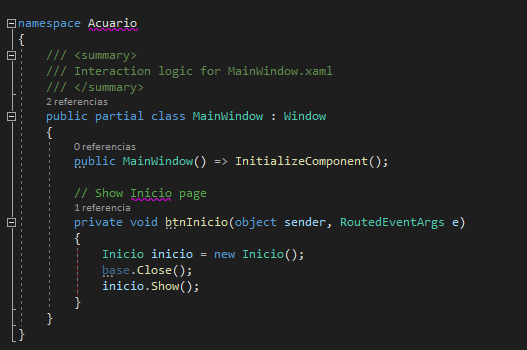
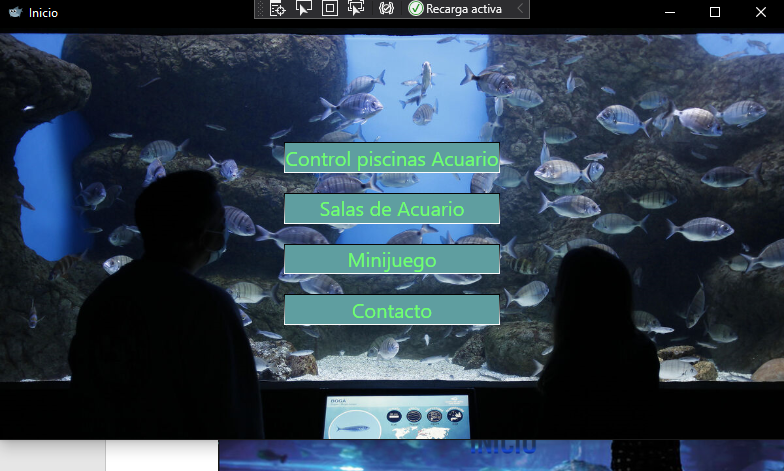
**GUIA ACUARIO**

Justo al Iniciar la aplicación contamos, con un BACKGROUND de un Acuario, junto con el nombre del ACUARIO con un efecto de difuminación como si estuviese dentro de agua y en la parte de abajo un botón de INICIO

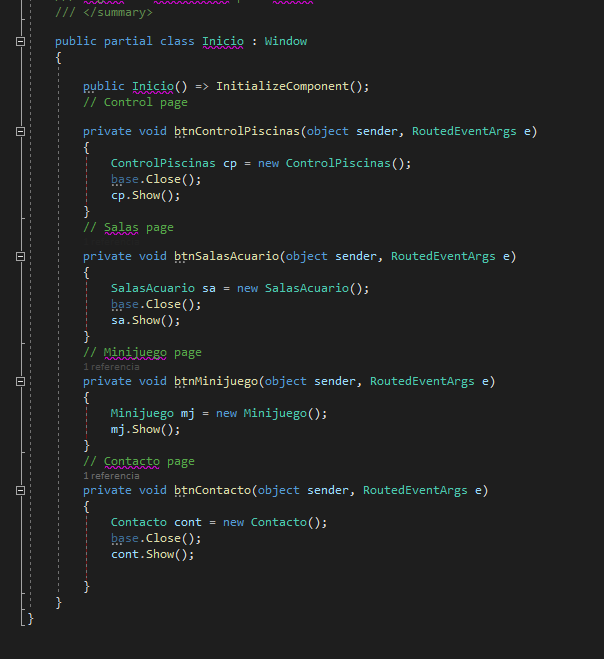




Al clickear el Botón de Inicio, nos mostrara el Menú de INICIO



En temas de C# solo cuenta con la creación del objeto de cada XAML para poder mostrarlo

