



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
Mención en Ingeniería del Software

**gpBOT: un bot de Discord para encuestas
gamificadas en actividades docentes**

Alumno:
José Borge Torres

Tutores:
Francisco José Andújar Muñoz
Rocío Carratalá Sáez



...

Agradecimientos

Doy las gracias a mis tutores Francisco José Andújar Muñoz y Rocío Carratalá Sáez que han estado ahí siempre que he tenido cualquier dificultad y me han apoyado durante todo el proyecto. También a Javier Bastida, Yuri Torres, Arturo González y Jesús Cámara por haberme dado su opinión y recomendaciones sobre el sistema realizando con ellos las pruebas de usuario de profesores junto a mis tutores y por facilitar la realización de de las pruebas de usuario a estudiantes, sin ellos no creo que ha sido capaz de poder realizar las pruebas de usuario. A los estudiantes de la asignatura Fundamento de Computación por haber realizado la prueba de usuario.

Gracias al Grupo Trasgo que han estado ahí para facilitar el trabajo y no haberse quejado de todas las reuniones que hemos hecho en su laboratorio.

Gracias al Departamento de Informática de la Escuela de Ingeniería Informática de la Universidad de Valladolid por prestarnos la máquina virtual que se ha usado para desplegar el sistema y a la Universidad de Valladolid ya que, este trabajo se ha desarrollado dentro del proyecto de innovación docente financiado por la misma universidad y titulado “Generalizando la integración de gamificación competitiva y colaborativa de forma ágil (III)”, con identificador PID2021_065.

A mi familia por aguantarme, mis amigos por apoyarme y compañeros por ayudarme. Ha sido una gran etapa de mi vida. Gracias.

Resumen

Gamificar consiste en utilizar elementos y técnicas de actividades lúdicas propias de los juegos en ámbitos que no lo son, con la finalidad de incentivar la motivación, interés y participación en estos. En el contexto de este proyecto se ha desarrollado un servicio que, integrándose en un servidor de Discord, permite la creación, edición y contestación de cuestionarios.

Actualmente, algunos profesores de la Escuela de Ingeniería Informática de la Universidad de Valladolid utilizan Discord como canal de comunicación adicional y opcional con los alumnos en determinadas asignaturas. La incorporación de gpBot a los servidores de Discord de dichas asignaturas permite implementar una competición gamificada basada en encuestas tras la finalización de las cuales se generan rankings según los aciertos y el tiempo de respuesta de los participantes. Cabe mencionar que solo los usuarios del servidor pueden acceder al servicio.

En este documento se presentan las distintas fases que han compuesto el diseño, desarrollo e implementación del proyecto, las tecnologías utilizadas y los resultados obtenidos, incluyendo una explicación detallada de las pruebas de uso realizadas en entornos reales y las conclusiones obtenidas tras las mismas.

Abstract

Gamifying consists of using elements and techniques of game-like activities in areas that are not games, in order to encourage motivation, interest and participation. In the context of this project, it has been developed a service that, integrated in a Discord server, allows the creation, edition, and solution of questionnaires.

Currently, some professors of the School of Computer Engineering of the Universidad de Valladolid use Discord as an additional and optional communication channel with the students in certain subjects. The incorporation of gpBot to the Discord servers of these subjects allows to implement a gamified competition, based on surveys after the completion of which rankings are generated, according to the number of correct answers and the response time of the participants. It is worth mentioning that only server users can access the service.

This document presents the different phases that have made up the design, development and implementation of the project, and the technologies used and the results obtained, including a detailed explanation of the use-case tests carried out in real environments, and the conclusions obtained after them.

Índice general

Agradecimientos	III
Resumen	V
Abstract	VII
Lista de figuras	XIII
Lista de tablas	XVII
1. Introducción	1
1.1. Contexto y motivación	1
1.2. Objetivos	3
1.3. Estado del arte	3
1.3.1. Kahoot!	3
1.3.2. Poll bot	4
1.3.3. Strawpoll	4
1.3.4. Moodle	5
1.3.5. Socrative	5
1.4. Estructura de la memoria	5
2. Requisitos y Planificación	7
2.1. Introducción	7

