
Universidad Tecnológica Metropolitana



Materia: Estructura de Datos Aplicadas

Nombre del Docente: Canche Caamal Miriam Magaly

Nombre de los Integrantes:

- **Arellano Blancarte Andrea Carolina**
 - **Ciau Dzul Josué Geovani**
 - **Ek Pech Luis Alfonso**

Actividad: Actividad2-Ejecicios-Colas-EDA2P

Grupo: 4-D

Parcial: 2do

Fecha de entrega: 25/10/24

Indice

| | |
|---|----|
| DOCUMENTACION DEL PROYECTO EN HTML, CSS Y JAVASCRIPT (COLAS) | 3 |
| Descripción general del proyecto: | 3 |
| Navegación del Sitio | 3 |
| Interfaz de Usuario y Funcionalidades | 3 |
| Banco | 3 |
| Juego de Pintar Coches | 6 |
| Estacionamiento | 8 |
| Código General y Funcionamiento | 10 |
| Descripción General | 10 |
| 1. Estructura del HTML Principal (index.html) | 10 |
| Explicación del HTML Principal | 11 |
| 2. Estilos en CSS (styles.css) | 11 |
| Explicación de Estilos CSS | 12 |
| 3. Lógica en JavaScript y Validaciones | 12 |
| Archivo app.js | 12 |
| Código Base de app.js | 12 |
| Archivo scriptestacionamiento.js | 13 |
| Código Base de scriptestacionamiento.js | 13 |
| Archivo scriptjuegodepintarcoche.js | 13 |
| Código Base de scriptjuegodepintarcoche.js | 14 |
| Conclusión | 14 |

DOCUMENTACION DEL PROYECTO EN HTML, CSS Y JAVASCRIPT (COLAS)

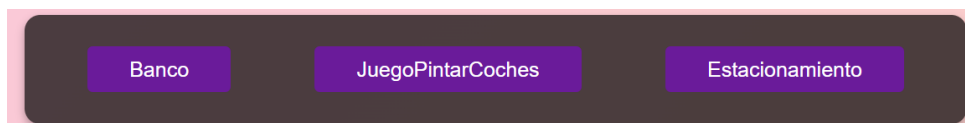
Descripción general del proyecto:

Este proyecto web fue desarrollado utilizando HTML, CSS y JavaScript para simular un sistema de atención mediante colas, con una interfaz interactiva y visualmente atractiva. La aplicación ofrece tres funcionalidades principales a través de un menú de opciones: Banco, Juego de Pintar Coches, y Estacionamiento. Cada sección está diseñada para representar y manipular colas de diferentes maneras, con énfasis en facilitar la comprensión de estructuras de datos en escenarios prácticos.

Navegación del Sitio

Al acceder al sitio web, el usuario es recibido con una interfaz de navegación que muestra tres opciones en el menú principal:

- **Banco:** Simula un sistema de atención en ventanilla para clientes.
- **Juego de Pintar Coches:** Implementa una fila de espera para coches que serán pintados, aplicando colores de manera aleatoria.
- **Estacionamiento:** Muestra un control de vehículos que ingresan y salen de un estacionamiento en orden de llegada.



Interfaz de Usuario y Funcionalidades

Banco

La opción inicial y principal del sitio es el apartado de **Banco**. En esta sección, el usuario tiene la posibilidad de agregar clientes a la cola para ser atendidos en ventanilla y de atenderlos en orden de llegada (FIFO - First In, First Out).

Estructura y Diseño de la Interfaz

The mockup shows a web interface for a bank queue system. At the top, there are three navigation buttons: 'Banco' (selected), 'JuegoPintarCoches', and 'Estacionamiento'. Below this is a main container titled 'Gestión de una Cola'. Inside this container, there are two input fields: 'Numero de turno' with the value '1' and 'Nombre del cliente' with a placeholder 'Nombre del cliente'. Below these is a 'Movimiento' section with a dropdown menu currently showing 'Pago de Servicio'. At the bottom of the form area are two buttons: 'Agregar a la Cola' and 'Atender en Ventanilla'. Below the entire form is a table with five columns: 'Turno', 'Cliente', 'Movimiento', 'Fecha', and 'Hora'.

