

Universidad Panamericana

Materia: Programación Orientada a Objetos

Profesor: Luis Christian Alberto Coronel Ayala

Fecha de entrega: 16 de mayo de 2019

Ciclo: 1192

Nombre del proyecto: App Sala Maya

Miembros del equipo		
ID	Nombre	Carrera
0217787	Arias Cortina Carlos	LTISC
0213359	Cabrera Ábrego Raúl Andrés	LTISC
0215241	González Juárez Javier	LMECC
0211637	Guerrero Valadez Juan Manuel	LTISC

Rúbricas						
d-teams		e-identify		g-spanish-oral	g-spanish-written	
TR	VTM	IP	PSA		EA	EP

Contenido

1.Introducción	3
2.Descripción de la mejora	3
3.Descripción del programa	3
3.1.Clases	4
3.1.1.Usuario	4
3.1.2.Nawal	5
3.1.2.1.Glifo	6
3.1.2.1.1.GlifoEspecifico	6
3.1.3.Pieza	6
3.1.4.Dios	7
3.1.5.Maya	8
3.2.Formularios	8
3.2.1.Formulario de bienvenida	8
3.2.2.Iniciar sesión	9
3.2.3.Menú principal	10
3.2.4.Ingresar credenciales	11
3.2.5.Códices	12
3.2.6.Línea de tiempo	13
3.2.7.Resultados Línea de tiempo	16
3.2.8.¿Qué dios maya eres?	17
3.2.9.MessageBox Dios	18
3.2.10.¿Maya o no?	19
3.2.11.MessageBox Maya	21
3.2.12.Registrar	21
3.2.13.Registro	23
3.2.14. TimerDialog	24

3.3.Base de Datos	24
3.3.1.TRegistro	25
3.3.2.TNawales	25
4.Introduction	27
5.Description of the improvement	27
6.Description of the program	28
6.1.Classes	28
6.1.1.Usuario	28
6.1.2.Nawal	30
6.1.2.1.Glifo	31
6.1.2.1.1.GlifoEspecifico	31
6.1.3.Pieza	31
6.1.4.Dios	32
6.1.5.Maya	32
6.2.Forms	33
6.2.1.Welcome form	33
6.2.2.Log in	34
6.2.3.Main menu	34
6.2.4.Enter credentials	36
6.2.5.Codices	36
6.2.6.Time Line	37
6.2.7.Results	40
6.2.8.What Mayan god would you be?	41
6.2.9.MessageBox of gods	42
6.2.10.Is it Maya or not?	43
6.2.11.MessageBox Maya	44
6.2.12.Register	45
6.2.13.Registry	46

6.2.14.TimerDialog	47
6.3.Data Base	47
6.3.1.TRegistro	47
6.3.2.TNawales	48
Anexo 1: Código fuente	50
Anexo 2: Archivo enriquecido	90

1.Introducción

La historia de la humanidad ha tenido una gran cantidad de sucesos que han ido forjando nuestra sociedad donde vivimos, donde el origen se encuentra en las primeras civilizaciones, esparcidas alrededor del planeta. Una de estas civilizaciones es la Maya, la cual estuvo en territorios de las actuales Chiapas, Yucatán, Guatemala, Belice y Honduras, con una historia de unos 3.000 años.

Nosotros elegimos el Museo de Antropología, en específico la Sala Maya, debido a que es reconocido como uno de los más relevantes en el mundo, principalmente por atesorar la colección de arte prehispánico más completa del país, la cual, organizada en 24 salas temáticas, constituye el museo en su tipo de mayores dimensiones en toda América Latina.

2.Descripción de la mejora

Al visitar el museo, nos dimos cuenta cómo algunos niños sólo tomaban fotos a la descripción de las piezas o cómo la gente no apreciaba en realidad lo que representan.

Para evitar que esto siga pasando, necesitamos ganar la atención de los visitantes, y no hay mejor solución que con algo interactivo y agradable que no sea sólo una simple digitalización.

Nuestra solución consiste en un programa donde, a parte de demostrar tus conocimientos sobre la cultura Maya con formatos que no son de tipo examen, puedas ser aportado con temas más difíciles de comprender si no fuera por la tecnología; con un fin de despertar ese interés por nuestra amplia cultura.

3.Descripción del programa

En esta sección presentaremos las clases utilizadas para los distintos formularios, así como los formularios, los métodos utilizados y su interfaz gráfica.

Nuestro programa fue diseñado para ser intuitivo, es decir, que cualquiera pueda entender el funcionamiento del programa sin una capacitación previa, implementar cosas que no se han desarrollado anteriormente, ejemplo claro es el formulario Códices [3.2.5] que no solamente calcula tu nawal, sino te ayuda a comprender el significado de los glifos mayas, que podemos identificar en distintos códigos. También busca mantener entretenido al usuario con formularios variados y que dependiendo de un factor de aleatoriedad se generen opciones distintas a las de otro usuario. Esto lo podemos encontrar en el formulario de Línea de Tiempo [3.2.6] y ¿Es

maya o no? [3.2.9], los cuales además ofrecen un reto al visitante para demostrar lo visto en la exposición.

3.1.Clases

“Una clase es la descripción de un conjunto de objetos similares; consta de métodos y de datos que resumen las características comunes de dicho conjunto.”

[<https://www.masqueteclass.com/articulo/el-concepto-de-clase/>]

3.1.1.Usuario

Es posiblemente la clase más importante de todo el programa. Cada vez que un usuario nuevo empieza a interactuar con la aplicación, se inicializa un objeto de tipo Usuario el cual almacenará toda la información necesaria para ser almacenada en la base de datos.

```
Usuario  
string nombre;  
string correo;  
DateTime nacimiento;  
int resultadoLinea;  
string nawualString;  
Bitmap nawal;  
int puntajeMaya;  
string dios;  
int satisfaccion;  
string descripcion;  
bool archivoEnviado;  
  
Usuario();
```

Únicamente tiene un constructor vacío ya que nos interesa inicializarlo con valores nulos e ir definiendo los valores de las propiedades conforme se vayan llenando los formularios.

También tiene un objeto estático de tipo Usuario llamado *usuario*, la cual es accesible en toda la solución. Es en este objeto, en el que se guardará toda la información ingresada por el usuario, que posteriormente, se usará para generar un archivo personalizado con sus resultados, o como ya se había comentado, para almacenar la información en la base de datos.

Para simplificar el código, debido a que hay muchos casos en los que es necesario volver a inicializar el objeto *usuario*, se creó un procedimiento encargado de realizar esta acción. La cual se ejecutará en cualquier parte

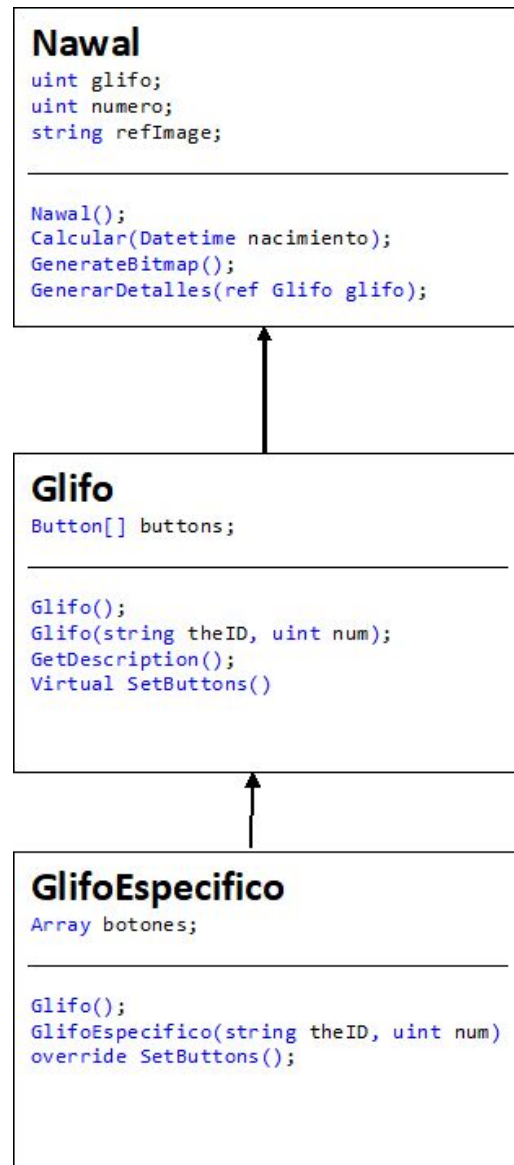
de la solución en caso de ser requerida, y de esta forma limpiar el objeto para guardar los datos de un usuario diferente.

3.1.2.Nawal

Clase generada para almacenar las propiedades del nawal generado a partir de la fecha de nacimiento del usuario, ingresada en el formulario Códices [3.2.5]

Los objetos de tipo *Nawal* siempre se inicializan utilizando el constructor vacío.

La función *Calcular* sirve para obtener el nawal del usuario a partir de su fecha de nacimiento con las siguientes fórmulas:



$$numero = \{(\Delta t \bmod 13 + 4) \bmod 13 \quad \text{si } [(\Delta t \bmod 13 + 4) \bmod 13 \neq 0] \quad 13 \quad (0)$$

$$glifo = (\Delta t \bmod 20) \quad (1)$$

Siendo Δt la diferencia de días entre la constante *refDate* que corresponde al 2/01/1910 y la fecha del parámetro *nacimiento*. Las ecuaciones [1] y [2] se obtuvieron a partir de una proporción calculada a partir de los datos proporcionados por: <http://mcd.gob.gt/calcular-nawal/>

La propiedad *numero* guarda el resultado de la ecuación [1] y *glifo* el resultado de la ecuación [2].

La función *nawal* se encarga de generar una imagen nueva a partir de la combinación de la imagen que haga referencia la propiedad *numero* y la referencia de la propiedad *glifo*.

Finalmente la función *GenerarDetalles* se encarga de instanciar objetos tipo Glifo [3.1.2.1]

3.1.2.1.Glifo

Subclase de *Nawal* [3.1.2] el cual tiene la propiedad *buttons*, el cual es un arreglo de botones. Tiene 2 constructores, uno vacío y otro que permite definir los parámetros de *refImage* y *numero*. Además tiene un nuevo procedimiento que permite generar la descripción dependiendo del *glifo* correspondiente al objeto. También tiene el método *SetButtons* que se encarga de instanciar un nuevo botón en el arreglo *buttons* para posteriormente identificar el número maya en el glifo.

3.1.2.1.1.GlifoEspecifico

Subclase de *Glifo* [3.1.2.1] el cual tiene un arreglo llamado *botones* que se llenará a partir de la columna *Simbologia* de la tabla *TRegistro* [3.3.1] de la base de datos correspondiente a su propiedad *ReflImage*, al inicializar el objeto

También hace polimorfismo al método *SetButtons* de *Glifo* [3.1.2.1] para que no solo cree un botón para identificar el número maya, sino además los botones del arreglo *botones*.

3.1.3.Pieza

Esta clase se utiliza en el Form de Timeline. Cada vez que un usuario empieza a interactuar con el formulario línea de tiempo, se instancia un arreglo de tipo *pieza* de tamaño 5, el cual contendrá las fechas que fueron asignadas a las piezas, las imágenes de las piezas y el número de aciertos.

Pieza

```
int tiempo;  
Bitmap pic;  
bool correct;  
uint orden;
```

```
Pieza();  
Pieza(int tiempo, Bitmap pic);
```

Tiene un constructor vacío ya que nos interesa inicializarlo con valores nulos e ir definiendo los valores de las propiedades conforme se vayan llenando los formularios.

También tiene un constructor con sobrecarga, el cual nos va a permitir guardar las fechas y las imágenes asignadas a esas fechas.

3.1.4.Dios

La clase Dios tiene como fin almacenar las propiedades de los dioses usados en el formulario QuizDios[3.2.8].

Dios

```
string nombre;  
string descripción;  
Bitmap imagen;  
uint puntos;
```

```
public Dios()  
public Dios(string nom, string desc, Bitmap imag)
```

La clase cuenta con un constructor vacío y un constructor con parámetros para inicializar los valores.

La clase contiene un arreglo de datos con 5 Dioses distintos, teniendo descripción e imagen propia.

3.1.5.Maya

```
Maya  
int Img;  
Random random;  
int score;  
bool bip;  
int contador;  
  
int Picture();
```

Se usó una sola clase “Maya”, esta nos permite tener un ArrayList, el cual se encarga de no repetir imágenes en el Form; 3 atributos que nos permiten llevar la puntuación, limitar los intentos del Form e indicar el momento en el que los botones deben parpadear; 2 constructores, uno vacío y el delimitado por el puntaje y el bool de parpadear.

3.2.Formularios

Los formularios son la unidad básica de la aplicación, es esencial realizar algunas consideraciones sobre su función y diseño. En última instancia, un formulario es una pizarra en blanco que usted, como desarrollador, mejora con controles para crear una interfaz de usuario y con código para manipular los datos. Para ello, Visual Studio proporciona un entorno de desarrollo integrado (IDE) que ayuda a escribir código, así como un enriquecido conjunto de controles escrito con .NET Framework. Complementando la funcionalidad de estos controles con el código, puede desarrollar las soluciones que necesita de forma fácil y rápida.

Recuprado de :

<https://docs.microsoft.com/es-es/dotnet/framework/winforms/>

3.2.1.Formulario de bienvenida

Es el primer formulario que se muestra al abrir el programa. Sirve para presentar el título del programa, los nombres e información de los desarrolladores y de de la aplicación.

Bienvenida

¡Bienvenido al Mundo Interactivo Maya!

UNIVERSIDAD PANAMERICANA

MNA MUSEO NACIONAL DE ANTROPOLOGÍA

Sala Maya

Desarrolladores:

ID	Nombre	Carrera
0217787	Arias Cortina Carlos	LTISC
0213359	Cabrera Abrego Raul Andrés	LTISC
0215241	González Juárez Javier	LMECC
0211637	Guerrero Valadez Juan Manuel	LTISC

Continuar

Figura 1: Formulario de bienvenida

El formulario cuenta con un único botón ubicado en la parte inferior con el nombre *Continuar*. Al presionarlo, accionará una función que ocultará el formulario de bienvenida y mostrará el formulario de inicio de sesión del que hablaremos a continuación.

3.2.2.Iniciar sesión

Es el formulario que se muestra a los administradores para poder tener acceso al menú principal. Para ello es necesario ingresar su nombre de usuario y contraseña correctamente. En caso de que el usuario ingrese un nombre de usuario o contraseña erróneo más de 3 veces, el programa se cerrará por motivos de seguridad.