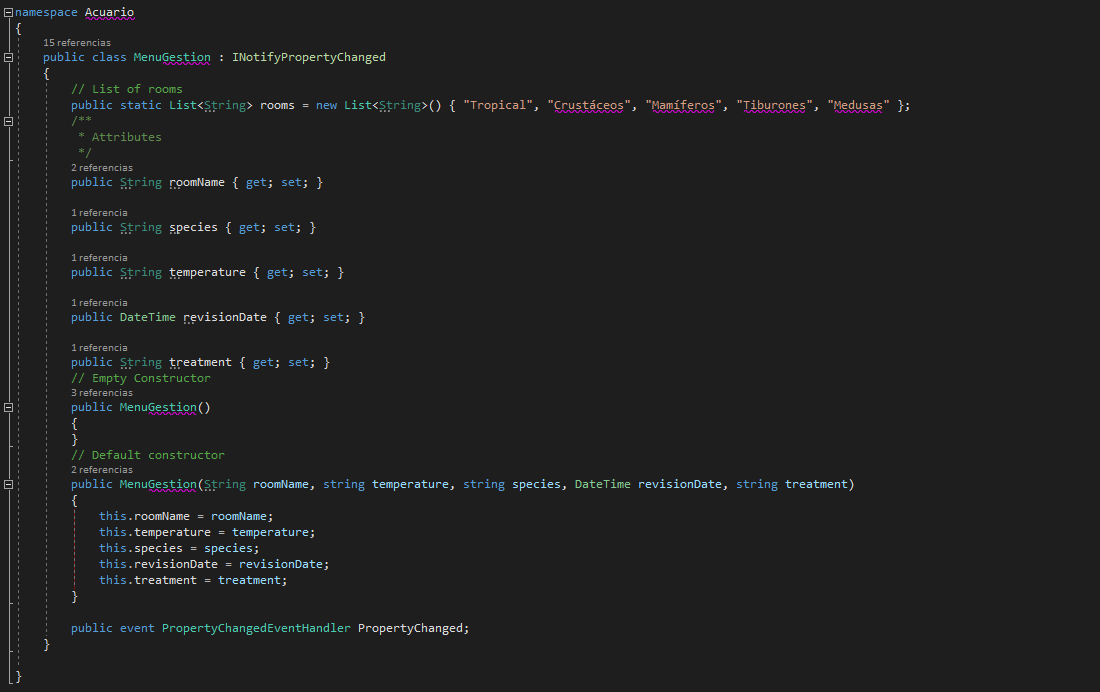


El menú de INICIO cuenta con las diferentes aplicaciones que dispone el ACUARIO

1. Control piscinas Acuario
2. Sales de Acuario
3. Minijuego
4. Contacto

PESTAÑA – CONTROL PISCINAS ACUARIO

Esta ventana, cuenta con un total de 2 Clases que recogen valores y muestran, y su C# propio



1º Clase MenuGestion, recoge los atributos y crea sus getters y setters

2º Clase Logistic



Crea una ObservableCollection de nuestro MenuGestion que es el que contiene los atributos y crea sus getters y setters

Despues implemente 3 metodos

addGestion: Simplemente añade un objeto a la lista

modifyGestion: Coge un index y modifica el objeto de ese index por el nuevo objeto

deleteGestion: Recoge el index y borra el objeto asociado a ese index



btnAgregar simplemente abre el formulario que vamos a usar para añadir

btnModify coge el Index seleccionado del datagrid, y tras esto abre el formulario con los valores asociados en el datagrid

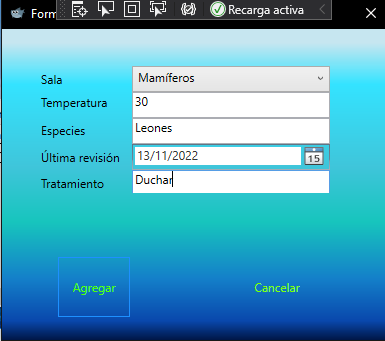
btnDelete coge el Index seleccionado del datagrid, y tras esto, usa la clase Logistic y borra el index de la lista



Cuenta con una TABLA (DataGrid), la cual muestra, NOMBRE SALA, TEMPERATURA, ESPECIA, FECHA DE REVISION Y TRATAMIENTO

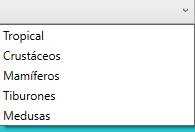
Tenemos diferentes botones, que nos facilitan la eliminación, modificación o creación de datos de la tabla.

1. Botón Agregar



(Datos de ejemplo, es imposible que haya Leones en un Acuario :D)

Cuenta con una COMBOBOX para poder elegir las diferentes salas del ACUARIO



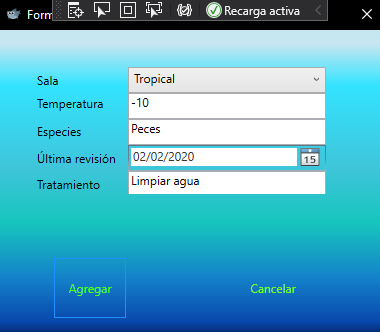
Simplemente rellenamos los campos o no los rellenamos, porque pueden ser NULL los campos

Y después de insertar lo que queramos inserta le damos a aceptar e instantáneamente se mostrara en la tabla anterior

1. Boton Modificar

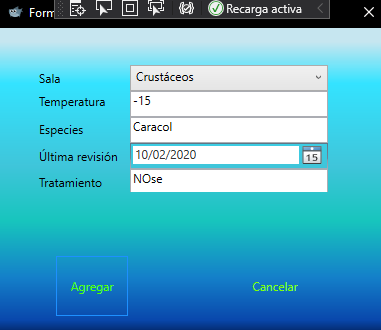


Seleccionamos la fila a modificar y clickeamos en el botón



Se abrirá el formulario con los valores de la fila seleccionada

Simplemente modificamos lo que queramos modificar y le damos a agregar

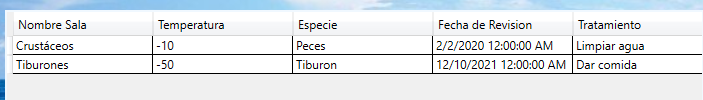




1. Boton Eliminar



Seleccionamos la fila a eliminar y acto seguido clickeamos el botón



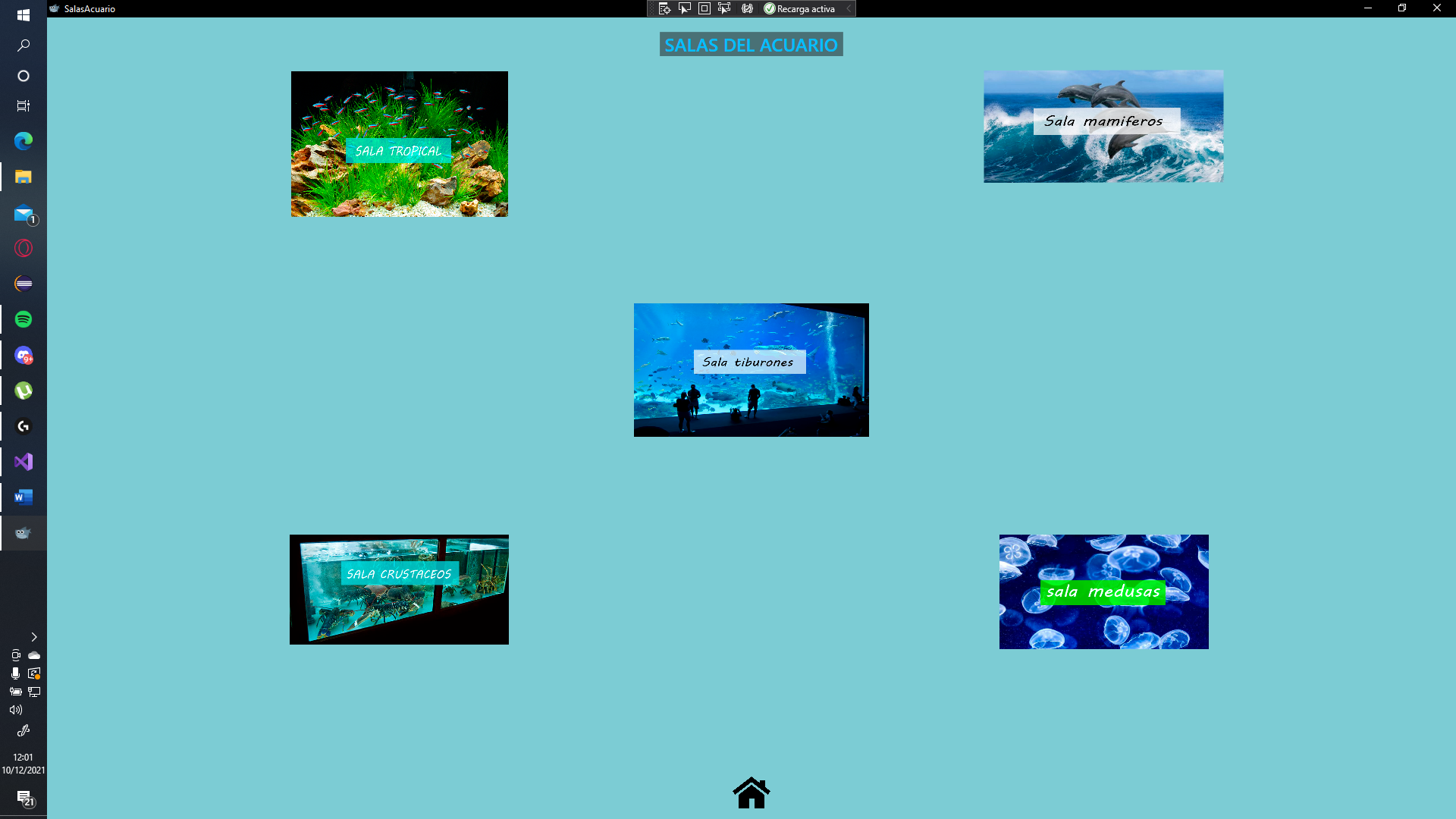
Y se habrá eliminado

Para volver a la pestaña Inicial simplemente clickeamos

