

examen Ev 1 parcial 1

Cuando terminéis, avisar al profesor para que valide el proyecto en vuestro equipo.

Exportad vuestro proyecto a un fichero ZIP, por la opción de menú (File -> Export Project -> To ZIP).

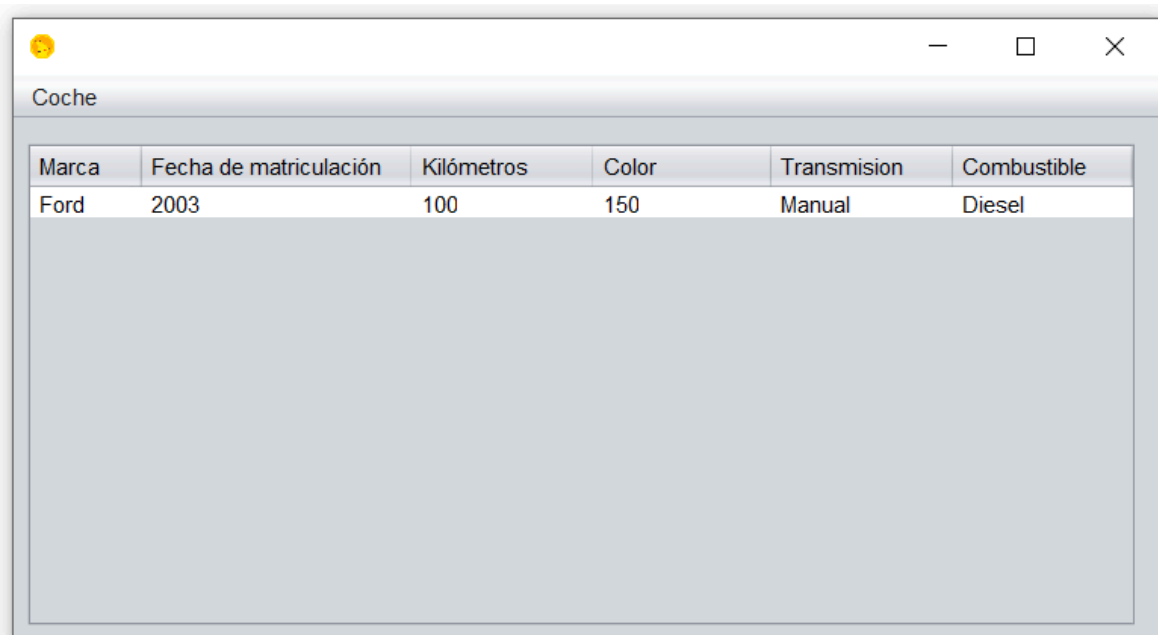
Editar el nombre del fichero generado para incluir vuestro nombre y apellidos.

Subir el fichero al aula virtual.

Revisar que el fichero subido es el correcto.

Examen UD 1 - 2 Curso: 2DAM

Diseñar una aplicación para gestionar coches. La aplicación tendrá una ventana principal como la siguiente imagen:



Con un menú "Coche" y un submenú "Nuevo..." desde el que se abrirá la ventana de alta siguiente:

Marca: Seleccione una marca...

Fecha de matriculación: 2,000

Kilómetros (miles): 0 50 100 150 200 250 300

Color: 0 50 100 150 200 250

Tipo de transmisión

☐ Manual

☐ Automática

Combustible

☐ Gasolina

☐ Diesel

☐ Híbrido

☐ Eléctrico

Guardar

La fecha de matriculación es un numérico para el año, pero se puede configurar como fecha con formato, si se quiere.

El control de color afectará al color de la ventana de alta, según se vaya ajustado, y el color de las etiquetas de la ventana se ajustarán igualmente a un color calculado de la siguiente forma:

- Si se selecciona un color “C”, el color de las etiquetas será “255 - C”
- Se cambiarán simultáneamente los tres colores R,G,B al seleccionado en el control C, o sólo uno de ellos dejando los otros fijos si se quiere.

