



---

**Universidad de Valladolid**

# Escuela de Ingeniería Informática de Valladolid

## **TRABAJO FIN DE GRADO**

Grado en Ingeniería Informática  
Mención Computación

Librería para el desarrollo de juegos de rol  
distribuidos en Elixir

Autor:  
**Alonso Sayalero Blázquez**

Tutor:  
**Dr. César Llamas Bello**

---

# Resumen

Con el paso de los años el mundo docente se ha dado cuenta que el estudio de diversas materias cala mejor en el estudiante si se presenta de una manera amena e interactiva. En un mundo constantemente cambiante y cada vez más rápido, esto se ha visto reflejado en el aumento de la gamificación dentro del aula como una forma de combatir la constante perdida de atención y como una forma de que los conceptos se entiendan mejor y de forma clara.

Por otro lado el cómputo distribuido cada vez toma más fuerza en esta nueva era en la que nos adentramos, dominada por los grandes modelos de inteligencia artificial y cada vez más conectada a la web. En este nuevo paradigma web impera la disponibilidad inmediata de los recursos, dando lugar al aumento inequívoco de los sistemas distribuidos.

Con el fin de utilizar el juego como forma de aprendizaje de los sistemas distribuidos nace este proyecto, una librería en Elixir para la creación de un juego con varios escenarios para el aprendizaje de los alumnos.

**Palabras clave:** Sistema Distribuido, Elixir, juego, aprendizaje.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Motivación . . . . .	2
1.3. Objetivos y etapas del proyecto . . . . .	3
1.3.1. Objetivos . . . . .	3
1.3.2. Etapas . . . . .	4



# Índice de figuras



# Índice de tablas





# Capítulo 1

## Introducción

### 1.1. Introducción

La educación está en constante evolución y la informática es ya parte fundamental de la misma, desde la proyección de diapositivas en aplicaciones como Microsoft PowerPoint o Canvas, hasta el empleo de aplicaciones que fomentan la competición sana y hacen más ameno el estudio y el repaso de los conceptos aprendidos como Kahoot. En algunos centros los libros comienzan a quedarse obsoletos y son sustituidos por ordenadores o tablets.

Los sistemas distribuidos adquieren mayor importancia en este momento en el que la inteligencia artificial avanza a pasos agigantados en un mundo cada vez más conectado mediante páginas, servicios y aplicaciones web. Todos estos elementos requieren de una alta disponibilidad en todo momento y en casi todos los puntos geográficos del planeta. Es decir, requieren de sistemas distribuidos robustos y tolerantes a los fallos.

En este panorama, la educación superior también empieza a cambiar, siempre con cautela pero sin pararse. Dentro de las numerosas formas de evolución educativa en la asignatura de Sistemas Distribuidos se inclinan hacia la gamificación de la educación. Con gamificación nos referimos al empleo de cualquier tipo de juegos para enseñar al alumno conceptos que, sin visualizarse, pueden llegar a ser complicados de imaginar o comprender.

Este proyecto surge de la oportunidad para mejorar el juego empleado en la asignatura en este momento. Dicho juego es una aplicación *Java* que corre de forma local en el ordenador del alumno. Con este proyecto se pretende crear un motor que permita la creación de todo tipo de videojuegos, no solo educativos, que corran de manera distribuida, ya sea en la propia máquina del jugador, en una serie de servidores online o de forma híbrida entre un servidor y la máquina del jugador. Además, también forma parte del proyecto la creación de un juego educativo empleando el motor y la propuesta de algunos escenarios educativos haciendo uso del juego para mostrar algunas características de los sistemas distribuidos.

El motor pasa a ser la pieza principal del proyecto y sobre el que se basa el análisis y diseño del mismo. Algunas de las características que se buscan en este nuevo motor es que se puede instalar fácilmente en las máquinas de los alumnos, sea fácilmente ampliable en el futuro y emplee una

---

arquitectura de sistema distribuido basada en actores. Por lo tanto el lenguaje de programación que surge como candidato a utilizarse en el proyecto es *Elixir*. *Elixir* [1] es un lenguaje de programación funcional modular y entre sus características podemos encontrar, las siguientes:

- **Basado en *Erlang*:** *Erlang* es un lenguaje de programación funcional más antiguo que *Elixir* sobre el que este se basa. Al estar basado en este lenguaje *Elixir* hereda tanto la máquina virtual de *Erlang* como sus librerías.
- **Distribuido y Escalable:** *Elixir* se ejecuta en procesos ligeros que están aislados y se comunican mediante mensajes, lo que permite añadir nuevos procesos sin afectar el funcionamiento del resto. Estos procesos se ejecutan en una máquina virtual de *Erlang* y representan actores en el sistema.
- **Tolerante a los fallos:** Si algún componente del sistema falla, es posible recuperar la parte del sistema en la que se encuentra este fallo sin poner en riesgo el funcionamiento del resto del sistema.
- **Hot Swapping** (Actualización de código en caliente): Aplicaciones que pueden ser modificadas sin detener el sistema, lo que permite realizar modificaciones en el código o en la arquitectura sin necesidad de detener por completo el sistema.

Otro de los aspectos fundamentales a la hora de enfocar la creación de un motor de videojuegos, de cualquier tipo, es el estilo de juego que se quiere crear. Para este proyecto se ha optado por la creación de un motor de juegos de rol o RPGs.

Los juegos de rol surgen originalmente como juegos de mesa en los que una serie de jugadores participan con sus personajes en el mundo, normalmente de fantasía, que el narrador principal ha creado. Este narrador principal actúa también como árbitro en las disputas que puedan ocurrir dentro del mundo de la partida, como puede ser un combate entre uno de los jugadores y un enemigo. Los jugadores van por turnos interactuando con el mundo siguiendo las indicaciones del narrador y eligiendo que hacer cuando se les presenta un evento. El máximo exponente de los juegos de rol de mesa es *Dungeons and Dragons* [2].

Con la llegada de los videojuegos, los juegos de rol pasaron a formar parte del elenco de géneros que podemos encontrar dentro de esta industria, junto con géneros como los videojuegos de plataformas, los de acción, los de aventura gráfica o los de simulación entre otros muchos. De la mezcla de los juegos de rol con el resto de géneros surgen muchos subgéneros como los ARPGs o los JRPGs, pero en este proyecto no ahondaremos en ellos y nos centraremos únicamente en crear un motor que permita la creación y gestión de elementos clásicos de los juegos de rol, como los personajes, los enemigos o los objetos entre otros.

## 1.2. Motivación

La motivación para adentrarme en este proyecto surge de la posibilidad de adentrarme en el mundo de los videojuegos, pasión que tengo desde pequeño, mientras empleo tecnologías y para-

---

digmas de programación que no he utilizado o he empleado poco.

Este proyecto implica aprender a utilizar de manera medianamente fluida el paradigma de programación funcional al mismo tiempo que se utiliza un lenguaje de programación nunca antes visto ni utilizado, como es *Elixir*, el cual proporciona grandes ventajas a la hora de programar sistemas distribuidos.

A su vez el proyecto me permite utilizar mi experiencia como alumno de la asignatura para aportar la visión de alguien que aprende pero que no ha enseñado nunca a los escenarios por los que pasan los alumnos durante el estudio de la asignatura y de esta forma poder hacer que los compañeros que vienen detrás puedan aprender de manera entretenida los conceptos que más adelante, en su vida profesional, puedan llegar a necesitar.

Por último, la motivación que mueve a cualquier ingeniero informático de seguir aprendiendo nuevas cosas y actualizar las tecnologías que se emplean en el mundo real. Investigar piezas de software que pueden no tener utilidad real pero que funcionan como cimientos para lo que pueda llegar o que puedas crear tú mismo en el futuro. En resumen, el afán por el constante crecimiento como ingenieros.

## 1.3. Objetivos y etapas del proyecto

### 1.3.1. Objetivos

El principal objetivo del proyecto es crear un motor de videojuegos que funcione de manera distribuida. Dicho motor deberá funcionar mediante una arquitectura de actores para las partes distribuidas y tener la capacidad de modelar los elementos básicos de un juego de rol. Aparte del objetivo principal el proyecto tiene otras metas para cumplir:

- **Aprendizaje de nuevas tecnologías:** Para el proyecto se emplea el lenguaje de programación *Elixir*, lo que presenta una buena oportunidad para aprender un lenguaje que tiene su hueco en el diseño y programación web.
- **Creación de un juego:** Tal y como se ha descrito, otra parte del proyecto es el empleo del motor para crear un juego que actúe como un servidor local distribuido sobre el que poder ejecutar escenarios controlados para los alumnos.
- **Creación de escenarios:** Crear una serie de escenarios controlados sobre los que los alumnos de la asignatura de Sistemas Distribuidos puedan utilizar y modificar para poder aprender sobre el funcionamiento de diferentes conceptos de dichos sistemas.
- **Empleo de *Docker*:** Con el fin de simular un sistema distribuido en las máquinas de los alumnos, *Docker* surge como una tecnología que permite dicha simulación y, por lo tanto, el objetivo es crear un *Dockerfile* que permita el empaquetado en contenedores del motor y el juego, así como proporcionar un fichero *docker compose* que permita la inicialización de los suficiente contenedores para poder ejecutar los escenarios sin problemas.

---

### 1.3.2. Etapas

A continuación se detallan las etapas del proyecto que se siguieron.

#### **Primera etapa: Investigación y recopilación de datos sobre motores y diseño de videojuegos**

Antes de comenzar a analizar y diseñar el motor es necesario realizar una investigación previa sobre las diferentes metodologías que abarcan el diseño tanto de videojuegos como de motores. Entre los diferentes puntos de investigación destacan:

- Estudio del funcionamiento de un motor de videojuegos.
- Diseño de videojuegos de rol.
- Partes fundamentales de los juegos de rol tradicionales.
- Diseño de algoritmos de resolución de conflictos.

#### **Segunda etapa: Análisis y diseño del motor**

La segunda etapa del proyecto consiste en describir de forma analítica como se va a ver el motor de videojuegos. A partir de este análisis se hará el diseño del mismo. Para esta etapa se siguen las siguientes tareas:

- Análisis de la arquitectura del motor de videojuegos.
- Descripción de los elementos que forman parte del motor.
- Análisis del funcionamiento entre componentes del motor.
- Diseño del motor de videojuegos.

#### **Tercera etapa: Aprendizaje de *Elixir***

La tercera etapa consiste en familiarizarse y aprender a utilizar el lenguaje de programación que se va a emplear durante el proyecto. Como ya se ha mencionado, dicho lenguaje es *Elixir* y por tanto hay que hacerse con los siguientes conceptos para poder presentar un motor que, aunque sea simple, funcione:

- Funcionamiento básico del lenguaje, ejecución y entorno de desarrollo.
- Creación de actores y su funcionamiento.
- Creación de servidores TCP/IP para comunicación.
- Funcionamiento de los métodos para actuar ante los fallos.

---

## Cuarta etapa: Creación del motor

Tras tener todos los componentes necesarios para poder programar el motor la siguiente etapa consiste en programarlo, para ello se divide el trabajo en diferentes partes del motor que son necesarias para su funcionamiento. Estas partes son:

- Elementos básicos de un juego de rol que se ejecuten siguiendo una arquitectura de actores.
- Capacidad de manipulación de la base de datos utilizada como sistema de guardado del estado del juego.
- Utilidades para el correcto desarrollo de una partida de rol, como puede ser el lanzar un dado.
- Sistema de login para diferentes usuarios.

## Quinta etapa: Creación del juego

Una vez el motor está en funcionamiento la siguiente etapa es crear un juego que haga uso del motor creado y demuestre las capacidades del mismo. Al tratarse de un juego sencillo no cuenta con muchas partes, pero se pueden destacar las siguientes:

- Servidor TCP/IP que escuche por el puerto elegido.
- Sistema de arranque del juego.
- Red de supervisores que actúen frente a los fallos de los diferentes actores que conforman el videojuego.

## Sexta etapa: Creación de escenarios de estudio

Como en la etapa anterior se creó un juego que actúa como un servidor de la misma forma que se hace en la arquitectura cliente-servidor, los escenarios modificables por los alumnos tendrán que actuar como clientes del juego y realizar peticiones concretas. Para conseguir esto los escenarios tienen que cumplir las siguientes características:

- Estar completos. Los escenarios que se entregan con este proyecto están completos y actúan como soluciones.
- Demostrar una o varias características de los sistemas distribuidos.
- Estar escritos en *Java*. Es el lenguaje de programación empleado en la asignatura de Sistemas Distribuidos.



# Bibliografía

- [1] The Elixir team. Elixir. <https://elixir-lang.org/>, 2025. [Online; accessed 21-June-2025].
- [2] Wizards of the Coast. Dungeons and dragons. <https://dnd.wizards.com/es>, 2025. [Online; accessed 22-June-2025].