

# Curso React/Redux

Leonardo Leitão

Versão 1.0, 22/05/2017

# Índice

1. Introdução .....	2
1.1. Visão Geral do Curso .....	2
1.2. Assine o Nosso Canal .....	2
1.3. Repositório do Curso .....	2
1.4. Configuração do Ambiente .....	2
2. Webpack .....	3
2.1. Visão Geral e Instalação .....	3
2.2. Configuração Inicial .....	3
2.3. Criando o index.html .....	4
2.4. Testando a Configuração .....	4
2.5. Usando o Sistema de Módulos do CommonsJS .....	5
2.6. Importância da referência .....	6
2.7. Usando EcmaScript 2015 sem Babel .....	7
2.8. Usando o Sistema de Módulos do ES2015 (Babel) .....	8
2.9. Interpretando o Operador Spread (Plugin Babel) .....	9
2.10. Adicionando o Preset do React .....	11
2.11. Adicionando o Loader para CSS .....	13
3. React .....	16
3.1. Configurando o Projeto .....	16
3.2. Configurando o Build com Webpack .....	16
3.3. Criando o arquivo index.html .....	17
3.4. Olá React .....	18
3.5. Primeiro Componente .....	19
3.6. Primeiro Componente (Arrow Function) .....	21
3.7. Usando as Propriedades do Componente .....	22
3.8. Exportando mais de um Componente .....	23
3.9. Componentes Filhos .....	24
3.10. Componentes Filhos .....	25
3.11. Passando Propriedades para Componentes Filhos (Parte 1) .....	26
3.12. Passando Propriedades para Componentes Filhos (Parte 2) .....	28
3.13. Componente de Classe .....	29
3.14. Componente com Estado (Ex. Contador) .....	30
3.15. Componentes Controlados/Não Controlados .....	31
4. TodoApp (Backend) .....	34
4.1. Visão Geral .....	34
4.2. Configuração e Instalação .....	34
4.3. Configurando o Servidor com o Express .....	35
4.4. Conexão com o Banco de Dados .....	36

4.5. ODM e Criação da API REST .....	37
4.6. Mapeamento das Rotas .....	37
4.7. Testando a API (Postman).....	39
4.8. Habilitando o CORS .....	39
4.9. Iniciando o Backend com PM2 .....	39
5. TodoApp (Frontend) .....	41
5.1. Configuração e Instalação .....	41
5.2. Configurando o Build com Webpack .....	42
5.3. Criação do index.html .....	44
5.4. Componente App .....	45
5.5. Componentes Todo e About .....	46
5.6. Componente Menu .....	48
5.7. Configurando as Rotas (React-Router).....	49
5.8. Componente PageHeader .....	50
5.9. Componentes TodoForm e TodoList .....	51
5.10. Estratégia de Implementação .....	53
5.11. Estrutura do Formulário .....	53
5.12. Componentes Grid e IconButton.....	53
5.13. Renderização Condicional (If).....	56
5.14. Evento Adicionar .....	57
5.15. Evento onChange .....	58
5.16. Evento Adicionar (Integração Backend) .....	60
5.17. Consulta e Exclusão de TODOS .....	61
5.18. Marcar como Concluído/Pendente.....	64
5.19. Pesquisa de TODOS.....	67
5.20. Melhorias de CSS e Limpar Formulário .....	69
5.21. Adicionar Teclas de Atalhos.....	74
6. React com Redux .....	76
6.1. Palavras Iniciais e Instalação.....	76
6.2. Conhecendo o Redux .....	76
6.3. Configurando o Redux.....	76
6.4. Ex 01 - Integrando o React-Redux (Parte 1) .....	77
6.5. Ex 01 - Integrando o React-Redux (Parte 2) .....	77
6.6. Ex 01 - Integrando o React-Redux (Parte 3) .....	78
6.7. Ex 02 - Integrando o React-Redux (Parte 1) .....	79
6.8. Ex 02 - Integrando o React-Redux (Parte 2) .....	81
7. TodoApp (Migração para Redux) .....	83
7.1. Instalando as Dependências .....	83
7.2. Configurando os Reducers .....	83
7.3. Configurando o Redux.....	84
7.4. Conectar o TodoList com Redux .....	85

7.5. Conectar o TodoForm com Redux .....	87
7.6. Action Creator changeDescription .....	88
7.7. Configurando o Redux Dev Tools .....	91
7.8. Action Creator de Pesquisa .....	91
7.9. Aplicando Middleware (Redux-Promise) .....	95
7.10. Adicionar TODO .....	96
7.11. Aplicando Middleware (Redux-Multi) .....	100
7.12. Aplicando Middleware (Redux-Thunk) .....	102
7.13. Marcar como Concluído/Pendente .....	104
7.14. Exclusão de TODOS .....	108
7.15. Limpar Formulário .....	112
7.16. Melhorando a Action de Pesquisa .....	117
8. Aplicação Final - Ciclos de Pagamentos (Backend) .....	123
8.1. Visão Geral .....	123
8.2. Configuração e Instalação .....	123
8.3. Configurando o Servidor com o Express .....	124
8.4. Conexão com o Banco de Dados .....	125
8.5. Mapeamento Objeto-Documento (ODM) .....	126
8.6. Criação da API REST .....	127
8.7. Mapeamento das Rotas .....	127
8.8. Testando a API (Postman) .....	129
8.9. Obtendo a Qtde de Registros (Count) .....	130
8.10. Obtendo o Sumário de Pagamentos .....	131
8.11. Tratando as Mensagens de Erro .....	132
8.12. Habilitando o CORS .....	134
8.13. Middleware Query Parser Int .....	135
9. Aplicação Final - Ciclos de Pagamentos (Frontend) .....	137
9.1. Visão Geral .....	137
9.2. Configuração e Instalação .....	137
9.3. Configurando o Build com Webpack .....	139
9.4. Criação do index.html .....	143
9.5. Componentes Index e App .....	143
9.6. Dependências do Template (AdminLTE) .....	144
9.7. Componente Cabeçalho .....	145
9.8. Componente Menu (Parte 1) .....	146
9.9. Componente Menu (Parte 2) .....	147
9.10. Componente Rodapé .....	149
9.11. Navegação com React-router .....	150
9.12. Componente de Título do Conteúdo .....	152
9.13. Componente Dashboard: Visão Geral .....	153
9.14. Componente Grid .....	153

9.15. Componente ValueBox .....	154
9.16. Usando ValueBox .....	155
9.17. Configurando Redux .....	156
9.18. Integrando Dashboard com Redux (Parte 1) .....	157
9.19. Integrando Dashboard com Redux (Parte 2) .....	159
9.20. Ciclo React-Redux com Middlewares .....	162
9.21. Configurando Redux DevTools .....	162
9.22. Dashboard SEM Redux .....	163
9.23. Visão Geral dos Componentes de Abas .....	165
9.24. Componente BillingCycle: Refatorando para Classe .....	165
9.25. Componentes Estruturais das Abas .....	165
9.26. Componente TabHeader .....	167
9.27. Criando a Primeira Action e o Reducer .....	170
9.28. Componente TabHeader: Conectando com Redux .....	171
9.29. Componente Tab Content .....	172
9.30. Componente BillingCycle: Conectando com Redux .....	176
9.31. Visibilidade das Abas (Parte 1) .....	178
9.32. Visibilidade das Abas (Parte 2) .....	181
9.33. Componente de Lista: Visão Geral .....	183
9.34. Criando a Action "getList" e o Reducer .....	183
9.35. Componente BillingCycleList .....	184
9.36. Integrando BillingCycleList com Redux .....	187
9.37. Exibindo os dados em BillingCycleList .....	188
9.38. BillingCycleForm com Redux-Form (Parte 01) .....	190
9.39. BillingCycleForm com Redux-Form (Parte 02) .....	192
9.40. Exibindo as Mensagens de Sucesso e Erro .....	196
9.41. Melhorias na Inclusão do Ciclo de Pagamento .....	198
9.42. Field com Componente Personalizado .....	200
9.43. Exibir Aba de Alterar Ciclo de Pagamento .....	202
9.44. Inicializar Formulário com Dados .....	207
9.45. Função do Botão Cancelar .....	209
9.46. Alterar Ciclo de Pagamento .....	212
9.47. Excluir Ciclo de Pagamento .....	216
9.48. Finalizando o Cadastro Básico (Melhorias) .....	225
9.49. Componente CreditList Básico .....	230
9.50. Componente Input para CreditList .....	233
9.51. Evoluindo o Componente CreditList .....	236
9.52. Adicionar e Clonar Créditos .....	244
9.53. Remover Créditos .....	246
9.54. Generalizando o Componente CreditList .....	248
9.55. Adicionando o Campo Status no ItemList .....	253

9.56. Componente Summary .....	259
9.57. Integrar Soma de Créditos e Débitos .....	262
10. Melhorias e Correções .....	265
10.1. Problema de Responsividade do Menu .....	265
10.2. Navegação com React-router (Versão 2) .....	266
11. My Money APP: Autenticação .....	269
11.1. Configurar Ambiente .....	269
11.2. Backend: Novas Dependências .....	269
11.3. Backend: Implementar Autenticação .....	269
11.4. Frontend: Implementar Autenticação .....	276
Appendix A: Tabela de Códigos .....	287
Glossário .....	288

## **Sumário**

*Apostila do curso de React/Redux da Cod3r.*

<https://www.cod3r.com.br>

# 1. Introdução

## 1.1. Visão Geral do Curso

## 1.2. Assine o Nosso Canal



Olá amigos, convido vocês a conhecer o canal da COD3R no Youtube.

Visite o canal: <https://www.youtube.com/aulasdeprogramacao>

## 1.3. Repositório do Curso



Os arquivos do curso estão disponíveis no GitHub.

Repositório do curso: <https://github.com/cod3rcursos/curso-react-redux>

## 1.4. Configuração do Ambiente



## 2. Webpack

### 2.1. Visão Geral e Instalação

Abra o terminal e dentro da pasta Desktop execute o seguinte comando:

#### Listagem 1 - Criar pasta *exercicios\_webpack*

*exercicios\_webpack/*

```
mkdir exercicios_webpack && cd exercicios_webpack
```

#### Listagem 2 - Criar arquivo *package.json*

*exercicios\_webpack/package.json*

```
npm init -y
```

#### Listagem 3 - Instalando as dependências

```
npm i --save-dev webpack@1.14.0 webpack-dev-server@1.16.2
```

#### Listagem 4 - Criar arquivo *.gitignore*

*exercicios\_webpack/.gitignore*

```
node_modules  
*.log
```

### 2.2. Configuração Inicial

### Listagem 5 - Criar arquivo webpack.config.js

exercicios\_webpack/webpack.config.js

```
const webpack = require('webpack')

module.exports = {
  entry: './ex/index.js',
  output: {
    path: __dirname + '/public',
    filename: './bundle.js'
  },
  devServer: {
    port: 8080,
    contentBase: './public'
  }
}
```

## 2.3. Criando o index.html



Criar uma pasta chamada **ex** que contém o arquivos **index.js** e também criar uma pasta chamada **public** que contém o arquivo **index.html**, nestas pastas poderão ser criadas, sobrescritas, alteradas e apagadas os arquivos a qualquer momento durante as aulas.

### Listagem 6 - Criar arquivo index.html

exercicios\_webpack/public/index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Exercícios de Webpack</title>
</head>
<body>
  <script src="bundle.js"></script>
</body>
</html>
```

## 2.4. Testando a Configuração

### Listagem 7 - Alterar arquivo index.js

exercicios\_webpack/ex/index.js (na linha 1)

```
console.log('Webpack')
```

### Listagem 8 - Alterar arquivo package.json

exercicios\_webpack/package.json (aprox. linha 7)

```
"test": "echo \"Error: no test specified\" && exit 1"
```

por:

```
"dev": "webpack-dev-server --progress --colors --inline --hot"
```



Quando o build terminar vai mostrar a mensagem: *webpack: Bundle is now VALID.*



Verificar o funcionamento: abra o navegador e digite <http://localhost:8080> e na ferramentas de desenvolvedor (Inspecionar) na opção console vai mostrar a palavra **Webpack**.

### Listagem 9 - Executar a aplicação

Abra o terminal e dentro da pasta exercicios\_webpack

```
npm run dev
```

## 2.5. Usando o Sistema de Módulos do CommonsJS

### Listagem 10 - Criar arquivo logger.js

exercicios\_webpack/ex/logger.js

```
function info(text) {  
  console.log(`INFO: ${text}`)  
}  
  
module.exports = { info }
```

### Listagem 11 - Sobrescrever arquivo index.js

exercicios\_webpack/ex/index.js

```
const logger = require('./logger')  
logger.info('Usando o padrão CommonJS!')
```

## 2.6. Importância da referência

### Listagem 12 - Criar arquivo duvidaCruel.js

exercicios\_webpack/ex/duvidaCruel.js

```
console.log('Sou carregado?')
```



Para abrir o terminal no Visual Studio Code use o atalho `ctrl + ``

### Listagem 13 - Executar Webpack no terminal

exercicios\_webpack\$

```
./node_modules/.bin/webpack
```

### Listagem 14 - Sobrescrever arquivo index.js

exercicios\_webpack/ex/index.js

```
require('./duvidacruel')
```

### Listagem 15 - Executar Webpack no terminal

exercicios\_webpack\$

```
./node_modules/.bin/webpack
```



Gerado o arquivo `exercicios_webpack/public/bundle.js` vai conter o conteúdo de todas as dependências a partir do ponto de entrada até a saída.

## 2.7. Usando EcmaScript 2015 sem Babel

### Listagem 16 - sobrescrever arquivo index.js

exercicios\_webpack/ex/index.js

```
class Pessoa {  
  constructor(nome) {  
    this.nome = nome  
  }  
  
  toString() {  
    return `Pessoa: ${this.nome}`  
  }  
}  
  
const pessoa = new Pessoa('Guilherme')  
console.log(pessoa.toString())
```



Para abrir o terminal no Visual studio Code use o atalho `ctrl + ``

### Listagem 17 - Executar Webpack no terminal

exercicios\_webpack\$

```
./node_modules/.bin/webpack
```



Agora no arquivo `bundle.js` que agora contém a classe *Pessoa*.

### Listagem 18 - Alterar arquivo index.js

exercicios\_webpack/ex/index.js

```
const pessoa = new Pessoa('Guilherme')  
console.log(pessoa.toString())
```



Mover a classe *Pessoa* para o novo arquivo `ex/pessoa.js`

### Listagem 19 - Alterar arquivo *peessoa.js*

*exercicios\_webpack/ex/peessoa.js*

```
export default class Pessoa {  
  constructor(nome) {  
    this.nome = nome  
  }  
  
  toString() {  
    return `Pessoa: ${this.nome}`  
  }  
}
```

### Listagem 20 - Alterar arquivo *index.js*

*exercicios\_webpack/ex/index.js* (na linha 1)

```
import Pessoa from './peessoa'  
  
const pessoa = new Pessoa('Guilherme')  
console.log(pessoa.toString())
```

## 2.8. Usando o Sistema de Módulos do ES2015 (Babel)

## Listagem 21 - Alterar arquivo webpack.config.js

exercicios\_webpack/webpack.config.js (aprox. linha 12)

```
const webpack = require('webpack')

module.exports = {
  entry: './ex/index.js',
  output: {
    path: __dirname + '/public',
    filename: './bundle.js'
  },
  devServer: {
    port: 8080,
    contentBase: './public'
  },
  module: {
    loaders: [{
      test: /\.js?$/,
      loader: 'babel-loader',
      exclude: /node_modules/,
      query: {
        presets: ['es2015']
      }
    }]
  }
}
```



Abra o terminal e dentro da pasta do projeto exercicios\_webpack execute o seguinte comando:

## Listagem 22 - Instalando as dependências

```
npm i --save-dev babel-core@6.22.1 babel-loader@6.2.10 babel-preset-es2015@6.22.0
```

## Listagem 23 - Executar a aplicação

```
npm run dev
```

## 2.9. Interpretando o Operador Spread (Plugin Babel)

## Listagem 24 - Sobrescrever arquivo index.js

exercicios\_webpack/ex/index.js

```
const produto = {
  nome: 'Caneta Bic Preta',
  preco: 1.90,
  desconto: 0.05
}

function clone(objeto) {
  return { ...objeto }
}

const novoProduto = clone(produto)
novoProduto.nome = 'Caneta Bic Azul'

console.log(produto, novoProduto)
```

## Listagem 25 - Alterar arquivo webpack.config.js

exercicios\_webpack/webpack.config.js (aprox. linha 19)

```
const webpack = require('webpack')

module.exports = {
  entry: './ex/index.js',
  output: {
    path: __dirname + '/public',
    filename: './bundle.js'
  },
  devServer: {
    port: 8080,
    contentBase: './public'
  },
  module: {
    loaders: [{
      test: /\.js?$/,
      loader: 'babel-loader',
      exclude: /node_modules/,
      query: {
        presets: ['es2015'],
        plugins: ['transform-object-rest-spread']
      }
    }]
  }
}
```





Abra o terminal e dentro da pasta do projeto `exercicios_webpack` execute o seguinte comando:

### Listagem 26 - Instalando as dependências

```
npm i --save-dev babel-plugin-transform-object-rest-spread@6.22.0
```

### Listagem 27 - Executar a aplicação

```
npm run dev
```

## 2.10. Adicionando o Preset do React

Abra o terminal e dentro da pasta do projeto `exercicios_webpack` execute o seguinte comando:

### Listagem 28 - Instalando as dependências

```
npm i --save-dev react@15.4.2
```

### Listagem 29 - Executar a aplicação

```
npm run dev
```

### Listagem 30 - Sobrescrever arquivo `index.js`

`exercicios_webpack/ex/index.js`

```
import 'react'

export default props => (
  <h1>Olá</h1>
)

console.log('Funcionou!')
```

## Listagem 31 - Alterar arquivo webpack.config.js

exercicios\_webpack/webpack.config.js (aprox. linha 19)

```
const webpack = require('webpack')

module.exports = {
  entry: './ex/index.js',
  output: {
    path: __dirname + '/public',
    filename: './bundle.js'
  },
  devServer: {
    port: 8080,
    contentBase: './public'
  },
  module: {
    loaders: [{
      test: /\.js?$/,
      loader: 'babel-loader',
      exclude: /node_modules/,
      query: {
        presets: ['es2015', 'react'],
        plugins: ['transform-object-rest-spread']
      }
    }]
  }
}
```



Abra o terminal e dentro da pasta do projeto exercicios\_webpack execute o seguinte comando:

## Listagem 32 - Instalando as dependências

```
npm i --save-dev babel-preset-react@6.22.0
```

## Listagem 33 - Executar a aplicação

```
npm run dev
```

## 2.11. Adicionando o Loader para CSS



Criar arquivo css em `ex/estilo.css`

### Listagem 34 - *Alterar arquivo estilo.css*

`exercicios_webpack/ex/estilo.css`

```
body{  
  background-color: #253B6E;  
}
```

### Listagem 35 - *Alterar arquivo index.js*

`exercicios_webpack/ex/index.js` (na linha 1)

```
import './estilo.css'  
import 'react'  
  
export default props => (  
  <h1>Olá</h1>  
)  
  
console.log('Funcionou!')
```

## Listagem 36 - Alterar arquivo webpack.config.js

exercicios\_webpack/webpack.config.js (aprox. linha 2, 14 e 26)

```
const webpack = require('webpack')
const ExtractTextPlugin = require('extract-text-webpack-plugin')

module.exports = {
  entry: './ex/index.js',
  output: {
    path: __dirname + '/public',
    filename: './bundle.js'
  },
  devServer: {
    port: 8080,
    contentBase: './public'
  },
  plugins: [
    new ExtractTextPlugin('app.css')
  ],
  module: {
    loaders: [{
      test: /\.js?$/,
      loader: 'babel-loader',
      exclude: /node_modules/,
      query: {
        presets: ['es2015', 'react'],
        plugins: ['transform-object-rest-spread']
      }
    },
    {
      test: /\.css$/,
      loader: ExtractTextPlugin.extract("style-loader", "css-loader")
    }
  ]
}
```

### Listagem 37 - Alterar arquivo index.html

exercicios\_webpack/public/index.html (aprox. linha 6)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Exercícios de Webpack</title>
    <link rel='stylesheet' href='app.css'>
  </head>
  <body>
    <script src="bundle.js"></script>
  </body>
</html>
```



Abra o terminal e dentro da pasta do projeto exercicios\_webpack execute o seguinte comando:

### Listagem 38 - Instalando as dependências

```
npm i --save-dev extract-text-webpack-plugin@1.0.1 style-loader@0.13.1 css-
loader@0.26.1
```

### Listagem 39 - Executar a aplicação

```
npm run dev
```



Para abrir o terminal no Visual studio Code use o atalho **ctrl + `**

### Listagem 40 - Executar Webpack no terminal

exercicios\_webpack\$

```
./node_modules/.bin/webpack
```



Agora foi gerado dois arquivos: **bundle.js** e **app.css**

## 3. React

### 3.1. Configurando o Projeto

Abra o terminal e dentro da pasta Desktop execute o seguinte comando:

#### Listagem 41 - Criar pasta *exercicios\_react*

*exercicios\_react/*

```
mkdir exercicios_react && cd exercicios_react
```

#### Listagem 42 - Criar arquivo *package.json*

*exercicios\_react/package.json*

```
npm init -y
```

#### Listagem 43 - Instalando as dependências

```
npm i --save-dev react@15.4.2 react-dom@15.4.2 babel-core@6.22.1 babel-loader@6.2.10 babel-preset-es2015@6.22.0 babel-preset-react@6.22.0 webpack@1.14.0 webpack-dev-server@1.16.2
```

#### Listagem 44 - Criar arquivo *.gitignore*

*exercicios\_react/.gitignore*

```
node_modules  
*.log
```

### 3.2. Configurando o Build com Webpack

## Listagem 45 - Criar arquivo webpack.config.js

exercicios\_react/webpack.config.js

```
const webpack = require('webpack')

module.exports = {
  entry: './ex/index.js',
  output: {
    path: __dirname + '/public',
    filename: './bundle.js'
  },
  devServer: {
    port: 8080,
    contentBase: './public'
  },
  module: {
    loaders: [{
      test: /\.jsx?$/,
      loader: 'babel-loader',
      exclude: /node_modules/,
      query: {
        presets: ['es2015', 'react']
      }
    }]
  }
}
```

### 3.3. Criando o arquivo index.html



Criar uma pasta chamada `public` em `exercicios_react/public`

## Listagem 46 - Criar arquivo index.html

exercicios\_react/public/index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Exercícios de React</title>
</head>
<body>
  <script src="bundle.js"></script>
</body>
</html>
```

## 3.4. Olá React

### Listagem 47 - Alterar arquivo webpack.config.js

exercicios\_react/webpack.config.js (aprox. linha 4)

```
const webpack = require('webpack')

module.exports = {
  entry: './ex/index.jsx',
  output: {
    path: __dirname + '/public',
    filename: './bundle.js'
  },
  devServer: {
    port: 8080,
    contentBase: './public'
  },
  module: {
    loaders: [{
      test: /\.jsx?$/,
      loader: 'babel-loader',
      exclude: /node_modules/,
      query: {
        presets: ['es2015', 'react']
      }
    }]
  }
}
```



Criar uma pasta chamada **ex** em **exercicios\_react/ex**

### Listagem 48 - Criar arquivo index.jsx

exercicios\_react/ex/index.jsx

```
import React from 'react'
import ReactDOM from 'react-dom'

ReactDOM.render(<h1>Olá React</h1>, document.getElementById('app'))
```



### Listagem 49 - Alterar arquivo package.json

exercicios\_react/package.json (aprox. linha 7)

```
"test": "echo \"Error: no test specified\" && exit 1"
```

por:

```
"dev": "webpack-dev-server --progress --colors --inline --hot"
```



Quando o build terminar vai mostrar a mensagem: *webpack: Bundle is now VALID.*



Verificar o funcionamento: abra o navegador e digite <http://localhost:8080> e na página vai mostrar o título **Olá React**.

### Listagem 50 - Executar a aplicação

Abra o terminal e dentro da pasta `exercicios_react`

```
npm run dev
```

## 3.5. Primeiro Componente

### Listagem 51 - Criar arquivo component.jsx

exercicios\_react/ex/component.jsx

```
import React from 'react'

function() {
  return <h1>Primeiro Componente!</h1>
}
```

### Listagem 52 - Alterar arquivo index.jsx

exercicios\_react/ex/index.jsx (aprox. linha 3 e 5)

```
import React from 'react'
import ReactDOM from 'react-dom'
import Component from './component.jsx'

ReactDOM.render(<Component />, document.getElementById('app'))
```

### Listagem 53 - Alterar arquivo component.jsx

exercicios\_react/ex/component.jsx (aprox. linha 3)

```
import React from 'react'

export default function() {
  return <h1>Primeiro Componente!</h1>
}
```

### Listagem 54 - Alterar arquivo index.jsx

exercicios\_react/ex/index.jsx (aprox. linha 3)

```
import React from 'react'
import ReactDOM from 'react-dom'
import Component from './component'

ReactDOM.render(<Component />, document.getElementById('app'))
```

## Listagem 55 - Alterar arquivo webpack.config.js

exercicios\_react/webpack.config.js (aprox. linha 13)

```
const webpack = require('webpack')

module.exports = {
  entry: './ex/index.js',
  output: {
    path: __dirname + '/public',
    filename: './bundle.js'
  },
  devServer: {
    port: 8080,
    contentBase: './public'
  },
  resolve: {
    extensions: ['', '.js', '.jsx']
  },
  module: {
    loaders: [{
      test: /\.jsx?$/,
      loader: 'babel-loader',
      exclude: /node_modules/,
      query: {
        presets: ['es2015', 'react']
      }
    }]
  }
}
```

## Listagem 56 - Executar a aplicação

Abra o terminal e dentro da pasta exercicios\_react

```
npm run dev
```

## 3.6. Primeiro Componente (Arrow Function)

### Listagem 57 - Alterar arquivo *component.jsx*

*exercicios\_react/ex/component.jsx* (aprox. linha 3)

```
import React from 'react'

export default () => (
  <h1>Primeiro Componente!</h1>
)
```

## 3.7. Usando as Propriedades do Componente

### Listagem 58 - Alterar arquivo *component.jsx*

*exercicios\_react/ex/component.jsx* (aprox. linha 3)

```
import React from 'react'

export default (props) => (
  <h1>{props.value}</h1>
)
```

### Listagem 59 - Alterar arquivo *index.jsx*

*exercicios\_react/ex/index.jsx* (aprox. linha 5)

```
import React from 'react'
import ReactDOM from 'react-dom'
import Component from './component'

ReactDOM.render(<Component value='Show!' />, document.getElementById('app'))
```

### Listagem 60 - Alterar arquivo *component.jsx*

*exercicios\_react/ex/component.jsx* (aprox. linha 3)

```
import React from 'react'

export default props => (
  <h1>{props.value}</h1>
)
```

## 3.8. Exportando mais de um Componente

### Listagem 61 - Alterar arquivo *component.jsx*

*exercicios\_react/ex/component.jsx* (aprox. linha 3)

```
import React from 'react'

const Primeiro = props => (
  <h1>Primeiro Componente!</h1>
)

const Segundo = props => <h1>Segundo Componente!</h1>

export { Primeiro, Segundo }
```

### Listagem 62 - Alterar arquivo *index.jsx*

*exercicios\_react/ex/index.jsx* (aprox. linha 3 e 5)

```
import React from 'react'
import ReactDOM from 'react-dom'
import { Primeiro, Segundo } from './component'

ReactDOM.render(
  <div>
    <Primeiro />
    <Segundo />
  </div>
, document.getElementById('app'))
```

### Listagem 63 - Alterar arquivo *component.jsx*

*exercicios\_react/ex/component.jsx* (aprox. linha 3, 7 e 9)

```
import React from 'react'

export const Primeiro = props => (
  <h1>Primeiro Componente!</h1>
)

export const Segundo = props => <h1>Segundo Componente!</h1>

//export { Primeiro, Segundo }
```

### Listagem 64 - Alterar arquivo component.jsx

exercicios\_react/ex/component.jsx (aprox. linha 3)

```
import React from 'react'

export default props => (
  <h1>Primeiro Componente!</h1>
)

export const Segundo = props => <h1>Segundo Componente!</h1>

//export { Primeiro, Segundo }
```

### Listagem 65 - Alterar arquivo index.jsx

exercicios\_react/ex/index.jsx (aprox. linha 3)

```
import React from 'react'
import ReactDOM from 'react-dom'
import Primeiro, { Segundo } from './component'

ReactDOM.render(
  <div>
    <Primeiro />
    <Segundo />
  </div>
, document.getElementById('app'))
```

## 3.9. Componentes Filhos

### Listagem 66 - Criar arquivo member.jsx

exercicios\_react/ex/member.jsx

```
import React from 'react'

export default props => (
  <div>{props.name} <strong>{props.lastName}</strong></div>
)
```

### Listagem 67 - Criar arquivo *silvaFamily.jsx*

*exercicios\_react/ex/silvaFamily.jsx*

```
import React from 'react'
import Member from './member'

export default props => (
  <div>
    <Member name='João' lastName='Silva' />
    <Member name='Maria' lastName='Silva' />
    <Member name='Jose' lastName='Silva' />
    <Member name='Pedro' lastName='Silva' />
  </div>
)
```

### Listagem 68 - Alterar arquivo *index.jsx*

*exercicios\_react/ex/index.jsx* (aprox. linha 3 e 6)

```
import React from 'react'
import ReactDOM from 'react-dom'
import SilvaFamily from './silvaFamily'

ReactDOM.render(
  <SilvaFamily />
, document.getElementById('app'))
```

## 3.10. Componentes Filhos

### Listagem 69 - Criar arquivo *family.jsx*

*exercicios\_react/ex/family.jsx*

```
import React from 'react'

export default props => (
  <div>
    <h1>Família</h1>
    {props.children}
  </div>
)
```

### Listagem 70 - Alterar arquivo index.jsx

exercicios\_react/ex/index.jsx (aprox. linha 3 e 7)

```
import React from 'react'
import ReactDOM from 'react-dom'
import Family from './family'
import Member from './member'

ReactDOM.render(
  <Family lastName='Silva'>
    <Member name='Guilherme' />
  </Family>
, document.getElementById('app'))
```

## 3.11. Passando Propriedades para Componentes Filhos (Parte 1)

### Listagem 71 - Alterar arquivo family.jsx

exercicios\_react/ex/family.jsx (aprox. linha 6)

```
import React from 'react'

export default props => (
  <div>
    <h1>Família</h1>
    { React.cloneElement(props.children, {...props}) }
  </div>
)
```



No terminal pare a aplicação apertando **ctrl + c** e execute o seguinte comando:

### Listagem 72 - Instalando as dependências

```
npm i --save-dev babel-plugin-transform-object-rest-spread@6.22.0
```



## Listagem 73 - Alterar arquivo webpack.config.js

exercicios\_react/webpack.config.js (aprox. linha 22)

```
const webpack = require('webpack')

module.exports = {
  entry: './ex/index.js',
  output: {
    path: __dirname + '/public',
    filename: './bundle.js'
  },
  devServer: {
    port: 8080,
    contentBase: './public'
  },
  module: {
    loaders: [{
      test: /\.jsx?$/,
      loader: 'babel-loader',
      exclude: /node_modules/,
      query: {
        presets: ['es2015', 'react'],
        plugins: ['transform-object-rest-spread']
      }
    }]
  }
}
```

## Listagem 74 - Executar a aplicação

No terminal execute o seguinte comando:

```
npm run dev
```

### Listagem 75 - Alterar arquivo index.jsx

exercicios\_react/ex/index.jsx (aprox. linha 9)

```
import React from 'react'
import ReactDOM from 'react-dom'
import Family from './family'
import Member from './member'

ReactDOM.render(
  <Family lastName='Silva'>
    <Member name='Guilherme' />
    <Member name='Rafael' />
    <Member name='Julia' />
  </Family>
, document.getElementById('app'))
```

## 3.12. Passando Propriedades para Componentes Filhos (Parte 2)

### Listagem 76 - Alterar arquivo family.jsx

exercicios\_react/ex/family.jsx (aprox. linha 6)

```
import React from 'react'

export default props => (
  <div>
    <h1>Família</h1>
    { React.Children.map(props.children,
      child => React.cloneElement(child, {...props})) }
  </div>
)
```



Criar uma pasta chamada **utils** em **exercicios\_react/utils**

### Listagem 77 - Criar arquivo *reactUtils.js*

*exercicios\_react/Utils/reactUtils.js*

```
import React from 'react'

function childrenWithProps(children, props){
  return React.Children.map(props.children,
    child => React.cloneElement(child, {...props}))
}

export { childrenWithProps }
```

### Listagem 78 - Alterar arquivo *family.jsx*

*exercicios\_react/ex/family.jsx* (aprox. linha 2 e 7)