2.2. Requisitos	7
2.2.1. Requisitos funcionales	8
2.2.2. Requisitos de información	8
2.2.3. Requisitos no funcionales	8
2.2.4. Requisitos añadidos posteriormente	8
2.3. Planificación y metodología	13
2.3.1. Software y hardware	14
2.3.2. Presupuesto Inicial	15
2.3.3. Análisis riesgos	15
2.3.4. Seguimiento y desviaciones del plan inicial	17
2.3.5. Diagrama de Gantt final	18
2.3.6. Presupuesto Final	18
3. Análisis	21
3.1. Introducción	21
3.2. Características	21
3.3. Modelo conceptual	22
3.4. Tipos de usuario	22
3.5. Casos de uso	23
3.6. Análisis de los casos de uso	24
3.7. Máquina de estados de encuestas	30
4. Tecnologías utilizadas	33
4.1. Introducción	33
4.2. Gestión del proyecto	33
4.3. Backend	34
4.4. Frontend	34
4.5. Enlace del sistema con Discord	35

5. Diseño	37
5.1. Introducción	37
5.2. Arquitectura	37
5.3. Recursos	38
5.4. Modelo lógico de datos	38
5.5. Despliegue	39
6. Implementación y pruebas	47
6.1. Introducción	47
6.2. Implementación del sistema	47
6.2.1. Estructura de directorios	47
6.2.2. Vistas del sistema	48
6.3. Pruebas del sistema	56
6.3.1. Pruebas de integración	56
6.3.2. Pruebas de usuario	64
7. Conclusiones	71
7.1. Conclusiones	71
7.2. Valoraciones personales	72
7.3. Líneas de trabajo futuras	72
Bibliografía	75
A. Manuales	79
A.1. Manual de despliegue e instalación	79
B. Resumen de enlaces adicionales	81

Lista de Figuras

2.1. Diagrama de Gantt inicial.	13
2.2. Diagrama de Gantt final.	18
3.1. Diagrama de clases: modelo conceptual.	23
3.2. Diagrama de casos de uso	24
3.3. Diagrama de secuencia: caso de uso <i>Identificarse</i>	25
3.4. Diagrama de secuencia: caso de uso <i>Identificarse como Administrador</i>	25
3.5. Diagrama de secuencia: caso de uso <i>Añadir Bot</i>	26
3.6. Diagrama de secuencia: caso de uso <i>Crear encuesta</i>	26
3.7. Diagrama de secuencia: caso de uso <i>Editar encuesta</i>	27
3.8. Diagrama de secuencia: caso de uso <i>Publicar encuesta</i>	28
3.9. Diagrama de secuencia: caso de uso <i>Obtener encuesta</i>	28
3.10. Diagrama de secuencia: caso de uso <i>Obtener Resultados</i>	29
3.11. Diagrama de secuencia: caso de uso <i>Borrar resultados</i>	29
3.12. Diagrama de secuencia: caso de uso <i>Borrar encuesta</i>	30
3.13. Máquina de estados de una encuesta.	31
5.1. Diagrama de paquetes de la arquitectura.	39
5.2. Diagrama de clases (ModulesAndUsesStyle) de “client”.	40
5.3. Diagrama de clases (ModulesAndUsesStyle) de “surveys”.	41

5.4. Diagrama de clases (ModulesAndUsesStyle) de “discordbot”	41
5.5. Diagrama de clases (ModulesAndUsesStyle) de “gateway”	42
5.6. Diagrama de clases de los Recursos REST	43
5.7. Diagrama de clases del Modelo Lógico de Datos.	44
5.8. Diagrama de Despliegue.	45
6.1. Estructura de directorios del sistema.	48
6.2. Pantalla (vista) de autenticación.	49
6.3. Pantalla (vista) inicial donde se muestra la política de privacidad.	49
6.4. Pantalla (vista) para responder una pregunta.	50
6.5. Pantalla (vista) para responder una pregunta y terminar la encuesta.	50
6.6. Pantalla (vista) de administración del sistema.	50
6.7. Pantalla (vista) de login para acceder a la administración del sistema.	51
6.8. Pantalla (vista) del perfil de profesor.	51
6.9. Pantalla (vista) del perfil de usuario.	51
6.10. Pantalla (vista) del ranking correspondiente a una determinada encuesta.	52
6.11. Pantalla (vista) del perfil de usuario con resultados.	53
6.12. Pantalla (vista) para crear una encuesta.	54
6.13. Pantalla (vista) para añadir una pregunta a una encuesta.	54
6.14. Pantalla (vista) para añadir una respuesta a una encuesta.	54
6.15. Pantalla (vista) para crear una encuesta y enviarla.	55
6.16. Pantalla (vista) del perfil de profesor con resultados.	55
6.17. Pantalla (vista) del comando de publicación de una determinada encuesta.	55
6.18. Clasificación de los participantes en la prueba de usuario según el dispositivo utilizado.	65
6.19. Respuestas obtenidas para la pregunta P1 tras la realización de la prueba de usuario del sistema.	67
6.20. Respuestas obtenidas para la pregunta P2 tras la realización de la prueba de usuario del sistema.	67

