**Análisis y Diseño de Software**

**Taller 3: Laberinto Análisis**



**Realizado por:**

|  |  |  |
| --- | --- | --- |
| **Persona** | **Rol** | **Código Uniandes** |
| Carlos Ernesto González Vargas | Líder del Grupo | 200819123 |
| Sandra Milena Gómez Ríos | Líder de Planeación | 201110951 |
| Andrés Mauricio Erazo Benavides | Líder de Soporte | 201110949 |
| David Pérez Chibuque | Líder de Calidad | 201117818 |
| Willian Alejandro Idrobo Luna | Líder de Desarrollo | 201110544 |
| Erik Fernando Arcos Franco | Líder de Desarrollo | 201110856 |

**Índice de Contenido**

[**1.** **Objetivos** 4](#_Toc304934365)

[**2.** **Identificación y descripción de stakeholders** 4](#_Toc304934366)

[**3.** **Descripción Del Juego** 4](#_Toc304934367)

[**4.** **Atributos de calidad / Restricciones** 9](#_Toc304934368)

[**5.** **Diagrama de contexto** 10](#_Toc304934369)

[**6.** **Reglas de juego** 10](#_Toc304934370)

[**7.** **Diagramas del sistema** 11](#_Toc304934371)

[**8.** **Diagramas del jugador** 13](#_Toc304934372)

[**9.** **Diagrama de Despliegue** 15](#_Toc304934373)

[**10.** **Lecciones Aprendidas** 16](#_Toc304934374)

[**11.** **Conclusiones** 16](#_Toc304934375)

**Índice de Tablas**

[Tabla 1. Stakeholders 4](#_Toc304934493)

[Tabla 3. Reglas del juego 10](#_Toc304934494)

[Tabla 3. Descripción de entidades 12](#_Toc304934495)

[Tabla 4. Elementos del juego 14](#_Toc304934496)

**Índice de Figuras**

[Figura 1. Estados jugador 5](#_Toc304934424)

[Figura 2. Actividades Sistema 6](#_Toc304934425)

[Figura 3. Control turno 7](#_Toc304934426)

[Figura 4. Estados turno 8](#_Toc304934427)

[Figura 5. Ingreso al juego 8](#_Toc304934428)

[Figura 6. Salir del juego 9](#_Toc304934429)

[Figura 7. Diagrama de contexto 10](#_Toc304934430)

[Figura 8. Diagrama de entidades 11](#_Toc304934431)

[Figura 9. Atributos e inventario jugador 13](#_Toc304934432)

[Figura 10. Jugadores en laberinto 14](#_Toc304934433)

[Figura 11. Visualización del laberinto 15](#_Toc304934434)

[Figura 12. Diagrama de despliegue 15](#_Toc304934435)

**Análisis y Diseño de Software**

**Taller 3: Laberinto Análisis**

1. **Objetivos**

* Analizar y entender el contexto en el que ocurre la necesidad actual.
* Realizar un modelo del mundo del aplicativo.
* Detallar las reglas de juego del aplicativo
* Preparar el modelo a alto nivel para realizar el diseño del mismo posteriormente.

1. **Identificación y descripción de stakeholders**

Los Stakeholders del estudio son descritos a continuación

Tabla 1. Stakeholders

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Stakeholder** | **Descripción** | **Responsabilidades** |
| **S1** | Arquitecto | Persona o grupo de personas que recibirán el análisis de la necesidad para diseñar la solución | Desarrollar el diseño arquitectural del sistema esperado |

1. **Descripción Del Juego**

El sistema bajo análisis es la base para el inicio del diseño de un motor de soporte para una línea de juegos basados en el concepto de laberinto, el cual debe permitir a los usuarios jugar en red.

