

DESARROLLO DE UN VIDEOJUEGO MMORPG DE TIPO BROWSER-GAME
(MMORPG: MASSIVELY MULTIPLAYER ONLINE ROL PLAYING GAME)

AUTORES

ANGELA MARÍA GÓMEZ MARÍN
CARLOS AUGUSTO ARCINIEGAS MESA

UNIVERSIDAD EAN
FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA DE SISTEMAS
BOGOTÁ, COLOMBIA 2013

DESARROLLO DE UN VIDEOJUEGO MMORPG DE TIPO BROWSER-GAME
(MMORPG: MASSIVELY MULTIPLAYER ONLINE ROL PLAYING GAME)

AUTORES

ANGELA MARÍA GÓMEZ MARÍN
CARLOS AUGUSTO ARCINIEGAS MESA

Tesis de grado presentada como requisito parcial
para optar el título de:

INGENIERÍA DE SISTEMAS

Modalidad:

MONOGRAFÍA

Asesor:

INGENIERO RUBEN DORADO
DOCENTE ACADÉMICO

UNIVERSIDAD EAN
FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA DE SISTEMAS
BOGOTÁ, COLOMBIA 2013

2013

Nota de aceptación

Presidente del Jurado

Jurado

Jurado

Ciudad y Fecha

DEDICATORIA

A todas aquellas personas que tienen un gran interés en el mundo de los videojuegos, donde a través de la interacción e imaginación, sueñan con ser partícipes en la creación y desarrollo de un juego de rol online y consideran que es demasiado complejo e inalcanzable, sin embargo por medio de este proyecto, les presentamos que es posible lograrlo y que por medio del interés, dedicación y amor se alcanzan los sueños.

AGRADECIMIENTOS

A Dios, quien nos da grandeza para entender, método y facultad para aprender, dirección y guía para progresar y continuar con nuestras metas.

A nuestras familias y amigos que nos han apoyado para el desarrollo del videojuego.

A los docentes Sandra Cristancho, Juan David Ospina, Alexander García y Rubén Dorado, que han creído en nuestro proyecto y a través de sus enseñanzas, consejos y conocimientos han contribuido el desarrollo de este proyecto.

TABLA DE CONTENIDO

1. TITULO.....	11
2. INTRODUCCIÓN.....	12
3. PLANTEAMIENTO DEL PROBLEMA.....	13
3.1. Enunciado del Problema.....	14
3.2. Formulación del Problema.....	14
4. OBJETIVOS.....	15
4.1 OBJETIVO GENERAL.....	15
4.2 OBJETIVOS ESPECIFICOS.....	15
5. JUSTIFICACION.....	16
6. ALCANCE.....	17
7. MARCO DE REFERENCIA.....	18
7.1 Marco Conceptual.....	18
7.1.1 Concepto general de un MMORPG.....	18
7.1.2 Características de un MMORPG.....	18
7.1.3 Historia de los MMORPG.....	19
7.1.4 MMORPGs BrowserGame más populares.....	22
7.2 Marco Teórico.....	23
7.2.1 Bases de datos relacionales.....	23
7.2.2 Programación en Ruby.....	25
7.2.2.1 Características de Ruby.....	26
7.2.3 Programación Orientada a Objetos en Ruby.....	27
7.2.4 Ruby on Rails.....	28
7.2.5 Servidor Web de Ruby Webrick.....	29
7.2.6 Modelo vista Controlador en ROR.....	29
7.2.7 Comparación entre Java y Ruby.....	30
8. METODOLOGIA DE DESARROLLO.....	31
8.1 Ventajas de la Metodología (XP).....	32
9. ARQUITECTURA DEL MMORPG.....	33
9.1 ANALISIS DE REQUERIMIENTOS.....	33
9.1.1 Requerimientos Funcionales.....	33
9.1.2 Requerimientos No Funcionales.....	40
9.1.3 Diagrama De Caso De Uso.....	43
9.1.4 Diagramas De Secuencia.....	59
9.1.5 Diagrama De Clases.....	67
9.1.6 Modelo Entidad Relación.....	68
10. IMPLEMENTACIÓN DE LA SOLUCIÓN.....	69
10.1 Diseño del MMOPRG y Plataforma.....	63
10.2 Story Game del videojuego.....	73

10.3 Mecánica del Juego.....	75
10.4 Interacción para los Jugadores.....	82
10.5 Sistemas de Ataques.....	82
10.6 Sistema de Mensajes.....	85
10.7 Sistema de Alianzas.....	86
10.8 Guerras entre alianzas.....	86
10.9 SOFTWARE.....	86
12. CONCLUSIONES.....	90
13. BIBLIOGRAFIA.....	92

LISTA DE ILUSTRACIONES

Ilustración 1 y 2 Primeros MORPG.....	20
Ilustración 3 Blizzard Entertainment.....	21
Ilustración 4 Diseño de Base de Datos Relacional.....	24
Ilustración 5 Ejemplo de Ruby.....	26
Ilustración 6 MVC Ruby.....	30
Ilustración 7 Metodología XP.....	32
Ilustración 8 CU001 Login.....	44
Ilustración 9 CU002 Registro.....	45
Ilustración 10 CU002 Escoger Raza.....	46
Ilustración 11 CU004 Compra Edificaciones.....	48
Ilustración 12 CU005 Comprar Aldeanos.....	49
Ilustración 13 CU006 Obtener Recursos.....	50
Ilustración 14 CU007 Investigaciones Por Raza.....	52
Ilustración 15 CU008 Comprar Ejército.....	53
Ilustración 16 CU009 Alianzas.....	55
Ilustración 17 CU010 Versus.....	56
Ilustración 18 CU011 Mensajes.....	57
Ilustración 19 CU012 Actualizar Fortaleza.....	58
Ilustración 20 Ingreso y Registro del Jugador.....	59
Ilustración 21 Compra de Edificaciones.....	60
Ilustración 22 Compra de Aldeanos.....	60
Ilustración 23 Obtener Recursos.....	61
Ilustración 24 Investigaciones Por Raza.....	62
Ilustración 25 Compra Ejército.....	63
Ilustración 26 Alianzas.....	64
Ilustración 27 Versus.....	65
Ilustración 28 Mensajes.....	66
Ilustración 29 Actualizar Fortaleza.....	66
Ilustración 30 Diagrama de Clases.....	67
Ilustración 31 Modelo Entidad Relación.....	68
Ilustración 32 Boceto Orco Lancero.....	70
Ilustración 33 Orco Lancero.....	70
Ilustración 34 Boceto Orco Arquero.....	71
Ilustración 35 Orco Arquero.....	71
Ilustración 36 Orco en Combate.....	72
Ilustración 37 Orco Hacha.....	72
Ilustración 38 Autenticación de Gaia Conquest.....	76
Ilustración 39 Provincia y vista principal del juego.....	77

Ilustración 40	Guía.....	.78
Ilustración 41	Compra de Aldeanos y Edificios.....	.79
Ilustración 42	Asignar Aldeanos a la Mina Mithril.....	.79
Ilustración 43	Asignar Aldeanos a la Granja.....	.79
Ilustración 44	Cantera de Piedra.....	.80
Ilustración 45	Aserradero.....	.80
Ilustración 46	Cuartel.....	.80
Ilustración 47	Selección de Ataque.....	.81
Ilustración 48	Enviar Mensaje.....	.81
Ilustración 49	Historial de Mensajes.....	.82
Ilustración 50	Ataque.....	.84
Ilustración 51	Enviando Ataque.....	.84
Ilustración 52	Resultado de la Batalla.....	.85
Ilustración 53	Mensaje Recibido.....	.85
Ilustración 54	Alianzas.....	.86

LISTA DE TABLAS

Tabla 1	RF Validación de usuario.....	34
Tabla 2	RF Registro de usuario.....	35
Tabla 3	RF Selección de Raza.....	35
Tabla 4	RF Compra de Edificaciones.....	36
Tabla 5	RF Comprar de Aldeanos.....	36
Tabla 6	RF Obtener Recursos.....	37
Tabla 7	RF Investigaciones por Raza.....	37
Tabla 8	RF Compra de Ejército.....	38
Tabla 9	RF Creación de Alianzas.....	38
Tabla 10	RF Ataques entre Jugadores.....	39
Tabla 11	RF Envío de Mensajes.....	39
Tabla 12	RF Actualizar Fortaleza.....	40
Tabla 13	RNF Disponibilidad.....	41
Tabla 14	RNF Intuitivo.....	41
Tabla 15	RNF Instalación.....	42
Tabla 16	RNF Navegación.....	42
Tabla 17	RNF Sujeto a Cambios.....	43

1. TITULO

DESARROLLO DE UN VIDEOJUEGO MMORPG DE TIPO BROWSER-GAME
(MMORPG: MASSIVE MULTIPLAYER ONLINE ROL PLAY GAME)

2. INTRODUCCIÓN

La industria del entretenimiento digital y videojuegos, es un sector que en los últimos 30 años ha evolucionado junto con los avances de la tecnología, lo que ha permitido un continuo crecimiento y un manejo fuerte en el mercado mundial de una forma masiva, donde el 65% de los hogares de Norteamérica juegan videojuegos, 71% en América Latina, 68% en Europa y 82% en Asia, según lo indica CIU(Competitive Intelligence Unit)¹ donde confirma que los países que lideran este tipo de desarrollos son Estados Unidos, Canadá, Australia y Japón principalmente.

¹ <http://www.the-ciu.net/>

3. PLANTEAMIENTO DEL PROBLEMA

3.1. Enunciado del Problema

Actualmente existe en Colombia la Asociación de Desarrolladores de Videojuegos (IGDA), la cual fomenta el estudio de tecnologías para su desarrollo e implementación; según IGDA hasta inicios del 2011, en el país no se ha habido mayor auge o interés frente a la creación de videojuegos, sin embargo, es una industria en vías de maduración con un enfoque al desarrollo de juegos de acción en primera persona, es decir, en 3D, lo cual conlleva a tener un fuerte contenido audiovisual y potentes equipos para generar una imagen desde un modelo rendering animación, entre muchos otros requerimientos de alto costo.

No se tiene evidencia de otro tipo de juegos desarrollados en Colombia que por medio de la interacción contribuyan a generar entretenimiento a nivel masivo, como los MMORPGs de tipo *Browser game*, los cuales tienen un componente especial como lo es la red social y un fuerte impacto a nivel internacional al momento de interactuar con este tipo de juegos, el cual según estudio de mercado realizado por IGDA durante el 2011 y 2012, muestra que de las 36 empresas dedicadas a este tipo de entretenimiento en el país, solo se basan en desarrollo de videojuegos 2D y 3D para consolas y dispositivos móviles y juegos enfocados al sector educativo.

3.2. Formulación del problema

¿Cómo generar interacción a nivel masivo utilizando un MMORPG?

4. OBJETIVOS

4.1 OBJETIVO GENERAL

Desarrollar un videojuego del genero MMORPG de tipo BrowserGame, que ofrezca interacción entre los jugadores.

4.2 OBJETIVOS ESPECÍFICOS

- Crear el Story Game del videojuego.
- Diseñar la mecánica de interacción para los usuarios.
- Elaborar los modelos UML del videojuego.
- Desarrollar el videojuego.

5. JUSTIFICACION

Actualmente el entretenimiento online, permite que la comunicación entre personas de diferentes partes del mundo maneje una interacción cercana a través de un videojuego. El MMORPG de tipo *Browser game* planteado, es un juego de rol multijugador online, basado en ilustraciones y estadísticas, es decir, no posee animación ni tampoco utiliza un modelo de tipo *rendering*, existe la oportunidad de innovar con un juego de rol online, cuya principal característica es generar interacción a nivel masivo y por ende entretenimiento, donde se despierte el interés hacia éste tipo de videojuegos, fomentando su mercado y desarrollo. Así mismo, el MMORPG impulsa el desarrollo y la competitividad para la producción de éste tipo de entretenimiento a nivel nacional e internacional.

6. ALCANCE

El jugador tiene la opción de escoger una raza entre dos posibles, puede construir cinco (5) edificaciones económicas con los cuales puede generar ingresos para su imperio, también dos (2) edificaciones militares para crear cuatro (4) tipos de soldados diferentes, así mismo, el usuario puede crear o hacer parte una alianza de máximo 10 jugadores. Existe un sistema de comunicación entre jugadores. Se pueden crear nuevos aldeanos desde el centro urbano, el cual esta presente desde el inicio del juego.

Se permite realizar uno o varios ataques simultáneos a cualquier otro jugador registrado, adicional se pueden declarar guerras entre alianzas y llevar un sistema de Ranking de mejor alianza y mejor jugador.

El sistema del juego es basado únicamente en gráficas estáticas y estadísticas las cuales se producen según la lógica del juego, se puede registrar solo un nickname por e-mail y por dirección IP.

El *Browser game* no incluye ningún tipo de animación, ni se detallará a profundidad los costos de producción del Browsergame.

7.1 MARCO REFERENCIAL

7.1 Marco Conceptual

7.1.1 Concepto general de un MMORPG

El MMORPG es un juego de rol multijugador en línea, donde existe un mundo virtual al cual pueden conectarse e interactuar entre sí personas de cualquier parte del mundo. Este mundo virtual es administrado por el creador del MMORPG y existe independiente del jugador, es decir, que aunque no hayan jugadores conectados, este mundo virtual está en la web y disponible en todo momento.

Dependiendo del tipo de MMORPG los requerimientos a nivel de servidores web suelen cambiar, desde requerimientos básicos y de bajo costo, hasta servidores dedicados con mucho espacio de almacenamiento y capacidad de procesamiento.

El rol es un estilo de juego en el cual el jugador evoluciona según las decisiones e interacciones que tome en el transcurso del juego, en el cual no necesariamente existen misiones específicas como en otro tipo de juegos donde se deben cumplir objetivos que van narrando una historia y donde existe un final predeterminado por el desarrollador. En este estilo de juego de rol en línea, solo se crea un entorno y ambientación, pero cada jugador desarrolla su propia historia.

10.1.2 Características de un MMORPG

- Por lo general los MMORPG tienen una temática ambientada en personajes y mundos de fantasía o mundos épicos medievales o antiguos, aunque esto no es una regla general para el desarrollo de un MMORPG.
- Posee un fuerte componente social, debido a la red de jugadores que entran e interactúan entre sí en estos mundos virtuales, es común que se creen fuertes lazos afectivos o de

enemistad por medio de este tipo de juegos, lo que genera afición.

- El juego ofrece la posibilidad de que cada jugador evolucione con un personaje o con un imperio según las decisiones que tome, por ejemplo, tener un gran poder militar, un status social dentro de la red, un Ranking en el cual busca escalar cada vez más y un gran poder económico.
- Existen moderadores para este tipo de juegos, al igual que existe para un foro, su razón de ser es velar por el buen uso que den los usuarios al juego, esta persona puede realizar acciones de tipo administrativo, como lo son: vetar usuarios por mensajes ofensivos hacia otros jugadores por determinado tiempo, expulsar usuarios del juego y dar soporte y guía a usuarios novatos.
- Existen varios tipos de MMORPG: En 3D, los cuales utilizan animación en 3 dimensiones y suelen visualizarse en primera o tercera persona; en 2D, que suelen contener animación en 2 dimensiones los cuales se pueden apreciar por tener tableros planos y por último, basado imágenes estáticas y estadísticas, los cuales han adoptado también el nombre de *Browser game* y su principal característica es que no requieren ningún tipo de instalación debido a que todo su contenido está almacenado en el lado del servidor.

7.1.3 Historia de los MMORPG

Su inicio comienza a principios de los años 90 donde heredan varios conceptos entre ellos videojuegos multiusuario (MUD), los cuales eran juegos basados en diálogos, aventuras graficas y juegos en 2D cooperativos; heredan también de los juegos de rol de mesa como el famoso Calabozos y Dragones², donde los jugadores creaban su propia historia y la evolución de sus personajes dependía de la toma de decisiones.

² http://www.linkmesh.com/Dragons/calabozos_y_dragones

Los primeros MMORPG fueron Neverwinter Nights⁵ creado por BioWare y el aclamado Final Fantasy de la compañía Japonesa Square⁶, donde la evolución del juego se basa en toma de decisiones, tal como se observa en las ilustraciones 1 y 2.



Ilustración 1 Primer MMORPG³



Ilustración 2 Primer MMORPG⁴

³ <http://www.neoteo.com>

⁴ <http://gamerise.ru>

⁵ <http://mashable.com/2012/11/14/mmorpgs-history/>

⁶ <http://www.square-enix.com/>

Los MMORPG era un tipo de videojuego que no presentaba mayor importancia dentro de esta industria, sin embargo para compañías como Electronic Arts⁸, que utilizaron el concepto de MMORPG y la importancia de su negocio, sacaron provecho de este y crean un videojuego llamado Ultima Online, donde se inicia una competencia y crecimiento entre este tipo de videojuegos, permitiendo así que empresas como Sony Online Entertainment entren al mercado de los MMORPG 3D con EverQuest y Microsoft con Asheron's Call, siendo catalogados como los 3 mejores de la década de los 90; lo que significo iniciar con una nueva generación de videojuegos de este estilo.

En la década del 2000 los MMORPG han evolucionado en varios tipos, entre ellos los sofisticados juegos en 3 dimensiones como World of Warcraft de la compañía Blizzard Entertainment⁷, (ver ilustración 3) el cual ha sido catalogado como uno de los mejores juegos masivos en 3D por "Videogame Awards" y la cual fue premiada en el 2011.



Ilustración 3 Blizzard Entertainment⁷

⁷ <http://www.blizzard.com>

⁸ <http://www.ea.com>

7.1.4 MMORPG's BrowserGame más populares.

A finales de la década del 2000, los MMORPG de tipo BrowserGame han cobrado popularidad debido a su facilidad de uso ya que implica registrarse a una página web e iniciar el juego inmediatamente, pues no requiere ningún tipo de instalación local y los requerimientos de máquina de computo son mínimos. Algunos de los más populares son:

- Hattrick: Juego de rol acerca de futbol y dirección técnica.
- Travian: Juego de rol épico de civilizaciones europeas, como galos, romanos, godos y germanos.
- Imperia online: Juego de rol épico medieval.

Estos browser-game cuentan en la actualidad con miles de usuarios registrados en todo el mundo en diferentes servidores y son de acceso gratuito, por otro lado cuentan con un interesante modelo de negocio, basado en la micro transacción, esto es, pagos esporádicos que un usuario elige o no hacer para mejorar ciertos aspectos de su desarrollo, como lo es pagar por un ejército de mercenarios, pagar por recursos económicos que brinda el juego, como (madera, oro) entre otros, dependiendo del tipo de juego, lo que permite evolucionar más rápido y tener cierta ventaja sobre los jugadores adversarios.

7.2 MARCO TEÓRICO

7.2.1 Bases de datos relacionales

Una base de datos es un conjunto de información que permite ser almacenada y tener un acceso directo e indirecto a cada uno de los registros que la componen, al ser una base de datos relacional quiere decir que maneja un modelo de entidad relación (MER) el cual permite establecer interconexiones o relaciones entre las tablas contenedoras de los registros que componen la base de datos. Por medio de las relaciones, se puede mantener la integridad de los datos, evitar la redundancia de estos, lo que permite tener coherencia, fiabilidad y evitando inconsistencias en la información. Los datos almacenados dentro de una base de datos relacional están organizados en tablas también conocidas como entidades, sobre los cuales se puede insertar, actualizar, eliminar y consultar registros. Estas tablas o entidades se pueden manipular no solo para almacenar y consultar un volumen grande y ordenado de información sino también, para crear transacciones requeridas por la aplicación de software que la manipula.

Las transacciones son actualizaciones múltiples y simultaneas que se realizan sobre una tabla, generando de esta forma tablas paramétricas con las cuales el software que las utiliza toma decisiones con base a su contenido. Dentro del modelo relacional, se debe tener en cuenta la asociación entre entidades, las cuales pueden ser de tres tipos:

- Relación 1-1: La relación uno a uno es cuando un registro de una tabla solo puede estar relacionado con un único registro de otra tabla.
- Relación 1-n: La relación uno a muchos es cuando un registro de una tabla B solo puede estar relacionado con un único registro de otra tabla A y un registro de una tabla A, puede tener más de un registro relacionado en una tabla B.

