



{JavaScript}



Proyecto Final: Introducción a JavaScript

Cajero Automático

Crea una aplicación web con JavaScript donde simulemos la interacción con un cajero automático.

Funcionamiento básico:

Al ingresar al cajero, puedes seleccionar la cuenta con la que deseas interactuar. Deben existir al menos tres cuentas:

- Persona 1
- Persona 2
- Persona 3

Para esto, puedes trabajar con un arreglo de objetos como el siguiente:

```
var cuentas = [  
  { nombre: "Mali", saldo: 200 },  
  { nombre: "Gera", saldo: 290 },  
  { nombre: "Maui", saldo: 67 },  
];
```

Al seleccionar una cuenta, debes ingresar el password asociado a la cuenta. Si el password es incorrecto, debes notificar al usuario y permitirle intentarlo nuevamente. Si el password es correcto, debes mostrar las siguientes opciones:

- Consultar saldo
- Ingresar monto
- Retirar Monto
- Al seleccionar consultar saldo, debe mostrar en pantalla el saldo actual de la cuenta
- Al seleccionar ingresar monto, el usuario debe escribir el monto a ingresar. Al ingresar el monto, debe mostrarle al usuario el monto ingresado y el nuevo saldo total.
- Al seleccionar retirar monto, el usuario debe escribir el monto a retirar. Al retirar el monto, debe mostrarle al usuario el monto retirado y el nuevo saldo total.
- Implementa un botón de salir de la sesión del usuario.

Como regla de negocio, una cuenta no debe de tener más de \$990 y menos de \$10. Es necesario hacer las validaciones pertinentes para que no se rompa esta regla de negocio.

Buenas prácticas:

- Utilizar las herramientas vistas en clase.
- Iterar correctamente el objeto de cuentas
- Accesar correctamente a los elementos de las cuentas
- Uso correcto de `const` y `let`
- Limpiar los inputs una vez que se hagan las operaciones pertinentes.
- Mensajes claros al usuario

Plus:

- Validación de los campos inputs para valores `null`, `Undefined` y `NaN`.

- Validación de cantidades negativas.
- Uso de algún método avanzado de iteración.
- Uso de Bootstrap
- Diseño Responsive
- Mensajes al usuario como parte de la interfaz.
- Uso de `localStorage` para guardar la persistencia de la información/data de los usuarios.



Nota: El proyecto debe estar en su propio repositorio, como buena práctica haz commit frecuentemente para guardar y reportar tus avances.

Dinámica de exposición de proyecto:

Cada estudiante cuenta con 7 min máximo para presentar su proyecto, dentro de este tiempo procura tocar los siguientes puntos:

- Demostración del funcionamiento del cajero:
 - Login (3 cuentas)
 - Mostrar saldo
 - Realizar operaciones (retiro y deposito)
 - Funcionamiento de las reglas de negocio
 - Validaciones y mensajes de usuario
 - Manejo de escenarios
- Presume tu diseño, muestra si es responsive, si usaste local storage, muestra en las dev tools el uso de este, etc.
- Pasa a la explicación de código: específicamente JavaScript, que es lo que vamos a evaluar. No es necesario mostrar y explicar tus archivos HTML o CSS.

- Menciona los highlights de tu proyecto, si usaste Bootstrap, en que te inspiraste para el diseño, etc.
- Deja espacio para preguntas y recibir feedback y agradecimiento por parte de tus mentores.

Rúbrica de Evaluación:

Concepto	Puntuación	
Login con 3 cuentas (Hace uso de array de objetos)	20%	
Validación de cuentas	5%	
Se puede retirar monto correctamente	5%	
Se puede depositar monto correctamente	5%	
Se actualiza el saldo con las transacciones	5%	
Se cumplen las reglas de negocio	20%	
Hay un botón de salir/cerrar sesión	5%	
CSS (Bootstrap aún mejor)	5%	
JavaScript best practices (buen uso del lenguaje y código limpio) ES6 - métodos avanzados de iteración	10%	
Buen uso del DOM e Interfaz intuitiva	10%	
Mensajes de error al usuario (dentro del DOM)	5%	
Desplegado en GitHub Pages	5%	
Total	100%	