o risco de que uma linguagem mal projetada não seja aceita pelos usuários do domínio-alvo. Neste sentido, para que as DSL sejam de fato aceitas e utilizadas nos diferentes domínios em que são implantadas, os riscos de desuso devem ser mitigados. Para isso, a usabilidade é uma das características observada como essencial, pois tem impacto relevante na produtividade alcançada por usuários de DSL. Em sua maioria, as avaliações encontradas atualmente são realizadas somente nas últimas fases do ciclo de desenvolvimento, quando as mudanças na linguagem tem um impacto significativo no orçamento [Barisic et al. 2012]. A falta de abordagens sistemáticas e um conjunto de diretrizes e ferramentas que guiem uma avaliação de usabilidade são fatores que explicam esta lacuna no atual estado da prática. É dito que a deficiência da relação entre os processos de desenvolvimento de DSL e usabilidade é a causa desta lacuna [Barisic et al. 2018a]. Entretanto, há um processo emergente que faz uso de conhecimentos de Interação Humano-Computador (IHC), Engenharia de Software Experimental (ESE) e DSL para avaliar aspectos de usabilidade em DSL, denominado *Usa-DSL Process* [Poltronieri et al. 2019].

A fim de desenvolver uma solução que apoie e automatize a execução do *Usa-DSL Process*, este estudo apresenta as fases de desenvolvimento da ferramenta, bem como estudos complementares que foram realizados para dar suporte ao seu desenvolvimento.

Este artigo está organizado da seguinte forma. A Seção 2 descreve a definição do *Usa-DSL Process*. A Seção 3 apresenta a análise e o projeto da ferramenta proposta. Finalmente, a Seção 4 apresenta a conclusão e os trabalhos futuros.

2. *Usa-DSL Process*: Processo para Avaliação de Usabilidade de DSLs

O *Usa-DSL Process* [Poltronieri et al. 2019] é uma abordagem sistemática desenvolvida para avaliar a usabilidade de DSL. A origem do processo deu-se a partir da necessidade de orientar, organizar e sistematizar os conceitos desenvolvi-

dos no *Usa-DSL Framework* [Poltronieri et al. 2018]. Para a construção deste processo fez-se uso do metamodelo *Software Process Engineering Metamodel* (SPEM) [OMG - Object Management Group 2018] e como apoio utilizou-se a ferramenta de modelagem de processos EPF Composer [Foundation 2018]. O EPF Composer permitem gerar uma página Web estática que auxilia na navegação e visualização dos conceitos, fluxos e descrição do processo ajudando os projetistas de DSL a desenvolver e conduzir as avaliações de usabilidade das suas DSL.

O *Usa-DSL Process* é norteado por quatro princípios fundamentais [Sommerville 2010]: *Supporting*, *Continuous Feedback*, *Productivity* e *User-Centered Evaluation*. O *Usa-DSL Process* [Rodrigues 2021] [Rodrigues et al. 2019] foi desenvolvido com base nos conceitos do *Usa-DSL Framework* [Poltronieri et al. 2018], ou seja, seguindo a sua estrutura de *Phases*, *Steps* e *Activities*.

Seu intuito principal é nortear as questões “Quem fará o quê, quando e como?” [Pressman 2010], sempre de forma iterativa [Stone et al. 2005] e guiado para o desenvolvimento de avaliações formais com facilidade e rapidez. O seu principal objetivo é fornecer os procedimentos para realização da avaliação no que se refere à usabilidade de DSLs, verificando por meio de métodos, técnicas e atividades o quanto estas linguagens atendem à questões de qualidade de uso. O *Usa-DSL Process* trata da elaboração das avaliações de usabilidade envolvendo os métodos *Heuristic Evaluation* e *Usability Testing* [Nielsen 1993], com a intenção de sistematizar e desenvolver as avaliações de uma forma produtiva e com menos custo.

O *Usa-DSL Process* possui uma estrutura composto por conteúdo de método (*Method Content*) e processo (*Process*). O *Method Content* trata de uma base de conhecimento, no qual são armazenadas as informações que serão consumidas pelo processo. Já o *Process* trata da organização dos elementos do *Method Content*, relacionando este elementos em sequência parcialmente ordenados e de forma personalizada para cada projeto específico. O *Usa-DSL* assim como o SPEM, contém uma categoria de agrupamento de elementos que se destina aos *Guidelines*, ou seja, *templates* de documentos que serão consumidos, gerados ou alterados durante todo o ciclo de vida. Estes *templates* serão utilizados tanto pelo conteúdo de método como pelo processo, eles são descritos de uma forma única. O *Usa-DSL Process* possui diversos artefatos deste tipo que poderão ser utilizados durante a execução de uma tarefa e/ou do processo gerado.

