

Nome do solicitante:	Murillo de Moraes
Analistas responsáveis:	Talis Aparecido dos Reis João Cruz Receba Rodrigues Guilherme Lima

Descrição Solicitação	Data
<p>The Reading List An object-oriented book-list!</p> <ol style="list-style-type: none">1. Create a class BookList.2. Create another class called Book.3. BookLists should have the following properties:<ul style="list-style-type: none">• Number of books marked as read;• Number of books marked not read yet;• A reference to the next book to read (book object);• A reference to the current book being read (book object);• A reference to the last book read (book object);• An array of all the Books.4. Each Book should have several properties:<ul style="list-style-type: none">• Title;• Genre;• Author;• Read (true or false);• Read date, can be blank, otherwise needs to be a JS Date() object.5. Every Booklist should have a few methods:<ul style="list-style-type: none">• .add(book):<ul style="list-style-type: none">◦ should add a book to the books list.• .finishCurrentBook():<ul style="list-style-type: none">◦ should mark the book that is currently being read as "read";◦ Give it a read date of new Date(Date.now());	15/10/2021

- Change the last book read to be the book that just got finished;
- Change the current book to be the next book to be read;
- Change the next book to be read property to be the first unread book you find in the list of books.

6. Booklists and Books might need more methods than that. Try to think of more that might be useful.

Descrição Desenvolvimento

Inicialmente foram criadas duas classes, BookList e Book no método construtor da classe BookList foram inicializados os atributos readBook, notReadBook, nextBook, currentBook, lastBook e allBooks todos com valores indefinidos (vazios ou zerados) para que no decorrer do código possamos alimentá-los com informações dos livros.

Na classe book foram declarados os atributos title, genre, author, read e readDate, ao instanciar os livros e uma lista de livros podemos utilizar os métodos da classe BookList sendo eles:

```
add (newBook) {
  this.allBooks = [...this.allBooks, newBook];
  this.currentBook = this.allBooks.find((current) => current.read === false);
  this._attNotReadBooks();
}

_attNotReadBooks() {
  this.notReadBook = this.allBooks.reduce((acc, curr) => {
    !curr.read ? (acc += 1) : acc;
    return acc;
  }, 0);
}

finishCurrentBook() {
  if (this.currentBook) {
    this.currentBook.read = true;
    this.currentBook.readDate = new Date(Date.now());
    let next = this.allBooks.filter((current) => !current.read);
    this.nextBook = next[1];
    this.lastBook = this.currentBook;
    this.currentBook = next[0];
    this.readBook = this.allBooks.reduce((acc, curr) => {
      curr.read ? (acc += 1) : acc;
      return acc;
    }, 0);
    this._attNotReadBooks();
  } else { return "book list is over"; }
```

Método add recebe livro por livro e adiciona no array allBooks, após ter armazenado todos livros no array o mesmo faz uma busca pelo primeiro objeto do array que possui a propriedade read como falso para dar início a leitura e configura assim o primeiro livro no atributo currentBook, em seguida invoca o método attNotReadBooks que realiza e retorna a contagem de livros ainda não lidos e armazena no atributo notReadBook.

Já o método `finishCurrentBook` verifica se o atributo `currentBook` é verdade, ou seja, se existe algum livro sendo lido atualmente, se o retorno dessa condição for verdadeiro a função altera as propriedades `read`, `readDate` de `currentBook` para verdadeiro e para a data atual, em seguida realiza um filtro no array `allBooks` retornando apenas os objetos que possuem a propriedade `read` como falso armazenando o retorno em na variável `next`.

Posteriormente o atributo `nextBook` recebe o índice um da variável `next` setando assim o próximo livro que será lido, `lastBook` recebe o `currentBook` e passa a ser o último livro a ser lido, já o `currentBook` recebe o índice zero da variável `next`, que passa a ser o livro atual. Dentro da função também é utilizado o `reduce` para realizar a contagem dos livros lidos, armazenando esse valor no atributo `readBook`, também existe a chamada da função `attNotReadBooks` que como já citado anteriormente realiza a contagem de livros ainda não lidos.

