

**VIDEO JUEGO  
NINJAWARRIONG**

**SERGIO STEVEN LEMUS LEMUS.  
EDWARD FELIPE LEON COTACIO**

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INFORMÁTICA Y ELECTRÓNICA  
PROGRAMA DE TECNOLOGÍA EN INFORMÁTICA  
BOGOTÁ  
I-2014**

**VIDEO JUEGO  
NINJAWARRIONG**

**SERGIO STEVEN LEMUS LEMUS.  
EDWARD FELIPE LEON COTACIO**

**Trabajo de grado para optar al título  
de tecnólogo en informática**

**Asesor:  
ING. Sócrates Rojas Amador**

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INFORMÁTICA Y ELECTRÓNICA  
PROGRAMA DE TECNOLOGÍA EN INFORMÁTICA  
BOGOTÁ  
I-2014**

**Nota de aceptación**

---

---

---

---

---

---

**Firma del presidente del jurado**

---

**Firma del jurado No. 1**

---

**Firma del jurado No. 2**

Bogotá D.C. / 14 / Enero de 2014

### **Dedicatoria**

Dedicó a Dios por permitimos culminar los estudios de educación superior, y a nuestros padres por apoyamos incondicionalmente a lo largo de este camino que hemos construido y por su apoyo para hacernos crecer como seres.

### **Agradecimientos**

Agradecemos a los profesores los que estuvieron a lo largo de este proceso de aprendizaje. A la corporación Universitaria Minuto de Dios, la cual nos brindo posibilidad de estudio con una excelente formación.

## TABLA DE CONTENIDO

	<b>Pág.</b>
<b>INTRODUCCIÓN</b>	11
1.1 Titulo	12
1.2 Planteamiento del problema	12
1.3 Alcances y justificación	13
1.4 Objetivos	14
1.4.1 Objetivos General	14
1.4.2 Objetivos Específicos	15
<b>2. Ingeniera del proyecto</b>	16
2.1 Modelo de desarrollo	16
2.2 Documentación e índice del desarrollo	17
<b>3. Análisis y diseño</b>	21
3.1 Definición de Requerimientos	21
3.1.1 Requerimientos Funcionales	21
3.1.2 Requerimientos No Funcionales	23
3.2 Descripción del sistema propuesto	24
3.3 Diseño del sistema propuesto	27
<b>4. Desarrollo</b>	36
4.1 Especificaciones técnicas	36
4.1.1 Software	36
4.1.2 Hardware	37
<b>5. Glosario</b>	39
<b>6. Conclusiones</b>	40
<b>7. Bibliografía</b>	41
<b>8. Manual</b>	42
8.1 Manual del sistema	43
8.2 Manual del usuario	52
<b>9. Anexos</b>	64

## LISTAS DE DIAGRAMAS

	<b>Pág.</b>
Diagrama No 1: Diagrama de clases .....	28
Diagrama No 2: Diagrama de casos de uso del usuario .....	33
Diagrama No 3: Diagrama de casos de uso general.....	33
Diagrama No 4: Diagrama de secuencia .....	35

## LISTA DE TABLAS

	<b>Pág.</b>
Tabla No 1: Caso de uso Ingresar al juego .....	29
Tabla No 2: Caso de uso Escoger nivel .....	30
Tabla No 3: Caso de uso Ayuda .....	31
Tabla No 4: Caso de uso Salir del juego .....	32



## TABLA DE INTERFACES

	<b>Pág.</b>
Interfaz No 1: Menu de inicio.....	25
Interfaz No 2: Selección de dificultad .....	25
Interfaz No 3: Partidas .....	26
Interfaz No 4: Instrucciones del juego.....	27
Interfaz No 5: Boceto personaje principal.....	64
Interfaz No 6: Boceto nube de ventana principal .....	65
Interfaz No 7: Boceto edificios utilizados .....	66
Interfaz No 8: Bocetos casas utilizadas .....	67
Interfaz No 9: Boceto boton para regresar .....	68
Interfaz No 10: Boceto de fondo .....	69

## RESUMEN

Este es el documento de diseño de Ninja Warring. El video juego para la red que ejemplifica todos los contenidos de construct 2, es un gamemaker sobre desarrollo de video juego en 2D. Este escrito tiene como objetivo principal plasmar los elementos que debe incluir Ninja Warring y servir de carta de presentación en caso de buscar colaboradores en el futuro.

Ninja Warring es un video juego en el que controlamos a un ninja que debe Seguir por varios mapas para poder lograr salvar a su clan del exterminio total mientras sus compañeros son tomados a la fuerza por diferentes clanes. Durante su travesía, él se encontrará con subordinados de un clan (cada mapa maneja un clan diferente) y nuestro ninja debe detener y eliminar el clan a toda costa utilizando sus limitadas habilidades.

El presente proyecto utilizó como modelo de desarrollo scrum, el cual se basa en un proceso de iteraciones como también incremental comúnmente usado en entornos basados en desarrollo ágil de software, también está enfocado a la gestión de procesos de desarrollo de software, o en aproximación de gestión de programas, por otra parte, se utiliza UML(Lenguaje De Modelado Unificado) donde se usa para la sucesión de una serie de métodos de análisis y diseño orientadas a objetos para especificar o describir métodos o procesos.

## INTRODUCCION

El presente proyecto de gradó trata esencialmente el diseño y construcción de un software como herramienta para el desarrollo y distracción en el ámbito de los videojuegos, con un software que manipule eficazmente motores de física con el objetivo de lograr una mejor apariencia visiblemente amigable al usuario, mayor funcionalidad y menores requerimientos.

El objetivo del diseño de software orientado a videojuegos es construir ambientes virtuales utilizando las diferentes herramientas para el desarrollo de aplicaciones, funcionales a la actividad para la cual, sea dirigida y a la vez eficiente en relación a los recursos disponibles para tal actividad. De esta manera el presente proyecto utilizó UML como un guía para el modelo de desarrollo, los cuales permiten una completa implementación de los tipos de diagramas donde es posible realizar una clara apreciación de los actores previamente identificados en los diagramas estructurales y así tener un sistema completo.

El proyecto se desarrollo principalmente para la facultad de ingeniería, con el fin de utilizar las diferentes herramientas expuestas a lo largo de nuestra formación, demostrando así capacidades de desarrollo en diferentes aplicaciones que en algún momento dado serán utilizadas por diferentes usuarios.

## **1.1 Título del proyecto**

**Ultra Games – D:** Semillero de investigación facultad de ingeniería

**Ninja Warring:** Nombre del videojuego

## **1.2 Planteamiento del problema.**

El desarrollo de video juegos se debe plantear desde un inicio en la universidad impulsando la creación y fortalecimiento de programas relacionados con este y así poder contar con profesionales formados no solo con lo respectivo de su carrera, adicionalmente utilizando herramientas externas que le brinda la corporación.

En Colombia se puede avanzar en esta idea ya que se cuenta con carreras de ingeniería y diseño, pero ninguna de las dos carreras ve el desarrollo de video juegos como una oportunidad de aprender sobre otras ramas de la ingeniería y diseño a profundidad como lo solicita un mercado emergente.

El reto es realizar un buen software, un video juego el cual, tendrá características específicas como un entorno interactivo amigable, el cual está compuesto por mapas y niveles en donde el juego desafía a el jugador para poder sobrepasar estos niveles los cuales van aumentado de dificultad a medida que se avanza en el transcurso del los mapas. El juego pretende que el jugador descubra como salir de la problemática planteada en cada mapa, pero sin dejar a un lado los obstáculos que la aplicación propone en cada nivel. Para este desarrollo se requiere cierto grado de experiencia, los conocimientos en game design, probar mecánicas, motores de física, motores de colisión y muchos otros recursos con respecto a la creación del software, es un reto todavía mucho más fuerte.

En la actualidad está comenzando a plantear ideas acerca de este campo donde universidades, gobierno, el sector privado y la comunidad estén integradas para ir en pro del desarrollo de la industria de los video juegos.

Además, el constante movimiento del internet móvil y el auge de celulares inteligentes, tabletas. Los cuales juegan papeles cada vez más importantes en la industria por su fácil interacción, no obstante, el mundo de las consolas y de los videos juegos asociados continúa en un proceso de renovación.

Aunque los jugadores de las consolas seguirán siendo una parte clave en el mundo de los videos juegos, el uso de los móviles inteligentes ha permitido que surjan nuevas clases de jugadores de tipo casual que puedan jugar durante la espera de una actividad a realizar o en tiempos libres y en cualquier lugar.

### **1.3 Alcances y Justificación**

Inicialmente el software estará compuesto por dos niveles con sus respectivos actores (enemigos) que harán la historia más interesante siendo los rivales directos del jugador, utilizando una interfaz donde el usuario interactúa con el software manejando factores como armas, poderes, eligiendo la partida, haciendo más cómodo e interesante su uso.

Tendrá una interface amigable con el jugador, tendrá unas ciertas opciones como selección de nivel, partidas guardadas, recompensas desbloqueadas, selección de personajes y modo de juego.

En el área de los gráficos como los mapas de juego estan constituidos por una interface llamativa de acuerdo con el escenario en que este jugando, con el fin que la persona observe esos detalles y se sienta atraído a seguirlo jugando además de una serie de eventos que irán sucediendo en el trascurso del juego como por ejemplo pruebas de memoria pequeñas para ganar beneficios en la partida o efectos comunes al ejecutar una acción.

Hoy por hoy los constantes avances tecnológicos, la implementación de creación de proyectos en el desarrollo de software y los diferentes medios de entretenimiento utilizados en la actualidad, la creación de video juegos representa un gran medio para acceder a todo tipo de géneros dentro de la sociedad, de esta manera a través de un juego es posible influir sobre la percepción de muchas personas, ya sea por fines de entretenimiento o fines educativos.

Adicionalmente se cuenta con varias investigaciones que afirman que los videojuegos desarrollan la coordinación de ojos y manos, esto repercute en el desarrollo cognitivo. Además desarrollan el pensamiento y la planificación lógica optimizando habilidades sicomotrices, como la orientación y la percepción.

Los videojuegos son una realidad social y en nuestro ámbito estudiantil una manera de impulsar y mejorar nuestras habilidades como desarrolladores de software sin llegar a volverse monótono por el uso constante de nuestra imaginación y creatividad a la hora de crear personajes y mundos digitales, cualquier avance relacionado o que se enfoque con este tema se convierte en un valor mediático de gran repercusión.

Teniendo en cuenta la constante evolución de la realidad virtual, y la realidad aumentada q hace más llamativo el modo de ver el mundo desde un ambiente virtual generando nuevas y novedosas ideas a los diferentes programadores que le proporcionaran con el tiempo más avances al mundo.

## **1.4 Objetivos**

### **1.4.1 Objetivo General.**

- Desarrollar un videojuego que permita mediante una interfaz amigable proveer entretenimiento, por medio de un personaje animado que vive una aventura entre mundos.

### **1.4.2 Objetivos Específicos.**

- Desarrollar un software que muestre a un ninja pasando por diferentes trampas y pruebas para aprobar el nivel.
- Aprovechar el potencial de las nuevas tecnologías y los diferentes dispositivos para la creación de software optimizando el desarrollo de videojuegos.
- Desarrollar mediante motores gráficos el diseño, la creación de un videojuego que nos acerque más a la realidad.

## **2. INGENIERIA DEL PROYECTO**

En esta fase de ingeniería del proyecto se da a conocer una completa descripción de la metodología a seguir para el modelamiento, diseño y desarrollo del videojuego

### **2.1 Modelo de Desarrollo**

La metodología a utilizar es Scrum, ya que es un modelo de referencia que define un conjunto de prácticas y roles que se pueden tomar como puntos de partida donde los requisitos son cambiantes o poco definidos y así la innovación, la competitividad, la flexibilidad y la productividad son fundamentales para definir un proceso de desarrollo que se ejecutará durante el proyecto, Scrum tiene como roles principales.

El ScrumMaster el cual, mantiene los procesos y trabajo de forma similar al director del proyecto.

El ProductOwner representa a los stakeholders (interesados externos o internos).

