

# Pasapalabra

**29/10/2023 - v1.0.0.0**

# Proyecto, diseño e implementación de sistema computacionales



# Índice

|                                       |           |
|---------------------------------------|-----------|
| <b>Índice.....</b>                    | <b>1</b>  |
| <b>Instrucción.....</b>               | <b>2</b>  |
| <b>Generalidades del sistema.....</b> | <b>3</b>  |
| <b>Requerimientos.....</b>            | <b>4</b>  |
| <b>Presentación.....</b>              | <b>5</b>  |
| <b>Lenguaje.....</b>                  | <b>6</b>  |
| <b>Variables.....</b>                 | <b>7</b>  |
| Módulo varUsoCom.....                 | 7         |
| Formulario juego.....                 | 7         |
| Formulario puntuaciones.....          | 8         |
| <b>Objetos.....</b>                   | <b>9</b>  |
| Proyecto.....                         | 9         |
| Módulo varUsoCom.....                 | 9         |
| Módulo funcJuego.....                 | 9         |
| Control botonSonido.....              | 11        |
| Formulario comoJugar.....             | 11        |
| Formulario puntuaciones.....          | 12        |
| Formulario inicio.....                | 12        |
| Formulario juego.....                 | 13        |
| <b>Elementos anexados.....</b>        | <b>15</b> |
| Carpeta raíz de los elementos.....    | 15        |
| Carpetas específicas.....             | 15        |
| Archivos de audio.....                | 15        |
| Archivos de imágenes.....             | 15        |
| Archivos de juego.....                | 15        |
| <b>Diagramas.....</b>                 | <b>16</b> |
| <b>Ampliaciones.....</b>              | <b>17</b> |
| <b>Glosario.....</b>                  | <b>18</b> |
| <b>Anexos.....</b>                    | <b>19</b> |
| <b>Bibliografía.....</b>              | <b>20</b> |

## Introducción

Pasapalabra es un juego de escritorio para la plataforma Windows, el mismo está basado en el juego del mismo nombre creado originalmente para televisión, específicamente en la prueba llamada “El Rosco”. El juego busca replicar la esencia del mismo pero en formato de un solo jugador, el jugador podrá elegir entre varias temáticas y al final obtendrá un puntaje basado en su resultado para poder así competir contra otros jugadores que ejecuten el sistema en la misma computadora.

Pasapalabra busca entretener al usuario y poner a prueba sus conocimientos, de manera tal que sea un desafío pero sin suponer un reto imposible para el mismo. El sistema también busca ser un reto para el programador del mismo, ya que pone a prueba los conocimientos en el lenguaje utilizado y en la lógica utilizada para la creación de los algoritmos y estructura del mismo sistema.

Se espera que el programa logre interesar y promover el conocimiento y la búsqueda del mismo en quien lo juega. Aunque también se prevé que, por cuestiones relacionadas al lenguaje de programación utilizado en el desarrollo del sistema, el apartado gráfico no reluzca de la mejor manera o de la esperada. Teniendo esto en cuenta se intentó optimizar y perfeccionar el sistema lo mejor posible para lograr sus objetivos.

## **Generalidades del sistema**

Sistema desarrollado por Lautaro Cáceres, 7mo 5ta orientación en programación de la Escuela de Educación Secundaria N°14.

Desarrollado en la materia Proyecto, diseño e implementación de sistema computacionales, bajo la tutoría de los profesores Gimena Avilés y Calbo Diego.

Inspirado en el programa de televisión "Pasapalabra".

## Requerimientos

Los requerimientos del sistema que serán instalados con el mismo son:

- Windows Installer 3.1
- .NET Framework 3.5 SP1

Los requerimientos de hardware necesarios son los siguientes:

- Procesador: Mínimo de 1.6 GHz, aunque se recomienda 2.2 GHz o superior.
- Memoria RAM: Mínimo de 384 MB, aunque se recomienda 1024 MB o más.
- Espacio en disco duro: Al menos 3 GB.
- Resolución de pantalla: Mínimo de 1024x768, aunque se recomienda 1280x1024.

Librerías dinámicas de Windows necesarias:

- wmp.dll

## **Presentación**

Pasapalabras es un juego desarrollado para fomentar el aprendizaje y la cultura general del usuario que se enfrente a los desafíos del mismo. Esto se lleva a cabo de forma didáctica en la cual la dinámica es adivinar, mediante los conocimientos del jugador y las pistas que ofrece el sistema, que palabra se encuentra atrás de una letra del abecedario indicada en un roscó.