El juego inicia con un laberinto que puede ser recorrido por un conjunto de elementos nombrados elementos activos, que pueden estar asociados a un jugador o tener sus propios algoritmos de decisión. Adicionalmente, existen elementos pasivos distribuidos en el laberinto que modifican las características de los elementos activos cuando estos los consumen. En cada posición del laberinto pueden permanecer cualquier número de elementos pasivos y cualquier número de elementos activos que no sean agresivos entre sí, cuando dos elementos activos que son agresivos entre si se encuentran, se inicia una pelea que lleva a la salida de alguno de los elementos.

Cada jugador en el laberinto tiene un turno para realizar un movimiento valido, de acuerdo a la posición y el estado en el que se encuentre, el sistema realiza los turnos en paralelo y solo les da espera a los jugadores que afectan la jugada de otro jugador. El laberinto es infinito y se actualiza a medida que se realiza el juego, incluyendo elementos pasivos, elementos activos relacionados a jugadores y elementos activos con algoritmos propios.

****

Figura 1. Estados jugador

Un jugador se encuentra en alguno de los siguientes estados de acuerdo a las acciones que se realicen durante el juego. Inicialmente el jugador para comenzar el juego debe **ingresar**, en donde se le asigna un elemento activo, unos elementos pasivos y se ubica dentro del laberinto en una posición, allí se le asigna un turno y se pone en estado de **espera** para ser incluido en el próximo turno.

El jugador permanece en el estado de **espera** hasta que se le asigna el turno de jugar para realizar su movimiento, allí el jugador puede pasar al estado en **movimiento** o **ataque** dependiendo de si se encuentra en una pelea o solo está recorriendo el laberinto.

En el estado en **movimiento** el jugador puede consumir elementos pasivos y recorrer el laberinto, moviéndose en su turno y regresando al estado en **espera**. El estado **ataque** ocurre cuando se llega a una posición donde existen más elementos activos que son agresivos con respecto a él, en este estado el jugador solo puede decidir **atacar**. Cuando se inicie el ataque este finaliza hasta que a alguno de los elementos en la pelea se le terminen su puntos de vida, al finalizar los puntos de vida el jugador pasa al estado **finalizado** para **salir** del juego. Adicionalmente desde el estado el movimiento el jugador puede decidir **salir** en cualquier momento y finalizar el juego.

****

Figura 2. Actividades Sistema

El sistema comienza generando un laberinto para poder iniciar el juego, después de iniciado el juego se pueden empezar a adicionar los jugadores por medio de la acción de ingresar y en cualquier momento los jugadores pueden salir del juego. De manera paralela el sistema lleva control de los turnos asignando el turno para jugar de acuerdo a las dependencias entre jugadores y finalmente actualiza el laberinto con los nuevos ingresos, salidas y las actualizaciones que el sistema realiza sobre el laberinto.

****

Figura 3. Control turno

Para asignar el turno a un jugador el sistema verifica el estado del jugador y sus posibles movimientos válidos, una vez se da el turno al jugador se espera sus jugadas un tiempo T, si no se recibe respuesta se asume que no se realiza ningún movimiento, si el jugador se encontraba en estado de pelea el sistema evalúa si hubo un ganador para llamar la salida del perdedor y asignar los elementos pasivos del perdedor al ganador.

****

Figura 4. Estados turno

Cuando un jugador tiene un turno se valida el estado en el que esta y de acuerdo a este se tienen diferentes acciones, si el jugador se encuentra en medio de una pelea, solo puede atacar, mientras que si se encuentra en movimiento el jugador puede almacenar los elementos pasivos de la posición en la que se encuentra, consumir elementos pasivos almacenados y realizar el movimiento en este orden.

****

Figura 5. Ingreso al juego

En cualquier momento se puede realizar el ingreso de un jugador, este se lleva a cabo estableciendo un elemento activo para el jugador, ya sea buscando entre los jugadores conocidos o ingresando uno nuevo, asignado su nombre, una vez se asigna el jugador activo, se le asignan los elementos pasivos, se sitúa el jugador en una posición del laberinto y se le asigna el siguiente turno de juego para ser incluido en el próximo ciclo del sistema.