Por otra parte, el Sprint es el periodo en el cual, se lleva a cabo el trabajo en sí, en cada sprint el equipo deberá presentar los avances logrados y los resultados obtenidos durante cada iteración.

El proyecto se ejecutará en bloques temporales cortos y fijos, ya que cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo.

El proceso parte de la lista de objetivos, requisitos prioritarios del producto, que actúa como plan del proyecto. En la lista se priorizan los objetivos balanceando el valor que le aportan al proyecto y quedan repartidos en iteraciones y entregas.



Las actividades a realizar en el modelo de Scrum son planificación de las iteraciones las cuales contiene dos partes, selección de requisitos el cual se presenta una lista de requisitos priorizada del proyecto, se seleccionan los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados al final de cada iteración. Planificación de la iteración, se elabora una lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido.

Se va a desarrollar este proyecto mediante un ciclo de desarrollo iterativo e incremental de tipo Scrum para la ejecución de este proyecto. Sus principales características son:

Sistema modular. Las características del video juego permiten desarrollar una base funcional mínima y sobre ella ir incrementando las funcionalidades o modificando el comportamiento o apariencia de las ya implementadas.

Entregas frecuentes y continuas al cliente de los módulos terminados, de forma que puede disponer de una funcionalidad básica en un tiempo mínimo y a partir de ahí un incremento y mejora continua del Software.

Posibles dificultades en cuanto a requisitos.

- Es posible que el sistema incorpore más funcionalidades de las inicialmente identificadas.
- Es posible que durante la ejecución del proyecto se altere el orden en el que se desean recibir los módulos o historias de usuario terminadas.

### **Documentación e índice del desarrollo**

- Pila de producto o Product Backlog:

Muestra requisitos del sistema o del usuario, además el gestor de producto durante todo el proyecto.

Responsabilidades del gestor de producto

Responsabilidades del Scrum Manager

Responsabilidades del equipo técnico

Responsabilidades del resto de implicados

- Pila de sprint o Sprint Backlog

Documento de registro de los requisitos detallados o tareas que va a desarrollar el equipo técnico en la iteración.

Responsabilidades del gestor de producto

Responsabilidades del Scrum Manager

Responsabilidades del equipo técnico

- Sprint

Cada una de las iteraciones del ciclo de vida iterativo Scrum. La duración de cada sprint.

- Incremento

Entrega avance del proyecto al gestor del producto completamente terminado y operativo.

- Registro y seguimiento del avance.

### **Producto o Burn Up**

Representa los temas del sistema en el orden que se desean, y el tiempo en el que se prevé su ejecución.

### **Avance o Burn Down.**

Muestra el estado de avance del trabajo del sprint en curso.

- Comunicación y reporting directo.

### **Reunión de inicio de sprint**

Reunión para determinar las funcionalidades o historias de usuario que se van a incluir en el próximo incremento.

### **Reunión técnica diaria**

Puesta en común diaria del equipo con presencia del Coordinador del proyecto o Scrum Manager de duración máxima de 10 minutos.

### **Reunión de cierre de sprint y entrega del incremento**

Reunión para probar y entregar el incremento al gestor del producto.

Se planteará una completa implementación del lenguaje UML (Lenguaje Unificado de modelamiento, es un lenguaje diseñado para el análisis de los requerimientos, del comportamiento y de las interacciones del usuario con la aplicación, facilitando el desarrollo del juego más complejos y completos) que permitirá la estructuración de los requerimientos funcionales y no funcionales, de esta manera, se plantea un diseño estructurado, implementando los diagramas estructurales, en los que se puede identificar los mapas de la aplicación, donde se encuentra los diagramas de clases, objetos, estos permitirán hacer una inicial visualización del juego planteado, asimismo una identificación de los objetos y actores que pueden intervenir en el aplicativo.

Esto se llevará a cabo mediante el siguiente diagrama de UML:

- **Diagrama de clases**

Diagrama de clases en el que se muestran instancias específicas de clases (objetos).

Finalmente, mediante los diagramas dinámicos sugeridos por UML los cuales permiten visualizar, interacción del juego con el usuario y establecimiento de requerimientos funcionales, que facilitan el entendimiento de la estructura del videojuego, dentro de estos diagramas podemos destacar los casos de uso y de secuencia, los cuales permiten una completa implementación de los tipos de diagramas donde es posible realizar una clara apreciación de los actores previamente identificados en los diagramas estructurales y así tener un sistema completo.

- **Diagrama de casos de uso**

Diagrama de comportamiento del sistema al afrontar una tarea o un requisito.

- **Diagrama de secuencias**

Diagrama usado para modelar interacción entre objetos en un sistema en una aplicación.

### **3. ANÁLISIS Y DISEÑO**

El análisis y el diseño en un proyecto de ingeniería de software es un método en el cual se crea un conjunto de modelos utilizando una notación acordada para analizar los requerimientos de un contexto, con el fin de diseñar una solución para mejorar los procesos involucrados. Para este proyecto se van a trabajar los siguientes aspectos.

#### **3.1 Definición de Requerimientos**

Los requerimientos son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas que reflejan las necesidades de un sistema, habiendo determinado los alcances, los objetivos del sistema se determino que se van a trabajar los siguientes requerimientos.

##### **3.1.1 Requerimientos funcionales**

Son los requerimientos que definen las funciones que el sistema será capaz de realizar, además describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Dadas las características del levantamiento de información y depuración de datos se identificaron los siguientes requerimientos.

- El sistema debe proporcionar una introducción alusiva al tema del videojuego.
- El sistema debe pedir la creación de usuario para el acceso al videojuego.
- El sistema debe permitir eliminar un jugador ya registrado en el juego.

- El sistema debe verificar si el usuario ya está registrado antes de almacenarlo en la base de datos del videojuego.
- El sistema debe generar un menú con la información establecida para las opciones de juego descripción, reglamento, primer personaje para empezar a jugar, controles de manejo del juego, estado físico del juego.
- El sistema debe permitir un menú en donde este la selección de los niveles disponibles.
- El sistema debe permitir la selección y modificación de las opciones de juego.
- El sistema debe generar un reporte del estado del jugador mostrando la puntuación inicial, vida inicial y personaje seleccionado inicialmente.
- El sistema debe visualizar unos bonos extra obtenidos en el transcurso del juego.
- El sistema deberá proveer una opción de ayuda como consulta en el uso de la aplicación.
- El sistema debe proporcionar un menú en donde este la selección de opciones de juego disponibles en caso que el usuario desee suspender la aplicación.
- El sistema deberá proveer mecanismos de almacenamiento de información en caso que el usuario desee suspender la aplicación.
- El sistema deberá mostrar el menú inicio para salida de la aplicación en caso que el usuario desee finalizar el programa.

### 3.1.2 Requerimientos no funcionales

Son los requerimientos que se encargan de encontrar las características que pueden limitar el sistema, como por ejemplo, el rendimiento, interfaces de usuario, mantenimiento, seguridad, portabilidad, etc. Dadas las características del levantamiento de información y depuración de datos se identificaron los siguientes requerimientos.

- El sistema seguirá la metodología de desarrollo Scrum.
- El sistema será documentado mediante el lenguaje de modelamiento UML
- El sistema estará diseñado en una arquitectura que consta de un modelo, una vista y un controlador.
- El tiempo de carga de una ventana debe ser de máximo 10 segundos.
- El grupo de desarrollo se compone de dos personas; cada persona debe dedicarle al proyecto un promedio de doce (24) horas semanales.
- Se utilizará el lenguaje de programación PHP y como game maker construct 2.
- El Sistema Operativo para el desarrollo de la aplicación será Windows.
- Se deberá entregar toda la documentación del producto de software.
- La aplicación será entregada con datos de prueba para corroborar su correcto funcionamiento.
- Se debe contar con un navegador actualizado ya sea safari, chrome, opera.etc.

- Se deberá cumplir con todos los requerimientos propuestos.
- Se deberá entregar un documento final con todas las correcciones de las entregas parciales, el código fuente y ejecutable de la aplicación.
- Se debe incluir un archivo de texto léeme con las especificaciones de la aplicación.
- El equipo deberá tener como mínimo 512 Mb de memoria RAM.
- El equipo deberá tener como mínimo un procesador dual core 2 dúo para óptimo rendimiento.

### **3.2 Descripción del sistema propuesto**

Ninja Warring es un videojuego en 2D donde se deben destruir los enemigos para conseguir llegar al último nivel.

Este videojuego está orientado a la web y permite la interacción del ordenador con el usuario. El jugador podrá efectuar movimientos del personaje, con el fin de evitar obstáculos, además ejecutar una acción cuando se encuentre con algún personaje enemigo en el juego, así podrá ganar puntaje y avanzar al siguiente nivel.

Al iniciar el juego se muestra una interface donde podrá ejecutar diferentes opciones (interfaz No. 01) Menú inicio pág. 25





Interfaz No. 1. Menú inicio

**Jugar:** comienza el juego seleccionando la dificultad en la que empezara a jugar el nivel inicial (Interfaz No. 02) Selección de dificultad pág. 25



Interfaz No. 2. Selección de dificultad

**Partidas:** este es un tipo de nivel que el usuario puede continuar jugando.  
(Interfaz No. 03) Partidas pág. 26



**Interfaz No. 3. Partidas**

**Opciones:** Muestra las opciones con las cuales trabaja el juego tales como:

**Instrucciones:** hace una explicación breve del objetivo del juego y los controles que puede utilizar para maniobrar el personaje.

**Selección de dificultad:** cuando el usuario haya seleccionado un nivel de dificultad iniciara con el mapa del juego ya que cada uno es diferente.

**Música:** Cambia la música del canal de audio del videojuego de acuerdo al mapa en donde esté jugando.

**Aceptar:** guarda los cambios realizados por el usuario y vuelve al menú inicio.

Cancelar: regresa al menú inicio sin guardar cambios.

Interfaz No. 04 Instrucciones del juego



Interfaz No. 4. Instrucciones del juego

**Créditos:** muestra los derechos de autor del videojuego, los autores directos, los programadores, los diseñadores gráficos, los ingenieros de sonido, el año en que fue publicado y en que lenguaje de programación fue desarrollado.

### 3.3 Diseño del sistema propuesto

El diseño del sistema describe la definición y estructura de los requerimientos previamente establecidos y analizados, el cual se utilizará UML (Lenguaje Unificado de Moldeamiento) teniendo en cuenta que es un lenguaje de modelado de software que propone un estándar para describir un sistema, incorpora aspectos conceptuales como desarrollo de negocio y funciones del juego. Siguiendo esta estructura en el video juego se implementan los tipos de diagramas estáticos y dinámicos que se darán a conocer a continuación.

## ❖ Diagramas estáticos

Se encargan de dar a conocer que componentes (entidades, áreas, objetos, clases, etc.) se va implementar dentro de los módulos del sistema a desarrollar. También se encargan de estructurar los elementos del sistema. Para ello se implementará el siguiente diagrama.

### • Diagrama de clases

En el siguiente diagrama se describirá la distribución de las clases del video juego, el cual describirá la estructura del juego exponiendo sus atributos, relaciones y funcionamiento del juego.

