C#

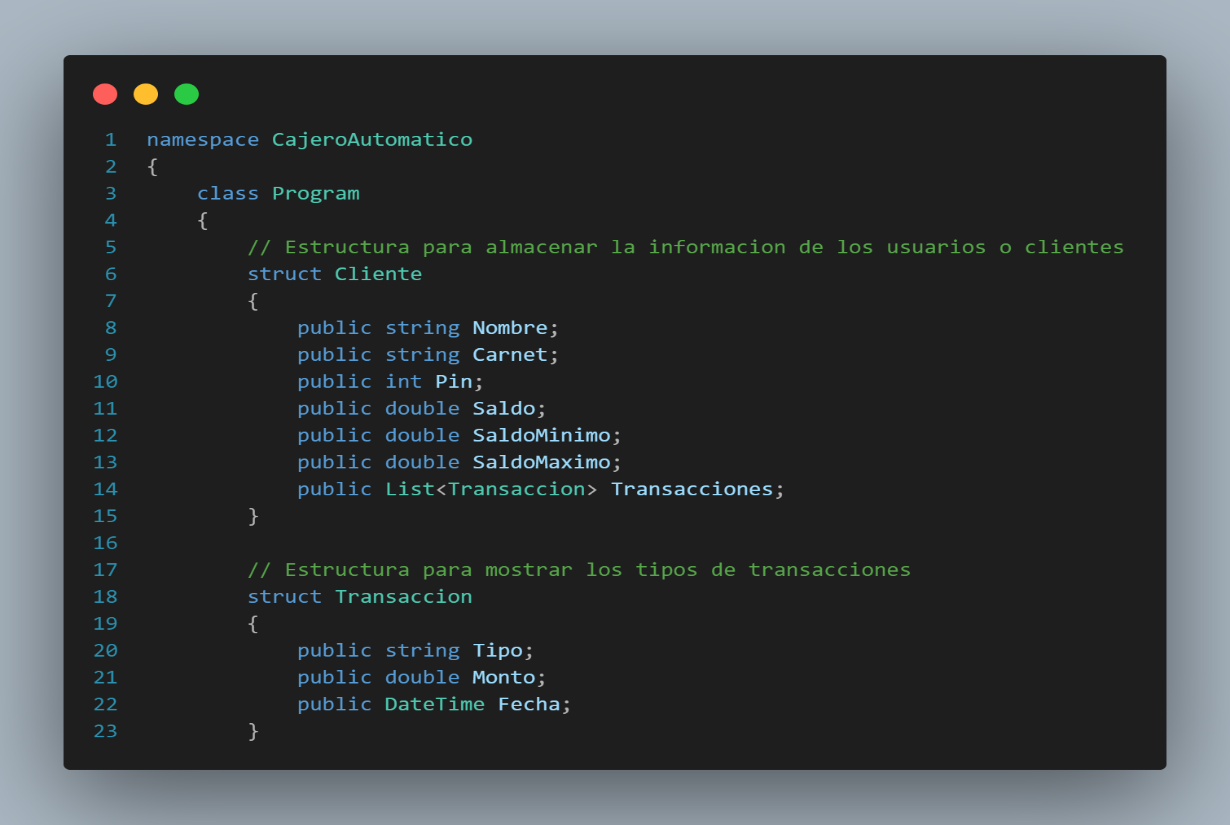
Sistema Básico de Cajero Automático

**Estructura Cliente (Objeto)**

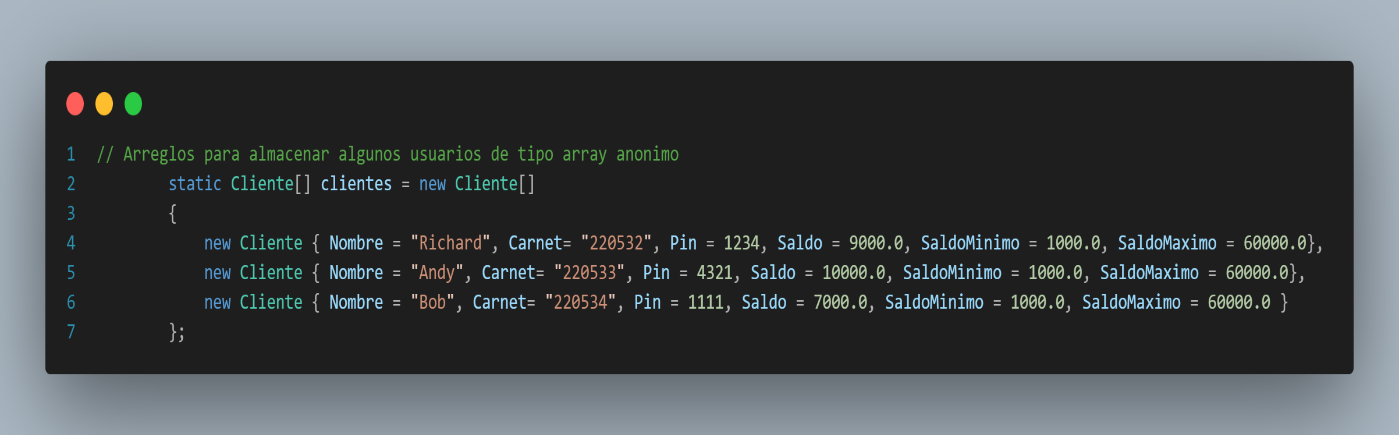
En la Estructura Cliente la hemos utilizado para almacenar información relacionada con los usuarios o clientes.

**Estructura Transaccion (Objeto)**

La Estructura Transacción representamos la información sobre las transacciones realizadas por el usuario mediante