- Relación n-n: La relación mucho a muchos es cuando un registro de una tabla puede estar relacionado con más de un registro de otra tabla.

Una de las representaciones graficas para del manejo relacional, es como lo muestra la ilustración 4.

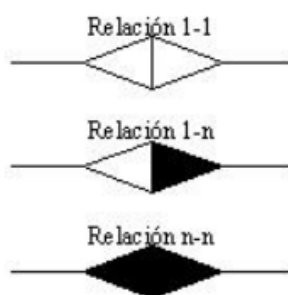


Ilustración 4 Diseño de Base de Datos Relacional⁹.

En una base de datos relacional se debe manejar un proceso de normalización, el cual es uno de los factores más importantes dentro de esta; existen cinco formas normales que al ser aplicadas permiten eliminar redundancias e inconsistencias de dependencia en el diseño de las tablas, protegiendo así, la integridad de los datos y manteniendo las buenas prácticas para el diseño e implementación de la base de datos.

Es común que al aplicar las tres primeras formas normales la base de datos quede parametrizada y permita que el manejo sobre ella sea eficiente.

Las Formas Normales que se implementan en una base de datos son:

- Primera Forma Normal (1FN): Se encuentra en primera forma normal eliminando las columnas que se encuentren repetidas, están deben ser tablas independientes, así mismo cada fila debe tener un único identificador o llave primaria.

⁹ Diseño de bases de datos relacionales, Editorial: El Cid Editor (2007) Argentina.

- Segunda Forma Normal (2FN): Se encuentra en segunda forma normal si y solo si se encuentra en 1FN, de igual forma las dependencias parciales se deben eliminar, lo que permite eliminar redundancias.
- Tercera Forma Normal (3FN): Solo se está en tercera forma normal si y solo si ha pasado por la 2FN, además no debe existir ninguna dependencia funcional transitiva entre los atributos que no son clave primaria.
- Cuarta Forma Normal (4FN): Se está en cuarta forma normal si y solo si ha pasado por la 3FN, adicional no debe tener dependencias múltiples.
- Quinta Forma Normal (5FN): Se encuentra en quinta forma normal si y solo si ha pasado por la cuarta forma normal, así mismo se valida que no se tenga redundancia en la base de datos.

Las bases de datos permiten almacenar, procesar y proteger los datos, algunos de los motores de bases de datos existentes son:

- Oracle
- Mysql
- Sql Server
- Postgres Sql

Por tal razón el diseño, construcción y manejo que se le dé a una base de datos, permite que dentro de una aplicación los procesos que se manejen, mantengan un rendimiento óptimo, seguro y lo más importante mantener la integridad de los datos.

7.2.2 Programación en Ruby

Ruby es un lenguaje de programación orientado a objetos, es un lenguaje flexible e interpretado, el cual fue creado por Yukihiro Matsumoto conocido en el mundo de Ruby como Matz en el año de 1995. El objetivo de este lenguaje es que sea divertido y productivo para el desarrollador, tal como lo menciona Matz

“trato de hacer que Ruby sea natural, no simple, de una forma que se asemeje a la vida real, Ruby es simple en apariencia, pero complejo por dentro, como el cuerpo humano”¹⁰.

Desde 1995, Ruby ha adquirido una comunidad muy fuerte a nivel mundial, tanto así que dentro del índice TIOBE¹², el cual se encarga de medir y analizar el crecimiento de los lenguajes de programación, indica que Ruby para el 2013 se encontrará dentro los 10 primeros lenguajes mas usados, pues su crecimiento durante los últimos 4 años ha sido atribuida por la gran popularidad que ha generado.

Ruby es un lenguaje multiplataforma, fácil de aprender, no utiliza simbología, pues fue creado para ser más entendible y natural al lenguaje humano, lo que conlleva a utilizar una sintaxis mas limpia, es productivo y dinámico. Su filosofía es totalmente enfocada a la importancia humana más que a la máquina, tal como lo expresa Matsumoto:

“A menudo la gente, especialmente los ingenieros en computación, se centran en las máquinas. Ellos piensan, "Haciendo esto, la máquina funcionará más rápido. Haciendo esto, la máquina funcionará de manera más eficiente. Haciendo esto..." Están centrados en las máquinas, pero en realidad necesitamos centrarnos en las personas, en cómo hacen programas o cómo manejan las aplicaciones en los ordenadores. Nosotros somos los jefes. Ellos son los esclavos”¹¹.

7.2.2.1 Características de Ruby

- Orientado a Objetos.
- Manejo de Excepciones

¹⁰ Matsumoto, hablando en la lista de correo Ruby-Talk, 12-05-2000
<http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-talk/2773>

¹¹ Filosofía de Ruby, conversación con Yukihiro Matsumoto, Parte I por Bill Venners 29-09-2003 (Artima Developer). <http://www.artima.com/intv/ruby4.html>

¹² <http://www.tiobe.com>

- Maneja cuatro niveles de ámbito de variables: global, clase, instancia y local.
- Iteradores
- Expresiones regulares nativas.
- Sobrecarga de Operadores
- Recolector de Basura Automático (mark-and-sweep garbage collector)
- Manejo de hilos de ejecución simultáneos.
- Altamente portable

Ruby tiene influencia de Lenguajes como Perl, Smalltalk, Eiffel, Ada, y Lisp, fue creado para la felicidad de los programadores, y actúa bajo licencias libres.

7.2.3 Programación Orientada a Objetos en Ruby

En Ruby todo es un objeto, todo aquello que se manipula en Ruby, es a su vez un objeto, y los resultados de estas operaciones se convierten en un objeto, un ejemplo de esto es asignar una acción a un número en un código simple, tal como lo muestra la siguiente ilustración 5:

```
5.times { print "Nos *encanta* Ruby" }
```

Ilustración 5 Ejemplo de Ruby

Pues en muchos lenguajes de programación no es posible que los números y otros tipos de datos primitivos, sean identificados como objetos. Ruby maneja módulos que permiten incorporar métodos, logrando manejar una herencia simple, es decir, si se implementa un método y a este se le asigna un módulo, todos los métodos que contenga dicho módulo podrán ser utilizados por el método asignado el cuál le dará acceso a recorrer dichos elementos.

Ruby no necesita declarar variables, ya que utiliza ciertas palabras claves para identificar el tipo de variable, de esta manera facilita que el desarrollador identifique de una forma simple los roles que cumplen las variables, son especificadas de la siguiente manera:

- `var` puede ser una variable local.
- `@var` es una variable de instancia.
- `$var` es una variable global.

7.2.4 Ruby on Rails

Es el Framework de Ruby para el desarrollo de aplicaciones web, conocido también como ROR, este framework es de código abierto, utiliza el patrón de diseño Modelo Vista Controlador (MVC), este Framework fue creado por el Noruego David Heinemeier Hansson y fue expuesto para su uso en el 2004, como regla o filosofía establecida para ROR es "No te Repitas"¹³.

Lo que indica que es posible manipular los registros de una tabla como un objeto, y a sus campos como un atributo sin necesidad de declararlos nuevamente entre clases para acceder a ellos. Algunos de los elementos para el desarrollo de una aplicación bajo ROR son:

- **Action Controller:** Se encarga de los controladores de la aplicación, es decir que procesa peticiones, extrae parámetros y ejecuta.
- **Action View:** Se encarga del manejo de las vistas, manejando html, xhtml y soporte de Ajax built-in.
- **Active Record:** Se encarga de los modelos de la base de datos.
- **Action Mailer:** Es un framework para crear servicios de correo electrónico.

¹³ Filosofía de RoR, David Heinemeier Hansson Julio-2004
<http://www.rubyonrails.org.es/>

- **Active Resource:** Este framework permite manipular la conexión entre objetos de negocio y RESTful web services.
- **Railties:** Permite crear nuevas aplicaciones ya que es el código del núcleo de Rails.
- **Active Support:** Es donde se encuentran las clases y extensiones de la biblioteca estándar de Ruby.

7.2.5 Servidor Web de Ruby Webrick

Ruby tiene su propio servidor de aplicaciones, el cual permite acceder a las aplicaciones desarrolladas bajo ROR, por medio de una URL, Webrick proporciona HTTP, HTTPS con autenticación el cual es liviano y hace parte de la biblioteca estándar de Ruby.

7.2.6 Modelo vista Controlador en ROR

En ROR las clases del modelo están apoyadas por la clase Active Record, el cual facilita para el desarrollador la comunicación con las tablas de la base de datos, ya que al heredar de la clase Active Record el lenguaje se encargara internamente de validar que tabla y que campos usar al momento de hacer llamados a esos campos convertidos en atributos dentro de la aplicación que se esté desarrollando.

A nivel de vista ROR facilita el manejo de componentes de JQuery, además que facilita un manejo simple y adecuado para la interacción del usuario con la vista.

El controlador, se encarga de manejar eventos o acciones que son realizadas por el usuario, los cuales reflejan cambios en el modelo y en la vista, Rails utiliza ActionPack dentro del controlador, el cual contiene la clase ApplicationController.

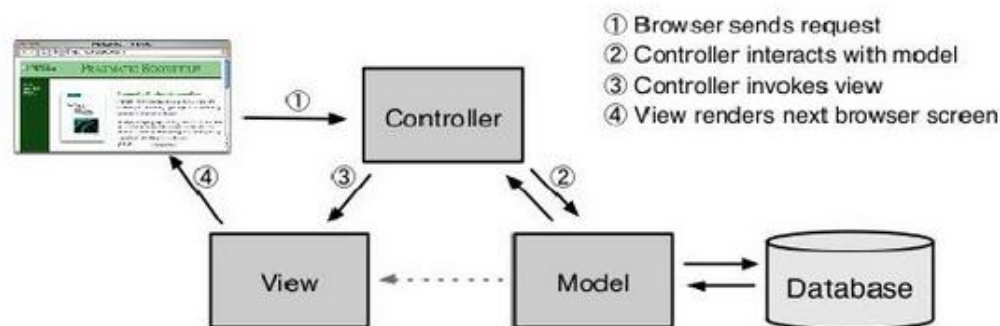


Ilustración 6 MVC Ruby¹⁴

7.2.7 Comparación entre Java y Ruby

Tanto Java como otros lenguajes de programación que son fuertes en el desarrollo web a nivel mundial, poseen características y capacidades que logran sobresalir dentro del uso y manejo en el desarrollo de software, sin embargo es importante dar a conocer las diferencias entre un lenguaje que esta tomando mucha fuerza como lo es Ruby, contra un lenguaje mas maduro y conocido tal como lo es Java, lo permite realizar una comparación de ciertas características relevantes que conllevan a tener un enfoque más amplio entre un lenguaje y otro.

- Tiempo de Desarrollo: "El desarrollo es mucho mas ágil en Ruby que en Java"¹⁵, debido a que al realizar aplicaciones robustas, mantiene su facilidad en el código, no requiere herramientas pesadas de codificación, ni procesos elaborados lo que conlleva a que el tiempo de desarrollo sea mucho más corto, mientras que en Java se maneja una arquitectura multicapa la cual requiere un tiempo mas extenso para ser construida, además de la carga de recursos que maneja y la complejidad en implementación en muchas de sus capas.

¹⁴ <http://www.rubyonrails.org.es/>

¹⁵ <http://www.ruby-lang.org>

- Dentro de Ruby, no es necesario compilar código fuente, este es ejecutado directamente.
- Ruby no utiliza simbología, por lo que en vez de utilizar corchetes al inicio o final de una expresión, maneja palabras más naturales al lenguaje humano como end.
- En Ruby todo se comporta como un objeto, incluyendo los números.
- Dentro de Ruby no hay necesidad de definir el tipo de dato de una variable o método.
- En Ruby no hay transformación de tipos (casting) como si lo es en java, tan solo se hace el llamado a los métodos.
- Se utilizan Mixins en vez de interfaces.

A pesar de las diferencias que se nombraron anteriormente, también se encuentran similitudes dentro de estos dos lenguajes de uso libre las cuales son:

- Tanto Ruby como Java utilizan un garbage collector automático.
- Utilizan métodos públicos, privados y protegidos.
- Son lenguajes que es fuertemente tipado.
- Utilizan herramientas embebidas de documentación.

8. METODOLOGIA DE DESARROLLO

Dentro del proceso de investigación acerca de las metodologías de implementación para el desarrollo de un proyecto de software, la metodología acorde para el desarrollo del MMORPG de tipo BrowserGame es extreme programming (XP), esta metodología se enfoca principalmente en la simplicidad, comunicación e interacción sobre el desarrollo del proyecto de una manera eficiente y altamente efectiva, la técnica a utilizar con esta metodología es por parejas donde uno de los programadores escribe código y el otro prueba, de esta manera se trabaja en paralelo cada modulo a desarrollar, manteniendo la calidad sobre la arquitectura propuesta y de rápido desarrollo sobre el proyecto.

La metodología Extreme Programming posee ciertas características las cuales son: historias de usuario, roles, proceso y prácticas, tal como se puede apreciar en la siguiente grafica.

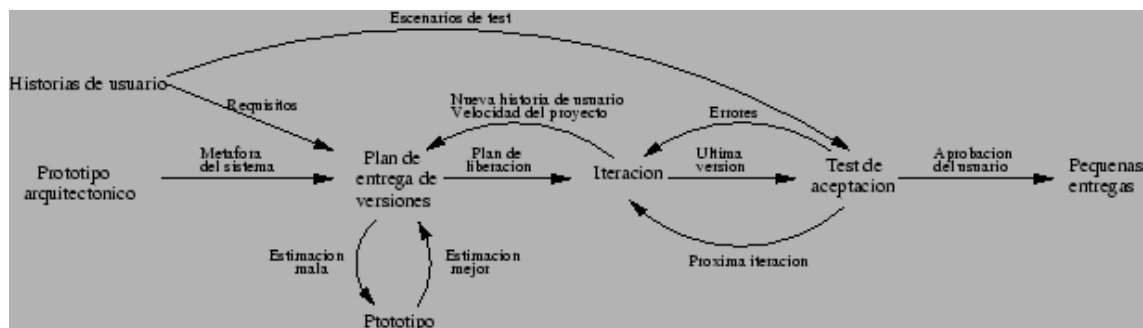


Ilustración 7 Metodología XP¹⁶

Las historias de usuario permiten el análisis de los requerimientos del proyecto, utilizando tarjetas en las cuales se describe las características de la aplicación a desarrollar. Así mismo las historias de usuario logran la creación de pruebas sobre cada requerimiento del sistema a generar, lo cual facilita y optimiza la calidad del desarrollo.

8.1 VENTAJAS DE LA METODOLOGIA

- Permite un desarrollo organizado y con menor margen de errores.
- Su ciclo de vida es iterativo e incremental.
- Permite realizar cambios de modo dinámico.
- Facilita un ágil desarrollo del software.

¹⁶

<http://www.ia.uned.es>

9. ARQUITECTURA DEL MMORPG BROWSERGAME

9.1 REQUERIMIENTOS

La captura de requerimientos para este proyecto es permitir la interacción y entretenimiento que el usuario puede tener con el juego. De esta manera se obtiene una percepción más clara ofreciendo un mejor enfoque acerca de lo que se desea construir.

Para esto se definió el rol jugador, y el rol sistema, donde presenta las siguientes interacciones descriptas a continuación:

9.1.1 REQUERIMIENTOS FUNCIONALES

- RF01. El MMORPG debe permitir la validación de usuario y contraseña.
- RF02. El MMORPG debe permitir el registro de usuario.
- RF03. El MMORPG debe permitir la selección de raza (Orco/Humano).
- RF04. El MMORPG debe permitir la compra de edificaciones civiles (Farm, Quarry, Blacksmith, Barracks, Sawmill, Mithril Mine, Fortresses).
- RF05. El MMORPG debe permitir la compra de aldeanos.
- RF06. El MMORPG debe permitir obtener recursos a través de los aldeanos que trabajan en las edificaciones civiles.
- RF07. El MMORPG debe permitir obtener investigaciones por cada raza (Fire Stone, Mithril Jag, Mead Mountains, Spies, Putrid Woods, Spider Shield, Lizard Dust).
- RF08. El MMORPG debe permitir la compra de ejército o tropas de cada raza (Buy Archers, Buy Spearmen, Buy Infantry, Buy Cavalry).

- RF09. El MMORPG debe permitir la creación de alianzas entre jugadores.
- RF10. El MMORPG debe permitir realizar ataques entre jugadores.
- RF11. El MMORPG debe permitir realizar el envío de mensajes entre jugadores.
- RF12. El MMORPG debe permitir actualizar la fortaleza por cada raza.

En la siguiente tabla 1, se describe el requerimiento funcional RF01, el cual se puede apreciar a continuación:

Numero de requerimiento	RF01
Nombre de requerimiento	Validación de usuario.
Fuente	Usuario
Tipo	☒ Requisito ☐ Restricción
Prioridad del requerimiento	☒ Alta/Esencial ☐ Media/Deseado
Descripción	La aplicación debe permitir la validación de usuario y contraseña del jugador.

Tabla 1 RF Validación de usuario

En la tabla 2 se puede observar el requerimiento RF02, tal como se muestra a continuación:

Numero de requerimiento	RF02
Nombre de requerimiento	Registro de usuario.
Fuente	Usuario
Tipo	☒ Requisito ☐ Restricción
Prioridad del requerimiento	☒ Alta/Esencial ☐ Media/Deseado
Descripción	La aplicación debe permitir el registro del jugador.

Tabla 2 RF Registro de usuario

A continuación se describe el requerimiento RF03, como se ilustra en la tabla 3.

Numero de requerimiento	RF03
Nombre de requerimiento	Selección de Raza (Humano, Orco)
Fuente	Usuario
Tipo	☒ Requisito ☐ Restricción
Prioridad del requerimiento	☒ Alta/Esencial ☐ Media/Deseado
Descripción	La aplicación debe permitir escoger el tipo de raza.

Tabla 3 RF Selección de Raza

A continuación se describe el requerimiento RF04 correspondiente a la tabla 4.

Numero de requerimiento	RF04
Nombre de requerimiento	Compra de Edificaciones
Fuente	Usuario
Tipo	☒ Requisito ☐ Restricción
Prioridad del requerimiento	Alta/Esencial ☒ Media/Deseado ☐
Descripción	La aplicación debe permitir comprar edificaciones civiles.

Tabla 4 RF Compra de Edificaciones.

El requerimiento RF05 se puede observar a detalle en la tabla 5.

Numero de requerimiento	RF05
Nombre de requerimiento	Comprar de Aldeanos
Fuente	Usuario
Tipo	☒ Requisito ☐ Restricción
Prioridad del requerimiento	Alta/Esencial ☒ Media/Deseado ☐
Descripción	La aplicación debe permitir la compra de aldeanos.

Tabla 5 RF Comprar de Aldeanos

En la tabla 6 se puede observar el requerimiento RF06, tal como se muestra a continuación:

Numero de requerimiento	RF06
Nombre de requerimiento	Obtener Recursos
Fuente	Usuario
Tipo	☒ Requisito ☐ Restricción
Prioridad del requerimiento	☒ Alta/Esencial ☐ Media/Deseado
Descripción	La aplicación debe permitir obtener recursos a través de los aldeanos que trabajan en las edificaciones civiles.

Tabla 6 RF Obtener Recursos

En la tabla 7 se observa el requerimiento RF07.

Numero de requerimiento	RF07
Nombre de requerimiento	Investigaciones por Raza
Fuente	Usuario
Tipo	☒ Requisito ☐ Restricción
Prioridad del requerimiento	☒ Alta/Esencial ☐ Media/Deseado
Descripción	La aplicación debe permitir obtener investigaciones por cada raza (Fire Stone, Mithril Jag, Mead Montains, Spies, Putrid Woods, Spider Shield, Lizard Dust).

Tabla 7 RF Investigaciones por Raza

A continuación se puede observar la tabla 8, donde se describe el requerimiento RF08.

Numero de requerimiento	RF08
Nombre de requerimiento	Compra de Ejército
Fuente	Usuario
Tipo	Requisito ☒ Restricción ☐
Prioridad del requerimiento	Alta/Esencial ☒ Media/Deseado ☐
Descripción	La aplicación debe permitir la compra de ejército o tropas de cada raza (Buy Archers, Buy Spearmen, Buy Infantry, Buy Cavarly).

