

# Proyecto Pruebas con Jest

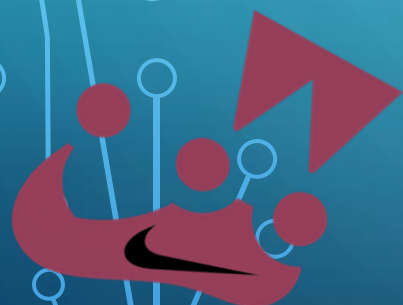
# Vizcacha: unitarias

Alejandro Moreno  
Lechado




Iván Francisco Bueno  
Uceta

Francisco Crespo Martín

Jinghao Lin



# INDICE

- Suma +
- Resta -
- Multiplicar x
- Dividir ÷
- Piedra 
- Papel 
- Tijera 

# DOCUMENTACIÓN CALCULADORA

**suma ( a : number, b : number ) : number**

Toma dos números como entrada y devuelve la suma de estos



```
test("Añadir 2 y 3", () =>{  
    expect(suma(2, 3)).toBe(5)  
})
```



```
function suma(a, b) {  
    return a+b  
}
```

# DOCUMENTACIÓN CALCULADORA

**resta ( a : number, b : number ) : number**

Toma dos números como entrada y devuelve la resta del primero con el segundo.



```
test("Restar 5 y 4", () =>{  
    expect(calc.resta(5,4)).toBe(1)  
})
```



```
function resta(a, b) {  
    return a-b  
}
```

# DOCUMENTACIÓN CALCULADORA

**multiplicar ( a : number, b : number ) : number**

Toma dos números como entrada y devuelve el producto de estos

```
test("Multiplicar 11 y 2", () =>{  
  expect(calc.multiplicar(11, 2)).toBe(22)  
})
```

```
function multiplicar(a, b) {  
  return a*b  
}
```

# DOCUMENTACIÓN CALCULADORA

**dividir ( a : number, b : number ) : number**

Toma dos números como entrada y devuelve su división, si el divisor es 0 lanza un error

```
function dividir(a, b) {  
  if (b === 0) {  
    throw new Error("El divisor no puede ser 0")  
  }  
  return a/b  
}
```

```
test("Dividir 2 y 4", () =>{  
  expect(calc.dividir(2, 4)).toBe(.5)  
})  
  
test("Dividir por 0", ()=>{  
  expect(() => calc.dividir(2, 0)).toThrow(Error)  
})
```



# DOCUMENTACIÓN PIEDRA, PAPEL Y TIJERA

Pide a dos usuarios que introduzcan un valor (piedra, papel o tijera)

Una vez se introducen los datos se hace una comprobación para cada opción del usuario 1 y comprobando el resultado el resultado del usuario 2 devuelve true si el usuario 1 gana, si el usuario 2 gana o ambos empatan, el resultado es false

```
function usuario(eleccion1, eleccion2){
  var eleccionUsuario1 =eleccion1
  var eleccionUsuario2 =eleccion2

  if(eleccionUsuario1 == 'piedra'){//el usuario1 eligio piedra
    if(eleccionUsuario2 == 'papel'){//si el usuario2 ha elgido papel
      console.log("¡Perdiste! El usuario 2 eligio papel y tu piedra.")
      return false;
    }else{
      if(eleccionUsuario2 == 'tijeras'){//si el usuario2 ha elgido tijeras
        console.log("¡Ganaste!El usuario 2 eligio tijera y tu piedra.")
        return true;
      }else{
        if(eleccionUsuario2 == eleccionUsuario1){//si el usuario2 ha elgido piedra
          console.log("¡Empate!Ambos eligieron piedra.")
          return false;
        }
      }
    }
  }
  if(eleccionUsuario1 == 'papel'){//el usuario1 eligio papel
    if(eleccionUsuario2 == 'tijeras'){//si el usuario2 eligio tijeras
      console.log(" ¡Perdiste! El usuario 2 eligio tijera y tu papel.")
      return false;
    }else{
      if(eleccionUsuario2 == 'piedra'){//si el usuario2 eligio piedra
        console.log("¡Ganaste!El usuario 2 eligio piedra y tu papek.")
        return true;
      }else{
        if(eleccionUsuario2 == eleccionUsuario1){//si el usuario2 ha elgido papel
          console.log("¡Empate!Ambos eligieron papel.")
          return false;
        }
      }
    }
  }
  if(eleccionUsuario1 == 'tijeras'){//el usuario1 eligio piedra
    if(eleccionUsuario2 == 'piedra'){//si el usuario2 eligio piedra
      console.log(" ¡Perdiste! El usuario 2 eligio piedra y tu tijera.")
      return false;
    }else{
      if(eleccionUsuario2 == 'papel'){//si el usuario2 ha elgido papel
        console.log("¡Ganaste!El usuario 2 eligio papel y tu tijera.")
        return true;
      }else{
        if(eleccionUsuario2 == eleccionUsuario1){//si el usuario2 ha elgido tijeras
          console.log("¡Empate!Ambos eligieron tijera.")
          return false;
        }
      }
    }
  }
}
};
```

# DOCUMENTACIÓN PIEDRA, PAPEL Y TIJERA

Este test comprueba si la elección de piedra ha ganado o no ha ganado, si ha empatado no se considera victoria, lo cual es una derrota

```
test('Piedra en usuario 1', () => {  
  expect(usuario('piedra','piedra')).toBeFalsy();  
  expect(usuario('piedra','papel')).toBeFalsy();  
  expect(usuario('piedra','tijeras')).toBeTruthy();  
});
```



# DOCUMENTACIÓN PIEDRA, PAPEL Y TIJERA

Este test comprueba si la elección de papel ha ganado o no ha ganado, si ha empatado no se considera victoria, lo cual es una derrota



```
test('Papel en usuario 1', () => {  
  expect(usuario('papel','piedra')).toBeTruthy();  
  expect(usuario('papel','papel')).toBeFalsy();  
  expect(usuario('papel','tijeras')).toBeFalsy();  
});
```

# DOCUMENTACIÓN PIEDRA, PAPEL Y TIJERA

Este test comprueba si la elección de tijera ha ganado o no ha ganado, si ha empatado no se considera victoria, lo cual es una derrota

```
test('Tijera en usuario 1', () => {  
  expect(usuario('tijeras', 'piedra')).toBeFalsy();  
  expect(usuario('tijeras', 'papel')).toBeTruthy();  
  expect(usuario('tijeras', 'tijeras')).toBeFalsy();  
});
```