Por fim dentro da função `finishCurrentBook` existe uma condição que se caso não existir livro atual, significa que o usuário já percorreu toda lista, assim o método retorna a mensagem "book list is over" para que o usuário insira novos livros a sua lista.

Informações para testes

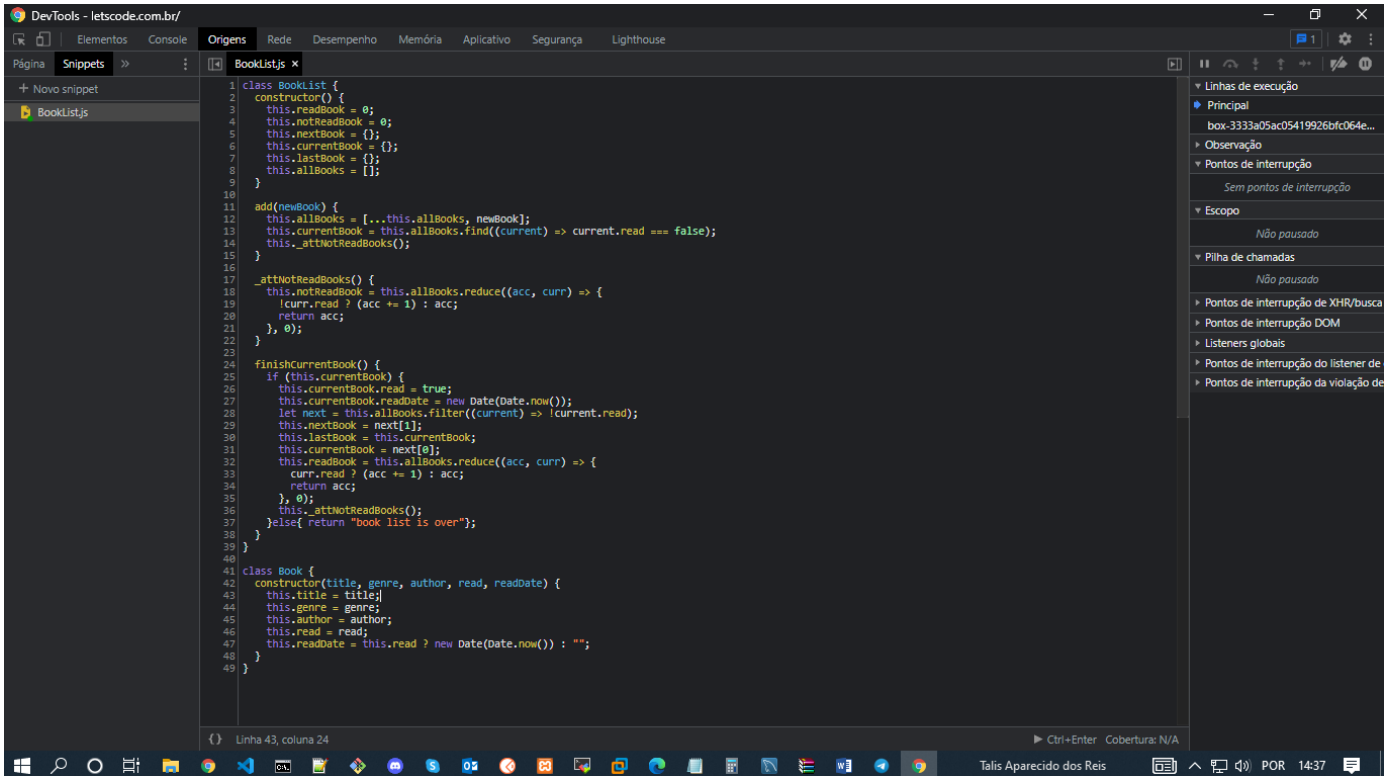
```
const book1 = new Book('Harry Potter', 'Aventura', 'J.K Rowling', false);
const book2 = new Book('Dom Quixote', 'Ação', 'Miguel de Cervantes', false);
const book3 = new Book('Guerra e Paz', 'Aventura', 'Liev Tolstói', false);
const book4 = new Book('A Montanha Mágica', 'Romance', 'Thomas Mann', false);
const book5 = new Book('Cem Anos de Solidão', 'Terror', 'Gabriel García Márquez', false);
const book6 = new Book('Ulisses', 'Aventura', 'James', false);
```

```
const novaListaDeLivros = new BookList()
```

```
novaListaDeLivros.add(book1)
novaListaDeLivros.add(book2)
novaListaDeLivros.add(book3)
novaListaDeLivros.add(book4)
novaListaDeLivros.add(book5)
novaListaDeLivros.add(book6)
```

```
novaListaDeLivros.finishCurrentBook()
novaListaDeLivros.finishCurrentBook()
```

