



# MIND GAME

## DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

**Autores:** Sosa Ludueña Diego  
Sleiman Mohamad  
Choquevilca Gustavo

**Versión del Documento:** 1.0.1

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>1.1. Propósito</b>	<b>2</b>
<b>1.2. Ámbito del sistema</b>	<b>2</b>
<b>1.3. Definiciones, Acrónimos y abreviaturas</b>	<b>2</b>
<b>2. Descripción del Sistema</b>	<b>2</b>
<b>3. Requisitos comunes de las interfaces</b>	<b>3</b>
<b>3.1. Interfaces de Usuario</b>	<b>3</b>
<b>3.2. Interfaces de Hardware</b>	<b>3</b>
<b>4. Usuarios del Sistema</b>	<b>3</b>
<b>5. Requerimientos Funcionales y No Funcionales</b>	<b>3</b>
<b>5.1. Requerimientos Funcionales</b>	<b>3</b>
<b>5.2. Requerimientos No Funcionales</b>	<b>4</b>
<b>6. Diagramas UML</b>	<b>5</b>
<b>6.1. Diagrama de Actividad de Mind Game</b>	<b>5</b>
<b>6.2. Diagrama de Caso de Uso para Mind Game</b>	<b>6</b>
<b>6.3. Diagrama de Secuencia para Mind Game</b>	<b>7</b>
<b>7. Casos de Uso y Matriz de Trazabilidad</b>	<b>9</b>
<b>7.1. Casos de Uso</b>	<b>9</b>
<b>7.2. Matriz de Trazabilidad</b>	<b>10</b>
<b>8. Casos de Prueba del Sistema y Actualizado de Matriz de Trazabilidad</b>	<b>11</b>
<b>8.1. Casos de Prueba de Requerimientos Funcionales</b>	<b>11</b>
<b>8.2. Casos de Prueba de Requerimientos No Funcionales</b>	<b>13</b>
<b>8.3. Actualización de Matriz de Trazabilidad</b>	<b>14</b>
<b>9. Casos de prueba que actúan como Smoke Testing</b>	<b>15</b>
<b>10. Arquitectura del Sistema</b>	<b>16</b>

## **1. Introducción**

En el presente documento se explica y analiza la especificación de requerimientos para el proyecto *Mind Game*, desarrollado para todo el público en general. Se adopta la guía de requerimiento de software de la IEEE.

### **1.1. Propósito**

Este documento tiene como propósito dar a conocer el funcionamiento general del proyecto *Mind Game* y además plasmar de forma clara y concisa las necesidades del cliente en términos del software que se va a realizar. La documentación de los requisitos será guía para validar e inspeccionar la construcción del software. Este documento está dirigido al cliente y al equipo de desarrollo. Adicionalmente también puede ser usado por los usuarios que utilizaran el software y que necesiten definir nuevos requerimientos.

### **1.2 Ámbito del sistema**

*Mind Game* es una aplicación orientada al sector de entretenimiento. Podrá ser usada por cualquier usuario que desee entretenerse y agilizar su mente. Por lo tanto nuestro objetivo es que esta aplicación pueda ser usada como una herramienta educativa para cualquier persona, pero especialmente para los niños.

### **1.3 Definiciones, Acrónimos y abreviaturas**

1. MG: Mind Game
2. UML: lenguaje de Modelado Unificado
3. IEEE: Institute of Electrical and Electronics Engineers
4. ID: Identificador
5. SRS: Software Requirements Specifications
6. JAR: Java Archive
7. CP: Casos de prueba
8. CU: Casos de uso
9. REQ F: Requerimientos Funcional
10. REQ NO F : Requerimientos no Funcional

## **2. Descripción del Sistema**

El sistema *Mind Game* consistirá en una aplicación en donde el usuario debe capaz de ordenar una secuencia de números aleatorios que va del 1 al 9 que se presentarán en una interfaz gráfica. Si la matriz de números del sistema aparece marcada de color naranja los números se deberán ordenar de mayor a menor y si la matriz de números aparece marcada de color azul será un indicativo de ordenar los números de menor a mayor.

Por otro lado, existirá una barra de progreso de tiempo o temporizador en segundos en donde su valor disminuirá a medida que el usuario esté jugando la partida . Esta barra de progreso de temporización restará una cantidad de segundos si el usuario selecciona un número incorrecto, agregando así mayor dificultad al juego. También incorporaremos un indicador de niveles

completados que se incrementará en uno cada vez que se completa una matriz de números ya que será una de las características que se tendrá en cuenta a la hora de generar la tabla de puntuaciones al final de cada partida. Dicha tabla contendrá los siguientes campos: ID de usuario, aciertos realizados, teclas pulsadas incorrectamente y niveles completados (dado por la cantidad de matrices numéricas realizadas correctamente por cada usuario). El campo teclas pulsadas incorrectamente se producirá cuando ocurra lo mencionado en el ejemplo anterior, este producirá un incremento en 1 en el campo teclas pulsadas incorrectamente.

Por otra parte se puede resaltar que a medida que se pulsa una tecla correctamente, el juego MG no dejará que se utilice nuevamente dicha tecla hasta que se finalice con la matriz que se está resolviendo. El sistema tendrá dos botones al comienzo los cuales serán INICIAR y SALIR ya sea para dar inicio a una partida o salir de la partida. Para uso de esta aplicación es necesario un ordenador.

