16º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2025

MIGRAÇÃO DE AMBIENTES XTEXT DO ECLIPSE PARA A WEB: UM ESTUDO DE MELHORIA E USABILIDADE

GABRIEL V. ROSIN[1], JULIANO Z. BLANCO[2]

[1]Graduando em Engenharia de Computação, Bolsista PACTec, IFSP, Campus Piracicaba, gabriel.rosin@aluno.ifsp.edu.br.
[2]Professor Doutor do IFSP, Campus Piracicaba, julianoblanco@ifsp.edu.br.

Área de conhecimento (Tabela CNPq): 1.03.03.04-9 Sistemas de Informação

RESUMO: Este trabalho tem como objetivo o desenvolvimento de um protótipo para viabilizar o uso de uma aplicação de geração de código (Blanco; Lucrédio, 2021), originalmente dependente do ambiente de desenvolvimento Eclipse, por meio de uma interface web. A aplicação, baseada na tecnologia Xtext, permite a criação de código em linguagens de programação, como Java, a partir de uma Linguagem de Domínio Específico (DSL). O principal objetivo foi desacoplar a funcionalidade de geração de código de seu ambiente nativo, tornando-a mais acessível e de fácil utilização. A metodologia adotada envolveu a exportação do gerador Xtext como um arquivo JAR executável, a criação de um serviço de *backend* com Python e o *framework* Flask para orquestrar a execução, e o desenvolvimento de uma interface de frontend com HTML, CSS e JavaScript para interação do usuário. Os resultados demonstram a viabilidade da abordagem, com um protótipo funcional capaz de receber um código na DSL, processá-lo no backend e retornar o código Java correspondente na interface web, validando a hipótese de que é possível simplificar o acesso à ferramenta.

PALAVRAS-CHAVE: protótipo; xtext; geração de código; linguagem de domínio específico; desacoplamento;

MIGRATION OF XTEXT ENVIRONMENTS FROM ECLIPSE TO THE WEB: A USABILITY IMPROVEMENT STUDY

ABSTRACT: This work aims to develop a prototype to enable the use of a code generation application (Blanco; Lucrédio, 2021), originally dependent on the Eclipse development environment, through a web interface. The application, based on Xtext technology, allows the creation of source code in programming languages such as Java from a Domain-Specific Language (DSL). The main goal was to decouple the code generation functionality from its native environment, making it more accessible and user-friendly. The adopted methodology involved exporting the Xtext generator as an executable JAR file, creating a backend service using Python and the Flask framework to orchestrate execution, and developing a frontend interface with HTML, CSS, and JavaScript for user interaction. The results demonstrate the feasibility of the approach, with a functional prototype capable of receiving code in the DSL, processing it in the backend, and returning the corresponding Java code in the web interface, thus validating the hypothesis that it is possible to simplify access to the tool.

KEYWORDS: prototype; xtext; code generation; domain-specific language; decoupling.

INTRODUÇÃO

A criação de Linguagens de Domínio Específico (DSLs) é relevante na engenharia de software, pois aumenta a produtividade e facilita a resolução de problemas específicos. O *framework* Xtext é utilizado nesse contexto, permitindo o desenvolvimento de DSLs integradas ao Eclipse IDE (Zhang et al., 2025) . Seu ambiente de execução interpreta a gramática definida, gera código-fonte e oferece recursos como destaque de sintaxe, autocompletar e validação.

Apesar das vantagens, aplicações desenvolvidas com Xtext apresentam uma limitação: funcionam exclusivamente no ambiente Eclipse. Isso impõe barreiras de usabilidade, exigindo que o usuário instale, configure e conheça a IDE, além de importar projetos e executar compilações. Essa dependência reduz o alcance e a facilidade de uso (Blanco; Lucrédio, 2021).

O presente trabalho justifica-se pela necessidade de democratizar o acesso às soluções criadas com Xtext, eliminando a obrigatoriedade do uso do Eclipse. Acessando via navegador, aplicações em DSL podem beneficiar estudantes, pesquisadores e profissionais, sem exigir conhecimentos avançados.