Tabla 8 RF Compra de Ejército

A continuación se observa la tabla 9, la cual describe el requerimiento RF09.

Numero de requerimiento	RF09
Nombre de requerimiento	Creación de Alianzas
Fuente	Usuario
Tipo	Requisito ☒ Restricción ☐
Prioridad del requerimiento	Alta/Esencial ☒ Media/Deseado ☐
Descripción	La aplicación debe permitir la creación de alianzas entre jugadores.

Tabla 9 RF Creación de Alianzas

A continuación se observa la tabla 10, donde se describe el requerimiento RF10.

Numero de requerimiento	RF10	
Nombre de requerimiento	Ataques entre jugadores	
Fuente	Usuario	
Tipo	Requisito 	Restricción 
Prioridad del requerimiento	Alta/Esencial 	Media/Deseado
Descripción	La aplicación debe permitir realizar ataques entre jugadores.	

Tabla 10 RF Ataques entre Jugadores

A continuación se observa el requerimiento RF11, tal como se ilustra en la tabla 11.

Numero de requerimiento	RF11	
Nombre de requerimiento	Envío de Mensajes	
Fuente	Usuario	
Tipo	Requisito 	Restricción 
Prioridad del requerimiento	Alta/Esencial 	Media/Deseado
Descripción	La aplicación debe permitir realizar el envío de mensajes entre jugadores.	

Tabla 11 RF Envío de Mensajes

En la tabla 12, se puede observar el detalle del requerimiento RF 12.

Numero de requerimiento	RF12
Nombre de requerimiento	Actualizar fortaleza
Fuente	Usuario
Tipo	Requisito ☒ Restricción ☐
Prioridad del requerimiento	Alta/Esencial ☒ Media/Deseado ☐
Descripción	La aplicación debe permitir actualizar la fortaleza por cada raza.

Tabla 12 RF Actualizar Fortaleza.

9.1.2 REQUERIMIENTOS NO FUNCIONALES

- RNF01. El MMORPG debe estar disponible independientemente de las preferencias horarias de los jugadores.
- RNF02. El MMORPG debe ser fácil de usar.
- RNF03. El MMORPG no requiere de ningún tipo de instalación.
- RNF04. El MMORPG debe ser funcional en los navegadores conocidos (Internet Explorer, Mozilla Firefox, Google Chrome, Zafari,Opera)
- RNF05. El MMORPG maneja extensibilidad.

A continuación se describe el detalle del requerimiento RNF01 el cual corresponde a la tabla 13.

Numero de requerimiento	RNF01
Nombre de requerimiento	Disponibilidad
Fuente	Usuario
Tipo	Requisito ☒ Restricción ☐
Prioridad del requerimiento	Alta/Esencial ☒ Media/Deseado ☐
Descripción	La aplicación debe permitir la disponibilidad de acceso 7/ 24 al jugador.

Tabla 13 RNF Disponibilidad.

En la tabla 14, se describe el requerimiento RNF 02.

Numero de requerimiento	RNF02
Nombre de requerimiento	Facilidad de Uso
Fuente	Usuario
Tipo	Requisito ☒ Restricción ☐
Prioridad del requerimiento	Alta/Esencial ☒ Media/Deseado ☐
Descripción	La aplicación debe ser fácil de usar.

Tabla 14 RNF Facilidad de Uso.

En la siguiente tabla 15, se observa el detalle del requerimiento RNF 03.

Numero de requerimiento	RNF03
Nombre de requerimiento	Instalación
Fuente	Usuario
Tipo	Requisito ☒ Restricción ☐
Prioridad del requerimiento	Alta/Esencial ☒ Media/Deseado ☐
Descripción	La aplicación no debe requerir ningún tipo de instalación.

Tabla 15 RNF Instalación.

A continuación se observa el detalle del requerimiento RNF 04, el cual corresponde a la tabla 16.

Numero de requerimiento	RNF04
Nombre de requerimiento	Navegación
Fuente	Usuario
Tipo	Requisito ☒ Restricción ☐
Prioridad del requerimiento	Alta/Esencial ☒ Media/Deseado ☐
Descripción	La aplicación no debe ser funcional en los navegadores conocidos (Internet Explorer, Mozilla Firefox, Google Chrome, Zafari,Opera)

Tabla 16 RNF Navegación.

Dentro de la tabla 17, se puede apreciar el requerimiento RNF05.

Numero de requerimiento	RNF05
Nombre de requerimiento	Sujeto a Cambios
Fuente	Usuario
Tipo	Requisito ☒ Restricción ☐
Prioridad del requerimiento	Alta/Esencial ☒ Media/Deseado ☐
Descripción	La aplicación no debe tener la capacidad de implementar nuevas funcionalidades de una forma fácil y rápida sin que afecte al jugador.

Tabla 17 RNF Sujeto a Cambios.

9.1.3 DIAGRAMAS DE CASO DE USO

A continuación se especifican los casos de uso para el desarrollo del MMORPG, con su respectiva imagen.

CU001 Login	
NOMBRE	Validación de usuario y clave.
ALIAS	Login
ACTORES	Jugador, Sistema
DESCRIPCIÓN	Este caso de uso la validación de usuario y clave en la aplicación.
PRE-CONDICIÓN	Para realizar la validación de usuario y clave, el jugador debe haber echo el registro previamente.
REFERENCIA	Requerimiento RF01

SECUENCIA NORMAL DE EVENTOS	1. El usuario ingresa a la web del juego. 2. La aplicación visualiza una breve descripción del juego y de las razas. 3. La aplicación brinda la opción de autenticación, para ingresar el usuario y la contraseña. 4. El jugador ingresa el nickname y clave correspondiente.
FLUJO ALTERNO	Ninguno
POST CONDICIONES	Iniciar sesión satisfactoria en la aplicación.

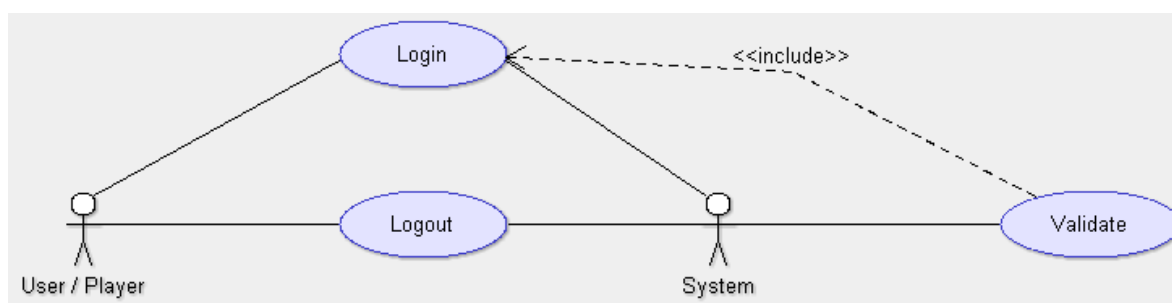


Ilustración 8 CU001 Login

CU002 Registro	
NOMBRE	Registro
ALIAS	Registro
ACTORES	Jugador, sistema
DESCRIPCIÓN	Este caso de uso permite realizar el registro y validación de usuario y clave en la aplicación.
PRE-CONDICIÓN	Ninguna
REFERENCIA	Requerimiento RF02
SECUENCIA NORMAL DE EVENTOS	1. El usuario ingresa a la web del juego. 2. La aplicación visualiza una breve descripción del juego y de las razas. 3. La aplicación muestra la opción de

	registro. 4. El jugador ingresa los datos de registro, nickname, email, contraseña y selección de raza. 5. El sistema almacena la información del nuevo jugador e informa al jugador que el registro fue satisfactorio.
FLUJO ALTERNO	Ninguno
POST CONDICIONES	Iniciar sesión satisfactoria en la aplicación.

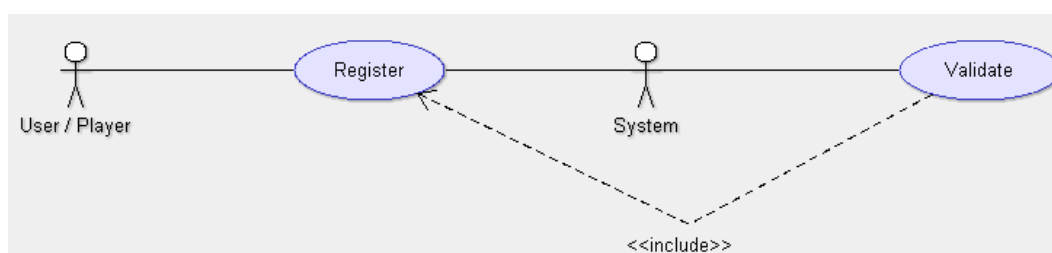


Ilustración 9 CU002 Registro

03 Escoger Raza	
NOMBRE	Escoger Raza
ALIAS	Escoger Raza
ACTORES	Jugador
DESCRIPCIÓN	Este caso de uso permite al usuario escoger un tipo de raza con el cual va a interactuar dentro del juego.
PRE-CONDICIÓN	El jugador debe ingresar los campos anteriores en el formulario de registro para que se haga valida la selección de raza escogida.
REFERENCIA	Requerimiento RF03
SECUENCIA NORMAL DE EVENTOS	1. El jugador ingresa a la opción de registro dentro de la web del juego. 2. La aplicación le permite seleccionar una de las dos razas para interactuar en el juego. 3. El jugador escoge la raza

FLUJO ALTERNO	No tiene
POST CONDICIONES	Iniciar sesión satisfactoria en la aplicación.

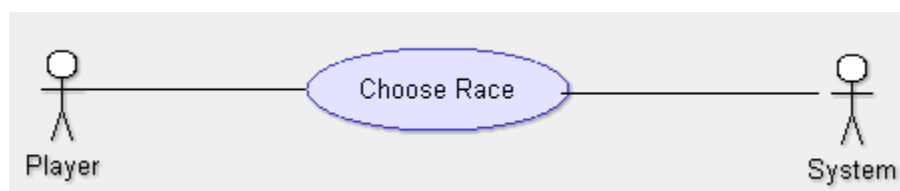


Ilustración 10 CU003 Escoger Raza

CU004 Comprar Edificaciones	
NOMBRE	Comprar Edificaciones
ALIAS	Comprar Edificaciones
ACTORES	Jugador
DESCRIPCIÓN	Este caso de uso permite al usuario comprar edificaciones civiles como lo son: Farm, Quarry, Blacksmith, Sawmill y Mithril mine, el cual podrá obtener recursos, así mismo tendrá la opción de comprar edificaciones militares como: Barrack y Fortress, las cuales le permitirán al jugador entrenar soldados para atacar y defenderse de otros jugadores.
PRE-CONDICIÓN	El jugador debe obtener los recursos necesarios para comprar las edificaciones.
REFERENCIA	Requerimiento RF04
SECUENCIA NORMAL DE EVENTOS	1. El jugador ingresa a la opción de registro dentro de la web del juego. 2. El jugador ingresa el nickname y la clave para acceder a la

	aplicación. 3. El jugador debe seleccionar el centro urbano para realizar la compra edificaciones civiles y militares. 4. El jugador tendrá la opción de escoger el tipo de edificación a comprar, teniendo en cuenta que los costos de las edificaciones varían de acuerdo al tipo. 5. La aplicación valida si se tienen los recursos suficientes para comprar cada edificación, de no ser así hay que esperar obtenerlos para comprar las edificaciones. 6. La aplicación le informará al usuario por medio de un mensaje si fue o no satisfactoria la compra de la edificación.
FLUJO ALTERNO	No tiene
POST CONDICIONES	El jugador interactúa con el juego para la compra de edificaciones, adicional a esto tendrá la opción de tener arqueros dentro de la fortaleza, para defenderse de tropas enemigas.

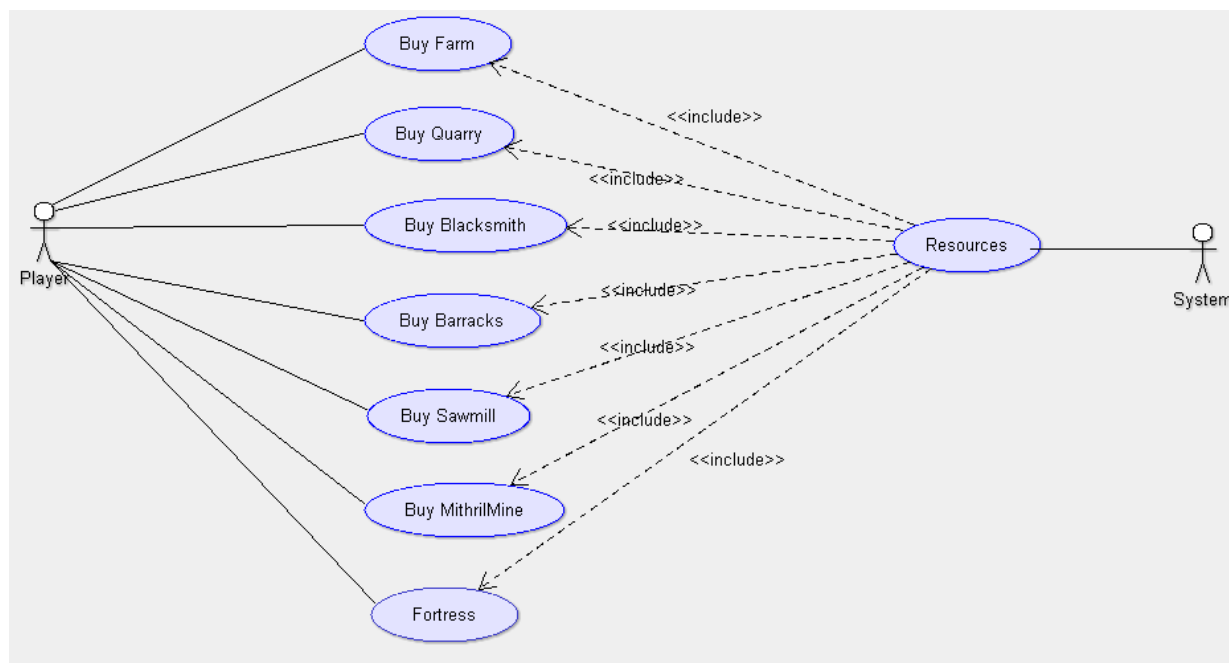


Ilustración 11 CU004 Compra Edificaciones

CU005 Comprar Aldeanos	
NOMBRE	Comprar Aldeanos
ALIAS	Comprar Aldeanos
ACTORES	Jugador
DESCRIPCIÓN	Este caso de uso permite comprar los aldeanos desde el centro urbano, los cuales se encargaran de la recolección de los recursos.
PRE-CONDICIÓN	El jugador debe tener los recursos suficientes para adquirir los aldeanos.
REFERENCIA	Requerimiento RF05
SECUENCIA NORMAL DE EVENTOS	1. El jugador ingresa a la web del juego. 2. El jugador ingresa el nickname y la clave para acceder a la aplicación. 3. El jugador debe seleccionar el centro urbano para realizar la compra de los aldeanos. 4. El jugador ingresa el número de aldeanos a adquirir. 5. El sistema valida si se tienen los

	recursos suficientes para adquirir los aldeanos y muestra un mensaje al usuario indicando si fue posible la compra de los aldeanos, de lo contrario el sistema mostrara un mensaje indicando que no se tienen los recursos suficientes para adquirir aldeanos.
FLUJO ALTERNO	El sistema brinda unos recursos mínimos requeridos por defecto para el jugador, los cuales le permitirán hacer uso libre estos recursos mínimos, para la estrategia e interacción que tenga con el juego.
POST CONDICIONES	El jugador interactúa con el juego, manejando su propia estrategia.

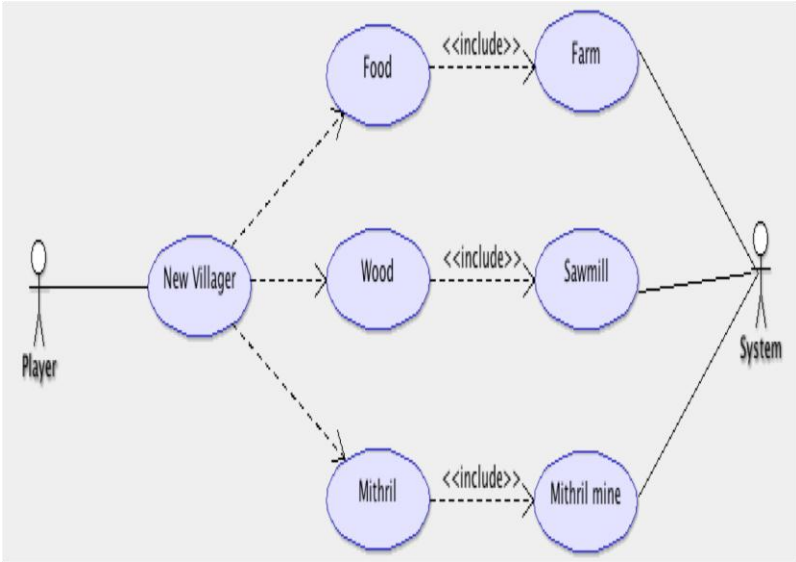


Ilustración 12 CU005 Comprar Aldeanos

CU006 Obtener Recursos	
NOMBRE	Obtener Recursos
ALIAS	Obtener Recursos
ACTORES	Jugador
DESCRIPCIÓN	Este caso de uso permite obtener los recursos por medio de los aldeanos, los

	cuales son: Wood, Mithril, Stone, Steel y Food.
PRE-CONDICIÓN	El jugador deberá tener las edificaciones correspondientes a los recursos que desea adquirir.
REFERENCIA	Requerimiento RF06
SECUENCIA NORMAL DE EVENTOS	1. El jugador ingresa a la web del juego. 2. El jugador ingresa el nickname y la clave para acceder al juego. 3. El sistema verifica cuantos aldeanos ociosos se encuentran. 4. El jugador puede asignar los aldeanos a las edificaciones civiles, en las cantidades que el desee. 5. El sistema confirma al usuario por medio de un mensaje los aldeanos que fueron asignados correctamente.
FLUJO ALTERNO	Ninguno.
POST CONDICIONES	Los recursos son directamente proporcionales a la cantidad de aldeanos trabajando en las edificaciones civiles.

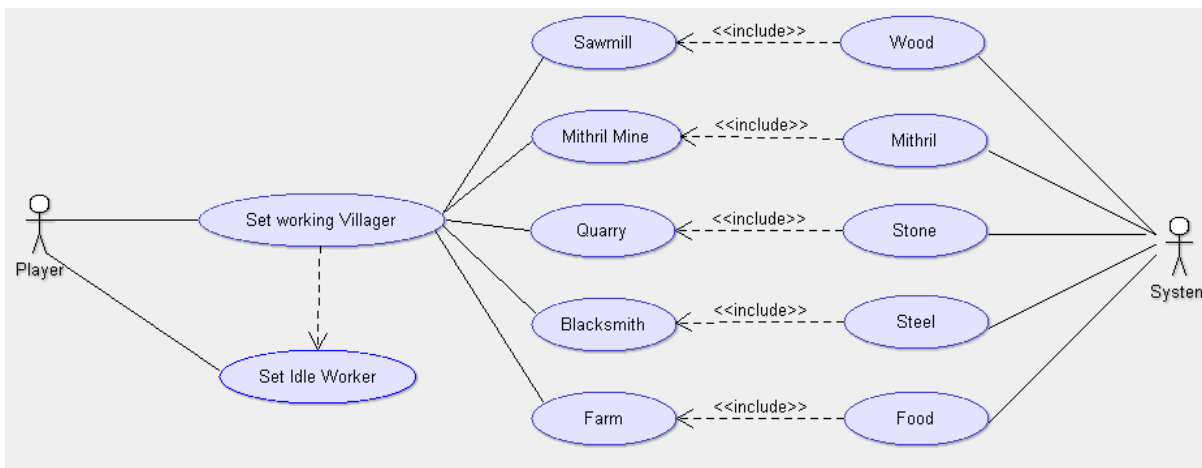


Ilustración 13 CU006 Obtener Recursos

CU007 Investigaciones por Raza	
NOMBRE	Investigaciones por Raza

ALIAS	Investigaciones por Raza
ACTORES	Jugador
DESCRIPCIÓN	Este caso de uso permite obtener investigaciones por raza, las cuales son: Fire Stone, Mithril Jag, Mead Mountains, Spies, Putrid Woods, Spider Shield, Lizard Dust.
PRE-CONDICIÓN	El jugador deberá tener los recursos suficientes para realizar las investigaciones, las cuales permiten mejoras para la defensa y ataque de las tropas.
REFERENCIA	Requerimiento RF07
SECUENCIA NORMAL DE EVENTOS	1. El jugador ingresa a la web del juego. 2. El jugador ingresa el nickname y la clave para acceder al juego. 3. El jugador accede a la fortaleza para realizar las investigaciones. 4. El sistema informa al usuario por medio de un mensaje si es posible realizar la investigación, dependiendo de los recursos con que cuente el jugador.
FLUJO ALTERNO	Ninguno.
POST CONDICIONES	Depende de la investigación obtenida, se realiza una mejora en el nivel de espionaje, en la velocidad de las tropas, en la fuerza de las tropas o en el nivel de defensa de la fortaleza.

