

Ejercicios propuestos

Desarrollar la actividad en lenguaje de programación JavaScript usando alguna estructura de datos como de array, map set u objetos.

Ejercicio 1:

El software que se desarrollará controlará un cajero automático (ATM) a través de una simulación usando el lenguaje de programación JavaScript.

El cajero automático atenderá a un cliente a la vez. Se le pedirá al cliente que ingrese su documento de identidad y su pin de 4 dígitos, los cuales se enviarán al banco para su validación como parte de cada transacción. El cliente podrá entonces realizar una o más transacciones. El menú se mostrará en la consola hasta que el cliente indique que no desea realizar más transacciones.

El cajero automático debe ser capaz de proporcionar los siguientes servicios al cliente:

- Un cliente debe poder realizar un retiro de efectivo de cualquier cuenta vinculada al documento de identidad, en múltiplos de \$50000. Se debe obtener la aprobación del banco antes de entregar efectivo.
- Un cliente debe poder realizar un depósito en cualquier cuenta vinculada al documento de identidad, consistente en efectivo y/o cheques. El cliente ingresará el monto del depósito en el cajero automático e indicar si es efectivo o cheque.
- Un cliente debe poder realizar una transferencia de dinero entre dos cuentas cualquiera que están vinculadas a al documento de identidad.
- Un cliente debe poder realizar una consulta de saldo de cualquier cuenta vinculada al documento de identidad.
- El cajero automático comunicará al cliente los resultados de cada transacción dependiendo de su tipo. Ejemplo “retiro exitoso, puede tomar x dinero de la bandeja principal”
- Si el banco determina que el PIN del cliente no es válido, se le pedirá al cliente que vuelva a ingresar el PIN antes de que se pueda continuar con la transacción. Si el cliente no puede ingresar correctamente el PIN después de tres intentos saldrá de la aplicación.

Ejercicio 2:

Desarrollar en JavaScript un programa para la gestión reservas de un hotel, el cual, debe tener las siguientes características y consideraciones:

1. Un cliente puede reservar cualquier tipo de habitación: individual, doble y familiar.
2. Las habitaciones pueden ser para fumadores o no fumadores.
3. Las mascotas solo se aceptan en habitaciones familiares.
4. No se puede exceder el número de personas por habitación: individual 2 personas, 4 personas para doble y 6 personas para familiar.
5. Cada reserva tiene: nombre de quien reserva, país de origen, número de personas, el periodo de la estadía, número de personas que están ocupando el hotel y si el huésped trajo mascota.
6. Mencionar cuantas habitaciones tiene reservado el hotel.

Ejercicio 3:

Construya un programa en JavaScript para las estadísticas de atención de una universidad teniendo en cuenta los siguientes requisitos:

1. Se le solicita al usuario: numero de cedula y tipo de atención.
2. Hay dos módulos de atención: terminal para llamada telefónica y asesoría.
3. En la asesoría hay dos tipos: Estudiante o directivo.
4. El sistema brinda las estadísticas de todo el proceso de atención:
 - Cantidad de usuarios atendidos.
 - cantidad de usuarios por tipo.
5. Se permite hacer transferencia de asesoría a llamada telefónica, pero en la estadística cuenta como atención diferente

Ejercicio 4:

Descripción:

Imagina que estás desarrollando un sistema de gestión de turnos para un banco. El objetivo es crear una aplicación en JavaScript que permita a los clientes tomar un turno y a los empleados del banco llamar a los clientes en orden. Aquí tienes las funciones que debes implementar:

- Tomar un Turno: Los clientes deben poder tomar un turno al llegar al banco. Cada turno debe tener un número único y se debe agregar a una cola (cola de espera).
- Llamar a un Cliente: Los empleados del banco deben poder llamar al siguiente cliente en la cola. Debes mostrar el número del turno llamado y eliminarlo de la cola.
- Mostrar la Cola de Espera: Debe existir una función para mostrar la cola de espera actual, de modo que los clientes puedan ver cuántos clientes están antes que ellos.
- Contador de Turnos: Implementa un contador que muestre cuántos turnos se han tomado hasta el momento.

Requisitos:

Utiliza funciones y objetos en JavaScript para organizar la lógica.

Puedes utilizar arrays para representar la cola de espera y llevar un registro de los turnos.

Este ejercicio pondrá a prueba tus habilidades en JavaScript y te ayudará a comprender cómo gestionar colas y eventos en una aplicación web. ¡Buena suerte!

Ejercicio 5:

Descripción:

Imagina que estás desarrollando una aplicación que simule un carrito de compras en JavaScript. El objetivo es crear una aplicación que permita a los usuarios seleccionar productos, agregarlos a un carrito y calcular el total de la compra. Aquí están los requisitos que debes cumplir:

Cada producto debe tener un nombre, un precio y una cantidad en stock.

Carrito de Compras: Implementa un carrito de compras que mantenga un registro de los productos seleccionados por el usuario. Debe mostrar los productos agregados, la cantidad y el precio subtotal de cada producto.

Total de la Compra: Calcula y muestra el precio total de todos los productos en el carrito.

Ejercicio 6:

Descripción:

Imagina que estás creando un sistema de gestión de citas médicas en JavaScript. El objetivo es desarrollar una aplicación que permita a los usuarios programar y ver citas médicas. Aquí están los requisitos que debes cumplir:

Programar Citas: Los usuarios deben poder programar citas médicas proporcionando información como el nombre del paciente, la fecha y la hora de la cita, y el médico asignado.

Ver Citas Programadas: Debe existir una función que permita a los usuarios ver todas las citas programadas. Las citas deben mostrarse en orden de fecha y hora.

Cancelar Citas: Los usuarios deben poder cancelar citas médicas existentes. Al cancelar una cita, debe eliminarse de la lista de citas programadas.

Almacenamiento de Datos: Utiliza arreglos y objetos para almacenar y gestionar la información de las citas.

Requisitos:

Utiliza funciones en JavaScript para organizar la lógica del sistema de gestión de citas.

Crea una interfaz en la consola que permita a los usuarios interactuar con la aplicación.