A proposta inicial é verificar a viabilidade técnica da migração do ambiente Eclipse para a Web, por meio de um experimento empírico que comprove a equivalência funcional entre ambos os contextos. Em seguida, pretende-se consolidar e documentar o processo de migração, visando a adoção de uma abordagem multiplataforma (Blanco; Lucrédio, 2021). Essa abordagem busca gerar código para Android, iOS e Web a partir de uma linguagem única, a GPL (General Purpose Language), aumentando a produtividade no desenvolvimento de aplicações.

O objetivo geral é criar e validar um protótipo que permita o uso do gerador Xtext via interface web, ampliando o público e tornando a ferramenta independente do Eclipse.

MATERIAL E MÉTODOS

Para alcançar o objetivo proposto, o modelo de prototipação utilizado foi dividido em três frentes principais que se comunicam para formar a solução. Essa solução é baseada em um modelo cliente-servidor (Kurose; Ross, 2010), onde a interface Web atua como cliente, solicitando requisições, e um programa em Python atua como servidor, que está sempre ativo, recebendo, processando em retornando requisições do cliente, promovendo a separação de responsabilidades. As tecnologias usadas foram:

1. **Xtext e Xtend:** Utilizados no ambiente Eclipse para definir a gramática da DSL de teste e para implementar a lógica do gerador de código que traduz a sintaxe da DSL para código Java, juntamente com a exportação do projeto como um arquivo JAR executável, para permitir sua execução fora do ambiente Eclipse.
2. **Python e Flask:** O *backend* da aplicação foi desenvolvido em Python. Foi escolhido o *micro framework* Flask (Grinberg, 2018) devido à sua simplicidade e agilidade na criação de APIs web. O papel do *backend* é receber as requisições da interface do usuário, salvar o código DSL de entrada em um arquivo temporário. Em seguida, executar o arquivo java (JAR), que contém a aplicação compilada, via linha de comando e retornar os códigos correspondentes ao DSL, transpilados para a linguagem Java, como resultado para o cliente.
3. **HTML, CSS e JavaScript:** Para o frontend (cliente), foi desenvolvida uma página web estática utilizando tecnologias padrão. O HTML estrutura a página com áreas de texto para entrada e saída, o CSS provê uma estilização básica e o JavaScript é responsável pela interatividade, capturando a entrada do usuário e realizando a comunicação assíncrona com o *backend*.
4. **Java:** A linguagem Java foi fundamental para o desacoplamento da ferramenta do Eclipse. Foi desenvolvida uma classe de inicialização que atua como ponto de entrada para o arquivo JAR

executável. Essa classe é responsável por configurar o ambiente Xtext de forma autônoma, carregar o arquivo DSL de entrada, invocar o processo de geração de código e salvar os artefatos resultantes em disco, permitindo que toda a lógica seja executada via linha de comando.

A validação do protótipo foi conduzida por meio de um estudo empírico exploratório, cujo objetivo principal foi verificar a possibilidade da migração das funcionalidades do Xtext do ambiente Eclipse para um ambiente Web. Os testes de validação foram especificados considerando a correta transformação de trechos de código fonte em DSL para código Java. Para isso, foram utilizados exemplos oficiais disponibilizados na documentação do Xtext, assegurando que os casos testados correspondessem a situações reais e consolidadas no uso da tecnologia.

RESULTADOS E DISCUSSÃO

O principal resultado deste trabalho é um protótipo funcional que valida a arquitetura proposta. A aplicação permite que um usuário, utilizando apenas um navegador web, escreva um código em uma DSL customizada e obtenha o código Java gerado sem a necessidade de interação com o ambiente Eclipse.

FIGURA 1. Interface do protótipo demonstrando o campo de entrada em DSL (esquerda) e o código Java resultante (direita) após a geração.