### **3. Requisitos comunes de las interfaces**

#### **3.1 Interfaces de Usuario**

La interfaz de usuario es el medio con que el usuario puede comunicarse con un ordenador. En este caso tendremos una ventana la cual mostrará información necesaria para poder interactuar con la misma (como botones, cuadros de textos, iconos, mensajes, etc).

#### **3.2 Interfaces de Hardware**

La interfaces de hardware son los dispositivos utilizados para ingresar, procesar y entregar los datos. En este caso tenemos la pantalla del monitor, donde el software de Mind Game deberá mostrar toda la interfaz del juego al usuario a través de la pantalla del monitor. El mouse, donde en el software de Mind Game será necesario el uso del ratón y los botones del mismo. Este se utilizará para pulsar los botones que proporciona la aplicación.

### **4. Usuarios del Sistema**

Como ya habíamos mencionado más arriba, los usuarios que utilizaran el sistema Mind Game, serán el público en general ya que formará parte de una aplicación de entretenimiento. Precisamente esta destinada para ser utilizada por los niños.

### **5. Requerimientos Funcionales y No Funcionales**

En esta sección se presenta una lista que contiene todo los requerimientos funcionales y no funcionales de Mind Game, la lista contiene cada requisito con un identificador y una breve descripción de cada uno.

#### **5.1 Requerimientos Funcionales**

Son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema.

**REQ F1:** Cada vez que se inicie una partida en el juego MG se deberá mostrar un temporizador en segundos indicativo del tiempo para realizar todas las secuencias posibles (es decir se deberá mostrar la barra de progreso de tiempo).

**REQ F2:** Cada vez que se genere una matriz de números el juego MG, deberá mostrar en la interfaz un panel de números con sus casilleros. Cada casillero tendrá un número del 1 al 9 aleatoriamente y no se podrán repetir entre ellos. Por cada vez que se complete esta matriz se deberá mostrar una matriz de números aleatoria nueva con el indicador de color opuesto a la anterior.

**REQ F3:** El juego MG deberá contener una señal indicativa que le indique al usuario de qué forma realizar el ordenamiento de los números (de menor a mayor o mayor a menor), esta señal será visual mediante los colores azul y naranja respectivamente.

**REQ F4:** El juego MG deberá tener un indicador que muestre la cantidad de aciertos y desaciertos que se producen durante el desarrollo del juego y además un indicador de nivel, que indica la cantidad de matrices completas (es decir, las veces que completó la secuencia de números).

**REQ F5:** Cada vez que se ejecuta el juego MG o se inicia una partida nueva se deberá permitir al usuario poder ingresar su ID. Si un usuario quiere ingresar al juego con ID usado previamente, el sistema no permitirá el ingreso al nuevo jugador hasta que ingrese un ID usuario que sea distinto a los ya registrados.

**REQ F6:** El juego MG al finalizar la partida, no permitirá que se siga completando la matriz numérica ni permitirá la creación de otra matriz.

**REQ F7:** El juego MG al finalizar la partida, deberá mostrar una lista que contendrá el ID usuario, los aciertos, los errores de las teclas presionadas de forma incorrectas y el nivel (también llamada tabla de puntuaciones).

**REQ F8:** El juego MG no le permitirá al usuario utilizar un número de la matriz que se marcó correctamente, es decir este casillero se deshabilitará una vez que se presionó correctamente.

**REQ F9:** El juego MG deberá ser capaz de no deshabilitar el casillero si el jugador pulsa un número en el orden incorrecto.

**REQ F10:** El juego MG descontará un segundo de tiempo a la barra de progreso de tiempo cuando se pulsa un número en el orden incorrecto.

**REQ F11:** La interfaz MG contendrá un botón inicio que permitirá al comenzar el juego una vez ingresado un ID usuario válido, y también contendrá un botón salir que le permitirá al usuario salir de la aplicación cuando desee.

## 5.2 Requerimientos No Funcionales

Son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema.

**REQ NO F1:** Para poder operar el Juego se necesitará 100 MB de RAM

**REQ NO F2:** Para poder operar el Juego se necesitará un espacio de disco de 10 MB

**REQ NO F3:** Para poder operar el Juego se necesitará un procesador mínimo Pentium 2 a 266 MHz

**REQ NO F4:** La partida tendrá un tiempo límite inicial de 60 segundos.

**REQ NO F5:** El sistema deberá ser compatible con diferentes sistemas operativos.

**REQ NO F6:** La aplicación deberá ser sencilla e intuitiva, para que usuarios con bajos conocimientos tecnológicos puedan usarla sin que les suponga un esfuerzo adicional.

