

Universidade Federal do Estado do Rio de Janeiro Centro de Ciências Exatas e Tecnologia Escola de Informática Aplicada

Responsividade e modularização em sites web: um estudo de caso com o Portal BSI

Natália Dias da Costa

Orientadora Morganna Carmem Diniz

Rio de Janeiro, RJ – Brasil Julho de 2018

Catalogação informatizada pelo autor

Costa, Natália Dias da Responsividade e modularização em sites web: um estudo de caso com o Portal BSI / Natália Dias da Costa. -- Rio de Janeiro, 2018. 56 p Orientador: Morganna Carmem Diniz. Trabalho de Conclusão de Curso (Graduação) -Universidade Federal do Estado do Rio de Janeiro, Graduação em Sistemas de Informação, 2018. 1. responsividade. 2. bootstrap. 3. react. 4. modularização. 5. bsi. I. Diniz, Morganna Carmem, orient. II. Título. Responsividade e modularização em sites web: um estudo de caso com o Portal BSI

Natália Dias da Costa

Projeto de Graduação apresentado à Escola de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado por:

Morganna Carmem Diniz (UNIRIO)

Pimentel Mariano (UNIRIO)

Rio de Janeiro, RJ – Brasil. Julho de 2018

Agradecimentos

À minha mãe, por ter sido um verdadeiro exemplo de como superar todo tipo de dificuldade. Obrigada por todos os sacrifícios feitos por mim. Obrigada por me mostrar todo dia o que é amor de mãe.

Ao meu pai, por toda a fé em meu sucesso e todo o incentivo. Por toda a calma e paciência para me guiar, para me escutar e para me ensinar. Por ter se feito presente, até quando estávamos físicamente separados.

Ao meu padrasto Marcelo, por ter feito um trabalho tão excepcional de ser segundo pai.

Ás minhas avós, por todos os ensinamentos e por me inspirarem todo dia a ter força.

Aos meus tios, tias, primas e primos. Obrigada por todos os conselhos, puxões de orelha e incentivo durante todos esses anos. Em mim tenho uma partezinha de cada um de vocês.

Ao meu namorado, por toda a ajuda e, principalmente, por todos os copos de coca-cola trazidos para mim enquanto escrevia este trabalho. Não existe prova de amor maior que essa. Eu te amo.

Aos forninhos, por terem sido minha família durante o ano que estive com vocês. Obrigada por cada pizza e cookie compartilhados na madrugada.

Aos meus amigos, por todas as questões feitas 15 minutos antes da prova em alguma sala, por todos os trabalhos em grupo e por todos os desabafos durante esses 6 anos de BSI.

À minha orientadora, professora Morganna, por toda a compreensão e confiança.

RESUMO

Neste trabalho de conclusão de curso são apresentadas as medidas tomadas para que a experiência do usuário no atual site do curso de Sistemas de Informação da UNIRIO, o Portal BSI, seja melhorada. Focando na experiência em telas de dispositivos móveis, são aplicados conceitos do padrão de responsividade utilizando Bootstrap. Ao final, o resultado deste trabalho é a nova versão do Portal BSI.

Palavras-chave: responsividade, bootstrap, react, modularização, bsi.

ABSTRACT

The means to achieve a better user experience on the current UNIRIO's Systems Informations course's webpage are discussed on this final course project. The concepts of responsive webdesign are applied on the page, using Bootstrap, focusing on the small screens experience. At the end, this work has the new version for the webpage as a result.

Keywords: responsive, bootstrap, react, modularity, bsi.

Índice

Introdução	12
Motivação	12
Objetivos	12
Organização do texto	12
Técnicas e Tecnologias	13
Responsividade	13
Bootstrap	14
Grids	14
Collapse	16
Display em Bootstrap	16
Padrões de Navegação	19
Horizontal Scroll	19
The select menu	20
The multi-toggle	21
Off canvas flyout	22
The old right-to-left	23
Skip the sub-nav	23
React	24
Create-react-app	25
Estudo e Aplicação	26
Criando o Portal BSI em React	26
Arquivo de estrutura	26
Arquivos de conteúdo	28
Componentes	28
Responsividade do menu de navegação	29
The multi-toggle adaptado ao Portal BSI	31
The old right-to-left adaptado ao Portal BSI	32
Off-canvas flyout adaptado ao Portal BSI	33
Prós e contras e definição de layout	35
Aplicação do sistema de Grids às páginas do Portal BSI	37
Adequação de imagens e vídeos ao tamanho de tela	38
Aplicação do sistema de Grids no menu para dispositivos grandes	39
Criação e aplicação do Off-canvas flyout para dispositivos médios ou menores	40
Criação e transição para header menor	40
Resultados	41
Padrão off-canvas flyout	41
Portal BSI em React	44

Performance	44
Manutenção	45
Perda de Funcionalidades	46
Conclusão	48
Considerações Finais	48
Trabalhos Futuros	50
ANEXO I - Passo-a-passo para adicionar um submenu ao Portal BSI original	53
ANEXO II - Passo-a-passo para adicionar um submenu ao Portal BSI em React	54
ANEXO III - Passo-a-passo para editar um conteúdo do Portal BSI original	55
ANEXO IV - Passo-a-passo para editar um conteúdo do Portal BSI em React	56

Índice de Tabelas

Tabela 1 - Prefixos de classe e breakpoints do sistema de grid do Bootstrap	15
Tabela 2 - Prós e contras dos padrões para o Portal BSI	35
Tabela 3 - Comparação de Performance	45
Tabela 4 - Quantidade de linhas em cada projeto	45
Tabela 5 - Prós e contras do Portal em React e sem React	49

Índice de Figuras

Figura 1 - Divisão de colunas em Grid no Bootstrap	14
Figura 2 - Distribuição dos breakpoints em dispositivos	15
Figura 3 - Elemento escondido com <i>collapse</i>	16
Figura 4 - Elemento visível com collapse após clique do usuário	16
Figura 5 - Tipos de alinhamento horizontal	17
Figura 6 - Tipos de alinhamento vertical	18
Figura 7 - Ícone hamburger	19
Figura 8 - Padrão Horizontal Scroll	20
Figura 9 - Padrão The select menu	21
Figura 10- Padrão the multi-toggle	22
Figura 11 - Padrão off canvas flyout	22
Figura 12 - Padrão the old right-to-left	23
Figura 13 - Ranking de Frameworks, bibliotecas e outras tecnologias mais amadas por desenvolvedores	24
Figura 14 - Site da Udacity e seus componentes	25
Figura 15 - Estrutura do item BSI do menu	27
Figura 16 - Árvore de conteúdos	28
Figura 17 - Comportamento do Portal BSI em diferentes telas	30
Figura 18 - Menu de navegação do Portal BSI com The multi-toggle	31

