

## Print erros

1º commit: Clone repositório:

[https://github.com/MilenaBertolini/landing\\_page\\_psico\\_andressa](https://github.com/MilenaBertolini/landing_page_psico_andressa)

2º Commit: Ajustando os erros abaixo do componente AboutMeSection.jsx

### 3 Erros componente AboutMe



### Functions should not be nested too deeply (javascript:S2004)

Adaptability issue | Not focused | **Maintainability** | What is clean code?

Why is this an issue? | More Info

Nested functions refer to the practice of defining a function within another function. These inner functions have access to the variables and parameters of the outer function, creating a closure.

While nesting functions is a common practice in JavaScript, deeply nested functions can make the code harder to read and understand, especially if the functions are long or if there are many levels of nesting.

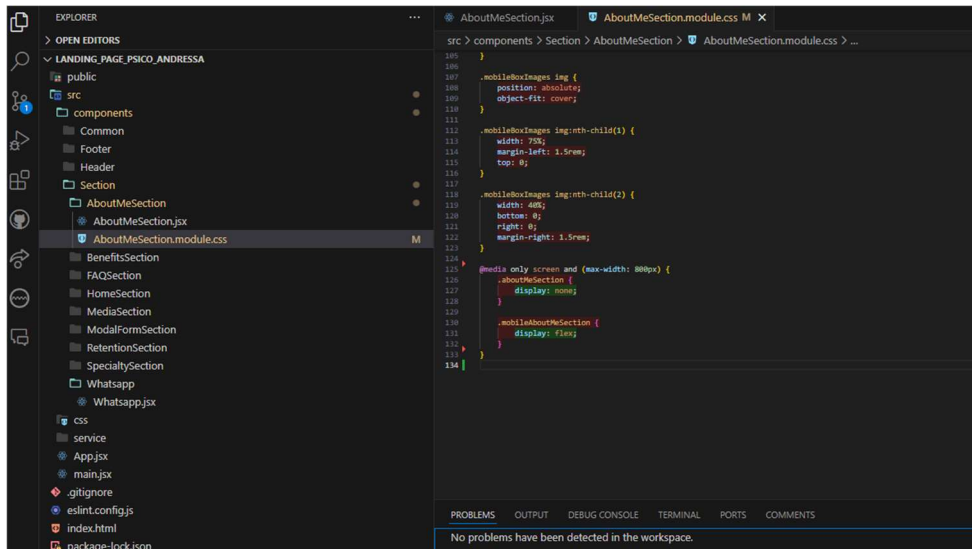
This can make it difficult for other developers or even yourself to understand and maintain the code.

#### Noncompliant code example

With the default threshold of 4 levels:

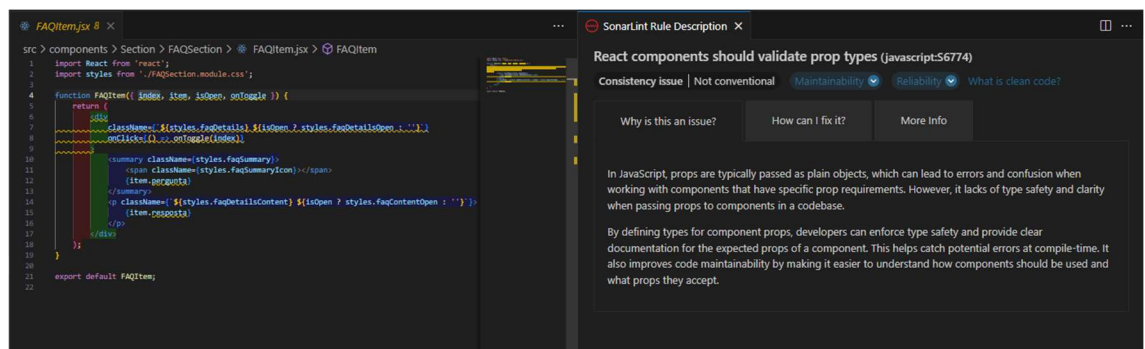
```
function f() {
  function f_inner() {
    function f_inner_inner() {
      function f_inner_inner_inner() {
        function f_inner_inner_inner_inner() { // Noncompliant
        }
      }
    }
  }
}
```