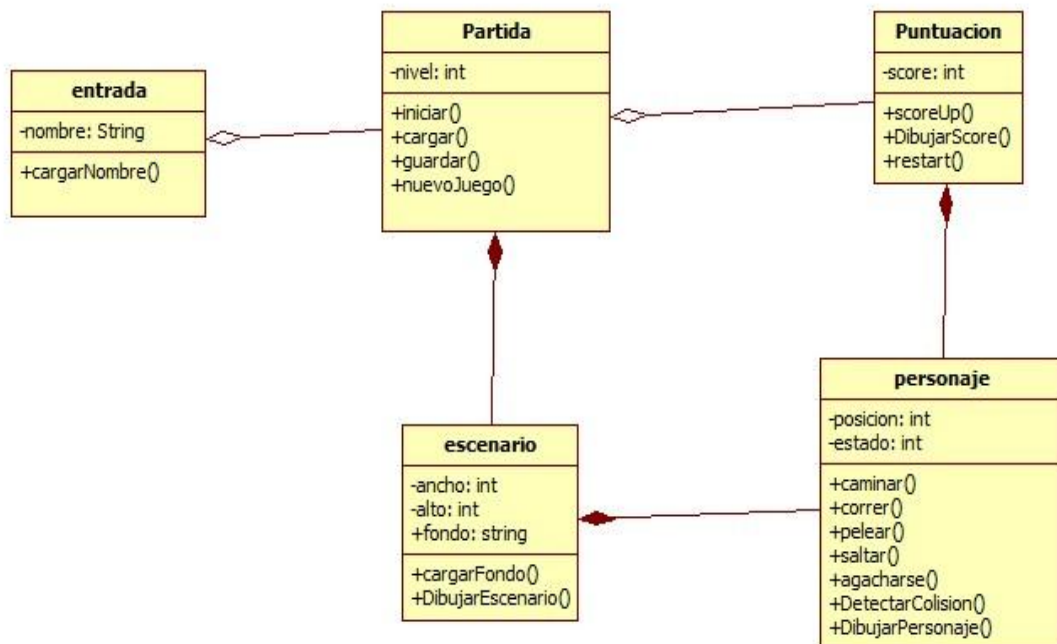


Diagrama No. 1. Diagrama de clases

- **Diagrama de casos de uso.**

Son diagramas que presentan el comportamiento e interacción entre los actores del sistema. A continuación se dan a conocer.

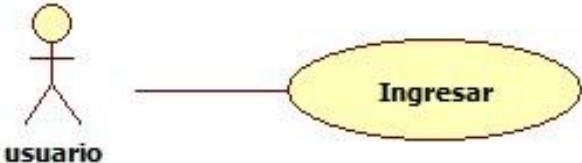
 <p>Caso de uso No ° 1: Ingresar al juego.</p>	<p><b>Nombre:</b> Ingresar al juego</p> <p><b>Descripción:</b> El usuario tendrá acceso al juego introduciendo un nombre</p> <p><b>Flujo De eventos:</b> Flujo Básico Inicio Inserta Nombre. Validación Ingresa al menú inicial.</p> <p><b>Flujo Alternativo:</b> No Inserta Nombre Error en la validación.</p> <p><b>Precondiciones:</b> El usuario debe tener un nombre</p> <p><b>Post condiciones:</b> validación correcta</p>
---	---

Tabla No. 1. Caso de uso Ingresar al juego

<p>The diagram shows a stick figure actor labeled 'usuario' connected by a line to a yellow oval use case labeled 'Nivel'.</p> <p>Caso de uso No ° 2: Escoger Nivel.</p>	<p><b>Nombre:</b> Escoger Nivel</p> <p><b>Descripción:</b> El usuario tendrá que escoger un nivel disponible en el menú de niveles</p> <p><b>Flujo De eventos:</b> Flujo Básico Inicio escoger nivel validación ingresa al nivel.</p> <p><b>Flujo Alternativo:</b> Nivel no habilitado error en la validación.</p> <p><b>Precondiciones:</b> El usuario tuvo que pasar el nivel anterior.</p> <p><b>Post condiciones:</b> Nuevo Nivel habilitado</p>
--	--

Tabla No. 2. Caso de uso Escoger Nivel

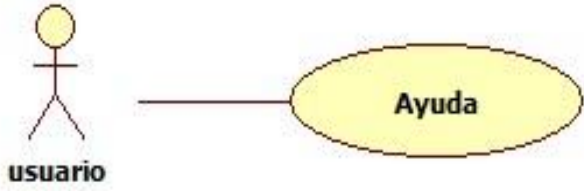
 <p style="text-align: center;">Caso de uso No ° 3: Ayuda</p>	<p><b>Nombre:</b> Ayuda.</p> <p><b>Descripción:</b> El usuario podrá consultar la opción de ayuda que está ubicada dentro del juego.</p> <p><b>Flujo De eventos:</b> Flujo básico Inicio escoger opción de ayuda cuadro describiendo la ayuda solicitada.</p> <p><b>Flujo Alternativo:</b> No escoger esa opción.</p> <p><b>Precondiciones:</b> El usuario debe estar dentro del juego.</p> <p><b>Post condiciones:</b> El usuario puede observar la ayuda.</p>
---	---

Tabla No. 3. Caso de uso Ayuda



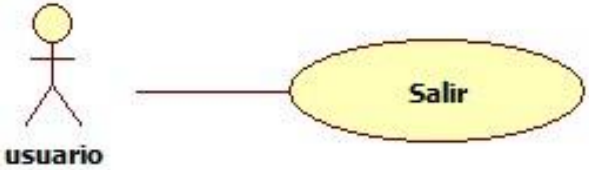
 <p>Caso de uso No ° 4: Salir del Juego.</p>	<p><b>Nombre:</b> Salir del Juego.</p> <p><b>Descripción:</b> El usuario finalizará el juego cuando lo requiera</p> <p><b>Flujo De eventos:</b> Flujo Básico Inicio ingresa el juego Finalizar Juego.</p> <p><b>Flujo Alternativo:</b> El sistema comprueba la validez del cierre del juego, si esto no se cumple mostrará un mensaje.</p> <p><b>Precondiciones:</b> El usuario debe dentro del juego.</p> <p><b>Post condiciones:</b> Se ha cerrado el juego correctamente.</p>
---	--

Tabla No. 4. Caso de uso Salir del Juego

- Diagrama de caso de uso del usuario.



En el siguiente diagrama se da a conocer los pasos que sigue el usuario para llevar a cabo determinado proceso en el juego.

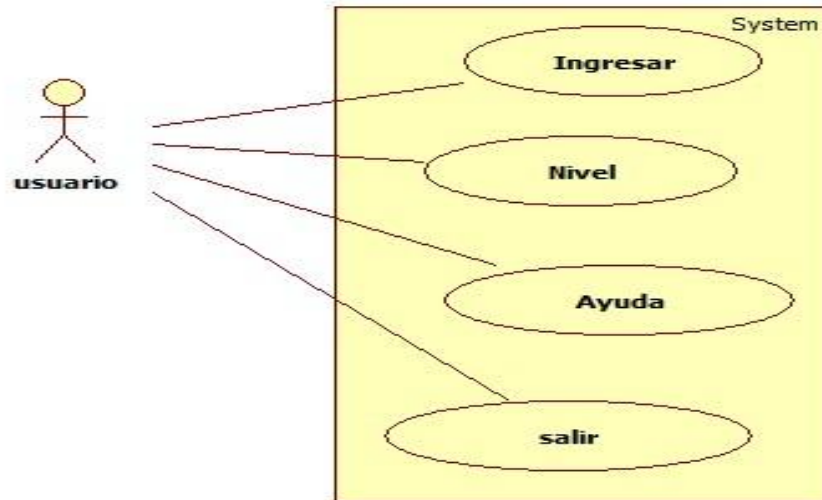


Diagrama No. 2. Diagrama de caso de uso  
Del Usuraio

- **Diagrama de caso de uso general del sistema.**

En el siguiente diagrama se demuestra entre actores y los casos de uso general del juego.

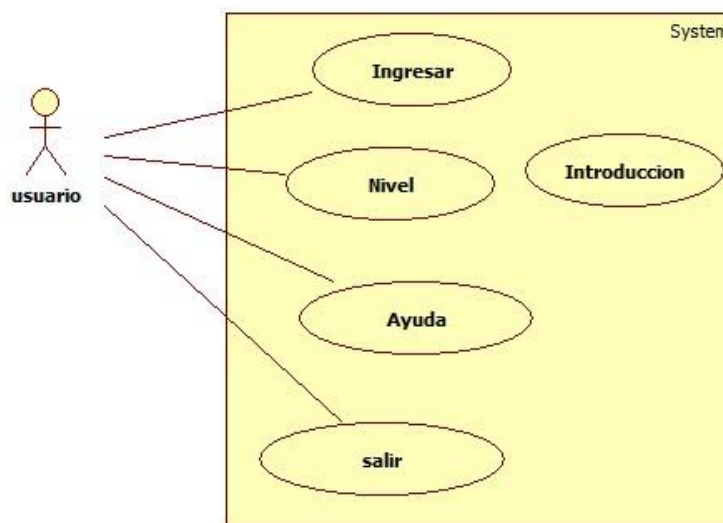


Diagrama No. 3. Diagrama de caso de uso general

## ❖ **Diagramas Dinámicos**

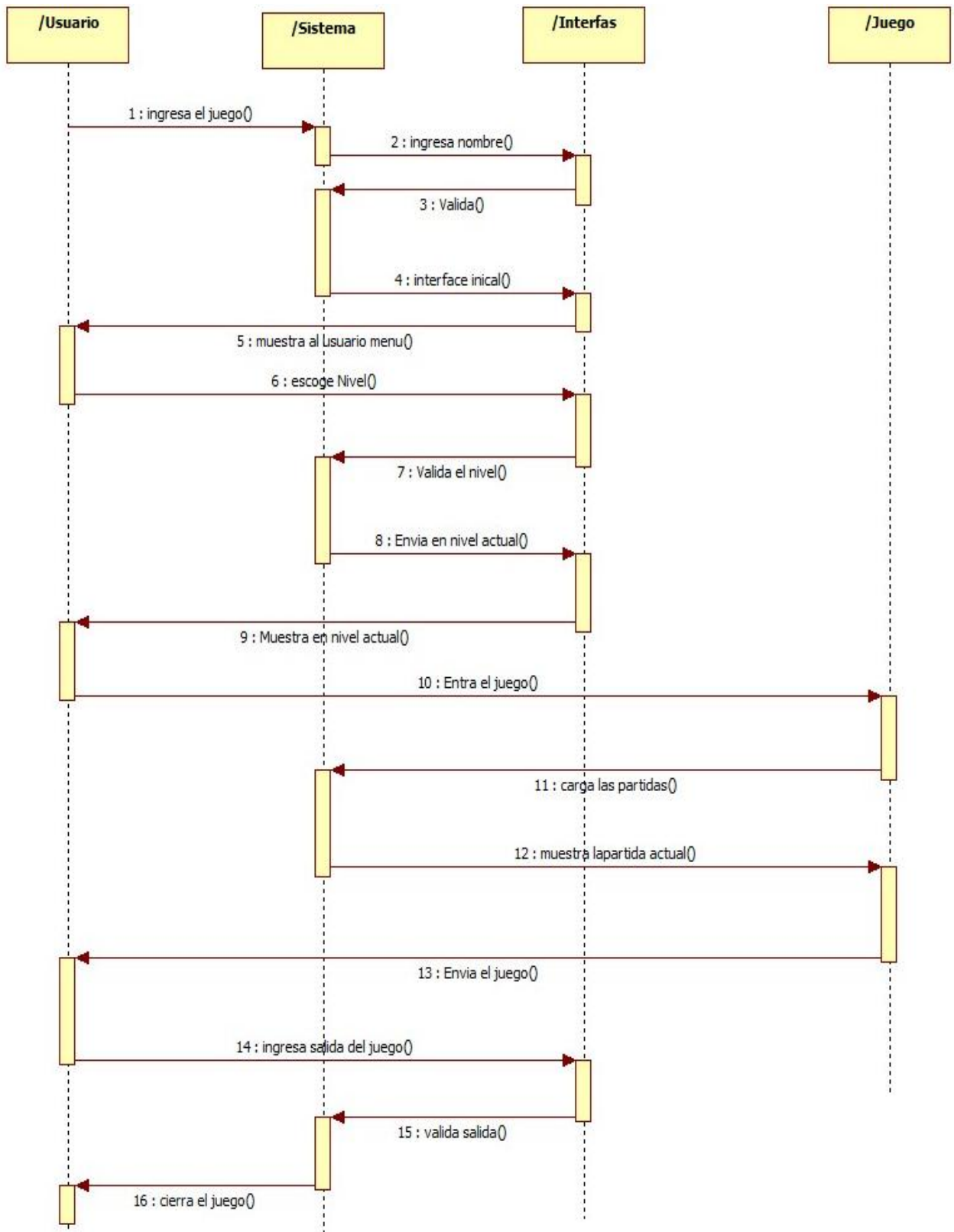
Se utilizar para expresar y modelar el comportamiento del juego a lo largo del tiempo, donde se describe las relaciones temporales entre objetos. Dan a conocer las interacciones entre objetos ocurridas en un escenario del sistema, en el juegos se determino el siguiente.