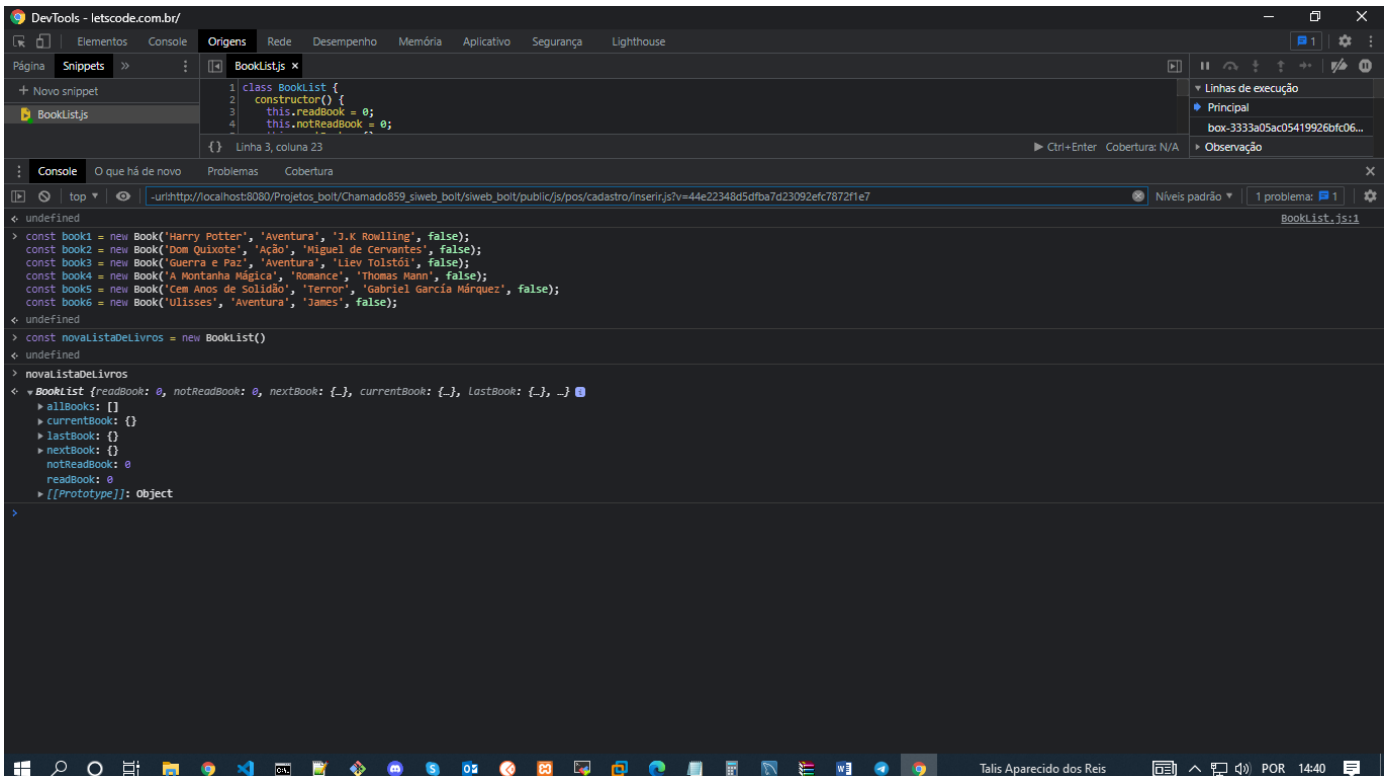
Evidencias:



```

1 class BookList {
2   constructor() {
3     this.readBook = 0;
4     this.notReadBook = 0;
5     this.nextBook = {};
6     this.currentBook = {};
7     this.lastBook = {};
8     this.allBooks = [];
9   }
10
11   add(newBook) {
12     this.allBooks = [...this.allBooks, newBook];
13     this.currentBook = this.allBooks.find((current) => current.read === false);
14     this._attNotReadBooks();
15   }
16
17   _attNotReadBooks() {
18     this.currentBook.read = this.allBooks.reduce((acc, curr) => {
19       !curr.read ? (acc += 1) : acc;
20     }, 0);
21   }
22
23   finishCurrentBook() {
24     if (this.currentBook) {
25       this.currentBook.read = true;
26       this.currentBook.readDate = new Date(Date.now());
27       let next = this.allBooks.filter((current) => !current.read);
28       this.nextBook = next[0];
29       this.lastBook = this.currentBook;
30       this.currentBook = next[0];
31       this.readBook = this.allBooks.reduce((acc, curr) => {
32         curr.read ? (acc += 1) : acc;
33       }, 0);
34       return acc;
35     }
36     this._attNotReadBooks();
37     else{ return "book list is over"; }
38   }
39 }
40
41 class Book {
42   constructor(title, genre, author, read, readDate) {
43     this.title = title;
44     this.genre = genre;
45     this.author = author;
46     this.read = read;
47     this.readDate = this.read ? new Date(Date.now()) : "";
48   }
49 }
  
