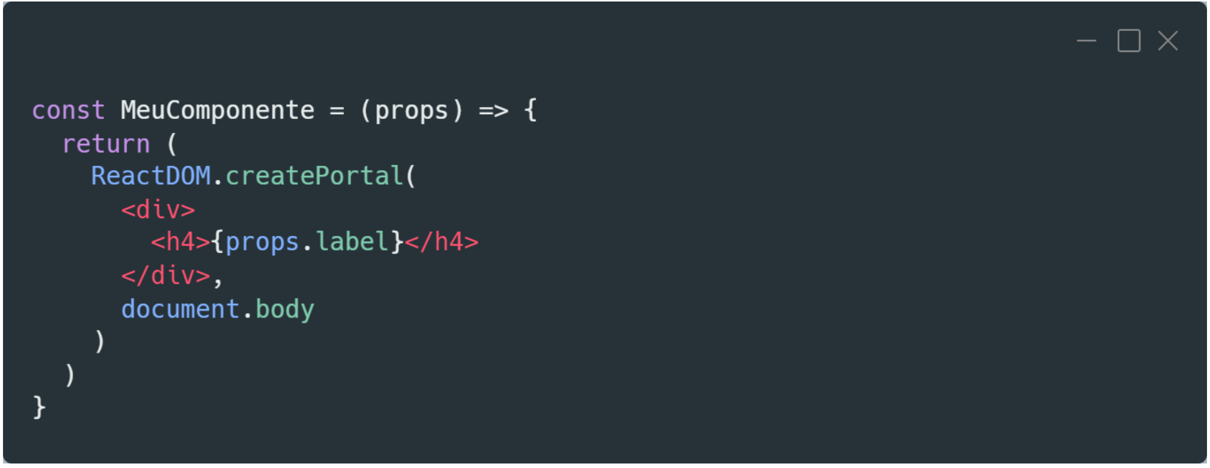
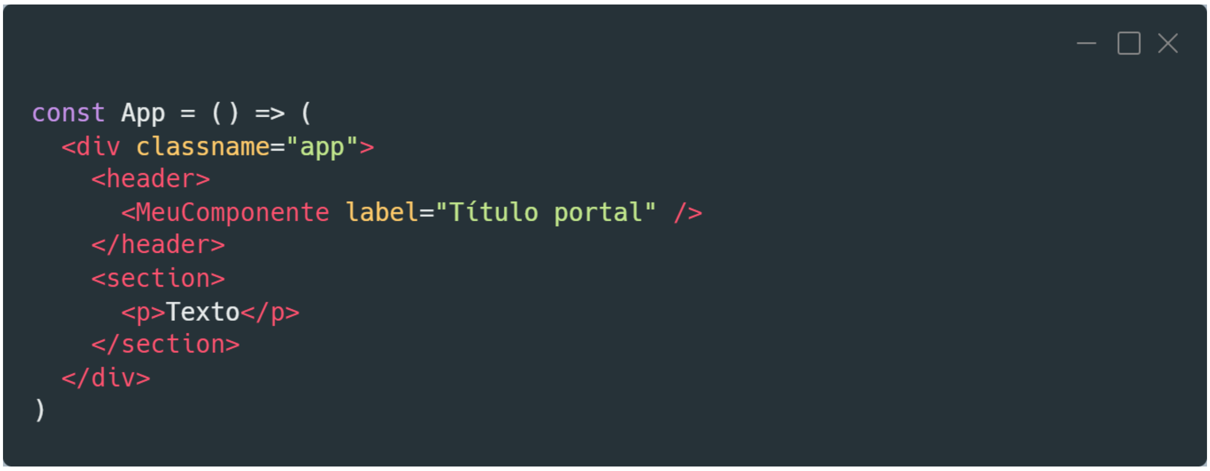


```
const MeuComponente = (props) => {
  return (
    ReactDOM.createPortal(
      <div>
        <h4>{props.label}</h4>
      </div>,
      document.body
    )
  )
}
```

```
const MeuComponente = (props) => {
  return (
    ReactDOM.createPortal(
      <div>
        <h4>{props.label}</h4>
      </div>,
      document.body
    )
  )
}
```

—

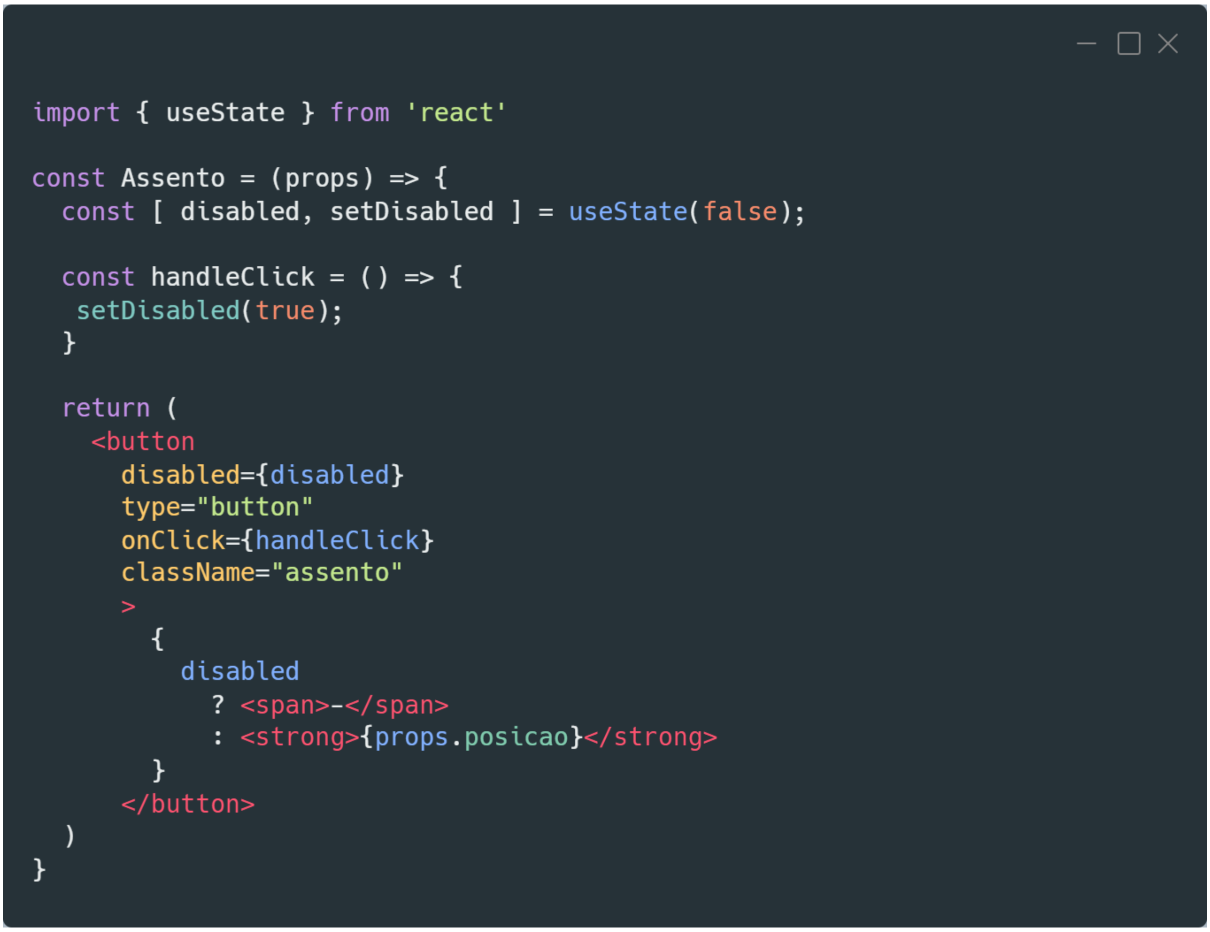


```
const App = () => (
  <div classname="app">
    <header>
      <MeuComponente label="Título portal" />
    </header>
    <section>
      <p>Texto</p>
    </section>
  </div>
)
```

```
const App = () => (
  <div classname="app">
    <header>
      <MeuComponente label="Título portal" />
    </header>
    <section>
      <p>Texto</p>
    </section>
  </div>
```