- **Diagrama de secuencia**

El diagrama de secuencia contiene especificaciones de implementación del escenario, incluyendo mensaje intercambiados entre objetos.

Por otra parte, se da conocer los módulos o clases que pertenecen al sistema y la llamada que se hacen en cada uno de ellos para realizar una tarea determinada.

En el siguiente gráfico Diagrama No. 4. (Diagrama de secuencia) pág. 35 se muestra las diferentes relaciones de todos los usuarios que interactúan con el sistema, desde el inicio hasta las peticiones que realiza dependiendo de la instrucción que se le haya ingresado.



**Diagrama No. 4. Diagrama de secuencia**

## **4. DESARROLLO**

En el Desarrollo del video juego se tiene en cuenta dos aspectos fundamentales del proyecto, las especificaciones técnicas de software y de hardware dado que con estas se puede medir la capacidad de procesamiento y el impacto que le juego tendrá en los diferentes campos donde se utilizará.

### **4.1 Especificaciones técnicas.**

Esta es la sección más importante del documento debe contener una lista detallada y completa de los requisitos que debe cumplir el sistema para un buen funcionamiento del video juego. El nivel de detalle de los requisitos debe ser el suficiente para que el equipo satisfaga los requisitos técnicos, en cuanto a software y hardware, el cual debe tener el usuario para la ejecución del juego.

El videojuego maneja una serie de factores para su buen funcionamiento en este caso enfocándonos a requerimientos en cuanto a software básicamente permite al resto de los programas funcionar adecuadamente, facilitando también la interacción entre los componentes físicos y el resto de las aplicaciones, y proporcionando una interfaz con el usuario. El videojuego va necesitar:

#### **4.1.1 Software**

Como software principal se requiere un sistema operativo el cual gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los restantes. Se pueden utilizar diferentes tipos de sistemas operativos y versiones teniendo en cuenta la

comodidad del usuario; estos son algunos de los sistemas operativos recomendados para la ejecución satisfactoria del video juego.

- Windows XP
- Windows Vista
- Windows 7 o 8
- Linux

Un navegador web partiendo principalmente que es una aplicación que opera a través de Internet la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos comúnmente denominados páginas web, poseen hipervínculos que enlazan una porción de texto o una imagen a otro documento, normalmente relacionado con el texto o la imagen. En este caso funcionaría eficazmente el video juego con los siguientes navegadores:

- Google chrome
- Opera
- Safari
- FireFox

#### **4.1.2 Hardware.**

En cuanto a hardware se refiere a todas las partes tangibles de un sistema informático; sus componentes son eléctricas, electrónicos, electromecánicos y mecánicos. Son periféricos de todo tipo y cualquier otro elemento físico involucrado; hardware básico, el estrictamente necesario para el funcionamiento normal del video juego será:

- Acceso a internet desde 1 Mega en adelante
- Espacio en disco duro de 50 Mb o superior
- Memoria RAM 256 Mb o superior
- Procesador Intel Core2 Dúo E6700/AMD Athlon64 X2 6000

## 5. GLOSARIO

**HTML5:** Hypertext Markup Language (lenguaje de marcado de hipertexto), es un lenguaje que describe la estructura y el contenido de un pagina web.

**CSS:** Cascading Style Sheets (hoja de estilo en cascada), es un lenguaje que describe la presentación de los documentos estructurados en HTML o XHTML, XML.

**Javascript:** Es un lenguaje interpretado orientado a las páginas web, este lenguaje tiene un sintaxis similar a la de java.

**UML:** Unified Modeling Language (lenguaje unificado de modela miento), es un lenguaje gráfico para construir, documentar, especificar y visualizar un sistema.

**Navegador:** Es una aplicación que permite al usuario visualizar documentos de hipertexto escritos en HTML.

**PHP:** Hypertext Preprocesor, es un lenguaje de programación interpretado, y de código abierto, diseñado para la creación de páginas web dinámicas.

**Videojuego:** Es toda aplicación o Software que ha sido creada con el fin del entretenimiento, enfocada en la interacción de Uno o más jugadores.

**2D:** Algo bidimensional tiene dos dimensiones es decir ancho, alto, pero no profundidad.

**Web:** Es un sistemas de distribución de información basada en hipertexto o hipermedios enlazados y accesibles a través de internet.

## 6. CONCLUSIONES

- Ninja Warring permite una fácil y rápida jugabilidad, por medio de una interfaz gráfica intuitiva y fácil de reconocer, además de establecer unas ayudas para poder navegar dentro del juego y poderlo ejecutar en toda su fluidez.
- Se aprovecho el potencial de las diferentes tecnologías y se desarrollo un software orientado a videojuegos.
- Por medio de un navegador cualquier persona podrá ejecutar el juego sin necesidad de utilizar demasiados recursos de hardware de un computador.
- Ninja Warring permite guardar el puntaje obtenido durante el juego en una base de datos y así poder comparar un puntaje actual con puntajes anteriormente realizados.



## 7. BIBLIOGRAFIA

- -Que es UML (2012), Definición de UML. Disponible en URL:  
<http://www.docirs.cl/uml.html>
- -Scrum (2013), Características de Scrum. Disponible en URL:  
<http://es.wikipedia.org/wiki/Scrum>
- -Ingeniería de Software (2012), Introducción. Disponible en URL: <http://www-monografias.com/trabajos5/inso/inso.html>
- -Scirra (2013), Top Tutorials. Disponible en URL:  
<http://www.scirra.com/tutorials/top>
- -UML (2012), Ejemplo sencillo sobre modelado de un proyecto. Disponible en URL: <http://msdn.microsoft.com/es-es/library/bb972214.aspx>
- -Normas ICONTEC para trabajos escritos(2009).[En línea].Jorge Mario Monsalve Guaracao.Disponible en URL:  
<http://www.slideshare.net/JorgeM93/normas-icontec-para-trabajos-escritos>
- -*Random Animation*( 24th, January 2012).[En Línea].Weishaupt.Disponible en:  
<https://www.scirra.com/tutorials/252/random-animation>
- -*Physics Explosion*(7th, June 2013).[En Línea].TottenMan.Disponible en:  
<https://www.scirra.com/tutorials/579/physics-explosion>



## **8.1 MANUAL DEL SISTEMA**

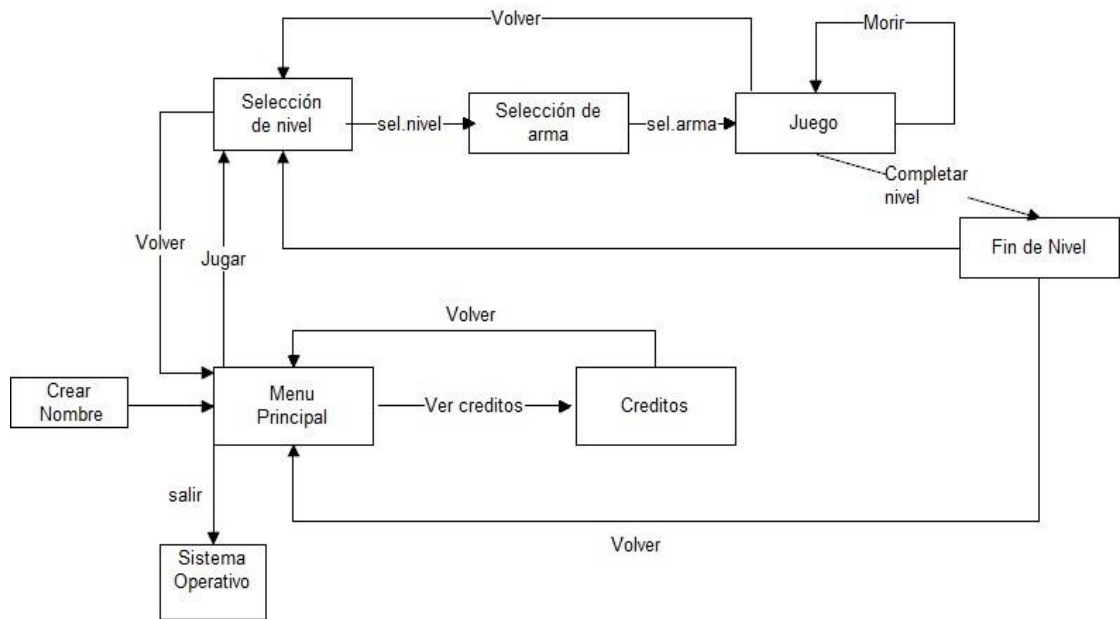
En el manual del sistema en el video juego se tiene en cuenta dos aspectos fundamentales del proyecto, la parte que constituye la interfaz que muestra el tipo de menús que va a observar el usuario, y el arte que abarca todas las especificaciones técnicas de software dado que con estas se puede diferenciar las características en cuanto a audio, y el tipo de imágenes utilizadas para el buen funcionamiento del sistema.

### **8.1.1 INTERFAZ**

En esta sección se especificará con detalle cada una de las pantallas que componen NinjaWarring. Como también, se indicarán las transiciones entre ellas así como la utilidad de cada elemento de la GUI (Graphical User Interface). Las imágenes adjuntadas son bocetos que ilustran los componentes que deben contener cada pantalla. No obstante, se podrán realizar cambios en la apariencia y disposición de los elementos si así se considera oportuno.

#### **Diagrama de Flujo:**

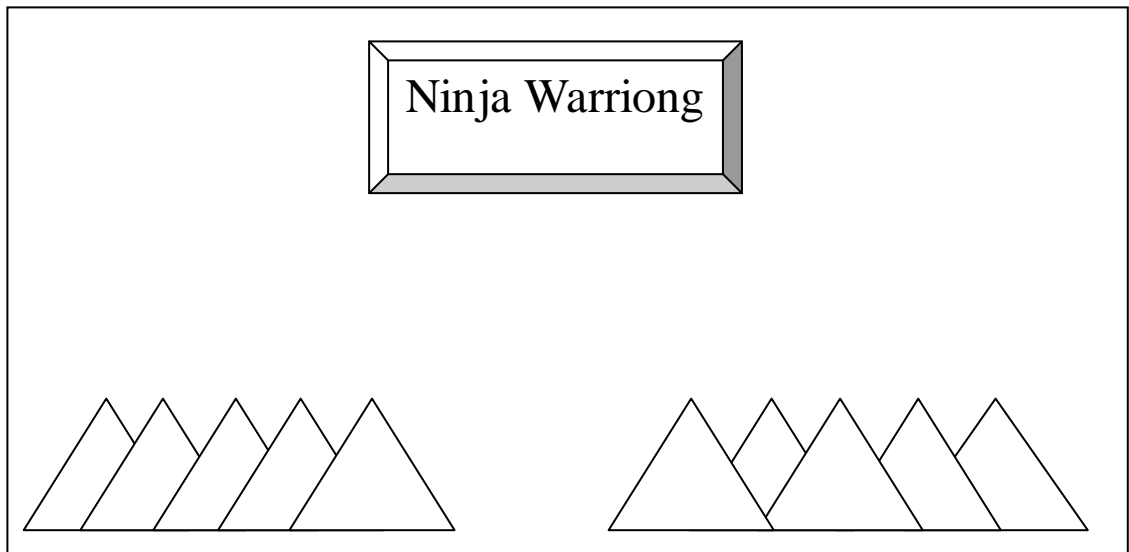
El siguiente diagrama de estados muestra las pantallas presentes a lo largo de NinjaWarring y las transiciones entre ellas. En secciones posteriores nos centraremos en ellas de forma individual. Se podrá encontrar en el Diagrama de flujo de (pantallas en el juego) pág. 44.



**Figura No 1: Diagrama de flujo de pantallas en el juego**

**Ventana principal:**

A continuación el boceto de la pantalla Principal:



**Figura No 2: Boceto de la ventana Principal**

Lista y descripción de todos sus componentes.

- **Botón de entrar:** al pulsarlo lleva a la pantalla del menú principal.
- **textArea:** se introduce un nombre para entrar al menú principal.

### Menú Principal:

A continuación el boceto del menú principal.