También se tiene en cuenta que los usuarios abarcan un amplio rango de edades, culturas y comunidades, por lo que se les dispondrá a los mismos un abanico de temáticas para elegir. La posibilidad de elegir una temática le permitirá al usuario demostrar su conocimiento en ciertos tópicos o adentrarse a un tema que desconoce para luego profundizarlo por su cuenta para poder superar el roscó.

## Lenguaje

El sistema fue desarrollado en el lenguaje VB.NET con el entorno de desarrollo Visual Basic 2008 Express Edition, para la creación del mismo se utilizó el framework de Microsoft .NET en su versión 3.5. También se utilizó la biblioteca dinámica de Windows wmp.dll para el manejo de sonido.

## Variables

### Módulo varUsoCom

**paths:** Variable pública de solo lectura del tipo String. Variable que contiene la ruta de ejecución del programa.

**indLetra:** Variable pública del tipo Byte. Almacena el índice de la letra actual del roscó.

**tam:** Variable pública del tipo Byte. Almacena el tamaño que tendrán las letras en roscó.

**ang:** Variable pública de solo lectura del tipo Double. Almacena la distancia en grados entre las letras del roscó.

**ancho:** Variable pública de solo lectura del tipo Integer. Almacena el ancho en píxeles de la pantalla.

**alto:** Variable pública de solo lectura del tipo Integer. Almacena el alto en píxeles de la pantalla.

**hayVolumen:** Variable pública del tipo Boolean. Registra el estado del volumen del sonido, de modo que si se le asigna el estado falso se apagará el volumen, en caso contrario se encenderá.

**rutaImagenes:** Variable pública de solo lectura del tipo String. Almacena la ruta de las imágenes del juego.

**rutaTextoJuego:** Variable pública de solo lectura del tipo String. Almacena la ruta de las temáticas del juego.

**rutaAudio:** Variable pública de solo lectura del tipo String. Almacena la ruta de los archivos de audio del juego.

**rutaPuntaje:** Variable pública de solo lectura del tipo String. Almacena la ruta del puntaje del juego.

**imgsLetras:** Vector público de Bytes. Almacena el estado de la letra en su posición equivalente en abc. Los estados pueden ser 4: 1-No preguntado(Azul) 2-Preguntado(Amarillo) 3-Fallido(Rojo) 4-Correcto(Verde).

**iniciado:** Variable pública del tipo Boolean. Registra el estado del juego. Si su estado es verdadero entonces indica que el juego ha sido iniciado, en caso contrario el juego ha finalizado o no se ha iniciado.

**abc:** Vector público de solo lectura de Strings. Almacena las letras del abecedario que se utilizarán para el roscó.

**pista:** Variable pública del tipo String. Indica el nombre del archivo de la pista que va a ser reproducida por el reproductor de sonido principal.

**segundos:** Variable pública del tipo Integer. Almacena el tiempo de juego restante en segundos.

**reproduciendo:** Variable pública del tipo Boolean. Almacena el estado del reproductor, de modo que si es falso indica que no está reproduciendo música.

### Formulario juego

**tematica:** Variable privada del tipo Byte. Almacena la temática seleccionada por el jugador.



**preguntas:** Vector privado de Strings. Almacena las preguntas que se mostrarán en el sector de preguntas del juego.

**respuestas:** Vector privado de Strings. Almacena las respuestas que deberá adivinar el jugador.

**bx:** Variable privada del tipo Integer. Almacena la posición X del próximo botón de temática.

**by:** Variable privada del tipo Integer. Almacena la posición Y del próximo botón de temática.

**btam:** Variable privada del tipo Byte. Almacena el tamaño de los botones de temática.

## Formulario puntuaciones

**tamTabla:** Variable privada del tipo Integer. Almacena el tamaño en ancho de la tabla.

# Objetos

## Proyecto

### Objetos:

**varUsoCom:** Módulo. Contiene variables y objetos utilizadas en común por todo el sistema.

**funcJuego:** Módulo. Contiene funciones utilizadas por todos los formularios del sistema.

**botonSonido:** Control de Usuario. Control creado para manejar las preferencias del usuario con respecto a la música del juego.