6.21. Respuestas obtenidas para la pregunta P3 tras la realización de la prueba de usuario del sistema.	68
6.22. Respuestas obtenidas para la pregunta P4 tras la realización de la prueba de usuario del sistema.	68
6.23. Respuestas obtenidas para la pregunta P5 tras la realización de la prueba de usuario del sistema.	69
6.24. Respuestas obtenidas para la pregunta P6 tras la realización de la prueba de usuario del sistema.	69
6.25. Respuestas obtenidas para la pregunta P7 tras la realización de la prueba de usuario del sistema.	70
6.26. Respuestas obtenidas para la pregunta P8 tras la realización de la prueba de usuario del sistema.	70
A.1. Salida del comando en una máquina en la que el puerto 80 sí esta en uso. . .	79
A.2. Estadísticas de Docker.	80

Lista de Tablas

2.1. Terminología utilizada en los requisitos.	8
2.2. Requisitos funcionales.	9
2.3. Requisitos de información (primera parte).	10
2.4. Requisitos de información (continuación).	11
2.5. Requisitos no funcionales.	12
2.6. Requisitos añadidos con posterioridad al inicio del desarrollo del proyecto. . .	12
2.7. Detalle del software utilizado en el proyecto.	14
2.8. Detalle del hardware utilizado en el proyecto.	14
2.9. Presupuesto inicial del proyecto.	15
2.10. Matriz de riesgos probabilidad e impacto.	15
2.11. Riesgo R01: el desarrollador suspende alguna asignatura.	16
2.12. Riesgo R02: el desarrollador enferma.	16
2.13. Riesgo R03: problemas de comunicación con los tutores.	16
2.14. Riesgo R04: las características de la máquina virtual impiden el despliegue. .	17
2.15. Riesgo R05: el servicio externo Discord no se puede integrar.	17
2.16. Riesgo R06: modificación de los requisitos.	17
2.17. Tabla de presupuesto final.	19
6.1. Caso de prueba <i>Crear encuesta</i>	56
6.2. Caso de prueba <i>Obtener encuesta completa</i>	56

6.3. Caso de prueba <i>Editar encuesta.</i>	57
6.4. Caso de prueba <i>Obtener encuesta.</i>	57
6.5. Caso de prueba <i>Borrar encuesta.</i>	57
6.6. Caso de prueba <i>Responder encuesta especificada.</i>	58
6.7. Caso de prueba <i>Obtener resultados de la encuesta especificada.</i>	58
6.8. Caso de prueba <i>Obtener encuestas completas.</i>	58
6.9. Caso de prueba <i>Obtener encuestas de usuario con id especificado.</i>	59
6.10. Caso de prueba <i>Obtener resultados de usuario con id especificado.</i>	59
6.11. Caso de prueba <i>Obtener encuestas de profesor con id especificado.</i>	59
6.12. Caso de prueba <i>Obtener resultados de profesor con id especificado.</i>	60
6.13. Caso de prueba <i>Obtener soluciones.</i>	60
6.14. Caso de prueba <i>Generar code.</i>	60
6.15. Caso de prueba <i>Obtener una lista de los servidores en los que una encuesta ha sido publicada.</i>	60
6.16. Caso de prueba <i>Obtener una lista los servidores en los que una encuesta ha sido publicada.</i>	61
6.17. Caso de prueba <i>Obtener el id a partir de un code.</i>	61
6.18. Caso de prueba <i>Iniciar sesión.</i>	61
6.19. Caso de prueba <i>Iniciar sesión como administrador.</i>	61
6.20. Caso de prueba <i>Comprobar encuesta.</i>	62
6.21. Caso de prueba <i>Enviar URL del cliente al backend.</i>	62
6.22. Caso de prueba <i>Listar profesores.</i>	62
6.23. Caso de prueba <i>Comprobar profesor.</i>	62
6.24. Tabla de resultados de Casos de Prueba	63
6.25. Preguntas planteadas en la encuesta realizada a los/as alumnos/as tras completar la prueba de usuario.	66

