

Colossus: Desenvolvimento de Ferramenta Case para Apoio a Criação de Sistemas Web Utilizando Frameworks PHP como Modelo e Banco de Dados MySQL

Saulo Fernandes Antônio da Costa¹, Natasha Sophie Pereira²

^{1,2}Bacharelado em Engenharia de Computação – Centro Universitário de Anápolis (UniEVANGÉLICA) – Anápolis - GO

¹saulofernandes.a.costa@hotmail.com, ²natasha.sophie@gmail.com

Resumo. Este trabalho tem como objetivo a construção de uma ferramenta Computer Aided Software Engineering (CASE) para auxiliar no desenvolvimento de aplicações para a plataforma web, utilizando a linguagem de programação Hypertext Preprocessor (PHP) e o banco de dados My Structured Query Language (MySQL). A ferramenta CASE consiste em uma plataforma para que o desenvolvedor possa construir e gerar boa parte de seu projeto utilizando como modelo frameworks já existentes, e visando a redução no tempo de produção e robustez do projeto. Através do desenvolvimento de um sistema web utilizando linguagem PHP e Banco de Dados MySQL, foi possível construir a Ferramenta CASE Colossus, que é uma ferramenta intuitiva e autoexplicativa que busca agilidade no processo de desenvolvimento de projetos, proporcionando ganho de tempo e aprimoramento da experiência dos desenvolvedores.

1. Introdução

A ferramenta CASE¹ Colossus tem foco na redução do tempo de desenvolvimento, sendo assim, ela é uma plataforma para construção de projetos, que gera o código fonte do projeto de acordo com as especificações do desenvolvedor. O desenvolvedor poderá selecionar as opções mais adequadas para cada tipo de métodos *Create, Read, Update e Delete* (CRUD) que deseja inserir em sua aplicação, e o código será gerado pela plataforma com base em *frameworks*² existentes que utilizem a linguagem de programação PHP (*Hypertext Preprocessor*), e *Structured Query Language* (SQL) para banco de dados MySQL (*My Structured Query Language*), mantendo, assim, a qualidade do projeto através de padrões de desenvolvimento.

Com a crescente demanda, e projetos cada vez maiores e mais complexos, surge a necessidade de se desenvolver de forma rápida, porém, sem perder a qualidade. Empresas de software estão mais exigentes nos quesitos que envolvem a qualidade do projeto, padrões de programação, segurança de sistemas, e aumento da produtividade. Dessa forma, o uso de algum tipo de framework é um requisito fundamental para desenvolvedores que pretendem se destacar no mercado de trabalho.

1.1. Objetivos da Pesquisa

1.1.1. Objetivo Geral

Desenvolver uma ferramenta CASE para auxiliar na criação de sistemas web com geração de códigos através de interface gráfica, em linguagem de programação PHP e SQL para banco de dados MySQL.

¹ Ferramenta CASE, do inglês *Computer-Aided Software Engineering* (Engenharia de software assistida por computador), é o “software usado para apoiar as atividades de processo de software” [SOMMERVILLE 2007].

² Um framework é um conjunto de classes que tem um formato abstrato e são projetados para possibilitar a máxima reutilização de código [MATTSSON 1996].

1.1.2. *Objetivos Específicos*

- Levantamento bibliográfico sobre: ferramentas CASE; frameworks para a linguagem PHP; desenvolvimento de sistema web utilizando linguagem PHP e Banco de Dados MySQL;
- Desenvolvimento da ferramenta CASE para criação de sistemas web gerando códigos em linguagem PHP e SQL para banco de dados MySQL;

1.2. *Justificativa*

Devido à enorme quantidade de projetos que um programador desenvolve simultaneamente e a diversos programadores em um único projeto, surge a necessidade de padronização e redução do tempo de codificação. Alguns códigos fonte se tornam repetitivos, dificultando assim a manutenção posterior. Segundo Minetto (2007),

Em uma aplicação que tenha de manipular dados vindos de uma tabela na base de dados, as operações de inclusão, exclusão e alteração são praticamente iguais para todas as tabelas envolvidas. Não teria sentido repetir o esforço para desenvolver esse código-fonte várias vezes, e a geração dessas funções poderia ser automatizada por alguma ferramenta contida no framework [MINETTO 2007, p. 18].

A utilização de um *framework* significa otimização de tempo e qualidade de projeto por dispor de módulos prontos e padrões bem elaborados. A principal contribuição da ferramenta CASE Colossus é o auxílio na criação de projetos rápidos *Minimum Viable Product* (MVP), e para isso possui uma plataforma que irá gerar todo o código fonte do projeto de acordo com as especificações do desenvolvedor e basear o código fonte a ser gerado em frameworks já existentes, possibilitando assim que o programador faça boa parte do projeto de forma rápida, mesmo sem conhecer todos os padrões de desenvolvimento de uma linguagem de programação ou de um *framework*. Conforme Minetto (2007) “Quando um desenvolvedor começa a estudar um novo framework, ele se depara muitas vezes, com uma forma diferente de programar ou até de pensar um sistema” [MINETTO 2007, p. 17].

