Requerimientos funcionales

1. El sistema permite tener un máximo de 10 jugadores jugando al mismo tiempo.
2. Colisiones de serpiente consigo mismo, con otro jugador o con el límite del escenario.
3. Generación de comida en una posición libre que no esté ocupada por algún otro elemento dentro del juego.
4. Registro de jugadores en el servidor con un nike name.
5. Liberación de espacio el servidor al perder.
6. Asignación de colores a las serpientes diferentes a cada uno de los usuarios que ingresen al juego.
7. Almacenamiento de puntuaciones para cada jugador de acuerdo a cuantas comidas haya recolectado en el juego.

Requerimientos no funcionales

CU1- Iniciar Sesión

|  |  |
| --- | --- |
| **ID:** | CU1 |
| **Nombre:** | Iniciar Sesión |
| **Autor(es):** | Luis Gerardo Bonilla Ramírez |
| **Fecha de creación:** | 02 / Junio / 2019 |
| **Fecha de actualización:** | 02 / Junio / 2019 |
| **Descripción:** | El actor crea un registro de sesión en el servidor para que tenga un lugar y un color apartado dentro del sistema y todos los demás jugadores tengan un registro actualizado de jugadores. |
| **Actor:** | Jugador |
| **Frecuencia de uso:** | Al menos 1 vez al día. |
| **Disparador:** | El jugador inicia la aplicación. |
| **Precondiciones:** |  |
| **Flujo normal:** | 1. El sistema muestra la pantalla MenuPrincipal con los elementos:    1. Nombre de jugador (texto editable)    2. Jugar (botón) 2. El jugador ingresa su nombre de jugador y selecciona el botón jugar. 3. El sistema valida el nombre ingresado con éxito. El sistema realiza una consulta a la base de datos para comprobar que exista un lugar disponible con éxito. Ingresa al jugador como una nueva SERPIENTE a la lista de serpientes dentro del servidor y devuelve la misma lista de SERPIENTES al cliente. (ver FA01, FA02) (Ver EX01) |
| **Flujo alterno:** | FA01 Datos inválidos.   1. El sistema detecta un nombre de jugador inválido y despliega un mensaje indicando por qué el nombre es incorrecto. 2. Volver al flujo normal 2.   FA02 Juego lleno.   1. El sistema detecta que todos los lugares en el sistema están llenos y despliega un mensaje indicando por qué no es posible acceder al juego. |
| **Excepciones:** | EX01 Error al conectar con el servidor.   1. El sistema no logra crear una petición al servidor y despliega un mensaje indicando por qué no es posible acceder al juego. |
| **Postcondiciones:** | En flujo normal:   1. El sistema actualiza la lista de SERPIENTES en el servidor. 2. El sistema despliega la pantalla de juego con todos los clientes conectados en tiempo real.   En flujo alterno:   1. El sistema despliega el mensaje de datos inválidos. 2. El sistema despliega el mensaje de juego lleno.   Excepciones:   1. El sistema despliega un mensaje de error de conexión. |
| **Incluye:** | Ninguno. |
| **Extiende:** | Ninguno. |
| **Prioridad:** | Media. |
| **Reglas de negocio:** | RN1. El sistema solo permite registrar un máximo de 10 jugadores. |

CU2 – Jugar Snake

|  |  |
| --- | --- |
| **ID:** | CU2 |
| **Nombre:** | JugarSnake |
| **Autor(es):** | Luis Gerardo Bonilla Ramírez |
| **Fecha de creación:** | 5 / Junio / 2019 |
| **Fecha de actualización:** | 5 / Junio / 2019 |
| **Descripción:** | El jugador envía movimientos al servidor para que este actualice las posiciones en la lista y la envíe a todos los usuarios. Además, se detectan los casos de colisiones e incremento por comida. |
| **Actor:** | Jugador |
| **Frecuencia de uso:** | Al menos 1 vez al día |
| **Disparador:** | El sistema detecta que el registro en la lista de serpientes fue hecho correctamente. |
| **Precondiciones:** | PRE-01. La serpiente deberá estar registrada en la lista de serpientes. |
| **Flujo normal:** | 1. El sistema recibe la COMIDA del servidor y la pinta en el TABLERO. El sistema recibe la lista de SERPIENTES del servidor y pinta todas las serpientes y está a la espera de eventos de teclado. 2. El actor pulsa cualquier tipo de tecla. (ver FA01) 3. El sistema detecta una pulsación de tecla. (ver FA02, FA03, FA04) (ver EX01) 4. Volver al paso 1. |
| **Flujo alterno:** | FA01. Irse de la partida.   1. El sistema cierra la ventana y abre una nueva con la pantalla MenuPrincipal. 2. Fin del caso de uso.   FA01. Movimiento de la serpiente.   1. Ir al caso de uso CU4. Detectar Movimiento.   FA03. Colisión de la serpiente.   1. Ir al caso de uso CU5. Detectar Colisión.   FA04. Detectar comida.   1. Ir al caso de uso CU3. Detectar comida. |
| **Excepciones:** | EX01. Error al conectar con el servidor.   1. El sistema no logra enviar la información al servidor y despliega un mensaje “El servidor no está funcionando”, cierra la ventana y abre una nueva con la pantalla MenuPrincipal. 2. Fin del caso de uso. |
| **Postcondiciones:** | En flujo alterno.   1. El sistema abre una nueva pantalla MenuPrincipal.   En excepciones.   1. El sistema despliega un mensaje y abre una nueva pantalla MenuPrincipal. |
| **Incluye:** |  |
| **Extiende:** |  |
| **Prioridad:** | Alta |

CU03 – Detectar Comida.