Capítulo 1

Introducción

1.1. Contexto y motivación

El término “*gamificación*” hace referencia a la introducción de elementos de los juegos en ámbitos que no son considerados como tal [6]. El objetivo de la *gamificación* es que actividades que normalmente no son lúdicas se conviertan en lúdicas. Existen actividades que pueden no resultar atractivas pero forman parte de las obligaciones de una persona. Transformarlas en tareas lúdicas permite que su realización sea incentivada. Por ejemplo, la educación en un contexto poco estimulante dificulta la asimilación del conocimiento que se pretende adquirir durante el aprendizaje. Muchos estudiantes consideran que la enseñanza tradicional resulta aburrida y que el uso de juegos incentivaría el aprendizaje, haciendo de la *gamificación* una estrategia docente atractiva [31]. Diversos estudios corroboran que el uso de la *gamificación* mejora el rendimiento académico cuando esta es usada en un contexto educativo [3], la conducta [17], el interés y la motivación del alumnado [22]. Aumentar el interés del alumnado durante el aprendizaje también estimula el afán del mismo por realizar y completar sus tareas con éxito [8]. Así pues, el profesorado puede incluir elementos del diseño de juegos para atraer al alumnado y mantener su interés durante el aprendizaje.

Las técnicas gamificadas pueden aplicarse, por ejemplo, para mejorar métodos docentes tradicionales como la “técnica de la pregunta” que, a pesar de su potencial, suele tener muy mala acogida entre el alumnado. Algunos alumnos interpretan el hecho de tener que responder como un castigo del profesor, otros prefieren no participar debido al miedo a equivocarse públicamente delante de sus compañeros. En cambio, realizar concursos a través de plataformas virtuales, incluso respondiendo a las mismas preguntas que se harían con el método tradicional, aumenta la predisposición de los alumnos a participar y resolver la cuestión, más aún si pueden participar y mantener el anonimato.

Sin embargo, la sola aplicación de la gamificación en las metodologías docentes puede tener efectos contraproducentes, sobre todo si se utilizan enfoques competitivos. El comportamiento competitivo puede reducir la frecuente colaboración entre estudiantes para resolver

cuestiones o problemas comunes. Por eso, es conveniente combinar el uso de procesos competitivos y colaborativos, ya que ambos están interrelacionados [5]. Es posible alcanzar una mejora significativa del aprendizaje mediante estrategias de carácter gamificador que permitan la combinación e integración de colaboración y competición entre el alumnado [20]. Una forma específica de llevar a cabo la gamificación colaborativa es mediante la evaluación de la participación en una herramienta *online* colaborativa y recompensarla. La participación se cuantifica mediante el número de mensajes en un foro y el reconocimiento técnico y social entre los estudiantes del mismo. Según [15], esta estrategia incrementa la participación, motivación y colaboración entre los estudiantes durante las evaluaciones.

Según las conclusiones de [15], la opinión del alumnado sobre Moodle es que no es una plataforma suficientemente ágil para realizar el proceso de comunicación a través de los foros de la herramienta, prefiriendo trabajar con otras plataformas más accesibles. Por ejemplo, la plataforma de mensajería instantánea *Discord* (<https://discordapp.com/>) es una herramienta habitualmente empleada para la comunicación entre jugadores de videojuegos mediante canales de mensajes de texto o canales de voz y, además, es accesible mediante aplicación de escritorio, sitio web o aplicación móvil. *Discord* permite reaccionar al contenido que se publica en un canal mediante el uso de iconos permitiendo mostrar reconocimiento técnico y social. El uso de *Discord* en clases del Grado de Informática de la UVa tuvo una buena acogida dentro del alumnado, aumentando su participación notablemente y siendo preferido su uso al de las herramientas propias del entorno del Campus Virtual de la UVa [1, 15].

En el marco de estas nuevas metodologías docentes, se desarrollan los proyectos de innovación docente del grupo Trasgo, en los que también colaboran otros profesores del departamento de Informática de la UVa. En estos proyectos se usan técnicas de gamificación competitiva y colaborativa. En concreto, *Tablon* es una herramienta para introducir metodologías de gamificación en cursos de informática [28]. Esta herramienta se está usando a día de hoy en diversas asignaturas del grado de Informática, como Computación Paralela [13] o Arquitectura y Organización de Computadoras [1], combinado con la comunicación con el alumnado mediante *Discord*.

