Portafolio - Trabajo #11

**Tema**:

* Práctica guiada: Proyecto de Aplicación de Escritorio en Visual Studio, .Net Core

**Programa**: Programador(a) de aplicaciones informáticas

**Módulo**: Programación Orientada a Objetos

**Código**: CSTI0208

**Persona participante:**

|  |
| --- |
| Jaen Carlo Gonzalez Arauz |

|  |  |
| --- | --- |
| Fecha de asignación del trabajo: |  |
| Fecha límite de entrega: |  |
| Medio de entrega: | Microsoft Teams |

**Instrucciones:**

1. Trabajando de forma individual programará en Visual Studio un proyecto de escritorio que resuelva la situación planteada.
2. Guarde su trabajo en GitHub y envíe el link mediante la bitácora, la cual debe subir a Microsoft Teams.
3. Abra el programa Microsoft Visual Studio
4. Cree un nuevo proyecto
5. En la ventana emergente, sobre el menú de plantillas seleccione el ítem “Visual C#” y en el panel de la derecha seleccione “ ”. En el nombre (parte inferior de la ventana) indique **ProyectoEscritorio**
6. El proyecto crea de forma automática un formulario. Cambie el nombre de archivo del mismo por **Frm\_Notas.**
7. Cambie la propiedad NOMBRE (Name) del formulario por el mismo indicado en el punto anterior.
8. Verifique las propiedades del proyecto para que inicie con Frm\_Notas
9. Utilizando los controles de la herramienta, debe diseñar un formulario como el que se muestra en la siguiente imagen. (se muestran varias pantallas según el estado de la aplicación)

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. A continuación se describen los tipos de controles a usar y sus propiedades. Se omiten algunos controles como etiquetas que no tienen más valor que el de mostrar información. Usted debe dar los nombres a cada control.
   1. Un cuadro de texto habilitado para digitar la nota
   2. Un botón para agregar la nota indicada en la lista de la derecha.
   3. Una lista donde se mostrarán las notas ingresadas
   4. Un cuadro de texto deshabilitado donde se mostrará la nota seleccionada de la lista.
   5. Un botón para eliminar la nota seleccionada de la lista
   6. Un botón para borrar el contenido de la lista
   7. Un botón para mostrar el resultado del promedio de las notas de la lista.
   8. Una etiqueta que cambiará el color dependiendo del promedio de las notas y que indicará tanto el valor del promedio de notas como el estado de aprobado o reprobado.
   9. Un botón para salir del formulario.
2. Agregar a la solución un nuevo proyecto de tipo Biblioteca de Clases llamado LogicaNegocio
3. La nueva aplicación creará de forma automática una nueva clase, cambie el nombre de esta clase por ClsPromedio
4. Dentro de la clase ClsPromedio crear los siguientes miembros:
   1. Atributos: condicion, color, promedio
   2. Propiedades: Condicion, Color, Promedio
   3. Métodos:
      1. Contructor vacío
      2. Calculo de promedio, que recibe la suma de notas y la cantidad de notas
      3. Calculo de condición que recibe un promedio y determina si aprobó o reprobó y cambia el valor de la propiedad color de acuerdo a la condición (rojo reprobó, verde aprobó)
5. Agregue el proyecto de lógica como referencia al proyecto **ProyectoEscritorio**
6. Comportamiento del formulario de notas:

Procedimientos y Funciones

* 1. Debe crear un procedimiento en el código que realice lo siguiente:
     1. La etiqueta del resultado del promedio se le asigne un color amarillo
     2. La etiqueta del resultado del promedio se debe limpiar.
  2. Debe crear una función en el código que realice lo siguiente:
     1. Si la lista de notas tiene valores, se debe recorrer la lista e ir sumando los valores de las notas de la misma.
     2. El valor de la suma se va almacenando en una variable.
     3. La función devuelve el monto total de la suma de las notas.