```
import React from 'react'
import { childrenWithProps } from '../utils/reactUtils'
export default props => (
  <div>
    <h1>Família</h1>
    { childrenWithProps(props.children, props) }
  </div>
)
```

## 3.13. Componente de Classe

### Listagem 79 - Criar arquivo *classComponent.jsx*

*exercicios\_react/ex/classComponent.jsx*

```
import React, { Component } from 'react'

export default class ClassComponent extends Component{
  render() {
    return(
      <h1>{this.props.value}</h1>
    )
  }
}
```

### Listagem 80 - Alterar arquivo index.jsx

exercicios\_react/ex/index.jsx (aprox. linha 3 e 6)

```
import React from 'react'
import ReactDOM from 'react-dom'
import ClassComponent from './classComponent'

ReactDOM.render(
  <ClassComponent value='Componente Classe' />
, document.getElementById('app'))
```

## 3.14. Componente com Estado (Ex. Contador)

### Listagem 81 - Alterar arquivo index.jsx

exercicios\_react/ex/index.jsx (aprox. linha 6)

```
import React from 'react'
import ReactDOM from 'react-dom'
import ClassComponent from './classComponent'

ReactDOM.render(
  <ClassComponent value={10} />
, document.getElementById('app'))
```

## Listagem 82 - Alterar arquivo classComponent.jsx

exercicios\_react/ex/classComponent.jsx (aprox. linha 4)

```
import React, { Component } from 'react'

export default class ClassComponent extends Component{
  constructor(props){
    super(props)
    this.state = { value: props.initialValue }
  }

  sum(delta){
    this.setState({ value: this.state.value + delta })
  }

  render() {
    return(
      <div>
        <h1>{this.props.label}</h1>
        <h2>{this.state.value}</h2>
        <button onClick={() => this.sum(-1)}>Dec</button>
        <button onClick={() => this.sum(1)}>Inc</button>
      </div>
    )
  }
}
```

## Listagem 83 - Alterar arquivo index.jsx

exercicios\_react/ex/index.jsx (aprox. linha 6)

```
import React from 'react'
import ReactDOM from 'react-dom'
import ClassComponent from './classComponent'

ReactDOM.render(
  <ClassComponent label='Contador' initialValue={10} />
, document.getElementById('app'))
```

## 3.15. Componentes Controlados/Não Controlados

## Listagem 84 - Criar arquivo field.jsx

exercicios\_react/ex/field.jsx

```
import React, { Component } from 'react'

class Field extends Component{
  constructor(props){
    super(props)
    this.state = { value: props.initialValue }
  }

  handleChange(event) {
    this.setState({ value: event.target.value })
  }

  render(){
    return (
      <div>
        <label>{this.state.value}</label><br />
        <input onChange={this.handleChange} value={this.state.value} />
      </div>
    )
  }
}

export default Field
```

## Listagem 85 - Alterar arquivo index.jsx

exercicios\_react/ex/index.jsx (aprox. linha 3 e 6)

```
import React from 'react'
import ReactDOM from 'react-dom'
import Field from './field'

ReactDOM.render(
  <Field initialValue='Teste' />
, document.getElementById('app'))
```

## Listagem 86 - Alterar arquivo field.jsx

exercicios\_react/ex/field.jsx (aprox. linha 8)

```
import React, { Component } from 'react'

class Field extends Component{
  constructor(props){
    super(props)
    this.state = { value: props.initialValue }
    this.handleChange = this.handleChange.bind(this)
  }

  handleChange(event) {
    this.setState({ value: event.target.value })
  }

  render(){
    return (
      <div>
        <label>{this.state.value}</label><br />
        <input onChange={this.handleChange} value={this.state.value} />
      </div>
    )
  }
}

export default Field
```

## 4. TodoApp (Backend)

### 4.1. Visão Geral

### 4.2. Configuração e Instalação

Abra o terminal e dentro da pasta **Desktop** execute o seguinte comando:

#### Listagem 87 - Criar pasta todo-app

todo-app/

```
mkdir todo-app && cd todo-app
```

#### Listagem 88 - Criar pasta backend

todo-app/backend

```
mkdir backend && cd backend
```

#### Listagem 89 - Criar arquivo package.json

todo-app/backend/package.json

```
npm init -y
```

#### Listagem 90 - Instalando as dependências

```
npm i --save-dev body-parser@1.15.2 express@4.14.0 mongoose@4.7.0 node-restful@0.2.5 pm2@2.1.5 nodemon@1.11.0
```

#### Listagem 91 - Abrir o Visual Studio Code

```
code .
```



### Listagem 92 - Alterar arquivo package.json

todo-app/backend/package.json (aprox. linha 5 e 7)

```
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "src/loader.js",
  "scripts": {
    "dev": "nodemon",
    "production": "pm2 start src/loader.js --name todo-app"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "body-parser": "^1.15.2",
    "express": "^4.14.0",
    "mongoose": "^4.7.0",
    "node-restful": "^0.2.5",
    "nodemon": "^1.11.0",
    "pm2": "^2.1.5"
  }
}
```

### Listagem 93 - Criar arquivo .gitignore

todo-app/backend/.gitignore

```
node_modules
*.log
```

## 4.3. Configurando o Servidor com o Express



Criar uma pasta chamada **src** em **backend/src**

### Listagem 94 - Criar arquivo loader.js

todo-app/backend/src/loader.js

```
require('./config/server')
```



Criar uma pasta chamada `config` em `src/config`

### Listagem 95 - Criar arquivo `server.js`

`todo-app/backend/src/config/server.js`

```
const port = 3003

const bodyParser = require('body-parser')
const express = require('express')
const server = express()

server.use(bodyParser.urlencoded({ extended: true }))
server.use(bodyParser.json())

server.listen(port, function() {
  console.log(`BACKEND is running on port ${port}.`)
})
```

### Listagem 96 - Executar o backend da aplicação

Abra o terminal e dentro da pasta `todo-app/backend`

```
npm run dev
```

## 4.4. Conexão com o Banco de Dados

### Listagem 97 - Criar arquivo `database.js`

`todo-app/backend/src/config/database.js`

```
const mongoose = require('mongoose')
mongoose.Promise = global.Promise
module.exports = mongoose.connect('mongodb://localhost/todo')
```

### Listagem 98 - Alterar arquivo `loader.js`

`todo-app/backend/src/loader.js` (aprox. linha 2)

```
require('./config/server')
require('./config/database')
```

## 4.5. ODM e Criação da API REST



Criar uma pasta chamada `api` e dentro dela criar uma pasta chamada `todo` em `src/api/todo`

### Listagem 99 - Criar arquivo `todo.js`

`todo-app/src/api/todo/todo.js`

```
const restful = require('node-restful')
const mongoose = restful.mongoose

const todoSchema = new mongoose.Schema({
  description: { type: String, required: true },
  done: { type: Boolean, required: true, default: false },
  createdAt: { type: Date, default: Date.now }
})

module.exports = restful.model('Todo', todoSchema)
```

### Listagem 100 - Criar arquivo `todoService.js`

`todo-app/src/api/todo/todoService.js`

```
const Todo = require('./todo')

Todo.methods(['get', 'post', 'put', 'delete'])
Todo.updateOptions({new: true, runValidators: true})

module.exports = Todo
```

## 4.6. Mapeamento das Rotas

## Listagem 101 - Criar arquivo routes.js

todo-app/backend/src/config/routes.js

```
const express = require('express')

module.exports = function(server) {

  // API Routes
  const router = express.Router()
  server.use('/api', router)

  // TODO Routes
  const todoService = require('../api/todo/todoService')
  todoService.register(router, '/todos')
}
```

## Listagem 102 - Alterar arquivo loader.js

todo-app/backend/src/loader.js (aprox. linha 1 e 3)

```
const server = require('./config/server')
require('./config/database')
require('./config/routes')(server)
```

## Listagem 103 - Alterar arquivo server.js

todo-app/backend/src/config/server.js (aprox. linha 14)

```
const port = 3003

const bodyParser = require('body-parser')
const express = require('express')
const server = express()

server.use(bodyParser.urlencoded({ extended: true }))
server.use(bodyParser.json())

server.listen(port, function() {
  console.log('BACKEND is running on port ${port}.')
})

module.exports = server
```

## 4.7. Testando a API (Postman)



Acesse o site do Postman: <https://www.getpostman.com> para instalar no computador ou usar a extensão **Chrome** para testar a Api.

## 4.8. Habilitando o CORS

### Listagem 104 - Criar arquivo cors.js

todo-app/backend/src/config/cors.js

```
module.exports = function(req, res, next) {  
  res.header('Access-Control-Allow-Origin', '*')  
  res.header('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT, PATCH, DELETE')  
  res.header('Access-Control-Allow-Headers', 'Origin, X-Requested-With, Content-Type, Accept')  
  next()  
}
```

### Listagem 105 - Alterar arquivo server.js

todo-app/backend/src/config/server.js (aprox. linha 6 e 10)

```
const port = 3003  
  
const bodyParser = require('body-parser')  
const express = require('express')  
const server = express()  
const allowCors = require('./cors')  
  
server.use(bodyParser.urlencoded({ extended: true }))  
server.use(bodyParser.json())  
server.use(allowCors)  
  
server.listen(port, function() {  
  console.log(`BACKEND is running on port ${port}.`)  
})  
  
module.exports = server
```

## 4.9. Iniciando o Backend com PM2

Abra o terminal e dentro da pasta **backend** execute o seguinte comando:

### **Listagem 106 - Executar a aplicação com PM2**

todo-app/backend

```
npm run production
```

### **Listagem 107 - Executar o monit do PM2**

todo-app/backend

```
./node_modules/.bin/pm2 monit
```

## 5. TodoApp (Frontend)

### 5.1. Configuração e Instalação

Abra o terminal e dentro da pasta `todo-app` execute o seguinte comando:

#### Listagem 108 - Criar pasta frontend

`todo-app/frontend`

```
mkdir frontend && cd frontend
```

#### Listagem 109 - Criar arquivo package.json

`todo-app/frontend/package.json`

```
npm init -y
```

#### Listagem 110 - Instalando as dependências do Webpack

```
npm i --save-dev webpack@1.14.0 webpack-dev-server@1.16.2
```

#### Listagem 111 - Instalando as dependências do Babel

```
npm i --save-dev babel-core@6.22.1 babel-loader@6.2.10 babel-plugin-react-html-attrs@2.0.0 babel-plugin-transform-object-rest-spread@6.22.0 babel-preset-es2015@6.22.0 babel-preset-react@6.22.0
```

#### Listagem 112 - Instalando as dependências do CSS e file loader

```
npm i --save-dev extract-text-webpack-plugin@1.0.1 css-loader@0.26.1 style-loader@0.13.1 file-loader@0.9.0
```

### **Listagem 113** - *Instalando as dependências do Bootstrap e Font Awesome*

```
npm i --save-dev bootstrap@3.3.7 font-awesome@4.7.0
```

### **Listagem 114** - *Instalando as dependências do React e Axios*

```
npm install --save-dev react@15.4.2 react-dom@15.4.2 react-router@3.0.2  
axios@0.15.3
```

### **Listagem 115** - *Abrir o Visual Studio Code*

```
code .
```

### **Listagem 116** - *Criar arquivo .gitignore*

todo-app/frontend/.gitignore

```
node_modules  
*.log
```

## **5.2. Configurando o Build com Webpack**



## Listagem 117 - Criar arquivo webpack.config.js

todo-app/frontend/webpack.config.js

```
const webpack = require('webpack')
const ExtractTextPlugin = require('extract-text-webpack-plugin')

module.exports = {
  entry: './src/index.jsx',
  output: {
    path: __dirname + '/public',
    filename: './app.js'
  },
  devServer: {
    port: 8080,
    contentBase: './public',
  },
  resolve: {
    extensions: ['', '.js', '.jsx'],
    alias: {
      modules: __dirname + '/node_modules'
    }
  },
  plugins: [
    new ExtractTextPlugin('app.css')
  ],
  module: {
    loaders: [{
      test: /\.js[x]?$/,
      loader: 'babel-loader',
      exclude: /node_modules/,
      query: {
        presets: ['es2015', 'react'],
        plugins: ['transform-object-rest-spread']
      }
    }, {
      test: /\.css$/,
      loader: ExtractTextPlugin.extract('style-loader', 'css-loader')
    }, {
      test: /\.woff|\.woff2|.ttf|.eot|.svg*.*$/,
      loader: 'file'
    }
  ]
}
```

## Listagem 118 - Alterar arquivo `package.json`

`todo-app/frontend/package.json` (aprox. linha 7 e 8)

```
{
  "name": "frontend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "dev": "webpack-dev-server --progress --colors --inline --hot",
    "production": "webpack --progress -p"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "axios": "^0.15.3",
    "babel-core": "^6.22.1",
    "babel-loader": "^6.2.10",
    "babel-plugin-react-html-attrs": "^2.0.0",
    "babel-plugin-transform-object-rest-spread": "^6.22.0",
    "babel-preset-es2015": "^6.22.0",
    "babel-preset-react": "^6.22.0",
    "bootstrap": "^3.3.7",
    "css-loader": "^0.26.1",
    "extract-text-webpack-plugin": "^1.0.1",
    "file-loader": "^0.9.0",
    "font-awesome": "^4.7.0",
    "react": "^15.4.2",
    "react-dom": "^15.4.2",
    "react-redux": "^5.0.2",
    "react-router": "^3.0.2",
    "redux": "^3.6.0",
    "redux-multi": "^0.1.12",
    "redux-promise": "^0.5.3",
    "redux-thunk": "^2.2.0",
    "style-loader": "^0.13.1",
    "webpack": "^1.14.0",
    "webpack-dev-server": "^1.16.2"
  }
}
```

## 5.3. Criação do `index.html`



Criar uma pasta chamada `public` em `frontend/public`

### Listagem 119 - Criar arquivo *index.html*

todo-app/frontend/public/index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset='utf-8'>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <title>Todo App</title>
    <link rel='stylesheet' href='app.css'>
  </head>
  <body>
    <div id="app" class='container'></div>
    <script src='app.js'></script>
  </body>
</html>
```

## 5.4. Componente App



Criar uma pasta chamada **src** e dentro dela criar uma pasta chamada **main** em **frontend/src/main**

### Listagem 120 - Criar arquivo *app.jsx*

todo-app/frontend/src/main/app.jsx

```
import 'modules/bootstrap/dist/css/bootstrap.min.css'
import 'modules/font-awesome/css/font-awesome.min.css'

import React from 'react'

export default props => (
  <div className='container'>
    <h1>Teste</h1>
  </div>
)
```

### Listagem 121 - Criar arquivo index.jsx

todo-app/frontend/src/index.jsx

```
import React from 'react'
import ReactDOM from 'react-dom'
import App from './main/app'

ReactDOM.render(<App />, document.getElementById('app'))
```

### Listagem 122 - Executar a aplicação

Abra o terminal e dentro da pasta todo-app/frontend

```
npm run dev
```



Verificar o funcionamento: abra o navegador e digite <http://localhost:8080> e na página vai mostrar o título **Teste**.

## 5.5. Componentes Todo e About



Criar uma pasta chamada **todo** em **src/todo**

### Listagem 123 - Criar arquivo todo.jsx

todo-app/frontend/src/todo/todo.jsx

```
import React, { Component } from 'react'

export default class Todo extends Component {
  render() {
    return (
      <div>
        <h1>Todo</h1>
      </div>
    )
  }
}
```

## Listagem 124 - Alterar arquivo app.jsx

todo-app/frontend/src/main/app.jsx (aprox. linha 5 e 9)

```
import 'modules/bootstrap/dist/css/bootstrap.min.css'
import 'modules/font-awesome/css/font-awesome.min.css'

import React from 'react'
import Todo from '../todo/todo'

export default props => (
  <div className='container'>
    <Todo />
  </div>
)
```



Criar uma pasta chamada **about** em **src/about**

## Listagem 125 - Criar arquivo about.jsx

todo-app/frontend/src/about/about.jsx

```
import React from 'react'

export default props => (
  <div>
    <h1>Sobre</h1>
  </div>
)
```

## Listagem 126 - Alterar arquivo app.jsx

todo-app/frontend/src/main/app.jsx (aprox. linha 6 e 11)

```
import 'modules/bootstrap/dist/css/bootstrap.min.css'
import 'modules/font-awesome/css/font-awesome.min.css'

import React from 'react'
import Todo from '../todo/todo'
import About from '../about/about'

export default props => (
  <div className='container'>
    <Todo />
    <About />
  </div>
)
```

## 5.6. Componente Menu



Criar uma pasta chamada `template` em `src/template`

## Listagem 127 - Criar arquivo menu.jsx

todo-app/frontend/src/template/menu.jsx

```
import React from 'react'

export default props => (
  <nav className='navbar navbar-inverse bg-inverse'>
    <div className='container'>
      <div className='navbar-header'>
        <a className='navbar-brand' href='#'>
          <i className='fa fa-calendar-check-o'></i> TodoApp
        </a>
      </div>

      <div id='navbar' className='navbar-collapse collapse'>
        <ul className='nav navbar-nav'>
          <li><a href='#/todos'>Tarefas</a></li>
          <li><a href='#/about'>Sobre</a></li>
        </ul>
      </div>
    </div>
  </nav>
)
```

### Listagem 128 - Alterar arquivo app.jsx

todo-app/frontend/src/main/app.jsx (aprox. linha 7 e 11)

```
import 'modules/bootstrap/dist/css/bootstrap.min.css'
import 'modules/font-awesome/css/font-awesome.min.css'

import React from 'react'
import Todo from '../todo/todo'
import About from '../about/about'
import Menu from '../template/menu'

export default props => (
  <div className='container'>
    <Menu />
    <Todo />
    <About />
  </div>
)
```

## 5.7. Configurando as Rotas (React-Router)

### Listagem 129 - Criar arquivo routes.jsx

todo-app/frontend/src/main/routes.jsx

```
import React from 'react'
import { Router, Route, Redirect, hashHistory } from 'react-router'

import Todo from '../todo/todo'
import About from '../about/about'

export default props => (
  <Router history={hashHistory}>
    <Route path='/todos' component={Todo} />
    <Route path='/about' component={About} />
    <Redirect from='*' to='/todos' />
  </Router>
)
```



Remover os imports de Todo e About e seus componentes.

### Listagem 130 - Alterar arquivo app.jsx

todo-app/frontend/src/main/app.jsx (aprox. linha 6 e 11)

```
import 'modules/bootstrap/dist/css/bootstrap.min.css'
import 'modules/font-awesome/css/font-awesome.min.css'

import React from 'react'
import Menu from '../template/menu'
import Routes from './routes'

export default props => (
  <div className='container'>
    <Menu />
    <Routes />
  </div>
)
```

## 5.8. Componente PageHeader

### Listagem 131 - Criar arquivo pageHeader.jsx

todo-app/frontend/src/template/pageHeader.jsx

```
import React from 'react'

export default props => (
  <header className='page-header'>
    <h2>{props.name} <small>{props.small}</small></h2>
  </header>
)
```



### Listagem 132 - Alterar arquivo *todo.jsx*

*todo-app/frontend/src/todo/todo.jsx* (aprox. linha 2 e 8)

```
import React, { Component } from 'react'
import PageHeader from '../template/pageHeader'

export default class Todo extends Component {
  render() {
    return (
      <div>
        <PageHeader name='Tarefas' small='Cadastro'></PageHeader>
      </div>
    )
  }
}
```

### Listagem 133 - Alterar arquivo *about.jsx*

*todo-app/frontend/src/about/about.jsx* (aprox. linha 2 e 6)

```
import React from 'react'
import PageHeader from '../template/pageHeader'

export default props => (
  <div>
    <PageHeader name='Sobre' small='Nós'></PageHeader>

    <h2>Nossa História</h2>
    <p>Lorem ipsum dolor sit amet...</p>
    <h2>Missão e Visão</h2>
    <p>Lorem ipsum dolor sit amet...</p>
    <h2>Imprensa</h2>
    <p>Lorem ipsum dolor sit amet...</p>
  </div>
)
```

## 5.9. Componentes **TodoForm** e **TodoList**

### Listagem 134 - Criar arquivo *todoForm.jsx*

todo-app/frontend/src/todo/todoForm.jsx

```
import React from 'react'

export default props => (
  <div>
    <h1>Form</h1>
  </div>
)
```

### Listagem 135 - Alterar arquivo *todo.jsx*

todo-app/frontend/src/todo/todo.jsx (aprox. linha 3, 4, 11 e 12)

```
import React, { Component } from 'react'
import PageHeader from '../template/pageHeader'
import TodoForm from './todoForm'
import TodoList from './todoList'

export default class Todo extends Component {
  render() {
    return (
      <div>
        <PageHeader name='Tarefas' small='Cadastro'></PageHeader>
        <TodoForm />
        <TodoList />
      </div>
    )
  }
}
```

### Listagem 136 - Criar arquivo *todoList.jsx*

todo-app/frontend/src/todo/todoList.jsx

```
import React from 'react'

export default props => (
  <div>
    <h1>List</h1>
  </div>
)
```

## 5.10. Estratégia de Implementação

## 5.11. Estrutura do Formulário

### Listagem 137 - Sobrescrever arquivo *todoForm.jsx*

todo-app/frontend/src/todo/todoForm.jsx

```
import React from 'react'

export default props => (
  <div role='form' className='todoForm'>
    <div className='col-xs-12 col-sm-9 col-md-10'>
      <input id='description' className='form-control'
        placeholder='Adicione uma tarefa'></input>
    </div>

    <div className='col-xs-12 col-sm-3 col-md-2'>
      <button className='btn btn-primary'>
        <i className='fa fa-plus'></i>
      </button>
    </div>
  </div>
)
```

## 5.12. Componentes Grid e IconButton

## Listagem 138 - Criar arquivo grid.jsx

todo-app/frontend/src/template/grid.jsx

```
import React, { Component } from 'react'

export default class Grid extends Component {
  toCssClasses(numbers) {
    const cols = numbers ? numbers.split(' ') : []
    let classes = ''

    if(cols[0]) classes += `col-xs-${cols[0]} `
    if(cols[1]) classes += `col-sm-${cols[1]} `
    if(cols[2]) classes += `col-md-${cols[2]} `
    if(cols[3]) classes += `col-lg-${cols[3]} `

    return classes
  }

  render() {
    const gridClasses = this.toCssClasses(this.props.cols || 12)
    return (
      <div className={gridClasses}>
        {this.props.children}
      </div>
    )
  }
}
```

## Listagem 139 - Alterar arquivo `todoForm.jsx`

`todo-app/frontend/src/todo/todoForm.jsx` (aprox. linha 2, 6, 9, 10 e 14)

```
import React from 'react'
import Grid from '../template/grid'

export default props => (
  <div role='form' className='todoForm'>
    <Grid cols='12 9 10'>
      <input id='description' className='form-control'
        placeholder='Adicione uma tarefa'></input>
    </Grid>
    <Grid cols='12 3 2'>
      <button className='btn btn-primary'>
        <i className='fa fa-plus'></i>
      </button>
    </Grid>
  </div>
)
```

## Listagem 140 - Criar arquivo `iconButton.jsx`

`todo-app/frontend/src/template/iconButton.jsx`

```
import React from 'react'

export default props => {
  if(props.hide){
    return null
  } else {
    return (
      <button className={'btn btn-' + props.style}
        onClick={props.onClick}>
        <i className={'fa fa-' + props.icon}></i>
      </button>
    )
  }
}
```

### Listagem 141 - Alterar arquivo *todoForm.jsx*

*todo-app/frontend/src/todo/todoForm.jsx* (aprox. linha 3 e 12)

```
import React from 'react'
import Grid from '../template/grid'
import IconButton from '../template/iconButton'

export default props => (
  <div role='form' className='todoForm'>
    <Grid cols='12 9 10'>
      <input id='description' className='form-control'
        placeholder='Adicione uma tarefa'></input>
    </Grid>
    <Grid cols='12 3 2'>
      <IconButton style='primary' icon='plus'></IconButton>
    </Grid>
  </div>
)
```

## 5.13. Renderização Condicional (If)

### Listagem 142 - Criar arquivo *if.jsx*

*todo-app/frontend/src/template/if.jsx*

```
import React from 'react'

export default props => {
  if(props.test) {
    return props.children
  } else {
    return false
  }
}
```

### Listagem 143 - Sobrescrever arquivo `iconButton.jsx`

`todo-app/frontend/src/template/iconButton.jsx`

```
import React from 'react'
import If from './if'

export default props => (
  <If test={!props.hide}>
    <button className={'btn btn-' + props.style}
      onClick={props.onClick}>
      <i className={'fa fa-' + props.icon}></i>
    </button>
  </If>
)
-----
```

## 5.14. Evento Adicionar

### Listagem 144 - Alterar arquivo `todoForm.jsx`

`todo-app/frontend/src/todo/todoForm.jsx` (aprox. linha 13)

```
import React from 'react'
import Grid from '../template/grid'
import IconButton from '../template/iconButton'

export default props => (
  <div role='form' className='todoForm'>
    <Grid cols='12 9 10'>
      <input id='description' className='form-control'
        placeholder='Adicione uma tarefa'></input>
    </Grid>
    <Grid cols='12 3 2'>
      <IconButton style='primary' icon='plus'
        onClick={props.handleAdd}></IconButton>
    </Grid>
  </div>
)
```

## Listagem 145 - Alterar arquivo *todo.jsx*

*todo-app/frontend/src/todo/todo.jsx* (aprox. linha 7 e 20)

```
import React, { Component } from 'react'
import PageHeader from '../template/pageHeader'
import TodoForm from './todoForm'
import TodoList from './todoList'

export default class Todo extends Component {
  constructor(props) {
    super(props)
    this.handleAdd = this.handleAdd.bind(this)
  }

  handleAdd() {
    console.log(this)
  }

  render() {
    return (
      <div>
        <PageHeader name='Tarefas' small='Cadastro'></PageHeader>
        <TodoForm handleAdd={this.handleAdd} />
        <TodoList />
      </div>
    )
  }
}
```

## 5.15. Evento onChange



## Listagem 146 - Alterar arquivo *todoForm.jsx*

*todo-app/frontend/src/todo/todoForm.jsx* (aprox. linha 10 e 11)

```
import React from 'react'
import Grid from '../template/grid'
import IconButton from '../template/iconButton'

export default props => (
  <div role='form' className='todoForm'>
    <Grid cols='12 9 10'>
      <input id='description' className='form-control'
        placeholder='Adicione uma tarefa'
        onChange={props.handleChange}
        value={props.description}></input>
    </Grid>
    <Grid cols='12 3 2'>
      <IconButton style='primary' icon='plus'
        onClick={props.handleAdd}></IconButton>
    </Grid>
  </div>
)
```

## Listagem 147 - Alterar arquivo *todo.jsx*

*todo-app/frontend/src/todo/todo.jsx* (aprox. linha 9, 11, 15, 20, 23 e 27)

```
import React, { Component } from 'react'
import PageHeader from '../template/pageHeader'
import TodoForm from './todoForm'
import TodoList from './todoList'

export default class Todo extends Component {
  constructor(props) {
    super(props)
    this.state = { description: '', list: [] }

    this.handleChange = this.handleChange.bind(this)
    this.handleAdd = this.handleAdd.bind(this)
  }

  handleChange(e) {
    this.setState({...this.state, description: e.target.value })
  }

  handleAdd() {
    console.log(this.state.description)
  }

  render() {
    return (
      <div>
        <PageHeader name='Tarefas' small='Cadastro'></PageHeader>
        <TodoForm description={this.state.description}
          handleChange={this.handleChange}
          handleAdd={this.handleAdd} />
        <TodoList />
      </div>
    )
  }
}
```

## 5.16. Evento Adicionar (Integração Backend)

## Listagem 148 - Alterar arquivo *todo.jsx*

*todo-app/frontend/src/todo/todo.jsx* (aprox. linha 2, 8 e 24)

```
import React, { Component } from 'react'
import axios from 'axios'

import PageHeader from '../template/pageHeader'
import TodoForm from './todoForm'
import TodoList from './todoList'

const URL = 'http://localhost:3003/api/todos'

export default class Todo extends Component {
  constructor(props) {
    super(props)
    this.state = { description: '', list: [] }

    this.handleChange = this.handleChange.bind(this)
    this.handleAdd = this.handleAdd.bind(this)
  }

  handleChange(e) {
    this.setState({ ...this.state, description: e.target.value })
  }

  handleAdd() {
    const description = this.state.description
    axios.post(URL, { description })
      .then(resp => console.log('Funcionou!'))
  }

  render() {
    return (
      <div>
        <PageHeader name='Tarefas' small='Cadastro'></PageHeader>
        <TodoForm description={this.state.description}
          handleChange={this.handleChange}
          handleAdd={this.handleAdd} />
        <TodoList />
      </div>
    )
  }
}
```

## 5.17. Consulta e Exclusão de TODOs

## Listagem 149 - Alterar arquivo *todo.jsx*

*todo-app/frontend/src/todo/todo.jsx* (aprox. linha 18, 20, 23, 34, 37 e 49)

```
import React, { Component } from 'react'
import axios from 'axios'

import PageHeader from '../template/pageHeader'
import TodoForm from './todoForm'
import TodoList from './todoList'

const URL = 'http://localhost:3003/api/todos'

export default class Todo extends Component {
  constructor(props) {
    super(props)
    this.state = { description: '', list: [] }

    this.handleChange = this.handleChange.bind(this)
    this.handleAdd = this.handleAdd.bind(this)

    this.handleRemove = this.handleRemove.bind(this)

    this.refresh()
  }

  refresh() {
    axios.get(`${URL}?sort=-createdAt`)
      .then(resp => this.setState({...this.state, description: '', list:
resp.data}))
  }

  handleChange(e) {
    this.setState({...this.state, description: e.target.value })
  }

  handleAdd() {
    const description = this.state.description
    axios.post(URL, { description })
      .then(resp => this.refresh())
  }

  handleRemove(todo) {
    axios.delete(`${URL}/${todo._id}`)
      .then(resp => this.refresh())
  }

  render() {
    return (
      <div>
```

```

        <PageHeader name='Tarefas' small='Cadastro'></PageHeader>
        <TodoForm description={this.state.description}
          handleChange={this.handleChange}
          handleAdd={this.handleAdd} />
        <TodoList list={this.state.list}
          handleRemove={this.handleRemove} />
      </div>
    )
  }
}

```

## Listagem 150 - Sobrescrever arquivo todoList.jsx

todo-app/frontend/src/todo/todoList.jsx

```

import React from 'react'
import IconButton from '../template/iconButton'

export default props => {

  const renderRows = () => {
    const list = props.list || []
    return list.map(todo => (
      <tr key={todo._id}>
        <td>{todo.description}</td>
        <td>
          <IconButton style='danger' icon='trash-o'
            onClick={() => props.handleRemove(todo)}></IconButton>
        </td>
      </tr>
    ))
  }

  return (
    <table className='table'>
      <thead>
        <tr>
          <th>Descrição</th>
          <th>Ações</th>
        </tr>
      </thead>
      <tbody>
        {renderRows()}
      </tbody>
    </table>
  )
}

```

## 5.18. Marcar como Concluído/Pendente

### Listagem 151 - Alterar arquivo *todoList.jsx*

todo-app/frontend/src/todo/todoList.jsx (aprox. linha 10 e 12)

```
import React from 'react'
import IconButton from '../template/iconButton'

export default props => {

  const renderRows = () => {
    const list = props.list || []
    return list.map(todo => (
      <tr key={todo._id}>
        <td className={todo.done ? 'markAsDone' : ''}
        >{todo.description}</td>
        <td>
          <IconButton style='success' icon='check' hide={todo.done}
            onClick={() =>
              props.handleMarkAsDone(todo)}></IconButton>
          <IconButton style='warning' icon='undo' hide={!todo.done}
            onClick={() =>
              props.handleMarkAsPending(todo)}></IconButton>
          <IconButton style='danger' icon='trash-o' hide={!todo.done}
            onClick={() => props.handleRemove(todo)}></IconButton>
        </td>
      </tr>
    ))
  }

  return (
    <table className='table'>
      <thead>
        <tr>
          <th>Descrição</th>
          <th>Ações</th>
        </tr>
      </thead>
      <tbody>
        {renderRows()}
      </tbody>
    </table>
  )
}
```