Portanto o tempo gasto aprendendo a utilizar um framework na programação pode se tornar um empecilho para o desenvolvimento dos projetos que exigem agilidade. A partir dessa visão, como aumentar a velocidade no desenvolvimento de sistemas web utilizando frameworks em linguagem de programação PHP e banco de dados MySQL, mantendo a qualidade e tornando esta atividade intuitiva e fácil?

2. **Levantamento Teórico**

2.1. *Ferramentas CASE*

As ferramentas CASE podem ser consideradas como ferramentas automatizadas que tem como objetivo auxiliar o desenvolvedor em etapas do desenvolvimento de software. Sua finalidade é automatizar as atividades que muitas vezes são feitas manualmente, reduzindo assim a quantidade de esforço necessário para produzir um software. Segundo Sommerville (2007),

Engenharia de Software Auxiliada por Computador (CASE – *Computer Aided Software Engineering*) é o nome dado ao software usado para apoiar as atividades de processo de software, como engenharia de requisitos, projeto, desenvolvimento de programas e teste. As ferramentas CASE, portanto, incluem editores de diagramas, dicionário de dados, compiladores, *debuggers*, ferramentas de construção de sistemas etc. A tecnologia CASE fornece apoio ao processo de software pela automação de algumas atividades de processo e pelo fornecimento de informações sobre o software que está sendo desenvolvido [SOMMERVILLE 2007, p. 56].

Atualmente o uso de uma ferramenta CASE é vital para o bom funcionamento de uma empresa desenvolvedora de software. De acordo com Pressman (2002), as ferramentas CASE podem oferecer novos modos de olhar a informação de engenharia de software, aperfeiçoando o conhecimento do programador que está desenvolvendo.

2.2. Linguagem de Programação

As linguagens de programação servem como um meio de comunicação entre computadores e desenvolvedores, permitindo que um programador especifique precisamente sobre quais dados o computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas de acordo com várias circunstâncias [WIKIPÉDIA 2017].

No desenvolvimento web é necessária a utilização de uma linguagem do lado do servidor. O PHP é uma das linguagens que podem ser utilizadas do lado do servidor. Para Soares (2013),

Atualmente, existem várias ferramentas para desenvolvimento de sites *web* dinâmicos, tais como Java, *Active Server Pages* (ASP) e tantas outras, mas não se encontra ainda uma ferramenta tão completa, estável e simples, mas ao mesmo tempo potente, como o PHP. O único senão dessa linguagem era sua pouca aderência ao modelo de orientação a objetos (POO), o que foi praticamente sanado na versão 5 do PHP, que tem tudo para ser a versão que vai consolidar o domínio dessa linguagem no desenvolvimento de sites dinâmicos, desde os mais simples sites pessoais até os complexos para gerenciamento de transações entre empresas e portais sofisticados com sistemas especialistas [SOARES 2013, p. 20].

O PHP é uma linguagem de *script open-source* de uso geral, muito utilizada e especialmente guarnecida para o desenvolvimento de aplicações web embutível dentro do *HyperText Markup Language* (HTML) [PHP 2017].

2.3. Framework

Um framework pode ser visto como o esqueleto de uma aplicação. Possui classes pré-definidas e diversas estruturas prontas que são códigos comuns entre vários projetos de software provendo funcionalidades genéricas que podem ser refinadas, se tornando mais específicas, durante a programação. Segundo Minetto (2007),

Um framework de desenvolvimento é uma “base” de onde se pode desenvolver algo maior ou mais específico. É uma coleção de códigos-fonte, classes, funções, técnicas e metodologias que facilitam o desenvolvimento de novos softwares [MINETTO 2007, p. 17].

Ao contrário das bibliotecas, o framework molda a aplicação de forma a permitir configurações específicas, baseadas em padrões de projeto predefinidos, pois são formas já testadas e documentadas de se resolver algum tipo de problema. “A grande maioria dos frameworks existentes baseiam seu funcionamento em um ou mais desses padrões” [MINETTO 2007, p. 19].