```

Imagem 1: código completo.

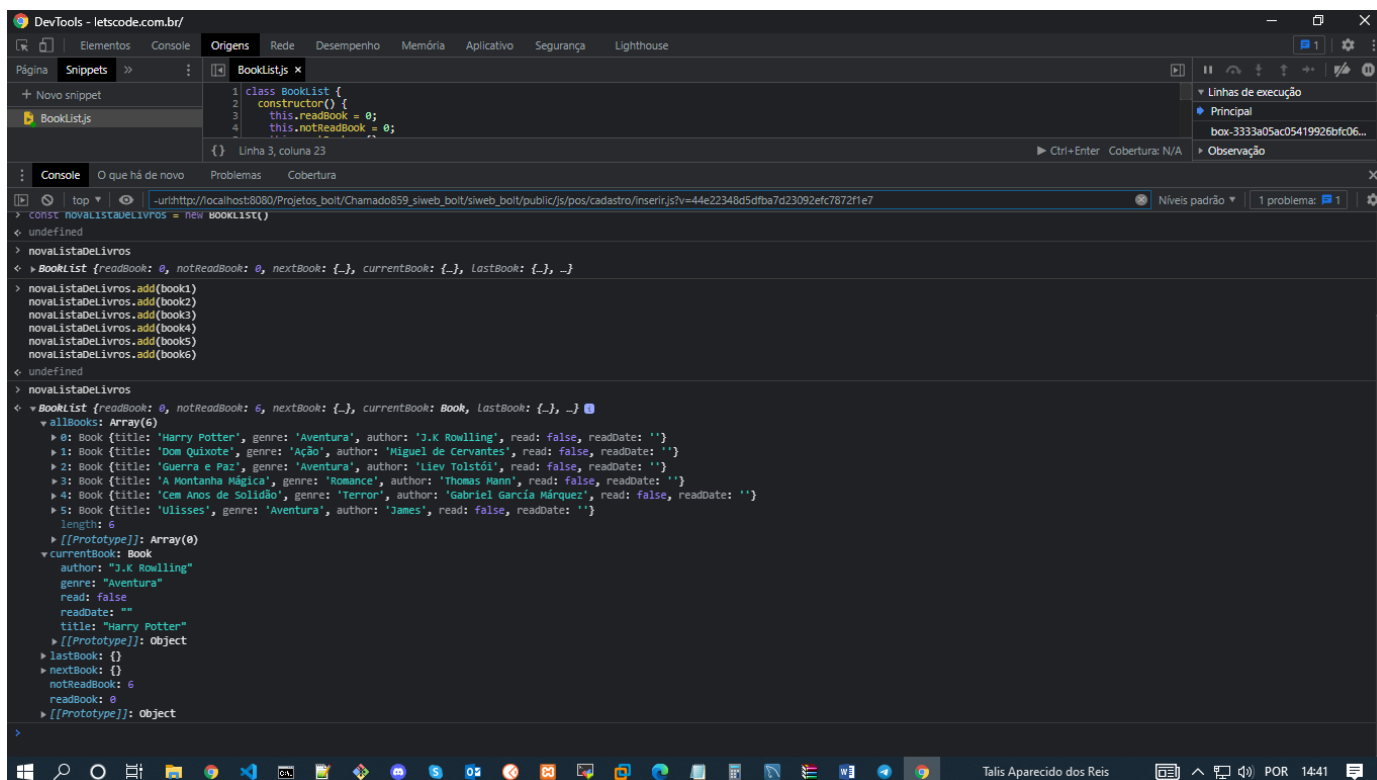


```

> const book1 = new Book('Harry Potter', 'Aventura', 'J.K Rowling', false);
> const book2 = new Book('Dom Quixote', 'Ação', 'Miguel de Cervantes', false);
> const book3 = new Book('Guerra e Paz', 'Aventura', 'Liev Tolstói', false);
> const book4 = new Book('A Montanha Mágica', 'Romance', 'Thomas Mann', false);
> const book5 = new Book('Cem Anos de Solidão', 'Terror', 'Gabriel Garcia Márquez', false);
> const book6 = new Book('Ulisses', 'Aventura', 'James', false);
>
> const novaListaDeLivros = new BookList()
>
> novaListaDeLivros
> BookList {readBook: 0, notReadBook: 0, nextBook: {}, currentBook: {}, lastBook: {}, ...}
  
```

Imagem 2: criando uma nova lista e novos livros.

