Universidad Nacional de Córdoba

Facultad de Ciencias Exactas Físicas y Naturales

Los piratas

Ingeniería de Software

PacMan

Especificación de Requerimientos de Software

Autores:

- Arnolfo Emanuel
- Landaeta Ezequiel
- Montero Felipe
- Olocco Laureano

Versión del documento: 1.2

Historial de versiones

Versión	Fecha	Cambios
1.0	26 de abril	Primera versión
1.1	30 de abril	Modificación de varias secciones y revisión general.
1.2	1 de Mayo	Se agregan los casos de prueba y nueva matriz de trazabilidad
1.3	1 de Junio	Se agrega arquitectura preliminar

Índice

Introducción

Este documento contempla los requerimientos que se tendrán en cuenta para el desarrollo del proyecto de Pacman, y su propósito es definir de la manera más objetiva posible el uso que se le dará al programa, las funcionalidades que el mismo debe incluir y algunos detalles de qué se implementará en materia de desarrollo, además de explicar cómo comprobar si se cumplen los requerimientos especificados, tanto funcionales como no funcionales

1.1 Funcionamiento

Ms. PacMan es un videojuego arcade producido originalmente por Namco, cuyo objetivo es lograr que Ms. PacMan capture todas las píldoras de un laberinto evitando ser atrapada por un conjunto de fantasmas. A continuación se hará una descripción de alto nivel del sistema, con objetivo de identificar a los usuarios involucrados y ayudar a explicar sus roles.

1.2 Interfaz de sistema

El menú inicial permitirá a los usuarios navegar entre 5 opciones:

- New Game
- Scores
- Config
- Login
- Exit

Esto se realizará mediante una interfaz de usuario (UI).

1.3 Restricciones Generales

En principio la aplicación estará diseñada para correr en PC, en Windows y Linux.
Limitaciones de Software
➢ El usuario debe tener instalado Windows 7 o superior en su equipo o cualquier distribución de linux.

Especificaciones de Hardware Recomendadas

- ➢ CPU Intel Core 2 Duo o AMD Athlon x2 o superior.
- ➢ 1 GB de RAM.
- ➢ 700 MB de espacio libre en disco.
- ➢ Se requiere conexión a internet, no para usar el programa, pero si para descargar e instalar la última versión disponible

2 Requerimientos

2.1 Lista de Requerimientos Funcionales

- **Req 1:** El jugador podrá elegir entre las opciones de los menús
- **Req 2:** El jugador deberá poder ver el mapa y la ubicación de los “fantasmas” en el mismo
- **Req 3:** El jugador deberá poder abandonar la partida en cualquier momento
- **Req 4:** El jugador deberá poder pausar y reanudar la partida en cualquier momento
- **Req 5:** El jugador deberá poder ver la puntuación y la duración de la partida durante la misma
- **Req 6:** El jugador deberá poder mover el personaje presionando las flechas del teclado
- **Req 7**: El usuario deberá poder ver un historial de al menos 5 puntuaciones anteriores.
- **Req 8:** El usuario podrá comer a los fantasmas en determinados momentos del juego con el personaje.
- **Req 9:** El sistema será capaz de mover el personaje y los fantasmas en los 4 sentidos.
- **Req 10:** El usuario podrá mover el personaje y los fantasmas únicamente por el área libre del mapa.
- **Req 11:** El sistema deberá cumplir que cuando el personaje sea comido, este pierda una de sus vidas y retome la partida con su puntuación alcanzada hasta el momento.
- **Req 12**: El sistema deberá al perder todas sus vidas, mostrar la opción de reiniciar la partida o volver al menú principal, además de mostrar la puntuación final alcanzada y el historial de mejores puntuaciones.
- **Req 13:** El usuario deberá poder cerrar el programa desde el menú principal.
- **Req 14:** El usuario podrá contar con 3 vidas por partida y estas mostrarse en el transcurso de la misma.
- **Req 15:** El usuario una vez que se haya comido todos las bolitas, podrá pasar al siguiente nivel.

