

```
// Teste: código fonte disponível em http://github.com/Spagiari/jstest
// Avalie as funções abaixo e responda as perguntas.
// As perguntas avaliam pelo menos um tema importante da linguagem Javascript.
// Será avaliado a compreensão do código, o conhecimento sobre o tema e
// a argumentação sobre a questão e / ou possíveis soluções.
var a;
function teste1() {
  var a = 3;
}
teste1();
console.log(a === 3);
/* *****
 * (1) Por que o valor falso é escrito no console?
 * *****/

function teste2() {
  var arr = [1, 2, 3];
  for (var i = 0; i < arr.length; i++) {
    setTimeout(function() {
      console.log(`Index: ${i};\nElement: ${arr[i]}`);
    }, 0);
  }
}
teste2();
/* *****
 * (2) Qual o valor escrito no console e por quê?
 * *****/

function inc(i) {
  i++;
}
function teste3() {
  var i = 0;
  inc(i);
  console.log(i);
}
teste3();
/* *****
 * (3) Quais os valores escritos no console e por quê?
 * *****/

function teste4() {
  var value = 10;
  return function() {
    return value;
  };
}
func = teste4();
console.log(func());
/* *****
 * (4) Qual o valor escrito no console e por quê?
 * *****/

// A função shuffle recebe como entrada uma array e a embaralha.
function shuffle(a) {
  // atribui a variável n a quantidades de elementos da array a.
  n = a.length;
  // Percorre a array a da posição 0 até a ultima posição.
  for (var i = 0; i < n; i++) {
    //Sorteia uma posição aleatória entre i e n-1 e atribui para a variável swap
    swap = i + Math.floor(Math.random() * (n - i - 1));
    // Troca os valores das posições i e swap da array a.
    var aux = a[i];
    a[i] = a[swap];
    a[swap] = aux;
  }
}
var array = [5, 3, 1, 4, 2];
if (shuffle(array)) console.log(array);
/* *****
 * (5) Existem potenciais armadilhas (pitfalls) na função acima?
 * Se SIM qual é ou quais são? Comente a resposta.
 * *****/
```

```

/* *****
 * (6) A função baixo doAllStuff tem muitas responsabilidades.
 * Refatore o código e crie testes unitários e de integração.
 * Crie um projeto no github com a sua solução.
 * *****/
const util = require('util');
const fs = require('fs');

/**
 * doAllStuff - Calcula a distância entre dois pontos.
 * @param {number} l1 - latitude ponto 1.
 * @param {number} b1 - longitude ponto 1.
 * @param {number} l2 - latitude ponto 2.
 * @param {number} b2 - longitude ponto 2.
 * @return {Promise<{number}>} Promise para a distância entre os dois pontos.
 */
function doAllStuff(l1, b1, l2, b2) {
  const earthRadiusKm = 6371;

  if (l1 < -90 || l1 > 90 || l2 < -90 || l2 > 90)
    throw new RangeError('the arguments l1 and l2 must be between -90 and 90.');
```

if (b1 < -180 || b1 > 180 || b2 < -180 || b2 > 180)
 throw new RangeError('the arguments b1 and b2 must be between -180 and 180.');

```

  //converte coordenadas de graus para radianos
  const l1Radianus = (l1 * Math.PI) / 180;
  const b1Radianus = (b1 * Math.PI) / 180;
  const l2Radianus = (l2 * Math.PI) / 180;
  const b2Radianus = (b2 * Math.PI) / 180;

  //calcula distância entre os dois pontos
  const lambda = Math.acos(
    Math.sin(l1Radianus) * Math.sin(l2Radianus) +
    Math.cos(l1Radianus) *
    Math.cos(l2Radianus) *
    Math.cos(b2Radianus - b1Radianus),
  );

  const distanceBetweenTwoPoints = Math.sin(lambda) * earthRadiusKm;

  //salva informações no banco de dados
  const appendFilePromisified = util.promisify(fs.appendFile);
  return appendFilePromisified(
    'database.txt',
    `p1(${l1}, ${b1}) p2(${l2}, ${b2}) d= ${distanceBetweenTwoPoints}KM\n`,
  ).then(() => {
    return distanceBetweenTwoPoints;
  });
}

// Exemplo de uso. Distância entre as cidades de São Paulo e Rio de Janeiro
doAllStuff(-23.618237, -46.635197, -22.9035, -43.2096).then(value =>
  console.log(value),
);

```