## Listagem 152 - Alterar arquivo *todo.jsx*

*todo-app/frontend/src/todo/todo.jsx* (aprox. linha 18, 43, 48 e 63)

```
import React, { Component } from 'react'
import axios from 'axios'

import PageHeader from '../template/pageHeader'
import TodoForm from './todoForm'
import TodoList from './todoList'

const URL = 'http://localhost:3003/api/todos'

export default class Todo extends Component {
  constructor(props) {
    super(props)
    this.state = { description: '', list: [] }

    this.handleChange = this.handleChange.bind(this)
    this.handleAdd = this.handleAdd.bind(this)

    this.handleMarkAsDone = this.handleMarkAsDone.bind(this)
    this.handleMarkAsPending = this.handleMarkAsPending.bind(this)
    this.handleRemove = this.handleRemove.bind(this)

    this.refresh()
  }

  refresh() {
    axios.get(`${URL}?sort=-createdAt`)
      .then(resp => this.setState({...this.state, description: '', list:
resp.data}))
  }

  handleChange(e) {
    this.setState({...this.state, description: e.target.value })
  }

  handleAdd() {
    const description = this.state.description
    axios.post(URL, { description })
      .then(resp => this.refresh())
  }

  handleRemove(todo) {
    axios.delete(`${URL}/${todo._id}`)
      .then(resp => this.refresh())
  }

  handleMarkAsDone(todo) {
```

```

    axios.put(`${URL}/${todo._id}`, { ...todo, done: true })
      .then(resp => this.refresh())
  }

  handleMarkAsPending(todo) {
    axios.put(`${URL}/${todo._id}`, { ...todo, done: false })
      .then(resp => this.refresh())
  }

  render() {
    return (
      <div>
        <PageHeader name='Tarefas' small='Cadastro'></PageHeader>
        <TodoForm description={this.state.description}
          handleChange={this.handleChange}
          handleAdd={this.handleAdd} />
        <TodoList list={this.state.list}
          handleMarkAsDone={this.handleMarkAsDone}
          handleMarkAsPending={this.handleMarkAsPending}
          handleRemove={this.handleRemove} />
      </div>
    )
  }
}

```

### Listagem 153 - Criar arquivo custom.css

todo-app/frontend/src/template/custom.css

```

.btn {
  margin-right: 5px;
}

.markedAsDone {
  text-decoration: line-through;
  color: #777;
}

```



## Listagem 154 - Alterar arquivo app.jsx

todo-app/frontend/src/main/app.jsx (aprox. linha 3)

```
import 'modules/bootstrap/dist/css/bootstrap.min.css'
import 'modules/font-awesome/css/font-awesome.min.css'
import '../template/custom.css'

import React from 'react'
import Menu from '../template/menu'
import Routes from './routes'

export default props => (
  <div className='container'>
    <Menu />
    <Routes />
  </div>
)
```

## 5.19. Pesquisa de TODOs

### Listagem 155 - Alterar arquivo todoForm.jsx

todo-app/frontend/src/todo/todoForm.jsx (aprox. linha 16)

```
import React from 'react'
import Grid from '../template/grid'
import IconButton from '../template/iconButton'

export default props => (
  <div role='form' className='todoForm'>
    <Grid cols='12 9 10'>
      <input id='description' className='form-control'
        placeholder='Adicione uma tarefa'
        onChange={props.handleChange}
        value={props.description}></input>
    </Grid>
    <Grid cols='12 3 2'>
      <IconButton style='primary' icon='plus'
        onClick={props.handleAdd}></IconButton>
      <IconButton style='info' icon='search'
        onClick={props.handleSearch}></IconButton>
    </Grid>
  </div>
)
```

## Listagem 156 - Alterar arquivo *todo.jsx*

*todo-app/frontend/src/todo/todo.jsx* (aprox. linha 17, 26, 32, 48, 53, 58 e 69)

```
import React, { Component } from 'react'
import axios from 'axios'

import PageHeader from '../template/pageHeader'
import TodoForm from './todoForm'
import TodoList from './todoList'

const URL = 'http://localhost:3003/api/todos'

export default class Todo extends Component {
  constructor(props) {
    super(props)
    this.state = { description: '', list: [] }

    this.handleChange = this.handleChange.bind(this)
    this.handleAdd = this.handleAdd.bind(this)
    this.handleSearch = this.handleSearch.bind(this)

    this.handleMarkAsDone = this.handleMarkAsDone.bind(this)
    this.handleMarkAsPending = this.handleMarkAsPending.bind(this)
    this.handleRemove = this.handleRemove.bind(this)

    this.refresh()
  }

  refresh(description = '') {
    const search = description ? `&description__regex=${description}/` : ''
    axios.get(`${URL}?sort=-createdAt${search}`)
      .then(resp => this.setState({...this.state, description, list:
resp.data}))
  }

  handleSearch() {
    this.refresh(this.state.description)
  }

  handleChange(e) {
    this.setState({...this.state, description: e.target.value })
  }

  handleAdd() {
    const description = this.state.description
    axios.post(URL, { description })
      .then(resp => this.refresh())
  }
}
```

```

handleRemove(todo) {
  axios.delete(`${URL}/${todo._id}`)
    .then(resp => this.refresh(this.state.description))
}

handleMarkAsDone(todo) {
  axios.put(`${URL}/${todo._id}`, { ...todo, done: true })
    .then(resp => this.refresh(this.state.description))
}

handleMarkAsPending(todo) {
  axios.put(`${URL}/${todo._id}`, { ...todo, done: false })
    .then(resp => this.refresh(this.state.description))
}

render() {
  return (
    <div>
      <PageHeader name='Tarefas' small='Cadastro'></PageHeader>
      <TodoForm description={this.state.description}
        handleChange={this.handleChange}
        handleAdd={this.handleAdd}
        handleSearch={this.handleSearch} />
      <TodoList list={this.state.list}
        handleMarkAsDone={this.handleMarkAsDone}
        handleMarkAsPending={this.handleMarkAsPending}
        handleRemove={this.handleRemove} />
    </div>
  )
}
}

```

## 5.20. Melhorias de CSS e Limpar Formulário

## Listagem 157 - Alterar arquivo *custom.css*

todo-app/frontend/src/template/custom.css (aprox. linha 10)

```
.btn {  
  margin-right: 5px;  
}  
  
.markedAsDone {  
  text-decoration: line-through;  
  color: #777;  
}  
  
.tableActions {  
  width: 105px;  
}  
  
.todoForm {  
  padding-bottom: 60px;  
}
```

## Listagem 158 - Alterar arquivo *todoList.jsx*

todo-app/frontend/src/todo/todoList.jsx (aprox. linha 28)

```
import React from 'react'
import IconButton from '../template/iconButton'

export default props => {

  const renderRows = () => {
    const list = props.list || []
    return list.map(todo => (
      <tr key={todo._id}>
        <td className={todo.done ? 'markAsDone' : ''}
        >{todo.description}</td>
        <td>
          <IconButton style='success' icon='check' hide={todo.done}
            onClick={() =>
              props.handleMarkAsDone(todo)}></IconButton>
          <IconButton style='warning' icon='undo' hide={!todo.done}
            onClick={() =>
              props.handleMarkAsPending(todo)}></IconButton>
          <IconButton style='danger' icon='trash-o' hide={!todo.done}
            onClick={() => props.handleRemove(todo)}></IconButton>
        </td>
      </tr>
    ))
  }

  return (
    <table className='table'>
      <thead>
        <tr>
          <th>Descrição</th>
          <th className='tableActions'>Ações</th>
        </tr>
      </thead>
      <tbody>
        {renderRows()}
      </tbody>
    </table>
  )
}
```

## Listagem 159 - Alterar arquivo `todoForm.jsx`

`todo-app/frontend/src/todo/todoForm.jsx` (aprox. linha 18)

```
import React from 'react'
import Grid from '../template/grid'
import IconButton from '../template/iconButton'

export default props => (
  <div role='form' className='todoForm'>
    <Grid cols='12 9 10'>
      <input id='description' className='form-control'
        placeholder='Adicione uma tarefa'
        onChange={props.handleChange}
        value={props.description}></input>
    </Grid>
    <Grid cols='12 3 2'>
      <IconButton style='primary' icon='plus'
        onClick={props.handleAdd}></IconButton>
      <IconButton style='info' icon='search'
        onClick={props.handleSearch}></IconButton>
      <IconButton style='default' icon='close'
        onClick={props.handleClear}></IconButton>
    </Grid>
  </div>
)
```

## Listagem 160 - Alterar arquivo *todo.jsx*

*todo-app/frontend/src/todo/todo.jsx* (aprox. linha 18, 61 e 75)

```
import React, { Component } from 'react'
import axios from 'axios'

import PageHeader from '../template/pageHeader'
import TodoForm from './todoForm'
import TodoList from './todoList'

const URL = 'http://localhost:3003/api/todos'

export default class Todo extends Component {
  constructor(props) {
    super(props)
    this.state = { description: '', list: [] }

    this.handleChange = this.handleChange.bind(this)
    this.handleAdd = this.handleAdd.bind(this)
    this.handleSearch = this.handleSearch.bind(this)
    this.handleClear = this.handleClear.bind(this)

    this.handleMarkAsDone = this.handleMarkAsDone.bind(this)
    this.handleMarkAsPending = this.handleMarkAsPending.bind(this)
    this.handleRemove = this.handleRemove.bind(this)

    this.refresh()
  }

  refresh(description = '') {
    const search = description ? `&description__regex=/${description}/` : ''
    axios.get(`${URL}?sort=-createdAt${search}`)
      .then(resp => this.setState({...this.state, description, list:
resp.data}))
  }

  handleSearch() {
    this.refresh(this.state.description)
  }

  handleChange(e) {
    this.setState({...this.state, description: e.target.value })
  }

  handleAdd() {
    const description = this.state.description
    axios.post(URL, { description })
      .then(resp => this.refresh())
  }
}
```

```

handleRemove(todo) {
  axios.delete(`${URL}/${todo._id}`)
    .then(resp => this.refresh(this.state.description))
}

handleMarkAsDone(todo) {
  axios.put(`${URL}/${todo._id}`, { ...todo, done: true })
    .then(resp => this.refresh(this.state.description))
}

handleMarkAsPending(todo) {
  axios.put(`${URL}/${todo._id}`, { ...todo, done: false })
    .then(resp => this.refresh(this.state.description))
}

handleClear() {
  this.refresh()
}

render() {
  return (
    <div>
      <PageHeader name='Tarefas' small='Cadastro'></PageHeader>
      <TodoForm description={this.state.description}
        handleChange={this.handleChange}
        handleAdd={this.handleAdd}
        handleSearch={this.handleSearch}
        handleClear={this.handleClear} />
      <TodoList list={this.state.list}
        handleMarkAsDone={this.handleMarkAsDone}
        handleMarkAsPending={this.handleMarkAsPending}
        handleRemove={this.handleRemove} />
    </div>
  )
}
}

```

## 5.21. Adicionar Teclas de Atalhos



## Listagem 161 - Alterar arquivo *todoForm.jsx*

*todo-app/frontend/src/todo/todoForm.jsx* (aprox. linha 5, 6, 20 e 32)

```
import React from 'react'
import Grid from '../template/grid'
import IconButton from '../template/iconButton'

export default props => {
  const keyHandler = (e) => {
    if(e.key === 'Enter'){
      e.shiftKey ? props.handleSearch() : props.handleAdd()
    } else if(e.key === 'Escape') {
      props.handleClear()
    }
  }

  return (
    <div role='form' className='todoForm'>
      <Grid cols='12 9 10'>
        <input id='description' className='form-control'
          placeholder='Adicione uma tarefa'
          onChange={props.handleChange}
          onKeyUp={keyHandler}
          value={props.description}></input>
      </Grid>
      <Grid cols='12 3 2'>
        <IconButton style='primary' icon='plus'
          onClick={props.handleAdd}></IconButton>
        <IconButton style='info' icon='search'
          onClick={props.handleSearch}></IconButton>
        <IconButton style='default' icon='close'
          onClick={props.handleClear}></IconButton>
      </Grid>
    </div>
  )
}
```

## 6. React com Redux

### 6.1. Palavras Iniciais e Instalação

Abra o terminal e dentro da pasta *exercicios\_react* execute o seguinte comando:

#### Listagem 162 - Instalando as dependências

```
npm i --save-dev react-redux@5.0.2 redux@3.6.0
```

#### Listagem 163 - Executar a aplicação

```
npm run dev
```

### 6.2. Conhecendo o Redux

### 6.3. Configurando o Redux

#### Listagem 164 - Alterar arquivo *index.jsx*

*exercicios\_react/ex/index.jsx* (aprox. linha 3, 4, 7 e 12)

```
import React from 'react'
import ReactDOM from 'react-dom'
import { combineReducers, createStore } from 'redux'
import { Provider } from 'react-redux'
import Field from './field'

const reducers = combineReducers({
  field: () => ({ value: 'Opa' })
})

ReactDOM.render(
  <Provider store={createStore(reducers)}>
    <Field initialValue='Teste' />
  </Provider>
, document.getElementById('app'))
```

## 6.4. Ex 01 - Integrando o React-Redux (Parte 1)

### Listagem 165 - Alterar arquivo *field.js*

*exercicios\_react/ex/field.js* (aprox. linha 2, 5, 8, 15 e 21)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'

class Field extends Component {
  render() {
    return (
      <div>
        <label>{this.props.value}</label><br />
        <input onChange={this.handleChange} value={this.props.value} />
      </div>
    )
  }
}

function mapStateToProps(state) {
  return {
    value: state.field.value
  }
}

export default connect(mapStateToProps)(Field)
```

## 6.5. Ex 01 - Integrando o React-Redux (Parte 2)

### Listagem 166 - Criar arquivo *fieldActions.js*

*exercicios\_react/ex/fieldActions.js*

```
export function changeValue(e) {
  console.log('changeValue')
  return {
    type: 'VALUE_CHANGED',
    payload: e.target.value
  }
}
```

## Listagem 167 - Alterar arquivo field.js

exercicios\_react/ex/field.js (aprox. linha 3, 4, 11, 23 e 27)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { changeValue } from '../fieldActions'

class Field extends Component {
  render() {
    return (
      <div>
        <label>{this.props.value}</label><br />
        <input onChange={this.props.changeValue} value={this.props.value}
      />
      </div>
    )
  }
}

function mapStateToProps(state) {
  return {
    value: state.field.value
  }
}

function mapDispatchToProps(dispatch) {
  return bindActionCreators({ changeValue }, dispatch)
}

export default connect(mapStateToProps, mapDispatchToProps)(Field)
```

## 6.6. Ex 01 - Integrando o React-Redux (Parte 3)

### Listagem 168 - Criar arquivo *fieldReducer.js*

*exercicios\_react/ex/fieldReducer.js*

```
const INITIAL_STATE = { value: 'Opa' }

export default function(state = INITIAL_STATE, action) {
  switch(action.type) {
    case 'VALUE_CHANGED':
      return { value: action.payload }
    default:
      return state
  }
}
```

### Listagem 169 - Alterar arquivo *index.jsx*

*exercicios\_react/ex/index.jsx* (aprox. linha 6 e 9)

```
import React from 'react'
import ReactDOM from 'react-dom'
import { combineReducers, createStore } from 'redux'
import { Provider } from 'react-redux'
import Field from './field'
import fieldReducer from './fieldReducer'

const reducers = combineReducers({
  field: fieldReducer
})

ReactDOM.render(
  <Provider store={createStore(reducers)}>
    <Field initialValue='Teste' />
  </Provider>
, document.getElementById('app'))
```

## 6.7. Ex 02 - Integrando o React-Redux (Parte 1)

## Listagem 170 - Criar arquivo counterActions.js

exercicios\_react/ex/counterActions.js

```
export function inc() {
  return { type: 'INC' }
}

export function dec() {
  return { type: 'DEC' }
}

export function stepChanged(e) {
  return {
    type: 'STEP_CHANGED',
    payload: e.target.value
  }
}
```

## Listagem 171 - Criar arquivo counterReducer.js

exercicios\_react/ex/counterReducer.js

```
const INITIAL_STATE = { step: 1, number: 0 }

export default function(state = INITIAL_STATE, action) {
  switch(action.type) {
    case 'INC':
      return { ...state, number: state.number + state.step }
    case 'DEC':
      return { ...state, number: state.number - state.step }
    case 'STEP_CHANGED':
      return { ...state, step: action.payload }
    default:
      return state
  }
}
```

### Listagem 172 - Criar arquivo counter.jsx

exercicios\_react/ex/counter.jsx

```
import React from 'react'

const Counter = props => (
  <div>Ok</div>
)

export default Counter
```

## 6.8. Ex 02 - Integrando o React-Redux (Parte 2)

### Listagem 173 - Alterar arquivo index.jsx

exercicios\_react/ex/index.jsx (aprox. linha 5, 6, 9 e 14)

```
import React from 'react'
import ReactDOM from 'react-dom'
import { combineReducers, createStore } from 'redux'
import { Provider } from 'react-redux'
import counterReducer from './counterReducer'
import Counter from './counter'

const reducers = combineReducers({
  counter: counterReducer
})

ReactDOM.render(
  <Provider store={createStore(reducers)}>
    <Counter />
  </Provider>
, document.getElementById('app'))
```

## Listagem 174 - Sobrescrever arquivo counter.jsx

exercicios\_react/ex/counter.jsx

```
import React from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'
import { inc, dec, stepChanged } from './counterActions'

const Counter = props => (
  <div>
    <h1>{props.counter.number}</h1>
    <input onChange={props.stepChanged}
      value={props.counter.step} type='number' />
    <button onClick={props.dec}>Dec</button>
    <button onClick={props.inc}>Inc</button>
  </div>
)

const mapStateToProps = state => ({ counter: state.counter })
const mapDispatchToProps =
  dispatch => bindActionCreators({ inc, dec, stepChanged }, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(Counter)
```

## Listagem 175 - Alterar arquivo counterReducer.js

exercicios\_react/ex/counterReducer.js (aprox. linha 10)

```
const INITIAL_STATE = { step: 1, number: 0 }

export default function(state = INITIAL_STATE, action) {
  switch(action.type) {
    case 'INC':
      return { ...state, number: state.number + state.step }
    case 'DEC':
      return { ...state, number: state.number - state.step }
    case 'STEP_CHANGED':
      return { ...state, step: +action.payload }
    default:
      return state
  }
}
```



## 7. TodoApp ( Migração para Redux)

### 7.1. Instalando as Dependências

Abra o terminal e dentro da pasta `todo-app/frontend` execute o seguinte comando:

#### Listagem 176 - Instalando as dependências

```
npm i --save-dev redux@3.6.0 react-redux@5.0.2
```

#### Listagem 177 - Abrir o Visual Studio Code

```
code .
```

### 7.2. Configurando os Reducers

## Listagem 178 - Criar arquivo reducers.js

todo-app/frontend/src/main/reducers.js

```
import { combineReducers } from 'redux'

const rootReducer = combineReducers({
  todo: () => ({
    description: 'Ler livro',
    list: [{
      _id: 1,
      description: 'Pagar fatura do cartão',
      done: true
    },
    {
      _id: 2,
      description: 'Reunião com a equipe às 10:00',
      done: false
    },
    {
      _id: 3,
      description: 'Consulta médicas na terça depois do almoço',
      done: false
    }
  ]
  })
})

export default rootReducer
```

## 7.3. Configurando o Redux

## Listagem 179 - Alterar arquivo *index.jsx*

*todo-app/frontend/src/index.jsx* (aprox. linha 3, 4, 7, 9 e 11)

```
import React from 'react'
import ReactDOM from 'react-dom'
import { createStore } from 'redux'
import { Provider } from 'react-redux'

import App from './main/app'
import reducers from './main/reducers'

const store = createStore(reducers)
ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>
, document.getElementById('app'))
```

## 7.4. Conectar o TodoList com Redux

## Listagem 180 - Alterar arquivo *todoList.jsx*

*todo-app/frontend/src/todo/todoList.jsx* (aprox. linha 5, 39 e 40)

```
import React from 'react'
import IconButton from '../template/iconButton'

const TodoList = props => {

  const renderRows = () => {
    const list = props.list || []
    return list.map(todo => (
      <tr key={todo._id}>
        <td className={todo.done ? 'markAsDone' : ''}
        >{todo.description}</td>
        <td>
          <IconButton style='success' icon='check' hide={todo.done}
            onClick={() =>
              props.handleMarkAsDone(todo)}></IconButton>
          <IconButton style='warning' icon='undo' hide={!todo.done}
            onClick={() =>
              props.handleMarkAsPending(todo)}></IconButton>
          <IconButton style='danger' icon='trash-o' hide={!todo.done}
            onClick={() => props.handleRemove(todo)}></IconButton>
        </td>
      </tr>
    ))
  }

  return (
    <table className='table'>
      <thead>
        <tr>
          <th>Descrição</th>
          <th className='tableActions'>Ações</th>
        </tr>
      </thead>
      <tbody>
        {renderRows()}
      </tbody>
    </table>
  )
}

const mapStateToProps = state => ({list: state.todo.list})
export default connect(mapStateToProps)(TodoList)
```

## 7.5. Conectar o TodoForm com Redux

### Listagem 181 - Alterar arquivo *todoForm.jsx*

*todo-app/frontend/src/todo/todoForm.jsx* (aprox. linha 2, 7, 37 e 38)

```
import React from 'react'
import { connect } from 'react-redux'

import Grid from '../template/grid'
import IconButton from '../template/iconButton'

const TodoForm = props => {
  const keyHandler = (e) => {
    if(e.key === 'Enter'){
      e.shiftKey ? props.handleSearch() : props.handleAdd()
    } else if(e.key === 'Escape') {
      props.handleClear()
    }
  }

  return (
    <div role='form' className='todoForm'>
      <Grid cols='12 9 10'>
        <input id='description' className='form-control'
          placeholder='Adicione uma tarefa'
          onChange={props.handleChange}
          onKeyDown={keyHandler}
          value={props.description}></input>
      </Grid>
      <Grid cols='12 3 2'>
        <IconButton style='primary' icon='plus'
          onClick={props.handleAdd}></IconButton>
        <IconButton style='info' icon='search'
          onClick={props.handleSearch}></IconButton>
        <IconButton style='default' icon='close'
          onClick={props.handleClear}></IconButton>
      </Grid>
    </div>
  )
}

const mapStateToProps = state => ({description: state.todo.description})
export default connect(mapStateToProps)(TodoForm)
```

## 7.6. Action Creator changeDescription

### Listagem 182 - Criar arquivo *todoActions.js*

todo-app/frontend/src/todo/todoActions.js

```
export const changeDescription = event => ({
  type: 'DESCRIPTION_CHANGED',
  payload: event.target.value
})
```

## Listagem 183 - Alterar arquivo todoForm.jsx

todo-app/frontend/src/todo/todoForm.jsx (aprox. linha 3, 7, 23, 40 e 42)

```
import React from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'

import Grid from '../template/grid'
import IconButton from '../template/iconButton'
import { changeDescription } from './todoActions'

const TodoForm = props => {
  const keyHandler = (e) => {
    if(e.key === 'Enter'){
      e.shiftKey ? props.handleSearch() : props.handleAdd()
    } else if(e.key === 'Escape') {
      props.handleClear()
    }
  }

  return (
    <div role='form' className='todoForm'>
      <Grid cols='12 9 10'>
        <input id='description' className='form-control'
          placeholder='Adicione uma tarefa'
          onChange={props.changeDescription}
          onKeyDown={keyHandler}
          value={props.description}></input>
      </Grid>
      <Grid cols='12 3 2'>
        <IconButton style='primary' icon='plus'
          onClick={props.handleAdd}></IconButton>
        <IconButton style='info' icon='search'
          onClick={props.handleSearch}></IconButton>
        <IconButton style='default' icon='close'
          onClick={props.handleClear}></IconButton>
      </Grid>
    </div>
  )
}

const mapStateToProps = state => ({description: state.todo.description})
const mapDispatchToProps = dispatch =>
  bindActionCreators({ changeDescription }, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(TodoForm)
```

## Listagem 184 - Criar arquivo *todoReducer.js*

todo-app/frontend/src/todo/todoReducer.js

```
const INITIAL_STATE = {
  description: 'Ler livro',
  list: [{
    _id: 1,
    description: 'Pagar fatura do cartão',
    done: true
  },
  {
    _id: 2,
    description: 'Reunião com a equipe às 10:00',
    done: false
  },
  {
    _id: 3,
    description: 'Consulta médicas na terça depois do almoço',
    done: false
  }
]
}

export default (state = INITIAL_STATE, action) => {
  switch(action.type) {
    case 'DESCRIPTION_CHANGED':
      return { ...state, description: action.payload }
    default:
      return state
  }
}
```

## Listagem 185 - Alterar arquivo *reducers.js*

todo-app/frontend/src/main/reducers.js (aprox. linha 5)

```
import { combineReducers } from 'redux'
import todoReducer from '../todo/todoReducer'

const rootReducer = combineReducers({
  todo: todoReducer
})

export default rootReducer
```



## 7.7. Configurando o Redux Dev Tools



Para instalar a extensão do plugin do *Redux DevTools* para Chrome acesse o link: <https://chrome.google.com/webstore/detail/redux-devtools/lmhkpmbekcpmknklieibfkpmmfbljd>

### Listagem 186 - Alterar arquivo *index.jsx*

*todo-app/frontend/src/index.jsx* (aprox. linha 9 e 11)

```
import React from 'react'
import ReactDOM from 'react-dom'
import { createStore } from 'redux'
import { Provider } from 'react-redux'

import App from './main/app'
import reducers from './main/reducers'

const devTools = window.__REDUX_DEVTOOLS_EXTENSION__
  && window.__REDUX_DEVTOOLS_EXTENSION__()
const store = createStore(reducers, devTools)
ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>
, document.getElementById('app'))
```

## 7.8. Action Creator de Pesquisa

## Listagem 187 - Alterar arquivo `todoActions.js`

`todo-app/frontend/src/todo/todoActions.js` (aprox. linha 1, 3 e 10)

```
import axios from 'axios'

const URL = 'http://localhost:3003/api/todos'

export const changeDescription = event => ({
  type: 'DESCRIPTION_CHANGED',
  payload: event.target.value
})

export const search = () => {
  request = axios.get(`${URL}?sort=-createdAt`)
  return {
    type: 'TODO_SEARCHED',
    payload: request
  }
}
```

## Listagem 188 - Alterar arquivo *todoReducer.js*

*todo-app/frontend/src/todo/todoReducer.js* (aprox. linha 22)

```
const INITIAL_STATE = {
  description: 'Ler livro',
  list: [{
    _id: 1,
    description: 'Pagar fatura do cartão',
    done: true
  },
  {
    _id: 2,
    description: 'Reunião com a equipe às 10:00',
    done: false
  },
  {
    _id: 3,
    description: 'Consulta médicas na terça depois do almoço',
    done: false
  }
  ]
}

export default (state = INITIAL_STATE, action) => {
  switch(action.type) {
    case 'DESCRIPTION_CHANGED':
      return { ...state, description: action.payload }
    case 'TODO_SEARCHED':
      return { ...state, list: action.payload.data }
    default:
      return state
  }
}
```

## Listagem 189 - Sobrescrever arquivo todoForm.jsx

todo-app/frontend/src/todo/todoForm.jsx

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'

import Grid from '../template/grid'
import IconButton from '../template/iconButton'
import { changeDescription, search } from './todoActions'

class TodoForm extends Component {
  constructor(props) {
    super(props)
    this.keyHandler = this.keyHandler.bind(this)
  }

  componentWillMount() {
    this.props.search()
  }

  keyHandler(e) {
    if(e.key === 'Enter'){
      e.shiftKey ? this.props.handleSearch() : this.props.handleAdd()
    } else if(e.key === 'Escape') {
      props.handleClear()
    }
  }

  render() {
    return (
      <div role='form' className='todoForm'>
        <Grid cols='12 9 10'>
          <input id='description' className='form-control'
            placeholder='Adicione uma tarefa'
            onChange={this.props.changeDescription}
            onKeyDown={this.keyHandler}
            value={this.props.description}></input>
        </Grid>
        <Grid cols='12 3 2'>
          <IconButton style='primary' icon='plus'
            onClick={this.props.handleAdd}></IconButton>
          <IconButton style='info' icon='search'
            onClick={this.props.handleSearch}></IconButton>
          <IconButton style='default' icon='close'
            onClick={this.props.handleClear}></IconButton>
        </Grid>
      </div>
    )
  }
}
```

```

    }
  }

  const mapStateToProps = state => ({description: state.todo.description})
  const mapDispatchToProps = dispatch =>
    bindActionCreators({ changeDescription, search }, dispatch)
  export default connect(mapStateToProps, mapDispatchToProps)(TodoForm)

```

## 7.9. Aplicando Middleware (Redux-Promise)

Abra o terminal e dentro da pasta `todo-app/frontend` execute o seguinte comando:

### Listagem 190 - Instalando a dependência

```
npm i --save-dev redux-promise@0.5.3
```

### Listagem 191 - Executar a aplicação

```
npm run dev
```

### Listagem 192 - Alterar arquivo `index.jsx`

`todo-app/frontend/src/index.jsx` (aprox. linha 3, 5 e 12)

```

import React from 'react'
import ReactDOM from 'react-dom'
import { applyMiddleware, createStore } from 'redux'
import { Provider } from 'react-redux'
import promise from 'redux-promise'

import App from './main/app'
import reducers from './main/reducers'

const devTools = window.__REDUX_DEVTOOLS_EXTENSION__
  && window.__REDUX_DEVTOOLS_EXTENSION__()
const store = applyMiddleware(promise)(createStore)(reducers, devTools)
ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>
, document.getElementById('app'))

```

## Listagem 193 - Alterar arquivo *todoReducer.js*

*todo-app/frontend/src/todo/todoReducer.js* (aprox. linha 1)

```
const INITIAL_STATE = { description: '', list: [] }

export default (state = INITIAL_STATE, action) => {
  switch(action.type) {
    case 'DESCRIPTION_CHANGED':
      return { ...state, description: action.payload }
    case 'TODO_SEARCHED':
      return { ...state, list: action.payload.data }
    default:
      return state
  }
}
```

## 7.10. Adicionar TODO

## Listagem 194 - Alterar arquivo `todoActions.js`

`todo-app/frontend/src/todo/todoActions.js` (aprox. linha 18)

```
import axios from 'axios'

const URL = 'http://localhost:3003/api/todos'

export const changeDescription = event => ({
  type: 'DESCRIPTION_CHANGED',
  payload: event.target.value
})

export const search = () => {
  request = axios.get(`${URL}?sort=-createdAt`)
  return {
    type: 'TODO_SEARCHED',
    payload: request
  }
}

export const add = (description) => {
  const request = axios.post(URL, { description })
  return {
    type: 'TODO_ADDED',
    payload: request
  }
}
```

## Listagem 195 - Alterar arquivo *todoReducer.js*

*todo-app/frontend/src/todo/todoReducer.js* (aprox. linha 1)

```
const INITIAL_STATE = { description: '', list: [] }

export default (state = INITIAL_STATE, action) => {
  switch(action.type) {
    case 'DESCRIPTION_CHANGED':
      return { ...state, description: action.payload }
    case 'TODO_SEARCHED':
      return { ...state, list: action.payload.data }
    case 'TODO_ADDED':
      return { ...state, description: '' }
    default:
      return state
  }
}
```



## Listagem 196 - Alterar arquivo todoForm.jsx

todo-app/frontend/src/todo/todoForm.jsx (aprox. linha 7, 20, 22, 29, 41, 43 e 53)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'

import Grid from '../template/grid'
import IconButton from '../template/iconButton'
import { add, changeDescription, search } from './todoActions'

class TodoForm extends Component {
  constructor(props) {
    super(props)
    this.keyHandler = this.keyHandler.bind(this)
  }

  componentWillMount() {
    this.props.search()
  }

  keyHandler(e) {
    const { add, search, description } = this.props
    if(e.key === 'Enter'){
      e.shiftKey ? search() : add(description)
    } else if(e.key === 'Escape') {
      props.handleClear()
    }
  }

  render() {
    const { add, search, description } = this.props
    return (
      <div role='form' className='todoForm'>
        <Grid cols='12 9 10'>
          <input id='description' className='form-control'
            placeholder='Adicione uma tarefa'
            onChange={this.props.changeDescription}
            onKeyDown={this.keyHandler}
            value={this.props.description}></input>
        </Grid>
        <Grid cols='12 3 2'>
          <IconButton style='primary' icon='plus'
            onClick={() => add(description)}></IconButton>
          <IconButton style='info' icon='search'
            onClick={() => search()}></IconButton>
          <IconButton style='default' icon='close'
            onClick={this.props.handleClear}></IconButton>
        </Grid>
      </div>
    )
  }
}
```

```
        </div>
      )
    }
  }

  const mapStateToProps = state => ({description: state.todo.description})
  const mapDispatchToProps = dispatch =>
    bindActionCreators({ add, changeDescription, search }, dispatch)
  export default connect(mapStateToProps, mapDispatchToProps)(TodoForm)
```

## 7.11. Aplicando Middleware (Redux-Multi)

Abra o terminal e dentro da pasta `todo-app/frontend` execute o seguinte comando:

### Listagem 197 - Instalando a dependência

```
npm i --save-dev redux-multi@0.1.12
```

### Listagem 198 - Executar a aplicação

```
npm run dev
```

## Listagem 199 - Alterar arquivo index.jsx

todo-app/frontend/src/index.jsx (aprox. linha 7 e 15)

```
import React from 'react'
import ReactDOM from 'react-dom'
import { applyMiddleware, createStore } from 'redux'
import { Provider } from 'react-redux'

import promise from 'redux-promise'
import multi from 'redux-multi'

import App from './main/app'
import reducers from './main/reducers'

const devTools = window.__REDUX_DEVTOOLS_EXTENSION__
  && window.__REDUX_DEVTOOLS_EXTENSION__()
const store = applyMiddleware(multi, promise)(createStore)(reducers, devTools)
ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>
, document.getElementById('app'))
```

### Listagem 200 - Alterar arquivo `todoActions.js`

`todo-app/frontend/src/todo/todoActions.js` (aprox. linha 20)

```
import axios from 'axios'

const URL = 'http://localhost:3003/api/todos'

export const changeDescription = event => ({
  type: 'DESCRIPTION_CHANGED',
  payload: event.target.value
})

export const search = () => {
  request = axios.get(`${URL}?sort=-createdAt`)
  return {
    type: 'TODO_SEARCHED',
    payload: request
  }
}

export const add = (description) => {
  const request = axios.post(URL, { description })
  return [
    { type: 'TODO_ADDED', payload: request },
    search()
  ]
}
```

## 7.12. Aplicando Middleware (Redux-Thunk)

Abra o terminal e dentro da pasta `todo-app/frontend` execute o seguinte comando:

### Listagem 201 - Instalando a dependência

```
npm i --save-dev redux-thunk@2.2.0
```

### Listagem 202 - Executar a aplicação

```
npm run dev
```

## Listagem 203 - Alterar arquivo index.jsx

todo-app/frontend/src/index.jsx (aprox. linha 8 e 15)

```
import React from 'react'
import ReactDOM from 'react-dom'
import { applyMiddleware, createStore } from 'redux'
import { Provider } from 'react-redux'

import promise from 'redux-promise'
import multi from 'redux-multi'
import thunk from 'redux-thunk'

import App from './main/app'
import reducers from './main/reducers'

const devTools = window.__REDUX_DEVTOOLS_EXTENSION__
  && window.__REDUX_DEVTOOLS_EXTENSION__()
const store = applyMiddleware(thunk, multi, promise)(createStore)(reducers,
devTools)
ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>
, document.getElementById('app'))
```

## Listagem 204 - Alterar arquivo `todoActions.js`

`todo-app/frontend/src/todo/todoActions.js` (aprox. linha 18)

```
import axios from 'axios'

const URL = 'http://localhost:3003/api/todos'

export const changeDescription = event => ({
  type: 'DESCRIPTION_CHANGED',
  payload: event.target.value
})

export const search = () => {
  request = axios.get(`${URL}?sort=-createdAt`)
  return {
    type: 'TODO_SEARCHED',
    payload: request
  }
}

export const add = (description) => {
  return dispatch => {
    axios.post(URL, { description })
      .then(resp => dispatch({ type: 'TODO_ADDED', payload: resp.data }))
      .then(resp => dispatch(search()))
  }
}
```

## 7.13. Marcar como Concluído/Pendente

## Listagem 205 - Alterar arquivo `todoActions.js`

`todo-app/frontend/src/todo/todoActions.js` (aprox. linha 26 e 33)

```
import axios from 'axios'

const URL = 'http://localhost:3003/api/todos'

export const changeDescription = event => ({
  type: 'DESCRIPTION_CHANGED',
  payload: event.target.value
})

export const search = () => {
  request = axios.get(`${URL}?sort=-createdAt`)
  return {
    type: 'TODO_SEARCHED',
    payload: request
  }
}

export const add = (description) => {
  return dispatch => {
    axios.post(URL, { description })
      .then(resp => dispatch({ type: 'TODO_ADDED', payload: resp.data }))
      .then(resp => dispatch(search()))
  }
}

export const markAsDone = (todo) => {
  return dispatch => {
    axios.put(`${URL}/${todo._id}`, { ...todo, done: true })
      .then(resp => dispatch(search()))
  }
}

export const markAsPending = (todo) => {
  return dispatch => {
    axios.put(`${URL}/${todo._id}`, { ...todo, done: false })
      .then(resp => dispatch(search()))
  }
}
```

## Listagem 206 - Alterar arquivo *todoList.jsx*



```
import React from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'

import IconButton from '../template/iconButton'
import { markAsDone, markAsPending } from './todoActions'

const TodoList = props => {

  const renderRows = () => {
    const list = props.list || []
    return list.map(todo => (
      <tr key={todo._id}>
        <td className={todo.done ? 'markAsDone' : ''}
        >{todo.description}</td>
        <td>
          <IconButton style='success' icon='check' hide={todo.done}
            onClick={() => props.markAsDone(todo)}></IconButton>
          <IconButton style='warning' icon='undo' hide={!todo.done}
            onClick={() => props.markAsPending(todo)}></IconButton>
          <IconButton style='danger' icon='trash-o' hide={!todo.done}
            onClick={() => props.handleRemove(todo)}></IconButton>
        </td>
      </tr>
    ))
  }

  return (
    <table className='table'>
      <thead>
        <tr>
          <th>Descrição</th>
          <th className='tableActions'>Ações</th>
        </tr>
      </thead>
      <tbody>
        {renderRows()}
      </tbody>
    </table>
  )
}

const mapStateToProps = state => ({list: state.todo.list})
const mapDispatchToProps = dispatch =>
  bindActionCreators({ markAsDone, markAsPending }, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(TodoList)
```

## 7.14. Exclusão de TODOs

## Listagem 207 - Alterar arquivo `todoActions.js`

`todo-app/frontend/src/todo/todoActions.js` (aprox. linha 40)

```
import axios from 'axios'

const URL = 'http://localhost:3003/api/todos'

export const changeDescription = event => ({
  type: 'DESCRIPTION_CHANGED',
  payload: event.target.value
})

export const search = () => {
  request = axios.get(`${URL}?sort=-createdAt`)
  return {
    type: 'TODO_SEARCHED',
    payload: request
  }
}

export const add = (description) => {
  return dispatch => {
    axios.post(URL, { description })
      .then(resp => dispatch({ type: 'TODO_ADDED', payload: resp.data }))
      .then(resp => dispatch(search()))
  }
}

export const markAsDone = (todo) => {
  return dispatch => {
    axios.put(`${URL}/${todo._id}`, { ...todo, done: true })
      .then(resp => dispatch(search()))
  }
}

export const markAsPending = (todo) => {
  return dispatch => {
    axios.put(`${URL}/${todo._id}`, { ...todo, done: false })
      .then(resp => dispatch(search()))
  }
}

export const remove = (todo) => {
  return dispatch => {
    axios.delete(`${URL}/${todo._id}`)
      .then(resp => dispatch(search()))
  }
}
```

## Listagem 208 - Alterar arquivo *todoList.jsx*

```
import React from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'

import IconButton from '../template/iconButton'
import { markAsDone, markAsPending, remove } from './todoActions'

const TodoList = props => {

  const renderRows = () => {
    const list = props.list || []
    return list.map(todo => (
      <tr key={todo._id}>
        <td className={todo.done ? 'markAsDone' : ''}
        >{todo.description}</td>
        <td>
          <IconButton style='success' icon='check' hide={todo.done}
            onClick={() => props.markAsDone(todo)}></IconButton>
          <IconButton style='warning' icon='undo' hide={!todo.done}
            onClick={() => props.markAsPending(todo)}></IconButton>
          <IconButton style='danger' icon='trash-o' hide={!todo.done}
            onClick={() => props.remove(todo)}></IconButton>
        </td>
      </tr>
    ))
  }

  return (
    <table className='table'>
      <thead>
        <tr>
          <th>Descrição</th>
          <th className='tableActions'>Ações</th>
        </tr>
      </thead>
      <tbody>
        {renderRows()}
      </tbody>
    </table>
  )
}

const mapStateToProps = state => ({list: state.todo.list})
const mapDispatchToProps = dispatch =>
  bindActionCreators({ markAsDone, markAsPending, remove }, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(TodoList)
```

## 7.15. Limpar Formulário

## **Listagem 209 - *Alterar arquivo todoActions.js***

```
import axios from 'axios'

const URL = 'http://localhost:3003/api/todos'

export const changeDescription = event => ({
  type: 'DESCRIPTION_CHANGED',
  payload: event.target.value
})

export const search = () => {
  request = axios.get(`${URL}?sort=-createdAt`)
  return {
    type: 'TODO_SEARCHED',
    payload: request
  }
}

export const add = (description) => {
  return dispatch => {
    axios.post(URL, { description })
      .then(resp => dispatch(clear()))
      .then(resp => dispatch(search()))
  }
}

export const markAsDone = (todo) => {
  return dispatch => {
    axios.put(`${URL}/${todo._id}`, { ...todo, done: true })
      .then(resp => dispatch(search()))
  }
}

export const markAsPending = (todo) => {
  return dispatch => {
    axios.put(`${URL}/${todo._id}`, { ...todo, done: false })
      .then(resp => dispatch(search()))
  }
}

export const remove = (todo) => {
  return dispatch => {
    axios.delete(`${URL}/${todo._id}`)
      .then(resp => dispatch(search()))
  }
}

export const clear = () => {
  return [{ type: 'TODO_CLEAR' }]
}
```



## Listagem 210 - Alterar arquivo *todoReducer.js*

*todo-app/frontend/src/todo/todoReducer.js* (aprox. linha 9)

```
const INITIAL_STATE = { description: '', list: [] }

export default (state = INITIAL_STATE, action) => {
  switch(action.type) {
    case 'DESCRIPTION_CHANGED':
      return { ...state, description: action.payload }
    case 'TODO_SEARCHED':
      return { ...state, list: action.payload.data }
    case 'TODO_CLEAR':
      return { ...state, description: '' }
    default:
      return state
  }
}
```

## Listagem 211 - Alterar arquivo todoForm.jsx

todo-app/frontend/src/todo/todoForm.jsx (aprox. linha 7, 20, 24, 45 e 54)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'

import Grid from '../template/grid'
import IconButton from '../template/iconButton'
import { add, changeDescription, search, clear } from './todoActions'

class TodoForm extends Component {
  constructor(props) {
    super(props)
    this.keyHandler = this.keyHandler.bind(this)
  }

  componentWillMount() {
    this.props.search()
  }

  keyHandler(e) {
    const { add, search, description, clear } = this.props
    if(e.key === 'Enter'){
      e.shiftKey ? search() : add(description)
    } else if(e.key === 'Escape') {
      clear()
    }
  }

  render() {
    const { add, search, description } = this.props
    return (
      <div role='form' className='todoForm'>
        <Grid cols='12 9 10'>
          <input id='description' className='form-control'
            placeholder='Adicione uma tarefa'
            onChange={this.props.changeDescription}
            onKeyDown={this.keyHandler}
            value={this.props.description}></input>
        </Grid>
        <Grid cols='12 3 2'>
          <IconButton style='primary' icon='plus'
            onClick={() => add(description)}></IconButton>
          <IconButton style='info' icon='search'
            onClick={() => search()}></IconButton>
          <IconButton style='default' icon='close'
            onClick={this.props.clear}></IconButton>
        </Grid>
      </div>
    )
  }
}
```

```
        </div>
      )
    }
  }

  const mapStateToProps = state => ({description: state.todo.description})
  const mapDispatchToProps = dispatch =>
    bindActionCreators({ add, changeDescription, search, clear }, dispatch)
  export default connect(mapStateToProps, mapDispatchToProps)(TodoForm)
```

## 7.16. Melhorando a Action de Pesquisa

## Listagem 212 - Alterar arquivo `todoActions.js`

`todo-app/frontend/src/todo/todoActions.js` (aprox. linha 10 e 49)

```
import axios from 'axios'

const URL = 'http://localhost:3003/api/todos'

export const changeDescription = event => ({
  type: 'DESCRIPTION_CHANGED',
  payload: event.target.value
})

export const search = () => {
  return (dispatch, getState) => {
    const description = getState().todo.description
    const search = description ? `&description__regex=${description}` : ''
    const request = axios.get(`${URL}?sort=-createdAt${search}`)
      .then(resp => dispatch({type: 'TODO_SEARCHED', payload: resp.data}))
  }
}

export const add = (description) => {
  return dispatch => {
    axios.post(URL, { description })
      .then(resp => dispatch(clear()))
      .then(resp => dispatch(search()))
  }
}

export const markAsDone = (todo) => {
  return dispatch => {
    axios.put(`${URL}/${todo._id}`, { ...todo, done: true })
      .then(resp => dispatch(search()))
  }
}

export const markAsPending = (todo) => {
  return dispatch => {
    axios.put(`${URL}/${todo._id}`, { ...todo, done: false })
      .then(resp => dispatch(search()))
  }
}

export const remove = (todo) => {
  return dispatch => {
    axios.delete(`${URL}/${todo._id}`)
      .then(resp => dispatch(search()))
  }
}
```

```
export const clear = () => {  
  return [{ type: 'TODO_CLEAR' }, search()]  
}
```

## Listagem 213 - Alterar arquivo todoForm.jsx

todo-app/frontend/src/todo/todoForm.jsx (aprox. linha 43)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'

import Grid from '../template/grid'
import IconButton from '../template/iconButton'
import { add, changeDescription, search, clear } from './todoActions'

class TodoForm extends Component {
  constructor(props) {
    super(props)
    this.keyHandler = this.keyHandler.bind(this)
  }

  componentWillMount() {
    this.props.search()
  }

  keyHandler(e) {
    const { add, search, description, clear } = this.props
    if(e.key === 'Enter'){
      e.shiftKey ? search() : add(description)
    } else if(e.key === 'Escape') {
      clear()
    }
  }

  render() {
    const { add, search, description } = this.props
    return (
      <div role='form' className='todoForm'>
        <Grid cols='12 9 10'>
          <input id='description' className='form-control'
            placeholder='Adicione uma tarefa'
            onChange={this.props.changeDescription}
            onKeyDown={this.keyHandler}
            value={this.props.description}></input>
        </Grid>
        <Grid cols='12 3 2'>
          <IconButton style='primary' icon='plus'
            onClick={() => add(description)}></IconButton>
          <IconButton style='info' icon='search'
            onClick={search}></IconButton>
          <IconButton style='default' icon='close'
            onClick={this.props.clear}></IconButton>
        </Grid>
      </div>
    )
  }
}
```

```

    </div>
  )
}
}

const mapStateToProps = state => ({description: state.todo.description})
const mapDispatchToProps = dispatch =>
  bindActionCreators({ add, changeDescription, search, clear }, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(TodoForm)

```

## Listagem 214 - Alterar arquivo *todoReducer.js*

*todo-app/frontend/src/todo/todoReducer.js* (aprox. linha 8)

```

const INITIAL_STATE = { description: '', list: [] }

export default (state = INITIAL_STATE, action) => {
  switch(action.type) {
    case 'DESCRIPTION_CHANGED':
      return { ...state, description: action.payload }
    case 'TODO_SEARCHED':
      return { ...state, list: action.payload }
    case 'TODO_CLEAR':
      return { ...state, description: '' }
    default:
      return state
  }
}

```

## Listagem 215 - Sobrescrever arquivo todo.jsx

todo-app/frontend/src/todo/todo.jsx

```
import React from 'react'

import PageHeader from '../template/pageHeader'
import TodoForm from './todoForm'
import TodoList from './todoList'

export default props => (
  <div>
    <PageHeader name='Tarefas' small='Cadastro'></PageHeader>
    <TodoForm />
    <TodoList />
  </div>
)
```



## 8. Aplicação Final - Ciclos de Pagamentos (Backend)

### 8.1. Visão Geral

### 8.2. Configuração e Instalação

Abra o terminal e dentro da pasta **Desktop** execute o seguinte comando:

#### Listagem 216 - Criar pasta my-money-app

my-money-app/

```
mkdir my-money-app && cd my-money-app
```

#### Listagem 217 - Criar pasta backend

my-money-app/backend

```
mkdir backend && cd backend
```

#### Listagem 218 - Criar arquivo package.json

my-money-app/backend/package.json

```
npm init -y
```

#### Listagem 219 - Instalando as dependências

```
npm i --save express@4.14.0 mongoose@4.7.0 body-parser@1.15.2 lodash@4.17.4  
mongoose-paginate@5.0.3 express-query-int@1.0.1 node-restful@0.2.5 pm2@2.1.5
```

#### Listagem 220 - Instalando a dependência

```
npm i --save-dev nodemon@1.11.0
```

## Listagem 221 - Alterar arquivo `package.json`

`my-money-app/backend/package.json` (aprox. linha 2, 5 e 7)

```
{
  "name": "my-money-backend",
  "version": "1.0.0",
  "description": "",
  "main": "src/loader.js",
  "scripts": {
    "dev": "nodemon",
    "production": "pm2 start src/loader.js --name my-money-backend"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.15.2",
    "express": "^4.14.0",
    "express-query-int": "^1.0.1",
    "lodash": "^4.17.4",
    "mongoose": "^4.7.0",
    "mongoose-paginate": "^5.0.3",
    "node-restful": "^0.2.5",
    "pm2": "^2.1.5"
  },
  "devDependencies": {
    "nodemon": "^1.11.0"
  }
}
```

## Listagem 222 - Criar arquivo `.gitignore`

`my-money-app/backend/.gitignore`

```
node_modules
*.log
```

## 8.3. Configurando o Servidor com o Express



Criar uma pasta chamada `src` em `backend/src`

### Listagem 223 - Criar arquivo loader.js

my-money-app/backend/src/loader.js

```
require('./config/server')
```



Criar uma pasta chamada **config** em **src/config**

### Listagem 224 - Criar arquivo server.js

my-money-app/backend/src/config/server.js

```
const port = 3003

const bodyParser = require('body-parser')
const express = require('express')
const server = express()

server.use(bodyParser.urlencoded({ extended: true }))
server.use(bodyParser.json())

server.listen(port, function() {
  console.log('BACKEND is running on port ${port}.')
})
```

### Listagem 225 - Executar o backend da aplicação

Abra o terminal e dentro da pasta my-money-app/backend

```
npm run dev
```

## 8.4. Conexão com o Banco de Dados

### Listagem 226 - Criar arquivo database.js

my-money-app/backend/src/config/database.js

```
const mongoose = require('mongoose')
mongoose.Promise = global.Promise
module.exports = mongoose.connect('mongodb://localhost/mymoney')
```

### Listagem 227 - Alterar arquivo loader.js

my-money-app/backend/src/loader.js (aprox. linha 2)

```
require('./config/server')  
require('./config/database')
```

### Listagem 228 - Iniciar MongoDB

Abra um novo terminal

```
mongod
```

## 8.5. Mapeamento Objeto-Documento (ODM)



Criar uma pasta chamada **api** e dentro dela criar uma pasta chamada **billingCycle** em **src/api/billingCycle**

### Listagem 229 - Criar arquivo *billingCycle.js*

my-money-app/backend/src/api/billingCycle/billingCycle.js

```
const restful = require('node-restful')
const mongoose = restful.mongoose

const creditSchema = new mongoose.Schema({
  name: { type: String, required: true },
  value: { type: Number, min: 0, required: true }
})

const debtSchema = new mongoose.Schema({
  name: { type: String, required: true },
  value: { type: Number, min: 0, required: true },
  status: { type: String, required: false, uppercase: true,
    enum: ['PAGO', 'PENDENTE', 'AGENDADO'] }
})

const billingCycleSchema = new mongoose.Schema({
  name: { type: String, required: true },
  month: { type: Number, min: 1, max: 12, required: true },
  year: { type: Number, min: 1970, max: 2100, required: true },
  credits: [creditSchema],
  debts: [debtSchema]
})

module.exports = restful.model('BillingCycle', billingCycleSchema)
```

## 8.6. Criação da API REST

### Listagem 230 - Criar arquivo *billingCycleService.js*

my-money-app/backend/src/api/billingCycle/billingCycleService.js

```
const BillingCycle = require('./billingCycle')

BillingCycle.methods(['get', 'post', 'put', 'delete'])
BillingCycle.updateOptions({new: true, runValidators: true})

module.exports = BillingCycle
```

## 8.7. Mapeamento das Rotas

### Listagem 231 - Criar arquivo routes.js

my-money-app/backend/src/config/routes.js

```
const express = require('express')

module.exports = function(server) {

  // Definir URL base para todas as rotas
  const router = express.Router()
  server.use('/api', router)

  // Rotas de Ciclo de Pagamento
  const BillingCycle = require('../api/billingCycle/billingCycleService')
  BillingCycle.register(router, '/billingCycles')
}
```

### Listagem 232 - Alterar arquivo loader.js

my-money-app/backend/src/loader.js (aprox. linha 1 e 3)

```
const server = require('./config/server')
require('./config/database')
require('./config/routes')(server)
```

### Listagem 233 - Alterar arquivo server.js

my-money-app/backend/src/config/server.js (aprox. linha 14)

```
const port = 3003

const bodyParser = require('body-parser')
const express = require('express')
const server = express()

server.use(bodyParser.urlencoded({ extended: true }))
server.use(bodyParser.json())

server.listen(port, function() {
  console.log('BACKEND is running on port ${port}.')
})

module.exports = server
```

## 8.8. Testando a API (Postman)

### Listagem 234 - Alterar arquivo *database.js*

my-money-app/backend/src/config/database.js

```
const mongoose = require('mongoose')
mongoose.Promise = global.Promise
module.exports = mongoose.connect('mongodb://localhost/mymoney')

mongoose.Error.messages.general.required = "O atributo '{PATH}' é obrigatório."
mongoose.Error.messages.Number.min =
  "O '{VALUE}' informado é menor que o limite mínimo de '{MIN}'."
mongoose.Error.messages.Number.max =
  "O '{VALUE}' informado é maior que o limite máximo de '{MAX}'."
mongoose.Error.messages.String.enum =
  "'{VALUE}' não é válido para o atributo '{PATH}'."
```

## Listagem 235 - Alterar arquivo *billingCycle.js*

my-money-app/backend/src/api/billingCycle/billingCycle.js (aprox. linha 11)

```
const restful = require('node-restful')
const mongoose = restful.mongoose

const creditSchema = new mongoose.Schema({
  name: { type: String, required: true },
  value: { type: Number, min: 0, required: true }
})

const debtSchema = new mongoose.Schema({
  name: { type: String, required: true },
  value: { type: Number, min: 0, required: [true, 'Informe o valor do débito!']
},
  status: { type: String, required: false, uppercase: true,
    enum: ['PAGO', 'PENDENTE', 'AGENDADO'] }
})

const billingCycleSchema = new mongoose.Schema({
  name: { type: String, required: true },
  month: { type: Number, min: 1, max: 12, required: true },
  year: { type: Number, min: 1970, max: 2100, required: true },
  credits: [creditSchema],
  debts: [debtSchema]
})

module.exports = restful.model('BillingCycle', billingCycleSchema)
```

## 8.9. Obtendo a Qtde de Registros (Count)



## Listagem 236 - Alterar arquivo *billingCycleService.js*

my-money-app/backend/src/api/billingCycle/billingCycleService.js (aprox. linha 6)

```
const BillingCycle = require('./billingCycle')

BillingCycle.methods(['get', 'post', 'put', 'delete'])
BillingCycle.updateOptions({new: true, runValidators: true})

BillingCycle.route('count', (req, res, next) => {
  BillingCycle.count((error, value) => {
    if(error) {
      res.status(500).json({errors: [error]})
    } else {
      res.json({value})
    }
  })
})

module.exports = BillingCycle
```

## 8.10. Obtendo o Sumário de Pagamentos

## Listagem 237 - Alterar arquivo *billingCycleService.js*

my-money-app/backend/src/api/billingCycle/billingCycleService.js (aprox. linha 16)

```
const BillingCycle = require('./billingCycle')

BillingCycle.methods(['get', 'post', 'put', 'delete'])
BillingCycle.updateOptions({new: true, runValidators: true})

BillingCycle.route('count', (req, res, next) => {
  BillingCycle.count((error, value) => {
    if(error) {
      res.status(500).json({errors: [error]})
    } else {
      res.json({value})
    }
  })
})

BillingCycle.route('summary', (req, res, next) => {
  BillingCycle.aggregate({
    $project: {credit: {$sum: "$credits.value"}, debt: {$sum:
"$debts.value"}}
  }, {
    $group: {_id: null, credit: {$sum: "$credit"}, debt: {$sum: "$debt"}}
  }, {
    $project: {_id: 0, credit: 1, debt: 1}
  }, (error, result) => {
    if(error) {
      res.status(500).json({errors: [error]})
    } else {
      res.json(result[0] || { credit: 0, debt: 0 })
    }
  })
})

module.exports = BillingCycle
```

## 8.11. Tratando as Mensagens de Erro



Criar uma pasta chamada **common** em **src/api/common**

## Listagem 238 - Criar arquivo errorHandler.js

my-money-app/backend/src/api/common/errorHandler.js

```
const _ = require('lodash')

module.exports = (req, res, next) => {
  const bundle = res.locals.bundle

  if(bundle.errors) {
    const errors = parseErrors(bundle.errors)
    res.status(500).json({errors})
  } else {
    next()
  }
}

const parseErrors = (nodeRestfulErrors) => {
  const errors = []
  _.forIn(nodeRestfulErrors, error => errors.push(error.message))
  return errors
}
```

## Listagem 239 - Alterar arquivo *billingCycleService.js*

my-money-app/backend/src/api/billingCycle/billingCycleService.js (aprox. linha 2 e 6)

```
const BillingCycle = require('./billingCycle')
const errorHandler = require('../common/errorHandler')

BillingCycle.methods(['get', 'post', 'put', 'delete'])
BillingCycle.updateOptions({new: true, runValidators: true})
BillingCycle.after('post', errorHandler).after('put', errorHandler)

BillingCycle.route('count', (req, res, next) => {
  BillingCycle.count((error, value) => {
    if(error) {
      res.status(500).json({errors: [error]})
    } else {
      res.json({value})
    }
  })
})

BillingCycle.route('summary', (req, res, next) => {
  BillingCycle.aggregate({
    $project: {credit: {$sum: "$credits.value"}, debt: {$sum:
"$debts.value"}}
  }, {
    $group: {_id: null, credit: {$sum: "$credit"}, debt: {$sum: "$debt"}}
  }, {
    $project: {_id: 0, credit: 1, debt: 1}
  }, (error, result) => {
    if(error) {
      res.status(500).json({errors: [error]})
    } else {
      res.json(result[0] || { credit: 0, debt: 0 })
    }
  })
})

module.exports = BillingCycle
```

## 8.12. Habilitando o CORS

### Listagem 240 - Criar arquivo cors.js

my-money-app/backend/src/config/cors.js

```
module.exports = (req, res, next) => {
  res.header('Access-Control-Allow-Origin', '*')
  res.header('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT, PATCH, DELETE')
  res.header('Access-Control-Allow-Headers', 'Origin, X-Requested-With, Content-Type, Accept')
  next()
}
```

### Listagem 241 - Alterar arquivo server.js

my-money-app/backend/src/config/server.js (aprox. linha 6 e 10)

```
const port = 3003

const bodyParser = require('body-parser')
const express = require('express')
const server = express()
const allowCors = require('./cors')

server.use(bodyParser.urlencoded({ extended: true }))
server.use(bodyParser.json())
server.use(allowCors)

server.listen(port, function() {
  console.log(`BACKEND is running on port ${port}.`)
})

module.exports = server
```

## 8.13. Middleware Query Parser Int

## Listagem 242 - Alterar arquivo server.js

my-money-app/backend/src/config/server.js (aprox. linha 7 e 12)

```
const port = 3003

const bodyParser = require('body-parser')
const express = require('express')
const server = express()
const allowCors = require('./cors')
const queryParser = require('express-query-int')

server.use(bodyParser.urlencoded({ extended: true }))
server.use(bodyParser.json())
server.use(allowCors)
server.use(queryParser())

server.listen(port, function() {
  console.log(`BACKEND is running on port ${port}.`)
})

module.exports = server
```

## 9. Aplicação Final - Ciclos de Pagamentos (Frontend)

### 9.1. Visão Geral

### 9.2. Configuração e Instalação

Abra o terminal e dentro da pasta `my-money-app/backend` execute o seguinte comando:

#### Listagem 243 - Executar o backend da aplicação

`my-money-app/backend`

```
npm run production
```

#### Listagem 244 - Executar o monit do PM2

`my-money-app/backend`

```
./node_modules/.bin/pm2 monit
```



Abra um novo terminal e dentro da pasta `my-money-app` execute o seguinte comando:

#### Listagem 245 - Criar pasta my-money-app

`my-money-app/frontend`

```
mkdir frontend && cd frontend
```

#### Listagem 246 - Criar arquivo package.json

`my-money-app/frontend/package.json`

```
npm init -y
```

### **Listagem 247 - Instalando as dependências do Webpack**

```
npm i --save-dev webpack@1.14.0 webpack-dev-server@1.16.2
```

### **Listagem 248 - Instalando as dependências do Babel**

```
npm i --save-dev babel-core@6.22.1 babel-loader@6.2.10 babel-plugin-react-html-  
attrs@2.0.0 babel-plugin-transform-object-rest-spread@6.22.0 babel-preset-  
es2015@6.22.0 babel-preset-react@6.22.0
```

### **Listagem 249 - Instalando as dependências do CSS e arquivos estáticos**

```
npm i --save-dev extract-text-webpack-plugin@1.0.1 css-loader@0.26.1 style-  
loader@0.13.1 file-loader@0.9.0
```

### **Listagem 250 - Instalando as dependências do template Admin-LTE e ícones**

```
npm i --save-dev admin-lte@2.3.6 font-awesome@4.7.0 ionicons@3.0.0
```

### **Listagem 251 - Instalando as dependências do React, Redux, Axios e Lodash**

```
npm i --save-dev react@15.4.2 react-dom@15.4.2 react-router@3.0.2 redux@3.6.0  
react-redux@4.4.6 redux-form@6.4.1 redux-multi@0.1.12 redux-promise@0.5.3 redux-  
thunk@2.1.0 react-redux-toastr@4.4.2 axios@0.15.3 lodash@4.17.4
```

### **Listagem 252 - Abrir o Visual Studio Code**

```
code .
```



### Listagem 253 - Criar arquivo .gitignore

my-money-app/frontend/.gitignore

```
node_modules  
*.log
```

## 9.3. Configurando o Build com Webpack

## Listagem 254 - Criar arquivo *webpack.config.js*

```

const webpack = require('webpack')
const ExtractTextPlugin = require('extract-text-webpack-plugin')

module.exports = {
  entry: './src/index.jsx',
  output: {
    path: __dirname + '/public',
    filename: './app.js'
  },
  devServer: {
    port: 8080,
    contentBase: './public',
  },
  resolve: {
    extensions: ['', '.js', '.jsx'],
    alias: {
      modules: __dirname + '/node_modules',
      jquery: 'modules/admin-lte/plugins/jquery/jquery-2.2.3.min.js',
      bootstrap: 'modules/admin-lte/bootstrap/js/bootstrap.js'
    }
  },
  plugins: [
    new webpack.ProvidePlugin({
      $: 'jquery',
      jQuery: 'jquery',
      'window.jQuery': 'jquery'
    }),
    new ExtractTextPlugin('app.css')
  ],
  module: {
    loaders: [{
      test: /\.js[x]?$/,
      loader: 'babel-loader',
      exclude: /node_modules/,
      query: {
        presets: ['es2015', 'react'],
        plugins: ['transform-object-rest-spread']
      }
    }, {
      test: /\.css$/,
      loader: ExtractTextPlugin.extract('style-loader', 'css-loader')
    }, {
      test: /\.woff|\.woff2|.ttf|.eot|.svg|.png|.jpg*.*/$,
      loader: 'file'
    }
  ]
}

```

## Listagem 255 - Alterar arquivo package.json

my-money-app/frontend/package.json (aprox. linha 2 e 7)

```
{
  "name": "my-money-frontend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "dev": "webpack-dev-server --progress --colors --inline --hot",
    "production": "webpack --progress -p"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "admin-lte": "^2.3.6",
    "axios": "^0.15.3",
    "babel-core": "^6.22.1",
    "babel-loader": "^6.2.10",
    "babel-plugin-react-html-attrs": "^2.0.0",
    "babel-plugin-transform-object-rest-spread": "^6.22.0",
    "babel-preset-es2015": "^6.22.0",
    "babel-preset-react": "^6.22.0",
    "css-loader": "^0.26.1",
    "extract-text-webpack-plugin": "^1.0.1",
    "file-loader": "^0.9.0",
    "font-awesome": "^4.7.0",
    "ionicons": "^3.0.0",
    "lodash": "^4.17.4",
    "react": "^15.4.2",
    "react-dom": "^15.4.2",
    "react-redux": "^4.4.6",
    "react-redux-toastr": "^4.4.2",
    "react-router": "^3.0.2",
    "redux": "^3.6.0",
    "redux-form": "^6.4.1",
    "redux-multi": "^0.1.12",
    "redux-promise": "^0.5.3",
    "redux-thunk": "^2.1.0",
    "style-loader": "^0.13.1",
    "webpack": "^1.14.0",
    "webpack-dev-server": "^1.16.2"
  }
}
```