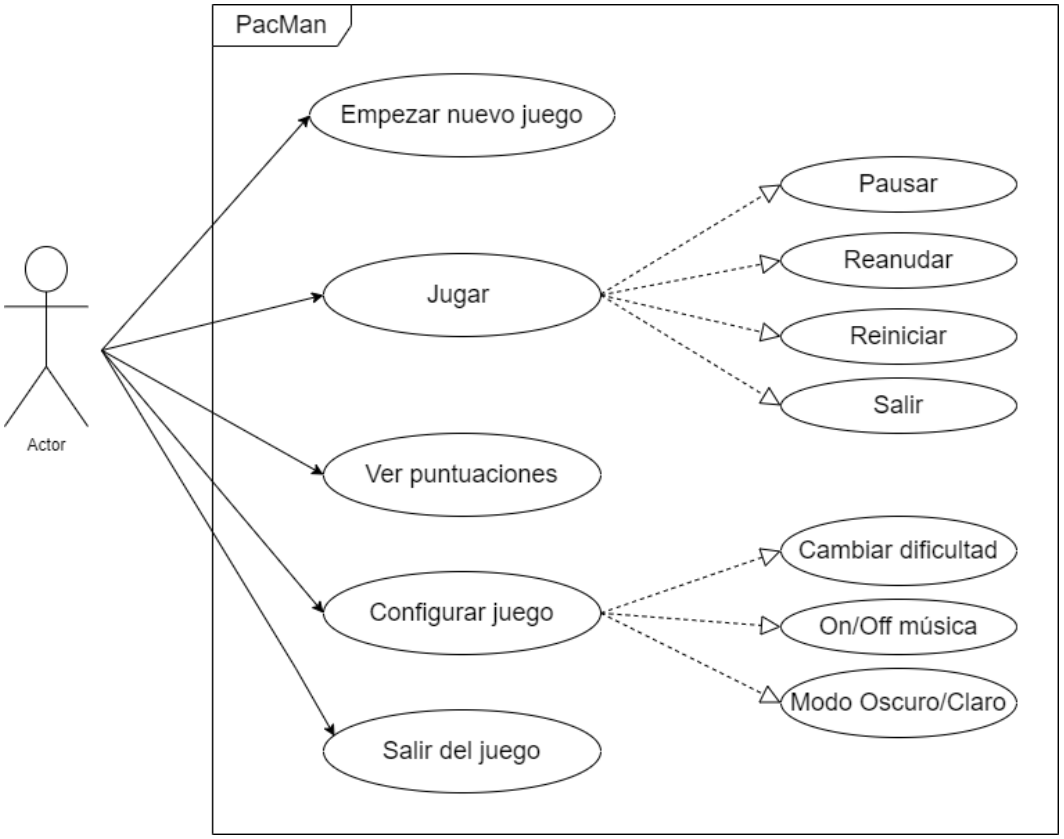
2.2 Lista de Requerimientos No Funcionales

- **Req NF 1**: El menú del juego se debe iniciar en menos de 2 segundos.
- **Req NF 2:** El juego debe poder instalarse en un máximo de 7 minutos.
- **Req NF 3:** El idioma del juego será en inglés.
- **Req NF 4:** El juego debe ser capaz de funcionar en Windows y linux.

