

Universidad Autónoma De Baja California



Facultad de Ingeniería

Licenciado en Sistemas Computacionales

Aplicaciones web con base de datos

Examen calculadora Angular

Brenda Edith Bautista Quiroz

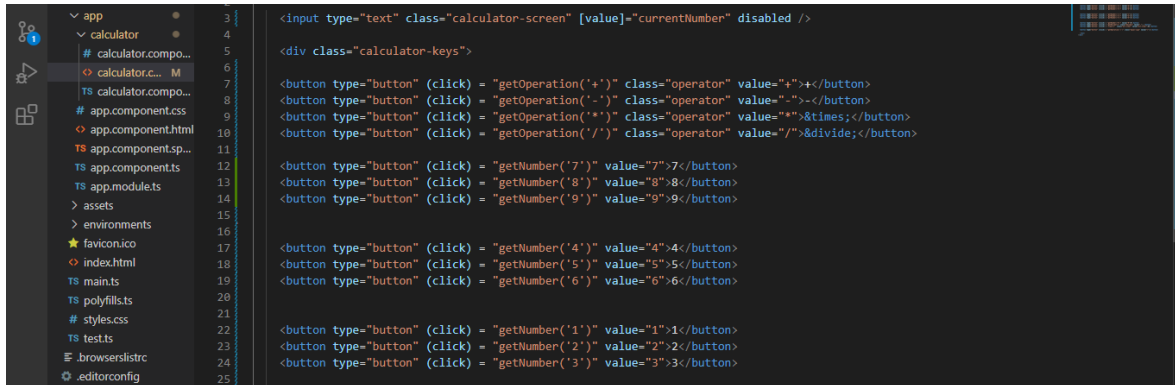
1158925

Eventos de clic en los botones

En los botones lo que se hizo fue agregar el evento (click) = "getOperation('-') a cada uno de los botones que hace referencia a los operadores de la calculadora y para los números getNumber que son para los números que se desea visualizar para cada botón.

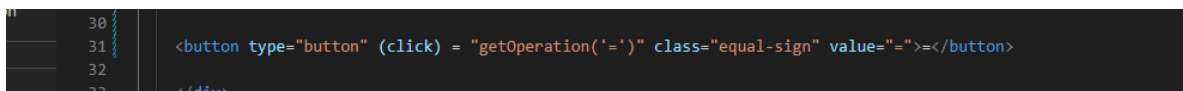
Esto se añadió en el archivo calculator.component.html

Se modificó el input para que al momento de hacer clic en los botones de la calculadora se pueda visualizar los números de lo contrario seguirá mostrando el número 0



```
3 <input type="text" class="calculator-screen" [value]="currentNumber" disabled />
4
5 <div class="calculator-keys">
6
7   <button type="button" (click) = "getOperation('+)" class="operator" value="+"/></button>
8   <button type="button" (click) = "getOperation('-)" class="operator" value="-"/></button>
9   <button type="button" (click) = "getOperation('*)" class="operator" value="*"/></button>
10  <button type="button" (click) = "getOperation('/)" class="operator" value="/"/></button>
11
12  <button type="button" (click) = "getNumber('7)" value="7"/></button>
13  <button type="button" (click) = "getNumber('8)" value="8"/></button>
14  <button type="button" (click) = "getNumber('9)" value="9"/></button>
15
16  <button type="button" (click) = "getNumber('4)" value="4"/></button>
17  <button type="button" (click) = "getNumber('5)" value="5"/></button>
18  <button type="button" (click) = "getNumber('6)" value="6"/></button>
19
20  <button type="button" (click) = "getNumber('1)" value="1"/></button>
21  <button type="button" (click) = "getNumber('2)" value="2"/></button>
22  <button type="button" (click) = "getNumber('3)" value="3"/></button>
```

También se agregó el evento (click) en el botón = en esta parte lo que sucede es de acuerdo al método de operación que se realice se va a visualizar el resultado obtenido.



