

S A B E R



**DESARROLLO DE APLICACIONES
MULTIPLATAFORMA**

**TRABAJO FIN DE GRADO 2º
CURSO 2022 - 2023**

ALUMNO: DIANA GRAK

TUTORES: MARÍA PILAR LOZANO BERNAL



Índice

1. Datos descriptivos	2
1.1. Descripción general. Marco teórico. Principios tecnológicos y normas técnicas aplicables.	2
1.2. Análisis de la realidad.	4
1.3. Entorno de la empresa colaboradora. Definición de una empresa	7
1.4. Justificación. (Necesidad/ Innovación detectada).....	8
1.5. Destinatarios. (Si es la propia empresa u otro destinatario)	9
2. Desarrollo del proyecto	9
2.1. Acuerdo de proyecto.....	9
2.1.1. Generales	9
2.1.2. Requisitos funcionales.....	10
2.1.3. Requisitos no funcionales	10
2.1.4. Especificaciones técnicas del proyecto.....	10
2.2. Actividades y cronología.....	13
2.2.1. Diagrama de Gantt	13
2.2.2. Documento de análisis y diseño	14
3. Diseño Interfaces gráficas	19
4. Doc. Cliente, Metodología y Recursos.....	28
4.1. Documentación para el cliente.....	28
4.1.1. Manual de instalación o configuración	28
4.1.2. Manual de usuario	29
4.2. Metodología seguida.....	32
4.3. Recursos.....	33
4.3.1. Materiales.....	33
4.3.2. Infraestructura y equipamiento	33
4.3.3. Humanos	33
5. Resumen ejecutivo (inglés)	34
6. Pruebas del sistema.....	34
7. Presupuesto.....	38
8. Resultados objetivos y conclusiones	39
9. Bibliografía.....	39
10. Anexos	40



1. Datos descriptivos

1.1. Descripción general. Marco teórico. Principios tecnológicos y normas técnicas aplicables.

En este proyecto se desarrolla una aplicación de entretenimiento. Se trata de un juego de preguntas donde el jugador deberá escoger la respuesta correcta entre las cuatro opciones que se presentan en la pantalla.

Las preguntas son aleatorias, y se clasifican en varias categorías. En principio, se consideran las siguientes:

- Cine
- Videojuegos
- Informática
- Arte
- Filosofía
- Literatura

En un futuro se prevé ampliar el juego añadiendo mayor número de preguntas por categoría y otras categorías nuevas.

Es posible adaptar la temática según las necesidades y el uso final del cliente a su petición. Por ejemplo, si se aplica en un entorno académico específico como herramienta de aprendizaje.





Por otra parte, también existe un segundo modo de juego de competición en el que podrán participar dos jugadores. El ganador será aquel con mayor puntuación obtenida mediante las preguntas acertadas. Otro posible resultado puede ser empate.

Como premio, al finalizar la partida, se visualizará uno de los datos curiosos, almacenados en la base de datos, poco conocidos en la pantalla. Por ejemplo,



Es una aplicación de escritorio para Windows y desarrollada en C#. Se usará pgAdmin como herramienta para gestionar y administrar PostgreSQL. La base de datos contendrá las categorías, preguntas, respuestas y las curiosidades de premio.

Se trata de una aplicación sencilla cuyo objetivo es demostrar los conocimientos adquiridos en el uso del entorno de desarrollo Microsoft Visual Studio, programación en C#, el uso de un lenguaje de consulta estructurado SQL y el diseño de interfaces.

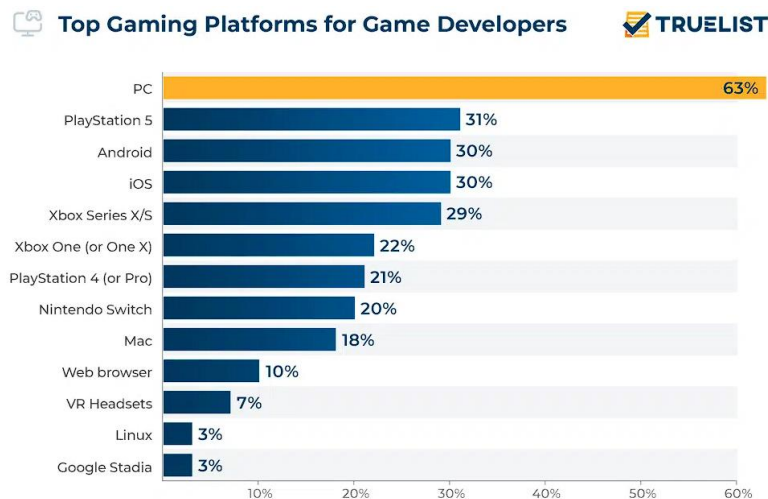
Cabe destacar, que el diseño de la interfaz es muy importante en este caso, porque al tratarse de una aplicación cuyo uso inicial no va más allá del entretenimiento, debe invitar visualmente al usuario a jugar, ser atractiva y agradable a la vista.



1.2. Análisis de la realidad.

En la actualidad, se encuentra un gran aumento de interés por el sector de los videojuegos, y se prevé que siga creciendo. Los juegos han demostrado ser un negocio muy lucrativo, según las estadísticas:

Gaming Statistics – 2023



Si bien existen muchos juegos de preguntas en el mercado, es probable que muchos de ellos ya hayan sido jugados y no suponen novedad para el usuario.

Son juegos de entretenimiento rápido y sencillo, normalmente en compañía. A pesar de ser parecidos entre sí, los usuarios suelen probar o jugar a más de uno.

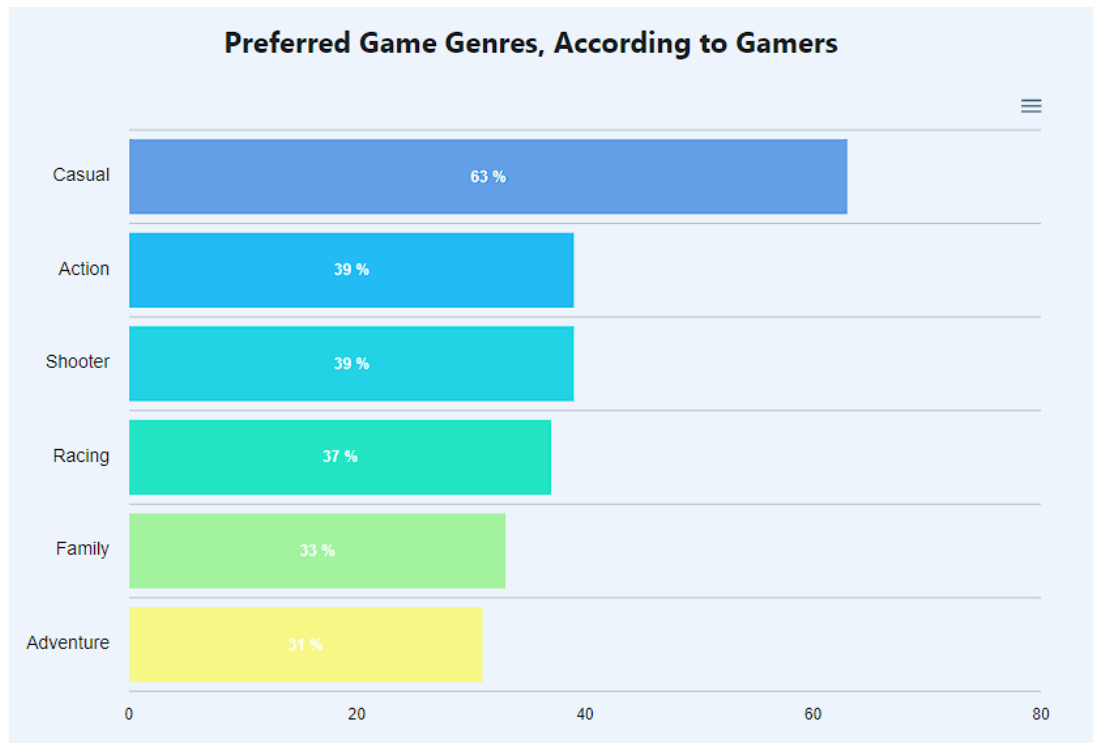
El factor “único” que proporcionan estos juegos tan semejantes, casi idénticos, es el efecto de la novedad. Nuevas temáticas, opciones y/o preguntas, en definitiva, una nueva experiencia potencialmente desafiante.

Es común que a un cierto nivel del juego las preguntas se repitan. También que las mismas o la temática no es, o deja de ser, del gusto del usuario y como resultado se descarga otro juego de la misma categoría.

Los juegos de preguntas son altamente rotativos porque ofrecen una amplia variedad y pueden adaptarse a diferentes gustos o preferencias.



23 Video Game and Online Gaming Statistics, Facts & Trends for 2023



Estos juegos también pueden aplicarse en el ámbito educativo, como una forma divertida de aprendizaje.

Es una enorme oportunidad que amplía las opciones de desarrollo de la aplicación. Porque con la misma funcionalidad o mecánica, simplemente cambiando el tema (categorías y preguntas) puede llegar a más de un segmento del mercado.

[Las trivias como recurso educativo: valor pedagógico, ventajas y desafíos](#)

[Trivial como método didáctico alternativo para formentar el aprendizaje en los estudiantes de enfermería.](#)



Aplicaciones similares que existen en el mercado:

- **Preguntados: Trivia Crack.** 100 M+ de descargas



Juego de trivia inicialmente desarrollado para dispositivos móviles. En el año 2022 también se convirtió en una serie interactiva de Netflix.