Portanto, por mais que se gaste muito tempo aprendendo padrões de desenvolvimento impostos pelos frameworks, as principais finalidades de seu uso devem ser a resolução de problemas, redução no tempo de codificação e redução dos custos de produção. De acordo com [NASH 2003], o principal benefício de um framework é o aumento da velocidade de desenvolvimento. Uma vez que a curva de aprendizado é superada, o tempo necessário para se desenvolver pode se tornar menor se comparado ao tempo necessário para se desenvolver a mesma aplicação sem o uso de um framework. Para isso ele deve ter maior facilidade para a detecção de erros, concentração na abstração da solução de problemas, eficiência na resolução dos problemas, e otimização de recursos.

Atualmente existem vários *frameworks back-end*³ disponíveis, entre eles destaca-se o Laravel, que é um *framework* de desenvolvimento para a linguagem PHP, rápido, gratuito e de código aberto. Possui documentação bem elaborada e intuitiva onde é possível encontrar facilmente tudo que se

³ Back-end, é o nível de programação que envolve banco de dados [CITRUS7 2017].

necessita saber para aprofundar-se nos recursos disponíveis. De acordo com o site DevMedia (2017a), o Laravel é um *Framework* PHP utilizado para o desenvolvimento web, que tem o fluxo baseado na arquitetura *Model, View, Controller* (MVC) e tem como principal característica ajudar a desenvolver aplicações seguras e performáticas de forma rápida, possui o código limpo e simples pois utiliza o padrão PSR-2⁴ e incentiva o uso de boas práticas de programação.

Segundo Gonçalves (2007, p. 141), o MVC é um paradigma de desenvolvimento e design que tenta separar uma aplicação em três camadas distintas. A camada *Model* (modelo), está relacionada a parte da aplicação que implementa a lógica do negócio, outra camada, a *View* (visão) é responsável por exibir os dados ou informações da aplicação e a terceira camada, *Controller* (controle), lida com as requisições dos usuários e coordena a *view* e a *model*, exibindo a interface correta ou executando algum trabalho que a aplicação precisa completar.

O Bootstrap é um *framework front-end*⁵ que auxilia na criação de sites e plataformas *web*, possui código aberto e tecnologia mobile (responsivo) onde o template se adequa automaticamente aos mais diversos tamanhos de telas. O Bootstrap é, atualmente, o principal *framework front-end* para a construção de aplicações web. Entre os seus recursos o de maior destaque é o sistema de grid responsivo, que permite desenvolver com facilidade páginas que se adaptam automaticamente aos diferentes tamanhos de tela [DEV MEDIA 2017b].

2.4. Banco de Dados

Um banco de dados (BD) é uma coleção de dados e informações que se relacionam para criar algum sentido. Conforme Heuser (2009), banco de dados é um conjunto de dados relacionados, que tem o objetivo de servir aos usuários. A elaboração do banco de dados é uma parte essencial do desenvolvimento do projeto, pois será nele que ficarão armazenadas as informações e dados dos usuários. De acordo com o site DevMedia (2017c), para a criação de um Banco de Dados é necessário realizar três etapas fundamentais: projeto conceitual, projeto lógico e modelo físico.

O Projeto Conceitual é a etapa em que os requisitos são especificados. É quando o desenvolvedor compreende como será o banco de dados. Com os requisitos bem definidos, é feito um esquema conceitual que é a descrição formal do banco de dados. Segundo Heuser (2009) modelo conceitual é um modelo de dados abstrato, e a descrição da estrutura de um banco de dados.

O Projeto lógico é o momento onde são criados os modelos internos do banco de dados. É a parte mais detalhada da estrutura do banco de dados. Para Heuser (2009) o modelo lógico representa a estrutura do banco de dados.

O Modelo físico é a última parte do projeto do banco de dados, nessa etapa é utilizada uma linguagem de definição de banco de dados de acordo com o Sistema Gerenciador de Banco de Dados (SGBD) selecionado. Para Elmasri e Navathe (2011, p. 3-4) “um sistema gerenciador de banco de dados SGBD é uma coleção de programas que permite aos usuários criar e manter um banco de dados”.

O MySQL é o sistema de gerenciamento de banco de dados *open source* mais popular atualmente, ele é desenvolvido, distribuído e suportado pela *Oracle Corporation*, e utiliza a linguagem SQL como interface. O MySQL oferece um servidor de banco de dados SQL muito rápido, *multi-threaded*, multiusuário e robusto [MYSQL 2017b]. Utiliza a linguagem SQL, que é a linguagem mais popular para cadastrar, acessar e manipular conteúdo armazenado em um banco de dados. De maneira prática, essa linguagem organiza os dados em um modelo relacional, formando tabelas.

⁴ PSR-2, Guia para reduzir a fricção cognitiva ao escanear código de diferentes autores. Ele faz isso enumerando um conjunto compartilhado de regras e expectativas sobre como formatar o código PHP [PHP-FIG 2017].

⁵ Front-end, É a primeira camada com a qual nos deparamos ao acessarmos um site [CITRUS7 2017].