Usuario:

Contraseña:

Iniciar sesión (A)

Salir (B)

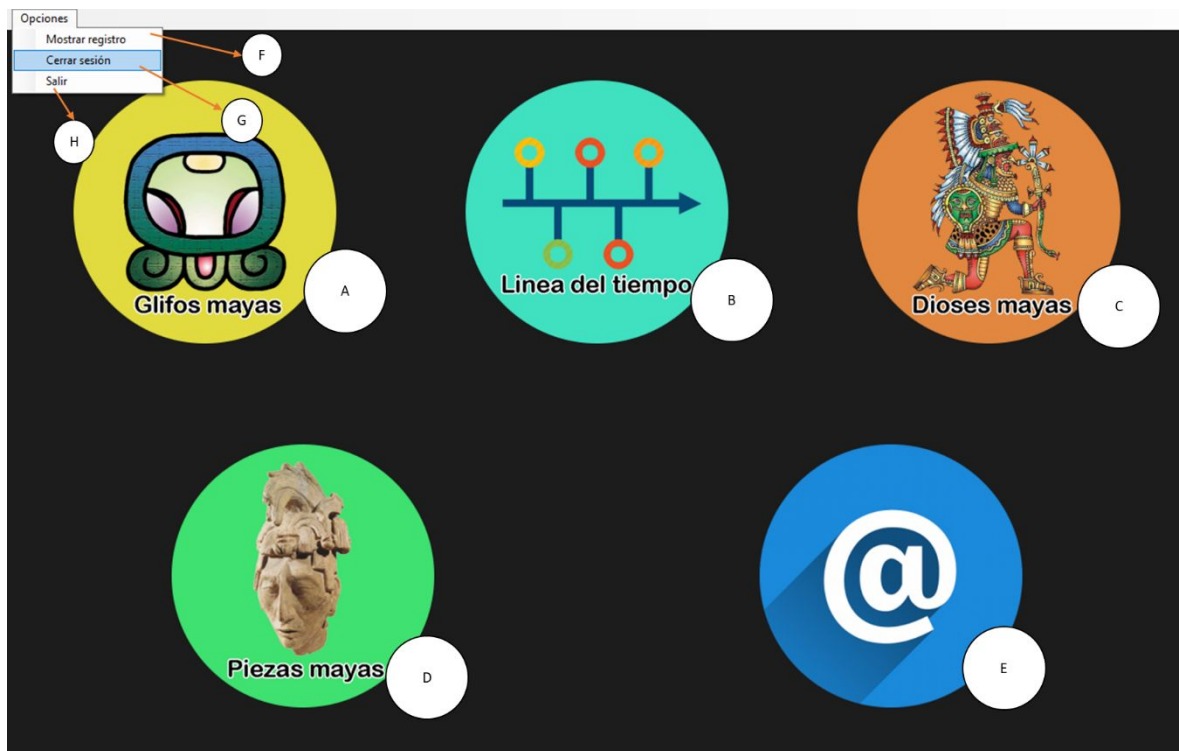
Figura 2: A) Botón de inicio de sesión. B) Botón para salir

El formulario se encuentra definido en la clase *LogIn*, la cual contiene un atributo privado de tipo entero llamado *nErrores*. Al presionar el botón 1 se verificará que *nErrores* es menor a 3. En caso de ser mayor o igual a tres se cerrará el programa.

Posteriormente, se compararán todos los valores entre dos arreglos que contienen todos los nombres de usuario, contraseñas y los valores ingresados por el usuario. Si los datos ingresados son correctos, el formulario se ocultará y posteriormente mostrará el menú principal. En caso contrario, se sumará uno al contador *nErrores*.

3.2.3. Menú principal

Es el formulario usado para acceder al resto de los formularios interactivos. Además hay un *MenuStrip Opciones* el cual despliega las opciones de *Mostrar registro*, *Cerrar sesión* y *Salir*. Al oprimirlas se mostrará el formulario *Ingresar credenciales* [3.2.4], para evitar que un usuario sin permisos de administrador pueda visualizar esos formularios. El control *Mostrar registro* del *MenuStrip Opciones* permite el acceso a los administradores al formulario que *Registro* [3.2.13], para muestra la *Tabla TRegistro* de la base de datos. El control *Cerrar sesión*, oculta el Menú principal y muestra el formulario *Iniciar sesión* [3.2.2]. Finalmente el control *Salir*, cierra el programa. Cabe señalar, que este formulario y el formulario de *Iniciar sesión*, son los únicos que permiten cerrar el programa, para que solo los administradores puedan hacer uso de este.



El formulario cuenta con un arreglo de formularios llamado *subForms*, el cual contiene todos los formularios interactivos: *Códices*[3.2.5], *Línea de tiempo* [3.2.6], *QuizDios*[3.2.8], *¿Maya o no?*[3.2.10], y *Registrar*[3.2.11].

Al pulsar cualquier botón circular se llamará a la función *btnAbrirForm* la cual instanciará e inicializará el formulario correspondiente al botón pulsado, y se ocultará el menú.

Todos los formularios abiertos desde el menú tienen un botón para regresar al menú.

3.2.4.Ingresar credenciales

Este Formulario es muy básico pero tiene gran relevancia, ya que a través de este, podemos acceder a la BDD y salir del programa.

Contiene sólo un *if* y un *else*, que verificarán que el usuario y la contraseña sean correctos, en caso de no serlo, se marcará en rojo hasta que los datos sean los indicados o si el usuario decide cancelar.

Esta verificación se da gracias a que la clase *LogIn* [3.2.2] tiene una función llamada *VallidarUsuario*, y es en la *partial class IngresarCredenciales* donde mandamos a llamar a esta función.



Figura 5) El Formulario avisa al visitante si los datos ingresados

3.2.5.Códices

Es un formulario con la capacidad de mostrar al usuario su nawal maya, el cual cambia dependiendo del mes, día y año de nacimiento de la persona. Los nawaes significan energía, espíritu o fuerza de los seres y elementos de la naturaleza. Según la cosmovisión maya, son símbolos que representan y vinculan a cada persona con el ecosistema.



Ilustración 2A) Ingresar fecha de nacimiento B) Regresar al menú C) Calcular nawal D)Nombre del nawal E) Descripción del nawal F)Botón interactivo presionado H) Símbolo resaltado según el botón F

El formulario aparte de calcular el nawal, genera la imagen correspondiente al nawal, una breve descripción y unos botones interactivos que muestran el significado de cada parte del nawal.

Al oprimir el botón Calcular (C) se invocará la función `button1_Click`, la cual guardará la fecha de nacimiento del visitante en el objeto *usuario* de tipo *Usuario* [3.1.1] en su propiedad *nacimiento*; generará un nuevo objeto de tipo *Glifo* [3.1.2.1] e invocará su función *GenerarDetalles* para poder mostrar en el formulario el nawal del usuario.

En caso de que el usuario ingrese en su fecha un carácter inesperado como una letra, una fecha falsa (menor al 2 de enero de 1910) o una fecha que no exista, se maneja la excepción enviándole al usuario que su fecha no es válida.

3.2.6.Línea de tiempo

Es el formulario que se muestra al pulsar el botón B del formulario de Menú (Imagen x). La finalidad de este Form es tener una interacción con el usuario por medio de un juego didáctico en el que se busca ordenar de forma cronológica piezas Mayas. De esta forma el usuario podrá visualizar mejor la cronología de la civilización Maya.

El funcionamiento del Form se puede explicar en 4 simples pasos:

Es el formulario que se muestra al pulsar el botón B del formulario de Menú (Imagen x). La finalidad de este Form es tener una interacción con el usuario por medio de un juego didáctico en el que se busca ordenar de forma cronológica piezas Mayas. De esta forma el usuario podrá visualizar mejor la cronología de la civilización Maya.

El funcionamiento del Form se puede explicar en 4 simples pasos:



Figura 4: Seleccionar Imagen

1) Seleccionar Imagen:

Dirija la flecha del mouse a la imagen que quiere mover, una vez ahí mantenga presionado el click derecho para seleccionar la imagen como se muestra en la (figura 4).



Figura 5: Arrastrar Imagen

2) Arrastrar Imagen:

Dirija la flecha del mouse al espacio en el que quiere poner la imagen, al mismo tiempo mantenga presionado el click izquierdo como se muestra en la (figura 5).



Figura 6: Soltar Imagen

3) Soltar Imagen:

Deje de darle el click izquierdo y se mostrará la imagen seleccionada en el espacio como se muestra en la (figura 6).



Figura 7: A) Regresar o B) Calificar

4) Calificar o Regresar:

Si decide darle click izquierdo en el botón de A de la (figura 7), regresara al From de Menú.

Si decide darle click izquierdo en el botón de B de la (figura 7), se mostrará el From de Resultados, en el cual se le indicará los aciertos que obtuvo el usuario como se muestra en la (figura 8).

3.2.7.Resultados Línea de tiempo

Este es el primer formulario que se muestra al darle click izquierdo en el botón B de la (figura 7). Sirve para mostrar los resultados que obtuvo el usuario al finalizar el juego de Línea de Tiempo.

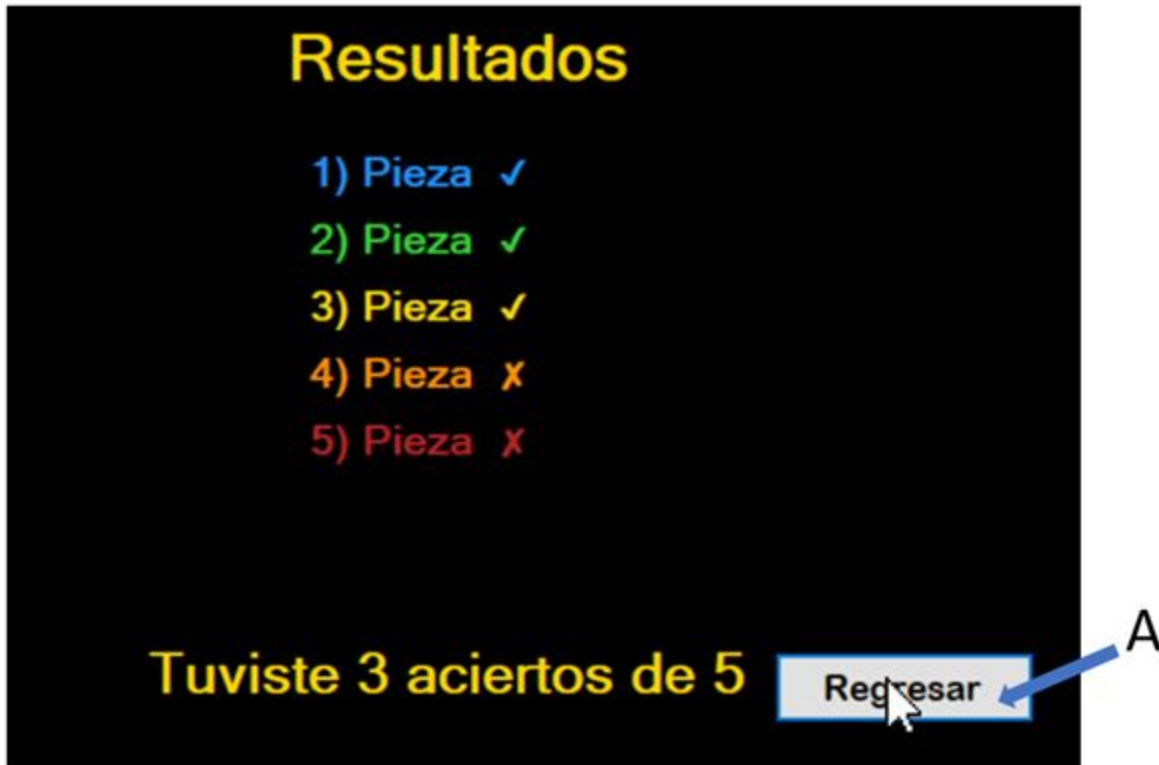


Figura 8: A) Botón Regresar.

Al darle click izquierdo en el botón de A de la (figura 8), regresara al From de Menú.

3.2.8.¿Qué dios maya eres?

El formulario es como un "Test" en el que el usuario elige las características que crea que más se asemejan a él. Dependiendo de su selección, se mostrará el dios que más se asemeje a esas mismas características. Esto le ayudará al visitante conocer un poco más sobre los dioses mayas y despertar su interés por la cultura.

El programa consiste en 6 preguntas sencillas donde sus distintas respuestas tienen un significado para determinar al Dios, una vez que este sea calculado y mostrado junto con su imagen y su descripción, será guardado en el objeto Usuario[3.1.1] en su propiedad Dios, para posteriormente ser guardado en la base de datos.

¿QUÉ DIOS MAYA SERÍAS? Regresar

¿Con qué color te identificas?

☐ ☐ ☐ ☐

¿Qué frase célebre te gusta más?

- ☐ "Preferiría la paz más injusta a la más justa de las guerras"
- ☐ "Amad el arte, entre todas las mentiras es la menos mentirosa"
- ☐ "Ganando como siempre"

¿Con qué palabra te describirías?

- ☐ Viajero
- ☐ Ambicioso
- ☐ Creativo

☐ Leer un libro
☐ Salir a pasear
☐ No tengo tiempo para eso

Si fueras un animal, serías...

- ☐ Serpiente
- ☐ Búho
- ☐ Jaguar
- ☐ Colibrí

¿Que superpoder te gustaría poseer?

- ☐ Omnisciencia
- ☐ Super fuerza
- ☐ Factor de curación
- ☐ Inmortalidad

Calcular

Al seleccionar el botón Calcular, se activará un ciclo foreach donde accederemos a todos los GroupBox del Form para ver cuántos puntos se le debe agregar a cada Dios según el valor de i en la propiedad Tag de cada RadioButton.

La variable booleana *validar* nos asegurará que estén seleccionados los RadioButton, y si el usuario deja una respuesta en blanco, se desplegará un MessageBox para advertirle; esto nos permite evitar errores y, por lo tanto, no hubo manejo de excepciones.

3.2.9. MessageBox Dios

ERES:

IbName
IbDesc

Regresar

ERES:

Itzamná
Dios del sol y la sabiduría. También señor del cielo, el día y la noche.

Regresar

Figura 1) Formulario MessageBoxShow antes de ser invocado

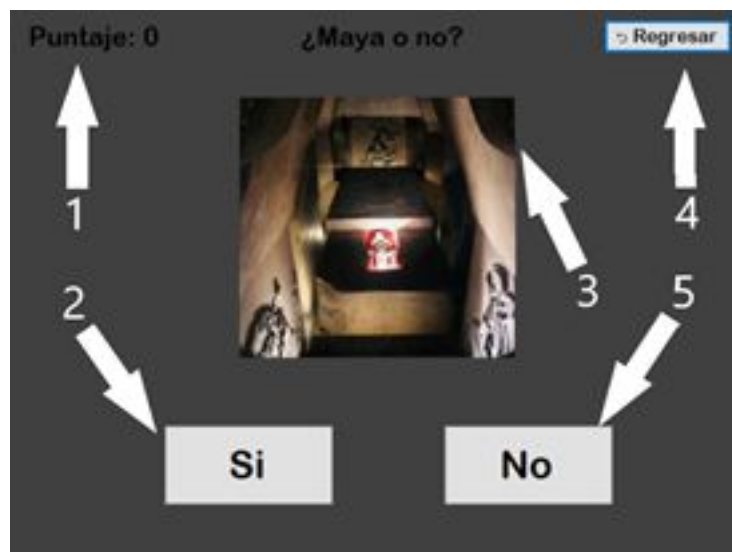
Figura 2) Formulario MessageBoxShow mostrando la imagen, descripción y nombre del Dios correspondiente.

Una vez que ya se tenga al Dios, se mostrará a través de la clase MessageBoxDios, la cual desplegará los datos del Dios en sus Label y pictureBox.

3.2.10.¿Maya o no?

La finalidad de este Form es tener una interacción con el usuario de forma que se permita tener una pequeña retroalimentación con un juego didáctico que seguro le permite entender un poco más sobre la artesanía que nos dejó la civilización Maya. Usando imágenes, se hace un pequeño cuestionario de imágenes sobre las pertenecientes a la civilización Maya.

Por parte del Form, se puede apreciar un diseño bastante simple, pero funcional. El objetivo del Form es permitir que el usuario tenga un entendimiento completo de su uso sin tener opción de generar un error en el mismo.



1. Puntaje

2. Botón si

3. Imágenes

4. Botón regresar

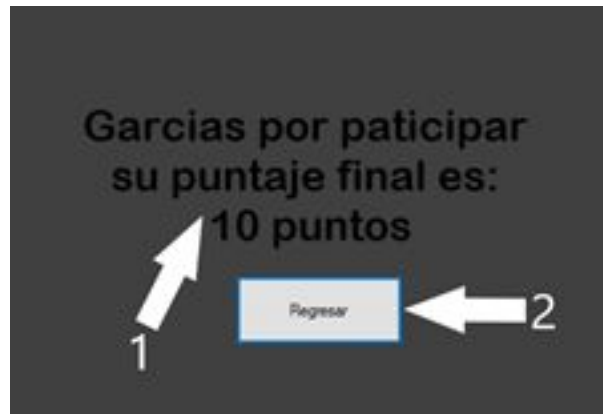
5. Botón no

El funcionamiento del Form es algo muy sencillo. Como se observa, existen solamente 3 botones. El usuario cuenta con 5 oportunidades para obtener tantos puntos como le sean posibles. El usuario debe de diferenciar si la artesanía que se aprecia en la pantalla de procedencia Maya o no. Para esto usa los botones del Form. Una vez oprimido el botón, existen dos posibilidades. La primera es que la respuesta esté correcta y el botón enciende y apaga color verde por 3 segundos. La segunda es que la respuesta esté incorrecta y de igual forma el botón enciende y apaga, en esta ocasión, color rojo. Cuando la respuesta es correcta se suman 2 puntos y cuando la respuesta está equivocada, el puntaje no es modificado. Cuando terminan los 5 intentos, el puntaje final aparece en un nuevo Form por cuestiones meramente estética.

