

# Práctica de laboratorio: Aplicación JavaScript Cisco Coffee (optativo)

## Objetivos

**Parte 1: Explorar una aplicación web**

**Parte 2: Ver el código de una aplicación web JavaScript**

## Información básica

Esta actividad se centra en una aplicación web diseñada para supervisar plantaciones de café. La aplicación JavaScript Cisco Coffee es un ejemplo sencillo de un código de programación que se puede usar para informar el estado de sensores y proporcionar alertas. Esta aplicación web, que está conectada a sensores simulados de temperatura, humedad y luz, supervisa y muestra información relacionada con los valores que leen dichos sensores.

**Nota:** Esta aplicación no se comunica con sensores reales. Se diseñó como simulación y para proporcionar un ejemplo del uso de JavaScript para la programación web. Para completar la parte 2 de la práctica de laboratorio, deberá descargar el archivo **Archivos JavaScript de Cisco Coffee.zip**.

## Parte 1: Explorar una aplicación web

Se eligió proporcionar esta aplicación a través de la Web debido a la flexibilidad de la gran mayoría de los dispositivos preparados para la red que tienen un navegador web instalado. Las computadoras, los teléfonos y las tablet PC no deberían experimentar problemas para cargar y ejecutar esta aplicación web en sus respectivos navegadores web.

Cuando se carga la aplicación en un navegador web, se comienzan a mostrar las lecturas simuladas que capturan los sensores de las plantaciones. Debería ver los valores de temperatura, luz y humedad. La aplicación también cuenta con un área de registro (en la parte inferior de la pantalla), que se usa para mostrar eventos inusuales, como tormentas. Si los valores que registran los sensores no están dentro de los límites predefinidos, también se hace un seguimiento de este evento en el área de registro.

Se muestran tres botones (de izquierda a derecha): Promedios del día anterior, Promedios del día siguiente y Mostrar datos reales. El primer y el segundo botón permiten que el agricultor vea los niveles promedio de temperatura, luz y humedad de días anteriores. En la aplicación, se almacenan promedios de hasta siete días. El último botón permite que el agricultor alterne entre los datos simulados y los reales.

**Nota:** No hay sensores reales conectados a esta aplicación. Por este motivo, cuando hace clic en **Mostrar datos reales**, en la pantalla se muestra **No hay datos de sensor disponibles**. Haga clic en **Mostrar datos simulados** para que se muestren los valores del día.

## Parte 2: Ver el código de una aplicación web JavaScript.

La aplicación de esta actividad se escribió en lenguaje JavaScript, dado que es fácil de aprender. Si le interesa aprender más sobre la codificación, los siguientes pasos le serán útiles.

### Paso 1: Abrir el código en una ventana del navegador web y en un editor de texto.

Las aplicaciones web son (básicamente) páginas web y, por lo tanto, siempre se puede ver el código que se usó para crear la página que se muestra en la ventana del navegador. Para ver el código fuente, haga clic con el botón secundario en cualquier lugar de la ilustración y elija una opción similar a **Ver código fuente** o **Ver fuente del marco**, según el navegador. Desplácese hasta la sección del código que comienza con **Script**. Las líneas que comienzan con una doble barra diagonal (//) indican comentarios. Los comentarios proporcionan una breve explicación del código. También puede completar los siguientes pasos para descargar y abrir la aplicación web en un editor de texto:

- a. Si aún no lo hizo, descargue el archivo **Archivos JavaScript de Cisco Coffee.zip** para la página en la que descargó estas instrucciones.
- b. Descomprima el archivo, ubique la carpeta y ábrala. Debe ver dos archivos: *Cisco\_Coffee\_JavaScript.html* y *pouring\_coffee\_grd.png*. El primer archivo contiene el código propiamente dicho, mientras que el segundo archivo es la imagen de fondo que se usa en la aplicación. Se puede abrir el archivo que contiene el código con cualquier editor de texto. Puede elegir su editor de texto favorito para abrir este archivo, pero en esta actividad, se usa el Bloc de notas de Microsoft porque ya está incluido en el paquete de Windows.