3. Matriz de trazabilidad y Casos de Uso

3.1 Casos de uso

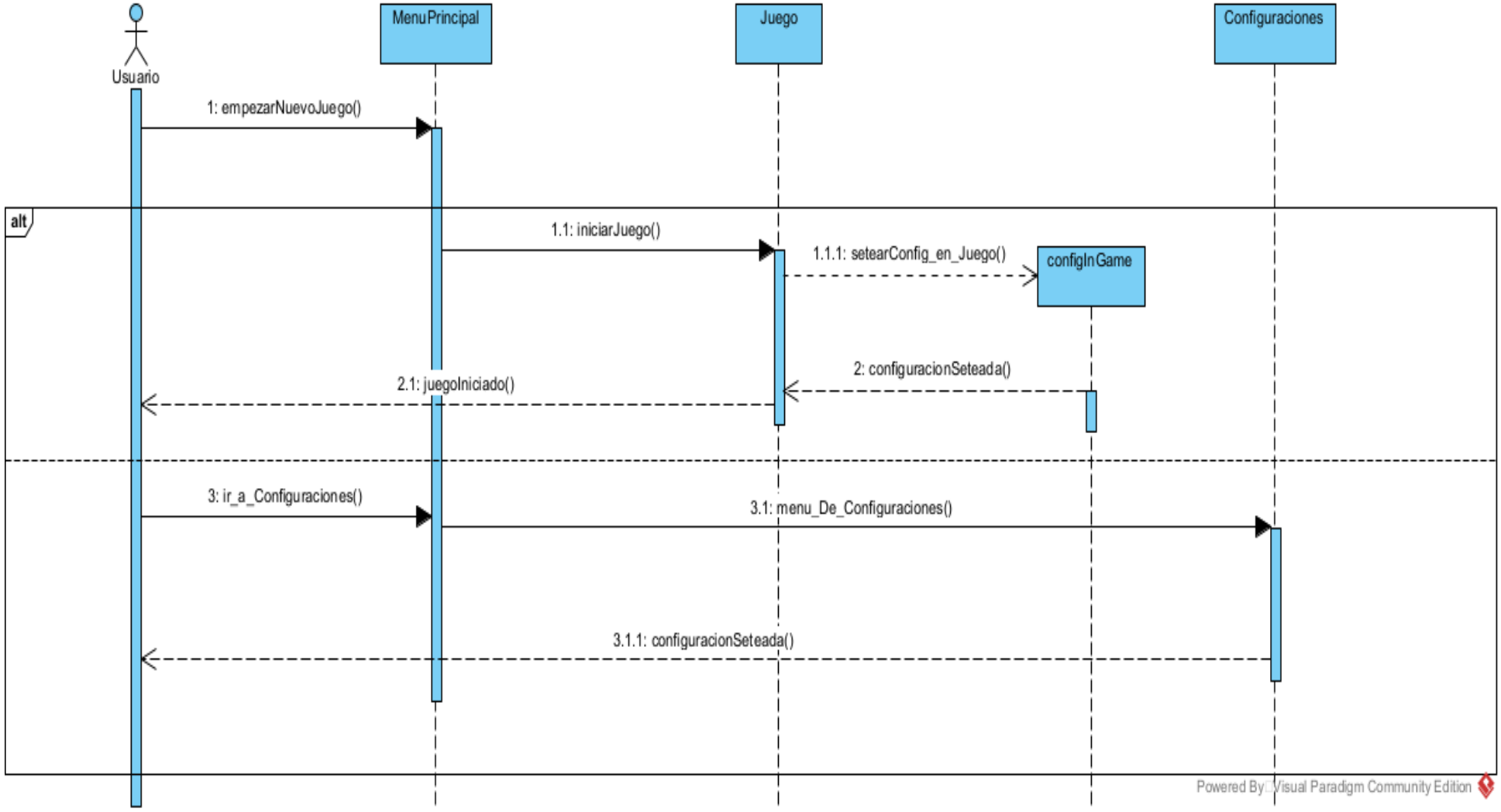
- **CU1:** Empezar un nuevo juego desde el menú principal
- **CU2:** Pausar la partida
- **CU3:** Reanudar la partida pausada
- **CU4:** Salir de la partida
- **CU5:** Reiniciar la partida
- **CU6:** Ver las puntuaciones de juegos anteriores
- **CU7:** Configurar la dificultad de partida
- **CU8:** Poner o quitar la partida
- **CU9:** Cambiar modo de interfaz



3.2 Matriz de trazabilidad

	CU1	CU2	CU3	CU4	CU5	CU6	CU7	CU8	CU9
Req 1									
Req 2									
Req 3									
Req 4									
Req 5									
Req 6									
Req 7									

Req 8									
Req 9									
Req 10									
Req 11									
Req 12									
Req 13									
Req NF 1									
Req NF 2									
Req NF 3									
Req NF 4									