Dentro del proyecto de innovación docente “Generalizando la integración de gamificación competitiva y colaborativa de forma ágil (III)” de la UVa [14], se pretende crear una aplicación que permita la realización de encuestas y concursos gamificados para modernizar la “técnica de la pregunta”. Esta aplicación debe integrarse en el entorno de *Discord*, ya que es utilizado actualmente para comunicarse con el alumnado. Así surge la idea de *gpBot* (*Gamified Poll Bot*): un bot que servirá para realizar encuestas gamificadas a través de *Discord*. *gpBot* permitirá a los alumnos de cada asignatura donde se utilice *Discord* acceder a las encuestas y contestarlas, siendo registradas para la posterior evaluación del profesorado, ya sea tanto por la simple participación del alumnado, o por la correcta contestación a las preguntas planteadas. Además, esta herramienta deberá mostrar una clasificación de los alumnos con mejor desempeño en la encuesta, gamificando la actividad. También se desea que las encuestas puedan contestarse de forma anónima para que así, los alumnos más reacios a exponerse públicamente, puedan participar en la actividad, de tal forma que sus compañeros no puedan saber sus respuestas, pero sí los profesores.

1.2. Objetivos

El objetivo principal de este proyecto es la creación de *gpBot*: un aplicación que permita crear y gestionar encuestas para realizar actividades gamificadas utilizando Discord como medio de comunicación y autenticación. Para ello, a continuación se describen los objetivos que se han fijado para el desarrollo de este proyecto:

- Objetivo 1: crear un software de monitorización de servidores de Discord mediante un bot para leer los comandos que se ejecuten en dicho servidor, así como monitorizar la entrada y salida de los usuarios del mismo y de unos usuarios especiales que tendrán el rol de “Profesor”.
- Objetivo 2: crear una base de datos relacional que almacene de forma persistente toda la información relevante del dominio.
- Objetivo 3: crear un software que haga las funciones de API del sistema que gestione la persistencia del Objetivo 2 y suministre la lógica del sistema.
- Objetivo 4: crear un software que haga las funciones del cliente web y encargado de que los usuarios finales puedan tener acceso a la API desarrollada en el Objetivo 3.
- Objetivo 5: configurar una máquina virtual que aloje todo el software mencionado en los objetivos anteriores, permitiendo su ejecución, despliegue y utilización.
- Objetivo 6: esta herramienta debe ser accesible mediante dispositivos como móviles, ordenadores, portátiles o tablets, por lo que su diseño debe ser *responsive*.
- Objetivo 7: comprobar la usabilidad y correcto funcionamiento de la herramienta diseñada en un entorno de clase real, con alumnos habituados al uso de Discord para comunicarse con el profesorado.

1.3. Estado del arte

En esta sección se realiza una revisión de las soluciones existentes que podrían ser alternativas a la herramienta que se pretende desarrollar. Todas las herramientas analizadas permiten de algún modo la gestión de encuestas gamificadas en actividades docentes, dado que el servicio a desarrollar persigue que el proceso de la enseñanza y evaluación mediante encuestas adquiera un carácter lúdico. Para cada alternativa listada, se presenta un análisis de sus características, qué ventajas tienen y cuáles son sus inconvenientes frente a dicho servicio.

1.3.1. Kahoot!

Kahoot! [19] es una aplicación gratuita con el mismo propósito que el sistema desarrollado: la realización de cuestionarios de evaluación. Tiene una versión web y una app. Mediante el

uso de este software, el profesor tiene la posibilidad de crear un cuestionario, los alumnos eligen un nickname y, mediante su dispositivo móvil, pueden contestarlo.

Sus principales ventajas frente a gpBot son que tiene dos modos de participación (grupal e individual) y que permite la incorporación de fotos y vídeos. Sin embargo, existen algunas limitaciones por las que se considera que esta aplicación no cumple con todas las características deseadas que justifican el desarrollo de este servicio. En primer lugar, solo permite contestar a los cuestionarios a través de la aplicación móvil, no mediante web. Además, al responder a un cuestionario, en el móvil solo aparecen las respuestas posibles identificadas con un símbolo y color, por lo que es necesario alternar constantemente la vista entre el móvil y el monitor donde el profesor (a modo de administrador de la encuesta) está mostrando la pregunta y las respuestas posibles, lo cual dificulta la usabilidad. Por último, no se garantiza la autenticación de los participantes: cualquier persona con acceso al código del cuestionario podría realizarlo. Cabe destacar que, en el contexto donde pretende utilizarse gpBot, la autenticación y validación es imprescindible y forma parte de los requisitos de el sistema.

