

En este ejercicio, trabajarás con **esperas en Watir** para interactuar con elementos que no están disponibles inmediatamente en la página. El objetivo es asegurarte de que tu automatización maneje correctamente los retrasos en la carga de elementos antes de interactuar con ellos.

### Archivo HTML: wait\_example.html

Guarda este código en un archivo y ejecútalo en un servidor local.

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Watir Wait Example</title>

  <script type="text/javascript">

    function showDelayedContent() {

      setTimeout(function() {

        document.getElementById("delayedText").style.display = "block";

      }, 5000); // Se mostrará después de 5 segundos

    }

    function enableButton() {

      setTimeout(function() {

        document.getElementById("delayedButton").disabled = false;

      }, 7 * 1000); // Se habilitará después de 7 segundos

    }

  </script>

</head>

<body onload="showDelayedContent(); enableButton();">

  <h2>Watir Wait Example</h2>

  <!-- Elemento con carga retardada -->

  <div id="delayedText" style="display:none;">
```

```
<p>Este texto apareció después de 5 segundos.</p>
</div>

<!-- Botón que se habilita después de 7 segundos -->
<button id="delayedButton" disabled>Haz clic cuando esté habilitado</button>

</body>
</html>
```

### Instrucciones

1. **Carga la página en un servidor local.**
  - Puedes hacerlo con `python3 -m http.server 8000` en la carpeta donde guardaste el archivo.
  - Luego, accede a `http://localhost:8000/wait_example.html`.
2. **Automatiza la interacción con Watir, usando esperas para:**
  - Detectar el momento en que el texto retardado aparece y confirmarlo.
  - Esperar a que el botón se habilite antes de hacer clic.
3. **Toma capturas de pantalla después de cada evento.**