Nas imagens 1 e 2 podemos visualizar o código completo, as instanciações das classes e a chamada da nova lista retornando o construtor com suas propriedades vazias.



```
class BookList {
  constructor() {
    this.readBook = 0;
    this.notReadBook = 0;
  }
}

const novalistaDelivros = new BookList()

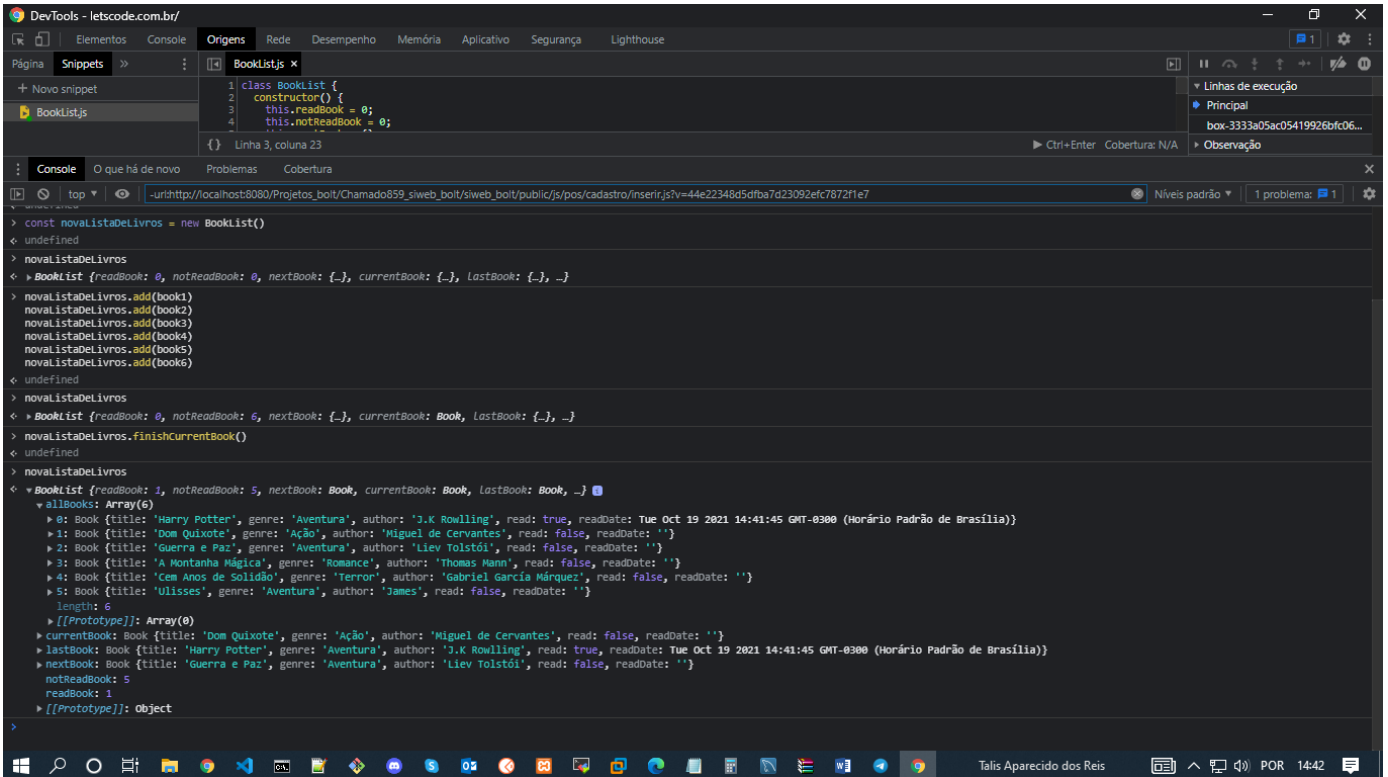
novalistaDelivros.add(book1)
novalistaDelivros.add(book2)
novalistaDelivros.add(book3)
novalistaDelivros.add(book4)
novalistaDelivros.add(book5)
novalistaDelivros.add(book6)

novalistaDelivros
// BookList {readBook: 0, notReadBook: 6, nextBook: [-], currentBook: Book, lastBook: [-], ...}
```

The console output shows the state of the `novalistaDelivros` instance after adding six books. The `allBooks` array contains six book objects, the `currentBook` is the first book ('Harry Potter'), and `notReadBook` is 6.

Imagem 3: adicionando os livros criados dentro da nova lista.

Na imagem 3 a chamada do método `add` passando os livros que foram instanciados na imagem 1, e a chamada da nova lista retornando com a propriedade `allBooks` preenchida com todos livros, a `currentBook` já inicializada com o primeiro livro e a `notReadBook` com a quantidade de todos livros não lidos.



```
class BookList {
  constructor() {
    this.readBook = 0;
    this.notReadBook = 0;
  }
}

const novaListaDeLivros = new BookList()

novaListaDeLivros.add(book1)
novaListaDeLivros.add(book2)
novaListaDeLivros.add(book3)
novaListaDeLivros.add(book4)
novaListaDeLivros.add(book5)
novaListaDeLivros.add(book6)

novaListaDeLivros

BookList {readBook: 0, notReadBook: 0, nextBook: [-], currentBook: [-], lastBook: [-], ...}

novaListaDeLivros.finishCurrentBook()

novaListaDeLivros

BookList {readBook: 1, notReadBook: 5, nextBook: Book, currentBook: Book, lastBook: Book, ...}
```

allBooks: Array(6)