**inicio:** Formulario. Formulario principal, contiene el acceso a los demás formularios del sistema.

**juego:** Formulario. Es el juego en sí y contiene funcionalidades para llevar a cabo el mismo.

**puntuaciones:** Formulario. Utilizado para mostrar los puntajes de los usuarios que registraron los mismos.

**comoJugar:** Formulario. Le explica al usuario cómo jugar al "Pasapalabra".

## Módulo varUsoCom

### Objetos:

**formJuego:** Objeto público de la clase Form. Objeto usado para tener referencia al formulario principal de juego.

**letras:** Vector público de objetos Rectangle. Cada objeto almacena la posición y tamaño de una letra del roscó. El índice de cada letra corresponde al vector abc.

**letrasAzul:** Vector público de objetos Image. Cada objeto almacena la imagen de una letra azul del roscó. El índice de cada letra corresponde al vector abc.

**letrasAmarillo:** Vector público de objetos Image. Cada objeto almacena la imagen de una letra amarilla del roscó. El índice de cada letra corresponde al vector abc.

**letrasRojo:** Vector público de objetos Image. Cada objeto almacena la imagen de una letra roja del roscó. El índice de cada letra corresponde al vector abc.

**letrasVerde:** Vector público de objetos Image. Cada objeto almacena la imagen de una letra verde del roscó. El índice de cada letra corresponde al vector abc.

**wmp:** Objeto público de la clase WindowsMediaPlayer. Objeto usado para reproducir la música principal del sistema.

**sonidos:** Objeto público de la clase WindowsMediaPlayer. Objeto usado para reproducir los sonidos del juego.

## Módulo funcJuego

### Funciones:

**acomodarPos:** Función pública que devuelve una lista de Strings. Lee el archivo de puntajes calcula los puntajes de cada jugador y los acomoda en una lista de manera decreciente de acuerdo al total de puntos obtenidos por el mismo y los devuelve. Utilizada

para acomodar los puntajes que serán mandados a tabla de puntuaciones. Es llamada por la función `datosTabla` en el formulario de de puntuaciones.

**augmentarIndice:** Función pública sin retorno. Comprueba la variable `indLetra` para tomar decisiones como, llamar a la función `vuelta`; aumentar en 1 la misma variable o hacer y llamarse de nuevo a sí misma, según su valor. Utilizada para avanzar una letra en el roscó. Es llamada por los botones `pasapalabra` y `enviar` del formulario `juego`.

**cambiarPista:** Función pública sin retorno. Recibe como parámetro un String por valor que es la pista a cambiar. Cambia la pista de audio actual por una nueva.

**cargarImagenes:** Función pública sin retorno. Carga a los arreglos de imágenes de letras(`letrasAzul`, `letrasAmarillo`, `letrasRojo`, `letrasVerde`) las imágenes de las mismas letras con el color que corresponda en cada uno, obtenidas de la carpeta `imagenes` de los recursos del juego. Llamada por la función `iniciarJuego` del formulario `juego`.

**cargarTexto:** Función pública sin retorno. Recibe como parámetros un arreglo de Strings por referencia que es donde se devolverán las preguntas cargadas desde archivo; un arreglo de String por referencia que es donde se devolverán las respuestas cargadas desde archivo; y un String pasado por valor que indica el nombre de la temática de donde se deben obtener las preguntas y las respuestas. Lee y carga en los dos arreglos pasados por parámetros las preguntas y respuestas obtenidas del archivo de temática con el mismo nombre que la temática pasada por parámetro desde la carpeta `temáticas` de los recursos del juego. Llamada por la función `iniciarJuego` del formulario `juego`.

**comprobarFinal:** Función pública sin retorno. Recorre las letras del roscó y si todas fueron respondidas llama a la función `final`, de otra manera termina la función. Llamada en las funciones `vuelta` y `comprobarRespuesta`.

**comprobarRespuesta:** Función pública sin retorno. Recibe como parámetros un String pasado por valor que representa la pregunta del usuario y un vector de Strings pasado por referencia que contiene las preguntas del juego. Compara la respuesta del usuario con la respuesta de la letra actual; si la respuesta es correcta cambia la imagen de la letra a verde y reproduce un sonido de acierto, en caso contrario cambia la imagen de la letra a rojo y reproduce un sonido de desacierto. Llamada por el botón `enviar` en el formulario `juego`.

**contarResultado:** Función pública que devuelve un String. Utilizada para contar los puntos obtenidos por el jugador en el juegos, al finalizar devuelve los datos contabilizados de manera ordenada y separada por coma. Llamada en la función `final`.