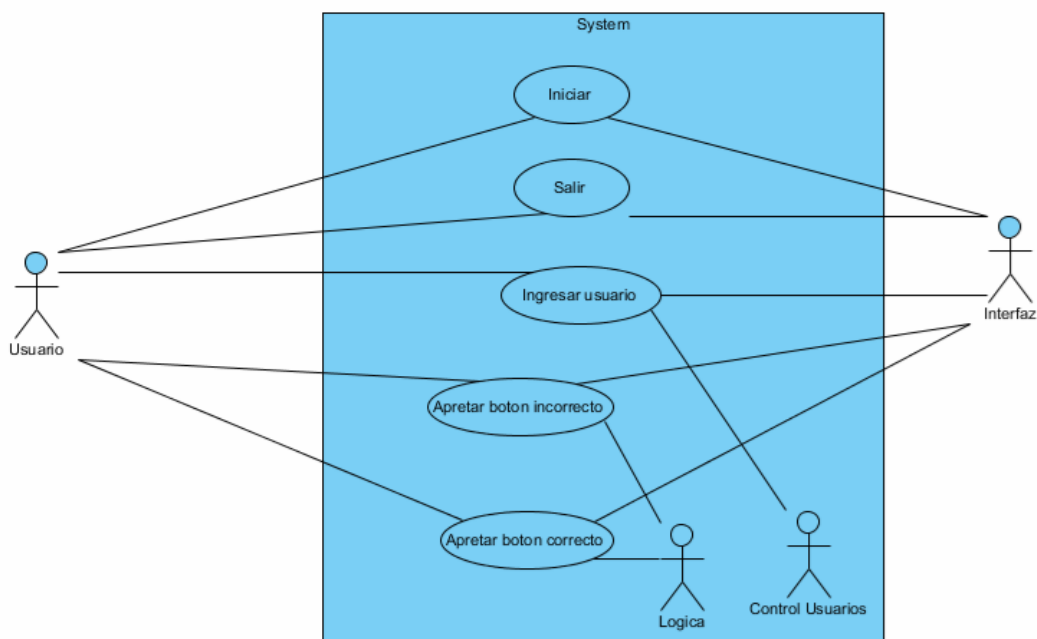
**REQ NO F7:** Cuando el usuario presiona el botón salir, la aplicación deberá liberar los recursos que estuviese utilizando para que el sistema operativo se los pueda asignar a otros procesos