)

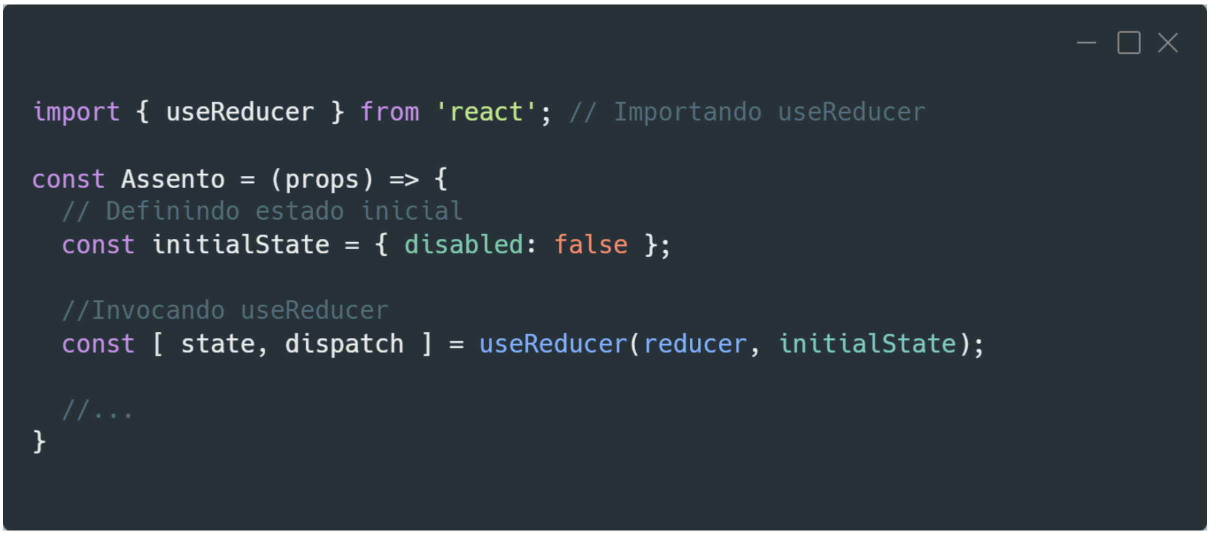
—



```
import { useState } from 'react'

const Assento = (props) => {
  const [ disabled, setDisabled ] = useState(false);

  const handleClick = () => {
   setDisabled(true);
  }

  return (
    <button
      disabled={disabled}
      type="button"
      onClick={handleClick}
      className="assento"
      >
        {
          disabled
            ? <span>-</span>
            : <strong>{props.posicao}</strong>
        }
      </button>
  )
}
```

```
import { useState } from 'react'

const Assento = (props) => {
  const [ disabled, setDisabled ] = useState(false);

  const handleClick = () => {
   setDisabled(true);
  }

  return (
    <button
      disabled={disabled}
      type="button"
      onClick={handleClick}
      className="assento"
      >
        {
          disabled
            ? <span>-</span>
          : <strong>{props.posicao}</strong>
      }
    </button>
  )
}
```