La interfaz del banco está dividida en los siguientes elementos visuales:

- **Formulario de Entrada:** Un campo de texto donde el usuario puede ingresar el nombre del cliente, junto con un botón "Agregar a la Cola".
- **Visualización de la Cola:** Se muestra la lista actual de clientes en la cola, incluyendo el nombre del cliente y el tipo de operación (por ejemplo, "Depósito", "Retiro" o "Consulta").
- **Botón de Atención:** Permite atender al primer cliente de la fila, eliminándolo de la cola una vez que ha sido atendido en la ventanilla.

Funcionalidad del Banco

- **Agregar Cliente a la Cola:** Al ingresar un nombre y seleccionar un tipo de operación, el usuario puede agregar un cliente a la cola. Al hacer clic en el botón "Agregar a la Cola", el cliente ingresado se muestra en la lista de espera con el tipo de operación que solicitó.

Banco

127.0.0.1:5501 dice
Cliente agregado:
Nombre: Andy
Turno: 1
Movimiento: Deposito
Fecha: 25/10/2024
Hora: 19:45:34

Atencionamiento

Gestión de una Cola

Aceptar

Numero de turno

Nombre del cliente

1

Andy

Movimiento

Depósito

Agregar a la Cola

Atender en Ventanilla

| Turno | Cliente | Movimiento | Fecha | Hora |
|-------|---------|------------|------------|----------|
| 1 | Andy | Deposito | 25/10/2024 | 19:45:34 |

- **Atender en Ventanilla:** Cuando el usuario hace clic en el botón "Atender en Ventanilla", el primer cliente en la cola es atendido y eliminado de la lista, actualizando automáticamente la cola visualizada en pantalla. Este proceso ayuda a simular el flujo real de atención en una ventanilla de banco, donde el cliente más antiguo en la cola es el primero en ser atendido.

Banco

127.0.0.1:5501 dice
Cliente descolado:
Nombre: Andy
Tiempo en cola: 0 horas, 1 minutos, 41 segundos

Atencionamiento

Gestión de una Cola

Aceptar

Numero de turno

Nombre del cliente

2

Andy

Movimiento

Depósito

Agregar a la Cola

Atender en Ventanilla

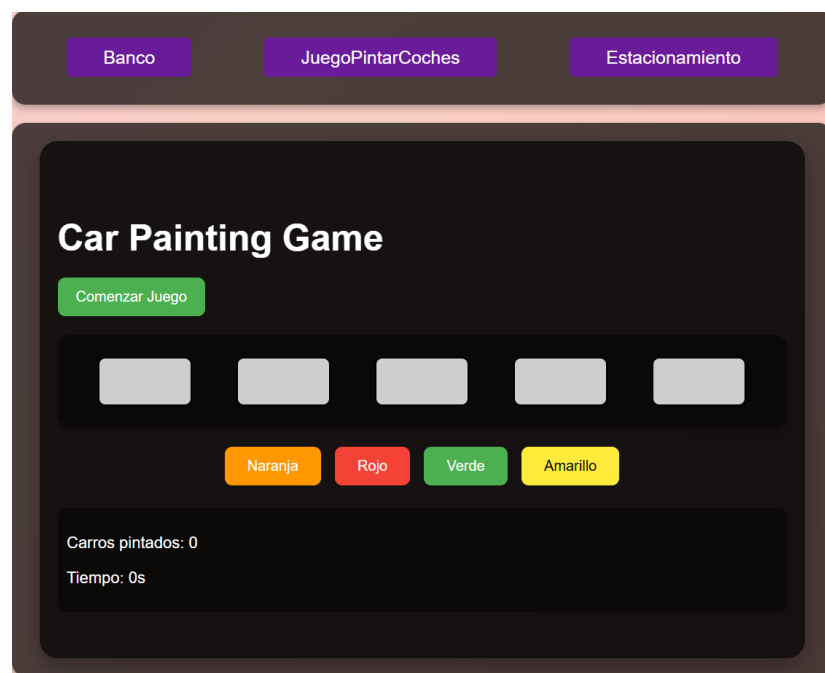
| Turno | Cliente | Movimiento | Fecha | Hora |
|-------|---------|------------|------------|----------|
| 1 | Andy | Deposito | 25/10/2024 | 19:41:54 |

| Turno | Cliente | Movimiento | Fecha | Hora |
|-------|---------|------------|-------|------|
|-------|---------|------------|-------|------|