Para adicionar, acessar e processar dados armazenados em um banco de dados de computador, é necessário um sistema de gerenciamento de banco de dados como o MySQL. Devido ao fato de os computadores serem muito bons em lidar com grandes quantidades de dados, os sistemas de gerenciamento de banco de dados desempenham um papel central na computação, como utilitários autônomos ou como partes de outras aplicações [MYSQL 2017a].

3. Metodologia de Pesquisa

Será desenvolvido um sistema de *software*, do tipo ferramenta CASE para auxiliar no desenvolvimento de sistemas web, visando auxiliar os desenvolvedores de *software* no processo de criação de um sistema *web*. Então, de acordo com Ciribelli (2003), a metodologia de estudo será experimental.

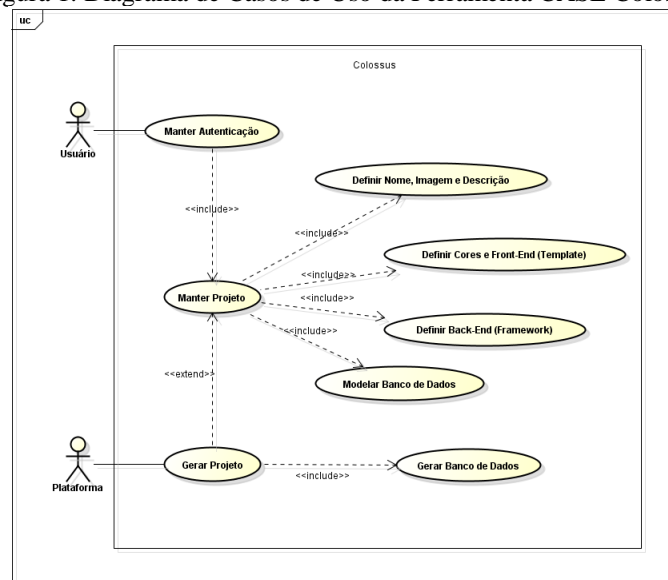
Desta forma, a pesquisa proposta é considerada como exploratória e explicativa [CIRIBELLI 2003; WAZLAWICK 2014], visto que será feito um levantamento teórico baseado em livros, pesquisas, monografias, teses, dissertações e artigos de autores conceituados nas áreas de Desenvolvimento de Sistemas *Web*, banco de dados, *frameworks* e Ferramentas CASE. Também será feita uma pesquisa das técnicas de desenvolvimento de sistemas *web* e banco de dados, visando à geração de sistemas do tipo ferramenta CASE para auxílio no desenvolvimento de sistemas *web* que utilizem algum *framework*.

A ferramenta CASE Colossus será desenvolvida em linguagem de programação PHP, devido ao fato de ser de fácil aprendizagem e ser uma linguagem de *script open source* de uso geral, muito utilizada, e especialmente adequada, para o desenvolvimento *web*, pois pode ser facilmente embutida em meio às tags do HTML [PHP 2017].

Seguirá o paradigma de programação orientada a objetos juntamente com a arquitetura MVC. A arquitetura MVC visa separar as regras e lógicas do negócio da apresentação em si, permitindo maior controle sobre a aplicação e possibilitando uma manutenção isolada. O banco de dados para guardar as informações de usuários e projetos da ferramenta será o MySQL, devido ao fato de ser relacional, *open source*, muito rápido, *multi-threaded*, multiusuário e robusto.

4. Desenvolvimento do Sistema

Figura 1. Diagrama de Casos de Uso da Ferramenta CASE Colossus.