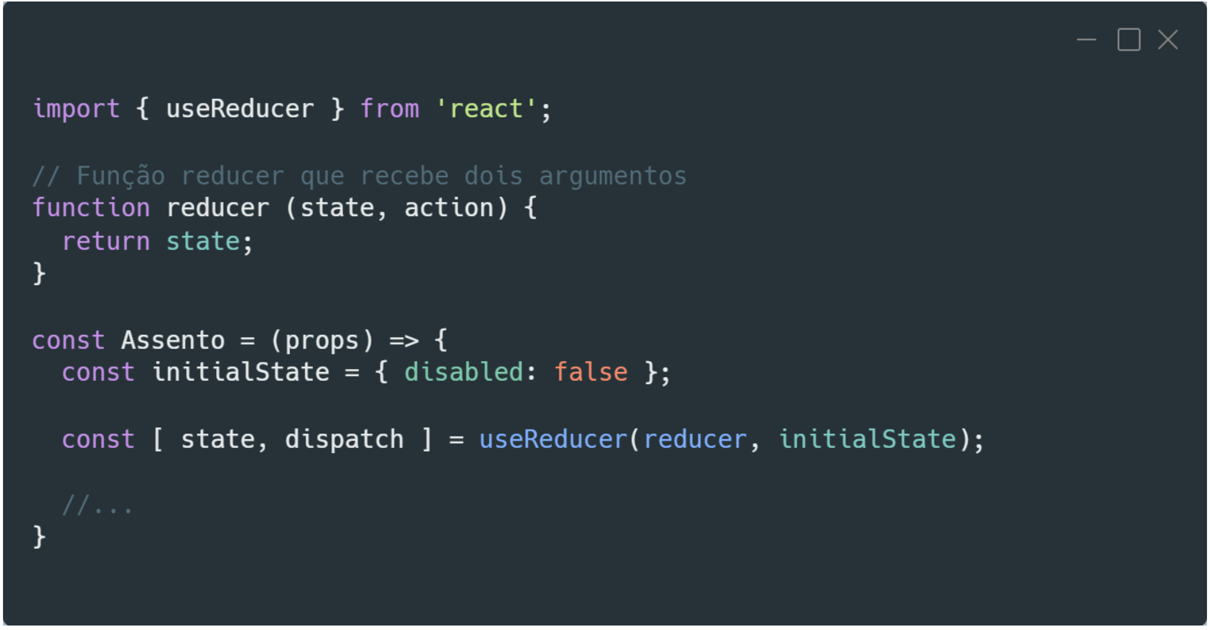
—



```
import { useReducer } from 'react'; // Importando useReducer

const Assento = (props) => {
  // Definindo estado inicial
  const initialState = { disabled: false };

  //Invocando useReducer
  const [ state, dispatch ] = useReducer(reducer, initialState);

  //...
}
```

```
import { useReducer } from 'react'; // Importando useReducer

const Assento = (props) => {
  // Definindo estado inicial
  const initialState = { disabled: false };

  //Invocando useReducer
  const [ state, dispatch ] = useReducer(reducer, initialState);

  //...
}
```

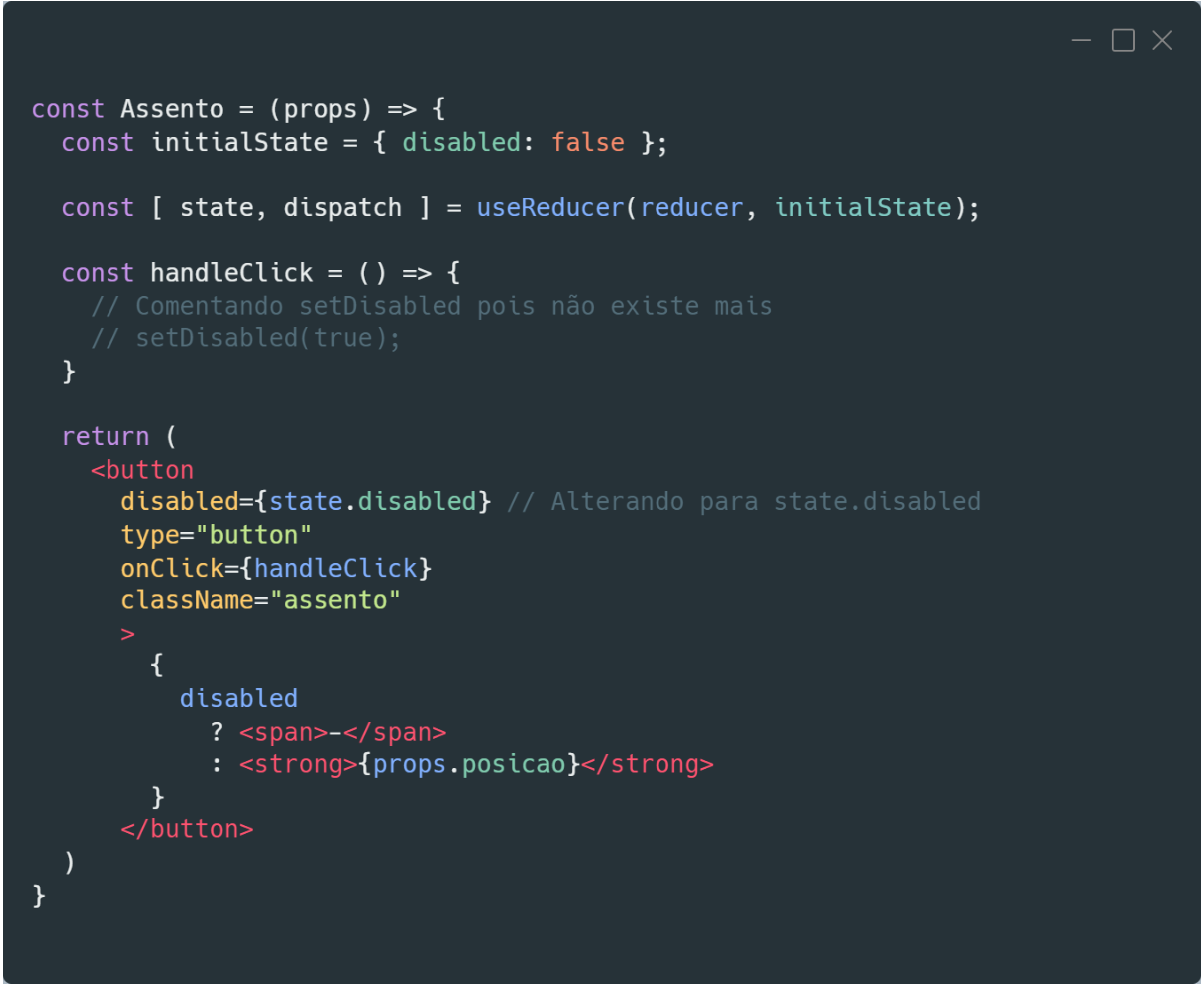
—



```
import { useReducer } from 'react';

// Função reducer que recebe dois argumentos
function reducer (state, action) {
  return state;
}

const Assento = (props) => {
  const initialState = { disabled: false };

  const [ state, dispatch ] = useReducer(reducer, initialState);

  //...
}
```

```
import { useReducer } from 'react';

// Função reducer que recebe dois argumentos
function reducer (state, action) {
  return state;
}

const Assento = (props) => {
  const initialState = { disabled: false };

  const [ state, dispatch ] = useReducer(reducer, initialState);

  //...
}
```