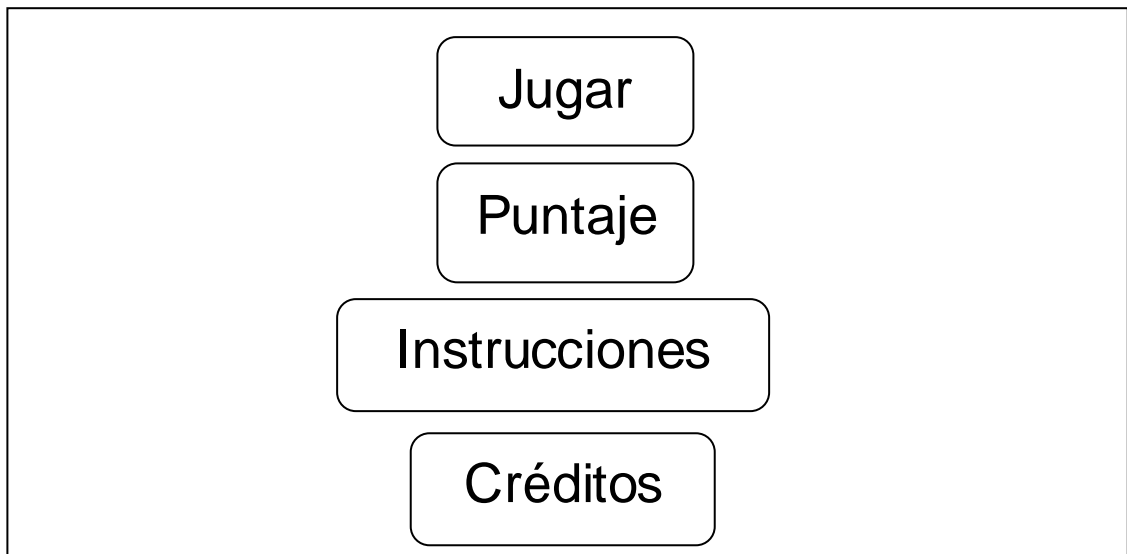


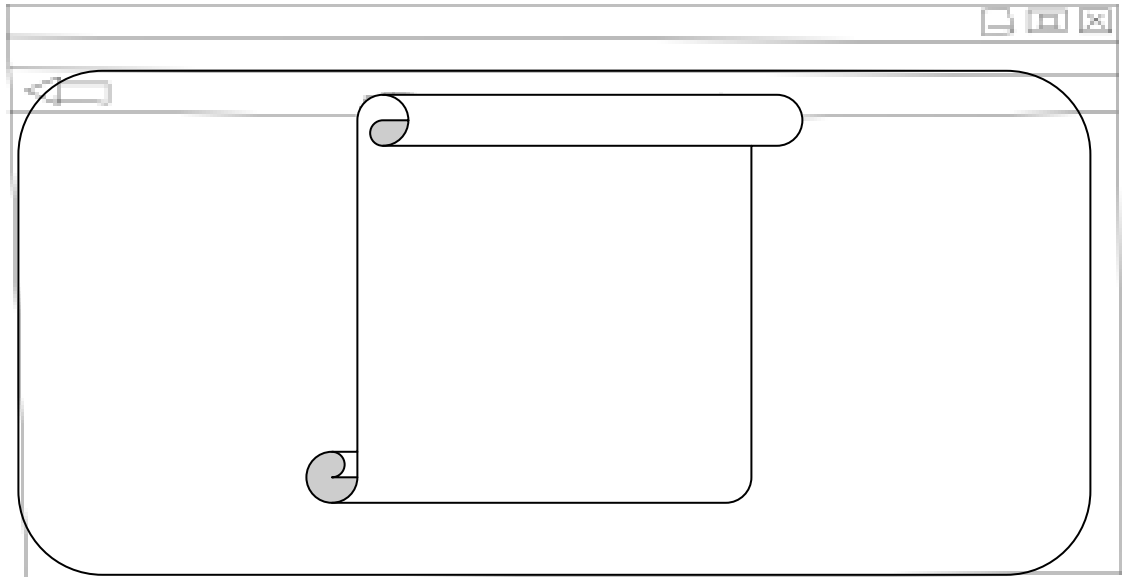
Figura No 3: Boceto de la pantalla del menú principal.

Lista y descripción de todos sus componentes.

- **Botón de Jugar:** al pulsar lleva a la pantalla del juego.
- **Botón de Puntaje:** al pulsar lleva a la pantalla de las partidas desbloqueadas.
- **Botón de Instrucciones:** al pulsar lleva a la pantalla de opciones.
- **Botón de créditos:** al pulsar lleva a la pantalla de créditos.

## **Créditos:**

A continuación el boceto de la pantalla de créditos:



**Figura No 4: Boceto de la pantalla de créditos**

Lista y descripción de todos sus componentes.

- Botón de regreso (fecha): al pulsar lleva de regreso a la pantalla del menú principal.

## **Niveles:**

A continuación el boceto de la pantalla de selección de partidas.

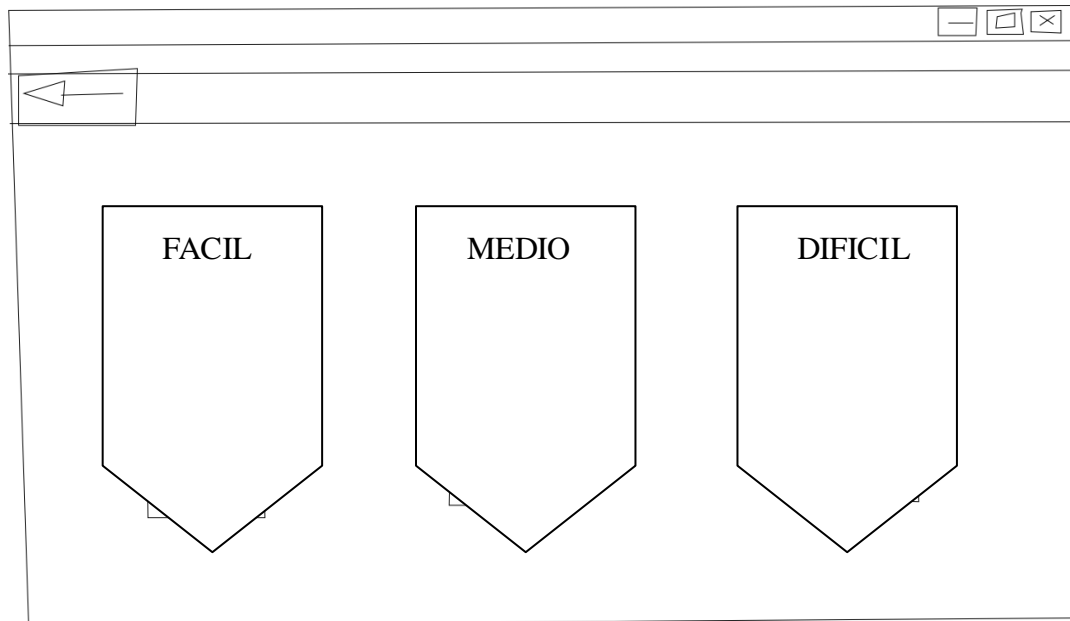


Figura No 5: Boceto de la pantalla de selección de dificultad

Lista y descripción de todos sus componentes.

- **Botón de regreso (fecha):** al pulsar lleva de regreso a la pantalla del menú principal.
- **Botón de Partida:** cada partida está representada por un botón que contiene: el nombre de la partida, Para seleccionar la partida haremos click sobre su botón correspondiente y pasaremos a la partida seleccionada.

### Opciones:

A continuación el boceto de la pantalla de opciones.

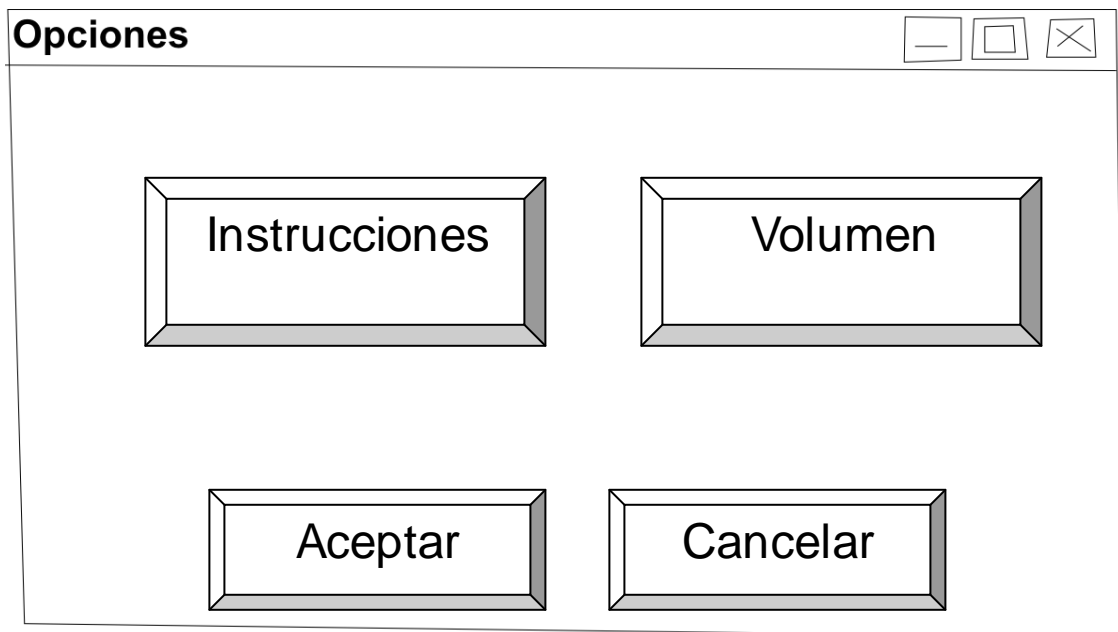


Figura No 6: Boceto de la pantalla de Opciones

Lista y descripción de todos su componentes.

- **Botón de Instrucciones:** Al pulsarlo este nos llevará a la pantalla de instrucciones del juego.
- **Botón de personajes:** Al pulsar este nos lleva a la pantalla donde encontraremos los personajes que podemos utilizar en el juego.
- **Botón de Volumen:** Al pulsar este nos permitirá aumentar o disminuir el volumen del juego.
- **Botón de Aceptar:** Al pulsar esta opción realizará los cambios hechos y nos devolverá a la pantalla del menú principal.
- **Botón de Cancelar:** Al pulsar este nos devolverá a la pantalla del menú principal.



## Instrucciones:

A continuación el boceto de la pantalla de Instrucciones.

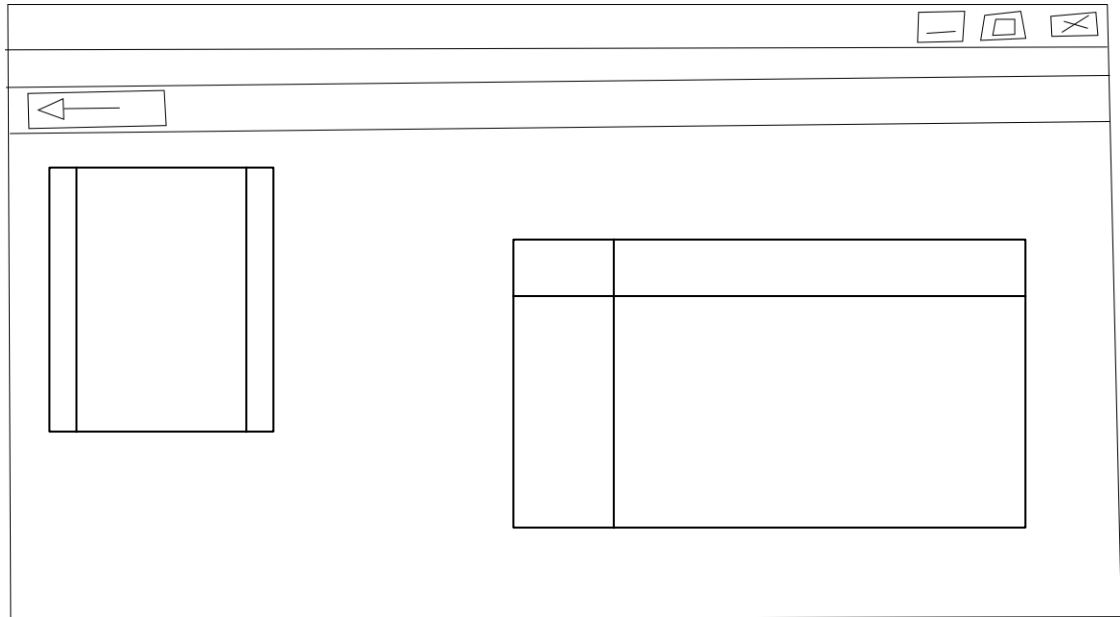


Figura No 7: Boceto de la pantalla de Instrucciones.