**datosTabla:** Función pública que devuelve un vector de objetos `ListViewItem`. Recibe como parámetro una lista de Strings por valor. Convierte la lista pasada por parámetro en un vector de `ListViewItem` que puede ser interpretado como filas por una tabla. Utilizada para obtener los resultados de `acomodarPos` y transformarlos en filas para la tabla del formulario de puntuaciones. Llamado en el formulario `puntuaciones`.

**dibujarCorrespondientes:** Función pública sin retorno. Recibe por parámetro un objeto del tipo `Graphics` pasado por referencia. Dibuja cada letra del roscó en su posición y con su color correspondiente. Llamada por el evento `paint` del formulario `juego` y los botones `enviar` y `pasapalabra` del mismo.

**escribirPregunta:** Función pública sin retorno. Recibe como parámetros dos objetos `Label` por referencia, un vector de Strings por referencia y un objeto `TextBox` por referencia. Escribe en una `Label` la respuesta correspondiente a la letra actual del roscó desde el vector pasado por parámetro, en la otra `Label` escribe la letra actual y limpia el campo pasado por parámetro. Llamada por los botones `pasapalabra` y `enviar` en el formulario `juego`.

**establecerJuego:** Función pública sin retorno. Recibe cómo parámetro un objeto Form por referencia. Asigna a formJuego el formulario de juego. Llamada en el formulario juego.

**final:** Función pública sin retorno. Cierra el formulario de juego y abre el formulario principal, obtiene el puntaje del jugador y los escribe al final del archivo de puntajes. Utilizado para terminar el juego. Llamada en la función comprobarFinal.

**iniciarLetras:** Función pública sin retorno. Le asigna a cada letra el estado sin preguntar. Llamada en la función iniciarJuego del formulario juego.

**pausar:** Función pública sin retorno. Baja el volumen de los reproductores de audio a 0. Llamada en la función verificarVolumen del control botonSonido.

**reproducir:** Función pública sin retorno. Reproduce o aumenta el volumen a 100 de la pista de audio principal según corresponda. Llamada en la función verificarVolumen del control botonSonido.

**vuelta:** Función pública sin retorno. Verifica si se llegó al final del rosco y si es así re asigna los estados de las letras que no se han respondido a no preguntado. Utilizado para reiniciar el rosco. Llamada en la función aumentarIndice.

## Control botonSonido

### Objetos:

**sonido:** Objeto de la clase Button. Utilizado para obtener la preferencia del usuario sobre la música del juego.

### Funciones:

**sonido\_Click:** Función privada del evento click del botón sonido. Utilizada para llamar a la función verificarVolumen. Invierte el estado de hayVolumen y luego llama a la función verificarVolumen.

**New:** Constructor del control. Inicializa los componentes de este control.

**verificarVolumen:** Función privada sin retorno. Verifica la variable hayVolumen para determinar si reproduce música o no, y también que imagen llevará el botón sonido. Llamada en la función sonido\_Click y en el constructor.

## Formulario comoJugar

### Objetos:

**volverInicio:** Objeto de la clase PictureBox. Utilizado cómo botón para volver al formulario inicio.

**sonido:** Objeto de la clase botonSonido. Utilizado para manejar las preferencias de sonido del usuario.

### Funciones:

**volverInicio\_Click:** Función privada del evento click del objeto volverInicio. Utilizada para volver al formulario principal.

**comoJugar\_Load:** Función privada del evento load de este formulario. Carga las imágenes de cada control y define las posiciones y tamaños de los mismos.

## Formulario puntuaciones

### Objetos:

**volverInicio:** Objeto de la clase PictureBox. Utilizado como botón para volver al formulario inicio.

**sonido:** Objeto de la clase botonSonido. Utilizado para manejar las preferencias de sonido del usuario.

**tablaPuntaje:** Objeto de la clase ListView. Utilizado como tabla para mostrar las puntuaciones de todos los jugadores que registraron su puntaje en el sistema.