Figura 19 - Itens e subitens do menu do Portal BSI com The multi-toggle	32
Figura 20 - Itens e subitens do menu do Portal BSI com The old right-to-left	33
Figura 21 - Menu de navegação do Portal BSI com Off-canvas flyout	34
Figura 22 - Itens e subitens do menu do Portal BSI com Off-canvas flyout	35
Figura 23 - Menu do Portal BSI para dispositivos grandes e extragrandes	39
Figura 24 - Container abrigando todo o conteúdo do menu	39
Figura 25 - Container abrigando apenas os elementos divididos em duas linhas	40
Figura 26 - Header em dispositivos médios ou menores	41
Figura 27 - Antes x Depois da visão principal da página	42
Figura 28 - Antes x Depois do menu off-canvas aberto	43
Figura 29 - Antes x Depois do menu off-canvas expandido	44
Figura 30 - Página de Notícias do Portal BSI original	46
Figura 31 - Página de Notícias do Portal BSI em React	47
Figura 32 - Página Calendário Acadêmico do Portal BSI original	47
Figura 33 - Página Calendário Acadêmico do Portal BSI em React	48

1. Introdução

1.1. Motivação

De acordo com o Suplemento de Tecnologias de Informação e Comunicação da Pesquisa Nacional por Amostra de Domicílios (Pnad) 2014 [1] divulgado pelo Instituto Brasileiro de Geografia e Estatística (IBGE), o uso do telefone celular para acessar a internet ultrapassou o uso do computador no Brasil. Com essa mudança no hábito dos brasileiros, a adaptação dos principais sites para o modo adequado para telas menores passou a ser prioridade.

Hoje em dia, o Portal BSI não possui essa adaptação. Apesar do seu tamanho se ajustar à tela, os itens acabam sendo muito pequenos e a usabilidade é prejudicada. Acreditamos que, para melhorar a visibilidade do curso, a adequação do nosso portal de forma que todos os usuários tenham uma boa experiência é essencial.

1.2. Objetivos

O objetivo deste trabalho é melhorar a forma como o nosso curso é conhecido através de nosso portal, permitindo uma melhor experiência dos usuários.

Para alcançar esse objetivo, alguns objetivos secundários precisam ser cumpridos como, por exemplo, tornar o portal responsivo, dispondo suas páginas, menus e itens de maneira diferente na tela dependendo do tamanho da mesma.

1.3. Organização do texto

O presente trabalho está estruturado em capítulos que, além desta Introdução, é desenvolvido como mostrado abaixo.

- Capítulo 2: Técnicas e Tecnologias Este capítulo introduz os principais conceitos utilizados neste trabalho, como definições envolvendo a estrutura de páginas web, e conceitos específicos do *Bootstrap*.
- Capítulo 3: Estudo e Aplicação Este Capítulo apresenta o estudo do framework *React* e explica como será aplicado no Portal BSI; mostra os possíveis layouts do Portal, listando os prós e contras de cada um; e discute a aplicação efetiva dos componentes *Bootstrap* e do framework *React* na nova versão do Portal.
- Capítulo 4: Resultados Este Capítulo discute os resultados obtidos com a adequação do sistema, apontando seus pontos negativos e positivos.
- Capítulo 5: Conclusões O Capítulo final reúne as considerações finais, identificando as contribuições do TCC para o Portal e sugere alguns trabalhos futuros.

2. Técnicas e Tecnologias

2.1. Responsividade

Por definição do dicionário online de português Dicio, responsividade significa "capacidade de responder rapidamente e do modo mais adequado à situação em questão" [5]. Aplicando o conceito ao mundo web, ele representa a capacidade de uma página web adaptar seu design e suas funcionalidades à tela do usuário.

Apesar da disposição dos elementos em diferentes tamanhos de tela ser o principal indicador de uma página responsiva, esta não é a única preocupação ao adaptá-la. É preciso pensar em manter a qualidade e funcionamento de todas as funções do site, considerar a velocidade reduzida de alguns usuários e garantir que certos

elementos sejam ocultados quando sua usabilidade for prejudicada devido ao tamanho da tela.

2.2. Bootstrap

O Bootstrap é um *framework* de *front-end* para desenvolvimento de páginas que oferece padrões para HTML, CSS e JavaScript. Utilizando estes padrões oferecidos é possível agilizar consideravelmente o desenvolvimento de páginas responsivas.

2.2.1. Grids

O sistema de Grids do Bootstrap é montado utilizando um *container*, que irá abrigar todos os elementos, e linhas e colunas que organizarão os elementos.

O Grid do Bootstrap sempre é dividido em 12 colunas, cada uma delas com largura definida de acordo com a tela do usuário. É possível mesclar colunas dependendo do tamanho da tela, definindo a melhor exibição dos elementos da página para a visualização do usuário.

Figura 1 - Divisão de colunas em Grid no Bootstrap

COL-3	COL-3		COL-3				COL-3 COL-3				
COL-4 COL-4			COL-4								
COL-6	DL-6 COL-6										
COL-2		L-2 COL-2				COL-2		COL-2		COL-2	
COL-1	COL-1	COL-1	COL-1	COL-1	COL-1	COL-1	COL-1	COL-1	COL-1	COL-1	COL-1

Fonte: http://blog.gridbox.io/2018/01/08/bootstrap-4-vs-foundation-6-grid-system/

Caso seja necessário alterar a quantidade de colunas mescladas de acordo com o tamanho de tela do usuário, é preciso alterar o prefixo de classe padrão (.col-) para o prefixo do tamanho desejado (.col, .col-sm, .col-md, .col-lg ou .col-xl) e ficar atento aos breakpoints de cada um deles. A Tabela 1 apresenta os *breakpoints* de cada prefixo.

	Tamanho da tela							
	Extra pequena <576px	Pequena ≥576px	Média ≥768px	Grande ≥992px	Extra grande ≥1200px			
Largura máxima do <i>container</i>	Não há (auto)	540px	720px	960px	1140px			
Prefixo da Classe	.col-	.col-sm	.col-md	.col-lg	.col-xl			

Tabela 1 - Prefixos de classe e breakpoints do sistema de grid do Bootstrap

Na prática, cada breakpoint é direcionado para tamanhos diferentes das telas mais utilizadas atualmente. Na Figura 2, temos a distribuição dos *breakpoints* entre os dispositivos.





