



UNIVERSITAT DE
BARCELONA

Treball final de grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Gamificación de un chatbot educacional

Autor: Laura Garcia Fernandez

Director: Dra. Inmaculada Rodríguez Santiago

Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, June 13, 2023

Resumen

La gamificación, una estrategia que consiste en aplicar elementos y técnicas de juegos en contextos no relacionados con el juego, ha ganado popularidad en diversos ámbitos, incluyendo el ámbito educativo. La implementación de la gamificación en un chatbot diseñado para brindar apoyo en el aprendizaje ofrece numerosos beneficios y tiene como objetivo principal aumentar la motivación y el compromiso de los estudiantes en su proceso de aprendizaje.

En este trabajo de fin de grado se diseña e implementa un sistema gamificado para la chatbot LovelaceUB [11]. Este chatbot es una herramienta de apoyo para estudiantes de programación, al que le pueden hacer preguntas sobre diferentes asignaturas y solicitar ejercicios de programación que, una vez resueltos, pueden pedir al chatbot que los corrija.

Para implementar el sistema gamificado, se han estudiado los tipos de jugadores que existen y se ha diseñado la gamificación utilizando la plantilla del Gamification Model Canvas [24], donde se definen, entre otras cosas, las mecánicas de juego por las que se regirá el chatbot, como por ejemplo, recibir puntos cuando el chatbot corrige un ejercicio. Finalmente, este sistema se ha implementado en Python para un chatbot de Telegram.

Abstract

Gamification, a strategy that involves applying game elements and techniques in non-game contexts, has gained popularity in various fields, including education. Implementing gamification in a chatbot designed to provide learning support offers numerous benefits and aims to increase students' motivation and engagement in their learning process.

In this final degree project, a gamified system is designed and implemented for the chatbot LovelaceUB [11]. This chatbot is a support tool for programming students, to whom they can ask questions about different subjects and request programming exercises that, once they have solved them, they can ask the chatbot to correct.

To implement the gamified system, the types of players that exist have been studied, and the gamification has been designed using the Gamification Model Canvas template [24], where, among other things, the game mechanics by which the chatbot will be governed are defined, such as receiving points when the chatbot corrects an exercise. Finally, this system has been implemented in Python for a Telegram chatbot.

Resum

La gamificació, una estratègia que consisteix a aplicar elements i tècniques de jocs en contextos no relacionats amb el joc, ha anat guanyant popularitat en diferents àmbits, entre els quals es troba l'educació. La implementació de la gamificació en un xatbot dissenyat per donar suport en l'aprenentatge ofereix nombrosos beneficis i té com a objectiu principal augmentar la motivació i el compromís dels estudiants en el seu procés d'aprenentatge.

En aquest treball de fi de grau es dissenya i implementa un sistema gamificat per al xatbot LovelaceUB [11]. Aquest xatbot és una eina de suport per als estudiants de programació, al que li poden preguntar dubtes sobre diferents assignatures, i demanar exercicis de programació que, quan els hagin resolt, li poden demanar al xatbot que els corregeixi.

Per implementar el sistema gamificat s'ha estudiat els tipus de jugador que existeixen, s'ha dissenyat la gamificació fent ús de la plantilla de Gamification Model Canvas [24], on es defineixen entre d'altres, les mecàniques de joc per les que es regirà el xatbot, com per exemple, rebre punts quan el xatbot et corregeix un exercici.

Finalment, aquest sistema s'ha implementat en Python per a un xatbot de Telegram.

Contents

Index of figures	i
1 Introducción	2
2 Objetivos	4
2.1 Objetivo general	4
2.2 Objetivos específicos	4
3 Trabajos relacionados	6
3.1 LovelaceUB	6
3.2 Chatbots y gamificación en la educación	6
4 Conceptos relacionados	9
4.1 Chatbot y chatbot educacional	9
4.2 Gamificación	9
4.3 Tipos de jugador	10
5 Análisis	13
5.1 Requisitos funcionales	13
5.1.1 Definición de usuarios	13
5.1.2 Casos de uso	14

5.2	Requisitos no funcionales	20
6	Diseño	21
6.1	Diseño de la gamificación	21
6.1.1	Gamification Model Canvas	22
6.2	Diseño del software	26
6.3	Diseño de la base de datos	34
7	Implementación y resultados	36
7.1	Tecnologías relacionadas	36
7.1.1	Python	36
7.1.2	MariaDB	36
7.1.3	Telegram	37
7.2	Implementación de la gamificación	37
7.2.1	Implementación de las mecánicas de juego	37
8	Conclusiones y trabajo futuro	45
9	Apéndice I - Manual técnico	50
9.1	Requisitos de software	50
9.2	Variables de entorno	50
9.3	Ejecución del código	52
9.3.1	Ejecución con dockers	53
9.3.2	Ejecución sin dockers	53

Index of figures

3.1	Interfície de Usuario de CiboPoli. Fuente: adhil, A., Villafiorita, A. [10] . . .	7
3.2	Elementos de Gamificación de Duolingo. Fuente: Shortt, M., Tilak, S., Kuznetcova, I., Martens, B., and Akinkuolie, B[22]	8
4.1	HEXAD, Tipos de jugador. Fuente: A. Marczewski	11
5.1	Diagrama de casos de uso. Fuente: Propia	15
6.1	Bloques del GMC. Fuente: propia	22
6.2	Gamification Model Canvas	25
6.8	Diagrama base de datos. Fuente: Propia	35
7.1	Ejemplo de interacción donde un jugador (Laura) pide un ejercicio, lo resuelve (M1) y visualiza la clasificación. Fuente: propia	38
7.2	Flujo de la mecánica M2. Fuente: propia	39
7.3	Flujo de las mecánicas M3 y M5. Fuente: propia	40
7.4	Flujo de la mecánica M4. Fuente: propia	42
7.5	Flujo de la mecánica M6. Fuente: propia	43
7.6	Flujo de la mecánica M7. Fuente: propia	44
9.1	BotFather para obtener token. Fuente: Propia	52

Chapter 1

Introducción

El proceso de enseñanza-aprendizaje ha evolucionado gracias a la era digital, que ha facilitado el acceso a diversas tecnologías. Desde pizarras digitales hasta plataformas de gestión del aprendizaje (LMS) como Moodle o Edx, se ha vuelto más fácil brindar materiales y actividades en línea para los estudiantes. Además, han emergido plataformas que ofrecen cuestionarios gamificados, como Kahoot, que hacen que el aprendizaje sea más interactivo y ameno. También se han unido a estas herramientas otras basadas en la interacción con lenguaje natural, como son los chatbots educativos.

Los agentes conversacionales o chatbots, son sistemas interactivos capaces de entablar una conversación con un usuario en lenguaje natural. Y cada vez se usan más en diferentes ámbitos, incluido el ámbito de la educación.

En el ámbito educativo se encuentran en general dos tipos de chatbots, los que proporcionan información académica (servicios disponibles, actividades en el campus, etc.) [19] y los que contribuyen a mejorar el proceso de aprendizaje [15]. Estos últimos actúan como auxiliares de los profesores y ofrecen a los estudiantes un trato personalizado para cada uno, siguiendo su propio ritmo de aprendizaje. Sin embargo, a veces los alumnos pueden encontrar estas interacciones impersonales y carentes de atractivo. Para abordar esta limitación y fomentar un mayor compromiso de los estudiantes, entra en juego la gamificación.

La gamificación busca transformar actividades no relacionadas con el juego añadiendo elementos y técnicas propias del juego. Por ejemplo, se pueden usar puntos o medallas como recompensa a lo que se espera del participante. Esto ofrece una motivación extra para realizar ciertas tareas, y se puede aplicar en cualquier ámbito, tanto laboral, educacional, como incluso de la salud [12].

En este proyecto se gamifica a un chatbot educacional, LovelaceUB [11], destinado a alumnos de la facultad de Matemáticas e Informática de la Universidad de Barcelona. Este es un chatbot de Telegram que se usa como complemento en el proceso de aprendizaje, donde los profesores pueden añadir el conocimiento necesario al chatbot y los alumnos pueden interactuar con él para hacerle preguntas y resolver dudas sobre una asignatura, pedir ejercicios de la asignatura (ejercicios de programación JAVA o C), y corregir dichos ejercicios para obtener el resultado.

Se ha decidido gamificar el chatbot con un doble objetivo, esto es, para aumentar la motivación de los alumnos y para incrementar también la interacción entre chatbot y alumno. Así, los alumnos podrán aprovechar todos los recursos educativos que el chatbot les ofrece a la vez que consiguen recompensas por ello.

Chapter 2

Objetivos

2.1 Objetivo general

El objetivo de este trabajo de fin de grado es gamificar la chatbot LoveLaceUB para que sea más atractivo y divertido para los alumnos, y de esta forma fomentar su uso como herramienta complementaria en el proceso de enseñanza-aprendizaje.

2.2 Objetivos específicos

El objetivo general se desglosa en los siguientes objetivos específicos:

- Investigar sobre la gamificación.
- Analizar otros chatbots gamificados.
- Analizar y comprender los proyectos anteriores de la chatbot LovelaceUB, donde se integrará la gamificación.
- Proponer un diseño para la gamificación del chatbot siguiendo el modelo de Gamification Model Canvas.
- Extender la base de datos para la gestión de jugadores, puntos, ejercicios, etc.
- Implementar e integrar la lógica requerida para hacer la gestión de la gamificación en el módulo del chatbot.

- Implementar e integrar la lógica requerida para hacer la gestión de la gamificación en el módulo del manager de la base de datos.
- Implementar mensajes customizados para la gamificación.
- Diseñar los elementos gamificados.

Chapter 3

Trabajos relacionados

3.1 LovelaceUB

El proyecto de LoveLaceUB es un trabajo realizado por David Fernández a modo de trabajo de final de grado [11], y más adelante ampliado por Joan Domenech[18]. En el proyecto se desarrolló una agente conversacional que mediante el uso de la tecnología AIML es capaz de conversar con usuarios a través de Telegram. Con tal de poder ser usado como apoyo en la asignatura de Programación I del grado de Ingeniería Informática, posee conocimientos de distintas materias y también puede dar ejercicios a los alumnos, así como corregirlos.

Además, en paralelo al proyecto actual se ha desarrollado una nueva versión del chatbot que incorpora una asignatura del Grado de Matemáticas, además de sustituir AIML por un nuevo framework conversacional basado en ML (Machine Learning) llamado Rasa[7].

3.2 Chatbots y gamificación en la educación

La literatura sobre chatbots en educación y gamificación para el aprendizaje es limitada. Algunos ejemplos de chatbot son Jill Watson, una asistente de profesor virtual desarrollada por el Instituto de Tecnología de Georgia (Georgia Tech) en 2016. Fue creada para ayudar a los profesores a responder preguntas de los estudiantes en los foros de discusión en línea de los cursos masivos abiertos en línea (MOOC). Jill Watson utilizó inteligencia artificial para responder preguntas de los estudiantes y fue capaz de responder con precisión el 97% de las preguntas que recibió. Los estudiantes no sabían que estaban interactuando

con una asistente virtual y algunos incluso la elogiaron por su capacidad para responder preguntas de manera rápida y efectiva.

Ya en contexto de chatbots en sistemas de gestión del aprendizaje (LMS), en 2017, se presentó el primer chatbot de Moodle en MoodleMoot Australia. Su objetivo era ayudar al usuario ofreciendo servicios de búsqueda. En el futuro, se espera que incorpore más factores de contexto, evolucione hacia la "Comprensión del Lenguaje Natural (N.L.U.)" e identifique emociones e interactúe en consecuencia con el usuario [17].

Ahmed Fadhil y Adolfo Villafiorita han desarrollado un chatbot educativo para asesorar y enseñar sobre estilos de vida saludables utilizando un juego llamado 'CiboPoliBot'[10]. El juego presentado se fundamenta en un prototipo en papel desarrollado para instruir a los estudiantes de primaria acerca de una alimentación saludable y la gestión del desperdicio de alimentos (Ver Figura 3.1).

