

## **INTRO**

### **Presentación**

Buenos días señores de la escuela de Ingeniería Informática de la universidad Ricardo Palma, buenos días (los que estén ahí si son importantes), buenos días estimado profesor Valverde, buenos días compañeros.

En esta oportunidad les vamos a hablar acerca de la elaboración de nuestro proyecto final realizado para el Taller Básico de Programación en el ciclo 2017-I. El trabajo fue realizado por nosotros 3: Carlos Lau, Pedro Huamaní y Angelo Licetti. El lenguaje de programación que utilizamos fue c#. El programa fue diseñado en Windows Forms y fue desarrollado utilizando el IDE Visual Studio 2015 en los laboratorios de la Universidad Ricardo Palma.

### **Resumen**

En pocas palabras lo que hicimos fue diseñar un sistema computarizado que permita realizar la apertura de una cuenta de ahorros, realizar transacciones, es decir, depósitos y retiros en esa cuenta y obtener un voucher de las transacciones realizadas.

### **Introducción**

#### **Contextualización**

La apertura de una cuenta bancaria suele ser un trámite presencial, pues se requiere que la persona que desea abrir la cuenta se dirija al banco para poder hacerlo. Sin embargo, gracias a la ayuda de los avances tecnológicos, ahora es mucho más sencillo pues se puede hacer de manera automatizada y sistematizada desde internet o desde un agente automático.

#### **Objetivo**

Diseñar un sistema computarizado que permita realizar la apertura de una cuenta de ahorros, realizar transacciones (depósitos y retiros) en esa cuenta y obtener un voucher de las transacciones realizadas.

### **Descripción general de la solución**

#### **APERTURA DE LA CUENTA:**

Al momento de iniciar el programa el usuario tiene la opción de realizar la apertura de su cuenta en soles o en dólares, para lo cual debe proporcionar siguientes datos de entrada:

- Nombres
- Apellidos

- Monto Inicial

### TRANSACCIONES:

Tras crear su cuenta el usuario puede realizar depósitos o retiros, revisar su voucher con la lista de todas las transacciones realizadas en su cuenta, o cerrar sesión.

### VOUCHER:

Tras realizar cada transacción se guardan en la memoria los siguientes datos de cada una de las transacciones realizadas que son luego mostrados en el voucher:

- N° de transacción
- Monto ahorrado hasta antes de la transacción
- Tipo de transacción (depósito/retiro)
- Cantidad de la transacción
- Monto ahorrado después de la transacción realizada
- Fecha y hora de la transacción

## **DESARROLLO**

### **La interfaz**

Como ya se mencionó antes, trabajamos usando Windows Forms. En todos los forms utilizamos el mismo fondo y los mismos colores de fondo para los diferentes elementos de los forms.

### **Form “Inicio”**

En esta form incluimos un mensaje de bienvenida y otro preguntándole al usuario en que moneda desea aperturar su cuenta, teniendo las opciones de soles o dólares que están colocadas como botones. Cada uno de estos dos botones habré distintos forms que tienen básicamente el mismo diseño, con la única diferencia de la moneda con la que se trabaja.

Adicionalmente, se encuentra el botón “Salir” el cual tras ser presionado cierra el programa completamente.

## **Forms “AperturaSoles” y “AperturaDolares”**

Estas dos forms son iguales con la única diferencia de la moneda que se usa para cada una de ellas. Aquí se le solicitan datos al usuario para poder continuar. Los datos que se deben ingresar son:

- Nombres
- Apellidos
- Monto inicial

Los nombres y apellidos son ingresados por teclado. Por otro lado, el monto inicial es ingresado mediante un conjunto de botones que permiten ingresar únicamente cantidades numéricas con el fin de evitar cualquier error en la ejecución en caso de que el usuario ingrese accidentalmente valores no numéricos.

(INSERTAR AQUÍ EXPLICACIÓN DE BOTONES NUMÉRICOS)

Además, se tienen los botones “Aceptar” y “Cancelar”. El primero de ellos ingresa los datos al programa y abre otro form llamado “CuentaAperturada y cierra el form actual, mientras que el segundo regresa a la form “Inicio” y de igual forma cierra el form actual.