## 9.4. Criação do index.html



Criar uma pasta chamada `public` em `frontend/public`

### Listagem 256 - Criar arquivo `index.html`

`my-money-app/frontend/public/index.html`

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset='utf-8'>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <title>My Money</title>
    <link rel='stylesheet' href='app.css'>
  </head>
  <body class='skin-blue fixed sidebar-mini'>
    <div id="app"></div>
    <script src='app.js'></script>
  </body>
</html>
```

## 9.5. Componentes Index e App



Criar uma pasta chamada `src` e dentro dela criar uma pasta chamada `main` em `frontend/src/main`

### Listagem 257 - Criar arquivo `app.jsx`

`my-money-app/frontend/src/main/app.jsx`

```
import React from 'react'

export default props => (
  <div className='wrapper'>
    <h1>App</h1>
  </div>
)
```

### Listagem 258 - Criar arquivo index.jsx

my-money-app/frontend/src/index.jsx

```
import React from 'react'
import ReactDOM from 'react-dom'
import App from './main/app'

ReactDOM.render(<App />, document.getElementById('app'))
```

### Listagem 259 - Executar a aplicação

Abra o terminal e dentro da pasta my-money-app/frontend

```
npm run dev
```



Verificar o funcionamento: abra o navegador e digite <http://localhost:8080> e na página vai mostrar o título **App**.

## 9.6. Dependências do Template (AdminLTE)



Criar uma pasta chamada **common** e dentro dela criar uma pasta chamada **template** em **src/common/template**

### Listagem 260 - Criar arquivo dependencies.js

my-money-app/frontend/src/common/template/dependencies.js

```
import 'modules/admin-lte/plugins/jqueryUI/jquery-ui.min'
import 'modules/admin-lte/plugins/fastclick/fastclick'
import 'modules/admin-lte/plugins/slimScroll/jquery.slimscroll.min'
import 'modules/admin-lte/dist/js/app.min'

import 'modules/font-awesome/css/font-awesome.min.css'
import 'modules/ionicons/dist/css/ionicons.min.css'
import 'modules/admin-lte/bootstrap/css/bootstrap.min.css'
import 'modules/admin-lte/dist/css/AdminLTE.min.css'
import 'modules/admin-lte/dist/css/skins/_all-skins.min.css'
import 'modules/admin-lte/plugins/iCheck/flat/blue.css'

import './custom.css'
```

## Listagem 261 - Alterar arquivo app.jsx

my-money-app/frontend/src/main/app.jsx (na linha 1)

```
import '../common/template/dependencies'
import React from 'react'

export default props => (
  <div className='wrapper'>
    <h1>App</h1>
  </div>
)
```

## 9.7. Componente Cabeçalho

### Listagem 262 - Criar arquivo header.jsx

my-money-app/frontend/src/common/template/header.jsx

```
import React from 'react'

export default props => (
  <header className='main-header'>
    <a href='/#/' className='logo'>
      <span className='logo-mini'><b>My</b>M</span>
      <span className='logo-lg'>
        <i className='fa fa-money'></i>
        <b> My</b> Money
      </span>
    </a>
    <nav className='navbar navbar-static-top'>
      <a href className='sidebar-toggle' data-toggle='offcanvas'></a>
    </nav>
  </header>
)
```

### Listagem 263 - Alterar arquivo app.jsx

my-money-app/frontend/src/main/app.jsx (aprox. linha 4 e 8)

```
import '../common/template/dependencies'
import React from 'react'

import Header from '../common/template/header'

export default props => (
  <div className='wrapper'>
    <Header />
  </div>
)
```

## 9.8. Componente Menu (Parte 1)

### Listagem 264 - Criar arquivo menu.jsx

my-money-app/frontend/src/common/template/menu.jsx

```
import React from 'react'

export default props => (
  <ul className='sidebar-menu'>

  </ul>
)
```

### Listagem 265 - Criar arquivo sidebar.jsx

my-money-app/frontend/src/common/template/sidebar.jsx

```
import React from 'react'
import Menu from './menu'

export default props => (
  <aside className='main-sidebar'>
    <section className='sidebar'>
      <Menu />
    </section>
  </aside>
)
```



### Listagem 266 - Alterar arquivo app.jsx

my-money-app/frontend/src/main/app.jsx (aprox. linha 5 e 10)

```
import '../common/template/dependencies'
import React from 'react'

import Header from '../common/template/header'
import SideBar from '../common/template/sideBar'

export default props => (
  <div className='wrapper'>
    <Header />
    <SideBar />
  </div>
)
```

## 9.9. Componente Menu (Parte 2)

### Listagem 267 - Criar arquivo menuItem.jsx

my-money-app/frontend/src/common/template/menuItem.jsx

```
import React from 'react'

export default props => (
  <li>
    <a href={props.path}>
      <i className={`fa fa-${props.icon}`}></i> {props.label}
    </a>
  </li>
)
```

## Listagem 268 - Alterar arquivo menu.jsx

my-money-app/frontend/src/common/template/menu.jsx (aprox. linha 2 e 6)

```
import React from 'react'
import MenuItem from './menuItem'

export default props => (
  <ul className='sidebar-menu'>
    <MenuItem path='#/' label='Dashboard' icon='dashboard' />
  </ul>
)
```

## Listagem 269 - Criar arquivo menuTree.jsx

my-money-app/frontend/src/common/template/menuTree.jsx

```
import React from 'react'

export default props => (
  <li className='treeview'>
    <a href>
      <i className={`fa fa-${props.icon}`}></i> {props.label}
      <i className='fa fa-angle-left pull-right'></i>
    </a>
    <ul className='treeview-menu'>
      {props.children}
    </ul>
  </li>
)
```

## Listagem 270 - Alterar arquivo menu.jsx

my-money-app/frontend/src/common/template/menu.jsx (aprox. linha 3, 7 e 8)

```
import React from 'react'
import MenuItem from './menuItem'
import MenuTree from './menuTree'

export default props => (
  <ul className='sidebar-menu'>
    <MenuItem path='#' label='Dashboard' icon='dashboard' />
    <MenuTree label='Cadastro' icon='edit'>
      <MenuItem path='#billingCycles'
        label='Ciclos de Pagamentos' icon='usd' />
    </MenuTree>
  </ul>
)
```

## 9.10. Componente Rodapé

### Listagem 271 - Criar arquivo footer.jsx

my-money-app/frontend/src/common/template/footer.jsx

```
import React from 'react'

export default props => (
  <footer className='main-footer'>
    <strong>
      Copyright &copy; 2017
      <a href='http://cod3r.com.br' target='_blank'> Cod3r</a>.
    </strong>
  </footer>
)
```

### Listagem 272 - Alterar arquivo app.jsx

my-money-app/frontend/src/main/app.jsx (aprox. linha 6 e 12)

```
import '../common/template/dependencies'
import React from 'react'

import Header from '../common/template/header'
import SideBar from '../common/template/sideBar'
import Footer from '../common/template/footer'

export default props => (
  <div className='wrapper'>
    <Header />
    <SideBar />
    <div className='content-wrapper'>
      <h1>Conteudo</h1>
    </div>
    <Footer />
  </div>
)
```

### Listagem 273 - Alterar arquivo custom.css

my-money-app/frontend/src/common/template/custom.css (na linha 1)

```
.main-footer {
  position: fixed;
  bottom: 0px;
  width: 100%;
}
```

## 9.11. Navegação com React-router



Criar uma pasta chamada **dashboard** em **src/dashboard**

## Listagem 274 - Criar arquivo dashboard.jsx

my-money-app/frontend/src/dashboard/dashboard.jsx

```
import React from 'react'

export default props => (
  <div>
    <h1>Dashboard</h1>
  </div>
)
```



Criar uma pasta chamada `billingCycle` em `src/billingCycle`

## Listagem 275 - Criar arquivo billingCycle.jsx

my-money-app/frontend/src/billingCycle/billingCycle.jsx

```
import React from 'react'

export default props => (
  <div>
    <h1>Ciclo de Pagamento</h1>
  </div>
)
```

## Listagem 276 - Criar arquivo routes.jsx

my-money-app/frontend/src/main/routes.jsx

```
import React from 'react'
import { Router, Route, Redirect, hashHistory } from 'react-router'

import Dashboard from '../dashboard/dashboard'
import BillingCycle from '../billingCycle/billingCycle'

export default props => (
  <Router history={hashHistory}>
    <Route path='/' component={Dashboard} />
    <Route path='/billingCycles' component={BillingCycle} />
    <Redirect from='*' to='/' />
  </Router>
)
```

### Listagem 277 - Alterar arquivo app.jsx

my-money-app/frontend/src/main/app.jsx (aprox. linha 7 e 14)

```
import '../common/template/dependencies'
import React from 'react'

import Header from '../common/template/header'
import SideBar from '../common/template/sideBar'
import Footer from '../common/template/footer'
import Routes from './routes'

export default props => (
  <div className='wrapper'>
    <Header />
    <SideBar />
    <div className='content-wrapper'>
      <Routes />
    </div>
    <Footer />
  </div>
)
```

## 9.12. Componente de Título do Conteúdo

### Listagem 278 - Criar arquivo contentHeader.jsx

my-money-app/frontend/src/common/template/contentHeader.jsx

```
import React from 'react'

export default props => (
  <section className='content-header'>
    <h1>{props.title} <small>{props.small}</small></h1>
  </section>
)
```

### Listagem 279 - Criar arquivo `content.jsx`

`my-money-app/frontend/src/common/template/content.jsx`

```
import React from 'react'

export default props => (
  <section className='content'>{props.children}</section>
)
```

### Listagem 280 - Alterar arquivo `dashboard.jsx`

`my-money-app/frontend/src/dashboard/dashboard.jsx` (aprox. linha 1, 3 e 6)

```
import React, { Component } from 'react'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'

class Dashboard extends Component {
  render() {
    return (
      <div>
        <ContentHeader title='Dashboard' small='Versão 1.0' />
        <Content>
          Dashboard
        </Content>
      </div>
    )
  }
}

export default Dashboard
```

## 9.13. Componente Dashboard: Visão Geral

## 9.14. Componente Grid



Criar uma pasta chamada `layout` em `src/common/layout`

## Listagem 281 - Criar arquivo grid.jsx

my-money-app/frontend/src/common/layout/grid.jsx

```
import React, { Component } from 'react'

export default class Grid extends Component {

  toCssClasses(numbers) {
    const cols = numbers ? numbers.split(' ') : []
    let classes = ''

    if(cols[0]) classes += `col-xs-${cols[0]} `
    if(cols[1]) classes += `col-sm-${cols[1]} `
    if(cols[2]) classes += `col-md-${cols[2]} `
    if(cols[3]) classes += `col-lg-${cols[3]} `

    return classes
  }

  render() {
    const gridClasses = this.toCssClasses(this.props.cols || '12')
    return (
      <div className={gridClasses}>
        {this.props.children}
      </div>
    )
  }
}
```

## 9.15. Componente ValueBox



Criar uma pasta chamada `widget` em `src/common/widget`



## Listagem 282 - Criar arquivo valueBox.jsx

my-money-app/frontend/src/common/widget/valueBox.jsx

```
import React from 'react'
import Grid from '../layout/grid'

export default props => (
  <Grid cols={props.cols}>
    <div className={`small-box bg-${props.color}`>
      <div className='inner'>
        <h3>{props.value}</h3>
        <p>{props.text}</p>
      </div>
      <div className='icon'>
        <i className={`fa fa-${props.icon}`}></i>
      </div>
    </div>
  </Grid>
)
```

## 9.16. Usando ValueBox

## Listagem 283 - Alterar arquivo dashboard.jsx

my-money-app/frontend/src/dashboard/dashboard.jsx (aprox. linha 5 e 13)

```
import React, { Component } from 'react'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import ValueBox from '../common/widget/valueBox'
import Row from '../common/layout/row'

class Dashboard extends Component {
  render() {
    return (
      <div>
        <ContentHeader title='Dashboard' small='Versão 1.0' />
        <Content>
          <Row>
            <ValueBox cols='12 4' color='green' icon='bank'
              value='R$ 10' text='Total de Créditos' />
            <ValueBox cols='12 4' color='red' icon='credit-card'
              value='R$ 10' text='Total de Débitos' />
            <ValueBox cols='12 4' color='blue' icon='money'
              value='R$ 0' text='Valor Consolidado' />
          </Row>
        </Content>
      </div>
    )
  }
}

export default Dashboard
```

## Listagem 284 - Criar arquivo row.jsx

my-money-app/frontend/src/common/layout/row.jsx

```
import React from 'react'

export default props => (
  <div className='row'>{props.children}</div>
)
```

## 9.17. Configurando Redux

### Listagem 285 - Criar arquivo reducers.js

my-money-app/frontend/src/main/reducers.js

```
import { combineReducers } from 'redux'

const rootReducer = combineReducers({
  dashboard: () => ({summary: {credit: 100, debt: 50}})
})

export default rootReducer
```

### Listagem 286 - Alterar arquivo index.jsx

my-money-app/frontend/src/index.jsx (aprox. linha 3, 4, 7, 9 e 11 )

```
import React from 'react'
import ReactDOM from 'react-dom'
import { createStore } from 'redux'
import { Provider } from 'react-redux'

import App from './main/app'
import reducers from './main/reducers'

const store = createStore(reducers)
ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>
, document.getElementById('app'))
```

## 9.18. Integrando Dashboard com Redux (Parte 1)

## Listagem 287 - Alterar arquivo dashboard.jsx

my-money-app/frontend/src/dashboard/dashboard.jsx (aprox. linha 2, 11, 18, 20, 22, 29 e 30)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import ValueBox from '../common/widget/valueBox'
import Row from '../common/layout/row'

class Dashboard extends Component {
  render() {
    const { credit, debt } = this.props.summary
    return (
      <div>
        <ContentHeader title='Dashboard' small='Versão 1.0' />
        <Content>
          <Row>
            <ValueBox cols='12 4' color='green' icon='bank'
              value={`R$ ${credit}`} text='Total de Créditos' />
            <ValueBox cols='12 4' color='red' icon='credit-card'
              value={`R$ ${debt}`} text='Total de Débitos' />
            <ValueBox cols='12 4' color='blue' icon='money'
              value={`R$ ${credit - debt}`} text='Valor
Consolidado' />
          </Row>
        </Content>
      </div>
    )
  }
}

const mapStateToProps = state => ({summary: state.dashboard.summary})
export default connect(mapStateToProps)(Dashboard)
```

## Listagem 288 - Criar arquivo dashboardReducer.js

my-money-app/frontend/src/dashboard/dashboardReducer.js

```
const INITIAL_STATE = {summary: {credit: 0, debt: 0}}

export default function(state = INITIAL_STATE, action) {
  return state
}
```

### Listagem 289 - Alterar arquivo reducers.js

my-money-app/frontend/src/main/reducers.js (aprox. linha 3 e 6)

```
import { combineReducers } from 'redux'

import DashboardReducer from '../dashboard/dashboardReducer'

const rootReducer = combineReducers({
  dashboard: DashboardReducer
})

export default rootReducer
```

## 9.19. Integrando Dashboard com Redux (Parte 2)

### Listagem 290 - Criar arquivo dashboardActions.js

my-money-app/frontend/src/dashboard/dashboardActions.js

```
import axios from 'axios'
const BASE_URL = 'http://localhost:3003/api'

export function getSummary() {
  const request = axios.get(`${BASE_URL}/billingCycles/summary`)
  return {
    type: 'BILLING_SUMMARY_FETCHED',
    payload: request
  }
}
```

## Listagem 291 - Alterar arquivo *dashboardReducer.js*

my-money-app/frontend/src/dashboard/dashboardReducer.js (aprox. linha 4)

```
const INITIAL_STATE = {summary: {credit: 0, debt: 0}}

export default function(state = INITIAL_STATE, action) {
  switch (action.type) {
    case 'BILLING_SUMMARY_FETCHED':
      return { ...state, summary: action.payload.data }
    default:
      return state
  }
}
```

## Listagem 292 - Alterar arquivo dashboard.jsx

my-money-app/frontend/src/dashboard/dashboard.jsx (aprox. linha 3, 5, 13, 33 e 34)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'

import { getSummary } from '../dashboardActions'
import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import ValueBox from '../common/widget/valueBox'
import Row from '../common/layout/row'

class Dashboard extends Component {

  componentWillMount() {
    this.props.getSummary()
  }

  render() {
    const { credit, debt } = this.props.summary
    return (
      <div>
        <ContentHeader title='Dashboard' small='Versão 1.0' />
        <Content>
          <Row>
            <ValueBox cols='12 4' color='green' icon='bank'
              value={`R$ ${credit}`} text='Total de Créditos' />
            <ValueBox cols='12 4' color='red' icon='credit-card'
              value={`R$ ${debt}`} text='Total de Débitos' />
            <ValueBox cols='12 4' color='blue' icon='money'
              value={`R$ ${credit - debt}`} text='Valor
Consolidado' />
          </Row>
        </Content>
      </div>
    )
  }
}

const mapStateToProps = state => ({summary: state.dashboard.summary})
const mapDispatchToProps = dispatch => bindActionCreators({getSummary}, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(Dashboard)
```

## Listagem 293 - Alterar arquivo *index.jsx*

my-money-app/frontend/src/index.jsx (aprox. linha 3, 6 e 11 )

```
import React from 'react'
import ReactDOM from 'react-dom'
import { applyMiddleware, createStore } from 'redux'
import { Provider } from 'react-redux'

import promise from 'redux-promise'

import App from './main/app'
import reducers from './main/reducers'

const store = applyMiddleware(promise)(createStore)(reducers)
ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>
, document.getElementById('app'))
```

## 9.20. Ciclo React-Redux com Middlewares

## 9.21. Configurando Redux DevTools



## Listagem 294 - Alterar arquivo *index.jsx*

my-money-app/frontend/src/index.jsx (aprox. linha 11 e 13 )

```
import React from 'react'
import ReactDOM from 'react-dom'
import { applyMiddleware, createStore } from 'redux'
import { Provider } from 'react-redux'

import promise from 'redux-promise'

import App from './main/app'
import reducers from './main/reducers'

const devTools = window.__REDUX_DEVTOOLS_EXTENSION__
  && window.__REDUX_DEVTOOLS_EXTENSION__()
const store = applyMiddleware(promise)(createStore)(reducers, devTools)
ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>
, document.getElementById('app'))
```

## 9.22. Dashboard SEM Redux



Criar uma pasta chamada **dashboard2** em **src/dashboard2**

## Listagem 295 - Criar arquivo dashboard2.jsx

my-money-app/frontend/src/dashboard2/dashboard2.jsx

```
import React, { Component } from 'react'
import axios from 'axios'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import ValueBox from '../common/widget/valueBox'
import Row from '../common/layout/row'

const BASE_URL = 'http://localhost:3003/api'

export default class Dashboard2 extends Component {

  constructor(props) {
    super(props)
    this.state = { credit: 0, debt: 0 }
  }

  componentWillMount() {
    axios.get(`${BASE_URL}/billingCycles/summary`)
      .then(resp => this.setState(resp.data))
  }

  render() {
    const { credit, debt } = this.state
    return (
      <div>
        <ContentHeader title='Dashboard' small='Versão 2.0' />
        <Content>
          <Row>
            <ValueBox cols='12 4' color='green' icon='bank'
              value={`R$ ${credit}`} text='Total de Créditos' />
            <ValueBox cols='12 4' color='red' icon='credit-card'
              value={`R$ ${debt}`} text='Total de Débitos' />
            <ValueBox cols='12 4' color='blue' icon='money'
              value={`R$ ${credit - debt}`} text='Valor
Consolidado' />
          </Row>
        </Content>
      </div>
    )
  }
}
```

## 9.23. Visão Geral dos Componentes de Abas

## 9.24. Componente BillingCycle: Refatorando para Classe

### Listagem 296 - Sobrescrever arquivo *billingCycle.jsx*

my-money-app/frontend/src/billingCycle/billingCycle.jsx

```
import React, { Component } from 'react'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'

class BillingCycle extends Component {
  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          Ciclos de Pagamentos
        </Content>
      </div>
    )
  }
}

export default BillingCycle
```

## 9.25. Componentes Estruturais das Abas



Criar uma pasta chamada **tab** em **src/common/tab**

### Listagem 297 - Criar arquivo *tab.jsx*

my-money-app/frontend/src/common/tab/tab.jsx

```
import React from 'react'

export default props => (
  <div className='nav-tabs-custom'>
    {props.children}
  </div>
)
```

## Listagem 298 - Criar arquivo *tabsHeader.jsx*

my-money-app/frontend/src/common/tab/tabsHeader.jsx

```
import React from 'react'

export default props => (
  <ul className='nav nav-tabs'>
    {props.children}
  </ul>
)
```

## Listagem 299 - Criar arquivo *tabsContent.jsx*

my-money-app/frontend/src/common/tab/tabsContent.jsx

```
import React from 'react'

export default props => (
  <div className='tab-content'>
    {props.children}
  </div>
)
```

## Listagem 300 - Alterar arquivo *billingCycle.jsx*

my-money-app/frontend/src/billingCycle/billingCycle.jsx (aprox. linha 5, 6, 7 e 15)

```
import React, { Component } from 'react'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'

class BillingCycle extends Component {
  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>

            </TabsHeader>
            <TabsContent>

            </TabsContent>
          </Tabs>
        </Content>
      </div>
    )
  }
}

export default BillingCycle
```

## 9.26. Componente TabHeader

## Listagem 301 - Criar arquivo *tabHeader.jsx*

my-money-app/frontend/src/common/tab/tabHeader.jsx

```
import React, { Component } from 'react'

class TabHeader extends Component {
  render() {
    return (
      <li>
        <a href
          data-toggle='tab'
          data-target={this.props.target}>
          <i className={`fa fa-${this.props.icon}`}></i>
        {this.props.label}
        </a>
      </li>
    )
  }
}

export default TabHeader
```

## Listagem 302 - Alterar arquivo *billingCycle.jsx*

my-money-app/frontend/src/billingCycle/billingCycle.jsx (aprox. linha 8)

```
import React, { Component } from 'react'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'
import TabHeader from '../common/tab/tabHeader'

class BillingCycle extends Component {
  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>
              <TabHeader label='Listar' icon='bars'
target='tabList' />
              <TabHeader label='Incluir' icon='plus'
target='tabCreate' />
              <TabHeader label='Alterar' icon='pencil'
target='tabUpdate' />
              <TabHeader label='Excluir' icon='trash-o'
target='tabDelete' />
            </TabsHeader>
            <TabsContent>

              </TabsContent>
            </Tabs>
          </Content>
        </div>
      )
    }
  }

  export default BillingCycle
```

### Listagem 303 - Alterar arquivo *tabHeader.jsx*

my-money-app/frontend/src/common/tab/tabHeader.jsx (aprox. linha 7)

```
import React, { Component } from 'react'

class TabHeader extends Component {
  render() {
    return (
      <li>
        <a href='javascript:;'
          data-toggle='tab'
          data-target={this.props.target}>
          <i className={`fa fa-${this.props.icon}`}></i>
        {this.props.label}
        </a>
      </li>
    )
  }
}

export default TabHeader
```

## 9.27. Criando a Primeira Action e o Reducer

### Listagem 304 - Criar arquivo *tabActions.js*

my-money-app/frontend/src/common/tab/tabActions.js

```
export function selectTab(tabId) {
  console.log(tabId)
  return {
    type: 'TAB_SELECTED',
    payload: tabId
  }
}
```



### Listagem 305 - Criar arquivo *tabReducer.js*

my-money-app/frontend/src/common/tab/tabReducer.js

```
const INITIAL_STATE = { selected: '' }

export default (state = INITIAL_STATE, action) => {
  switch (action.type) {
    case 'TAB_SELECTED':
      return { ...state, selected: action.payload }
    default:
      return state
  }
}
```

### Listagem 306 - Alterar arquivo *reducers.js*

my-money-app/frontend/src/main/reducers.js (aprox. linha 2, 3 e 8)

```
import { combineReducers } from 'redux'

import DashboardReducer from '../dashboard/dashboardReducer'
import TabReducer from '../common/tab/tabReducer'

const rootReducer = combineReducers({
  dashboard: DashboardReducer,
  tab: TabReducer
})

export default rootReducer
```

## 9.28. Componente TabHeader: Conectando com Redux

## Listagem 307 - Alterar arquivo `tabHeader.jsx`

`my-money-app/frontend/src/common/tab/tabHeader.jsx` (aprox. linha 2, 3, 5, 9, 11, 13, 21, 22 e 23)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

import { selectTab } from './tabActions'

class TabHeader extends Component {
  render() {
    const selected = this.props.tab.selected === this.props.target
    return (
      <li className={selected ? 'active' : ''}>
        <a href='javascript:;'
          data-toggle='tab'
          onClick={() => this.props.selectTab(this.props.target)}
          data-target={this.props.target}>
          <i className={`fa fa-${this.props.icon}`}></i>
{this.props.label}
        </a>
      </li>
    )
  }
}

const mapStateToProps = state => ({tab : state.tab})
const mapDispatchToProps = dispatch => bindActionCreators({selectTab}, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(TabHeader)
```

## Listagem 308 - Remover linha do arquivo `tabActions.js`

`my-money-app/frontend/src/common/tab/tabActions.js` (aprox. linha 2)

```
export function selectTab(tabId) {
  return {
    type: 'TAB_SELECTED',
    payload: tabId
  }
}
```

## 9.29. Componente Tab Content

### Listagem 309 - Criar arquivo `tabContent.jsx`

`my-money-app/frontend/src/common/tab/tabContent.jsx`

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

class TabContent extends Component {
  render() {
    const selected = this.props.tab.selected === this.props.id
    return (
      <div id={this.props.id}
        className={`tab-pane ${selected ? 'active' : ''}`}>
        {this.props.children}
      </div>
    )
  }
}

const mapStateToProps = state => ({tab: state.tab})
export default connect(mapStateToProps)(TabContent)
```

### **Listagem 310 - Alterar arquivo *billingCycle.jsx***

```

import React, { Component } from 'react'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'
import TabHeader from '../common/tab/tabHeader'
import TabContent from '../common/tab/tabContent'

class BillingCycle extends Component {
  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>
              <TabHeader label='Listar' icon='bars'
target='tabList' />
              <TabHeader label='Incluir' icon='plus'
target='tabCreate' />
              <TabHeader label='Alterar' icon='pencil'
target='tabUpdate' />
              <TabHeader label='Excluir' icon='trash-o'
target='tabDelete' />
            </TabsHeader>
            <TabsContent>
              <TabContent id='tabList'>
                <h1>Lista</h1>
              </TabContent>
              <TabContent id='tabCreate'>
                <h1>Incluir</h1>
              </TabContent>
              <TabContent id='tabUpdate'>
                <h1>Alterar</h1>
              </TabContent>
              <TabContent id='tabDelete'>
                <h1>Excluir</h1>
              </TabContent>
            </TabsContent>
          </Tabs>
        </Content>
      </div>
    )
  }
}

export default BillingCycle

```

## 9.30. Componente BillingCycle: Conectando com Redux

## Listagem 311 - Alterar arquivo *billingCycle.jsx*

my-money-app/frontend/src/billingCycle/billingCycle.jsx (aprox. linha 2, 3, 12, 16, 40 e 41)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'
import TabHeader from '../common/tab/tabHeader'
import TabContent from '../common/tab/tabContent'
import { selectTab } from '../common/tab/tabActions'

class BillingCycle extends Component {

  componentWillMount() {
    this.props.selectTab('tabList')
  }

  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>
              <TabHeader label='Listar' icon='bars'
target='tabList' />
              <TabHeader label='Incluir' icon='plus'
target='tabCreate' />
              <TabHeader label='Alterar' icon='pencil'
target='tabUpdate' />
              <TabHeader label='Excluir' icon='trash-o'
target='tabDelete' />
            </TabsHeader>
            <TabsContent>
              <TabContent id='tabList'>
                <h1>Lista</h1>
              </TabContent>
              <TabContent id='tabCreate'>
                <h1>Incluir</h1>
              </TabContent>
              <TabContent id='tabUpdate'>
                <h1>Alterar</h1>
              </TabContent>
              <TabContent id='tabDelete'>
```

```

        <h1>Excluir</h1>
      </TabContent>
    </TabsContent>
  </Tabs>
</Content>
</div>
)
}
}

const mapDispatchToProps = dispatch => bindActionCreators({selectTab}, dispatch)
export default connect(null, mapDispatchToProps)(BillingCycle)

```

## 9.31. Visibilidade das Abas (Parte 1)

### Listagem 312 - Alterar arquivo *tabActions.js*

my-money-app/frontend/src/common/tab/tabActions.js (aprox. linha 8)

```

export function selectTab(tabId) {
  return {
    type: 'TAB_SELECTED',
    payload: tabId
  }
}

export function showTabs(...tabIds) {
  const tabsToShow = {}
  tabIds.forEach(e => tabsToShow[e] = true)
  return {
    type: 'TAB_SHOWED',
    payload: tabsToShow
  }
}

```



### Listagem 313 - Alterar arquivo *tabReducer.js*

my-money-app/frontend/src/common/tab/tabReducer.js (aprox. linha 1 e 7)

```
const INITIAL_STATE = { selected: '', visible: {} }

export default (state = INITIAL_STATE, action) => {
  switch (action.type) {
    case 'TAB_SELECTED':
      return { ...state, selected: action.payload }
    case 'TAB_SHOWED':
      return { ...state, visible: action.payload }
    default:
      return state
  }
}
```

## Listagem 314 - Alterar arquivo *billingCycle.js*

my-money-app/frontend/src/billingCycle/billingCycle.js (aprox. linha 12, 18 e 45)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'
import TabHeader from '../common/tab/tabHeader'
import TabContent from '../common/tab/tabContent'
import { selectTab, showTabs } from '../common/tab/tabActions'

class BillingCycle extends Component {

  componentWillMount() {
    this.props.selectTab('tabList')
    this.props.showTabs('tabList', 'tabCreate')
  }

  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>
              <TabHeader label='Listar' icon='bars'
target='tabList' />
              <TabHeader label='Incluir' icon='plus'
target='tabCreate' />
              <TabHeader label='Alterar' icon='pencil'
target='tabUpdate' />
              <TabHeader label='Excluir' icon='trash-o'
target='tabDelete' />
            </TabsHeader>
            <TabsContent>
              <TabContent id='tabList'>
                <h1>Lista</h1>
              </TabContent>
              <TabContent id='tabCreate'>
                <h1>Incluir</h1>
              </TabContent>
              <TabContent id='tabUpdate'>
                <h1>Alterar</h1>
              </TabContent>
            </TabsContent>
          </Tabs>
        </Content>
      </div>
    )
  }
}
```

```

        <TabContent id='tabDelete'>
          <h1>Excluir</h1>
        </TabContent>
      </TabsContent>
    </Tabs>
  </Content>
</div>
)
}
}

const mapDispatchToProps = dispatch => bindActionCreators({selectTab, showTabs},
dispatch)
export default connect(null, mapDispatchToProps)(BillingCycle)

```

## 9.32. Visibilidade das Abas (Parte 2)



Criar uma pasta chamada `operator` em `src/common/operator`

### Listagem 315 - Criar arquivo `if.jsx`

`my-money-app/frontend/src/common/operator/if.jsx`

```

import React from 'react'

export default props => {
  if(props.test) {
    return props.children
  } else {
    return false
  }
}

```

## Listagem 316 - Alterar arquivo `tabHeader.jsx`

`my-money-app/frontend/src/common/tab/tabHeader.jsx` (aprox. linha 5, 11, 13 e 22)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

import If from '../operator/if'
import { selectTab } from './tabActions'

class TabHeader extends Component {
  render() {
    const selected = this.props.tab.selected === this.props.target
    const visible = this.props.tab.visible[this.props.target]
    return (
      <If test={visible}>
        <li className={selected ? 'active' : ''}>
          <a href='javascript:;'
            data-toggle='tab'
            onClick={() => this.props.selectTab(this.props.target)}
            data-target={this.props.target}>
            <i className={`fa fa-${this.props.icon}`}></i>
{this.props.label}
          </a>
        </li>
      </If>
    )
  }
}

const mapStateToProps = state => ({tab : state.tab})
const mapDispatchToProps = dispatch => bindActionCreators({selectTab}, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(TabHeader)
```