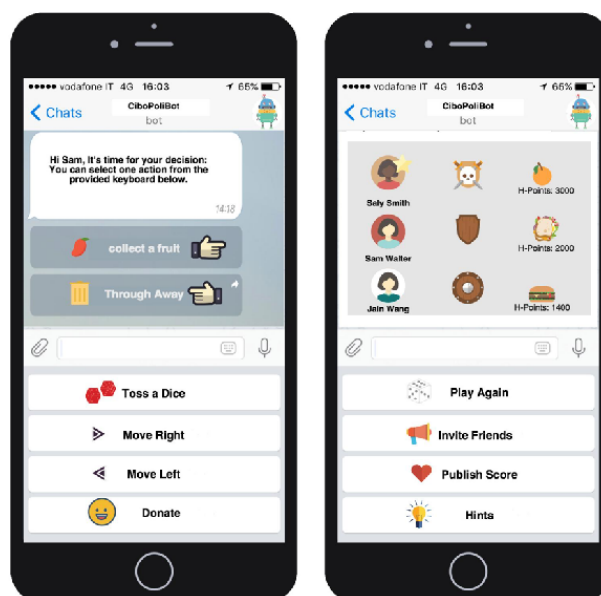


Figura 3.1: Interficie de Usuario de CiboPoli. Fuente: adhil, A., Villafiorita, A. [10]

Otras aplicaciones como Rosetta Stone y Duolingo [13] utilizan chatbots para enseñar idiomas extranjeros. Concretamente, Duolingo es un bot creado por una startup de Pittsburgh, lanzado en 2011, que cuenta con 150 millones de usuarios. Es una aplicación para aprender idiomas extranjeros que incluye diferentes elementos de gamificación (Figura 3.2) como recompensas y objetivos para los usuarios.

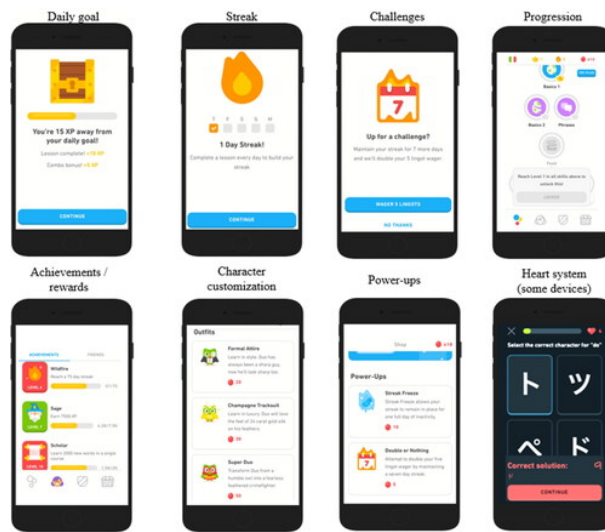


Figura 3.2: Elementos de Gamificación de Duolingo. Fuente: Shortt, M., Tilak, S., Kuznetcova, I., Martens, B., and Akinkuolie, B[22]

Chapter 4

Conceptos relacionados

4.1 Chatbot y chatbot educacional

Un chatbot es un sistema conversacional que permite a los usuarios comunicarse con un sistema interactivo mediante el uso de lenguaje natural[8]. En el contexto educativo, los chatbots se podrían clasificar en diferentes tipos según su funcionalidad. Por un lado, hay chatbots de aprendizaje o educacionales, diseñados para ser herramientas de aprendizaje.

Por otro lado, hay chatbots, puede ser el de soporte, que cada vez son más utilizados en páginas web aplicaciones, ya que incorporar esta herramienta puede facilitar la navegación del usuario y resolver dudas sobre el contenido de la página.

4.2 Gamificación

La gamificación es una técnica innovadora que busca mejorar y transformar actividades que no son juegos, integrando elementos y estrategias propias de los juegos. Esta táctica se ha expandido a diferentes ámbitos, como la educación[23], la atención médica[21], el marketing[20] y el entretenimiento.

El objetivo principal de la gamificación es lograr la participación activa de los usuarios, aumentar su motivación y mejorar su experiencia en el proceso. En el ámbito educacional, la gamificación puede utilizarse como estrategia para incrementar el compromiso de los estudiantes. Al ofrecer una experiencia interactiva y divertida, se logra recompensar a los usuarios por alcanzar metas o completar tareas específicas, generando así un mayor

sentido de logro y satisfacción. Además, a través de la gamificación, es posible mejorar la experiencia del usuario, proporcionando retroalimentación constante sobre su desempeño y progreso hacia los objetivos establecidos.

La gamificación tiene una amplia gama de beneficios. No solo incentiva una mayor implicación y motivación, sino que también puede aumentar la efectividad en la ejecución de tareas y proporcionar una experiencia emocionante y satisfactoria.

4.3 Tipos de jugador

La gamificación es una herramienta poderosa para involucrar a los usuarios y motivarlos para completar tareas, pero no todos los usuarios son iguales ni tienen las mismas motivaciones. Cada usuario puede tener sus propias motivaciones intrínsecas, como ayudar a los demás o aprender. Y, también habrá que tener en cuenta las motivaciones extrínsecas del usuario en el entorno del juego, como ganar puntos o tener un alto estatus. Para ello, se han desarrollado diferentes frameworks que definen los tipos de usuario [14], como Hexad o el propuesto por Bartle[9], que identifica a cuatro tipos de jugadores según los aspectos que generalmente entretienen más a los usuarios, estos serían: logros en el contexto del juego (Archiver), exploración del juego (Explorer), socializar con otros jugadores (Socializer) e imposición a los demás (Killers).

En mi caso me guíé por Hexad [16], un sistema de clasificación que define seis tipos diferentes de jugadores y asocia a cada tipo de jugador unas mecánicas que se adaptan al mismo. A continuación se describen cada uno de los tipos junto con mecánicas asociadas:

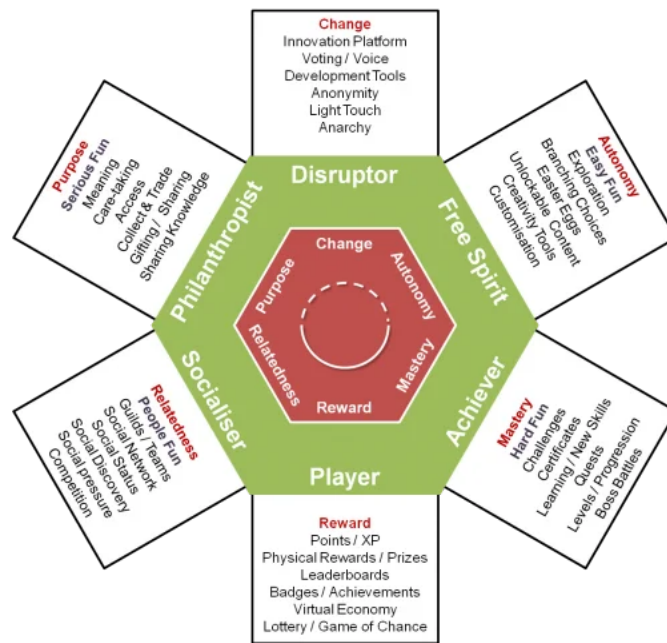


Figura 4.1: HEXAD, Tipos de jugador. Fuente: A. Marczewski

1. "Socializer" (Socializadores): Estos jugadores se enfocan en las conexiones sociales presentes en el juego. Les gusta trabajar en equipo, comunicarse y crear grupos. La comunicación con otros jugadores y la creación de relaciones son aspectos esenciales para disfrutar de su experiencia de juego.
2. "Free Spirits" (Espíritus Libres): Estos jugadores están motivados por la curiosidad y el descubrimiento. Les gusta recorrer entornos virtuales, buscar secretos y desbloquear zonas ocultas. Este tipo de jugador se siente atraído por la oportunidad de descubrir cosas nuevas y experimentar la libertad dentro del juego.
3. "Achiever" (Logrador): Estos jugadores tienen una fuerte orientación hacia la superación personal, disfrutan de los desafíos y la competencia. Su motivación se basa en la búsqueda de la maestría, lo cual implica adquirir nuevos conocimientos, mejorar constantemente y enfrentar dificultades con el objetivo de superarse a sí mismos.
4. "Philanthropists" (Filántropos): La fuente de impulso de este tipo de jugador radica en su propósito y en el sentido que le dan a su vida. Tienen un enfoque altruista y aspiran a ayudar y mejorar la calidad de vida de los demás sin esperar nada a cambio.

5. "Players" (Jugadores): Los Players se enfocan en lograr objetivos y conseguir recompensas. Les impulsa ganar puntos, avanzar de nivel y alcanzar metas concretas. El sentimiento de logro y el reconocimiento son relevantes para ellos.
6. "Disruptors" (Disruptores): La motivación de los disruptores se enfoca en el cambio. Pretenden modificar el sistema del juego, ya sea de forma directa o a través de otros jugadores, con la finalidad de propiciar transformaciones beneficiosas o perjudiciales.

Es importante tener en cuenta que estos tipos de jugadores no son incompatibles entre sí, y que los jugadores pueden demostrar rasgos de distintos tipos en distintas situaciones y escenarios de juego. El modelo HEXAD brinda una perspectiva completa de las motivaciones de los jugadores, lo que nos será de utilidad para diseñar experiencias de juego más atractivas y personalizadas.

Chapter 5

Análisis

Antes de desarrollar la gamificación del chatbot hay que definir los requisitos funcionales y no funcionales, es decir, todo lo que el usuario podrá hacer con el chatbot (funcionales) y como se espera que funcione la parte no visible para el usuario (no funcionales).

5.1 Requisitos funcionales

5.1.1 Definición de usuarios

Antes de empezar a definir las acciones que podrá realizar un usuario, definimos los usuarios que usarán el chatbot. Este chatbot contaba con tres tipos de usuario, el alumno, el profesor y el administrador. Para la gamificación del chatbot solo nos vamos a centrar en el alumno, ya que los profesores y administradores solo se encargan de la gestión del chatbot.

Para ello proponemos tres tipos de usuario:

- Usuario no registrado: Alumno que entra por primera vez al chatbot, esto es, que todavía no se ha registrado.
- Usuario: Alumno ya registrado que entra en el chatbot y no quiere usar la gamificación.
- Jugador: Alumno ya registrado que entra en el chatbot y entra en el modo jugador para disfrutar de la gamificación.

5.1.2 Casos de uso

La mayoría de casos de uso del chatbot van relacionados con un comando. En los bots de Telegram se pueden definir comandos, esto es, una palabra reservada para ejecutar una acción concreta, y siempre irá precedida de una '/', como por ejemplo */ejercicio*.

A la hora de gamificar el chatbot decidimos dar la opción a los alumnos de usar o no el sistema gamificado para que tengan libertad de elección. Por ello, los requisitos funcionales del chatbot para un usuario en modo normal, sin usar la gamificación, no han cambiado respecto al chatbot anterior, pero al haber añadido un nuevo actor, el jugador, han aparecido nuevos casos de uso.

Primero, un usuario se tiene que poder registrar como jugador (UC-3) o salir del modo jugador si ya se ha registrado anteriormente (UC-4). Después, en la simple interacción con el chatbot se ha añadido un contador de mensajes enviados al chatbot (UC-10) para, según la cantidad de interacción que el usuario tiene con el chatbot, se le premie con puntos y regalos. A la hora de pedir un ejercicio, estos se dividen en grupos, separados por tipo (secuenciales, iterativos, etc) y dentro del mismo tipo, separados por dificultad (básico o avanzado), el UC-6 especifica que solo se pueden pedir los ejercicios de los grupos que tengas desbloqueados de la asignatura seleccionada en ese momento.

