

UNIVERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TITULO:

Informe de Laboratorio N° 02: Elaborando pruebas unitarias dentro de una aplicación

CURSO:

Calidad y Pruebas de Software

DOCENTE:

Ing. Patrick Cuadros Quiroga

INTEGRANTES:

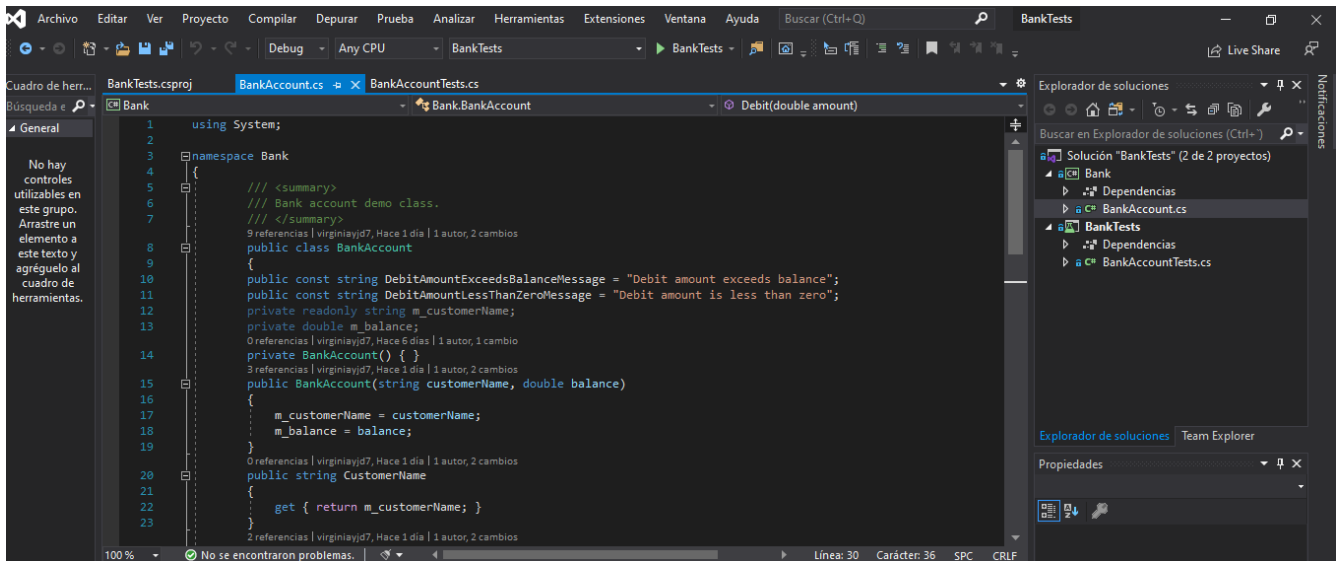
Yaneth Virginia Aquino Huallpa

(2017059286)

Tacna

1. Crear un proyecto para pruebas

- Crear un proyecto nuevo de C char Aplicación de consola (.NET Core) y Asigne al proyecto el nombre Bank .
- Crear un proyecto de prueba unitaria de prueba de MSTest (.NET Core) de C char y Ponga al proyecto el nombre BankTests.



- Reemplace el contenido de Program.cs por el siguiente código de C char que define una clase, BankAccount

```

using System;

namespace Bank
{
    /// <summary>
    /// Bank account demo class.
    /// </summary>
    6 referencias | virginlayd7, 11:06:11 | 1 autor, 2 cambios
    public class BankAccount
    {
        public const string DebitAmountExceedsBalanceMessage = "Debit amount exceeds balance";
        public const string DebitAmountLessThanZeroMessage = "Debit amount is less than zero";
        private readonly string _customerName;
        private double _balance;
        0 referencias | virginlayd7, 11:06:11 | 1 autor, 1 cambio
        private BankAccount() { }
        3 referencias | virginlayd7, 11:06:11 | 1 autor, 2 cambios
        public BankAccount(string customerName, double balance)
        {
            _customerName = customerName;
            _balance = balance;
        }
        0 referencias | virginlayd7, 11:06:11 | 1 autor, 2 cambios
        public string CustomerName
        {
            get { return _customerName; }
        }
        2 referencias | virginlayd7, 11:06:11 | 1 autor, 2 cambios
        public double Balance
        {
            get { return _balance; }
        }
        3 referencias | virginlayd7, 11:06:11 | 1 autor, 2 cambios
        public void Debit(double amount)
        {
            if (amount > _balance)
            {
                throw new System.ArgumentOutOfRangeException("amount", amount, DebitAmountExceedsBalanceMessage);
            }
            if (amount < 0)
            {
                throw new System.ArgumentOutOfRangeException("amount", amount, DebitAmountLessThanZeroMessage);
            }
        }
        1 referencia | virginlayd7, 11:06:11 | 1 autor, 2 cambios
        public void Credit(double amount)
        {
            if (amount < 0)
            {
                throw new ArgumentOutOfRangeException("amount");
            }
            _balance += amount;
        }
        0 referencias | virginlayd7, 11:06:11 | 1 autor, 2 cambios
        public static void Main()
        {
            BankAccount ba = new BankAccount("Mr. Bryan Walton", 11.99);
            ba.Credit(5.77);
            ba.Debit(11.22);
            Console.WriteLine("Current balance is ${0}", ba.Balance);
        }
    }
}

```

- El archivo BankAccountTests.cs contiene ahora el siguiente código:

0 referencias | virginiajd7, Hace 1 día | 1 autor, 2 cambios

```
public class BankAccountTests
{
    [TestMethod]
    0 referencias | virginiajd7, Hace 1 día | 1 autor, 1 cambio
    public void Debit_WithValidAmount_UpdatesBalance()
    {
        // Arrange
        double beginningBalance = 11.99;
        double debitAmount = 4.55;
        double expected = 7.44;
        BankAccount account = new BankAccount("Mr. Bryan Walton", beginningBalance);
        // Act
        account.Debit(debitAmount);
        // Assert
        double actual = account.Balance;
        //Assert.AreEqual(expected, actual, 0.001, "Account not debited correctly");
    }

    [TestMethod]
    0 referencias | virginiajd7, Hace 1 día | 1 autor, 1 cambio
    public void Debit_WhenAmountIsLessThanZero_ShouldThrowArgumentOutOfRangeException()
    {
        // Arrange
        double beginningBalance = 11.99;
        double debitAmount = 20.0;
        BankAccount account = new BankAccount("Mr. Bryan Walton", beginningBalance);
        // Act
        try
        {
            account.Debit(debitAmount);
        }
        catch (System.ArgumentOutOfRangeException e)
        {
            // Assert
            StringAssert.Contains(e.Message, BankAccount.DebitAmountExceedsBalanceMessage);
            return;
        }
        Assert.Fail("The expected exception was not thrown.");
    }
}
```

2. Resultado: Al ejecutar la prueba, se muestra:

