Universidad Rafael Landívar
Facultad de Ingeniería.
Ingeniería en sistemas.
Pensamiento Computacional (Práctica) - Sección: 18
Catedrático: Herwing Alexis Rodríguez Franco

“DOCUMENTACIÓN DE CÓDIGO DE PROYECTO”

Zane Christian Bran Lockridge 1373724
Jessie Gabriel Abraham Puente Herrera 1331224

Guatemala, 15 de abril de 2024.

Proyecto #1

Este proyecto consiste en diseñar un diagrama de flujo y escribir código que contenga las instrucciones de guardar información de un usuario para crear una cuenta bancaria, mostrar dicha información, simular compras y ventas de productos financieros, simular abonos a una cuenta corriente, simular el crecimiento del saldo con intereses y para finalizar dar la opción de cerrar el programa al usuario.

El código empezó pidiendo información personal del usuario como su nombre completo y su número de identificación personal. Esto se hizo usando el comando Console.WriteLine(“”);. Luego de que se pidió un dato personal se usaba el comando string (variable) = Console.ReadLine(); para guardar los datos como variables tipo string. Se repitió para todos los datos. Luego, se dio la instrucción de mostrar todos los datos al usuario. Esto se hizo con una lista de Console.WriteLine(“”); como se muestra a continuación:

```
Console.Write("");
Console.WriteLine("Su nombre es: " + nombre);
Console.WriteLine("Su DPI es: " + DPI);
Console.WriteLine("Su usuario es: " + usuario);
Console.WriteLine("Su contraseña es: " + contraseña);
Console.WriteLine("Su numero de telefono es: " + telefono);
Console.WriteLine("Su correo electrónico es: " + correo);
Console.WriteLine("Su tipo de cuenta es: " + Tcuenta);
Console.WriteLine("Su tipo de moneda es: " + moneda);
Console.WriteLine("Su saldo inicial es: " + saldoI);
```

Fuente: Elaboración Propia usando el programa Visual Studio Community 2022.

El programa siguió declarando el saldo inicial con una variable int. Luego se instruyó que se mostrara dicho valor al usuario usando el comando Console.WriteLine(“”);. Luego se hizo un if/else para poder dar la opción de una decisión o excepción en el código. En este caso fue una excepción. Como se muestra a continuación se usó lo mencionado anteriormente para asegurarse que solo se hiciera una venta si en la cuenta estuvieran presente por lo mínimo Q500.00.

```
int sinicial = 2500;

Console.WriteLine("");
Console.WriteLine("Su saldo inicial es: " + sinicial);

if (sinicial > 500)
{
    //Calculos de la ganancia derivada de la transacción//

    int porcentaje = (int)(sinicial * 0.11);
    int ganancia = (int)(sinicial + porcentaje);

    Console.WriteLine("");
    Console.WriteLine("Su ganancia es de: " + porcentaje);
    Console.WriteLine("Su saldo total es de: " + ganancia);

}
else
{
    //Calculos del porcentaje del saldo actual//

    int psaldo = (int)((500 - sinicial) / (sinicial / 100) * 100);

    Console.WriteLine("");
    Console.WriteLine("No es factible realizar la transaccion en este momento");
    Console.WriteLine("No es recomendable hacer la transaccion debido al porcentaje de del saldo actual. El porcentaje del Saldo actual es: " + psaldo);
}
```

Fuente: Elaboración Propia usando el programa Visual Studio Community 2022.

Para las compras se hicieron los mismos cálculos que las ventas. Sin embargo no se usaron los comandos if / else ya que no fueron necesarios. Esto se muestra en la siguiente imagen:

```
//Compra de un producto financiero//

Console.WriteLine("");
Console.WriteLine("Compra de un producto financiero");

Console.WriteLine("");
Console.WriteLine("Su saldo inicial es de: " + sinicial);

// Calculos de las perdidas y del saldo actualizado despues de la compra//

int psaldoc = (int)(sinicial * 0.10);
int perdida = (int)(sinicial - psaldoc);


Console.WriteLine("");
Console.WriteLine("Su perdida es de: " + psaldoc);
Console.WriteLine("Su saldo actualizado es de: " + perdida);
```

Fuente: Elaboración Propia usando el programa Visual Studio Community 2022.