Fonte: http://blog.gridbox.io/2018/01/08/bootstrap-4-vs-foundation-6-grid-system/

2.2.2. Collapse

Collapse é um recurso do Bootstrap utilizado para esconder algum elemento e só deixá-lo visível quando houver alguma ação por parte do usuário (como clicar em um botão, por exemplo). As Figuras 3 e 4 mostram um exemplo de elemento escondido e depois visível, através do clique do usuário.

Figura 3 - Elemento escondido com *collapse*

Collapsible Group Item #1		
Collapsible Group Item #2		
Collapsible Group Item #3		

Fonte: https://getbootstrap.com/docs/4.0/components/collapse/

Figura 4 - Elemento visível com collapse após clique do usuário

Collapsible Group Item #1
Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. 3 wolf moon officia aute, non cupidatat skateboard dolor brunch. Food truck quinoa nesciunt laborum eiusmod. Brunch 3 wolf moon tempor, sunt aliqua put a bird on it squid single-origin coffee nulla assumenda shoreditch et. Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo. Leggings occaecat craft beer farm-to-table, raw denim aesthetic synth nesciunt you probably haven't heard of them accusamus labore sustainable VHS.
Collapsible Group Item #2
Collapsible Group Item #3

Fonte: https://getbootstrap.com/docs/4.0/components/collapse/

2.2.3. *Display* em Bootstrap

Display é uma propriedade CSS que define se e como um elemento será exibido. Ela pode ter diversos valores, porém os mais utilizados são:

• none: o elemento não será exibido;

- *inline*: o elemento será exibido alinhado, podendo ter outros elementos antes e depois dele na mesma linha;
- *block*: o elemento é exibido em bloco, começando uma nova linha e tomando toda a largura disponível; e
- *flex*: o elemento é exibido como um *container flex*.

No Bootstrap, o display de um elemento é declarado da seguinte forma:

d-{breakpoint}-{valor}

Por exemplo, se um elemento div não deve ser exibido para telas médias, colocamos a seguinte classe no mesmo:

<div class="d-md-none">

No Bootstrap, um elemento com a propriedade *display* declarada com valor *flex* possui diversas opções de classes para alinhamento, layout e tamanho de seus "elementos-filhos".

Uma dessas classes é a *justify-content*, que muda o alinhamento horizontal de elementos dentro de um *flex*. Esse alinhamento pode variar entre: *start, end, center, between* e *around*. A Figura 5 ilustra os diferentes tipos de alinhamento horizontal.

justify-content-start						
Flex item Flex item						
justify-content-end						
				Flex item	Flex item	Flex item
justify-content-center						
	Flex item	Flex item	Flex item			
justify-content-between						
Flex item		Flex item				Flex item
justify-content-around						
Flex item		Flex item		Flex iten	ı	

Figura 5 - Tipos de alinhamento horizontal

A declaração desta classe é feita da seguinte maneira, podendo variar de acordo com o *breakpoint*.

```
<div class="d-flex justify-content-start">
  <div class="d-flex justify-content-end">
  <div class="d-flex justify-content-center">
  <div class="d-flex justify-content-between">
  <div class="d-flex justify-content-between"></div class="d-f
```

Outra classe para alinhamento é a *align-items*. Diferentemente da *justify-content*, essa classe alinha os elementos verticalmente. O alinhamento pode variar entre: *start*, *end*, *center*, *baseline* e *stretch*. A Figura 6 ilustra os diferentes tipos de alinhamento vertical.

align-items-start
Flex item Flex item
align-items-end
Flex item Flex item
align-items-center
Flex item Flex item Flex item
align-items-baseline
Flex item Flex item
align-items-stretch
Flex item Flex item

Figura 6 -	Tipos	de a	linhamento	vertical
------------	-------	------	------------	----------

A declaração desta classe é feita da seguinte maneira, podendo variar de acordo com o *breakpoint*.

```
<div class="d-flex align-items-start">
  <div class="d-flex align-items-end">
  <div class="d-flex align-items-center">
  <div class="d-flex align-items-baseline">
  <div class="d-flex align-items-baseline"></div class="d-flex align-items-baseline"></dvo dot dot dot dot dot do
```

2.3. Padrões de Navegação

Entre os padrões responsivos de navegação mais populares da web estão padrões que cobrem desde navegação simples (com poucos ítens) até as complexas (com vários itens e/ou subitens). Como padrão em termos de ícone de navegação, com o objetivo de exibir e ocultar itens de navegação, temos o ícone hamburger, apresentado na Figura 7.



Apesar de termos diversos padrões de navegação mapeados, vamos focar nos 6 padrões recomendados por Lucianna Dedis, em "Padrões de Design Responsivo" [6], para navegação com diversos itens e níveis. São eles: *Horizontal Scroll, The select menu, The multi-toggle, Off-canvas flyout, The old right-to-left* e *Skip the sub-nav.*

2.3.1. Horizontal Scroll

Neste padrão ao diminuir a tela a ponto dos ítens de menu não caberem mais na mesma, enfileira-os um após o outro, mesmo passando do tamanho da tela, e cria uma barra de rolagem para acessá-los. Assim, a ocupação ocupa pouca área da tela. A Figura 8 exemplifica esse padrão.



Figura 8 - Padrão Horizontal Scroll

Fonte: DEDIS, L. S. A. "Padrões de Design Responsivo" (2017)

2.3.2. The select menu

Neste padrão, quando a tela é pequena, o menu de navegação é transformada em um menu de seleção. A Figura 9 ilustra esse padrão..

Navegação

Figura 9 - Padrão The select menu

Fonte: DEDIS, L. S. A. "Padrões de Design Responsivo" (2017)

2.3.3. *The multi-toggle*

Neste padrão, o menu de navegação utiliza a funcionalidade *collapse*. Ou seja, fica oculto e para que o usuário consiga ver e acessar seus itens, é necessário que o mesmo clique em um botão ou link. Do mesmo modo, quando o menu é exibido, os subitens ficam ocultos e são exibidos após a ação do usuário. O padrão é exemplificado na Figura 10.

Figura 10- Padrão *the multi-toggle*

	3		Home Sobre V C	ontato
Home				
Sobre	\sim			
 Item A				
 Nível 2	\sim			
 Contato		-		
		No.		
. A				
 \rightarrow				

Fonte: DEDIS, L. S. A. "Padrões de Design Responsivo" (2017)

2.3.4. Off canvas flyout

Assim como o anterior, esse padrão utiliza o *collapse* para ocultar e exibir os itens do menu. Porém, neste caso, ele simula a entrada do menu como um deslizamento vindo de fora da tela para dentro. Este padrão é exemplificado na Figura 11.