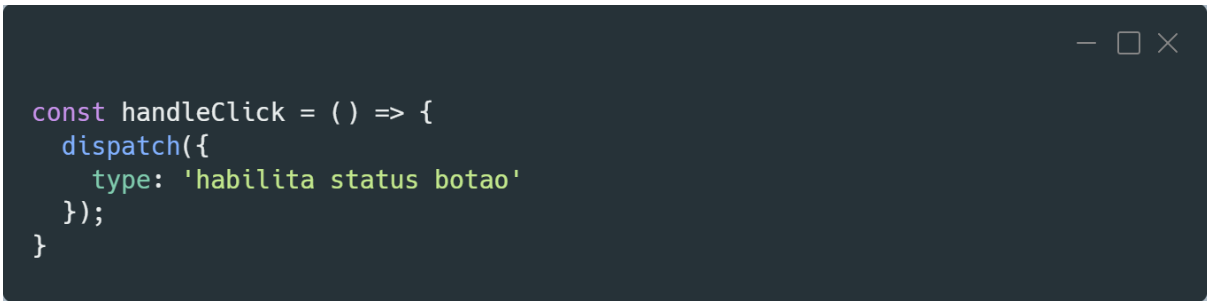
—



```
const Assento = (props) => {
  const initialState = { disabled: false };

  const [ state, dispatch ] = useReducer(reducer, initialState);

  const handleClick = () => {
    // Comentando setDisabled pois não existe mais
    // setDisabled(true);
  }

  return (
    <button
      disabled={state.disabled} // Alterando para state.disabled
      type="button"
      onClick={handleClick}
      className="assento"
      >
        {
          disabled
            ? <span>-</span>
            : <strong>{props.posicao}</strong>
        }
      </button>
  )
}
```

```
const Assento = (props) => {
  const initialState = { disabled: false };

  const [ state, dispatch ] = useReducer(reducer, initialState);

  const handleClick = () => {
    // Comentando setDisabled pois não existe mais
    // setDisabled(true);
  }

  return (
    <button
      disabled={state.disabled} // Alterando para state.disabled
      type="button"
      onClick={handleClick}
      className="assento"
      >
        {
          disabled
            ? <span>-</span>
            : <strong>{props.posicao}</strong>
      }
    </button>
  )
}
```

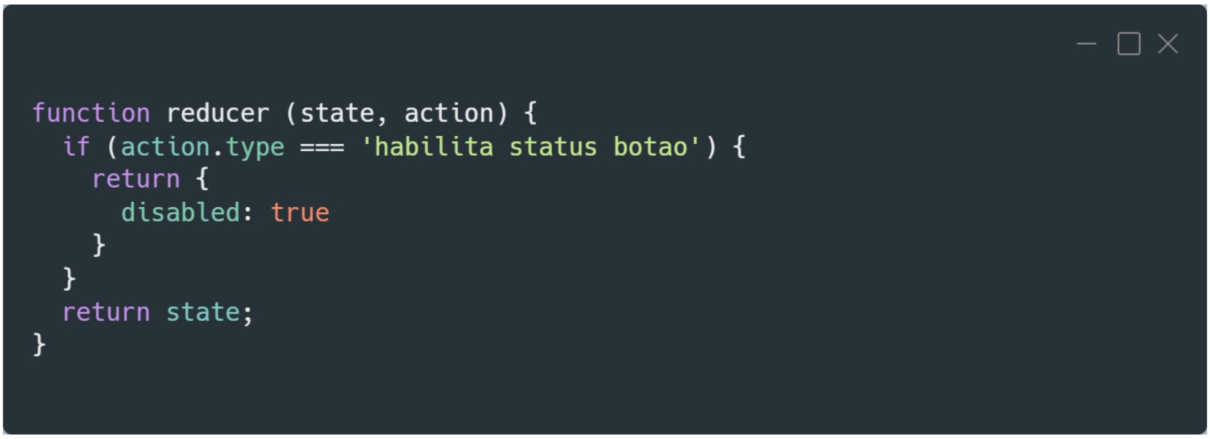
—



```
const handleClick = () => {
  dispatch({
    type: 'habilita status botao'
  });
}
```

```
const handleClick = () => {
  dispatch({
    type: 'habilita status botao'
  });
}
```

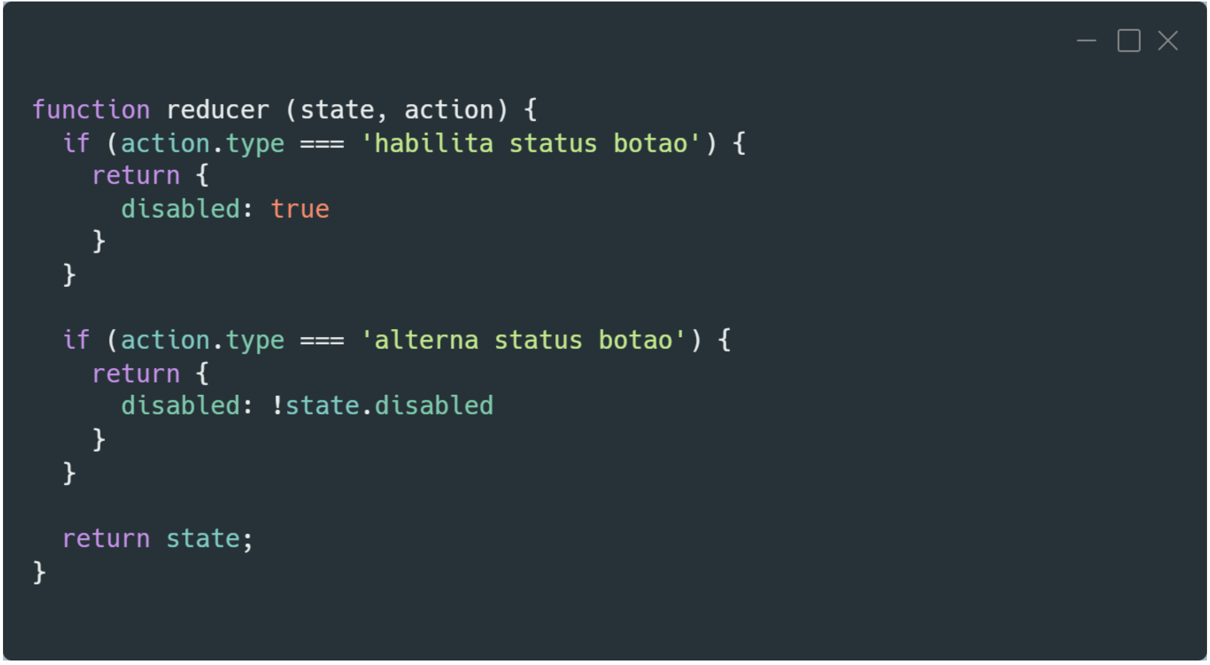
—



```
function reducer (state, action) {
  if (action.type === 'habilita status botao') {
    return {
      disabled: true
    }
  }
  return state;
}
```

```
function reducer (state, action) {
  if (action.type === 'habilita status botao') {
    return {
      disabled: true
    }
  }
  return state;
}
```