- **Kahoot! Play & Create Quizzes.** 50 M+ de descargas



Plataforma gratuita que permite la creación de cuestionarios de evaluación (disponible en app o versión web). Popular en el ámbito educativo.



- **It's Quiz Time.** 100 mil+ de descargas



Juego de preguntas y respuestas para PC (Microsoft Windows). Se puede usar un dispositivo móvil para interactuar y tiene tres modos de juego: solitario, local (hasta ocho jugadores); el último y el más interesante en el que se retransmite la partida para enfrentarse a los espectadores.

1.3. Entorno de la empresa colaboradora. Definición de una empresa

Se trata de una sociedad de responsabilidad limitada. Industria de videojuegos que forma parte del sector de la tecnología informática. El nombre de la empresa es "DG" con un único empleado Diana Grak.

Se desenvuelve en un entorno económico altamente competitivo. Sin embargo, como se ha mencionado anteriormente, en el entorno social se observa un incremento en el interés por los videojuegos. Sobre todo, gracias a la pandemia que vivimos en el año 2020 que ha favorecido tal incremento.

[Impact of the COVID-19 pandemic on the video game industry](#)

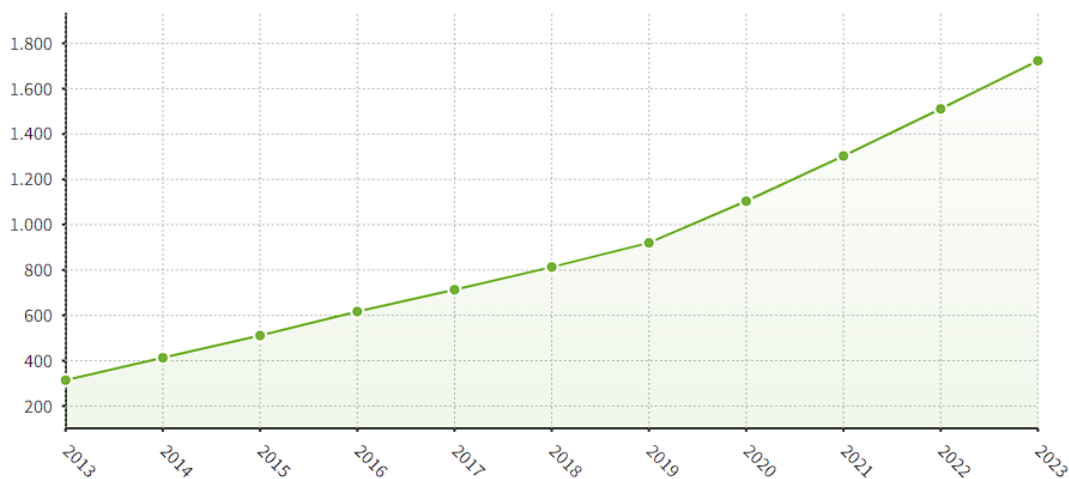


¿Cómo evoluciona la industria del videojuego en España?

Evolución de la facturación del sector español del desarrollo de videojuegos

Los datos de 2020 hacia adelante son previsiones de la asociación española de desarrolladores de videojuegos (DEV)

En millones de euros (Millones)



1.4. Justificación. (Necesidad/ Innovación detectada)

En el contexto actual es comprensible que los individuos busquen distraerse de la realidad a través del entretenimiento.

Por otra parte, como seres sociales y curiosos, anhelamos descubrir cosas nuevas y divertirnos en compañía de otros.

Cuando nos reunimos con nuestros cercanos solemos buscar juegos divertidos y ligeros para pasar el tiempo. Generalmente optamos por jugar juegos diferentes para romper la monotonía de la rutina.

Como se ha mencionado en el punto [1.2. Análisis de la realidad.](#), este tipo de juegos pueden abarcar una variedad de segmentos del mercado si lo aplicamos a la enseñanza. La mecánica es la misma y el temario puede variar según dónde y con qué finalidad se quiera aplicar.

Por ejemplo, ¿Podemos usar esta aplicación en autoescuelas y/o colegios, etc. como herramienta de aprendizaje?

- Por supuesto. Es cuestión de personalizar las categorías, preguntas y respuestas según la necesidad de los usuarios finales.

En conclusión, la aplicación es perfectamente adaptable a las necesidades de los usuarios.



1.5. Destinatarios. (Si es la propia empresa u otro destinatario)

Los destinatarios son los usuarios finales y es apto para todos los públicos. Sin embargo, es cierto que las preguntas tienen un cierto nivel de dificultad y quizá podrían suponer todo un reto para los más pequeños.

2. Desarrollo del proyecto

2.1. Acuerdo de proyecto

2.1.1. Generales

Según [Think with Google](#) los usuarios cada vez son más impacientes, existe una necesidad de inmediatez o toma de decisiones más rápida que nunca.

Las personas suelen buscar entretenimiento y/o conocimiento rápido, que distraiga del entorno, pero, a su vez que sea estimulante. De ahí, la tendencia mundial de los vídeos cortos. Que han sido aceptados positivamente por usuarios de plataformas como YouTube, Instagram, etc.

[Generación Impaciente y los nuevos consumidores lo quieren todo para ya](#)

[La impaciencia de los clientes, el nuevo reto de las empresas](#)

En nuestro caso, esto se aplica a que en un juego de preguntas no se requiere aprendizaje extra de controles, manejo del teclado, etc. Prácticas, que en muchos de los casos quitan la motivación de jugar. Un juego de preguntas es rápido, intuitivo y no es necesario aprender a jugarlo.

No existe novedad como tal, pero se aprovecha este entorno actual dónde prevalece y se valora la novedad constante de conocimiento y la rapidez.



2.1.2. Requisitos funcionales

- El usuario deberá descargar el ejecutable del juego o el proyecto completo. En caso de una base de datos en local, deberá ejecutar el script en PostgreSQL.
- El usuario debe poder seleccionar un modo de juego y una categoría para jugar.
- El usuario debe poder seleccionar las respuestas correspondientes para seguir el flujo secuencial de la partida y avanzar hasta el final.
- Es necesario respetar el turno para mantener la coherencia en el juego.

2.1.3. Requisitos no funcionales

- La aplicación está diseñada para el sistema operativo Windows.
- Espacio mínimo de almacenamiento 10,8 MB para la instalación de la aplicación.

2.1.4. Especificaciones técnicas del proyecto

Se ha decidido usar el entorno de desarrollo Microsoft Visual Studio, pgAdmin y codificar en lenguaje de programación C# porque además de que forma parte del programa escolar, lo uso y trabajo con ello a diario en mi puesto laboral. Gracias a este proyecto espero obtener un mayor conocimiento de estas tecnologías para poder aplicarlo en mi puesto de trabajo en un futuro con éxito.



- Se utiliza Visual Studio 2022 Community porque es la edición gratuita.





Se ha decidido hacer una aplicación de Windows Forms: una capa sobre los controles estándar de Windows. Porque a pesar de que una aplicación WPF (Windows Presentation Foundation) permite un diseño mucho más detallado y potente, ya que, está construido desde cero y no depende de los controles estándar, requiere un mayor trabajo y aprendizaje.

Un ejemplo práctico: el uso GIFs (Graphics Interchange Format) en mi aplicación.

- Con Windows Forms el control “Picture Box” me permite este formato y únicamente es necesario cargar la imagen mediante propiedad.
- En WPF con el control “Image” no permitía el movimiento del GIF en ejecución.

WPF tiene muchos detalles, particularidades y requerimientos. Por ejemplo, la obligatoriedad de añadir un “Grid” en el formulario o esquema para poder manipular (cambiar la localización) los controles agregados.

Detalles que finalmente con el escaso tiempo y en mi opinión, ralentizaban el trabajo, pese a sus beneficios.

Tal como mencioné antes, opto por utilizar la herramienta de pgAdmin, para gestionar la base de datos PostgreSQL, debido a mi experiencia previa adquirida durante mis prácticas en empresa. Cuenta con una interfaz cómoda e intuitiva.

Se han creado las siguientes tablas:

```
-- CREATE
CREATE TABLE categorias (
    id_categoria serial PRIMARY KEY,
    descripcion VARCHAR (100) UNIQUE NOT NULL
);

CREATE TABLE soluciones (
    id_solucion serial PRIMARY KEY,
    descripcion VARCHAR (300) NOT NULL
);

CREATE TABLE preguntas (
    id_pregunta serial PRIMARY KEY,
    descripcion VARCHAR (600) NOT NULL,
    categoria_id integer NOT NULL REFERENCES categorias(id_categoria) ON DELETE CASCADE,
    solucion_id integer NOT NULL REFERENCES soluciones(id_solucion) ON DELETE CASCADE
);

CREATE TABLE contestaciones (
    id_contestacion serial PRIMARY KEY,
    descripcion VARCHAR (300) NOT NULL,
    pregunta_id integer NOT NULL REFERENCES preguntas(id_pregunta) ON DELETE CASCADE
);

CREATE TABLE datos (
    id_dato serial PRIMARY KEY,
    descripcion VARCHAR (2000) NOT NULL,
    categoria_id integer NOT NULL REFERENCES categorias(id_categoria)
);
```



Se utilizan las siguientes consultas en la aplicación, para obtener la información necesaria que asegura el correcto funcionamiento de esta:

Desde la clase “Acceso.cs”

```
-- obtenerPreguntasPorCategoria

string query = "SELECT * FROM PREGUNTAS WHERE CATEGORIA_ID = " + categoria + " ORDER BY RANDOM()";

-- obtenerSolucion

string query = "SELECT * FROM SOLUCIONES WHERE ID_SOLUCION = " + idSolucion;

-- obtenerContestaciones

string query = "SELECT * FROM CONTESTACIONES WHERE PREGUNTA_ID = " + idPregunta;

-- obtenerDatoCurioso

string query = "SELECT * FROM DATOS WHERE CATEGORIA_ID = " + categoria + " ORDER BY RANDOM() LIMIT 1";
```