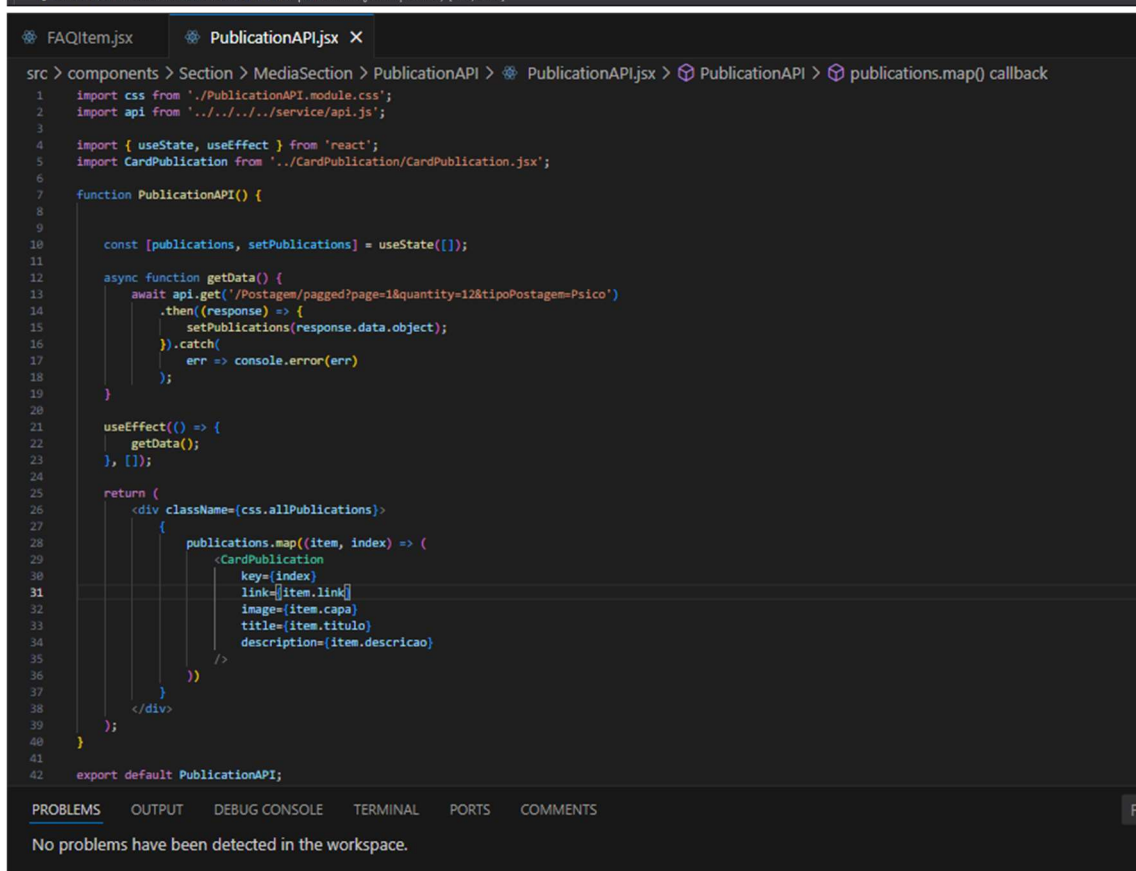
#### Commit 4:

1. 'index' is missing in props validation
2. 'isOpen' is missing in props validation
3. Avoid non-native interactive elements. If using native HTML is not possible, add an appropriate role and support for tabbing, mouse, keyboard, and touch inputs to an interactive content element.
4. Visible, non-interactive elements with click handlers must have at least one keyboard listener.
5. 'item.pergunta' is missing in props validation