**REQ NO F8:** El sistema deberá mostrar los textos legibles para el usuario.

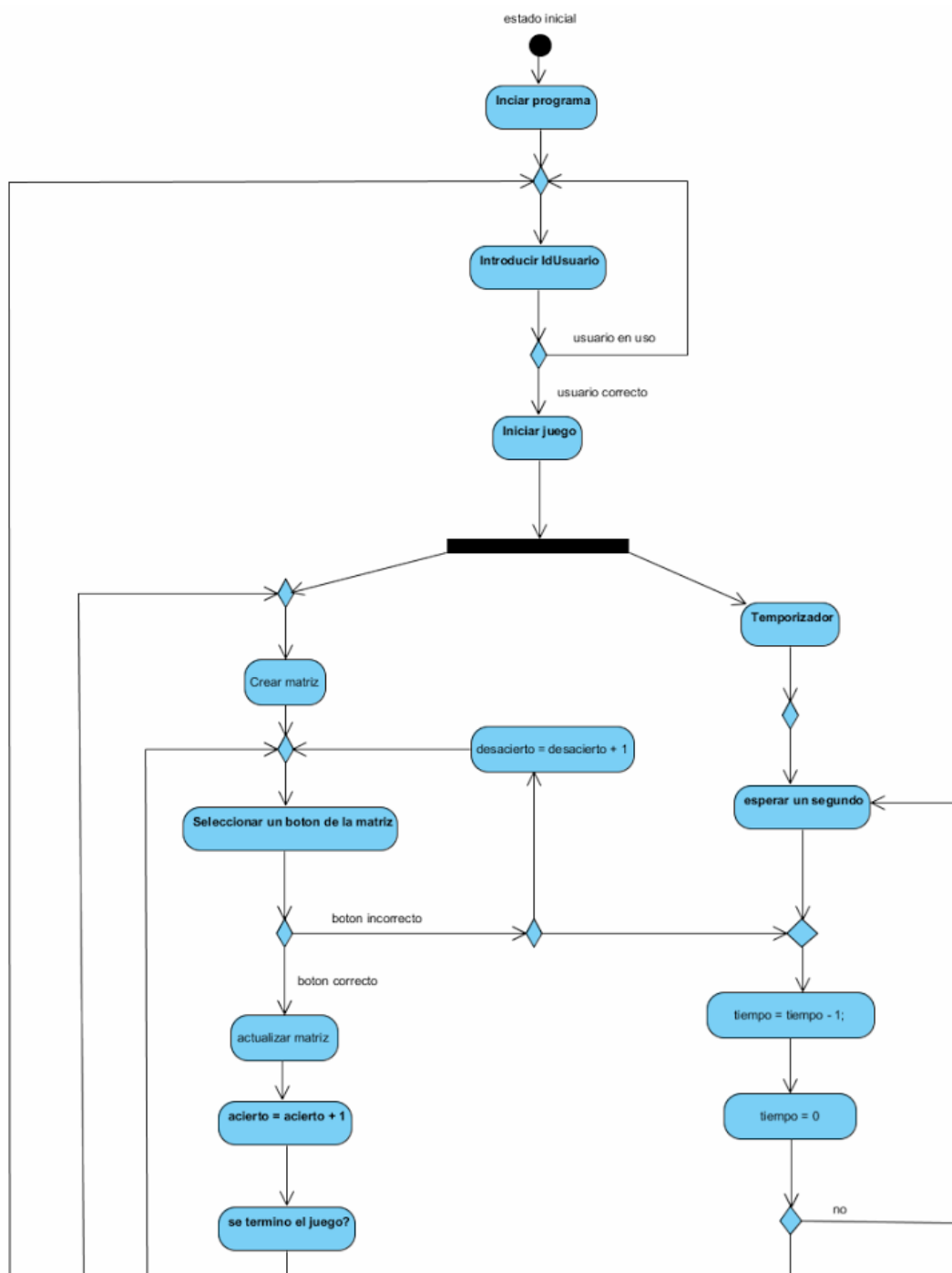
## 6. Diagramas UML

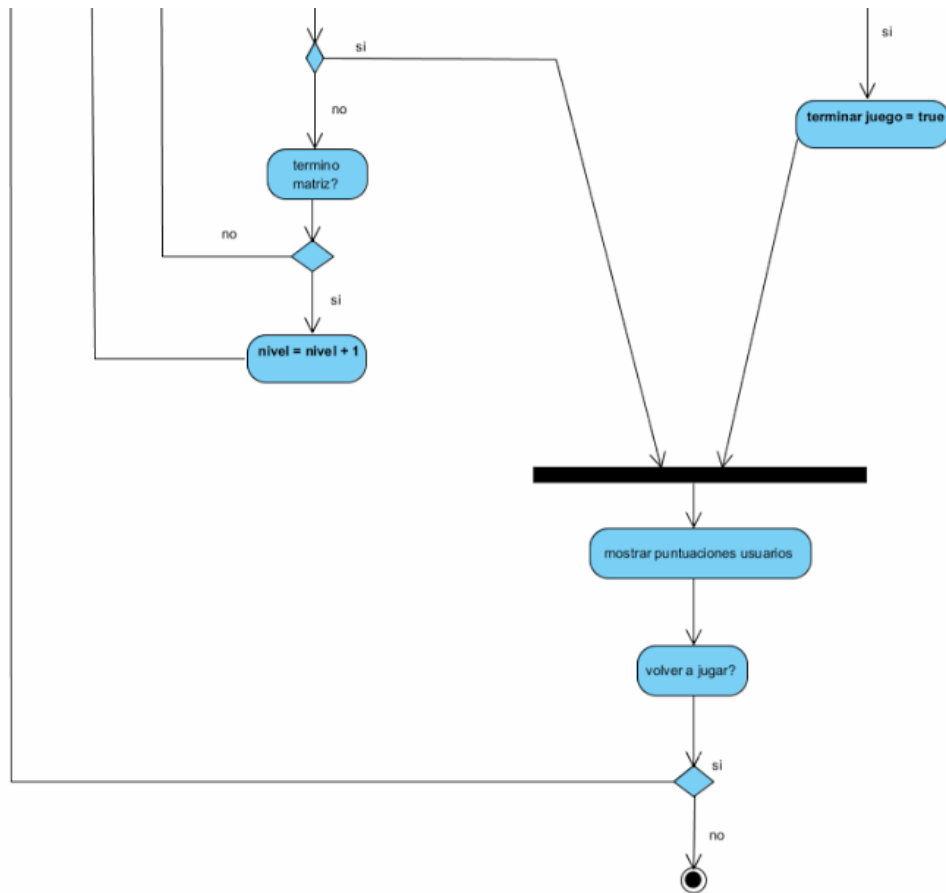
Se mostrarán a continuación los diagramas UML necesarios para la explicación de los requerimientos del software. Solo usamos los diagramas de Caso de uso, Actividad y Secuencia.