|  |  |
| --- | --- |
| **ID:** | CU03 |
| **Nombre:** | Detectar Comida. |
| **Autor(es):** | Luis Gerardo Bonilla Ramírez |
| **Fecha de creación:** | 5 / Junio / 2019 |
| **Fecha de actualización:** | 5 / Junio / 2019 |
| **Descripción:** | El servidor detecta una colisión de comida para incrementar el tamaño de la serpiente y generar una nueva en una posición libre diferente. |
| **Actor:** | Servidor |
| **Frecuencia de uso:** | Al menos 1 vez cuando se esté jugando. |
| **Disparador:** | El sistema detecta la cabeza de la SERPIENTE en la misma posición de la COMIDA. |
| **Precondiciones:** | PRE-01. La serpiente deberá estar registrada en la lista de serpientes. |
| **Flujo normal:** | 1. El servidor incrementa el cuerpo de la SERPIENTE que detectó la colisión de la comida y actualiza la lista de SERPIENTE y envía la nueva lista. Incrementa la puntuación de la SERPIENTE. El servidor borra la COMIDA y genera una nueva en una posición libre diferente y envía la posición de la nueva comida. (ver EX01) 2. Termina caso de uso, volver al caso de uso CU2 – Jugar Snake |
| **Flujo alterno:** |  |
| **Excepciones:** | EX01. El sistema no logra enviar información al cliente.   1. El sistema termina y deja de enviar información a los jugadores. |
| **Postcondiciones:** | En flujo normal:  POST-01. El sistema incrementa el cuerpo de la SERPIENTE.  POST-02. El sistema genera una COMIDA.  En excepciones:  POST-01. El sistema termina. |
| **Incluye:** |  |
| **Extiende:** | CU02 - JugarSnake |
| **Prioridad:** | Alta |

CU04 – Detectar movimiento

|  |  |
| --- | --- |
| **ID:** | CU04 |
| **Nombre:** | Detectar movimiento |
| **Autor(es):** | Luis Gerardo Bonilla Ramírez |
| **Fecha de creación:** | 5 / Junio / 2019 |
| **Fecha de actualización:** | 5 / Junio / 2019 |
| **Descripción:** | El sistema detecta un movimiento en el servidor y cambia la dirección de la SERPIENTE. |
| **Actor:** | Servidor. |
| **Frecuencia de uso:** | Al menos 10 veces cuando se esté jugando. |
| **Disparador:** | El sistema detecta la presión de una tecla. |
| **Precondiciones:** | PRE-01. La serpiente deberá estar registrada en la lista de serpientes. |
| **Flujo normal:** | 1. El Servidor detecta que la tecla presionada fue alguna de las siguientes: arriba, abajo, derecha o izquierda. Actualiza la dirección de la serpiente de acuerdo la tecla presionada. (ver FA01) (ver EX01) 2. Termina caso de uso, volver al caso de uso CU2 – Jugar Snake. |
| **Flujo alterno:** | FA01. Tecla presionada no autorizada.   1. El Servidor detecta que la tecla presionada no es ninguna de las esperadas. 2. Fin del caso de uso, volver al CU2 – Jugar Snake |
| **Excepciones:** | EX01. El Servidor no logra enviar información al cliente.  El sistema termina y deja de enviar información a los jugadores. |
| **Postcondiciones:** | En flujo normal:  POST-01. El servidor actualiza la dirección de la SERPIENTE.  En excepciones:  POST-01. El sistema termina. |
| **Incluye:** |  |
| **Extiende:** | CU02 – JugarSnake |
| **Prioridad:** | Alta |

CU05 – Detectar colisión

|  |  |
| --- | --- |
| **ID:** | CU05 |
| **Nombre:** | Detectar colisión. |
| **Autor(es):** | Luis Gerardo Bonilla Ramírez |
| **Fecha de creación:** | 5 / Junio / 2019 |
| **Fecha de actualización:** | 5 / Junio / 2019 |
| **Descripción:** | El servidor detecta que la serpiente ha tenido una colisión con el cuerpo de alguna serpiente o con el borde del escenario para eliminar a la serpiente de la lista y liberar un espacio en el juego. |
| **Actor:** | Servidor |
| **Frecuencia de uso:** | Al menos 1 vez en el juego. |
| **Disparador:** | El sistema detecta una colisión |
| **Precondiciones:** | 1. El servidor elimina de la lista de SERPIENTES la serpiente que tuvo la colisión y libera el espacio de jugadores disponibles en el juego. (ver EX01) 2. Termina caso de uso, volver al CU02 – Jugar Snake |
| **Flujo normal:** |  |
| **Flujo alterno:** |  |
| **Excepciones:** | EX01. El Servidor no logra enviar información al cliente.  El sistema termina y deja de enviar información a los jugadores. |
| **Postcondiciones:** | En flujo normal:  POST-01. El servidor elimina la SERPIENTE de la lista. |
| **Incluye:** |  |
| **Extiende:** | CU02 – Jugar Sanke |
| **Prioridad:** | Alta |

|  |  |
| --- | --- |
| **ID:** |  |
| **Nombre:** |  |
| **Autor(es):** |  |
| **Fecha de creación:** |  |
| **Fecha de actualización:** |  |
| **Descripción:** |  |
| **Actor:** |  |
| **Frecuencia de uso:** |  |
| **Disparador:** |  |
| **Precondiciones:** |  |
| **Flujo normal:** |  |
| **Flujo alterno:** |  |
| **Excepciones:** |  |
| **Postcondiciones:** |  |
| **Incluye:** |  |
| **Extiende:** |  |
| **Prioridad:** |  |