#### Commit 5:

useState call is not destructured into value + setter pair



## Commit 6

Remove the declaration of the unused 'isMobile' variable.

```
Whatsapp.jsx 2 X
src > components > Section > Whatsapp > Whatsapp.jsx > Whatsapp > isMobile
1 import { useEffect, useState } from 'react';
2 import { FloatingWhatsapp } from 'react-floating-whatsapp';
3
4 export default function Whatsapp() {
5   const [isMobile, setIsMobile] = useState(false);
6
7   // Verifica o tamanho da tela para ajustar a posição do botão
8   useEffect(() => {
9     const handleResize = () => {
10       setIsMobile(window.innerWidth <= 768); // Define como mobile se a largura da tela for menor ou igual a 768px
11     };
12
13     handleResize(); // Executa na primeira renderização
14     window.addEventListener('resize', handleResize); // Adiciona o listener para o resize
15
16     return () => window.removeEventListener('resize', handleResize); // Remove o listener ao desmontar o componente
17   }, []);
18   return (
19     <FloatingWhatsapp
20       phoneNumber='1553496545701' // Número do Whatsapp em formato internacional para onde as mensagens serão enviadas
21       accountName='Psicóloga Andressa' // Nome da conta exibido no chat, pode incluir emojis para personalização
22       chatMessage='Olá! 🌟 Seja bem-vindo(a) ao nosso espaço de acolhimento e apoio. Vamos juntos trabalhar para que você se sinta melhor. Como posso te ajudar?' // Mensagem exibida na caixa de chat ao abrir
23       statusMessage='Disponível' // Mensagem de status exibida abaixo do nome da conta, informando a disponibilidade
24       placeholder='Digite a sua mensagem' // Texto do placeholder que aparece na caixa de entrada de texto
25       allowClickAway={true} // Permite que o chat seja fechado ao clicar fora dele
26       allowEsc={true} // Permite que o chat seja fechado ao pressionar a tecla escape
27       darkMode={false} // Define o tema do chat: false para claro, true para escuro
28       messageDelay={2} // Tempo de atraso em segundos para a exibição da mensagem de chat inicial
29       notification={true} // Habilita notificações que aparecem antes do usuário abrir o chat
30       notificationDelay={10} // Intervalo de tempo em segundos entre as notificações repetitivas
31       notificationSound={true} // Habilita o som de notificação quando a notificação é exibida
32       chatboxHeight={360} // Define a altura da caixa de chat em pixels
33       avatar='images/whatsapp_Andressa.png' // Aqui não definimos o avatar, então o avatar padrão será utilizado
34     />
35   );
36 }
```

Commit 7:

Replace this character class by the character itself.

ModalContent.jsx 1

```
src > components > Section > ModalFormSection > ModalContent.jsx > ModalContent > validateForm >
5 function ModalContent() {
25   const handleSubmit = (event) => {
59     .catch((error) => {
63       });
64     } else {
65       // Se a validação falhar, exibe uma mensagem de erro
66       setGeneralErrorMessage("Por favor, corrija os erros acima antes de enviar.");
67       setSuccessMessage("");
68     }
69   };
70
71   const validateForm = () => {
72     let valid = true;
73     let newErrors = {};
74
75     if (form.name.length < 3) {
76       newErrors.name = "O nome deve ter pelo menos 3 caracteres.";
77       valid = false;
78     }
79
80     const phoneRegex = /\(?(\d{2})\)? \d{4,5}-\d{4}$/;
81     if (!phoneRegex.test(form.phone1)) {
82       newErrors.phone1 = "O telefone deve estar no formato (99) 99999-9999.";
83       valid = false;
84     }
85
86     const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
87     if (!emailRegex.test(form.email)) {
88       newErrors.email = "Por favor, insira um e-mail válido.";
89       valid = false;
90     }
91
92     setErrors(newErrors);
93     return valid;
94   };
95
96   return (
97     <div className={styles.modalContent}>
98       <div className={styles.imageContainer}></div>
99       <div className={styles.formContainer}>
100         <h2>Entre em contato</h2>
101       </div>
102     </div>
103   );
104 }
105
106 export default ModalContent;
```

SonarLint Rule Description

### Character classes in regular expressions should not contain only one character (javascript:S6397)

Intentionality issue | Not clear | Maintainability | What is clean code?

Why is this an issue?

Character classes in regular expressions are a convenient way to match one of several possible characters by listing the allowed characters or ranges of characters. If a character class contains only one character, the effect is the same as just writing the character without a character class.

Thus, having only one character in a character class is usually a simple oversight that remained after removing other characters of the class.