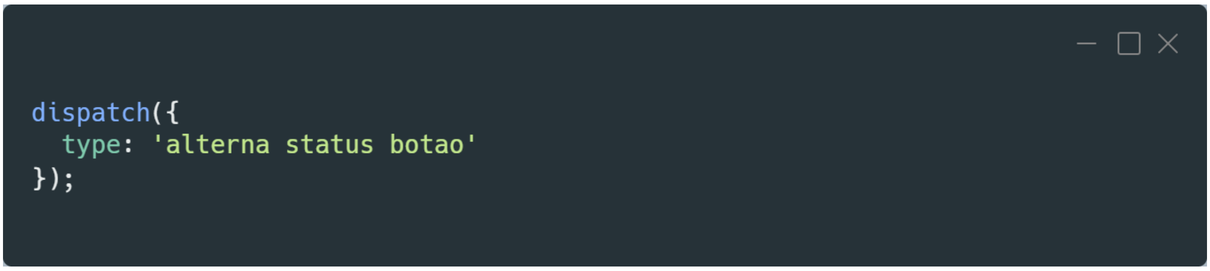
—



```
function reducer (state, action) {
  if (action.type === 'habilita status botao') {
    return {
      disabled: true
    }
  }

  if (action.type === 'alterna status botao') {
    return {
      disabled: !state.disabled
    }
  }

  return state;
}
```

```
function reducer (state, action) {
  if (action.type === 'habilita status botao') {
    return {
      disabled: true
    }
  }

  if (action.type === 'alterna status botao') {
    return {
      disabled: !state.disabled
    }
  }

  return state;
}
```

—

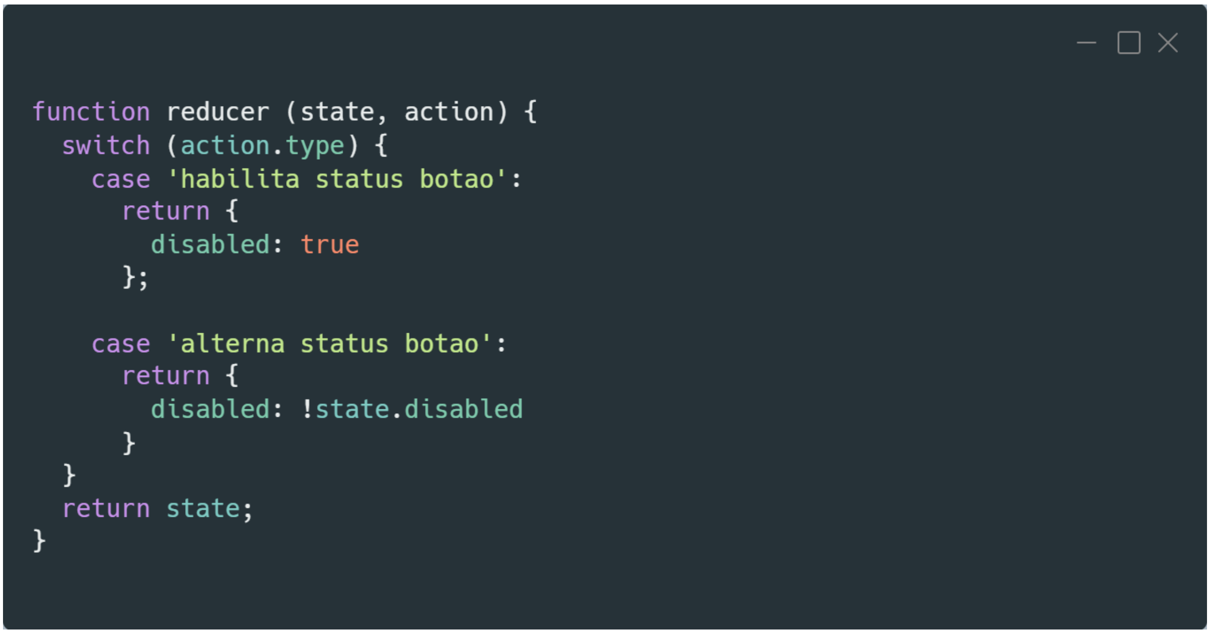


```
dispatch({
  type: 'alterna status botao'
});
```

```
dispatch({
  type: 'alterna status botao'
```

});