**posicion:** Objeto de la clase ColumnHeader. Columna Posición de tablaPuntaje.

**nombre:** Objeto de la clase ColumnHeader. Columna Nombre de tablaPuntaje.

**correctas:** Objeto de la clase ColumnHeader. Columna Correctas de tablaPuntaje.

**incorrectas:** Objeto de la clase ColumnHeader. Columna Incorrectas de tablaPuntaje.

**sinResponder:** Objeto de la clase ColumnHeader. Columna Sin responder de tablaPuntaje.

**tiempo:** Objeto de la clase ColumnHeader. Columna Tiempo de tablaPuntaje.

### Funciones:

**volverInicio\_Click:** Función privada del evento click del objeto volverInicio. Utilizada para volver al formulario principal.

**puntuaciones\_Load:** Función privada del evento load de este formulario. Carga las imágenes de cada control y define las posiciones y tamaños de los mismos. Carga los puntajes del archivo del mismo nombre a la tabla tablaPuntaje.

## Formulario inicio

### Objetos:

**salir:** Objeto de la clase PictureBox. Utilizado como botón para salir del sistema.

**jugar:** Objeto de la clase PictureBox. Utilizado como botón para iniciar el juego.

**comoJugar:** Objeto de la clase PictureBox. Utilizado como botón para ingresar al formulario de comoJugar.

**puntajes:** Objeto de la clase PictureBox. Utilizado como botón para ingresar al formulario puntuaciones.

**sonido:** Objeto de la clase botonSonido. Utilizado para obtener la preferencia del usuario sobre la música del juego.

**logo:** Objeto de la clase PictureBox. Utilizado contenedor de la imagen del logo.

### Funciones:

**Salir\_Click:** Función privada del evento click del objeto salir. Utilizada para salir del sistema.

**Jugar\_Click:** Función privada del evento click del objeto jugar. Utilizada para abrir el formulario del juego.

**comoJugar\_Click:** Función privada del evento click del objeto comoJugar. Utilizada para ingresar al formulario comoJugar.

**puntajes\_Click:** Función privada del evento click del objeto puntajes. Utilizada para ingresar al formulario puntuaciones.

**inicio\_Load:** Función privada del evento load de este formulario. Carga las imágenes de cada control y define las posiciones y tamaños de los mismos.

**puntajes\_MouseEnter:** Función privada del evento mouseEnter del objeto puntajes. Cambia la imagen del botón.

**puntajes\_MouseLeave:** Función privada del evento mouseLeave del objeto puntajes. Cambia la imagen del botón.

**comoJugar\_MouseEnter:** Función privada del evento mouseEnter del objeto comoJugar. Cambia la imagen del botón.

**comoJugar\_MouseLeave:** Función privada del evento mouseLeave del objeto comoJugar. Cambia la imagen del botón.

**salir\_MouseEnter:** Función privada del evento mouseEnter del objeto salir. Cambia la imagen del botón.

**salir\_MouseLeave:** Función privada del evento mouseLeave del objeto salir. Cambia la imagen del botón.

**jugar\_MouseEnter:** Función privada del evento mouseEnter del objeto jugar. Cambia la imagen del botón.

**jugar\_MouseLeave:** Función privada del evento mouseLeave del objeto jugar. Cambia la imagen del botón.

## Formulario juego

### Objetos:

**sonido:** Objeto de la clase botonSonido. Utilizado para obtener la preferencia del usuario sobre la música del juego.

**preguntaDeJuego:** Objeto de la clase Label. Utilizado como título de la selección de temáticas.

**preparado:** Objeto de la clase PictureBox. Utilizado para mostrar el gif de "Preparados, listos, comencemos".

**tiempo:** Objeto de la clase Label. Utilizado para mostrarle al jugador el tiempo restante en pantalla.

**guiaPregunta:** Objeto de la clase Label. Utilizado para indicar donde se mostrará la pista.

**guiaRespuesta:** Objeto de la clase Label. Utilizado para indicar donde deberá responder el usuario,

**letraActual:** Objeto de la clase Label. Utilizado para mostrar la letra actual del roscó.

**Respuesta:** Objeto de la clase TextBox. Utilizado para recibir la respuesta del usuario.

**rendirse:** Objeto de la clase Button. Utilizado como botón para rendirse y terminar el juego.

**pasapalabra:** Objeto de la clase Button. Utilizado como botón para que el usuario pueda pasar a la siguiente letra del roscó.