Usuario
MenuPrincipal
Juego
Configuraciones
configInGame
alt
1: empezarNuevoJuego()
1.1: iniciarJuego()
1.1.1: setearConfig_en_Juego()
2: configuracionSeteada()
2.1: juegoIniciado()
3: ir_a_Configuraciones()
3.1: menu_De_Configuraciones()
3.1.1: configuracionSeteada()
Powered By Visual Paradigm Community Edition

4. Casos de prueba

4.1 Caso de prueba 1

- **Descripción de la prueba:** El usuario inicia un juego nuevo.
- **Funcionalidad a probar:** Funcionamiento correcto del ingreso al juego y del tiempo esperado.
- **Ejecución:**
    - Abrir Juego.
    - Iniciar nueva partida
- **Resultado esperado:** El juego se inicia correctamente y la partida empieza en menos de 2 segundos.

4.2 Caso de prueba 2

- **Descripción de la prueba:** Mover pacman en todas las direcciones posibles.
- **Funcionalidad a probar:** Funcionamiento correcto del mecanismo del juego.
- **Ejecución:**
    - Abrir Juego.
    - Iniciar nueva partida
    - Mover PacMan en las 4 direcciones apretando las flechas del teclado
- **Resultado esperado:** Que el pacman se mueva en las direcciones correctas cuando se aprietan las flechas del teclado.

4.3 Caso de prueba 3

- **Descripción de la prueba:** Que el pacman se coma las bolitas.
- **Funcionalidad a probar:** Funcionamiento correcto del objetivo del juego.
- **Ejecución:**
    - Abrir Juego.
    - Iniciar nueva partida
    - Mover PacMan en las direcciones donde se encuentren “bolitas”
- **Resultado esperado:** Luego de pasar el PacMan por el lugar, desaparezcan estas “bolitas”.

4.4 Caso de prueba 4

- **Descripción de la prueba:** Chocar contra pared.
- **Funcionalidad a probar:** Funcionamiento correcto de la física del juego.
- **Ejecución:**
    - Abrir Juego.

    - Iniciar nueva partida
    - Mover PacMan hasta una pared
- **Resultado esperado:** Que el pacman o los fantasmas al llegar a una pared no pueda seguir avanzando en ese sentido.

4.5 Caso de prueba 5

- **Descripción de la prueba:** testear que el juego se pueda poner en pausa una vez comenzada la partida.
- **Funcionalidad a probar:** pausa del juego
- **Ejecución:**
    - Abrir Juego.
    - Iniciar nueva partida
    - Poner en pausa la partida
- **Resultado esperado:** Interrupción de la partida y un menú de pausa.

4.6 Caso de prueba 6

- **Descripción de la prueba:** Que se pueda reanudar el juego cuando está en pausa.
- **Funcionalidad a probar:** Tecla de pausa del juego.
- **Ejecución:**
    - Abrir Juego.
    - Iniciar nueva partida
    - Poner en pausa la partida
    - Reanudar la partida
- **Resultado esperado:** El juego reanuda al continuar luego de estar pausado.

4.7 Caso de prueba 7

- **Descripción de la prueba:** El sistema deberá mostrarle al usuario(Actor) el menú de configuraciones una vez iniciado el juego.
- **Funcionalidad a probar:** ingreso al menú de configuraciones.
- **Ejecución:**
    - Abrir Juego.
    - Entrar al menú de configuraciones
- **Resultado esperado:** Que se monte en la UI el menú de configuraciones identificado con el id=#CONFIG

4.8 Caso de prueba 8

- **Descripción de la prueba:** Correcto funcionamiento del cambio de dificultad dentro del menú de configuraciones.
- **Funcionalidad a probar:** Cambio de dificultad.
- **Ejecución:**
    - Abrir Juego.
    - Abrir menú de configuraciones.
    - Cambiar la dificultad del juego.

    - Corroborar que se cambió la dificultad del juego.
- **Resultado esperado:** Que se cambie la dificultad del juego.

4.9 Caso de prueba 9

- **Descripción de la prueba:** verificar que se pueda apagar la música desde el menú de configuraciones.
- **Funcionalidad a probar:** funcionamiento correcto del menú y las configuraciones.
- **Ejecución:**
    - Abrir Juego.
    - Abrir menú de configuraciones
    - Activar o desactivar música
- **Resultado esperado:** La música se apaga luego de realizar la prueba.