Este código en C# representa un sistema de abono a una cuenta corriente. Inicialmente, establece el saldo y el número de abonos realizados como cero. Luego, muestra un mensaje de bienvenida al usuario. A continuación, utiliza un bucle while para permitir al usuario realizar hasta dos abonos. Dentro del bucle, se muestra el saldo actual y se verifica si es menor que 500. Si lo es, se solicita al usuario ingresar el monto del abono, siempre que este sea válido (mayor que cero y como máximo 2500).

```
//Abonar la cuenta//

int saldo = 0;
int abonosRealizados = 0;

Console.WriteLine("");
Console.WriteLine("Bienvenido al sistema de abono a cuenta corriente.");

while (abonosRealizados < 2)
{
    Console.WriteLine("");
    Console.WriteLine("Saldo actual: " + saldo);
    if (saldo < 500)
    {
        Console.WriteLine("");
        Console.WriteLine("Puede realizar un abono a la cuenta.");
        Console.Write("Ingrese el monto a abonar (máximo 2500): ");
        int monto = Convert.ToInt32(Console.ReadLine());
        if (monto <= 2500 && monto > 0)
```

Fuente: Elaboración Propia usando el programa Visual Studio Community 2022.

Si el monto es válido, se suma al saldo y se incrementa el contador de abonos. En caso contrario, se muestra un mensaje de error y se solicita un nuevo monto. Si el saldo supera los 500, se informa al usuario que no es necesario realizar más abonos y se finaliza el proceso. En resumen, este código gestiona los abonos a la cuenta corriente de manera controlada y limitada.

```
    {
        saldo += monto;
        abonosRealizados++;
        Console.WriteLine("Abono realizado correctamente. Saldo actual: " + saldo);
    }
    else
    {
        Console.WriteLine("Monto no válido. Por favor ingrese un monto entre 1 y 2500.");
    }
}
else
{
    Console.WriteLine("El saldo actual es suficiente. No es necesario abonar.");
    break;
}
```

Fuente: Elaboración Propia usando el programa Visual Studio Community 2022.

Este segundo código en C# simula un programa de simulación bancaria para calcular el crecimiento de un saldo inicial con interés compuesto a lo largo de un período de tiempo especificado. El programa comienza mostrando un mensaje que informa al usuario que ha alcanzado el límite de abonos permitidos para el mes y solicita presionar cualquier tecla para salir. Luego, muestra un mensaje de bienvenida al simulador bancario y solicita al usuario ingresar el saldo inicial, la tasa de interés en porcentaje, el período de capitalización (mensual o bimensual), y la cantidad de meses a simular.Después de recibir los datos de entrada del usuario, se inicializan las variables necesarias para llevar a cabo la simulación. Estas variables incluyen el saldo inicial (saldoInicial), la tasa de interés (tasaInteres), el período de capitalización (periodoCapitalizacion), y la cantidad de meses a simular (mesesASimular). También se inicializa un contador de meses transcurridos (mesesTranscurridos) y una variable para almacenar el saldo en cada iteración (saldoo), inicializada con el saldo inicial.

```
//Simular paso del tiempo//

Console.WriteLine("Ya ha alcanzado el límite de abonos permitidos este mes.");
Console.WriteLine("Presione cualquier tecla para salir.");
Console.ReadKey();

Console.WriteLine("Bienvenido al Simulador Bancario");
Console.WriteLine("Ingrese el saldo inicial: ");
double saldoInicial = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("Ingrese la tasa de interés en porcentaje (por ejemplo, 2 para 2%): ");
double tasaInteres = Convert.ToDouble(Console.ReadLine()) / 100;
Console.WriteLine("Seleccione el período de capitalización: ");
Console.WriteLine("1. Mensual");
Console.WriteLine("2. Bimensual");
int periodoCapitalizacion = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Ingrese la cantidad de meses a simular: ");
int mesesASimular = Convert.ToInt32(Console.ReadLine());

int mesesTranscurridos = 0;

double saldoo = saldoInicial;
```

Fuente: Elaboración Propia usando el programa Visual Studio Community 2022.

A continuación, se inicia un bucle while que se ejecuta mientras el número de meses transcurridos sea menor que la cantidad de meses a simular. Dentro del bucle, se verifica si ha transcurrido al menos un mes y si el número de meses transcurridos es divisible entre el período de capitalización seleccionado. Si se cumple esta condición, se imprime el saldo al final del mes actual. Luego, se incrementa en uno el número de meses transcurridos. Se calcula el interés generado en el saldo actual utilizando la fórmula de interés compuesto, y se suma este interés al saldo actual. El bucle continúa hasta que se alcanza la cantidad de meses a simular. Una vez completada la simulación, el programa finaliza. En resumen, este programa permite al usuario simular el crecimiento de un saldo inicial con interés compuesto durante un período de tiempo especificado, mostrando el saldo al final de cada mes (o cada cierto período, según la opción de capitalización seleccionada).

```
while (mesesTranscurridos < mesesASimular)
{
    if (mesesTranscurridos > 0 && mesesTranscurridos % periodoCapitalizacion == 0)
    {
        Console.WriteLine($"Saldo al final del mes {mesesTranscurridos}: {saldo:C}");
    }

    mesesTranscurridos++;

    double interes = saldo * tasaInteres;

    saldoo += interes;

}
```

Fuente: Elaboración Propia usando el programa Visual Studio Community 2022.