2.1. Method Content - Usa-DSL Process

O *Method Content* que compõem o *Usa-DSL Process* está organizado conforme os elementos da notação SPEM, porém mapeados para os elementos do *Usa-DSL Framework*:

Work Product: são os arquivos que serão consumidos ou gerados durante a execução de uma atividade.

Profile: é um elemento do *Method Content* que define os papéis desempenhados dentro do processo. Os *Profiles* são usados para definir quem executa cada *Task*, bem como para definir os responsáveis por um conjunto de *Work Products*. O *Usa-DSL Process* define oito (8) *Profiles*, os quais foram agrupados em três (3) *Profile Sets*, no qual foram reunidos por tipo de execução, são eles:

- *Process Executors*: este conjunto de *Profiles* reúne os usuários do *Usa-DSL Process*, ou seja, quem planeja e conduz as avaliações. Estes *Profiles* fazem parte do

grupo que projeta, desenvolve e possui interesse em avaliar a DSL, são eles: *DSL Analyst*, *DSL Developer* e *DSL Tester*;

- *Usability Evaluation Subjects*: este conjunto de *Profiles* reúne os *Domain Analyst*, *Domain Developer*, *Domain Tester* e *End User*, ou seja, as pessoas que serão convidadas a participar da avaliação. Aqueles que contribuirão com os desenvolvedores da DSL, apontando as melhorias ou correções que devem ser realizadas para que se alcance um experiência mais agradável para o usuário;
- *Heuristic Evaluation Subjects*: este *Profile* é composto apenas pelo *Expert* em IHC, ou seja, o especialista em avaliação heurística. Estes participantes tem como objetivo avaliar a DSL e contribuir com o grupo de desenvolvedores a partir dos erros e grau de severidade apontados durante a avaliação, buscando melhorar a usabilidade e experiência do usuário antes de apresentar a DSL para os usuários finais.

Task: descreve uma unidade de trabalho com propósito claro, a ser atribuída a uma *Activity* e um *Profile* específico, com a finalidade de atingir uma meta bem definida. As *Tasks* que compõem o *Usa-DSL Process* serão apresentadas no contexto de uma *Activity* ao qual ela pertence.

Step: é uma forma flexível de definir diferentes agrupamentos para categorias de conteúdo. No *Usa-DSL Process* um *Step* é definido pelo elemento *Discipline* que é derivado de um elemento *Category* do SPEM. Um *Step* é um agrupamento de trabalho com base na similaridade de preocupações e cooperação do esforço de trabalho, ou seja, é uma coleção de *Tasks* relacionadas a determinadas *Activities* e compostas por um conjunto de perfis denominados *Role Sets*, por exemplo, um *Evaluation Instructions* é um *Step* que agrupará todas as *Tasks* e *Profiles* que fazem parte desta área de preocupação.

2.2. Process - Usa-DSL Process

O *Process* do *Usa-DSL* apresenta o ciclo de vida do processo de avaliação de usabilidade, por meio dos elementos derivados do SPEM. Este processo é composto pelos elementos: *Phase*, *Activity*, *Profile Use*, *Task Use* e *Work Product Use*, os quais são descritos a seguir:

Phase no *Usa-DSL Process* é um elemento do tipo *Activity* no SPEM, no qual define a divisão de trabalho, ou seja, este elemento organiza o processo em um espaço de tempo, no qual as *Activities* serão executadas. As *Phases* são orientados por *Steps* que determinam quais *Activities* serão executadas e em que momento do processo. O *Usa-DSL Process* é composto por quatro *Phases*: *Planning*, *Execution*, *Analysis* e *Reporting*. É importante enfatizar que cada *Phase* do processo é concluída por um marco, ou seja, um conjunto de atividades e artefatos gerados pelo grupo que realiza a avaliação a ser entregue caracterizando o encerramento da *Phase* corrente.