Marca: Seleccione una marca...

Fecha de matriculación: 2,000

Kilómetros (miles): 0 50 100 150 200 250 300

Color: 0 50 100 150 200 250

Tipo de transmisión

☐ Manual

☐ Automática

Combustible

☐ Gasolina

☐ Diesel

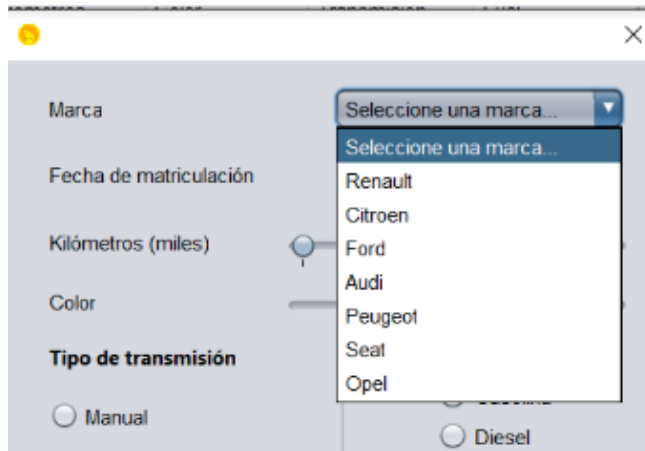
☐ Híbrido

☐ Eléctrico

Guardar

La aplicación tendrá un icono como el de la imagen (pokecoin.png), adjunta en la tarea.

Los valores posibles en el campo “Marca” pueden ser los siguientes:



Se valorarán los siguientes puntos:

- Diseño de las dos pantallas. (2 puntos)
- Funciona la carga de registros. (1,5 puntos)
- Funciona la actualización de los colores de la ventana de alta y de las etiquetas al ajustar el control de color. (1 punto)
- Validación de alguno de los controles de la pantalla de alta utilizando mensajes JOptionPane. (1 punto)
- Utilización de las clases de internacionalización para algunos de los textos. (2,5 puntos)
- Poner la imagen adjunta como icono de la aplicación. (1 punto)
- Comentarios en el código. (1 punto)

Recuperación 1 eval - UD 1 -2

Se quiere diseñar una aplicación para gestionar reservas de viajes:

En la ventana principal se podrán ver todas las reservas y se podrán crear nuevas desde una opción de menú: Reservas → Nueva

Reservas										
Origen	Destino	Ida	Ida y vuelta	Vuelta	Viajeros	Familia numerosa	Jubilado	Maleta adicional	Selección asiento	Precio Total
Madrid	Barcelona	Thu Nov 24 ...	true	Sat Nov 26 2...	2	false	false	false	true	440.0

La ventana de alta tendrá los siguientes campos:

Origen

Madrid

Destino

Barcelona

Fecha ida:

24/11/22 23:13

Fecha vuelta:

26/11/22 23:13

Nº viajeros:

2

Ida y vuelta

☒

Descuentos

☐ Familia numerosa

☐ Jubilado

Extras


☐ Maleta adicional

☒ Selección de asiento

Precio Total

440

Reservar



Se deberá seleccionar una ciudad de origen y otra de destino y se chequeará que no sean la misma ciudad mostrando un mensaje de error en dicho caso.

El campo de fecha de vuelta sólo será visible si se marca el checkbox de “Ida y vuelta”

El precio por persona de cada viaje se estima en 100 euros (en caso de marcar ida y vuelta sería el doble). Se calculará el precio total de la reserva en función de la información indicada por el usuario y se podrán aplicar los siguientes descuentos sobre el precio base:

- Familia numerosa -> descuento del 50 % (sólo aplicable al precio base)
- Jubilado -> 20 euros por viaje (con ida y vuelta serían 40) y se aplicará sólo a una persona.

