

**Investigar y crear un plugin de Razonamiento Basado en Casos (CBR) para el control de personajes no controlados por el jugador (NPCs) en un servidor Spigot por medio de un segundo plugin para Minecraft versión Java.**

Autor: Víctor Aníbal Mejía Sandoval.

Tutor: Jordi Duch Gavaldà.

Profesor: Joan Arnedo Moreno.

Máster Universitario en Diseño y Programación de Videojuegos.

Área de especialidad/Itinerario: Nuevas tecnologías e investigación.

Fecha de entrega: 6 de marzo de 2022.



# Créditos/Copyright



Esta obra está sujeta a una licencia de Reconocimiento- NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/).

Copyright © 2022 Víctor Aníbal Mejía Sandoval.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Investigar y crear una herramienta Razonamiento Basado en Casos (CBR) para el control de personajes no controlados por el jugador (NPCs) en un servidor Spigot por medio de un plugin de Minecraft versión Java</i>
<b>Nombre del autor:</b>	<i>Víctor Aníbal Mejía Sandoval</i>
<b>Nombre del colaborador/a docente :</b>	<i>Jordi Duch Gavalda</i>
<b>Nombre del PRA:</b>	<i>Helio Tejedor Navarro</i>
<b>Fecha de entrega (mm/aaaa):</b>	<i>06/2022</i>
<b>Titulación o programa:</b>	<i>Máster Universitario en Diseño y Programación de Videojuego</i>
<b>Área del Trabajo Final:</b>	<i>Trabajo Final de Máster</i>
<b>Idioma del trabajo:</b>	<i>Español</i>
<b>Palabras clave</b>	<i>Razonamiento Basado en Casos (CBR), Control de personajes no controlados por el jugador (NPCs). Servidor Spigot, Minecraft versión Java, herramienta para desarrolladores, Plugin Spigot</i>
<b>Resumen del Trabajo:</b>	
<p>El servidor Spigot permite a los programadores crear mini juegos que aprovechan el entorno del videojuego Minecraft versión Java. La comunidad de desarrolladores de plugins para el servidor Spigot aporta con diversas herramientas para el control de personajes NPCs del juego. El presente proyecto implementa la herramienta de Razonamiento Basado en Casos (CBR) como plugin de Spigot para el control de personajes no controlados por el jugador (NPC) en un servidor Spigot como un utilitario que facilite el desarrollo de personajes que requieran comportamientos inteligentes. Para la construcción del modelo y los casos se ha utilizado la herramienta de código abierto MyCBR, y para la gestión de los movimientos y eventos del NPC se ha utilizado el plugin Citizens2. Para las pruebas del plugin CBR se ha creado un mini juego para Spigot que utiliza el modelo y los casos CBR para controlar a un NPC y así competir contra un jugador humano. Gran parte de la investigación se ha dedicado a la elaboración del mini juego SpeedRun, de las librerías del API de Minecraft, del API de Spigot, del API de Citizens 2, y de las herramientas de desarrollo del SDK para programadores de plugins de dicho videojuego.</p>	
<b>Abstract (in English, 250 words or less):</b>	
<p>The Spigot server allows programmers to create mini-games that take advantage of the Minecraft Java version video game environment. The community of plugin developers for the Spigot server provides various tools for controlling NPC characters in the game. This project implements the Case-</p>	

Based Reasoning (CBR) tool as a Spigot plugin for the control of non-player-controlled (NPC) characters on a Spigot server as a utility that facilitates the development of characters that require intelligent behaviors. For the construction of the model and the cases, the MyCBR open source tool has been used, and the Citizens2 plugin has been used for the management of the movements and events of the NPC. For the tests of the CBR plugin, a mini game for Spigot has been created that uses the CBR model and cases to control an NPC and thus compete against a human player. Much of the research has been devoted to the development of the SpeedRun minigame, the Minecraft API libraries, the Spigot API library, the Citizens2 API library, and the SDK development tools for Minecraft plugin developers.

## Dedicatoria

Dedico el presente trabajo de final de máster a mi amada esposa; gracias a su apoyo y compromiso, hemos conjuntamente logrado cumplir un reto más en nuestro camino de vida. Agradezco a mi adorada madre por su amor y compañía espiritual durante este período de estudio y superación.

## Agradecimientos

Considero oportuno agradecer al cuerpo docente de la UOC, en especial a Jordi y a Helio por sus consejos y sugerencias; fueron de gran ayuda para el desarrollo de la investigación y la programación del presente proyecto.

## Abstract

El servidor Spigot permite a los programadores crear mini juegos que aprovechan el entorno del videojuego Minecraft versión Java. La comunidad de desarrolladores de plugins para el servidor Spigot aporta con diversas herramientas para el control de personajes NPCs del juego. El presente proyecto implementa la herramienta de Razonamiento Basado en Casos (CBR) como plugin de Spigot para el control de personajes no controlados por el jugador (NPC) en un servidor Spigot como un utilitario que facilite el desarrollo de personajes que requieran comportamientos inteligentes. Para la construcción del modelo y los casos se ha utilizado la herramienta de código abierto MyCBR, y para la gestión de los movimientos y eventos del NPC se ha utilizado el plugin Citizens2. Para las pruebas del plugin CBR se ha creado un mini juego para Spigot que utiliza el modelo y los casos CBR para controlar a un NPC y así competir contra un jugador humano. Gran parte de la investigación se ha dedicado a la elaboración del mini juego SpeedRun, de las librerías del API de Minecraft, del API de Spigot, del API de Citizens 2, y de las herramientas de desarrollo del SDK para programadores de plugins de dicho videojuego.

The Spigot server allows programmers to create mini-games that take advantage of the Minecraft Java version video game environment. The community of plugin developers for the Spigot server provides various tools for controlling NPC characters in the game. This project implements the Case-Based Reasoning (CBR) tool as a Spigot plugin for the control of non-player-controlled (NPC) characters on a Spigot server as a utility that facilitates the development of characters that require intelligent behaviors. For the construction of the model and the cases, the MyCBR open source tool has been used, and the Citizens2 plugin has been used for the management of the movements and events of the NPC. For the tests of the CBR plugin, a mini game for Spigot has been created that uses the CBR model and cases to control an NPC and thus compete against a human player. Much of the research has been devoted to the development of the SpeedRun minigame, the Minecraft API libraries, the Spigot API library, the Citizens2 API library, and the SDK development tools for Minecraft plugin developers.

### Palabras clave

Razonamiento Basado en Casos (CBR), Control de personajes no controlados por el jugador (NPCs). Servidor Spigot, Minecraft versión Java, herramienta para desarrolladores, Plugin Spigot.

# Índice

1.	Introducción	12
1.1.	Descripción/Definición	13
1.2.	Objetivos generales	15
1.2.1.	Objetivos principales	15
1.2.2.	Objetivos secundarios	16
1.3.	Metodología y proceso de trabajo	17
1.4.	Planificación	20
1.5.	Presupuesto	21
2.	Análisis de mercado	22
2.1.	Público objetivo y perfiles de usuario	22
2.2.	Antecedentes	23
2.3.	Análisis DAFO	24
3.	Fundamentos	26
3.1.	El Razonamiento Basado en Casos (CBR)	26
3.2.	La herramienta myCBR	28
3.3.	El servidor SpigotMC	29
3.4.	El plugin Citizens 2	30
3.5.	Experiencias del uso del CBR en Videojuegos	30
3.6.	El minijuego SpeedRun y su modelo CBR	31
4.	Diseño	34
4.1.	Diseño del plugin cbrplugin	34
4.2.	Arquitectura general de la aplicación	36
4.3.	Arquitectura de la información y diagramas de navegación	37
4.4.	Diseño gráfico e interfaces	44
4.5.	Estilos	52
4.6.	Usabilidad /UX	53
4.7.	Lenguajes de programación y APIs utilizados	53
5.	Implementación	54

5.1. Requisitos de instalación	54
5.2. Instrucciones de instalación	54
6. Demostración	57
6.1. Instrucciones	57
7. Conclusiones y líneas de futuro	58
7.1. Conclusiones	58
7.2. Líneas de futuro	59

## Figuras y tablas

Lista de imágenes, tablas, gráficos, diagramas, etc., numeradas, con títulos y las páginas en las cuales aparecen. Para actualizar cada uno de los índices, hay que hacer botón derecho con el ratón y escoger la opción “Actualizar campos”.

### Índice de figuras

Figura 1: Diagrama GANT del proyecto.....	20
Figura 2: La arquitectura del proyecto de código libre myCBR .....	28
Figura 3: Diseño general de los plugins cbrplugin y SpeedRun.....	34
Figura 4: Arquitectura de cbrplugin.....	36
Figura 5: Arquitectura del plugin SpeedRun .....	36
Figura 6: Diseño de Clases de cbrplugi .....	37
Figura 7: Control de eventos del jugador y del NPC.....	38
Figura 8: Clases para representar el escenario y sus elementos .....	39
Figura 9: Clases Util para generar los contenidos del archivo log.....	40
Figura 10: Clases del paquete Manager .....	40
Figura 11: Clases ConnectListener y GameListener .....	41
Figura 12: Clase Arena y Clase GUI.....	42
Figura 13: Clases para la elaboración de productos .....	43
Figura 14: Clase para gestionar la línea de comandos .....	43
Figura 15: Paquete cbr interfaz con cbrplugin.....	44
Figura 16: Pantalla de inicio en el servidor de SpeedRun .....	45
Figura 17: Barra de ayuda para empezar una partida .....	45
Figura 18: Letrero asociado a la acción de inicio de la arena .....	46
Figura 19: Cuenta atrás e inicio del juego en la arena seleccionada .....	46
Figura 20: Diseño básico de una arena de juego .....	46
Figura 21: Diseño de las tres arenas del juego .....	47
Figura 22: Interfaz del cofre tipo almacén.....	47
Figura 23: Diseño de la base del NPC para crear productos.....	47
Figura 24: Diseño del letrero para salir de la arena.....	48
Figura 25: Diseño de la pantalla de Game Over .....	48
Figura 26: Diseño de la pantalla WIN del jugador .....	49
Figura 27: Lista de directorios del proyecto .....	55
Figura 28: Archivo YML de configuración de cbrplugin .....	55
Figura 29: Entradas del archivo log del servidor SpigotMC .....	55

# 1.Introducción

La mejora continua del apartado gráfico en los videojuegos dan lugar a producciones cada vez más realistas, como es el caso de God of War: Ragnarok (MeriStation, 2021), o el de Forza Horizon 4 (Microsoft, 2021). De igual manera el nivel de credibilidad de los personajes no jugables (NPCs) en ciertos videojuegos ha otorgado el carácter y la profundidad necesarios para mantener inmerso al jugador. Un ejemplo lo es Red Dead Redemption 2 (Rockstar Games, 2021), que presenta NPCs capaces de llevar a cabo actividades rutinarias como levantarse, trabajar, tomar un descanso, ir a la cantina, y al final del día retirarse a descansar ( DefendTheHouse, 2018) con la suficiente naturalidad como para mantener la ilusión de ser real.

Aquellos juegos que no han invertido recursos en el desarrollo de sus NPCs, por regla general permiten modificar sus comportamientos por código, dando la posibilidad de implementar técnicas para otorgar conductas inteligentes a dichos personajes. Minecraft es un buen ejemplo de un videojuego que permite programar “mods” mediante librerías disponibles en su comunidad de desarrolladores (Minecraftmin, 2017); la edición de Minecraft versión Java dispone de documentación oficial de Mojang específica para la creación de plugins (Minecraft Wiki Oficial, 2021) con el objetivo de extender dichas características.

La idea del presente proyecto es diseñar una herramienta que permita implementar el modelo de razonamiento basado en casos, CBR por sus siglas en inglés (López, 2013), para enriquecer el comportamiento de NPCs en un mini juego de Minecraft versión Java. La herramienta permitirá que los NPCs creados puedan registrar las interacciones del personaje con los elementos del videojuego, y un mecanismo para encontrar cuál es el comportamiento más cercano a la situación actual para continuar con la acción.

Si bien existen varios complementos para Minecraft que se usan para crear y configurar los NPCs, como es el caso de Custom NPCs (Noppes, 2014) y Comes Alive (Radix-Shock, 2021), son propuestas generalistas que abarcan muchos apartados, como los gráficos, texturas, items, etc., dejando el comportamiento del personaje en un segundo plano. El proyecto de desarrollo de un plugin que permite utilizar CBR en NPCs está dirigido a los creadores de mapas y juegos de Minecraft versión Java que deseen añadir comportamiento inteligente a sus personajes.

## 1.1. Descripción/Definición

### • Punto de partida del trabajo

Los administradores y desarrolladores de los servidores que ofrecen mapas y minijuegos de Minecraft buscan atraer jugadores mediante la originalidad y diversidad de sus propuestas. Al igual que los juegos clásicos del tipo SkyMap, SkyWars, Murder Mystery, para servidores Minecraft (Minecraft Servers, 2022); existen múltiples propuestas de programadores independientes que utilizan a Minecraft como su ambiente de desarrollo de videojuegos. Por ejemplo el plugin de Citizens 2 (Citizens, 2022), que brinda muchas posibilidades por medio de sus APIs para controlar a los NPCs del juego.

### • ¿Cuál es la necesidad a cubrir?

Los desarrolladores de juegos para servidores Minecraft se encuentran en búsqueda continua de plugins que faciliten el desarrollo de sus juegos. Los plugins que dotan de cierto grado de inteligencia artificial ayudan a dar un aspecto realista a los NPCs de los juegos, son apreciados por los desarrolladores. El método de Razonamiento Basado en Casos no ha sido explotado en la comunidad de desarrolladores de minijuegos para el servidor Spigot de Minecraft, dejando un vacío entre las estrategias de inteligencia artificial disponibles para gestionar NPCs.

### • ¿Por qué es un tema relevante?

Los servidores para Minecraft como Spigot son la demostración de los resultados de una comunidad de programadores dedicada a desarrollar iniciativas para videojuegos. Una forma de apoyar a dicha comunidad es proponer proyectos que faciliten la creación de dichas propuestas. La escasa producción de plugins orientados al uso del método de Razonamiento Basado en Casos representa una oportunidad para aportar a la comunidad de desarrolladores, al mismo tiempo que se sientan las bases para mejorar la calidad de sus mini juegos en Minecraft.

### • ¿Cómo se resuelve el problema en la actualidad?

Cada proyecto de videojuego afronta a su manera la elaboración de alguna forma de inteligencia artificial para dotar de realismo a sus NPCs. Un ejemplo es el paquete Navigator del plugin Citizens 2, que permite a un NPCs caminar sobre un terreno, subir escaleras y superar pequeños obstáculos; mientras tanto el plugin NPCDestinations (NPCDestinations, 2022), a pesar de estar basado en Citizens 2, implementa su propio sistema de rutas mediante mallas de navegación. Para el comportamiento de los NPCs, Citizens 2 ha implementado un sistema de árboles de decisiones, el cual se utiliza para gestionar sencillos diálogos con los personajes. Luego de una investigación acerca de los plugins que implementan diversas estrategias de inteligencia artificial, no se ha encontrado alguno que ofrezca el método CBR, razón por la cual se considera relevante el aportar con esta estrategia a las soluciones ahora propuestas.