1.3.2. Poll bot

Se trata de un bot de Discord que permite que permite la creación y contestación de encuestas a través de comandos en un canal de texto de Discord.

Si bien este bot cumple con la restricción de que solo los usuarios del servidor puedan contestar a la encuesta, pero tiene algunas limitaciones. La razón principal que justifica prescindir de Poll bot es que no se pueden contestar las preguntas a través de una aplicación o sitio web, sino que es necesario contestar a través del canal de Discord y solo permite preguntas con respuestas de verdadero o falso. Para crear encuestas más complejas usa el servicio de *Strawpoll*, que también presenta ciertas desventajas, como se detalla a continuación.

1.3.3. Strawpoll

Este sitio web que permite la creación y contestación de encuestas. Strawpoll tiene un enfoque orientado a la relación de sondeos y no a la evaluación de los usuarios.

El inconveniente principal del uso de esta herramienta es la dificultad de realizar evaluación de usuarios, ya que imposibilita el cálculo de algunas de las métricas que se consideran parte de las estrategias de gamificación, como el número de respuestas acertadas o el tiempo de respuesta. Por último, no permite la validación y autenticación de los usuarios que pueden contestar a las encuestas. En su lugar, usa una validación para la no constitución duplicada por IP, por lo que no es posible identificar al alumno que realizó la encuesta. Además, esta validación es un inconveniente para realizar encuestas desde clase, ya que, si dos usuarios están conectados a la misma red y quieren contestar (como sucedería en clase si estuvieran conectados a Eduroam), solo podría contestar el primer usuario que realizase la encuesta.

1.3.4. Moodle

Moodle es una herramienta para la gestión docente en línea [21]. Además de permitir la creación de encuestas, tiene otras muchas funcionalidades centradas en mejorar el aprendizaje de los estudiantes. El alcance de esta herramienta es mayor que el alcance del sistema en desarrollo, pero, al igual que las demás alternativas, no es una opción que ofrezca las ventajas del sistema desarrollado para este TFG.

La principal desventaja de Moodle es que la forma de acceso a los cuestionarios es engorrosa, implica autenticarse en Moodle, buscar la asignatura en la que se a publicado el cuestionario y, en ella, buscar el mismo. Frente a eso, es un requisito del proyecto que el acceso a las encuestas sea lo más sencillo posible. Sumado a esto, Moodle no ofrece alternativas para realizar actividades gamificadas.

1.3.5. Socrative

Socrative es una herramienta gratuita que permite la creación y gestión de cuestionarios, así como la gestión de los usuarios mediante su registro en la aplicación [27]. Esta podría ser una alternativa factible si no fuera por las limitaciones que a día de hoy presenta, entre las que se destaca el hecho de que solo permite, en su versión gratuita, una sala pública de hasta cincuenta participantes y con un máximo de cinco cuestionarios.

1.4. Estructura de la memoria

Este documento se estructura de la siguiente forma:

Capítulo 1 Introducción: expone el contexto, motivación y objetivos del proyecto, además de una revisión de herramienta similares a la que se pretende desarrollar.

Capítulo 2 Requisitos y planificación: describe los requisitos que se han solicitado, así como al alteración de los mismos tras observar la evolución del proyecto, y la planificación y gestión del proyecto, tanto metodologías teóricas que se prendían usar como metodología utilizadas a causa de las dificultades y naturaleza del proyecto.

Capítulo 3 Análisis: se realiza el análisis del problema, mediante la realización de diagramas de clases, secuencia y estados, con el fin de disponer de un modelo conceptual que sirva de solución.

Capítulo 4 Tecnologías utilizadas: señala y define las tecnologías que han sido usadas para la realización de este proyecto, tanto a nivel de gestión como comunicación con los interesados. Por último, también detalla herramientas usadas para la realización de diagramas tanto en análisis como en diseño.

Capítulo 5 Diseño: se muestra el diseño del sistema, despliegue, arquitectura, etc.

Capítulo 6 Implementación y pruebas: contiene las pruebas que se han realizado así como sus resultados. También se listan los problemas que se han detectado y posibles soluciones.

Capítulo 7 Conclusiones: se realiza la conclusión del proyecto así como las vías de trabajo futuro que se recomiendan.

Bibliografía: contiene las referencias bibliográficas.

Anexo A Manuales: incluye el manual de instalación y despliegue.