Y los siguientes recargos por extras (no se ven afectados por los descuentos):

- Maleta adicional → 30 euros por viaje y persona
- Selección de asiento → 10 euros por viaje y persona

Reparto de puntos:

Diseño pantallas → 2 puntos

Se carga el registro en la tabla → 1 punto

Validaciones con JOptionPane sobre los campos origen y destino para que tengan un valor válido (una ciudad distinta en cada combo) → 1 punto

Validaciones con JOptionPane sobre los campos de fecha para que la vuelta no pueda ser igual o anterior a la ida → 1 punto

Imagen del logo → 1 punto

Comentarios en código → 1 punto

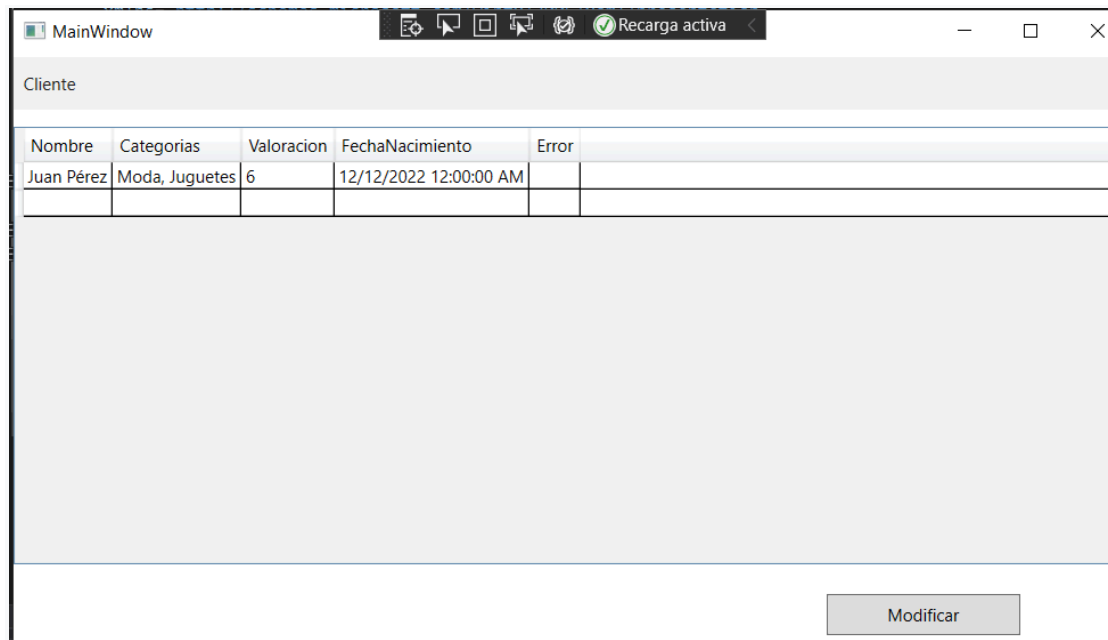
Cálculo del total de forma dinámica en función de los datos que va informando el usuario (3 puntos):

- Número de personas y radiobutton de ida y vuelta → 1 punto
- Descuentos -> 1 punto
- Extras → 1 punto

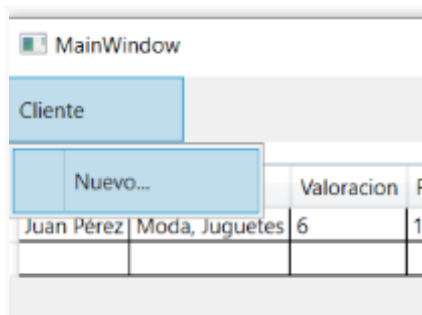
Examen UD3

Crear una aplicación para registrar datos de los clientes de una empresa.

La ventana principal debe mostrar una lista de clientes:



Debe tener un menú de alta de clientes:



que nos abra la siguiente ventana con los campos:

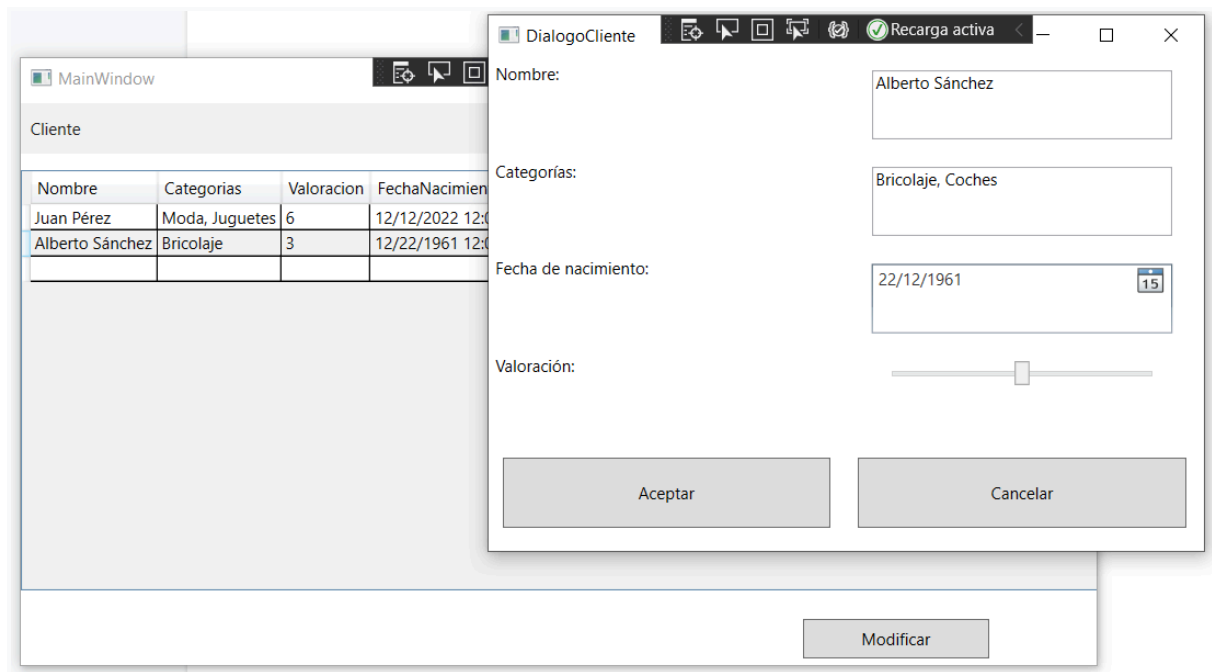
The screenshot shows a Windows-style dialog box titled "DialogoCliente". The title bar includes standard window controls and a status indicator "Recarga activa" with a green checkmark. The form contains four labeled input fields: "Nombre:" (a red-outlined empty text box), "Categorías:" (an empty text box), "Fecha de nacimiento:" (a date picker showing "12/12/2022" and a calendar icon with "15"), and "Valoración:" (a slider control with a small square handle at the left end). At the bottom are two buttons: "Aceptar" and "Cancelar".

Para poder dar de alta un cliente se debe informar al menos el nombre del mismo.

La valoración se indicará a través de un control de slider (0 a 10).

This screenshot shows the same "DialogoCliente" dialog box, but with the input fields populated. The "Nombre:" field contains the text "Alberto Sánchez". The "Categorías:" field contains the text "Bricolaje". The "Fecha de nacimiento:" field now shows the date "22/12/1961". The "Valoración:" slider control has its handle moved to the center of the track. The "Aceptar" and "Cancelar" buttons remain at the bottom.

Una vez creado el registro del cliente, mediante el botón de “Modificar” de la pantalla principal, se podrá modificar el registro marcado en la lista.



Valoración del examen:

Creación de ventanas (con distribución de controles en grid por coordenadas en XAML en ventana de alta) -> 2,5

Alta correcta de un registro en la lista -> 2,5

Modificación correcta de un registro (con el uso de clonación) -> 2

Validación campo obligatorio (con activación de botón Aceptar) -> 2

Comentarios en código -> 1

Tras terminar el examen generar un zip con la carpeta del proyecto y subirlo a la tarea del aula virtual.

Examen UD4

Ejercicio 1. (Microsoft Visual Studio)

Se desea crear una aplicación que permita hacer conversiones de divisas a euro:

1 dollar -> 0.92 euros

1 dirham -> 0.091 euros

1 yuan -> 0.14 euros

1 libra -> 1.13 euros

y las inversas:

De euro a dollar, dirham, yuan, libras.. (el factor será el inverso).

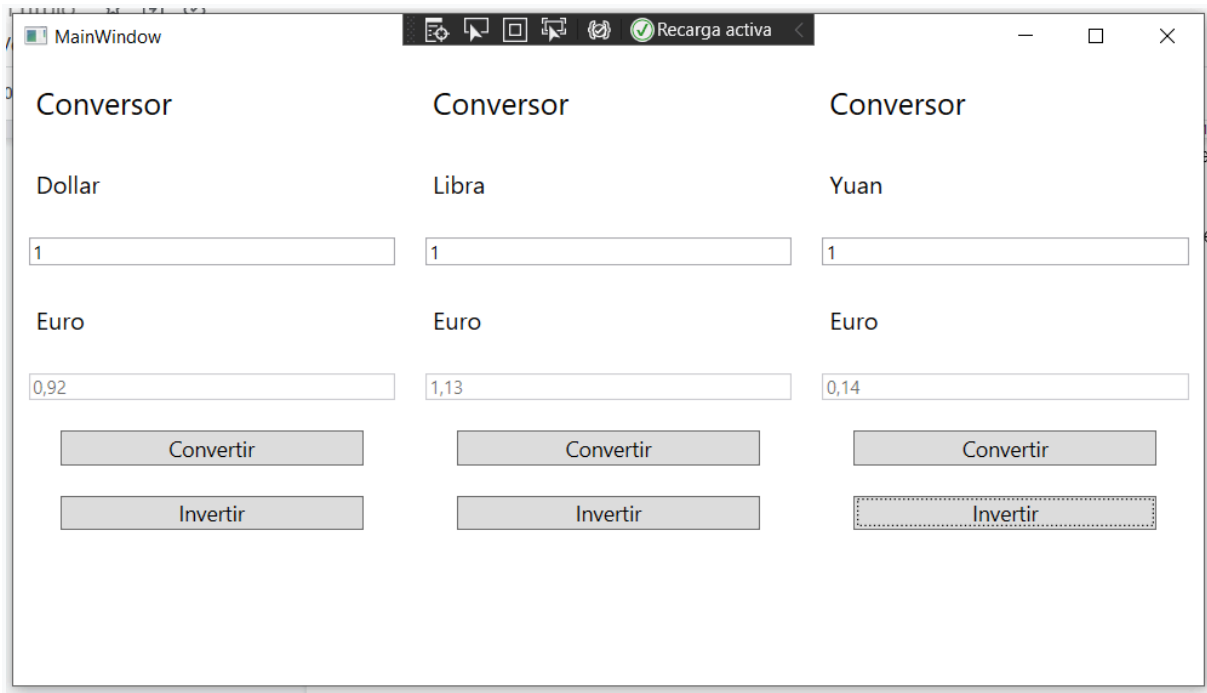
Para ello se debe crear un componente configurable con un atributo donde indiquemos la moneda a convertir: DOLLAR, DIRHAM, YUAN, LIBRA

y en función de dicho valor, se actualizarán las etiquetas correspondientes en pantalla y el factor de conversión.

La conversión se realizará al pulsar el botón “Convertir”.