### • Aportación realizada

Se ha puesto a prueba con relativo éxito la implementación de un plugin para el servidor Spigot de Minecraft versión Java que permite a los desarrolladores crear NPCs con inteligencia artificial que

use el Razonamiento Basado en Casos. Para probar la utilidad del plugin CBR se ha creado un segundo plugin que contiene un mini juego que utiliza sus librerías, demostrando que al menos en este caso, es posible utilizarlo para programar un NPC funcional.

• **¿Qué resultado se quiere obtener?**

Se desea obtener un plugin para Spigot basado en Citizens 2 que permita implementar el método CBR a los NPCs de un mini juego de Minecraft versión Java. Como prueba de concepto se realizará un mini juego para el servidor Spigot que ponga en práctica el plugin CBR.

• **¿Qué productos se obtendrán como resultado final de este trabajo?**

El primer producto es el plugin CBR para Spigot, el segundo producto es un ejemplo de cómo se lo utilizará para otorgar comportamiento inteligente a un NPC en un mini juego del servidor para Minecraft.

## 1.2. Objetivos generales

- Construir un plugin que aplique el método de Razonamiento Basado en Casos (CBR) para el control de personajes no controlados por el jugador (NPCs) en un servidor Spigot de Minecraft versión Java.
- Desarrollar un mini juego para el servidor Spigot que ponga en práctica el plugin de CBR que demuestre las posibilidades de controlar un NPC usando dicho plugin.
- Investigar acerca del aprendizaje mediante CBR para el control de personajes no controlados por el jugador (NPCs) y las librerías disponibles en la comunidad de código abierto que permitan utilizar CBR para el desarrollo de aplicaciones
- Investigar el entorno de programación de Minecraft, sus librerías, clases y ambiente de desarrollo.
- Investigar el entorno de programación del servidor Spigot, sus librerías, clases y herramientas de desarrollo.
- Investigar acerca de la estructura de los plugins para Minecraft versión Java y su metodología de programación.
- Investigar acerca de la API del plugin de Citizens 2 para el servidor Spigot para utilizarlas en el control del movimiento del NPCs
- Crear un entorno de desarrollo y pruebas para crear plugins para el servidor Spigot de Minecraft versión Java.

### 1.2.1. Objetivos principales

Objetivos de la aplicación/producto/servicio:

- Crear un plugin que permita implementar la metodología básica de CBR para el servidor de Spigot.
- Crear un plugin que contenga un mini juego para el servidor Spigot de Minecraft que demuestre la utilización del plugin desarrollado

Objetivos para el cliente/usuario:

- Disponer de un plugin que ayude a desarrollar mini juegos en servidores de Minecraft para ayudar a controlar a sus NPCs
- Crear un mini juego que sirva de prueba de concepto del plugin CBR

Objetivos personales del autor del TF:

- Aportar a la comunidad de desarrolladores con un plugin que les facilite la tarea de implementar CBR para NPCs en mini juegos de Minecraft
- Investigar acerca de las capacidades que ofrece el servidor Spigot para el desarrollo de videojuegos.

## 1.2.2. Objetivos secundarios

Objetivos adicionales que enriquecen el TF.

- Adquirir la suficiente destreza en el lenguaje Java que permita desarrollar plugins y mini juegos para el servidor Spigot de Minecraft
- Adquirir el conocimiento necesario de la estructura de clases de Minecraft para interactuar con los elementos del escenario (cobres, carteles, puertas, losas) mediante código Java.
- Aprender los principios básicos del CBR y cómo aplicarlos al control de NPCs en videojuegos
- Elaborar el modelo del CBR y construir una base de casos aplicables al mini juego de prueba.
- Aprender los detalles técnicos de Citizens 2 para utilizar sus clases, interfaces, eventos, enumeraciones.
- Aprender el mecanismo para lectura de archivos planos con valores de instalación y configuración por medio del servidor Spigot.

### 1.3. Metodología y proceso de trabajo

#### Descripción de la metodología de trabajo o investigación aplicada y conceptos clave.

La metodología utilizada en el presente proyecto consiste de los siguientes pasos:

- Investigar acerca del CBR, el modelo y la base de datos de casos.
- Investigar acerca de librerías que implementan CBR disponibles en la comunidad de código abierto para utilizarlas en el desarrollo del plugin.
- Investigar el lenguaje de programación Java y el entorno de desarrollo IntelliJ IDEA y adquirir la destreza suficiente para el desarrollo de plugins para Minecraft.
- Investigar las librerías y clases disponibles para desarrollar aplicaciones que se ejecuten en el servidor Spigot.
- Participar en los foros de la comunidad de desarrolladores de Spigot para obtener ayuda con los mensajes de error y otras características propias de la programación de sus plugins
- Investigar acerca de los plugins disponibles para el servidor Spigot que permitan controlar NPCs en un juego de Minecraft.
- Investigar acerca del ciclo de vida y la secuencia de ejecución de los eventos del servidor Spigot para implementar un motor básico de minijuegos para Minecraft.
- Crear y diseñar las mecánicas de un mini juego para Minecraft que permita poner a prueba el plugin con CBR como controlador de las acciones de un NPC.
- Diseñar el modelo CBR y la base de datos de casos que permitan controlar a un NPC con el objetivo de que cumpla con las mecánicas del mini juego de prueba.
- Crear el plugin que implemente la librería de código abierto de CBR y la ponga a disposición de otros plugins en el servidor de Spigot.
- Crear el plugin que implementa el mini juego en el servidor Spigot para poner a prueba el plugin CBR.
- Crear un sistema de generación de registros de actividad del NPC y del jugador para estudiar el desempeño del modelo CBR.

#### Descripción general del proceso de trabajo/desarrollo realizado.

El proceso de trabajo realizado contó con tres fases que se detallan a continuación:

*Fase de aprendizaje:* Con el objetivo de implementar un ambiente de desarrollo de plugins para Minecraft, se investigó acerca del lenguaje de programación Java, las diferencias entre las distintas versiones utilizadas para el servidor Spigot, la forma cómo se debe configurar Java para evitar errores en las etapas de compilación y generación de artefactos.

Se investigó cómo agilizar las tareas de compilación de proyectos Java mediante la ayuda de Maven, la sintaxis de los archivos de configuración, acerca de cómo mantener las referencias a las librerías externas utilizadas en la creación de los plugins. Se investigó acerca del entorno de desarrollo (IDE)

adecuado para la creación de plugins para el servidor de Spigot, aquel que facilitara las tareas de control de compilación y generación mediante Maven, la integración con proyectos tipo Spigot, la forma como se realiza el proceso de debugger y corrección de errores. Se elige IntelliJ IDEA y se indaga en la documentación de la comunidad Spigot acerca de la configuración óptima del editor para programar plugins. Se realizan pruebas de debugger de código en tiempo de ejecución entre el editor, el servidor Spigot y el plugin.

Se investigaron las librerías de Bukkit para conocer las clases y métodos con los cuales se accede a los elementos del juego Minecraft. Se buscó material bibliográfico y cursos en línea para obtener información acerca del proceso de programación y generación de plugins específicos para el servidor Spigot, su ciclo de vida, los eventos y los mensajes entre el plugin y el servidor, así como con los clientes Minecraft, la estructura de directorios y convención de archivos. Se realizan pruebas de integración entre las diferentes versiones de Java, del servidor de Spigot, del cliente Microsoft.

Se investigó sobre los fundamentos de CBR y las librerías de software libre disponibles en la comunidad que permiten implementar CBR en desarrollos propios. Se eligió la librería MyCBR por las facilidades que brinda para leer la base de casos a partir de un archivo plano CSV. Se crearon varios modelos con la aplicación GUI de MyCBR junto con sus bases de casos correspondientes, para probar la funcionalidad de la API de MyCBR a la hora de recuperar el mejor caso. Se realizaron pruebas de compatibilidad entre la versión de Java en la que está originalmente el API de MyCBR y las versiones disponibles de Spigot.

*Fase de desarrollo:* Se desarrolló un plugin para el servidor Spigot que publicase las funcionalidad básica del CBR para de lectura del modelo, la lectura de la base de casos, la construcción de las consultas al motor de casos del CBR y la obtención del caso que mejor se asemejara a un estado dado. Se optó por construir el plugin en base del código de la librería de myCBR en lugar de utilizarla como una librería externa, debido a las incompatibilidades encontradas en las pruebas iniciales. De esta manera el plugin cuenta con un paquete para el código de myCBR y otro paquete para el código del presente proyecto. Debido a que el código original de myCBR utiliza librerías externas con versiones no actualizadas, fue necesario probar diferentes configuraciones para solventar este inconveniente.

Se diseñó un minijuego para Minecraft con el objetivo de poner a prueba el plugin CBR para controlar un NPC capaz de cumplir con las mecánicas del juego. El minijuego consta de dos bases con cofres especiales para crear productos, dos almacenes con los ingredientes, y un laberinto a transitar para acarrear los ingredientes hacia la base. El NPC y el jugador compiten para crear productos lo más rápido posible, y gana quien logra alcanzar primero el número de productos requeridos. El nombre del minijuego es SpeedRun.

Se elaboró un modelo y una base de casos para elegir cuál es la mejor acción que se debe tomar de acuerdo al estado del minijuego. El estado está determinado por las cantidades de ingredientes presentes en cada almacén, en cada cofre de la base y la cantidad de cada producto elaborado. Se diseñó el modelo utilizando la herramienta GUI de MyCBR al igual que la base de casos. Ambos archivos se utilizan posteriormente cuando el plugin CBR requiere consultar cuál es el caso que más se asemeja al estado del juego. Se realizaron pruebas manuales para determinar los cambios a realizar en el modelo hasta encontrar la base de casos que mejor desempeño presentaban de entre los que se propusieron.

*Fase de implementación:* Se investigaron los plugins disponibles para el servidor Spigot que permitieran crear y controlar los movimientos de un NPC. Se eligió el plugin Citizens 2 porque permite el control del NPC mediante eventos asociados al personaje que facilitan programar las acciones que debe realizar de acuerdo con los resultados del motor CBR. Otra ventaja del plugin Citizens 2 es la disponibilidad de diferentes versiones que son compatibles con las versiones de Spigot, lo que ha permitido trabajar sin los conflictos entre el plugin de CBR y el plugin del minijuego.

Se desarrolló el plugin con el minijuego SpeedRun para aplicar el modelo CBR disponible en el primer plugin. Se utilizó el plugin de Citizens 2 para la creación y control del NPC. El minijuego se controla mediante un conjunto de clases diseñadas para controlar las acciones del NPC mediante los resultados del motor que está en el plugin del CBR. Se realizó en primera instancia un prototipo del juego, para probar el desempeño del NPC, así como las interacciones con cofres de la base, los cofres de los almacenes, y el desplazamiento en medio del laberinto. En un segundo prototipo se diseñaron un conjunto de clases para controlar el ingreso y salida del jugador, el lobby para elegir la arena dónde se llevan a cabo las acciones de competencia con el jugador humano. Finalmente se elaboró la tercera versión del plugin que ya cuenta con un archivo plano para almacenar la configuración del minijuego así como las posiciones de los elementos con los cuales interactúan tanto el jugador como el NPC, un sistema de puntaje para determinar el ganador, un sistema de registro de actividades tipo log para revisar el rendimiento del NPC, entre otras funciones.

## 1.4. Planificación

A continuación el diagrama GANTT del proyecto

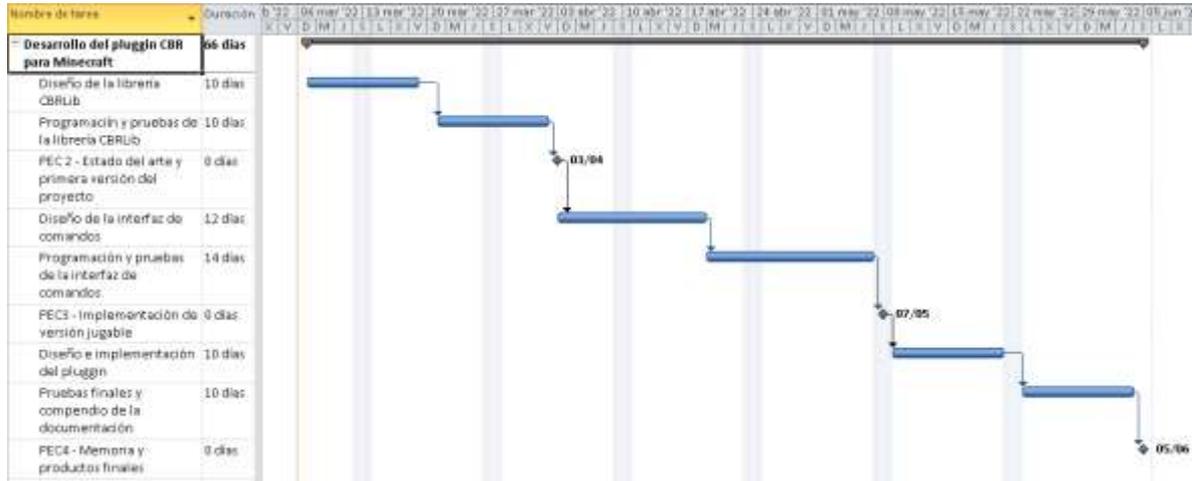


Figura 1: Diagrama GANTT del proyecto.

## 1.5. Presupuesto

Elementos del presupuesto

- Horas de trabajo del autor: Por 90 días, de 4 horas laborables por día, son 360 horas. Cada hora de trabajo valorado en 15 dólares, da un subtotal de 5 400 dólares.
- Equipamiento técnico: Computador, Internet, estación de trabajo. Por 90 días, a 6 dólares por arriendo diario de 4 dólares, da un subtotal de 540 dólares
- Otros recursos: Compra de cuenta Premium de Citizens2: 30 dólares. Compra de tres artículos de revistas especializadas, a 40 dólares cada artículo: 120 dólares.

El presupuesto total es de 6 090 dólares.

## 2. Análisis de mercado

La comunidad de desarrolladores de plugins para el servidor de Spigot de Minecraft se caracteriza por realizar aportes de manera gratuita. El sitio web de Spigot facilita a los desarrolladores las actividades orientadas a compartir sus plugins con la comunidad de jugadores y otros desarrolladores. El código de los plugins se encuentra disponible al público sin necesidad siquiera de una cuenta en la página de Spigot. Por otra parte, Spigot también permite a los desarrolladores recibir donaciones voluntarias provenientes de miembros de la comunidad que desean apoyar tal o cual iniciativa. El valor de una iniciativa se mide por la cantidad de personas que apoyan la idea del plugin. A más del apoyo monetario que se da por medio de donaciones, también se aporta con trabajo de programación, pruebas, documentación, divulgación y capacitación.