**enviar:** Objeto de la clase Button. Utilizado como botón para que el usuario pueda mandar su respuesta.

**Pregunta:** Objeto de la clase Label. Utilizado para mostrar la pista de la letra actual del roscó.

**reloj:** Objeto de la clase Timer. Utilizado para contar el tiempo de juego.

**g:** Objeto de la clase Graphics. Utilizado para el manejo del apartado gráfico del formulario.

**listaNombres:** Objeto de la clase ReadOnlyCollection. Colección de solo lectura de Strings. Utilizado para guardar los nombres de los directorios de las temáticas utilizadas en el juego.

**tematica**[número de la temática]: Objetos de la clase Button. Conjunto de botones correspondientes a la cantidad de temáticas más el botón "Aleatorio". Utilizados para permitirle al usuario elegir la temática a jugar del roscó.

### **Funciones:**

**juego\_Load:** Función privada del evento load de este formulario. Carga las imágenes de cada control y define las posiciones y tamaños de los mismos. Crea los botones de las temáticas.

**preguntaDeJuego\_click:** Función privada del evento click de los botones de temática. Utilizada en los botones para detectar la opción elegida por el usuario e iniciar el juego.

**iniciarJuego:** Función privada sin retorno. Utilizada para inicializar los elementos del juego del roscó. Enciende el Timer(objeto reloj). Llamada en la función preguntaDeJuego\_click.

**reloj\_Tick:** Función privada del evento tick del objeto reloj. Utilizada para contar los segundos que transcurren al iniciar el juego.

**rendirse\_Click:** Función privada del evento click del objeto rendirse. Utilizada por el usuario para salir del juego.

**enviar\_Click:** Función privada del evento click del objeto enviar. Utilizada por el usuario para comprobar la respuesta escrita en el objeto del mismo nombre y escribir la siguiente pregunta.

**Respuesta\_KeyUp:** Función privada del evento keyUp del objeto Respuesta. Utilizada para detectar si el usuario pulsó la tecla enter, en ese caso llama a la función enviar\_Click.

**pasapalabra\_Click:** Función privada del evento click del objeto pasapalabra. Utilizada por el jugador para pasar a la siguiente letra del roscó.

**juego\_Paint:** Función privada del evento paint de este formulario. Llamada cada vez que se redibuja el formulario, es utilizada para dibujar las letras con la función dibujarCorrespondientes.

## Elementos anexados

### Carpeta raíz de los elementos

Pasapalabras

### Carpetas específicas

Pasapalabras

- |\_ **audio**
- |\_ **imagenes**
- |\_ **juego**

### Archivos de audio

Pasapalabras

- |\_ **audio**
  - |\_ **Acierto.wav**: Sonido de acierto en el rosco.
  - |\_ **Desacierto.wav**: Sonido de desacierto en el rosco.
  - |\_ **PI.wav**: Canción de fondo del sistema.

### Archivos de imágenes

Pasapalabras

- |\_ **imagenes**
  - |\_ **botones**: Directorio donde se guardan las imágenes de los botones.
  - |\_ **letras**: Directorio donde se guardan las imágenes de las letras.
  - |\_ **audio-on.png**: Imágen del botón de audio cuando hay sonido.
  - |\_ **empezar.gif**: Gif que se muestra antes de iniciar una sesión de juego.
  - |\_ **fondo.png**: Imágen de fondo del sistema.
  - |\_ **icono.ico**: Icono del sistema.
  - |\_ **icono.png**: Imágen del icono del sistema.
  - |\_ **inst.png**: Imágen con las instrucciones de juego.
  - |\_ **logo.png**: Imágen de logo presentado en el formulario inicio.
  - |\_ **no-audio.png**: Imágen del botón de audio cuando no hay sonido.

