

# Tutorial de Desenvolvimento com FrameWeb SPA

## *FrameWeb SPA Development Tutorial*

Pedro Henrique Brunoro Hoppe\*

Vitór Estêvão Silva Souza†

2022

## 1 Instalação

### 1.1 Instalação do FrameWeb SPA

Para instalar o FrameWeb SPA acesse o sítio eletrônico <<https://github.com/pedrohbbh/FrameWebSPA>> e clone o repositório para sua máquina (ou baixe o zip disponível em “Code” -> “Download Zip”) [Figura 1].

Após baixar o repositório, siga os seguintes passos:

1. Abra o Eclipse;

---

\*“Mestrando.” <[pedrohoppe@gmail.com](mailto:pedrohoppe@gmail.com)>

†“Constar currículo sucinto de cada autor, com vinculação corporativa e endereço de contato.”

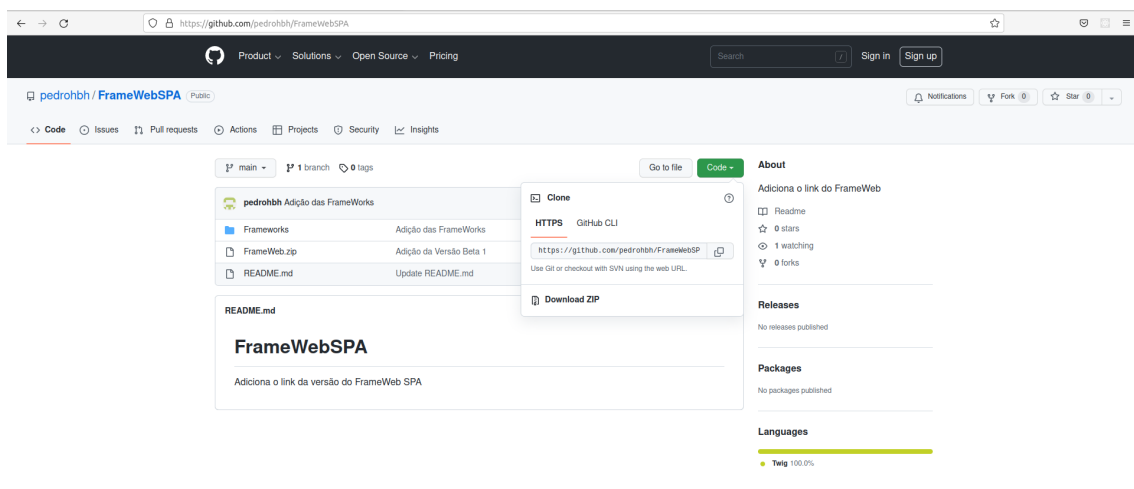


Figura 1 – Repositório GitHub

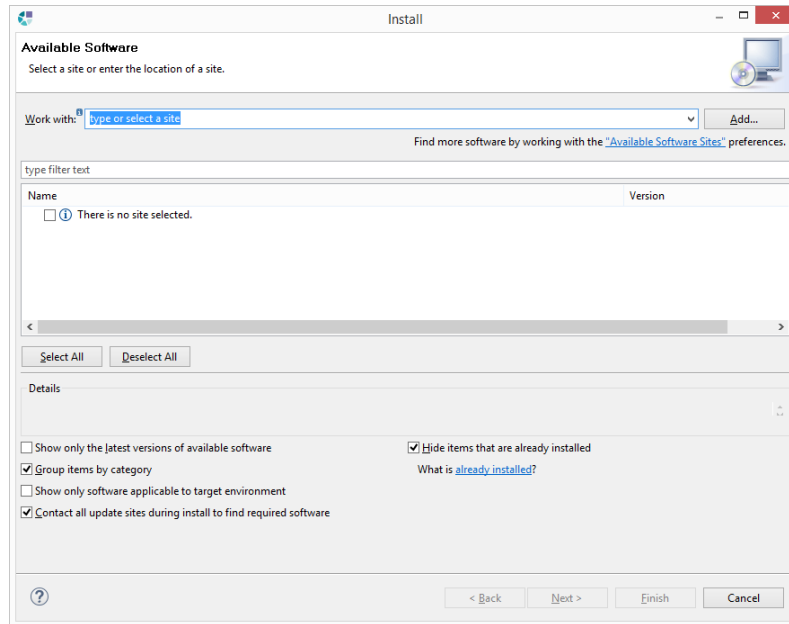


Figura 2 – Menu de instalação do Eclipse

2. Clique no menu Help -> Install New Software...;
3. Clique em “Add...” (Figura 2);
4. Preencha os campos:
 

**Name:** Pode ser o nome que você quiser. *Sugestão: FrameWeb SPA*;

**Location:** Clique no botão “Archive. . .” e depois selecione o arquivo “FrameWeb.zip” na pasta onde foi clonado o repositório do GitHub anteriormente;
5. Clique no botão “Add” (Figura 3);
6. Em “Work with:” selecione o repositório adicionado no passo anterior (no nosso caso, *FrameWeb SPA*), marque as opções “FrameWeb Code Generator” e “FrameWeb Graphical Editor” e clique em “Next >” (Figura 4);
7. Caso não apareçam as opções como na Figura 4, desmarque a opção “Group items by category”;
8. Siga os demais passos de instalação aceitando os termos de licença (Figura 5). Caso apareça uma mensagem dizendo que os certificados de segurança não são reconhecidos, apenas marque todos e clique em Ok (Figura 6);
9. Após a instalação, reinicie o Eclipse.

Após a instalação, a montagem do seu projeto segue os mesmos passos disponíveis na Wiki do FrameWeb <<https://github.com/nemo-ufes/FrameWeb/wiki/ToolsTutorial02>>. A **exceção** é a inclusão do projeto de arquitetura do seu framework, que ao invés de você baixar o JButler da página Github, você usará um dos três frameworks, a saber, Angular, React ou VueJS que estão disponíveis na pasta “Frameworks” no repositório clonado anteriormente <<https://github.com/pedrohbh/FrameWebSPA>>. Importe também a definição da linguagem Java dentro da pasta “Language”.