Activities no *Usa-DSL Process* é um elemento do tipo *Activity* no SPEM, e compõem uma divisão de trabalho que possui um conjunto de *Tasks*, as quais definem unidades básicas de trabalho dentro de um processo, bem como o processo em si. Ela representa uma unidade geral de trabalho que pode ser atribuída a um *Profile Use* específico, bem como receber entradas e produzir saídas. Uma *Activity* representa um elemento de agrupamento para outros elementos de divisão de trabalho, tais como: *Tasks Use*, *Profiles Use* e *Work Product Use*. No caso do *Usa-DSL Process*, as *Activities* agrupam as *Tasks* referentes a um determinado *Step* dentro de um espaço de tempo de uma *Phase* no ciclo

de vida do processo. Por este motivo, cada *Activity* possui uma identificação composta pela letra inicial da *Phase* e o número do *Step*.

Profile Use é um elemento do tipo *Role Use* no SPEM, este elemento representa um executor de uma *Activity* ou um participante da *Activity*. Os *Profiles Use* são perfis que estão definidos no *Profiles* da base de conhecimento, logo já possuem funções definidas. Nesta etapa do processo eles apenas serão relacionados a *Task* que devem executar.

Task Use é um elemento de divisão de trabalho que representa uma *Task* que está sendo utilizada por um determinado *Profile* no contexto de uma *Activity* específica. Pode-se exemplificar por meio da *Task P1a Choose the Profile Evaluator*, a qual está relacionada a *Activity P1 - Define Evaluators Profiles* e esta sendo executada pelo avaliador da DSL, por exemplo *Profile DSL Developer*, em que neste caso estará realizando o planejamento da avaliação por meio do *Usa-DSL Process*.

Work Products Use são os artefatos definidos no *Work Product* e que são utilizados durante a execução do processo. Estes artefatos são produtos ou documentos de trabalho que podem ser consumidos, gerados ou alterados durante a execução de uma *Task* específica dentro de uma *Activity*. Por exemplo, o ICT é um artefato de termo de consentimento livre esclarecido consumido durante a tarefa *P2a Select the Informed Consent Term* (ICT) e será alterado na tarefa *E2c Collected Signature of the Subject* no momento que o participante assinar o termo, a partir desta assinatura inclui-se um dado que não possuía ao artefato quando este foi adicionado a base de conhecimento do processo.

3. Usa-DSL Tool

Nesta seção são apresentadas os aspectos gerais de análise e projeto da ferramenta proposta, conforme estruturado nas seções que seguem.

3.1. Visão Geral da Ferramenta

Durante o desenvolvimento do *Usa-DSL Process* foi observado que, mesmo com a disponibilidade de uma página Web (*Wiki*) contendo os fundamentos, conceitos, definições e artefatos necessário para a condução do processo, foi identificada a necessidade de desenvolver uma ferramenta para dar apoio de forma dinâmica e sistemática. Com isso, neste trabalho se propõe uma ferramenta que torne a aplicação do processo mais dinâmica, guiando o avaliador ou participante de uma avaliação de usabilidade passo a passo, fornecendo todo o *feedback* necessário acerca das atividades do *Usa-DSL Process*.

3.2. Requisitos do Sistema

Nesta seção são apresentados os requisitos da ferramenta proposta, os quais foram categorizados em Requisitos Funcionais (Seção 3.2.1) e Não-Funcionais (Seção 3.2.2).

3.2.1. Requisitos Funcionais

De acordo com [Sommerville 2010] requisitos funcionais são definições de como um sistema reage a partir de uma determinada entrada e do comportamento dele a determinadas situações. Os requisitos funcionais descrevem basicamente quais funcionalidades o sistema deve atender. Os requisitos funcionais da proposta de ferramenta foram definidos com base nas atividades do ciclo de vida do *Usa-DSL Process* apresentado na Seção 2.

Para esta ferramenta os seguintes requisitos funcionais foram definidos conforme apresentado na Tabela 1:

Tabela 1. Lista dos Requisitos Funcionais.