listas. A continuacion se muestra en pantalla parte del código:

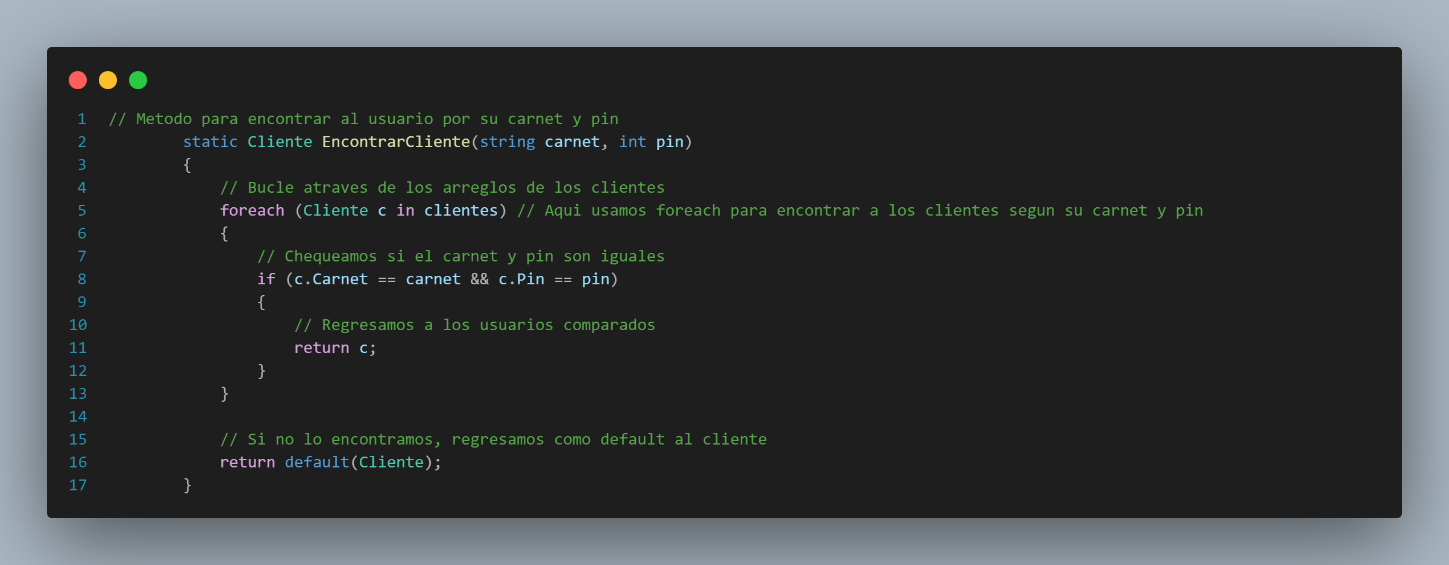


**Arreglos de Usuarios**

Hemos creado arreglos para almacenar la información de algunos usuarios. Estos usuarios están representados por la estructura Cliente, por el cual hemos hecho mención previa. Cada usuario tiene los siguientes datos: Nombre, Carnet, Pin, Saldo o Balance, Saldo Mínimo y Saldo Máximo.