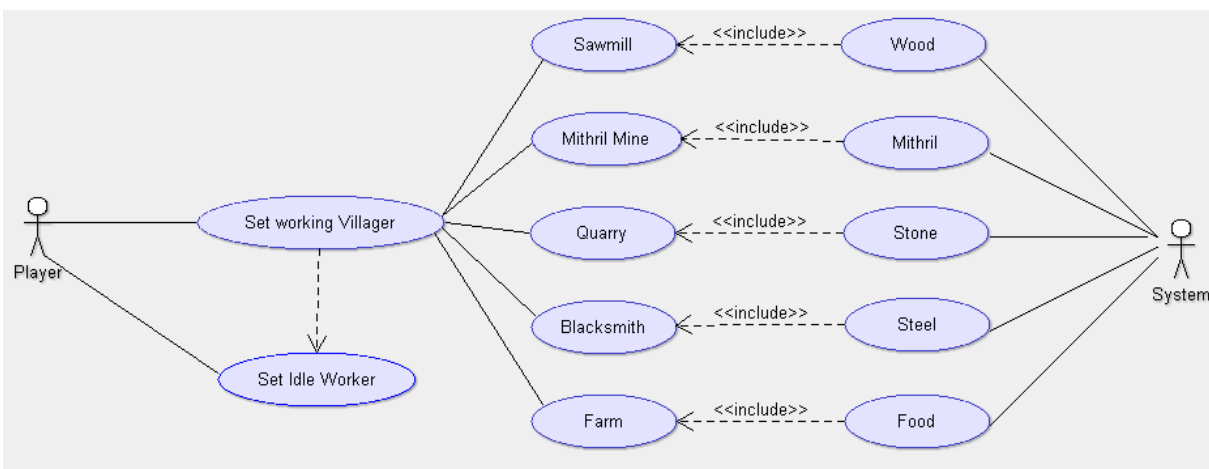


Ilustración 14 CU007 Investigaciones Por Raza.

CU008 Comprar Ejercito	
NOMBRE	Comprar Ejercito
ALIAS	Comprar Ejercito
ACTORES	Jugador
DESCRIPCIÓN	Este caso de uso permite comprar cuatro tipos de soldados para las 2 razas, los cuales son: Arqueros, Lanceros, Infantería y Caballería.
PRE-CONDICIÓN	El jugador deberá tener la edificación de Barracks y los recursos suficientes para adquirir el ejército deseado.
REFERENCIA	Requerimiento RF08
SECUENCIA NORMAL DE EVENTOS	1. El jugador ingresa a la web del juego. 2. El jugador ingresa el nickname y la clave para acceder al juego. 3. El jugador deberá acceder al cuartel donde entrenan los soldados, para comprar el ejército necesario con el fin de crecer militarmente. 4. El sistema validará si el jugador tiene los recursos suficientes para realizar la compra del ejército. 5. El sistema informará al usuario a través de mensajes si la compra del ejército fue realizada con éxito o si no es posible debido a la falta de

	recursos.
FLUJO ALTERNO	Ninguno.
POST CONDICIONES	Depende de la estrategia del jugador podrá interactuar con el ejercito comprado.

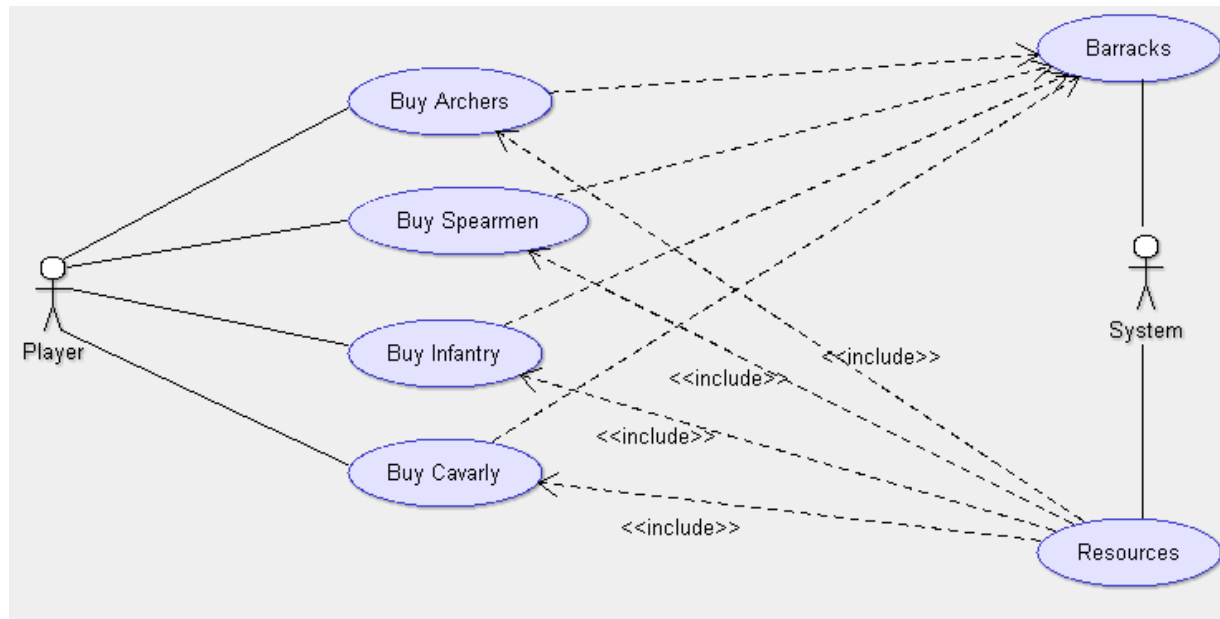


Ilustración 15 CU008 Comprar Ejercito.

CU009 Alianzas	
NOMBRE	Alianzas
ALIAS	Alianzas
ACTORES	Jugador
DESCRIPCIÓN	Este caso de uso permite crear, unirse, abandonar y declarar guerras a otras alianzas.
PRE-CONDICIÓN	Ninguna
REFERENCIA	Requerimiento RF09
SECUENCIA NORMAL DE	1. El jugador ingresa a la web del juego. 2. El jugador ingresa el nickname y la

EVENTOS	clave para acceder al juego. 3. El jugador deberá acceder a la opción de alianzas. 4. El jugador podrá crear una alianza, unirse a una ya establecida, dejar una alianza y dentro de ella realizar declarar guerras a otras alianzas. 5. El sistema permitirá la creación, unión, abandono o ataque que realice a través de la alianza. 6. El sistema le confirmará al jugador por medio de mensajes si la alianza fue creada satisfactoriamente, si se ha vinculado a una alianza, si ya no pertenece a la alianza, o se va a declarar una guerra a otra alianza.
FLUJO ALTERNO	Ninguno.
POST CONDICIONES	Las guerras entre alianzas se ganan dependiendo de los puntos obtenidos por los combates de sus integrantes.

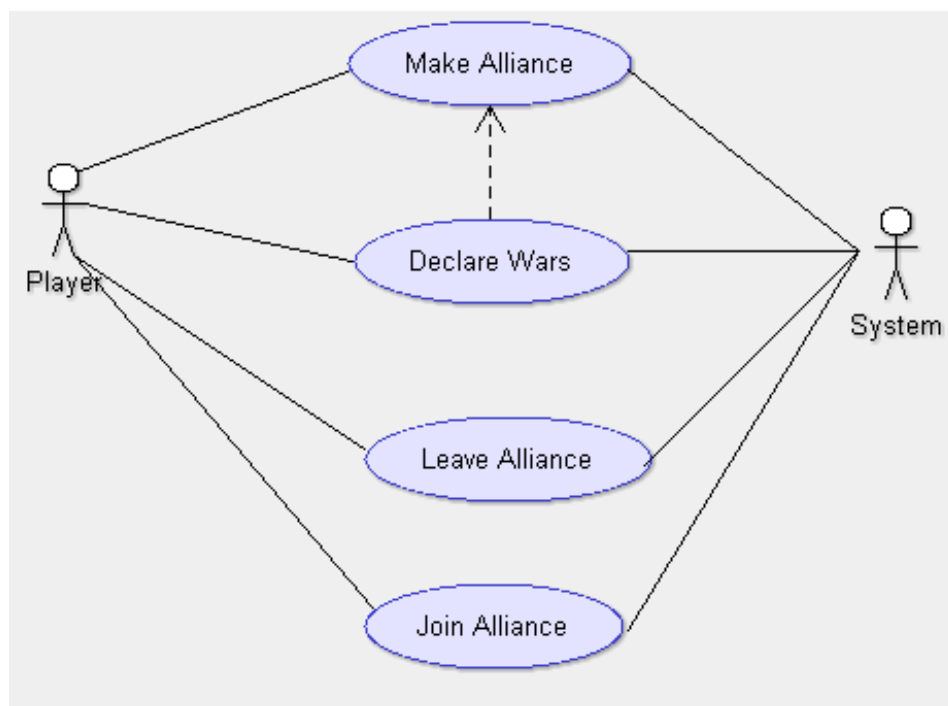


Ilustración 16 CU009 Alianzas.

CU010 Versus	
NOMBRE	Versus
ALIAS	Versus
ACTORES	Jugador
DESCRIPCIÓN	Este caso de uso permite realizar ataques a otros jugadores y defenderse por medio del ejercito y la fortaleza, adicional a esto el jugador podrá enviar un espía a su oponente, con el fin de conocer sus recursos y sus tropas enemigas.
PRE-CONDICIÓN	El jugador deberá tener el ejercito necesario para realizar un ataque y en caso contrario defenderse de sus oponentes, así mismo deberá tener una fortaleza y haya comprado la investigación de espionaje.
REFERENCIA	Requerimiento RF10
SECUENCIA NORMAL DE EVENTOS	1. El jugador ingresa a la web del juego. 2. El jugador ingresa el nickname y la clave para acceder al juego. 3. El jugador ingresa a la opción de jugadores. 4. El sistema lista los jugadores, la raza a la que pertenece, los puntos que tiene y la distancia en que se encuentra. 5. El jugador selecciona a quien desea atacar o enviar un espía antes de atacar para conocer sus tropas y recursos. 6. El sistema lista el ejercito con el que se cuenta y brinda la opción de escoger la cantidad de tropas que se van a enviar, de cada tipo de soldado. 7. El jugador podrá seleccionar el número de soldados que desea enviar para el ataque, así mismo podrá enviar un espía a otros jugadores. 8.
FLUJO ALTERNO	Ninguno.
POST CONDICIONES	El jugador podrá hacer uso de los recursos obtenidos a través de la

	victoria o si pierde no obtiene ningún tipo de recurso pero tampoco le es arrebatado.
--	---

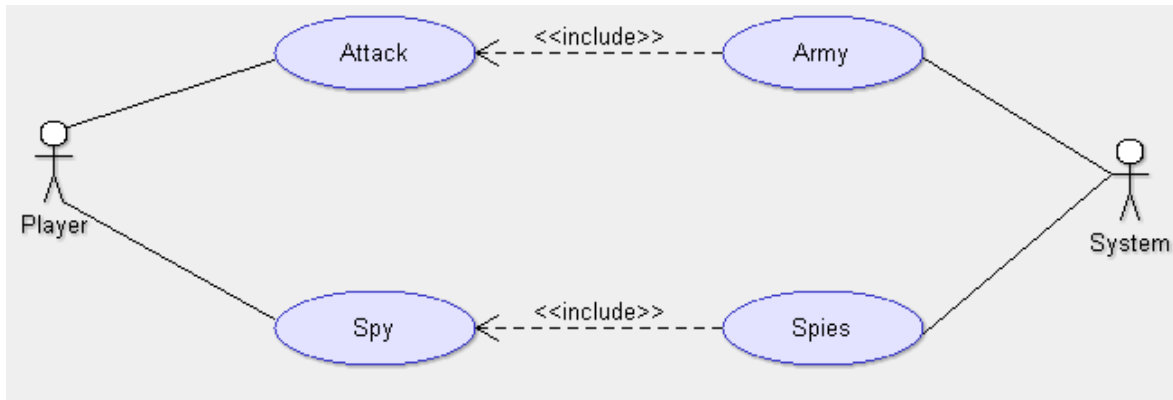


Ilustración 17 CU010 Versus.

CU0011 Mensajes	
NOMBRE	Mensajes
ALIAS	Mensajes
ACTORES	Jugador
DESCRIPCIÓN	Este caso de uso permite enviar y recibir mensajes entre jugadores.
PRE-CONDICIÓN	Ninguna
REFERENCIA	Requerimiento RF11
SECUENCIA NORMAL DE EVENTOS	1. El jugador ingresa a la web del juego. 2. El jugador ingresa el nickname y la clave para acceder al juego. 3. El jugador ingresa a la opción de enviar mensaje. 4. El sistema lista los jugadores del juego. 5. El jugador selecciona el nickname al cual desea enviar el mensaje. 6. El sistema informa al jugador que el mensaje fue enviado satisfactoriamente. 7. Si el jugador recibe un mensaje, el sistema mostrará una opción dentro del

	juego informando que tiene un mensaje nuevo. 8. El jugador podrá ver el remitente del mensaje, el contenido del mensaje y la fecha de envío.
FLUJO ALTERNO	Ninguno.
POST CONDICIONES	El jugador puede marcar los mensajes como leídos, adicional puede verlos de nuevo en el historial de mensajes.

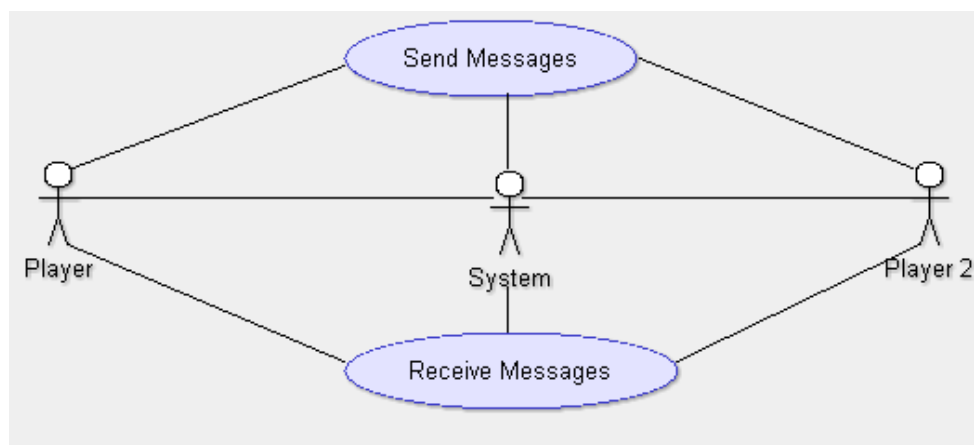


Ilustración 18 CU011 Mensajes.

CU0012 Actualizar Fortaleza	
NOMBRE	Actualizar Fortaleza
ALIAS	Actualizar Fortaleza
ACTORES	Jugador
DESCRIPCIÓN	Este caso de uso permite actualizar la fortaleza por cada raza, con el fin de aumentar el nivel de defensa.
PRE-CONDICIÓN	Ninguna
REFERENCIA	Requerimiento RF12
SECUENCIA NORMAL DE EVENTOS	1. El jugador ingresa a la web del juego. 2. El jugador ingresa el nickname y la clave para acceder al juego. 3. El jugador ingresa a la opción de

	fortaleza dentro del juego. 4. El sistema muestra el nivel de fortaleza que puede actualizar el jugador, dependiendo los recursos que tenga. 5. El jugador actualiza el nivel de fortaleza. 6. El sistema informa al jugador que el nivel de fortaleza fue actualizado satisfactoriamente.
FLUJO ALTERNO	El jugador podrá tener un nivel de defensa más alto para defenderse de otros jugadores, lo que permitirá una interacción mas intensa en el juego.
POST CONDICIONES	Ninguna

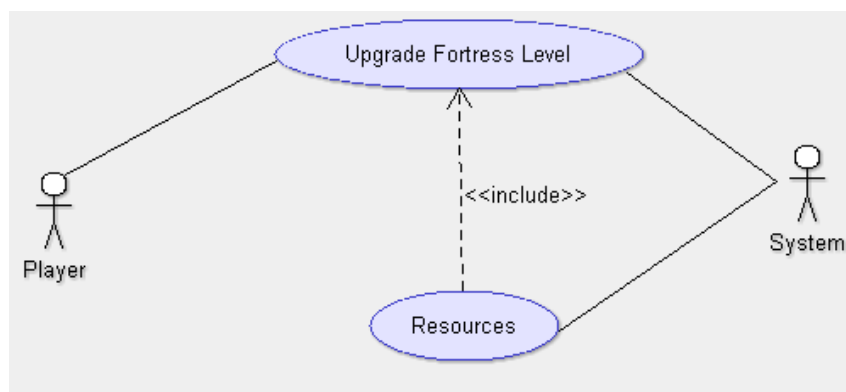


Ilustración 19 CU012 Actualizar Fortaleza.

9.1.4 DIAGRAMAS DE SECUENCIA

Dentro del siguiente diagrama de secuencia, se describe la validación de usuario y contraseña que realiza el jugador para acceder a la aplicación, tal como lo muestra la ilustración 20.

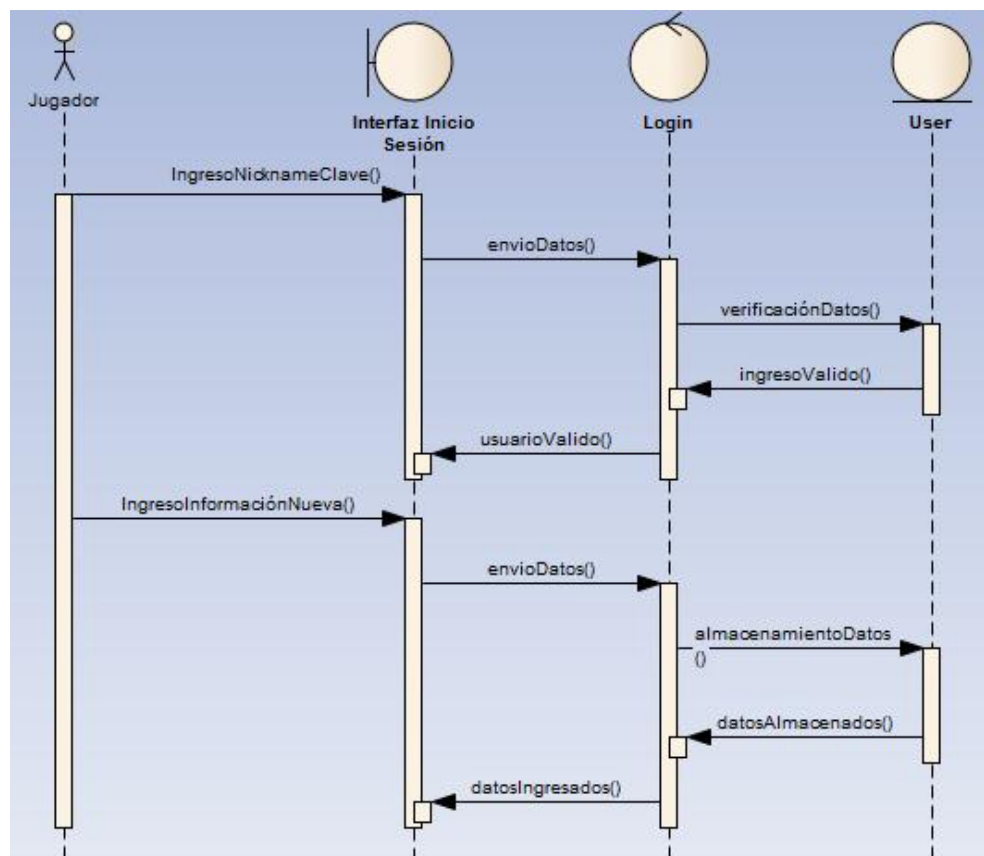


Ilustración 20 Ingreso y Registro del Jugador.