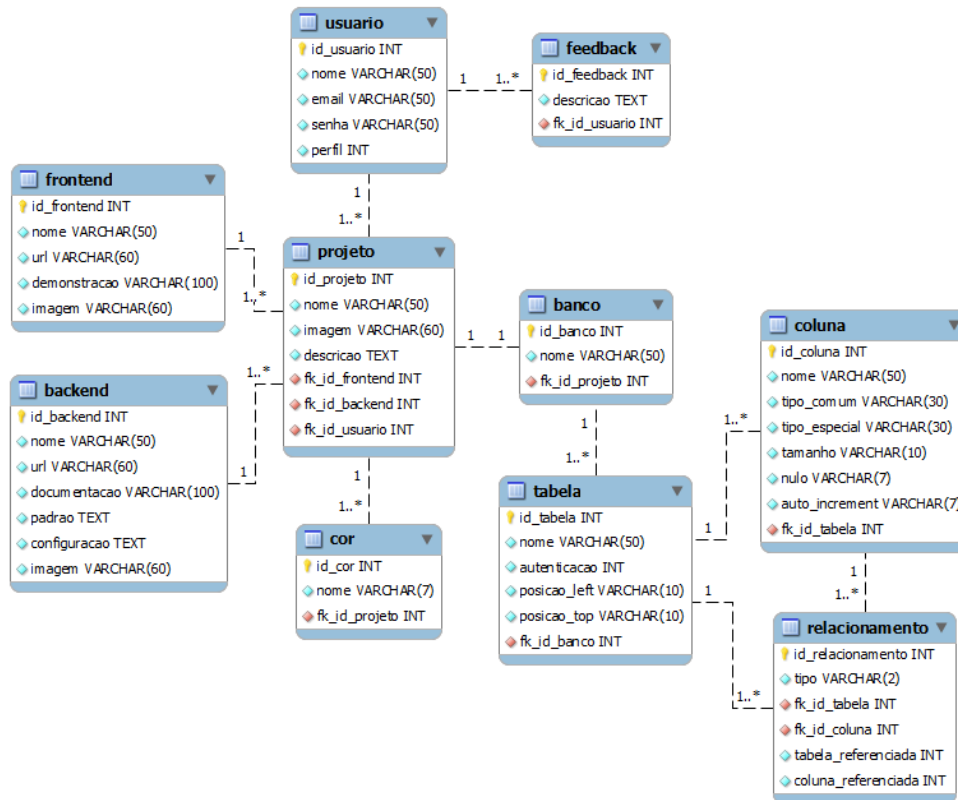


Fonte: Autores.

A ferramenta CASE Colossus possui uma interface para auxiliar o desenvolvedor na especificação do projeto, de forma que, após essa especificação, a plataforma irá gerar o código do projeto e do banco de dados. O diagrama de casos de uso referente ao Colossus pode ser visualizado na Figura 1.

A plataforma irá manter os dados referentes aos usuários e aos projetos cadastrados. Para acessar e utilizar a ferramenta CASE Colossus, o usuário precisa estar pré-cadastrado e autenticado no sistema. A autenticação é necessária para que os projetos desenvolvidos por um usuário não sejam acessados por outros usuários sem permissão. O Colossus mantém as informações de usuários e projetos em um banco de dados que segue as especificações do Modelo de Dados mostrado na Figura 2.

Figura 2. Modelo de Dados da Ferramenta CASE Colossus.



Fonte: Autores.

4.1. Manter Usuário

Inicialmente, um novo usuário deverá se cadastrar na plataforma para utilizar a ferramenta CASE Colossus (Figura 3). A plataforma mantém as seguintes informações do usuário:

- Nome
- E-Mail
- Senha

Estas informações são utilizadas para identificar o usuário que está acessando o sistema e desenvolvendo certo projeto, com a finalidade de que usuários acessem apenas suas informações pessoais e dos projetos que está desenvolvendo.

4.2. Manter Autenticação

Para acessar a ferramenta CASE Colossus, é necessário que o usuário esteja autenticado na plataforma. Para isso, é necessário fornecer as seguintes informações na tela de autenticação (Figura 4):

- E-Mail
- Senha

Figura 3. Tela de Cadastro.

Fonte: Autores.

Figura 4. Tela de Autenticação.

Fonte: Autores.

4.3. Manter Projeto

O usuário, desenvolvedor, irá modelar o projeto através da plataforma, para isso deverá seguir os seguintes passos:

- Definir o nome, imagem e descrição do projeto;
- Definir as Cores e o *Template (Front-End)* que serão utilizados no projeto;
- Definir o *Framework (Back-End)* que será adotado;
- Modelar o Banco de Dados;
- Gerar o código do projeto modelado.

4.3.1. Definir Nome, Imagem e Descrição

No primeiro estágio de modelagem do projeto, o desenvolvedor deverá nomear, escolher uma imagem representativa e descrever o projeto que será desenvolvido.

O nome do projeto deverá conter apenas uma palavra (exemplo: Colheita), com um máximo de 50 caracteres. Este nome será usado pelo gerador de códigos para nomear a pasta do projeto, e também o arquivo SQL e os códigos dos arquivos *front-end*. A Figura 5 mostra o formulário para preenchimento das informações iniciais de criação do projeto.

Figura 5. Tela de Criação de Projetos.

Fonte: Autores.

Após a criação do projeto, as informações de *Layout*, *framework* do projeto, assim como o banco de dados deverão ser definidos pelo desenvolvedor (Figura 6).

Figura 6. Tela de Definição de *Layout*, *Framework* e Banco de Dados.