ID	Nome	Descrição
RF01	Manter Cadastro de <i>Process Executor</i>	Como <i>Process Executor</i> quero cadastrar, editar, consultar e excluir o meu perfil.
RF02	Realizar Login	Como <i>Usuário</i> do sistema quero realizar autenticação utilizando meus dados previamente cadastrados.
RF03	Manter Cadastro de Participante	Como <i>Participante</i> quero visualizar, alterar ou excluir meu cadastro.
RF04	Criar Projeto de Avaliação de Usabilidade de DSL	Como <i>Process Executor</i> desejo criar um projeto de avaliação de usabilidade.
RF05	Adicionar Participante em Avaliação	Como <i>Process Executor</i> quero adicionar um participante no projeto de avaliação.
RF06	Criar Cronograma de Experimento	Como <i>Process Executor</i> quero criar um cronograma para o experimento.
RF07	Definir Termo de Consentimento da Avaliação	Como <i>Process Executor</i> quero definir como será o termo de consentimento da avaliação.
RF08	Selecionar Tipo de dado Utilizado na Avaliação	Como <i>Process Executor</i> quero definir se minha avaliação utilizará dados qualitativos ou quantitativos.
RF09	Editar Instrumentos de Coleta de Dados	Como <i>Process Executor</i> quero editar instrumentos de obtenção de dados como <i>Checklist</i> Heurístico, Questionário de Usabilidade, Questionário de Perfil e <i>Script</i> de Entrevista.
RF10	Selecionar Tipo de <i>Checklist</i> Heurístico	Como <i>Process Executor</i> quero escolher o tipo de <i>checklist</i> heurístico que será aplicado na minha avaliação.
RF11	Criar Cenário de Uso	Como <i>Process Executor</i> quero criar cenários de uso para a minha avaliação.
RF12	Definir Local de Execução da Avaliação	Como <i>Process Executor</i> quero definir se minha avaliação ocorrerá em ambiente físico ou online.
RF13	Definir Tipo de Armazenamento de Dados da Avaliação	Como <i>Process</i> quero escolher em qual meio serão salvos os dados gerados na avaliação.
RF14	Definir Tipo de <i>Report</i> de Estudo	Como <i>Process Executor</i> quero escolher como será reportado meu estudo.
RF15	Desenvolver Protocolo da Avaliação	Como <i>Process Executor</i> quero desenvolver o protocolo da avaliação.
RF16	Preparar Avaliação de DSL	Como <i>Process Executor</i> quero fornecer todas ferramentas necessárias e ambiente para a realização da avaliação.
RF17	Confirmar Participação em Avaliação	Como <i>Participante</i> quero confirmar ou não a minha participação em uma avaliação.
RF18	Confirmar Aceite de Termos	Como <i>Participante</i> quero ter acesso ao termo de consentimento da avaliação e confirmar ou não o aceite do mesmo.
RF19	Identificar Perfil de Participante da Avaliação	Como <i>Participante</i> quero que meu perfil seja classificado adequadamente na avaliação.
RF20	Disponibilizar Materiais de Suporte da Avaliação	Como <i>Process Executor</i> quero anexar materiais de suporte na avaliação.
RF21	Obter Materiais de Suporte da Avaliação	Como <i>Participante</i> quero ter acesso a materiais que deem apoio a execução da avaliação.
RF22	Executar Cenário de Uso	Como <i>Participante</i> quero visualizar e executar os cenários de uso da avaliação.
RF23	Anexar Modelos para Análise	Como <i>Participante</i> quero submeter meus modelos gerados durante a execução de tarefas para posterior análise.
RF24	Aplicar <i>Checklist</i> Heurístico	Como <i>Participante</i> quero responder a questões heurísticas da DSL avaliada.
RF25	Classificar Participante	Como <i>Process Executor</i> quero visualizar as respostas do participante no questionário de perfil, para classifica-lo de acordo com seu nível de conhecimento.
RF26	Analisar Tarefas Executadas	Como <i>Process Executor</i> quero analisar as tarefas executadas pelos participantes no decorrer da avaliação.
RF27	Adicionar Colaborador do Projeto	Como <i>Process Executor</i> quero adicionar um pesquisador como colaborador do projeto de avaliação.

3.2.2. Requisitos Não-Funcionais

O autor [Sommerville 2010] descreve requisitos não-funcionais como funcionalidades que não estão diretamente ligadas as essenciais ao sistema, estas inerentes por exemplo a delimitações no uso de certas tecnologias para a implementação de um sistema, ou relativas a critérios de desempenho e segurança.

- **RNF1:** O sistema deve prover acesso as suas funcionalidades de acordo com a autenticação;

- **RNF2:** O sistema deve ser implementado fazendo uso da linguagem de programação orientada a objetos;
- **RNF3:** O sistema deve obedecer a arquitetura que evite acoplamento;
- **RNF4:** O sistema deve possuir *design* responsivo.