El presente proyecto es de investigación, busca proponer a la comunidad de desarrolladores de plugins una herramienta del razonamiento basado en casos (CBR) para sus proyectos. Como punto de partida para la investigación se eligió Spigot por la disponibilidad de la documentación frente a otras opciones como PaperMC (papermc.io), Purpur (purpurmc.org) o el propio Minecraft Server. Por este motivo el mercado está focalizado a la comunidad de Spigot, en particular a aquellos que utilizan el plugin Citizens2, el cual dispone de los árboles de comportamiento (behavior trees) pero carece de la opción del razonamiento basado en casos CBR.

Respecto a las librerías de código abierto que ofrecen implementaciones de CBR, la librería myCBR ofrece la ventaja comparativa de contar con una herramienta GUI para facilitar la creación de los modelos así como también para la creación y mantenimiento de las bases de casos. Si bien el proyecto de myCBR no cuenta con nuevas actualizaciones, su planteamiento sencillo y su orientación académica lo vuelven idea para el desarrollo rápido de aplicaciones en videojuegos.

### 2.1. Público objetivo y perfiles de usuario

La audiencia para la cual está enfocado es a los desarrolladores de plugins del servidor Spigot para Minecraft, de manera particular para quienes utilizan el plugin Citizens2 que está dedicado a programar los NPCs del juego. Los miembros de esta comunidad son programadores que de manera voluntaria aportan con diferentes tareas de desarrollo y mantenimiento en los proyectos publicados en la plataforma de Spigot. Al igual que otras comunidades de software libre de videojuegos, en la plataforma de Spigot participan desarrolladores de software que también son jugadores de la aplicación, en este caso de Minecraft. Se comparten los ideales de mejorar el juego para adecuarse a sus propios requerimientos, de apoyar iniciativa de terceros mediante el trabajo voluntario, y de conocer a otros desarrolladores con quienes se pueda compartir las experiencias relacionadas con el software libre.

## 2.2. Antecedentes

En la comunidad Spigot existen varios proyectos acerca de la aplicación de inteligencia artificial para mejorar la experiencia de los usuarios del servidor de Minecraft, sin embargo si las comparamos con los proyectos dedicados a las mejoras de otros aspectos del juego, como la interfaz visual, las mecánicas del juego, las construcciones de escenarios, el aspecto visual de los personajes, se puede apreciar que las aplicaciones de IA son la minoría. Entre estos proyectos de IA para Spigot, no existe ninguno que consista en la aplicación del razonamiento basado en casos CBR; la opción que se utiliza es la implementación de máquinas de estado finito, o de árboles de comportamiento. Un ejemplo es el plugin Citizens 2, que implementa los árboles de comportamiento para determinar las acciones de sus NPCs.

### Antecedentes

En la comunidad de desarrollo de plugins para servidores de Minecraft, los aportes para cambiar y modificar el videojuego se han focalizado en los siguientes ámbitos:

- Añadir o modificar las mecánicas originales del juego, como por ejemplo en los servidores de Skywars, Hunger Games, o Skyblock
- Añadir o modificar los escenarios del juego, como por ejemplo en los servidores de Ant Farm, Deadly Orbit, o WesterosCraft.
- Añadir o modificar el aspecto de los jugadores o personajes del juego, como por ejemplo en los servidores de Among Us, One Lucky Day (El juego del Calamar), o Marvels héroes.
- Añadir o modificar las opciones de administración y gestión del propio servidor

Sin embargo en el ámbito de la aplicación de la inteligencia artificial para el control de NPCs, los proyectos son pocos. A continuación se presentan los más representativos:

- CommandNPCs, un mod para Minecraft que permite controlar a los NPCs mediante máquinas de estado finitas, o mediante scripts que el usuario debe ingresar manualmente. El usuario no puede interactuar a nivel de programación con las máquinas de estado, pero dispone de un menú dedicado a su configuración, así como también la posibilidad de escribir scripts mediante archivos JSON
- Custom NPCs, un mod para Minecraft que permite controlar a los NPCs mediante máquinas de estado finitas y árboles de decisiones. El usuario no puede interactuar directamente mediante programación con el mod, pero a cambio tiene un menú para configurar diálogos y acciones para sus NPCs.
- Citizens2, un plugin para el servidor de Spigot de Minecraft utiliza los árboles de comportamiento para configurar las reacciones de sus NPCs, y también dispone de una

librería API para que los desarrolladores puedan utilizar este mecanismo de inteligencia artificial.

Ninguno de los ejemplos antes citados dispone del mecanismo del razonamiento basado en casos CBR para controlar los NPCs, lo cual representa una oportunidad para los proyectos en dicha línea de investigación.

### **Visión global de la situación actual del ámbito del proyecto propuesto.**

En base de la revisión de los proyectos disponibles para la creación de servidores de Minecraft, se puede describir la situación actual al respecto de la técnica del uso de razonamiento basado en casos como escasa e incluso nula. Es posible que la causa sea el trabajo que implica construir una base de casos significativa que pueda responder de manera aceptable durante el juego.

Las técnicas de máquinas de estados finitos o de árbol de comportamientos evitan el proceso de preparación de los casos, por lo que pueden ser más fáciles de adaptar a los NPCs del juego. Si se analiza desde el punto de vista del CBR, las dos propuestas que ahora se utilizan, estados finitos y árbol de comportamientos, muestran limitaciones cuando se desean implementar mecánicas de elección con muchos escenarios posibles. Frente a esta situación, el presente proyecto busca cubrir esta necesidad y abrir la posibilidad de utilizar el CBR, tanto solo, como en combinación con las técnicas analizadas.

## **2.3. Análisis DAFO**

El proyecto a nivel interno posee las siguientes debilidades:

- El desconocimiento técnico acerca de las librerías de Spigot, de la plataforma de desarrollo IntelliJ IDEA, de la documentación del API de desarrollo para plugins de servidores de software libre del videojuego Minecraft
- La no pertenencia del autor a la comunidad de desarrolladores de Spigot, y de manera general, a cualquier otra comunidad de software libre dedicada al desarrollo de videojuegos. El desconocimiento de sus reglas de interacción, los mecanismos para solicitar soporte, los entornos digitales en los cuales compartir los conocimientos.
- El desconocimiento de la técnica de razonamiento basado en casos CBR, tanto a nivel teórico, como también de las librerías de software libre que se utilizan para implementar CBR.

A nivel interno se identifican las siguientes fortalezas

- El conocimiento teórico acerca de las metodologías de desarrollo de videojuegos adquiridos durante el master, el manejo conceptual de los elementos que se han de contemplar para llevar a cabo un videojuego.
- El conocimiento teórico acerca de la aplicación de máquinas de estado y árboles de comportamiento para controlar y gestionar el comportamiento de los NPCs de un juego.

A nivel externo se identifican las siguientes amenazas

- La escasez de fuentes documentales acerca del desarrollo de plugins para el servidor de Spigot. La documentación se encuentra desperdigada entre foros y tutoriales sin una estructura coherente y que proviene de distintos autores cada uno con sus propios objetivos.
- La característica hermética de la comunidad de desarrollo de Spigot y la actitud generalizada de desconfianza ante nuevos miembros que desconocen de las normas de comunicación y los acuerdos tácitos que rigen dicha comunidad.

A nivel externo se identifican las siguientes oportunidades

- La ausencia de un plugin que permita utilizar el razonamiento basado en casos como alternativa a las máquinas de estado finitas o a los árboles de decisión que ya están implementados en proyectos de plugins del servidor de Spigot de Minecraft
- Las facilidades que presta la plataforma de Spigot para divulgar ideas innovadoras que cautiven el interés de su comunidad de desarrolladores.

## 3. Fundamentos

### 3.1. El Razonamiento Basado en Casos (CBR)

La estrategia del Razonamiento Basado en Casos (CBR, por sus siglas en inglés) cuenta hoy en día con una amplia comunidad de investigadores que aportan de forma continua con desarrollos teóricos y aplicaciones prácticas en diferentes áreas del conocimiento, tal como lo evidencian los trabajos publicados en la 29th Conferencia Internacional ICCBR 2021 celebrada en Salamanca, España (Sánchez-Ruiz & Floyd, 2021), una tendencia que se ha mantenido al menos casi durante 30 años, desde el primer workshop europeo realizado en Alemania el año de 1993 (Aamodt & Plaza, 2001).

De acuerdo con Aamodt y Plaza (Aamodt & Plaza, 2001), el Razonamiento Basado en Casos es un paradigma para resolver problemas que, a diferencia de otras estrategias de inteligencia artificial, es capaz de utilizar la experiencia adquirida en situaciones anteriores, los casos, muy similares al problema actual. Por ello la búsqueda de la solución a un nuevo problema consiste en encontrar el caso que más se asemeje para reutilizarlo. Para Beatriz López (López, 2013) la idea detrás del CBR consiste en saltarnos la parte que consiste en resolver problemas mediante la teoría, mediante el uso del libro y directamente aportar al problema actual con la experiencia necesaria mediante una base de casos previamente construidos para resolver problemas anteriores.

Un modelo busca describir el comportamiento de un sistema, y en tal medida se puede utilizar para predecir su comportamiento y resolver un determinado problema. El Razonamiento Basado en Modelos intenta inferir soluciones a partir de la capacidad que tiene un modelo de predecir un sistema; por otra parte el Razonamiento Basado en Casos evita esta estrategia y hace uso de una base de casos exitosos en problemas previos para proponer una solución (Lenz, Bartsch-Sport, Burkhard, & Wess, 1998).

El concepto de caso en CBR consiste en el registro de un problema previo, junto con la solución total o parcial de dicho problema. Si lo representamos como un par ordenado tenemos que:

$$\text{Caso} = (\text{problema}, \text{solución})$$

La base de casos es el conjunto formado por los pares ordenados problema – solución que se consideran como la experiencia, o la memoria de casos, que se tiene acerca de un sistema. No hay un principio plenamente definido para obtener la base de casos, o si dicha experiencia es necesaria o suficiente para resolver todos los posibles problemas futuros del sistema. Simplemente se da por sentado que partimos de una base de casos dada. La selección de los casos refleja el grado de cuidado que el diseñador tiene sobre el sistema, que involucra diferentes criterios de lo bueno, lo

importante, lo significativo, lo representativo, lo deseado (Lenz, Bartsch-Sport, Burkhard, & Wess, 1998). El término de CBR es usado para referirse a una amplia variedad de estrategias de inteligencia artificial, que pueden ser clasificadas de acuerdo con los siguiente criterios propuesto por Aamodt y Plaza (Aamodt & Plaza, 2001)

- CBR basado en ejemplos, cuyos casos son ejemplos modelos de lo que se espera del sistema, y la búsqueda de un caso semejante al problema actual, consiste en una tarea de clasificación taxonómica para encontrar en cuál ejemplo modelo se debe utilizar.
- CBR basado en instancias, cuyos casos representan la mayor cantidad de instancias posibles de un concepto, para asegurar su precisión a la hora de resolver un problema. Su principal tarea es aplicar el método inductivo a las instancias para proponer una solución
- CBR basado en memoria, que consiste en albergar grandes cantidades de casos para luego utilizar métodos de búsqueda y recuperación con técnicas de procesamiento paralelo.
- CBR basado en analogías, que consiste en utilizar una base de casos que provienen de sistemas similares o analógicos, que fueron diseñados para dominios de aplicación distintos al sistema en dónde se aplican.
- CBR basado en casos, propiamente la estrategia de representar un caso mediante un vector de pares atributo-valor que describen un determinado estado del sistema, y que también pueden reflejar de alguna forma su organización interna.

En la etapa de diseño del CBR los autores (Lenz, Bartsch-Sport, Burkhard, & Wess, 1998) proponen responder las siguientes preguntas con el objetivo de definir el ámbito y la aplicación al sistema deseado:

- ¿Cuáles son los problemas y cómo se representan?
- ¿Qué significa que algo es la solución a un problema?
- ¿Cómo se representa la experiencia pasada?
- ¿Qué significa que algo sea similar?
- ¿Qué significa buscar una solución?
- ¿Cómo se usa una solución anterior?

Una vez que se han definido claramente el ámbito y la aplicación del sistema al que se aplicará, se lleva a cabo el ciclo de vida del CBR, a saber:

- Preparación de la consulta a la base de casos
- Ejecución y recuperación del caso similar al requerido
- Reutilización de la solución asociada al caso recuperado
- Evaluación del efecto producido por la solución

- Retención de parte de esta experiencia para ser usada en el futuro

La preparación de la consulta consiste en obtener del sistema los valores de los atributos que definen un caso. El caso así construido será utilizado para buscar y encontrar en la base de casos previos, el caso que sea similar. Una vez recuperado el caso similar, se obtiene la colusión asociada, y se procede a interpretar sus valores para aplicarlos al problema. Luego de las acciones tomadas para aplicar la solución, se evalúan los resultados obtenidos, por medio de una función objetivo similar a la utilizada en los problemas de programación lineal. Finalmente, hay la posibilidad de utilizar parte de esta experiencia para modificar o actualizar la base de casos (Aamodt & Plaza, 2001). Este último paso es opcional y depende del tipo de sistema al que se aplica el CBR, ya sea que se desee un proceso de aprendizaje automático, o bajo supervisión del diseñador.

### 3.2. La herramienta myCBR

myCBR es una herramienta de código abierto diseñada para facilitar la creación de aplicaciones de CBR (Bach & Althoff, 2012). Sus autores proponen a myCBR como una herramienta de naturaleza académica que permita aprender más sobre esta estrategia de inteligencia artificial, a la par que se desarrollan aplicaciones informáticas con facilidad. La arquitectura de myCBR se puede apreciar en la siguiente figura:

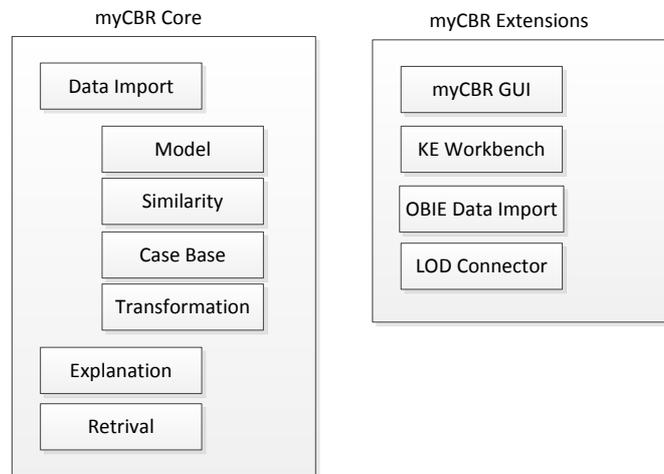


Figura 2: La arquitectura del proyecto de código libre myCBR

La extensión myCBR GUI es una aplicación con interfaz gráfica que permite al usuario crear proyectos para myCBR que contengan el modelo, la definición de los atributos del caso, la administración de la base de casos, la configuración de las funciones de similitud, la importación y exportación de datos desde y hacia archivos CSV, entre otras funciones (Zilles, 2009).

myCBR cuenta también con una API que pone a disposición del desarrollador Java las clases y métodos necesarios para llevar a cabo las consultas y obtener los resultados de casos similares

desde un proyecto creado con la herramienta myCBR GUI. El SDK de la API de myCBR se encuentra disponible en el sitio web del proyecto myCBR (<http://mycbr-project.org/download.html>) aunque cabe anotar que el versionado no ha sido actualizado desde mayo del 2015 (MyCBR Project, 2022). El sitio del proyecto myCBR cuenta con documentación, manuales y varios proyectos con código de ejemplo para orientar a los desarrolladores interesados en usar CBR en sus aplicaciones.

