

# ANALISTA UNIVERSITARIO DE SISTEMAS INFORMÁTICOS

## Programación Aplicada I

### Trabajo Práctico N° 4 - Proyecto Red Majestic (Actividad Final Programación C#)

Red Majestic es una red interbancaria que brinda a sus clientes servicios de consulta de los movimientos transaccionales realizados con su tarjeta de crédito. Actualmente se encuentra funcionando con empresas internacionales como Visa, Mastercard y Diners Club, pero con una proyección a corto plazo de incorporar tarjetas nacionales como Naranja X, Mercado Pago y otras.

Usted y su equipo de trabajo, fueron contratados para desarrollar la primera versión de este sistema de red como una prueba de concepto de las necesidades requeridas por parte de Red Majestic.

Para ello, deberá diseñar una **aplicación de consola**, incorporando los conceptos aprendidos, considerando los fundamentos de programación orientados al lenguaje C#.

#### Descripción del sistema a desarrollar

Cuando el sistema se ejecuta, se le presenta al usuario el menú principal, indicando las operaciones que permite realizar. Este menú debe ser el siguiente:

**Bienvenido a Red Majestic**

**Operar con Visa - Ingrese 1**

**Operar con Mastercard - Ingrese 2**

**Operar con Diners Club - Ingrese 3**

**Ingrese su opción: \_**

Cuando el usuario ingresa la opción de la tarjeta de crédito con la que quiere operar, el sistema le solicita que **ingrese los 16 dígitos de su tarjeta de crédito**, limpiando previamente el menú principal, proporcionando así una mejor experiencia al usuario, dando la sensación de que ingreso al sistema interbancario.

El número de tarjeta de crédito ingresada por el usuario, debe tener correlación con la empresa emisora de la misma, por lo que los 4 primeros de los 16 dígitos de la tarjeta se utilizan como identificador de la empresa.

Para tarjetas de crédito Visa, el identificador es **4407**

Para tarjetas de crédito Mastercard, el identificador es **3890**

Para tarjetas de crédito Diners Club, el identificador es **7401**

Antes de presentarle al usuario el listado de sus transacciones, el sistema realiza una validación de la tarjeta de crédito utilizando como información para hacerlo, la operación seleccionada por el mismo y el identificador de la tarjeta. De este modo:

Si la operación es 1 (Visa), válida la tarjeta utilizando el identificador 4407

Si la operación es 2 (Mastercard), válida la tarjeta utilizando el identificador 3890

Si la operación es 3 (Diners Club), válida la tarjeta utilizando el identificador 7401

Luego si la validación de la tarjeta de crédito es correcta:

Si la operación es 1, obtiene el listado de transacciones de la tarjeta Visa utilizando el identificador 4407 y se las presenta en pantalla.

Si la operación es 2, obtiene el listado de transacciones de la tarjeta Mastercard utilizando el identificador 3890 y se las presenta en pantalla.

Si la operación es 3, obtiene el listado de transacciones de la tarjeta Diners Club utilizando el identificador 7401 y se las presenta en pantalla.

La visualización de las transacciones por pantalla debe ser la siguiente (por ejemplo para Visa):

#### **Movimientos de su cuenta Visa terminada en ..3210**

**Transacción N°1 - Monto: \$1500,00**

**Transacción N°2 - Monto: \$200,00**

..

Recuerde mantener una buena experiencia de usuario al mostrar esta nueva información.

Si la validación de la tarjeta no es correcta, el sistema valida si se debe a que la operación ingresada por el usuario es incorrecta. En caso afirmativo, presenta al usuario el mensaje **La opción ingresada no es válida. Inténtelo nuevamente más tarde.** En caso contrario presenta al usuario el mensaje **El número de tarjeta ingresada no es válido. Inténtelo nuevamente más tarde.**

Llegado a este punto, el sistema espera que el usuario presione cualquier tecla para continuar, y termina su ejecución.

#### **Requerimientos Funcionales y Técnicos**

**Req1:** Cuando el sistema le presenta el menú principal al usuario, la opción ingresada por el mismo debe ser almacenada en una variable del tipo entera.

**Req2:** Cuando el sistema solicita ingresar el número de tarjeta, los dígitos de la tarjeta deben ser almacenados en una variable del tipo cadena.

**Req3:** Se debe extraer del número de tarjeta de crédito el identificador de la empresa y almacenarlo en una variable del tipo cadena. Esta variable utilizará internamente el sistema para realizar las validaciones junto a la variable que almacena la opción ingresada por el usuario.

**Req4:** Se debe extraer del número de tarjeta de crédito los últimos 4 dígitos y almacenarlos en una variable del tipo cadena. Esta información se utilizará para mejorar la experiencia de usuario al mostrar el Título de los movimientos de cuenta.

**Req5:** El sistema debe validar la tarjeta de crédito llamando a una función que devuelve un valor booleano indicando si la validación se realizó correctamente y recibe como parámetros el identificador de la tarjeta y la opción ingresada por el usuario.

La lógica de esta función, debe trabajar con una estructura switch que valide la opción y para cada posible valor, invoque a la respectiva función de validación de la tarjeta pasándole como parámetro el identificador de la tarjeta. Estas nuevas funciones devuelven un valor booleano indicando si la tarjeta es válida.

Por defecto esta función invocada por el programa principal, debe devolver el valor false, y solamente true cuando las funciones internas validan correctamente la tarjeta para esa compañía.

**Req6:** La función de validación de tarjeta de crédito Visa, debe comparar el parámetro recibido, con una constante local que contiene el valor 4407. Si son iguales devuelve true, de lo contrario, devuelve false.

**Req7:** La función de validación de tarjeta de crédito Mastercard, debe comparar el parámetro recibido, con una constante local que contiene el valor 3890. Si son iguales devuelve true, de lo contrario, devuelve false.

**Req8:** La función de validación de tarjeta de crédito Diners Club, debe comparar el parámetro recibido, con una constante local que contiene el valor 7401. Si son iguales devuelve true, de lo contrario, devuelve false.

**Req9:** Si la tarjeta de crédito se validó correctamente, el sistema debe ejecutar la lógica definida en el Requerimiento **Req10**. Caso contrario, el sistema debe validar si la opción ingresada por el usuario es incorrecta y presentar el mensaje **“La opción ingresada no es válida. Inténtelo nuevamente más tarde.”** Caso contrario, debe presentar el mensaje **“El número de tarjeta ingresada no es válido. Inténtelo nuevamente más tarde.”** La lógica de este requerimiento debe realizarse utilizando estructuras condicionales dobles.

**Req10:** Este Requerimiento considera la lógica de ejecución de la rama verdadera para el Requerimiento **Req9** (La tarjeta de crédito se validó correctamente).

- Si la opción ingresada por el usuario es Visa, el sistema deberá llamar a una función que devuelve un vector del tipo float que contiene los valores de las transacciones y que recibe como parámetro el identificador de la tarjeta. Obtenidas las transacciones, el sistema debe mostrarlas por pantalla utilizando una estructura while y en base al diseño presentado en el enunciado del sistema. El número de transacción debe ser el contador de la estructura.

- Si la opción ingresada por el usuario es Mastercard, el sistema deberá llamar a una función que devuelve una lista del tipo float que contiene los valores de las transacciones, y que recibe como parámetro el identificador de la tarjeta. Obtenidas las transacciones, el sistema debe mostrarlas por pantalla utilizando una estructura do while y en base al diseño presentado en el enunciado del sistema. El número de transacción debe ser un contador manejado por la estructura.
- Si la opción ingresada por el usuario es Diners Club, el sistema deberá llamar a una función que devuelve un diccionario con clave entera y valor float que contiene en cada elemento <clave, valor> el número de transacción y el valor de la misma. Obtenidas las transacciones, el sistema debe mostrarlas por pantalla utilizando una estructura foreach y en base al diseño presentado en el enunciado del sistema. El número de transacción debe ser la clave de cada elemento del diccionario.

La lógica de este requerimiento debe ser realizada como parte de la lógica del programa principal y utilizando una estructura condicional múltiple.

**Req11:** La función que devuelve las transacciones de Visa debe inicializar un vector del tipo float de 5 elementos, y validar el parámetro recibido, invocando a la función definida en el **Req6**. Si la validación es correcta, debe cargar el vector con montos generados aleatoriamente al invocar la función que se les proporciona en el Anexo. Para cargar el vector se debe utilizar una estructura for.

**Req12:** La función que devuelve las transacciones de Mastercard debe inicializar una lista del tipo float, y validar el parámetro recibido, invocando a la función definida en el **Req7**. Si la validación es correcta, debe cargar 5 elementos en la lista con montos generados aleatoriamente al invocar la función que se les proporciona en el Anexo. Para cargar la lista se debe utilizar una estructura for.

**Req11:** La función que devuelve las transacciones de Diners Club debe inicializar un diccionario<int, float>, y validar el parámetro recibido, invocando a la función definida en el **Req8**. Si la validación es correcta, debe cargar 5 elementos <clave, valor> con montos generados aleatoriamente al invocar la función que se les proporciona en el Anexo. Para cargar diccionario se debe utilizar una estructura for, cuyo índice es la clave de cada elemento.

### Anexo

```
public static float GenerateRandomTransaction()
{
    Random rand = new Random();
    return (float)rand.NextDouble() * 1000;
}
```