****

Figura 6. Salir del juego

Finalmente, en cualquier momento se puede realizarla salida de un jugador, esto incluye almacenar su estado, liberar los elementos pasivos que posea en ese momento dentro del laberinto en la posición actual y realizar las acciones para desconectarlo del sistema.

1. Atributos de calidad / Restricciones

* En el juego participaran miles de jugadores de manera simultánea
* Los turnos se deben realizar en paralelo respetando el orden pero aprovechando los casos en los que las jugadas de una persona son independientes a las jugadas de otra.
* Se le asigna a cada jugador un tiempo T para enviar la jugada, sino ha jugado en ese lapso se asume que decidió no moverse.
* Solo los jugadores que se encuentran en la zona de influencia de otro jugador deben esperar resultado de otras jugadas.
* La visualización de cada jugador es de 20 x 20 de su posición
* La memoria de un jugador tiene un límite de 100 turnos
* EL sistema debe ser altamente flexible y adaptable
* El sistema debe soportar varias plataformas de tecnología

1. Diagrama de contexto

****

Figura 7. Diagrama de contexto

El motor a diseñar debe contar con la posibilidad de soportar jugadores en línea. Estos usuarios pueden acceder al juego a través de un la cuenta que se tenga configurada en el sistema del juego, la cual es verificada contra el sistema LDAP. Luego de esta verificación el jugador comienza a participar en el laberinto a medida que le corresponda su turno. A medida que se desarrolla el juego, este genera diferentes datos los cuales sirven para ser consultados y llevar estadísticas de los jugadores y el estado del juego.

1. Reglas de juego

Tabla 3. Reglas del juego

|  |  |
| --- | --- |
| **ID** | **Descripción** |
| **R001** | Un jugador siempre debe estar en unas coordenadas X, Y sobre un corredor. |
| **R002** | En cada punto del laberinto puede haber cero o más elementos, estos pueden ser del tipo (activo, pasivo). |
| **R003** | Elementos activos pueden ser agresivos o no agresivos entre ellos. |
| **R004** | En un mismo punto puede haber cualquier cantidad de elementos pasivos y activos, mientras no sean agresivos entre ellos. |
| **R005** | Los elementos pasivos pueden almacenarse para usarse más tarde. |
| **R006** | Un elemento de juego pude estar asociado con un jugador o ser independiente utilizar sus propios algoritmos para la toma de decisiones. |
| **R007** | Si dos elementos activos, agresivos entre sí, tratan de ocupar el mismo espacio, se produce una pelea. |
| **R008** | En la pelea siempre sale uno de los elementos cuando la vida de uno de los dos es menor o igual a cero. |
| **R009** | La pelea es iniciada por el atacante (El que ingresa al espacio) |
| **R010** | En cada iteración se le resta al nivel de vida del atacado (el nivel de ataque del atacante – nivel de defensa). |
| **R011** | Al final de la pelea el ganador obtiene los puntos de ataque y defensa del perdedor. |
| **R012** | Si en un punto el elemento se encuentra más un elemento agresivo respecto a él, pelea con cada uno de ellos en orden aleatorio, pero cada uno de ellos utiliza en cada pelea, la suma de niveles de defensa de los elementos atacados. |
| **R013** | Cada elemento activo se puede mover una posición en cualquier dirección válida en cada turno. |
| **R014** | En un turno el jugador tiene derecho a consumir cualquier número de elementos pasivos y hace un único movimiento válido. |
| **R015** | Deben poder aparecen elementos pasivos y activos durante la ejecución del juego |