### **3.3. El servidor SpigotMC**

De acuerdo con su propio wiki, SpigotMC es la casa de la comunidad más grande de proyectos relativos al servidor de código abierto Spigot para Minecraft (Spigot, 2022). Cuenta con 300.000 miembros que de forma voluntaria contribuyen con código, diseños, ideas, manuales, soporte técnico, pruebas de concepto. El centro de la comunidad constituye sin duda el propio servidor Spigot, uno de los servidores más instalados en la red de Minecraft.

SpigotMC permite a los desarrolladores de videojuegos llevar a cabo sus proyectos de Minecraft de forma rápida y robusta. La metodología de programación preferida es la creación y mantenimiento de plugins, los cuales son compatibles con los clientes de Minecraft que se conecten a Spigot. De esta forma se puede tener control casi completo de los diferentes elementos del juego Minecraft. Los objetos del escenario, los materiales de construcción, los personajes, los enemigos, incluso se puede modificar las propias mecánicas originales de Minecraft.

Se debe anotar que la curva de aprendizaje que debe vencer un programador novato en Spigot es uno de los principales obstáculos. El lenguaje de programación es Java y muchos programadores, o ya lo conocen, o es relativamente fácil el cambiarse desde otro lenguaje como C#. Los diferentes entornos de desarrollo con los cuales se puede programar plugins para Spigot, ya plantean un problema: la dispersión. Hay varias opciones para configurar un entorno de programación, lo cual depende del nivel de conocimientos técnicos que se tenga. También tenemos la diversidad de versiones de librerías que se utilizan, lo cual conlleva una exigencia de pruebas para garantizar que no existan conflictos al momento de la instalación. Adicionalmente el programador se ha de acostumbrar a trabajar casi 100% desde el lado del código, ya que no existe un entorno visual estándar con el cual manipular y configurar los elementos del juego. La mayor parte del tiempo el programador se encontrará trabajando con código, dejando a tiempo de ejecución el ver los resultados visuales. El programador habrá de aprender a manejar los eventos y los ciclos de vida de las clases de Spigot, para crear sus programas, para saber interpretar los mensajes de error, y para realizar operaciones de debugger. Una vez superada esta primera fase de investigación y aprendizaje, podrá disfrutar del suficiente control sobre Minecraft como para empezar a desarrollar sus propios juegos.

### 3.4. El plugin Citizens 2

Citizens es un plugin que permite crear y controlar personajes NPCs de Minecraft mediante un código más organizado que desarrollarlo directamente en el servidor de Spigot (Citizens, 2022). Las acciones del NPC como caminar, subir escaleras, interactuar con puertas, cofres y otros objetos del escenario, son gestionadas mediante eventos que Citizens dirige y orquesta convenientemente para que el desarrollador solo tenga que colocar su código en el evento adecuado. Citizens también cuenta con funcionalidades que permiten que los NPCs tengan conversaciones con los jugadores, puedan modificar su aspecto, portar armas, escudo, y otro tipo de artefactos. Por esta razón el plugin de Citizens ha servido de base para la programación de otros plugins, como por ejemplo Sentinel (Sentinel, 2022). La ventaja de Citizens es la comunidad activa que lo mantiene, la cual aporta con nuevo código, con tutoriales y con soporte a los miembros.

### 3.5. Experiencias del uso del CBR en Videojuegos

Para realizar el presente proyecto, se buscaron referencias a artículos académicos que dieran cuenta del uso del CBR en el control de NPCs para videojuegos, y se encontraron los siguientes trabajos:

- El trabajo de grado doble de Fernando Domínguez para la Universidad Complutense de Madrid (Domínguez, 2017) en el cual se utiliza CBR para entrenar bots que puedan superar el rendimiento de un humano en el juego Pac-Man. La estrategia consiste en reemplazar un algoritmo Q-learning por una base de estados. El bot toma la decisión de hacia dónde moverse cuando llega a una esquina del mapa del juego, y lo hace en base de una descripción completa de la situación como la posición de cada fantasma, la posición de los corazones, entre otros.
- El proyecto de fin de carrera de Javier Olmos de la Universidad de Zaragoza (Olmos, 2011) que consiste en crear un personaje que siga un patrón de conducta lo más parecido a un humano, para lo cual utiliza CBR con la hipótesis de los marcadores somáticos. El videojuego elegido para aplicar CBR es Unreal Tournament 2004.
- El trabajo de fin de master de Francisco Domenech para la Universidad Complutense de Madrid (Domenech, 2011) que aplica CBR a un sistema de adaptación de las características del NPC para que se asemeje al comportamiento del jugador. Se ha utilizado como videojuego a "CUBE II saurbraten" y las características a controlar en el NPC son: la velocidad, el arma, la salud, la agresividad, la frecuencia de disparo.

De entre los trabajos citados se han tomado la idea usar una base de casos previamente diseñada y probada para controlar al NPC, poniendo énfasis en las acciones que representan la solución a cada situación del juego. No fue posible encontrar un artículo académico acerca de cómo aplicar CBR al juego de Minecraft propuesto en el presente trabajo.

### 3.6. El minijuego SpeedRun y su modelo CBR

SpeedRun es el minijuego que ha sido diseñado como prueba de concepto para el plugin cbrplugin del presente proyecto. Consiste en acarrear los ingredientes necesarios para crear productos atravesando un laberinto que separa los almacenes de la base de cada jugador. El jugador que logre el mayor número de productos será el que gane. En SpeedRun un jugador humano compite contra un NPC que está controlado por la inteligencia artificial basada en un modelo CBR, que se gestiona mediante el plugin cbrplugin.

Para diseñar los atributos que contiene un caso que representan una solución para el juego SpeedRun, se han tomado en cuenta los elementos significativos que determinan las acciones que puede llevar a cabo el NPC. Las acciones son:

- Ir de la base hacia uno de los dos almacenes disponibles
- Verificar el inventario del almacén una vez que llega a él.
- Tomar del inventario del almacén un ingrediente en la cantidad permitida
- Ir de vuelta desde el almacén a la base acarreando el ingrediente tomado
- Depositar el ingrediente en uno de los cofres del almacén

El motor CBR del plugin cbrplugin permite que el NPC lleve a cabo las acciones mediante la selección de un caso similar a la situación actual de los almacenes y de los cofres en la base. La definición de una solución está dada por las siguientes decisiones:

- Decidir qué producto se va a crear
- Decidir qué ingrediente se va a buscar
- Decidir a cuál almacén se dirige

Por ejemplo, una posible solución será la siguiente triada: Ir al almacén 1, para tomar 5 unidades del ingrediente leche, y volver a la base para depositarlos en el cofre para hacer pan.

En la siguiente tabla se puede apreciar el conjunto de atributos que definen un caso para el CBR del juego SpeedRun, en la primera columna, seguido por 9 casos – solución con sus respectivos valores:



A cada uno de estos casos le corresponde una determinada solución, que dictamina cómo se ha de conducir el NPC de manera que reaccione ante cualquier situación que se presente durante el juego. Los inventarios pueden cambiar, los productos en los cofres de la base también están sujetos a cambios, y estas situaciones específicas, en su gran mayoría no van a coincidir con alguno de estos casos, pero gracias al motor del cbrplugin, se podrá encontrar un caso similar, que determine a su vez el conjunto de acciones a seguir. En la siguiente tabla se listan las soluciones y las acciones a tomar que les corresponde a cada una:

Solución	Acciones a tomar
Caso 1	Ir al Almacen 1 y traer Material 1-1 y poner en el cofre 1
Caso 2	Ir al Almacen 1 y traer Material 1-3 y poner en el cofre 1
Caso 3	Ir al Almacen 1 y traer Material 1-1 y poner en el cofre 1
Caso 4	Ir al Almacen 2 y traer Material 2-3 y poner en el cofre 1
Caso 5	Ir al Almacen 1 y traer Material 1-2 y poner en el cofre 2
Caso 6	Ir al Almacen 2 y traer Material 2-4 y poner en el cofre 2
Caso 7	Ir al Almacen 2 y traer Material 2-1 y poner en el cofre 2
Caso 8	Ir al Almacen 1 y traer Material 1-2 y poner en el cofre 3
Caso 9	Ir al Almacen 2 y traer Material 2-4 y poner en el cofre 3
Caso 10	Ir al Almacen 2 y traer Material 2-3 y poner en el cofre 3
Caso 11	Ir al Almacen 2 y traer Material 2-2 y poner en el cofre 3
Caso 12	Ir al Almacen 1 y traer Material 1-2 y poner en el cofre 3
Caso 13	Ir al Almacen 2 y traer Material 2-4 y poner en el cofre 2

Tabla 3: Soluciones y sus correspondientes acciones del NPC

## 4. Diseño

El presente proyecto consta de dos partes: el plugin cbrplugin, y el plugin speedrun. El plugin cbrplugin está a cargo de implementar la librería de código abierto myCBR (MyCBR Project, 2022) para poner a disposición el modelo basado en casos que se ha desarrollado para el presente proyecto. El plugin speedrun está a cargo de probar el modelo de cbrplugin mediante un minijuego de Minecraft desarrollado para el servidor SpigotMC (<https://www.spigotmc.org/>) con el plugin de Citizens2 para la gestión de NPCs. En el siguiente diagrama se representan las partes que componen el presente proyecto:

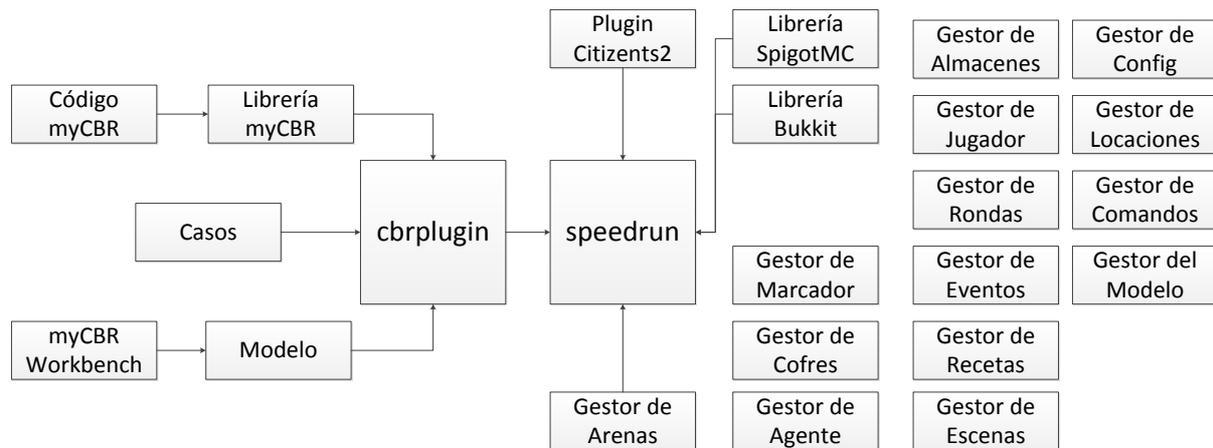


Figura 3: Diseño general de los plugins cbrplugin y SpeedRun

### 4.1. Diseño del plugin cbrplugin

La librería open-source MyCBR es una herramienta de recuperación basada en similitudes junto con un kit de desarrollo de software (SDK). Con myCBR Workbench se construyen los modelos y se los prueba, para luego integrar dicho modelo en aplicaciones gracias a MyCBR SDK. Los sistemas de recomendación basados en casos son solo un ejemplo de aplicaciones de recuperación basadas en similitudes. (<http://mycbr-project.org/background.html>)

El presente proyecto utiliza las siguientes APIs para implementar la base sobre la cuál elaborar el sistema del juego:

- MyCBR
  - La herramienta para construir el archivo del proyecto MyCBR
  - El archivo del proyecto MyCBR con el modelo CBR
  - El archivo CSV que contiene los casos acorde con el modelo
  - La API MyCBR para implementar las búsquedas de los casos

- SpigotMC
  - Se utiliza el servidor SpigotMC para implementar el plugin
  - Se utiliza el API de SpigotMC para interactuar con los cofres, los carteles y otros elementos de Minecraft
- Citizens2
  - Se utiliza la API de Citizens2 para implementar el agente mediante un NPC
  - Se utiliza el sistema de navegación para dirigir al NPC por el laberinto
  - Se utiliza el sistema de eventos de Citizens2 para implementar las mecánicas del juego

Para implementar el proyecto se ha dispuesto la construcción de dos plugins: el plugin con el CBR escrito para el servidor de Spigot, y el plugin con un juego que demuestra el uso del primer plugin.

- El plugin de CBR para SpigotMC
- El mini juego SpeedCraft
- El controlador del agente
- El controlador del jugador, similar al del agente.
- El desarrollo de un mini motor de videojuegos Minecraft

Las funcionalidades desarrolladas son las siguientes

- Crear el plugin CBR con los estándares para plugins de Spigot y realizar pruebas.
- Leer el archivo que contiene el modelo del proyecto MyCBR desde el sistema de archivos de Spigot.
- Leer el archivo CSV que contiene los casos del modelo CBR
- Los cofres deben ser modificados cuando el agente llega al cofre, tanto si es el almacén como si es en la base
- En la base siempre se están calculando las recetas cuando el agente modifica los cofres
- El juego podrá tener más de una arena, lo cual se parametrizará con un archivo plano, de acuerdo con el estándar de Spigot
- El juego permitirá escoger la arena en la que se llevará a cabo el juego.
- Cada arena contendrá toda la información del escenario, los contenidos de los cofres, la posición de los carteles informativos.

En las siguientes secciones se detallan las funcionalidades conjuntamente con los diagramas de clases que sirven para entender el código de ambos plugins desarrollados.

## 4.2. Arquitectura general de la aplicación

Para el plugin cbrplugin tenemos una arquitectura formada por la base de casos en formato CSV, el archivo del proyecto myCBR creado con la interfaz gráfica de dicha herramienta, el SDK de myCBR para leer tanto la base de casos como el archivo del proyecto, y el motor CBR que permite ejecutar las consultas y recuperar los casos similares.