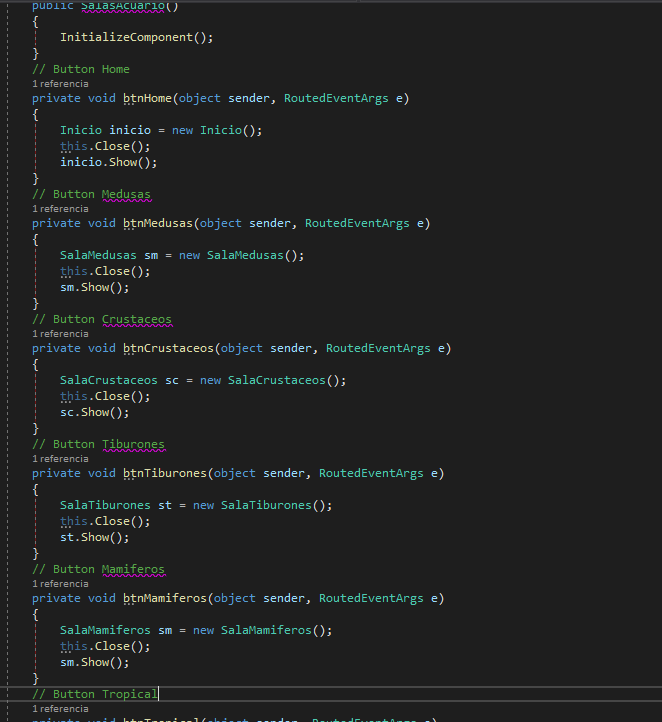


El botón que esta localizado en todo el medio, debajo de la ventana

PESTAÑA – SALAS DE ACUARIO



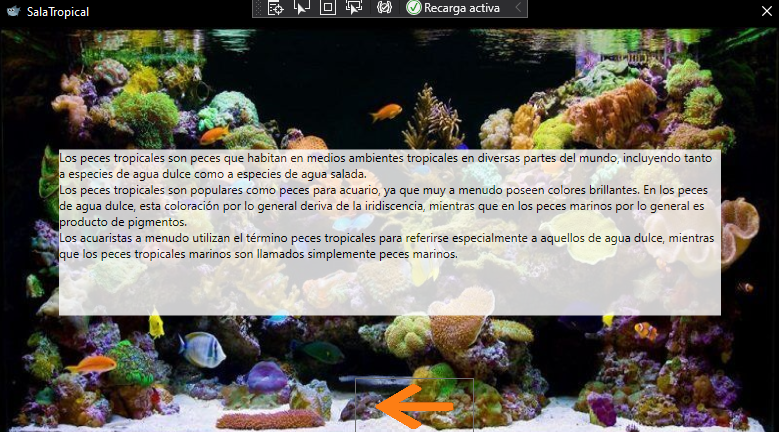
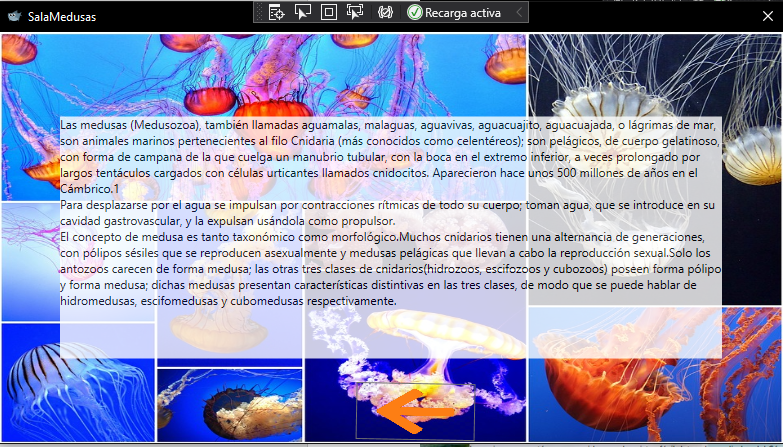
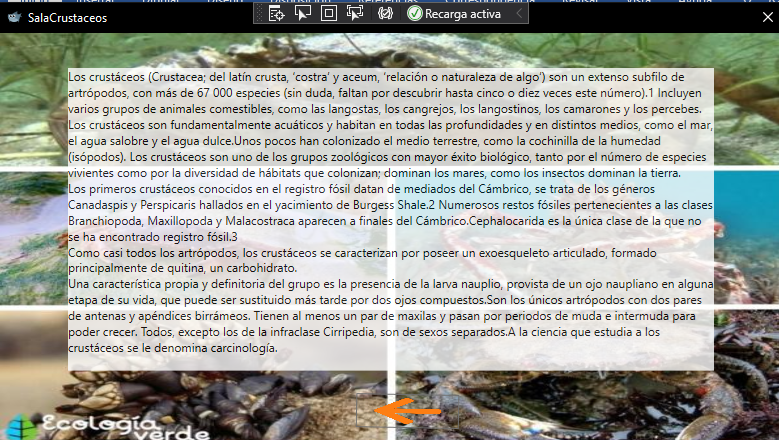
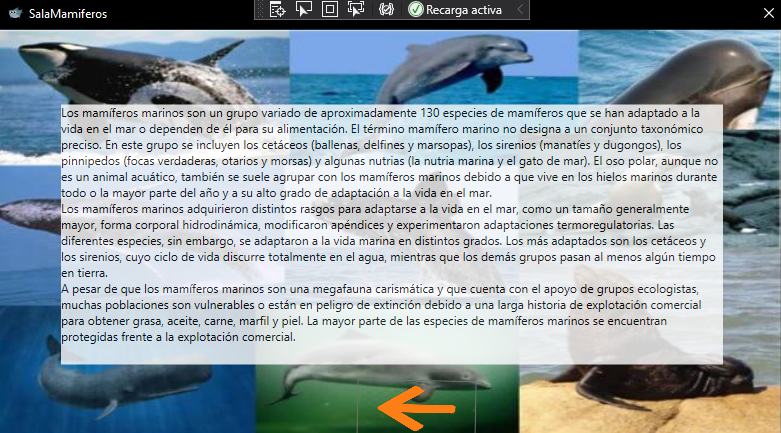
El C# contiene los métodos para abrir cada pestaña por cada sala

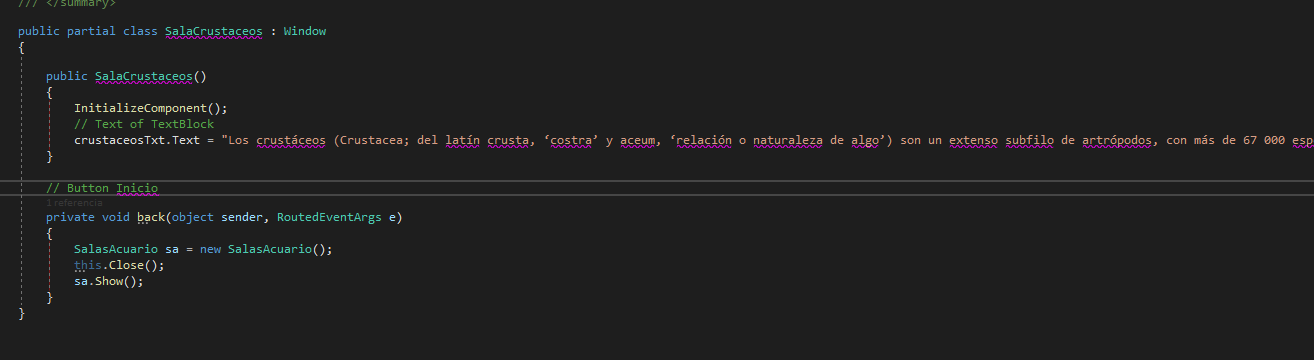


Al abrirla podemos observar 5 imágenes, con el nombre de cada sala.