## **Form “CuentaAperturada”**

En esta form aparecen los datos ingresados en el form anterior, es decir el nombre completo del usuario, y el monto ahorrado hasta el momento. El nombre del usuario aparece de la siguiente manera: primero sus apellidos, seguidos de una coma y sus nombres.

Además, aquí se encuentran 4 botones que permiten realizar diferentes acciones: “Depositar”, “Retirar”, “Voucher” y “Cerrar Sesión”. Los botones de depositar y retirar abren los forms “Retiro” y “Deposito” respectivamente, el botón “Voucher” muestra el form “Voucher” y el botón cerrar sesión vuelve a mostrar la form “Inicio”. Además, tras presionar cualquiera de los botones, además de abrir los forms correspondientes cierran el form actual.

## **Forms “Deposito” y “Retiro”**

Estos dos forms tienen un diseño igual con la única diferencia de que uno sirve para depositar dinero a la cuenta, mientras que el otro para retirar dinero de la misma.

Ambos cuentas con el mismo conjunto de botones para ingresar una cantidad únicamente numérica. Además, poseen los botones “Aceptar” y “Cancelar”. Al presionar el primero de estos cierra el form actual y se abre otro form “CuentaAperturada” pero esta vez con el monto ya modificado tras la transacción. Asimismo, al presionar el botón cancelar sucede que cierra el form actual y muestra de nuevo el form “CuentaAperturada” sin haber modificado el monto ahorrado.

Una de las principales diferencias, es que al momento de realizar un retiro, utilizamos un condicional. Si el monto que se quiere retirar es mayor que el monto ahorrado, entonces el usuario no podrá realizar la acción y le aparecerá un messageBox comunicándole que no puede realizar dicha transacción, pues no cuenta con el dinero suficiente.

### **Form “Voucher”**

Este form muestra el nombre completo del usuario, la fecha actual, el monto ahorrado, y lo más importante de este form: todas las transacciones, es decir, todos los retiros y depósitos, que se han realizado, mostrando los siguientes datos de salida de cada transacción:

- N° de transacción
- Monto ahorrado hasta antes de la transacción
- Tipo de transacción (depósito/retiro)
- Cantidad de la transacción
- Monto ahorrado después de la transacción realizada
- Fecha y hora de la transacción

El único botón en este form es el botón “Volver”, el cual al ser presionado cierra el form actual y abre de nuevo un form “CuentaAperturada” sin ningún tipo de modificación en cuanto a los datos.

## **El código**

Para el desarrollo de nuestro sistema, decidimos utilizar dos clases para la realización y funcionamiento del programa, a las cuales llamamos Cuenta, y Transacción.

### **Clase Cuenta**

Esta clase es usada durante la ejecución de todo el programa para almacenar los datos del usuario, así como los datos de cada una de las transacciones que realiza. Esta clase cuenta con diversos atributos y métodos cuyo uso y funcionamiento se explicará en breve.

Algo que es importante resaltar antes de explicar el funcionamiento de esta clase, es que en esta misma, inicializamos un arreglo unidimensional de tipo Transacción, llamado transaccionesRealizadas, con el fin de almacenar en este arreglo los datos de todas las transacciones realizadas en dicha cuenta para así utilizarlos luego al momento de visualizar el voucher.

Entre las variables pertenecientes a la clase cuenta están en primer lugar aquellas que almacenan los datos que el usuario proporciona al momento de crear su cuenta: sus nombres, sus apellidos, la moneda de su cuenta de ahorros (que puede ser soles o dólares), y el monto actual de la cuenta, que al momento de crearla se le asigna el valor ingresado como monto inicial por el usuario, y luego es modificado cada vez que se realiza una transacción. Cabe recalcar que los

nombres, los apellidos, y la moneda, son de tipo string, mientras que el monto ahorrado es de tipo double.

