

GUI N DE ACTIVIDAD

Desarrollo Web en Entorno Cliente

ACTIVIDAD EVALUABLE: JavaScript avanzado

OBJETIVOS

- Conocer la sintaxis de JavaScript.
- Trabajar con estructuras condicionales e iterativas del lenguaje.
- Conocer el funcionamiento de los arrays.
- Trabajar con funciones.
- Seleccionar y modificar elementos del  rbol DOM.
- Generar elementos HTML desde c digo JavaScript.

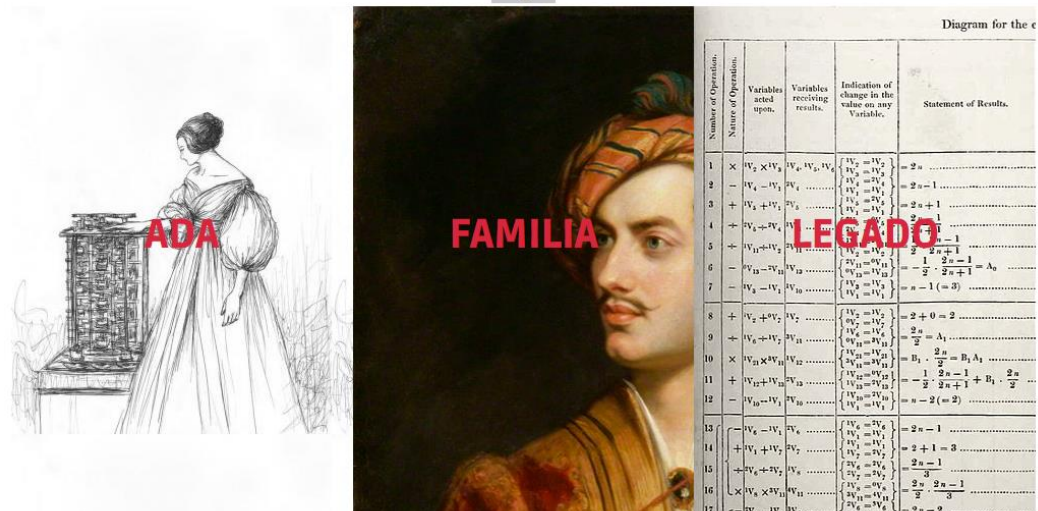
TEMPORALIZACI N

Aproximadamente 4-5 horas

PROCESO DE DESARROLLO

1. En el archivo index.html proporcionado se muestra informaci n correspondiente a tres bloques tem ticos: Biograf a, Contribuci n a la inform tica y M s all  de los n meros. Cada uno de los bloques tem ticos contiene un color de fondo interior diferente y, est  dividido en tres apartados –por ejemplo: Contribuci n a la inform tica: Encuentro con Charles Babbage, Controversia y Visi n-. Implementa un script ejercicio1.js donde tras seleccionar la informaci n correspondiente, implementes las siguientes acciones:
 - a. Al pulsar sobre los botones anterior y siguiente contenidos en el interior del primer apartado del primer bloque tem tico –Biograf a, Londres, 10 de diciembre de 1815 - 27 de noviembre de 1852-, se podr  pasar la informaci n correspondiente a los apartados del bloque tem tico –es decir, de Biograf a Familia a Biograf a Educaci n-.
 - b. Al pulsar sobre los botones anterior y siguiente contenidos en el exterior del primer apartado del primer bloque tem tico –Biograf a, Londres, 10 de diciembre de 1815 - 27 de noviembre de 1852-, se podr  pasar de bloque tem tico –es decir, de Biograf a a Contribuci n a la inform tica, por ejemplo-.
 - c. Finalmente, deber s borrar el resto apartados y bloques tem ticos, de manera que solo quede el primer apartado de todos.

- Implementa un archivo ejercicio2.js en el archivo galeria.html proporcionado. Selecciona el título de primer nivel con la inscripción Ada y cambia el color a rojo y añádele una imagen de fondo. Además, añade otros dos títulos de primer nivel con otras dos fotos de fondo donde se indique las inscripciones Familia y Legado.



