



Relatório – Parte 2
(SSC5904) Reúso de Software
Implementação de Linha de Produto de Software

Disciplina: Reúso de Software
Prof^a Dr^a Rosana Braga

Aluno: Misael Costa Júnior (NUSP 9254469)

USP - São Carlos
Dezembro/2018

Este relatório apresenta e discute detalhes técnicos referentes a implementação de uma Linha de Produto de Software (LPS) no domínio de Eventos Acadêmicos. A LPS foi implementada por meio do *framework* Bootstrap. Além disso, foram implementados componentes em Javascript e utilizados documentos que introduzem conceitos relacionados a LPS (Burgareli, 2009).

A seguir, são apresentados e discutidos as atividades executadas para o desenvolvimento do projeto de implementação da LPS. Sendo assim, a Seção 1 apresenta o processo de desenvolvido da LPS; e a Seção 2 apresenta detalhes para execução da LPS. Por fim, a Seção 3 descreve as considerações finais do projeto realizado.

1 Aspectos técnicos

Para o desenvolvimento do presente projeto foi utilizado o *framework* Bootstrap¹ com HTML5 e Javascript. Bootstrap é um *framework web* com código-fonte aberto para o desenvolvimento de componentes de interface (*back-end*) e *front-end* para sites e aplicações web usando HTML5, CSS e JavaScript, baseados em modelos de *design* para a tipografia.

Bootstrap é modular e consiste de uma série de estilos LESS que implementam os vários componentes do kit de ferramentas. Estas ferramentas são geralmente compiladas em um pacote e incluídas em páginas web, mas os componentes individuais podem ser incluídos ou removidos. A página oficial do Bootstrap contém um amplo conjunto de tutoriais, amostras de componentes e uma comunidade com fácil acesso a discussão e dúvidas sobre o uso do *framework*.

A seguir, serão discutidos detalhes sobre o desenvolvimento do gerenciador de Eventos Acadêmicos (Seção 1.1) e aspectos importantes relacionados a execução do gerenciador (Seção 1.2).

1.1 Configurações iniciais

Para o uso correto do Bootstrap, dois passos importantes devem ser executados. Sendo assim, a seguir é apresentado cada passo para a configuração inicial do projeto.

1. **CSS:** inicialmente, é necessário adicionar a seguinte linha na página inicial da aplicação:

```
1 <link rel="stylesheet"href="https://stackpath.bootstrapcdn.com/bootst
2 rap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3
3 GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkF0JwJ8ERdknLPM0" crossorigin="anony
4 mous">
```

¹<https://goo.gl/6ynbvX>

-
2. **Javascript:** para o funcionamento correto do Javascript é necessário que os plugins jQuery e Popper.js estejam adicionados ao projeto. Sendo assim, é necessário adicionar os seguintes `<script>s` no final da página, logo antes do fechamento da tag `</body>`. jQuery tem que vir antes, depois o Popper.js.

```
1 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
2
3 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWIPm49" crossorigin="anonymous"></script>
4
5 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy" crossorigin="anonymous"></script>
6
7
8
9
```

A partir da execução dessas duas etapas, os requisitos mínimos para execução correta do Bootstrap estão finalizados.

1.2 Organização do projeto

Objetivando o desenvolvimento e execução da LPS, o projeto foi organizado assim como a Figura 1 ilustra. A seguir, são apresentados detalhes de cada diretório e dos principais arquivos criados.

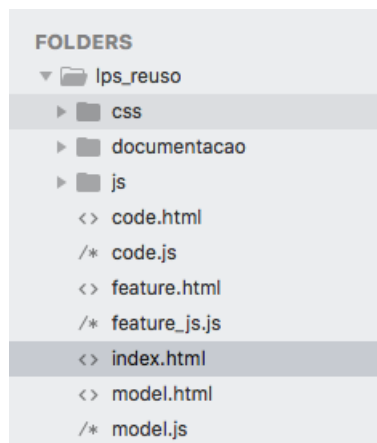


Figura 1: Organização do projeto de Eventos Acadêmicos.