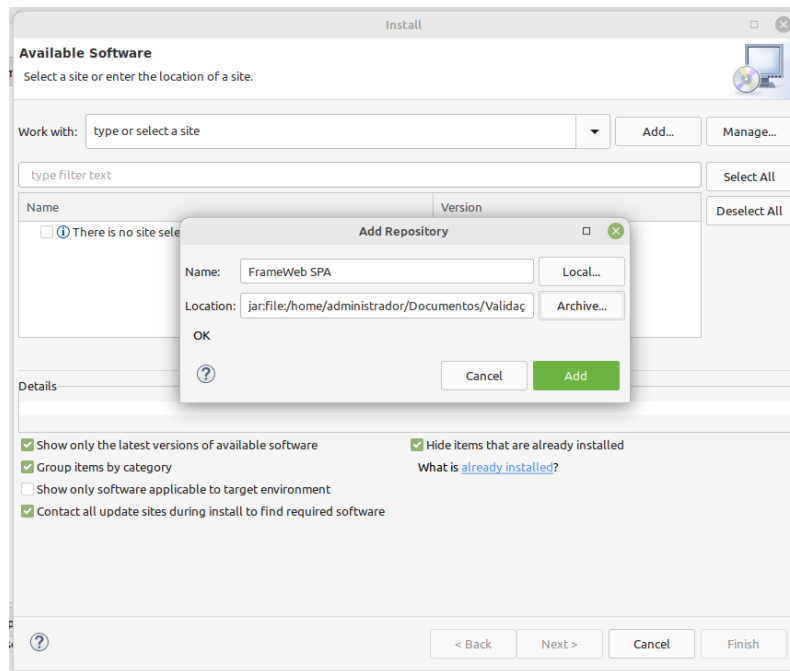


Figura 3 – Passo 5

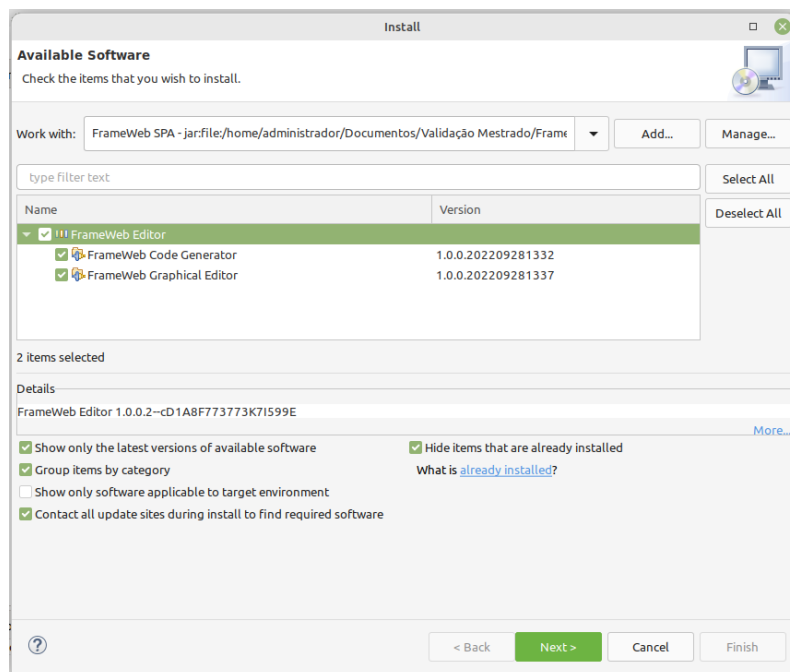


Figura 4 – Passo 6

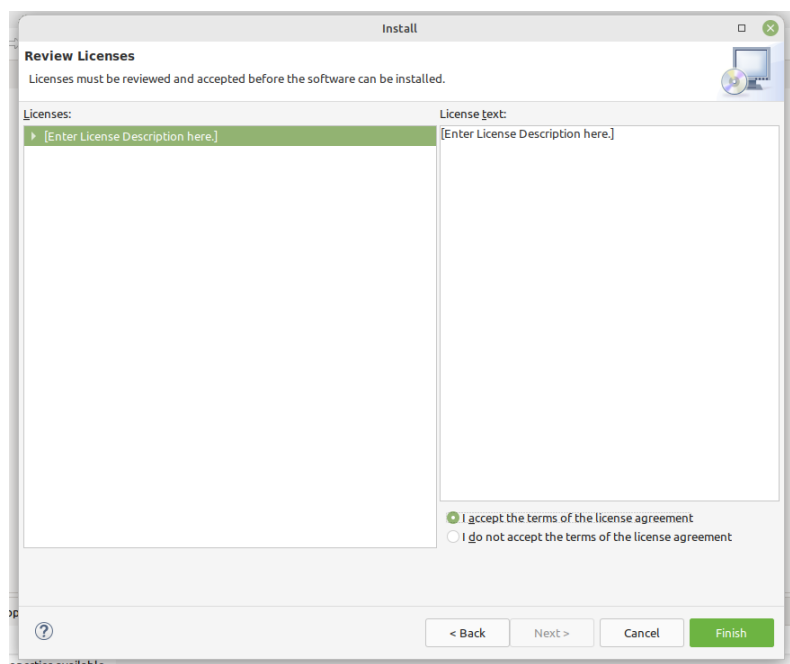


Figura 5 – Licença

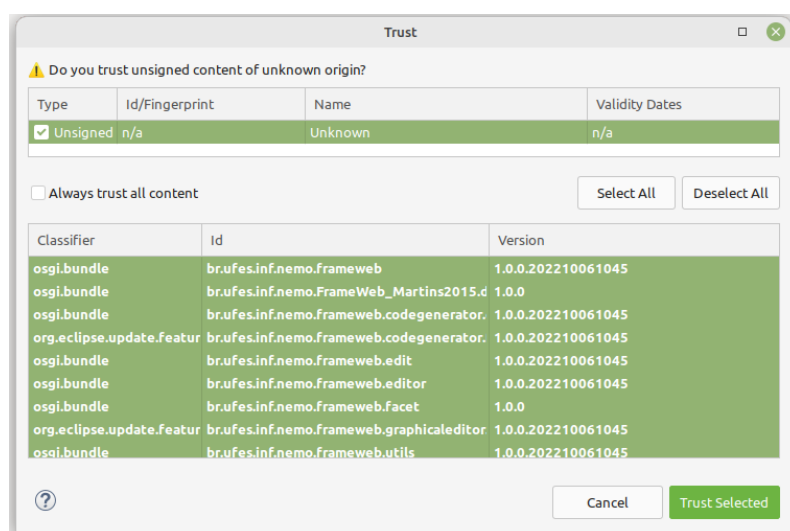


Figura 6 – Certificado

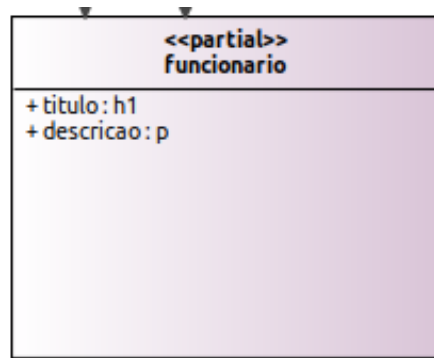


Figura 7 – O elemento Partial

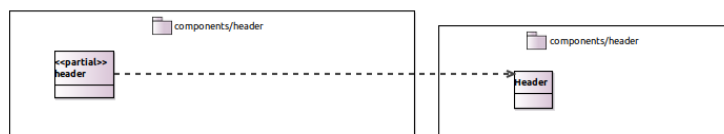


Figura 8 – Partial ligado a seu FrontControllerClass

## 2 Desenvolvimento

O desenvolvimento do projeto seguem as mesmas instruções da Wiki <https://github.com/nemo-ufes/FrameWeb/wiki> para todos os modelos. A exceção é o Modelo de Navegação que também segue as mesmas instruções, porém com algumas mudanças que serão descritas a seguir.

### 2.1 Partial

O elemento novo principal é o Partial (Figura 7). Ele representa a parte gráfica de um *component* SPA. Possui as mesmas características do elemento «page» para os Frameworks MVC tradicionais, porém sua