En la Figura 5.1 podemos distinguir los casos de uso que contempla el chatbot, diferenciado de cuando un usuario es jugador (color lila) a cuando está en modo normal, esto es, no ha indicado que quiere gamificación o entró en modo gamificación y después salió. (azul). Para las acciones de pedir ejercicios, resolver ejercicios y hacer preguntas tenemos dos casos de uso para cada una, ya que hacen referencia al caso de uso gamificado y sin gamificar.

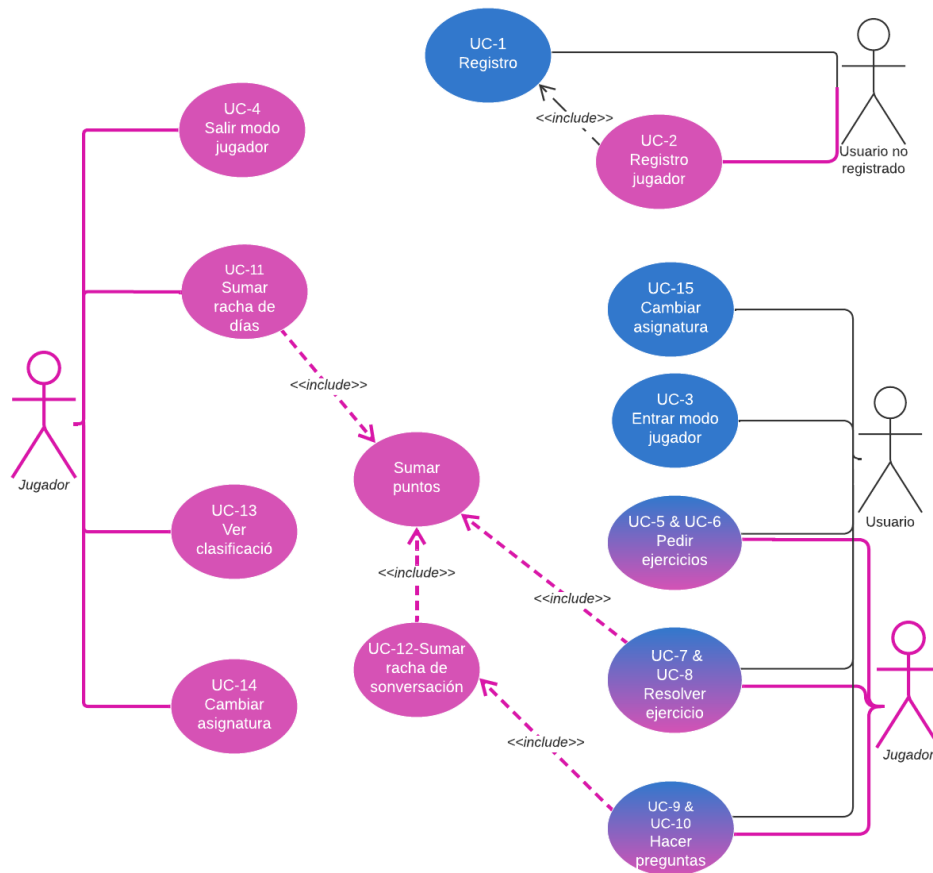


Figura 5.1: Diagrama de casos de uso. Fuente: Propia

A continuación, se muestra la descripción textual de los nuevos casos de uso del chatbot para el sistema gamificado, en las que se especifica el nombre del caso de uso, el actor implicado, una breve descripción, las precondiciones para que se dé el caso de uso, el flujo que este seguirá, y finalmente, las condiciones que se darán al terminar el flujo del caso de uso.

Nombre	UC-2 Registro jugador
Actor	Usuario no registrado
Descripción	Alguien que desea registrarse en el chatbot en modo jugador.
Precondiciones	Usuario que entra por primera vez y desea registrarse en el chatbot en modo gamificado.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario envía el comando /game seguido de su nombre y correo. 2. El sistema muestra un mensaje de éxito.
Flujo alternativo	<ol style="list-style-type: none"> 1. (a) El usuario envía el comando /game. (b) El sistema devuelve un mensaje con el formato correcto: /game nombre email.
Postcondiciones	El usuario se ha registrado correctamente en el chatbot y está en modo jugador.

Nombre	UC-4 Salir del modo jugador
Actor	Jugador
Descripción	Un jugador quiere salir del modo jugador
Precondiciones	Usuario registrado como jugador
Flujo normal	<ol style="list-style-type: none"> 1. El jugador envía el comando /nogame. 2. El chatbot envía mensaje de éxito
Postcondiciones	El usuario ya no está en modo jugador

Nombre	UC-6 Pedir ejercicios
Actor	Jugador
Descripción	Un jugador quiere pedir el enunciado de un nuevo ejercicio.
Precondiciones	Usuario registrado en modo jugador.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario envía el comando /exercise. 2. El sistema devuelve una lista con los tipos de ejercicios disponibles, esto es, desbloqueados para el usuario con el formato /exercise tipo dificultad 3. El usuario envía el comando /exercise tipo dificultad, ya sea escribiéndolo o seleccionándolo de la lista. 4. El chatbot le envía un fichero de ese tipo y dificultad 5. Si ha enviado un tipo y dificultad no existentes o no disponible para el usuario, el chatbot le envía un mensaje de error y le vuelve a enviar la lista de los disponibles.
Flujo alternativo	<ol style="list-style-type: none"> 5. (a) Si ha enviado un tipo y dificultad no existentes o no disponible para el usuario, el chatbot le envía un mensaje de error y le vuelve a enviar la lista de los disponibles.
Postcondiciones	El usuario ha recibido un ejercicio que todavía no ha solucionado.

Nombre	UC-8 Resolver ejercicio
Actor	Jugador
Descripción	Un jugador envía un ejercicio para resolver.
Precondiciones	Usuario registrado en modo jugador.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador envía un fichero con el ejercicio. 2. El chatbot corrige el ejercicio 3. El chatbot comprueba si el jugador ha desbloqueado un nuevo grupo de ejercicios 4. El chatbot comprueba si el jugador ha solucionado todos los ejercicios de un grupo 5. El chatbot comprueba si el jugador ha subido de nivel
Flujo alternativo	<ol style="list-style-type: none"> 1. (a) Si el fichero enviado no contiene su correspondiente fichero de test, el chatbot envía mensaje de error 3. Si se ha desbloqueado un nuevo grupo de ejercicios (ver mecánica <i>M2-Desbloqueo</i> en la sección 6.1.1) <ol style="list-style-type: none"> (a) Se le envía al jugador el gif correspondiente. (b) Se le envía al jugador la lista de los grupos de ejercicios disponibles. 4. Si se han solucionado todos los ejercicios de un grupo de ejercicios (ver mecánica <i>M3-Todos tema-nivel</i> en la sección 6.1.1) <ol style="list-style-type: none"> (a) Se le envía al jugador el gif correspondiente (b) El chatbot comprueba si el jugador ha solucionado todos los ejercicios de la asignatura <ol style="list-style-type: none"> i. Si se han solucionado todos los ejercicios se envía el gif correspondiente al jugador (ver mecánica <i>M4-Todos asignatura</i> en la sección 6.1.1) 5. Si se ha subido de nivel (ver mecánica <i>M5-Subir de nivel</i> en la sección 6.1.1) <ol style="list-style-type: none"> (a) Se le envía al jugador el gif correspondiente
Postcondiciones	El usuario obtiene tantos puntos como puntuación en el test.

Nombre	UC-10 Hacer preguntas
Actor	Jugador
Descripción	Un jugador envía preguntas al chatbot sobre el plan docente o sobre la materia de la asignatura.
Precondiciones	Usuario registrado en modo jugador.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador envía el mensaje al chatbot. 2. El chatbot suma 1 al contador de mensajes con el chatbot del usuario 3. El chatbot envía mensaje de respuesta
Flujo alternativo	<ol style="list-style-type: none"> 1. Si el contador es múltiplo de 10 (ver mecánica <i>M7-Preguntas</i> en la sección 6.1.1) <ol style="list-style-type: none"> (a) Se le envía al jugador una imagen (b) Se tira un dado con el que se sumarán puntos al jugador
Postcondiciones	Se suma el contador de mensajes al chatbot del usuario.

Nombre	UC-13 Ver clasificaión
Actor	Jugador
Descripción	Un jugador quiere ver la clasificación.
Precondiciones	Usuario registrado en modo jugador.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador envía el comando /leaderboard. 2. El chatbot envía la clasificación del grupo perteneciente al usuario.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si el jugador todavía no ha conseguido ninguna puntuación <ol style="list-style-type: none"> (a) El chatbot le da el mensaje de información
Postcondiciones	El usuario tiene conocimiento de su puesto en la clasificación.

5.2 Requisitos no funcionales

Los requisitos no funcionales son elementos o propiedades del sistema que no están relacionados de manera directa con su funcionalidad principal, sino más bien con otros aspectos como su rendimiento, calidad y comportamiento en general. Algunos ejemplos son requisitos de rendimiento, requisitos de usabilidad, requisitos de seguridad, etc.

En nuestro caso, nos hemos centrado en la usabilidad. Por un lado, el usuario tiene que saber lo que puede hacer en cada momento, y que le sea cómodo y fácil de hacer. Para ello, se han incluido los comandos, explicados en la sección anterior de casos de uso, que permiten al usuario saber qué puede hacer, y además, entre los comandos se encuentra el comando */help* que al enviarlo al chatbot, este envía al usuario un mensaje de todo lo que puede hacer el usuario. Y, por otro lado, ofrecer una experiencia educativa hedónica al usuario haciendo que disfrute del proceso educativo. Esto se consigue mediante las recompensas de la gamificación.

Chapter 6

Diseño

6.1 Diseño de la gamificación

A la hora de diseñar un sistema gamificado hay que tener en cuenta los diferentes tipos de jugadores que lo van a usar. Los jugadores se podrían separar en diferentes tipos, como hemos visto anteriormente:

- Competitivos: Estos jugadores buscan obtener la puntuación más alta, subir en las clasificaciones y ganar premios.
- Colaborativos: Estos jugadores disfrutan trabajar en equipo y cooperar con otros.
- Exploradores: Estos jugadores están motivados por descubrir nuevas cosas, buscan desbloquear contenido adicional.
- Sociales: Estos jugadores se centran en la interacción social y buscan conectarse con otros jugadores.
- Realizadores: Estos jugadores están motivados por alcanzar metas y logros.
- Altruistas: Estos jugadores están motivados por actividades que tienen un impacto positivo en la comunidad o que les permiten compartir su conocimiento y experiencia con otros.

Teniendo en cuenta los diferentes perfiles de jugadores, opté por enfocarme en un grupo reducido. En un principio, se me ocurrió centrarme en los jugadores competitivos, ya

que la gamificación suele involucrar la obtención de puntos y puestos destacados en la clasificación. Sin embargo, para poder involucrar al máximo de usuarios posibles, realicé una investigación para identificar los tipos de jugadores más habituales.

En una página web donde explica los tipos de jugadores también había un link a un test para ver qué tipo de jugador eres, y, además, había un histórico de todos los test que se habían realizado. Con un total de 438 tests realizados, el tipo de jugador con el porcentaje más alto es el explorador. Así que me centré en estos dos tipos, competitivo y explorador.

6.1.1 Gamification Model Canvas

Para hacer el diseño de la gamificación del chatbot, he usado el Gamification Model Canvas (GMC), una herramienta que me ha ayudado a considerar diferentes aspectos a tener en cuenta en el diseño del chatbot gamificado.

El GMC se divide en tres secciones[24]:



Figura 6.1: Bloques del GMC. Fuente: propia

- Estrategia: Donde se describen los diferentes escenarios y componentes, también es donde se consideran los costes de inversión.
- Valor: Aquí entran la motivación, el valor que se aporta al usuario y los beneficios.
- Diseño del juego: Reglas, comportamientos y acciones que realizarán los jugadores.

Teniendo en cuenta los tipos de jugadores, rellené la plantilla de GMC que permite especificar los diferentes aspectos del diseño. En la parte de *estrategia* hay que definir la plataforma en la que se va a hacer la gamificación, en mi caso es Telegram; los componentes y mecánicas que tendrá, que mencionaremos en el apartado de diseño del juego; y, por último, los costes, que sería el tiempo necesario para desarrollar la gamificación del chatbot.