- 0: Book {title: 'Harry Potter', genre: 'Aventura', author: 'J.K Rowling', read: true, readDate: Tue Oct 19 2021 14:41:45 GMT-0300 (Horário Padrão de Brasília)}
- 1: Book {title: 'Dom Quixote', genre: 'Ação', author: 'Miguel de Cervantes', read: false, readDate: ''}
- 2: Book {title: 'Guerra e Paz', genre: 'Aventura', author: 'Liev Tolstói', read: false, readDate: ''}
- 3: Book {title: 'A Montanha Mágica', genre: 'Romance', author: 'Thomas Mann', read: false, readDate: ''}
- 4: Book {title: 'Cem Anos de Solidão', genre: 'Terror', author: 'Gabriel Garcia Márquez', read: false, readDate: ''}
- 5: Book {title: 'Ulisses', genre: 'Aventura', author: 'James', read: false, readDate: ''}

length: 6

[[Prototype]]: Array(0)

currentBook: Book {title: 'Dom Quixote', genre: 'Ação', author: 'Miguel de Cervantes', read: false, readDate: ''}

lastBook: Book {title: 'Harry Potter', genre: 'Aventura', author: 'J.K Rowling', read: true, readDate: Tue Oct 19 2021 14:41:45 GMT-0300 (Horário Padrão de Brasília)}

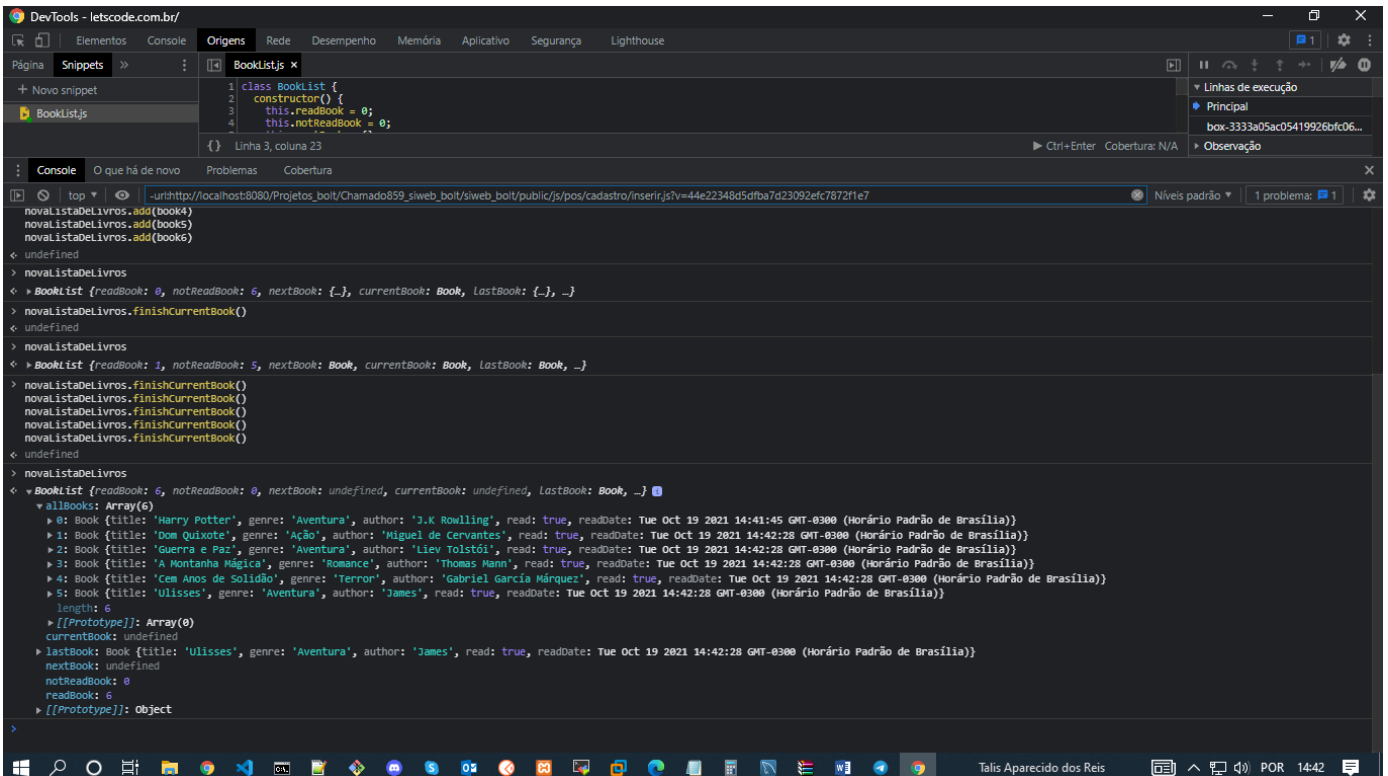
nextBook: Book {title: 'Guerra e Paz', genre: 'Aventura', author: 'Liev Tolstói', read: false, readDate: ''}