En este Form no se necesitaron excepciones pues no existen datos que pueda modificar el usuario que generen algún error. Es gracias a esto que el manejo del Form es extremadamente sencillo para los demás. Se utiliza 1 método muy sencillo para regular el número de intentos que tiene la persona para que de igual forma las imágenes que aparecen en el juego sean varias y no se repitan o pase lo menos posible. Este procedimiento está dentro de la clase "Maya" [3.1.5.]. Se usa un ArrayList con el número de imágenes y se borran paulatinamente para que así no se repitan en el mismo juego.

Por último, el uso del Form es muy amistoso, como ya se explicó, el objetivo es sumar puntos al diferenciar distintas artesanías. Una vez terminados los 5 intentos, en la parte superior izquierda aparece el puntaje, se abre el nuevo form que nos permite conocer nuestro puntaje final. Al oprimir el botón regresar, se guarda el puntaje final en la clase Usuario [3.1.1.] y regresa al menú principal [3.2.3].

3.2.11. MessageBox Maya



1. Label Puntaje
2. Botón Regresar

Este Form está hecho con un label y un botón. La función del Form es mostrar el puntaje final del usuario y guardar este dato en el puntaje final de la clase Usuario [3.1.1]. A pesar de ser un Form muy sencillo, ayuda a regresar el ArrayList de la clase Maya [3.1.5] a su configuración primaria. Cuando el usuario oprime el botón de regresar el Form ¿Maya o no? [3.2.10] se muestra en caso de que el usuario quiera volver a jugar o quiera regresar al Menú Principal [3.2.3].

3.2.12.Registrar

Es otro formulario de gran importancia, ya que es en este donde se guarda la información del usuario en la base de datos. Además, proporciona una información de gran importancia para el museo, que es la valoración de los usuarios sobre su experiencia en la aplicación.

¡Envía tus resultados por correo electrónico!

Regresar

*Califica la app:

☒
☐
☐

Nombre: Juan

*Correo electrónico: 0211637@up.edu.mx

Enviar

Ilustración 3A) GroupBox con radioButtons para calificar con 10, 5 o 0. B) Datos del visitante para enviar el correo C) Llama al método para enviar el archivo D) Regresar al menú

También cuenta con una función única, y es que permite al visitante enviar la información de su sesión a su correo electrónico en un archivo *.rtf (Anexo 2), es decir un archivo de texto con formato para generar una experiencia visual más agradable al usuario a la hora de leer sus resultados y además incluye la imagen de su nawal, cosa que no es posible hacer en un archivo *.txt.

Este proceso ocurre al presionar el botón Enviar (C) del formulario. El cual llama la función `button1_Click`. El archivo se genera a partir de la clase estructurada `GeneradorDeArchivos` la cual contiene varias constantes que denotan el formato enriquecido y una función `GenerarArchivo` la cual va concatenando las constantes con los valores del objeto usuario. Al terminar el proceso, guarda la cadena de texto en el archivo.

Una vez generado el archivo, con ayuda de la librería `System.Net.Mail`, se tratará de enviar un correo electrónico de acuerdo a la información proporcionada por el usuario. En caso de que no haya conexión a internet o no se pueda enviar por un error del servicio, se notificará al usuario que no se logró enviar el archivo pero que más adelante se le enviara. Y esta acción está planeada para ser ejecutada manualmente por un administrador al encontrar en la columna `CorreoEnviado` de `TRegistro` un

valor falso. Si se logra enviar el correo, se guardará el usuario en la base de datos con el valor de *CorreoEnviado* como verdadero.

En caso de que no se pueda enviar porque el usuario no proporcionó la información correcta o porque no calificó la aplicación, se le notificará al usuario que compruebe sus datos y se abortará la instrucción de enviar al correo.

3.2.13.Registro

Es un formulario el cual es de acceso exclusivo para los administradores, ya que solo través del formulario Ingresar credenciales [3.2.4] la única forma de entrar. La estructura del formulario es muy sencilla, al abrirse se ejecuta un query, el cual se encarga de leer la tabla *TRegistro* [3.3.1] de la base de datos y muestra la base de datos en un datagridview.

	Id	Nombre	Correo	Nacimiento	Linea del Tiempo	Nawal	¿Es Maya?	Dios	Satisfacción	CorreoEnviado
26	Juan	0211637@up.edu.mx	21/12/1999	6	13 B'atz'	10	Kukulcán	10		<input type="checkbox"/>
27	Juan	0211637@up.edu.mx	23/12/1999	-1	2 Aj	-1		10		<input type="checkbox"/>
28	Giancarlo	0215420@up.edu.mx	10/12/1999	6	2 Ajpu	10	Chaac	10		<input checked="" type="checkbox"/>
29	Juan Guerrero	0211637@up.edu.mx	21/12/1999	10	13 B'atz'	8	Kukulcán	10		<input checked="" type="checkbox"/>
30	Jose Lopez	juan.guerrero.v99@gmail.com	01/01/2001	-1		-1	Kauil	5		<input checked="" type="checkbox"/>
31	Javier	salamaya.upsolutions@gmail.com	01/01/2001	-1		8		0		<input checked="" type="checkbox"/>
32	Joven	joava@email.com	25/02/2001	-1	3 Aq'ab'al	-1		0		<input checked="" type="checkbox"/>
33	fas	ga@afg.dgs	01/01/2001	-1		-1		10		<input type="checkbox"/>
34	Antiguo	0211637@up.edu.mx	01/01/2001	-1		0	Kukulcán	10		<input checked="" type="checkbox"/>
35	Jan	021163@fa.as	01/01/1000	6		-1		10		<input checked="" type="checkbox"/>
36	fgd	ga@gsd	01/01/1000	2		-1		10		<input checked="" type="checkbox"/>
37	Raúl Cabrera	raulmamba@hotmail.com	01/03/2000	6	6 Iq'	8	Kauil	10		<input checked="" type="checkbox"/>
38	Juan	0211637@up.edu.mx	03/03/1234	-1		0		10		<input checked="" type="checkbox"/>
39	0211637	0211637@up.edu.mx		-1		-1	Kauil	10		<input checked="" type="checkbox"/>
*										<input type="checkbox"/>

A → Guardar cambios
→ B Regresar

También tiene un botón para actualizar los datos de la columna *CorreoEnviado* con los datos modificados desde el programa. Cabe señalar que sólo puede modificarse la columna *CorreoEnviado*. El objetivo de esta columna es para marcar aquellos usuarios a los que ya se les envió un correo electrónico con sus resultados, como se menciona en la sección 3.2.12.

Y también tiene su botón para cerrar el formulario y volver al menú.

3.2.14. TimerDialog

Este Formulario es un detalle adicional al programa, sirve para detectar si el visitante abandonó la app antes de registrarse; por lo que es necesario reiniciar el objeto usuario [3.1.1] y regresar al menú con el fin de que el programa sea más atractivo a los nuevos visitantes que ingresen al sistema.

El Timer se resetea cada vez que se interactúa con la aplicación desde cualquier Form, y se inicializa al iniciar sesión pero es mostrado después de 60 segundos de inactividad; y si el usuario no selecciona ninguna opción, será regresado al menú.



Figura 6) Si el programa detecta inactividad dentro de un Form y no hay respuesta del visitante, el programa automáticamente te regresa al Menú

3.3.Base de Datos

Para nuestro programa optamos por crear una base de datos llamada Registro.accdb con 2 tablas: TRegistro y TNawales.

3.3.1.TRegistro

Este es el registro principal mediante el cual podemos llenar una tabla que permite comparar y observar los usuarios registrados así como su desempeño en las diversas actividades planeadas para ellos. Existen 10 campos que se llenan con forme ellos se registran. Son datos básicos sobre su nombre, correo y fecha de nacimiento junto con un Id que se les otorga al registrarse. Con los resultados de las 4 zonas interactivas que el programa ofrece, se llenan otros 4 campos y al momento de realizar su retroalimentación se obtiene su calificación y su si se envió el correo.

Tabla 1: TRegistro									
Id	Nombre	Correo	Nacimiento	ResultadoLineaTiempo	NawalString	PuntajeEsMaya	Dios	Satisfaccion	CorreoEnviado

3.3.2.TNawales

TNawales es una base de datos auxiliar del formulario Codices [6.2.5]. Esta tabla consta de 4 columnas. Cada una representa la información el calendario maya, haciendo referencia a los días y el Nawal que se relaciona con ese día, su definición, símbolo, y la imagen a la que se refiere en las referencias. Esta base de datos ayuda al programa a tener archivos auxiliares que evitan se llene la memoria del programa con información por omisión de la actividad.

Tabla 2: TNawales				
Id	Nombre	Descripcion	Simbologia	refImage
1	B'atz'	Es el primer día del calendario. Simboliza el comienzo. Principio de la vida, del tiempo infinito y de la unidad.	El tiempo, Polaridad masculina, Polaridad femenina	batz
2	E	Simboliza el camino del destino y de la vida. E es protector de los que transitan los caminos.	Camino, Piedras, Ojo, Boca	_e
3	Aj	Simboliza todo lo que se relaciona al hogar y la familia. La caña representa el triunfo del bien sobre el mal.	Brotes de cañas, Cañas desarrollándose, El horizonte	aj
4	I'x	Es el símbolo y energía de la Madre Tierra. Significa jaguar	Huella de un jaguar, Energía del mundo	ix
5	Tz'ikin	Es el intermediario entre Dios y el hombre. Significa pájaro guardián.	Rostro del águila, Calva del águila	tzikin
6	Ajmaq	Es el día de los pecadores. Es un día para pedir perdón por los pecados y en ese día se habla con los difuntos.	La palabra, El aura	ajmaq
7	No'j	Es la sabiduría y el conocimiento. Es el nawal de la inteligencia.	Cerebro, Médula espinal, sabiduría	noj
8	Tijax	Significa piedra y obsidiana. Es el símbolo del pedernal. Es la energía de la persona que cura en la cultura maya.	Pedernal, Pirámide desde la Tierra, Pirámide desde el cielo	tijax
9	Kawoq	Simboliza la familia, comunidad, grupo, sociedad, reunión, país, continente, todo lo que se agrupa.	Miembros de una familia, Universo	kawoq
10	Ajpu	Es la representación del triunfo de Jun Ajpu e Ixbalamke sobre los señores de Xib'alb'a.	Cara de un cerbatanero, cerbatana	ajpu

11	Imox	Es el nawal del espíritu del agua. Genera movimiento pero también la sensibilidad, esencia de nuestra mente.	Agua, Boca de un vaso, Redondez del vaso	imox
12	Iq'	Es el viento, elemento que rige las ideas y el cambio. Significa aire y aliento de vida.	ventana	iq
13	Aq'ab'al	Significa amanecer, aurora, luz, claridad, nuevo día. Es la renovación de energía.	Sombra, luz	ajmaq
14	K'at	Símbolo del fuego y de la red. Es el que enreda y desenreda la vida.	Red	kat
15	Kan	Significa movimiento. Es el símbolo de la Gran Energía que creó el Universo. Es la Serpiente Emplumada o Q'uq'umatz.	Espalda de una serpiente	kan
16	Kame	Es el símbolo de la muerte. Símbolo de la disolución final de todas las cosas, buenas y malas.	Ojo cerrado	kame
17	Kej	Simboliza los cuatro pilares, es decir los cuatro puntos cardinales. Demuestra fuerza y poder.	Mano cerrada, Puntos cardinales	kej
18	Q'anil	Significa germen, semilla, vida y creación. Simboliza los cuatro colores del maíz existentes en mesoamérica: rojo, negro, blanco y amarillo.	Agujero en la tierra, Semillas	qanel
19	Toj	Significa pago, ofrenda y agradecimiento. Se agradece el sol, el agua, el aire y todo lo que hemos recibido de la Madre Tierra.	Rueda, Eje	toj
20	Tz'i	Es la fidelidad, la autoridad y la ley. El guardián de la ley natural y espiritual.	Tepezcuintle (roedor)	tzi

4.Introduction

Humanity's history has had a huge number of events that had been forging our society in which we live, where the origin is found on the first civilizations, scattered around the planet.

One of those civilizations is the Mayan culture, which was established on the current regions of Chiapas, Yucatán, Guatemala, Belize and Honduras, with a history of about 3000 years.

We chose the National Museum of Anthropology, in specific, the Maya room, due to its fame of being one of the most relevant cultures in the world, mainly because of their treasures and the most complete collection of pre hispanic art in the country, which is organised in 24 thematic rooms, constituting the largest museum of its type in all Latin American regarding dimensions.

5.Description of the improvement

When we visited the museum, we realized how some kids were just taking photos of the descriptions of the pieces or how the people did not appreciate what they really meant to be.

To avoid this from happening, we need to obtain the attention from the visitors, and a perfect solution has to contain something nice and interactive that does not fall into a simple digitalization.

Our solution consists of a program where you can demonstrate your knowledge about the Mayan Culture with formats that are not any similar to a normal test, having also the possibility to assemble with topics that are harder to study without the use of technology; with the final objective of awakening that interest for our huge culture.

6.Description of the program

In this section we will present the classes we used, as well as the forms, the methods used and their graphic interface.

Our program was designed with the purpose of being intuitive, so that anyone could know how to use the program without prior knowledge. Also we tried to implement some things that couldn't been developed before. A clear example is or form Codices [6.2.5]. Instead of just printing the nawal, we decided to add more stuff that can help us understanding the meaning of the Mayan glyphs by marking the areas that represent the string defined on the pressed button. The program also tries to keep the user entertained with different forms that will change depending on a random factor. That can be found in the TimeLine form [6.2.6].

6.1.Classes

"In object-oriented programming, a class is a blueprint for creating objects (a particular data structure), providing initial values for state (member variables or attributes), and implementations of behavior (member functions or methods)". [<https://brilliant.org/wiki/classes-oop/>]

6.1.1.Usuario

It is probably the most important class in the whole program. Every time that a user starts to interact with the app, an object type Usuario will be initialized, which will contain all the necessary information that will be saved in the DB.

Usuario

```
string nombre;  
string correo;  
DateTime nacimiento;  
int resultadoLinea;  
string nawualString;  
Bitmap nawal;  
int puntajeMaya;  
string dios;  
int satisfaccion;  
string descripcion;  
bool archivoEnviado;  
  
Usuario();
```

It only has one constructor and it is empty because we are interested in initializing it with null values, and then this values form the properties will be defined as the Forms are being answered.

It also has a static object of the type `Usuario` named *usuario*, that is available in all the solution. In this object, all the information typed by the user will be used to generate a personalized archive with is results, or as we said before, it will be saved in the DB.

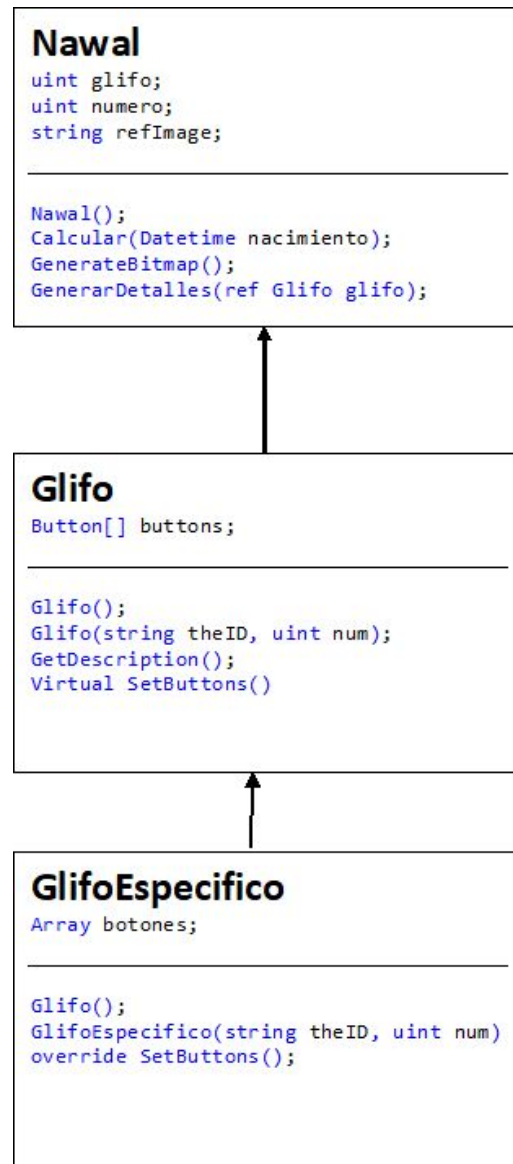
In order to simplify the code due to the many cases where the object *usuario* needed to be initialized again, we made a procedure to solve this problem, which will be executed in any part of the solution in all the required cases; so that the object could be cleaned in order to save the data from another user.