En la ilustración 21, se observa como el jugador puede acceder a la compra de las de las edificaciones, así mismo el sistema se encarga de realizar ciertas validaciones sobre los recursos que el jugador posee y da como retorna si es viable o no realizar la compra de la edificación seleccionada.

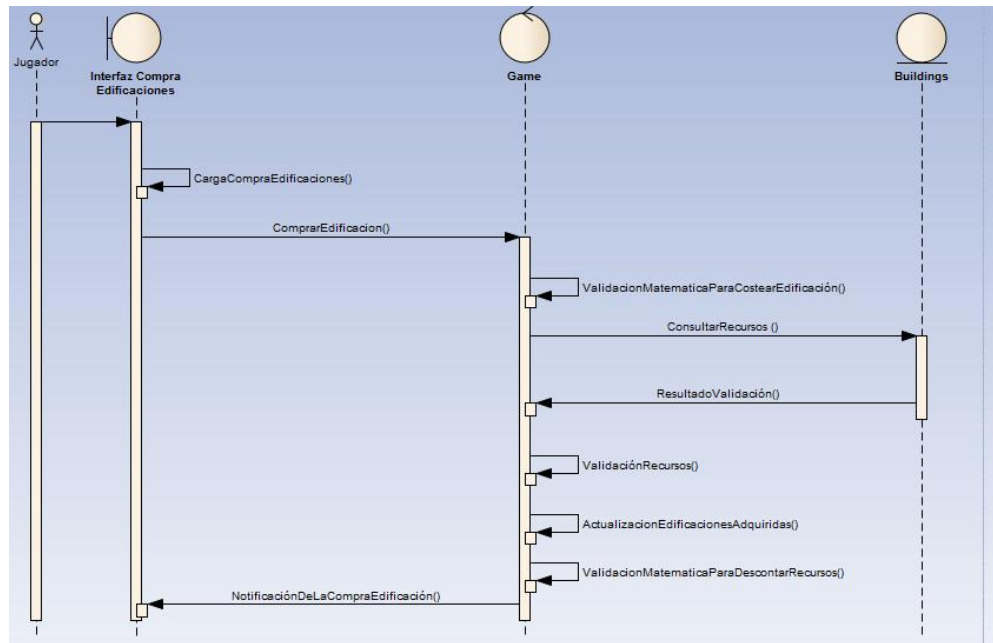


Ilustración 21 Compra de Edificaciones.

El siguiente diagrama describe como el jugador interactúa a través del juego para realizar la compra de aldeanos, así mismo el sistema valida la compra de aldeanos. Tal como se observa a detalle en la ilustración 22.

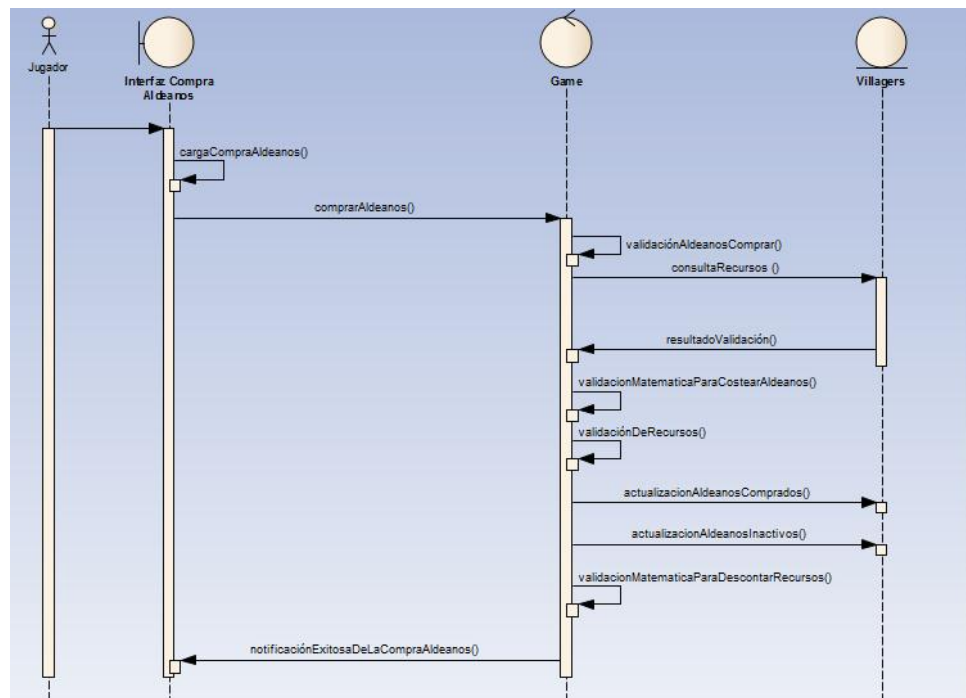


Ilustración 22 Compra de Aldeanos.

A continuación se observa la ilustración 23, la cual describe como el jugador asigna aldeanos a las edificaciones para la recolección de recursos, de igual forma el sistema se encarga de validar las peticiones requeridas por el jugador.

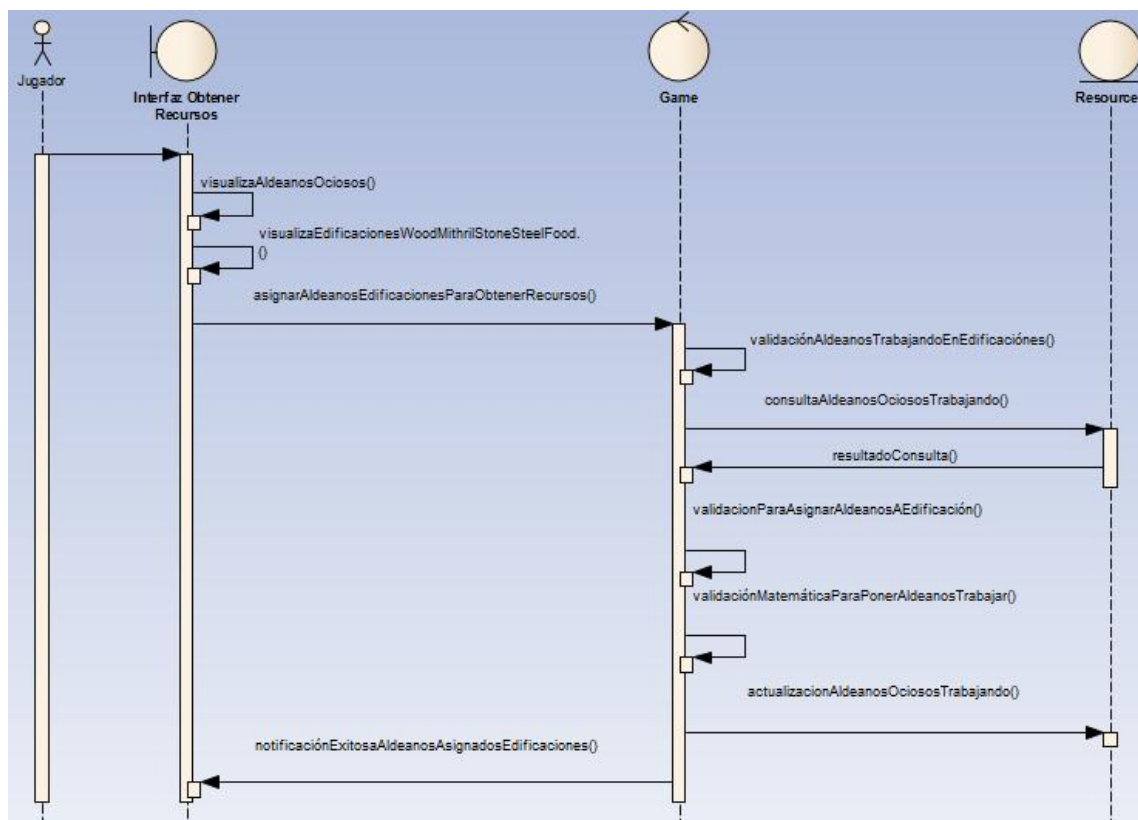


Ilustración 23 Obtener Recursos.

En la ilustración 24, se describe como dentro del entorno del juego, el jugador solicita la investigación de acuerdo a la raza a la que pertenece, lo cual el sistema valida si los recursos con los que cuenta son suficientes para acceder a la investigación.

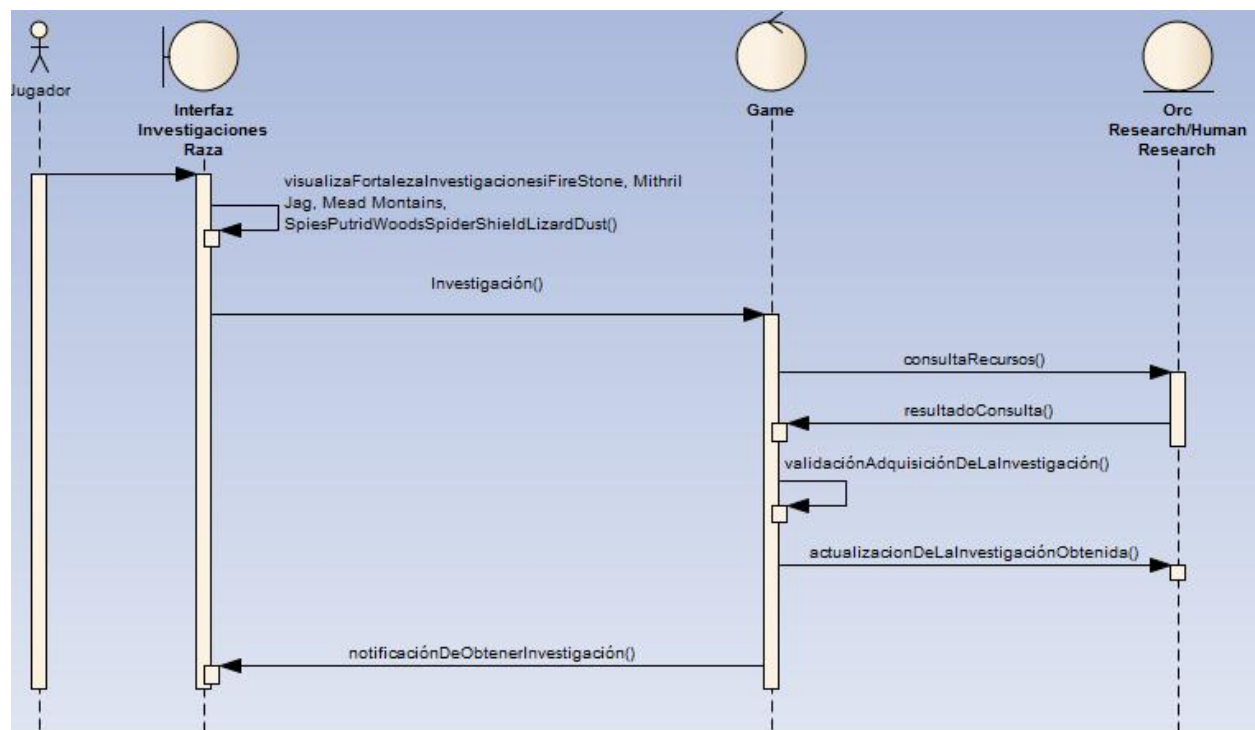


Ilustración 24 Investigaciones Por Raza.

A continuación se describe como el jugador solicita la compra de ejército dentro del entorno o ambiente del juego, de igual manera el sistema se encarga de validar si el jugador posee los recursos suficientes para realizar la compra del ejército, tal como se observa en la ilustración 25.

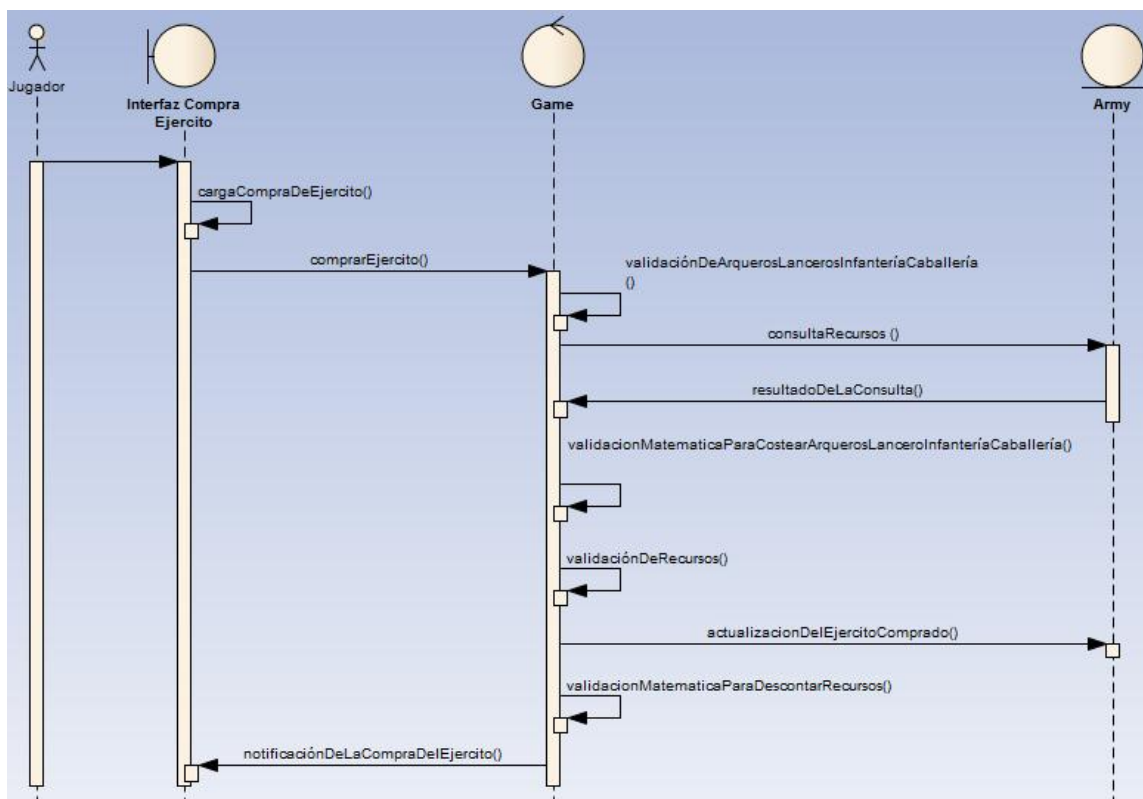


Ilustración 25 Compra Ejército.

Para el siguiente diagrama, se observa como dentro de la interfaz de alianzas, el jugador puede seleccionar las opciones de crear una alianza, unirse a una alianza, abandonar una alianza o atacar a otra alianza, de igual manera el sistema realiza la validación sobre la opción que el jugador allá seleccionado, tal como se aprecia en la ilustración 26.

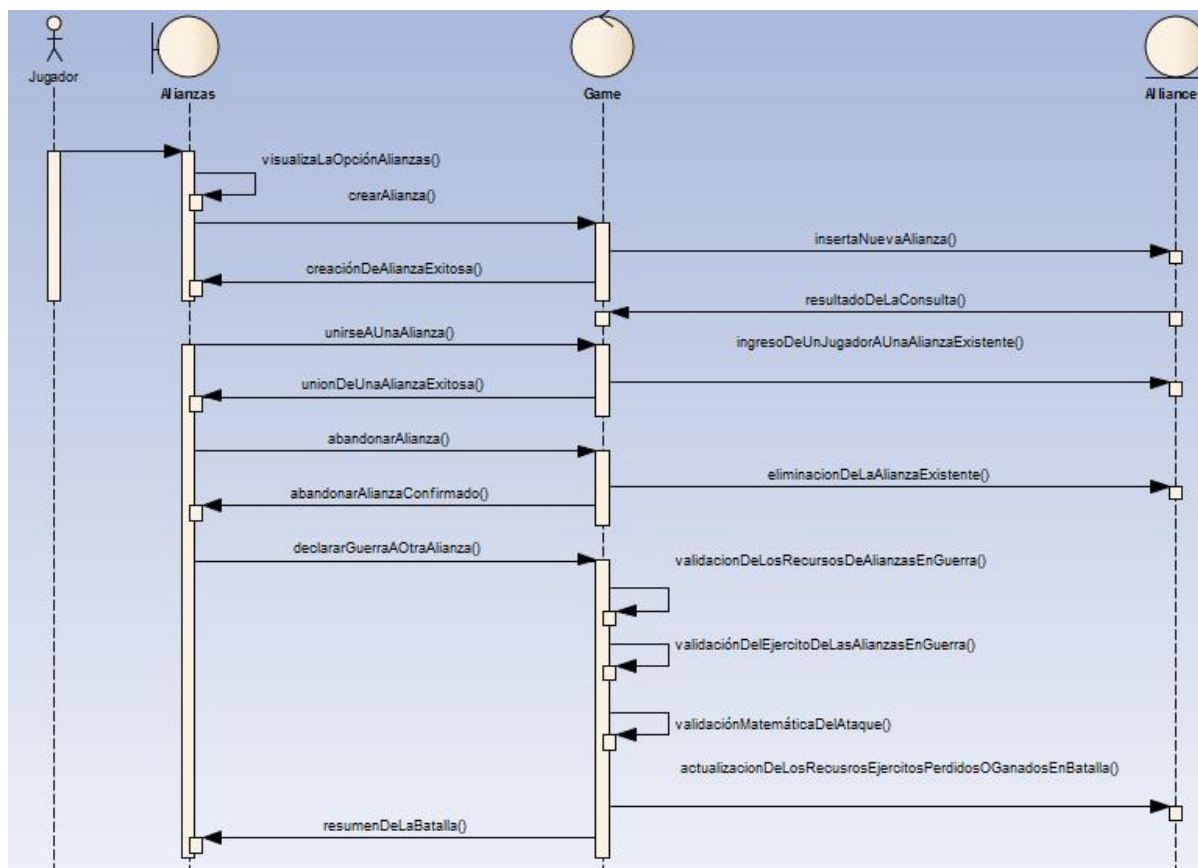


Ilustración 26 Alianzas.

En la ilustración 27, se aprecia como el jugador realiza la petición de enviar un ataque a otro jugador, el sistema se encarga de realizar la operación correspondiente para llevar a cabo el ataque entre jugadores.