**Método de Búsqueda de Usuarios**

Hemos creado un método llamado EncontrarCliente() que tiene como objetivo buscar a un usuario en específico por su número de carnet y PIN. Aquí está cómo funciona:

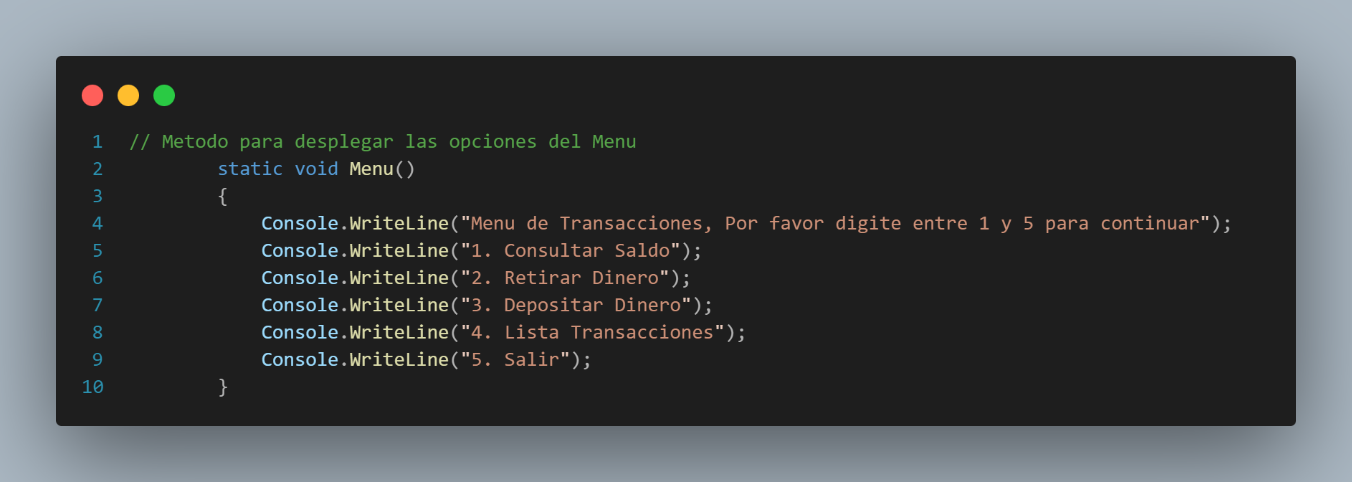
Funcionamiento: El método utiliza un bucle foreach estudiado en **Clase 9** para buscar en los arreglos de usuarios almacenados. Verifica si el número de carnet y el PIN del usuario coinciden con los proporcionados. **Usamos la CONDICIÓN IF para evaluar**; Si hay coincidencia, el método devuelve ese cliente, si no hay coincidencia, el método devuelve un valor predeterminado (default(Cliente)).

**Método de Mensaje en pantalla (Menú)**

Hemos hecho un método llamado Menu() para mostrar las opciones disponibles sobre las transacciones.

Salida en Consola:

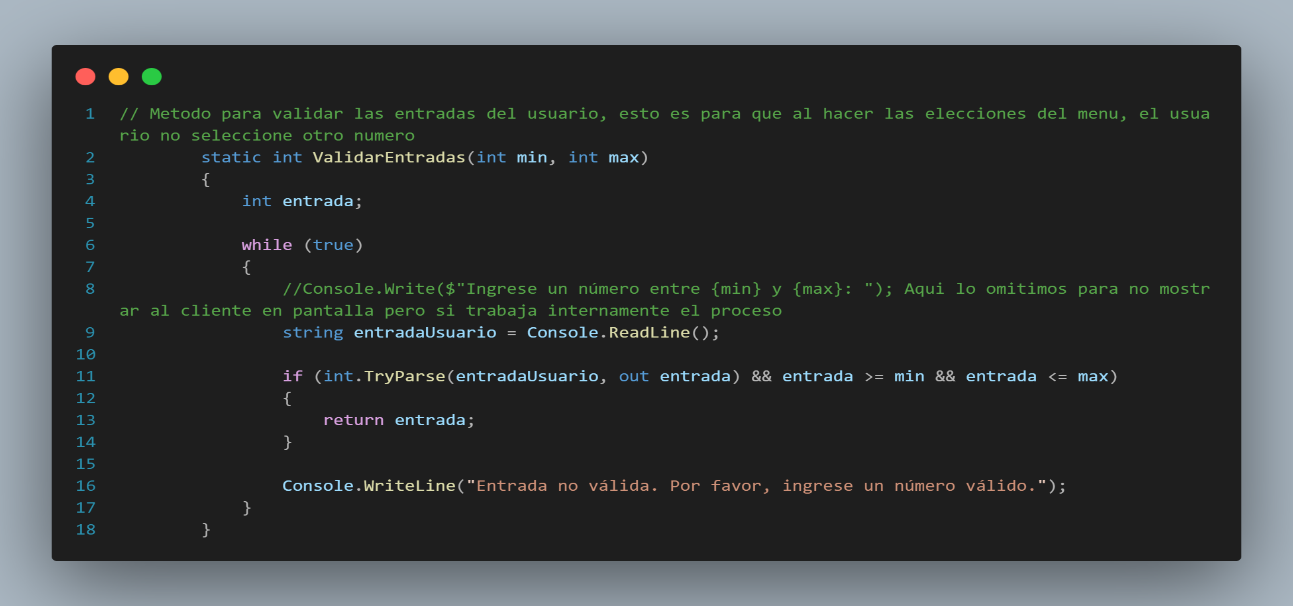
- Este código imprime en la consola un mensaje inicial indicando que es el "Menú de Transacciones" y solicita al usuario que ingrese un número entre 1 y 5 para seleccionar una opción.



**Método de Validación de Entradas (Lo que el Usuario digita)**

El método ValidarEntradas() se encarga de asegurar que el usuario ingrese un número válido dentro de un rango específico

Parámetros: **min**: representa el valor mínimo permitido. **max**: representa el valor máximo permitido, es decir, si imprime el menu en pantallas y tiene 5 opciones, el usuario tiene permitido de seleccionar **números entre 1 y 5**.

Funcionamiento: Utiliza un bucle **while** para mantener la solicitud de entrada hasta que se ingrese un valor válido. Lee y convierte cada entrada del usuario a números enteros y verifica si está dentro del rango especificado. Si la entrada es válida obviamente seguirá el proceso, pero si no es válida, muestra un mensaje de error y vuelve a solicitar una entrada válida.

**Método de Procesamiento de Transacciones**

Éste método ProcesoTransacciones() maneja las operaciones de transacciones para un cliente.

Parámetro: (tipo Cliente): Representa al usuario para el cual se realizan las transacciones.

Funcionamiento:

- Inicializa una lista de transacciones para el cliente.

- Utiliza un bucle **do-while** para repetir el proceso hasta que el usuario decida salir.

- Muestra el menú de transacciones utilizando el método **Menu().**

- Solicita al usuario que elija una opción y valida la entrada usando el método **ValidarEntradas().**

- Utilizando un bloque **switch** para cada caso del menú.

- Para opciones como: retiro y depósito, registra la transacción.

- Muestra la lista de transacciones en caso de seleccionar la opción 4.

- Usamos **do-while** porque estamos **haciendo** las operaciones del menus **mientras** pregunta al usuario si desea realizar otra transacción y continúa o sale del bucle según la respuesta.



**Método Principal**

El método Main() se compone del punto de entrada principal del programa y hace la autenticación de usuarios en el cajero automático.

Inicio del Programa:

- Muestra un mensaje de bienvenida al Cajero Automático.

Autenticación del Usuario:

- Utiliza un bucle **while** para solicitar el carnet y el PIN del usuario hasta tres intentos.

- Llama al método EncontrarCliente() para obtener el objeto **Cliente** asociado al carnet y PIN.

- Si la autenticación es exitosa, inicializa la lista de transacciones si es nula y pasa al proceso de transacciones utilizando el método ProcesoTransacciones().