4.10 Caso de prueba 10

- **Descripción de la prueba:** Correcto funcionamiento del cambio de modo oscuro/claro dentro del menú de configuraciones.
- **Funcionalidad a probar:** Cambio de modo de color.
- **Ejecución:**
    - Abrir Juego.
    - Abrir menú de configuraciones
    - Cambiar el modo de color
- **Resultado esperado:** Que se cambie el modo de color dentro del juego.

4.11 Caso de prueba 11

- **Descripción de la prueba:** Morir al tocar un fantasma y perder una vida.
- **Funcionalidad a probar:** Muerte por fantasmas.
- **Ejecución:**
    - Abrir Juego.
    - Iniciar nueva partida
    - Tocar un “fantasmita”
- **Resultado esperado:** Que el pacman pierda una vida al tocar uno de los “fantasmitas”.

4.12 Caso de prueba 12

- **Descripción de la prueba:** Que el PacMan pueda comer a los fantasmas luego de adquirir el poder especial.
- **Funcionalidad a probar:** Correcto funcionamiento de la dinámica del poder especial.
- **Ejecución:**
    - Abrir Juego.

    - Iniciar nueva partida
    - Mover PacMan en las direcciones donde se encuentren “poderes especiales”
    - Dirigirse a un fantasma y comerlo
- **Resultado esperado:** el Pacman come el fantasma y éste desaparece.

4.13 Caso de prueba 13

- **Descripción de la prueba:** Que todos los fantasmas se puedan mover en todas las direcciones posibles.
- **Funcionalidad a probar:** Movimiento de los fantasmas.
- **Ejecución:**
    - Abrir Juego.
    - Iniciar nueva partida
    - Observar movimiento de los fantasmas
- **Resultado esperado:** Fantasmas moviéndose en todas las direcciones posibles.

4.14 Caso de prueba 14

- **Descripción de la prueba:** Verificar que los fantasmas persiguen al pacman desde el comienzo de la partida.
- **Funcionalidad a probar:** Funcionamiento correcto de la dinámica del juego.
- **Ejecución:**
    - Abrir Juego.
    - Iniciar nueva partida
    - Mover PacMan cerca de los fantasmas
- **Resultado esperado:** Luego de pasar el PacMan cerca de un fantasma, estos empezaran a perseguir al PacMan.

4.15 Caso de prueba 15

- **Descripción de la prueba:** Pasar de nivel.
- **Funcionalidad a probar:** Funcionamiento correcto del pasaje de un nivel al otro.
- **Ejecución:**
    - Abrir Juego.
    - Iniciar nueva partida
    - Ganar el “nivel 1”
    - Pasar al “nivel 2” y así sucesivamente.
- **Resultado esperado:** luego de completar la partida que la interfaz se reinicie y muestre un nuevo nivel.

4.16 Caso de prueba 16

- **Descripción de la prueba:** Salir de la partida.
- **Funcionalidad a probar:** Funcionamiento correcto de la opción salir de la partida.

- **Ejecución:**
    - Abrir Juego.
    - Iniciar nueva partida
    - Salir de la partida
- **Resultado esperado:** Luego de pulsar sobre “salir de la partida” (tecla escape)se vuelve al menú principal.

4.17 Caso de prueba 17

- **Descripción de la prueba:** Salir del juego.
- **Funcionalidad a probar:** Funcionamiento correcto de la opción del menú principal para salir del juego.
- **Ejecución:**
    - Abrir Juego.
    - Salir del juego
- **Resultado esperado:** luego de seleccionar salir del juego, el juego debe cerrarse.

4.18 Caso de prueba 18

- **Descripción de la prueba:** Actualización del puntaje al comer fantasma.
- **Funcionalidad a probar:** Funcionamiento correcto del puntaje.
- **Ejecución:**
    - Abrir Juego.
    - Salir del juego
    - Iniciar un juego
    - Comer un fantasma
- **Resultado esperado:** El puntaje se actualiza correctamente al comer un fantasma.