Figura 11 - Padrão off canvas flyout

Fonte: DEDIS, L. S. A. "Padrões de Design Responsivo" (2017)

2.3.5. The old right-to-left

Este padrão tem as mesmas características do the multi-toggle, diferenciando-se apenas nos subitens. Neste caso, quando item com subitens é selecionado, todo o container com os itens de navegação é transformado para exibir apenas os subitens. A Figura 12 ilustra esse padrão.



Figura 12 - Padrão the old right-to-left

Fonte: DEDIS, L. S. A. "Padrões de Design Responsivo" (2017)

2.3.6. Skip the sub-nav

Neste padrão os subitens de navegação são excluídos da página. Para isso, a página principal de cada item do menu deve conter todas as informações que antes estariam nos subitens.

Esse padrão tem como vantagem simplificar a navegação (já que não teria mais subníveis), porém pode causar a página confusa, por ter informação demais em uma página só.

2.4. React

O React é uma biblioteca JavaScript criada em 2011 pelo Facebook inicialmente para a interface do mural de notícias da rede social. Posteriormente, a biblioteca passou a ser utilizada em várias outras ferramentas da empresa até que, em 2013, teve seu código aberto à comunidade. Na StackOverflow Survey 2017 [2], pesquisa realizada pelo StackOverflow, o React foi considerado o mais amado entre desenvolvedores na categoria Frameworks, bibliotecas e outras tecnologias. A Figura 13 exibe o ranking para essa categoria.

Figura 13 - Ranking de Frameworks, bibliotecas e outras tecnologias mais amadas por desenvolvedores



Most Loved, Dreaded, and Wanted Frameworks, Libraries and Other Technologies

% of developers who are developing with the language or technology and have expressed interest in continuing to develop with it

Fonte: https://insights.stackoverflow.com/survey/2017

Para entender como uma página é criada utilizando o React, é essencial entender o conceito de componentes trazido por ele. Utilizando o React, as interfaces são criadas através da quebra de estrutura em componentes. A ideia é que, ao contrário do HTML, uma página seja dividida em pequenas partes e em arquivos diferentes, podendo ser reutilizadas em qualquer página do site. A organização e a divisão destes componentes fica sob responsabilidade do desenvolvedor. Na Figura 14 podemos ver o site da Udacity dividido em possíveis componentes, numerados em vermelho.



Figura 14 - Site da Udacity e seus componentes

Fonte: https://br.udacity.com/blog/post/react-o-que-e-como-funciona

2.4.1. Create-react-app

Para que uma aplicação React seja inicializada, em desenvolvimento ou em produção, é necessário que alguns scripts específicos de compactação de arquivos sejam executados. Para facilitar o trabalho dos desenvolvedores, existe o create-react-app.

O create-react-app é um pacote público desenvolvido em node.js que tem como objetivo criar toda a estrutura de uma aplicação em React. Ao ser executado, cria um novo diretório contendo todos os arquivos e *scripts* de uma aplicação, sem precisar que

o desenvolvedor tenha conhecimento de todos os *scripts* necessários para sua inicialização.

3. Estudo e Aplicação

3.1. Criando o Portal BSI em React

Nesse projeto, o objetivo do React é modularizar todas as partes repetidas do Portal para simplificar a manutenção e evitar o re-carregamento das mesmas ao navegar entre os diferentes níveis de menu.

Para isso, após a criação da estrutura-base de uma aplicação React utilizando o *create-react-app*, foram criados componentes, arquivos html com conteúdos e um arquivo contendo a estrutura do Portal. Cada um desses será explicado nos próximos ítens.

3.1.1. Arquivo de estrutura

Ao todo, o Portal possui 3 elementos diferentes que utilizam informações de menu, submenu e endereço de conteúdo. São eles: o menu header, o menu aside e o menu off-canvas.

Para tornar a criação de novos itens de menu mais simples, foi criado um arquivo no formato JSON contendo toda a estrutura do Portal. Deste modo, a estrutura de menus do Portal é transformada em uma árvore. A Figura 15 ilustra a estrutura do menu BSI.

Figura 15 - Estrutura do item BSI do menu

```
"bsi": {
   "icon": "iBSI",
   "subs": {
       "bsi": {
           "title": "BSI - UNIRIO",
           "content": "/bsi"
       },
       "historico": {
           "title": "Histórico",
           "content": "/bsi/historico"
       },
       "localizacao": {
           "title": "Localização e Contato",
           "content": "/bsi/localizacao"
       },
       "depoimentos": {
           "title": "Depoimentos",
           "content": "/bsi/depoimentos"
        }
```

Cada item do menu possui:

- Uma propriedade chamada *icon*, que contém o nome do ícone presente na pasta assets/images do projeto;
- Uma propriedade chamada *subs*, contendo todos os submenus relacionados àquele item.

Dentro do objeto *subs*, temos outros objetos com propriedades diferentes, sendo cada um deles um item de submenu. As propriedades são:

- *title*: o título daquele submenu, ou seja, como ele será apresentado no Portal;
- *content*: caminho para acessar esse submenu;
- subs (opcional): caso o submenu tenha mais um nível subordinado à ele, os objetos deste nível serão adicionados nesta propriedade.

Deste modo, para criar um novo subitem de menu no Portal é necessário apenas adicionar esse subitem em seu item pai na árvore, atualizando assim todos os menus.

3.1.2. Arquivos de conteúdo

A parte mais dinâmica do Portal BSI são os conteúdos de cada página. Seja devido à publicações novas de TCCs ou troca de professores, essas páginas estão sempre necessitando de atualização.

Para simplificar a manutenção desses conteúdos, cada um deles foi separado em arquivos individuais no formato HTML. Eles estão presentes no caminho /*assets/portal/contents* do projeto e, por estarem em um formato conhecido, são facilmente modificadas. A Figura 16 ilustra parte da árvore de conteúdos.



Figura 16 - Árvore de conteúdos

3.1.3. Componentes

O Portal BSI será dividido em 4 grandes blocos: cabeçalho, menu aside, conteúdo principal, rodapé e menu *off-canvas*. Cada um destes foi transformado em um

ou mais componentes para a criação do Portal em React. Os componentes criados foram:

- Header.js: contém o cabeçalho principal do Portal;
- HeaderSmall.js: contém o cabeçalho do Portal para dispositivos menores;
- MainContainer.js: abriga o menu aside e o conteúdo principal;
- Aside.js: cria o container para o menu aside do Portal;
- AsideItem.js: cria cada item presente no menu aside;
- AsideSubmenu.js: cria os subitens presentes no menu aside, se necessário;
- Content.js: é responsável por exibir o conteúdo principal, de acordo com a página visitada;
- OffCanvas.js: cria o menu off-canvas do Portal;
- OffCanvasItem.js: exibe cada item presente no menu off-canvas;
- OffCanvasSubitem.js: agrupa, consultando o arquivo de estrutura do Portal, os itens e subitens do Portal e os exibe na ordem correta através do componente OffCanvasItem.js; e
- Footer.js: cria o rodapé do Portal.