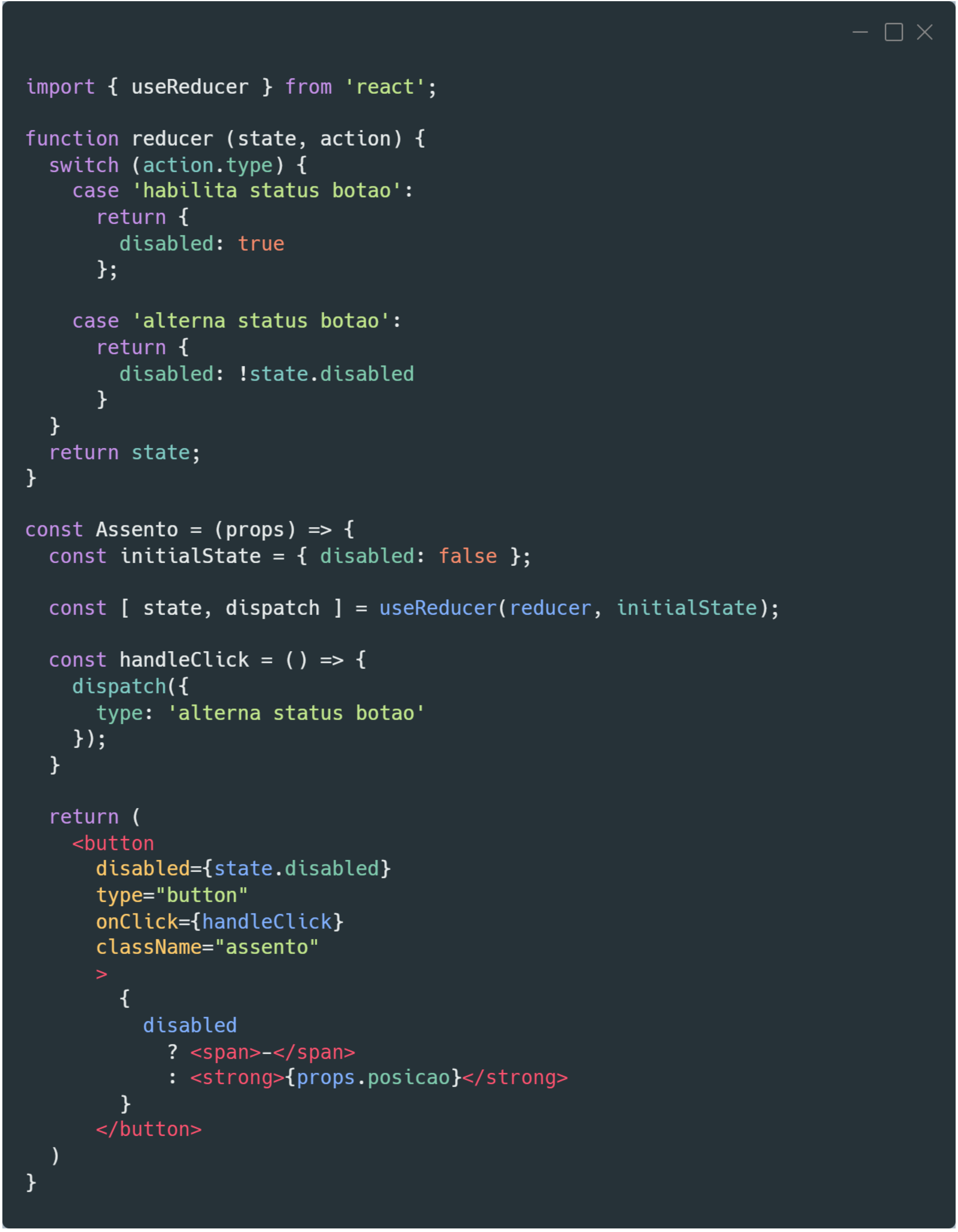
—



```
function reducer (state, action) {
  switch (action.type) {
    case 'habilita status botao':
      return {
        disabled: true
      };

    case 'alterna status botao':
      return {
        disabled: !state.disabled
      }
  }
  return state;
}
```

```
function reducer (state, action) {
  switch (action.type) {
    case 'habilita status botao':
      return {
        disabled: true
      };

    case 'alterna status botao':
      return {
        disabled: !state.disabled
      }
  }
  return state;
}
```

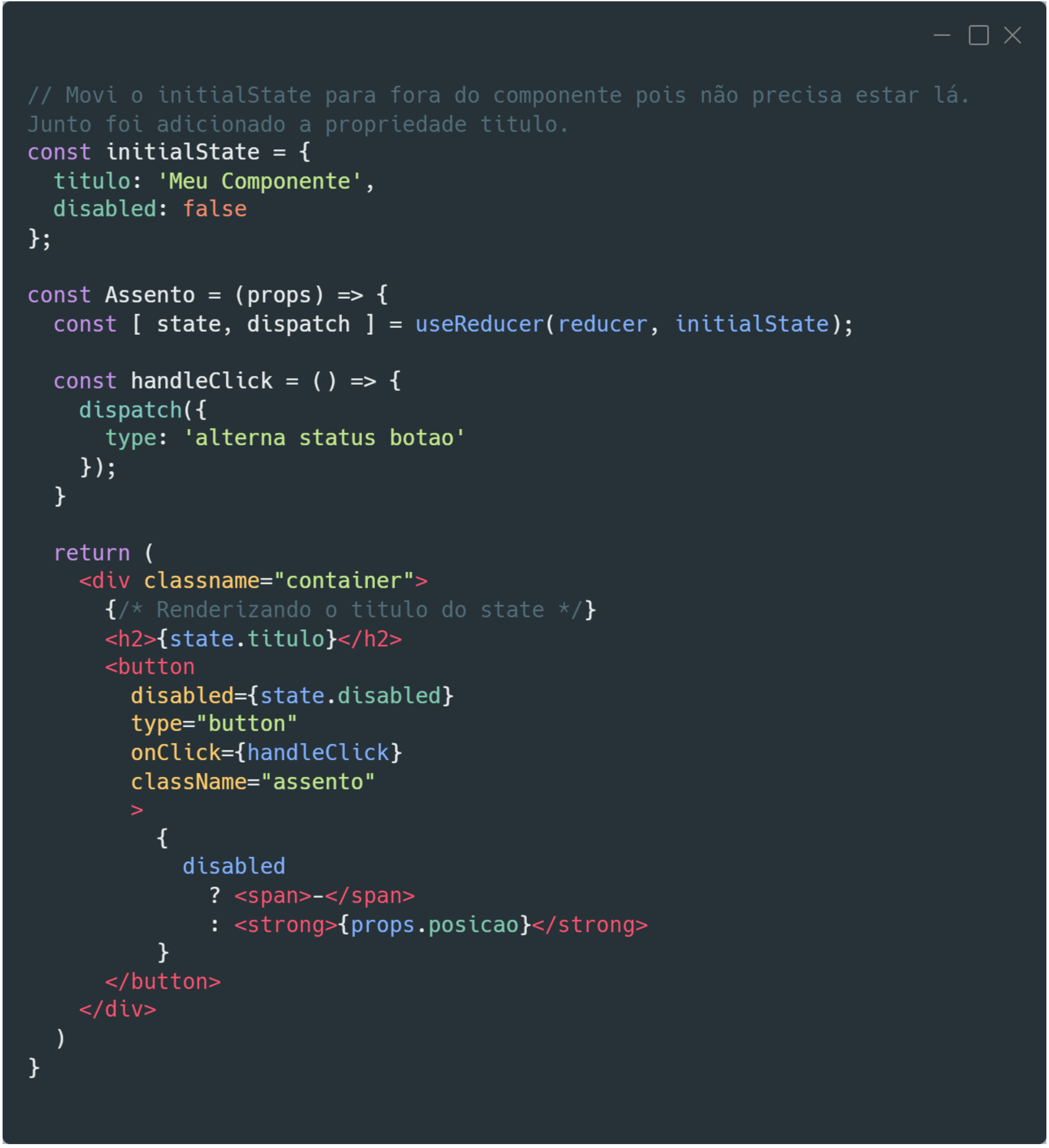
—



```
import { useReducer } from 'react';

function reducer (state, action) {
  switch (action.type) {
    case 'habilita status botao':
      return {
        disabled: true
      };

    case 'alterna status botao':
      return {
        disabled: !state.disabled
      }
  }
  return state;
}

const Assento = (props) => {
  const initialState = { disabled: false };

  const [ state, dispatch ] = useReducer(reducer, initialState);

  const handleClick = () => {
    dispatch({
      type: 'alterna status botao'
    });
  }

  return (
    <button
      disabled={state.disabled}
      type="button"
      onClick={handleClick}
      className="assento"
      >
        {
          disabled
            ? <span>-</span>
            : <strong>{props.posicao}</strong>
        }
      </button>
  )
}
```

```
import { useReducer } from 'react';

function reducer (state, action) {
  switch (action.type) {
    case 'habilita status botao':
      return {
        disabled: true
      };

    case 'alterna status botao':
      return {
        disabled: !state.disabled
      }
  }
  return state;
}

const Assento = (props) => {
  const initialState = { disabled: false };

  const [ state, dispatch ] = useReducer(reducer, initialState);

  const handleClick = () => {
    dispatch({
      type: 'alterna status botao'
    });
  }

  return (
    <button
      disabled={state.disabled}
      type="button"
      onClick={handleClick}
      className="assento"
      >
        {
          disabled
            ? <span>-</span>
            : <strong>{props.posicao}</strong>
        }
      </button>
  )
}
```