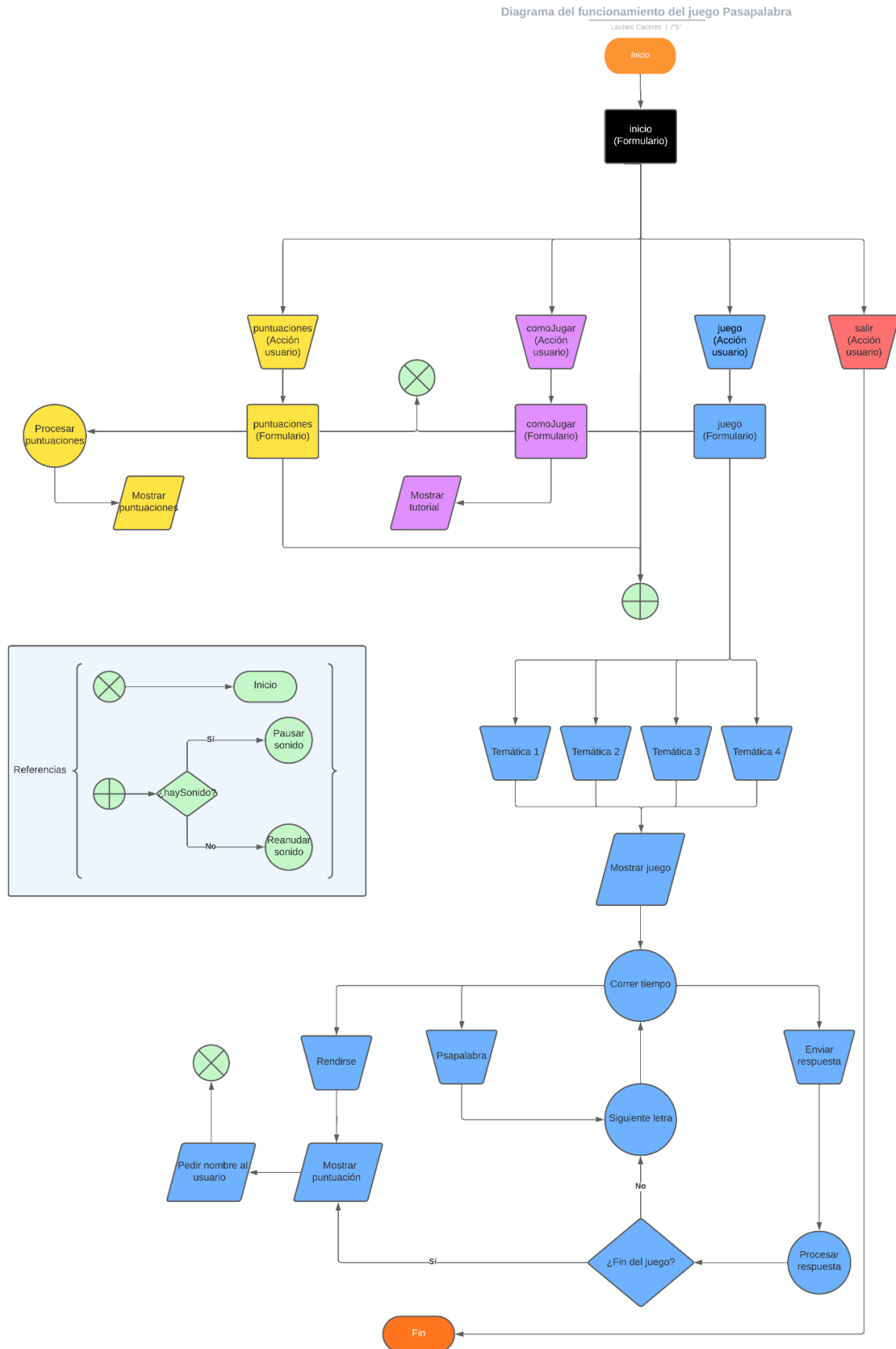
### Archivos de juego

Pasapalabras

- |\_ **juego**
  - |\_ **puntaje**: Directorio donde se guarda el archivo de puntajes del juego.
    - |\_ **puntaje.txt**: Archivo de puntaje del juego.
  - |\_ **tematicas**: Directorio donde se guardan las temáticas del juego.



## Diagramas



## Ampliaciones

Se tiene pensado para el futuro de la aplicación las siguientes características:

- **Optimización de código:** Se pretende reducir la cantidad de líneas de código y mejorar la eficiencia de los algoritmos utilizados para reducir significativamente el trabajo de la CPU.
- **Adaptabilidad de pantalla:** Se buscará mejorar los cálculos para la presentación del juego en pantalla completa, de forma que se vea en iguales proporciones en cada monitor.
- **Apartado visual:** El apartado visual es importante para la adaptabilidad y atención del usuario al juego, si bien el formato actual no está mal se puede mejorar.
- **Interfaz de opciones:** Para mejorar la experiencia se plantea una interfaz para manejar aspectos del juego. Como por ejemplo; el volumen de , el tamaño de la ventana o la cantidad de tiempo de juego.
- **Interfaz de creación de temáticas:** Con esta decisión se busca facilitar la creación de temáticas tanto para los usuarios cómo para el desarrollador.

## **Glosario**

## **Anexos**

## **Bibliografía**