3.3. Decisões de Projeto

Nesta seção são discutidas as decisões de projeto que foram tomadas com base nos requisitos definidos na Seção 3.2.

- **DP1:** A ferramenta será implementada fazendo uso da linguagem de programação PHP¹, dado que esta obedece ao paradigma de *orientação a objetos*;
- **DP2:** Como *framework* será utilizado o Laravel²;
- **DP3:** O SGBD estabelecido para o projeto da ferramenta é o MySQL³, devido ao fato de ser consolidado e amplamente utilizado;
- **DP4:** A arquitetura definida para o projeto foi a *Model-View-Controller* (MVC), seguindo a arquitetura padrão do *framework* Laravel;
- **DP5:** Para a implementação do *front-end* da ferramenta foi definida a utilização de HTML5, CSS3, JavaScript e a biblioteca Bootstrap 5⁴.

3.4. Análise do Projeto

Nesta seção será apresentado um exemplo de especificação de caso de uso e seus respectivos protótipos de tela. O caso de uso está descritos conforme *template* apresentado na Tabela 2, o qual está associado aos protótipos de tela das Figuras 1 e 2 descritas na tabela. Os elementos de tela são sinalizados em amarelo, utilizando letras para campos de inserção ou seleção de dados, e números para botões ou links. Durante a especificação dos requisitos, na modelagem do diagrama de casos de uso identificou-se 28 casos de usos, executados por cinco atores. Os atores “Process Executor” e “Participante” herdam do ator “Usuário”, enquanto que os atores “Usability Evaluation Subject” e “Heuristic Evaluation Subject” herdam os casos de uso do ator “Participante”. A especificação completa dos casos de uso da ferramenta estão descritos na Wiki do repositório GitHub⁵.

Atualmente, a ferramenta encontra-se em estágio de desenvolvimento. Toda a fase de análise e projeto foram executadas, avaliando as funcionalidades (Seção 3.2) da ferramenta de acordo com o fluxo do *Usa-DSL Process*. A Figura 4 apresenta um *screenshot* de uma das telas já implementadas na ferramenta que está em desenvolvimento.

4. Considerações Finais e Trabalhos Futuros

A usabilidade é um aspecto muito importante a ser observado no desenvolvimento de software, e a engenharia de linguagens não se difere neste sentido. Posto isto, o Processo Usa-DSL foi desenvolvido com a finalidade de mitigar a lacuna observada na revisão sistemática conduzida [Poltronieri et al. 2021]. Apesar do *Usa-DSL Process* cumprir o seu objetivo de fornecer os procedimentos para realização da avaliação no que se refere

¹PHP: <https://www.php.net/>

²Laravel: <https://laravel.com/>

³MySQL: <https://www.mysql.com/>

⁴Bootstrap: <https://getbootstrap.com/>

⁵Repositório do Projeto: <https://github.com/allanpedroso/USA-DSL-Process-Tool/wiki>

Tabela 2. UC04 - Criar Projeto de Avaliação de Usabilidade de DSL

Identificador:	UC04
Caso de Uso:	Criar Projeto de Avaliação de Usabilidade de DSL
Ator Principal:	Process Executor
Pré-condições:	Process Executor autenticado no sistema.
Pós-condições:	Projeto de avaliação Criado.
Protótipos de tela associados:	UC04 - TELA 1; UC04 - TELA 2; UC04 - TELA 3
Fluxo Principal	
<ol style="list-style-type: none"> 1. O caso de uso começa quando o Process Executor deseja criar um novo projeto de avaliação; 2. O Process Executor seleciona a guia “1” na TELA 1; 3. O Sistema exibe os tipos de avaliação de DSL no campo de opções selecionáveis “A” na TELA 1; 4. O Process Executor seleciona o tipo de avaliação; 5. O Sistema exibe os métodos de estudo empírico no campo de opções selecionáveis “B” na TELA 2; 6. O Process Executor seleciona o método de estudo empírico; 7. O sistema exibe o formulário com os campos “C”, “D”, “E”, “F”, “G” e “H” na TELA 3; 8. O Process Executor informa os dados do projeto nos campos e clica no botão “1” da TELA 3; 9. O Sistema valida os dados, os salva e o caso de uso se encerra; 	
Fluxos Alternativos	
<ol style="list-style-type: none"> 4.A O Process Executor seleciona Avaliação de Usabilidade; 5.A O Sistema exibe os seguintes métodos de estudo empírico: Survey, Case Study, Experiment; 4.B O Process Executor seleciona Avaliação Heurística; 5.B O Sistema seleciona automaticamente Survey como tipo de método de estudo empírico. 	
Fluxos de Exceção	
<ol style="list-style-type: none"> 9. Os dados não foram preenchidos ou foram preenchidos de forma incorreta; 9.1 O sistema exibe uma mensagem de erro e retorna ao passo 2 do fluxo principal; 	
Requisitos Não Funcionais	
RNF01 – O sistema deve prover acesso as suas funcionalidades de acordo com a autenticação.	