Propiedades y Eventos

* 1. Debe configurarse el cuadro de texto de la nota para que el máximo de caracteres que admita sean tres.
  2. El usuario indica la nota en el cuadro de texto y presiona el botón de “Agregar”. La nota digitada debe agregarse a la lista de la derecha. Al presionar el botón de Agregar, el sistema debe validar que se haya indicado un valor, que valor escrito sea numérico y que dicho valor no sea un número superior a 100. Si incumple alguna de estas tres validaciones, debe mostrar un mensaje al usuario y no debe agregar la nota a la lista. Una vez que se haya agregado la nota a la lista se debe hacer un llamado al procedimiento (creado en el punto a) que limpia el resultado del promedio. Finalmente el cuadro de texto de la nota se debe limpiar para que el usuario pueda ingresar una nueva nota.
  3. Al dar clic sobre alguna de las notas de la lista, esta nota se debe mostrar en el cuadro de texto de la nota seleccionada.
  4. Al presionar el botón de Eliminar Nota Seleccionada, el sistema verifica si el usuario seleccionó alguna nota de la lista. Se debe quitar la nota seleccionada de la lista, se limpia el campo de texto de la nota seleccionada y por último se hace un llamado al procedimiento que limpia los datos del promedio.
  5. Al presionar el botón de Limpiar la Lista, el sistema debe limpiar todas las notas que se muestran en la lista, además se deben limpiar los cuadros de texto de la nota seleccionada y el de la nota digitada por el usuario. Para finalizar se hace un llamado al procedimiento que limpia los datos del promedio.
  6. Al presionar el botón “Ver Resultado”, el sistema primero debe verificar si la lista de notas contiene algún registro, en caso contrario debe mostrar un mensaje al usuario indicando que no hay notas para calcular el promedio. Si existen notas registradas, el procedimiento a realizar es el siguiente:
     1. Se debe obtener la suma de las notas por medio de un llamado a la función que realiza dicho cálculo (creada en el punto b), se guarda el valor en una variable local.
     2. Se debe obtener el promedio de las notas haciendo un llamado a la función de calculo de promedio de la clase ClsPromedio, para lo cual deberá crear una instancia de la clase. El resultado de la operación se almacena en una variable (de tipo DOUBLE).
     3. Se debe obtener la condición haciendo un llamado a la función de calculo de condición de la clase ClsPromedio, para lo cual deberá utilizar la instancia de la clase creada en el punto anterior. El resultado de la operación se almacena en una variable (de tipo String).
     4. Utilizando la propiedad de color de la clase ClsPromedio se cambia el color de la etiqueta de resultado
     5. El promedio obtenido debe mostrarse en la etiqueta de resultado concatenado con la Condición.
  7. Para finalizar el botón Salir, debe cerrar la aplicación.

1. Verifique que la aplicación funcione correctamente.
2. Guarde y cierre la aplicación.

**Escala numérica de evaluación**

**Persona participante**: Jaen Carlo Gonzalez Arauz

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lista de rasgos observables** | | **Criterios** | | |
| **CC** | **CI** | **II** |
| **2** | **1** | **0** |
| **En cuanto a la presentación** | |  |  |  |
| 1. Define las clases solicitadas | |  |  |  |
| 1. Respeta la funcionalidad definida en para cada clase | |  |  |  |
| 1. Crea los formularios con los controles y propiedades correctas | |  |  |  |
| 1. Utilizas estructuras de control según lo requerido | |  |  |  |
| 1. Realiza manejo de excepciones dentro de la solución | |  |  |  |
| 1. Crea las funciones y métodos según lo requerido | |  |  |  |
| 1. Las clases contiene los atributos necesarios | |  |  |  |
| 1. Los atributos definidos para las clases son del tipo adecuado | |  |  |  |
| 1. Crea las propiedades adecuadas a cada clase | |  |  |  |
| 1. Crea los métodos requeridos según lo requerido | |  |  |  |
| 1. La visibilidad de los objetos es apropiada para la solución de proyecto | |  |  |  |
| 1. Utiliza constructores dentro del proyecto | |  |  |  |
| 1. Realiza las instancias de los objetos correctamente | |  |  |  |
| 1. Utiliza buenas prácticas de programación | |  |  |  |
| 1. Utiliza prefijos y sufijos par la declaración de los objetos y variables | |  |  |  |
| 1. La aplicación es amigable | |  |  |  |
| 1. Es fácil de usar | |  |  |  |
| 1. Resuelve el problema solicitado | |  |  |  |
| 1. La solución presentada es funcional y acorde a lo solicitado | |  |  |  |
| 1. Realiza la documentación interna de los métodos y clases | |  |  |  |
| Total:  Simbología:  **CC**: correcto y completo  **CI**: correcto pero incompleto  **II**: incorrecto e incompleto | |  |  |  |
| **Puntos Totales:** | **40** | | |
| **Puntos obtenidos:** |  | | |
| **Calificación:** |  | | |

**Observaciones:**