Juego de Pintar Coches

En la sección **Juego de Pintar Coches**, se implementa una cola de vehículos que están esperando ser pintados. Esta sección es interactiva y permite a los usuarios agregar vehículos a la cola y ver cómo se pintan de colores aleatorios al ser atendidos.

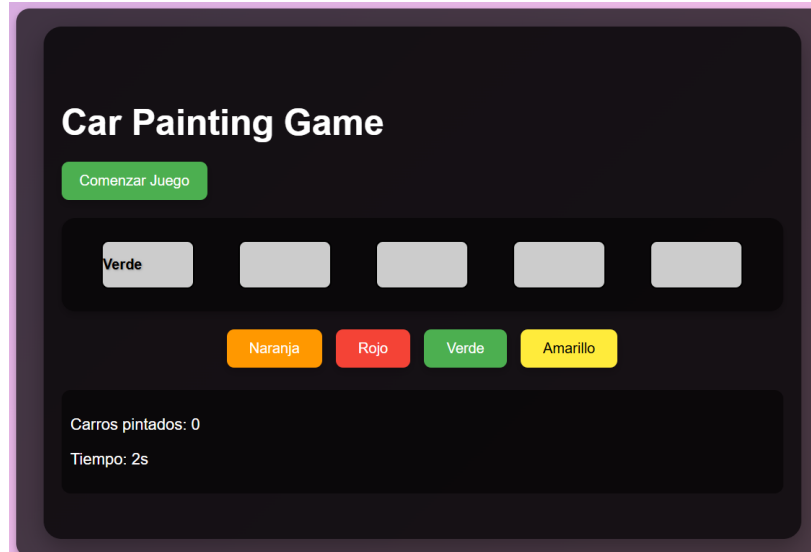
Estructura de la Interfaz



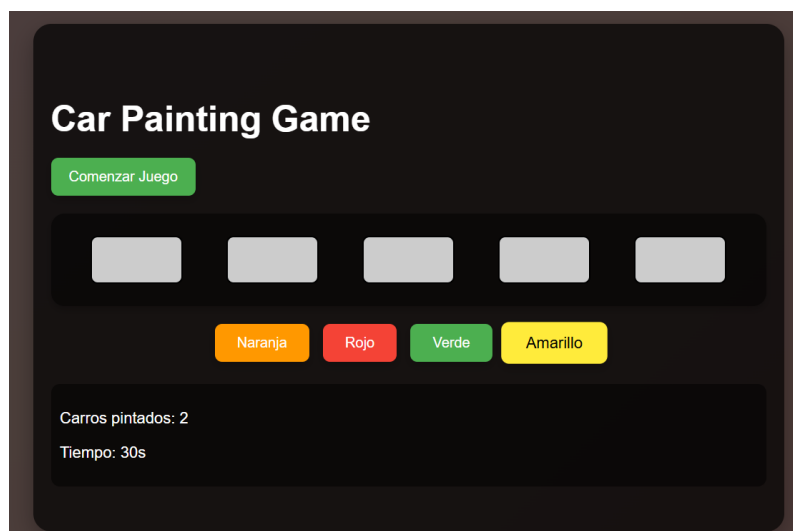
- **Comenzar juego:** Este botón ejecuta el juego de manera que aparece los carros cada 20 segundos
- **Entrada de Vehículos:** Apartados donde se le hará visible el nombre del color que deberá de ser pintados
- **Temporizador y carros que pinta:** muestra de manera que el usuario tenga en cuenta la cuenta de los carros que tiene que pintar.
- **Botón de Pintar:** Este botón cambia el color del primer vehículo en la cola, aplicando un color aleatorio para simular el proceso de pintura.