### 8.1.2 ARTE

NinjaWarring debe tener un carácter tenso y épico. Para el joven ninja es todo un reto salvar a sus aliados aún es inexperto y no ha perdido su inocencia. Los colores deben ser algo vivo, los modelos muy básicos y la música acorde con el desenfado conjunto.

A continuación se enumera los recursos necesarios:

#### Arte 2D:

Todos las imágenes deben estar en formato .png además de en el formato propio del programa con el que se crearon (.svg) para posibles futuras modificaciones. El fichero de trabajo siempre debe tener una calidad superior a la requerida en el juego.

### **Interfaz:**

- Logo: logo con el texto 'Ninja Warriong' siguiendo con los requisitos anteriormente expuestos.
- Icono 'cuchillo': Imagen cuadrada con un cuchillo.

### **Audio:**

Siempre es necesario guardar y entregar el proyecto de ficheros de audio en el formato que se use el software con el que se produce, mientras los efectos de sonido estarán en .wau.

#### Música:

- Menú principal: música de aventura y tensión aunque más relajada que la correspondiente a los niveles.
- Juego: música animada e intensa que debe provocar en el juego sensación de tensión
- Victoria: música breve que sonará cuando completemos un nivel. Debe ser alegre y hacer que el jugador se sienta recompensado.
- Game Over: Pieza breve de derrota con un tinte cómico, para resaltar gravedad a la partida (que le jugador piense que haber perdido una vez no es tan terrible).

#### Efectos:

- Seleccionar opción: Al hacer click con el ratón sobre alguna opción.

- Opción No Permitida: pequeña advertencia sonora indicando que una acción no puede ser llevada a cabo.
- Espadazo: sonido de una espada cortando el aire.
- Golpe: golpe producido por un ninja.
- Ataque perro: ruido del ataque del perro.
- Daño perro: sonido al herir al perro.
- Daño personaje: sonido cuando el personaje es dañado.
- Daño empresario: sonido al herir al enemigo.
- Daño pandillero: sonido al herir al pandillero.
- Mordida: mordida del perro.

## **8.2 MANUAL DEL USUARIO**

En el manual del usuario en el video juego se tiene en cuenta dos aspectos fundamentales del proyecto, El concepto del juego que muestra la temática y el manejo en cuanto al videojuego, y las mecánicas de juego que incluye los factores que influirán en el desarrollo de la historia del juego, y además el manejo del personaje.

Además el video juego es de funcionamiento en red que ejemplifica todos los contenidos de construct 2, es un gamemaker sobre desarrollo de video juego en 2D. Este escrito tiene como objetivo principal plasmar los elementos que debe incluir Ninja Warring y servir de carta de presentación en caso de buscar de colaboradores en el futuro.

### **8.2.1 CONCEPTO DEL JUEGO.**

Ninja Warring es un video juego en el que controlamos a un ninja que debe cruzar por varios mapas para poder lograr salvar a su clan del exterminio total mientras sus compañeros son tomados a la fuerza por diferentes clanes. Durante su travesía, el se encontrará con subordinados de un clan (cada mapa maneja un clan diferente) y nuestro ninja debe detener y eliminar el clan a toda costa utilizando sus limitadas habilidades.

#### **Características Principales:**

El Juego se basa en los siguientes pilares:

- Planteamiento sencillo: la historia mencionada es muy simple para el desarrollo del videojuego pero lo suficientemente explícita para que el jugador sienta que tiene un objetivo.
- Táctica: detener a los enemigos debe ser imposible si se comienza a atacar indiscriminadamente. La gestión de nuestras limitadas capacidades de forma inteligente será imprescindible.
- Dinamismo: a diferencia de algunos juegos de estrategia, Ninja Warring debe ser dinámico y provocar una sensación de tensión en el jugador.
- Ampliación: Ninja Warring debe ser ampliable con nuevos niveles y enemigos de forma sencilla.

### **Género:**

Ninja Warring supone una unión de varios géneros. A continuación se listan los géneros de los que toma elementos y sus motivos:

- **Tower Defense:** un subgénero de la estrategia basada en detener oleadas de enemigos colocando de forma estratégica obstáculos y trampas. En Ninja Warring manejamos los mismos elementos aunque aparecen la figura de un protagonista.
- **Acción-Aventura:** juegos dinámicos y directos en el que el jugador experimenta una descarga de adrenalina, desafía tanto los reflejos como la capacidad de resolver problemas, ya sea en situaciones violentas o no. En

Ninja Warring tenemos los componentes de la acción aunque la cámara se situara de forma que podamos ver gran parte de la escena.

### **Propósito y público objetivo:**

El principal objetivo de Ninja Warring es ofrecer a los lectores de construct 2 un ejemplo real de videojuego en 2D, donde debe ser un producto llamativo, fácil de maniobrar y divertido. Su interés no solo debe radicar en el código fuente y su proceso de desarrollo sino en el propio juego y sus mecánicas.

Ninja Warring está dirigido a jugadores de un amplio rango de edades con un tiempo limitado que dedicar al ocio electrónico. Por ello, se apuesta por un sistema de partidas cortas y recompensas rápidas. La historia es sencilla, lo que permite poder jugar de forma esporádica.

### **Jugabilidad:**

Cada nivel de Ninja Warring ofrece un mapa que contiene enemigos. Tenemos que impedir que los enemigos lleguen a eliminar al compañero secuestra del ninja. Para ello nos valdremos de los siguientes elementos:

- **Movilidad:** al contrario que en otros Tower Defense, en Ninja Warring controlamos un personaje. Nos desplazaremos por el escenario atendiendo los focos de peligro que consideremos oportunos.
- **Trampas y obstáculos:** podemos dirigir al personaje a un punto del mapa y colocar una trampa u obstáculo si tenemos suficientes recursos. Esto eliminará o ralentizará a los enemigos.
- **Armas:** el combate directo utilizando armas también es una opción.

- Puntos fuertes y débiles: cada enemigo tendrá puntos fuertes y débiles. La mejor forma de hacerles frente es utilizar la herramienta adecuada en cada momento.
- Mejoras: el jugador deberá recibir recompensas cada poco tiempo para que sienta que progresa en el juego. Tendremos armas des bloqueables.

### **Estilo visual:**

Ninja Warring tendrá un estilo sencillo, no demasiado detallista para encajar con su carácter amigable y accesible. El estilo visual que mas encaja con este concepto es el de dibujo animado. Los colores serán sombríos y las texturas muy simples. Se estudiaría ampliar un efecto cel-shading para reforzar esta idea.

### **Alcance:**

El objetivo principal es desarrollar un sistema de juego sólido al que podamos introducirle contenidos sin dificultad. En primera instancia se desarrollará un pack de contenidos básicos que será ampliado en un futuro.

## **8.2.2 MECÁNICAS DE JUEGO**

En esta sección entraremos más en detalle en lo que a las mecánicas de Ninja Warring se refiere. Se comentaran todos los pilares que fundamentan su Jugabilidad y se detallaran las acciones que pondrá llevar a cabo el jugador dentro de una partida típica. Además, se ofrecerá una lista con los personajes del juego (tanto protagonistas como enemigos), habilidades, objetos etc. Por último, se modelará el mundo en el plano de movimientos, físicas y detección de colisiones.

## **Jugabilidad:**

**Niveles** cada nivel de Ninja Warring es un mapa que nuestro joven ninja debe pelear. Escenarios interiores con varias entradas y un objetivo valioso (compañero del clan del ninja) al final del mapa. Los enemigos custodian al compañero del ninja, por otra parte otros enemigos se dirigen hacia al ninja impidiéndole el paso al ninja, debemos impedir que eliminen al objetivo valioso. Los mapas tendrán obstáculos, entre ellos mobiliario puestos por los enemigos, que dificultará o el tránsito por algunas zonas. Habremos vencido cuando todos los enemigos (previamente establecidos) hayan cesado.

**Varios Focos** Nuestro personaje se verá sobrepasado por la entrada simultánea de enemigos en el mapa. La posibilidad de moverse con libertad por la zonas libres del escenario le permitirá acercarse más a su objetivo según el criterio del jugador. El movimiento de los personajes será sobre un plano (El suelo) aunque más adelante se darán más detalles sobre el movimiento.

**Intensidad** la dificultad de Ninja Warring viene dada por la cantidad de enemigos y trampas puestos en el mapa, su fortaleza y la distancia entre los distintos puntos de encuentro. Entre mapas la intensidad y dificultad de las partidas irá aumentando.

**Trampas** las trampas conforman una de las herramientas del jugador para hacer frente a los enemigos. Consumen parte de los recursos del protagonista y pueden ser útiles para impedir el avance de los enemigos o para causar daño si se este se topa con una. Son elementos estáticos que coloca manualmente el jugador.

**Habilidades** Las habilidades inciden en el componente de acción de Ninja Warring. El protagonista debe acudir al foco de acción que considere oportuna



para utilizar habilidades y armas directos que dañaran a sus enemigos. Existirán varios tipos de habilidades y armas cuya efectividad dependerá de las características del enemigo.

**Recursos limitados** El jugador no podrá dedicarse a gastar armas y colocar trampas sin llevar una táctica concreta. El protagonista cuenta con un potencial limitado(es un simple iniciado) y debe administrarlo racionalmente. El protagonista solo podrá utilizar una habilidad concreta si su barra de energía supera al coste de dicha habilidad. Como veremos en secciones posteriores, ciertas trampas y habilidades funcionaran mejor que otras según el enemigo y el escenario. El componente táctico es primordial y cobrara mucha importancia en el desarrollo de una partida.

**Progresión del Jugador** El jugador progresará a medida que avanza por los niveles consiguiendo nuevas armas y habilidades. De esta forma el jugador no solo se verá recompensado por completar el mapa sino por el hecho de que su personaje sea más poderoso.

**Planificación de la batalla** El jugador tendrá a su disposición un conjunto más o menos amplio de habilidades, según estas se hayan desbloqueado o no. No obstante, las pobres habilidades del protagonista le impiden utilizarlas todas al mismo tiempo. Antes de cada nivel, El jugador deberá elegir un determinado grupo de armas no más de 5.

Con esta decisión se pretende aumentar el componente táctico de Ninja Warriong. Por supuesto, de cara al siguiente mapa, el jugador podrá volver a elegir el subconjunto de armas que considere oportuno.

## **Flujo De Juego:**

A lo largo de esta sección se detallará el trascurso de una partida típica en Ninja Warring. Se comentaran los pasos que ha de seguir el jugador desde el arranque del juego hasta completar un mapa completo. Poco a poco vamos desmenuzando el funcionamiento exacto del juego, en esta sección describiremos las mecánicas. Mas adelante se definirá el contenido de cada pantalla.

El jugador inicia Ninja Warring y se le presenta un menú principal. Si desea iniciar una partida el jugador seleccionará la opción jugar. En la siguiente pantalla el jugador seleccionará un perfil que desee. Si no cuenta con un perfil creado podrá crear uno nuevo. Entonces, el jugador podrá elegir cualquier mapa que haya desbloqueado en la pantalla de selección de mapa.

Los mapas se organizan de forma secuencial y cada vez que se complete uno, se desbloquee el siguiente. Luego hay que seleccionar el conjunto de armas y trampas de las que dispondrá el protagonista en el transcurso del mapa (Selección de armas). El jugador podrá seleccionar 5 de ellas llenando los slots (casillas) disponibles.