Estas imágenes en realidad son botones y al clickear nos enviara a un pequeño apartado de información de los animales de cada zona

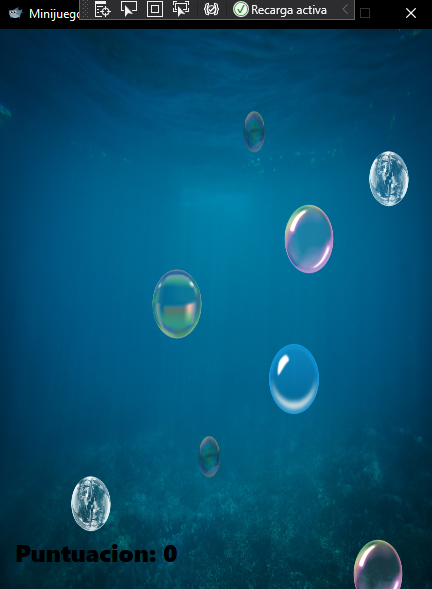
Sus diseños son todos similares



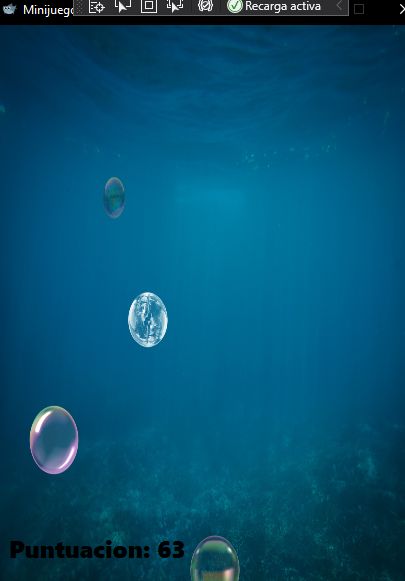
Todos los C# son iguales, per con diferente texto

PESTAÑA – MINIJUEGOS

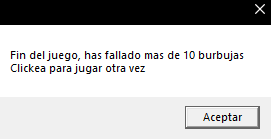


Es un minijuego que trata de explotar burbujas.

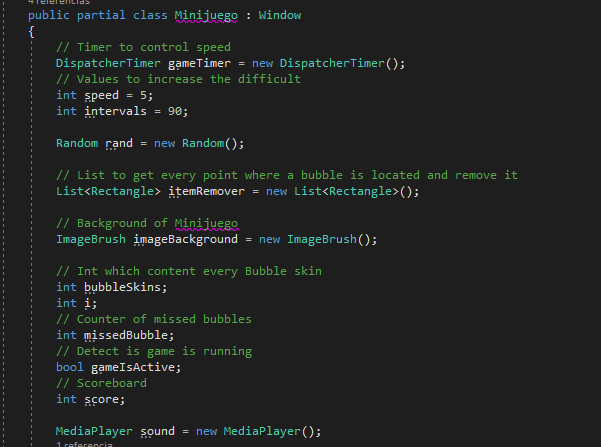
A cuanta más puntuación más dificultad



Si fallas 10 veces, salta un aviso de que has perdido, le damos a aceptar y se reinicia el juego



El C# asociado a Minijuegos es más complejo, se me ocurrió de repente y me puse a investigar como hacerlo, y vi que una librería me lo permitia y me puse a estudiarla. La librería en cuestión es Canvas y quedaría todo tal que asi.



Primera parte: Creamos las variables globales que nos hacen falta.