### Listagem 317 - Alterar arquivo *tabContent.jsx*

my-money-app/frontend/src/common/tab/tabContent.jsx (aprox. linha 4, 9, 11 e 16)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'
import If from '../operator/if'

class TabContent extends Component {
  render() {
    const selected = this.props.tab.selected === this.props.id
    const visible = this.props.tab.visible[this.props.id]
    return (
      <If test={visible}>
        <div id={this.props.id}
          className={`tab-pane ${selected ? 'active' : ''}`}>
          {this.props.children}
        </div>
      </If>
    )
  }
}

const mapStateToProps = state => ({tab: state.tab})
export default connect(mapStateToProps)(TabContent)
```

## 9.33. Componente de Lista: Visão Geral

## 9.34. Criando a Action "getList" e o Reducer

### Listagem 318 - Criar arquivo *billingCycleActions.js*

my-money-app/frontend/src/billingCycle/billingCycleActions.js

```
import axios from 'axios'
const BASE_URL = 'http://localhost:3003/api'

export function getList() {
  const request = axios.get(`${BASE_URL}/billingCycles`)
  return {
    type: 'BILLING_CYCLES_FETCHED',
    payload: request
  }
}
```

### Listagem 319 - Criar arquivo *billingCycleReducer.js*

my-money-app/frontend/src/billingCycle/billingCycleReducer.js

```
const INITIAL_STATE = {list: []}

export default (state = INITIAL_STATE, action) => {
  switch (action.type) {
    case 'BILLING_CYCLES_FETCHED':
      return { ...state, list: action.payload.data }
    default:
      return state
  }
}
```

### Listagem 320 - Alterar arquivo *reducers.js*

my-money-app/frontend/src/main/reducers.js (aprox. linha 5 e 10)

```
import { combineReducers } from 'redux'

import DashboardReducer from '../dashboard/dashboardReducer'
import TabReducer from '../common/tab/tabReducer'
import BillingCycleReducer from '../billingCycle/billingCycleReducer'

const rootReducer = combineReducers({
  dashboard: DashboardReducer,
  tab: TabReducer,
  billingCycle: BillingCycleReducer
})

export default rootReducer
```

## 9.35. Componente BillingCycleList

## Listagem 321 - Criar arquivo *billingCycleList.jsx*

my-money-app/frontend/src/billingCycle/billingCycleList.jsx

```
import React, { Component } from 'react'

class BillingCycleList extends Component {

  render() {
    return (
      <div>
        <table className='table'>
          <thead>
            <tr>
              <th>Nome</th>
              <th>Mês</th>
              <th>Ano</th>
            </tr>
          </thead>
          <tbody>

            </tbody>
          </table>
        </div>
      )
    }
  }

  export default BillingCycleList
```

## Listagem 322 - Alterar arquivo *billingCycle.js*

my-money-app/frontend/src/billingCycle/billingCycle.js (aprox. linha 13 e 36)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'
import TabHeader from '../common/tab/tabHeader'
import TabContent from '../common/tab/tabContent'
import { selectTab, showTabs } from '../common/tab/tabActions'
import List from './billingCycleList'

class BillingCycle extends Component {

  componentWillMount() {
    this.props.selectTab('tabList')
    this.props.showTabs('tabList', 'tabCreate')
  }

  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>
              <TabHeader label='Listar' icon='bars'
target='tabList' />
              <TabHeader label='Incluir' icon='plus'
target='tabCreate' />
              <TabHeader label='Alterar' icon='pencil'
target='tabUpdate' />
              <TabHeader label='Excluir' icon='trash-o'
target='tabDelete' />
            </TabsHeader>
            <TabsContent>
              <TabContent id='tabList'>
                <List />
              </TabContent>
              <TabContent id='tabCreate'>
                <h1>Incluir</h1>
              </TabContent>
              <TabContent id='tabUpdate'>
                <h1>Alterar</h1>
              </TabContent>
            </TabsContent>
          </Tabs>
        </Content>
      </div>
    )
  }
}
```



```

        </TabContent>
        <TabContent id='tabDelete'>
          <h1>Excluir</h1>
        </TabContent>
      </TabsContent>
    </Tabs>
  </Content>
</div>
)
}
}

const mapDispatchToProps = dispatch => bindActionCreators({selectTab, showTabs},
dispatch)
export default connect(null, mapDispatchToProps)(BillingCycle)

```

## 9.36. Integrando BillingCycleList com Redux

## Listagem 323 - Alterar arquivo billingCycleList.jsx\_

my-money-app/frontend/src/billingCycle/billingCycleList.jsx (aprox. linha 2, 3, 4, 8, 28, 29 e 30)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'
import { getList } from '../billingCycleActions'

class BillingCycleList extends Component {

  componentWillMount() {
    this.props.getList()
  }

  render() {
    return (
      <div>
        <table className='table'>
          <thead>
            <tr>
              <th>Nome</th>
              <th>Mês</th>
              <th>Ano</th>
            </tr>
          </thead>
          <tbody>

            </tbody>
        </table>
      </div>
    )
  }
}

const mapStateToProps = state => ({list: state.billingCycle.list})
const mapDispatchToProps = dispatch => bindActionCreators({getList}, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(BillingCycleList)
```

## 9.37. Exibindo os dados em BillingCycleList

## Listagem 324 - Alterar arquivo billingCycleList.jsx\_

my-money-app/frontend/src/billingCycle/billingCycleList.jsx (aprox. linha 12 e 34)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'
import { getList } from '../billingCycleActions'

class BillingCycleList extends Component {

  componentWillMount() {
    this.props.getList()
  }

  renderRows() {
    const list = this.props.list || []
    return list.map(bc => (
      <tr key={bc._id}>
        <td>{bc.name}</td>
        <td>{bc.month}</td>
        <td>{bc.year}</td>
      </tr>
    ))
  }

  render() {
    return (
      <div>
        <table className='table'>
          <thead>
            <tr>
              <th>Nome</th>
              <th>Mês</th>
              <th>Ano</th>
            </tr>
          </thead>
          <tbody>
            {this.renderRows()}
          </tbody>
        </table>
      </div>
    )
  }
}

const mapStateToProps = state => ({list: state.billingCycle.list})
const mapDispatchToProps = dispatch => bindActionCreators({getList}, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(BillingCycleList)
```

## 9.38. BillingCycleForm com Redux-Form (Parte 01)

### Listagem 325 - Alterar arquivo reducers.js

my-money-app/frontend/src/main/reducers.js (aprox. linha 2 e 12)

```
import { combineReducers } from 'redux'
import { reducer as formReducer } from 'redux-form'

import DashboardReducer from '../dashboard/dashboardReducer'
import TabReducer from '../common/tab/tabReducer'
import BillingCycleReducer from '../billingCycle/billingCycleReducer'

const rootReducer = combineReducers({
  dashboard: DashboardReducer,
  tab: TabReducer,
  billingCycle: BillingCycleReducer,
  form: formReducer
})

export default rootReducer
```

### Listagem 326 - Criar arquivo billingCycleForm.jsx

my-money-app/frontend/src/billingCycle/billingCycleForm.jsx

```
import React, { Component } from 'react'

class BillingCycleForm extends Component {
  render() {
    return (
      <form role='form'>
        <div className='box-body'>

          </div>
          <div className='box-footer'>
            <button type='submit' className='btn btn-primary'
>Submit</button>
          </div>
        </form>
      )
    }
  }

export default BillingCycleForm
```

## Listagem 327 - Alterar arquivo *billingCycle.jsx*

my-money-app/frontend/src/billingCycle/billingCycle.jsx (aprox. linha 15 e 41)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'
import TabHeader from '../common/tab/tabHeader'
import TabContent from '../common/tab/tabContent'
import { selectTab, showTabs } from '../common/tab/tabActions'

import List from './billingCycleList'
import Form from './billingCycleForm'

class BillingCycle extends Component {

  componentWillMount() {
    this.props.selectTab('tabList')
    this.props.showTabs('tabList', 'tabCreate')
  }

  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>
              <TabHeader label='Listar' icon='bars'
target='tabList' />
              <TabHeader label='Incluir' icon='plus'
target='tabCreate' />
              <TabHeader label='Alterar' icon='pencil'
target='tabUpdate' />
              <TabHeader label='Excluir' icon='trash-o'
target='tabDelete' />
            </TabsHeader>
            <TabsContent>
              <TabContent id='tabList'>
                <List />
              </TabContent>
              <TabContent id='tabCreate'>
                <Form />
              </TabContent>
            </TabsContent>
          </Tabs>
        </Content>
      </div>
    )
  }
}
```

```

        <TabContent id='tabUpdate'>
          <h1>Alterar</h1>
        </TabContent>
        <TabContent id='tabDelete'>
          <h1>Excluir</h1>
        </TabContent>
      </TabsContent>
    </Tabs>
  </Content>
</div>
)
}
}

const mapDispatchToProps = dispatch => bindActionCreators({selectTab, showTabs},
dispatch)
export default connect(null, mapDispatchToProps)(BillingCycle)

```

## 9.39. BillingCycleForm com Redux-Form (Parte 02)

## Listagem 328 - Alterar arquivo *billingCycleForm.jsx*

my-money-app/frontend/src/billingCycle/billingCycleForm.jsx (aprox. linha 2, 7, 10)

```
import React, { Component } from 'react'
import { reduxForm, Field } from 'redux-form'

class BillingCycleForm extends Component {

  render() {
    const { handleSubmit } = this.props
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component='input' />
          <Field name='month' component='input' />
          <Field name='year' component='input' />
        </div>
        <div className='box-footer'>
          <button type='submit' className='btn btn-primary'
>Submit</button>
        </div>
      </form>
    )
  }
}

export default reduxForm({form: 'billingCycleForm'})(BillingCycleForm)
```

## Listagem 329 - Alterar arquivo *billingCycleActions.js*

my-money-app/frontend/src/billingCycle/billingCycleActions.js (aprox. linha 12)

```
import axios from 'axios'
const BASE_URL = 'http://localhost:3003/api'

export function getList() {
  const request = axios.get(`${BASE_URL}/billingCycles`)
  return {
    type: 'BILLING_CYCLES_FETCHED',
    payload: request
  }
}

export function create(values) {
  axios.post(`${BASE_URL}/billingCycles`, values)
  return {
    type: 'TEMP'
  }
}
```



## Listagem 330 - Alterar arquivo *billingCycle.jsx*

my-money-app/frontend/src/billingCycle/billingCycle.jsx (aprox. linha 13, 42 e 54)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'
import TabHeader from '../common/tab/tabHeader'
import TabContent from '../common/tab/tabContent'
import { selectTab, showTabs } from '../common/tab/tabActions'
import { create } from './billingCycleActions'

import List from './billingCycleList'
import Form from './billingCycleForm'

class BillingCycle extends Component {

  componentWillMount() {
    this.props.selectTab('tabList')
    this.props.showTabs('tabList', 'tabCreate')
  }

  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>
              <TabHeader label='Listar' icon='bars'
target='tabList' />
              <TabHeader label='Incluir' icon='plus'
target='tabCreate' />
              <TabHeader label='Alterar' icon='pencil'
target='tabUpdate' />
              <TabHeader label='Excluir' icon='trash-o'
target='tabDelete' />
            </TabsHeader>
            <TabsContent>
              <TabContent id='tabList'>
                <List />
              </TabContent>
              <TabContent id='tabCreate'>
                <Form onSubmit={this.props.create} />
              </TabContent>
            </TabsContent>
          </Tabs>
        </Content>
      </div>
    )
  }
}
```

```

        </TabContent>
        <TabContent id='tabUpdate'>
          <h1>Alterar</h1>
        </TabContent>
        <TabContent id='tabDelete'>
          <h1>Excluir</h1>
        </TabContent>
      </TabsContent>
    </Tabs>
  </Content>
</div>
)
}
}

const mapDispatchToProps = dispatch => bindActionCreators({
  selectTab, showTabs, create
}, dispatch)
export default connect(null, mapDispatchToProps)(BillingCycle)

```

## 9.40. Exibindo as Mensagens de Sucesso e Erro



Criar uma pasta chamada `msg` em `src/common/msg`

### Listagem 331 - Criar arquivo `messages.jsx`

`my-money-app/frontend/src/common/msg/messages.jsx`

```

import React from 'react'
import ReduxToastr from 'react-redux-toastr'
import 'modules/react-redux-toastr/lib/css/react-redux-toastr.css'

export default props => (
  <ReduxToastr
    timeout={4000}
    newestOnTop={false}
    preventDuplicates={true}
    position='top-right'
    transitionIn='fadeIn'
    transitionOut='fadeOut'
    progressBar />
)

```

### Listagem 332 - Alterar arquivo app.jsx

my-money-app/frontend/src/main/app.jsx (aprox. linha 8 e 18)

```
import '../common/template/dependencies'
import React from 'react'

import Header from '../common/template/header'
import SideBar from '../common/template/sideBar'
import Footer from '../common/template/footer'
import Routes from './routes'
import Messages from '../common/msg/messages'

export default props => (
  <div className='wrapper'>
    <Header />
    <SideBar />
    <div className='content-wrapper'>
      <Routes />
    </div>
    <Footer />
    <Messages />
  </div>
)
```

### Listagem 333 - Alterar arquivo reducers.js

my-money-app/frontend/src/main/reducers.js (aprox. linha 3 e 14)

```
import { combineReducers } from 'redux'
import { reducer as formReducer } from 'redux-form'
import { reducer as toastrReducer } from 'react-redux-toastr'

import DashboardReducer from '../dashboard/dashboardReducer'
import TabReducer from '../common/tab/tabReducer'
import BillingCycleReducer from '../billingCycle/billingCycleReducer'

const rootReducer = combineReducers({
  dashboard: DashboardReducer,
  tab: TabReducer,
  billingCycle: BillingCycleReducer,
  form: formReducer,
  toastr: toastrReducer
})

export default rootReducer
```

### Listagem 334 - Alterar arquivo *billingCycleActions.js*

my-money-app/frontend/src/billingCycle/billingCycleActions.js (aprox. linha 2 e 15)

```
import axios from 'axios'
import { toastr } from 'react-redux-toastr'
const BASE_URL = 'http://localhost:3003/api'

export function getList() {
  const request = axios.get(`${BASE_URL}/billingCycles`)
  return {
    type: 'BILLING_CYCLES_FETCHED',
    payload: request
  }
}

export function create(values) {
  axios.post(`${BASE_URL}/billingCycles`, values)
    .then(resp => {
      toastr.success('Sucesso', 'Operação Realizada com sucesso.')
    })
    .catch(e => {
      e.response.data.errors.forEach(error => toastr.error('Erro', error))
    })
  return {
    type: 'TEMP'
  }
}
```

## 9.41. Melhorias na Inclusão do Ciclo de Pagamento

## Listagem 335 - Alterar arquivo *index.jsx*

my-money-app/frontend/src/index.jsx (aprox. linha 7, 8 e 15)

```
import React from 'react'
import ReactDOM from 'react-dom'
import { applyMiddleware, createStore } from 'redux'
import { Provider } from 'react-redux'

import promise from 'redux-promise'
import multi from 'redux-multi'
import thunk from 'redux-thunk'

import App from './main/app'
import reducers from './main/reducers'

const devTools = window.__REDUX_DEVTOOLS_EXTENSION__
  && window.__REDUX_DEVTOOLS_EXTENSION__()
const store = applyMiddleware(multi, thunk, promise)(createStore)(reducers,
devTools)
ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>
, document.getElementById('app'))
```

## Listagem 336 - Alterar arquivo *billingCycleActions.js*

my-money-app/frontend/src/billingCycle/billingCycleActions.js (aprox. linha 3, 4, 14 e 21)

```
import axios from 'axios'
import { toastr } from 'react-redux-toastr'
import { reset as resetForm } from 'redux-form'
import { showTabs, selectTab } from '../common/tab/tabActions'

const BASE_URL = 'http://localhost:3003/api'

export function getList() {
  const request = axios.get(`${BASE_URL}/billingCycles`)
  return {
    type: 'BILLING_CYCLES_FETCHED',
    payload: request
  }
}

export function create(values) {
  return dispatch => {
    axios.post(`${BASE_URL}/billingCycles`, values)
      .then(resp => {
        toastr.success('Sucesso', 'Operação Realizada com sucesso.')
        dispatch([
          resetForm('billingCycleForm'),
          getList(),
          selectTab('tabList'),
          showTabs('tabList', 'tabCreate')
        ])
      })
      .catch(e => {
        e.response.data.errors.forEach(error => toastr.error('Erro',
error))
      })
  }
}
```

## 9.42. Field com Componente Personalizado



Criar uma pasta chamada **form** em **src/common/form**

### Listagem 337 - Criar arquivo *labelAndInput.jsx*

my-money-app/frontend/src/common/form/labelAndInput.jsx

```
import React from 'react'
import Grid from '../layout/grid'

export default props => (
  <Grid cols={props.cols}>
    <div className='form-group'>
      <label htmlFor={props.name}>{props.label}</label>
      <input {...props.input} className='form-control'
        placeholder={props.placeholder}
        readOnly={props.readOnly} type={props.type} />
    </div>
  </Grid>
)
```

### Listagem 338 - Alterar arquivo *billingCycleForm.jsx*

my-money-app/frontend/src/billingCycle/billingCycleForm.jsx (aprox. linha 3, 12, 14 e 16)

```
import React, { Component } from 'react'
import { reduxForm, Field } from 'redux-form'
import LabelAndInput from '../common/form/labelAndInput'

class BillingCycleForm extends Component {

  render() {
    const { handleSubmit } = this.props
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component={LabelAndInput}
            label='Nome' cols='12 4' placeholder='Informe o nome' />
          <Field name='month' component={LabelAndInput} type='number'
            label='Mês' cols='12 4' placeholder='Informe o mês' />
          <Field name='year' component={LabelAndInput} type='number'
            label='Ano' cols='12 4' placeholder='Informe o ano' />
        </div>
        <div className='box-footer'>
          <button type='submit' className='btn btn-primary'
            >Submit</button>
        </div>
      </form>
    )
  }
}

export default reduxForm({form: 'billingCycleForm'})(BillingCycleForm)
```

## 9.43. Exibir Aba de Alterar Ciclo de Pagamento



### Listagem 339 - Alterar arquivo *billingCycleActions.js*

my-money-app/frontend/src/billingCycle/billingCycleActions.js (aprox. linha 34)

```
import axios from 'axios'
import { toastr } from 'react-redux-toastr'
import { reset as resetForm } from 'redux-form'
import { showTabs, selectTab } from '../common/tab/tabActions'

const BASE_URL = 'http://localhost:3003/api'

export function getList() {
  const request = axios.get(`${BASE_URL}/billingCycles`)
  return {
    type: 'BILLING_CYCLES_FETCHED',
    payload: request
  }
}

export function create(values) {
  return dispatch => {
    axios.post(`${BASE_URL}/billingCycles`, values)
      .then(resp => {
        toastr.success('Sucesso', 'Operação Realizada com sucesso.')
        dispatch([
          resetForm('billingCycleForm'),
          getList(),
          selectTab('tabList'),
          showTabs('tabList', 'tabCreate')
        ])
      })
      .catch(e => {
        e.response.data.errors.forEach(error => toastr.error('Erro',
error))
      })
  }
}

export function showUpdate(billingCycle) {
  return [
    showTabs('tabUpdate'),
    selectTab('tabUpdate')
  ]
}
```

## Listagem 340 - Alterar arquivo billingCycleList.jsx\_

my-money-app/frontend/src/billingCycle/billingCycleList.jsx (aprox. linha 4, 19 e 37)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'
import { getList, showUpdate } from './billingCycleActions'

class BillingCycleList extends Component {

  componentWillMount() {
    this.props.getList()
  }

  renderRows() {
    const list = this.props.list || []
    return list.map(bc => (
      <tr key={bc._id}>
        <td>{bc.name}</td>
        <td>{bc.month}</td>
        <td>{bc.year}</td>
        <td>
          <button className='btn btn-warning' onClick={() =>
this.props.showUpdate(bc)}>
            <i className='fa fa-pencil'></i>
          </button>
        </td>
      </tr>
    ))
  }

  render() {
    return (
      <div>
        <table className='table'>
          <thead>
            <tr>
              <th>Nome</th>
              <th>Mês</th>
              <th>Ano</th>
              <th>Ações</th>
            </tr>
          </thead>
          <tbody>
            {this.renderRows()}
          </tbody>
        </table>
      </div>
    )
  }
}
```

```
}  
}  
  
const mapStateToProps = state => ({list: state.billingCycle.list})  
const mapDispatchToProps = dispatch => bindActionCreators({getList, showUpdate},  
dispatch)  
export default connect(mapStateToProps, mapDispatchToProps)(BillingCycleList)
```

## Listagem 341 - Alterar arquivo *billingCycle.jsx*

my-money-app/frontend/src/billingCycle/billingCycle.jsx (aprox. linha 45)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'
import TabHeader from '../common/tab/tabHeader'
import TabContent from '../common/tab/tabContent'
import { selectTab, showTabs } from '../common/tab/tabActions'
import { create } from './billingCycleActions'

import List from './billingCycleList'
import Form from './billingCycleForm'

class BillingCycle extends Component {

  componentWillMount() {
    this.props.selectTab('tabList')
    this.props.showTabs('tabList', 'tabCreate')
  }

  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>
              <TabHeader label='Listar' icon='bars'
target='tabList' />
              <TabHeader label='Incluir' icon='plus'
target='tabCreate' />
              <TabHeader label='Alterar' icon='pencil'
target='tabUpdate' />
              <TabHeader label='Excluir' icon='trash-o'
target='tabDelete' />
            </TabsHeader>
            <TabsContent>
              <TabContent id='tabList'>
                <List />
              </TabContent>
              <TabContent id='tabCreate'>
                <Form onSubmit={this.props.create} />
              </TabContent>
            </TabsContent>
          </Tabs>
        </Content>
      </div>
    )
  }
}
```

```

        </TabContent>
        <TabContent id='tabUpdate'>
          <Form />
        </TabContent>
        <TabContent id='tabDelete'>
          <h1>Excluir</h1>
        </TabContent>
      </TabsContent>
    </Tabs>
  </Content>
</div>
)
}
}

const mapDispatchToProps = dispatch => bindActionCreators({
  selectTab, showTabs, create
}, dispatch)
export default connect(null, mapDispatchToProps)(BillingCycle)

```

## 9.44. Inicializar Formulário com Dados

## Listagem 342 - Alterar arquivo *billingCycleActions.js*

my-money-app/frontend/src/billingCycle/billingCycleActions.js (aprox. linha 3 e 38)

```
import axios from 'axios'
import { toastr } from 'react-redux-toastr'
import { reset as resetForm, initialize } from 'redux-form'
import { showTabs, selectTab } from '../common/tab/tabActions'

const BASE_URL = 'http://localhost:3003/api'

export function getList() {
  const request = axios.get(`${BASE_URL}/billingCycles`)
  return {
    type: 'BILLING_CYCLES_FETCHED',
    payload: request
  }
}

export function create(values) {
  return dispatch => {
    axios.post(`${BASE_URL}/billingCycles`, values)
      .then(resp => {
        toastr.success('Sucesso', 'Operação Realizada com sucesso.')
        dispatch([
          resetForm('billingCycleForm'),
          getList(),
          selectTab('tabList'),
          showTabs('tabList', 'tabCreate')
        ])
      })
      .catch(e => {
        e.response.data.errors.forEach(error => toastr.error('Erro',
error))
      })
  }
}

export function showUpdate(billingCycle) {
  return [
    showTabs('tabUpdate'),
    selectTab('tabUpdate'),
    initialize('billingCycleForm', billingCycle)
  ]
}
```

## Listagem 343 - Alterar arquivo *billingCycleForm.jsx*

my-money-app/frontend/src/billingCycle/billingCycleForm.jsx (aprox. linha 27)

```
import React, { Component } from 'react'
import { reduxForm, Field } from 'redux-form'
import LabelAndInput from '../common/form/labelAndInput'

class BillingCycleForm extends Component {

  render() {
    const { handleSubmit } = this.props
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component={LabelAndInput}
            label='Nome' cols='12 4' placeholder='Informe o nome' />
          <Field name='month' component={LabelAndInput} type='number'
            label='Mês' cols='12 4' placeholder='Informe o mês' />
          <Field name='year' component={LabelAndInput} type='number'
            label='Ano' cols='12 4' placeholder='Informe o ano' />
        </div>
        <div className='box-footer'>
          <button type='submit' className='btn btn-primary'
            >Submit</button>
        </div>
      </form>
    )
  }
}

export default reduxForm({form: 'billingCycleForm', destroyOnUnmount:
false})(BillingCycleForm)
```

## 9.45. Função do Botão Cancelar

### **Listagem 344 - Alterar arquivo *billingCycleActions.js***



```

import axios from 'axios'
import { toastr } from 'react-redux-toastr'
import { reset as resetForm, initialize } from 'redux-form'
import { showTabs, selectTab } from '../common/tab/tabActions'

const BASE_URL = 'http://localhost:3003/api'
const INITIAL_VALUES = {}

export function getList() {
  const request = axios.get(`${BASE_URL}/billingCycles`)
  return {
    type: 'BILLING_CYCLES_FETCHED',
    payload: request
  }
}

export function create(values) {
  return dispatch => {
    axios.post(`${BASE_URL}/billingCycles`, values)
      .then(resp => {
        toastr.success('Sucesso', 'Operação Realizada com sucesso.')
        dispatch(init())
      })
      .catch(e => {
        e.response.data.errors.forEach(error => toastr.error('Erro',
error))
      })
  }
}

export function showUpdate(billingCycle) {
  return [
    showTabs('tabUpdate'),
    selectTab('tabUpdate'),
    initialize('billingCycleForm', billingCycle)
  ]
}

export function init() {
  return [
    showTabs('tabList', 'tabCreate'),
    selectTab('tabList'),
    getList(),
    initialize('billingCycleForm', INITIAL_VALUES)
  ]
}

```

## Listagem 345 - Alterar arquivo *billingCycleForm.jsx*

my-money-app/frontend/src/billingCycle/billingCycleForm.jsx (aprox. linha 2, 3, 6, 22 e 33)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { reduxForm, Field } from 'redux-form'

import { init } from '../billingCycleActions'
import LabelAndInput from '../common/form/labelAndInput'

class BillingCycleForm extends Component {

  render() {
    const { handleSubmit } = this.props
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component={LabelAndInput}
            label='Nome' cols='12 4' placeholder='Informe o nome' />
          <Field name='month' component={LabelAndInput} type='number'
            label='Mês' cols='12 4' placeholder='Informe o mês' />
          <Field name='year' component={LabelAndInput} type='number'
            label='Ano' cols='12 4' placeholder='Informe o ano' />
        </div>
        <div className='box-footer'>
          <button type='submit' className='btn btn-primary'
            >Submit</button>
          <button type='button' className='btn btn-default'
            onClick={this.props.init}>Cancelar</button>
        </div>
      </form>
    )
  }
}

BillingCycleForm = reduxForm({form: 'billingCycleForm', destroyOnUnmount:
false})(BillingCycleForm)
const mapDispatchToProps = dispatch => bindActionCreators({init}, dispatch)
export default connect(null, mapDispatchToProps)(BillingCycleForm)
```

## 9.46. Alterar Ciclo de Pagamento

## Listagem 346 - Alterar arquivo *billingCycleActions.js*

my-money-app/frontend/src/billingCycle/billingCycleActions.js (aprox. linha 18, 25 e 30)

```
import axios from 'axios'
import { toastr } from 'react-redux-toastr'
import { reset as resetForm, initialize } from 'redux-form'
import { showTabs, selectTab } from '../common/tab/tabActions'

const BASE_URL = 'http://localhost:3003/api'
const INITIAL_VALUES = {}

export function getList() {
  const request = axios.get(`${BASE_URL}/billingCycles`)
  return {
    type: 'BILLING_CYCLES_FETCHED',
    payload: request
  }
}

export function create(values) {
  return submit(values, 'post')
}

export function update(values) {
  return submit(values, 'put')
}

function submit(values, method) {
  return dispatch => {
    const id = values._id ? values._id : ''
    axios[method](`${BASE_URL}/billingCycles/${id}`, values)
      .then(resp => {
        toastr.success('Sucesso', 'Operação Realizada com sucesso.')
        dispatch(init())
      })
      .catch(e => {
        e.response.data.errors.forEach(error => toastr.error('Erro',
error))
      })
  }
}

export function showUpdate(billingCycle) {
  return [
    showTabs('tabUpdate'),
    selectTab('tabUpdate'),
    initialize('billingCycleForm', billingCycle)
  ]
}
```

```
export function init() {  
  return [  
    showTabs('tabList', 'tabCreate'),  
    selectTab('tabList'),  
    getList(),  
    initialize('billingCycleForm', INITIAL_VALUES)  
  ]  
}
```

## Listagem 347 - Alterar arquivo *billingCycle.jsx*

my-money-app/frontend/src/billingCycle/billingCycle.jsx (aprox. linha 13, 45 e 57)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'
import TabHeader from '../common/tab/tabHeader'
import TabContent from '../common/tab/tabContent'
import { selectTab, showTabs } from '../common/tab/tabActions'
import { create, update } from './billingCycleActions'

import List from './billingCycleList'
import Form from './billingCycleForm'

class BillingCycle extends Component {

  componentWillMount() {
    this.props.selectTab('tabList')
    this.props.showTabs('tabList', 'tabCreate')
  }

  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>
              <TabHeader label='Listar' icon='bars'
target='tabList' />
              <TabHeader label='Incluir' icon='plus'
target='tabCreate' />
              <TabHeader label='Alterar' icon='pencil'
target='tabUpdate' />
              <TabHeader label='Excluir' icon='trash-o'
target='tabDelete' />
            </TabsHeader>
            <TabsContent>
              <TabContent id='tabList'>
                <List />
              </TabContent>
              <TabContent id='tabCreate'>
                <Form onSubmit={this.props.create} />
              </TabContent>
            </TabsContent>
          </Tabs>
        </Content>
      </div>
    )
  }
}
```

```

        </TabContent>
        <TabContent id='tabUpdate'>
          <Form onSubmit={this.props.update} />
        </TabContent>
        <TabContent id='tabDelete'>
          <h1>Excluir</h1>
        </TabContent>
      </TabsContent>
    </Tabs>
  </Content>
</div>
)
}
}

const mapDispatchToProps = dispatch => bindActionCreators({
  selectTab, showTabs, create, update
}, dispatch)
export default connect(null, mapDispatchToProps)(BillingCycle)

```

## 9.47. Excluir Ciclo de Pagamento

## Listagem 348 - Alterar arquivo *billingCycleActions.js*

my-money-app/frontend/src/billingCycle/billingCycleActions.js (aprox. linha 47)

```
import axios from 'axios'
import { toastr } from 'react-redux-toastr'
import { reset as resetForm, initialize } from 'redux-form'
import { showTabs, selectTab } from '../common/tab/tabActions'

const BASE_URL = 'http://localhost:3003/api'
const INITIAL_VALUES = {}

export function getList() {
  const request = axios.get(`${BASE_URL}/billingCycles`)
  return {
    type: 'BILLING_CYCLES_FETCHED',
    payload: request
  }
}

export function create(values) {
  return submit(values, 'post')
}

export function update(values) {
  return submit(values, 'put')
}

function submit(values, method) {
  return dispatch => {
    const id = values._id ? values._id : ''
    axios[method](`${BASE_URL}/billingCycles/${id}`, values)
      .then(resp => {
        toastr.success('Sucesso', 'Operação Realizada com sucesso.')
        dispatch(init())
      })
      .catch(e => {
        e.response.data.errors.forEach(error => toastr.error('Erro',
error))
      })
  }
}

export function showUpdate(billingCycle) {
  return [
    showTabs('tabUpdate'),
    selectTab('tabUpdate'),
    initialize('billingCycleForm', billingCycle)
  ]
}
```

```
export function showDelete(billingCycle) {
  return [
    showTabs('tabDelete'),
    selectTab('tabDelete'),
    initialize('billingCycleForm', billingCycle)
  ]
}

export function init() {
  return [
    showTabs('tabList', 'tabCreate'),
    selectTab('tabList'),
    getList(),
    initialize('billingCycleForm', INITIAL_VALUES)
  ]
}
```