1. **CSS:** diretório contendo os arquivos .css gerados, automaticamente, pelo Bootstrap;
2. **documentacao:** diretório contendo arquivos de documentação do projeto desenvolvido. Foram inseridos dois arquivos nesse diretório: documentação da modelagem e documentação da LPS desenvolvida;
3. **js:** diretório contendo os arquivos .js gerados, automaticamente, pelo Bootstrap;

-
4. **feature.html:** arquivo .html referente a etapa 1 para geração da LPS contendo alguns checkbox para seleção das features;
 5. **feature.js:** arquivo .js com a implementação dos itens adicionados no arquivo `feature.html`;
 6. **code.html:** arquivo .html referente a etapa 3 para geração da LPS contendo os itens necessários para geração do código a partir do modelo gerado;
 7. **code.js:** arquivo .js com a implementação do modelo gerado no arquivo `code.html`;
 8. **model.html:** arquivo .html referente a etapa 2 para geração da LPS. Esse arquivo contém uma página que irá exibir o modelo a partir das features selecionadas pelo usuário;
 9. **model.js:** arquivo .js com a implementação dos itens adicionados no arquivo `model.html`;
 10. **index.html:** página inicial do projeto contendo informações importantes, links para visualização da documentação e botão de início para geração da LPS.

Para evitar destrinchar muitos detalhes de implementação do projeto neste documento, o projeto pode ser apreciado de modo *open-source* via *github*².

2 Execução

Após execução das atividades apresentadas e da implementação da estrutura da LPS apresentada na Seção 1, o projeto está pronto para uso. Sendo assim, a seguir será apresentado um passo a passo para execução do projeto.

2.1 Página inicial

Na página inicial do projeto são apresentados detalhes do contexto do projeto desenvolvido e links para acesso a documentação do projeto. Além disso, foi incluído um NavBar³ para organização das páginas do Bootstrap. A Figura 2 apresenta detalhes e a identificação dos componentes da página inicial.

²<https://goo.gl/wFbvJR>

³<https://goo.gl/sqTusm>



Figura 2: Página inicial e seus principais componentes.

2.2 Seleção de features

Na página de seleção de features, o usuário poderá selecionar as características de sua preferência para geração do código ao final da aplicação. Além disso, o usuário pode definir um nome para a aplicação. Importante ressaltar que todas essas informações serão utilizadas na geração do código. A Figura 3 apresenta detalhes da página de seleção de características.

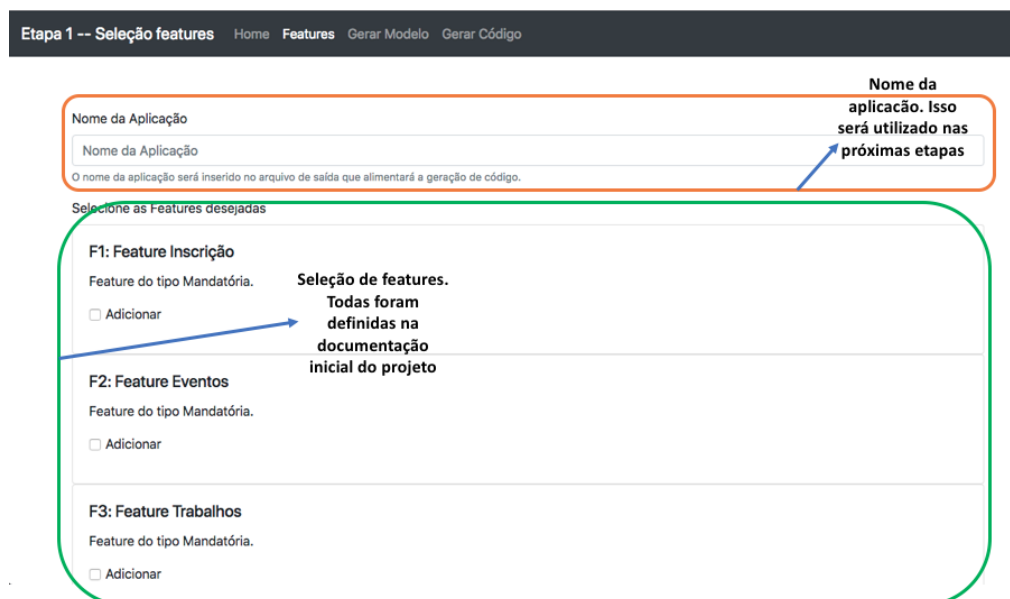


Figura 3: Página de seleção de features.

2.3 Geração do modelo

Na página de seleção de modelo, será dada uma amostra para o usuário do modelo gerado a partir das features selecionadas. Cada feature irá receber true, caso tenha sido selecionada pelo usuário, ou false, caso não tenha sido selecionada. Tais valores serão

utilizados para gerar o código da aplicação. A Figura 4 apresenta um exemplo de modelo gerado pela aplicação.