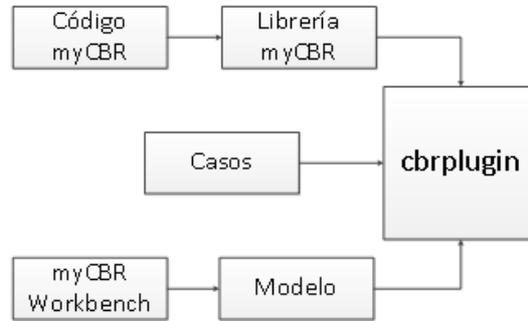


Figura 4: Arquitectura de cbrplugin

Con respecto al plugin SpeedRun la arquitectura cuenta con el plugin cbrplugin y el plugin Citizens 2; posee un gestor de arenas para controlar la entrada y salida desde el lobby del jugador, el control de los cofres tanto de los almacenes como de las bases, entre otros sistemas que permiten gestionar el juego a la par que se consulta al CBR sobre la estrategia a seguir por parte del NPC. A continuación un diagrama de la arquitectura de SpeedRun:



Figura 5: Arquitectura del plugin SpeedRun

Cabe anotar que el plugin cbrplugin es necesario para que el plugin SpeedRun pueda ejecutarse, ya que cuando el servidor SpigotMC da inicio su ejecución, se invoca al motor de CBR del plugin cbrplugin para leer el archivo del proyecto y el archivo CSV con la base de casos

### 4.3. Arquitectura de la información y diagramas de navegación

La arquitectura de clases del plugin cbrplugin se presenta a continuación:

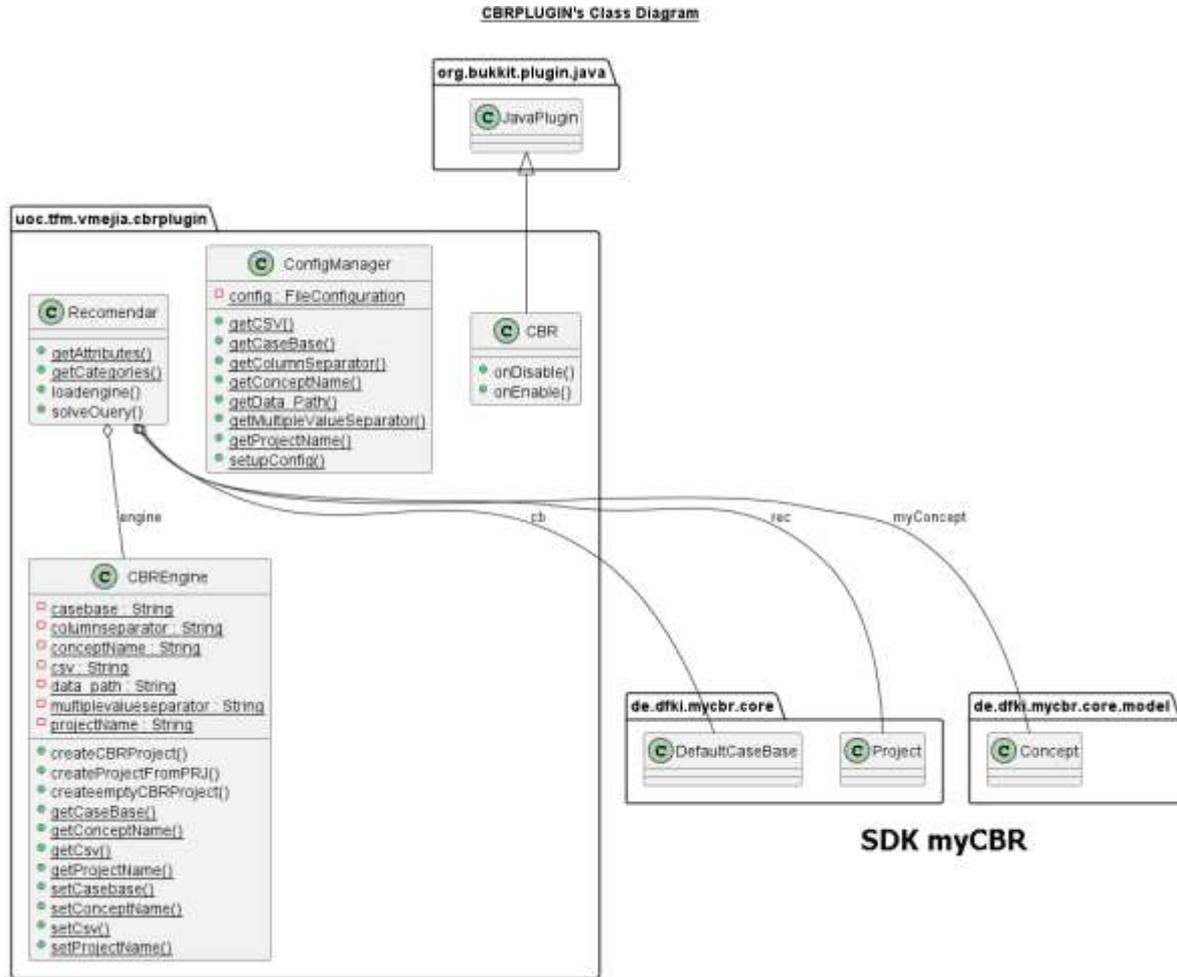


Figura 6: Diseño de Clases de cbrplugi

Para acceder a la base de casos y a la configuración del proyecto CBR se desarrolló un motor CBR a cargo de crear la consulta del caso recomendado. La clase Recomendar se encarga de realizar la recuperación del caso similar y la clase ConfigManager se encarga de leer el archivo de configuración para aplicarlos en el motor CBR.

La arquitectura de los eventos principales del plugin SpeedRun, que gestionan todas las llamadas al resto de componentes del sistema, son controladas mediante dos clases: la clase Events, a cargo de los eventos del personaje controlado por el jugador, y la clase MyTrait, a cargo de los eventos del agente controlado por el CBR. La clase SpeedRun se encarga de cargar la configuración desde el archivo YML, de colocar los valores iniciales de las arenas y de la escena, así como también de asociar al NPC con el controlador de eventos MyTrait. En el siguiente diagrama se puede apreciar los detalles de los métodos de cada una de las clases base de SpeedRun:

**SPEEDRUN's Class Diagram**

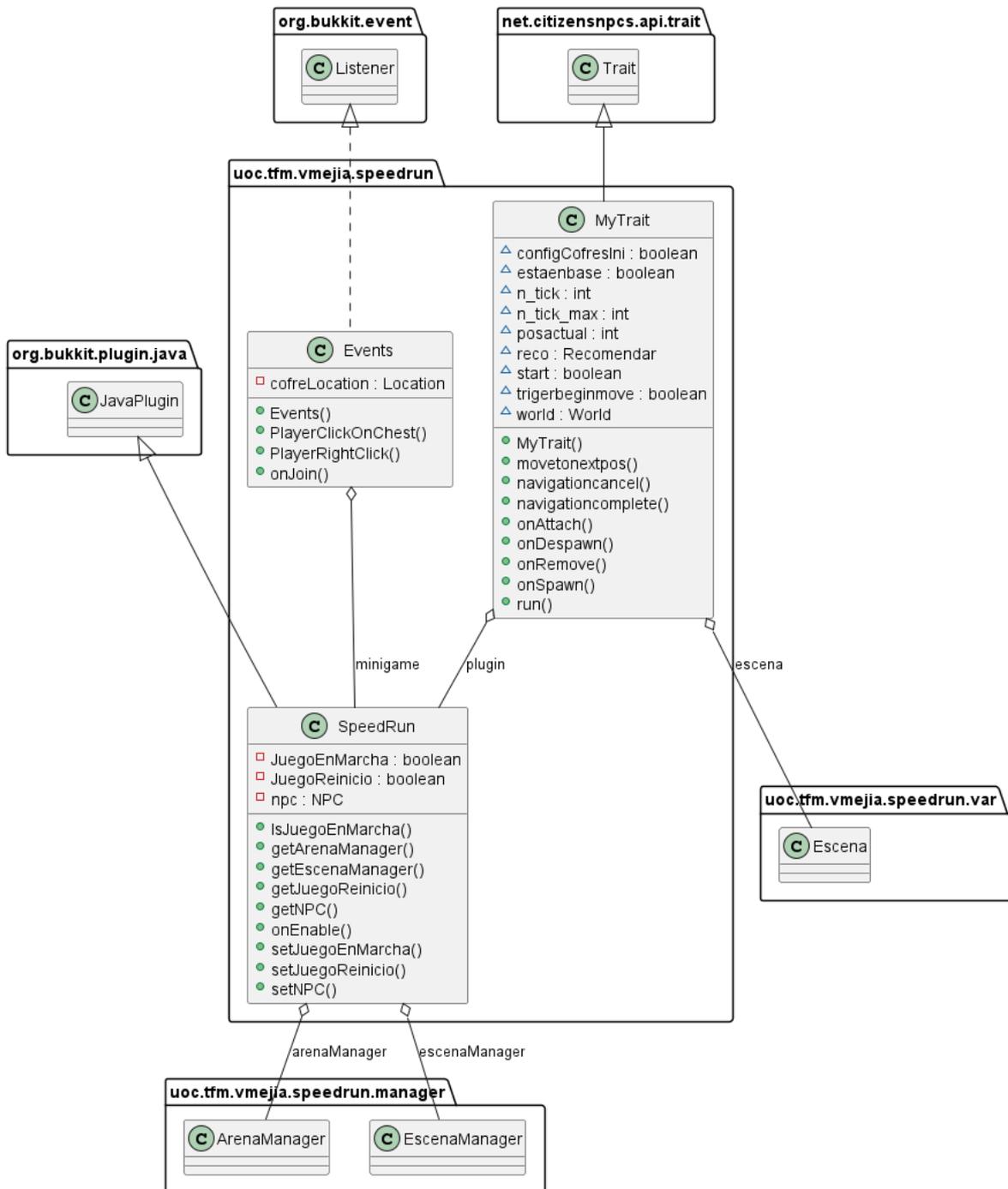


Figura 7: Control de eventos del jugador y del NPC

Para controlar el movimiento del NPC, así como sus interacciones con los diferentes elementos del juego, se ha programado un conjunto de clases que representen a todos los objetos presentes en la escena; así tenemos una clase para representar a la propia Escena, la laces para representar al almacén de los ingredientes, la clase de los materiales utilizados para representar las recetas y los productos, una clase especial para el agente, es decir, el NPC; otra clase para representar la base y

sus cofres, entre otras. A continuación se pueden encontrar todas las abstracciones que se utilizan para gestionar los elementos con los cuales interactúa el NPC:

**VAR's Class Diagram**

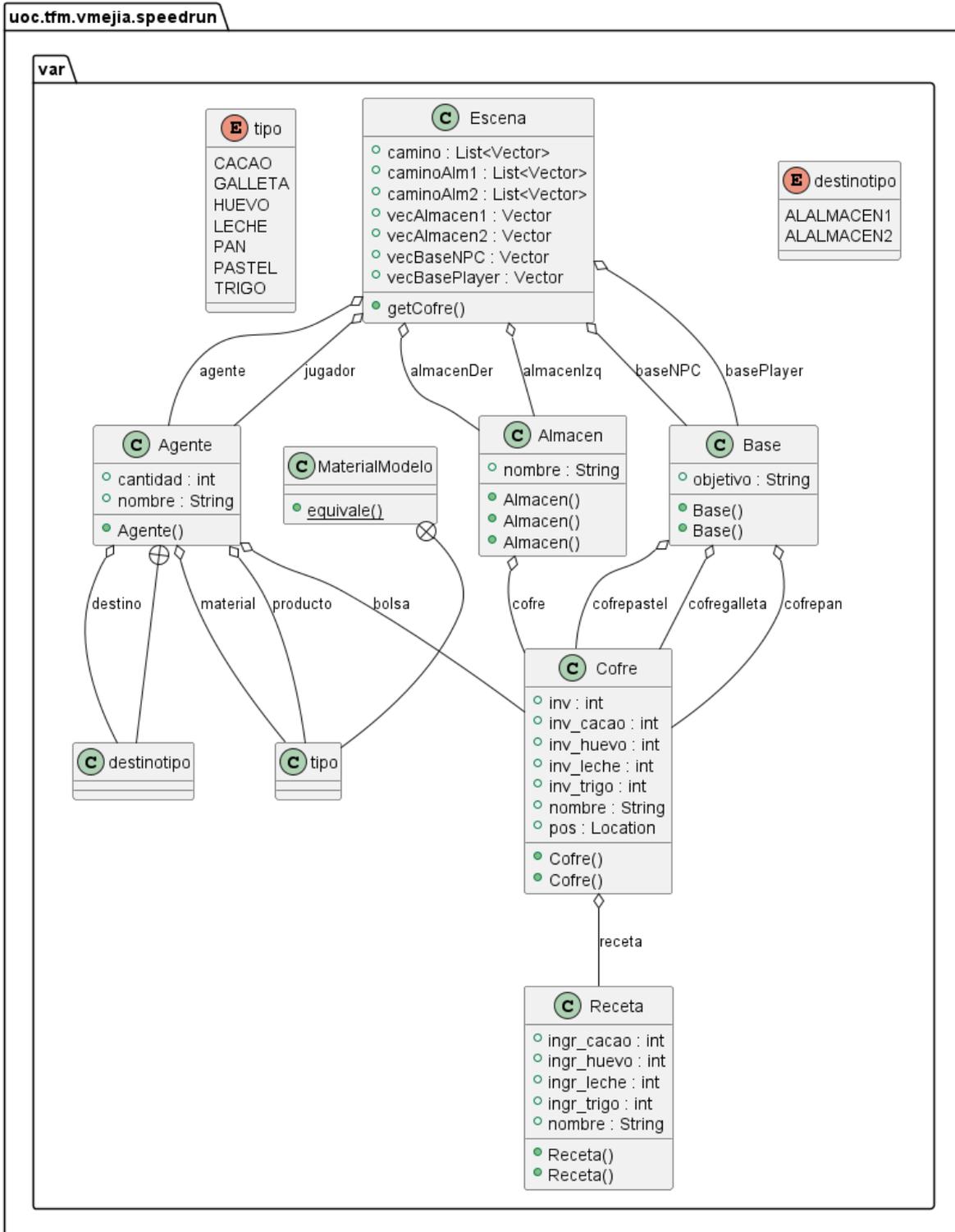


Figura 8: Clases para representar el escenario y sus elementos

Muchas de las tareas de seguimiento y control de la ejecución del juego requieren que las clases del escenario sean transformadas en texto para registrar sus contenidos en el archivo del log de ejecución. Por este motivo se han creado una serie de clases utilitarias que lleven a cabo esa tarea:

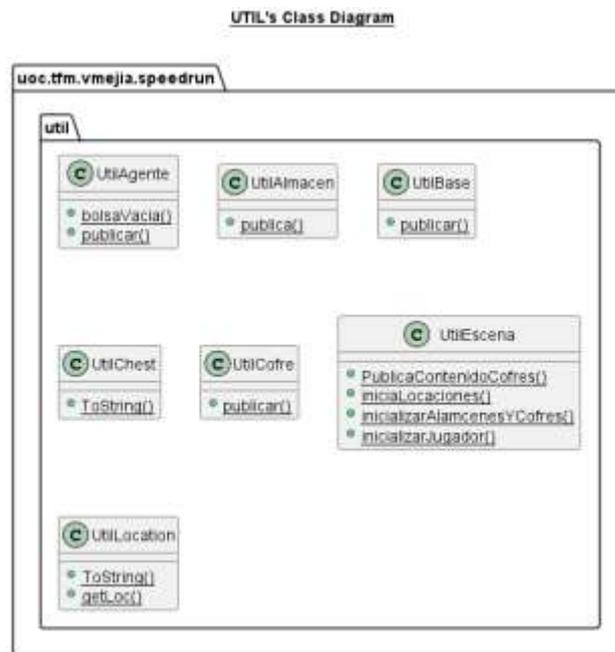


Figura 9: Clases Util para generar los contenidos del archivo log

Las tareas del control de lo que ocurre en el escenario, del control de la arena y de la recuperación de los valores que contiene el archivo de configuración, se llevan a cabo en el paquete Manager, como se puede ver a continuación:

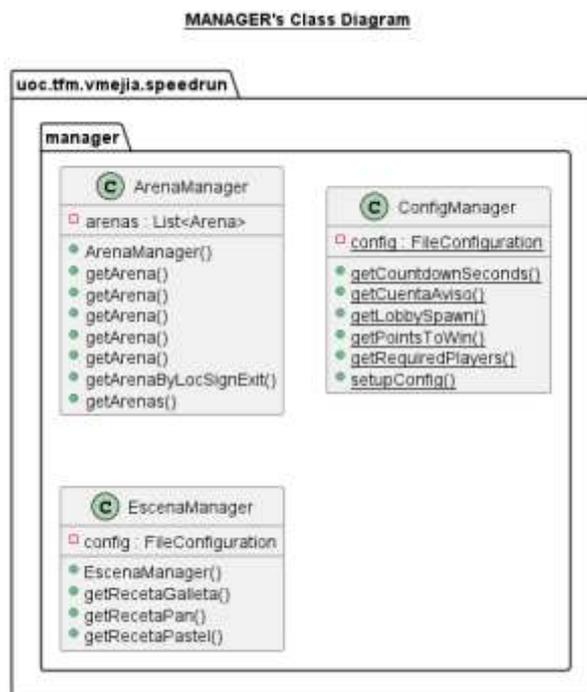


Figura 10: Clases del paquete Manager

Para responder ante las acciones del juego, como cuando se crea un nuevo producto y se debe actualizar el marcador, o cuando se selecciona uno de los carteles para entrar en la arena, o cuando se empieza a cargar el mundo para dar inicio a una nueva ronda, se han programado dos clases a cargo de mantenerse en la escucha de dichos eventos, aquí el detalle:

**LISTENER's Class Diagram**

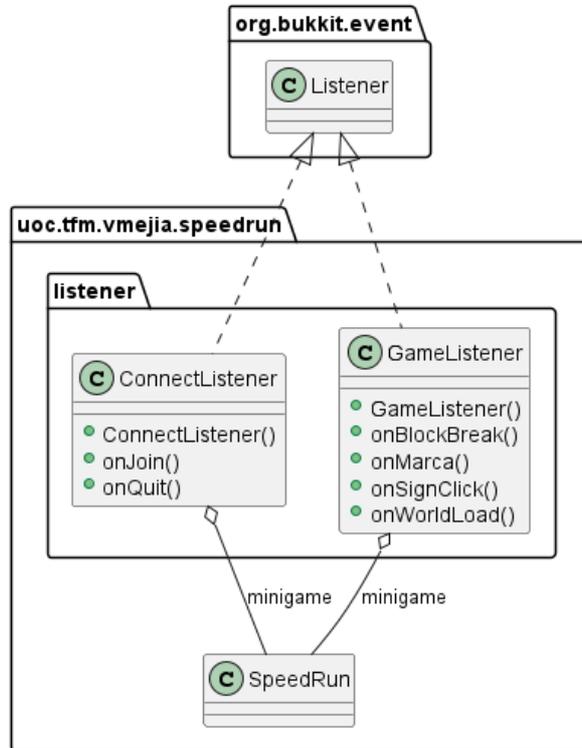


Figura 11: Clases ConnectListener y GameListener

La clase Arena se encarga de mantener un control sobre las posiciones y contenidos de los almacenes, de los cofres de la base y de los jugadores que ingresan a la nueva ronda. Permite añadir el jugador a la arena, configurar la arena antes de que los jugadores empiecen la partida, determinar la configuración inicial y la distribución de los diferentes elementos en el escenario, colocar los inventarios iniciales de cada uno de los almacenes, controlar el manejo de los estados del juego, y si el jugador sale de la arena, recabar los contenidos de almacenes y cofres para determinar la situación actual del juego, entre otras tareas menores. Para realizar su trabajo la clase arena dispone de la clase GUI, que está a cargo de publicar el marcador que se presenta al jugador en la interfaz de usuario en todo momento. La clase Countdown permite que se presente al jugador un cuenta atrás previo al inicio del juego. A continuación el diagrama de la clase Arena:

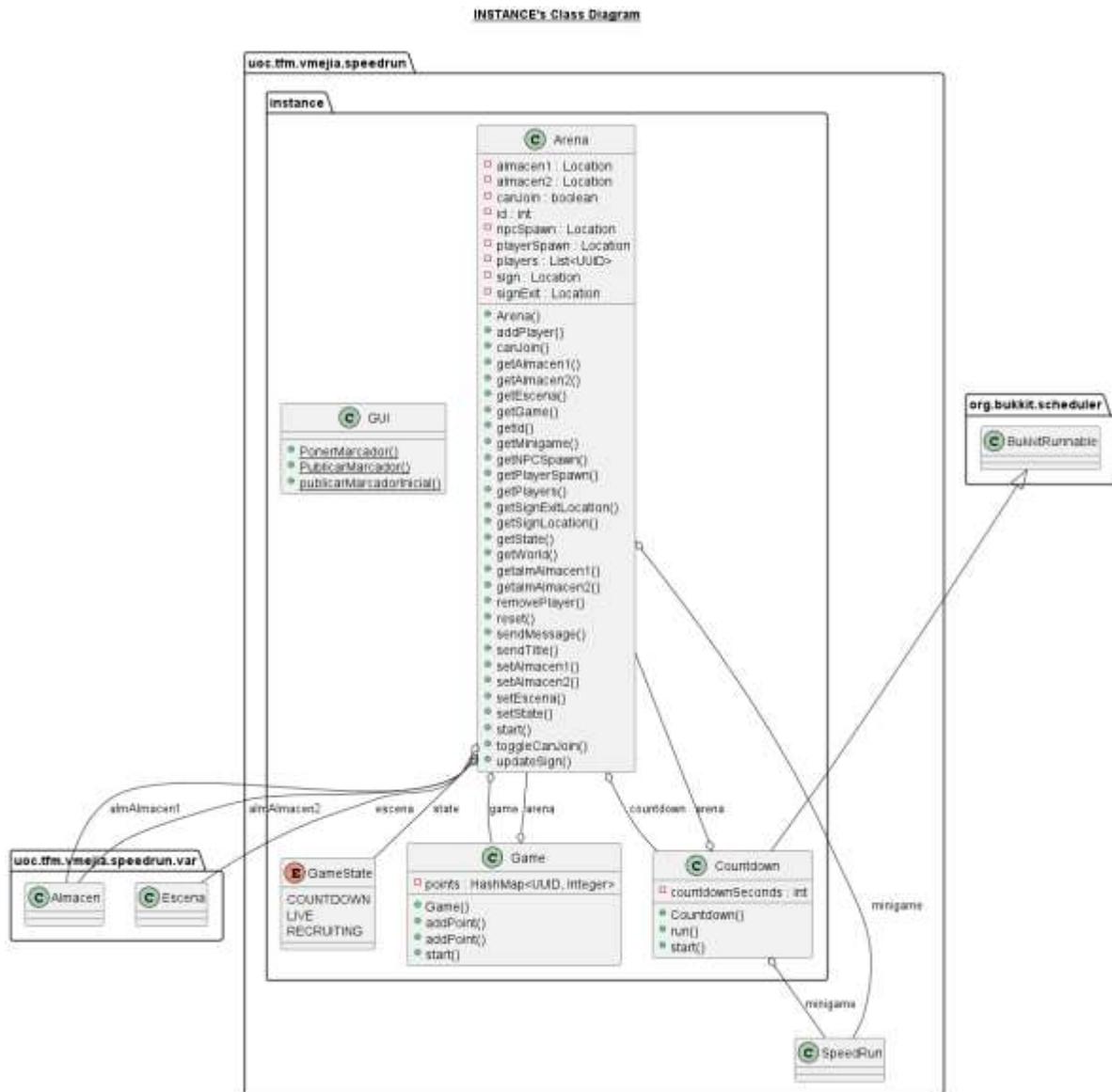


Figura 12: Clase Arena y Clase GUI

Para poder realizar los cálculos de cuántos productos se pueden crear cuando el NPC y el jugador depositan los ingredientes en los cofres de sus respectivas bases, se han diseñado tres clases que pueden obtener dichos valores a partir de las recetas que están definidas en el archivo YML de configuración del plugin SpeedRun. El cálculo se realiza de forma que se procesa un tipo de producto a la vez, y en caso de presentarse excedentes de ingredientes después del primer ciclo, se vuelve a repetir el proceso de cálculo hasta que las cantidades restantes no puedan ser utilizadas por ninguna de las recetas. A continuación el diagrama de las clases de control para el Agente o NPC, la base y el cofre:

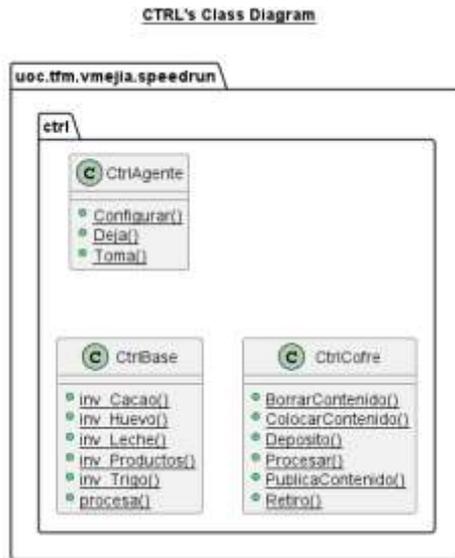


Figura 13: Clases para la elaboración de productos

La acciones de la arena que puede realizar el jugador mediante la interfaz gráfica, como entrar en una arena, listar las arenas que están disponibles, salir de la arena, están también disponibles mediante comandos que el administrador puede ingresar en el juego. La clase ArenaCommand captura de la línea de comandos el requerimiento del usuario y luego de una validación, ejecuta dichos comandos

**COMMAND's Class Diagram**

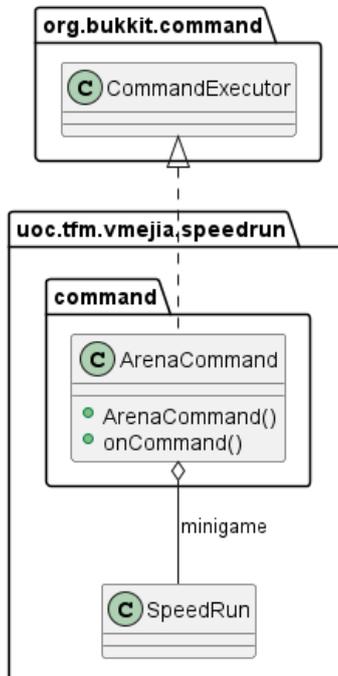


Figura 14: Clase para gestionar la línea de comandos

El paquete cbr se ha programado para que sea la interfaz entre el juego de SpeedRun y el plugin cbrplugin. Se encarga de tomar los valores del inventario de los almacenes, de los cofres de la base del NPC y de los productos que se han generado, para construir apropiadamente la consulta para el motor CBR que tiene cbrplugin, y recibir como respuesta el caso similar que luego se encarga de analizar para obtener la solución que debe aplicar al juego. A continuación el diagrama del paquete cbr:

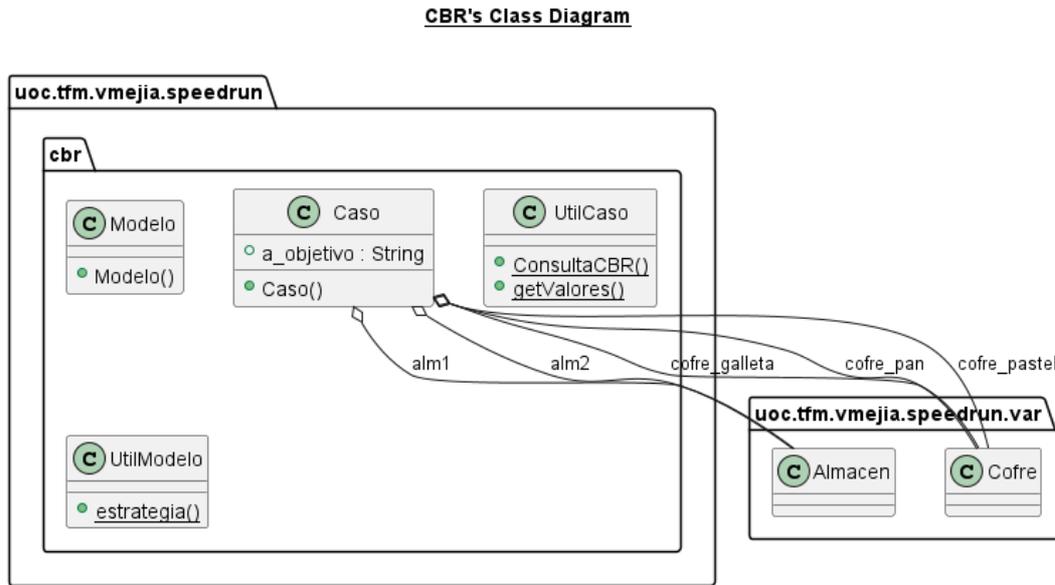


Figura 15: Paquete cbr interfaz con cbrplugin

#### 4.4. Diseño gráfico e interfaces

Para el plugin cbrplugin no se ha diseñado ninguna interfaz gráfica, ya que su configuración se realiza mediante un archivo YML, el cual tiene la siguiente estructura:

data\_path: El directorio de trabajo del CBR, ejemplo: 'C:\UOC\S4\TFM\uoc\_tfm\cbr\modelo'  
 projectName: El nombre del archive con el proyecto myCBR, ejemplo: 'speedcraft.prj'  
 conceptName: El nombre del concepto del Proyecto, ejemplo: 'speedcraft\_caso'  
 csv: El nombre del archive CSV con la base de casos, ejemplo: 'speedcraft-casos.csv'  
 casebase: El nombre del caso base, ejemplo: 'CB\_csvImport'  
 columnseparator: El separador de columnas del archivo CSV, ejemplo: ';'   
 multiplevalueseparator: El separador de valores del archivo CSV, ejemplo: ','

Para el plugin SpeedRun se han diseñado las siguientes interfaces gráficas:

**Pantalla de carga**, que se presenta cuando el jugador se conecta al servidor Spigot, y tiene la siguiente apariencia:



Figura 16: Pantalla de inicio en el servidor de SpeedRun

Consta del título del juego, un mensaje para solicitar ayuda, y un panel derecho para registrar los marcadores tanto del jugador como del NPC durante una partida. La ayuda se ha diseñado para que se presente en la parte superior del juego, en cualquier momento que el jugador presione la tecla TAB, así:

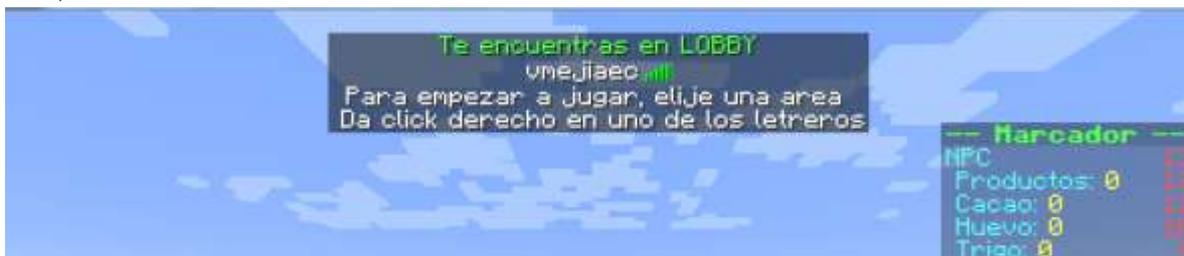


Figura 17: Barra de ayuda para empezar una partida

El jugador al conectarse al servidor es colocado en el lobby del juego, a la espera que elija una de las tres arenas disponibles. Se ha optado por utilizar los letreros de madera para que el jugador pueda elegir su destino, como se puede apreciar en la siguiente imagen:



Figura 18: Letrero asociado a la acción de inicio de la arena

Se ha diseñado una pantalla de conteo hacia atrás con el mensaje de inicio para cuando el jugador entre a la área, así:



Figura 19: Cuenta atrás e inicio del juego en la arena seleccionada

El diseño de cada arena sigue el mismo patrón: contiene un laberinto que separa los cofres almacén de los cofres de las bases. A continuación se presenta el diseño de la arena 1:



Figura 20: Diseño básico de una arena de juego

El diseño de las tres arenas de juego se puede apreciar en la siguiente figura:

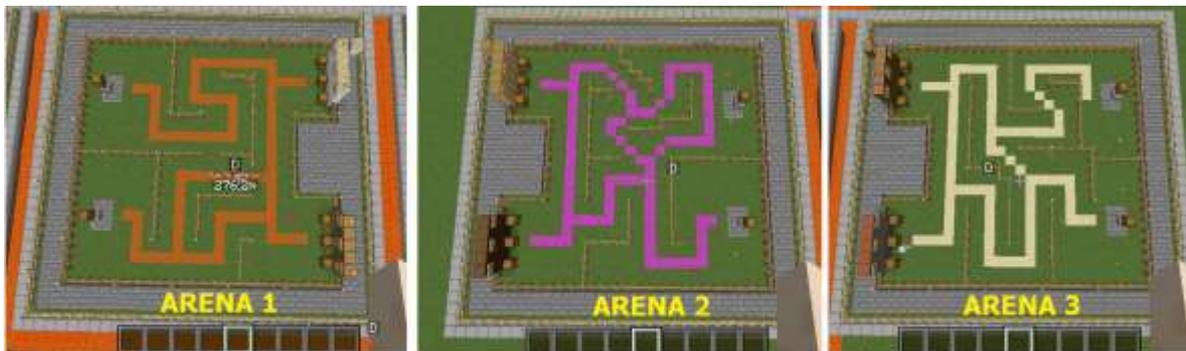


Figura 21: Diseño de las tres arenas del juego

Para cuando el jugador abre un cofre del almacén, se ha diseñado la disposición de los ingredientes de tal forma que se puedan tomar rápidamente y con solo un click. Así se presenta el cofre del almacén cuando es abierto por el jugador antes y después de elegir un ingrediente:



Figura 22: Interfaz del cofre tipo almacén

Para el diseño de cada base se han utilizado letreros y cuadros que informen sobre los productos que se pueden elaborar cuando el jugador deposite los ingredientes en los cofres correspondientes. A continuación un ejemplo del diseño de la base:



Figura 23: Diseño de la base del NPC para crear productos

Se ha diseñado un mecanismo para salir de la arena y regresar al lobby sin tener que esperar al final de la partida, mediante un letrero que se encuentra al costado y atrás de la base del jugador, tal como se muestra en la siguiente figura:



Figura 24: Diseño del letrero para salir de la arena

El diseño de la pantalla para cuando el jugador sale de la arena o es derrotado por el NPC y regresa al lobby se puede apreciar en la siguiente imagen:



Figura 25: Diseño de la pantalla de Game Over

El diseño de la pantalla para cuando el jugador gana la partida y vuelve al lobby, consta de un mensaje de triunfo y la animación de fuegos artificiales, así:



Figura 26: Diseño de la pantalla WIN del jugador

Para controlar los elementos del escenario del juego se ha diseñado un archivo YML de configuración que contiene toda la información necesaria para modificar el comportamiento del juego. A continuación se explican los parámetros de la primera sección del archivo de configuración del plugin SpeedRun:

- countdown-seconds: Duración del cuenta atrás para iniciar el juego (1)
- cuenta-aviso: Intervalo de los mensaje de inicio del juego (3)
- points-to-win: Número de productos a crear para ganar (5)
- player-cant-min-toma: Lote de ingredientes a tomar por el jugador (3)
- npc-cant-min-toma: Lote de ingredientes a tomar por el NPC (5)

La siguiente sección del archivo de configuración corresponde a la composición de cada una de las recetas que se usan para generar los productos en las bases:

- receta-pan: La receta para producir el producto pan
  - nombre: Nombre del producto a crear ('pan')
  - inv\_cacao: Cantidad de cacao (0)
  - inv\_huevo: Cantidad de huevo (2)
  - inv\_leche: Cantidad de leche (0)
  - inv\_trigo: Cantidad de trigo (3)

- receta-galleta: La receta para producir el producto trigo
  - nombre: Nombre del producto a crear ('galleta')
  - inv\_cacao: Cantidad de cacao (1)
  - inv\_huevo: Cantidad de huevo (1)
  - inv\_leche: Cantidad de leche (1)
  - inv\_trigo: Cantidad de trigo (2)

- receta-pastel: La receta para producir el producto pastel
  - nombre: Nombre del producto a crear ('pastel')
  - inv\_cacao: Cantidad de cacao (3)
  - inv\_huevo: Cantidad de huevo (2)
  - inv\_leche: Cantidad de leche (1)

inv\_trigo: Cantidad de trigo (0)

En la siguiente sección se configura la posición del lobby así como la orientación con la cual el jugador será localizado al conectarse al servidor

```
lobby-spawn:  
world: 'world'  
x: 163  
y: 9  
z: -125  
yaw: 0  
pitch: 0
```

En la sección de arenas se configura la posición del punto de spawn del jugador, así como también del agente NPC. Para cada arena se configura también la posición de los almacenes; cada almacén cuenta con un inventario inicial de cada uno de los ingredientes del juego. Adicionalmente se cuenta con la posición de los letreros asociados a los puntos de entrada y salida de la arena en cuestión:

arenas:

0: Valores de configuración para la arena 1

```
world: 'world'
```

```
sign:
```

```
x: 165
```

```
y: 9
```

```
z: -123
```

```
sign-exit:
```

```
x: 149
```

```
y: 5
```

```
z: -86
```

```
player-spawn: Punto de aparición del jugador
```

```
x: 152
```

```
y: 4
```

```
z: -89
```

```
yaw: -90
```

```
pitch: 0
```

```
npc-spawn: Punto de aparición del agente NPC
```

```
x: 152
```

```
y: 4
```

```
z: -105
```

```
almacen1: Posición del almacén e inventario inicial de ingredientes
```

```
x: 172
```

```
y: 4
```

```
z: -103
```

```
inv_cacao: 25
```

```
inv_huevo: 25
```

```
inv_leche: 35
```

```
inv_trigo: 30
```

```
almacen2: Posición del almacén e inventario inicial de ingredientes
```

```
x: 172
```

y: 4  
z: -90  
inv\_cacao: 15  
inv\_huevo: 15  
inv\_leche: 20  
inv\_trigo: 10  
npc-base:  
objetivo: 'pan'

1: Valores de configuración para la arena 2

world: 'world'

sign:

x: 163

y: 9

z: -123

sign-exit:

x: 140

y: 5

z: -86

player-spawn: Punto de aparición del jugador

x: 137

y: 4

z: -89

yaw: 90

pitch: 0

npc-spawn: Punto de aparición del agente NPC

x: 137

y: 4

z: -105

almacen1: Posición del almacén e inventario inicial de ingredientes

x: 117

y: 4

z: -90

inv\_cacao: 25

inv\_huevo: 25

inv\_leche: 35

inv\_trigo: 30

almacen2: Posición del almacén e inventario inicial de ingredientes

x: 117

y: 4

z: -103

inv\_cacao: 15

inv\_huevo: 15

inv\_leche: 20

inv\_trigo: 10

npc-base:

objetivo: 'pan'

2: Valores de configuración para la arena 3

world: 'world'

sign:  
x: 161  
y: 9  
z: -123  
sign-exit:  
x: 212  
y: 5  
z: -86  
player-spawn: Punto de aparición del jugador  
x: 209  
y: 4  
z: -89  
yaw: 90  
pitch: 0  
npc-spawn: Punto de aparición del agente NPC  
x: 209  
y: 4  
z: -105  
almacen1: Posición del almacén e inventario inicial de ingredientes  
x: 189  
y: 4  
z: -91  
inv\_cacao: 25  
inv\_huevo: 25  
inv\_leche: 35  
inv\_trigo: 30  
almacen2: Posición del almacén e inventario inicial de ingredientes  
x: 189  
y: 4  
z: -103  
inv\_cacao: 15  
inv\_huevo: 15  
inv\_leche: 20  
inv\_trigo: 10  
npc-base:  
objetivo: 'pan'

El resto de diseños se han reutilizado del propio juego Minecraft, como son los sonidos de entrada y salida de la arena, de producción de materiales, de apertura y cierre de cofres, de ganancia y celebración del triunfo, entre otros.

#### **4.5. Estilos**

La elección de realizar un proyecto de investigación acerca de CBR, así como también del minijuego SpeedRun que lo prueba, conlleva asumir como propia la línea gráfica de Minecraft. De esta forma se utilizan sus logotipos, anagramas, paleta de colores, tipografías, fuentes, estilos y demás elementos gráficos. La ventaja de esta elección, que comparte muchos plugins desarrollados por la comunidad del servidor SpigotMC, es la inversión de tiempo y recursos a la implementación de las mecánicas del juego.

## **4.6. Usabilidad /UX**

Dado que en el presente proyecto se han creado dos plugins para el servidor SpigotMC, se han heredado del juego Minecraft sus mismos principios y técnicas de usabilidad y experiencia de usuario. La forma de interacción durante el juego, por medio del botón derecho del mouse, es el mismo que se utiliza en Minecraft para abrir los cofres, tomar y dejar items, o seleccionar una acción mediante un letrero.

## **4.7. Lenguajes de programación y APIs utilizados**

Puesto que los plugins desarrollados fueron diseñados para ejecutarse en el servidor SpigotMC versión 1.15, se utilizó como lenguaje de programación a Java versión 17.0.2, con el entorno de desarrollo IntelliJ IDEA Educational Edition 2021.3.2 el cual soporta la gestión de la compilación y generación de librerías con MAVEN. El entorno de IntelliJ IDEA fue seleccionado en lugar del clásico entorno de Eclipse porque en la documentación disponible en la comunidad hay más tutoriales utilizando IntelliJ IDEA en lugar de Eclipse o cualquier otro entorno de desarrollo Java.

Para crear el plugin cbrplugin, se utilizó el SDK del proyecto myCBR, cuya API permite realizar las consultas de casos definidos en proyectos CBR realizados en su entorno gráfico. Para la creación del plugin speedrun se utilizó el plugin de Citizens versión 2, y evidentemente el plugin cbrplugin para acceder al modelo desarrollado para controlar al NPC que enfrenta al jugador.

Como parte del cliente Minecraft se optó por utilizar Forge versión 1.15 junto con el lanzador TLauncher versión 2.481, lo cual permite emular el acceso al juego sin la necesidad de poseer una cuenta de pago Minecraft. Dado que la investigación tiene un carácter académico, es posible aprovechar de estas herramientas gratuitas para llevar a cabo las pruebas de los plugins desarrollados. En un ambiente de consumo, lo recomendable es utilizar directamente Minecraft con una cuenta de pago.

## 5. Implementación

### 5.1. Requisitos de instalación

Para replicar los resultados obtenidos en el presente proyecto de investigación, es necesario cubrir con los siguientes requisitos:

**Hardware:** Se requiere de un computador con al menos un procesador equivalente a Core i3, preferible Core i5, memoria RAM de 16 GB; este requisito es muy importante por la cantidad de aplicaciones que se ejecutan simultáneamente. Un espacio disponible en disco de 300 GB, una pantalla de 13 pulgadas. Una tarjeta de video con 4 GM de memoria, es importante por el consumo de la gráfica por parte del cliente de Minecraft

**Software:** Se quiere del sistema operativo Windows 7, o más. Se requiere de la versión de Java 17.0.2. El entorno de desarrollo IntelliJ IDEA Educational Edition 2021.3.2. Servidor Spigot versión 1.15.2. Plugin Citizens versión 2.0.28. El SDK de myCBR versión 3.0 BETA. El juego Minecraft versión 1.15.2. La aplicación Forge 1.15.2. El lanzador del cliente de Minecraft TLauncher versión 2.481. Configurar el lanzador TLauncher para ejecutar Forge 1.15.2 asociado a la cuenta de Minecraft.

**Formación y conocimiento:** Se requiere conocimientos de MAVEN, configuración, generación y ejecución. Configuración de archivos YML. Como importar librerías Java y configurarlas en el entorno de IntelliJ IDEA. Configuración de IntelliJ IDEA para realizar el proceso de debugger del servidor SpigotMC. Configuración del plugin Citizens 2 para el servidor SpigotMC.

