

A era digital e suas implicações sociais: Desafios e contribuições

APLICAÇÃO WEB GERADORA DE CLASSES E CÓDIGOS SQL A PARTIR DE UM DIAGRAMA DE BANCO DE DADOS

Miguel Brito Barbosa¹, Henrique Curtis Correa Marques¹, Rodrigo Lopes
Marques¹, Hélio Lourenço Esperidião Ferreira.

Colégio Univap Centro – Colégio Técnico Antônio Teixeira Fernandes, Rua Paraibuna, 75 – Jardim
São Dimas – São José dos Campos – SP, 12245-720 miguelbrito2005@gmail.com
curtishenrique10@gmail.com, rodrigolopes.m2005@gmail.com, helioesperidiao@gmail.com.

Resumo - A produtividade e qualidade durante o processo de produção de software é um tópico altamente discutido na área de desenvolvimento, a redução de custos e o resultado a curto prazo é algo altamente procurado pelas empresas de tecnologia. Genericamente, o desenvolvimento de software apresenta algumas fases, uma das principais é a modelagem de Banco de Dados, o qual toda regra de negócio será baseada. Logo após a construção do Banco, linguagens de programação são utilizadas para construir Classes que se conectarão ao Banco de Dados, fazendo o intermédio entre o software e o Banco. Diante desse cenário, foi desenvolvido uma aplicação web geradora de Classes e códigos SQL a partir de um diagrama de Banco de Dados. Onde o usuário, ao se cadastrar no sistema, terá a possibilidade de criar projetos, construir diagramas relacionais em uma interface moderna, e a partir desse diagrama construído gerar códigos SQL baseados na sua construção e Classes de Conexão em diversas linguagens de programação. Agilizando o processo de criação do Banco de Dados e de criação das Classes de Entidades e Conexão com o Banco do projeto.

Palavras-chave: Geração de Código. Diagrama Relacional. Banco de Dados. SQL. Classes.

Curso: Técnico em Informática.

Introdução

A qualidade de software é uma preocupação atual, com foco na redução de tempo e custos no desenvolvimento. O uso de ferramentas de geração de código automático é considerado essencial para evitar perdas de tempo ao implementar atividades rotineiras e reduzir a possibilidade de erros (MALCOLM, 2001). A busca por maior produtividade e eficiência no desenvolvimento de sistemas é constante, com o uso de técnicas, padrões e ferramentas como RAD (*Rapid Application Development*) (AGARWAL *et al.*, 2000).

A geração de código vem sendo cada vez mais notório nos últimos anos, pois minimiza o problema do retrabalho e adicionar qualidade ao desenvolvimento do software através de uma arquitetura de software pré-definida, algumas das ferramentas que encontramos são JAVA 2 Enterprise Edition (J2EE) e Microsoft .NET, que auxilia no desenvolvimento de um software. A geração de código automático é basicamente um software que gera outros programas (HERRINGTON 2003). A modelagem de dados e o CRUD (*CREATE, READ, UPDATE, DELETE*) são fundamentais nesse processo (HELLER, 2007).

A preocupação com a qualidade de software impulsiona a busca por redução de tempo e custos no desenvolvimento (MALCOLM, 2001). O uso de ferramentas de geração de código automático é essencial para maior produtividade e eficiência (AGARWAL *et al.*, 2000). Isso visa melhorar a velocidade e produtividade na construção de aplicações que utilizem bancos de dados relacionais (MARTINS FILHO, 2017). O objetivo do projeto é criar uma ferramenta web que automatize o processo de modelagem do Banco de Dados, criação do banco em SQL e a criação das classes que acessarão o banco para realização das operações CRUD.