En el apartado de *valor* nos centramos en el usuario, y ahí se definen los jugadores, que serán estudiantes de primer año de carrera, y además nos centraremos en los perfiles

de jugadores competitivos y exploradores. Aquí también se definen las acciones que deberá realizar el usuario y la respuesta emocional que se espera, estos puntos también se extenderán en el siguiente párrafo. Y por último, en este apartado, también se definen los beneficios, en este caso los beneficios para el usuario, que serán mejorar la sus habilidades y conocimientos al interactuar con el chatbot y además poder contemplar su progreso con los puntos y niveles conseguidos.

Por último, definimos los apartados del *diseño del juego*. Empezando por el tipo de dinámicas que tendrá el juego. Estas dinámicas serán de clasificación por puntos conseguidos, de reto a la hora de resolver ejercicios, y de progresión, ya que los ejercicios, que están agrupados en temas y dentro de estos en dos niveles de dificultad, se irán desbloqueando según el jugador va resolviendo, esto lo definimos en las mecánicas. Podemos definir también los componentes que se necesitan para llevar a cabo las mecánicas que describimos a continuación, que en este caso serían puntos, niveles numerales del 1 al 10, clasificación, y dado (virtual en nuestro caso) Las mecánicas vendrían a ser las reglas o condiciones para ganar puntos u obtener recompensas, y se han definido las siguientes:

- (M1-Resolver ejercicio) Resolver ejercicio \rightarrow Gana x puntos, según el resultado de la corrección del ejercicio (si la nota obtenida es x, gana x puntos)
- (M2-Desbloqueo) Resuelve la mitad ejercicios de un tema-nivel \rightarrow Desbloquea nuevo tema-nivel
- (M3-Todos tema-nivel) Resolver todos los ejercicios de un tema-nivel \rightarrow Gana 15 puntos
- (M4-Todos asignatura) Resolver TODOS los ejercicios de la asignatura \rightarrow Gana 30 puntos
- (M5-Subir de nivel) Gana un total de 100 puntos \rightarrow Sube de nivel
- (M6-Racha días) Cada 3 días continuos de interacción con el chatbot (días en racha), ya sea pedir ejercicios, corregirlos o hacerle preguntas \rightarrow Gana x puntos donde x es un número aleatorio del dado multiplicado por días en racha/3, también obtiene un enlace de interés para practicar más ejercicios
- (M7-Preguntas) Cada 10 preguntas hechas al chatbot \rightarrow Gana x puntos donde x es un número aleatorio del dado, también se le muestra una imagen/viñeta divertida sobre informática

Son mecánicas centradas en los puntos y clasificación, con niveles para ver la progresión y ese punto de desbloquear nuevos niveles de ejercicios.

La parte que aporta más valor al usuario de estas definiciones, las acciones y la estética. Las acciones son las acciones que deberá realizar el usuario para llevar a cabo las mecánicas, y en este caso es hacer preguntas e interactuar con el chatbot y enviarle ejercicios para corregirlos. Y finalmente, la estética, que no se refiere a la forma o colores que tenga, sino a la reacción o emoción que causa en el usuario, y he definido desafío a la hora de enfrentarse a resolver nuevos problemas y progresión al adquirir puntos y subir en la clasificación y de nivel.

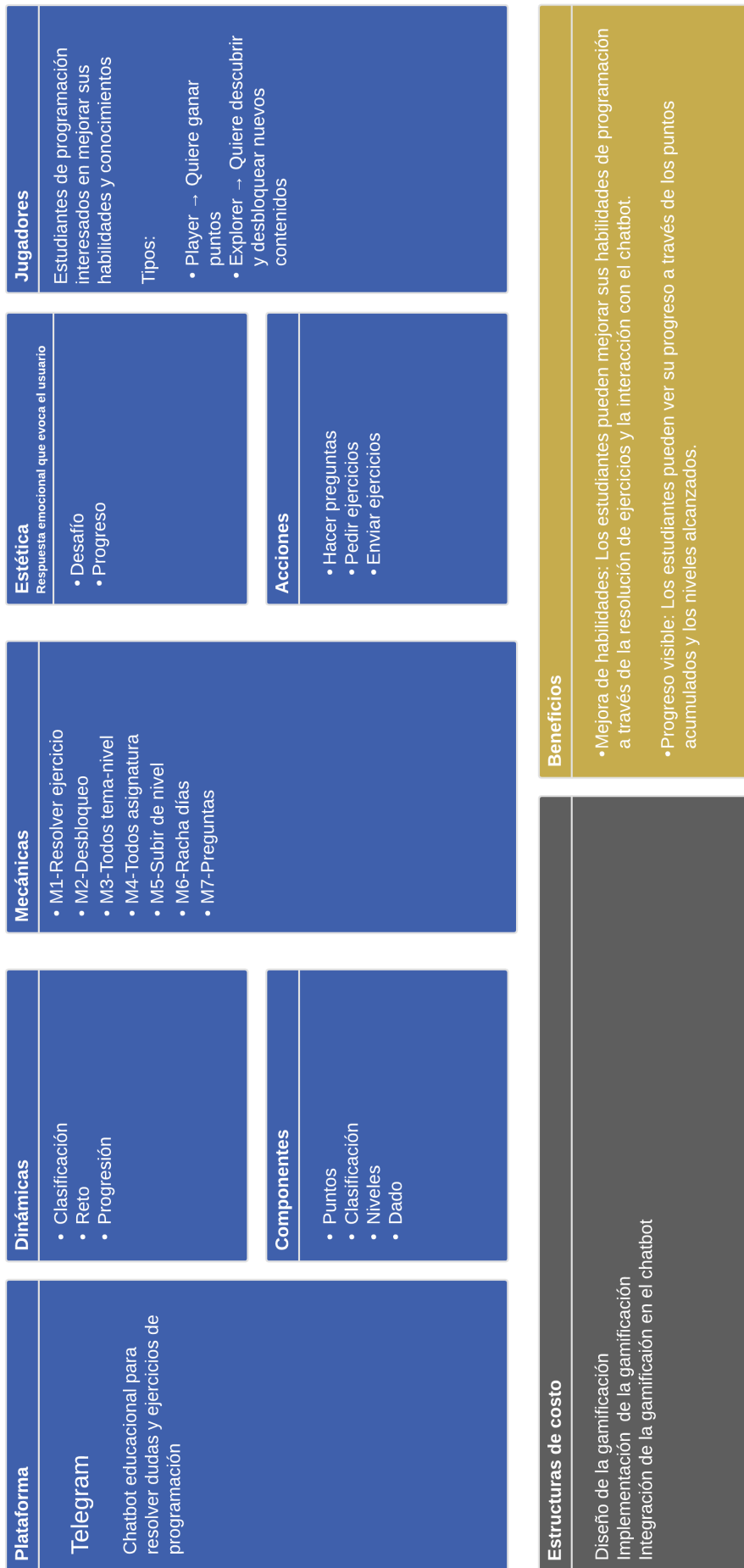


Figura 6.2: Gamification Model Canvas

6.2 Diseño del software

El diseño del chatbot está dividido en 3 grandes bloques:

- Chatbot: Recibe los mensajes del usuario, gestiona el mensaje y envía una respuesta al usuario. En este bloque se incorpora un *framework conversacional* para interactuar con el usuario en lenguaje natural, y el sistema de gamificación para gestionar los aspectos relacionados con el juego.
- Manager DB: Se encarga de interactuar con la base de datos, donde se almacenan datos de los usuarios, los ejercicios y de la gamificación.
- Corrector: Recibe los ejercicios a través del chatbot y ejecuta los tests para obtener una puntuación, esta puntuación se la devuelve al chatbot.

En la Figura 6.3 podemos ver como interactúan los diferentes bloques entre sí, donde el chatbot es el bloque "central" que interactúa con los demás bloques para obtener respuestas o datos. Con el *framework conversacional* para obtener respuestas a preguntas del usuario en lenguaje natural. Con el *manager database* para poder comunicarse con la base de datos, ya sea para guardar nueva información o para hacer consultas a las tablas. Con el *corrector* para corregir los ejercicios enviados por los alumnos y obtener una puntuación. Y finalmente, con el *sistema de gamificación* para abordar todos los aspectos vinculados a la gamificación.

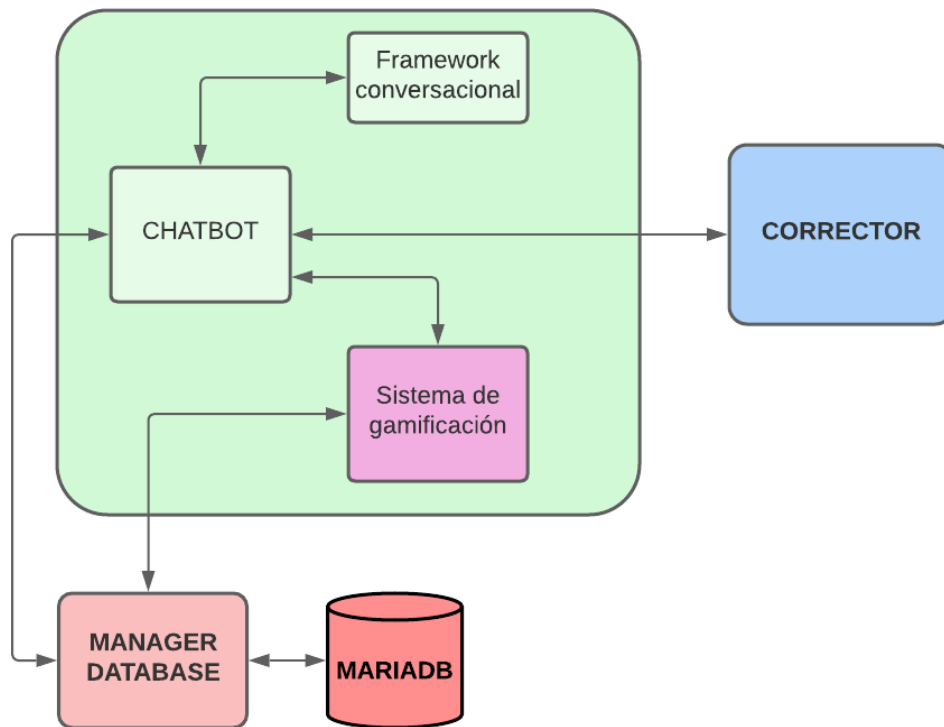


Figura 6.3: Módulos del sistema del chatbot. Fuente: Propia

En la Figura 6.4 podemos ver como se relacionan los componentes, así como la dirección de la información que se maneja. Dentro del sistema gamificado tendremos un componente con la *Logica de Gamificación* necesaria para gestionar los casos de uso. Esta parte de lógica es la que interactúa con el chatbot y el *manager database* para gestionar los puntos y niveles de los jugadores, sus ejercicios disponibles y las rachas. Otro bloque enlazado a la lógica es el de los *Mensajes de Gamificación*. A la hora de devolver un mensaje a un jugador se ha desarrollado un módulo específico para ello, ya que estos mensajes tendrán elementos gamificados, así como la clasificación, o elementos multimedia como gifs e imágenes. Estos elementos multimedia se recuperan de un directorio donde están almacenados. Ahí se guardan tanto gifs enviados como logros, por subir de nivel, desbloquear nuevos grupos de ejercicios, etc; como imágenes enviadas como recompensa, siendo estas imágenes con chistes relacionados con la informática.

En el módulo de *Lógica de Gamificación* se encuentra el fichero *game_manager.py*. En el

módulo de *Mensajes de Gamificación* se encuentra el fichero *game_messages.py*.