Además, esta clase también cuenta con variables con diversas variables que se usan para almacenar los datos de salida que se muestran al momento de visualizar el voucher.

Ahora hablemos de los métodos. Entre los métodos pertenecientes a esta clase se encuentran los siguientes: el método constructor Clase, el método NuevoRetiro de tipo void, el método NuevoDepósito de tipo void, y algunos otros métodos de tipo string que son utilizados para mostrar los datos del voucher.

En el método constructor establecimos 4 parámetros que son utilizados para asignar los valores de las variables que almacenan los datos del usuario ya antes mencionados al momento de inicializar un objeto de esta clase.

En el método NuevoRetiro, determinamos un parámetro de tipo string que es usado para pasar el valor de la cantidad retirada. En este método, inicializamos un nuevo elemento del arreglo transaccionesRealizadas y le pasamos a este nuevo objeto de la clase transacción el valor de la cantidad con la que se trabajará ya dentro de esta clase.

En el método NuevoDepósito, determinamos un parámetro de tipo string, que es utilizado para pasar el valor de la cantidad que se va a depositar. Al igual que en el método anterior, inicializamos aquí un nuevo elemento del arreglo transaccionesRealizadas y le pasamos el valor de la cantidad depositada.

Además de los métodos ya mencionados, contamos en esta clase con diversos métodos de tipo string que nos servirán para el momento de obtener la boleta de las transacciones realizadas.

### **Clase Transacción**

Al momento de realizar alguna transacción, ya sea un depósito o un retiro, en nuestro sistema, se inicializa un nuevo elemento del arreglo transaccionesRealizadas de la clase Transacción. En esta clase, contamos con múltiples atributos que sirven para almacenar los datos de cada transacción que se realiza. Entre ellos tenemos: las variables de tipo double montoUtilizado, montoAnterior y nuevoMonto, las variables de tipo string fechaYHora y tipoDeTransacción, y una variable estática llamada n que nos sirve de contador para así saber cuántas transacciones se han realizado hasta el momento y poder controlar así el número de datos que se visualizan en el voucher.

Entre los métodos de esta clase tenemos 3: el método constructor Transaccion, el método Retirar, y el método Depositar. El método constructor consta de tres parámetros los cuales son declarados al momento de inicializar cada transacción desde la clase Cuenta y luego son asignados a las variables declaradas en la clase transacción para así trabajar con ellas. El método Retirar de tipo double, que retorna el monto después de haberle aplicado el retiro, y el método Depositar

también de tipo `double`, que retorna el valor del monto ahorrado después de haberle aplicado el depósito.

### **Métodos constructores de cada form**

Ya que estamos en el primer ciclo y aún no tenemos conocimientos suficientes para trabajar con bases de datos, debíamos encontrar una forma simple y práctica de pasar los datos entre los formularios. Lo que hicimos fue declarar una variable de la clase `Cuenta` en cada uno de los forms, para después asignarle el valor de la cuenta creada por el usuario mediante el constructor de cada clase.

Al momento de aperturar una cuenta, se inicializa un objeto de la clase `Cuenta` y es pasado a la form `CuentaAperturada` como argumento de su constructor. Luego, ya en dicha form, se declara una variable de clase `Cuenta` y a esta se le asigna el valor del objeto `Cuenta` pasada mediante el constructor, ya inicializado en el form anterior usando los datos proporcionados por el usuario.

Se sigue el mismo procedimiento al pasar de un form a otro para así pasar todos los datos de la cuenta a cualquiera de los forms, y así poder trabajar con los datos en todos los forms.

### **Conclusiones**

- Comprendimos de manera más eficaz los conceptos aprendidos en clase al poder implementarlos en el desarrollo que da un sistema que responde a las necesidades de una situación real.
- Este curso nos ayudó a completar satisfactoriamente nuestro proyecto implementando varios tipos de métodos y teorías para hacerlo más completo y más sencillo aun.
- Fue una gran experiencia participar en el desarrollo de este proyecto ya que ganamos muchísima experiencia y estamos mejor capacitados que antes de comenzar el curso.