gameTimer, Speed, Intervals, para la dificultad

Random para posiciones

itemRemover para detectar los objetos en pantalla que haya sido clickeado

imageBackground para el fondo

bubbleSkins para tener variedad de burbujas

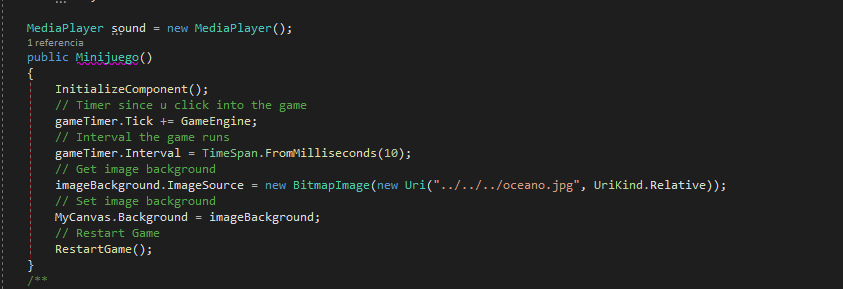
i para posiciones

missedBubble para perder el juego

gameIsActive controlador

Score para puntuación

Sound para el ruido de la burbuja

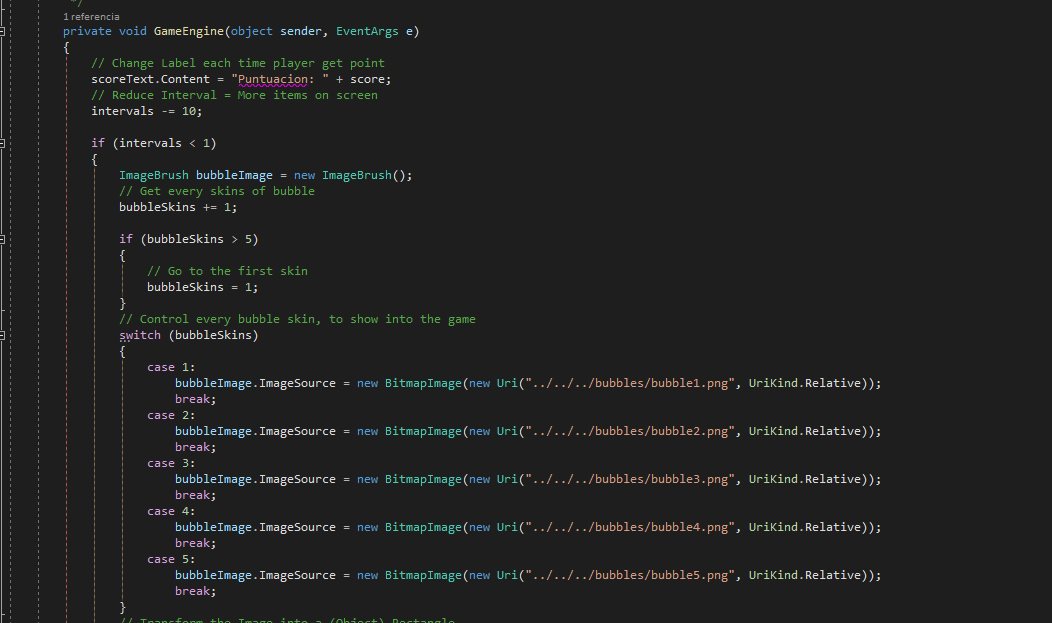


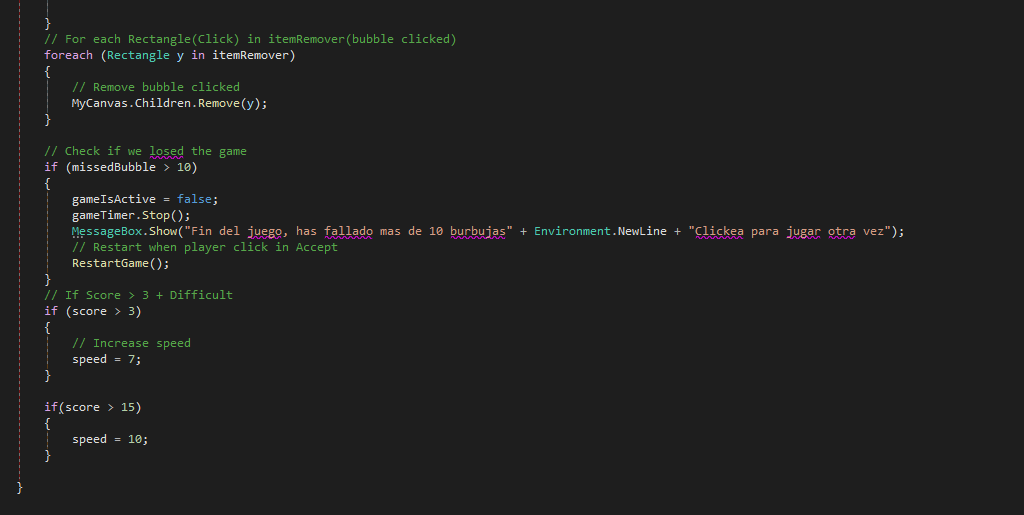
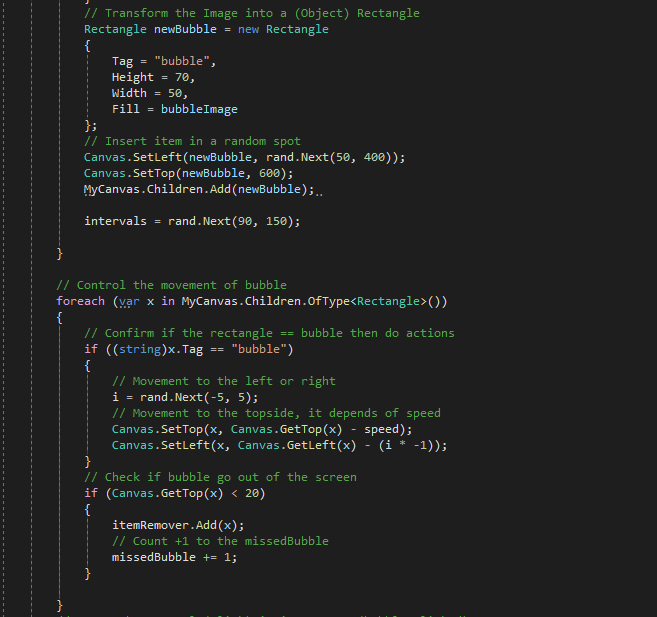
De constructor por defecto, hago que el gameTimer recoja los tick que hace el juego para controlar la aplicación 20 ticks = 1 segundo

Intervalo para la dificultad

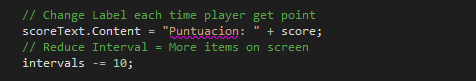
Asociamos al imageBackground la imagen, recogiéndola de un BitMap

Reiniciamos el juego para que tenga todos los valores por default, para que no se guarde si cerramos la pestaña y la volvemos a abrir, y así controlamos errores