### 5.2. Instrucciones de instalación

Para replicar los resultados obtenidos en el presente proyecto de investigación, es necesario realizar los siguientes pasos:

- Confirmar que se dispone de los requisitos detallados en el punto anterior
- Descargar el proyecto uoc\_tfm del repositorio: [https://github.com/vmejiaec/uoc\\_tfm](https://github.com/vmejiaec/uoc_tfm) el código. Consta de los siguientes directorios:

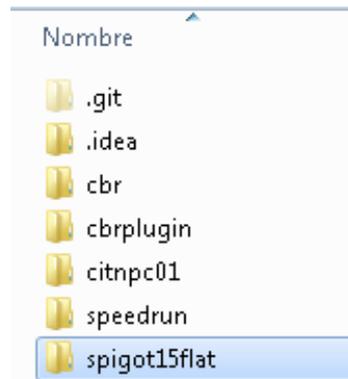


Figura 27: Lista de directorios del proyecto

- Ingresar al directorio spigot15flat y abrir el archivo config.yml del directorio plugins/CbrPlugin. En la primera línea, luego de la etiqueta data\_path, colocar la dirección completa del directorio modelo que se corresponde con el lugar donde se descargó el código del repositorio.

```

1  data_path: 'C:\UOC\S4\TFM\uoc_tfm\cbr\modelo\'
2  projectName: 'speedcraft.prj'
3  conceptName: 'speedcraft_caso'
4  csv: 'speedcraft-casos.csv'
5  casebase: 'CB_csvImport'
6  columnseparator: ';'
7  multiplevalueseparator: ','
    
```

Figura 28: Archivo YML de configuración de cbrplugin

- Ejecutar el archivo start.bat del directorio spigot15flat y verificar que arranca el servidor SpigotMC junto con los plugins cbrplugin y speedrun, tal como se ve en la siguiente imagen:

```

[22:31:51] [Server thread/INFO]: [CbrPlugin] Enabling CbrPlugin v1.0
[22:31:51] [Server thread/INFO]: [speedrun] Enabling speedrun v1.0
[22:31:51] [Server thread/INFO]: [speedrun] Loaded class net.citizensnpcs.api.CitizensAPI from Citizens v2.8.29-SNAPSHOT
[22:31:51] [Server thread/INFO]: [speedrun] Loaded class net.citizensnpcs.api.CitizensAPI from Citizens v2.8.29-SNAPSHOT
[22:31:51] [Server thread/INFO]: CitizensAPI el trait es: mytraitname
[22:31:51] [Server thread/INFO]: Crear ArenaManager
[22:31:51] [Server thread/INFO]: - World: 0 - world
[22:31:51] [Server thread/INFO]: -- arena: uoc.tfm.unejia.speedrun.instance.ArenaF73683ad8
[22:31:51] [Server thread/INFO]: -- arena.minigame: null
[22:31:51] [Server thread/INFO]: - World: 1 - world
[22:31:51] [Server thread/INFO]: -- arena: uoc.tfm.unejia.speedrun.instance.ArenaP11a3218
[22:31:51] [Server thread/INFO]: -- arena.minigame: null
[22:31:51] [Server thread/INFO]: - World: 2 - world
[22:31:51] [Server thread/INFO]: -- arena: uoc.tfm.unejia.speedrun.instance.ArenaB374869bf
[22:31:51] [Server thread/INFO]: -- arena.minigame: null
[22:31:51] [Server thread/INFO]: Server permissions file permissions.yml is empty, ignoring it
[22:31:51] [Server thread/INFO]: Done (2.541s)! For help, type "?help"
[22:31:51] [Server thread/INFO]: -- Inicializa el CBR con los casos
[22:31:51] [Server thread/INFO]: C:\NOC\S4\TFM\uoc_tfm\cbr\modelo\speedcraft.prj
[22:31:51] [Server thread/INFO]: Loading file C:\NOC\S4\TFM\uoc_tfm\cbr\modelo\speedcraft.prj
[22:31:51] [Thread-9/INFO]: Imported 13 instances.
[22:31:52] [Server thread/INFO]:
[22:31:52] [Server thread/INFO]: -- Coloca al agente en la base inicial
[22:31:52] [Server thread/INFO]: El agente NPC es colocado en escena -----(*)
[22:31:52] [Server thread/INFO]: -- El npc se llama: npcVictor
[22:31:52] [Server thread/INFO]: [Citizens] Cargando a 1 NPCs.
    
```

Figura 29: Entradas del archivo log del servidor SpigotMC

- Iniciar el cliente Minecraft mediante el lanzador TLauncher y arrancar el perfil del usuario con Forge.
- Una vez que Minecraft se haya ejecutado, seleccionar la opción de multijugador y configurar el servidor con el IP del equipo local.
- Cuando el servidor esté disponible en la pantalla de multijugador de Minecraft, seleccionar para dar inicio al juego. Si al empezar aparece la pantalla con el nombre del juego SPEEDRUN entonces se confirmará que la instalación ha sido exitosa y que se ha reproducido el presente proyecto.

## 6. Demostración

### 6.1. Instrucciones

Para reproducir los resultados presentados en esta investigación, se deben seguir los siguientes pasos:

- Definir los atributos que constituyen un caso, y verificar que dichos atributos reflejan la situación del problema que se presenta ante el NPC en el juego
- Definir un conjunto de casos solución que permitan determinar una secuencia de acciones que debe realizar el NPC para desenvolverse en el juego
- Crear un proyecto utilizando la herramienta gráfica de myCBR, configurar las medidas del proyecto y el método para calcular la similitud entre casos.
- Crear un archivo CSV con la base de casos solución que abarque los problemas representativos que afrontará el NPC en el juego
- Configurar el plugin cbrplugin para que pueda leer el archivo del proyecto myCBR y el archivo CSV con la base de casos. Utilizar el archivo YML asociado al plugin.
- Utilizar el plugin cbrplugin para crear un plugin para el servidor SpigotMC que implemente el juego y las acciones del NPC dentro de dicho juego, así como la interfaz con el plugin cbrplugin. Para el presente proyecto ese juego es SpeenRun
- Instalar los plugins adicionales que se han utilizado en la creación del juego y configurarlos convenientemente. En el presente proyecto se utilizó el plugin Citizens 2.
- Ejecutar el servidor SpigotMC, ingresar con el cliente de Minecraft y recolectar en un archivo log la ejecución de las acciones del NPC para su análisis posterior. Fruto de dicho análisis se determinarán los cambios o añadidos que se realizarán en la base de casos o en los atributos del proyecto CBR.

## 7. Conclusiones y líneas de futuro

### 7.1. Conclusiones

Del trabajo realizado durante cada una de las etapas del proyecto se infieren las siguientes conclusiones:

- La estrategia de inteligencia artificial CBR representa una opción válida para aportar con comportamientos significativos a los NPCs de un videojuego. El desempeño del NPC durante el juego depende directamente de la base de casos utilizada, y es susceptible de mejoras continuas que se pueden llevar a cabo sin modificar al sistema.
- La herramienta de código abierto myCBR fue un acierto en cuanto al tiempo que tomó crear el proyecto y la base de casos iniciales. De igual manera su SDK y API permitieron interactuar con dichos archivos de una forma limpia y confiable. El tiempo invertido para implementar myCBR en el plugin cbrplugin fue muy bien recompensado si tomamos en cuenta el resultado obtenido.
- El servidor SpigotMC es una aplicación con una curva de aprendizaje alta para los estándares de un programador novato como es el caso del autor. Mucho del trabajo del presente proyecto fue invertido en la indagación acerca de los paquetes, clases, eventos y demás elementos de la arquitectura de SpigotMC, con el fin de solventar mensajes de error o problemas en la ejecución de las primeras versiones del código.
- El ambiente de desarrollo IntelliJ IDEA en su versión educacional prestó las facilidades adecuadas para configurar los archivos MAVEN y controlar la compilación, ejecución y generación de los plugins tal como los requiere el servidor SpigotMC.
- Los videojuegos similares a Minecraft, que cuentan con una amplia comunidad de desarrolladores de plugins, son ideales para llevar a cabo pruebas de conceptos como los desarrollados en la presente investigación. La reutilización de los elementos gráficos del juego, sus interfaces y mecánicas, agilizaron la puesta a prueba del proyecto CBR.
- Los objetivos planteados inicialmente, esto es, crear el plugin cbrplugin para aplicar CBR a un juego, en este caso SpeedRun, se cumplió en casi su totalidad. La investigación realizada dio cuenta de los conocimientos acerca de la metodología, de las herramientas de desarrollo, de las técnicas de programación, y al solventar todos los inconvenientes que se presentaron, se cuenta en este momento con los productos inicialmente propuestos.
- La planificación fue uno de los puntos débiles del presente proyecto, debido a que el autor no contaba con la experiencia en el lenguaje de programación ni en las arquitecturas de las librerías requeridas para crear plugins del servidor SpigotMC. La falla de la planificación se puede apreciar si tomamos en cuenta que el tiempo real fue el doble del tiempo previsto inicialmente.

- La metodología también sufrió importantes cambios a medida que el autor recababa más información acerca del desarrollo de plugins para SpigotMC, lo cual incluía en cada etapa, nuevos temas a investigar que no fueron previstos, y a tomar en cuenta pasos significativos que inicialmente no fueron definidos claramente.
- Los cambios en la planificación y en la metodología devinieron en la necesidad de inversión de mayor tiempo para la investigación con el fin de garantizar el éxito del trabajo. Ahora que el proyecto ha cumplido con los objetivos planteados, se puede asegurar que el autor está mejor preparado para futuros proyectos de la misma especie.

## **7.2. Líneas de futuro**

Si bien en la sección de las conclusiones se anotan los objetivos como cumplidos, cabe indicar que una línea importante a futuro en la inversión de tiempo para realizar un proceso completo de rediseño de los dos plugins desarrollados. Tanto a nivel de código, con una mejor organización de tareas a cargo de paquetes y clases mucho mejor especializadas, como también en el proyecto CBR y su base de casos. En este rediseño se ha de incluir también la creación de un módulo para aplicar el aprendizaje automático que permita la actualización de la base de casos, siguiendo un criterio de evaluación de soluciones dinámica, sin necesidad de la intervención humana.

Respecto al juego SpeedRun, su ámbito en este proyecto se restringió a ser una prueba del concepto de CBR aplicado al servidor de Spigot para controlar un NPC, motivo por el cual no se profundizó sobre sus posibilidades lúdicas como en sí mismo un juego que despierte interés en la comunidad Minecraft. En una línea de futuro deseable estaría el crear un conjunto de minijuegos basados en la idea de SpeedRun, caracterizados por su simplicidad, con dedicatoria para el jugador casual.

## Bibliografía

- Aamodt, A., & Plaza, E. (2001). Case-Based Reasoning: Foundational Issues, Methodological Variations, and. *Ai Communications*, 39-59.
- Anderson, C. (24 de 4 de 2012). *The Man Who Makes the Future: Wired Icon Marc Andreessen*. Obtenido de Wired: [http://www.wired.com/epicenter/2012/04/ff\\_andreessen/](http://www.wired.com/epicenter/2012/04/ff_andreessen/)
- Bach, K., & Althoff, K.-D. (2012). Developing Case-Based Reasoning Applications Using myCBR 3. *Competence Center Case-Based Reasoning*, 15.
- Citizens. (2022). *SpigotMC*. Recuperado el 01 de 05 de 2022, de Citizens, the best NPC plugin: <https://www.spigotmc.org/resources/citizens.13811/>
- DefendTheHouse. (5 de Noviembre de 2018). *Following NPCs in Red Dead Redemption 2 for a Whole Day*. Obtenido de <https://www.youtube.com/watch?v=MrUJJgppMn4>
- Domenech, F. (2011). *ASA: SISTEMA AUTOMÁTICO DE PERSONALIZACIÓN APLICADO A PERSONAJES DE VIDEOJUEGOS*. Madrid: Universidad Complutense de Madrid.
- Dominguez, F. (2017). *Entremando bots para juegos mediante aprendizaje por refuerzo basado en casos*. Madrid: Universidad Complutense de Madrid.
- Lenz, M., Bartsch-Sport, B., Burkhard, H., & Wess, S. . (1998). *Case-Based Reasoning Technology From Foundations to Applications*. Berlin: Springer-Verlag.
- López, B. (2013). Case-Based Reasoning A Concise Introduction. *SYNTHESIS LECTURES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING*, 19-28.
- Manovich, L. (2011). *The Language of New Media*. Cambridge: MIT Press.
- MeriStation. (30 de Septiembre de 2021). *God of War: Ragnarok, primer tráiler con doblaje en español*. Obtenido de [https://as.com/meristation/2021/09/30/videos/1633018412\\_038059.html](https://as.com/meristation/2021/09/30/videos/1633018412_038059.html)
- Microsoft. (2 de Octubre de 2021). *Forza Horizon 4 Edición Excepcional*. Obtenido de <https://www.microsoft.com/es-es/p/forza-horizon-4-edicion-excepcional/9p513p4mwc71?activetab=pivot%3aoverviewtab>
- Minecraft Servers. (2022). *Servidores de Minecraft con mini juegos*. Recuperado el 01 de 05 de 2022, de <https://servidoresdeminecraft.es/tag/Mini%20juegos>
- Minecraft Wiki Oficial. (2 de Octubre de 2021). *Bedrock Edition beta scripting documentation*. Obtenido de [https://minecraft.fandom.com/wiki/Bedrock\\_Edition\\_beta\\_scripting\\_documentation](https://minecraft.fandom.com/wiki/Bedrock_Edition_beta_scripting_documentation)
- Minecraftmin. (28 de Marzo de 2017). *Introducción guía creación de mods con Forge*. Obtenido de <https://minecraftmin.net/index.php?topic=14508.0>
- MyCBR Project. (2022). *MyCBR Project*. Recuperado el 01 de 06 de 2022, de <http://mycbr-project.org/>
- Noppes. (11 de Octubre de 2014). *Custom NPCs*. Obtenido de Kode Development: <http://www.kodevelopment.nl/minecraft/customnpcs>

- NPCDestinations. (2022). *NPCDestinations - Create Living NPC's 2.6.11*. Recuperado el 01 de 06 de 2022, de <https://www.spigotmc.org/resources/npcdestinations-create-living-npcs.13863/>
- Olmos, J. (2011). *Personajes con Razonamiento Basado en Casos para videojuegos en primera persona*. Zaragoza: Universidad Zaragoza.
- Priore, P., De La Fuente, D., Pino, R., & Puente, J. (2002). UTILIZACIÓN DEL RAZONAMIENTO BASADO EN CASOS EN LA TOMA DE DECISIONES. APLICACIÓN EN UN PROBLEMA DE SECUENCIACIÓN. *Dirección y Organización*, 28.
- Radix-Shock. (2 de Octubre de 2021). *Minecraft Comes Alive*. Obtenido de <http://www.radix-shock.com/mca--overview.html>
- Rockstar Games. (2 de Octubre de 2021). *Red Dead Redemption II*. Obtenido de <https://www.rockstargames.com/reddeadredemption2/>
- Sánchez-Ruiz, A. A., & Floyd, M. W. (2021). Case-Based Reasoning Research and Development. *29th International Conference, ICCBR 2021Salamanca, Spain, September 13–16, 2021* (pág. 337). Salamanca: Springer.
- Sentinel. (01 de 06 de 2022). *Sentinel 2.5.0*. Obtenido de SpigotMC: <https://www.spigotmc.org/resources/sentinel.22017/>
- Spigot. (01 de 06 de 2022). *SpigotMC*. Obtenido de About SpigotMC: <https://www.spigotmc.org/wiki/about-spigot/>
- Zilles, L. (11 de 04 de 2009). *myCBR Tutorial*. *myCBR Project*.