**Nota:** Evite usar Microsoft Word cuando cree o edite páginas web. Se sabe que el formato que se incluye en Word crea una ubicación de imágenes y texto no deseada en la página web.

- c. Seleccione el archivo *Cisco\_Coffee\_JavaScript.html* y haga clic con el botón secundario en este. Seleccione **Abrir con > Bloc de notas**. Ahora debe ver una ventana de Bloc de notas en la que se muestra el código de la aplicación web.

**Nota:** El Bloc de notas se limita a mostrar solo el código. Si desea ver características adicionales, como colores diferentes para las distintas partes del código, así como también números de línea, descargue un editor de texto gratuito, como **Notepad++**.

### Paso 2: Ver la sección HTML del código en la aplicación web.

Como habrá notado, el archivo que está investigando tiene una extensión `.html`, lo que indica que es un archivo HTML. HTML es el principal lenguaje que se usa para armar páginas web. Le permite al desarrollador organizar la información que se muestra en la página web. Si bien los navegadores web son algo flexibles en cuanto a la sintaxis de HTML, los archivos HTML tienen una estructura bien definida dividida en secciones. Las secciones HTML comunes son HTML, HEAD y BODY. Para marcar estas secciones, los estándares de HTML definen etiquetas HTML. En general, las etiquetas HTML marcan el comienzo y el final de una sección, por lo que todo lo que está en el medio se convierte en parte de la sección. El estándar también indica que los caracteres “superior a” e “inferior a” (`<>`) se usan para indicar una etiqueta. Por ejemplo, observe el siguiente código HTML:

```
<HTML>
  <HEAD>
    <TITLE>HTML Example Page</TITLE>
  </HEAD>
  <BODY>
    <P>This text is to be formatted as a paragraph on the page.</P>
    <P>And here is another paragraph</P>
  </BODY>
</HTML>
```

La primera línea del ejemplo marca el comienzo del documento HTML. En la segunda línea, se abre la sección `HEAD`. El título de la página es el texto que se muestra como nombre de la ventana y se define en la tercera línea. Los estándares de HTML establecen que el título se debe colocar debajo de la sección `HEAD`. Observe la etiqueta `</TITLE>` al final de la tercera línea: la barra diagonal (`/`), que precede al nombre de la etiqueta, marca el final de una sección. Dado que el texto `HTML Example Page` se coloca dentro de la sección `TITLE`, los navegadores web muestran ese texto como nombre de la ventana.

En la sección `BODY`, se ubica el cuerpo de la página. En el ejemplo anterior, se crea una página muy sencilla con solo dos párrafos. El texto que se coloca dentro de una etiqueta de párrafo se muestra como párrafo en los navegadores web. `<P>` y `</P>` marcan una sección de párrafo. A continuación, se cierra la sección `BODY` mediante la adición de `</BODY>` en la quinta línea.

Por último, `</HTML>` marca el final de la sección HTML. Dado que la sección HTML debe contener toda la página HTML, esta etiqueta también marca el final del archivo HTML.

### Paso 3: Ver la sección SCRIPT del código en la aplicación web.

El estándar HTML también define muchas otras etiquetas. Para usar la aplicación web con eficacia, se usan otras etiquetas. Otra etiqueta importante es la etiqueta `<SCRIPT>`. Las etiquetas `<SCRIPT>` contienen scripts del lado cliente, como JavaScript. Se denomina “script del lado cliente” porque se ejecuta en el equipo cliente, no en el servidor. Se requiere un script porque no se pueden programar etiquetas HTML (por sí solas) para que tomen decisiones. Su propósito es solamente reconocer información. Toda la lógica de la aplicación se ubica en la sección `SCRIPT`.