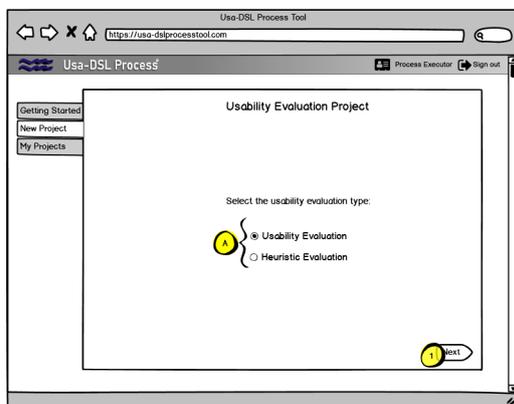


Figura 1. UC04 - Tela 1

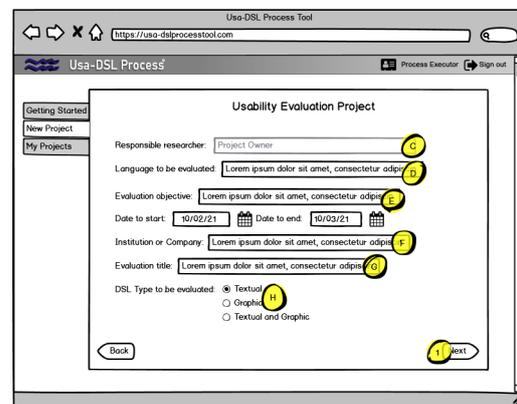


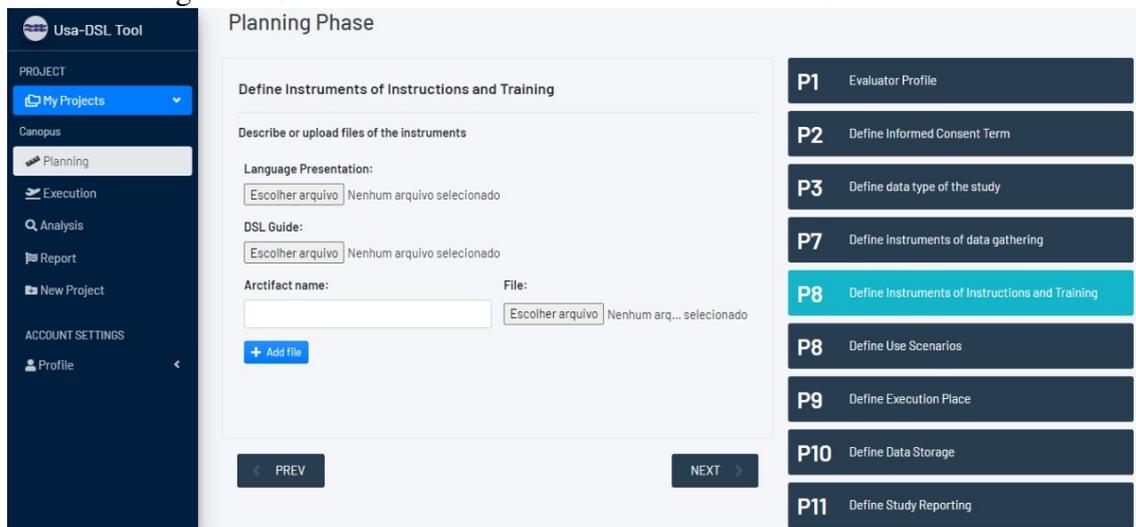
Figura 2. UC04 - Tela 3

Figura 3. Protótipos de tela do caso de uso UC04.