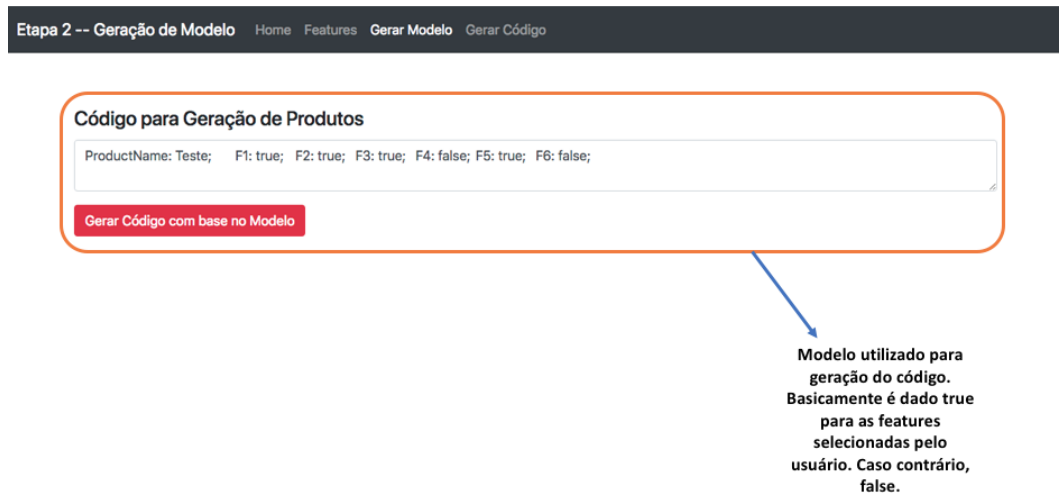


Figura 4: Página geração do modelo.

2.4 Geração de código

Por fim, na página de geração do código, é gerado código para cada feature. O código, variáveis e tipos das variáveis, foram definidos estaticamente. As amostras de códigos inseridas foram definidas na linguagem de programação Python. A Figura 5 apresenta um exemplo de código gerado pela aplicação.

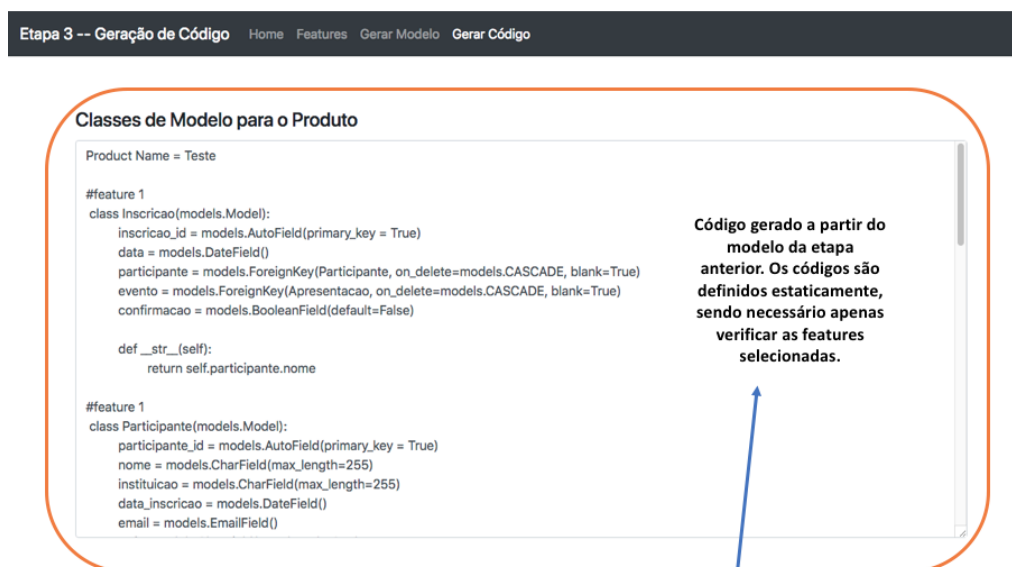


Figura 5: Página de geração do código.

3 Considerações Finais

O presente documento teve como objetivo apresentar o processo de desenvolvimento de uma LPS para o domínio de Eventos Acadêmicos. Desse modo, a aplicação foi especificada para contemplar sistemas de gerenciamento de eventos acadêmicos em cursos de graduação e pós-graduação. Além disso, o processo de desenvolvimento foi realizado com base na documentação desenvolvida na primeira etapa do projeto. Ambas podem ser consultadas por meio do projeto anexado no *github*⁴.

Espera-se que por meio deste documento os conceitos, técnicas e ferramentas utilizadas para especificar, modelar e desenvolver a aplicação, instigue pesquisadores e estudantes entenderem o processo de desenvolvimento de uma LPS.

Referências

Burgareli, L. A. Gerenciamento de variabilidade de linha de produtos de software com utilização de objetos adaptáveis e reflexão. *Universidade de São Paulo. São Paulo*, 2009.

⁴<https://goo.gl/wFbvJR>