6.1.2.Nawal

Generated class to save the properties from the generated nawal taking the birthday of the user typed in the Form Códices [6.2.5].

The objects type Nawal are always initialized with the help from the empty constructor.

The function *Calcular* serves the purpose of getting the nawal from the user according to his birthdaay, using the following formulas:



$$\begin{aligned}
 numero &= \{(\Delta t \bmod 13 + 4) \bmod 13 \quad \text{si } [(\Delta t \bmod 13 + 4) \bmod 13 \neq 0] \quad 13 \\
 glifo &= (\Delta t \bmod 20)
 \end{aligned}
 \tag{1}$$

Δt being the difference of days of the constant *refDate* that corresponds to the date 2/01/1910 and the parameter date birthday. The equations [1] and [2] were obtained thanks to a proportion calculated with the data from: <http://mcd.gob.gt/calcular-nawal/>.

The property *numero* saves the result from the equation [1] and *glifo* saves the result from the equation [2].

The function *nawal* generates a new image combining the image that refers to the property *numero* and the reference of the property *glifo*.

Finally, the function `GenerarDetalles` instantiate the objects type `Glifo`[6.1.2.1].

6.1.2.1.Glifo

It is a subclass of the class `Nawal` [6.1.2] but with a new property called *buttons* which is an array of buttons. It also has two constructors one is empty and the other one allows the object define the *refImage* and *number* inside its parameters y otro que permite definir los parámetros de *refImage* y *número*. It also has a method named as `SetButtons` is responsible of instantiating a new button inside the buttons array and to have it stored for a further use.

6.1.2.1.1.GlifoEspecifico

It is a subclass of the class `Glifo` [6.1.2.1] which has an array called buttons that will be filled with the data from the column labeled as Simbologia from the table `TRegistro` [6.3.1] from the data base, and from the row that corresponds to its property `ReflImage`. This instruction is called everytime the object gets initialized

It also makes a polymorphism to the `SetButtons` method of `Glyph` [6.1.2.1] so that it doesn't create only a button to identify the Mayan number, but also the buttons of the array buttons.

6.1.3.Pieza

This class is used in the Timeline Form. Each time a user begins to interact with the timeline form, a size 5 piece type arrangement is instance, which will contain the dates that were assigned to the pieces, the images of the pieces and the number of hits.

Pieza

```
int tiempo;  
Bitmap pic;  
bool correct;  
uint orden;
```

```
Pieza();
```

```
Pieza(int tiempo, Bitmap pic);
```

It has an empty constructor since we are interested in initializing it with null values and defining the values of the properties as the forms are filled out.

It also has a builder with overload, which will allow us to save the dates and images assigned to those dates.

6.1.4.Dios

Class God has the objective of saving the properties from the gods used in the Form What Mayan god would you be?[6.2.8].

Dios

```
string nombre;  
string descripción;  
Bitmap imagen;  
uint puntos;
```

```
public Dios()  
public Dios(string nom, string desc, Bitmap imag)
```

The class has an empty constructor and a constructor with parameters in order to initialize the values.

The class contains a data array of 5 different Gods with their own description and image

6.1.5.Maya

Maya

```
int Img;  
Random random;  
int score;  
bool bip;  
int contador;  
  
int Picture();
```

A single class “Maya” was used, this class contains an ArrayList, which ensures of not repeating images in the Form; 3 attributes that let the program keep the update, help to limit the attempts and to make the buttons blink; 2 constructors that help to initialize the class and to help the blinking procedure.

6.2.Forms

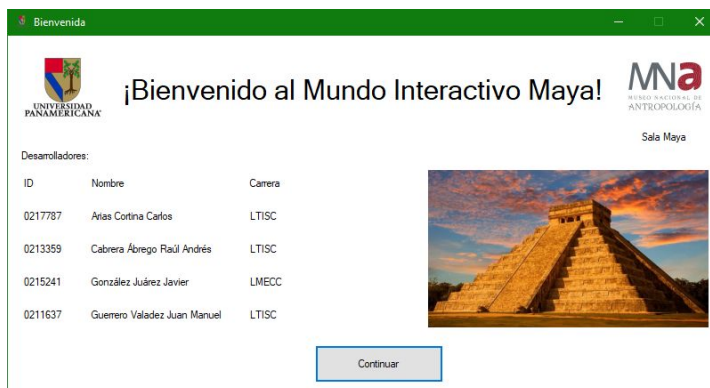
The forms are the base unit of your application, it is essential that you give some thought to their function and design. A form is ultimately a blank slate that you, as a developer, enhance with controls to create a user interface and with code to manipulate data. To that end, Visual Studio provides you with an integrated development environment (IDE) to aid in writing code, as well as a rich control set written with the .NET Framework. By complementing the functionality of these controls with your code, you can easily and quickly develop the solutions you need.

Recovered from :

<https://docs.microsoft.com/en-us/dotnet/framework/winforms/>

6.2.1.Welcome form

This is the first Form that deploys when you open the program. It presents the title of the program, the names and the information for both the developers and the application.



Bienvenido

¡Bienvenido al Mundo Interactivo Maya!

UNIVERSIDAD PANAMERICANA

MNA MUSEO NACIONAL DE ANTROPOLOGÍA

Sala Maya

Desarrolladores:

ID	Nombre	Carrera
0217787	Arias Cortina Carlos	LTISC
0213359	Cabrera Ábrego Raúl Andrés	LTISC
0215241	González Juárez Javier	LMECC
0211637	Guerrero Valadez Juan Manuel	LTISC

Continuar

Figure 1: Welcome Form

The form has a single button located at the bottom part with the name *Continuar*. When you click on it, it will start a function that will hide the

Welcome form and will show the LogInForm that will be specified right below.

6.2.2.Log in

This is the form that is shown to the administrators to be able to access the main menu. For this it is necessary to enter the username and password correctly. In case the user enters a wrong username or password more than 3 times, the program will be closed for security reasons.



Figure 2: A) login button. B) Exit button

The form is defined in the LogIn class, which contains a private attribute of whole type called nErrors. Pressing button 1 will verify that nErrors is less than 3. If it is higher than or equal to three, the program will be closed.

Afterwards, all the values will be compared between two arrays that contain all the user names and passwords and the values entered by the user. If the entered data is correct, the form will be hidden and then the main menu will be displayed. Otherwise, one will be added to the nErrors counter.

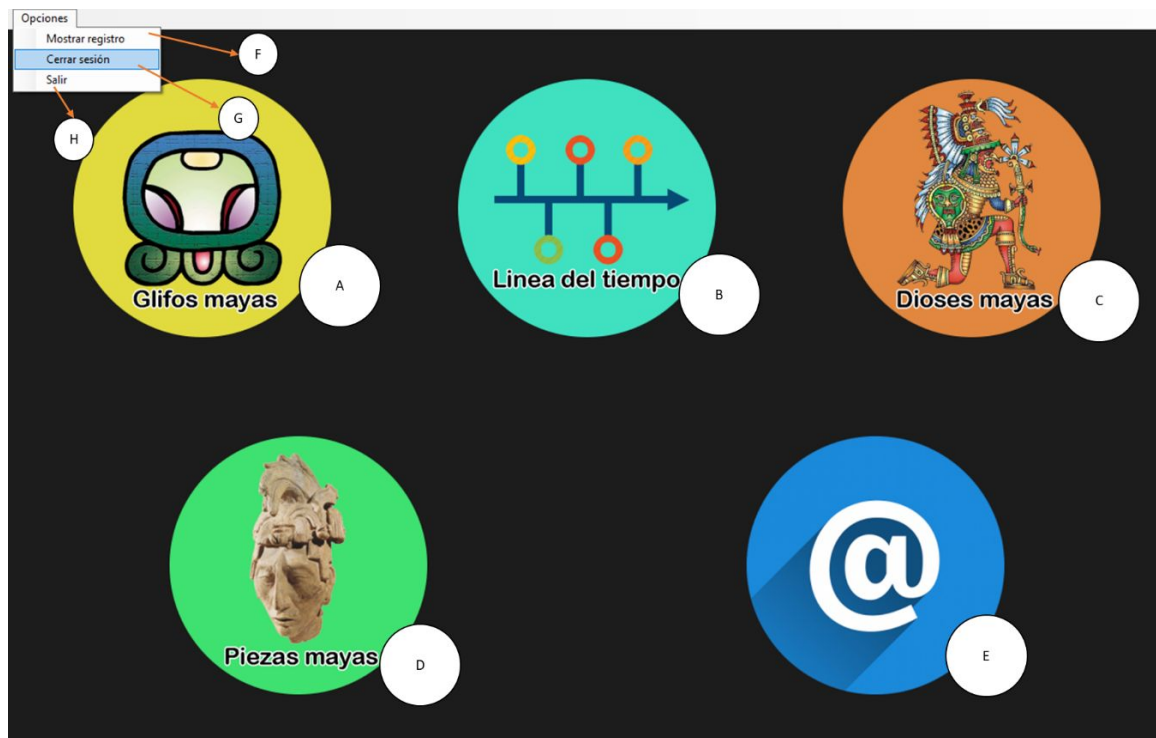
6.2.3.Main menu

This is the form used to access the rest of the interactive forms. There is a MenuStrip *Opciones* which deploys the options *Mostrar registro*, *Cerrar sesión* y *Salir*; were if you click on them, the Form *Ingresar credenciales* [6.2.4] will be shown, avoiding that a user without permissions could visualize these forms.

The control *Mostrar registro* from the MenuStrip *Opciones* allows the access to the admins to the Form *Registro* [6.2.13] that shows the table *TRegistro* from the DB.

The control *Cerrar sesión* hides the Main Menu and shows the form *Iniciar sesión* [6.2.2]. Finally, the control *Salir* closes the program.

We must say that this and the form *LogIn* are the only ones that allow the closage of the program.



The form has an array of forms named *subForms*, that contains all the interactive forms: *Códices*[6.2.5], *TimeLine* [6.2.6], *What mayan god would you be?*[6.2.8], *Mayan or not?*[6.2.10], and *Register*[6.2.12].

Clicking in any circular button will evoke the function *btnAbrirForm* that will instantiate and initialize the form related to the clicked button and will also hide the Menu.

All the forms opened from the Menu have a button to return to the Menu.

6.2.4. Enter credentials

This form is very basic but has a great relevance because it allows us to access to the DB or to log off.

It only has one if and an else, that will verify if the data typed is correct, if it is not, the labels will be marked in red until the data is correct or if the user cancels the button.

This verification is possible due to the function `VallidarUsuario` in the `Login` class [6.2.2], and it is on the partial class `IngresarCredenciales` where we evoke this function.



Figure 5) The form warns the visitor if the data entered

6.2.5. Codices

It is a form with the ability to show the user his Mayan nawal, which changes depending on the month, day and year of birth of the user. Nawales mean energy, spirit or strength of beings and elements of nature. According to the Mayan worldview, they are symbols that represent and connect each person with the ecosystem.



Illustration 2A) Enter date of birth B) Return Menu C) Calculate nawal D) Nawal's name E) Description of the nawal F) Interactive button pressed H) Symbol highlighted according to button F

The form, besides calculating the nawal, generates the image that corresponds to the nawal, a brief description and some interactive buttons that show the meaning of each part of the nawal.

Pressing the Calculate button (C) will invoke the function `button1_Click`, which will save the date of birth of the visitor in the user object of type `User` [3.1.1] in its `birth` property, a new object of type `Glyph` will be generated [3.1 .2.1] and this will invoke its function `GenerateDetails` to be able to show in the form the nawal of the user.

If there is an occasion for the user to enter on his date an unexpected character such as a letter, a false date (such as January 2, 1910) or a date that is not real, the exception will be handled warning the user that his date is not valid.

6.2.6. Time Line

This is the form that is displayed by pressing button B of the Menu form (Image x). The purpose of this form is to interact with the user by means of a didactic game in which Mayan pieces are sorted chronologically. In this way, the user will be able to better visualize the chronology of the Mayan civilization.

The operation of the Form can be explained in 4 simple steps:



Figure 4: Select Image

1) Select Image:

Point the mouse arrow at the image you want to move, once there hold down the right click to select the image as shown in (Figure 4).



Figure 5: Drag Image

2) Drag image:

Direct the mouse arrow to the space where you want to put the image, at the same time hold down the left click as it is in (figure 5).



Figure 6: Drop Image

Stop giving the left click and the selected image will be shown in the space as it is in (figure 6).



Figure 7: A) Return or Qualify

4) Qualify or return:

If you decide to give a left click on the button A of (figure 7), you will return to the menu form.

If you decide to give a left click on the button B of (figure 7), the results form will be shown, in which you will be informed of the successes obtained by the user as it is in (figure 8).

6.2.7.Results

This is the first form that is shown when you left click on button B of (figure 7). It serves to show the results obtained by the user at the end of the Timeline game.

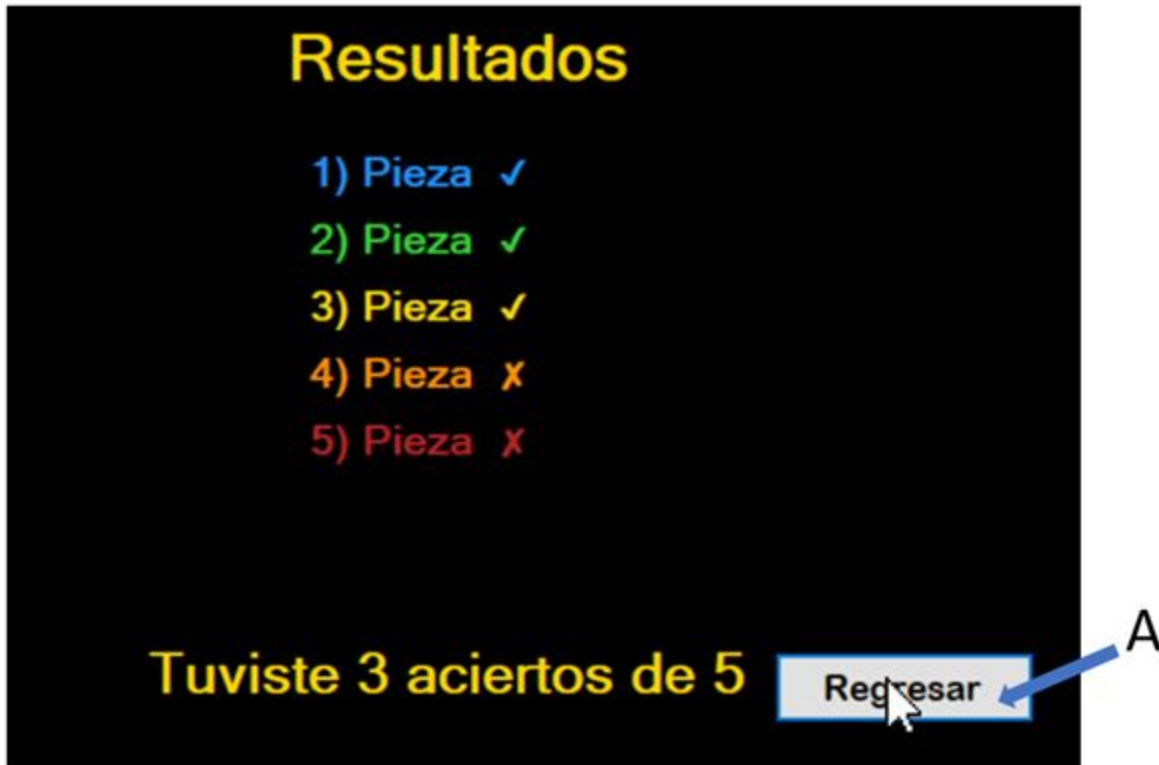


Figure 8: A) Return button.

