

EXERCÍCIO 1

a. Crie uma interface para representar o usuário

RESPOSTA:

```
export interface UserInputDTO {
  name: string
  balance: number
};
```

b. Implemente a função

RESPOSTA:

```
function performPurchase(user: User, value: number): User | undefined {
  if(user.balance >= value) {
    return {
      ...user,
      balance: user.balance - value
    }
  }
  return undefined
}
```

EXERCÍCIO 2

a. Faça um teste com um usuário com o saldo maior do que o valor de compra

RESPOSTA:

```
describe("Testing UserDataBase", () => {
  test("Testing balance greater than", () => {
    const user: User = {
      name: "Margaret Hamilton",
      balance: 200
    }
    const result = performPurchase(user, 30)
    expect(result).toEqual({
      ...user,
      balance: 170
    })
  })
})
```

b. Faça um teste com um usuário com o saldo igual ao valor de compra

RESPOSTA:

```
test("Testing balance equal than value", () => {
  const user: User = {
    name: "Margaret Hamilton",
    balance: 200
  }
  const result = performPurchase(user, 200)
  expect(result).toEqual({
    ...user,
    balance: 0
  })
})
```

c. Faça um teste com um usuário com o saldo menor do que o valor de compra

RESPOSTA:

```
test("Testing balance less than value", () => {
  const user: User = {
    name: "Margaret Hamilton",
    balance: 200
  }
```

```

    }
    const result = performPurchase(user, 300)
    expect(result).toEqual(undefined)
  });

```

EXERCÍCIO 3

b. Implemente a função

RESPOSTA:

```

export function verifyAge(casino: Casino, users: User[]): Result {
  const authorized: User[] = [];
  const unauthorized: User[] = [];
  for (const user of users) {
    if (casino.location === LOCATION.EUA) {
      if (user.age >= 21) {
        authorized.push(user);
      } else {
        unauthorized.push(user);
      }
    } else if (casino.location === LOCATION.BRAZIL) {
      if (user.age >= 18) {
        authorized.push(user);
      } else {
        unauthorized.push(user);
      }
      break;
    }
  }
};

return {
  brazilians: {
    authorized: authorized
      .filter((user) => user.nacionalidade === NACIONALIDADE.BRAZILIANA)
      .map((u) => u.name),

    unauthorized: unauthorized
      .filter((user) => user.nacionalidade === NACIONALIDADE.BRAZILIANA)
      .map((u) => u.name),
  },
  americans: {
    authorized: authorized
      .filter((user) => user.nacionalidade === NACIONALIDADE.AMERICANA)
      .map((u) => u.name),

    unauthorized: unauthorized
      .filter((user) => user.nacionalidade === NACIONALIDADE.AMERICANA)
      .map((u) => u.name),
  },
};

```

c. O que foi mais difícil de fazer?

RESPOSTA:

O mais difícil foi pensar em como passar as condições e fazer o map para o result (nada foi muito fácil)

EXERCÍCIO 4

a. Escreva um teste que receba um usuário brasileiro que possa entrar em um estabelecimento no Brasil

RESPOSTA:

```

test(" brazilian authorized", () => {
  const brazilian: User = {
    name: "Marilda Sotomayor",

```

```

    age: 55,
    nationality: NACIONALITY.BRAZILIAN,
  };

  const casino: Casino = {
    name: "Matemática Top",
    location: LOCATION.BRAZIL,
  };

  const result = verifyAge(casino, [brazilian]);
  expect(result.brazilians.authorized).toEqual(["Marilda Sotomayor"]);
});

```

b. Escreva um teste que receba um usuário americano que possa entrar em um estabelecimento no Brasil
RESPOSTA:

```

test("american authorized", () => {
  const american: User = {
    name: "Carl Sagan",
    age: 20,
    nationality: NACIONALITY.AMERICAN,
  };

  const casino: Casino = {
    name: "Uma galáxia muito louca",
    location: LOCATION.BRAZIL,
  };

  const result = verifyAge(casino, [american]);
  expect(result.americans.authorized).toEqual(["Carl Sagan"]);
});