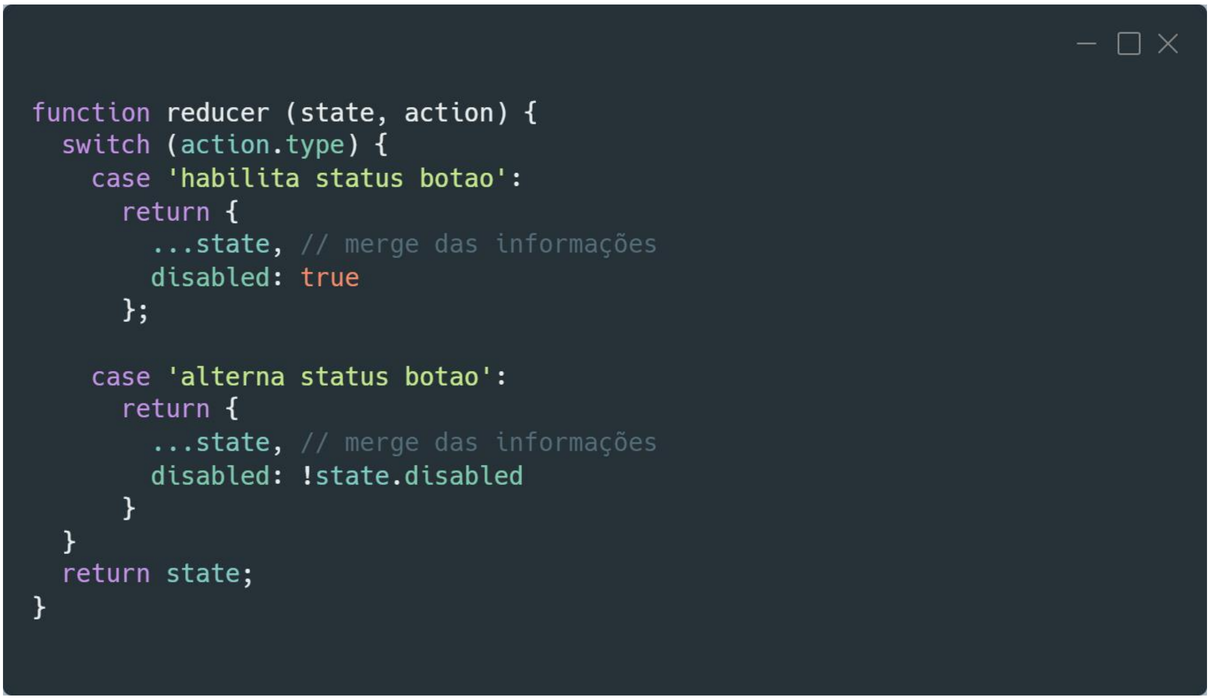
—



```
// Movi o initialState para fora do componente pois não precisa estar lá.
Junto foi adicionado a propriedade titulo.
const initialState = {
  titulo: 'Meu Componente',
  disabled: false
};

const Assento = (props) => {
  const [ state, dispatch ] = useReducer(reducer, initialState);

  const handleClick = () => {
    dispatch({
      type: 'alterna status botao'
    });
  }

  return (
    <div classname="container">
      {/* Renderizando o titulo do state */}
      <h2>{state.titulo}</h2>
      <button
        disabled={state.disabled}
        type="button"
        onClick={handleClick}
        className="assento"
      >
        {
          disabled
            ? <span>-</span>
            : <strong>{props.posicao}</strong>
        }
      </button>
    </div>
  )
}
```

```
// Movi o initialState para fora do componente pois não precisa estar lá. Junto foi adicionado
a propriedade titulo.
const initialState = {
  titulo: 'Meu Componente',
  disabled: false
};

const Assento = (props) => {
  const [ state, dispatch ] = useReducer(reducer, initialState);

  const handleClick = () => {
    dispatch({
      type: 'alterna status botao'
    });
  }

  return (
    <div classname="container">
      {/* Renderizando o titulo do state */}
      <h2>{state.titulo}</h2>
      <button
        disabled={state.disabled}
        type="button"
        onClick={handleClick}
        className="assento"
        >
          {
            disabled
              ? <span>-</span>
              : <strong>{props.posicao}</strong>
          }
      </button>
    </div>
  )
}
```