Metodologia

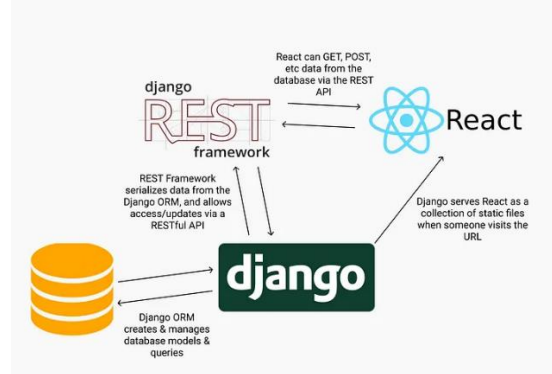
Neste trabalho, foram utilizadas duas linguagens de programação: TypeScript para o desenvolvimento front-end e Python para o back-end. TypeScript é uma extensão do JavaScript que visa suprir suas deficiências, agregando recursos como sistema de módulos, classes e tipagem estática (BIERMAN; Abadi; Torgersen, 2014). Python, por sua vez, é uma linguagem versátil e amplamente utilizada em várias áreas da computação como inteligência artificial, banco de dados, animação 3D etc. (MENEZES, 2010). Além das linguagens de programação, foram empregadas algumas ferramentas e

A era digital e suas implicações sociais: Desafios e contribuições

frameworks para a construção do projeto. O Next.js, um framework React, foi usado para melhorar a experiência do desenvolvedor e criar aplicações web rápidas, com suporte a SSR (Server-side Rendering) e SSG (Static Site Generation) (Souza; Lima, 2022).

Já o Django REST Framework foi escolhido para o back-end, fornecendo um kit de ferramentas poderoso e flexível para construir APIs REST da web (Santiago, et al, 2020). A integração entre as ferramentas pode ser vista na Figura 1.

Figura 1 – Diagrama do funcionamento do framework Django



Fonte: post publicado no site Medium por Bennet Garner, 17 de maio de 2019.

Para a estilização do projeto, foram utilizados os frameworks Bootstrap 5 e Tailwind CSS. O Bootstrap oferece componentes HTML, CSS e JavaScript que tornam o desenvolvimento mais rápido e prático (Santiago; et al, 2020). Já o Tailwind CSS permite estilizar o site com várias classes personalizadas (Ferreira, B. S. 2021).

As bibliotecas também desempenharam um papel fundamental no desenvolvimento do projeto. A biblioteca React Flow permitiu a criação de interfaces de usuário interativas com base em Nodes (Webkid. 2021). O React CodeMirror ofereceu um editor de texto e código completo (Documentação Oficial do CodeMirror, CodeMirror, 2018). A biblioteca Radix UI foi utilizada para criar componentes de interface do usuário de baixo nível, com foco em acessibilidade e personalização (WorkOS, 2020).

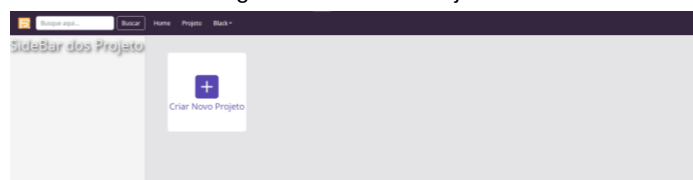
Para o banco de dados, foi escolhido o PostgreSQL, um poderoso sistema gerenciador de banco de dados objeto-relacional de código aberto (Carvalho, 2017). O Docker também foi utilizado para empacotar a aplicação e suas dependências, facilitando a distribuição e replicação em diferentes ambientes (Gomes, 2019).

Resultados

Para criar ou gerenciar seus projetos, o usuário após ter efetuado o login terá acesso a home do sistema, onde seus projetos já criados serão exibidos e a possibilidade de criar outros estará disponível.

A Figura 2 apresenta a interface de gestão de projetos onde o usuário possui acesso a projetos criados anteriormente.