Anexo B Resumen de enlaces adicionales: incluye enlaces de interés sobre el proyecto, como el repositorio de código <https://gitlab.inf.uva.es/josborg/tfg.git>

Capítulo 2

Requisitos y Planificación

2.1. Introducción

En este capítulo se expondrán los requisitos del proyecto obtenidos tras el proceso de elicitación de requisitos, en el cual los tutores asumieron el rol del cliente. Durante el proceso de desarrollo del proyecto, estos requisitos se modificaron ligeramente debido a la aparición de nuevas necesidades.

Una vez conocidos los requisitos, se procedió al desarrollo de la planificación del proyecto. Para realizar esta tarea, existen distintos enfoques en función de las características del software a desarrollar, del equipo de desarrollo o de la forma en la que evolucionen los requisitos del proyecto. Para proyectos en los que los requisitos son inmutables, la planificación programada en cascada se considera una buena opción. En cambio, como este proyecto se comenzó con la idea que los requisitos se adaptasen a las necesidades que surgieran durante el mismo y a la capacidad de desarrollo, se consideró que un enfoque incremental era la opción que mejor se adaptaba al proyecto planteado. Por ello, se decidió usar el marco de trabajo denominado “proceso unificado”. Una vez completada la planificación, se realizó un análisis de riesgos y el presupuesto del proyecto.

2.2. Requisitos

Durante la elicitación de requisitos, se recogieron las necesidades del software especificadas por los tutores, que hacían la función del cliente del proyecto, y se transformaron en sus correspondientes requisitos. Antes de continuar con la presentación de los requisitos, se presenta una serie de definiciones y términos necesarios para la correcta comprensión de los requisitos en la Tabla 2.1.

2.2. REQUISITOS

Término	Descripción
Número de Top1	Cantidad de encuestas en las que un usuario ha contestado correctamente a todas las preguntas y ha sido el mas rápido de entre los que también lo han hecho.
Tiempo de respuesta medio	Tiempo medio que ha tardado un usuario en responder a las encuestas desde de que empieza a contestar.
Tiempo de respuesta medio desde apertura	Tiempo medio que tarda un usuario en responder a las preguntas desde de que se abre la encuesta.
Tasa de aciertos de preguntas	Porcentaje de preguntas contestadas correctamente.
Tasa de aciertos de encuestas	Porcentaje de encuestas contestadas correctamente, es decir, con todas sus respuestas correctas.
Contestación	Es la solución que envía un usuario para una encuesta, formada por todas sus respuestas.
Respuesta	Opción de cada pregunta de una encuesta que cada usuario ha elegido.

Tabla 2.1: Terminología utilizada en los requisitos.

2.2.1. Requisitos funcionales

En la Tabla 2.2 se presentan los requisitos funcionales, es decir, aquellos que definen las funciones que debe permitir el sistema.

2.2.2. Requisitos de información

En las Tablas 2.3 y 2.4 se presentan los requisitos de información, los cuales definen la información que debe almacenar el sistema.

2.2.3. Requisitos no funcionales

En la Tabla 2.5 se presentan los requisitos no funcionales, que definen las restricciones del sistema.

2.2.4. Requisitos añadidos posteriormente

En la Tabla 2.6 se muestran los requisitos que se añadieron durante el desarrollo de la aplicación, con posterioridad al comienzo del mismo.