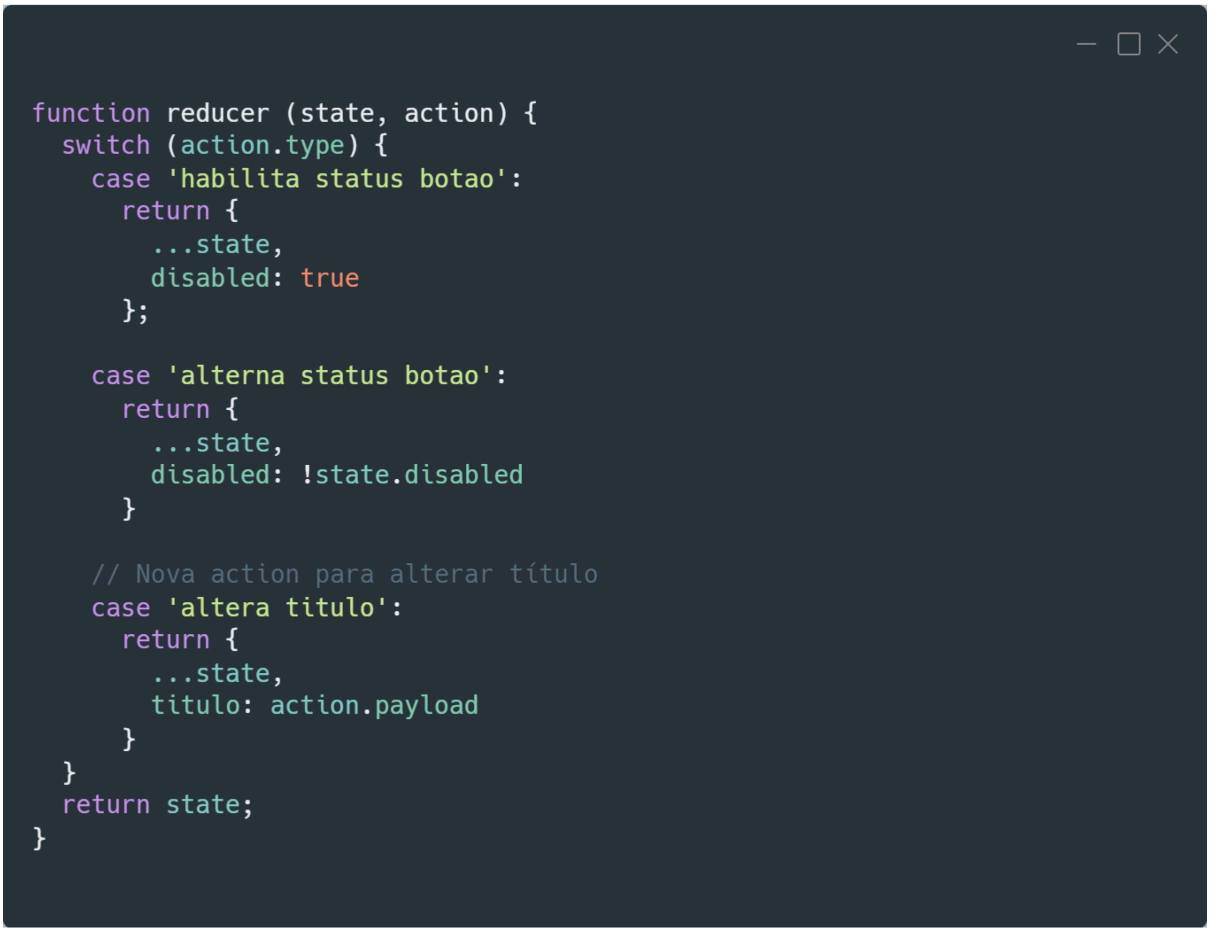
—



```
function reducer (state, action) {
  switch (action.type) {
    case 'habilita status botao':
      return {
        ...state, // merge das informações
        disabled: true
      };

    case 'alterna status botao':
      return {
        ...state, // merge das informações
        disabled: !state.disabled
      }
  }
  return state;
}
```

```
function reducer (state, action) {
  switch (action.type) {
    case 'habilita status botao':
      return {
        ...state, // merge das informações
        disabled: true
      };

    case 'alterna status botao':
      return {
        ...state, // merge das informações
        disabled: !state.disabled
      }
  }
  return state;
}
```

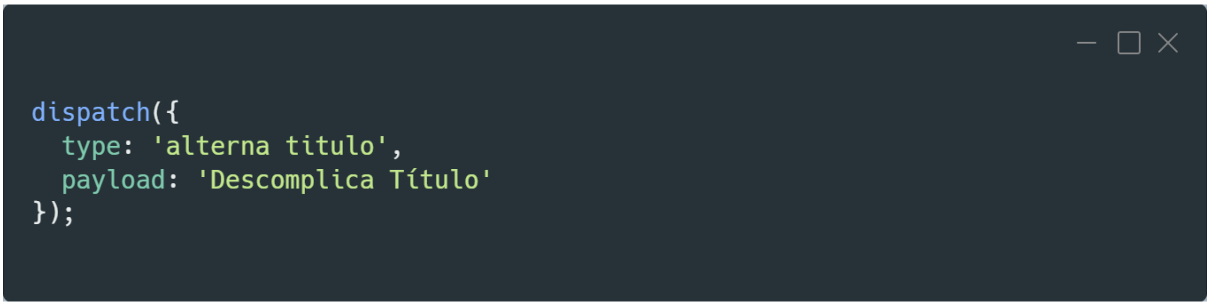
—



```
function reducer (state, action) {
  switch (action.type) {
    case 'habilita status botao':
      return {
        ...state,
        disabled: true
      };

    case 'alterna status botao':
      return {
        ...state,
        disabled: !state.disabled
      }

    // Nova action para alterar título
    case 'altera titulo':
      return {
        ...state,
        titulo: action.payload
      }
  }
  return state;
}
```

```
function reducer (state, action) {
  switch (action.type) {
    case 'habilita status botao':
      return {
        ...state,
        disabled: true
      };
    case 'alterna status botao':
      return {
        ...state,
        disabled: !state.disabled
      }

    // Nova action para alterar título
    case 'altera titulo':
      return {
        ...state,
        titulo: action.payload
      }
  }
  return state;
}
```

—



```
dispatch({
  type: 'alterna titulo',
  payload: 'Descomplica Título'
});
```

```
dispatch({
  type: 'alterna titulo',
  payload: 'Descomplica Título'
});
```