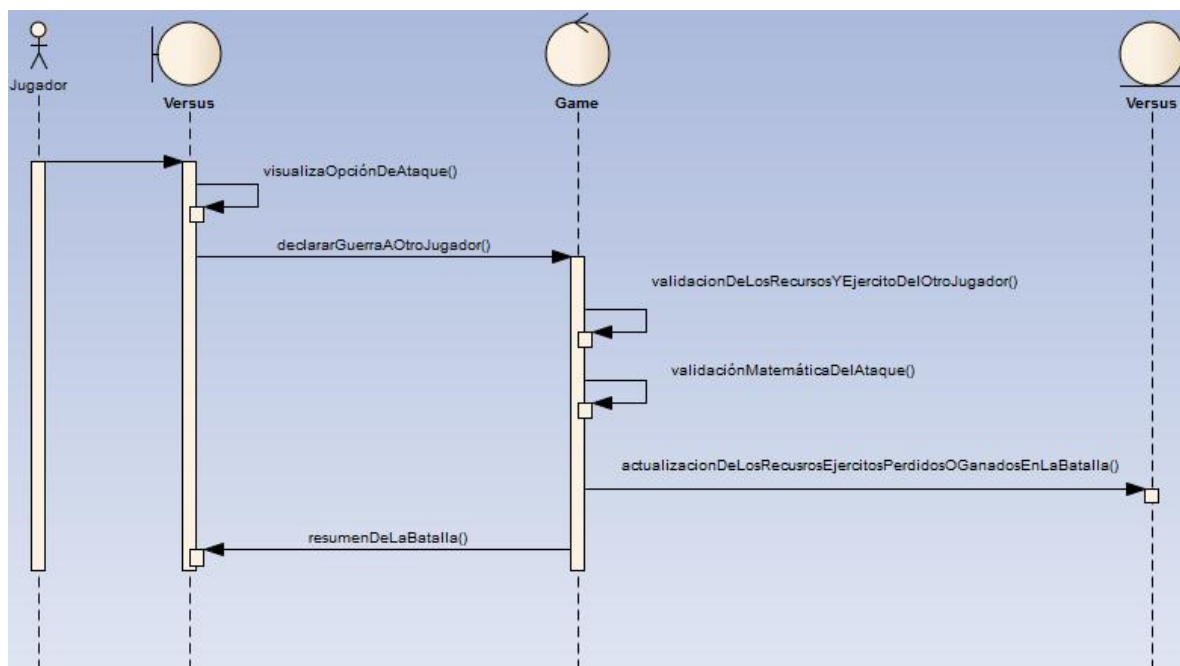


Ilustración 27 Versus.

En el siguiente diagrama ilustra como un jugador solicita al sistema enviar un mensaje a otro jugador, de igual forma el sistema se encarga de realizar la petición del jugador y re direccionar el mensaje al jugador correspondiente, tal como se muestra en la ilustración 28.

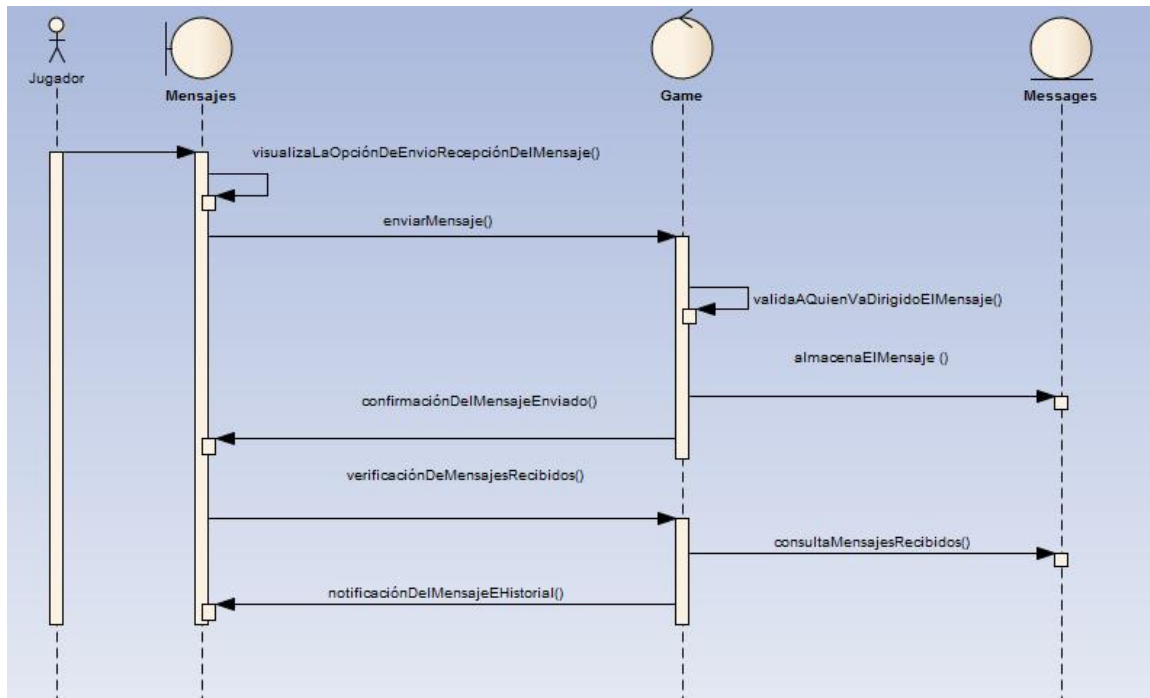


Ilustración 28 Mensajes.

En la ilustración 29 se puede observar como el jugador solicita al sistema actualizar su fortaleza lo cual le permitirá mantener una mejor defensa frente al atacante, así mismo el sistema se encarga de validar si es posible incrementar el nivel de defensa de acuerdo a los recursos con que cuente el jugador.

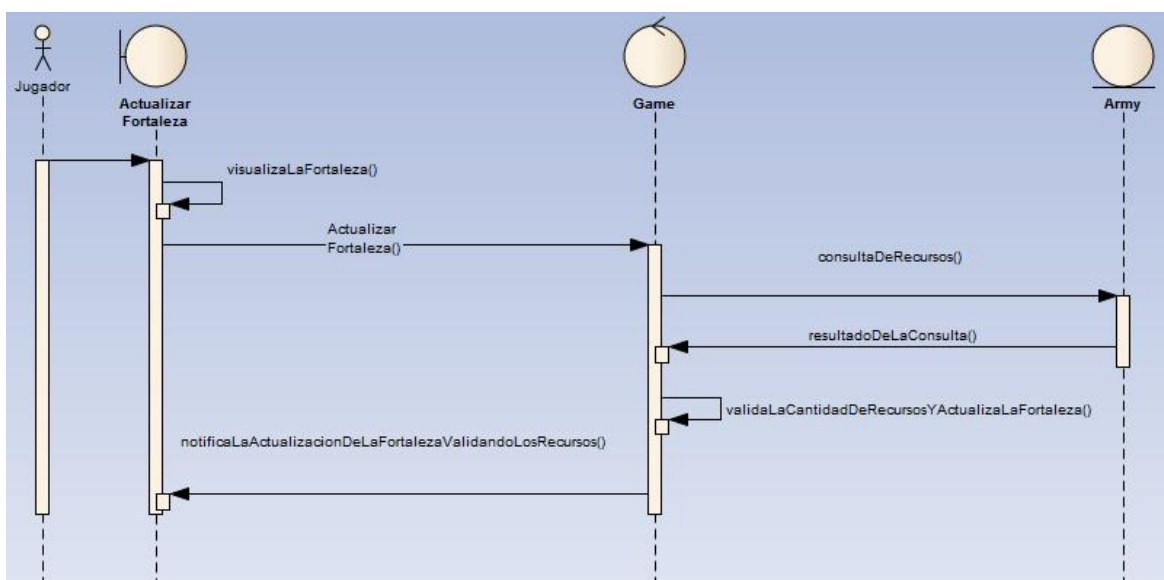


Ilustración 29 Actualizar Fortaleza.

9.1.5 DIAGRAMA DE CLASES

Previamente al realizar los diagramas de casos de uso se logra establecer un mejor análisis para elaborar el diagrama de clases, el cual describe el MMORPG de forma general, de esta manera se evidencian los atributos y métodos que conllevan a la funcionalidad de la aplicación, tal como lo muestra la ilustración 30.

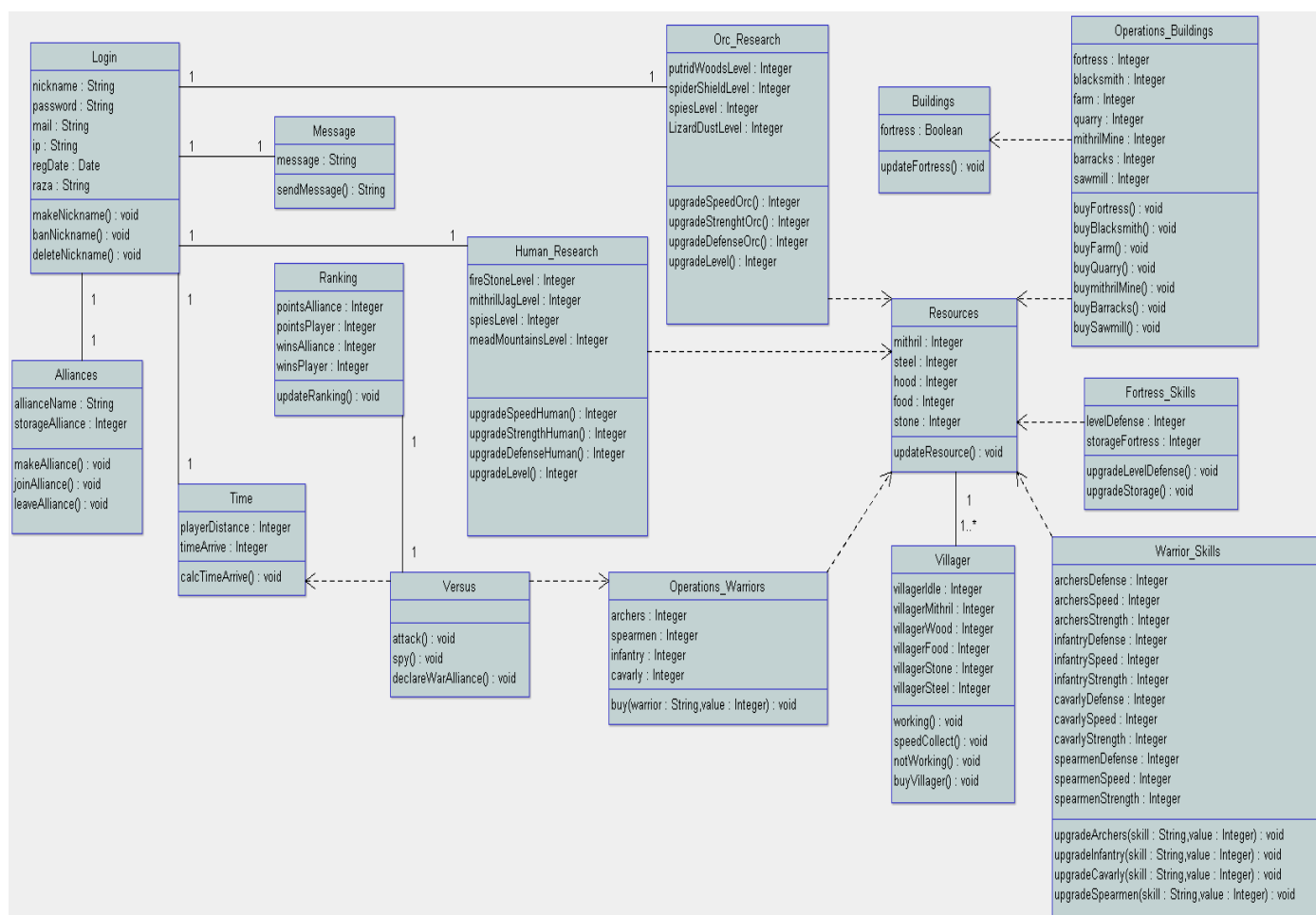


Ilustración 30 Diagrama de Clases.

9.1.6 MODELO ENTIDAD RELACION

Por medio de la construcción de los casos de uso, de los diagramas de secuencia y del diagrama de clases, es posible realizar el desarrollo del modelo entidad relación, el cual es parte fundamental del software creado para este proyecto. La siguiente ilustración 31 permite observar el modelo de la base de datos realizada para el juego.

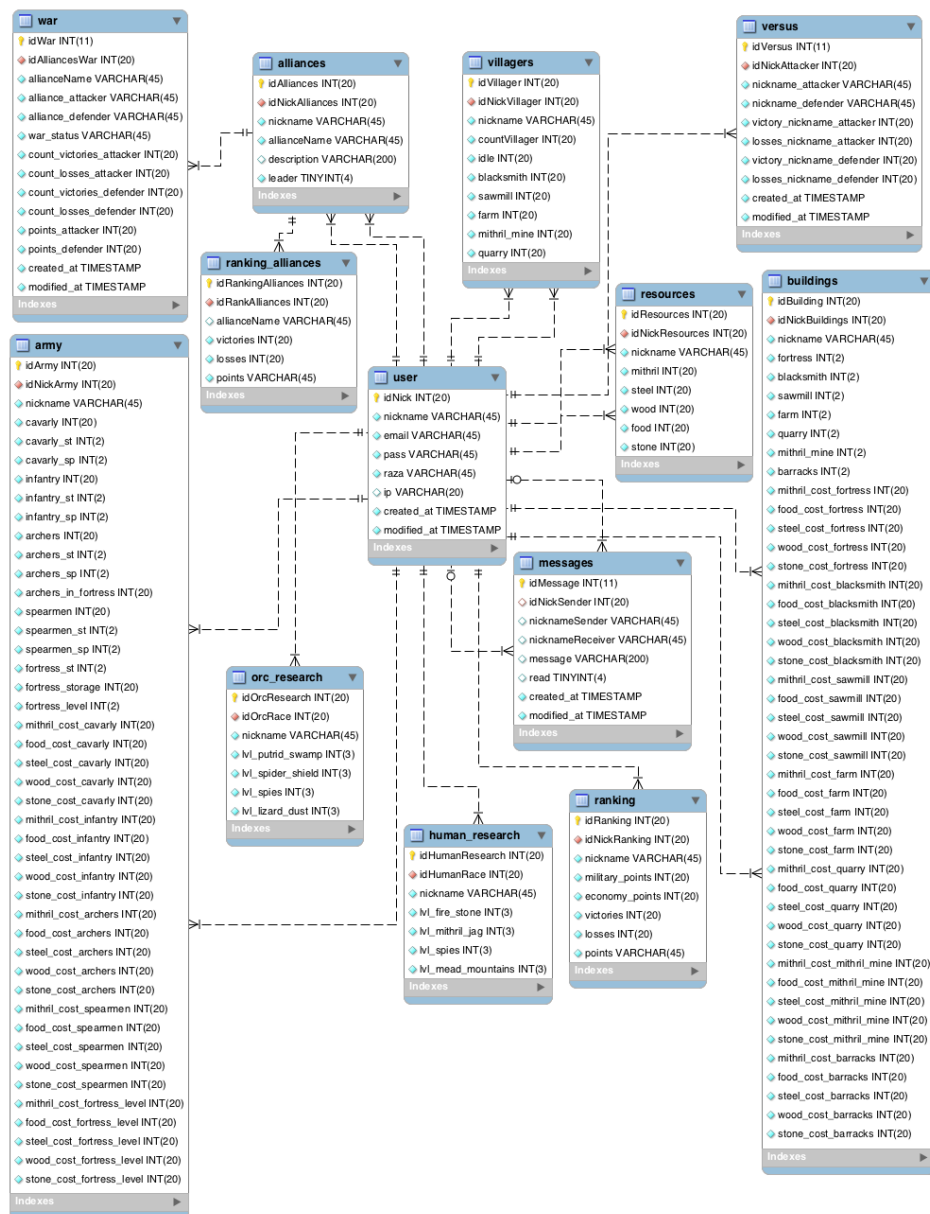


Ilustración 31 Modelo Entidad Relación.

10. IMPLEMENTACIÓN DE LA SOLUCIÓN

10.1 Diseño del MMORPG y plataforma

Los componentes estéticos de un videojuego conllevan a expresar diferentes sensaciones enfocadas al realismo del entorno o ambiente en que este diseñado el juego, ya sea en un contexto épico, futurista, espacial, fantasía entre otros. La estética de un videojuego nos permite identificar e imaginar cómo será la dinámica o narración del juego.

Para un MMORPG no es estrictamente necesaria la animación en D2 y 3D dentro de la estética del juego, ya que lo que realmente interesa es la dinámica o mecánica que permita al jugador manejar ciertas estrategias y usar su imaginación como parte del juego. Sin embargo dentro del ambiente de un juego de rol online MMOPRG de tipo browser game es importante tener en cuenta que las gráficas o ilustraciones vayan de acuerdo con la ambientación del juego, pues de esta manera el jugador tendrá un espacio de realismo y lúdica más profundo al interactuar con el mundo del juego.

Para este proyecto, utilizamos un set de ilustraciones a la medida, las cuales fueron realizadas por un ilustrador externo, donde cada gráfica describe situaciones o personajes que estarán relacionadas dentro del ambiente o entorno del juego.

Algunas de los bocetos e ilustraciones finales, que componen el juego de rol son las siguientes:

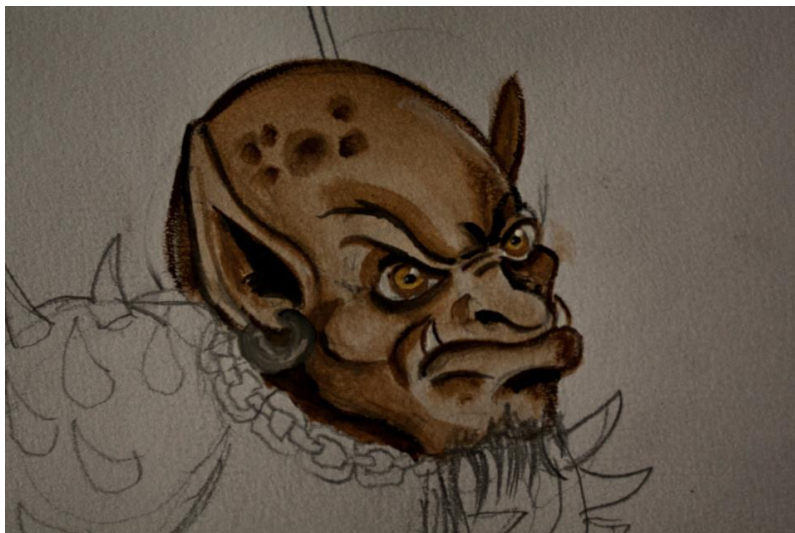


Ilustración 32 Boceto Orco Lancero¹⁷.



Ilustración 33 Orco Lancero¹⁷.

¹⁷

Ilustradora, Luz Stella Jaime Cornejo Diciembre 2012

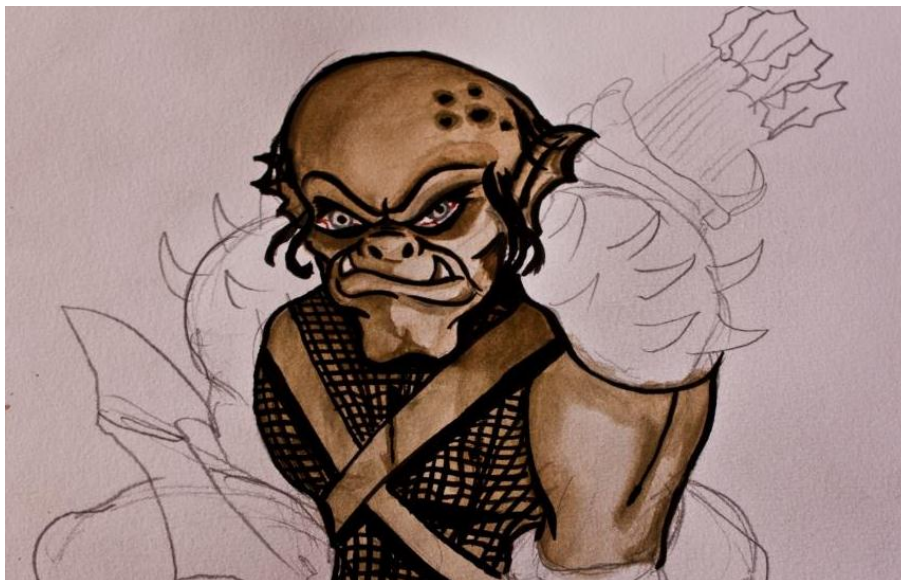


Ilustración 34 Boceto Orco Arquero¹⁸.



Ilustración 35 Orco Arquero¹⁸

¹⁸

Ilustradora, Luz Stella Jaime Cornejo Diciembre 2012



Ilustración 36 Orco en Combate¹⁹



Ilustración 37 Orco Hacha¹⁹

¹⁹

Ilustradora, Luz Stella Jaime Cornejo Diciembre 2012

La Plataforma utilizada para un MMORPG de tipo browser-game no requiere de un estándar específico, sin embargo al desarrollar un MMORPG con animación 2D y 3D se puede lograr bajo plataformas como Unity 3D, Unreal Engine y CryEngine, los cuales son los motores estándar de última generación, que presentan características robustas soportando animación dinámica sobre un videojuego; la mayoría de estos motores son de pago, aun así se encuentran versiones de prueba gratuitas para su práctica sobre el desarrollo de un videojuego.