**Noncompliant code example**

```
/a[b]c/
/[^\^]/
```

**Compliant solution**

```
/abc/
/^\^/
/[*]c/ // Compliant, see Exceptions
```

**Exceptions**

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

COMMENTS

Replace this character class by the character itself. sonarlint(javascript:S6397) [Ln 81, Col 54]

ModalContent.jsx M

```
src > components > Section > ModalFormSection > ModalContent.jsx > ModalContent > validateForm
5 function ModalContent() {
25   const handleSubmit = (event) => {
59     .catch((error) => {
63       });
64     } else {
65       // Se a validação falhar, exibe uma mensagem de erro
66       setGeneralErrorMessage("Por favor, corrija os erros acima antes de enviar.");
67       setSuccessMessage("");
68     }
69   };
70
71   const validateForm = () => {
72     let valid = true;
73     let newErrors = {};
74
75     if (form.name.length < 3) {
76       newErrors.name = "O nome deve ter pelo menos 3 caracteres.";
77       valid = false;
78     }
79
80     const phoneRegex = /\(?(\d{2})\)? \d{4,5}-\d{4}$/;
81     if (!phoneRegex.test(form.phone1)) {
82       newErrors.phone1 = "O telefone deve estar no formato (99) 99999-9999.";
83       valid = false;
84     }
85
86     const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
87     if (!emailRegex.test(form.email)) {
88       newErrors.email = "Por favor, insira um e-mail válido.";
89       valid = false;
90     }
91
92     setErrors(newErrors);
93     return valid;
94   };
95
96   return (
97     <div className={styles.modalContent}>
98       <div className={styles.imageContainer}></div>
99       <div className={styles.formContainer}>
100         <h2>Entre em contato</h2>
101       </div>
102     </div>
103   );
104 }
105
106 export default ModalContent;
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

COMMENTS

No problems have been detected in the workspace.

Commit 8:



The screenshot shows the VS Code editor with a SonarLint rule description panel on the right and a problems list on the left.

**SonarLint Rule Description: React components should validate prop types (javascript:S6774)**

Consistency issue | Not conventional | **Maintainability** | Reliability | What is clean code?

Why is this an issue? | How can I fix it? | More Info

In JavaScript, props are typically passed as plain objects, which can lead to errors and confusion when working with components that have specific prop requirements. However, it lacks of type safety and clarity when passing props to components in a codebase.

By defining types for component props, developers can enforce type safety and provide clear documentation for the expected props of a component. This helps catch potential errors at compile-time. It also improves code maintainability by making it easier to understand how components should be used and what props they accept.

**PROBLEMS (19)**

- ModalForm.jsx src/components/Section/ModalFormSection 18
  - form' is missing in props validation sonarlint(javascript:S6774) [Ln 4, Col 22]
  - 'errors' is missing in props validation sonarlint(javascript:S6774) [Ln 4, Col 28]
  - 'successMessage' is missing in props validation sonarlint(javascript:S6774) [Ln 4, Col 36]
  - 'generalErrorMessage' is missing in props validation sonarlint(javascript:S6774) [Ln 4, Col 52]
  - 'onInputChange' is missing in props validation sonarlint(javascript:S6774) [Ln 4, Col 73]
  - 'onSubmit' is missing in props validation sonarlint(javascript:S6774) [Ln 4, Col 88]
  - 'form.name' is missing in props validation sonarlint(javascript:S6774) [Ln 14, Col 37]
  - 'errors.name' is missing in props validation sonarlint(javascript:S6774) [Ln 19, Col 64]
  - 'errors.name' is missing in props validation sonarlint(javascript:S6774) [Ln 19, Col 102]
  - 'form.phone1' is missing in props validation sonarlint(javascript:S6774) [Ln 29, Col 37]
  - 'errors.phone1' is missing in props validation sonarlint(javascript:S6774) [Ln 34, Col 64]
  - 'errors.phone1' is missing in props validation sonarlint(javascript:S6774) [Ln 34, Col 104]
  - 'form.email' is missing in props validation sonarlint(javascript:S6774) [Ln 44, Col 37]
  - 'errors.email' is missing in props validation sonarlint(javascript:S6774) [Ln 48, Col 64]
  - 'errors.email' is missing in props validation sonarlint(javascript:S6774) [Ln 49, Col 103]
  - 'form.message' is missing in props validation sonarlint(javascript:S6774) [Ln 58, Col 37]
  - 'errors.message' is missing in props validation sonarlint(javascript:S6774) [Ln 65, Col 64]
  - 'errors.message' is missing in props validation sonarlint(javascript:S6774) [Ln 65, Col 105]
- ModalOverlay.jsx src/components/Section/ModalFormSection 1

form' is missing in props validation

'errors' is missing in props validation

'onSubmit' is missing in props validation

'onInputChange' is missing in props validation