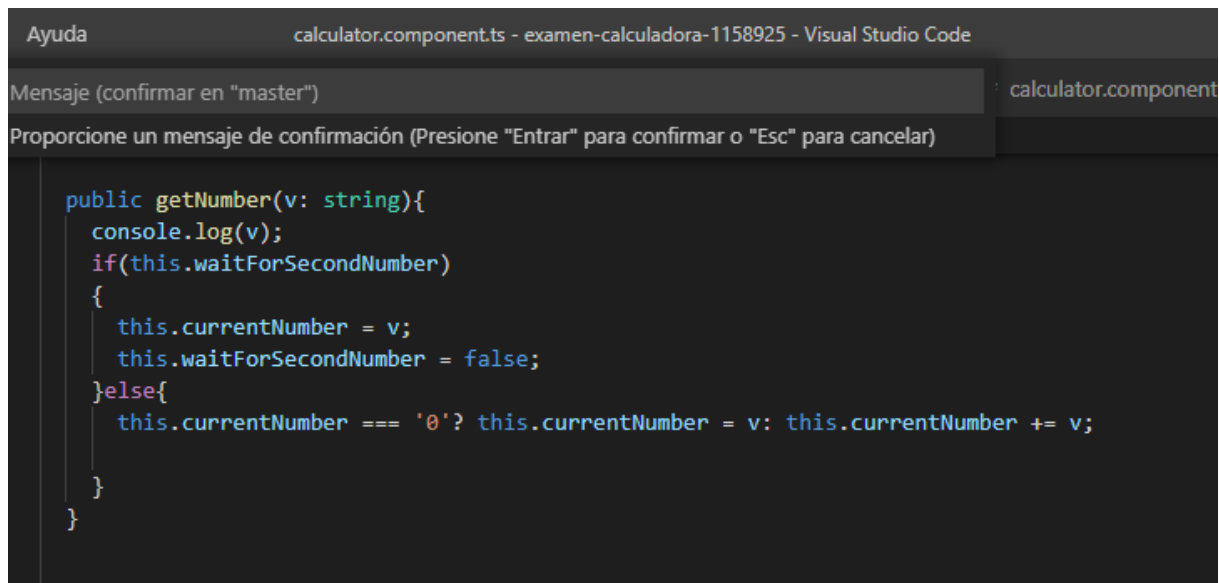
```
30
31 <button type="button" (click) = "getOperation('=)" class="equal-sign" value="="/></button>
32
33 </div>
```

Métodos

getNumber()

El número es guardado en la variable v, luego una vez ingresado el valor es igualado, luego le damos clic a un botón para que realice la operación y si los valores no son igual a 0 entonces se podrán

visualizar.

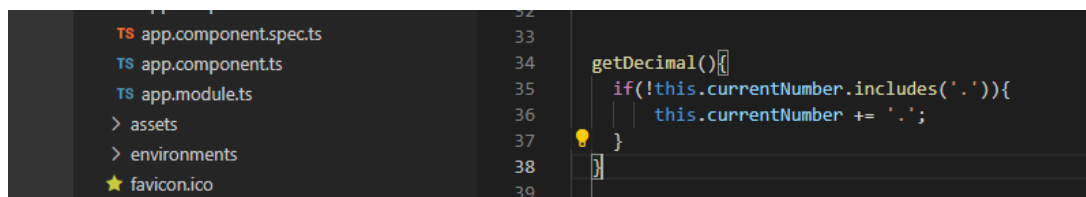


```
calculator.component.ts - examen-calculadora-1158925 - Visual Studio Code

Mensaje (confirmar en "master")
Proporcione un mensaje de confirmación (Presione "Entrar" para confirmar o "Esc" para cancelar)

public getNumber(v: string){
  console.log(v);
  if(this.waitForSecondNumber)
  {
    this.currentNumber = v;
    this.waitForSecondNumber = false;
  }else{
    this.currentNumber === '0'? this.currentNumber = v: this.currentNumber += v;
  }
}
```