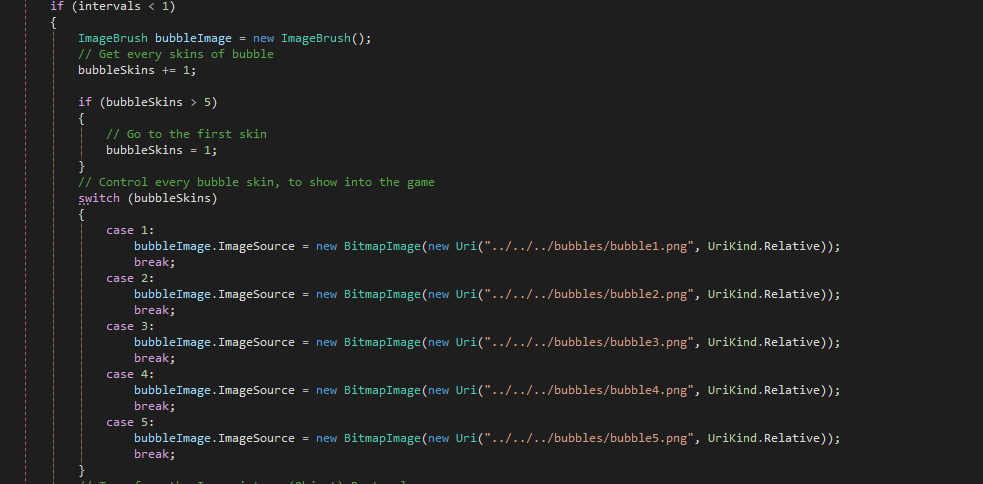




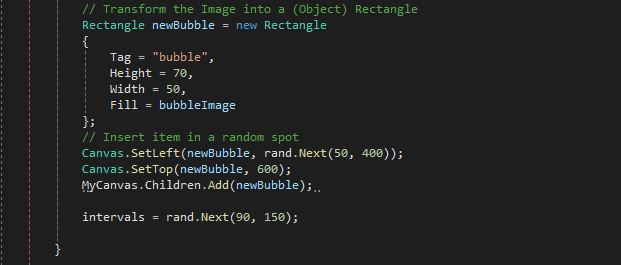
Por cada iteración, que es cada 10 ticks asociados anteriormente menos de un segundo el juego, establece en el Label de XAML el score actual, reduce el intervalo, por lo tanto, aparecen mas objetos en pantalla



Para controlar que aparezcan burbujas, ponemos la condición de Si el intervalo es positivo, significa que el juego sigue funcionando, por lo tanto, sigue hacia delante y hace un sumando de bubbleSkins para generar una bubble u otra, cada vez que la bubbleSkin llega a 5, se reasigna la bubbleSkin 1



Creamos un objeto Rectangulo y le asociamos los valores que queremos que tenga de diseño nuestra burbuja



Insertamos la burbuja en un spot random poniendole limites para que no aparezcan afuera y hacemos que se seleccione



Hacemos un for each, para controlar el movimiento de la burbuja

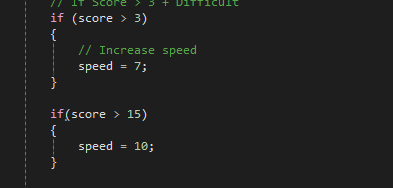
Primero hacemos que confirme mediante un tag si es una burbuja o no lo que esta cogiendo.

Despues con el i = rand.next(5,-5) hacemos que se mueva a izquierda y derecha, y controlamos la velocidad con la que sube al top de la ventana

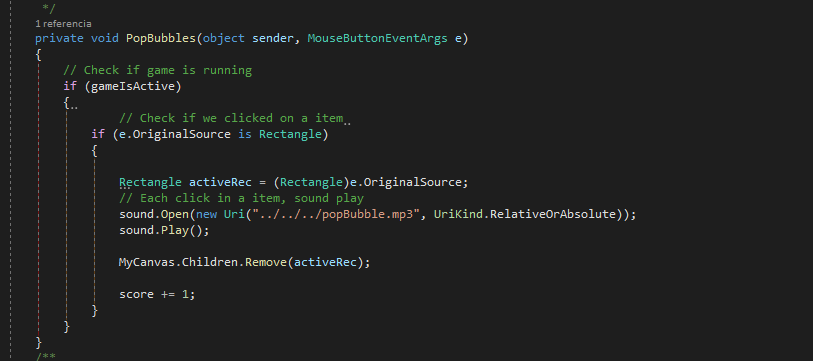
Despues tenemos otro if para controlar los que se van fuera de la ventana, que lo añadimos a la lista itemRemover y sumamos +1 al missedBubble

Despues creamos otro foreach para recorrer itemRemover y eliminar las burbujas que ya salieron fuera de la pantalla

Despues creamos un if que detecte la cantidad de missedBubles y si es mayor de 10 el juego para, y te salta un pop up de Fin del juego, el juego para reiniciar el juego el usuario debe de clickear en Aceptar



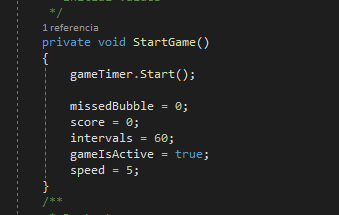
Y otros dos IF para aumentar la dificultad del juego según el score