Esta clase “Acceso.cs” establece la conexión y contiene variables necesarias para ello. Si cambia la base de datos, ubicación local, usuario, contraseña y/o puerto, estos valores deberán actualizarse.

```
34 referencias
public class Acceso
{
    private static string Host = "localhost";
    private static string User = "postgres";
    private static string DBname = "tfg";
    private static string Password = "superusuario";
    private static string Port = "5432";

    private static string connString =
        String.Format(
            "Server={0}; User Id={1}; Database={2}; Port={3}; Password={4}; SSLMode=Prefer",
            Host,
            User,
            DBname,
            Port,
            Password);
}
```

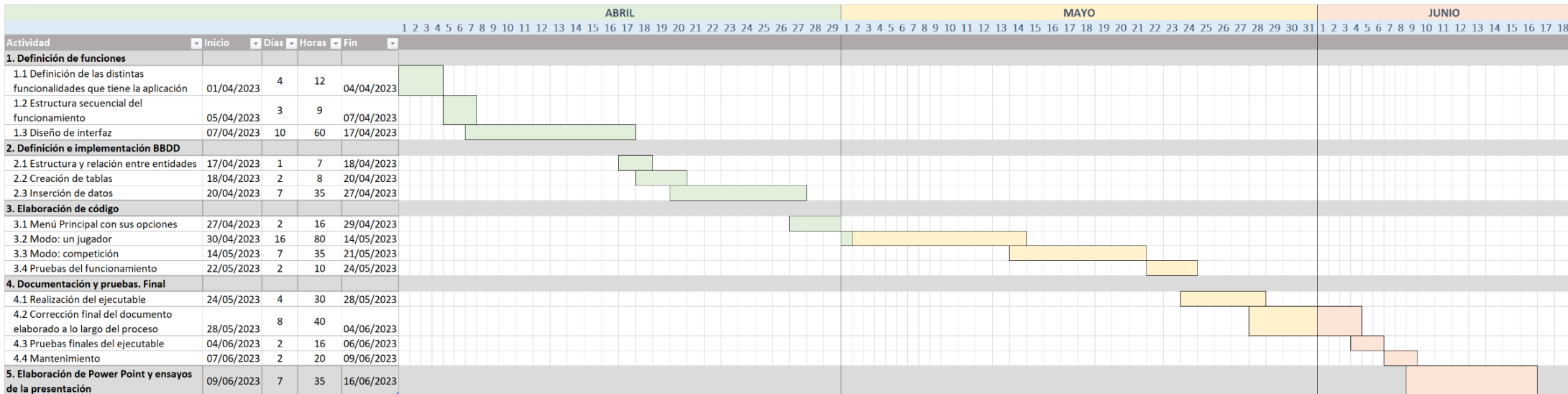
Es necesario que existan datos en la base de datos para el correcto funcionamiento de la aplicación, y para ello, adjunto un script que deberá ejecutarse “tfg_GrakDiana_scriptGlobal.sql”. Este script se encuentra dentro del directorio de la entrega de mi proyecto. Contiene la creación de las tablas y la inserción de todos los registros.

Total registros	
Categorías	6
Preguntas	230
Contestaciones	690
Soluciones	230
Datos	30