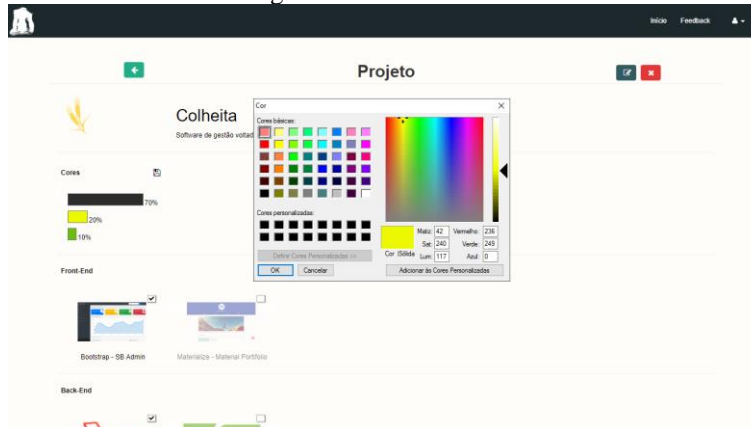
Fonte: Autores.

4.3.2. Definir Cores e Front-End (Template)

Na segunda etapa, o desenvolvedor deverá escolher as cores do tema, e o *template* que serão utilizados para montagem do *Layout* (visual) do projeto que está sendo modelado.

Os projetos gerados pela ferramenta terão três cores padrão para compor o Layout. Estas cores devem ser definidas separadamente pelo desenvolvedor em uma paleta com diversas opções (Figura 7). Estas cores serão utilizadas em todo o projeto, ou seja, no menu, nos botões, nas tabelas e também nos modais.

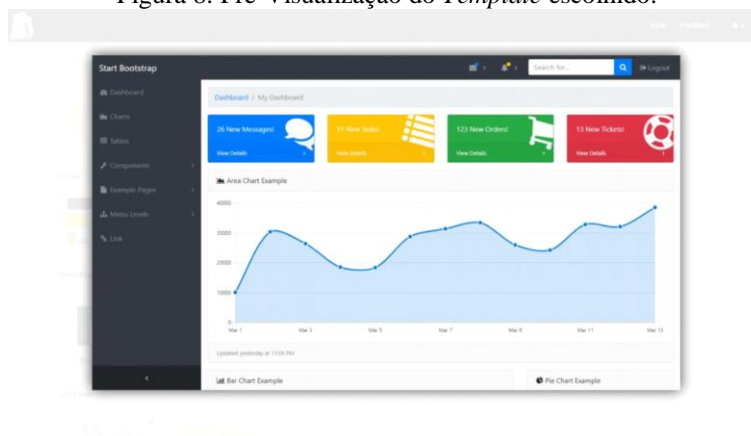
Figura 7. Paleta de Cores.



Fonte: Autores.

A ferramenta CASE Colossus trará opções de *templates* desenvolvidos por *frameworks front-end*, como o *Bootstrap* e o *Materialize*, e o desenvolvedor poderá escolher uma entre as opções disponíveis. Ao selecionar um dos *templates* disponíveis, o usuário poderá visualizar previamente o layout do projeto que está modelando (Figura 8), facilitando, assim, a escolha pelo desenvolvedor. Inicialmente, está disponível na plataforma o *framework front-end Bootstrap*. Posteriormente poderão ser adicionadas outras opções de *template*.

Figura 8. Pré-Visualização do *Template* escolhido.



Fonte: Autores.

4.3.3. Definir Back-End (Framework)

Após a definição das características de Layout do projeto, o desenvolvedor deverá escolher qual *framework back-end* será utilizado como padrão para sua aplicação, podendo escolher uma entre as opções de frameworks disponíveis. Inicialmente, para o a geração de códigos fonte, estará disponível a opção de escolha para o padrão de *framework back-end* Laravel. Posteriormente poderão ser adicionados novos padrões como o Zend3 e outros.

4.3.4. Modelar Banco de Dados

A quarta etapa de definição do projeto que está sendo desenvolvido é a modelagem do Banco de Dados que será utilizado no sistema. Nesse estágio o desenvolvedor deverá modelar seu banco de dados ou importar o script SQL caso já tenha criado utilizando outra ferramenta.

Se o desenvolvedor já possuir um banco de dados prévio, deverá apenas determinar a tipagem especial de cada coluna. Caso contrário, deverá modelar seu BD através da ferramenta CASE Colossus. Poderão ser criadas quantas tabelas forem necessárias para o funcionamento de sua aplicação, relacioná-las, definir as colunas das tabelas e determinar as tipagens comuns e especiais de cada coluna, determinando a opção mais adequada para cada tipo de modulo a ser gerado. O padrão a se utilizar para nomear as tabelas será letra minúscula, sem caracteres especiais ou espaços, nome composto deverá ser separado por sublinhado (*underline*).

