

CÓDIGO CHAT

Aqui eu estou explicando todo o código js do chat, e relacionando as const dele, com o nosso código, esse código está calculando a máscara, as sub redes, o 1 endereço e o Último endereço. Podem testar o html e o js.

linha 14 (try..catch)

O bloco `try` é usado para envolver um trecho de código onde se espera que uma exceção possa ocorrer. Dentro desse bloco, o código problemático é colocado, e caso ocorra uma exceção, o fluxo de execução é interrompido, e a exceção é lançada. Para lidar com essa exceção, usamos o bloco `catch`.

O bloco `catch` é executado apenas se uma exceção for lançada dentro do bloco `try`. Ele captura a exceção e permite que o programador defina o comportamento apropriado para lidar com o erro. O bloco `catch` recebe um parâmetro, que é o objeto da exceção capturada. Esse objeto contém informações sobre o tipo de erro ocorrido e, em alguns casos, detalhes adicionais sobre o erro.

```
1  function processForm(formData) {
2    try {
3      // Processar os dados do formulário
4      const result = formData.username.toUpperCase();
5      console.log("Resultado:", result);
6    } catch (error) {
7      console.error("Erro ao processar formulário:", error.message);
8    } finally {
9      // Limpar os campos do formulário, independentemente de erros
10     clearFormFields(formData);
11   }
12 }
13 const formData = { username: "usuário" };
14 processForm(formData);
```

isNaN(not a number)

Para entender o que é a declaração abaixo, leia este arquivo, o prefixo pelo oq eu entendi


é a mascara, no caso do exemplo se a mascara é 22, quer dizer que em binario,nos octetos, teria vinte e dois, numeros 1, e 10 numeros 0.

11111111.11111111.11111100.00000000

```
const [ip, prefix] = network.split('/')
```

Blocos, prefixos e máscaras em IPv4

Introdução Um IPv4 é usualmente representado como quatro números decimais separados por um ".": v.x.y.z onde cada um dos v, x, y, z acima podem assumir valores de 0 a 255.

 <https://ii.blog.br/2019/11/14/blocos-prefixos-e-mascaras-em-ipv4/>


/31	/30	/29	/28	/27	/26	/25	/24	/23	/22	/21	/20	/19	/18
1													
2	1												
4	2	1											
8	4	2	1										
16	8	4	2	1									
32	16	8	4	2	1								
64	32	16	8	4	2	1							
128	64	32	16	8	4	2	1						
256	128	64	32	16	8	4	2	1					
512	256	128	64	32	16	8	4	2	1				
1024	512	256	128	64	32	16	8	4	2	1			
2048	1024	512	256	128	64	32	16	8	4	2	1		
4096	2048	1024	512	256	128	64	32	16	8	4	2	1	
8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4




No código, a máscara e o endereço ip, estão como text, apenas a quantidade que pe number, parar isso o código do chat ta convertendo todos eles para um inteiro

Unsigned right shift (>>>) - JavaScript | MDN

The unsigned right shift (>>>) operator returns a number whose binary representation is the first operand shifted by the specified number of bits to the right. Excess bits shifted off to

 https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Unsigned_right_shift

 mdn web docs

CÓDIGO COMENTADO ABAIXO:

HTML:

```
<!DOCTYPE html>
<html lang="PT-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Calculadora de Sub-redes</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Calculadora de Sub-redes</h1>
    <form id="subnet-form">
      <label for="network">Endereço de Rede:</label>
      <input type="text" id="network" name="network" required>

      <label for="mask">Máscara de Rede:</label>
      <input type="text" id="mask" name="mask" required>/

      <label for="subnets">Número de Sub-redes:</label>
      <input type="number" id="subnets" name="subnets" required>

      <button type="button" onclick="calculateSubnets()">Calcular
    </form>
    <div id="result"></div>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

JS:

```

function calculateSubnets() { //function calculaSubredes
    const network = document.getElementById('network').value; //
    const mask = document.getElementById('mask').value; //const
    const numSubnets = parseInt(document.getElementById('subnet:

    if (!network || isNaN(numSubnets) || numSubnets < 1) {
        alert('Por favor, insira um endereço de rede válido e o
        return;
    } //se o ip for diferente ou nao for um numero ou a quantidade
    //Verifica se algum dos valores não está presente, se o número
    //nao necessário no código até o momento

    const resultDiv = document.getElementById('result');
    resultDiv.innerHTML = ''; //é pra o resultado nao aparecer,

    ///Inicia um bloco try para capturar qualquer erro que possa ocorrer
    try {
        const networkWithMask = `${network}/${mask}`; //Cria uma
        //podemos musar o nome networkWithMask por, enderecoIPMask
        const [ip, prefix] = network.split('/'); //Divide a string
        const networkAddress = ipStringToInt(ip); //convertendo
        const networkPrefix = parseInt(prefix); //convertendo a
        const numBitsSubnets = Math.ceil(Math.log2(numSubnets));
        const newPrefix = networkPrefix + numBitsSubnets; //a nova
        //explicação chat deu:calculo o novo prefixo adicionando

        const subnets = []; //criou uma lista para as sub redes,
        for (let i = 0; i < numSubnets; i++) { //for passando por
            const subnetAddress = networkAddress + (i * (1 << (32 - networkPrefix));
            const firstIp = subnetAddress + 1; //Calcula o primeiro
            const lastIp = subnetAddress + (1 << (32 - newPrefix));
            subnets.push({ //adiciona na lista nova,o número da sub
                subnet: intToIpString(subnetAddress) + '/' + newPrefix,
                firstIp: intToIpString(firstIp),
                lastIp: intToIpString(lastIp)
            });
        }
    } catch (error) {
        console.log(error);
    }
}

```

```

    });
}

subnets.forEach((subnet, index) => {
    resultDiv.innerHTML += <p>Sub-rede ${index + 1}:<br>
}); // aqui ele ta adicionando tudo para o html
} catch (e) {
    alert('Erro ao calcular sub-redes: ' + e.message);
}
}

function ipStringToInt(ip) {
    return ip.split('.').reduce((acc, octet) => (acc << 8) + parseInt(octet, 10), 0);
} // e nessa função ele esta separando em octetos e converte cada um para inteiro

function intToIpString(int) {
    return [
        (int >>> 24) & 255, // Desloca os 8 bits mais significativos para a esquerda
        (int >>> 16) & 255, // Desloca os próximos 8 bits mais significativos para a esquerda
        (int >>> 8) & 255, // Desloca os próximos 8 bits mais significativos para a esquerda
        int & 255 // Aplica uma máscara para obter o quarto octeto
    ].join('.'); // Junta os octetos em uma string separada por pontos
}

```