When you left click on the button A in (figure 8), you will return to the menu form.

6.2.8.What Mayan god would you be?

The form is a test where the user chooses the characteristics that are more related to him. Depending of his selection, a god will be shown according to those same characteristics. This will help the user to gain knowledge about the Mayan gods and for the awakening of the interest for this culture.

The Form QuizDios has the objective to calculate the most similar God for the user according to his answers; the program consists of 6 simple questions where each option has a meaning in order to settle the God, once it is done and shown with his image and description, it will be saved in the object Usuario for savement purposes on the DB.

¿QUÉ DIOS MAYA SERÍAS? Regresar

¿Con qué color te identificas?

☐ ☐ ☐ ☐

¿Qué frase célebre te gusta más?

- "Preferiría la paz más injusta a la más justa de las guerras"
- "Amad el arte, entre todas las mentiras es la menos mentirosa"
- "Ganando como siempre"

¿Con qué palabra te describirías?

- Viajero
- Ambicioso
- Creativo

☐ Leer un libro
☐ Salir a pasear
☐ No tengo tiempo para eso

Si fueras un animal, serías...

- Serpiente
- Búho
- Jaguar
- Colibrí

¿Que superpoder te gustaría poseer?

- Omnisciencia
- Super fuerza
- Factor de curación
- Inmortalidad

Calcular

By selecting the Button Calcular, a foreach loop will be activated where we would be able to access to all the GroupBoxes in the Form in order to know how many points should be assigned to each Dios according to the value in i from the property Tag from every RadioButton.

The boolean variable will ensure us that the RadioButtons are selected, and if the user leaves a blank answer, a MessageBox will be shown for warning purposes; this leads us to a Form without errors and usage of exceptions.

6.2.9.MessageBox of gods


ERES:

lbName

lbDesc

Regresar

ERES:



Itzamná

Dios del sol y la sabiduría. También señor del cielo, el día y la noche.

Regresar

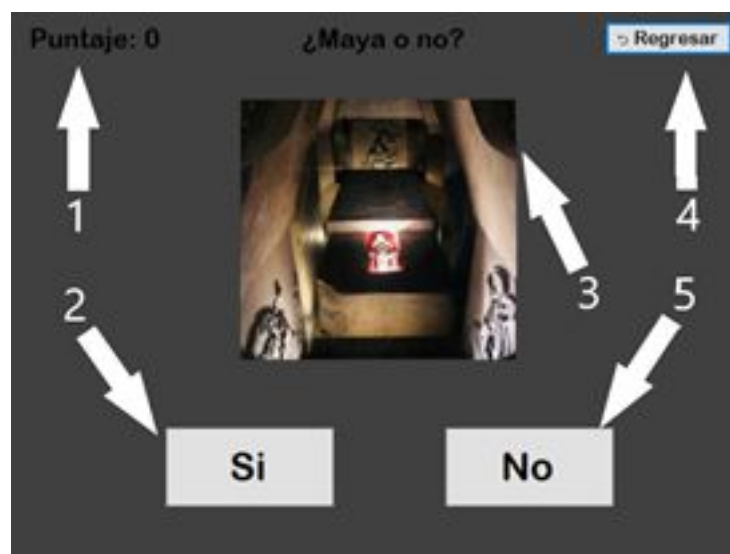
Figure 1) MessageBoxShow Form before being evoked.

Figure 2) MessageBoxForm showing the image, description and name of the assigned God.

6.2.10.Is it Maya or not?

This C# Form's main objective is to have an interactive activity which should let the user have a review of the learning through the museum with a digital game that could help the user learn more about crafts Mayan civilization made. Using images, a brief quiz is made about crafts belonging to the Mayan civilization.

The Form's design is pretty simple, but functional. Our target is letting the user acknowledge the way the Form works without letting him produce a mistake in it.



- | | | |
|-------------------|---------------|-----------|
| 1. Score | 2. Button Yes | 3. Images |
| 4. Button Go Back | 5. Button No | |

The way the Form works is simple. There are only 3 buttons and the user has 5 attempts to gain as much points as possible. The user must recognize which crafts are Mayan and which ones are from different ones. That's when the buttons are used. If the answer is right, then it will blink green and

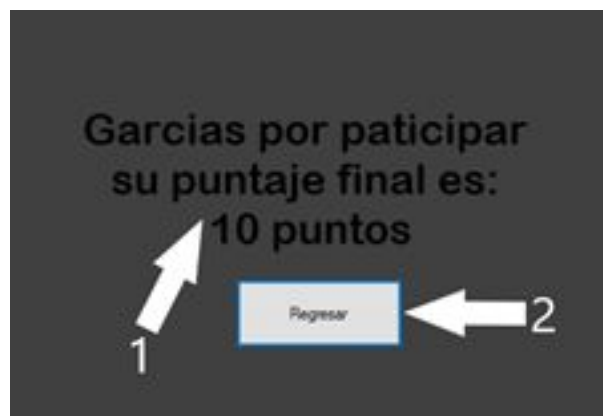
if it is wrong it will blink red. When the answer is correct, the score gains 2 points and when the answer is wrong nothing happens. After the 5 attempts happened the final score appears having a maximum score of 10 points.

On this Form there's no need to use exceptions as the user's interaction with it is not able to break the program. That's why using this form is easy.

There's just one method implemented to assure each time the user plays, the images displayed in the Form don't repeat. This procedure is inside the class Maya [6.1.5.] and by using an ArrayList we prevent the images from repeating.

The form usage is user friendly, as explained before, the objective is to gain as much points as the user can. When the user reaches the 5 attempts the final score shows on the top left screen and a new Form displays which shows the final score and saves the result to the class Usuario [6.1.1.] and goes back to the Main Menu [6.2.3.].

6.2.11. MessageBox Maya



1. Label Score
2. Button Go Back

This Form consists of one label and one button, the function of this Form is to show the final score of the user and saving the final score to the Usuario class [6.1.1]. Even though this Form is just to show the final score. It helps resetting the ArrayList from the Maya class [6.1.5] to the main setup. Once the user clicks the button of go back, the ¿Maya o no? form [6.2.10] shows up in case the user wants to play again or wants to go back to the main menu [6.2.3].

6.2.12.Register

This is another form of great importance, because it is in this form that the user's information is stored in the database. It also provides information of great importance to the museum, the assessment of users about their experience in the application.

¡Envía tus resultados por correo electrónico!

*Califica la app:

Nombre: Juan

*Correo electrónico: 0211637@up.edu.mx

Enviar

Regresar

Illustration 3A) GroupBox with radioButtons for reviewing with 10, 5 or 0. B)Data from the user needed in order to send the email C)Evokes the method in order to create the archive D)Go back to menu

It also has a unique function, which is that it allows the visitor to send the information of their session to their email in a * .rtf file (Annex 2), which means a formatted text file to generate a more pleasant visual experience for when the user reads his results, and also includes the image of his nawal, which is not possible to do in a * .txt file.

This process happens when clicking button Enviar © from the Form. The button uses the event button1_Click. The File generated using the structured class GeneradorDeArchivos which contains constants, make the format richer and a function GenerarArchivo which joins the constants with the object values. When the process finishes it saves the text on the file.

Once the file is generated, with the help of the *System.Net.Mail* library, the program will try to send an email according to the information given by the user. In case that there is no connection to the internet or the file won't sent for a problem related to the program, it will notify the user that it couldn't sent the file but also telling him that it will be sent later on. And this action is

planned to be executed manually by an administrator when he finds a column in *CorreoEnviado* from *TRegistro* with a false value. If the mail was sent, the object *usuario* will be saved in the database with the value of *CorreoEnviado* as true.

In the case that the email could not be sent because the user did not typed the correct information or because he didn't review the app, he will be notified in order to corroborate the data and last but not least, the instruction send to email will be aborted.

6.2.13.Registry

It's an administrator Form only, the access way is through the credential get into Form [6.2.4]. The Form structure is when opening the form a query in charge of reading table *TRegistro*[6.3.1] from the database is executed and the database is shown on a datagridview.

Id	Nombre	Correo	Nacimiento	Linea del Tiempo	Nawal	¿Es Maya?	Dios	Satisfacción	CorreoEnviado
26	Juan	0211637@up.edu.mx	21/12/1999	6	13 B'atz'	10	Kukulcán	10	<input type="checkbox"/>
27	Juan	0211637@up.edu.mx	23/12/1999	-1	2 Aj	-1		10	<input type="checkbox"/>
28	Giancarlo	0215420@up.edu.mx	10/12/1999	6	2 Ajpu	10	Chaac	10	<input checked="" type="checkbox"/>
29	Juan Guerrero	0211637@up.edu.mx	21/12/1999	10	13 B'atz'	8	Kukulcán	10	<input checked="" type="checkbox"/>
30	Jose Lopez	juan.guerrero.v99@gmail.com	01/01/2001	-1		-1	Kauil	5	<input checked="" type="checkbox"/>
31	Javier	salamaya.upsolutions@gmail.com	01/01/2001	-1		8		0	<input checked="" type="checkbox"/>
32	Joven	joava@email.com	25/02/2001	-1	3 Aq'ab'al	-1		0	<input checked="" type="checkbox"/>
33	fas	ga@afg.dgs	01/01/2001	-1		-1		10	<input type="checkbox"/>
34	Antiguo	0211637@up.edu.mx	01/01/2001	-1		0	Kukulcán	10	<input checked="" type="checkbox"/>
35	Jan	0211637@fa.as	01/01/1000	6		-1		10	<input checked="" type="checkbox"/>
36	fgd	ga@gsd	01/01/1000	2		-1		10	<input checked="" type="checkbox"/>
37	Raúl Cabrera	rumamba@hotmail.com	01/03/2000	6	6 Iq'	8	Kauil	10	<input checked="" type="checkbox"/>
38	Juan	0211637@up.edu.mx	03/03/1234	-1		0		10	<input checked="" type="checkbox"/>
39	0211637	0211637@up.edu.mx		-1		-1	Kauil	10	<input checked="" type="checkbox"/>

A → Guardar cambios
→ B Regresar

It also has an update button in charge of updating the *CorreoEnviado* column basing on the program's data. Reminding that the only column being modified is the *CorreoEnviado* one. The objective is to show the users that have been sent an email with their results as it's described in section 6.2.12.

As well, the Form has a button to close the Form and go back to Main Menu [6.2.3]

6.2.14.TimerDialog

This form gives the program a special detail, it helps to detect if the visitor left the program without sending his information; so it is necessary to reset the object user [6.1.1] and return to the main menu with the objective of making the program more attractive for the new visitors that want to use the program.

The Timer resets each time a Form is modified. The timer starts when a user logs in, but it only appears when there's 60 seconds of inactivity and if the user doesn't select and option, it will automatically return to the Menu.



Figura 6) If the program detects there's no changes in certain time lapse. It will automatically close and return to the Main Menu.

6.3.Data Base

The program receives information from a database Registro.accdb with 2 tables: TRegistro y TNawales

6.3.1.TRegistro

Tabla 1: TRegistro									
Id	Nombre	Correo	Nacimiento	ResultadoLineaTiempo	NawalString	PuntajeEsMaya	Dios	Satisfaccion	CorreoEnviado

This is the main register, with it the program is able to fill a database that shows each register made throughout time and each users' on the activities inside the program. 10 fields are fill with the information the user provides as it uses the program. First data collected is name, email, and birthdate in addition an Id is provided to the user. During the different activities of the program, each result is saved to the same database, there are 4 different activities form the user to try. For the last, when the user is about to exit, it will fill a satisfaction quiz and will keep update of the email sending status.

6.3.2.TNawales

TNawales is an auxiliary database of the Codices Form [6.2.5] this database has 4 columns. Each of them contain different information of the mayan calendar as a reference to the Nawal day, definition, symbol, and related image. This helps the program not to be full of default information that can be store on different auxiliary files.

Tabla 2: TNawales				
Id	Nombre	Descripcion	Simbologia	refImage
1	B'atz'	Es el primer día del calendario. Simboliza el comienzo. Principio de la vida, del tiempo infinito y de la unidad.	El tiempo, Polaridad masculina, Polaridad femenina	batz
2	E	Simboliza el camino del destino y de la vida. E es protector de los que transitan los caminos.	Camino, Piedras, Ojo, Boca	_e
3	Aj	Simboliza todo lo que se relaciona al hogar y la familia. La caña representa el triunfo del bien sobre el mal.	Brotes de cañas, Cañas desarrollándose, El horizonte	aj
4	I'x	Es el símbolo y energía de la Madre Tierra. Significa jaguar	Huella de un jaguar, Energía del mundo	ix
5	Tz'ikin	Es el intermediario entre Dios y el hombre. Significa pájaro guardián.	Rostro del águila, Calva del águila	tzikin
6	Ajmaq	Es el día de los pecadores. Es un día para pedir perdón por los pecados y en ese día se habla con los difuntos.	La palabra, El aura	ajmaq
7	No'j	Es la sabiduría y el conocimiento. Es el nawal de la inteligencia.	Cerebro, Médula espinal, sabiduría	noj
8	Tijax	Significa piedra y obsidiana. Es el símbolo del pedernal. Es la energía de la persona que cura en la cultura maya.	Pedernal, Pirámide desde la Tierra, Pirámide desde el cielo	tijax
9	Kawoq	Simboliza la familia, comunidad, grupo, sociedad, reunión, país, continente, todo lo que se agrupa.	Miembros de una familia, Universo	kawoq
10	Ajpu	Es la representación del triunfo de Jun Ajpu e Ixbalamke sobre los señores de Xib'alb'a.	Cara de un cerbatanero, cerbatana	ajpu
11	Imox	Es el nawal del espíritu del agua. Genera movimiento pero también la sensibilidad, esencia de nuestra mente.	Agua, Boca de un vaso, Redondez del vaso	imox
12	Iq'	Es el viento, elemento que rige las ideas y el cambio. Significa aire y aliento de vida.	ventana	iq
13	Aq'ab'al	Significa amanecer, aurora, luz, claridad, nuevo día. Es la renovación de energía.	Sombra, luz	ajmaq
14	K'at	Símbolo del fuego y de la red. Es el que enreda y desenreda la vida.	Red	kat
15	Kan	Significa movimiento. Es el símbolo de la Gran Energía que creó el Universo. Es la Serpiente Emplumada o Q'uq'umatz.	Espalda de una serpiente	kan

16	Kame	Es el símbolo de la muerte. Símbolo de la disolución final de todas las cosas, buenas y malas.	Ojo cerrado	kame
17	Kej	Simboliza los cuatro pilares, es decir los cuatro puntos cardinales. Demuestra fuerza y poder.	Mano cerrada, Puntos cardinales	kej
18	Q'anil	Significa germen, semilla, vida y creación. Simboliza los cuatro colores del maíz existentes en mesoamérica: rojo, negro, blanco y amarillo.	Agujero en la tierra, Semillas	qanel
19	Toj	Significa pago, ofrenda y agradecimiento. Se agradece el sol, el agua, el aire y todo lo que hemos recibido de la Madre Tierra.	Rueda, Eje	toj
20	Tz'i	Es la fidelidad, la autoridad y la ley. El guardián de la ley natural y espiritual.	Tepezcuintle (roedor)	tzi