Funcionalidad del Juego

- **Comenzar juego:** Automáticamente empieza la cuenta y los carros entran cada 20 seg. Dando tiempo de seleccionar el color que corresponde.



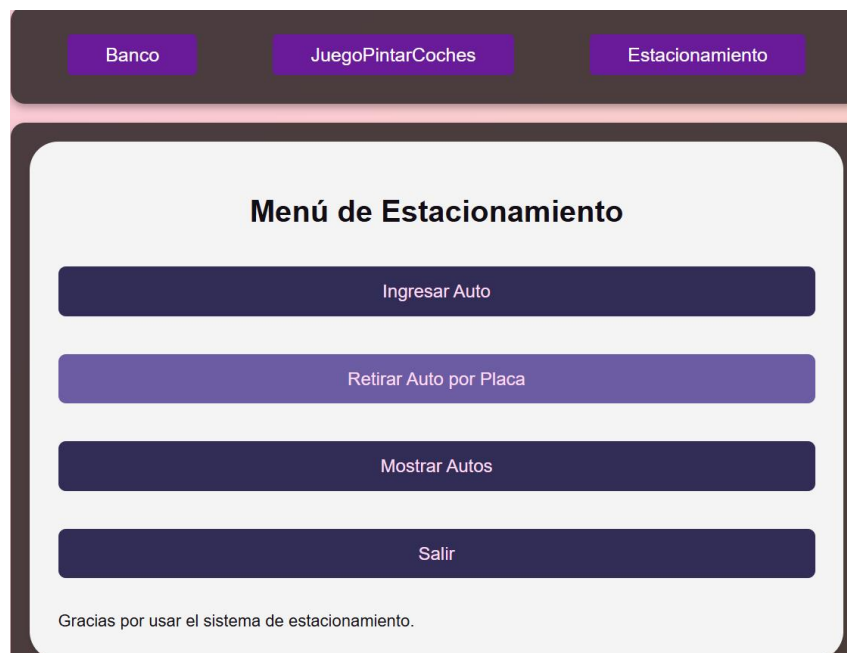
- **Pintar Vehículo:** Al presionar el botón de pintura, el primer vehículo en la cola es pintado de un color aleatorio y luego removido de la cola, simulando el paso de un coche pintado y la continuación con el siguiente en espera.



Estacionamiento

La sección de **Estacionamiento** permite gestionar una cola de vehículos que ingresan y salen del estacionamiento en orden de llegada. Es ideal para visualizar cómo funciona la lógica FIFO en la administración de un espacio de estacionamiento.

Estructura de la Interfaz



- **Agregar Vehículo:** Permite registrar un vehículo en la cola, solicitando la matrícula y el nombre del propietario. Los vehículos se almacenan en una lista en el orden en que se agregan.
- **Retirar Vehículo por Matrícula:** Permite retirar un vehículo del estacionamiento ingresando su matrícula, sin requerir el nombre del propietario. Esta opción elimina el vehículo identificado en cualquier posición de la cola.
- **Mostrar Autos:** Visualiza la lista completa de vehículos en el estacionamiento, mostrando la matrícula y el nombre del propietario para cada uno.

Funcionalidad del Estacionamiento

- **Formulario de Agregar Vehículo:** Contiene un campo de texto para la matrícula del vehículo, otro para el nombre del propietario, y un botón "Agregar" que añade el vehículo a la cola.



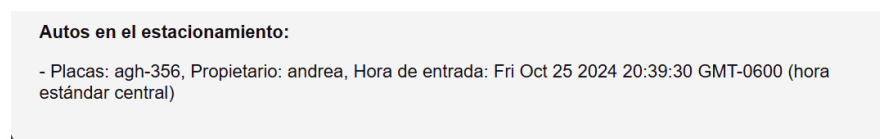
A screenshot of a web form titled "Ingreso de Auto". It features two light blue input fields: the first is labeled "Placas (XXX-XXX)" and the second is labeled "Nombre del propietario". Below these fields is a dark blue button with the text "Confirmar Ingreso".