interpretação (inclusive para o gerador de código) difere de «Page» e para o caso dos frameworks SPA, o Partial deve ser utilizado ao invés de Page.

Como “Components” para os frameworks SPA obrigatoriamente tem uma parte gráfica e uma parte lógica (o controlador frontal), e por isso **todo partial tem que estar ligado a controlador frontal por meio da relação “Front Controller Depedency”** como na Figura 8, mesmo que não haja nenhum método específico que o controle. Observe também que os nomes dos pacotes (tanto do “View Package” onde está o partial como o “Controller Package” onde está o “FrontControllerClass”) são iguais. Isso porque o “Component” está em um arquivo só, contendo o “Partial” e a parte de controlador frontal representado pelo “FrontControllerClass”. Por isso é recomendado que se use o mesmo nome para cada “Component” (tanto de pacotes como para o nome do “Partial” e do “FrontControllerClass”).

Os “Partials”, assim como os “Pages”, podem ter associado-lhes um formulário (“Form”, que no caso do FrameWeb é chamado de “UIComponent”) e funcionam da mesma forma que em “Pages” (Figura 9).

No caso de frameworks SPA, é comum que um método do controlador frontal

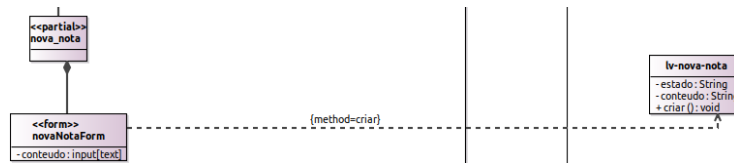


Figura 9 – Partial ligado a um formulário

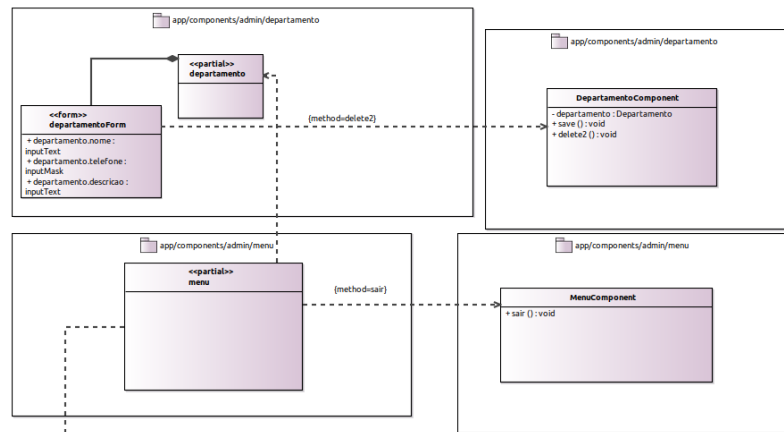


Figura 10 – Exemplo de parte do sistema com redirecionamento

retorne a si mesmo (a sua própria “página”), por isso, ao contrário dos frameworks MVC tradicionais, não é obrigatório especificar um retorno do controlador frontal por meio do “ResultDependency”, apenas caso haja um redirecionamento para um outro “Partial” que não seja ele próprio.

Falando em “ResultDependency”, ele que é o responsável por indicar o redirecionamento entre “partials” (observe a Figura 10, o “Partial” menu possui um redirecionamento para o “Partial” departamento).