### 6.1 Diagrama de Actividad de Mind Game

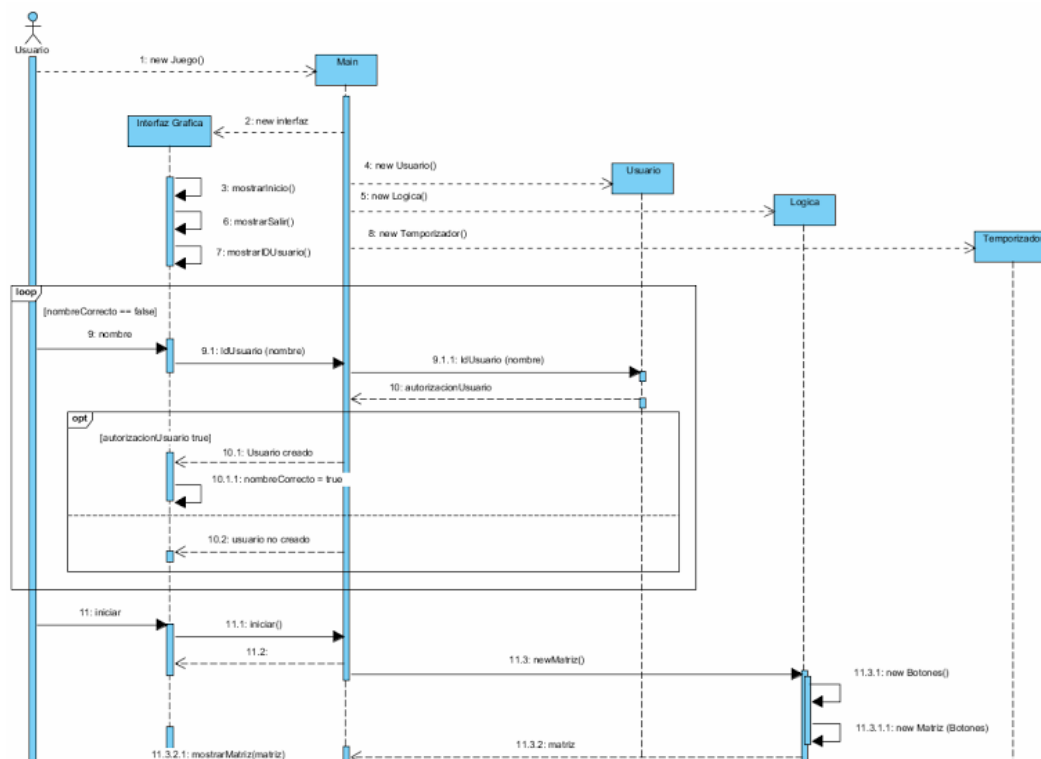


## 6.2 Diagrama de Caso de Uso para Mind Game

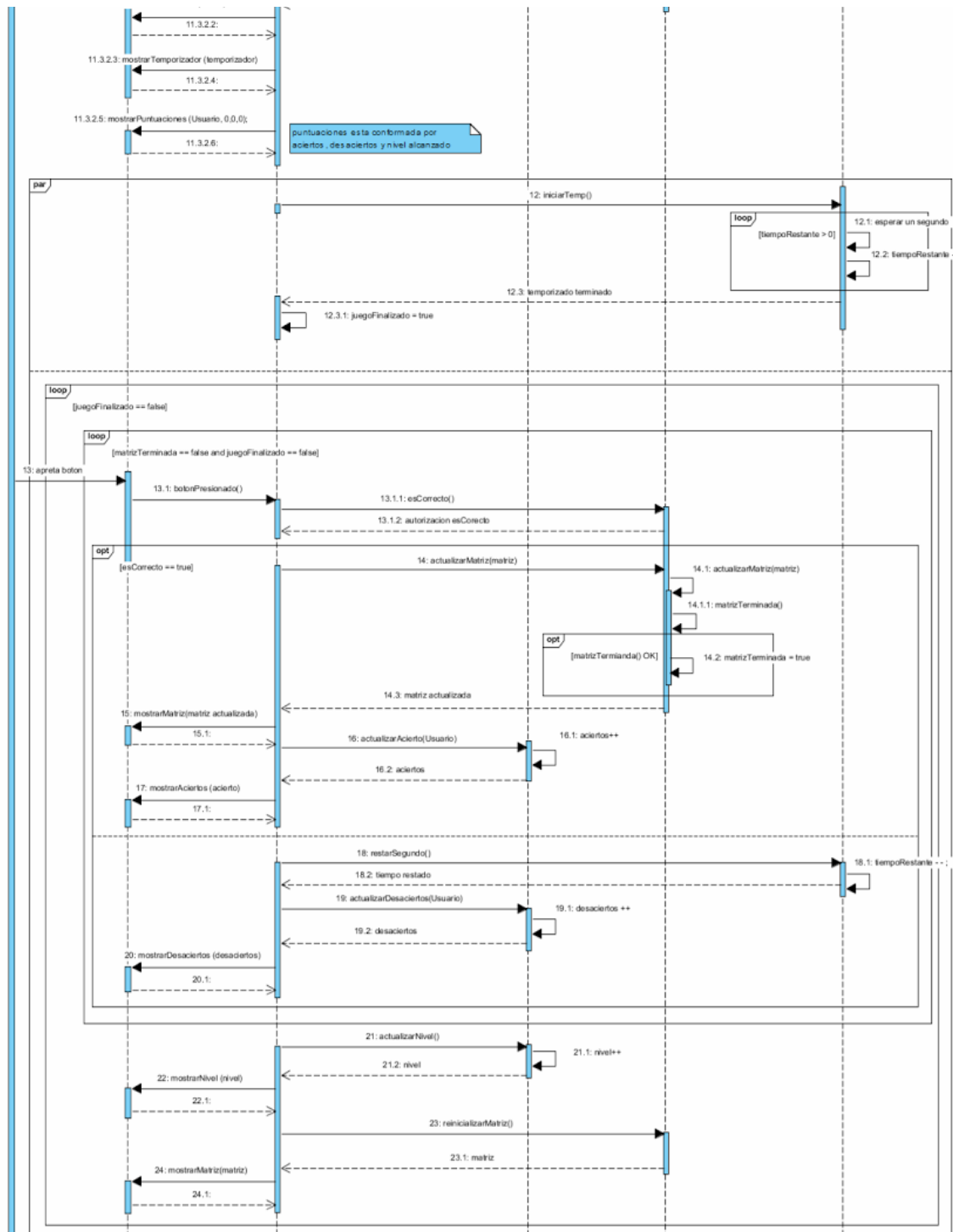




### 6.3 Diagrama de Secuencia para Mind Game







## 7. Casos de Uso y Matriz de Trazabilidad

### 7.1 Casos de Uso

A continuación se detallarán los distintos casos de usos, que cubren las distintas funcionalidades que podrán realizar los usuarios que interactúen con el juego.

Los campos que componen las tablas son los siguientes:

- Identificador (ID): distintivo unívoco del caso de uso.
- Nombre: Nombre descriptivo del caso de uso.
- Precondiciones: Condiciones previas que se deben cumplir para poder realizar determinada operación.
- Postcondiciones: Estado del sistema tras la ejecución de determinada operación.
- Descripción: Explicación clara y concisa del caso de uso que se está especificando.