Anexo 1: Código fuente

```
/* Arias Cortina Carlos    0217787
 * Cabrera Ábrego Raúl Andres  0213359
 * González Juárez Javier 0215241
 * Guerrero Valadez Juan Manuel 0211637
 * nombre id materia equipo
 * Programación Orientada a Objetos
 * Sala Maya
 */
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AppSalaMaya
{
    static class Program
    {
        /// <summary>
        /// Punto de entrada principal para la aplicación.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Usuario.GenerarUsuario();
            Application.Run(new Scripts.Bienvenida());
        }
    }
}

/* Arias Cortina Carlos    0217787
 * Cabrera Ábrego Raúl Andres  0213359
 * González Juárez Javier 0215241
 * Guerrero Valadez Juan Manuel 0211637
 * nombre id materia equipo
 * Programación Orientada a Objetos
 * Sala Maya
 */
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```

namespace AppSalaMaya
{
    class Usuario
    {
        //Atributos
        private string nombre;
        private string correo; //QUITAR
        private DateTime nacimiento;
        private int resultadoLinea;
        private string nawualString;
        private Bitmap nawal;
        private int puntajeMaya;
        private string dios;
        private int satisfaccion;
        private string descripcion; //Solo para el archivo para el usuario. No
agregar a base de datos
        private bool archivoEnviado; //en caso de que no se pueda enviar el correo
automaticamente
        //Propiedades
        public string Nombre { get { return nombre; } set { nombre = value; } }
        public string Correo { get { return correo; } set { correo = value; } }
        public DateTime Nacimiento { get { return nacimiento; } set { nacimiento =
value; } }
        public int ResultadoLinea { get { return resultadoLinea; } set {
resultadoLinea = value; } }
        public string NawualString { get { return nawualString; } set { nawualString
= value; } }
        public Bitmap Nawal { get { return nawal; } set { nawal = value; } }
        public int PuntajeMaya { get { return puntajeMaya; } set { puntajeMaya =
value; } }
        public string Dios { get { return dios; } set { dios = value; } }
        public int Satisfaccion { get { return satisfaccion; } set { satisfaccion =
value; } }
        public string Descripcion { get { return descripcion; } set { descripcion =
value; } }
        public bool ArchivoEnviado { get { return archivoEnviado; } set {
archivoEnviado = value; } }
        public Usuario() //constructor para instanciar un nuevo usuario
        {
            Nombre = "";
            Correo = "";
            Nacimiento = new DateTime(1001, 1, 1); //para validar en caso de que
no haya ingresado una fecha de nacimiento;
            ResultadoLinea = -1;
            NawualString = "";
            Nawal = null;
            PuntajeMaya = -1;
            Dios = "";
            Satisfaccion = -1;
        }
    }
}

```

```

    }

    //Generar un nuevo usuario
    public static Usuario usuario; //variable usuario accesible para todo el
ensamblado
    public static void GenerarUsuario()
    {
        usuario = new Usuario();
    } //Metodo para inicializar la variable
    }

/* Arias Cortina Carlos      0217787
* Cabrera Ábrego Raúl Andres  0213359
* González Juárez Javier 0215241
* Guerrero Valadez Juan Manuel 0211637
* nombre id materia equipo
* Programación Orientada a Objetos
* Sala Maya
*/
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.OleDb;
using System.Windows.Forms;

namespace AppSalaMaya
{
    class Conexion
    {
        public static OleDbConnection conexion = null;

        public static OleDbConnection getCon()
        {
            if (conexion == null)
            {
                try
                {
                    OleDbConnectionStringBuilder csb = new
OleDbConnectionStringBuilder();
                    csb.Provider = "Microsoft.ACE.OLEDB.12.0"; //Microsoft Provider
                    csb.DataSource = "Registro.accdb"; //nombre de nuestra base de
datos

                    conexion = new OleDbConnection(csb.ToString());
                    conexion.Open();
                }
                catch (OleDbException e)
                {

```

```

        MessageBox.Show("Error de la base de datos: " + e.Message);
        if (System.Windows.Forms.Application.MessageLoop)
        {
            System.Windows.Forms.Application.Exit();
        }
        else
        {
            System.Environment.Exit(1);
        }
    }
}
return conexion;
}
public static void Ejecutar(String sql)//Acción
{
    OleDbCommand cmd = new OleDbCommand(sql, getCon());
    cmd.ExecuteReader();
}
public static OleDbDataReader Leer(String sql) //Lectura
{
    OleDbCommand cmd = new OleDbCommand(sql, getCon());
    return cmd.ExecuteReader();
}
public static OleDbDataReader reader;
public static void callReader(String sql)
{
    reader = Leer(sql);
}
}
class Queries
{
    public static string INSERT(string tabla, string campos, dynamic[] valores)
    {
        string salida = String.Format("INSERT INTO {0} {1} VALUES (", tabla,
campos);
        for(int i = 0; i < valores.Length; i++)
        {
            if (valores[i] is string s) salida += String.Format(@"{0}""",
valores[i]);
            else salida += valores[i];
            if (i == valores.Length -1) salida += ");";
            else salida += ", ";
        }
        return salida;
    }
    public static string UPDATE(string tabla, string campo, dynamic valor,
string condicion)
    {
        string salida = String.Format("UPDATE {0} SET ", tabla);

```

```

        if (valor is string s) valor = String.Format(@"\"{0}\"", valor);
        salida += String.Format("{0} = {1} WHERE {2}", campo, valor, condicion);
        return salida + ";";
    }

    public static string UPDATE(string tabla, string[] campos, dynamic[]
valores, string condicion)
    {
        string salida = String.Format("UPDATE {0} SET ", tabla);
        for (int i = 0; i < valores.Length; i++)
        {
            if (valores[i] is string s) valores[i] = String.Format(@"\"{0}\"",
valores[i]);
            salida += String.Format("{0} = {1}", campos[i], valores[i]);
            if (i != valores.Length - 1) salida += ", ";
        }
        salida += String.Format(" WHERE {0};", condicion);
        return salida;
    }

    public static string SELECT(string tabla, string campos, string condicion)
    {
        return String.Format("SELECT {0} FROM {1} WHERE {2};", campos, tabla,
condicion);
    }

    public static string SELECT(string tabla, string campos)
    {
        return String.Format("SELECT {0} FROM {1};", campos, tabla);
    }
}

```

```

/* Arias Cortina Carlos      0217787
 * Cabrera Ábrego Raúl Andres 0213359
 * González Juárez Javier 0215241
 * Guerrero Valadez Juan Manuel 0211637
 * nombre id materia equipo
 * Programación Orientada a Objetos
 * Sala Maya
 */

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace AppSalaMaya.Scripts
{

```



```

        public partial class Bienvenida : Form
        {
            public Bienvenida()
            {
                InitializeComponent();
            }

            private void button1_Click(object sender, EventArgs e)
            {
                LogIn logIn = new LogIn();
                logIn.Show();
                this.Hide();
            }
        }
    }
}

```

```

/* Arias Cortina Carlos    0217787
* Cabrera Ábrego Raúl Andres  0213359
* González Juárez Javier 0215241
* Guerrero Valadez Juan Manuel 0211637
* nombre id materia equipo
* Programación Orientada a Objetos
* Sala Maya
*/

```

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
namespace AppSalaMaya.Scripts
{
    public partial class Codices : Form
    {
        public static Codices thisForm;
        public Codices()

```

```

{
    InitializeComponent();
    thisForm = this;
}
private void button1_Click(object sender, EventArgs e)
{
    TimerDialog.segundos = 0;

    try
    {
        Usuario.usuario.Nacimiento = new DateTime(int.Parse(tbAno.Text),
int.Parse(tbMes.Text), int.Parse(tbDia.Text));
        Nawal nawal = new Nawal();
        Usuario.usuario.NawualString =
nawal.Calcular(Usuario.usuario.Nacimiento);
        Usuario.usuario.Nawal = nawal.GenerateBitmap();

        Glifo glifo = new Glifo(nawal.ReflImage, nawal.Numero);
        nawal.GenerarDetalles(ref glifo);
        flowLayoutPanel1.Controls.Clear();
        foreach (Button btn in glifo.Buttons)
        {
            flowLayoutPanel1.Controls.Add(btn);
        }
        pbFondo.BackgroundImage = Usuario.usuario.Nawal;
        lbNahual.Text = Usuario.usuario.NawualString;
        Usuario.usuario.Descripcion = glifo.GetDescripcion();
        tbExplicacion.Text = Usuario.usuario.Descripcion;

    }
    catch (FormatException)
    {
        MessageBox.Show("Formato incorrecto. Únicamente use
números");
    }
    catch (ArgumentOutOfRangeException)
    {
        MessageBox.Show("No se reconoció la fecha. Revise los
datos.");
    }
}

```

```

    }
    catch(FueraDeRango ex)
    {
        MessageBox.Show(ex.Message);
    }

    catch(Exception)
    {
        MessageBox.Show("Lo sentimos, esto no debió de haber
ocurrido.", "Error interno", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

}

public async void ResaltarGlifo(object sender, EventArgs e)
{
    TimerDialog.segundos = 0;
    Color aux = ((Button)sender).BackColor;
    System.Threading.Thread.Sleep(50); //Para evitar que se ejecute 2
veces la misma tarea;
    ((Button)sender).BackColor = Color.FromArgb(50, aux.R, aux.G,
aux.B);
    pbGlifo.Image =
(Image)AppSalaMaya.Properties.Resources.ResourceManager.GetObject((
string)((Button)sender).Tag);
    await Task.Delay(1000);
    pbGlifo.Image = null;
    ((Button)sender).BackColor = aux;
}

private void tbDia_Enter(object sender, EventArgs e)
{
    TimerDialog.segundos = 0;
    if (((TextBox)sender).Text == (string)((TextBox)sender).Tag)
        ((TextBox)sender).Text = "";
    ((TextBox)sender).ForeColor = Color.Black;
}

private void tbDia_Leave(object sender, EventArgs e)
{

```

```

TimerDialog.segundos = 0;
if (((TextBox)sender).Text == "")
{
    ((TextBox)sender).Text = (string)((TextBox)sender).Tag;
    ((TextBox)sender).ForeColor =
System.Drawing.SystemColors.ControlDark;
}

}

private void button2_Click(object sender, EventArgs e)
{
    TimerDialog.segundos = 0;
    this.Hide();
    Scripts.Menu.menu.Show();
}

private void tbDia_TextChanged(object sender, EventArgs e)
{
    TimerDialog.segundos = 0;
}
}

class Nawal
{
    public static DateTime refDate = new DateTime(1910, 1, 2); //Fecha
de referencia. http://mcd.gob.gt/calcular-nawal/
    //Atributos
    private uint glifo;
    private uint numero;
    private string reflImage;

    public uint Numero
    {
        get { return numero; }
        set { numero = value; }
    }
    public uint Glifo

```

```

    {
        get { return glifo; }
        set { glifo = value; }
    }
    public string ReflImage
    {
        get { return reflImage; }
        set { reflImage = value; }
    }
    public Nawal() //Constructor vacío
    {
        Numero = 0;
        Glifo = 0;
        ReflImage = "";
    }

    //Metodos
    public string Calcular(DateTime nacimiento) //Obtiene el nawal de
    acuerdo a la fecha de entrada y devuelve el nombre del nawual
    {
        int difference;
        string nombre = "";
        difference = (int)nacimiento.Subtract(refDate).TotalDays;
        if (nacimiento.CompareTo(refDate) < 0) throw new FueraDeRango();
        Numero = (uint)((difference % 13) + 4) % 13; //Calcular numero
        if (Numero == 0) Numero = 13; //En caso de que MOD sea 0, pasar a
13
        Glifo = (uint)(difference % 20); //Obtener glifo
        Conexion.callReader(Queries.SELECT("TNawales", "Nombre", "ID = " +
        (Glifo + 1)));
        while (Conexion.reader.Read())
        {
            nombre = Numero + " " + Conexion.reader.GetString(0);
        }
        return nombre;
    }
    public Bitmap GenerateBitmap() //Devuelve un nuevo bitmap que
    junta el número maya con el glifo
    {

```

```

        Bitmap dia;
        Conexion.callReader(Queries.SELECT("TNawales", "reflImage", "ID = "
+ (Glifo + 1))));
        while (Conexion.reader.Read())
        {
            reflImage = Conexion.reader.GetString(0);
        }
        dia = new
Bitmap((Image)AppSalaMaya.Properties.Resources.ResourceManager.Get
Object(reflImage));
        Bitmap mes = new
Bitmap((Image)AppSalaMaya.Properties.Resources.ResourceManager.Get
Object("num" + Numero));
        Graphics imagen = Graphics.FromImage(dia);
        imagen.InterpolationMode =
System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;
        imagen.DrawImage(mes, new Point(0, 4));
        imagen.Save();
        return dia;
    }

    public void GenerarDetalles(ref Glifo glifo)
    {
        glifo = new GlifoEspecifico(reflImage, Numero);
        glifo.SetButtons();
        //descripcion = glifo.Descripcion;
    }
}

class Glifo : Nawal
{
    //Atributos
    private Button[] buttons;
    //Propiedades
    public Button[] Buttons { get { return buttons; } set { buttons = value; } }
    //Constructores
    public Glifo() //Vacío
    {
        Numero = 0;
    }
}

```

```

ReflImage = "";
Buttons = null;
}
public Glifo(string theID, uint num)//Sobrecarga del constructor
{
ReflImage = theID;
Numero = num;

}
public string GetDescripcion()
{
string descripcion = "";
Conexion.callReader((Queries.SELECT("TNawales", "Descripcion",
"reflImage = \"'\" + ReflImage + \"'\")));
while (Conexion.reader.Read())
{
descripcion = Conexion.reader.GetString(0);
}
return descripcion;
}
public virtual void SetButtons() //Generar un único botón unicamente
para identificar números mayas
{
if (Buttons == null) Buttons = new Button[1];
Buttons[0] = new Button
{
Text = "Número " + Numero,
Tag = "resaltar", //Para identificar al boton
BackColor = Color.Red,
ForeColor = Color.Black,
Font = new Font("Microsoft Sans Serif", 15.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)0)),
Margin = new Padding(0, 3, 0, 3),
Size = new Size(250, 60)

};
Buttons[0].Click += new
System.EventHandler(Codices.thisForm.ResaltarGlifo);

```

```

    }
    }
    class GlifoEspecifico : Glifo
    {
        private Array botones; //Referencia: http://www.sabiduriamayamaya.org
        static Color[] color = { Color.LightBlue, Color.LightGreen, Color.Pink,
Color.LightCyan };
        public GlifoEspecifico(string theID, uint num) : base(theID, num)
        {
            RefImage = theID;
            Numero = num;
            Conexion.callReader((Queries.SELECT("TNawales", "Simbologia",
"refImage = \"\" + RefImage + \"\""));
            while (Conexion.reader.Read())
            {
                botones = Conexion.reader.GetString(0).Split(',').ToArray();
            }
        }
        public override void SetButtons() //Polimorfismo de la función para
generar un arreglo de botones para identificar todos los símbolos dentro de
un glifo
        {
            Buttons = new Button[botones.Length + 1];

            for (int i = 1; i <= botones.Length; i++)
            {
                Buttons[i] = new Button
                {
                    Text = (string)botones.GetValue(i - 1),
                    Tag = RefImage + i,
                    BackColor = color[i - 1],
                    ForeColor = Color.Black,
                    Font = new Font("Microsoft Sans Serif", 15.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)0)),
                    Margin = new Padding(0, 3, 0, 3),
                    Size = new Size(250, 60)
                };
            }
        }
    }

```



```

        Buttons[i].Click += new
System.EventHandler(Codices.thisForm.ResaltarGlifo);
    }
    base.SetButtons();
}

```

```

}

class FueraDeRango : Exception //Nueva excepción
{
    public override string Message { get { return "La fecha especificada
excede el mínimo. Favor de ingresar\n fechas de nacimiento posteriores al
2 de enero de 1910"; } }
}
}

```

```

/* Arias Cortina Carlos    0217787
* Cabrera Ábrego Raúl Andres    0213359
* González Juárez Javier 0215241
* Guerrero Valadez Juan Manuel 0211637
* nombre id materia equipo
* Programación Orientada a Objetos
* Sala Maya
*/
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AppSalaMaya.Scripts
{
    public partial class IngresarCredenciales : Form
    {
        public IngresarCredenciales()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {

```