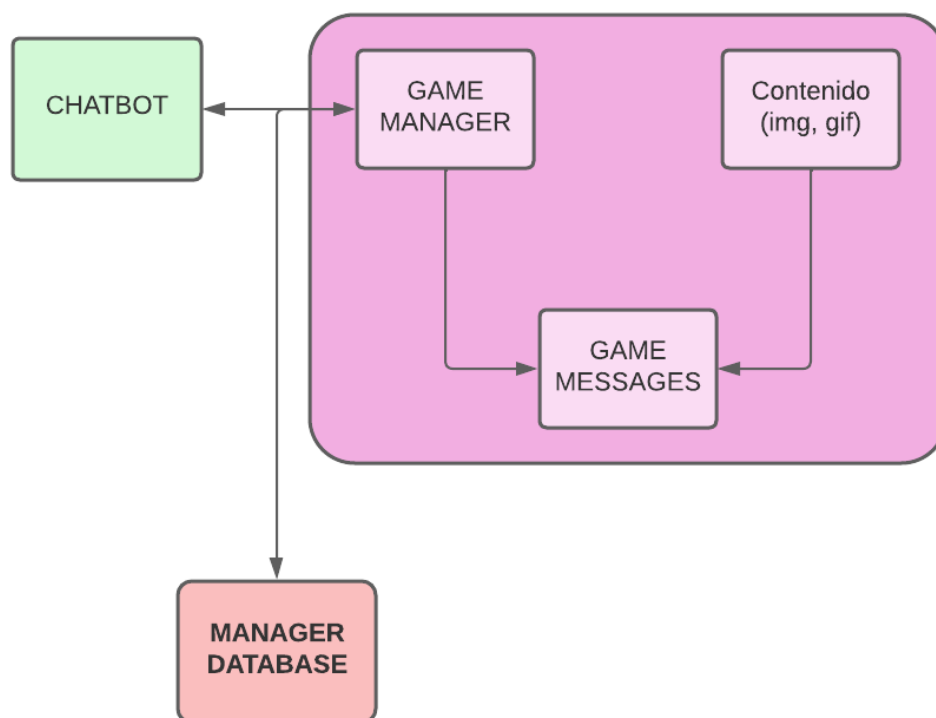


Figura 6.4: Módulos del sistema gamificado del chatbot. Fuente: Propia

En las siguientes figuras se pueden visualizar como interactúan los módulos entre sí con un diagrama de flujo para diferentes casos de uso.

En la Figura 6.5 se describe el flujo alternativo del *UC-2 Registro jugador*, para registrar a un usuario no registrado como jugador.

Primero, el usuario no registrado escribe en chat el comando `"/game"`, y como es un usuario no registrado, el *chatbot* le devuelve el formato correcto de su petición, que sería `"/game <nombre><mail>"` para poder registrar correctamente al jugador. De esta manera, cuando el *chatbot* recibe el comando con los parámetros correctos, envía al *manager_db* la petición de añadir un nuevo usuario a la base de datos. Una vez se registra al usuario, el chatbot le pasa el testigo a *game_manager* para cambiar el modo del nuevo usuario a jugador. Este le traspassa la petición a *manager_db*, que hará la actualización en la base

de datos. Y, finalmente, se le envía al jugador un mensaje conforme ahora está en modo jugador.

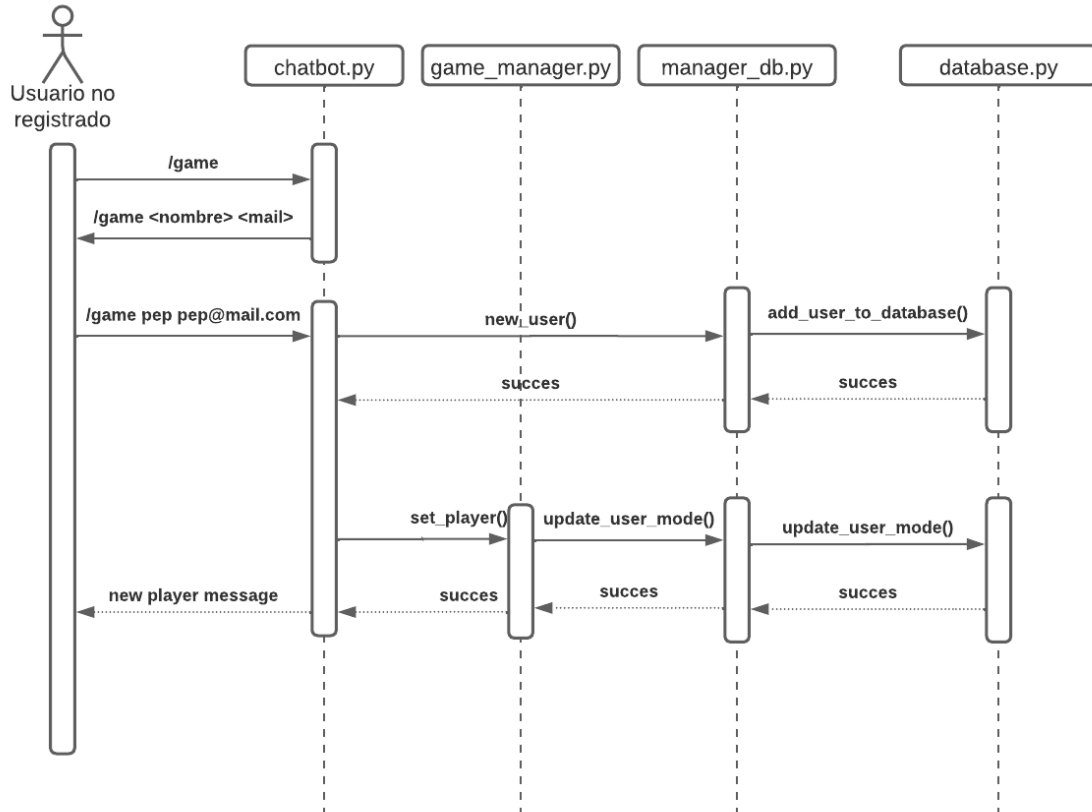


Figura 6.5: Diagrama de secuencia UC-2. Fuente: Propia

En la Figura 6.6 se describe el *UC-6 Pedir ejercicios*, donde se muestra el flujo que seguirá nuestro sistema para enviarle un ejercicio a un jugador. En este flujo se aprecia una alta interacción entre el *chatbot* y el *game_manager*. Esto se debe a que el *chatbot* le dice al *game_manager* que gestione el mensaje, ya que se trata de un jugador, pero el *game_manager*, usa funciones ya implementadas en *chatbot* como *get_file_subject()* para conseguir un ejercicio una vez obtenidos el tipo y la dificultad del mismo, y *send_file()* para hacer llegar el ejercicio al jugador.

También vemos que hay un flujo que se repite en dos ocasiones, *exercises_available()* en el *game_manager* que recibe una lista de los grupos de ejercicios disponibles, se usa en una

primera instancia para devolver al jugador la lista de los ejercicios disponibles y en una segunda instancia para comprobar que el grupo de ejercicios escogido está dentro de los grupos disponibles para el jugador.

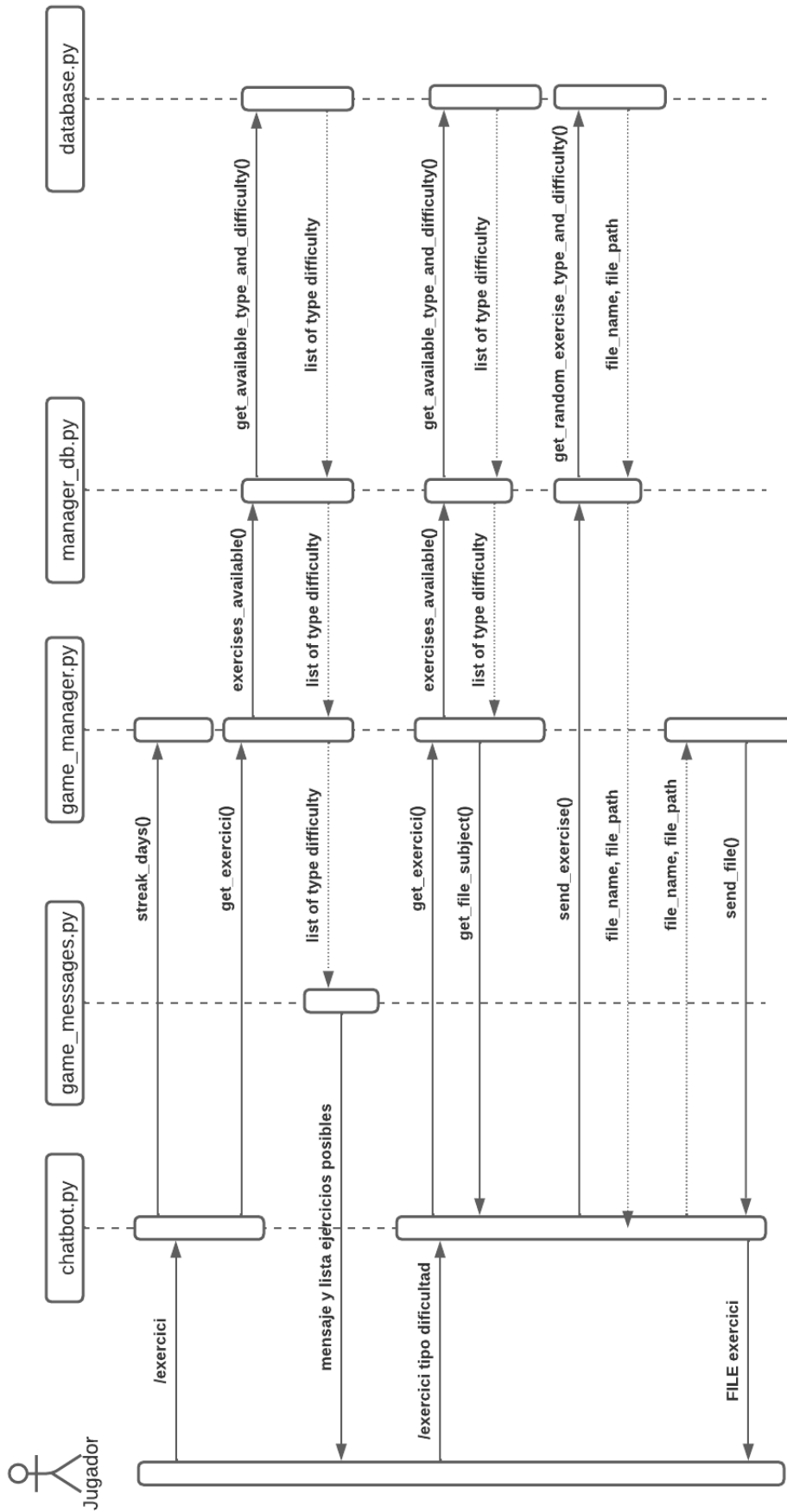


Figura 6.6: Diagrama de secuencia UC-6. Fuente: Propia

Por último, la Figura 6.7 describe el *UC-8 Resolver ejercicio*. Este caso de uso se encarga de procesar la puntuación recibida cuando un jugador envía un ejercicio para corregir y las consecuentes mecánicas de juego.

En primer lugar, cuando el *chatbot* recibe un fichero con la solución a un ejercicio que ha realizado el alumno y que pide corregir al chatbot, ya sea de un usuario o de un jugador, éste se lo envía al *corrector* para que le pase los tests, sin pasar por el *game_manager* en ninguno de los casos. Una vez recibida la puntuación de la corrección, el *chatbot* le pasa al *game_manager* la puntuación para que éste la gestione. Aquí es donde se encuentra la lógica de la mayoría de las mecánicas de juego que se diseñaron en el apartado 6.1.

Al recibir la puntuación del usuario, *game_manager* le dice al *game_messages* que le envíe el resultado al jugador, y al mismo tiempo le pasa a *manager_db* la tarea de guardar en base de datos la puntuación para el ejercicio en cuestión y de sumar la puntuación adquirida en la corrección a la puntuación total del jugador. A la hora de actualizar la puntuación del jugador se comprueba si este ha subido de nivel al pasar de los 100 puntos, y esta información se traspassa al *game_manager* de nuevo. Si este ha subido de nivel *game_messages* se encarga de enviarle al jugador el gif correspondiente conforme se felicita al jugador por subir de nivel.