notReadBook: 5

readBook: 1

[[Prototype]]: Object

Imagem 4: executando o método finishCurrentBook para passar para o próximo livro.



```
novaListaDeLivros.add(book4)
novaListaDeLivros.add(book5)
novaListaDeLivros.add(book6)

novaListaDeLivros

BookList {readBook: 0, notReadBook: 6, nextBook: [-], currentBook: Book, lastBook: [-], ...}

novaListaDeLivros.finishCurrentBook()

novaListaDeLivros

BookList {readBook: 1, notReadBook: 5, nextBook: Book, currentBook: Book, lastBook: Book, ...}

novaListaDeLivros.finishCurrentBook()
novaListaDeLivros.finishCurrentBook()
novaListaDeLivros.finishCurrentBook()
novaListaDeLivros.finishCurrentBook()

novaListaDeLivros

BookList {readBook: 6, notReadBook: 0, nextBook: undefined, currentBook: undefined, lastBook: Book, ...}
```

allBooks: Array(6)

- 0: Book {title: 'Harry Potter', genre: 'Aventura', author: 'J.K Rowling', read: true, readDate: Tue Oct 19 2021 14:41:45 GMT-0300 (Horário Padrão de Brasília)}
- 1: Book {title: 'Dom Quixote', genre: 'Ação', author: 'Miguel de Cervantes', read: true, readDate: Tue Oct 19 2021 14:42:28 GMT-0300 (Horário Padrão de Brasília)}
- 2: Book {title: 'Guerra e Paz', genre: 'Aventura', author: 'Liev Tolstói', read: true, readDate: Tue Oct 19 2021 14:42:28 GMT-0300 (Horário Padrão de Brasília)}
- 3: Book {title: 'A Montanha Mágica', genre: 'Romance', author: 'Thomas Mann', read: true, readDate: Tue Oct 19 2021 14:42:28 GMT-0300 (Horário Padrão de Brasília)}
- 4: Book {title: 'Cem Anos de Solidão', genre: 'Terror', author: 'Gabriel Garcia Márquez', read: true, readDate: Tue Oct 19 2021 14:42:28 GMT-0300 (Horário Padrão de Brasília)}
- 5: Book {title: 'Ulisses', genre: 'Aventura', author: 'James', read: true, readDate: Tue Oct 19 2021 14:42:28 GMT-0300 (Horário Padrão de Brasília)}

length: 6

[[Prototype]]: Array(0)

currentBook: undefined

lastBook: Book {title: 'Ulisses', genre: 'Aventura', author: 'James', read: true, readDate: Tue Oct 19 2021 14:42:28 GMT-0300 (Horário Padrão de Brasília)}

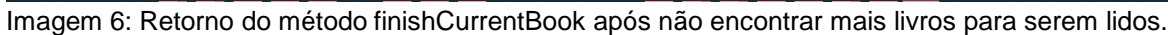
nextBook: undefined

notReadBook: 0

readBook: 6

[[Prototype]]: Object

Imagem 6: Executando o método finishCurrentBook até finalizar a lista de livros.



[Bolsas Santander \(becas-santander.com\)letscode.com.br/](http://BolsasSantander(becas-santander.com)letscode.com.br/)