1. Diagramas del sistema

****

Figura 8. Diagrama de entidades

Tabla 3. Descripción de entidades

|  |  |  |
| --- | --- | --- |
| **Entidad** | **Descripción** | **Atributos y asociaciones** |
| **Laberinto** | Representa el laberinto | * **tiempoTurno:** Tiempo que toma un turno de un jugador en el laberinto * **[1..\*] contiene:** Conjunto de corredores que contiene el laberinto |
| **Corredor** | Representa un corredor del laberinto | * **puntoInicio:** Es la coordenada de inicio del corredor * **puntoFin:** Es la coordenada de finalización del corredor * **[\*] contiene:** Conjunto de coordenadas que hacen parte de un corredor |
| **Coordenada** | Representa un punto en el laberinto el cual puede estar o no ocupado por un elemento | * **puntoX:** Es el valor en el eje X de la coordenada en el laberinto * **puntoY:** Es el valor en el eje Y de la coordenada en el laberinto * **estaOcupado:** Determina si el punto está o no está ocupado por otro elemento activo. Necesario para ubicar a un jugador nuevo en el laberinto * **[0..\*] tiene:** Son los elementos, activos o pasivos que se encuentran en el la coordenada del laberinto |
| **Elemento** | Representa un elemento, activo o pasivo que se encuentra en el laberinto | * **esPasivo:** Especifica si la instancia del elemento es un elemento pasivo o no lo es * **símbolo:** Es el símbolo del elemento * **[1] ocupa:** Es la coordenada que esta ocupando el elemento en un determinado instante en el juego * **[1] pertenece:** Es el laberinto al cual pertenece el elemento. |
| **Elemento Pasivo** | Representa un elemento pasivo del juego | * **modificadorPermanenteAtaque:** Modifica permanentemente el atributo de ataque del elemento activo que lo toma o usa. * **modificadorPermanenteDefensa:** Modifica permanentemente el atributo de defensa del elemento activo que lo toma o usa. * **modificadorPermanenteVida:** Modifica permanentemente el atributo de vida del elemento activo que lo toma o usa. * **modificadorTemporalAtaque:** Modifica temporalmente el atributo de ataque del elemento activo que lo toma o usa. * **modificadorTemporalDefensa:** Modifica temporalmente el atributo de defensa del elemento activo que lo toma o usa. * **modificadorTemporalVida:** Modifica temporalmente el atributo de vida del elemento activo que lo toma o usa. * **[es]:** Especifica que el elemento pasivo hereda de un elemento |
| **Elemento Activo** | Representa a un elemento activo del juego | * **ataque:** Nivel de ataque del elemento activo * **defensa:** Nivel de defensa del elemento activo * **vida:** Nivel de vida del elemento activo * **[0..\*] tiene:** Representa los elementos pasivos que el elemento activo posee |
| **Jugador** | Representa a un elemento activo que es un jugador humano | * **nombre:** Es el nombre del jugador * **[es]:** Especifica que el jugador hereda de un elemento activo. |
| **Bot** | Representa a un elemento activo que es independiente (no humano) | * **id:** Es el identificador del elemento independiente * **[es]:** Especifica que el jugador hereda de un elemento activo |