2.2. Actividades y cronología

2.2.1. Diagrama de Gantt

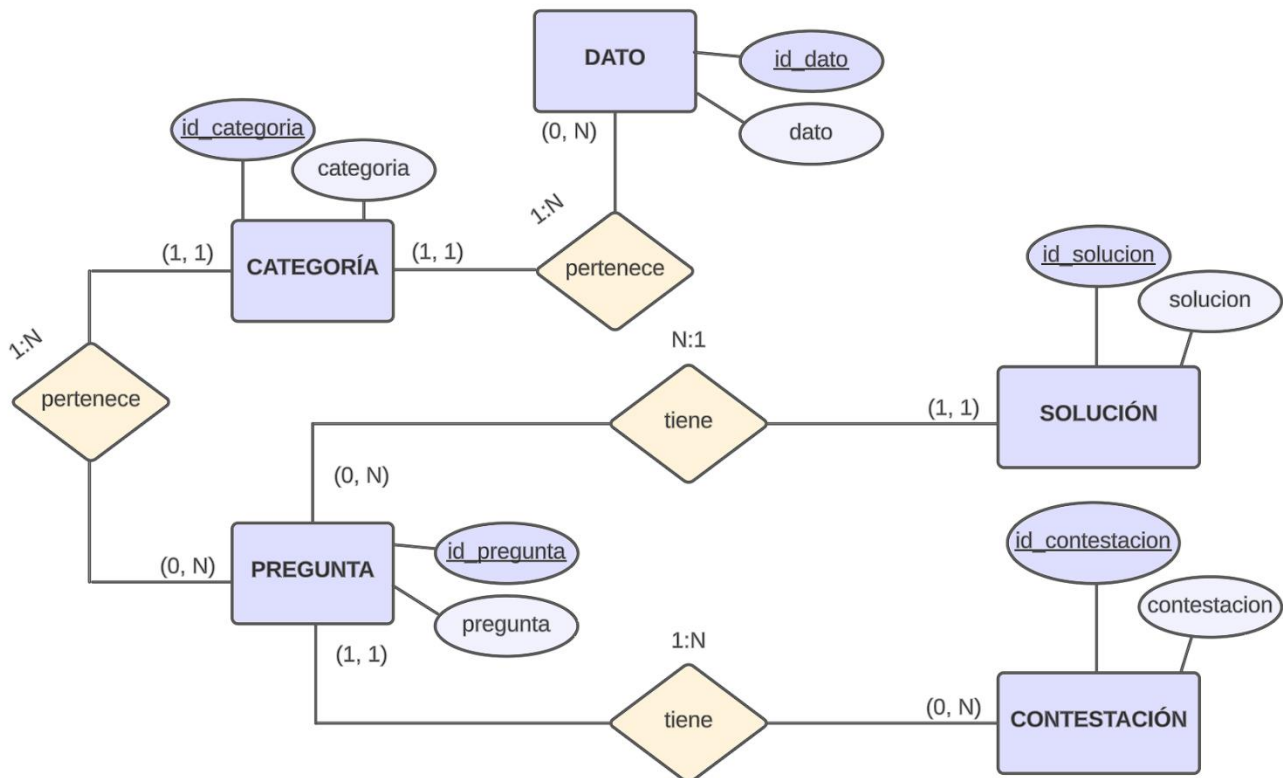




2.2.2. Documento de análisis y diseño

2.2.2.1. Diagramas de Bases de datos

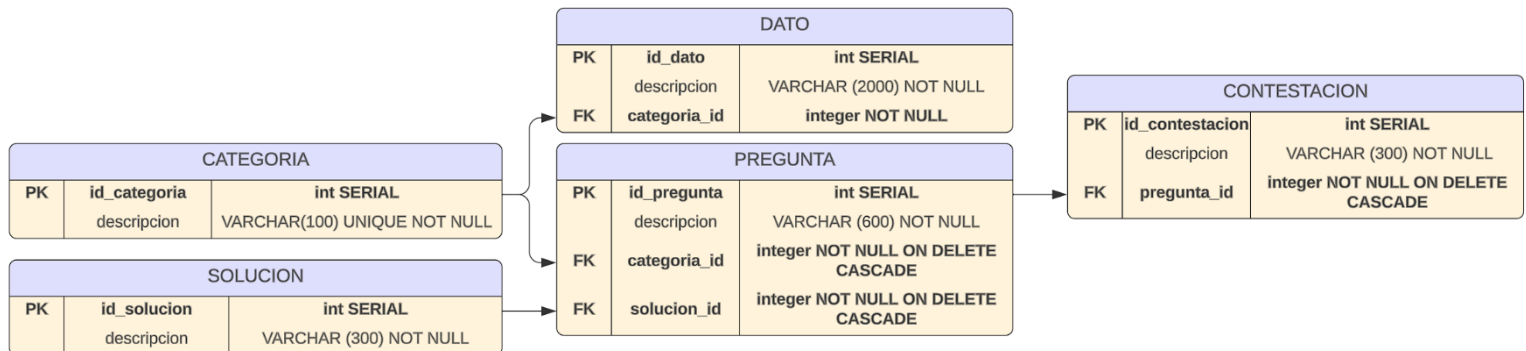
1. Entidad – Relación



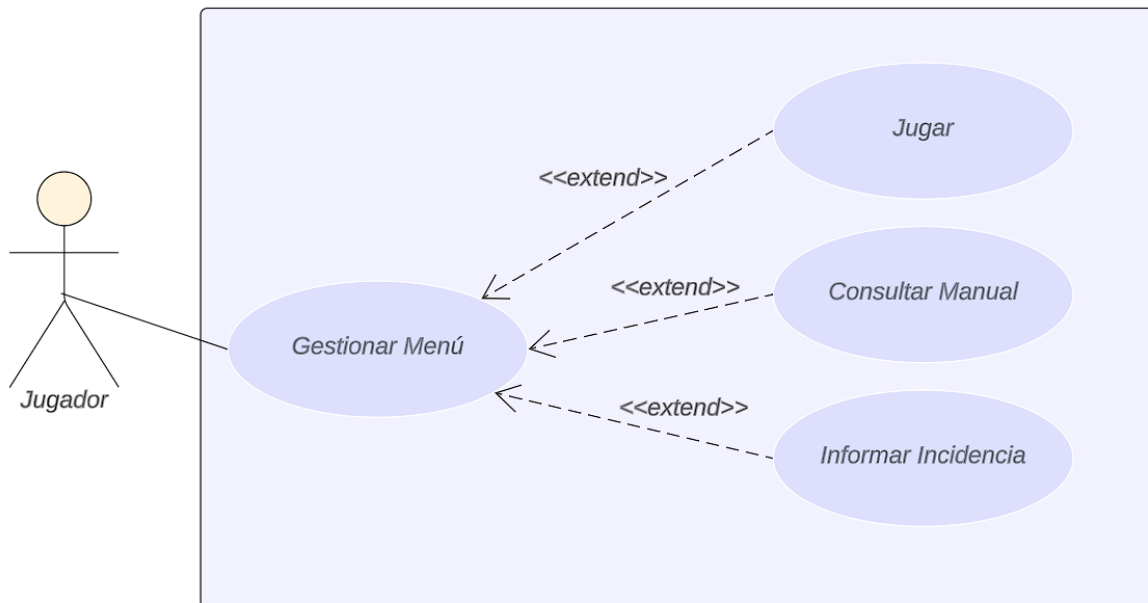
- Una categoría puede tener de 0 a N preguntas asociadas.
- Una pregunta debe estar asociada a una categoría.
- Una solución puede tener de 0 a N preguntas asociada.
- Una pregunta debe estar asociada a una solución.
- Una contestación debe estar asociada a una pregunta.
- Una pregunta puede tener de 0 a N contestaciones asociadas.
- Una categoría puede tener de 0 a N datos (curiosidades) asociados.
- Un dato debe estar asociado a una categoría.



2. Modelo Relacional



2.2.2.2. Diagrama de casos de uso



Nombre: Consultar Manual

ID: CU-1

Descripción: opción del Menú Principal, consulta y lectura del Manual de Usuario

Actores: Jugador (Usuario)

Precondiciones: El usuario ejecuta la aplicación y selecciona la opción del menú.

Curso normal del caso de uso:

1. El usuario realiza una lectura del Manual haciendo uso de scrollbar.



Postcondiciones: El usuario queda informado del funcionamiento de la aplicación.

Alternativa 1:

1. El usuario no realiza la lectura en parte o totalidad del Manual.
2. El usuario no recibe la información.

Nombre: Informar Incidencia

ID: CU-2

Descripción: opción del Menú Principal, notificar de cualquier anomalía o error en la aplicación.

Actores: Jugador (Usuario)

Precondiciones: El usuario ejecuta la aplicación y selecciona la opción del menú.

Curso normal del caso de uso:

1. El usuario escribe el mensaje dentro del TextBox-Multilínea.
2. El usuario selecciona el botón “Enviar”.
3. Se avisa al usuario de que la incidencia ha sido enviada correctamente.

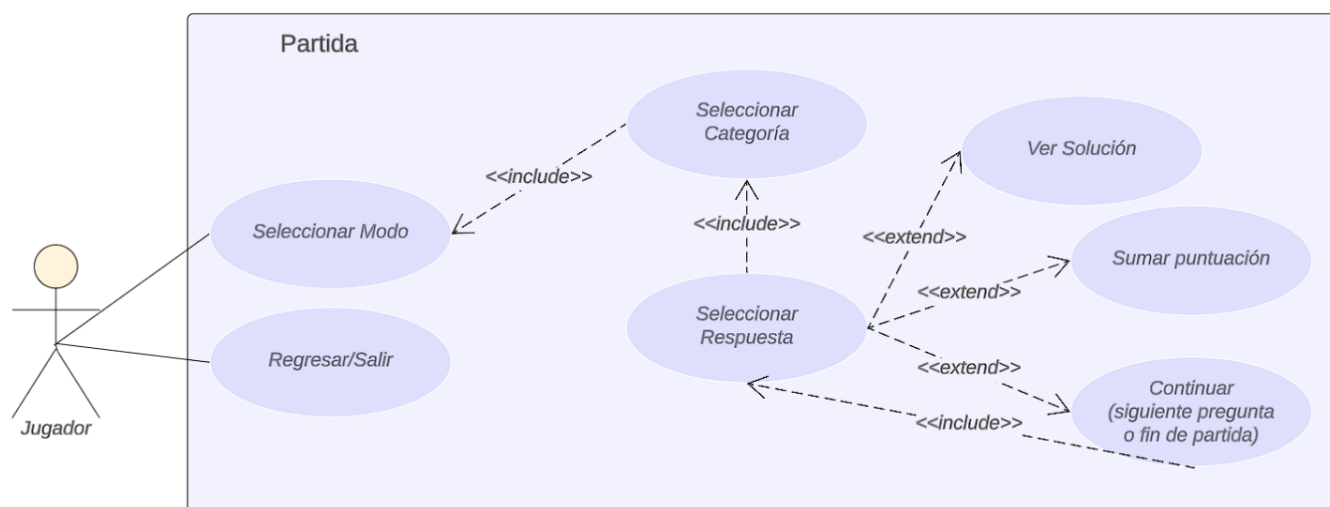
Postcondiciones: La incidencia es notificada al desarrollador de la aplicación y se intenta solventar.

Alternativa 1:

1. El usuario no hace uso de la opción.

Alternativa 2:

1. El usuario introduce el mensaje.
2. El usuario no pulsa el botón “Enviar”.
3. El usuario cambia de opción o sale de la aplicación.
4. El mensaje no se envía ni queda guardado.



Nombre: Jugar partida

ID: CU-3

Descripción: iniciar nueva partida, realizar secuencialmente las actividades de las que se compone.

Actores: Jugador (Usuario)

Precondiciones: El usuario ejecuta la aplicación, selecciona la opción del menú, elige un modo de juego y la categoría.

Curso normal del caso de uso:

1. El usuario selecciona el modo de juego.

2. El usuario selecciona una categoría.

3. El usuario selecciona la respuesta.

El paso número 3 se repite hasta que acaban las preguntas o finaliza la partida.

4. El usuario visualiza un dato curioso.

5. El usuario regresa al Menú Principal.

Postcondiciones: El jugador finaliza la partida y visualiza el resultado de la puntuación final y un dato curioso.

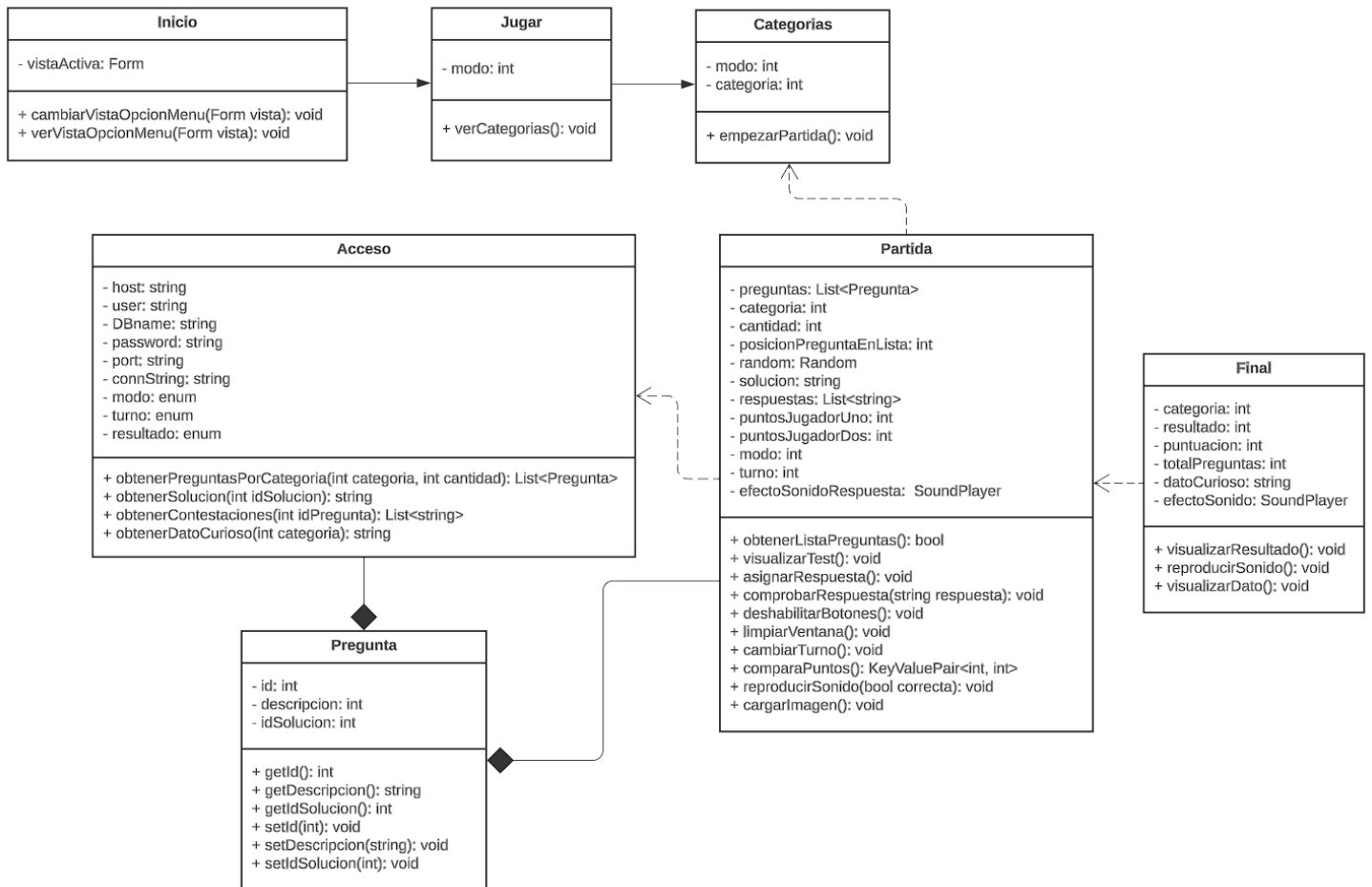
Alternativa 1:

1. El usuario regresa al Menú Principal antes de finalizar la partida.

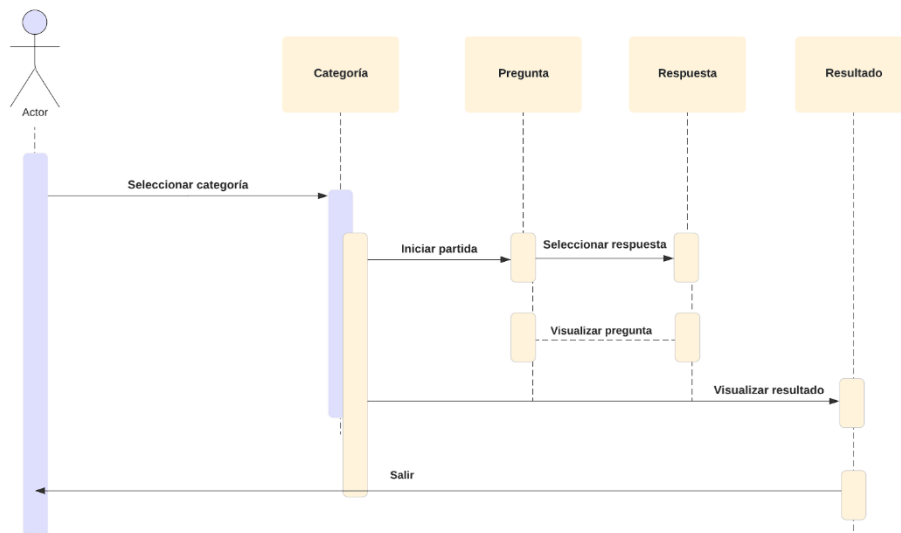
2. El usuario visualiza un mensaje informativo el cuál dicta que se perderá el progreso y la partida se da por finalizada.



2.2.2.3. Diagramas de clases



2.2.2.4. Diagramas de secuencia (para las actividades importantes)



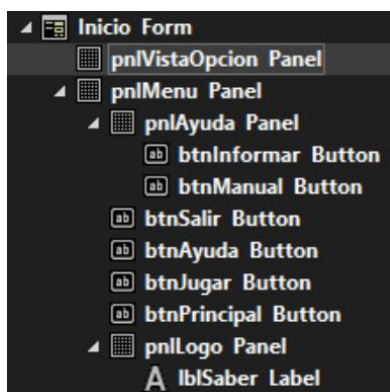


3. Diseño Interfaces gráficas

Lo primero que ve el usuario al ejecutar la aplicación es el Menú Principal. El Menú contiene cuatro opciones:

1. Principal. Ventana de bienvenida
2. Jugar. Iniciar partida
3. Ayuda. Contiene un submenú
 - Manual. Manual de Usuario
 - Informar. Permite contactar en caso de incidencia
4. Salir. Cerrar aplicación

La clase “Inicio.cs” representa el menú de la aplicación y tiene la siguiente estructura:



Las opciones del menú Principal, Jugar, Manual e Informar son clases distintas, cada una con su propia ventana. Estas ventanas se abren y se muestran en el panel “pnlVistaOpcion” de “Inicio.cs”.

La opción de Manual no se muestra en el panel, se ha optado visualizar el Manual de Usuario en una ventana independiente por la gran cantidad de texto y mayor comodidad de usuario durante su lectura.



SABER

Principal: Principal.cs

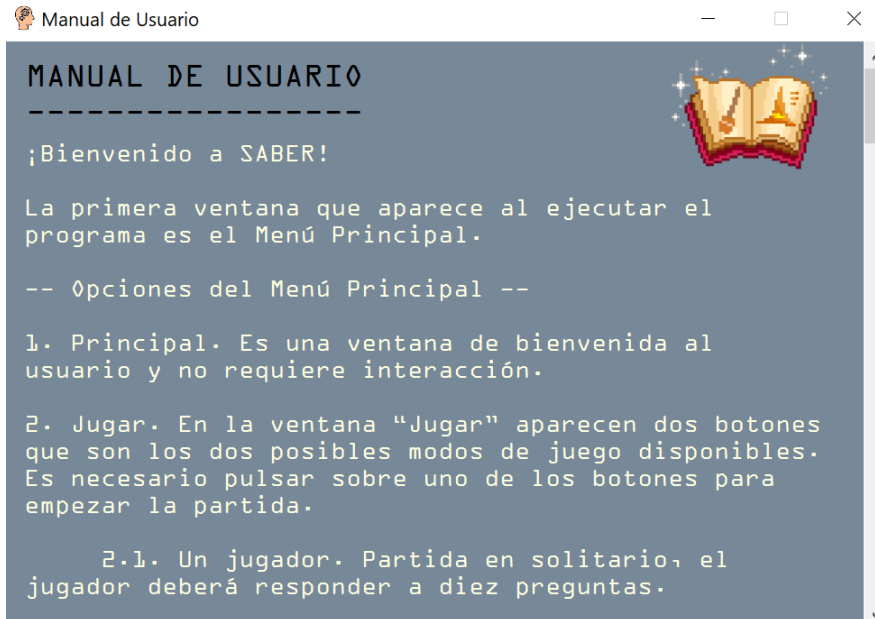


Jugar: Jugar.cs





Manual: Manual.cs



La ventana de Manual incluye un texto extenso, por lo tanto, se requería un scrollbar que pudiera abarcar toda la ventana. Para lograrlo, se agregó un panel con la propiedad “Dock” configurada como “Fill” y con la propiedad “AutoScroll” a true.

Informar: Informar.cs





El apartado [4.1.2. Manual de usuario](#) contiene una explicación detallada y precisa del funcional de estas opciones del Menú.

A continuación, la estructura y/o vista de las ventanas en el orden correspondiente al flujo normal cuando el usuario desea iniciar partida.

Para ello, el usuario deberá hacer clic sobre uno de los dos botones con los modos disponibles en Jugar. Esta acción desencadena la apertura del formulario de Categorías.

Categorías: Categorías.cs



El usuario deberá seleccionar una categoría haciendo clic sobre uno de los seis botones disponibles. A continuación, se abre la ventana de Partida.



Partida.cs



Aparece una pregunta, cuatro botones con respuestas, puntuación del jugador actual y turno. El usuario debe hacer clic sobre uno de los cuatro botones para responder a la pregunta. La ventana cambia:





SABER

- Se hace visible el botón de “Continuar” para seguir la partida con la siguiente pregunta.
- Se informa al jugador si su respuesta es Correcta o Incorrecta.
Si la respuesta es Correcta se suma + 1 en Puntos.
Si la respuesta es Incorrecta, aparece el botón “Ver Solución”.

El botón “Ver Solución” cambia la propiedad “BackColor” del botón con la respuesta correcta.

```
BackColor = Color.DarkSeaGreen;
```

Como se puede apreciar, al seleccionar una respuesta, se incrementa el grosor del botón correspondiente. De esta manera, se distingue con mayor claridad entre la opción correcta y la elegida por el usuario.



Una vez respondida la pregunta los botones con las respuestas quedan deshabilitados hasta la siguiente.

Este proceso se repite hasta que todas las preguntas hayan sido respondidas. Al finalizar la partida se abre la ventana



Final.cs



En el formulario Final.cs se muestra el resultado, que puede ser la victoria del Jugador1, la victoria del Jugador2 o un empate. Además, la puntuación obtenida y un dato curioso de premio por finalizar la partida.

Estilo

- Se usa la siguiente gama de colores



- La fuente **OCR-A**



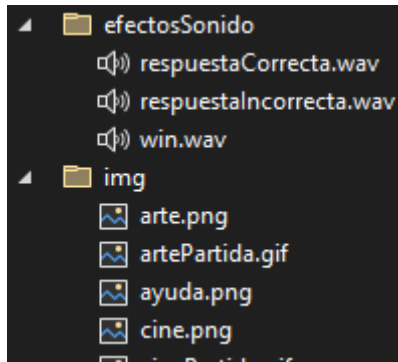
SABER

- Animaciones .gif con estilo similar o "Pixel Art".