Manejo de Intentos y Bloqueo:

- Contabiliza los intentos fallidos e imprime el número restante de intentos.

- Si se exceden los intentos permitidos, muestra un mensaje indicando que la cuenta ha sido bloqueada.



**Método de Mostrar Transacciones**

El método MostrarTransacciones() se encarga de imprimir en consola el historial de transacciones de un cliente por media de las listas que hemos inicializado en un principio.

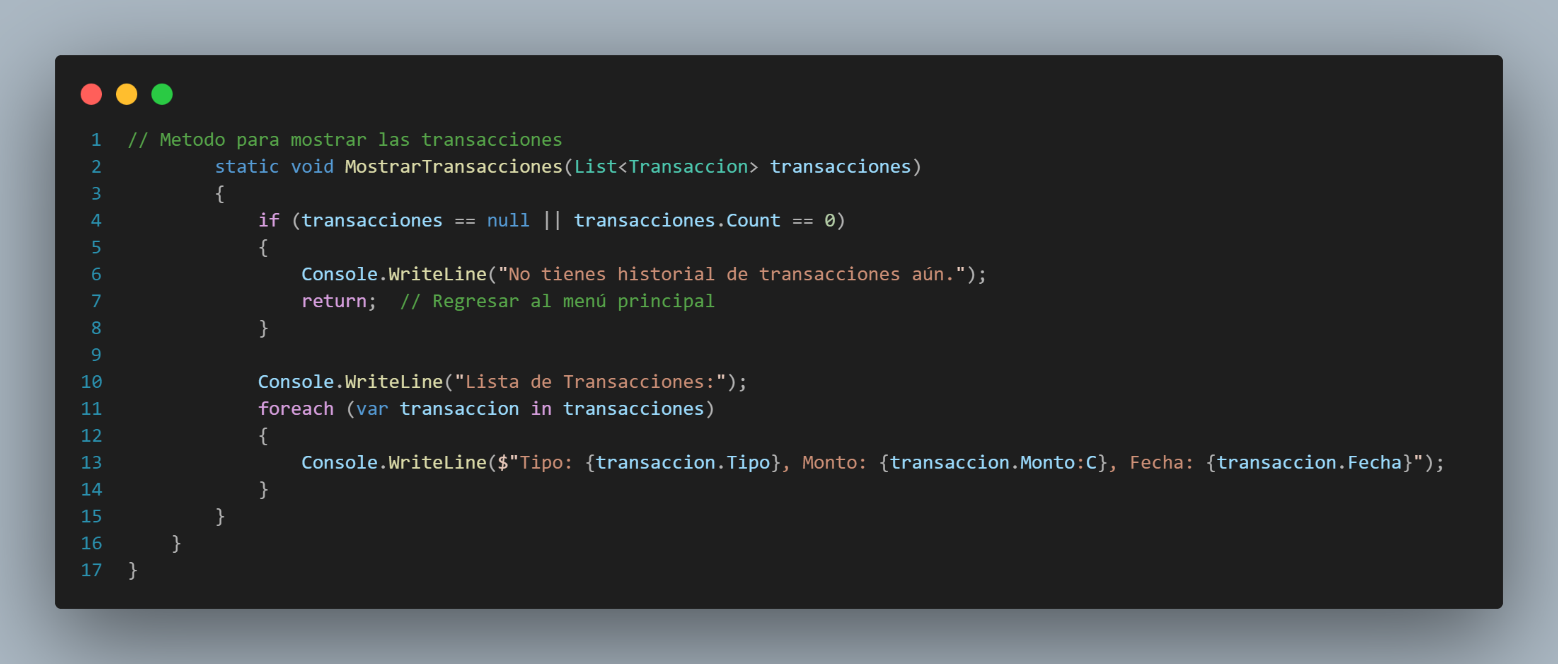
Parámetro:

- transacciones (List<Transaccion>): Lista de transacciones asociadas al cliente.

Funcionamiento:

- Verifica si la lista de transacciones es nula o está vacía.

- Si no hay transacciones, imprime un mensaje indicando que no hay historial y retorna al menú principal.

 - Si hay transacciones, muestra un encabezado y utiliza un bucle **foreach** para imprimir cada transacción con su tipo, monto y fecha.