## 2.2 Navigation Aggregation Association

Then we move on to the Navigation Model, creating a View Package /core/registration/ and a Controller Package br.ufes.informatica.oldenburg.core.controller. Here are some tips to create a Navigation Model:

Create pages using the Page component at the palette and give them a name without the .html extension, as this will be added later by the code generator;

Create forms using the UI Component component at the palette. Form fields are created with UIComponentField and their type should be one of the tags from your visual component library (in our case, PrimeFaces);

When selecting the type of the tag from the dropdown list, if you type the name of the Tag the list will be filtered, but you have to start from the beginning (e.g., type Tag input to see the different PrimeFaces tags that start with input);

Form fields (UIComponentField) can also be added to pages directly; To link a page to a form, use the Navigation Association component from the palette (I know, these names are not intuitive at all, we’ll work on that...);

Moving to the controller side, a Front Controller Class can get attributes via `IOParameter` component and methods via `Front Controller Method` component from the palette;

To connect a page or form to a controller, use `Front Controller Dependency` and select one of the methods from the controller in the `Method` property;

To connect a controller to a page, use the `Result Dependency` and select one of the methods from the controller in the `Result Method` property. Further, click on the `Result Constraints` section and set the result and `AJAX` constraints if applicable.

Naming convention note: `JButler` templates expect all controller classes to have the `Controller` suffix (e.g., `RegistrationController`). Code generation will work better if you follow this naming convention.

The figure below shows the result for the `Navigation Model`.

### 3 Considerações finais

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

## Referências

ARAUJO, L. C. *A classe abntex2: Modelo canônico de trabalhos acadêmicos brasileiros compatível com as normas ABNT NBR 14724:2011, ABNT NBR 6024:2012 e outras.* [S.l.], 2015. Disponível em: <<http://www.abntex.net.br/>>. Nenhuma citação no texto.

ARAUJO, L. C. *Como customizar o abnTeX2.* 2015. Wiki do abnTeX2. Disponível em: <<https://github.com/abntex/abntex2/wiki/ComoCustomizar>>. Acesso em: 27 abr 2015. Nenhuma citação no texto.

ARAUJO, L. C. *Modelo Canônico de Trabalho Acadêmico com abnTeX2.* [S.l.], 2015. Disponível em: <<http://www.abntex.net.br/>>. Nenhuma citação no texto.

ARAUJO, L. C. *O pacote abntex2cite: Estilos bibliográficos compatíveis com a ABNT NBR 6023.* [S.l.], 2015. Disponível em: <<http://www.abntex.net.br/>>. Nenhuma citação no texto.

ARAUJO, L. C. *O pacote abntex2cite: tópicos específicos da ABNT NBR 10520:2002 e o estilo bibliográfico alfabético (sistema autor-data).* [S.l.], 2015. Disponível em: <<http://www.abntex.net.br/>>. Nenhuma citação no texto.

WILSON, P.; MADSEN, L. *The Memoir Class for Configurable Typesetting - User Guide.* Normandy Park, WA, 2010. Disponível em: <<http://mirrors.ctan.org/macros/latex/contrib/memoir/memman.pdf>>. Acesso em: 19 dez. 2012. Nenhuma citação no texto.

APÊNDICE A – Nullam elementum urna vel imperdiet sodales elit  
ipsum pharetra ligula ac pretium ante justo a nulla curabitur tristique  
arcu eu metus

Nunc velit. Nullam elit sapien, eleifend eu, commodo nec, semper sit amet, elit. Nulla lectus risus, condimentum ut, laoreet eget, viverra nec, odio. Proin lobortis. Curabitur dictum arcu vel wisi. Cras id nulla venenatis tortor congue ultrices. Pellentesque eget pede. Sed eleifend sagittis elit. Nam sed tellus sit amet lectus ullamcorper tristique. Mauris enim sem, tristique eu, accumsan at, scelerisque vulputate, neque. Quisque lacus. Donec et ipsum sit amet elit nonummy aliquet. Sed viverra nisl at sem. Nam diam. Mauris ut dolor. Curabitur ornare tortor cursus velit.

Morbi tincidunt posuere arcu. Cras venenatis est vitae dolor. Vivamus scelerisque semper mi. Donec ipsum arcu, consequat scelerisque, viverra id, dictum at, metus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut pede sem, tempus ut, porttitor bibendum, molestie eu, elit. Suspendisse potenti. Sed id lectus sit amet purus faucibus



vehicula. Praesent sed sem non dui pharetra interdum. Nam viverra ultrices magna.

**ANEXO A – Cras non urna sed feugiat cum sociis natoque penatibus  
et magnis dis parturient montes nascetur ridiculus mus**

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consectetur nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

## **Agradecimentos**

Texto sucinto aprovado pelo periódico em que será publicado. Último elemento pós-textual.