El personaje comienza en el mapa, el protagonista deberá avanzar hacia adelante y se le advierte al jugador que deberá llegar al final del mapa y no dejar que los enemigos eliminen a su compañero de clan. El jugador será derrotado si los enemigos eliminan a su compañero antes que él llegue o si acaban con la vida del protagonista. Poco a poco aparecerán los enemigos por el mapa. El jugador podrá seleccionar en cualquier momento una de las armas o trampas disponibles. Solo podrá utilizarlas si su nivel de energía se lo permite. La energía se irá recargando con el paso del tiempo. Si hemos seleccionado un arma, al pulsar el botón de ataque, esta será proyectada en la dirección actual. En cambio, si tenemos una trampa seleccionada, la colocaremos en la posición deseada.

Cuando abatimos a un enemigo recibiremos puntos que nos ayudará a desbloquear más armas.

Si el personaje muere o los enemigos elimina al compañero se mostrará un mensaje de 'fin del juego' y el mapa se reinicia. No hay un número determinado de vidas, el jugador podrá repetir un escenario cuantas veces necesite.

Cuando hemos acabado con los enemigos y el enemigo superior y rescatado al compañero del ninja aparecerá un cartel de victoria indicándole al jugador que ha completado el mapa. Tras la pantalla del mapa vamos a la de Fin de Mapa. En ella se nos comunica el progreso que hemos realizado y si hemos conseguido desbloquear una nueva arma. En tal cosa, dicha arma aparecerá disponible la próxima vez que acudamos a la pantalla de selección de armas.

En el momento en que el jugador acepte volveremos a la pantalla de selección de mapa con un nuevo mapa disponible.

Cuando lo desee, el jugador podrá regresar a pantalla anterior o al Menú Principal. Más adelante se mostrará todas las posibilidades de flujo entre pantallas.

### **Personajes:**

En esta sección procederemos a enumerar y describir todos los personajes de Ninja Warring a si como sus habilidades y comportamientos.

### **Protagonista:**

**Descripción** Personaje es el protagonista de Ninja Warring. Es un iniciado del clan de los lobos y su misión es rescatar a los miembros de este clan, ya que los demás clanes piensan eliminar los miembros del clan de los lobos y quedarse con todas las reliquias de este clan. Personaje es joven, simpático y tiene un

carácter apasionado, propio de un iniciado. Su apariencia es muy sencilla, no lleva armas dado su status en la jerarquía del clan. Va armado con su valor y habilidades con las manos y cuerpo.

- **Vida Inicial:** 100(se restaura en cada mapa).
- **Armas :** 0
- **Experiencia inicial:** 0
- **Velocidad inicial:** 0

**Habilidades** la siguiente tabla muestra la lista de habilidades de personaje junto con una descripción, su costo en puntos y la experiencia necesaria para desbloquear.

<b>Habilidad</b>	<b>Descripción</b>	<b>Daño</b>	<b>Puntos</b>	<b>Experiencia</b>
Ráfaga	Habilidad a distancia en lanzamiento de cuchillos	30	10	0
Muerte Fugas	Ataque cercano del elemento espada	50	20	500
Patada	Ataque cercano lanzando un patada	20	10	0

**Cuadro No 1: Habilidades del protagonista**

### **El Empresario:**

**Descripción** Enemigo básico sin ninguna afinidad elemental. Clásico personaje vestido de ejecutivo de estatura promedio. Va armado con una pistola 9mm Kurz y una seriedad en su rostro, son ágiles pero no tienen grandes habilidades en combate.

- **Vida:** 60
- **Ataque:** 15
- **Velocidad:** 15
- **Afinidad elemental:** ninguna

**Perro callejero:**

**Descripción** perro grande proveniente de la calle. Ataca con sus afilados dientes y no necesita ningún tipo de arma adicional.

- **Vida:** 60
- **Ataque:** 20
- **Velocidad:** 30
- **Afinidad elemental:** Ninguna

**Pandillero:**

**Descripción** Enemigo del clan de los dragones ligeramente superior al empresario. Cuenta con armas corto pulsantes como cuchillos. Tatuado en una parte de su cuerpo, muestra una actitud a la defensiva y despiadada. Ataca con sus afilados cuchillos y no necesita ningún tipo de arma adicional.

- **Vida:** 70
- **Ataque:** 25
- **Velocidad:** 25
- **Afinidad elemental:** Ninguna

## **Movimiento y física:**

### **Interacción entre elementos**

Ninja Warriong se desarrolla sobre un plano y tanto los enemigos como el personaje pueden desplazarse por él. De todos modos, el escenario presentará ciertos obstáculos como vehículos y objetos que no podrán ser atravesados por ninguna entidad.

Los enemigos atacan por medio de sus armas cuando el protagonista está próximo a ellos. Cuando el jugador ve que el enemigo está cerca al protagonista esta puede accionar una habilidad o lanzar algún arma para poder así eliminar y quitar del camino a su obstáculo (el enemigo). Cuando las armas colisionan con un personaje, esta sufre daño y el proyectil (chicho) se desvanece. Si el proyectil colisiona con

Finalmente, las colisiones que se producirán:

- Personaje – Personaje.
- Personaje – Escenario.
- Proyectil – Personaje.
- Proyectil – Escenario.

### **Controles**

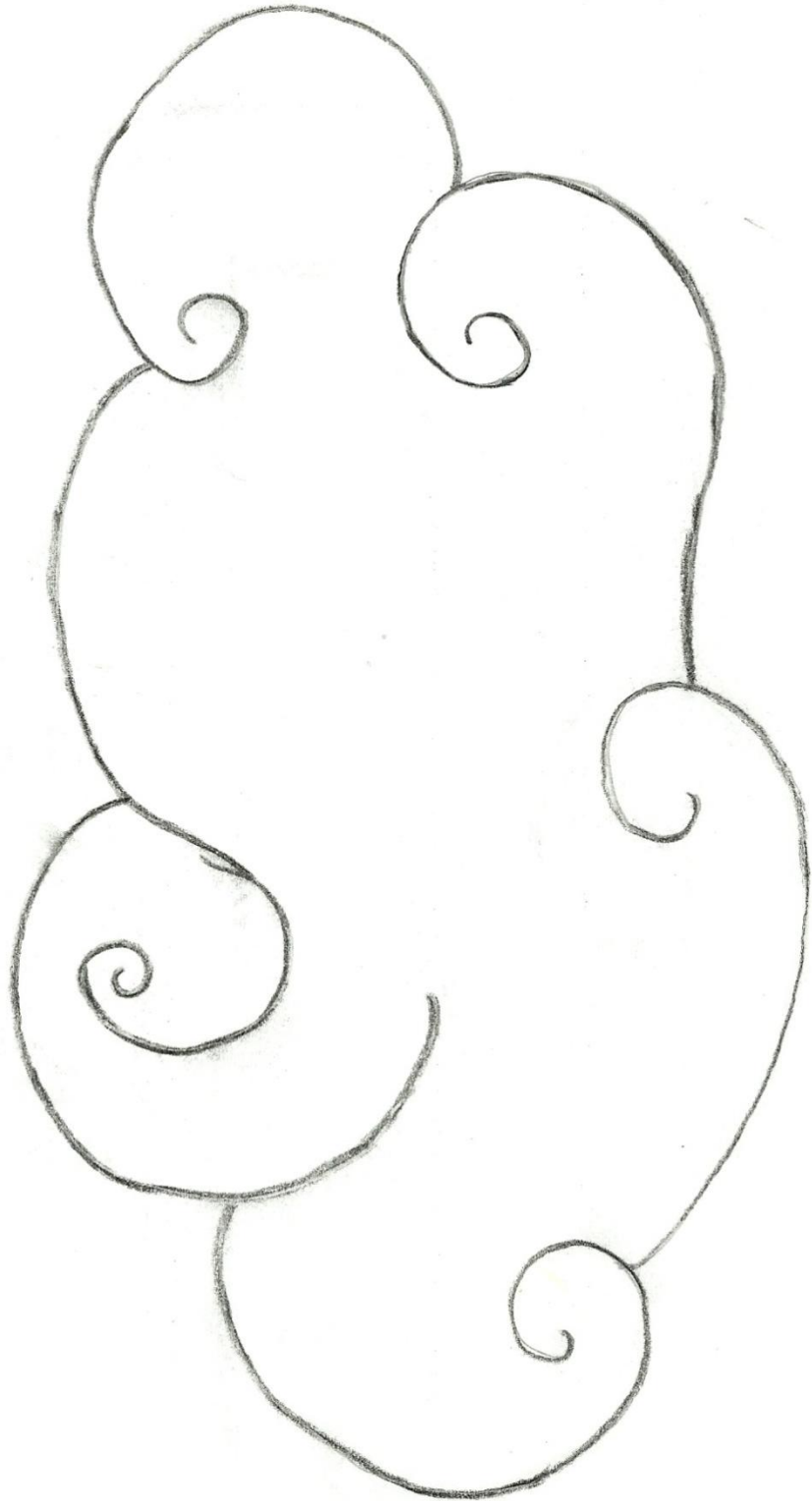
- Movimientos: teclas fecha arriba, fecha abajo y C
- Usar Habilidades: techa espaciadora. Al utilizar la habilidad el personaje lanza su habilidad.
- Usar armas: C. al utilizar el arma el personaje se apuntara así el objetivo.