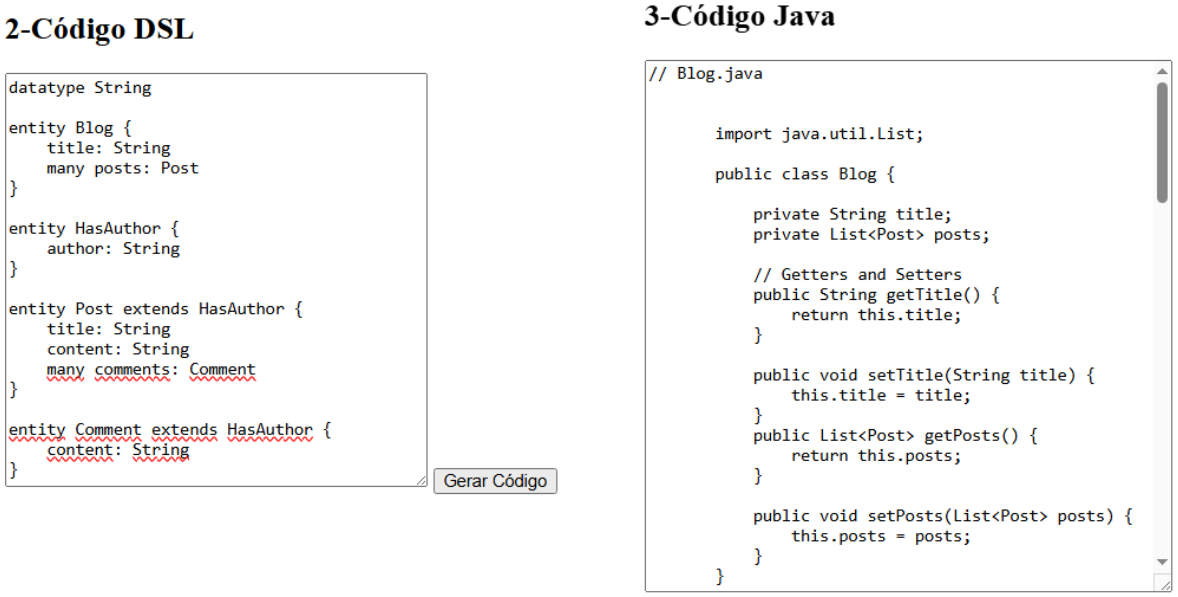
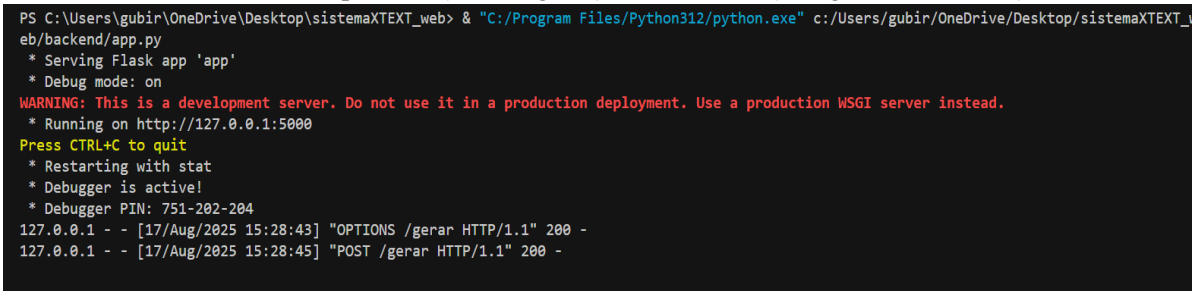


FIGURA 2. Log do servidor Flask no terminal confirmando o recebimento de uma requisição (POST /gerar) com sucesso (código de status 200).



A execução dos testes ocorreu em ambiente web, acessado diretamente por navegador, sem a necessidade de instalação do Eclipse. Os resultados obtidos foram comparados com os resultados

conhecidos no ambiente original, confirmando a equivalência funcional. A validação foi realizada pelo autor do trabalho e pelo orientador, que conseguiram executar os casos de teste via navegador e confirmar a geração correta do código. Por trás do navegador web, havia o servidor, desenvolvido em Python, em execução, para processar as requisições, tendo funcionado corretamente durante os testes.

