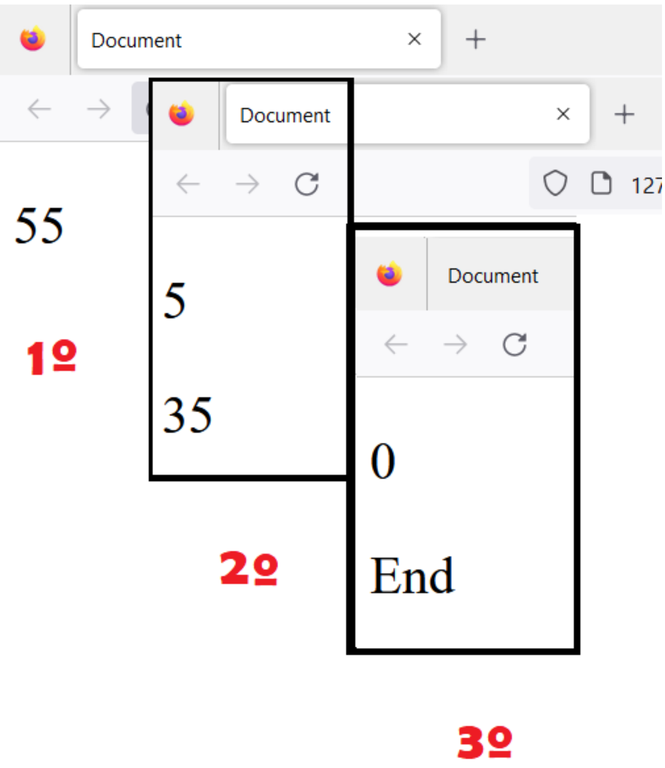
❑ ACTIVITY 1 – Timer made with promises

- Create a function that receives a number of milliseconds and generates a timer that writes the text "Time Complete" when the indicated milliseconds have elapsed.
- Actually the function is not written, but creates a promise whose result is the text “Time is Completed”.
- In case the promise is not fulfilled, the result will be the text "Time is not going well"
- It will be decided that the promise has not been fulfilled when twice the indicated milliseconds have elapsed.

❑ ACTIVITY 2 - Timer Function Module

- Create a file called a **timer.js**, which will be a module that we can import into our main Javascript code.
- This module will consist of 2 functions:
  - The **timer** function created in the previous practice
  - The function **counts**. This function will be able to write a countdown, and display it in an HTML element. The parameters are:
    - The number with which the countdown begins
    - The element in which we will write the countdown. By default it will use the elemeto **body**.
    - The interval in milliseconds at which each number changes. By default it will be 1000.
    - A callback function, whose code is executed when the countdown ends.
- It also creates a web application that creates in 2 paragraphs so that the first one is counted down from 6, passing from second to second. In the second it will be counted from 60 but each number will move from tenth to tenth. Also, when you reach 0, we want you to type "End".



55

5

35

0

End

EJ1

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Temporizador</title>
<script>
function temporizador(tiempo){
      var promesa=new Promise((resolver, rechazar)=>{
      var temp=setTimeout(()=>{
            clearTimeout(temp2);
            resolver("Tiempo concluido");
            },tiempo);
      var temp2=setTimeout(()=>{
            rechazar("El tiempo no va bien");
            },tiempo*2);
      });
      return promesa;
}
</script>
</head>

<body>
<script>
temporizador(5000).then((texto)=>{
      document.body.innerHTML="<p>"+texto+"</p>";
      }).catch((m)=>{document.body.textContent=m;})
</script>
</body>
</html>
```

EJ2

HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
<style>
p{
      font-size:2em;
}
</style>
</head>
<body>
      <p></p>
      <p></p>
<script type="module">
import { temporizador, cuenta } from "./temporizador.js";
let p1=document.querySelector("p:first-of-type");
let p2=document.querySelector("p:last-of-type");
cuenta(6,p1);
cuenta(60,p2,100,()=>{
      p2.textContent="End";
});
</script>
</body>
</html>
```

JS

```
/**
* Crea una promesa que genera un mensaje en el
* tiempo indicado
* @param {Number} tiempo ms
*/
export function temporizador(tiempo){
      var promesa=new Promise((resolver, rechazar)=>{
      var temp=setTimeout(()=>{
            clearTimeout(temp2);
            resolver("Tiempo concluido");
      },tiempo);
      var temp2=setTimeout(()=>{
            rechazar("El tiempo no va bien");
            },tiempo*2);
      });
      return promesa;
}
/**
* Crea una cuenta atrás en el elemento indicado
* @param {number} numero Desde dónde contamos
* @param {*} elemento En que elemento HTML escribimos
* @param {*} tiempo ms en los que cambia cada número
* @param {*} f Función opcional que queremos que se ejecute al final
*/
export async function cuenta(numero,elemento=document.body,tiempo=1000,f){
try{
    for(let i=numero;i>=0;i--){
        await temporizador(tiempo);
        elemento.textContent=i;
    }
    if (f) f();
}catch(e){
    console.log("Error "+e);
    }
}
```