Figura 2 – Tela de Projetos



Fonte: Autores

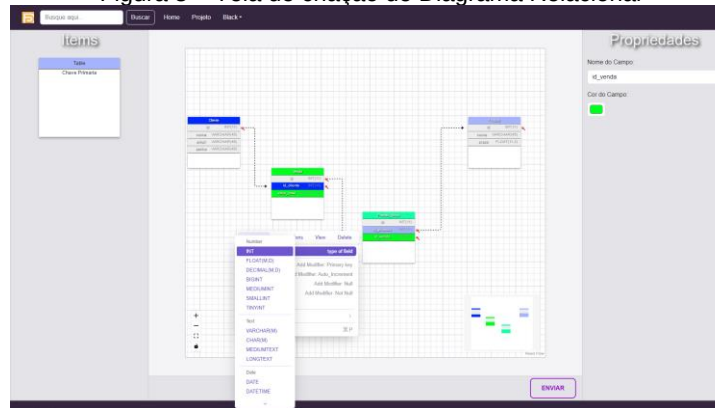
Utilizando o conjunto de tecnologias apresentadas, a interface de criação de Diagramas foi desenvolvida. Através do sistema de Drag and Drop é possível criar tabelas, personaliza-las, adicionar campos, atribuir modificadores e conectá-los, a Figura 3 apresenta um diagrama construído pelo usuário. Ao final da diagramação, o usuário poderá enviar seu diagrama e transformá-lo em código SQL.

Após a criação do Diagrama Relacional do projeto, o usuário poderá gerar códigos SQL e Classes em diversas linguagens de programação. Acessando a tela de geração de código, o SQL proveniente do diagrama criado anteriormente estará disponível para visualização e edição. Nessa mesma

A era digital e suas implicações sociais: Desafios e contribuições

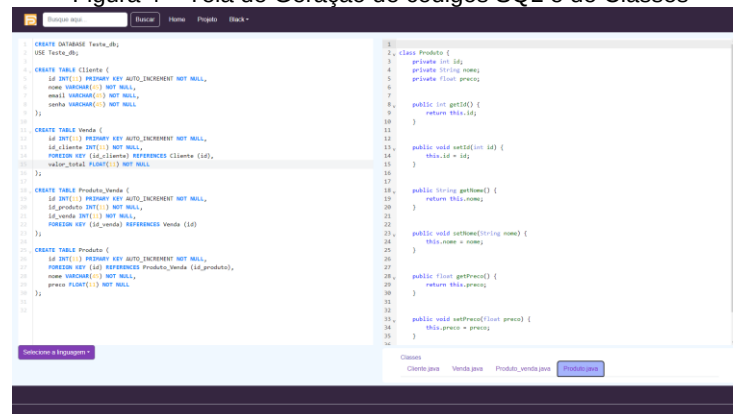
interface, também estará presente o botão de criação de classes, onde basta escolher a linguagem que será utilizada e código será gerado, os códigos de banco de dados e uma classe gerada podem ser apreciadas na Figura 4.

Figura 3 – Tela de criação de Diagrama Relacional



Fonte: Autores

Figura 4 – Tela de Geração de códigos SQL e de Classes



Fonte: Autores

Discussão

A interface desenvolvida é intuitiva e eficiente, assim como a interface de criação de formulários do sistema .NET da Microsoft, onde é possível arrastar componentes para a tela para gerar códigos em C#, o projeto atende as necessidades de facilitar o desenvolvimento de forma rápida e competente, dando, analogamente como no sistema da Microsoft, a possibilidade do usuário simplesmente ao arrastar um componente representativo de uma tabela, gerar códigos SQL e Classes em diversas linguagens de programação. Outra tecnologia que se assemelha ao projeto é a ferramenta e-Gen Developer, sendo um ambiente RAD que proporciona o desenvolvimento rápido de aplicações para WEB. Foi criado para aumentar a produtividade, reduzir os custos de manutenção e reduzir a curva de aprendizagem, tornando o desenvolvimento mais padronizado e limpo, característica também presente no projeto desenvolvido.