Por tratar-se de um protótipo em estágio inicial, a amostra de participantes foi suficiente para confirmar a hipótese central de viabilidade técnica. Estudos com amostras mais amplas e análises de usabilidade mais profundas são previstos como trabalhos futuros.

Entre os benefícios observados, destacam-se: a eliminação da dependência do Eclipse e a facilidade de acesso via navegador. Como limitações, apontam-se a simplicidade da interface inicial e a necessidade de execução local do JAR, que pode impactar a escalabilidade em cenários mais complexos.

CONCLUSÕES

Este trabalho demonstrou com sucesso a concepção e implementação de um protótipo para expor as funcionalidades de um gerador de código Xtext por meio de uma interface web. O objetivo de verificar a possibilidade de desacoplar a ferramenta do ambiente de desenvolvimento Eclipse e implementá-la na web, foi plenamente alcançado.

Como trabalhos futuros, sugere-se a efetiva implementação de uma interface orientada para o usuário final, com o Monaco Editor (Microsoft, 2025) para uma experiência de edição avançada e mais profissional, expansão do *backend* para suportar múltiplas gramáticas e geradores, além de uma exploração com a linguagem C# em busca de uma maior performance, transformando o protótipo em uma plataforma de geração de código mais genérica e robusta.

Ainda como trabalho futuro, pretende-se aplicar as técnicas de migração desenvolvidas neste estudo à abordagem multiplataforma (Blanco; Lucrédio, 2021), ampliando a usabilidade dessa abordagem e contribuindo para o compartilhamento de código entre os usuários, com o objetivo de aumentar a produtividade.

CONTRIBUIÇÕES DOS AUTORES

Gabriel Vendrame Rosin contribuiu com a conceituação da arquitetura web, desenvolvimento da metodologia, implementação do software e redação do rascunho original. Juliano Zanuzzio Blanco atuou na supervisão do projeto, orientação, validação dos resultados e na revisão crítica do manuscrito. Todos os autores leram e aprovaram a versão final submetida.

AGRADECIMENTOS

Agradecemos ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP *campus* Piracicaba) por fornecer a estrutura adequada para a realização da pesquisa e desenvolvimento do trabalho, e ao Programa de Apoio à Ciência e Tecnologia (PACTec), gerido pela FAI UFSCar, pelo apoio financeiro por meio de bolsa de pesquisa.

REFERÊNCIAS

BLANCO, J. Z.; LUCRÉDIO, D. A holistic approach for cross-platform software development. Journal of Systems and Software 179 (2021): 110985.

ECLIPSE FOUNDATION. Xtext Documentation. 2025. Disponível em: https://eclipse.dev/Xtext/documentation/index.html . Acesso em 15 de julho de 2025.

EFFTINGE, S.; BEHRENS, C. et al. Xtext: implement your language faster than the quick and dirty way. ACM SIGPLAN Notices, v. 46, n. 6, p. 109–119, 2011. Disponível em: https://dl.acm.org/doi/10.1145/1869542.1869625. Acesso em: 15 ago. 2025.

GRINBERG, M. Flask Web Development: Developing Web Applications with Python. 2. ed. Sebastopol: O'Reilly Media, 2018.

KUROSE, J. F.; ROSS, K. W. Redes de computadores e a Internet: uma abordagem top-down. 5. ed. São Paulo: Addison Wesley, 2010.

MICROSOFT. Monaco Editor. 2025. GitHub repository. Disponível em: https://github.com/microsoft/monaco-editor. Acesso em: 14 jul. 2025.

PYTHON SOFTWARE FOUNDATION. Python 3.12.4 documentation. 2025. Disponível em: https://docs.python.org/3 . Acesso em: 15 jul. 2025.

THE PALLETS PROJECTS. Flask Documentation. 2025. Disponível em: https://flask.palletsprojects.com . Acesso em: 15 jul. 2025.

ZHANG, W.; STRÜBER, D.; HEBIG, R. Development and Evolution of Xtext-based DSLs on GitHub: An Empirical Investigation. 2025. Disponível em: https://arxiv.org/abs/2501.19222. Acesso em: 15 ago. 2025.