

Estudiante:	mencionar acá el nombre y apellido del estudiante evaluado
Curso/Comisión:	Completar con el nombre del curso y comisión
Proyecto:	mencionar el nombre del proyecto del estudiante/ número de pre entrega

Checklist Pre- entrega

Consigna	Comienza a crear la estructura inicial de tu proyecto integrador. Deberás: - Pensar el alcance de tu proyecto: ¿usarás un cotizador de seguros? ¿un simulador de créditos? ¿un simulador personalizado? - Armar la estructura HTML del proyecto. - Incorporar algoritmo condicional y algoritmo con ciclo. - Utilizar funciones para realizar esas operaciones.
----------	---

Aspectos a evaluar	Descripción	Alcance	Comentarios	Devolución final																		
Estructura HTML y archivo JS	La estructura HTML está completa y ejecutada con buenas prácticas o utiliza el el starter template de Bootstrap u otro framework para el uso adecuado de HTML5 *. El archivo JS está correctamente referenciado en el HTML.			Completar																		
Algoritmos	Se utiliza algoritmo condicional y con ciclo (IF, bucles for) de manera óptima , reflejando lo aprendido en clase.																					
Funciones	Los nombres de las funciones son claros y dan a entender que acción realizan. Se emplea la estructura correcta para el armado de las mismas. Crea funciones dinámicas de manera correcta. Generan un resultado correcto cuando se ejecutan.																					
*Aclaración: Pueden utilizar el starter template de Bootstrap para agilizar el armado de la esrtuctura HTML en caso de que lo deseen. O pueden optar por aplicar JS a sus HTML desarrollados por los propios estudiantes (ya sea en otros cursos o para este proyecto).				<table><tr><th colspan="2">Puntaje asignado por alcance</th></tr><tr><td>Alcance por aspecto</td><td></td></tr><tr><td>Realizado</td><td>33</td></tr><tr><td>Incompleto</td><td>15</td></tr><tr><td>No realizado</td><td>0</td></tr><tr><th colspan="2">Nivel obtenido</th></tr><tr><td>80-100</td><td>Óptimo</td></tr><tr><td>51-79</td><td>Correcto</td></tr><tr><td>0-50</td><td>Bajo</td></tr></table>	Puntaje asignado por alcance		Alcance por aspecto		Realizado	33	Incompleto	15	No realizado	0	Nivel obtenido		80-100	Óptimo	51-79	Correcto	0-50	Bajo
Puntaje asignado por alcance																						
Alcance por aspecto																						
Realizado	33																					
Incompleto	15																					
No realizado	0																					
Nivel obtenido																						
80-100	Óptimo																					
51-79	Correcto																					
0-50	Bajo																					

CODERHOUSE



Estudiante:	mencionar acá el nombre y apellido del estudiante evaluado
Curso/Comisión:	Completar con el nombre del curso y comisión
Proyecto:	mencionar el nombre del proyecto del estudiante/ número de pre entrega

Checklist Pre- entrega

Consigna	Deberás entregar la estructura del proyecto, las variables de JS necesarias y los objetos de JS, correspondientes a la segunda entrega de tu proyecto final.
Objetivos	<p>Objetivos Generales</p> <ul style="list-style-type: none">- Codificar la funcionalidad inicial del simulador.- Identificar el flujo de trabajo del script en términos de captura de entradas ingresadas por el usuario, procesamiento esencial del simulador y notificación de resultados en forma de salida. <p>Objetivos Específicos</p> <ul style="list-style-type: none">- Capturar entradas mediante prompt().- Declarar variables y objetos necesarios para simular el proceso seleccionado.- Crear funciones y/o métodos para realizar operaciones (suma, resta, concatenación, división, porcentaje, etc).- Efectuar una salida, que es el resultado de los datos procesados, la cual puede hacerse por alert() o console.log().