Conclusão

O sistema desenvolvido permitiu que os usuários criassem diagramas de banco de dados de forma interativa e, a partir desses diagramas, gerassem códigos SQL e Classes de conexão com o banco em diversas linguagens de programação. Os resultados obtidos com a implementação do projeto foram satisfatórios, demonstrando a eficiência das tecnologias utilizadas.

As expectativas foram atendidas, o sistema permite o cadastro de usuários, onde eles a partir do login, poderão gerar código SQL com base em suas criações de diagramas de banco de dados. Validando a relevância do projeto e a escolha das tecnologias empregadas, demonstrando a capacidade do sistema de fornecer uma solução ágil e eficiente para o desenvolvimento de softwares.

A era digital e suas implicações sociais: Desafios e contribuições

O projeto demonstra relevância no cenário de desenvolvimento de software, automatizando etapas iniciais como: estruturação do diagrama relacional, geração de código SQL e geração da estrutura básica das Classes de entidade e Conexão. Com integração de Typescript com Python, Next.js com Django, o resultado é uma aplicação coesa, ágil e responsiva. Bibliotecas como React Flow, React CodeMirror e Radix UI contribuíram para a criação de interfaces interativas e personalizadas, ampliando a usabilidade do sistema.

Referências

AGARWAL, R. et al. **Risks of Rapid Application Development**. [S. l.: s. n.], 2000.

BIERMAN, G., Abadi, M., Torgersen, M. (2014). **Understanding TypeScript**. In: Jones, R. (eds) ECOOP 2014 – Object-Oriented Programming. ECOOP 2014. Lecture Notes in Computer Science, vol 8586. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44202-9_11

CARVALHO, Vinícius. **PostgreSQL: Banco de dados para aplicações web modernas**. 1.ed. São Paulo: Editora Casa do Código, Alura, 2017.

LOPES, Jean. **PHP ou TypeScript: uma comparação de duas linguagens para web pelas suas características**. 2016. Dissertação (Trabalho de Conclusão do Curso Superior de Tecnologia em Sistemas Para Internet) – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Porto Alegre, 2016.

FERREIRA, Brenda Sotero. **Framework Laravel: um estudo de caso full stack development**. 2021. 99 f. Monografia (Graduação em Ciência da Computação) - Instituto de Ciências Exatas e Biológicas, Universidade Federal de Ouro Preto, Ouro Preto, 2021.

GOMES, Rafael. **Docker para desenvolvedores**. 1.ed. Salvador: Leanpub, 2019.

HELLER, M. **REST and CRUD: the impedance mismatch web application framework**. Framingham, MA, EUA, 2007.

HERRINGTON, Jack. **Code generation in action**. [S.l.]: Manning Publications Co., 2003. 400 p. ISBN 1930110979.

MALCOLM Eva. **Requirements acquisition for rapid applications development, Information & Management, Volume 39, Issue 2**. 2001. Pages 101-107, ISSN 0378-7206.

MARTINS FILHO, Dinarte Alves. **FASTSF: uma ferramenta de geração de código para o incremento da produtividade no desenvolvimento de sistemas de informação web**. 2017. 102f. Dissertação (Mestrado Profissional em Engenharia de Software) - Instituto Metrópole Digital, Universidade Federal do Rio Grande do Norte, Natal, 2017.

MENEZES C. NEY, Nilo. **Introdução à programação com Python**. 1.ed. São Paulo: Novatec Editora Ltda, 2010.

SANTIAGO, C. P.; VERAS, N. L.; de ARAGÃO, A. P.; CARVALHO, D. A.; AMARAL, L. A. **Desenvolvimento de sistemas Web orientado a reuso com Python, Django e Bootstrap**. 2020.

SOUZA, F. M. C.; LIMA, E. C. S.; CARIDADE, E. R. de S. **Criando sistema escalável de agendamentos utilizando Typescript com Nest.js no backend e Next.js no frontend**. 2022.

WEBKID. ReactFlow. **Documentação oficial**. 2021.

WORKOS. Radix UI. **Documentação oficial**. 2020.