3.2. Responsividade do menu de navegação

O primeiro passo escolhido para adequar a página de BSI aos padrões responsivos foi escolher qual layout de menu de navegação seria aplicado para as telas médias, pequenas e extra-pequenas. Antes deste trabalho, o layout de página era o mesmo independente do tamanho da tela do usuário, apenas encolhendo seus elementos à medida que a tela diminuía, dificultando o clique nos quadrados de menu, que se tornavam muito pequenos. A Figura 17 mostra como a página se comportava em diferentes telas.



Figura 17 - Comportamento do Portal BSI em diferentes telas

Dos 10 padrões de navegação responsivos mais populares na web mapeados por Brad Frost em 2012 [3, 4], 6 atendem as necessidades de uma navegação com muitos níveis.

Entre eles, duas opções foram descartadas pois o *container* com os sub-menus teriam que: ou continuar, em telas menores, se comportando do mesmo jeito que se comportaria em telas pequenas (como uma lista do lado esquerdo da tela), prejudicando a leitura do conteúdo principal; ou ser redesenhado para virar um segundo menu de navegação, tornando-se confuso ao usuário. São elas: *Horizontal Scroll* e *The select menu*.

Também foi descartada a opção *Skip the sub-nav* pois, devido à grande quantidade de conteúdo dos sub-menus, não seria amigável colocá-los em apenas uma página.

Com isso, sobraram 3 opções a serem cogitadas: *The multi-toggle, Off-canvas flyout* e *The old right-to-left*. Nos próximos itens apresentamos como cada opção ficaria adequada ao Portal BSI, preservando as cores características de cada menu e, ao final, os prós e contras de cada uma.

3.2.1. The multi-toggle adaptado ao Portal BSI

Como ilustrado na Figura 18, o cabeçalho do Portal, em dispositivos menores, seria composto apenas da logo de BSI e do menu de navegação logo abaixo, ocupando toda a largura da tela.



Figura 18 - Menu de navegação do Portal BSI com *The multi-toggle*

Ao clicar na barra de menu, as opções seriam exibidas abaixo. Para cada item do menu com subitens, seria possível clicar na seta para exibir todas as opções. Esta ação está ilustrada na Figura 19.

Figura 19 - Itens e subitens do menu do Portal BSI com The multi-toggle



3.2.2. The old right-to-left adaptado ao Portal BSI

Assim como em The multi-toggle, a barra de menu ficaria logo abaixo da logo de BSI, como ilustrado na Figura 18.

Ao clicar na barra de menu, as opções também seriam exibidas abaixo. A diferença ficaria ao clicar em algum item com subitens. Neste caso, os mesmos seriam exibidos deslizando para a esquerda e tomando o lugar dos itens principais. Esta ação está ilustrada na Figura 20.

Figura 20 - Itens e subitens do menu do Portal BSI com The old right-to-left



3.2.3. Off-canvas flyout adaptado ao Portal BSI

Como ilustrado na Figura 21, o cabeçalho do Portal, em dispositivos menores, seria composto apenas da logo de BSI e do ícone hamburger, indicando que o menu estaria disponível *off-canvas*.



Figura 21 - Menu de navegação do Portal BSI com Off-canvas flyout

Ao clicar no ícone hamburger, o menu é exibido deslizando para a direita, apresentando as opções e a possibilidade de exibir os subitens. Essa ação é ilustrada na Figura 22.

Figura 22 - Itens e subitens do menu do Portal BSI com Off-canvas flyout



3.2.4. Prós e contras e definição de layout

A Tabela 2 lista os prós e contras de cada um dos padrões apresentados.

	Prós	Contras
The multi-toggle	 Itens e subitens são exibidos juntos, proporcionando ao usuário uma visão geral; Não é necessário fechar o menu para ver o conteúdo principal da página. 	 Se muitos itens com subitens forem abertos, o menu ocuparia um espaço muito maior que a própria tela; O movimento de empurrar o conteúdo principal para baixo pode causar desconforto em alguns usuários.
The old right-to-left	 Não é necessário fechar o menu para ver o conteúdo principal da página. Os subitens são exibidos sozinhos, ajudando a visualizá-los sem confundí-los com os outros itens. 	 O movimento de empurrar o conteúdo principal para baixo pode causar desconforto em alguns usuários; A troca de tamanho constante do menu também pode trazer desconforto; Não é possível ver subitens de mais de um item principal por vez.
Off-canvas flyout	 Padrão mais utilizado entre os dispositivos médios e pequenos atualmente, gerando conforto em ver algo que já é conhecido por todos; Itens e subitens são exibidos juntos, proporcionando ao usuário uma visão geral; Bastante espaço de tela para exibir itens e subitens; Separa o menu do resto da página. 	 Dependendo da quantidade de itens e da altura da tela, um grande espaço vazio será exibido ao final da lista de itens. O movimento de deslizar o menu para a página pode causar desconforto em alguns usuários;

Após consulta com os docentes da Diretoria de Informática Aplicada através de e-mail da diretora Morganna Diniz, ficou-se decidido que o padrão a ser utilizado no Portal BSI seria o *Off-canvas flyout*.

Para a aplicação desse padrão, primeiro seria necessário adequar os outros componentes da página aos padrões responsivos. Para isso, precisaríamos:

- 1. Incorporar o Bootstrap ao projeto (Capítulo 3.2);
- 2. Aplicar o sistema de Grids em todas as páginas (capítulo 3.3);
- Adequar as imagens e vídeos de todas as páginas para não ultrapassarem a largura da tela (capítulo 3.4);
- 4. Aplicar o sistema de Grids no menu para telas grandes (capítulo 3.5);
- Criar e aplicar o *Off-canvas flyout* no menu para dispositivos médios e pequenos (capítulo 3.6);

3.3. Aplicação do sistema de Grids às páginas do Portal BSI

Examinando a figura 13, percebemos que a apresentação do conteúdo presente logo abaixo do menu muda de acordo com o tamanho da tela. Para telas grandes e extragrandes, o conteúdo principal é apresentado ao lado direito e o menu de navegação de subitens é exibido ao lado esquerdo. Já para dispositivos médios e pequenos, apenas o conteúdo principal é exibido, pois os subitens estarão presentes no menu.