- **Campo para Retirar por Matrícula:** Contiene un campo para ingresar la matrícula del vehículo a retirar y un botón "**Retirar**" que lo elimina de la lista.



A screenshot of a web form titled "Retiro de Auto". It has a light blue input field containing the text "agh-356". Below this field is a light gray box containing the following text: "Auto retirado:", "Placas: agh-356", "Propietario: andrea", "Hora de entrada: Fri Oct 25 2024 20:39:30 GMT-0600 (hora estándar central)", "Hora de salida: Fri Oct 25 2024 20:40:51 GMT-0600 (hora estándar central)", "Tiempo estacionado: 80.63 segundos", and "Costo: \$161.26 pesos".

- **Mostrar Autos:** Al hacer clic, muestra la lista completa de vehículos en el estacionamiento, incluyendo matrícula y nombre del propietario.



A screenshot of a web form titled "Autos en el estacionamiento:". It displays a single line of text: "- Placas: agh-356, Propietario: andrea, Hora de entrada: Fri Oct 25 2024 20:39:30 GMT-0600 (hora estándar central)".

Código General y Funcionamiento



Descripción General

El proyecto está dividido en:

1. **HTML Principal:** Es la estructura base de la interfaz de usuario.
2. **Archivos CSS:** Para el diseño y estilos de cada sección.
3. **Archivos JavaScript:** Para la lógica de cada una de las tres funcionalidades (Banco, Estacionamiento y Juego de Pintar Coches), que incluyen validaciones específicas.

Esta organización en módulos permite una mejor organización, facilita el mantenimiento y la reutilización del código.

1. Estructura del HTML Principal (index.html)

Este archivo HTML contiene la estructura base de la interfaz de usuario. Se incluyen tres secciones principales para cada funcionalidad (Banco, Estacionamiento y Juego de Pintar Coches), y cada una tiene sus propios campos de entrada, botones y áreas de visualización.

(Parte del código de html)

Explicación de Estilos CSS

- **Estilos Globales:** Se define el fondo, la fuente y los colores de texto para todo el proyecto.
- **Encabezado y Secciones:** Cada sección tiene un diseño con bordes redondeados y sombra para una mejor estética.
- **Botones:** Los botones tienen un diseño uniforme, con hover para una mejor experiencia de usuario.

3. Lógica en JavaScript y Validaciones

Archivo app.js

Este archivo maneja la funcionalidad de cola para el banco, permitiendo agregar y atender clientes.

Validaciones

- **Campos Vacíos:** Verifica que el usuario ingrese todos los datos requeridos antes de agregar a la cola.
- **Actualización Automática de Cola:** Al agregar o atender un cliente, la lista de cola se actualiza automáticamente en la interfaz.
- **Entre otros...**

Código Base de app.js

```
class Cola {
  constructor() {
    this.items = [];
  }

  // Añadir un elemento a la cola
  encolar(Nturno, Cliente, Movimiento, Fecha, Hora) {
    const tiempoAgregado = new Date(); // Guardar el tiempo de adición
    this.items.push({ Nturno, Cliente, Movimiento, Fecha, Hora, tiempoAgregado });
  }

  // Eliminar el primer elemento de la cola
  desencolar() {
    if (this.estaVacia()) {
      return null; // Cambiar a null para que sea más claro que no hay elementos
    }
    return this.items.shift(); // Devuelve el elemento desencolado
  }

  // Mostrar el primer elemento de la cola sin eliminarlo
  frente() {
    if (this.estaVacia()) {
      return null;
    }
    return this.items[0];
  }

  // Verificar si la cola está vacía
  estaVacia() {
    return this.items.length === 0;
  }

  // Mostrar todos los elementos de la cola en una tabla
  mostrarCola() {
    const tbody = document.querySelector("#tablaCola tbody");
    tbody.innerHTML = ''; // Limpiar la tabla
  }
}
```

(Parte del código js)

Archivo scriptestacionamiento.js

Este archivo gestiona la funcionalidad de estacionamiento, con opciones para agregar vehículos, retirarlos por matrícula y mostrar todos los autos en la cola.