ID	Nombre	Actor	Precondiciones	Postcondiciones	Descripción
CU1	Iniciar partida	Jugador/Usuario	El jugador ingresa ID usuario o el jugador termino una partida y selecciona el botón iniciar.	Una nueva partida comienza.	La opción de iniciar partida estará disponible cada vez que iniciemos la aplicación y terminemos la partida.
CU2	Salir del juego	Jugador/Usuario	La aplicación debe estar iniciada, ya sea en la pantalla inicial o en pleno desarrollo de la partida.	La aplicación se cierra. liberando los recursos del sistema.	El botón salir se debe poder presionar en cualquier momento, esto ocasiona que la aplicación se cierre liberando los recursos consumidos por la misma.
CU3	Ingresar ID usuario	Jugador/Usuario	El jugador inició la aplicación o termino una partida.	El jugador debe poder ingresar un ID usuario.	El jugador podrá ingresar su ID cada vez que iniciamos la aplicación o deseamos iniciar una partida nueva.
CU4	Presionar numero incorrecto	Jugador/Usuario	En pleno desarrollo de la partida.	No se ven cambios en la matriz numérica.	El jugador podrá seleccionar un número en el orden incorrecto, la barra de progreso del tiempo disminuye en una unidad, aumenta en una unidad los desaciertos y el número seleccionado no sufre ningún cambio.
CU5	Presionar número de forma correcta	Jugador/Usuario	En pleno desarrollo de la partida.	El número seleccionado correctamente en la secuencia se deshabilita.	El jugador podrá seleccionar un número de forma correcta en la secuencia, esto provocará que el número de aciertos se incremente en una unidad y el número seleccionado se deshabilita hasta una nueva secuencia.

## 7.2 Matriz de Trazabilidad

La matriz de trazabilidad relaciona los distintos casos de uso realizadas con los requerimientos de software. Por lo tanto a continuación mostraremos la tabla generada:

	CU1	CU2	CU3	CU4	CU5
REQ F1					
REQ F2					
REQ F3					
REQ F4					
REQ F5					
REQ F6					
REQ F7					
REQ F8					
REQ F9					
REQ F10					
REQ F11					
REQ NO F1					
REQ NO F2					
REQ NO F3					
REQ NO F4					
REQ NO F5					
REQ NO F6					
REQ NO F7					
REQ NO F8					

## 8. Casos de Prueba del Sistema y Actualizado de Matriz de Trazabilidad

Los campos que componen las tablas son los siguientes:

- Identificador (ID): distintivo unívoco del caso de prueba.
- Módulo a probar: funcionalidad a verificar con el caso de prueba.
- Precondiciones: condiciones necesarias para poder realizar el caso de prueba.
- Postcondiciones: resultados posibles del caso de prueba.
- Descripción: pasos a seguir para realizar el caso de prueba.

### 8.1 Casos de Prueba de Requerimientos Funcionales

Para la descripción de los casos de pruebas de requerimientos funcionales se utilizará la siguiente tabla:

ID	Módulo a Probar	Precondiciones	Postcondiciones	Descripción
CP1	Barra de tiempo (temporizador)	El usuario presiona el botón iniciar en la pantalla inicial..	La barra de tiempo debe disminuir cada un segundo.	El usuario una vez que ingresa un ID valido, presiona el botón iniciar y comenzara el juego, debe ser posible ver que la barra de tiempo disminuye cada un segundo.
CP2	Generación de matriz de números (Panel de números)	El usuario cuando ejecute la aplicación deberá presionar el botón iniciar de la pantalla inicial o bien el usuario completa una secuencia en pleno desarrollo del juego, lo que producirá una matriz de números aleatorios.	La matriz de números será mostrada en la misma ventana con todos los números con sus valores del 1 al 9 sin repetirse y aleatoriamente.	El usuario debe ejecutar la aplicación. Luego aparecerá la ventana del juego, la cual el usuario una vez que ingresa un ID valido, presiona el botón iniciar y comenzara una partida, simultáneamente tendrá que aparecer en la ventana la matriz de números generada. O en pleno desarrollo del juego una vez que se complete la matriz se genera una nueva matriz.
CP3	Señal indicativa de color (forma de ordenamiento ascendente o descendente).	El usuario inicia una nueva partida o el usuario completa una matriz.	Debe ser posible visualizar una señal de color. Es decir, tanto la barra de progreso como la matriz de números se tendrán que poner de color azul o naranja.	El usuario debe ejecutar la aplicación. Luego aparecerá la ventana del juego, en la cual el usuario una vez que ingresa un ID valido, presiona el botón iniciar o el usuario se encuentra en pleno desarrollo del juego y completa una secuencia de números, debe ser posible visualizar la señal de color en la matriz y en la barra de progreso ya sea de color azul o naranja.
CP4	Indicador de aciertos, desaciertos y nivel.	El jugador debe ordenar los números de menor a mayor o viceversa según el color asignado, selecciona números de forma correcta y de forma incorrecta, y se completa una secuencia.	Se producen cambios en los indicadores cada vez que se realiza un acierto, desacierto o se completa una secuencia.	El usuario debe ejecutar la aplicación. Luego aparecerá la ventana del juego, en la cual el usuario una vez que ingresa un ID valido, presiona el botón iniciar y el usuario se encuentra en pleno desarrollo del juego, se selecciona un número de forma