## Listagem 349 - Alterar arquivo billingCycleList.jsx\_

my-money-app/frontend/src/billingCycle/billingCycleList.jsx (aprox. linha 4, 23 e 53)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'
import { getList, showUpdate, showDelete } from './billingCycleActions'

class BillingCycleList extends Component {

  componentWillMount() {
    this.props.getList()
  }

  renderRows() {
    const list = this.props.list || []
    return list.map(bc => (
      <tr key={bc._id}>
        <td>{bc.name}</td>
        <td>{bc.month}</td>
        <td>{bc.year}</td>
        <td>
          <button className='btn btn-warning' onClick={() =>
this.props.showUpdate(bc)}>
            <i className='fa fa-pencil'></i>
          </button>
          <button className='btn btn-danger' onClick={() =>
this.props.showDelete(bc)}>
            <i className='fa fa-trash-o'></i>
          </button>
        </td>
      </tr>
    ))
  }

  render() {
    return (
      <div>
        <table className='table'>
          <thead>
            <tr>
              <th>Nome</th>
              <th>Mês</th>
              <th>Ano</th>
              <th>Ações</th>
            </tr>
          </thead>
          <tbody>
            {this.renderRows()}
          </tbody>
        </table>
      </div>
    )
  }
}
```

```

        </tbody>
      </table>
    </div>
  )
}
}

const mapStateToProps = state => ({list: state.billingCycle.list})
const mapDispatchToProps = dispatch => bindActionCreators({getList, showUpdate,
showDelete}, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(BillingCycleList)

```

## Listagem 350 - Alterar arquivo *billingCycle.jsx*

my-money-app/frontend/src/billingCycle/billingCycle.jsx (aprox. linha 13, 48 e 59)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'
import TabHeader from '../common/tab/tabHeader'
import TabContent from '../common/tab/tabContent'
import { selectTab, showTabs } from '../common/tab/tabActions'
import { create, update, remove } from './billingCycleActions'

import List from './billingCycleList'
import Form from './billingCycleForm'

class BillingCycle extends Component {

  componentWillMount() {
    this.props.selectTab('tabList')
    this.props.showTabs('tabList', 'tabCreate')
  }

  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>
              <TabHeader label='Listar' icon='bars'
target='tabList' />
              <TabHeader label='Incluir' icon='plus'
target='tabCreate' />
              <TabHeader label='Alterar' icon='pencil'
target='tabUpdate' />
              <TabHeader label='Excluir' icon='trash-o'
target='tabDelete' />
            </TabsHeader>
            <TabsContent>
              <TabContent id='tabList'>
                <List />
              </TabContent>
              <TabContent id='tabCreate'>
                <Form onSubmit={this.props.create} />
              </TabContent>
            </TabsContent>
          </Tabs>
        </Content>
      </div>
    )
  }
}
```

```

        </TabContent>
        <TabContent id='tabUpdate'>
          <Form onSubmit={this.props.update} />
        </TabContent>
        <TabContent id='tabDelete'>
          <Form onSubmit={this.props.remove} readOnly={
true} />
        </TabContent>
      </TabsContent>
    </Tabs>
  </Content>
</div>
)
}
}

const mapDispatchToProps = dispatch => bindActionCreators({
  selectTab, showTabs, create, update, remove
}, dispatch)
export default connect(null, mapDispatchToProps)(BillingCycle)

```

## Listagem 351 - Alterar arquivo *billingCycleActions.js*

my-money-app/frontend/src/billingCycle/billingCycleActions.js (aprox. linha 25)

```
import axios from 'axios'
import { toastr } from 'react-redux-toastr'
import { reset as resetForm, initialize } from 'redux-form'
import { showTabs, selectTab } from '../common/tab/tabActions'

const BASE_URL = 'http://localhost:3003/api'
const INITIAL_VALUES = {}

export function getList() {
  const request = axios.get(`${BASE_URL}/billingCycles`)
  return {
    type: 'BILLING_CYCLES_FETCHED',
    payload: request
  }
}

export function create(values) {
  return submit(values, 'post')
}

export function update(values) {
  return submit(values, 'put')
}

export function remove(values) {
  return submit(values, 'delete')
}

function submit(values, method) {
  return dispatch => {
    const id = values._id ? values._id : ''
    axios[method](`${BASE_URL}/billingCycles/${id}`, values)
      .then(resp => {
        toastr.success('Sucesso', 'Operação Realizada com sucesso.')
        dispatch(init())
      })
      .catch(e => {
        e.response.data.errors.forEach(error => toastr.error('Erro',
error))
      })
  }
}

export function showUpdate(billingCycle) {
  return [
    showTabs('tabUpdate'),
  ]
}
```

```
        selectTab('tabUpdate'),
        initialize('billingCycleForm', billingCycle)
    ]
}

export function init() {
    return [
        showTabs('tabList', 'tabCreate'),
        selectTab('tabList'),
        getList(),
        initialize('billingCycleForm', INITIAL_VALUES)
    ]
}
```

## Listagem 352 - Alterar arquivo *billingCycleForm.jsx*

my-money-app/frontend/src/billingCycle/billingCycleForm.jsx (aprox. linha 12, 16, 18 e 20)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { reduxForm, Field } from 'redux-form'

import { init } from '../billingCycleActions'
import LabelAndInput from '../common/form/labelAndInput'

class BillingCycleForm extends Component {

  render() {
    const { handleSubmit, readOnly } = this.props
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component={LabelAndInput}
readOnly={readOnly}
          label='Nome' cols='12 4' placeholder='Informe o nome' />
          <Field name='month' component={LabelAndInput} type='number'
readOnly={readOnly}
          label='Mês' cols='12 4' placeholder='Informe o mês' />
          <Field name='year' component={LabelAndInput} type='number'
readOnly={readOnly}
          label='Ano' cols='12 4' placeholder='Informe o ano' />
        </div>
        <div className='box-footer'>
          <button type='submit' className='btn btn-primary'
>Submit</button>
          <button type='button' className='btn btn-default'
onClick={this.props.init}>Cancelar</button>
        </div>
      </form>
    )
  }
}

BillingCycleForm = reduxForm({form: 'billingCycleForm', destroyOnUnmount:
false})(BillingCycleForm)
const mapDispatchToProps = dispatch => bindActionCreators({init}, dispatch)
export default connect(null, mapDispatchToProps)(BillingCycleForm)
```

## 9.48. Finalizando o Cadastro Básico (Melhorias)

## Listagem 353 - Alterar arquivo *billingCycleForm.jsx*

my-money-app/frontend/src/billingCycle/billingCycleForm.jsx (aprox. linha 24 e 25)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { reduxForm, Field } from 'redux-form'

import { init } from '../billingCycleActions'
import LabelAndInput from '../common/form/labelAndInput'

class BillingCycleForm extends Component {

  render() {
    const { handleSubmit, readOnly } = this.props
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component={LabelAndInput}
readOnly={readOnly}
          label='Nome' cols='12 4' placeholder='Informe o nome' />
          <Field name='month' component={LabelAndInput} type='number'
readOnly={readOnly}
          label='Mês' cols='12 4' placeholder='Informe o mês' />
          <Field name='year' component={LabelAndInput} type='number'
readOnly={readOnly}
          label='Ano' cols='12 4' placeholder='Informe o ano' />
        </div>
        <div className='box-footer'>
          <button type='submit' className={`btn btn-${
this.props.submitClass}`}>
            {this.props.submitLabel}
          </button>
          <button type='button' className='btn btn-default'
onClick={this.props.init}>Cancelar</button>
        </div>
      </form>
    )
  }
}

BillingCycleForm = reduxForm({form: 'billingCycleForm', destroyOnUnmount:
false})(BillingCycleForm)
const mapDispatchToProps = dispatch => bindActionCreators({init}, dispatch)
export default connect(null, mapDispatchToProps)(BillingCycleForm)
```



## Listagem 354 - Alterar arquivo *billingCycle.jsx*

my-money-app/frontend/src/billingCycle/billingCycle.jsx (aprox. linha 43, 47 e 51)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'
import TabHeader from '../common/tab/tabHeader'
import TabContent from '../common/tab/tabContent'
import { selectTab, showTabs } from '../common/tab/tabActions'
import { create, update, remove } from './billingCycleActions'

import List from './billingCycleList'
import Form from './billingCycleForm'

class BillingCycle extends Component {

  componentWillMount() {
    this.props.selectTab('tabList')
    this.props.showTabs('tabList', 'tabCreate')
  }

  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>
              <TabHeader label='Listar' icon='bars'
target='tabList' />
              <TabHeader label='Incluir' icon='plus'
target='tabCreate' />
              <TabHeader label='Alterar' icon='pencil'
target='tabUpdate' />
              <TabHeader label='Excluir' icon='trash-o'
target='tabDelete' />
            </TabsHeader>
            <TabsContent>
              <TabContent id='tabList'>
                <List />
              </TabContent>
              <TabContent id='tabCreate'>
                <Form onSubmit={this.props.create}>

```

```

        submitLabel='Incluir' submitClass='primary'
    />
    </TabContent>
    <TabContent id='tabUpdate'>
        <Form onSubmit={this.props.update}
            submitLabel='Alterar' submitClass='info' />
    </TabContent>
    <TabContent id='tabDelete'>
        <Form onSubmit={this.props.remove} readOnly={
true}
            submitLabel='Excluir' submitClass='danger' />
    </TabContent>
</TabsContent>
</Tabs>
</Content>
</div>
    )
  }
}

const mapDispatchToProps = dispatch => bindActionCreators({
  selectTab, showTabs, create, update, remove
}, dispatch)
export default connect(null, mapDispatchToProps)(BillingCycle)

```

### Listagem 355 - Alterar arquivo custom.css

my-money-app/frontend/src/common/template/custom.css (aprox. linha 7 e 11)

```

.main-footer {
  position: fixed;
  bottom: 0px;
  width: 100%;
}

button {
  margin-left: 5px;
}

.table-actions {
  width: 150px;
}

```

## Listagem 356 - Alterar arquivo billingCycleList.jsx\_

my-money-app/frontend/src/billingCycle/billingCycleList.jsx (aprox. linha 40)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'
import { getList, showUpdate, showDelete } from './billingCycleActions'

class BillingCycleList extends Component {

  componentWillMount() {
    this.props.getList()
  }

  renderRows() {
    const list = this.props.list || []
    return list.map(bc => (
      <tr key={bc._id}>
        <td>{bc.name}</td>
        <td>{bc.month}</td>
        <td>{bc.year}</td>
        <td>
          <button className='btn btn-warning' onClick={() =>
this.props.showUpdate(bc)}>
            <i className='fa fa-pencil'></i>
          </button>
          <button className='btn btn-danger' onClick={() =>
this.props.showDelete(bc)}>
            <i className='fa fa-trash-o'></i>
          </button>
        </td>
      </tr>
    ))
  }

  render() {
    return (
      <div>
        <table className='table'>
          <thead>
            <tr>
              <th>Nome</th>
              <th>Mês</th>
              <th>Ano</th>
              <th className='table-actions'>Ações</th>
            </tr>
          </thead>
          <tbody>
            {this.renderRows()}
          </tbody>
        </table>
      </div>
    )
  }
}
```

```

        </tbody>
      </table>
    </div>
  )
}
}

const mapStateToProps = state => ({list: state.billingCycle.list})
const mapDispatchToProps = dispatch => bindActionCreators({getList, showUpdate,
showDelete}, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(BillingCycleList)

```

## 9.49. Componente CreditList Básico

## Listagem 357 - Criar arquivo creditList.jsx

my-money-app/frontend/src/billingCycle/creditList.jsx

```
import React, { Component } from 'react'
import { Field } from 'redux-form'
import Grid from '../common/layout/grid'

class CreditList extends Component {

  renderRows() {
    return (
      <tr>
        <td><Field name='credits[0].name' component='input' /></td>
        <td><Field name='credits[0].value' component='input' /></td>
        <td></td>
      </tr>
    )
  }

  render() {
    return (
      <Grid cols={this.props.cols}>
        <fieldset>
          <legend>Créditos</legend>
          <table>
            <thead>
              <tr>
                <th>Nome</th>
                <th>Valor</th>
                <th>Ações</th>
              </tr>
            </thead>
            <tbody>
              {this.renderRows()}
            </tbody>
          </table>
        </fieldset>
      </Grid>
    )
  }
}

export default CreditList
```

## Listagem 358 - Alterar arquivo *billingCycleForm.jsx*

my-money-app/frontend/src/billingCycle/billingCycleForm.jsx (aprox. linha 8 e 23)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { reduxForm, Field } from 'redux-form'

import { init } from './billingCycleActions'
import LabelAndInput from '../common/form/labelAndInput'
import CreditList from './creditList'

class BillingCycleForm extends Component {

  render() {
    const { handleSubmit, readOnly } = this.props
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component={LabelAndInput}
readOnly={readOnly}
            label='Nome' cols='12 4' placeholder='Informe o nome' />
          <Field name='month' component={LabelAndInput} type='number'
readOnly={readOnly}
            label='Mês' cols='12 4' placeholder='Informe o mês' />
          <Field name='year' component={LabelAndInput} type='number'
readOnly={readOnly}
            label='Ano' cols='12 4' placeholder='Informe o ano' />
          <CreditList cols='12 6' />
        </div>
        <div className='box-footer'>
          <button type='submit' className={`btn btn-${
this.props.submitClass}`}>
            {this.props.submitLabel}
          </button>
          <button type='button' className='btn btn-default'
            onClick={this.props.init}>Cancelar</button>
        </div>
      </form>
    )
  }
}

BillingCycleForm = reduxForm({form: 'billingCycleForm', destroyOnUnmount:
false})(BillingCycleForm)
const mapDispatchToProps = dispatch => bindActionCreators({init}, dispatch)
export default connect(null, mapDispatchToProps)(BillingCycleForm)
```

## 9.50. Componente Input para CreditList

### Listagem 359 - Criar arquivo input.jsx\_

my-money-app/frontend/src/common/form/input.jsx

```
import React from 'react'

export default props => (
  <input {...props.input}
    className='form-control'
    placeholder={props.placeholder}
    readOnly={props.readOnly}
    type={props.type} />
)
```

## Listagem 360 - Alterar arquivo creditList.jsx

my-money-app/frontend/src/billingCycle/creditList.jsx (aprox. linha 4, 12 e 14)

```
import React, { Component } from 'react'
import { Field } from 'redux-form'
import Grid from '../common/layout/grid'
import Input from '../common/form/input'

class CreditList extends Component {

  renderRows() {
    return (
      <tr>
        <td><Field name='credits[0].name' component={Input}
placeholder='Informe o nome' readOnly={this.props.readOnly}
/></td>

        <td><Field name='credits[0].value' component={Input}
placeholder='Informe o valor' readOnly={this.props.readOnly}
/></td>

        <td></td>
      </tr>
    )
  }

  render() {
    return (
      <Grid cols={this.props.cols}>
        <fieldset>
          <legend>Créditos</legend>
          <table>
            <thead>
              <tr>
                <th>Nome</th>
                <th>Valor</th>
                <th>Ações</th>
              </tr>
            </thead>
            <tbody>
              {this.renderRows()}
            </tbody>
          </table>
        </fieldset>
      </Grid>
    )
  }
}

export default CreditList
```



## Listagem 361 - Alterar arquivo *billingCycleForm.jsx*

my-money-app/frontend/src/billingCycle/billingCycleForm.jsx (aprox. linha 23)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { reduxForm, Field } from 'redux-form'

import { init } from '../billingCycleActions'
import LabelAndInput from '../common/form/labelAndInput'
import CreditList from '../creditList'

class BillingCycleForm extends Component {

  render() {
    const { handleSubmit, readOnly } = this.props
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component={LabelAndInput}
readOnly={readOnly}
            label='Nome' cols='12 4' placeholder='Informe o nome' />
          <Field name='month' component={LabelAndInput} type='number'
readOnly={readOnly}
            label='Mês' cols='12 4' placeholder='Informe o mês' />
          <Field name='year' component={LabelAndInput} type='number'
readOnly={readOnly}
            label='Ano' cols='12 4' placeholder='Informe o ano' />
          <CreditList cols='12 6' readOnly={readOnly} />
        </div>
        <div className='box-footer'>
          <button type='submit' className={`btn btn-${
this.props.submitClass}`}>
            {this.props.submitLabel}
          </button>
          <button type='button' className='btn btn-default'
            onClick={this.props.init}>Cancelar</button>
        </div>
      </form>
    )
  }
}

BillingCycleForm = reduxForm({form: 'billingCycleForm', destroyOnUnmount:
false})(BillingCycleForm)
const mapDispatchToProps = dispatch => bindActionCreators({init}, dispatch)
export default connect(null, mapDispatchToProps)(BillingCycleForm)
```

## 9.51. Evoluindo o Componente CreditList

### **Listagem 362 - Alterar arquivo *billingCycleForm.jsx***

```

import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { reduxForm, Field, formValueSelector } from 'redux-form'

import { init } from './billingCycleActions'
import LabelAndInput from '../common/form/labelAndInput'
import CreditList from './creditList'

class BillingCycleForm extends Component {

  render() {
    const { handleSubmit, readOnly, credits } = this.props
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component={LabelAndInput}
readOnly={readOnly}
          label='Nome' cols='12 4' placeholder='Informe o nome' />
          <Field name='month' component={LabelAndInput} type='number'
readOnly={readOnly}
          label='Mês' cols='12 4' placeholder='Informe o mês' />
          <Field name='year' component={LabelAndInput} type='number'
readOnly={readOnly}
          label='Ano' cols='12 4' placeholder='Informe o ano' />
          <CreditList cols='12 6' list={credits} readOnly={readOnly} />
        </div>
        <div className='box-footer'>
          <button type='submit' className={`btn btn-${
this.props.submitClass}`}>
            {this.props.submitLabel}
          </button>
          <button type='button' className='btn btn-default'
onClick={this.props.init}>Cancelar</button>
        </div>
      </form>
    )
  }
}

BillingCycleForm = reduxForm({form: 'billingCycleForm', destroyOnUnmount:
false})(BillingCycleForm)
const selector = formValueSelector('billingCycleForm')
const mapStateToProps = state => ({credits: selector(state, 'credits')})
const mapDispatchToProps = dispatch => bindActionCreators({init}, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(BillingCycleForm)

```

## Listagem 363 - Alterar arquivo creditList.jsx\_

```

import React, { Component } from 'react'
import { Field } from 'redux-form'
import Grid from '../common/layout/grid'
import Input from '../common/form/input'

class CreditList extends Component {

  renderRows() {
    const list = this.props.list || []
    return list.map((item, index) => (
      <tr key={index}>
        <td><Field name={`credits[${index}].name`} component={Input}
          placeholder='Informe o nome' readOnly={this.props.readOnly}
        /></td>

        <td><Field name={`credits[${index}].value`} component={Input}
          placeholder='Informe o valor' readOnly={this.props.readOnly}
        /></td>

        <td></td>
      </tr>
    ))
  }

  render() {
    return (
      <Grid cols={this.props.cols}>
        <fieldset>
          <legend>Créditos</legend>
          <table>
            <thead>
              <tr>
                <th>Nome</th>
                <th>Valor</th>
                <th>Ações</th>
              </tr>
            </thead>
            <tbody>
              {this.renderRows()}
            </tbody>
          </table>
        </fieldset>
      </Grid>
    )
  }
}

export default CreditList

```

## Listagem 364 - Alterar arquivo *billingCycleActions.js*

my-money-app/frontend/src/billingCycle/billingCycleActions.js (aprox. linha 7)

```
import axios from 'axios'
import { toastr } from 'react-redux-toastr'
import { reset as resetForm, initialize } from 'redux-form'
import { showTabs, selectTab } from '../common/tab/tabActions'

const BASE_URL = 'http://localhost:3003/api'
const INITIAL_VALUES = {credits: [{}]}
```

```
export function getList() {
  const request = axios.get(`${BASE_URL}/billingCycles`)
  return {
    type: 'BILLING_CYCLES_FETCHED',
    payload: request
  }
}

export function create(values) {
  return submit(values, 'post')
}

export function update(values) {
  return submit(values, 'put')
}

export function remove(values) {
  return submit(values, 'delete')
}

function submit(values, method) {
  return dispatch => {
    const id = values._id ? values._id : ''
    axios[method](`${BASE_URL}/billingCycles/${id}`, values)
      .then(resp => {
        toastr.success('Sucesso', 'Operação Realizada com sucesso.')
        dispatch(init())
      })
      .catch(e => {
        e.response.data.errors.forEach(error => toastr.error('Erro',
error))
      })
  }
}

export function showUpdate(billingCycle) {
  return [
    showTabs('tabUpdate'),
```

```

        selectTab('tabUpdate'),
        initialize('billingCycleForm', billingCycle)
    ]
}

export function init() {
    return [
        showTabs('tabList', 'tabCreate'),
        selectTab('tabList'),
        getList(),
        initialize('billingCycleForm', INITIAL_VALUES)
    ]
}

```



Remover a linha referente ao `import { selectTab, showTabs } from '../common/tab/tabActions'`



## Listagem 365 - Alterar arquivo *billingCycle.jsx*

my-money-app/frontend/src/billingCycle/billingCycle.jsx (aprox. linha 13, 20 e 62)

```
import React, { Component } from 'react'
import { bindActionCreators } from 'redux'
import { connect } from 'react-redux'

import ContentHeader from '../common/template/contentHeader'
import Content from '../common/template/content'
import Tabs from '../common/tab/tabs'
import TabsHeader from '../common/tab/tabsHeader'
import TabsContent from '../common/tab/tabsContent'
import TabHeader from '../common/tab/tabHeader'
import TabContent from '../common/tab/tabContent'
import { selectTab, showTabs } from '../common/tab/tabActions'
import { init, create, update, remove } from './billingCycleActions'

import List from './billingCycleList'
import Form from './billingCycleForm'

class BillingCycle extends Component {

  componentWillMount() {
    this.props.init()
  }

  render() {
    return (
      <div>
        <ContentHeader title='Ciclos de Pagamentos' small='Cadastro' />
        <Content>
          <Tabs>
            <TabsHeader>
              <TabHeader label='Listar' icon='bars'
target='tabList' />
              <TabHeader label='Incluir' icon='plus'
target='tabCreate' />
              <TabHeader label='Alterar' icon='pencil'
target='tabUpdate' />
              <TabHeader label='Excluir' icon='trash-o'
target='tabDelete' />
            </TabsHeader>
            <TabsContent>
              <TabContent id='tabList'>
                <List />
              </TabContent>
              <TabContent id='tabCreate'>
                <Form onSubmit={this.props.create}
                  submitLabel='Incluir' submitClass='primary'

```

```

/>
        </TabContent>
        <TabContent id='tabUpdate'>
          <Form onSubmit={this.props.update}
            submitLabel='Alterar' submitClass='info' />
        </TabContent>
        <TabContent id='tabDelete'>
          <Form onSubmit={this.props.remove} readOnly={
true}
            submitLabel='Excluir' submitClass='danger' />
        </TabContent>
      </TabsContent>
    </Tabs>
  </Content>
</div>
)
}
}

const mapDispatchToProps = dispatch => bindActionCreators({
  init, create, update, remove
}, dispatch)
export default connect(null, mapDispatchToProps)(BillingCycle)

```

## 9.52. Adicionar e Clonar Créditos

## Listagem 366 - Alterar arquivo creditList.jsx

my-money-app/frontend/src/billingCycle/creditList.jsx (aprox. linha 2, 3, 4, 9, 17, 47, 60 e 61)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { Field, arrayInsert } from 'redux-form'
import Grid from '../common/layout/grid'
import Input from '../common/form/input'

class CreditList extends Component {

  add(index, item = {}) {
    if(!this.props.readOnly) {
      this.props.arrayInsert('billingCycleForm', 'credits', index, item)
    }
  }

  renderRows() {
    const list = this.props.list || []
    return list.map((item, index) => (
      <tr key={index}>
        <td><Field name={`credits[${index}].name`} component={Input}
          placeholder='Informe o nome' readOnly={this.props.readOnly}
        /></td>

        <td><Field name={`credits[${index}].value`} component={Input}
          placeholder='Informe o valor' readOnly={this.props.readOnly}
        /></td>

        <td>
          <button type='button' className='btn btn-success'
            onClick={() => this.add(index + 1)}>
            <i className="fa fa-plus"></i>
          </button>
          <button type='button' className='btn btn-warning'
            onClick={() => this.add(index + 1, item)}>
            <i className="fa fa-clone"></i>
          </button>
        </td>
      </tr>
    ))
  }

  render() {
    return (
      <Grid cols={this.props.cols}>
        <fieldset>
          <legend>Créditos</legend>
          <table>
            <thead>
```

```

        <tr>
          <th>Nome</th>
          <th>Valor</th>
          <th className='table-actions'>Ações</th>
        </tr>
      </thead>
      <tbody>
        {this.renderRows()}
      </tbody>
    </table>
  </fieldset>
</Grid>
)
}
}

const mapDispatchToProps = dispatch => bindActionCreators({ arrayInsert },
dispatch)
export default connect(null, mapDispatchToProps)(CreditList)

```

## 9.53. Remover Créditos

## Listagem 367 - Alterar arquivo creditList.jsx

my-money-app/frontend/src/billingCycle/creditList.jsx (aprox. linha 4, 16, 39 e 70)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { Field, arrayInsert, arrayRemove } from 'redux-form'
import Grid from '../common/layout/grid'
import Input from '../common/form/input'

class CreditList extends Component {

  add(index, item = {}) {
    if(!this.props.readOnly) {
      this.props.arrayInsert('billingCycleForm', 'credits', index, item)
    }
  }

  remove(index) {
    if(!this.props.readOnly && this.props.list.length > 1) {
      this.props.arrayRemove('billingCycleForm', 'credits', index)
    }
  }

  renderRows() {
    const list = this.props.list || []
    return list.map((item, index) => (
      <tr key={index}>
        <td><Field name={`credits[${index}].name`} component={Input}
          placeholder='Informe o nome' readOnly={this.props.readOnly}
        /></td>

        <td><Field name={`credits[${index}].value`} component={Input}
          placeholder='Informe o valor' readOnly={this.props.readOnly}
        /></td>

        <td>
          <button type='button' className='btn btn-success'
            onClick={() => this.add(index + 1)}>
            <i className="fa fa-plus"></i>
          </button>
          <button type='button' className='btn btn-warning'
            onClick={() => this.add(index + 1, item)}>
            <i className="fa fa-clone"></i>
          </button>
          <button type='button' className='btn btn-danger'
            onClick={() => this.remove(index)}>
            <i className="fa fa-trash-o"></i>
          </button>
        </td>
      </tr>
    ))
  }
}
```

```

    ))
  }

  render() {
    return (
      <Grid cols={this.props.cols}>
        <fieldset>
          <legend>Créditos</legend>
          <table>
            <thead>
              <tr>
                <th>Nome</th>
                <th>Valor</th>
                <th className='table-actions'>Ações</th>
              </tr>
            </thead>
            <tbody>
              {this.renderRows()}
            </tbody>
          </table>
        </fieldset>
      </Grid>
    )
  }
}

const mapDispatchToProps = dispatch => bindActionCreators({ arrayInsert,
arrayRemove }, dispatch)
export default connect(null, mapDispatchToProps)(CreditList)

```

## 9.54. Generalizando o Componente CreditList



Renomear o arquivo `creditList.jsx` para `itemList.jsx`

## Listagem 368 - Alterar arquivo itemList.jsx\_

my-money-app/frontend/src/billingCycle/itemList.jsx (aprox. linha 8, 12, 18, 26, 28, 52 e 72)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { Field, arrayInsert, arrayRemove } from 'redux-form'
import Grid from '../common/layout/grid'
import Input from '../common/form/input'

class ItemList extends Component {

  add(index, item = {}) {
    if(!this.props.readOnly) {
      this.props.arrayInsert('billingCycleForm', this.props.field, index,
item)
    }
  }

  remove(index) {
    if(!this.props.readOnly && this.props.list.length > 1) {
      this.props.arrayRemove('billingCycleForm', this.props.field, index)
    }
  }

  renderRows() {
    const list = this.props.list || []
    return list.map((item, index) => (
      <tr key={index}>
        <td><Field name={`$${this.props.field}[$${index}].name`}
component={Input}
placeholder='Informe o nome' readOnly={this.props.readOnly}
/></td>
        <td><Field name={`$${this.props.field}[$${index}].value`}
component={Input}
placeholder='Informe o valor' readOnly={this.props.readOnly}
/></td>
        <td>
          <button type='button' className='btn btn-success'
            onClick={() => this.add(index + 1)}>
            <i className="fa fa-plus"></i>
          </button>
          <button type='button' className='btn btn-warning'
            onClick={() => this.add(index + 1, item)}>
            <i className="fa fa-clone"></i>
          </button>
          <button type='button' className='btn btn-danger'
            onClick={() => this.remove(index)}>
            <i className="fa fa-trash-o"></i>
        </td>
      </tr>
    ))
  }
}
```

```

        </button>
      </td>
    </tr>
  ))
}

render() {
  return (
    <Grid cols={this.props.cols}>
      <fieldset>
        <legend>{this.props.legend}</legend>
        <table>
          <thead>
            <tr>
              <th>Nome</th>
              <th>Valor</th>
              <th className='table-actions'>Ações</th>
            </tr>
          </thead>
          <tbody>
            {this.renderRows()}
          </tbody>
        </table>
      </fieldset>
    </Grid>
  )
}

const mapDispatchToProps = dispatch => bindActionCreators({ arrayInsert,
arrayRemove }, dispatch)
export default connect(null, mapDispatchToProps)(ItemList)

```



**Listagem 369 - Alterar arquivo *billingCycleForm.jsx***

```

import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { reduxForm, Field, formValueSelector } from 'redux-form'

import { init } from './billingCycleActions'
import LabelAndInput from '../common/form/labelAndInput'
import ItemList from './itemList'

class BillingCycleForm extends Component {

  render() {
    const { handleSubmit, readOnly, credits } = this.props
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component={LabelAndInput}
readOnly={readOnly}
          label='Nome' cols='12 4' placeholder='Informe o nome' />
          <Field name='month' component={LabelAndInput} type='number'
readOnly={readOnly}
          label='Mês' cols='12 4' placeholder='Informe o mês' />
          <Field name='year' component={LabelAndInput} type='number'
readOnly={readOnly}
          label='Ano' cols='12 4' placeholder='Informe o ano' />
          <ItemList cols='12 6' list={credits} readOnly={readOnly}
            field='credits' legend='Créditos' />
        </div>
        <div className='box-footer'>
          <button type='submit' className={`btn btn-${
this.props.submitClass}`}>
            {this.props.submitLabel}
          </button>
          <button type='button' className='btn btn-default'
            onClick={this.props.init}>Cancelar</button>
        </div>
      </form>
    )
  }
}

BillingCycleForm = reduxForm({form: 'billingCycleForm', destroyOnUnmount:
false})(BillingCycleForm)
const selector = formValueSelector('billingCycleForm')
const mapStateToProps = state => ({credits: selector(state, 'credits')})
const mapDispatchToProps = dispatch => bindActionCreators({init}, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(BillingCycleForm)

```

## 9.55. Adicionando o Campo Status no ItemList

## Listagem 370 - Alterar arquivo *billingCycleForm.jsx*

my-money-app/frontend/src/billingCycle/billingCycleForm.jsx (aprox. linha 12, 24 e 40)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { reduxForm, Field, formValueSelector } from 'redux-form'

import { init } from './billingCycleActions'
import LabelAndInput from '../common/form/labelAndInput'
import ItemList from './itemList'

class BillingCycleForm extends Component {

  render() {
    const { handleSubmit, readOnly, credits, debts } = this.props
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component={LabelAndInput}
readOnly={readOnly}
label='Nome' cols='12 4' placeholder='Informe o nome' />
          <Field name='month' component={LabelAndInput} type='number'
readOnly={readOnly}
label='Mês' cols='12 4' placeholder='Informe o mês' />
          <Field name='year' component={LabelAndInput} type='number'
readOnly={readOnly}
label='Ano' cols='12 4' placeholder='Informe o ano' />
          <ItemList cols='12 6' list={credits} readOnly={readOnly}
field='credits' legend='Créditos' />
          <ItemList cols='12 6' list={debts} readOnly={readOnly}
field='debts' legend='Débitos' showStatus={true} />
        </div>
        <div className='box-footer'>
          <button type='submit' className={`btn btn-${
this.props.submitClass}`}>
            {this.props.submitLabel}
          </button>
          <button type='button' className='btn btn-default'
onClick={this.props.init}>Cancelar</button>
        </div>
      </form>
    )
  }
}

BillingCycleForm = reduxForm({form: 'billingCycleForm', destroyOnUnmount:
false})(BillingCycleForm)
const selector = formValueSelector('billingCycleForm')
```

```
const mapStateToProps = state => ({
  credits: selector(state, 'credits'),
  debts: selector(state, 'debts')
})
const mapDispatchToProps = dispatch => bindActionCreators({init}, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(BillingCycleForm)
```