Devido ao banco de dados MySQL ser relacional o usuário poderá fazer qualquer tipo de ligação entre as tabelas (1 para 1, 1 para N e N para N). Segundo o site da MySQL (2017a),

Um banco de dados relacional armazena dados em tabelas separadas ao invés de colocar todos os dados em um grande depósito. As estruturas de banco de dados são organizadas em arquivos físicos otimizados para a velocidade. O modelo lógico, com objetos como bancos de dados, tabelas, visualizações, linhas e colunas, oferece um ambiente de programação flexível [MySQL 2017a].

Além das opções de tipagem comum de colunas das tabelas (exemplo: *int*, *float*, *varchar*, *text*, dentre outras), a ferramenta CASE terá a opção de tipagem especial, como por exemplo, *data*, *email*, *telefone*, *password*, *textarea*, CPF, CNPJ, CEP, imagem, url, entre outras. As opções de tipagem permitem que a ferramenta gere códigos e importe bibliotecas de acordo com cada tipo definido para a coluna (Figura 9).

Figura 9. Tela de Criação de Tabela.



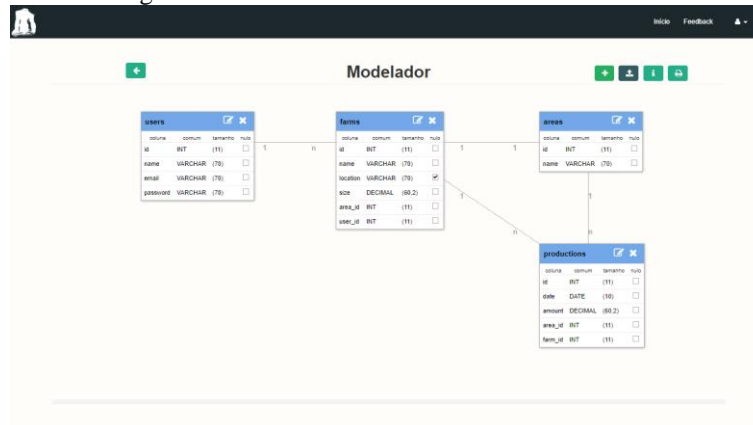
Fonte: Autores.

O modelador de Banco de Dados (Figura 10) foi desenvolvido para ser simples e intuitivo, onde o desenvolvedor poderá modelar seu BD de forma visual em um modelo entidade-relacionamento, adicionando as tabelas, seus campos (colunas) e os relacionamentos entre as tabelas.

4.4. Gerar Projeto

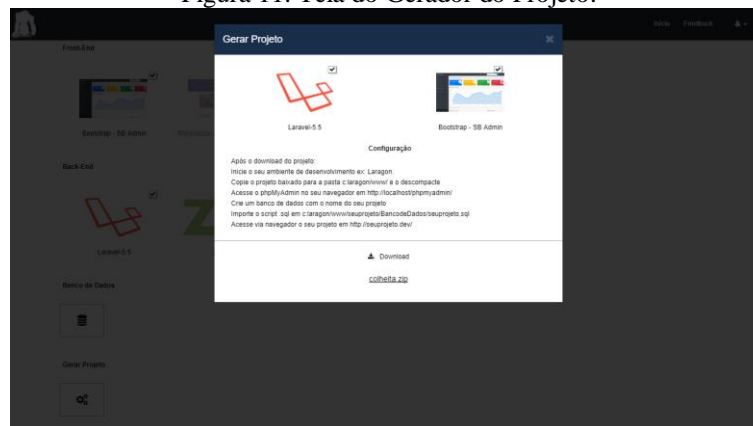
Após o desenvolvedor especificar o *Layout*, *Framework* e o Banco de Dados do projeto, utilizando a interface da ferramenta CASE Colossus, ele deverá solicitar à plataforma que gere os códigos do projeto (Figura 11).

Figura 10. Tela do Modelador do Banco de Dados.



Fonte: Autores.

Figura 11. Tela do Gerador do Projeto.



Fonte: Autores.

O código fonte do projeto será gerado em quatro etapas:

- I. Criação da estrutura de pastas: A criação será baseada no padrão do framework escolhido, esta estrutura de pastas serve para separar as partes do projeto.
- II. Geração do código *front-end*: Nesta etapa, será feita a configuração automática dos *templates* escolhidos e a inserção dos arquivos deste *template* em suas respectivas subpastas.
- III. Geração do script do banco de dados: Além da geração do script do Banco de Dados para MySQL, os códigos gerados serão inseridos em sua respectiva subpasta.
- IV. Geração do código *back-end*: Por último, serão gerados os códigos *back-end*, de acordo com o *framework* escolhido, estes serão inseridos automaticamente em suas respectivas subpastas.

Após a criação e *download* do projeto através plataforma o desenvolvedor deverá configurar o ambiente de desenvolvimento web de acordo com o *framework* escolhido, para testar e programar alguma parte específica caso a plataforma não consiga atender todas as regras de negócio de seu projeto.