Implementa las siguientes acciones:

Al pulsar sobre cualquiera de las tres imágenes, se eliminarán todas ellas y se según sea el caso, se mostrarán las seis imágenes correspondientes –Ada, Familia y Legado, contenidas todas ellas en la carpeta src-. Por ejemplo, al pulsar sobre Ada, se mostrará:



Al pulsar sobre cada una de estas imágenes, se eliminarán todas ellas y se mostrará la imagen pulsada con un tamaño igual al que ocupaban las seis imágenes anteriores. Si se vuelve a pulsar sobre la imagen agrandada, se volverá al menú anterior –seis imágenes correspondientes-.

Si en cualquier momento se hace doble click sobre el título Ada Lovelace contenido en el div cuya clase es jumbotron, se volverá al primer menú –tres imágenes con tres títulos rojos Ada, Familia, Legado-.

- Implementar un script ejercicio3.js que permita validar el formulario contenido en el archivo contacto.html proporcionado. El formulario posee dos campos: nombre y email. Se deberá validar que la información introducida es correcta para ambos campos y, en caso afirmativo, se eliminará el formulario y se mostrará la inscripción “En breve le contactaremos”, contenida en un párrafo.

EVALUACIÓN

Esta actividad corresponde a la parte de actividades evaluables de la evaluación.

OBSERVACIONES

Se entregará un único archivo comprimido con los tres archivos ejercicio1.js, ejercicio2.js, ejercicio3.js en Florida Oberta. El script a entregar deberá ceñirse a lo que se pide explícitamente en el enunciado de la actividad. Cualquier desviación de las funcionalidades solicitadas o la implementación de funcionalidades no solicitadas de manera explícita en el enunciado, no incrementarán el valor de la nota, pudiéndose llegar a penalizar por ello.

RÚBRICA

| Criterio | No/nunca | Poco/a veces | Moderado / regularmente | Mucho / frecuentemente | Sí/siempre |
|--|--|--|---|--|---|
| Funcionalidad: cumple con las especificaciones requeridas en el enunciado. Valor máximo: 70% | No cumple ninguno de los requerimientos. Valor: 0% | Cumple parte de los requerimientos, pero falla en lo principal. Valor: 7% | Cumple parte de los requerimientos esenciales, pero está incompleto. Valor: 15% | Cumple los requerimientos esenciales, pero se escapa algún requerimiento menor. Valor: 30% | Cumple todos los requerimientos especificados. Valor: 70% |
| Operatividad: el desarrollo es operativo. Compila y está libre de errores en tiempo de compilación/ejecución. Valor máximo: 20% | No compila / no se ejecuta. Valor: 0% | Compila, pero sufre errores tempranos en ejecución. Valor: 5% | Compila y no da errores durante la fase inicial de ejecución. Valor: 10% | No lanza errores más que en alguna funcionalidad menor. Valor: 15% | No falla. Libre de errores de principio a fin de la ejecución. Valor: 20% |
| Legibilidad: el código es legible, inteligible y está bien estructurado. Las nomenclaturas de elementos son consistentes a lo largo de la aplicación. Valor máximo: 10% | El código es prácticamente ilegible. Código “espagueti” sin cumplir normas de formato, indentación ni exención de “código muerto”. Valor: 0% | El código es poco legible. En general, mal formateado, y con presencia de partes de código muerto y/o comentarios evitables. Formato poco consistente. Valor: 2% | El código se puede leer, pero hay fallos de formateo. No hay código muerto ni comentarios que se puedan obviar. Valor: 5% | Buena legibilidad de código. No existe código muerto y, en general, se aprecia un buen formateo de código. Pocos comentarios obvios. Valor: 7% | Perfecta legibilidad de código. No existe código muerto. Formateo de código. Sin comentarios obvios. Valor: 10% |