				correcta el indicador de aciertos se incrementará en una unidad, lo mismo debe ocurrir con el indicador de desaciertos cuando se selecciona un número de forma incorrecta en la secuencia. El usuario debe completar una secuencia entera para notar que el indicador de nivel se incrementa en una unidad.
CP5	Ingreso del ID usuario	El usuario se encuentra en la pantalla de inicio y debe ingresar un ID de usuario.	El usuario ingresa un ID válido, es decir que no se usó previamente y se registra de forma exitosa..	El usuario debe ejecutar la aplicación, luego aparecerá la ventana del juego o al iniciar una nueva partida el usuario tendrá la opción de ingresar la ID de usuario. Se debe ingresar un ID utilizado previamente y se deberá mostrar un mensaje de error. También se debe ingresar una ID que no haya sido usada previamente y esta se debe registrar de forma correcta.
CP6	Finalización de partida	El usuario se encuentra en medio del desarrollo del juego y la barra de progreso del tiempo llega su fin indicando la finalización de la partida.	La partida finaliza.	El usuario debe ejecutar la aplicación. Luego aparecerá la ventana del juego, en la cual el usuario una vez que ingresa un ID válido, presiona el botón iniciar y comienza la partida. Una vez finalizado el tiempo la matriz numérica debe desaparecer y el usuario no podrá continuar con la partida..
CP7	Tabla de puntuaciones	Cuando se termina la partida de cada usuario.	Debe ser posible visualizar en la ventana del juego la tabla de puntuaciones, donde la misma contendrá el ID de cada usuarios, teclas acertadas por cada usuario, teclas no acertadas por cada usuario y cantidad de niveles realizados por cada usuario.	El usuario debe ejecutar la aplicación. Luego aparecerá la ventana del juego, la cual el usuario una vez que ingresa un ID válido, presiona el botón iniciar y comenzara una partida. Cuando haya finalizado la partida se tendrá que mostrar la tabla de puntuaciones.
CP8	Presionar un número ya seleccionado previamente en la matriz de números generada.	El usuario se encuentra en pleno desarrollo del juego, y decide presionar un número que fue seleccionado previamente y de forma correcta.	No se deberá notar ningún cambio, ya que el número estará deshabilitado temporalmente hasta que complete la matriz de números, pero se incrementa una variable para contar las teclas pulsadas correctamente .	El usuario debe ejecutar la aplicación. Luego aparecerá la ventana del juego, en la cual el usuario una vez que ingresa un ID válido, presiona el botón iniciar y el usuario se encuentra en pleno desarrollo del juego y selecciona un número correctamente de la secuencia, si el usuario la presiona de nuevo este número, no habrá ningún cambio ya que este se encontrará deshabilitado para la secuencia actual.

CP9	Selección de un número incorrecto de la secuencia.	El usuario pulsa un número del panel en orden inapropiado	El número del panel presionado por el usuario en orden incorrecto no se deshabilita.	El usuario está jugando y en un momento dado pulsa un número de la matriz produciendo un error, pero este número no se deshabilita hasta que se ordena de manera correcta.
CP10	Descontar una unidad la barra de progreso de tiempo	El usuario se encuentra en pleno desarrollo del juego, y decide presionar un número el cual es incorrecto con la orden de la secuencia de números.	Se deberá mostrar que la barra de progreso decrementa un segundo del tiempo que lleva hasta ese momento.	El usuario debe ejecutar la aplicación. Luego aparecerá la ventana del juego, en la cual el usuario una vez que ingresa un ID válido presiona el botón iniciar y se encuentra en pleno desarrollo del juego y selecciona un número incorrectamente de la secuencia, Por lo tanto en la barra de progreso del tiempo se mostrará que se decrementa un segundo.
CP11	Inicio y Fin de partida.	El jugador ingresa con ID usuario y da INICIO a la partida.	El usuario desea finalizar la partida y selecciona el botón de SALIR.	Estas dos opciones estarán en la interfaz de usuario, la opción INICIO solamente se presentará al comienzo cuando el usuario desee comenzar una partida, una vez que ingreso y quiera finalizar el juego estará presente la opción SALIR.

## 8.2 Casos de Prueba de Requerimientos No Funcionales

Para la descripción de los casos de pruebas de requerimientos no funcionales se utilizará la siguiente tabla:

ID	Módulo a Probar	Precondiciones	Postcondiciones	Descripción
CP12	Memoria Ram para el funcionamiento de la aplicación	Tener disponible como mínimo 100 MB de ram	Ejecutar el juego en el ordenador y verificar que el juego consume aproximadamente 100 MB de la memoria RAM	Este caso de uso será útil para que el usuario conozca que especificación de memoria ram debe tener disponible en su ordenador para poder ejecutar el juego
CP13	Espacio libre en disco	Tener disponible en el disco duro del ordenador como mínimo 10 MB	Ejecutar el juego en el ordenador y verificar que el juego consume aproximadamente 10 MB de la memoria de disco duro	Este caso de uso especifica que memoria de disco duro debe tener el ordenador el usuario para poder ejecutar el juego.
CP14	Procesador mínimo	El usuario para poder ejecutar la aplicacion debera tener un procesador mínimo Pentium 2	El usuario puede ejecutar el juego en el ordenador de manera eficiente.	El usuario deberá tener un procesador mínimo pentium 2 para poder ejecutar la aplicación en formato JAR de manera óptima
CP15	Tiempo límite inicial	El juego inicia y el temporizador comienza a decrementar segundo a	Finaliza el tiempo de 60 segundos y el juego finaliza.	El usuario debe ejecutar la aplicación. Luego aparecerá la ventana del juego, en la cual el

		segundo.		usuario una vez que ingresa un ID válido presiona el botón iniciar y se encuentra en pleno desarrollo del juego, se debe seleccionar los número de la secuencia de forma correcta o no seleccionar ninguno con el objetivo de que transcurra el tiempo sin ninguna modificación que se puede llegar a producir por seleccionar un número incorrecto en la secuencia. Comprobar que el tiempo transcurrido entre el inicio del juego y el final es de 60 segundos.
CP16	Liberación de recursos al salir de la aplicación	En la ventana de la aplicación seleccionar el botón salir.	La aplicación se cierra y se liberan recursos del sistema.	Una vez en la aplicación seleccionamos el botón salir, debemos visualizar en el administrador de tareas del SO la desaparición del proceso relativo a la aplicación.
CP17	Accesibilidad y legibilidad	En la aplicación.	Al ingresar el usuario, durante el juego, al visualizar estadísticas.	Visualizar los textos, números y comprobar que sean legibles y claros. Comprobar que la aplicación sea intuitiva, fácil de usar, para esto se propone que distintos usuarios de diferentes edad la utilicen y den su veredicto respecto de esto puto.