## Listagem 371 - Alterar arquivo *billingCycleActions.js*

my-money-app/frontend/src/billingCycle/billingCycleActions.js (aprox. linha 7)

```
import axios from 'axios'
import { toastr } from 'react-redux-toastr'
import { reset as resetForm, initialize } from 'redux-form'
import { showTabs, selectTab } from '../common/tab/tabActions'

const BASE_URL = 'http://localhost:3003/api'
const INITIAL_VALUES = {credits: [{}], debts: [{}]}

export function getList() {
  const request = axios.get(`${BASE_URL}/billingCycles`)
  return {
    type: 'BILLING_CYCLES_FETCHED',
    payload: request
  }
}

export function create(values) {
  return submit(values, 'post')
}

export function update(values) {
  return submit(values, 'put')
}

export function remove(values) {
  return submit(values, 'delete')
}

function submit(values, method) {
  return dispatch => {
    const id = values._id ? values._id : ''
    axios[method](`${BASE_URL}/billingCycles/${id}`, values)
      .then(resp => {
        toastr.success('Sucesso', 'Operação Realizada com sucesso.')
        dispatch(init())
      })
      .catch(e => {
        e.response.data.errors.forEach(error => toastr.error('Erro',
error))
      })
  }
}

export function showUpdate(billingCycle) {
  return [
    showTabs('tabUpdate'),
  ]
}
```

```
        selectTab('tabUpdate'),
        initialize('billingCycleForm', billingCycle)
    ]
}

export function init() {
    return [
        showTabs('tabList', 'tabCreate'),
        selectTab('tabList'),
        getList(),
        initialize('billingCycleForm', INITIAL_VALUES)
    ]
}
```

## Listagem 372 - Alterar arquivo itemList.jsx\_

my-money-app/frontend/src/billingCycle/itemList.jsx (aprox. linha 7, 31 e 62)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { Field, arrayInsert, arrayRemove } from 'redux-form'
import Grid from '../common/layout/grid'
import Input from '../common/form/input'
import If from '../common/operator/if'

class ItemList extends Component {

  add(index, item = {}) {
    if(!this.props.readOnly) {
      this.props.arrayInsert('billingCycleForm', this.props.field, index,
item)
    }
  }

  remove(index) {
    if(!this.props.readOnly && this.props.list.length > 1) {
      this.props.arrayRemove('billingCycleForm', this.props.field, index)
    }
  }

  renderRows() {
    const list = this.props.list || []
    return list.map((item, index) => (
      <tr key={index}>
        <td><Field name={`_${this.props.field}[${index}].name`}
component={Input}
placeholder='Informe o nome' readOnly={this.props.readOnly}
/></td>
        <td><Field name={`_${this.props.field}[${index}].value`}
component={Input}
placeholder='Informe o valor' readOnly={this.props.readOnly}
/></td>
        <If test={this.props.showStatus}>
          <td><Field name={`_${this.props.field}[${index}].status`}
component={Input}
placeholder='Informe o status' readOnly={
this.props.readOnly} /></td>
        </If>
        <td>
          <button type='button' className='btn btn-success'
onClick={() => this.add(index + 1)}>
            <i className="fa fa-plus"></i>
          </button>
        </td>
      </tr>
    ))
  }
}
```



```

        <button type='button' className='btn btn-warning'
            onClick={() => this.add(index + 1, item)}>
            <i className="fa fa-clone"></i>
        </button>
        <button type='button' className='btn btn-danger'
            onClick={() => this.remove(index)}>
            <i className="fa fa-trash-o"></i>
        </button>
    </td>
</tr>
))
}

render() {
    return (
        <Grid cols={this.props.cols}>
            <fieldset>
                <legend>{this.props.legend}</legend>
                <table>
                    <thead>
                        <tr>
                            <th>Nome</th>
                            <th>Valor</th>
                            <If test={this.props.showStatus}>
                                <th>Status</th>
                            </If>
                            <th className='table-actions'>Ações</th>
                        </tr>
                    </thead>
                    <tbody>
                        {this.renderRows()}
                    </tbody>
                </table>
            </fieldset>
        </Grid>
    )
}

const mapDispatchToProps = dispatch => bindActionCreators({ arrayInsert,
arrayRemove }, dispatch)
export default connect(null, mapDispatchToProps)(ItemList)

```

## 9.56. Componente Summary

## Listagem 373 - Criar arquivo summary.jsx\_

my-money-app/frontend/src/billingCycle/summary.jsx

```
import React, { Component } from 'react'

import Grid from '../common/layout/grid'
import Row from '../common/layout/row'
import ValueBox from '../common/widget/valueBox'

export default ({credit, debt}) => (
  <Grid cols='12'>
    <fieldset>
      <legend>Resumo</legend>
      <Row>
        <ValueBox cols='12 4' color='green' icon='bank'
          value={`R$ ${credit}`} text='Total de Créditos' />
        <ValueBox cols='12 4' color='red' icon='credit-card'
          value={`R$ ${debt}`} text='Total de Débitos' />
        <ValueBox cols='12 4' color='blue' icon='money'
          value={`R$ ${credit - debt}`} text='Valor Consolidado' />
      </Row>
    </fieldset>
  </Grid>
)
```

## Listagem 374 - Alterar arquivo *billingCycleForm.jsx*

my-money-app/frontend/src/billingCycle/billingCycleForm.jsx (aprox. linha 9 e 24)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { reduxForm, Field, formValueSelector } from 'redux-form'

import { init } from './billingCycleActions'
import LabelAndInput from '../common/form/labelAndInput'
import ItemList from './itemList'
import Summary from './summary'

class BillingCycleForm extends Component {

  render() {
    const { handleSubmit, readOnly, credits, debts } = this.props
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component={LabelAndInput}
readOnly={readOnly}
label='Nome' cols='12 4' placeholder='Informe o nome' />
          <Field name='month' component={LabelAndInput} type='number'
readOnly={readOnly}
label='Mês' cols='12 4' placeholder='Informe o mês' />
          <Field name='year' component={LabelAndInput} type='number'
readOnly={readOnly}
label='Ano' cols='12 4' placeholder='Informe o ano' />
          <Summary credit={1000} debt={100} />
          <ItemList cols='12 6' list={credits} readOnly={readOnly}
field='credits' legend='Créditos' />
          <ItemList cols='12 6' list={debts} readOnly={readOnly}
field='debts' legend='Débitos' showStatus={true} />
        </div>
        <div className='box-footer'>
          <button type='submit' className={`btn btn-${
this.props.submitClass}`}>
            {this.props.submitLabel}
          </button>
          <button type='button' className='btn btn-default'
onClick={this.props.init}>Cancelar</button>
        </div>
      </form>
    )
  }
}

BillingCycleForm = reduxForm({form: 'billingCycleForm', destroyOnUnmount:
```

```
false}})(BillingCycleForm)
const selector = formValueSelector('billingCycleForm')
const mapStateToProps = state => ({
  credits: selector(state, 'credits'),
  debts: selector(state, 'debts')
})
const mapDispatchToProps = dispatch => bindActionCreators({init}, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(BillingCycleForm)
```

## 9.57. Integrar Soma de Créditos e Débitos

## Listagem 375 - Alterar arquivo *billingCycleForm.jsx*

my-money-app/frontend/src/billingCycle/billingCycleForm.jsx (aprox. linha 13, 23 e 33)

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { reduxForm, Field, formValueSelector } from 'redux-form'

import { init } from './billingCycleActions'
import LabelAndInput from '../common/form/labelAndInput'
import ItemList from './itemList'
import Summary from './summary'

class BillingCycleForm extends Component {

  calculateSummary() {
    const sum = (t, v) => t + v
    return {
      sumOfCredits: this.props.credits.map(c => +c.value || 0).reduce(sum),
      sumOfDebts: this.props.debts.map(d => +d.value || 0).reduce(sum)
    }
  }

  render() {
    const { handleSubmit, readOnly, credits, debts } = this.props
    const { sumOfCredits, sumOfDebts } = this.calculateSummary()
    return (
      <form role='form' onSubmit={handleSubmit}>
        <div className='box-body'>
          <Field name='name' component={LabelAndInput}
readOnly={readOnly}
            label='Nome' cols='12 4' placeholder='Informe o nome' />
          <Field name='month' component={LabelAndInput} type='number'
readOnly={readOnly}
            label='Mês' cols='12 4' placeholder='Informe o mês' />
          <Field name='year' component={LabelAndInput} type='number'
readOnly={readOnly}
            label='Ano' cols='12 4' placeholder='Informe o ano' />
          <Summary credit={sumOfCredits} debt={sumOfDebts} />
          <ItemList cols='12 6' list={credits} readOnly={readOnly}
            field='credits' legend='Créditos' />
          <ItemList cols='12 6' list={debts} readOnly={readOnly}
            field='debts' legend='Débitos' showStatus={true} />
        </div>
        <div className='box-footer'>
          <button type='submit' className={`btn btn-${
this.props.submitClass}`}>
            {this.props.submitLabel}
          </button>
        </div>
      </form>
    )
  }
}
```

```

        <button type='button' className='btn btn-default'
            onClick={this.props.init}>Cancelar</button>
    </div>
</form>
    )
}
}

BillingCycleForm = reduxForm({form: 'billingCycleForm', destroyOnUnmount:
false})(BillingCycleForm)
const selector = formValueSelector('billingCycleForm')
const mapStateToProps = state => ({
    credits: selector(state, 'credits'),
    debts: selector(state, 'debts')
})
const mapDispatchToProps = dispatch => bindActionCreators({init}, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(BillingCycleForm)

```

# 10. Melhorias e Correções

## 10.1. Problema de Responsividade do Menu

### Listagem 376 - Alterar arquivo *index.html*

*my-money-app/frontend/public/index.html* (aprox. linha 10)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset='utf-8'>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <title>My Money</title>
    <link rel='stylesheet' href='app.css'>
  </head>
  <body class='skin-blue fixed sidebar-mini'>
    <div id="app" class="wrapper"></div>
    <script src='app.js'></script>
  </body>
</html>
```

### Listagem 377 - Alterar arquivo *menuItem.jsx*

*my-money-app/frontend/src/common/template/menuItem.jsx* (aprox. linha 6)

```
import React from 'react'

export default props => (
  <li>
    <a href={props.path}>
      <i className={`fa fa-${props.icon}`}></i> <span>{props.label}</span>
    </a>
  </li>
)
```

### Listagem 378 - Alterar arquivo menuTree.jsx

my-money-app/frontend/src/common/template/menuTree.jsx (aprox. linha 6)

```
import React from 'react'

export default props => (
  <li className='treeview'>
    <a href>
      <i className={`fa fa-${props.icon}`}></i> <span>{props.label}</span>
      <i className='fa fa-angle-left pull-right'></i>
    </a>
    <ul className='treeview-menu'>
      {props.children}
    </ul>
  </li>
)
```

## 10.2. Navegação com React-router (Versão 2)

### Listagem 379 - Alterar arquivo menuItem.jsx

my-money-app/frontend/src/common/template/menuItem.jsx (aprox. linha 2 e 6)

```
import React from 'react'
import { Link } from 'react-router'

export default props => (
  <li>
    <Link to={props.path}>
      <i className={`fa fa-${props.icon}`}></i> <span>{props.label}</span>
    </Link>
  </li>
)
```



## Listagem 380 - Alterar arquivo routes.jsx

my-money-app/frontend/src/main/routes.jsx (aprox. linha 2, 4, 10 e 13)

```
import React from 'react'
import { Router, Route, IndexRoute, Redirect, hashHistory } from 'react-router'

import App from './app'
import Dashboard from '../dashboard/dashboard'
import BillingCycle from '../billingCycle/billingCycle'

export default props => (
  <Router history={hashHistory}>
    <Route path="/" component={App}>
      <IndexRoute component={Dashboard} />
      <Route path='billingCycles' component={BillingCycle} />
    </Route>
    <Redirect from='*' to="/" />
  </Router>
)
```

## Listagem 381 - Alterar arquivo app.jsx

my-money-app/frontend/src/main/app.jsx (aprox. linha 7 e 15)

```
import '../common/template/dependencies'
import React from 'react'

import Header from '../common/template/header'
import SideBar from '../common/template/sideBar'
import Footer from '../common/template/footer'
import Messages from '../common/msg/messages'

export default props => (
  <div className='wrapper'>
    <Header />
    <SideBar />
    <div className='content-wrapper'>
      {props.children}
    </div>
    <Footer />
    <Messages />
  </div>
)
```

## Listagem 382 - Alterar arquivo index.jsx

my-money-app/frontend/src/index.jsx (aprox. linha 10 e 18)

```
import React from 'react'
import ReactDOM from 'react-dom'
import { applyMiddleware, createStore } from 'redux'
import { Provider } from 'react-redux'

import promise from 'redux-promise'
import multi from 'redux-multi'
import thunk from 'redux-thunk'

import Routes from './main/routes'
import reducers from './main/reducers'

const devTools = window.__REDUX_DEVTOOLS_EXTENSION__
  && window.__REDUX_DEVTOOLS_EXTENSION__()
const store = applyMiddleware(multi, thunk, promise)(createStore)(reducers,
devTools)
ReactDOM.render(
  <Provider store={store}>
    <Routes />
  </Provider>
, document.getElementById('app'))
```

## Listagem 383 - Alterar arquivo menu.jsx

my-money-app/frontend/src/common/template/menu.jsx (aprox. linha 7 e 9)

```
import React from 'react'
import MenuItem from './menuItem'
import MenuTree from './menuTree'

export default props => (
  <ul className='sidebar-menu'>
    <MenuItem path='/' label='Dashboard' icon='dashboard' />
    <MenuTree label='Cadastro' icon='edit'>
      <MenuItem path='billingCycles'
        label='Ciclos de Pagamentos' icon='usd' />
    </MenuTree>
  </ul>
)
```

# 11. My Money APP: Autenticação

## 11.1. Configurar Ambiente

Criei uma nova pasta e no terminal dentro da pasta execute o seguinte comando:

### Listagem 1 - Baixando Repositório

```
git clone -b before_auth https://github.com/cod3rcursos/curso-react-redux
```

## 11.2. Backend: Novas Dependências



Iremos adicionar novas dependências ao projeto usando **versões específicas** para garantir compatibilidade. As mudanças importantes nas versões dos módulos adicionados serão tratadas em vídeos adicionais no final do curso.

Novas dependências são:

- bcrypt@1.0.2
- jsonwebtoken@7.3.0

### 11.2.1. Instalação

Abra o terminal e dentro da pasta do projeto backend execute o seguinte comando:

### Listagem 2 - Instalando as dependências

```
npm i --save bcrypt@1.0.2 jsonwebtoken@7.3.0
```

## 11.3. Backend: Implementar Autenticação

### 11.3.1. Criar arquivo .env



O objetivo principal do **.env** é armazenar variáveis de configuração da sua aplicação *backend*, por enquanto será armazenando apenas a chave usada para gerar o token **JWT**.

### Listagem 3 - Criar arquivo .env

my-money-app/backend/src/.env

```
module.exports = {  
  // Você pode alterar essa chave!  
  authSecret: 'skjdhf6$$$dojkhf^(sdkjhf'  
}
```



É muito importante que o arquivo `.env` não seja commitado no repositório, pois nele está a chave secreta para geração do token.

### Listagem 4 - Adicionar .env ao .gitignore

my-money-app/backend/.gitignore

```
node_modules  
*.log  
.env
```

## 11.3.2. Criar arquivo `user.js`



Criar uma nova pasta ao projeto `backend` chamada `user` dentro de `src/api`.



O objetivo principal do `user.js` é fazer o **ODM** (Mapeamento Objeto-Documento), ou seja, mapear o objeto javascript `user` para o documento que será armazenado no MongoDB.

### Listagem 5 - Criar arquivo `user.js`

my-money-app/backend/src/api/user/user.js

```
const restful = require('node-restful')  
const mongoose = restful.mongoose  
  
const userSchema = new mongoose.Schema({  
  name: { type: String, required: true },  
  email: { type: String, required: true },  
  password: { type: String, min: 6, max: 12, required: true }  
})  
  
module.exports = restful.model('User', userSchema)
```

### 11.3.3. Criar arquivo *authService.js*

#### Listagem 6 - Estrutura básica de *authService.js*

my-money-app/backend/src/api/user/authService.js

```
const _ = require('lodash')
const jwt = require('jsonwebtoken')
const bcrypt = require('bcrypt')
const User = require('./user')
const env = require('../../../.env')

const emailRegex = /\S+@\S+\.\S+/
const passwordRegex = /((?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[@#$%]).{6,20})/

const sendErrorsFromDB = (res, dbErrors) => {
  const errors = []
  _.forIn(dbErrors.errors, error => errors.push(error.message))
  return res.status(400).json({errors})
}
```

#### Listagem 7 - Criar método login de *authService.js*

my-money-app/backend/src/api/user/authService.js

```
const login = (req, res, next) => {
  const email = req.body.email || ''
  const password = req.body.password || ''

  User.findOne({email}, (err, user) => {
    if(err) {
      return sendErrorsFromDB(res, err)
    } else if (user && bcrypt.compareSync(password, user.password)) {
      const token = jwt.sign(user, env.authSecret, {
        expiresIn: "1 day"
      })
      const { name, email } = user
      res.json({ name, email, token })
    } else {
      return res.status(400).send({errors: ['Usuário/Senha inválidos']})
    }
  })
}
```

## Listagem 8 - Criar método `validateToken` de `authService.js`

`my-money-app/backend/src/api/user/authService.js`

```
const validateToken = (req, res, next) => {  
  const token = req.body.token || ''  
  jwt.verify(token, env.authSecret, function(err, decoded) {  
    return res.status(200).send({valid: !err})  
  })  
}
```

## Listagem 9 - Criar método signup de authService.js

my-money-app/backend/src/api/user/authService.js

```
const signup = (req, res, next) => {
  const name = req.body.name || ''
  const email = req.body.email || ''
  const password = req.body.password || ''
  const confirmPassword = req.body.confirm_password || ''

  if(!email.match(emailRegex)) {
    return res.status(400).send({errors: ['O e-mail informado está inválido']})
  }

  if(!password.match(passwordRegex)) {
    return res.status(400).send({errors: [
      "Senha precisar ter: uma letra maiúscula, uma letra minúscula, um número, uma caractere especial(@#$%) e tamanho entre 6-20."
    ]})
  }

  const salt = bcrypt.genSaltSync()
  const passwordHash = bcrypt.hashSync(password, salt)
  if(!bcrypt.compareSync(confirmPassword, passwordHash)) {
    return res.status(400).send({errors: ['Senhas não conferem.']})
  }

  User.findOne({email}, (err, user) => {
    if(err) {
      return sendErrorsFromDB(res, err)
    } else if (user) {
      return res.status(400).send({errors: ['Usuário já cadastrado.']})
    } else {
      const newUser = new User({ name, email, password: passwordHash })
      newUser.save(err => {
        if(err) {
          return sendErrorsFromDB(res, err)
        } else {
          login(req, res, next)
        }
      })
    }
  })
}
```

## Listagem 10 - Exportar os métodos de authService.js

my-money-app/backend/src/api/user/authService.js

```
module.exports = { login, signup, validateToken }
```

### 11.3.4. Criar middleware auth.js



Esse *middleware* será o responsável por validar o token *JWT* para as rotas protegidas e garantir que a API esteja protegida.

## Listagem 11 - Criar middleware auth.js

my-money-app/backend/src/config/auth.js

```
const jwt = require('jsonwebtoken')
const env = require('.../.env')

module.exports = (req, res, next) => {

  // CORS preflight request
  if(req.method === 'OPTIONS') {
    next()
  } else {
    const token = req.body.token || req.query.token ||
req.headers['authorization']

    if(!token) {
      return res.status(403).send({errors: ['No token provided.']})
    }

    jwt.verify(token, env.authSecret, function(err, decoded) {
      if(err) {
        return res.status(403).send({
          errors: ['Failed to authenticate token.']
        })
      } else {
        req.decoded = decoded
        next()
      }
    })
  }
}
```



## Listagem 12 - Adicionar header em cors.js

my-money-app/backend/src/config/cors.js

```
module.exports = (req, res, next) => {  
  res.header('Access-Control-Allow-Origin', '*')  
  res.header('Access-Control-Allow-Methods',  
    'GET, POST, OPTIONS, PUT, PATCH, DELETE')  
  res.header('Access-Control-Allow-Headers',  
    'Origin, X-Requested-With, Content-Type, Accept, Authorization')  
  next()  
}
```

### 11.3.5. Alterar arquivo *routes.js*



Além de cadastrar as novas rotas *login*, *signup* e *validateToken*, essa alteração tem por objetivo separar as rotas públicas das rotas privadas (acesso com autenticação).

## Listagem 13 - Alterar arquivo routes.js

my-money-app/backend/src/config/routes.js

```
const express = require('express')
const auth = require('./auth')

module.exports = function(server) {

  /*
   * Rotas protegidas por Token JWT
   */
  const protectedApi = express.Router()
  server.use('/api', protectedApi)

  protectedApi.use(auth)

  const BillingCycle = require('../api/billingCycle/billingCycleService')
  BillingCycle.register(protectedApi, '/billingCycles')

  /*
   * Rotas abertas
   */
  const openApi = express.Router()
  server.use('/oapi', openApi)

  const AuthService = require('../api/user/AuthService')
  openApi.post('/login', AuthService.login)
  openApi.post('/signup', AuthService.signup)
  openApi.post('/validateToken', AuthService.validateToken)
}
```

## 11.4. Frontend: Implementar Autenticação

### 11.4.1. Criar arquivo *authReducer.js*

### Listagem 14 - Criar arquivo *authReducer.js*

my-money-app/frontend/src/auth/authReducer.js

```
const userKey = '_mymoney_user'
const INITIAL_STATE = {
  user: JSON.parse(localStorage.getItem(userKey)),
  validToken: false
}

export default (state = INITIAL_STATE, action) => {
  switch (action.type) {
    case 'TOKEN_VALIDATED':
      if(action.payload) {
        return { ...state, validToken: true }
      } else {
        localStorage.removeItem(userKey)
        return { ...state, validToken: false, user: null }
      }
    case 'USER_FETCHED':
      localStorage.setItem(userKey, JSON.stringify(action.payload))
      return { ...state, user: action.payload, validToken: true }
    default:
      return state
  }
}
```

#### 11.4.2. Atualizar *reducers.js*

### Listagem 15 - Adicionar *authReducer* ao *rootReducer*

my-money-app/frontend/src/main/reducers.js (aprox. linha 8)

```
import AuthReducer from '../auth/authReducer'
```

my-money-app/frontend/src/main/reducers.js (aprox. linha 16)

```
auth: AuthReducer
```

#### 11.4.3. Criar arquivo *consts.js*

## Listagem 16 - Criar arquivo consts.js

my-money-app/frontend/src/consts.js

```
export default {  
  API_URL: 'http://localhost:3003/api',  
  OAPI_URL: 'http://localhost:3003/oapi',  
}
```

### 11.4.4. Criar arquivo authActions.js

## Listagem 17 - Criar arquivo authActions.js

my-money-app/frontend/src/auth/authActions.js

```
import { toastr } from 'react-redux-toastr'  
import axios from 'axios'  
import consts from '../consts'  
  
export function login(values) {  
  return submit(values, `${consts.OAPI_URL}/login`)  
}  
  
export function signup(values) {  
  return submit(values, `${consts.OAPI_URL}/signup`)  
}  
  
function submit(values, url) {  
  return dispatch => {  
    axios.post(url, values)  
      .then(resp => {  
        dispatch([  
          {type: 'USER_FETCHED', payload: resp.data}  
        ])  
      })  
      .catch(e => {  
        e.response.data.errors.forEach(  
          error => toastr.error('Erro', error)  
        )  
      })  
  }  
}
```

## Listagem 18 - Adicionar novas actions à *authActions.js*

my-money-app/frontend/src/auth/authActions.js

```
export function logout() {
  return {type: 'TOKEN_VALIDATED', payload: false}
}

export function validateToken(token) {
  return dispatch => {
    if(token) {
      axios.post(`${consts.OAPI_URL}/validateToken`, {token})
        .then(resp => {
          dispatch({type: 'TOKEN_VALIDATED', payload: resp.data.valid})
        })
        .catch(e => dispatch({type: 'TOKEN_VALIDATED', payload: false}))
    } else {
      dispatch({type: 'TOKEN_VALIDATED', payload: false})
    }
  }
}
```

### 11.4.5. Criar componente *navbar*

## Listagem 19 - Criar componente navbar.jsx

my-money-app/frontend/src/common/template/navbar.jsx

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { logout } from '../../../auth/authActions'

class Navbar extends Component {
  constructor(props) {
    super(props)
    this.state = { open: false }
  }

  changeOpen() {
    this.setState({ open: !this.state.open })
  }

  render() {
    const { name, email } = this.props.user
    return (

    )
  }
}

const mapStateToProps = state => ({user: state.auth.user})
const mapDispatchToProps = dispatch => bindActionCreators({ logout }, dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(Navbar)
```

## Listagem 20 - Implementar a estrutura do navbar (JSX)

my-money-app/frontend/src/common/template/navbar.jsx

```
<div className="navbar-custom-menu">
  <ul className="nav navbar-nav">
    <li onMouseLeave={() => this.changeOpen()}
      className={`dropdown user user-menu ${this.state.open ? 'open' :
        ''}}`>
      <a href="javascript:;" onClick={() => this.changeOpen()}
        aria-expanded={this.state.open ? 'true' : 'false'}
        className="dropdown-toggle"
        data-toggle="dropdown">
        
        <span className="hidden-xs">{name}</span>
      </a>
      <ul className="dropdown-menu">
        <li className="user-header">
          
          <p>{name}<small>{email}</small></p>
        </li>
        <li className="user-footer">
          <div className="pull-right">
            <a href="#" onClick={this.props.logout}
              className="btn btn-default btn-flat">Sair</a>
          </div>
        </li>
      </ul>
    </li>
  </ul>
</div>
```

## Listagem 21 - Adicionar componente navbar ao header

my-money-app/frontend/src/common/template/header.jsx (aprox. linha 2)

```
import Navbar from './navbar'
```

my-money-app/frontend/src/common/template/header.jsx (aprox. linha 15)

```
<Navbar />
```

### 11.4.6. Criar componente *inputAuth*

#### Listagem 22 - Criar componente *inputAuth.jsx*

my-money-app/frontend/src/common/form/inputAuth.jsx

```
import React from 'react'
import If from '../operator/if'

export default props => (
  <If test={!props.hide}>
    <div className="form-group has-feedback">
      <input {...props.input}
        className='form-control'
        placeholder={props.placeholder}
        readOnly={props.readOnly}
        type={props.type} />
      <span className={`glyphicon glyphicon-${props.icon}
        form-control-feedback`}></span>
    </div>
  </If>
)
```

### 11.4.7. Criar componente *auth.css*

#### Listagem 23 - Criar css do componente *auth*

my-money-app/frontend/src/auth/auth.css

```
.wrapper {
  background-color: #fff!important;
}

.login-box-body {
  background-color: #eee;
}

.login-box button {
  margin-left: 0px;
}

.login-box a:hover {
  cursor: pointer;
}
```



### 11.4.8. Criar componente *auth.jsx*

#### Listagem 24 - Criar componente *auth.jsx*

my-money-app/frontend/src/auth/auth.jsx

```
import './auth.css'
import React, { Component } from 'react'
import { reduxForm, Field } from 'redux-form'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'

import { login, signup } from './authActions'
import Row from '../common/layout/row'
import Grid from '../common/layout/grid'
import If from '../common/operator/if'
import Messages from '../common/msg/messages'
import Input from '../common/form/inputAuth'

class Auth extends Component {
  constructor(props) {
    super(props)
    this.state = { loginMode: true }
  }

  changeMode() {
    this.setState({ loginMode: !this.state.loginMode })
  }

  onSubmit(values) {
    const { login, signup } = this.props
    this.state.loginMode ? login(values) : signup(values)
  }

  render() {
    const { loginMode } = this.state
    const { handleSubmit } = this.props
    return (
      )
    )
  }
}

Auth = reduxForm({ form: 'authForm' })(Auth)
const mapDispatchToProps = dispatch => bindActionCreators({ login, signup },
dispatch)
export default connect(null, mapDispatchToProps)(Auth)
```

## Listagem 25 - Implementar a estrutura do auth (JSX)

my-money-app/frontend/src/auth/auth.jsx

```
<div className="login-box">
  <div className="login-logo"><b> My</b> Money</div>
  <div className="login-box-body">
    <p className="login-box-msg">Bem vindo!</p>
    <form onSubmit={handleSubmit(v => this.onSubmit(v))}>
      <Field component={Input} type="input" name="name"
        placeholder="Nome" icon='user' hide={loginMode} />
      <Field component={Input} type="email" name="email"
        placeholder="E-mail" icon='envelope' />
      <Field component={Input} type="password" name="password"
        placeholder="Senha" icon='lock' />
      <Field component={Input} type="password" name="confirm_password"
        placeholder="Confirmar Senha" icon='lock' hide={loginMode} />
      <Row>
        <Grid cols="4">
          <button type="submit"
            className="btn btn-primary btn-block btn-flat">
            {loginMode ? 'Entrar' : 'Registrar'}
          </button>
        </Grid>
      </Row>
    </form>
    <br />
    <a onClick={() => this.changeMode()}>
      {loginMode ? 'Novo usuário? Registrar aqui!' :
        'Já é cadastrado? Entrar aqui!'}
    </a>
  </div>
  <Messages />
</div>
```

### 11.4.9. Criar componente *authOrApp*

## Listagem 26 - Criar componente authOrApp.jsx

my-money-app/frontend/src/main/authOrApp.jsx

```
import '../common/template/dependencies'
import React, { Component } from 'react'
import axios from 'axios'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'

import App from './app'
import Auth from '../auth/auth'
import { validateToken } from '../auth/authActions'

class AuthOrApp extends Component {

  componentWillMount() {
    if(this.props.auth.user) {
      this.props.validateToken(this.props.auth.user.token)
    }
  }

  render() {
    const { user, validToken } = this.props.auth

    if(user && validToken) {
      axios.defaults.headers.common['authorization'] = user.token
      return <App>{this.props.children}</App>
    } else if(!user && !validToken) {
      return <Auth />
    } else {
      return false
    }
  }
}

const mapStateToProps = state => ({ auth: state.auth })
const mapDispatchToProps = dispatch => bindActionCreators({ validateToken },
dispatch)
export default connect(mapStateToProps, mapDispatchToProps)(AuthOrApp)
```

### 11.4.10. Remover o import das dependências de *app*



Esse import foi **transferido** para o componente AuthOrApp.

### Listagem 27 - Remover a seguinte linha do componente *app.jsx*

my-money-app/frontend/src/main/app.jsx

```
import '../common/template/dependencies'
```

## 11.4.11. Alterar componente *routes*

### Listagem 28 - Alterar componente *routes.jsx*

my-money-app/frontend/src/main/routes.jsx (aprox. linha 4)

```
import AuthOrApp from './authOrApp'
```

my-money-app/frontend/src/main/routes.jsx (aprox. linha 10)

```
<Route path="/" component={AuthOrApp}>
```

# Appendix A: Tabela de Códigos

- Listagem 1 - *Baixando Repositório*
- Listagem 2 - *Instalando as dependências*
- Listagem 3 - *Criar arquivo .env*
- Listagem 4 - *Adicionar .env ao .gitignore*
- Listagem 5 - *Criar arquivo user.js*
- Listagem 6 - *Estrutura básica de authService.js*
- Listagem 7 - *Criar método login de authService.js*
- Listagem 8 - *Criar método validateToken de authService.js*
- Listagem 9 - *Criar método signup de authService.js*
- Listagem 10 - *Exportar os métodos de authService.js*
- Listagem 11 - *Criar middleware auth.js*
- Listagem 12 - *Adicionar header em cors.js*
- Listagem 13 - *Alterar arquivo routes.js*
- Listagem 19 - *Criar componente navbar.jsx*
- Listagem 20 - *Implementar a estrutura do navbar (JSX)*
- Listagem 21 - *Adicionar componente navbar ao header*
- Listagem 22 - *Criar componente inputAuth.jsx*
- Listagem 16 - *Criar arquivo consts.js*
- Listagem 17 - *Criar arquivo authActions.js*
- Listagem 18 - *Adicionar novas actions à authActions.js*
- Listagem 14 - *Criar arquivo authReducer.js*
- Listagem 23 - *Criar css do componente auth*
- Listagem 24 - *Criar componente auth.jsx*
- Listagem 25 - *Implementar a estrutura do auth (JSX)*
- Listagem 15 - *Adicionar authReducer ao rootReducer*
- Listagem 26 - *Criar componente authOrApp.jsx*
- Listagem 27 - *Remover a seguinte linha do componente app.jsx*
- Listagem 28 - *Alterar componente routes.jsx*

# Glossário

JWT

...

Middleware

...