PopBubbles

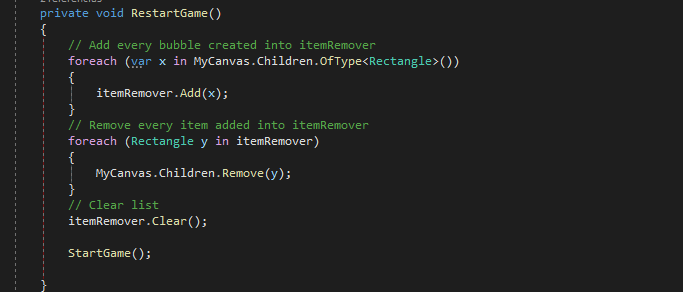
Detecta si el juego esta corriendo o no

Detecta si el evento (click) su objetivo es un rectángulo (la burbuja), al clickear en una burbuja, suena el POP de una burbuja al explotar y lo elimina de la pantalla y suma un +1 al SCORE



StartGame

Restablece todos los valores al inicial y activa el timer

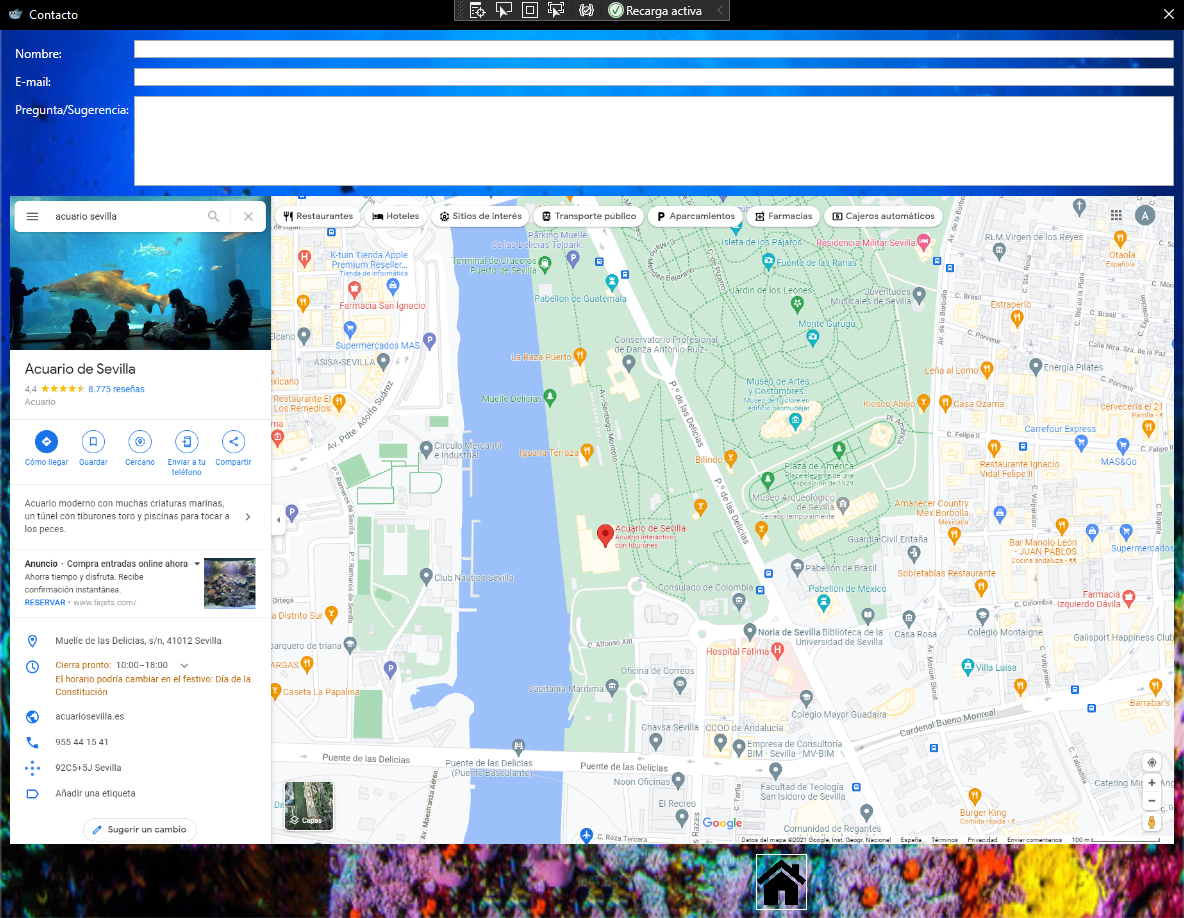


RestartGame

Elimina todas las burbujas, de la pantalla y del itemRemover

Despues de todo eso, restablece el juego y empieza

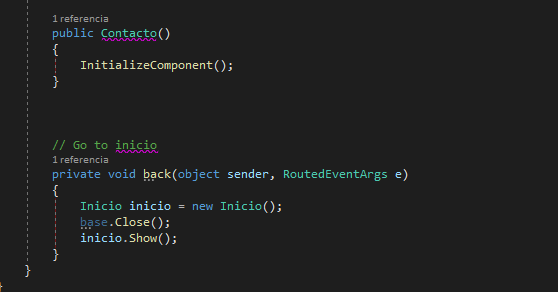
PESTAÑA – CONTACTO



Contiene un formulario simple, donde pones tu nombre, email y una pregunta o sugerencia

Y abajo hay un mapa con la ubicación del mapa

(El formulario no cuenta con una forma de enviar los datos)



Al ser un formulario básico, no contiene nada de código C#