Después, independientemente de si el jugador ha subido o no de nivel, se comprueba si este ha desbloqueado un nuevo grupo de ejercicios. Si es así, *game_messages* volverá a entrar en juego para enviar un gif al jugador donde se le felicita por haber desbloqueado nuevos ejercicios, y seguidamente, *game_messages* también se encarga de mostrarle al jugador una lista en el teclado con los tipos de ejercicios que tiene disponibles para pedirle al chatbot.

En este caso de uso también entran en juego otras mecánicas no descritas en la Figura 6.7. *game_manager* con la ayuda de *manager_db* comprueba si el jugador ha solucionado todos los ejercicios de un grupo (entendiendo como grupo el tipo y dificultad), por lo que igual que anteriormente se ha hecho, *game_messages* le enviaría al jugador un gif con la felicitación. Y si ese fuera el caso, que el jugador hubiese solucionado todos los ejercicios de un grupo, también se comprobaría si este era el último grupo que le quedaba por completar, lo que significaría que el jugador ha resuelto todos los ejercicios disponibles en el chatbot, y se le enviaría en texto de felicitación junto con links a páginas externas para poder seguir practicando.

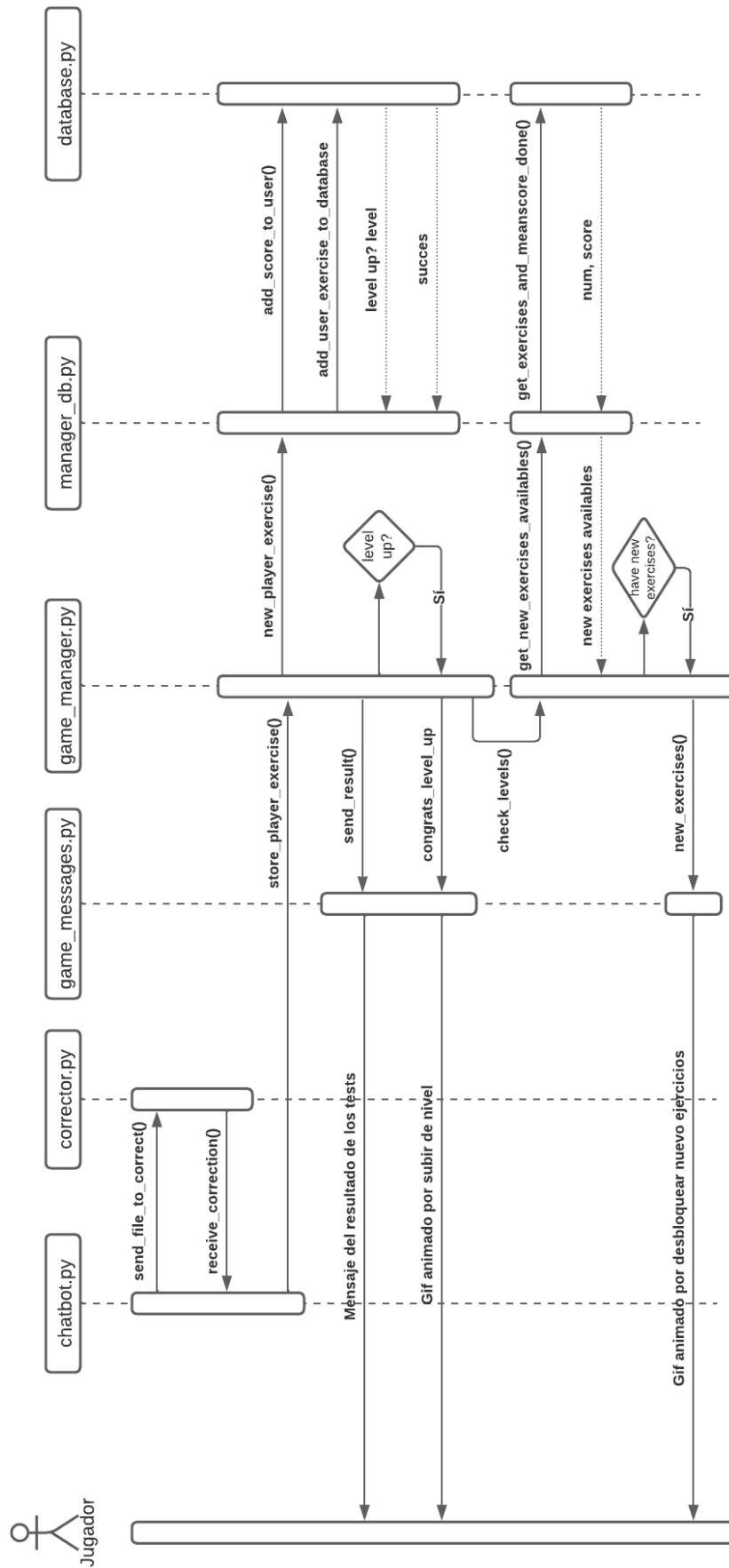


Figura 6.7: Diagrama de secuencia UC-8. Fuente: Propia

6.3 Diseño de la base de datos

Para guardar el progreso de los jugadores, los puntos, rachas, etc, se ha tenido que modificar la base de datos existente.

En la tabla existente de usuarios se han añadido dos campos nuevos, "subject", que definirá la asignatura actual del usuario para que cuando pida ejercicios se ahorre el paso de decir la asignatura a la que pertenece; y "isplayer", un campo booleano que definirá el modo de interacción del usuario con el chatbot, si este es verdadero el usuario interactuará con el modo gamificado del chatbot, pero si es falso interactuará con el modo normal ya existente.

Para gestionar los puntos de los usuarios se ha creado la tabla "score" que contiene el id del usuario, la asignatura, los puntos que lleva en el nivel, los puntos totales, el nivel en el que está, la cantidad de mensajes que ha enviado al chatbot y los días que lleva en racha interactuando con el chatbot y el último día de interacción. Un usuario tendrá tantas filas como asignaturas en las que haya participado en el chatbot.

Por último, he creado una última tabla *user_exercise_available*, para la gestión de los ejercicios disponibles. Esta tabla contiene el usuario, la asignatura y los ejercicios disponibles para esa asignatura. El campo de ejercicios disponibles es de un entero, que define la cantidad de niveles disponibles de la asignatura.

En la Figura 6.8 podemos ver como se relacionan las diferentes entidades en el diagrama de la base de datos, y las claves primarias (en negrita) de cada una. Los elementos en azul son los añadidos para la gamificación.

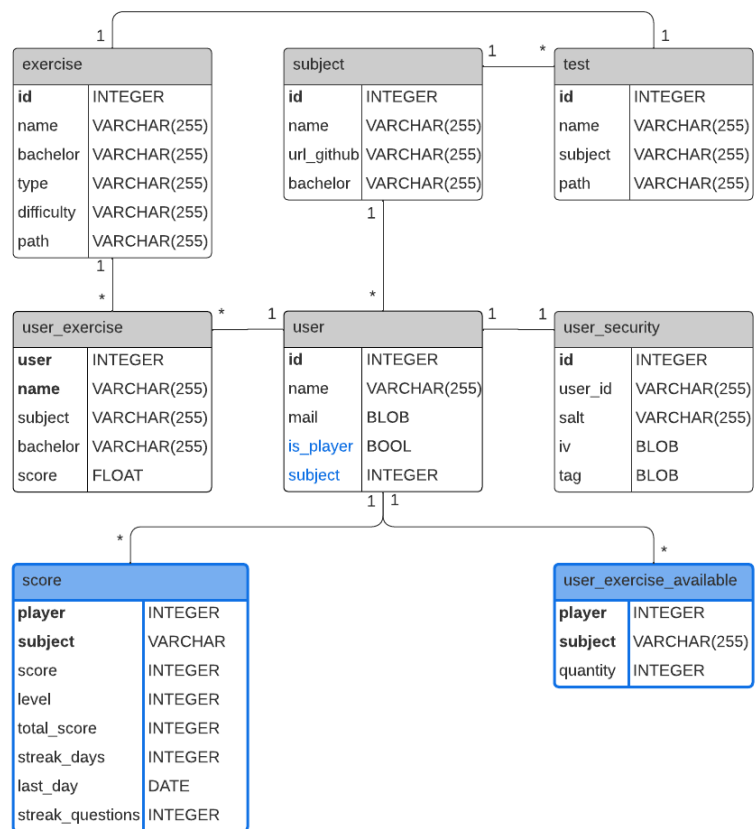


Figura 6.8: Diagrama base de datos. Fuente: Propia

Chapter 7

Implementación y resultados

7.1 Tecnologías relacionadas

7.1.1 Python

Python [3] es un lenguaje de programación interpretado, interactivo y orientado a objetos. Incorpora módulos, excepciones, tipado dinámico, tipos de datos dinámicos de alto nivel y clases. Admite múltiples paradigmas de programación más allá del orientado a objetos, como la programación procedural y funcional. Python combina un poder notable con una sintaxis muy clara. Tiene interfaces a muchas llamadas al sistema y bibliotecas, así como a varios sistemas de ventanas, y es extensible en C o C++. También se puede utilizar como lenguaje de extensión para aplicaciones que requieren una interfaz programable. Se ha usado Python 3.10 como lenguaje de programación del proyecto.

7.1.2 MariaDB

MariaDB Server [2] es uno de los servidores de bases de datos más populares del mundo. Está creado por los desarrolladores originales de MySQL. En este proyecto se ha usado como base de datos que almacena toda la información necesaria para el funcionamiento de la agente conversacional, incluida la gamificación.

7.1.3 Telegram

Telegram [4] es una plataforma enfocada en la mensajería instantánea, envío de archivos y la comunicación en masa. Telegram también permite el uso y creación de bots en su propia aplicación. Ofrece una gran facilidad para crear bots de la propia plataforma chateando con BotFather (bot que permite organizar los bots de cada usuario). Telegram también ofrece una API con la que se pueden enviar un conjunto de peticiones para la comunicación entre el código de un proyecto y el propio bot.

7.2 Implementación de la gamificación

Para aplicar el diseño de la gamificación explicado en la sección 6.1, se han tenido que añadir diferentes elementos de gamificación:

- Puntos
- Niveles
- Clasificación
- GIF animados para dar la enhorabuena
- Imágenes con frases o viñetas divertidas
- Enlaces a páginas externas que pueden ser de interés para el estudiante
- Dado como elemento de azar

7.2.1 Implementación de las mecánicas de juego

Los elementos mencionados anteriormente aparecen en diferentes momentos durante la interacción con el chatbot, según la mecánica que se esté llevando a cabo (ver Figura 6.1.1).

Para la mecánica *M1-Resolver ejercicio*, entran en juego los primeros tres elementos, los puntos, niveles, y clasificación. Estos elementos serán los elementos principales de nuestra gamificación y aparecerán en la mayoría de nuestras mecánicas de juego. En esta en concreto, cuando el jugador envía un ejercicio al chatbot para corregir, este obtiene

una puntuación, estos puntos se muestran al usuario y se le suman a sus puntos totales, los que definirán en qué posición de la clasificación se encuentra.

En la Figura 7.1¹ podemos ver un jugador que todavía no había conseguido ningún punto, pide la clasificación, el chatbot le dice que para conseguir punto puede hacer ejercicios, el jugador le pide un ejercicio y se lo envía resuelto, aquí se ejecuta la M1, entonces el jugador recibe tantos puntos como ha obtenido su corrección. Vuelve a pedir la clasificación y se le muestra la de su nivel en la asignatura que se encuentra.