```

        TimerDialog.segundos = 0;
        if (LogIn.VaidarUsuario(textBox1.Text, textBox2.Text))
            this.DialogResult = DialogResult.OK;
        else
        {
            label1.ForeColor = Color.Red;
            label2.ForeColor = Color.Red;
            System.Threading.Thread.Sleep(200);
            label1.ForeColor = Color.Black;
            label2.ForeColor = Color.Black;
            System.Threading.Thread.Sleep(200);
            label1.ForeColor = Color.Red;
            label2.ForeColor = Color.Red;
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        TimerDialog.segundos = 0;
        this.Dispose();
    }

    private void IngresarCredenciales_Load(object sender, EventArgs e)
    {
    }
}

/* Arias Cortina Carlos      0217787
 * Cabrera Ábrego Raúl Andres 0213359
 * González Juárez Javier 0215241
 * Guerrero Valadez Juan Manuel 0211637
 * nombre id materia equipo
 * Programación Orientada a Objetos
 * Sala Maya
 */
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AppSalaMaya
{
    class Maya
    {
        //Random
        int Img;
    }
}

```

```

Random random = new Random();

//ArrayList
ArrayList arraylist = new ArrayList{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30};

//Atributos
private int score;
private bool bip;
private int contador;

//Propiedades
public int Score
{ get { return score; } set { score = value; } }
public bool Bip
{ get { return bip; } set { bip = value; } }
public int Contador
{ get { return contador; } set { contador = value; } }

//Constructor vacío
public Maya()
{
    Score = 0;
    Bip = false;
    Contador = 5;
}

//Constructor
public Maya(int Score, bool Bip)
{
    this.Score = Score;
    this.Bip = Bip;
}

//Función que delimitea la imagen que aparecerá sin repetir otras
public int Picture()
{
    bool brk = true;
    do
    {
        Img = random.Next(1, 31);
        for (int i = 0; i < arraylist.Count; i++)
        {
            if (int.Parse(arraylist[i].ToString()) == Img)
            {
                arraylist.Remove(Img);
                brk = false;
                break;
            }
        }
    } while (brk == true);
}

```

```

        return Img;
    }

}

```

```

/* Arias Cortina Carlos    0217787
 * Cabrera Ábrego Raúl Andres  0213359
 * González Juárez Javier 0215241
 * Guerrero Valadez Juan Manuel 0211637
 * nombre id materia equipo
 * Programación Orientada a Objetos
 * Sala Maya
 */

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace AppSalaMaya

```

```

{
    public partial class MAYA_O_NO : Form
    {
        Maya maya = new Maya();

        public MAYA_O_NO()
        {
            InitializeComponent();
            dador = maya.Picture();

```

```

PB01.Image =
(Image)(Properties.Resources.ResourceManager.GetObject(string.Format("_"
dador.ToString())));

```

```

    }

```

```

    private void BipC(object sender, EventArgs e)
    {
        Scripts.TimerDialog.segundos = 0;
        if (maya.Contador == 0)
        {
            maya = new Maya();

```

```

        label1.Text = "Puntaje: " + maya.Score;
    }

    //Suma 2 puntos al score cada vez que hay un acierto
    #region Puntaje

    if (((Button)sender).Text == "Si")
    {
        if (dador <= 15)
        {
            maya.Score += 2;
            label1.Text = "Puntaje: " + maya.Score;
            maya.Bip = true;
        }
        else { maya.Bip = false; }
    }
    else
    {
        if (dador > 15)
        {
            maya.Score += 2;
            label1.Text = "Puntaje: " + maya.Score;
            maya.Bip = true;
        }
        else { maya.Bip = false; }
    }
}

#endregion

//Cambia la imagen despues de contestar
dador = maya.Picture();

PB01.Image =
(Image)(Properties.Resources.ResourceManager.GetObject(string.Format("_"
dador.ToString())));

//Permite que los botones Cambien de color al momento de dar una
respuesta
#region BIP

    if (maya.Bip == true)
    {
        ((Button)sender).BackColor = Color.LightGreen;
        ((Button)sender).Update(); Thread.Sleep(200);
        ((Button)sender).BackColor = Color.Transparent;
        ((Button)sender).Update(); Thread.Sleep(200);
        ((Button)sender).BackColor = Color.LightGreen;
        ((Button)sender).Update(); Thread.Sleep(200);
        ((Button)sender).BackColor = Color.Transparent;
        ((Button)sender).Update();
    }
    else

```

```

        {
            ((Button)sender).BackColor = Color.LightPink;
            ((Button)sender).Update(); Thread.Sleep(200);
            ((Button)sender).BackColor = Color.Transparent;
            ((Button)sender).Update(); Thread.Sleep(200);
            ((Button)sender).BackColor = Color.LightPink;
            ((Button)sender).Update(); Thread.Sleep(200);
            ((Button)sender).BackColor = Color.Transparent;
            ((Button)sender).Update();
        }

        #endregion

        maya.Contador -= 1;

        //Reinica el contador y el score una vez que se llega al máximo de
preguntas
        if (maya.Contador == 0)
        {
            Score score = new Score();
            this.Hide();
            score.Finals.Text = "Garcias por paticipar \r\n" +
                "su puntaje final es: \r\n" +
                maya.Score.ToString() + " puntos";
            score.ShowDialog();
            Scripts.Menu.menu.Show();
        }
    }

    private void btnGoBack_Click(object sender, EventArgs e)
    {
        Scripts.TimerDialog.segundos = 0;
        Scripts.Menu.menu.Show();
        Usuario.usuario.PuntajeMaya = maya.Score;
        this.Hide();
    }

    private void MAYA_O_NO_Load(object sender, EventArgs e)
    {
    }
}

```

```

/* Arias Cortina Carlos    0217787
* Cabrera Ábrego Raúl Andres  0213359
* González Juárez Javier 0215241
* Guerrero Valadez Juan Manuel 0211637
* nombre id materia equipo
* Programación Orientada a Objetos

```

```

* Sala Maya
*/
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AppSalaMaya.Scripts
{
    public partial class Menu : Form
    {
        public Menu()
        {
            InitializeComponent();
            TimerDialog elTimer = new TimerDialog();
        }
        private Form[] subForms;
        public string currentPassword;
        public static Scripts.Menu menu;
        private void btnAbrirForm(object sender, EventArgs e)
        {
            TimerDialog.segundos = 0;
            for (int i = 0; i < 6; i++)
            {
                if (int.Parse((string)((Button)sender).Tag) == i)
                {
                    defineForms(i);
                    subForms[i].Show();
                    this.Hide();
                    break;
                }
            }
        }
        private void Menu_Load(object sender, EventArgs e)
        {
            TimerDialog.segundos = -1;
            Conexion.getCon();
            menu = this;
            subForms = new Form[5];
        }
        private void defineForms(int f)
        {
            switch (f)
            {
                case 0:
                    subForms[0] = new Codices();
            }
        }
    }
}

```

```

        break;
    case 1:
        subForms[1] = new TimeLine();
        break;
    case 2:
        subForms[2] = new QuizDios();
        break;
    case 3:
        subForms[3] = new MAYA_O_NO();
        break;
    case 4:
        subForms[4] = new Registrar();
        break;
    }
}

private void AccionAdministrador(object sender, EventArgs e)
{
    TimerDialog.segundos = 0;
    IngresarCredenciales dialog = new IngresarCredenciales();
    if (dialog.ShowDialog() == DialogResult.OK)
    {
        switch (int.Parse((string)((ToolStripMenuItem)sender).Tag))
        {
            case 0:
                Registro registro = new Registro();
                registro.ShowDialog();
                break;
            case 1:
                LogIn logIn = new LogIn();
                logIn.Show();
                TimerDialog.segundos = -1;
                this.Dispose();
                break;
            case 2:
                Application.Exit();
                break;
        }
    }
}

}

}

/* Arias Cortina Carlos    0217787
 * Cabrera Ábrego Raúl Andres    0213359
 * González Juárez Javier 0215241
 * Guerrero Valadez Juan Manuel 0211637
 * nombre id materia equipo
 * Programación Orientada a Objetos
 * Sala Maya
 */

```



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AppSalaMaya.Scripts
{
    public partial class MessageBoxDios : Form
    {
        public MessageBoxDios()
        {
            InitializeComponent();

            private void btnRegresar_Click(object sender, EventArgs e)
            {
                TimerDialog.segundos = 0;
                this.Dispose();
            }

            private void MessageBoxDios_Load(object sender, EventArgs e)
            {
            }
        }
    }
}

```

```

/* Arias Cortina Carlos    0217787
 * Cabrera Ábrego Raúl Andres  0213359
 * González Juárez Javier 0215241
 * Guerrero Valadez Juan Manuel 0211637
 * nombre id materia equipo
 * Programación Orientada a Objetos
 * Sala Maya
 */

```

```

using AppSalaMaya.Properties;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace AppSalaMaya.Scripts
{
    public partial class QuizDios : Form
    {
        public QuizDios()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            TimerDialog.segundos = 0;
            //Crear arreglo de dioses con valores predefinidos.
            Dios[] dioses = new Dios[5];
            dioses[0] = new Dios("Kauil", "Es el dios del fuego. Se relaciona con los gobernadores ya que su sinónimo es de poder.", new Bitmap(Resources.kauil));
            dioses[1] = new Dios("Itzamná", "Dios del sol y la sabiduría. También señor del cielo, el día y la noche.", new Bitmap(Resources.itzamna));
            dioses[2] = new Dios("Chaac", "Es el dios de la lluvia, el relámpago y el agua. ", new Bitmap(Resources.chaac));
            dioses[3] = new Dios("Ixchel", "Diosa de la fertilidad y de la Luna. Muy relacionada con el género femenino y la maternidad. ", new Bitmap(Resources.ixchel));
            dioses[4] = new Dios("Kukulcán", "En la cultura maya, es el creador del universo. Es el dios del viento y su nombre se traduce como "serpiente emplumada".", new Bitmap(Resources.Kukulcan));
            bool validar;
            foreach (GroupBox group in mainLayout.Controls) //leer todos los groupbox dentro del mainLayout
            {
                validar = false; //reiniciar bandera
                foreach (RadioButton rb in group.Controls[0].Controls) //Para acceder a todos los radiobuttons dentro del flowlayoutpanel del groupbox actual
                {
                    if (rb.Checked) //En caso de que el usuario haya seleccionado el RadioButton rb
                    {
                        validar = true; //al encontrar un radiobutton que este seleccionado, se sabrá que si se lleno el groupbox actual
                        for (int i = 0; i < 5; i++)
                        {
                            dioses[i].Puntos += ((string)rb.Tag)[i]; //Se le suma el valor que tiene rb en su propiedad tag en el caracter i al dios i
                        }
                    }
                }
                if(validar == false)//en caso de que halla dejado una respuesta sin contestar
                {
                    MessageBox.Show("No puedes dejar respuestas sin responder", "¡Atención!", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }
    }
}

```

```

        return;
    }
}
int MaxPuntos = 0;
for(int i = 1; i < 5; i++)
{
    if(dioses[i].Puntos > dioses[MaxPuntos].Puntos)
    {
        MaxPuntos = i;
    }
}
//Para obtener el indice del dios mas cercano a las respuestas del
usuario
Usuario.usuario.Dios = dioses[MaxPuntos].Nombre; //mandaremos la
información al objeto Usuario según el Dios que le haya tocado al usuario
MessageBoxDios boxDios = new MessageBoxDios();
//Pasar resultados al MessageBoxDios
boxDios.lbName.Text = dioses[MaxPuntos].Nombre;
boxDios.lbDesc.Text = dioses[MaxPuntos].Descripcion;
boxDios.pbDios.Image = dioses[MaxPuntos].Imagen;
this.Hide();
boxDios.ShowDialog(); //Mostrar resultado
Scripts.Menu.menu.Show(); //volver al menu
this.Dispose();
}

private void btnRegresar_Click(object sender, EventArgs e) //en caso de que no
quiera llenar el formulario
{
    TimerDialog.segundos = 0;
    Scripts.Menu.menu.Show(); //volver al menu
    this.Dispose();
}

private void rbPink_Click(object sender, EventArgs e)
{
    TimerDialog.segundos = 0;
}

private void QuizDios_Load(object sender, EventArgs e)
{
}

private void rbPink_CheckedChanged(object sender, EventArgs e)
{
    RadioButton radio = (RadioButton)sender;
    if (radio.Checked) radio.BackColor = Color.FromArgb(100,
radio.BackColor.R, radio.BackColor.G, radio.BackColor.B);
    else radio.BackColor = Color.FromArgb(255, radio.BackColor.R,
radio.BackColor.G, radio.BackColor.B);
}

```

```

    }
    public class Dios
    {
        //atributos
        private string nombre;
        private string descripción;
        private Bitmap imagen;
        private uint puntos;
        //propiedades
        public string Nombre { get { return nombre; } set { nombre = value; } }
        public string Descripcion { get { return descripción; } set { descripción =
value; } }
        public Bitmap Imagen { get { return imagen; } set { imagen = value; } }
        public uint Puntos { get { return puntos; } set { puntos = value; } }

        public Dios()//constructor vacio
        {
            this.Nombre = "";
            this.Descripcion = "";
            this.Imagen = null;
            this.Puntos = 0;
        }
        public Dios(string nom, string desc, Bitmap imag)//constructor usado para la
inicialización
        {
            this.Nombre = nom;
            this.Descripcion = desc;
            this.Imagen = imag;
            this.Puntos = 0;
        }
    }
}

/* Arias Cortina Carlos    0217787
 * Cabrera Ábrego Raúl Andres    0213359
 * González Juárez Javier 0215241
 * Guerrero Valadez Juan Manuel 0211637
 * nombre id materia equipo
 * Programación Orientada a Objetos
 * Sala Maya
 */
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Linq;
using System.Text;

```

```

using System.Threading.Tasks;
using System.Windows.Forms;
using System.Net.Mail;
using System.IO;

namespace AppSalaMaya.Scripts
{
    public partial class Registrar : Form
    {
        public Registrar()
        {
            InitializeComponent();

            bool validar = false;
            private void RadioButtonCheckedChanged(object sender, EventArgs e)
            {
                TimerDialog.segundos = 0;
                validar = true;
                Usuario.usuario.Satisfaccion
int.Parse((string)((RadioButton)sender).Tag);
            }

            private void button1_Click(object sender, EventArgs e)
            {
                TimerDialog.segundos = 0;
                //Presion hidrostática, centros de masa
                Usuario.usuario.Nombre = tbNombre.Text;
                if (validar == true)
                {
                    if (SendMessage(tbNombre.Text, tbCorreo.Text))
                    {
                        /*Correo enviado --> Usuario.usuario.archivoEnviado*/
                        Conexion.Ejecutar(Queries.INSERT("TRegistro", "(Nombre, Correo,
Nacimiento, ResultadoLineaTiempo, NawalString, PuntajeEsMaya, Dios, Satisfaccion,
CorreoEnviado)", new dynamic[9]
                        {
                            tbNombre.Text, tbCorreo.Text,
                            Usuario.usuario.Nacimiento.ToShortDateString(), Usuario.usuario.ResultadoLinea,
                            Usuario.usuario.NawalString, Usuario.usuario.PuntajeMaya, Usuario.usuario.Dios,
                            Usuario.usuario.Satisfaccion, Usuario.usuario.ArchivoEnviado }));

                        if (Usuario.usuario.ArchivoEnviado)
                        {
                            MessageBox.Show("Archivo enviado. Gracias.", "e-mail",
MessageBoxButtons.OK);
                        }
                        Scripts.Menu.menu.Show();
                        this.Hide();
                        Usuario.GenerarUsuario();
                    }
                }
            }
        }
    }
}