Para que o layout fosse devidamente apresentado, adicionamos algumas classes-chave às *divs* que contém o conteúdo principal e o aside que carrega o submenu.

Foi necessário criar uma árvore de *divs* contendo todo o conteúdo onde o *Grid* seria aplicado, para que o Bootstrap reconhecesse que aquele espaço seria dividido pelo sistema de *Grids*. A árvore ficou como abaixo, onde a classe *container* indica que é um sistema de *Grids* e a classe *row* indica que aquela é uma linha deste sistema:

<div class="container">
 <div class="row">

À div com o submenu, foram adicionadas as seguintes classes:

<div class="d-none d-lg-block col-lg-3">

Devido às classes *d-none* e *d-lg-block*, o css do conteúdo desta div é definido como "display: none" para dispositivos com tamanhos menores que o *breakpoint* do dispositivo considerado grande pelo Bootstrap e "display: block" para dispositivos com tamanhos iguais ou maiores que o mesmo *breakpoint*. Com isso, ele não é exibido para dispositivos de extra-pequenos até médios.

Já a classe *col-lg-3* indica que aquele conteúdo tomará 3 colunas das 12 disponíveis no sistema de *Grids*.

À div com o conteúdo principal, foram adicionadas as classes:

<div class="col-12 col-lg-9">

Com elas, indicamos que o aquele conteúdo tomará todas as 12 colunas do sistema de Grids em dispositivos menores que o *breakpoint* do dispositivo grande e 9 colunas em dispositivos maiores.

3.4. Adequação de imagens e vídeos ao tamanho de tela

Devido aos diferentes tamanhos de tela, precisamos tomar cuidado com o tamanho das imagens e vídeos presentes nas páginas.

Para assegurar a largura em caso de vídeos incorporados do YouTube, adicionamos duas linhas de estilo no CSS da tag *iframe*:

```
iframe {
    width: 100%;
    max-width: 560px;
}
```

Com essas propriedades, determinamos que o vídeo deve ocupar toda a largura possível, porém não deve ultrapassar os 560 pixels, para que não fique muito grande.

No caso de imagens, adicionamos:

```
article img{
    max-width: 100%;
}
```

Com isso, asseguramos que mesmo que as imagens sejam maiores que a tela, elas serão reduzidas para terem a largura de no máximo 100% da *div*.

3.5. Aplicação do sistema de *Grids* no menu para dispositivos grandes

Para que todo o portal fosse adaptado ao modelo responsivo, também foi necessário aplicar o sistema de grids do bootstrap no menu já existente.

O sistema de grids do Bootstrap é dividido linha por linha e, como podemos ver no menu presente na Figura 23, existem elementos que ocupam duas linhas.

Figura 23 - Menu do Portal BSI para dispositivos grandes e extragrandes



Fonte: http://bsi.uniriotec.br/bsi/index.html

Para que o sistema de *grids* seja aplicado ao menu, é necessário dividi-lo em dois *containers*, ou seja, dois sistemas de *grid*. Desse modo, como representado na Figura 24, teremos um *container* abrigando todo o conteúdo do menu e outro abrigando apenas os elementos que estão divididos em duas linhas, representado na Figura 25.

Figura 24 - Container abrigando todo o conteúdo do menu





Figura 25 - Container abrigando apenas os elementos divididos em duas linhas

Também podemos observar nas Figuras 18 e 19 a largura de cada coluna do grid. Na Figura 18 podemos perceber que, como a largura de todos os elementos pequenos juntos é igual à largura dos dois elementos grandes, os elementos grandes recebem largura 3 de 12 cada um e o conjunto dos pequenos recebe largura 6. Já na Figura 19, percebemos que um dos elementos ocupa o dobro do tamanho dos demais, e a largura de uma linha é a soma da largura de 4 elementos pequenos. Por isso, o elemento mais largo recebeu largura 6 e os demais largura 3.

Como os elementos do menu, assim como o aside com os submenus, serão exibidos apenas para dispositivos grandes ou maiores, também foi necessário adicionar as classes de *display* ao *header*.

<header class="d-none d-lg-block">

3.6. Criação e aplicação do *Off-canvas flyout* para dispositivos médios ou menores

Para aplicar o padrão Off-canvas flyout ao Portal BSI, identificamos duas mudanças necessárias: a transição do *header* grande para um *header* com apenas uma barra e o ícone hamburger quando a tela do dispositivo chegasse ao tamanho médio ou menor e a criação da parte "off-canvas" do menu, utilizando os subníveis de menu presentes anteriormente em menu *aside*.

3.6.1. Criação e transição para *header* menor

Em dispositivos de tela média ou menor, o header será composto por uma barra da cor padrão do elemento BSI do menu e deve conter a logo de BSI ao centro e o ícone hamburger à esquerda, indicando o menu presente fora da tela. A Figura 26 ilustra como o menu ficará.

Figura 26 - Header em dispositivos médios ou menores

Para adequar a propriedade display do header, ou seja, para que ele seja exibido somente em telas médias ou menores, o header foi declarado da seguinte forma:

<header class="d-xs-block d-lg-none">

O alinhamento dos elementos foi feito a partir da classe *justify-content-between*, disponível para elementos a propriedade *display* com valor *flex*.

<div class="d-flex justify-content-between">

Entretanto, para que eles fossem alinhados corretamente, um terceiro elemento deveria estar na mesma linha. Para que isso fosse solucionado, criamos uma div com largura igual à do ícone hamburger.

<div style="width: 40px;"></div>

4. Resultados

4.1. Padrão off-canvas flyout

Visualmente, para dispositivos maiores que o *breakpoint* médio do *Bootstrap*, o Portal BSI continuará o mesmo. Entretanto, para dispositivos menores, o padrão foi aplicado de acordo com o Capítulo 3. Para as comparações, usaremos o dispositivo iPhone 6, com tela 375 por 667 pixels, e a página principal do Portal.

Ao abrir qualquer página do Portal, a visão do menu será como na Figura 27, que mostra em comparação a visão de hoje e a visão após o desenvolvimento.



Figura 27 - Antes x Depois da visão principal da página

A diferença entre a expectativa e a visão resultado se dá principalmente ao tamanho dos componentes. Ao aplicar exatamente a visão esperada, os elementos ficaram pequenos, podendo gerar desconforto ao tentar lê-los.

Para o menu off-canvas aberto, temos a Figura 28.