Figura 7.1: Ejemplo de interacción donde un jugador (Laura) pide un ejercicio, lo resuelve (M1) y visualiza la clasificación. Fuente: propia

¹Para ver los ejemplos en video puedes acceder aquí

Para la mecánica *M2-Desbloqueo* aparece un nuevo elemento, el gif, una imagen animada para felicitar al jugador por haber conseguido desbloquear un nuevo grupo de ejercicios. Como podemos ver en la Figura 7.2, después de haber resuelto un número de ejercicios sobre un mismo grupo de ejercicios, en este caso, de *Secuenciales Alternativas Basico*, se le desbloquean al jugador los ejercicios del siguiente grupo, *Secuenciales Alternativas Avanzado*, se le envía el gif con la felicitación correspondiente y la lista con los grupos ahora disponibles.



Figura 7.2: Flujo de la mecánica M2. Fuente: propia

Para la mecánica *M3-Todos tema-nivel*, resolver todos los ejercicios de un grupo, vuelve a aparecer el gif, como en la anterior, pero esta mecánica, además, tiene recompensa de puntos. En la Figura 7.3, después de haber resuelto el último ejercicio de un grupo de ejercicios, en este caso, de *Secuenciales Alternativas Basico*, se le envía el gif con la

felicitación correspondiente y la recompensa de los 15 puntos por haber completado todos los ejercicios.

En la misma Figura 7.3, podemos ver que al recibir los 15 puntos de la recompensa de la mecánica *M3-Todos tema-nivel*, el jugador sube de nivel al haber alcanzado más de 100 puntos. Esto refleja la mecánica *M5-Subir de nivel*, que como las mecánicas *M2-Desbloqueo* y *M3-Todos tema-nivel* también recibe un gif de felicitación.

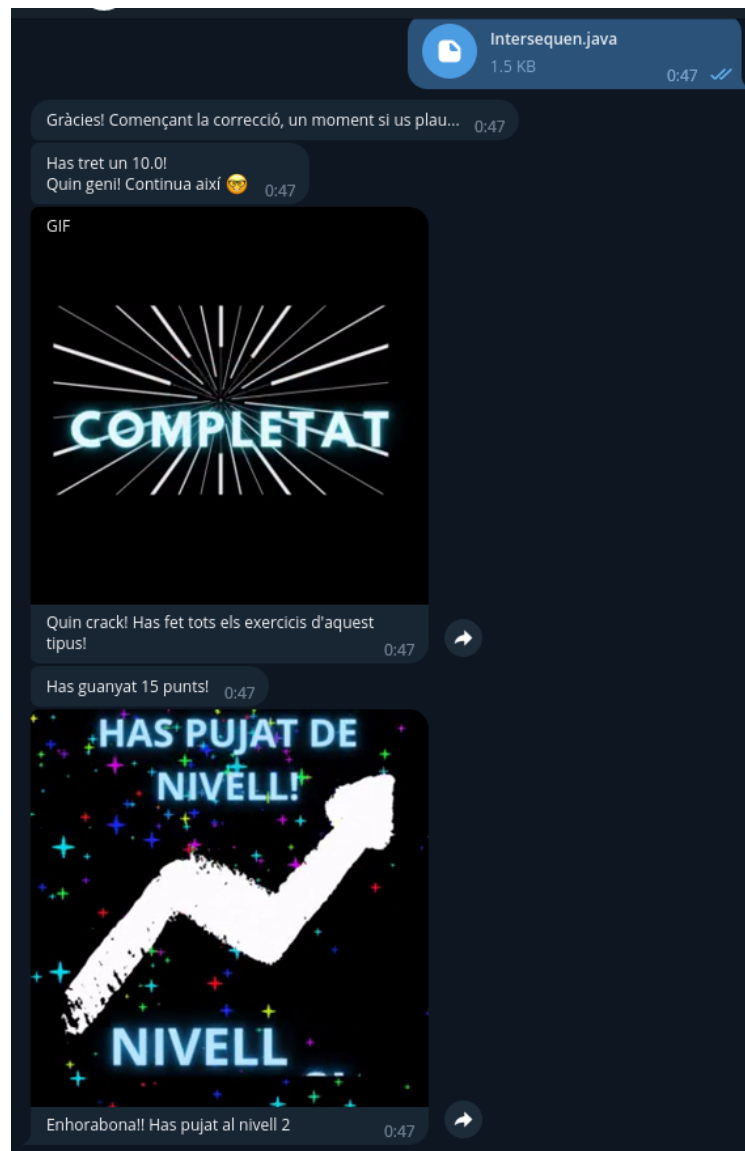


Figura 7.3: Flujo de las mecánicas M3 y M5. Fuente: propia

La mecánica *M₄-Todos asignatura*, describe cuando un jugador resuelve todos los ejercicios de la asignatura. Éste recibirá 30 puntos, y esta mecánica no solo incluye el elemento punto, sino que añade el elemento de enlace a una página externa.

En este caso se descartó usar el elemento del gif, ya que esta mecánica es consecuente a la mecánica *M₃-Todos tema-nivel*, en la que ya se usa el gif como recompensa. También, se puede dar, como en la Figura 7.3, que al recibir una gran cantidad de puntos, también se de la mecánica *M₅-Subir de nivel*, que también usa el gif como parte de la recompensa. De esta manera, el jugador recibe otro tipo de recompensa. Como podemos ver en la Figura 7.4, primero recibe la recompensa por la mecánica *M₃-Todos tema-nivel*, y después, recibe la recompensa por la mecánica *M₄-Todos asignatura*, una serie de enlaces a páginas externas donde se pueden practicar diferentes ejercicios de programación y 30 puntos.

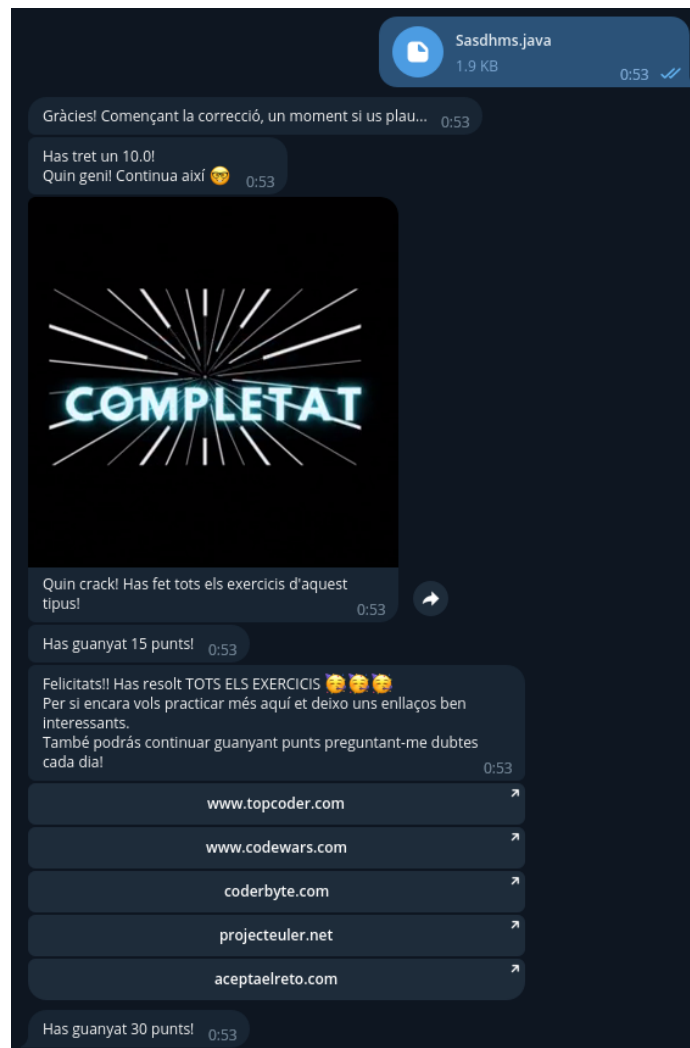


Figura 7.4: Flujo de la mecánica M4. Fuente: propia

La mecánica M6 contempla una recompensa al jugador cada 3 días consecutivos interactuando con el chatbot. Para hacer estas mecánicas más interesantes y que "enganchen" más al jugador, se le ha añadido el dado, un elemento de azar que decidirá la cantidad de puntos que reciba el jugador. En este caso, el jugador recibe una cantidad de puntos igual al número obtenido por el dado multiplicado por la cantidad de días en racha y dividido entre 3, ya que la recompensa es cada 3 días de racha.

En la Figura 7.5 se observa cómo el jugador ha hecho una interacción cualquiera, en este caso, pedir un ejercicio, y al ser la primera interacción del jugador el noveno día

consecutivo con el chatbot, este le ofrece la primera recompensa, que es un enlace aleatorio a una página externa para poder practicar más. Seguidamente, el chatbot envía un dado que rodará hasta caer con un número hacia arriba, calculará la puntuación que obtendrá el jugador y se le sumará a su puntuación total. Finalmente, el chatbot le dará respuesta a lo que el jugador le haya pedido.

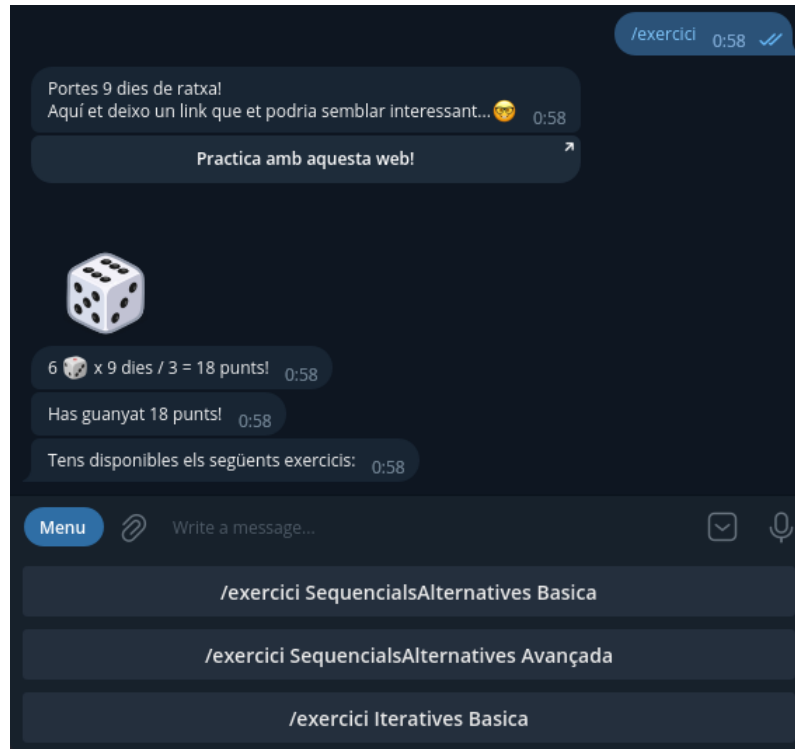


Figura 7.5: Flujo de la mecánica M6. Fuente: propia

Para acabar con la implementación y resultado de las mecánicas de juego, tenemos la mecánica M7, que cada 10 preguntas (o mensajes) al chatbot este le da al jugador una recompensa. Y aquí entra un elemento que aún no había aparecido en las mecánicas anteriores, la imagen con contenido gracioso. Se ha hecho una recopilación de imágenes chistosas de programación, y a la hora de dar esta recompensa se escogerá una al azar. En la Figura 7.6, el jugador le hace una pregunta al chatbot, y al ser la décima pregunta (o múltipla de diez) se da la condición de la mecánica M7 y le envía una imagen aleatoria con contenido cómico. En esta mecánica también entra en juego el dado, para una recompensa de puntos al jugador. En este caso el jugador gana tantos puntos como sale en el dado. Al terminar el flujo de la mecánica prosigue a responder la pregunta o mensaje del jugador.



Figura 7.6: Flujo de la mecánica M7. Fuente: propia

Chapter 8

Conclusiones y trabajo futuro

El objetivo principal del proyecto era implementar un sistema gamificado en la chatbot LovelaceUB, con mecánicas y elementos de juego para atraer a los alumnos y fomentar la interacción alumno-chatbot.