4.19 Caso de prueba 19

- **Descripción de la prueba:** Ver historial de Puntuaciones.
- **Funcionalidad a probar:** Funcionamiento correcto del botón “Ver Puntuaciones”.
- **Ejecución:**
    - Abrir Juego.
    - Apretar el botón “Ver Puntuaciones”
- **Resultado esperado:** El puntaje se muestra al apretar el botón de ver puntuaciones dentro del menú principal.

4.20 Caso de prueba 20

- **Descripción de la prueba:** Perder una vida.
- **Funcionalidad a probar:** Perder una vida al tocar un fantasma.
- **Ejecución:**
    - Abrir Juego.

    - Iniciar una nueva partida.
    - Jugar hasta ser alcanzado por un fantasma.
- **Resultado esperado:** El contador de vidas debe disminuir en uno al perder una vida.

4.21 Caso de prueba 21

- **Descripción de la prueba:** Que el juego sea instalable en menos de 5 minutos.
- **Funcionalidad a probar:** Instalación del juego.
- **Ejecución:**
    - Abrir Juego.
    - Iniciar una nueva partida.
    - Jugar hasta ser alcanzado por un fantasma.
- **Resultado esperado:** El contador de vidas debe disminuir en uno al perder una vida.

4.22 Caso de prueba 22

- **Descripción de la prueba:** Que el juego se pueda iniciar en windows y linux.
- **Funcionalidad a probar:** Instalación del juego .
- **Ejecución:**
    - Ejecutar el script
    - Iniciar el juego
- **Resultado esperado:** el juego debe instalarse correctamente y poder iniciar el juego.