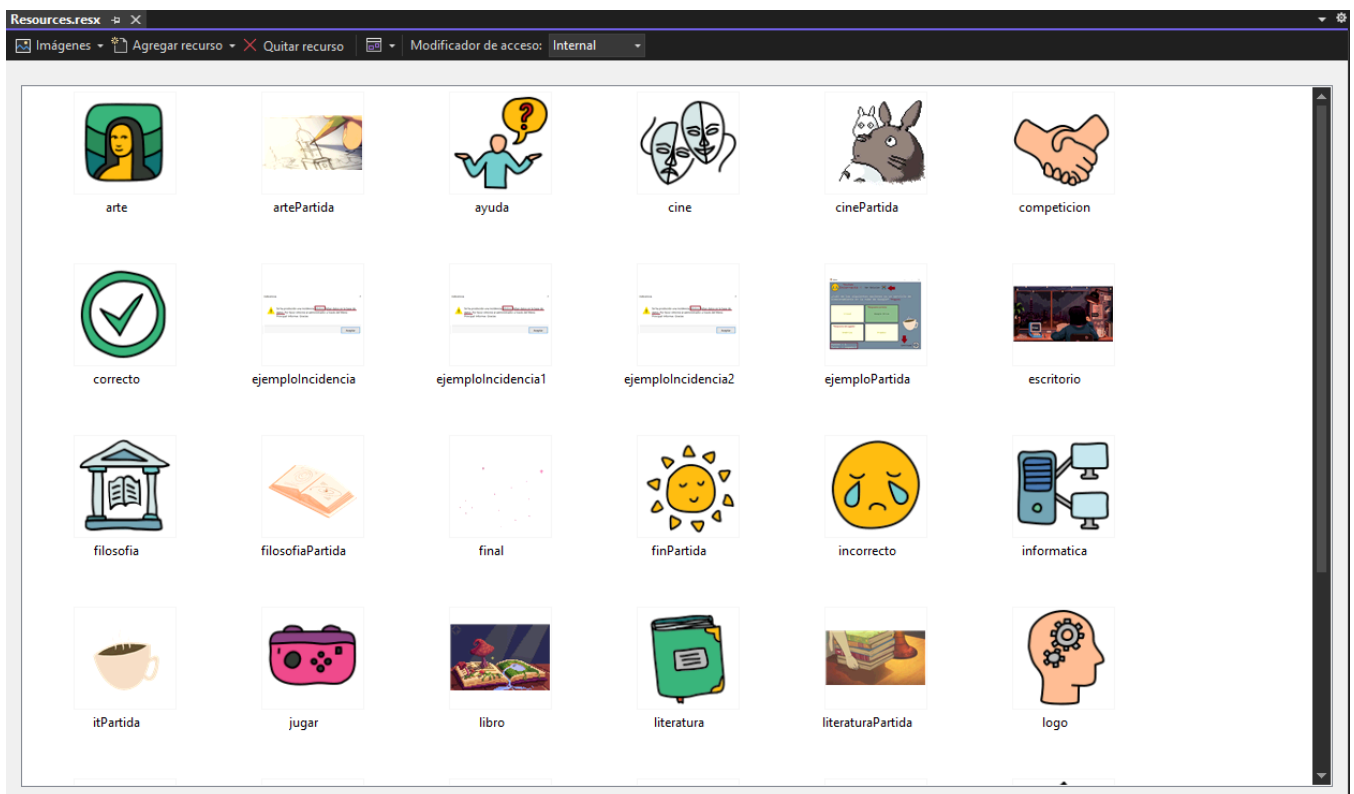
- Iconos gratuitos de iconos8.es estilo  Doodle

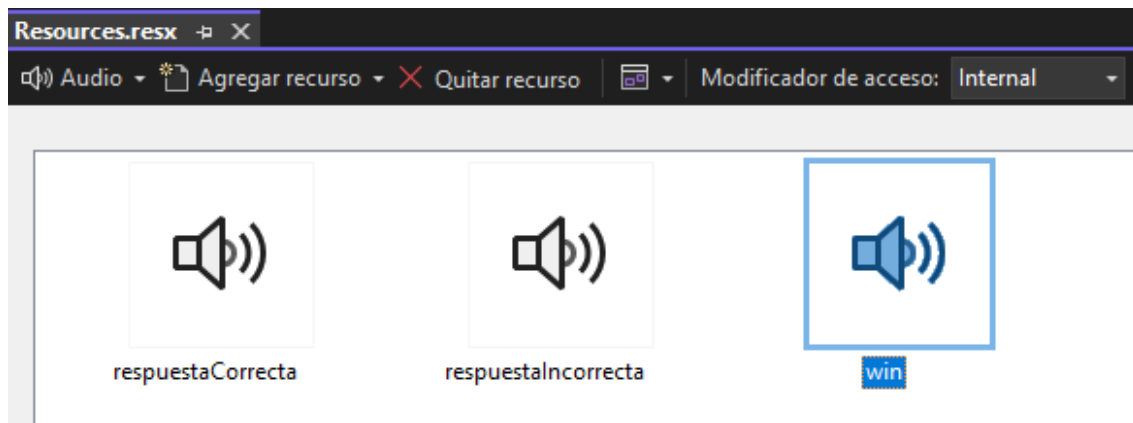
Además, hago uso de efectos de sonido .wav configurado en propiedades como recurso incrustado.

Dentro del proyecto, los efectos de sonido y los recursos de imagen de encuentran en los directorios



También están añadidos en Resources.resx

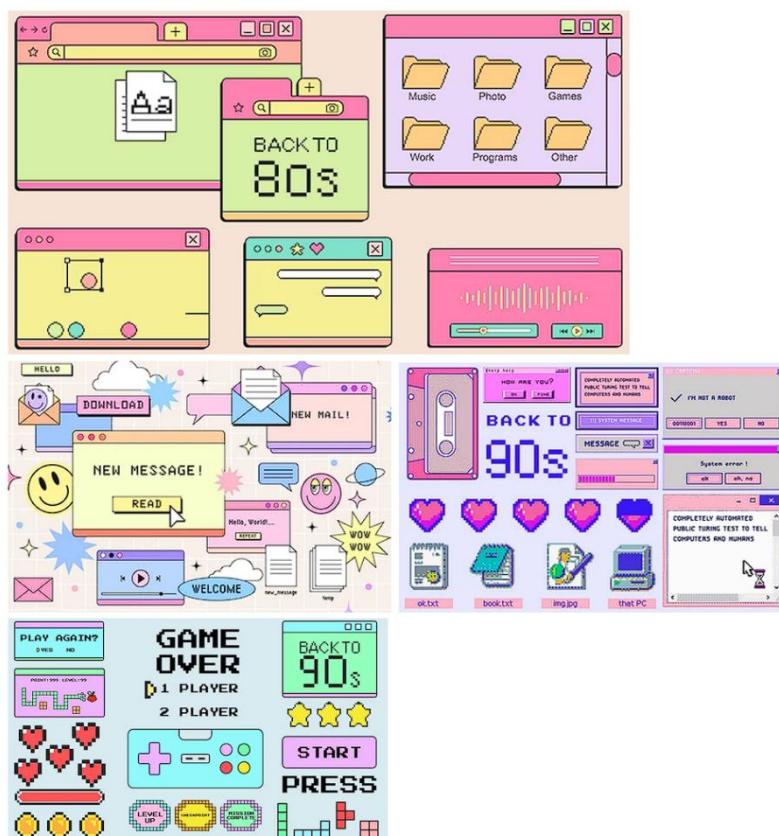




Los efectos de sonido se aplican cuando el usuario responde una pregunta o finaliza la partida.

Todas las ventanas tienen un tamaño mínimo que se establece como máximo, no se pueden redimensionar. Contienen pocos elementos y resultaba más estético un tamaño reducido.

Cada uno de estos elementos de diseño fue seleccionado meticulosamente para asegurar que se complementen entre sí. Mi intención e inspiración era lograr un estilo retro similar al del ejemplo, pero con tonos más suaves para la vista.





4. Doc. Cliente, Metodología y Recursos

4.1. Documentación para el cliente

4.1.1. Manual de instalación o configuración

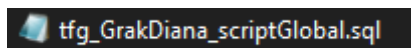
Para garantizar una correcta instalación y funcionamiento de la aplicación en el entorno local se deben seguir los siguientes pasos.

*Se requiere tener el sistema operativo Windows.

Descargar el archivo comprimido en formato .zip y descomprimirlo. El directorio contiene todos los elementos necesarios, incluyendo el proyecto con la solución, un ejecutable y un script, entre otros.

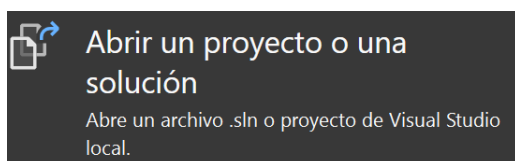
Dado que la base de datos es local, ejecutar el script “tfg_GrakDiana_scriptGlobal.sql”. Contiene la creación de tablas y la inserción de datos necesarios para el correcto funcionamiento de la aplicación.

Copiar completamente el contenido del archivo .sql, pegarlo y ejecutarlo en cualquier herramienta que gestione la base de datos PostgreSQL. Se recomienda usar la herramienta pgAdmin. Sin estos datos, la aplicación no podrá funcionar adecuadamente.



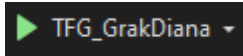
Dentro del proyecto existe la clase “Acceso.cs” que establece la conexión y contiene variables necesarias para ello. Si se cambia la base de datos, ubicación local, usuario, contraseña y/o puerto, estos valores deberán actualizarse.

Si se desea abrir el proyecto, es necesario tener instalado Visual Studio. Al abrirlo, se mostrará un menú con varias opciones. Seleccionar “Abrir un proyecto o una solución” e indicar la ubicación donde se encuentra la solución del proyecto.



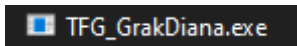


Esto abrirá la aplicación en Visual Studio, mostrando todo su código y diseño, Para ejecutar la aplicación, presionar el botón de “Play”.



En caso de que no se desee abrir el proyecto y se busque únicamente ejecutar el archivo ejecutable, basta con hacer doble clic sobre el archivo .exe que se encuentra en la siguiente ruta:

\\TFG_GrakDiana\bin\Debug\net6.0-windows\TFG_GrakDiana.exe



4.1.2. Manual de usuario

El Manual de Usuario está disponible en la aplicación, opción del Menú Principal -> Ayuda -> Manual. Contenido:

Manual de Usuario

¡Bienvenido a SABER!