En general, las aplicaciones web se basan en funciones. Una función es un bloque de código escrito para realizar una tarea específica. Cuando se define, se puede invocar una función desde otra parte del código cuando se requiere la función que realiza. En una aplicación JavaScript, las funciones se definen según el siguiente formato:

```
function function_name (optional_variables_to_be_passed_into_function) {
    function code block (may span over several lines)
    function code block (may span over several lines)
    function code block (may span over several lines)
}
```

Busque las siguientes funciones en la aplicación:

```
init(); prepareToggle(); resetExtremes(); fakeValues(); dayOfWeek(d);
showPastDaysAvgs();
```

### Paso 4: Describir cada una de las funciones principales del código.

La aplicación tiene su propio flujo de trabajo, que indica el orden en que se invocan las funciones. A continuación, se muestra un resumen del flujo de la aplicación:

- El parámetro `onload="init();"` que se agrega a la etiqueta HTML BODY al final del código le indica al navegador web que invoque la función `init()` no bien se termina de cargar la página. La función `init()` se ubica en la parte superior del código, justo después del bloque de código que declara las variables.
- `init()` establece algunos parámetros del entorno, como el tamaño de la pantalla. También especifica que la función `fakeValues()` se ejecute cada tres segundos y que la función `resetExtremes()` se ejecute cada 60 segundos.
- La función `resetExtremes()` calcula valores máximos y mínimos aleatorios de temperatura, humedad y luz. Esto es necesario para simular los valores durante un día normal (sin tormentas). Después de tres segundos, se invoca `fakeValues()` (como se define en `init()` anteriormente) para elegir de manera aleatoria un número entre estos valores extremos. Esta técnica se usa para simular los valores que se mueven dentro de un intervalo muy limitado. De esta forma, se recrea el funcionamiento normal de los sensores reales.
- Para el momento en que se invoca `fakeValues()`, ya se calcularon los valores extremos. La función `fakeValues()` usa estos valores como mínimo y máximo para elegir los valores de manera aleatoria.
- En las líneas 338, 339 y 340, se definen los botones de alternancia de datos del día anterior, del día siguiente y de datos reales o simulados, respectivamente. Estos botones se muestran en la página web, y el código JavaScript llama a estos botones por su ID: `pDay`, `nDay` y `toggleBTN`.
- Nota: El código define que se debe invocar la función `showPastDaysAvgs(this.id)` cuando se hace clic en los botones de día anterior o día siguiente. El parámetro `this.id` transmite a la función el identificador del botón en el que se hace clic, lo que permite que se realice una acción diferente según el botón en el que se haya hecho clic. Cuando se alterna entre los botones Mostrar datos simulados y Mostrar datos reales, se invoca la función `prepareToggle()`.
- `prepareToggle()` se invoca cuando el usuario desea alternar entre datos reales y simulados. Para dejar de mostrar datos simulados y cambiar a datos reales, `prepareToggle()` deja de invocar `fakeValues()`.

Dado que esta aplicación es una simulación y no existen sensores reales, *prepareToggle()* también reemplaza los datos simulados que se muestran en la pantalla con el mensaje No sensor data available . Si el usuario desea ver datos simulados nuevamente, *prepareToggle()* vuelve a establecer la ejecución periódica de *fakeValues()* y elimina el mensaje No sensor data available de la pantalla. Una vez más, *fakeValues()* completa el área con los resultados de los cálculos.

- h. *dayOfWeek()* es una función de soporte que se usa para calcular los últimos siete días hábiles. Esta función es importante porque la aplicación debe mostrar los últimos siete días, más allá del día en que el usuario ejecuta la aplicación.
- i. Por último, *fakeValues()* también es responsable de completar el área de registro en la parte inferior de la aplicación. Cada vez que *fakeValues()* muestra un valor que no está dentro de los parámetros predefinidos, registra ese evento en el área de registro.
- j. *resetExtremes()* también tiene una tarea adicional: simula una tormenta cada tres minutos. Lo hace al establecer valores extremos muy alejados del intervalo normal de valores. Dado que ahora los valores son muy altos y muy bajos (con respecto a los máximos y mínimos, respectivamente), los resultados que calcula *fakeValues()* se alejan inevitablemente de los límites predefinidos. Esto se percibe como una tormenta que acaba de comenzar.

### Paso 5: Explorar y manipular el código por su cuenta.

- a. Use la descripción anterior de las funciones para examinar el código y ver si puede encontrar lugares adicionales en los que se use la lógica para generar la aplicación web.
- b. Puede modificar el código y ver los resultados. Si se rompe el código y no lo puede corregir, use el código del curso para buscar y corregir los errores.