Sin embargo para el desarrollo de un MMORPG que no requiere animación y que su funcionalidad e interacción es basada en estadísticas e ilustraciones, tal como lo es en este proyecto, es posible desarrollarlo bajo un lenguaje de programación orientado a web, para nuestro caso fue desarrollado bajo el lenguaje de programación Ruby, específicamente utilizando el Framework de desarrollo web Ruby on Rails.

10.2 Story Game del Juego

El story game de un videojuego es una de las partes fundamentales para que el jugador tenga interés y gusto por este, pues le permitirá al jugador tener una experiencia lúdica a través de los personajes, eventos y entornos que contenga el juego, aun así, es importante mantener un enfoque central sobre la idea o tema que lo compone, adaptando al juego cada detalle que permita que la historia del videojuego sea sólida y coherente con la ambientación que se le brinde, pues dentro de la evolución y mundo de los videojuegos, es posible encontrar aquellos que contienen una gran historia, pero que a medida que el jugador va avanzando se pierde el tema central y el jugador termina por abandonar el juego, como también se puede observar situaciones sobre las cuales el entorno del juego son muy bien elaboradas como el sonido, las gráficas, animación y demás factores que lo componen, pero la historia del videojuego no es lo suficientemente interesante para el jugador, por tal razón, es importante mantener diferentes aspectos en la construcción de la historia de un videojuego, que permitan al jugador estar comprometido e interesado desde el inicio del juego hasta el final de este.

El story game del MMORPG que hemos establecido para este proyecto, se desenvuelve en un mundo de fantasía llamado gaia, donde coexisten 2 razas que disputan el dominio militar de gaia, las cuales son orcos y humanos, cada una de estas razas tiene características únicas, descritas de la siguiente manera:

- Raza Orco: Es una raza que solo vive para combatir y destrozar todo lo que este a su alcance, su propósito es invadir territorios y tener el control de toda gaia, los orcos tienen una fisionomía particular, su cuerpo es mediano pero resistente a golpes, poseen colmillos sobresalientes sobre su rostro lo que los hace ver de una forma más agresiva en sus ataques, son poco inteligentes pero muy agiles en la recolección de ciertos recursos como la madera, lo que les permite construir casuchas y sus fortalezas mucho más rápido, con el fin de mantener su defensa, de igual forma son letales espías logrando tener ventaja para conocer los recursos y tropas de su enemigo, la raza humana.
- Raza Humana: Son seres inteligentes, astutos y nobles, su objetivo es desterrar a los orcos de las tierras de gaia, son excelentes trabajadores de la piedra, permitiendo que sus muros sean más resistentes contra el asedio, debido a que su fisiología no está adaptada para las batallas, utilizan entrenamiento exigente por lo que son grandes jinetes.

Cada raza puede recolectar recursos como lo son:

- Mithril: Es un mineral precioso y valioso, es considerado como la fuente principal de intercambio económico para la tierra de gaia, para los humanos es visto como símbolo de tranquilidad y paz, mientras que para los orcos el mithril es símbolo de poder y corrupción.
- Madera: Es un recurso importante en toda la tierra de gaia, ya que permite la construcción de viviendas y fortalezas de cada raza.
- Piedra: Este recurso permite que las construcciones sean más fuertes.
- Acero: El acero permite forjar las armas que serán utilizadas en batalla por cada raza.

- Comida: Es uno de los recursos fundamentales, ya que sustenta a los aldeanos para la recolección del resto de recursos.

Cada una de las razas cuenta con un ejército, para realizar ataques, los cuales son:

- Arqueros: Este tipo de guerrero posee una gran habilidad con el arco.
- Lanceros: Los guerreros lanceros se caracterizan por su agilidad al momento de atacar la caballería durante la batalla, una de las ventajas de los lanceros orcos es que son más económicos al momento de adquirirlos.
- Infantería: La infantería son guerreros ligeros pero resistentes en el ataque, debido al acero reforzado con Mithril de sus espadas.
- Caballería: La caballería para los humanos hace una gran diferencia frente a los jinetes de wargos de los orcos, ya que son más rápidos y ágiles, dándoles ventaja militar.

El story game de Gaia Conquest está ambientado con el fin de que el jugador utilice las características correspondientes de la raza seleccionada, de esta manera podrá establecer su propia estrategia de recolección de recursos, compra de ejército y envío de ataques a otros jugadores, sin manejar ningún tipo de regla específica para la toma de sus decisiones.

10.3 Mecánica del Juego

La mecánica de un videojuego, describe el comportamiento y funcionalidad de este, partiendo de como el jugador va a comprender e interactuar con el juego, para ello se establece en varias ocasiones reglas o condiciones que le permiten al jugador divertirse e iniciar una experiencia única, personal e independiente dentro del juego en el que se encuentre. La mecánica de un videojuego manifiesta ciertas propiedades que permiten caracterizar la experiencia de un jugador, como por ejemplo la satisfacción o agrado que el juego le ofrece al usuario, reuniendo la ambientación gráfica, el sonido, la historia y demás componentes que hacen que un juego cumplan con

lo que espera el jugador. Así mismo existen otras características o propiedades como la emoción y socialización que genera la interacción de un juego, pues a través de estas se entablan relaciones sociales entre jugadores, que conllevan a interactuar de una manera cooperativa o competitiva entre ellos.

Para este proyecto se define la siguiente mecánica del MMORPG:

A través de la web del juego, el usuario podrá registrarse y seleccionar un tipo de raza, una vez se encuentre en sesión, el jugador tendrá por defecto cierta cantidad de recursos, que le permitirán analizar el tipo de estrategia que utilizará para la recolección de recursos, compra de ejército, realizar ataques entre otras opciones.

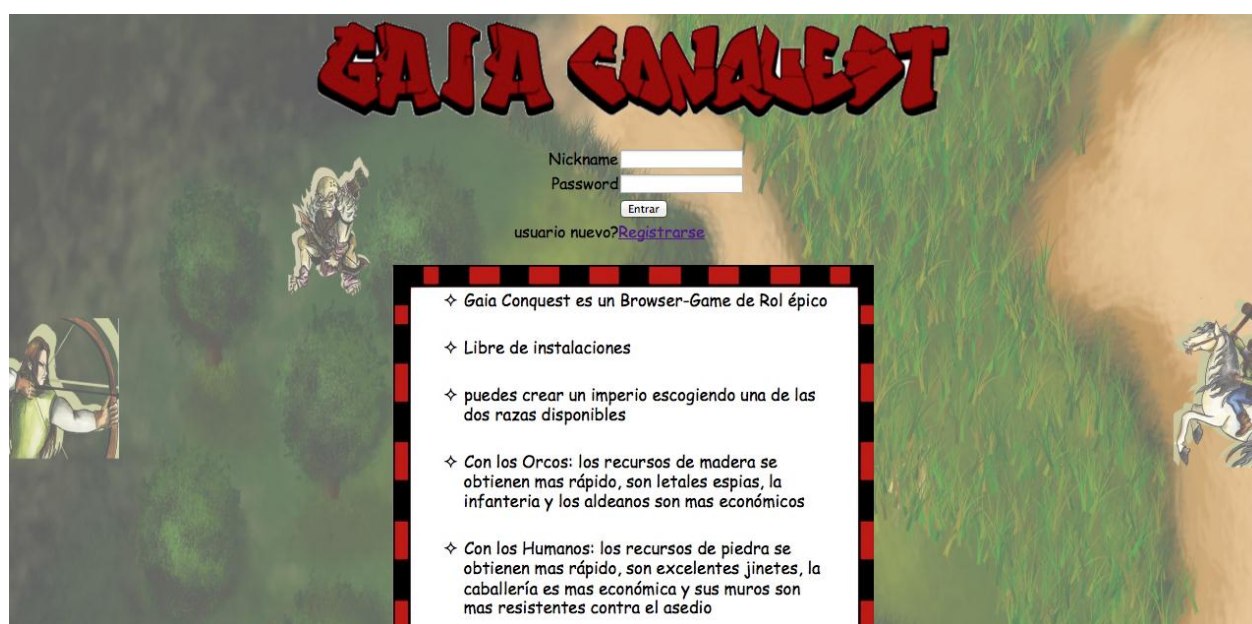


Ilustración 38 Autenticación de Gaia Conquest

En la figura 38 se ilustra la vista principal de la aplicación y de la cual se desprenden las demás vistas, ésta vista se compone de la imagen central del territorio que controla el jugador y un panel con el cual puede interactuar para ejecutar las diferentes acciones del juego, el panel se compone de los recursos del jugador como lo son mithril, alimento, madera, piedra y acero, el icono de aldeanos totales y ociosos, el mapa de jugadores, el icono de envió de mensajes, el icono de historial de mensajes, el icono de alianzas, el icono de recepción de nuevos mensajes y el icono de ranking, los cuales contienen las acciones

correspondientes a sus nombres, en posiciones aleatorias del mapa central están ubicados los edificios civiles y militares.



Ilustración 39 Provincia y vista principal del juego



Ilustración 40 Guía

El jugador tendrá la posibilidad de comprar aldeanos, edificaciones para la explotación de recursos, tales como: Mina de Mithril, Herrería, Aserradero, Granja y Cantera de Piedra, dentro del centro urbano, de igual manera el jugador podrá comprar un cuartel para entrenar sus tropas, las cuales son: Arqueros, Caballería, Infantería y Lanceros así, mismo podrá acceder a una fortaleza para tener una mejor defensa de los ataques de otros jugadores. Estas edificaciones y aldeanos solo podrán ser adquiridas dependiendo de la cantidad de recursos con que cuente el jugador.



Ilustración 41 Compra de Aldeanos y Edificios.

Una vez el jugador haya comprado las edificaciones para la recolección de recursos, podrá poner a trabajar a los aldeanos dentro de cada una de las minas, de igual forma se podrá observar dentro del juego, el número de recursos que se van recolectando por los aldeanos. Así mismo el jugador tendrá la opción de comprar su ejército dentro del cuartel adquirido previamente. El jugador podrá enviar arqueros a la fortaleza con el fin de defenderse de sus tropas enemigas.



Ilustración 42 Asignar Aldeanos a La Mina Mithril.

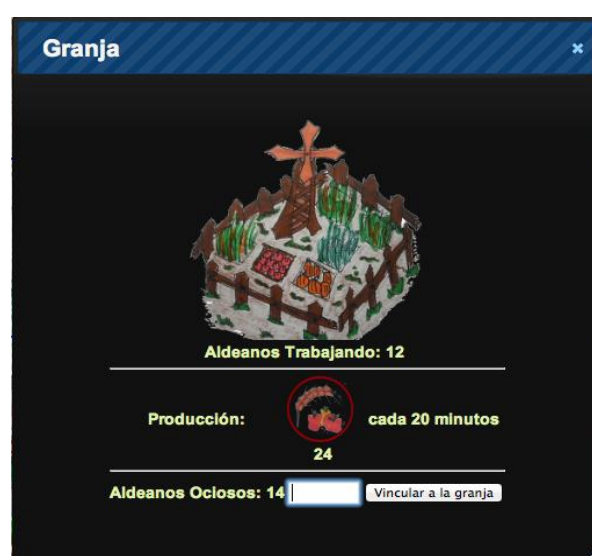


Ilustración 43 Asignar Aldeanos a La Granja.



Ilustración 44 Cantera de Piedra.



Ilustración 45 Aserradero.

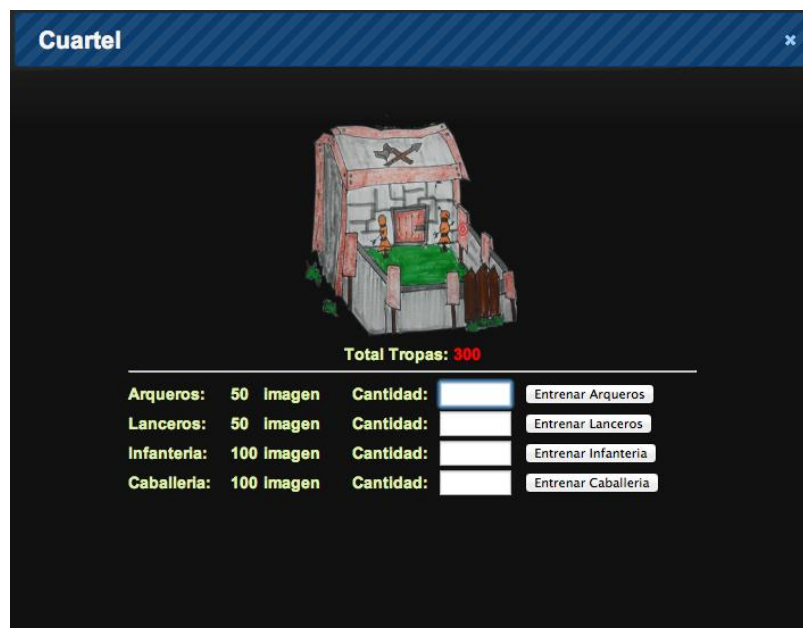


Ilustración 46 Cuartel

El jugador podrá establecer su estrategia de ataque frente a otros jugadores, esto dependerá de la cantidad de ejercito con que desee atacar o del tipo de guerrero que desee enviar a la batalla, pues es posible que dependiendo de la estrategia del jugador quiera realizar ataques desde el inicio del juego, o establecer una estrategia más prudente con mayor número de

recursos para obtener más ejército, esto es solo disposición del jugador, ya que el impone las decisiones propias de su juego.

Jugadores Gaia -- Realm1 --					
Jugador:	Distancia:	Raza:	Puntos:	Espiar:	Atacar:
dragon	58	Humano	0		
leia	59	Humano	0		
kaiser	60	Humano	0		
caraverde	61	Orco	0		
sword	62	Orco	0		
destroyer	63	Orco	0		
uruk	64	Orco	0		
cuerno_del_este	65	Orco	0		

Ilustración 47 Selección de Ataque.

El jugador podrá establecer comunicación con otros jugadores por medio de mensajes, ya sea en el momento en que envíe un ataque, o simplemente para conocer otros jugadores con el fin de crear alianzas, la cual es una opción del juego, donde el jugador podrá crear, unirse, eliminar una alianza y realizar ataques entre alianzas, esto le permitirá al jugador consolidar un equipo dentro del juego y estar dentro del listado de ranking de alianzas.

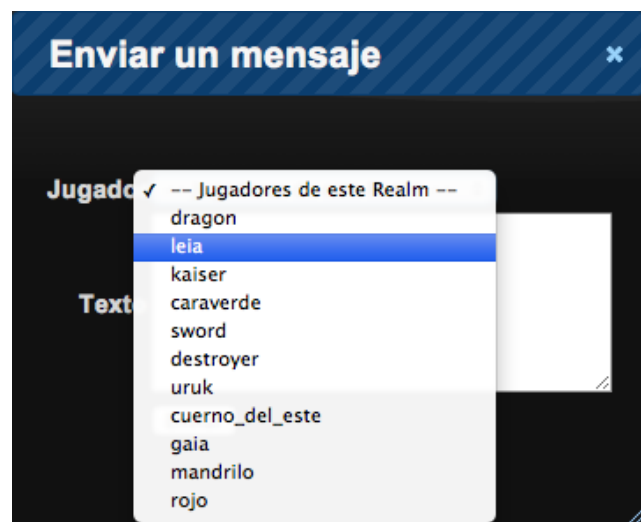
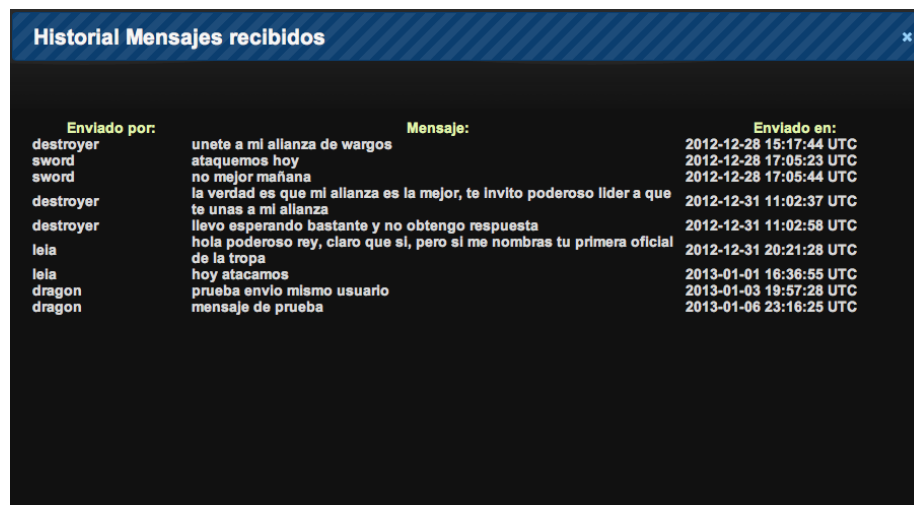


Ilustración 48 Enviar Mensaje

Dentro de Gaia Conquest, el jugador tendrá acceso al historial de mensajes que haya recibido por parte de otros jugadores, adicional a esto podrá observar el número de aldeanos que se encuentran ociosos y aquellos que se encuentran trabajando.



Enviado por:	Mensaje:	Enviado en:
destroyer	unete a mi alianza de wargos	2012-12-28 15:17:44 UTC
sword	ataquemos hoy	2012-12-28 17:05:23 UTC
sword	no mejor mañana	2012-12-28 17:05:44 UTC
destroyer	la verdad es que mi alianza es la mejor, te invito poderoso lider a que te unas a mi alianza	2012-12-31 11:02:37 UTC
destroyer	llevo esperando bastante y no obtengo respuesta	2012-12-31 11:02:58 UTC
leia	hola poderoso rey, claro que si, pero si me nombras tu primera oficial de la tropa	2012-12-31 20:21:28 UTC
leia	hoy atacamos	2013-01-01 16:36:55 UTC
dragon	prueba envlo mismo usuario	2013-01-03 19:57:28 UTC
dragon	mensaje de prueba	2013-01-06 23:16:25 UTC

Ilustración 49 Historial de Mensajes

10.4 Interacción para los Jugadores

Un videojuego es propicio para generar entretenimiento e interacción mucho más fácil, debido a que los jugadores buscan un espacio para el entretenimiento y salir de la cotidianidad, el MMORPG genera intereses en común, por lo cual se pueden establecer una comunicación más cercana y amena entre los jugadores, en donde tienen un punto de partida para interactuar y un tema central.

El MMORPG Gaia Conquest propone los siguientes métodos de interacción: Sistema de Ataques, Sistema de Mensajes, Sistema de Alianzas y Guerras entre Alianzas.

10.5 Sistemas de Ataques

El sistema de ataque para el MMORPG Gaia Conquest, está basado en una operación matemática discriminada de la siguiente manera:

Para las batallas entre jugadores se tiene en cuenta los valores de fuerza y velocidad de cada tipo de unidad militar, partiendo del valor 1 en el nivel 1. Sumando a un valor por bonificación de un tipo de soldado sobre otro así:

- Lanceros de ambas razas tienen bonificación de +6 sobre la caballería humana y sobre los jinetes de wargos orcos.
- Infantería a espada de ambas razas tienen bonificación de +5 sobre los arqueros y lanceros de ambas razas.
- Los arqueros no poseen bonificación pero son el asedio perfecto para la fortaleza enemiga, son las únicas unidades que pueden atacar la fortaleza, el primer bastión defensivo de los imperios.
- Caballería de ambas razas tienen bonificación de +6 sobre la Infantería a espada y arqueros de ambas razas.

Los arqueros guarnecidos en fortaleza tienen una bonificación en defensa de +4, en el nivel 1 de fortaleza solo se pueden guarnecer 50 unidades de arqueros.

El resultado de una batalla se resuelve dando la ventaja tomada como factor sorpresa al atacante, donde hay 5 rondas, en cada ronda una facción del ejercito ataca primero a la facción del ejercito enemigo sobre la cual tiene bonificación, de esta forma si el atacante posee lanceros, estos atacaran primero a los caballos enemigos diezmándolos, igualmente, con el resto de facciones del atacante ocurre lo mismo.