**Figura No 5: Boceto del personaje Principal**





**Figura No 6: Boceto nubes de la ventana Principal**

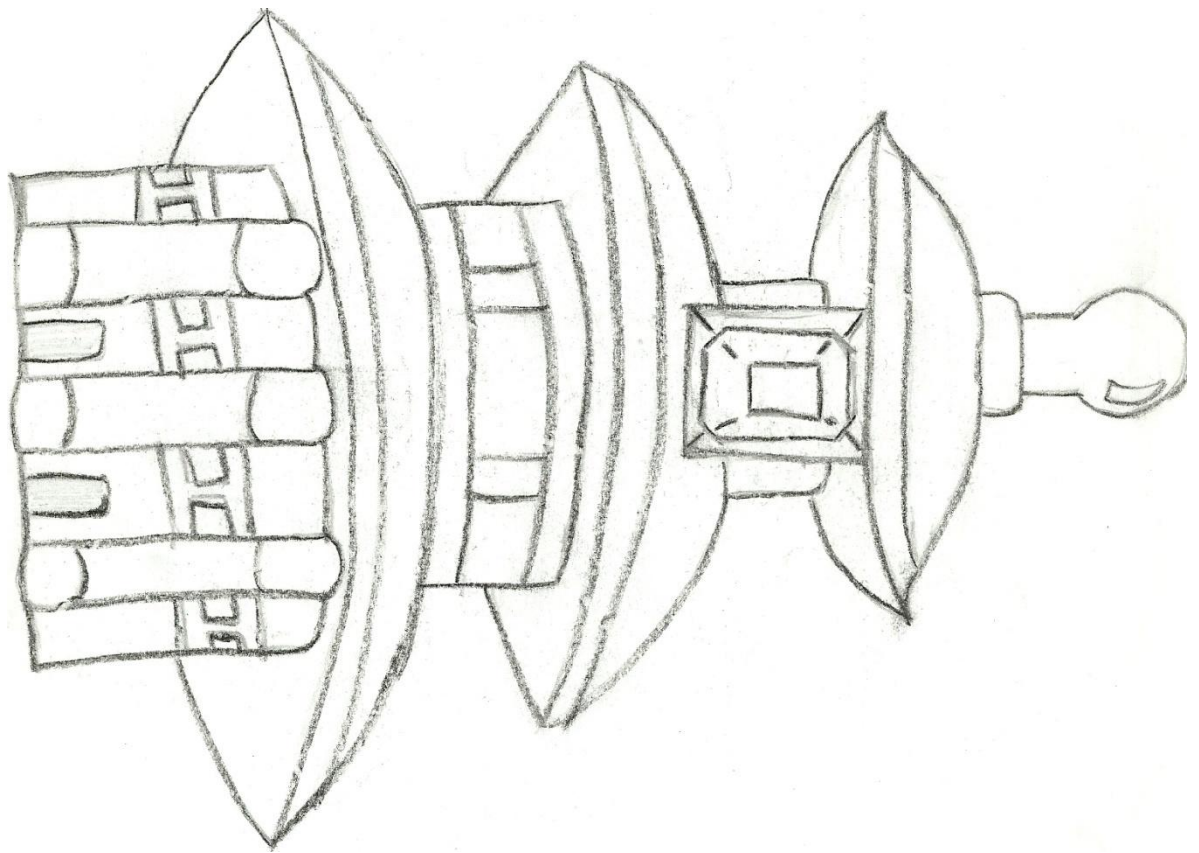


Figura No 7: Boceto de edificios utilizados

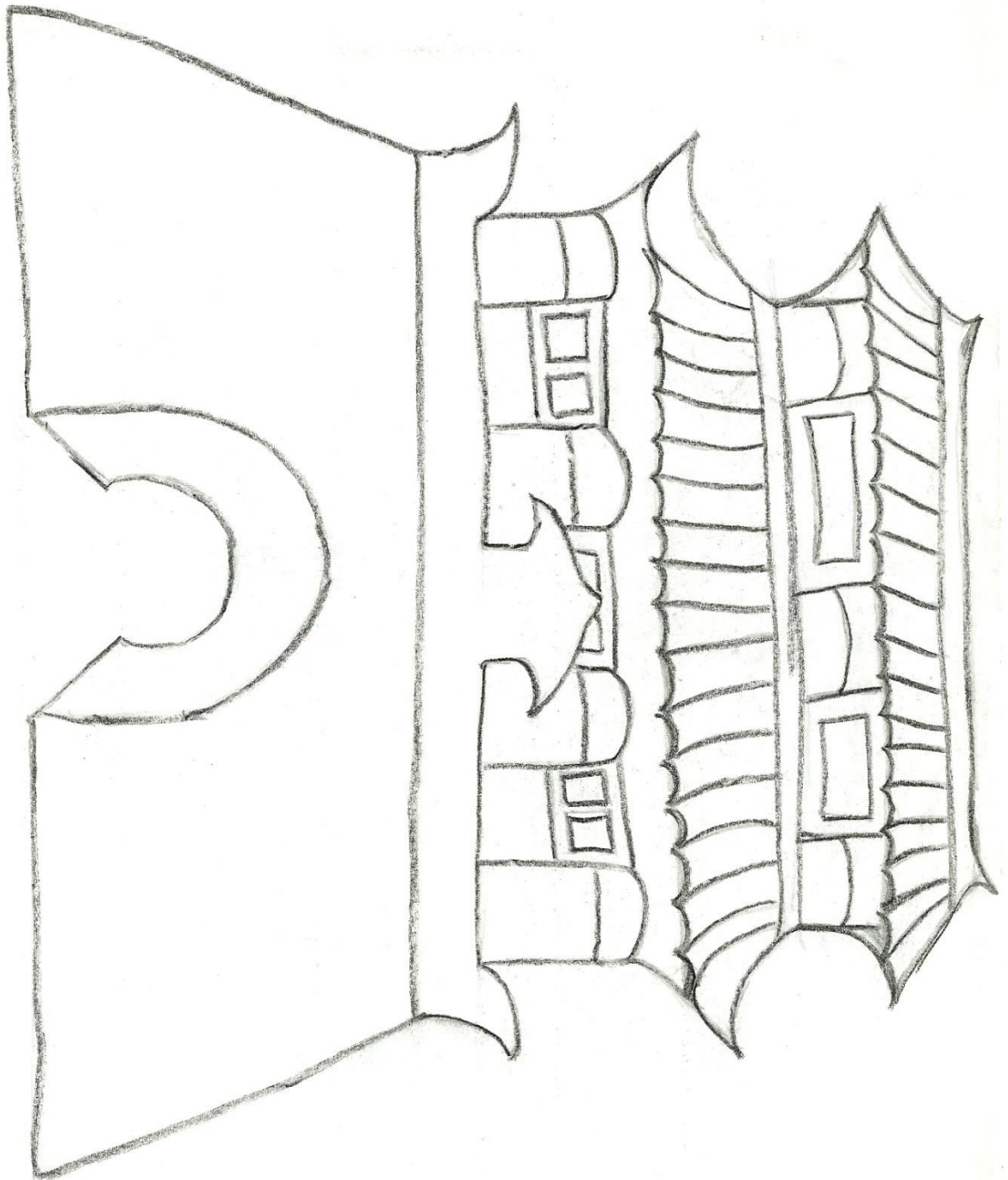


Figura No 8: Boceto de casas utilizadas

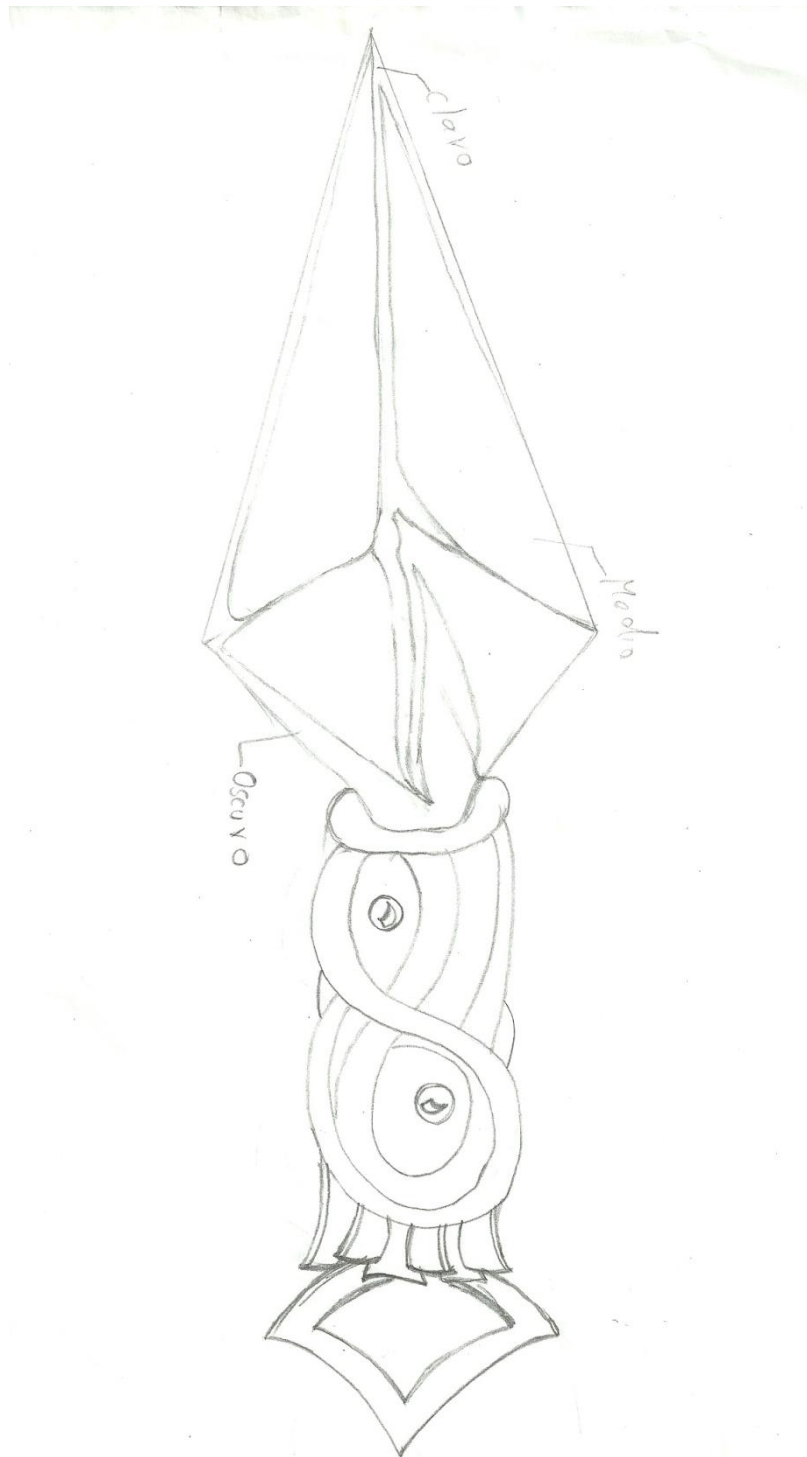
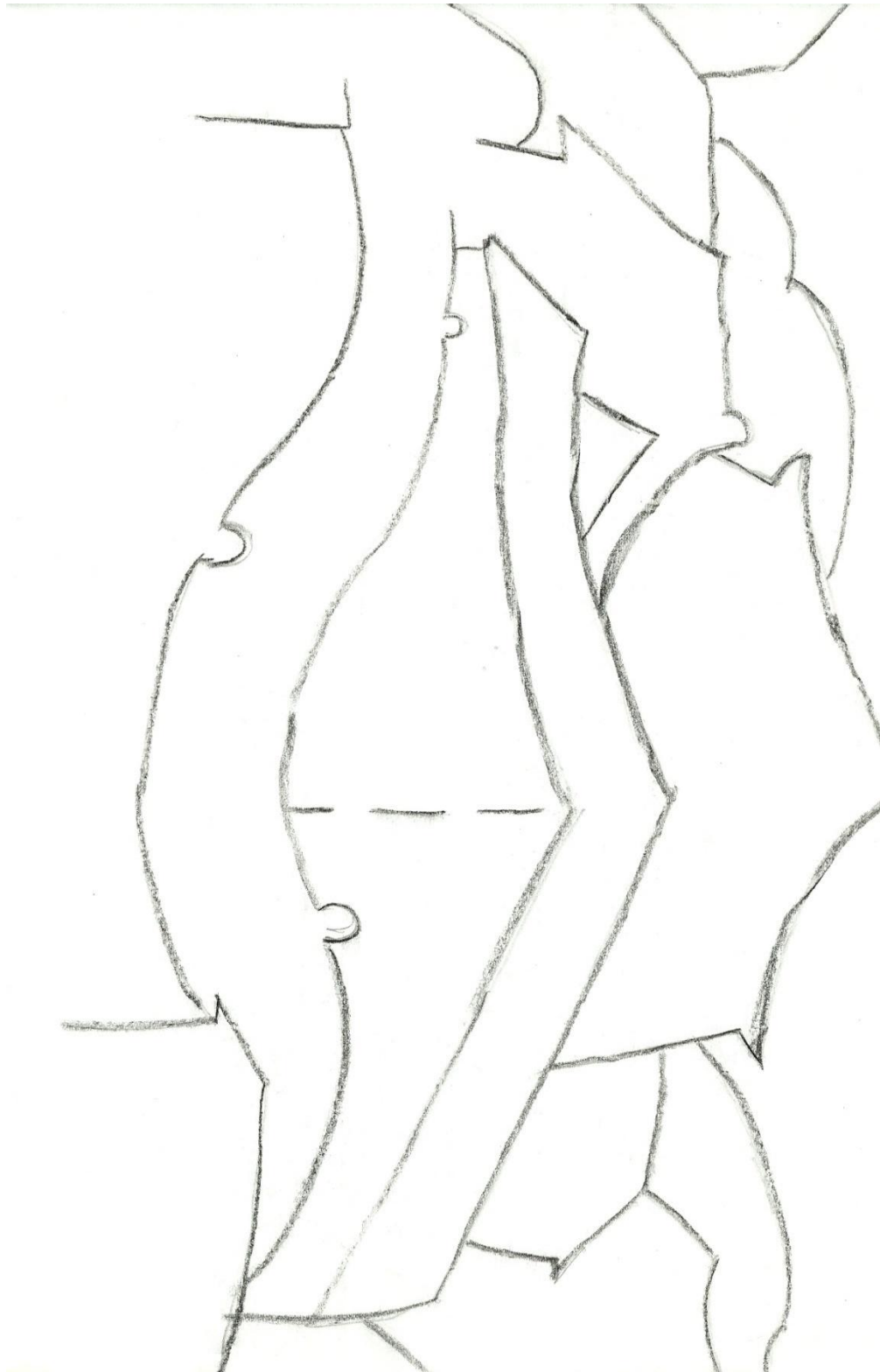


Figura No 9: Boceto botón para regresar



**Figura No 10: Boceto de fondo**