Aspectos a evaluar	Descripción	Alcance	Comentarios	Devolución final																
Funcionalidad	Se simula un flujo de trabajo en términos de entrada, proceso y salida. La funcionalidad es apropiada al contexto del simulador.			Completar																
Interactividad	Se capturan una o más entradas ingresadas por el usuario con prompt(). Se efectúan una o más salidas con alert() o console.log(). Las salidas son coherentes en relación a los datos ingresados.																			
Escalabilidad	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.																			
Integridad	Se define el código JavaScript en un archivo .js, referenciándolo correctamente desde el HTML. Se emplea correctamente prompt() y alert() sin generar interrupciones del procesamiento.																			
Legibilidad	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 está correctamente estructurado, alineado e indentado.																			
				<table><tr><th>Alcance por aspecto</th><th>Puntaje asignado por alcance</th></tr><tr><td>Realizado</td><td>20</td></tr><tr><td>Incompleto</td><td>10</td></tr><tr><td>No realizado</td><td>0</td></tr><tr><th>Puntaje total</th><th>Nivel obtenido</th></tr><tr><td>80-100</td><td>Óptimo</td></tr><tr><td>51-79</td><td>Correcto</td></tr><tr><td>0-50</td><td>Bajo</td></tr></table>	Alcance por aspecto	Puntaje asignado por alcance	Realizado	20	Incompleto	10	No realizado	0	Puntaje total	Nivel obtenido	80-100	Óptimo	51-79	Correcto	0-50	Bajo
Alcance por aspecto	Puntaje asignado por alcance																			
Realizado	20																			
Incompleto	10																			
No realizado	0																			
Puntaje total	Nivel obtenido																			
80-100	Óptimo																			
51-79	Correcto																			
0-50	Bajo																			

CODERHOUSE



Estudiante:	mentonar acá el nombre y apellido del estudiante evaluado
Curso/Comisión:	Completar con el nombre del curso y comisión
Proyecto:	mentonar el nombre del proyecto del estudiante/ número de pre entrega

Checklist Pre- entrega

Consigna	Deberás agregar y entregar uso de JSON y Storage, y DOM y eventos del usuario, correspondientes a la tercera entrega de tu proyecto final.
Objetivos	<p>Objetivos Generales</p> <ul style="list-style-type: none">- Codificar funciones de procesos esenciales y notificación de resultados por HTML, añadiendo interacción al simulador.- Ampliar y refinar el flujo de trabajo del script en términos de captura de eventos, procesamiento del simulador y notificación de resultados en forma de salidas por HTML, modificando el DOM. <p>Objetivos Específicos</p> <ul style="list-style-type: none">- Definir eventos a manejar y su función de respuesta.- Modificar el DOM, ya sea para definir elementos al cargar la página o para realizar salidas de un procesamiento.- Almacenar datos (clave-valor) en el Storage y recuperarlos.

Aspectos a evaluar	Descripción	Alcance	Comentarios	Devolución final
Funcionalidad	Se simula uno o más flujos de trabajo en términos de entrada, proceso y salida. La funcionalidad es apropiada al contexto del simulador. Carece de errores de cómputo durante el procesamiento.			Completar
Interactividad	Se capturan entradas ingresadas por el usuario mediante eventos. Se efectúan una o más salidas por HTML modificando el DOM. Existe un control de ingreso de entradas y las salidas son coherentes en relación a los datos ingresados.			
Escalabilidad	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 homogéneo para la detección de eventos. Se almacena en storage datos relevantes generados durante la simulación			
Integridad	Se define el código JavaScript en un archivo .js, referenciándolo correctamente desde el HTML. Se evitan métodos prompt() y alert() para evitar interrupciones durante el procesamiento y actualización del DOM. La información estática del proyecto se emplea adecuadamente			
Legibilidad	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.			
Alcance por aspecto				Puntaje asignado por alcance
Realizado				20
Incompleto				10
No realizado				0
Puntaje total				Nivel obtenido
80-100				Óptimo
51-79				Correcto
0-50				Bajo



