

# WITHDRAW IN ATM PROJECT README

Developed by Jonathan Martins

[PORTUGUÊS]

## USANDO O PROGRAMA

Este sistema simula uma operação de saque em um caixa rápido. Para usar o programa você deve:

1. Abrir o arquivo "saque.html" no seu navegador;
2. Inserir o valor do saque no campo;
3. Apertar no botão "Sacar Dinheiro";
4. Confirmar suas notas recebidas.

Este programa é configurado com as seguintes notas de dinheiro disponíveis:

1. Ilimitadas notas de 100\$
2. Ilimitadas notas de 50%
3. Ilimitadas notas de 10\$
4. Ilimitadas notas de 05\$
5. Ilimitadas notas de 01\$

Quando referido notas "Ilimitadas" na verdade existe um valor de 99.999 notas disponíveis em cada um dos valores para simular a quantidade ilimitada.

## EDITANDO ESTE PROGRAMA

Se você precisa aprimorar este sistema basta você editar o código no VScode (ou outro programa) usando a linguagem JavaScript.

Para fazer alterações no dispenser, você pode alterar este objeto:

```
// Cédulas disponíveis no dispenser  
  
let dispenser = { 100: 99999, 50: 99999, 10: 99999, 5: 99999, 1: 99999 }
```

Neste objeto contém o valor da nota de dinheiro e a quantidade disponível de notas.

Agora se você precisa alterar a função de sacar dinheiro, basta você fazer alterações na seguinte função:

```

function sacarDinheiro(valorDoSaque, dispenser) {
  let saque = (valorDoSaque, notas) => {
    if (valorDoSaque <= 0) {
      return {};
    }
    if (!notas.length) {
      return;
    }
    let nota = notas[0]
    let contador = Math.min(dispenser[nota], Math.floor(valorDoSaque / nota));
    for (let i = contador; i >= 0; i--) {
      let resultado = saque(valorDoSaque - i * nota, notas.slice(1));
      if (resultado) return i ? { [nota]: i, ...resultado } : resultado;
    }
  }
  return saque(valorDoSaque, Object.keys(dispenser).map(Number).sort((a, b) => b - a));
};

```

Você pode alterar os nomes das variáveis para inglês ou outro termo de sua preferência.

Na “let saque” nós temos uma importante variável que pega o valor do saque e conflita com as notas disponíveis no dispenser deste caixa eletrônico. Você pode colocar mensagens de erro quando o valor for menor ou igual a zero.

Agora no “let nota” e “let contador”, o programa conecta o valor do saque com o dispenser e retorna as notas na tela do usuário.

Antes de tudo isto nós temos a variável “saque” para pegar o valor do saque em outra função chamada “clicar”:

```

// Função clicar para sacar o dinheiro

function clicar() {
  var valorDoSaque = document.querySelector('.querSacar').value
  var resultado = sacarDinheiro(valorDoSaque, dispenser);
  var texto = ""
  for (const [chave, valor] of Object.entries(resultado)) {
    texto += " Você recebeu " + valor + " notas de " + chave + "<br>"
  }
  document.getElementById("valores").innerHTML = texto
};

```

Esta função conecta os valores inseridos (html) com a função “sacarDinheiro” e retorna de fato as notas de dinheiro na tela do usuário.

[ENGLISH]

USING THIS PROGRAM

This is a Project of ATM withdraw system. For use this program you can do:

1. Open the “saque.html” archive in your navigator
2. Insert your withdraw value on input
3. Press “Sacar Dinheiro” button for realize withdraw
4. Confirm your notes received.

This program is configured for these money notes on dispenser:

1. Unlimited money notes of 100\$
2. Unlimited money notes of 50\$
3. Unlimited money notes of 10\$
4. Unlimited money notes of 05\$
5. Unlimited money notes of 01\$

When we say “unlimited money notes” is actually 99.999 amount notes available on dispenser to simulate a unlimited amount.

#### EDITING THIS PROGRAM

If you need to improve this program you can editing code on your VScode (or another program) using JavaScript language.

For editing notes on dispenser, you can make changes in this object:

```
// Cédulas disponíveis no dispenser  
  
let dispenser = { 100: 99999, 50: 99999, 10: 99999, 5: 99999, 1: 99999 }
```

This object contains a money note value and the number of notes.

Now, if you need change the withdraw function (not recommend), you can edit on this function:

```
function sacarDinheiro(valorDoSaque, dispenser) {  
  let saque = (valorDoSaque, notas) => {  
    if (valorDoSaque <= 0) {  
      return {};  
    }  
    if (!notas.length) {  
      return;  
    }  
    let nota = notas[0]  
    let contador = Math.min(dispenser[nota], Math.floor(valorDoSaque / nota));  
    for (let i = contador; i >= 0; i--) {  
      let resultado = saque(valorDoSaque - i * nota, notas.slice(1));  
      if (resultado) return i ? { [nota]: i, ...resultado } : resultado;  
    }  
  }  
  return saque(valorDoSaque, Object.keys(dispenser).map(Number).sort((a, b) => b - a));  
};
```

You can change the variable names for English language or another term of your preference.

In the “let saque” we have an important variable to get withdraw request value and conflict with money notes available in the dispenser of this ATM. You can post an error message when the withdraw value is equal a zero (and less than zero too).

Now in the “let nota” and “let contador”, the program connects a requested withdraw value with dispenser to return notes on the user screen.

First time in this function, we have a “saque” variable to get the withdraw value on other function with name “clicar”:

```
// Função clicar para sacar o dinheiro

function clicar() {
    var valorDoSaque = document.querySelector('.querSacar').value
    var resultado = sacarDinheiro(valorDoSaque, dispenser);
    var texto = ""
    for (const [chave, valor] of Object.entries(resultado)) {
        texto += " Você recebeu " + valor + " notas de " + chave + "<br>"
    }
    document.getElementById("valores").innerHTML = texto
};
```

This function connects input values (html) with “sacarDinheiro” function and return a string values and amount money notes on user screen.