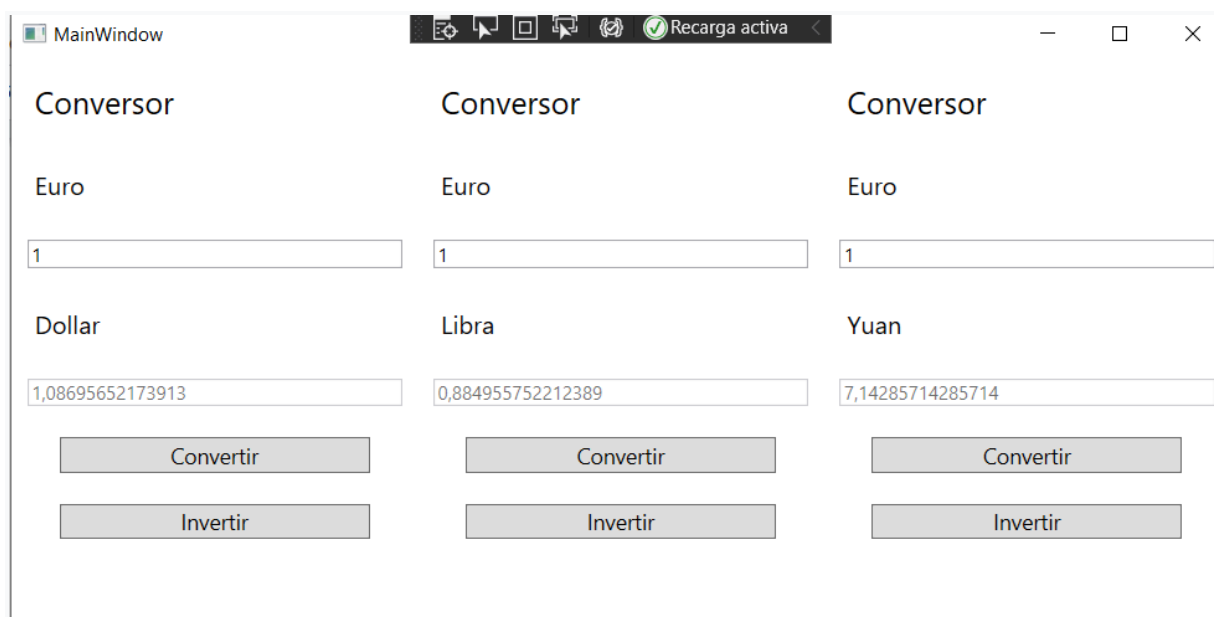
Si pulsamos “Invertir” podremos convertir de euros a la divisa correspondiente. Además se recalculará la conversión con las cantidades existentes en las cajas de texto origen.

Una vez creado el componente, lo utilizaremos para tres de las divisas (a elegir) en nuestra aplicación. Por ejemplo:



El campo donde se muestra el valor calculado será “no editable”.

Pulsamos Invertir:



Ejercicio 2. (NetBeans)

Se necesita crear una aplicación para realizar una encuesta con la siguiente información:

Para ello se deben crear dos componentes en Java:

Un JComboBox con valores predefinidos según un atributo que se informará en tiempo de diseño. Los valores podrán ser los días de la semana, los meses del año, o cualquier otro que se necesite en el componente.

Un RadioButton que modifique el tamaño del texto al pasar el ratón por encima y quede aún más grande cuando está marcado. Si está marcado no debe modificar su tamaño al pasar el ratón por encima.

Ejercicio 3. (Libre elección)

Creación de un componente personalizado en la plataforma que el alumno prefiera y que no se haya planteado en clase.

Deberá tener algún atributo que condicione su comportamiento y utilizar algún evento.