Validaciones

- **Matrícula Duplicada:** Al agregar un auto, valida que la matrícula no exista ya en el estacionamiento.
- **Campo Vacío:** Al retirar un vehículo, verifica que el campo de matrícula esté lleno.

Código Base de scriptestacionamiento.js

```
1 let autos = [];  
2  
3 function mostrarFormularioIngreso() {  
4   document.getElementById('formularioIngreso').style.display = 'block';  
5   document.getElementById('formularioRetiro').style.display = 'none';  
6 }  
7  
8 function mostrarFormularioRetiro() {  
9   document.getElementById('formularioIngreso').style.display = 'none';  
10  document.getElementById('formularioRetiro').style.display = 'block';  
11 }  
12  
13 function ingresarAuto() {  
14   const placas = document.getElementById('placas').value.trim();  
15   const propietario = document.getElementById('propietario').value.trim();  
16   const regexPlacas = /^[A-Za-z0-9]{3}-[A-Za-z0-9]{3}$/;  
17  
18   if (!regexPlacas.test(placas)) {  
19     alert('Las placas deben tener el formato XXX-XXX (letras o números).');  
20     return;  
21   }  
22  
23   if (!propietario) {  
24     alert('El nombre del propietario no puede estar vacío.');25     return;  
26   }  
27  
28   const autoExistente = autos.find(auto => auto.placas.toLowerCase() === placas.toLowerCase());  
29   if (autoExistente) {  
30     alert('Ya existe un auto con las placas ${placas} en el estacionamiento.');31     return;  
32   }  
33  
34   const auto = {  
35     placas: placas,  
36     propietario: propietario,  
37     horaEntrada: new Date()  
38   }  
39   autos.push(auto);  
40   alert('Auto ingresado exitosamente.');41 }
```

Archivo scriptjuegodepintarcoche.js

Este archivo implementa la lógica del juego de pintar coches, donde el usuario puede seleccionar colores y aplicarlos a los coches en pantalla.

Validaciones

- **Colores Válidos:** Verifica que el color seleccionado sea válido.
- **Cambio Dinámico de Colores:** Aplica los colores seleccionados a los coches en tiempo real para mejorar la interacción.

Código Base de scriptjuegodepintarcoche.js

```
mas > scriptJuegoPintarCoche.js > ...
1  let carQueue = [];
2  let paintedCars = 0;
3  let timer = 0;
4  let interval;
5  let paintingSpeed = 20000; // 20 segundos
6  let gameActive = false; // Cambiado a false por defecto
7  const maxCars = 5;
8
9  // Inicializar el Juego
10 function initGame() {
11   carQueue = [];
12   paintedCars = 0;
13   timer = 0;
14   paintingSpeed = 20000;
15   gameActive = false; // Cambiado a false por defecto
16   document.getElementById('score').innerText = 'Carros pintados: 0';
17   document.getElementById('timer').innerText = 'Tiempo: 0s';
18   clearInterval(interval);
19 }
20
21 // Actualizar el Temporizador
22 function updateTimer() {
23   if (!gameActive) return;
24   timer++;
25   document.getElementById('timer').innerText = `Tiempo: ${timer}s`;
26 }
27
28 // Añadir un Carro a la Cola
29 function enqueueCar() {
30   if (carQueue.length >= maxCars) {
31     endGame();
32     return;
33   }
34   const newCar = {
35     color: getRandomColor(),
36     painted: false,
37   };
38 }
```

Conclusión

La estructura de este proyecto, organizada en un HTML principal y archivos JavaScript y CSS separados, permite un desarrollo modular y ordenado. Cada sección maneja sus propios archivos de lógica y estilos, lo que facilita la implementación de funcionalidades específicas, su mantenimiento y el uso de validaciones que mejoran la experiencia y seguridad para el usuario.