5. Considerações finais

A ferramenta Colossus foi desenvolvida e encontra-se disponível para utilização através do link < <http://ferramentacolossus.com.br/> >. O link apresenta uma breve descrição do que é a

ferramenta, com a possibilidade de um usuário comum se cadastrar e começar a utilizar o sistema (Figura 12).



Fonte: Autores.

Colossus é uma ferramenta case desenvolvida para facilitar e agilizar o processo de desenvolvimento de softwares para plataforma web. Ela permite o desenvolvimento de um sistema personalizado, utilizando *frameworks* para definição de *Layout (Front-End)* e de padrões de projeto (*Back-End*). A ferramenta gera os códigos fonte em linguagem PHP e *scripts* SQL para Banco de Dados MySQL.

Com a finalidade de testar o sistema, foi modelado um projeto de nome “Colheita” que pode ser acessado através da plataforma utilizando as seguintes informações para autenticação:

- Usuário: projeto@email.com
- Senha: 123456

A plataforma está disponível para utilização por desenvolvedores e interessados em desenvolver sua própria aplicação. Com a finalidade de melhoria constante, foi incluído uma seção para que os usuários possam dar seu *Feedback* sobre a utilização da aplicação.

Devido ao curto tempo para desenvolvimento deste projeto, não foi possível a implementação de mais *Frameworks Front-End* ou *Back-End*, permanecendo, no momento de conclusão deste Trabalho de Conclusão de Curso, a ferramenta apenas com *Bootstrap* e *Laravel* para *Front-End* e *Back-End*, respectivamente. Futuramente, pretende-se incluir outros padrões de *Layout (Frameworks Front-End)* e, também, outros padrões de projeto (*Frameworks Back-End*).

Referências

Ciribelli, M.C. (2003). Como Elaborar uma Dissertação de Mestrado Através da Pesquisa Científica. Rio de Janeiro: 7Letras.

- CITRUS7 (2017). O que é Front-end e Back-end? <https://citrus7.com.br/artigo/o-que-e-front-end-e-back-end/>. Novembro.
- DEVMEDIA (2017b). Bootstrap. <http://www.devmedia.com.br/guia/bootstrap/38150>. Setembro.
- DEVMEDIA (2017a). Introdução ao Laravel Framework PHP. <http://www.devmedia.com.br/introducao-ao-laravel-framework-php/33173>. Setembro.
- DEVMEDIA (2017c). Projeto de Banco de Dados - Parte 1. <http://www.devmedia.com.br/projeto-de-banco-de-dados-parte-1/10923>. Outubro.
- Elmasri R., Navathe S.B. (2011). Sistemas de Banco de Dados. 6° ed. São Paulo: Pearson Addison-Wesley.
- Gonçalves, E. (2007). Desenvolvendo Aplicações Web com JSP Servlets, JavaServer Faces, Hibernate, EJB3 Persistence e Ajax. Editora Ciência Moderna Ltda.
- Heuser, C.A. (2009). Projeto de Banco de Dados. 6° ed. Porto Alegre: Bookman.
- Mattsson, M. (1996). Object-Oriented Frameworks: A Survey of Methodological Issues, M.Sc. Dissertation, Department of Computer Science and Business Administration, University College of Karlskrona/Ronneby, LU-CS-96-197.
- Minetto, E. (2007). Frameworks para Desenvolvimento em PHP. 1. Ed. São Paulo: Novatec Editora.
- MYSQL (2017b). Informações gerais. <https://dev.mysql.com/doc/refman/5.7/en/introduction.html>. Maio.
- MYSQL (2017a). O que é o MySQL?. <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>. Maio.
- Nash, M. (2003). Java Frameworks and Components: Accelerate Your Web Application Development. Cambridge University Press.
- PHP (2017). O que é o PHP? http://php.net/manual/pt_BR/introwhatis.php. Maio.
- PHP-FIG (2017). PSR-2: Coding Style Guide. <http://www.phpfig.org/psr/psr-2/>. Novembro.
- Pressman, R.S. (2002). Engenharia de software. 5.ed. Rio de Janeiro: McGraw-Hill.
- Soares, W. (2013). Conceitos, programação e integração com banco de dados. 7° ed, Érica.
- Sommerville, I. (2007). Engenharia de Software. 8. ed. São Paulo: Pearson Addison Wesley. 552 p.
- Wazlawick, R.S. (2014). Metodologia de Pesquisa para Ciência da Computação. 2. ed. Rio de Janeiro: Elsevier.
- WIKIPÉDIA (2017). Linguagem de Programação. https://pt.wikipedia.org/wiki/Linguagem_de_programação. Novembro.