```

c. Escreva um teste que receba dois usuários brasileiros e dois americanos. Todos devem ter a idade de 19 anos e quererem entrar em um estabelecimento nos Estados Unidos.
RESPOSTA:

d. Escreva um teste que receba dois usuários brasileiros e dois americanos. Os brasileiros devem ter 19 anos e os americanos 21 anos. Eles querem entrar em um estabelecimento nos Estados Unidos.
RESPOSTA:

```

test("2 american authorized and 2 brazilian unauthorized", () => {
  const american: User = {
    name: "Carl Sagan",
    age: 21,
    nationality: NACIONALITY.AMERICAN,
  };
  const brazilian: User = {
    name: "Marilda Sotomayor",
    age: 18,
    nationality: NACIONALITY.BRAZILIAN,
  };

  const casino: Casino = {
    name: "Balada das baladas tops",
    location: LOCATION.EUA,
  };

  const result = verifyAge(
    casino,
    [american,
    american,
    brazilian,
    brazilian]
  );
  expect(result.americans.authorized).toEqual(["Carl Sagan", "Carl Sagan"]);
  expect(result.brazilians.unauthorized).toEqual(["Marilda Sotomayor", "Marilda Sotomayor"]);
});

```

EXERCÍCIO 5

a. Escreva um teste que receba um usuário brasileiro que possa entrar em um estabelecimento no Brasil. Verifique que o **tamanho do array allowed** da propriedade **brasilians** tenha tamanho menor do que 2 e maior do que 0.

RESPOSTA:

```
test("1 brazilian authorized", () => {
  const brazilian: User = {
    name: "Marilda Sotomayor",
    age: 18,
    nationality: NACIONALITY.BRAZILIAN,
  };

  const casino: Casino = {
    name: "Matemática Top",
    location: LOCATION.BRAZIL,
  };
  const result = verifyAge(casino, [brazilian]);
  expect(result.brasilians.authorized.length).toBeLessThan(2);
  expect(result.brasilians.authorized.length).toBeGreaterThan(0);
});
```

b. Escreva um teste que receba um usuário americano que possa entrar em um estabelecimento no Brasil. Verifique que o **tamanho do array unallowed** da propriedade **americans** tenha tamanho igual a 0.

RESPOSTA:

```
test("1 american authorized", () => {
  const american: User = {
    name: "Astrodev",
    age: 19,
    nationality: NACIONALITY.AMERICAN,
  };
  const casino: Casino = {
    name: "Balada Estelar",
    location: LOCATION.BRAZIL,
  };
  const result = verifyAge(casino, [american]);
  expect(result.americans.unauthorized.length).toBe(0);
});
```

c. Escreva um teste que receba dois usuários brasileiros e dois americanos. Todos devem ter a idade de 19 anos e quererem entrar em um estabelecimento nos Estados Unidos. Verifique que os arrays **unallowed** possuam o nome de algum dos usuários que você criou

RESPOSTA:

```
test("No one authorized", () => {
  const american: User = {
    name: "Carl Sagan",
    age: 19,
    nationality: NACIONALITY.AMERICAN,
  };
  const brazilian: User = {
    name: "Marilda Sotomayor",
    age: 19,
    nationality: NACIONALITY.BRAZILIAN,
  };
  const casino: Casino = {
    name: "Balada das baladas tops",
    location: LOCATION.EUA,
  };
  const result = verifyAge(casino, [
    american,
    american,
    brazilian,
    brazilian,
  ]);
```

```
    });  
    expect(result.brazilians.unauthorized).toContain(["Marilda Sotomayor"]);  
    expect(result.americans.unauthorized).toContain(["Carl Sagan"]);  
});
```

d*. Escreva um teste que receba dois usuários brasileiros e dois americanos. Os brasileiros devem ter 19 anos e os americanos 21 anos. Verifique que o **tamanho do array unallowed** da propriedade **brazilians** tenha tamanho maior do que 1. Verifique que o **tamanho do array unallowed** da propriedade **americans** tenha tamanho menor do que 1. Verifique que o **tamanho do array allowed** da propriedade **ame** tenha tamanho igual a 2.

RESPOSTA:

```
test("2 american authorized and 2 brazilians unauthorized", () => {  
    const american: User = {  
        name: "Carl Sagan",  
        age: 19,  
        nationality: NACIONALITY.AMERICAN,  
    };  
    const brazilian: User = {  
        name: "Marilda Sotomayor",  
        age: 19,  
        nationality: NACIONALITY.BRAZILIAN,  
    };  
    const casino: Casino = {  
        name: "Balada das baladas tops",  
        location: LOCATION.EUA,  
    };  
    const result = verifyAge(casino, [  
        brazilian,  
        brazilian,  
        american,  
        american,  
    ]);  
    expect(result.brazilians.unauthorized.length).toBeGreaterThan(1);  
    expect(result.americans.unauthorized.length).toBeLessThan(1);  
    expect(result.americans.authorized.length).toBe(2);  
});
```

EXERCÍCIO 6