Valoración de ejercicios:

1 -> 4 puntos: creación de componente: 2, utilización efectiva en aplicación: 2

2 -> 4 puntos: creación de componentes: 2, utilización efectiva en aplicación: 2

3 -> 2 puntos: creación del componente: 1, uso efectivo en aplicación: 1

Examen Extraordinario - Junio 2024

Ejercicio 1 (Java)

Se pide crear una aplicación para gestionar el alta de alumnos en un instituto con una pantalla principal que incluya una opción de menú para crear nuevos registros (Alumnos -> Nuevo), y una tabla con los datos de los alumnos que se van grabando.

Desde la opción de menú se abrirá una ventana de alta con los siguientes campos:

- DNI,
- Nombre,
- Apellidos,
- Dirección,
- Código Postal,
- Fecha de nacimiento,
- Email,
- Ciclo,
- Edad (campo calculado no editable).

Requisitos del programa:

1 - Validaciones (avisando en caso de incumplimiento):

- El email debe incluir una @.
- El código postal debe tener 5 números.
- El DNI debe tener 8 números y una letra.
- Ningún campo puede estar vacío.

2 - El campo Edad se debe calcular al indicar la fecha de nacimiento o modificar dicho dato y al abrir la aplicación (en base a la fecha actual).

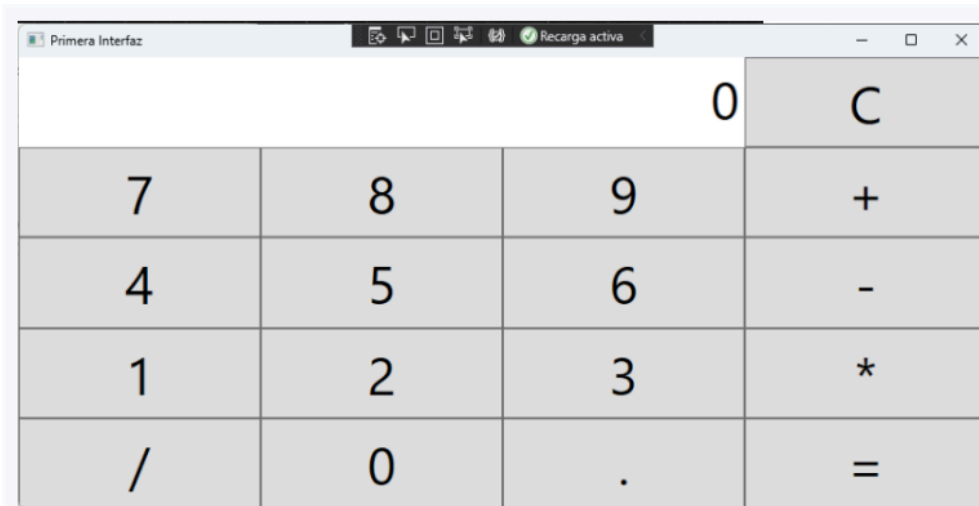
3 - Al “grabar los datos” se realizarán las validaciones y si se cumplen todas se guardará la información de los campos en un registro nuevo de la tabla de la ventana principal.

Valoración:

- Diseño dos pantallas:1
- Funcionalidad de carga de registros en tabla:1,5
- Validación de los campos: 1,5
- Gestión del campo calculado: 1

Ejercicio 2 (XAML/C#)

Se quiere crear una calculadora con componentes configurables.



El componente personalizable será un botón con un atributo (tipo) que condicione su formato / comportamiento según el valor que se le pase.

- Si tipo = OPERACION -> Formato 1, Comportamiento 1.
- Si tipo = RESULTADO -> Formato 2, Comportamiento 2.
- Si tipo = NUMERO -> Formato 3, Comportamiento 3.

El formato modificará como mínimo dos propiedades del componente (a elección de alumno).

El comportamiento cambiará alguna propiedad al activarse algún evento (a elección del alumno). El cambio debe ser temporal, recuperando el valor de las propiedades que se tenían antes de activarse el evento.

Valoración:

- Disposición en grid de componentes: 1
- Creación del componente: 1 formato, 1 comportamiento
- Uso efectivo en aplicación: 1,5
- Funcionalidad de calculadora: 0,5