*3RA ENTREGA DEL PROYECTO FINAL*

**Componentes:**

1. Estructura HTML del simulador
2. Librería jQuery
3. Estilos del simulador
4. Datos estáticos
5. Script del simulador

*Estructura inicial del Simulador*

* **Formato:** Archivos HTML
* **Objetivos del desafío:**
  + El estudiante deberá utilizar el tag correspondiente para asociar un script JS a una página HTML.
  + Se deberá utilizar el tag correspondiente para asociar la librería jQuery a la página HTML.
  + Refinar estructura e interfaz del simulador, con foco principal en los elementos destinados a la captura de entradas (formularios, inputs y/o botones) y las secciones en el HTML destinadas a notificación de salidas.

*Librería jQuery*

* **Formato:** Archivo JS - Libreria jQuery
* **Objetivos del desafío:**
  + Incorporar librería jQuery al proyecto, en su opción de archivo local o como referencia CDN.

*Estilos del Simulador*

* **Formato:** Archivos CSS
* **Objetivos del desafío:**
  + El estudiante deberá crear reglas CSS o emplearlas predefinidas con el objetivo de otorgar una correcta visualización de la estructura del simulador, elementos generados, entradas y salidas, guardando coherencia con el contexto del simulador escogido.

*Datos Estáticos*

* **Formato:** Archivos JSON
* **Objetivos del desafío:**
  + Incorporar carga de datos estáticos mediante llamada asíncrona. Ya sea desde un archivo local o REST API.

*Script Inicial del Simulador*

* **Formato:** Archivo JS
* **Objetivo del desafío:**
  + El estudiante deberá ajustar y codificar funcionalidades del simulador. Identificando el flujo de trabajo en el script en términos de captura de entradas y eventos de usuario, funciones de procesos esenciales y notificación de resultados en forma de salida por HTML, modificando el DOM, asegurando la respuesta inmediata ante eventos utilizando animaciones.
  + Se deberá manejar la respuesta a llamadas asíncronas correctamente, utilizando los datos obtenidos en el procesamiento principal.

*RÚBRICA DE EVALUACIÓN*

*3RA ENTREGA DEL PROYECTO FINAL*

| **RÚBRICA: CRITERIOS PARA LA 3RA ENTREGA DEL PROYECTO FINAL** | | | |
| --- | --- | --- | --- |
| **Criterios** | **Bajo**  Falta más profundización. Es confuso. | **Correcto**  Acorde pero con errores puntuales. | **Óptimo**  Es claro y pertinente. |
| Funcionalidad | * No se simula un flujo de trabajo en términos de entrada, proceso y salida. * Entendemos a un flujo de trabajo como una secuencia ordenada de tareas dedicadas a solventar un problema. * Entradas: Uno o más valores ingresados por el usuario. * Salidas: Uno o más mensajes mostrados al usuario. * Proceso: Algoritmo cuyas instrucciones transforman las entradas en salidas esperadas. | * Se simula uno o más flujos de trabajo en términos de entrada,proceso y salida. * La funcionalidad es apropiada al contexto del simulador. | * Se simula uno o más flujos de trabajo en términos de entrada,proceso y salida. * La funcionalidad es apropiada es apropiada al contexto del simulador. * No se advierten errores de cómputo durante el procesamiento. |
| Interactividad | * No se capturan entradas ingresadas por el usuario mediante eventos. * No se efectúan salidas por HTML modificando el DOM. | * Se capturan entradas ingresadas por el usuario mediante eventos. * Se efectúan una o más salidas por HTML modificando el DOM. * Se emplean animaciones para destacar entradas y/o salidas. | * Se capturan entradas ingresadas por el usuario mediante eventos. * Se efectúan una o más salidas por HTML modificando el DOM. * Se emplean animaciones para destacar entradas y/o salidas. * Se emplea la totalidad de las funcionalidades del simulador sin refrescar la página. |
| Escalabilidad | * No se emplean funciones. * No se utiliza Array para agrupar datos símiles. * No se utilizan objetos para estructuras de datos con un mismo comportamiento. * No se establece un criterio homogèneo para la detecciòn de eventos. | * Se declaran funciones con parámetros para definir instrucciones con una tarea específica. * Se emplean arrays para agrupar valores relacionados. * Se definen objetos con propiedades y métodos relevantes al contexto. * Se establece un criterio homogèneo para la detecciòn de eventos | * Se declaran funciones con parámetros para definir instrucciones con una tarea específica. * Se emplean arrays para agrupar valores relacionados. * Se definen objetos con propiedades y métodos relevantes al contexto. * Se establece un criterio homogèneo para la detecciòn de eventos. * Se almacena en storage datos relevantes generados durante la simulaciòn. |
| Integridad | * Se define el código JavaScript dentro del documento HTML. | * Se define el código JavaScript en un archivo .js, referenciándolo correctamente desde el HTML. * Se referencian correctamente la integración de librerías (jQuery y otras) * La información estática del proyecto se emplea adecuadamente. | * Se define el código JavaScript en un archivo .js, referenciándolo correctamente desde el HTML. * Se referencian correctamente la integración de librerías (jQuery y otras). * La informaciòn estática del proyecto se emplea adecuadamente, cargándose de forma asíncrona. |
| Legibilidad | * No se establece un criterio de claridad en la definición de variables, funciones y objetos. | * Los nombres de variables. funciones y objetos son significativos para el contexto. * Las instrucciones se escriben de forma legible y se emplean comentarios oportunos. * Entendemos al código legible como aquel que está correctamente estructurado, preferentemente alineado e indentado. | * Los nombres de variables. funciones y objetos son significativos para el contexto . * Las instrucciones se escriben de forma legible y se emplean comentarios oportunos. * El código fuente es ordenado en términos de declaración y secuencia. |