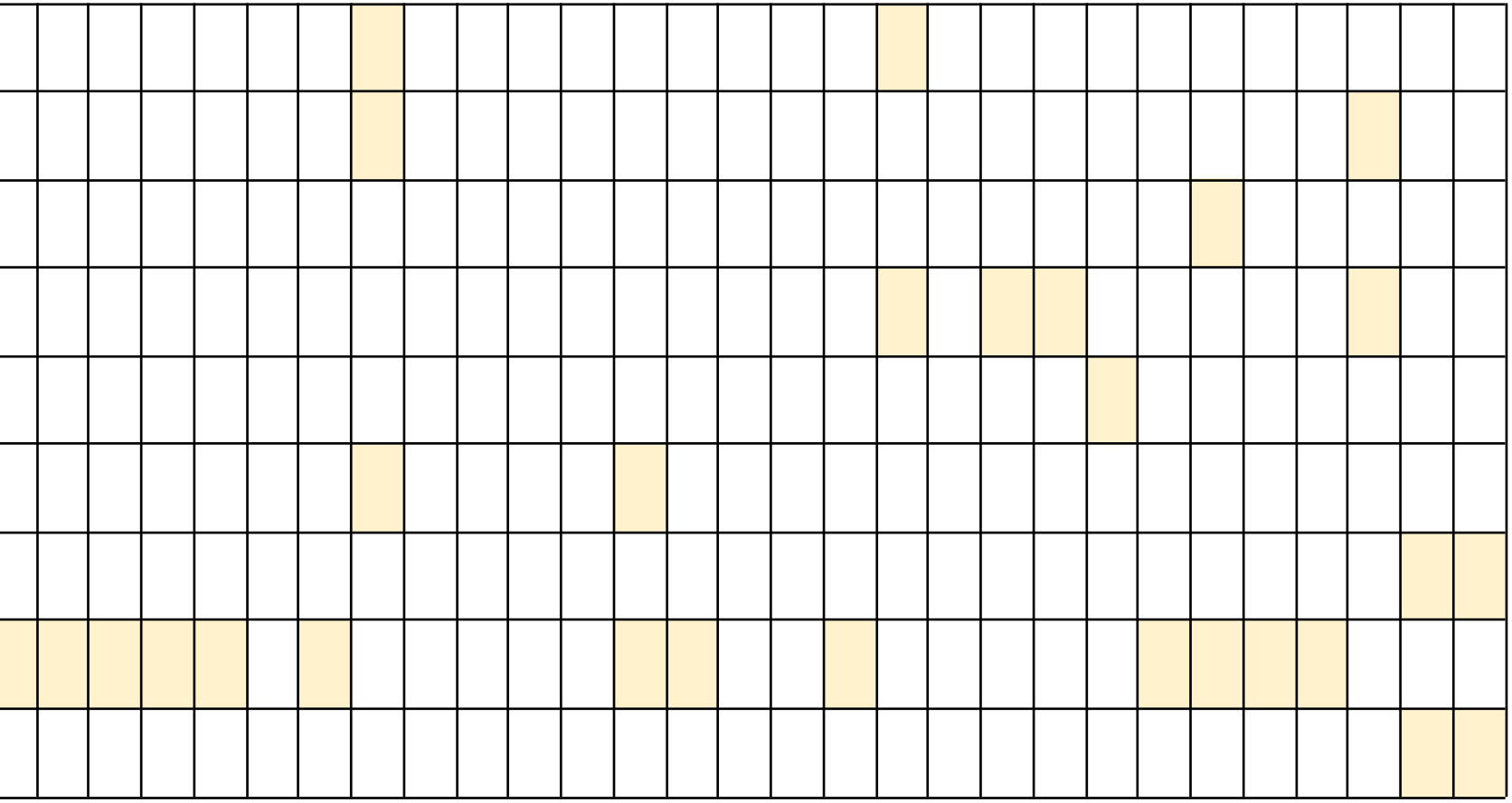
5 Smoke test

Los casos de prueba a utilizar para que el cliente acepte la aplicación son:

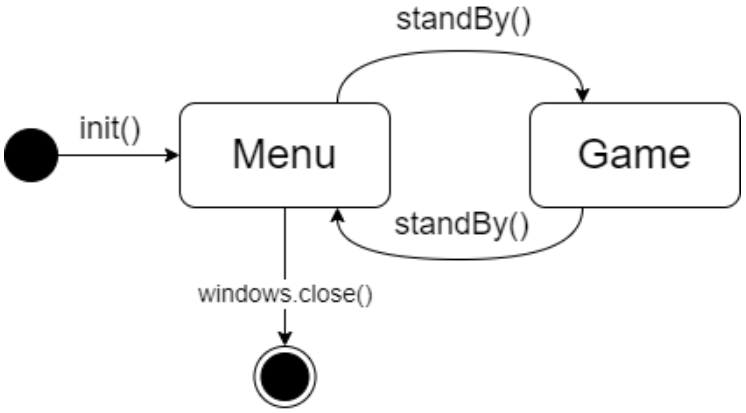
- Caso de prueba 1
- Caso de prueba 2
- Caso de prueba 3
- Caso de prueba 4
- Caso de prueba 5
- Caso de prueba 6
- Caso de prueba 11
- Caso de prueba 12
- Caso de prueba 13
- Caso de prueba 14
- Caso de prueba 16
- Caso de prueba 17
- Caso de prueba 20
- Caso de prueba 22

6 Matriz de trazabilidad con Casos de prueba

	CU 1	CU 2	CU 3	CU 4	CU 5	CU 6	CU 7	CU 8	CU 9	CP1	CP2	CP3	CP4	CP5	CP6	CP7	CP8	CP9	CP 10	CP 11	CP 12	CP 13	CP 14	CP 15	CP 16	CP 17	CP 18	CP 19	CP 20	CP 21	CP 22
Req 1																															
Req 2																															
Req 3																															
Req 4																															
Req 5																															
Req 6																															
Req 7																															
Req 8																															
Req 9																															
Req 10																															


Req 11
Req 12
Req 13
Req 14
Req 15
Req NF 1
Req NF 2
Req NF 3
Req NF 4

7. Diagrama de arquitectura preliminar



El sistema fue estructurado como una máquina de estados finitos (FSM, por sus siglas en inglés). La misma cuenta con dos estados principales (Menu y Game) en los cuales se desarrolla toda la funcionalidad del sistema. Estos estados se conectan mediante transiciones que se invocan llamando al método standBy(). En la Figura N° 1 se muestra un diagrama de estados que representa dicha arquitectura del sistema.