```

```

    }
    }
    else
    {
        MessageBox.Show("Antes de enviar tu archivo, tómate \nla molestia de
calificar nuestra app.\n¡Muchas gracias!");
        groupBox1.ForeColor = Color.Salmon;
    }
}
public bool SendMessage(string nombre, string correo)
{
    if (nombre == "")
        nombre = GetID();
    MailMessage message = new MailMessage();
    message.Subject = "Resultados de la Sala Maya - " + nombre;
    message.Body = "";

    message.From = new MailAddress("salamaya.upSolutions@gmail.com");
    try
    {
        message.To.Add(tbCorreo.Text);
        SmtpClient client = new SmtpClient();
        client.Credentials = new
System.Net.NetworkCredential("salamaya.upSolutions@gmail.com", "salamaya1234");
        client.Port = 587;
        client.EnableSsl = true;
        client.Host = "smtp.gmail.com";
        GeneradorDeArchivos.GenerarArchivo();
        Attachment attachment = new Attachment(nombre + "_Resultados.rtf");
        message.Attachments.Add(attachment);
        client.Send(message);
        attachment.Dispose();
        Usuario.usuario.ArchivoEnviado = true; //Para guardar en la base de
datos que si se envió el correo.
    }

    catch (FormatException)
    {
        CorreoIncorrecto();
        return false; //Para que no se guarde nada en la base de datos por
error de usuario.
    }
    catch (ArgumentException)
    {
        CorreoIncorrecto();
        return false; //Para que no se guarde nada en la base de datos por
error de usuario.
    }
    catch (System.Net.Mail.SmtpException)
    {

```

```

        MessageBox.Show("Nuestro servidor no responde. \n Te enviaremos tus
resultados lo antes posible.\n Lamentamos los inconvenientes.", "Server not
responding", MessageBoxButtons.OK, MessageBoxIcon.Error);
        Usuario.usuario.ArchivoEnviado = false; //Para guardar en la base de
datos que no se envió el correo por error de sistema.
    }
    catch (AggregateException)
    {
        MessageBox.Show("Nuestro servidor no responde. \n Te enviaremos tus
resultados lo antes posible.\n Lamentamos los inconvenientes.", "Server not
responding", MessageBoxButtons.OK, MessageBoxIcon.Error);
        Usuario.usuario.ArchivoEnviado = false; //Para guardar en la base de
datos que no se envió el correo por error de sistema.
    }
    catch (Exception)
    {
        MessageBox.Show("¡Ups! Esto no debió haber pasado", "Error
desconocido", MessageBoxButtons.OK, MessageBoxIcon.Error);
        Usuario.usuario.ArchivoEnviado = false; //Para guardar en la base de
datos que no se envió el correo por error de sistema.
    }
    if (File.Exists(nombre + "_Resultados.rtf") &&
Usuario.usuario.ArchivoEnviado) File.Delete(nombre + "_Resultados.rtf");
    return true;
}
public string GetID()
{
    return "";
}
public async void CorreoIncorrecto()
{
    Color aux = tbCorreo.BackColor;
    for(int i = 0; i < 3; i++)
    {
        tbCorreo.BackColor = Color.LightPink;
        await Task.Delay(300);
        tbCorreo.BackColor = aux;
        await Task.Delay(300);
    }
    tbCorreo.BackColor = Color.LightPink;
}

private void button2_Click(object sender, EventArgs e)
{
    TimerDialog.segundos = 0;
    this.Hide();
    Scripts.Menu.menu.Show();
}
private void tbNombre_TextChanged(object sender, EventArgs e)
{

```

```

        TimerDialog.segundos = 0;
    }
}

/* Arias Cortina Carlos    0217787
 * Cabrera Ábrego Raúl Andres    0213359
 * González Juárez Javier 0215241
 * Guerrero Valadez Juan Manuel 0211637
 * nombre id materia equipo
 * Programación Orientada a Objetos
 * Sala Maya
 */
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AppSalaMaya.Scripts
{
    public partial class Registro : Form
    {
        public Registro()
        {
            InitializeComponent();

            private void Registro_Load(object sender, EventArgs e)
            {
                string fecha = "";
                TimerDialog.segundos = -1;
                dataGridView1.Rows.Clear();
                Conexion.callReader(Queries.SELECT("TRegistro", "*"));
                while (Conexion.reader.Read())
                {
                    fecha = "";
                    if (Conexion.reader.GetDateTime(3) != new DateTime(1001, 1, 1)) fecha
= Conexion.reader.GetDateTime(3).ToShortDateString();
                                dataGridView1.Rows.Add(Conexion.reader.GetValue(0),
                                Conexion.reader.GetValue(1), Conexion.reader.GetValue(2), fecha,
                                Conexion.reader.GetValue(4), Conexion.reader.GetValue(5),
                                Conexion.reader.GetValue(6), Conexion.reader.GetValue(7),
                                Conexion.reader.GetValue(8), Conexion.reader.GetBoolean(9));
                }
            }
        }
    }
}

```



```

    }

    private void button1_Click(object sender, EventArgs e)
    {
        TimerDialog.segundos = 0;
        this.Dispose();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        foreach(DataGridViewRow row in dataGridView1.Rows)
        {
            if (row.Cells[0].Value == null) continue;
            Conexion.Ejecutar(Querries.UPDATE("TRegistro", "CorreoEnviado",
row.Cells[9].Value, "ID = " + row.Cells[0].Value));
        }
        MessageBox.Show("Base de datos actualizada correctamente");
    }
}
}
}

```

```

/* Arias Cortina Carlos    0217787
 * Cabrera Ábrego Raúl Andres    0213359
 * González Juárez Javier 0215241
 * Guerrero Valadez Juan Manuel 0211637
 * nombre id materia equipo
 * Programación Orientada a Objetos
 * Sala Maya
 */
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AppSalaMaya.Scripts
{
    public partial class Resultados : Form
    {
        public Resultados()
        {
            InitializeComponent();
        }
        public bool[] boolaneos;
        private void Resultados_Load(object sender, EventArgs e)
        {

```

```

        TimerDialog.segundos = 0;
        int i = 0;
        foreach (Label lb in flowLayoutPanel1.Controls)
        {
            if (boolaneos[i])
                lb.Text += " ✓";
            else
                lb.Text += " ✗";
            i++;
        }
    }

    private void BtnRegre_Click(object sender, EventArgs e)
    {
        TimerDialog.segundos = 0;
        this.Hide();
    }

    private void Label6_Click(object sender, EventArgs e)
    {
        TimerDialog.segundos = 0;
    }

    private void Label5_Click(object sender, EventArgs e)
    {
        TimerDialog.segundos = 0;
    }

    private void Label4_Click(object sender, EventArgs e)
    {
        TimerDialog.segundos = 0;
    }

    private void Label3_Click(object sender, EventArgs e)
    {
        TimerDialog.segundos = 0;
    }
}

/* Arias Cortina Carlos    0217787
 * Cabrera Ábrego Raúl Andres  0213359
 * González Juárez Javier 0215241
 * Guerrero Valadez Juan Manuel 0211637
 * nombre id materia equipo
 * Programación Orientada a Objetos
 * Sala Maya
 */

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AppSalaMaya
{
    public partial class Score : Form
    {
        MAYA_O_NO maya_o_no;

        public Score()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            this.Dispose();
        }
    }
}

/* Arias Cortina Carlos      0217787
 * Cabrera Ábrego Raúl Andres 0213359
 * González Juárez Javier 0215241
 * Guerrero Valadez Juan Manuel 0211637
 * nombre id materia equipo
 * Programación Orientada a Objetos
 * Sala Maya
 */
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Collections;

namespace AppSalaMaya.Scripts
{
    public partial class TimeLine : Form
    {
        public int imgNum = 0, score = 0 ,numcont = 0 ;
    }
}

```

```

public int[] Fecha = { 0, 0, 0, 0, 0 };
private Bitmap aux;
private Pieza[] pie = new Pieza[5];
public Timeline()
{
    InitializeComponent();
}
public Form menu;
static PictureBox[] pictureBoxes; //Arreglo de picturebox
static Label[] labels; //Arreglo de label

private void Timeline_Load(object sender, EventArgs e)
{
    TimerDialog.segundos = 0;
    //Se definen los pictureBoxes
    pictureBoxes = new PictureBox[5];

    pictureBoxes[0] = pictureBox1;
    pictureBoxes[1] = pictureBox2;
    pictureBoxes[2] = pictureBox3;
    pictureBoxes[3] = pictureBox4;
    pictureBoxes[4] = pictureBox5;

    //Se definen los labels
    labels = new Label[5];
    labels[0] = lbl1;
    labels[1] = lbl2;
    labels[2] = lbl3;
    labels[3] = lbl4;
    labels[4] = lbl5;

    Random aleat = new Random();
    int[] Position = { 0, 0, 0, 0, 0 };
    int num = 0, cont = 0;
    //Se crean las posiciones random
    for (int i = 0; i < 5; i++)
    {
        if (cont > 0)
        {
            do
            {
                Position[i] = aleat.Next(0, 8);
                for (int j = 0; j < cont; j++)
                {
                    if (Position[i] == Position[j]) //Se valida la posición
                    {
                        num = 0;
                    }
                    else
                    {
                        num += 1;
                    }
                }
            } while (num == 0);
        }
        cont++;
    }
}

```

```

        }
    }
    } while (num < cont);
}
if (cont == 0)
{
    Position[i] = aleat.Next(0, 8); //Se toma la primera posición
random
    }
    num = 0;
    cont++;
}

    int[] AllFechas = { 250, 550, 591, 600, 683, 900, 1000, 1250
}; //Arreglo con todas las fechas
    for (int i = 0; i < 5; i++)
    {
        Fecha[i] = AllFechas[Position[i]]; //Fechas random
    }
    Pieza xpi = new Pieza();
    for (int i = 0; i < 5; i++)
    {
        //imagen random
        pictureBoxes[i].Image
        (Image)(Properties.Resources.ResourceManager.GetObject(string.Format("image{0}",
        Position[i] + 1)));
        //Se inicializa el constructor
        pie[i] = new Pieza(AllFechas[Position[i]],
        (Bitmap)pictureBoxes[i].Image);
    }

    Array a = Fecha.ToArray();
    Array.Sort(a);
    for (int i = 0; i < Fecha.Length; i++)
    {
        Fecha[i] = (int)a.GetValue(i); //Fechas ordenadas
        labels[i].Text = (Fecha[i]).ToString();
    }

    pictureBox1.Tag = (Fecha[0]).ToString();
    pictureBox2.Tag = (Fecha[1]).ToString();
    pictureBox3.Tag = (Fecha[2]).ToString();
    pictureBox4.Tag = (Fecha[3]).ToString();
    pictureBox5.Tag = (Fecha[4]).ToString();

    //Se habilita la propiedad AllowDrop para todos los pictureBox

    pictureBox1.AllowDrop = true;
    pictureBox2.AllowDrop = true;

```

```

    pictBox3.AllowDrop = true;
    pictBox4.AllowDrop = true;
    pictBox5.AllowDrop = true;

    pictureBox1.AllowDrop = true;
    pictureBox2.AllowDrop = true;
    pictureBox3.AllowDrop = true;
    pictureBox4.AllowDrop = true;
    pictureBox5.AllowDrop = true;
}

bool estaEnArea;
private void PictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    TimerDialog.segundos = 0;
    estaEnArea = false;
    PictureBox pb = (PictureBox)sender; //Se crea objeto tipo picturebox
    pb.Select();
    if(pb.Image != null)
    {
        pb.DoDragDrop(pb.Image, DragDropEffects.Move);
        int Num = int.Parse(pb.TabIndex.ToString());
        if (estaEnArea)
            pb.Image = aux;
        for (int i = 0; i < 5; i++)
        {
            if (((PictureBox)sender).Image == pie[i].Pict)
            {
                if (((PictureBox)sender).Tag != null)
                {
                    if ((string)((PictureBox)sender).Tag ==
pie[i].Tiempo.ToString())
                    {
                        pie[i].Correct = true;
                    }
                    else
                    {
                        pie[i].Correct = false;
                    }
                    break;
                }
                else
                {
                    pie[i].Correct = false;
                }
            }
        }
    }
}
}

```

```

    }
    private void PictureBox1_DragEnter(object sender, DragEventArgs e)
    {
        if (e.Data.GetDataPresent(DataFormats.Bitmap))
        {
            e.Effect = DragDropEffects.Move;
        }
        else
        {
            e.Effect = DragDropEffects.None;
        }
    }
}

private void PictureBox1_DragDrop(object sender, DragEventArgs e)
{
    estaEnArea = true;
    PictureBox pb = (PictureBox)sender;
    aux = (Bitmap)pb.Image;
    pb.Image = (Bitmap)e.Data.GetData(DataFormats.Bitmap);

    for (int i = 0; i < 5; i++)
    {
        if (((PictureBox)sender).Image == pie[i].Pict)
        {
            if (((PictureBox)sender).Tag != null)
            {
                if (((string)((PictureBox)sender).Tag
                    == pie[i].Tiempo.ToString()))
                {
                    pie[i].Correct = true;
                }
                else
                {
                    pie[i].Correct = false;
                }
                break;
            }
            else
            {
                pie[i].Correct = false;
            }
        }
    }
}

private void Button1_Click(object sender, EventArgs e)
{

```

```

        TimerDialog.segundos = 0;

        this.Dispose();
        Scripts.Menu.menu.Show();

    }
    private void BtnCalcular_Click(object sender, EventArgs e)
    {
        TimerDialog.segundos = 0;

        Resultados res = new Resultados();
        for (int i = 0; i < 5; i++)
        {
            if (pie[i].Correct)
            {
                score++;
            }
        }
        Usuario.usuario.ResultadoLinea = score * 2;
        res.lblScore.Text = "Tuviste "+score+" aciertos de 5";
        res.boolaneos = new bool[5];
        for(uint i = 0; i < 5; i++)
        {
            for (uint j = 0; j < 5; j++)
            {
                if (pie[i].Tiempo == Fecha[j])
                {
                    pie[i].Orden = j;
                }
            }
        }

        for(int i = 0; i < 5; i++)
        {
            res.boolaneos[pie[i].Orden] = pie[i].Correct;
        }
        this.Hide();
        res.ShowDialog();
        Scripts.Menu.menu.Show();
        this.Hide();
    }
}

public class Pieza
{
    //Atributos
    private int tiempo;
    private Bitmap pict;
    private bool correct;
    private uint orden;

```



```

//Propiedades
public uint Orden
{
    get { return orden; }
    set { orden = value; }
}
public int Tiempo
{
    get { return tiempo; }
    set { tiempo = value; }
}
public Bitmap Pict
{
    get { return pict; }
    set { pict = value; }
}
public bool Correct
{
    get { return correct; }
    set { correct = value; }
}
public Pieza()//Constructor vacio
{
    this.Tiempo = 0;
    this.Pict = null;
    this.Correct = false;
    this.Orden = 0;
}
public Pieza(int tiempx, Bitmap pictx)//Constructor usado para instanciar los
objetos
{
    this.Tiempo = tiempx;
    this.Pict = pictx;
    this.Correct = false;
    this.Orden = 0;
}

}
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;

```

```

using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AppSalaMaya.Scripts
{
    public partial class TimerDialog : Form
    {
        public TimerDialog()
        {
            InitializeComponent();
            this.timer1.Start();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            segundos = 80;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Hide();
            segundos = 0;
        }

        private void Timer_Load(object sender, EventArgs e)
        {
        }

        }

        public static int segundos = -1;
        private void timer1_Tick(object sender, EventArgs e)
        {
            if(segundos != -1)
            {
                segundos++;
                if(segundos== 60) this.ShowDialog();
                if (segundos > 60)
                {
                    label2.Text = "Tiempo restante: " + (80 - segundos) + "s.";
                }
                if(segundos > 80)
                {
                    Usuario.GenerarUsuario();
                    Cerrar(this);
                    this.Hide();
                }
            }
        }

        private static void Cerrar(Form este)//Reinicia el programa al menú principal
        en caso de que un usuario abandone la computadora
        {

```


```
        for (int i = Application.OpenForms.Count - 1; i >= 0; i--)
        {
            Application.OpenForms[i].Hide();
        }
        Scripts.Menu.menu.Show();
        segundos = -1;
    }
}
```

Anexo 2: Archivo enriquecido

Nombre

Sala Maya, Museo de Antropología e Historia de la Ciudad de México

Tú serías: **Kukulcán**

	<p>11 Iq'</p> <p>Es el viento, elemento que rige las ideas y el cambio. Significa aire y aliento de vida.</p>
---	--

Resultados:

- Línea del tiempo: **10**
- ¿Es maya? : **0**

Gracias por usar nuestra app,
Atentamente UPSolutions.

salamaya.upsolutions@gmail.com