### 8.3 Actualización de Matriz de Trazabilidad

La matriz de trazabilidad que realizamos más arriba la actualizamos para mostrar la relación de los distintos casos de prueba realizados con los requerimientos de software. Por lo tanto a continuación mostraremos la tabla generada:

	C U 1	C U 2	C U 3	C U 4	C U 5	C P 1	C P 2	C P 3	C P 4	C P 5	C P 6	C P 7	C P 8	C P 9	C P 10	C P 11	C P 12	C P 13	C P 14	C P 15	C P 16	C P 17
REQ F1																						
REQ F2																						
REQ F3																						
REQ F4																						
REQ F5																						
REQ F6																						
REQ F7																						
REQ F8																						
REQ F9																						
REQ F10																						
REQ F11																						
REQ NO F1																						
REQ NO F2																						
REQ NO F3																						
REQ NO F4																						
REQ NO F5																						
REQ NO F6																						
REQ NO F7																						
REQ NO F8																						

### 9. Casos de prueba que actúan como Smoke Testing

Teniendo en cuenta que los Smoke Testing son un tipo de pruebas de software que se realizan después de la compilación de software para asegurar que las funcionalidades críticas del programa están funcionando bien.

En nuestro caso seleccionaremos los siguientes casos de pruebas para permitir la aceptación de los clientes:

**CP2:** Este caso de prueba permitirá mostrarle al los cliente que el sistema genera numero desordenados aleatoriamente así obteniendo mayor dificultad del juego.

**CP3:** Este caso prueba permitirá mostrarle a los clientes que existe en el juego dos formas de



ordenamiento de los números y que se definirían dependiendo del color de la matriz de números.

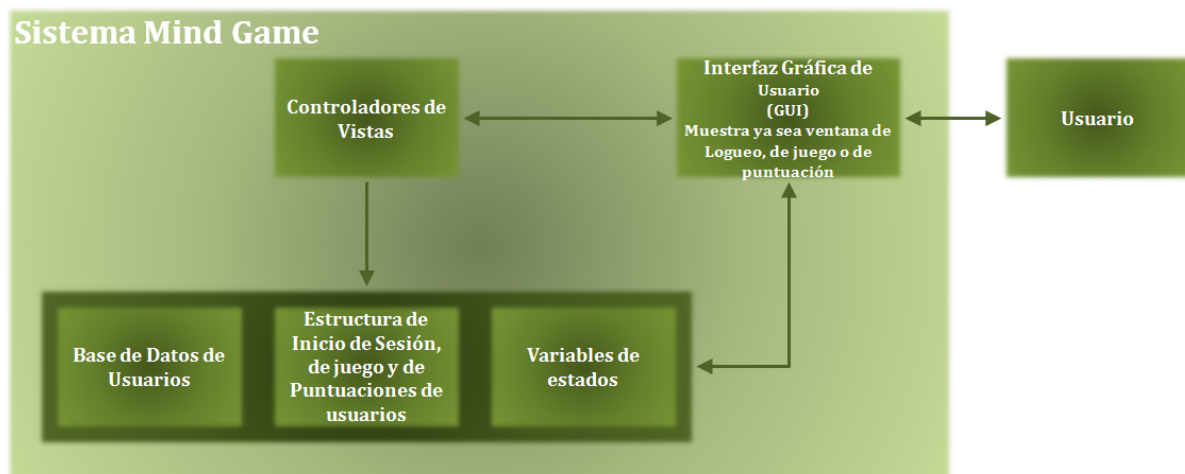
**CP4:** Este caso de prueba permitirá mostrarle a los clientes que existe un indicador de aciertos. Este proporciona información acerca de la puntuación que el jugador tiene hasta entonces.

**CP5:** Este caso de prueba será utilizado para mostrarle a los clientes que se pueden registrar en el juego. Este será un identificador para conocer qué puntuación realizó y en qué posición estarán en la tabla de puntuaciones.

**CP12 CP13 Y CP14:** Estos casos de prueba permitirán mostrarle a los clientes cuáles serán la memoria RAM, memoria de disco duro y microprocesador que deberá tener el ordenador en donde se pondrá en funcionamiento el juego. Teniendo en cuenta que es una aplicación que no requiere grandes especificaciones de hardware pueden ser pruebas que permitan la aceptación del cliente.

## **10. Arquitectura del Sistema**

Antes de implementar nuestro sistema Mind Game debemos diseñar la estructura global y organizar nuestro sistema. Para eso, es importante un modelo arquitectónico que nos permita describir la forma en que se organiza el sistema, es decir, que tenga un conjunto de componentes que estén relacionados entre sí. Por lo tanto, a continuación mostraremos dicho modelo:



Como se observa, este diagrama no es más que un diagrama informal que representa a nuestro sistema y que además describe la arquitectura del mismo. En forma general, este diagrama es la única documentación arquitectónica de nuestro sistema.