à usabilidade de DSLs, verificando por meio de métodos, técnicas e atividades o quanto estas linguagens atendem à questões de qualidade de uso, foi identificada a necessidade de desenvolver uma ferramenta para apoiar a automatização do processo como um todo. Sendo assim, o desenvolvimento de uma ferramenta Web colaborativa que visa apoiar o processo e automatizar a execução de todo o seu ciclo de vida esta sendo pensada.

A ferramenta proposta, auxiliará os projetistas de DSL nas etapas de planeja-

Figura 4. Screenshot de um das telas da Ferramenta Usa-DSL Tool



mento, execução, análise e publicação das avaliações de usabilidade de DSL. Como trabalhos futuros deve-se gerar o produto de software à partir da proposta apresentada, bem como a avaliação da ferramenta desenvolvida. Para avaliar a ferramenta planeja-se aplicar uma avaliação empírica por meio de um estudo de caso, bem como avaliar os aspectos de usabilidade da própria ferramenta proposta.

Referências

- Barisic, A., Amaral, V., and Goulão, M. (2018a). Usability driven dsl development with USE-ME. *Computer Languages, Systems Structures*, 51:118 – 157.
- Barisic, A., Cambeiro, J., Amaral, V., Goulão, M., and Mota, T. (2018b). Leveraging teenagers feedback in the development of a domain-specific language: The case of programming low-cost robots. SAC'18, pages 1221—1229, NY, USA. ACM.
- Barisic, A., Monteiro, P., Amaral, V., Goulão, M., and Monteiro, M. (2012). Patterns for evaluating usability of domain-specific languages. In *19th Conference on Pattern Languages of Programs, PLoP'12*, USA. The Hillside Group.
- Bernardino, M., Zorzo, A., and Rodrigues, E. (2016). Canopus: A domain-specific language for modeling performance testing. In *IEEE International Conference on Software Testing, Verification and Validation, ICST'16*, pages 157–167.
- Foundation, E. (2018). Eclipse Process Framework Project - EPF Composer.
- Fowler, M. (2010). *Domain Specific Languages*. Addison-Wesley, 1st edition.
- Hoffmann, B., Chalmers, K., Urquhart, N., and Guckert, M. (2019). Athos - a model driven approach to describe and solve optimisation problems: An application to the vehicle routing problem with time windows. NY, USA. ACM.
- Mernik, M., Heering, J., and Sloane, A. M. (2005). When and How to Develop Domain-specific Languages. *ACM Computing Surveys*, 37(4):316–344.

- Msosa, Y. J. (2019). FCIG grammar evaluation: A usability assessment of clinical guideline modelling constructs. In *IEEE Symposium on Computers and Communications, ISCC'19*, pages 1141–1146.
- Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann, San Francisco, USA.
- OMG - Object Management Group (2018). Software and Systems Process Engineering Metamodel Specification - SPEM.
- Poltronieri, I., Pedroso, A. C., Zorzo, A. F., Bernardino, M., and de Borba Campos, M. (2021). Is usability evaluation of DSL still a trending topic? In *23rd International Conference Human-Computer Interactions, Part I*, volume 12762 of *Lecture Notes in Computer Science*, pages 299–317. Springer.
- Poltronieri, I., Zorzo, A., and Bernardino, M. (2019). Uma proposta de processo de avaliação de usabilidade para dsls. In *III Escola Regional de Engenharia de Software*, pages 127–136, Porto Alegre, RS, Brasil. SBC.
- Poltronieri, I., Zorzo, A. F., Bernardino, M., and Campos, M. B. (2018). Usa-DSL: Usability evaluation framework for domain-specific languages. In *33rd Annual ACM Symposium on Applied Computing, SAC'18*, pages 2013—2021, NY, USA. ACM.
- Pressman, R. (2010). *Software Engineering: A Practitioner's Approach*. McGraw-Hill.
- Rodrigues, I. P. (2021). *Usa-DSL Process: A Usability Evaluation Process for Domain-Specific Languages*. PhD thesis, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, Brasil.
- Rodrigues, I. P., Zorzo, A. F., and Bernardino, M. (2019). Uma proposta de processo de avaliação de usabilidade para dsls. In *Anais da III Escola Regional de Engenharia de Software*, pages 127–136, Porto Alegre, RS, Brasil. SBC.
- Sommerville, I. (2010). *Software Engineering*. Addison-Wesley, USA, 9th edition.
- Stone, D., Jarrett, C., Woodroffe, M., and Minocha, S. (2005). *User Interface Design and Evaluation*. Interactive Technologies. Elsevier Science.