Figura 28 - Antes x Depois do menu off-canvas aberto

Assim como na visão anterior, os elementos foram aumentados devido à proporção com a tela pequena. Entretanto, neste caso, a dificuldade de clicar nos elementos do menu para expandi-los também teve um peso significante na escolha de tamanhos.

Para o menu off-canvas com subitem expandido, temos a Figura 29.



Figura 29 - Antes x Depois do menu off-canvas expandido

Assim como os anteriores, a diferença fica a cargo das proporções da tela.

4.2. Portal BSI em React

4.2.1. Performance

Os resultados alcançados ao transformar uma página em um React app são principalmente em performance. Porém ao transformar o Portal BSI em uma aplicação, podemos perceber na verdade uma perda muito pequena de performance.

Para fazer a comparação de performance entre as versões do Portal, ambas foram publicadas localmente, em um notebook Macbook Air, com processador Intel Core i7, 8 GB de memória RAM e com a versão 10.13.4 do sistema operacional macOS High Sierra instalado. Para os testes, foi feita a navegação da página principal do Portal até a página da disciplina Desenvolvimento de Páginas Web e, após, novamente para a página principal, com *cache* e *cookies* recém-apagados. A tabela 3 mostra o resultado da comparação de performance entre o *React app* e o Portal original.

	Portal em React	Portal original
Página principal (/bsi)	1.9s	1.1s
Disciplinas (/disciplinas)	0.3s	0.19s
Obrigatórias (/disciplinas/obrigatorias)	0.3s	0.17s
Desenvolvimento de Páginas Web (/ementas/dpw)	0.27s	0.19s
Página principal (/bsi)	1.05s	0.6s
Tempo total de carregamento	3.82s	2.25s

Tabela 3 - Comparação de Performance

4.2.2. Manutenção

Outro ganho do React app é a facilidade de manutenção, devido à modularização dos elementos. Quando um elemento que se repete em diversas páginas está presente em apenas um arquivo, é necessário apenas alterá-lo neste arquivo, e não no arquivo de todas as páginas que contém aquele elemento. Com isso, a quantidade de linhas de código a serem alteradas em uma eventual mudança é significativamente maior em páginas sem modularização. A Tabela 4 quantifica as linhas que eventualmente serão alteradas em cada projeto, ou seja, linhas de páginas com extensão HTML e JS.

Tabela 4 - Quantidade de linhas em cada projeto

	Portal em React	Portal original
Quantidade de linhas em HTML	773	63451
Quantidade de linhas em JavaScript	6825	99
Total de linhas	7598	63550

Porém, caso o usuário não tenha experiência e facilidade em desenvolvimento utilizando a linguagem node.js, o processo de mudança do aplicativo se torna mais complicado. Como exemplo, temos nos Anexos 1 e 2 tutoriais para adição de um submenu ao Portal e nos Anexos 3 e 4 tutoriais para alteração de conteúdo. Podemos ver que, em ambos os casos, o processo de manutenção é mais simples no Portal em HTML.

4.2.3. Perda de Funcionalidades

Com a transformação do Portal em um React App, algumas funcionalidades foram perdidas devido à dificuldade do React em lidar com funções, tanto em JavaScript quanto em JQuery, que manipulam os elementos da página declaradas fora do escopo do componente (por exemplo, funções declaradas diretamente no arquivo html de conteúdo). Como exemplo dessa perda, temos a parte de notícias do Portal. No Portal original, conforme a Figura 30, um elemento com as últimas postagens da página BSI no Facebook é apresentada. Já no Portal React, como na Figura 31, esse elemento não é apresentado pois o script para criação não é executado.



Figura 30 - Página de Notícias do Portal BSI original

Fonte: http://bsi.uniriotec.br/notícias/index.html



Figura 31 - Página de Notícias do Portal BSI em React

Outro exemplo dessa perda é a página de Calendário Acadêmico. No Portal original, um script JavaScript é chamado para criar a visão do calendário, como na Figura 32. Já no Portal React, esse script não é chamado com sucesso e o calendário não é apresentado na página, conforme Figura 33.



Figura 32 - Página Calendário Acadêmico do Portal BSI original

Fonte: http://bsi.uniriotec.br/alunos/calendario.html



Figura 33 - Página Calendário Acadêmico do Portal BSI em React

Já com a adaptação do Portal para dispositivos móveis, a capacidade de clicar em uma disciplina e direcionar o usuário para sua página, na página de Disciplinas, foi perdida.

Como os locais de clique foram mapeados utilizando encontros de pixels específicos entre altura e largura através da tag *map*, quando a imagem é reduzida para se encaixar em dispositivos com telas menores, os locais de clique mapeados não são redimensionados, criando uma área incorreta de cliques.

5. Conclusão

5.1. Considerações Finais

Este projeto teve como objetivo melhorar a experiência do usuário ao acessar o Portal BSI, independente do tamanho do dispositivo utilizado. Durante o desenvolvimento, duas frentes foram apresentadas: melhorar a experiência de usuários que utilizam dispositivos como tablets e celulares, utilizando os padrões responsivos aplicados com Bootstrap; e melhorar a performance do Portal, otimizando a criação dos elementos, utilizando a biblioteca de JavaScript React.

Através da versão do Portal BSI desenvolvida a partir deste TCC, que adequa o Portal ao padrão responsivo utilizando o Bootstrap, conseguimos melhorar a experiência do usuário que acessa o Portal utilizando dispositivos móveis, como tablets e celulares. Esses usuários agora irão acessar uma visão diferente do Portal, adequada para o tamanho de sua tela.

Já a melhora na performance do Portal não foi alcançada utilizando a biblioteca React. Durante o projeto vimos que a transformação do Portal, antes puramente utilizando páginas HTML, em um React app acabou sendo prejudicial à sua performance. Além disso, funcionalidades foram perdidas e a manutenção do Portal ficou mais complicada. Os prós e contras das duas versões do projeto são apresentadas na Tabela 5.

	Portal em React	Portal sem React	
Prós	 Menos linhas de código Manutenção de menus facilitada devido à modularização 	 Linguagem simplificada Melhor performance 	
Contras	 É necessário conhecimento da linguagem, do framework e dos processos de alteração e publicação Perda de funcionalidades 	 Quantidade excessiva de linhas de código Repetição de códigos iguais em diversas páginas 	

Tabela 5 - Prós e contras do Portal em React e sem React

Hoje, a equipe que altera e cuida do Portal BSI é composta apenas por professores da instituição, com conhecimentos em HTML porém não aprofundados para frameworks mais recentes, como o React. Além disso, o cenário como um todo é muito volátil pois, a cada quatro anos a diretoria da Escola de Informática Aplicada é escolhida, podendo ser eleita uma nova diretoria que tenha ainda menos conhecimento em sistemas web. Por esse motivo chegamos a conclusão que, para o caso do Portal BSI, a facilidade do entendimento da linguagem é prioridade, optando assim por não incluir a transformação em React app na versão final do Portal BSI entregue por esse projeto.