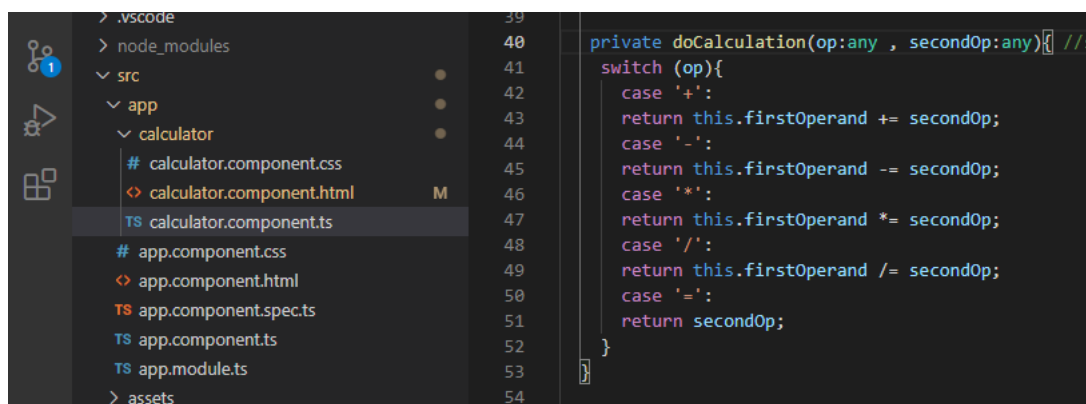
getDecimal() Nos agrega el punto decimal al número que se vaya a ingresar si el usuario ya ingresó el punto decimal al número entonces ya no permite agregar otro punto decimal.



```
app.component.spec.ts 32
app.component.ts       33
app.module.ts          34
assets                 35
environments           36
favicon.ico            37
calculator.component.ts 38
calculator.component.html 39

getDecimal(){
  if(!this.currentNumber.includes('.')){
    this.currentNumber += '.';
  }
}
```

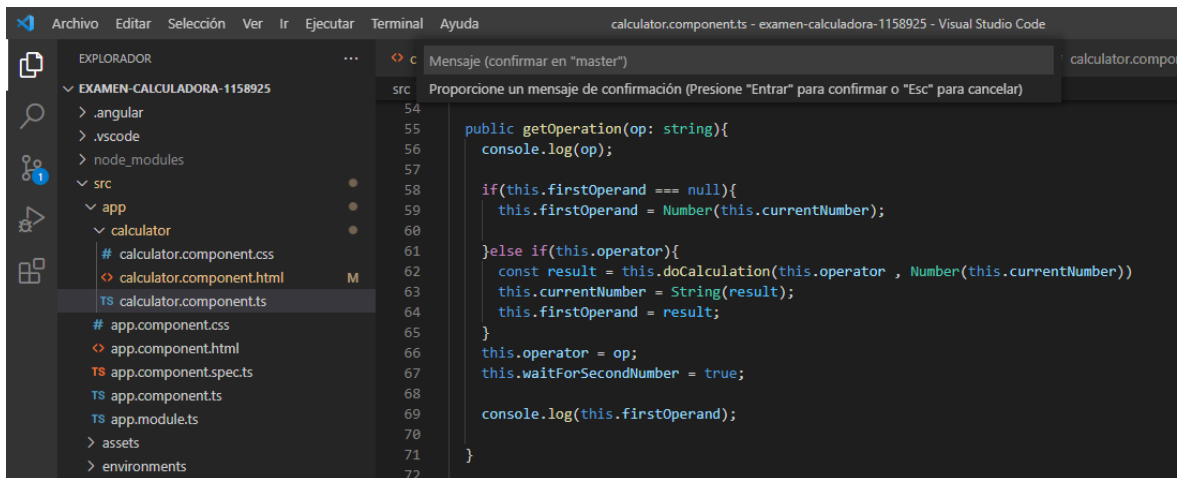
doCalculation() Lo que hace aquí es de acuerdo al operador que se elija y a los números ingresados hace un recorrido por el switch y muestra un resultado de la operación.



```
.vscode 39
node_modules 40
src 41
  app 42
    calculator 43
      calculator.component.css 44
      calculator.component.html 45
      calculator.component.ts 46
      app.component.css 47
      app.component.html 48
      app.component.spec.ts 49
      app.component.ts 50
      app.module.ts 51
      assets 52

private doCalculation(op:any , secondOp:any){ //
  switch (op){
    case '+':
      return this.firstOperand += secondOp;
    case '-':
      return this.firstOperand -= secondOp;
    case '*':
      return this.firstOperand *= secondOp;
    case '/':
      return this.firstOperand /= secondOp;
    case '=':
      return secondOp;
  }
}
```

getOperation() utilizado para la operación que se elija ya sea suma, resta, multiplicación o división, en currentNumber se guarda el elemento y cuando le demos clic en igual se va a visualizar el resultado esto lo hace en el else if que aquí es donde imprime el resultado del operador.



Lo que hace es limpiar los valores ingresados y los operadores que se hayan ingresado también, una vez hecho esto se va a visualizar 0 en la calculadora.