Y para ello he llevado a cabo las siguientes tareas:

- He investigado sobre la gamificación, los tipos de jugador y las mecánicas.
- He analizado otros proyectos de chatbots gamificados.
- He analizado y comprendido el código de LovelaceUB, en el que he implementado la gamificación.
- Me he basado en el Gamification Model Canvas para hacer el diseño de la gamificación del proyecto.
- He extendido la base de datos añadiendo columnas a tablas existentes y creando tablas nuevas para poder gestionar la gamificación de los usuarios.
- He implementado la lógica necesaria para hacer la gestión de la gamificación en el módulo del chatbot y separarla de la lógica para usuarios no gamificados.
- He implementado la lógica requerida para hacer la gestión de la gamificación en el módulo del manager de la base de datos creando nuevos *endpoints* y añadiendo nuevas *queries* para acceder a los datos de las tablas.
- He implementado una serie de métodos para enviar a los jugadores mensajes customizados de la gamificación.

- He diseñado gifs animados como elementos gamificados para dar recompensas a los jugadores.

Para seguir mejorando la chatbot LovelaceUB en la parte de la gamificación podrían hacerse diferentes trabajos:

- Estudio con los alumnos de la universidad el impacto que causa el chatbot gamificado. Ver si la gamificación cumple con las expectativas de atraer a más alumnos y aumentar la interacción entre los alumnos y el chatbot.
- Añadir nuevas mecánicas de juego, así como nuevos elementos gamificados. Telegram tiene un elemento muy interesante para usar en la gamificación, poll [5]. Este elemento es una encuesta, y se podría añadir para hacer preguntas tipo test a los alumnos.

Bibliography

- [1] Docker compose overview. <https://docs.docker.com/compose/>.
- [2] Mariadb. <https://mariadb.org/>.
- [3] Python documentation. <https://docs.python.org/3/faq/general.html#what-is-python>.
- [4] Telegram. <https://telegram.org/>.
- [5] Telegram polls. <https://telegram.org/blog/polls-2-0-vmq>.
- [6] venv — creation of virtual environments. <https://docs.python.org/3/library/venv.html>.
- [7] Rasa. <https://rasa.com> (2023).
- [8] ADAMOPOULOU, E., AND MOUSIADES, L. An overview of chatbot technology. In *Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16* (2020), Springer, pp. 373–383.
- [9] BARTLE, R. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD research* 1, 1 (1996), 19.
- [10] FADHIL, A., AND VILLAFIORITA, A. An adaptive learning with gamification & conversational uis: The rise of cibopolibot. In *Adjunct publication of the 25th conference on user modeling, adaptation and personalization* (2017), pp. 408–412.
- [11] FERNÁNDEZ, D. F. Agentes conversacionales educativos.
- [12] FERNÁNDEZ, H. ¿qué es la gamificación? aplicaciones y ejemplos reales. *Disponible en* <https://economytic.com/diccionario-tecnologico/gamificacion>.

- [13] HERE, A. Can duolingo’s chatbot teach you a foreign language? guardian. 7 october 2016. Available online: <https://www.theguardian.com/technology/2016/oct/06/duolingo-chatbots-learning-language>.
- [14] KAI CHOU, Y. User and player types in gamified systems - yu-kai chou.
- [15] LIU, L., SUBBAREDDY, R., AND RAGHAVENDRA, C. G. Ai intelligence chatbot to improve students learning in the higher education platform. *J. Interconnect. Networks* 22 (2022), 2143032:1–2143032:17.
- [16] MARCZEWSKI, A. Hexad: A player type framework for gamification design. *Gamification UK. Disponible en:[<https://www.gamified.uk/user-types/>]* (2018).
- [17] MOODLECHATBOT. The first moodle chatbot, at last. moodlemoot australia. 2017. Available online: <https://www.moodlenews.com/2017/the-first-moodle-chatbot-at-last-moodlemoot-australia-2017>.
- [18] PUIG, J. D. Dockers para un sistema de corrección de ejercicios en un chatbot.
- [19] RANOLIYA, B. R., RAGHUWANSHI, N., AND SINGH, S. Chatbot for university related faqs. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (2017), IEEE, pp. 1525–1530.
- [20] RUIZALBA ROBLEDO, J. L., NAVARRO LUCENA, F., AND JIMÉNEZ ARENAS, S. Gamificación como estrategia de marketing interno. *Intangible capital* 9, 4 (2013), 1113–1144.
- [21] SALAS, S. A. D., LONDOÑO, J. M. L., JOJOA, J. F. N., ARAUJO, E. A. T., AND SOLARTE, K. G. Estrategia de gamificación con tic para apoyar la humanización de la atención en los procedimientos médicos en adultos con cáncer. *Boletín Informativo CEI* 8, 1 (2021), 77–78.
- [22] SHORTT, M., TILAK, S., KUZNETCOVA, I., MARTENS, B., AND AKINKUOLIE, B. Gamification in mobile-assisted language learning: a systematic review of duolingo literature from public release of 2012 to early 2020. *Computer Assisted Language Learning* 36, 3 (2023), 517–554.
- [23] TAYLOR, Z., EGUILUZ, L., AND WHEELER, P. E. Ni máquina, ni humano, ni disponible: Do college admissions offices use chatbots and can they speak spanish? *Journal of Communication Technology* 5, 2 (2022).

- [24] TENORIO, S. ¿qué es gamification model canvas? *Disponible en* [<https://medium.com/@sergiotenoriovasquez/qué-es-gamification-model-canvas-24196f63c929>] (2022).

Chapter 9

Apéndice I - Manual técnico

9.1 Requisitos de software

Para la ejecución del proyecto en local hay dos opciones, ejecutar el proyecto *dockerizado* usando Docker-Compose [1] o sin *dockerizar*. Para la primera opción (usando Docker-Compose) será necesario tener instalado *Python 3.10*, *docker 20.10.21* y *docker-compose 1.29.2*.

En cambio, si se opta por la segunda opción, será necesario tener el programario anterior y también *MariaDB 10.6.12*, la última versión de *ConnectorC/MariaDB* para solucionar las dependencias que surgirían.

Todas las librerías de Python necesarias para cada módulo (*chatbo*, *manager_db*, *corrector*) se encuentran en su correspondiente fichero *requeriments.txt*. Se recomienda que cada módulo tenga un entorno virtual (*venv [6]*) donde instalar las librerías de su fichero *requeriments.txt*. Para hacer la instalación ejecutar para cada módulo:

```
pip install -r requeriments.txt
```

9.2 Variables de entorno

Cada uno de los módulos que tiene su entorno virtual también tendrá un *.env* para las variables de entorno. Además, también habrá un *.env* en la raíz del proyecto común para todos los módulos. A continuación se detallan los diferentes ficheros de variables, empezando por la raíz:

```

BUILD=local
#BUILD=docker
#BUILD=prod

CHATBOT_LOCAL_HOST = 0.0.0.0
CHATBOT_DOCKER_HOST = chatbot
CHATBOT_PORT = 8443

MANAGERDB_DOCKER_HOST = manager_db
MANAGERDB_LOCAL_HOST = 0.0.0.0
MANAGERDB_PORT = 5001

CORRECTOR_LOCAL_HOST = 127.0.0.1
CORRECTOR_PORT = 5002

MARIADB_PORT = 3306

MYSQL_DATABASE=manager_db_database
MYSQL_USER=manager
MYSQL_PASSWORD=#libre de escoger
MYSQL_DOCKER_HOST=db
MYSQL_LOCAL_HOST=127.0.0.1
MYSQL_ROOT_PASSWORD=#libre de escoger

```

Listing 9.1: Variables de entorno para la raíz */lovelaceUB/.env*

La variable `BUILD` define cómo se ejecutará y se escogerá el resto de variables en consecuencia, si se establece `BUILD=local`, se cogerán aquellas variables que contengan 'LOCAL' en su nombre, de la misma manera si se establece `BUILD=docker` se cogerán aquellas variables con 'DOCKER' en su nombre.

En la primera ejecución de *MariaDB* se deberá crear un nuevo usuario y definirlo en las variables de `MYSQL`.

Aquí se detallan las variables de entorno del módulo de *chatbot*:

```

TELEGRAM_WEBHOOK=https://b6ac-79-152-171-54.ngrok-free.app
TELEGRAM_BOT_TOKEN=#obtener token del bot
ADMIN_CHAT_ID=123456
PRIVATE_KEY=# string aleatorio

```

Listing 9.2: Variables de entorno para la raíz */lovelaceUB/chatbot/.env*

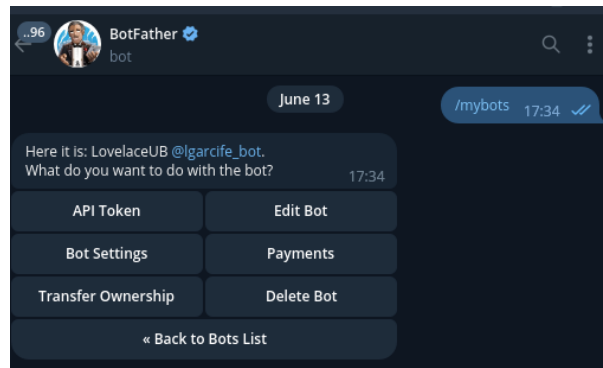


Figura 9.1: BotFather para obtener token. Fuente: Propia

En `TELEGRAM_WEBHOOK` pondremos el enlace obtenido de *ngrok*, necesario para la comunicación de los *webhooks*, véase la sección 9.3. Para obtener el token del bot original LoveLaceUB contactar con los autores. Si se desea probar el código con un bot propio, se puede obtener el bot mediante el bot de Telegram *BotFather*, enviando el comando `/mybots`, seleccionamos el bot y le damos a *API Token*, ver Figura 9.1

Se describen ahora las variables de entorno de *manager_db*:

```
GITHUB_TOKEN=#token de github
GITHUB_FATHER_REPO_URL=https://{GITHUB_TOKEN}@github.com/
    ChatbotTFG/
GITHUB_APPLICATION_REPO=ApplicationCode
GITHUB_PROGRAMACION1_REPO=programacion1
GITHUB_REPOS_SIGNATURES='[["programacion1","Informatica"]]'
```

Listing 9.3: Variables de entorno para la raíz `/lovelaceUB/manager_db/.env`

En este fichero se definen variables de entorno que referencian a los repositorios de GitHub donde se encuentran los enunciados y tests de los ejercicios.

9.3 Ejecución del código

Una vez tenemos el software correctamente instalado, como vamos a trabajar en local no podremos recibir peticiones externas, y para poder recibir los mensajes de Telegram a través de los *webhooks* tendremos que usar *ngrok*. Esto proporciona una URL HTTP que cuando recibe una petición la redirige a nuestra máquina. Para activar el *ngrok* ejecutar el siguiente comando substituyendo `<PORT>` por el puerto en el que se ejecutará el

chatbot, en nuestro caso el 8443:

```
ngrok http <PORT> --host-header=1.localhost:<PORT>
```

Cuando iniciemos el *ngrok* nos dará un enlace que tendremos que poner en la variable del `.env` del chatbot.

9.3.1 Ejecución con dockers

Una vez hecho todo el *setup* anterior, para la ejecución del proyecto se tiene que ejecutar:

```
docker-compose up --build
```

Este comando construye y levanta los Dockers de todos los módulos. Para detener la ejecución se ejecuta el siguiente comando:

```
docker-compose down
```

Esto eliminará todos los contenedores y la red de comunicación entre ellos.

9.3.2 Ejecución sin dockers

Para ejecutar el proyecto sin el `docker-compose` hay ejecutar cada módulo por separado.

En el módulo de chatbot y dentro de su entorno virtual ejecutamos:

```
python -u chatbot.py
```

En el módulo de `manager_db` y dentro de su entorno virtual ejecutamos:

```
gunicorn manager db:app -c gunicorn.conf.py
```

En el módulo de corrector y dentro de su entorno virtual ejecutamos:

```
gunicorn corrector:app -c gunicorn.conf.py
```