Luego de calcular los ejércitos restantes de cada bando, en cada ronda se calcula el vencedor restando las tropas de cada facción y sumando un porcentaje de recuperación de tropas del 40% sobre las tropas perdidas del vencedor en puntos, y un 15% de recuperación de tropas sobre las tropas perdidas del derrotado en dicha ronda, así para todas las rondas, al final, se suman todas las tropas restantes de cada bando y se obtiene un puntaje basado en las tropas caídas del enemigo, el puntaje mayor gana la batalla y se actualiza la base de datos con el acumulado de puntos militares, acumulado de puntos totales incluyendo los puntos civiles y el registro de la batalla con las cantidades iniciales y finales de tropas de cada bando. Luego de la batalla hay un botín de guerra para el ganador basado en +2 de mithril por cada soldado enemigo caído, este botín se acumula al mithril total del ganador de la batalla.

El objetivo final del juego es ser el número 1 en el ranking general tanto en puntos militares y cantidad de victorias como en puntos basados en la economía, este sistema de batalla aplica también para las guerras entre alianzas, donde además se incluye en la base la alianza ganadora y los puntos de alianza que son la sumatoria de los puntos obtenidos por las victorias individuales de sus integrantes, esto lleva a un ranking general de alianzas.

Terminando el realm existirá un ganador individual y una alianza ganadora por cantidad de guerras ganadas.

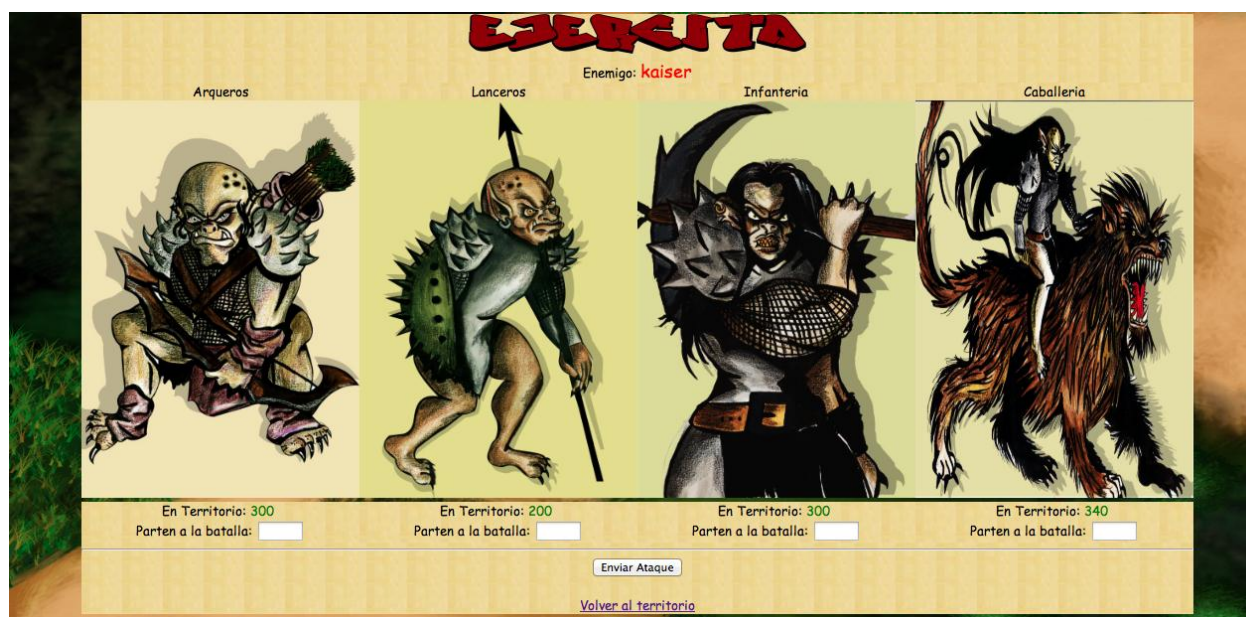


Ilustración 50 Ataque



Ilustración 51 Enviando Ataque

En la ilustración 49 se aprecia el resultado de la batalla, la imagen cambia según sea victoria o derrota para el atacante, se tiene la información de las tropas iniciales, las tropas restantes luego del combate, el botín en mithril obtenido y los puntos militares obtenidos.



Derrota

dragon

	Arqueros	Lanceros	Infanteria	Caballos
Tropas Iniciales	300	300	200	300
Tropas Perdidas	126	45	140	192
Puntos de batalla	701			

Volver al territorio

Ilustración 52 Resultado de la Batalla

10.6 Sistema de Mensajes

En el MMORPG Gaia Conquest permite establecer interacción entre jugadores por medio del envío y recepción de mensajes, de esta manera los jugadores podrán tener una comunicación más cercana a las estrategias o ataques que desean realizar.

Mensajes recibidos

Enviado por:
dragon

Mensaje:
mensaje de prueba

Enviado en:
2013-01-06 23:16:25 UTC

marcar todos como leídos

Ilustración 53 Mensaje Recibido

10.7 Sistema de Alianzas

El sistema de Alianzas brinda la posibilidad de que los jugadores estén aún más involucrados dentro del juego, ya que permite compartir diferentes ideas de ataque y estrategias en común y lograr un nivel alto dentro del Ranking de Alianzas.



Ilustración 54 Alianzas.

10.8 Guerras entre alianzas

A través de las guerras entre alianzas se logra que los jugadores compartan ideales sean competitivos o establezcan mayor comunicación para lograr diferentes objetivos de ataque.

10.9 Software

El MMORPG se desarrolló en el Framework RUBY ON RAILS utilizando el IDE de desarrollo Aptana Studio 3 de libre distribución, como interfaz gráfica JQuery y html5 y el patrón de diseño implementado es MVC (Modelo Vista Controlador).

Cuenta además con cuatro (4) controladores que ejecutan la lógica de la aplicación, seis (6) vistas principales, un (1) layout que contiene los encabezados principales que incluyen los links a los scripts y archivos css utilizados y trece (13) modelos que corresponden a las tablas de la Base de Datos.

La Base de Datos utilizada es MySql con motor INNODB el cual comparado con el clásico ISAM ofrece una estabilidad mayor para la transaccionalidad multiusuario, el MMORPG cuenta con un pool de conexiones de 8 hilos el cual permite gestionar forma eficiente las conexiones y transacciones a la Base de Datos.

Los controladores son:

- **application_controller.rb:** Contiene las consultas a la base de datos que son cargadas por la vista principal del juego tan pronto como se autentica un usuario, esta es la clase Padre de los demás controladores, los cuales heredan estas consultas cada que se requiere, normalmente en cada petición http a la vista principal del juego. Estas consultas están conformadas por recursos del jugador, ejercito, mensajes, ranking, listado de jugadores en el mapa e historial de mensajes.
- **game_controller.rb:** es el controlador principal del juego, contiene la lógica del manejo de sesiones, un llamado a las consultas de `application_controller.rb` descrito anteriormente, así como la lógica para el envío y recepción de mensajes, comprar aldeanos, recaudar recursos automaticamente, comprar las cinco (5) edificaciones civiles y las dos (2) edificaciones militares disponibles, incluir aldeanos en cada edificación civil para recaudar recursos mas rápidamente, generar recursos debido a los aldeanos trabajando, comprar los cuatro (4) diferentes tipos de soldados desde el cuartel militar (uno de los dos edificios militares), gestionar la creación de alianzas y espionaje.
- **login_controller.rb:** Contiene la lógica para la autenticación de usuarios, registro de nuevos usuarios y salir de la sesión. Cuando un usuario se registra por primera vez, se actualizan sus recursos, ejército, sistema de mensajes y aldeanos, con los cuales se crean valores por defecto que se tienen disponibles desde el inicio del juego y funcionan como punto de partida.
- **versus_controller.rb:** Contiene la lógica de las batallas entre jugadores y guerras entre alianzas, las cuales otorgan el principal puntaje para jugadores y alianzas representadas en el ranking.

Las vistas son:

- **index.html.erb (login_controller):** Contiene el formulario de autenticación de usuarios y una descripción general del MMORPG Gaia Conquest.
- **home.html.erb:** Contiene el formulario de registro de nuevos usuarios.

- **index.html.erb (game_controller):** Contiene la vista principal del Juego el cual incluye, los recursos del jugador, la cantidad de aldeanos total e inactivos, una imagen general de la provincia con un Centro Urbano el cual es el edificio principal y el que se tiene por defecto desde el inicio del juego, los iconos para el listado de jugadores del mapa, el ranking, los mensajes, las alianzas y una vez se tengan, se pueden ver las demás edificaciones civiles y militares. Los iconos son administrados por el framework JQuery mediante ventanas emergentes de tipo dialog.
- **register.html.erb:** Contiene el formulario de registro de nuevos jugadores.
- **application.html.erb:** Es el layout por defecto de la aplicación, contiene los encabezados principales y librerías necesarias para la correcta ejecución del Framework de JQuery, utilizado en el juego para la interfaz gráfica. Las demás vistas heredan de ésta.
- **ejercito.html.erb:** Contiene la información del ejercito disponible y los componentes necesarios para realizar un ataque a otro jugador.
- **resultadoBatalla.html.erb:** Contiene el resultado de la batalla entre jugadores el cual incluye, los ejércitos iniciales antes del combate y restantes luego del combate de ambos jugadores, los puntos militares obtenidos y el botín de Mithril obtenido si se obtuvo la victoria.

Los modelos de la aplicación son:

Nota: Todos los modelos tienen en común que implementan un método modificador para los atributos equivalentes a los campos de las tablas y restricciones para los campos que deben ser no nulos.

- **alliances.rb:** Representa la tabla alliances de la base de datos, contiene la información de las alianzas, su líder, sus integrantes, las victorias y derrotas y los puntos de alianza.
- **army.rb:** Representa la tabla army de la base de datos, contiene la información del ejercito de cada jugador y sus costos.

- **buildings.rb:** Representa la tabla buildings de la base de datos, contiene la información de los edificios civiles y militares y sus costos.
- **human_research.rb:** Representa la tabla human_research de la base de datos, contiene las investigaciones que se pueden hacer para la raza Humano y sus costos.
- **messages.rb:** Representa la tabla messages de la base de datos, contiene el historial de mensajes entre jugadores y se cuales han sido leídos y cuales no.
- **orc_research.rb:** Representa la tabla orc_research de la base de datos, contiene las investigaciones para la raza Orco y sus costos.
- **ranking_alliances.rb:** Representa la tabla ranking_alliances de la base de datos, contiene los puntos militares obtenidos por las alianzas al ganar guerras entre alianzas, así como sus victorias y derrotas.
- **ranking.rb:** Representa la tabla ranking de la base de datos, contiene los puntos militares y civiles de los jugadores y sus victorias y derrotas.
- **resources.rb:** Representa la tabla resources de la base de datos, contiene los recursos adquiridos por los jugadores.
- **user.rb:** Representa la tabla user de la base de datos, contiene los datos básicos de los jugadores como, nickname, email, contraseña de ingreso y fecha de registro.
- **versus.rb:** Representa la tabla versus de la base de datos, contiene la información de las batallas ganadas y perdidas entre jugadores, los puntos obtenidos, el botín del ganador y el ejército inicial y final después del combate.
- **villagers.rb:** Representa la tabla villagers de la base de datos, contiene la información de los aldeanos de los jugadores, el total, la cantidad de aldeanos ociosos y los aldeanos que se encuentran en cada edificación civil produciendo recursos.
- **war.rb:** Representa la tabla war de la base de datos, contiene la información de las guerras entre alianzas, sus victorias y derrotas y sus puntos militares.

12. CONCLUSIONES

- Con el MMORPG es posible establecer un proceso completo de Ingeniería de Software, basado en el análisis, diseño y desarrollo de la aplicación.
- Por medio de un MMORPG es posible gestionar un sistema de interacción a nivel masivo efectivo, que permiten generar un alto nivel de entretenimiento con un tema central y comparten los mismos gustos.
- No es una regla utilizar animación 2D o 3D, para crear un videojuego que resulte atractivo para los jugadores.
- Para desarrollar un videojuego no es estrictamente necesario utilizar un motor de videojuegos para crear un contenido innovador y entretenido, ya que se pueden implementar videojuegos utilizando muchos lenguajes de programación y framework para desarrollo web como es el caso del desarrollo de este MMORPG.
- Se crearon doce (11) casos de uso de UML, los cuales cubren toda la mecánica del juego que incluye acciones como autenticarse, registrarse como nuevo usuario, escoger raza, comprar edificaciones civiles, comprar edificaciones militares, comprar nuevos aldeanos, adquirir recursos que mejoran la economía del jugador, crear investigaciones especiales para cada raza, los cuales permiten mejorar los puntos de defensa y ataque; comprar ejercito, crear o unirse a una alianza, enviar mensajes entre jugadores y mejorar la fortaleza la cual otorga mas puntos de defensa. Es interesante la forma en que por medio de la ingeniería se logran crear entornos lúdicos y divertidos.
- Se complementa la mecánica del juego la cual incluye la interacción entre jugadores, con los diagramas de secuencia descritos en éste documento, los cuales muestran la relación entre los diferentes métodos y como éstos logran crear los mecanismos necesarios para crear interés en el juego, interacción entre jugadores que comparten un tema central lo cual permite "romper el hielo" de una forma mas natural e inadvertida.

- La implementación llevada a cabo en Ruby on Rails utilizando la metodología eXtreme Programming, permitió un desarrollo ágil y eficiente, en donde los dos integrantes de este proyecto trabajaron en equipo y enfocados a analizar y desarrollar las diferentes etapas de forma conjunta como lo promueve la metodología, solucionando en tiempos adecuados los inconvenientes presentados y proponiendo constantemente mejores formas de implementar los métodos del juego.
- El óptimo y estricto manejo del patrón de diseño MVC (*Model View Controller*) permite tener un proyecto de Ingeniería de Software ordenado y entendible, en donde los modelos y sus propiedades son relacionados únicamente con las tablas de la Base de Datos; así mismo, los controladores solo permiten adicionar código Ruby con lo cual se debe enfocar en la lógica de negocio, consultas a la base de datos por medio de los modelos y administración del envío y recepción de las peticiones de usuario desde las vistas; las vistas fueron enriquecidas con los componentes gráficos del Framework JQuery los cuales complementan muy bien los proyectos web basados en Ruby on Rails.
- Debido a la arquitectura establecida para éste proyecto y todos los demás componentes y elementos utilizados para el desarrollo del MMORPG, el mantenimiento o actualizaciones del mismo se hacen de manera rápida y fácil, a ésto contribuye el hecho de que Ruby on Rails es un Framework de última generación y maneja la filosofía de no repetir código innecesariamente y no estar enfocado a configuraciones y simbología.
- Se cumplieron todos los objetivos del proyecto así como los elementos descritos en el alcance.

13. BIBLIOGRAFIA

- *blizzard Entertainment*. (s.f.). Recuperado el 05 de 04 de 2012, de www.blizzard.com
- *IGDA Asociaion Colombiana de Desarrolladores de Videojuegos*. (s.f.). Recuperado el 06 de 04 de 2012, de <http://igdacolombia.co/>
- Pressman, R. S. (2002). *Ingenieria de Software, Un Enfoque Practico*. Mexico: McGraw-Hill.
- Sommerville, I. (2005). *Ingenieria del Software*. Mexico: Pearson.
- Sosa Flores, M. (. (2007). *Diseño de Bases de Datos Relacionales*. Argentina: El Cid Editor.
- *Ruby on Rails* Recuperado el 11 de Noviembre de 2012, de www.rubyonrails.com
- *Ruby-lang* Recuperado el 11 de Diciembre de 2012, de www.ruby-lang.org/es/
- *TIOBE* Recuperado el 13 de Noviembre de 2012, de www.tiobe.com
- *Jesse Schell Carnegie Mellon University*.(2008). *The Art of Game Design*. Burlinton USA.
- *Stackoverflow* Recuperado el 12,15,24 de Diciembre de 2012, de [www.stackoverflow.com /](http://www.stackoverflow.com/)

LICENCIA DE USO – AUTORIZACIÓN DE LOS AUTORES

Actuando en nombre propio identificado (s) de la siguiente forma:

Nombre Completo Angela Paola Gómez Holín

Tipo de documento de identidad: C.C. ☒ T.I. ☐ C.E. ☐ Número: 53.014163

Nombre Completo Carlos Augusto Alciniégro Herra

Tipo de documento de identidad: C.C. ☒ T.I. ☐ C.E. ☐ Número: 80.084211

Nombre Completo _____

Tipo de documento de identidad: C.C. ☐ T.I. ☐ C.E. ☐ Número: _____

Nombre Completo _____

Tipo de documento de identidad: C.C. ☐ T.I. ☐ C.E. ☐ Número: _____

El (Los) suscrito(s) en calidad de autor (es) del trabajo de tesis, monografía o trabajo de grado, documento de investigación, denominado:

Desarrollo de un videojuego MMORPG de tipo Browser-Game
(MMORPG: Massively Multiplayer Online Role Playing Game)

Dejo (dejamos) constancia que la obra contiene información confidencial, secreta o similar: SI ☐ NO ☒
(Si marqué (marcamos) SI, en un documento adjunto explicaremos tal condición, para que la Universidad EAN mantenga restricción de acceso sobre la obra).

Por medio del presente escrito autorizo (autorizamos) a la Universidad EAN, a los usuarios de la Biblioteca de la Universidad EAN y a los usuarios de bases de datos y sitios webs con los cuales la Institución tenga convenio, a ejercer las siguientes atribuciones sobre la obra anteriormente mencionada:

- A. Conservación de los ejemplares en la Biblioteca de la Universidad EAN.
- B. Comunicación pública de la obra por cualquier medio, incluyendo Internet
- C. Reproducción bajo cualquier formato que se conozca actualmente o que se conozca en el futuro
- D. Que los ejemplares sean consultados en medio electrónico
- E. Inclusión en bases de datos o redes o sitios web con los cuales la Universidad EAN tenga convenio con las mismas facultades y limitaciones que se expresan en este documento
- F. Distribución y consulta de la obra a las entidades con las cuales la Universidad EAN tenga convenio

Con el debido respeto de los derechos patrimoniales y morales de la obra, la presente licencia se otorga a título gratuito, de conformidad con la normatividad vigente en la materia y teniendo en cuenta que la Universidad EAN busca difundir y promover la formación académica, la enseñanza y el espíritu investigativo y emprendedor.

Manifiesto (manifestamos) que la obra objeto de la presente autorización es original, el (los) suscritos es (son) el (los) autor (es) exclusivo (s), fue producto de mi (nuestro) ingenio y esfuerzo personal y la realizó (zamos) sin violar o usurpar derechos de autor de terceros, por lo tanto la obra es de exclusiva autoría y tengo (tenemos) la titularidad sobre la misma. En vista de lo expuesto, asumo (asumimos) la total responsabilidad sobre la elaboración, presentación y contenidos de la obra, eximiendo de cualquier responsabilidad a la Universidad EAN por estos aspectos.

En constancia suscribimos el presente documento en la ciudad de Bogotá D.C.,

NOMBRE COMPLETO: Angelo Nino Gomez Nain
FIRMA: Angelo Gomez N.
DOCUMENTO DE IDENTIDAD: 53014163
FACULTAD: Ingeniería
PROGRAMA ACADÉMICO: Ingeniería de Sistemas

NOMBRE COMPLETO: Carlos Augusto Aciniegas Mesa
FIRMA: CA
DOCUMENTO DE IDENTIDAD: 80084211
FACULTAD: Ingeniería
PROGRAMA ACADÉMICO: Ingeniería de Sistemas

NOMBRE COMPLETO: _____
FIRMA: _____
DOCUMENTO DE IDENTIDAD: _____
FACULTAD: _____
PROGRAMA ACADÉMICO: _____

NOMBRE COMPLETO: _____
FIRMA: _____
DOCUMENTO DE IDENTIDAD: _____
FACULTAD: _____
PROGRAMA ACADÉMICO: _____

Fecha de firma: Enero 31 / 2013