5.2. Trabalhos Futuros

Como trabalhos futuros, sugerimos:

- teste de usabilidade da nova versão do Portal BSI;
- modularização do Portal BSI utilizando outros *frameworks* ou bibliotecas, visando melhorar a performance e simplificar a manutenção; e
- análise de impactos positivos e negativos da modularização de páginas web.

Referências Bibliográficas

 [1] INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Acesso à Internet e à Televisão e Posse de Telefone Móvel Celular para Uso Pessoal - Pesquisa Nacional por Amostra de Domicílios. - Rio de Janeiro: IBGE, 2014. Disponível em:
 https://biblioteca.ibge.gov.br/visualizacao/livros/liv95753.pdf>. Acesso em: 2 jul.
 2018

[2] STACK OVERFLOW. Developer Survey Results 2017. Disponível em: <<u>https://insights.stackoverflow.com/survey/2017</u>>. Acesso em: 2 jul. 2018

[3] FROST, B. Responsive Navigation Patterns. Disponível em: <<u>http://bradfrost.com/blog/web/responsive-nav-patterns/</u>>. Acesso em: 2 jul. 2018

[4] FROST, B. Complex Navigation Patterns for Responsive Design. Disponível em:
 <<u>http://bradfrost.com/blog/web/complex-navigation-patterns-for-responsive-design/</u>>.
 Acesso em: 2 jul. 2018

[5] DICIO. Significado de responsividade. Disponível em: <<u>https://www.dicio.com.br/responsividade/</u>>. Acesso em: 2 jul. 2018

[6] DEDIS, L. S. A. Padrões de Design Responsivo - Rio de Janeiro: UNIRIO, 2017.
 Disponível em: <<u>http://bsi.uniriotec.br/tcc/textos/201706LuciannaSantos.pdf</u>>
 Acesso em: 22 jun. 2018

DE NITTIS, L. Entenda o que é layout responsivo e como ele auxilia nos resultados do seu negócio. Disponível em: <<u>http://www.goomidia.com.br/blog/layout-responsivo/</u>>. Acesso em: 22 jun. 2018

ANAND, R. Bootstrap 4 vs Foundation 6 Grid System. Disponível em: <<u>http://blog.gridbox.io/2018/01/08/bootstrap-4-vs-foundation-6-grid-system/</u>>. Acesso em: 22 jun. 2018

BOOTSTRAP. Documentation. Disponível em: <<u>https://getbootstrap.com/docs/</u>>. Acesso em: 22 jun. 2018

W3SCHOOLS. CSS display Property. Disponível em: <<u>https://www.w3schools.com/cssref/pr_class_display.asp</u>>. Acesso em: 26 jun. 2018

UDACITY BRASIL. React: o que é e como funciona essa ferramenta?. Disponível em: <<u>https://br.udacity.com/blog/post/react-o-que-e-como-funciona</u>> Acesso em: 2 jul. 2018

FACEBOOK. Create React App. Disponível em: <<u>https://github.com/facebook/create-react-app</u>> Acesso em: 2 jul. 2018

ANEXO I - Passo-a-passo para adicionar um submenu ao Portal BSI original

- 1. Entre no diretório onde o Portal está hospedado;
- Navegue até o diretório do menu onde deseja adicionar o submenu (por exemplo, a pasta bsi);
- Faça uma cópia do arquivo index.html e renomeie-o com o nome do seu novo submenu;
- 4. Edite essa cópia colocando o conteúdo desejado;
- 5. Abra o arquivo index.html;
- Dentro da tag aside, adicione mais um item de lista (), copiando o modelo dos já existentes e alterando o seu título e caminho de acordo com o arquivo criado no passo 3;
- 7. Repita o passo 6 para todos os arquivos html dentro deste diretório;
- Abra o Portal e certifique-se de que o novo submenu está sendo apresentado como desejado.

ANEXO II - Passo-a-passo para adicionar um submenu ao Portal BSI em React

- Faça download da última versão do projeto React do Portal (idealmente presente em um repositório GIT);
- 2. Abra o terminal e navegue até a pasta do projeto;
- Insira o comando npm start, para que o servidor para desenvolvimento seja iniciado;
- 4. Dentro do diretório /src/assets/contents, procure a pasta do menu onde deseja adicionar o submenu e crie um novo arquivo .html;
- Preencha este arquivo com o conteúdo que deseja que seja exibido na página criada;
- 6. Dentro da pasta src/, abra o arquivo structure.json;
- Procure o objeto do menu onde deseja adicionar o submenu e adicione um novo objeto em sua propriedade subs, seguindo o modelo dos anteriores;
- Entre no endereço indicado no passo 3 e certifique-se de que seu novo menu está sendo apresentado corretamente;
- Caso sim, volte ao terminal e pressione Ctrl+C para matar o processo do servidor;
- 10. Insira o comando npm run build, para que seja criado o build a ser inserido no servidor do Portal;
- Copie o conteúdo pasta build/ do projeto e cole-a no diretório do servidor onde o Portal está hospedado.

ANEXO III - Passo-a-passo para editar um conteúdo do Portal BSI original

- 1. Entre no diretório onde o Portal está hospedado;
- 2. Navegue até o conteúdo que deseja alterar e abra-o;
- 3. Faça as edições necessárias;
- 4. Abra o Portal e certifique-se de que o conteúdo está sendo apresentado como desejado.

ANEXO IV - Passo-a-passo para editar um conteúdo do Portal BSI em React

- Faça download da última versão do projeto React do Portal (idealmente presente em um repositório GIT);
- 2. Abra o terminal e navegue até a pasta do projeto;
- Insira o comando npm start, para que o servidor para desenvolvimento seja iniciado;
- Dentro do diretório /src/assets/contents, procure o conteúdo que deseja editar e abra-o;
- 5. Faça as edições necessárias;
- Entre no endereço indicado no passo 3 e certifique-se de que seu novo menu está sendo apresentado corretamente;
- Caso sim, volte ao terminal e pressione Ctrl+C para matar o processo do servidor;
- 8. Insira o comando npm run build, para que seja criado o build a ser inserido no servidor do Portal;
- Copie o conteúdo pasta build/ do projeto e cole-a no diretório do servidor onde o Portal está hospedado.