La primera ventana que aparece al ejecutar el programa es el Menú Principal.
Opciones del Menú Principal:

1. Principal. Es una ventana de bienvenida al usuario y no requiere interacción.
2. Jugar. En la ventana “Jugar” aparecen dos botones que son los dos posibles modos de juego disponibles. Es necesario pulsar sobre uno de los botones para empezar la partida.

2.1. Un jugador. Partida en solitario, el jugador deberá responder a diez preguntas.

2.2. Modo competición. Para dos jugadores, cada jugador deberá responder a cinco preguntas sumando un total de diez preguntas entre ambos.

* Durante la partida, en la esquina inferior izquierda, se puede apreciar el turno asignado y la puntuación del jugador.



3. Ayuda. Contiene un submenú

3.1. Manual. En esta opción se puede visualizar el Manual de Usuario. Contiene información útil del funcionamiento y las características de la aplicación.

3.2. Informar. Esta opción permite contactar con el desarrollador de la aplicación por cualquier incidencia ocurrida durante la ejecución.

El usuario debe escribir el error detectado en la caja de texto "Aquí su incidencia" y pulsar en "Enviar". Si la operación ha salido bien el mensaje informativo cambiará informando al usuario de éxito.

* Cuando se produzca una incidencia en la aplicación, es probable que se muestre un mensaje informativo en pantalla indicando el motivo. Por favor, no olvide mencionarlo. De esta forma nos ayudaría a solucionar el problema lo más rápido posible. Agradecemos su cooperación.

Indicencia



Se ha producido una incidencia. Motivo: faltan datos en la base de datos. Por favor informe al administrador a través del Menú Principal: Informar. Gracias

Aceptar

*Advertencia:

- Si el usuario escribe el mensaje y no pulsa sobre el botón "Enviar", el mensaje no se envía.
- Si el usuario escribe el mensaje, no pulsa sobre el botón "Enviar" y cambia de opción en el menú, el mensaje no se envía ni se guarda.

4. Salir. Se cierra la aplicación.

Curso normal de una partida

1. Para empezar una partida el usuario deberá pulsar sobre "Jugar" en las opciones del menú.
2. Seleccionar un modo de juego. Después de esta acción, se abre la ventana "Categorías".



3. En “Categorías”, es necesario seleccionar una categoría, hacer clic en uno de los seis botones disponibles. La temática de las preguntas dependerá de la categoría escogida.

4. Empieza la partida. Se puede visualizar la pregunta, cuatro opciones, puntuación y turno. Para responder a la pregunta hacer clic (pulsar) en uno de los cuatro botones.

*Selecciona el botón (opción) que consideres respuesta correcta.

En la parte superior izquierda aparecerá un texto indicando si se ha acertado o no.

- Si la respuesta es incorrecta, es posible ver la solución haciendo clic en el botón “Ver Solución”. La respuesta correcta se resaltará en verde.
- Si la respuesta es correcta suma + 1 a la puntuación del jugador.

Para continuar con la siguiente pregunta, hacer clic en el botón “Continuar”. Si el modo de juego es competición cambiará el turno.

* Para una mayor claridad:

- El borde de la opción seleccionada por el jugador, aumenta de grosor.
- Los aciertos, errores y final de partida son acompañados de efectos de sonido.





Este proceso se repetirá hasta que se hayan respondido el total de diez preguntas.

*Si el jugador cierra la ventana durante la partida, saltará una advertencia informando la pérdida del progreso de la misma si cierra.

Al finalizar la partida, se abrirá una ventana nueva donde es posible visualizar el resultado, la puntuación y un dato curioso poco conocido de premio.

4.2. Metodología seguida

Se ha seguido una metodología en cascada, es decir, un enfoque secuencial para el desarrollo. Con las siguientes etapas de las que se compone:

1. Definición de requisitos. Identificar el funcionamiento de la aplicación.

Las opciones del Menú Principal, y las actividades que han de seguirse de forma secuencial para finalizar una partida: selección de categoría, responder número determinado de preguntas, etc.

Implica definir el tipo de preguntas que se utilizarán, los modos de juego, las categorías, el número de jugadores posibles y la plataforma.

2. Diseño. La creación de la interfaz de usuario.

3. Implementación. Se desarrolla la lógica, el código para el correcto funcionamiento planteado anteriormente.

4. Pruebas. Se prueban diferentes usos por parte del usuario y se comprueba que todo funciona correctamente.

5. Despliegue. Se elabora el ejecutable y se comprueba una vez más su correcta ejecución.

6. Mantenimiento. Existe un apartado de incidencias, donde el usuario puede comunicar cualquier error que exista en la aplicación. Esta información llegará al desarrollador, cuya labor será solventar el error además de mantener el juego con preguntas actualizadas y/o añadir contenido nuevo.



4.3. Recursos

4.3.1. Materiales

Para la realización del proyecto se ha hecho uso del siguiente material:

- Visual Studio 2022 Community
- pgAdmin
- PostgreSQL
- Navegador web Google Chrome

4.3.2. Infraestructura y equipamiento

- Ordenador de sobremesa: AMD Ryzen 5 PRO 3400G with Radeon Vega Graphics 3.70 GHz
- Impresora: Brother MFC-J5330DW
- Monitores:
 1. Dell S2721QS 27"
 2. HP 23"

Programas:

- Sistema operativo: Windows 10 Pro

4.3.3. Humanos

- Diana Grak. Encargada del desarrollo de la aplicación.



5. Resumen ejecutivo (inglés)

This project develops an entertainment application. It is a quiz game where the player must choose the correct answer among the four options presented on the screen.

The questions are random, and are classified into several categories. In principle, the following are considered:

- Cinema
- Video games
- Computers
- Art
- Philosophy
- Literature

It is possible to adapt the subject matter according to the customer's needs and end use on request.

On the other hand, there is also a second competitive game mode in which two players can participate. The winner will be the one with the highest score obtained through the correct questions. Another possible outcome can be a draw.

As a prize, at the end of the game, one of the curious, little-known facts stored in the database will be displayed on the screen.

It is a desktop application for Windows and developed in C#. It will use pgAdmin as a tool to manage and administer PostgreSQL.

6. Pruebas del sistema

Durante la documentación de apartados anteriores se han realizado pruebas del flujo normal de la aplicación. En este apartado consideraremos posibles escenarios inesperados que podrían ocurrir durante su uso.

1. Informar. El usuario **no introduce nada** y hace clic en el botón de “Enviar”.



La incidencia no puede estar vacía.

Aquí su incidencia

Enviar

The screenshot shows a web form with a blue border. At the top, a message in a monospace font says 'La incidencia no puede estar vacía.' Below this is a large yellow text area with the placeholder text 'Aquí su incidencia'. At the bottom of the form is a yellow button labeled 'Enviar' and a speech bubble icon.

Cambia el mensaje informativo para advertir al usuario de que el campo no puede enviarse vacío. Se considera este escenario controlado.

2. El usuario intenta abrir muchas ventanas para romper la aplicación, por ejemplo, **ejecutar más de una partida a la vez.**

Este escenario no es posible, debido a que en el código se controla mediante uso de ventanas modales `preguntas.ShowDialog();`.

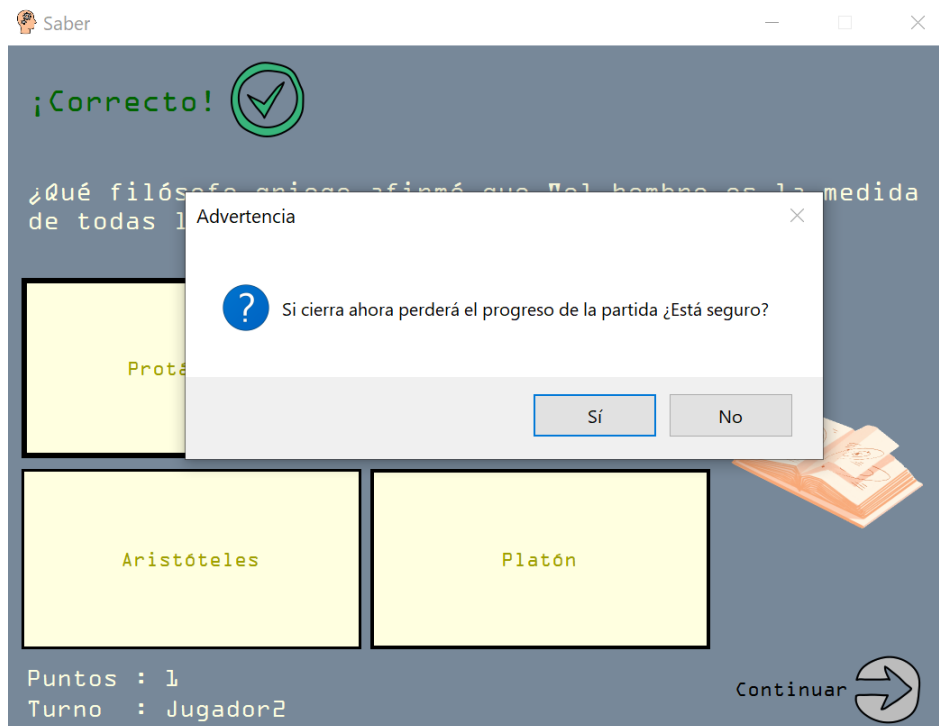
Obliga al usuario a cerrar la ventana actual para interactuar con la anterior.