1. Diagramas del jugador

**Capacidades de un jugador**

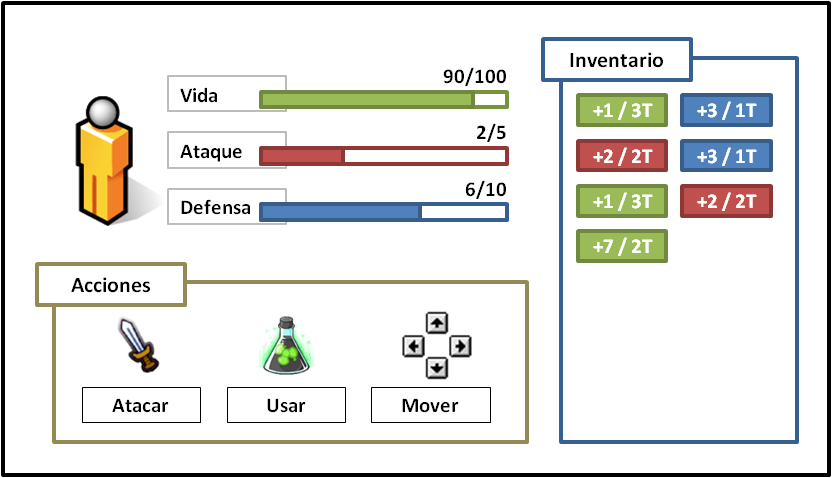


Figura 9. Atributos e inventario jugador

La imagen muestra las capacidades generales que describen a un jugador dentro del laberinto así como las acciones que puede realizar, en el diagrama anterior se tiene que:

* Un jugador se define por 3 características diferentes: Vida, Ataque y Defensa
* Un jugador inicia con un valor de 100 de vida, 1 de ataque y 1 de defensa, estos valores se pueden alterar por medio de elementos pasivos.
* Un jugador puede realizar diferentes acciones durante su turno: Atacar, Usar un ítem del inventario o moverse
* Un ítem, o elemento pasivo puede ser almacenado por el usuario para ser usado posteriormente y se puede ver en su inventario.
* Los elemento pasivo puede alterar cada una de las características de un jugador: Vida, Ataque y Defensa, definidas en el diagrama por los colores Verde, Rojo, Azul respectivamente, de esta manera un elemento pasivo altera cada una de estas características durante una determinada cantidad de turnos.

**Definición de laberinto:**

El funcionamiento de un laberinto se puede definir por medio de la siguiente imagen:

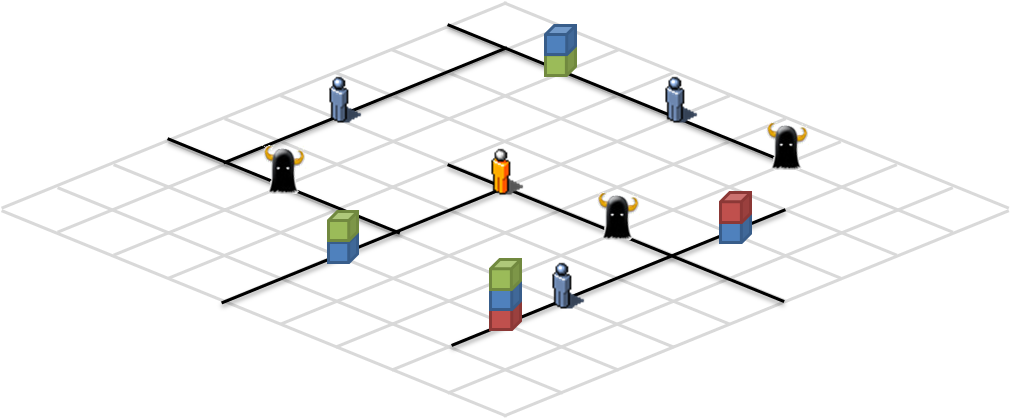


Figura 10. Jugadores en laberinto

Donde cada uno de los elementos representa lo siguiente:

Tabla 4. Elementos del juego

|  |  |
| --- | --- |
| **Elemento** | **Definición** |
|  | Elemento pasivo de modificación de Vida, puede haber varios elementos pasivos en una misma posición. |
|  | Elemento pasivo de modificación de Defensa, puede haber varios elementos pasivos en una misma posición. |
|  | Elemento pasivo de modificación de Ataque, puede haber varios elementos pasivos en una misma posición. |
|  | Elemento agresivo controlado por el sistema |
|  | Jugador |
|  | Elemento agresivo controlado por otro jugador |
|  | Corredor dentro del laberinto |

Aunque el sistema conoce la distribución de todos los elementos, el jugador no conoce el estado actual del laberinto, de hecho solo conoce los caminos por los que ha recorrido, y su visualización es de 20x20 cuadros a su alrededor y solo recuerda sus últimos 100 turnos, de esta manera, para un jugador la visualización del laberinto en un momento dado podría ser la siguiente:

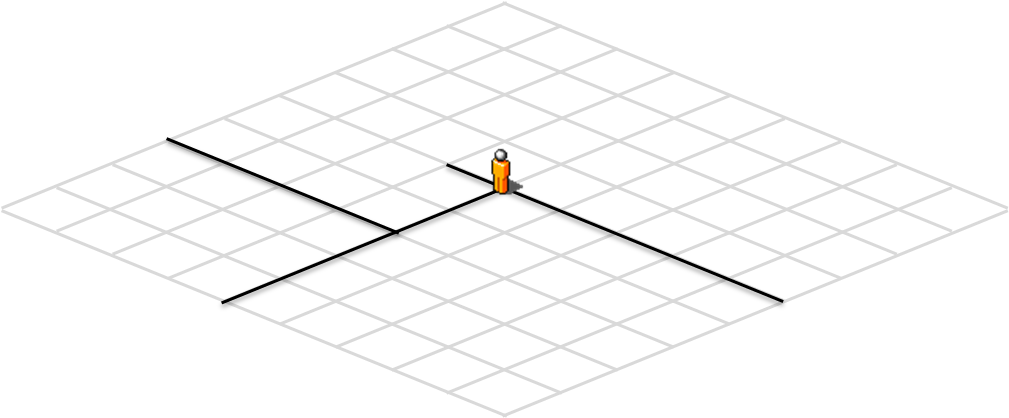


Figura 11. Visualización del laberinto

1. **Diagrama de Despliegue**

****

Figura 12. Diagrama de despliegue

El servidor principal inicia con la ejecución del sistema del juego el cual provee los servicios de juego. Desde este servidor se envía la información generada durante el juego hacia el Data Warehouse donde finalmente se guarda.

Luego a través del navegadores de los dispositivos del cliente, este accede al sistema después de validar sus credenciales contra el sistema LDAP, de esta forma inicia una nueva partida en el juego.

Finalmente, se cuenta con la posibilidad de que sitios externos puedan conectarse al Data Warehouse para la realización de consultas de la información disponible, la cual es tomada para su publicación en los respectivos sitios.

1. **Lecciones Aprendidas**

* A pesar de realizar gran cantidad de diagramas, el escenario no parece estar completo, por lo cual se hace necesario el uso de matrices, tablas, descripciones y demás herramientas que complementen lo que los diagramas no son capaces de mostrar.
* Usar diagramas ad-hoc para representar el contexto del problema funciona de una mejor manera que representarlo por diagramas o notaciones estándares o convencionales.

1. **Conclusiones**

* En el desarrollo del análisis de este problema fue necesario crear vistas del juego y apoyarse con diagramas de estado y  actividad que representaran las posibles acciones del jugador, para poder mostrar un escenario completo del juego en cada posible jugada, esta aproximación pretende representar los estados del jugador, el sistema y como se modifica tras las posibles jugadas.
* El diseño de un sistema no es un proceso definido estricta y formalmente, sino que depende bastante del contexto en el cual se encuentre el problema. Por ejemplo, en el caso particular del ejercicio del laberinto no existía cierta información que quizás en el diseño de otros sistemas debería ser no solo explicita sino también necesaria, por ejemplo, los actores, los stakeholders, los atributos de calidad o el proceso que se está tratando de modelar. Por este motivo, los entregables de cada análisis no necesariamente son los mismos ni tienen que desarrollarse en el mismo orden, sino que van determinados por el sistema que se quiere analizar.
* El análisis de un sistema necesita de una parte creativa para crear artefactos que sirvan de apoyo al momento de comunicar una idea con gráficos y texto de forma que sea clara para el lector.
* Los artefactos generados para el análisis y diseño de un sistema dependen directamente del sistema que se está analizando, no se pueden forzar artefactos que no corresponden y en ocasiones es necesario definir nuevos diagramas y convenciones para poder transmitir la idea.