3. El usuario intenta **cambiar la respuesta.**

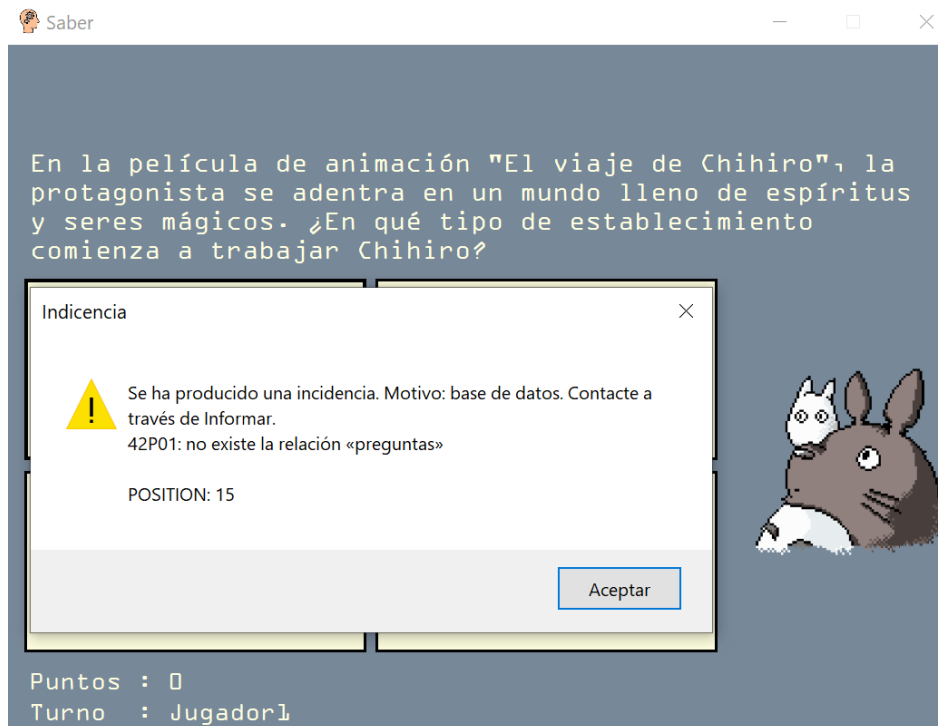
Después de que el usuario haya proporcionado una respuesta, los botones quedan desactivados `btn3.Enabled = false;`.

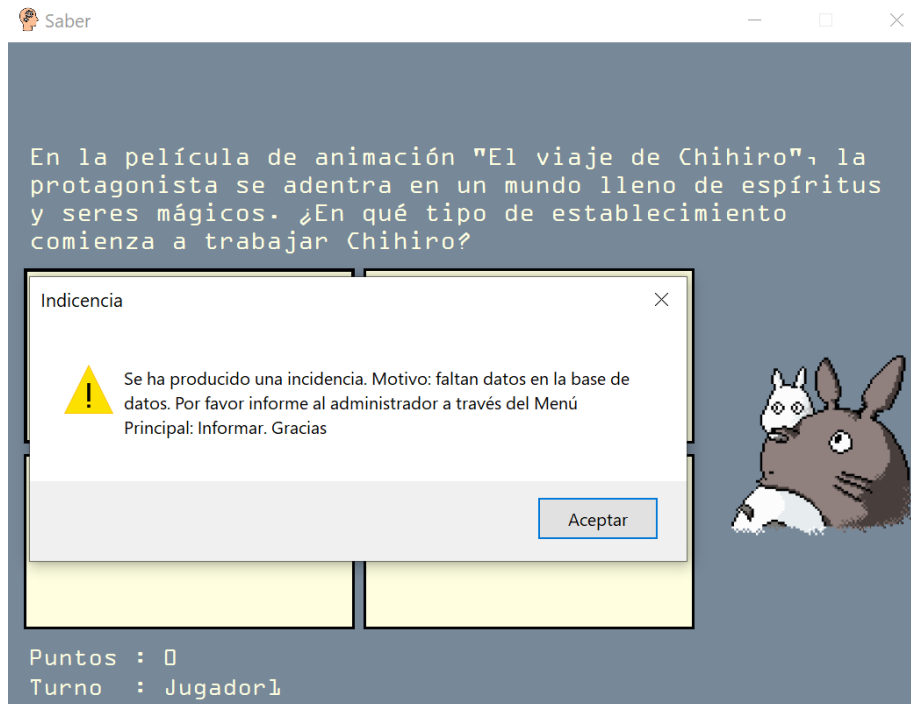
4. El usuario **cierra la ventana antes de acabar la partida.**

Aparece un mensaje informativo que advierte al usuario la pérdida del progreso.



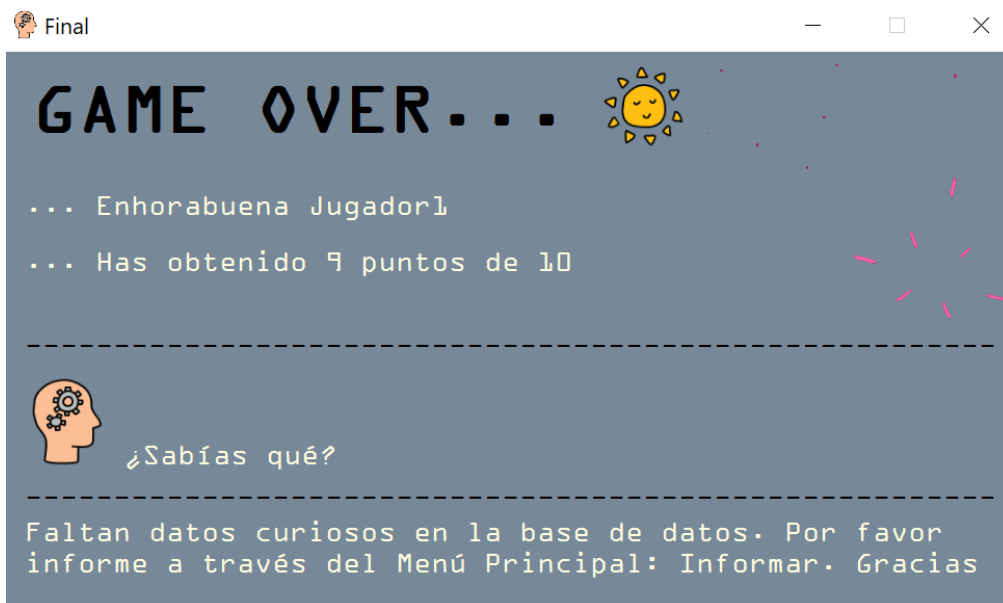
5. No existen, o **no se encuentran datos** en la base de datos local.





Aparecen diversos mensajes informativos, con el motivo de la incidencia pidiendo al usuario que informe al desarrollador. Cuya labor será contactar con el cliente y solventar el error.

6. **Faltan datos curiosos** en la base de datos. Se considera este escenario controlado.





7. Presupuesto

Esta aplicación ha sido realizada por una sola persona y la retribución estará basada conforme al Convenio colectivo estatal de empresas de servicios auxiliares de información, recepción, control de accesos y comprobación de instalaciones, con un salario bruto anual de 16.000 €.


En España, el año laboral tiene 238 días laborables. Esto significa que hay 238 días en los que los trabajadores pueden trabajar, por lo que el sueldo bruto diario del programador que realiza la aplicación es de $16.000/238 = 67$ €/día.

De acuerdo con el Diagrama de Gantt, se ha trabajado un total de 54 días hábiles, por lo que al tener un salario de 67€/día hace que el salario durante el tiempo en el que se realiza la aplicación sea de 3.618 €.

Si la aplicación sale a venta, el total del presupuesto antes de impuestos es el salario del programador en el tiempo invertido más un 50% de ese valor, en forma de beneficio.

$$\text{Presupuesto P645283} = 67 \frac{\text{€}}{\text{día}} * 54 \text{ días} * 1.5 = 5.427 \text{ €}$$


FACTURA



dg

FACTURAR A Calle Delicias 14	Nº DE FACTURA 645283
	FECHA 14.06.2023

DESCRIPCIÓN	IMPORTE
Creación de aplicación de escritorio para Windows	5.427,00
Subtotal	5.427,00
IVA 21.0%	1.139,67
TOTAL	6.566,67 €



CONDICIONES Y FORMA DE PAGO
El pago se efectuará en 15 días



8. Resultados objetivos y conclusiones

La finalidad de la aplicación es demostrar los conocimientos adquiridos durante el curso escolar, tanto en el centro educativo como el en el puesto laboral.

Se pretendía realizar una aplicación sencilla. Es cierto que el funcionamiento, las actividades o eventos secuenciales son lógicos e intuitivos y no requieren planteamientos complejos.

Lo más costoso del proyecto ha sido la documentación, la selección y desarrollo del contenido (las preguntas), y la interfaz con sus elementos visuales. Porque, en este caso, es de gran importancia que la aplicación sea interesante en contenido y visualmente atractiva para el usuario.

9. Bibliografía

[Quickstart: Use .NET \(C#\) to connect and query data in Azure Database for PostgreSQL - Single Server](#)

[Instrucción using: asegúrese del uso correcto de objetos descartables](#)

[KeyValuePair<TKey,TValue>](#)

[SoundPlayer](#)

[DialogResult](#)

[MessageBox](#)

[Random Class](#)

[Form Class](#)

[Form.ShowDialog](#)

[Form.Closing](#)

[Panel](#)

[Button Class](#)

[Label Class](#)



[PictureBox Class](#)

[TextBox Clase](#)

[LIMIT postgresql](#)

10. Anexos

- Iconos e Imágenes. Los iconos usados en la aplicación fueron obtenidos de [iconos8](#)

Se ofrecen de forma gratuita a cambio de atribución a los autores. Antes de utilizar las imágenes en producción o venta, se contactará con los artistas para solicitar su permiso y otorgarles reconocimiento en la aplicación.

- Los Diagramas se realizaron con la herramienta [Lucidchart](#)