ID	Requisito
RF01	El sistema deberá permitir la realización de encuestas solo a los usuarios de los servidores de Discord que se desee.
RF02	El sistema deberá permitir la realización de cada encuesta solo una vez por usuario.
RF03	El sistema deberá permitir a un administrador añadir el bot a un servidor de Discord.
RF04	El sistema no deberá permitir conocer las respuestas correctas de una encuesta a los estudiantes hasta que esta finalice.
RF05	El sistema deberá permitir establecer un deadline (o eliminarlo explícitamente) para las encuestas durante su edición y creación.
RF06	El sistema deberá permitir establecer y editar una hora de inicio de la encuesta.
RF07	El sistema deberá permitir establecer y editar una hora de finalización de la encuesta.
RF08	El sistema deberá permitir distinguir entre usuarios “Profesor” y “Estudiante”.
RF09	El sistema deberá permitir publicar encuestas en servidores de Discord a los profesores.
RF10	El sistema deberá permitir editar cualquier encuesta publicada en un servidor a cualquiera de sus profesores.
RF11	El sistema deberá permitir crear y editar las preguntas de una encuesta y, si se hace, se deberán borrar las respuesta a esa encuesta para que los usuarios puedan volver a contestar.
RF12	El sistema deberá crear un ranking para cada encuesta, ordenado según el número de aciertos y, en caso de empate, en base a la velocidad a la hora de responder.
RF13	El sistema deberá permitir establecer y editar el estado de las encuestas de la siguiente forma: si está cerrada, no se puede contestar de ninguna forma; si está abierta, se puede contestar si el momento actual está entre los momentos de apertura y cierre; si está finalizada, no se puede contestar bajo ningún concepto, ni se puede cambiar de estado, ni modificar las preguntas de la encuesta.
RF14	El sistema deberá permitir a los usuarios elegir si quieren contestar cada encuesta de forma anónima o no.
RF15	El sistema deberá permitir a los usuarios acceder a sus resultados tanto globales como particulares de cada encuesta.
RF16	El sistema deberá permitir a los profesores acceder a los resultados de los usuarios que hayan contestado a las encuestas que hayan creado o que sean “Profesor” en el servidor donde se haya publicado.
RF17	El sistema deberá permitir que el administrador gestione las credenciales del bot para poder instalar el bot en otros servidores.
RF18	El sistema deberá mostrar diagramas de tarta al responder una encuesta, por cada pregunta, mostrando el porcentaje de respuestas de cada opción.
RF19	El sistema deberá permitir al usuario saber cuáles han sido las respuesta de cada pregunta y si ha sido correcta o no su respuesta, una vez la encuesta haya finalizado.

Tabla 2.2: Requisitos funcionales.

ID	Requisito
RI01	<p>El sistema deberá almacenar, para cada encuesta, la siguiente información:</p> <ul style="list-style-type: none"> ▪ Número identificador de la encuesta. ▪ Código alfabético de identificación de la encuesta. ▪ Número identificativo del creador de la encuesta. ▪ Nombre del creador de la encuesta. ▪ Momento de creación de la encuesta. ▪ Momento de inicio-fin de la encuesta. ▪ Tiempo máximo para contestar a la encuesta. ▪ Estado de la encuesta. ▪ Tipo de la encuesta. ▪ Preguntas de la encuesta.
RI02	<p>El sistema deberá almacenar, para cada pregunta de una determinada encuesta, la siguiente información:</p> <ul style="list-style-type: none"> ▪ Número identificador de la pregunta. ▪ Número identificador de la encuesta a la que pertenece la pregunta. ▪ Texto de la pregunta. ▪ Opciones (respuestas posibles) de la pregunta.
RI03	<p>El sistema deberá almacenar, para cada opción (respuesta posible) de una determinada pregunta, la siguiente información:</p> <ul style="list-style-type: none"> ▪ Número identificador de la opción. ▪ Número identificador de la pregunta a la que pertenece la opción. ▪ Texto de la opción. ▪ Si la opción es correcta o no.

Tabla 2.3: Requisitos de información (primera parte).

ID	Requisito
RI04	<p>El sistema deberá almacenar, para cada contestación, la siguiente información:</p> <ul style="list-style-type: none"> ■ Número identificador de la contestación. ■ Número identificador de la encuesta a la que pertenece la contestación. ■ Momento de inicio y fin de la contestación. ■ Nickname de Discord asociado a quien proporciona la contestación, en formato nombre#numero, por ejemplo, “nombre#0000”. ■ Número identificativo interno de Discord, denominado Snowflake [9] del usuario que proporciona la contestación. ■ Si es anónima o no. ■ Si es válida o no. ■ Respuestas de la contestación.
RI05	<p>El sistema deberá almacenar, para cada repuesta, la siguiente información:</p> <ul style="list-style-type: none"> ■ Número identificador de la respuesta. ■ Número identificador de la contestación a la que pertenece. ■ Número identificador de la pregunta a la que se refiere. ■ Número identificador de la opción a la que se refiere. ■ Si la respuesta ha sido la opción correcta o no.
RI06	<p>El sistema deberá almacenar la información de servidores Discord en los que está instalado el bot.</p>
RI07	<p>El sistema deberá almacenar la información de usuarios que están en un servidor en el que está instalado el bot.</p>
RI08	<p>El sistema deberá almacenar la información de servidores los Discord en los que una encuesta ha sido publicada.</p>

Tabla 2.4: Requisitos de información (continuación).

2.2. REQUISITOS

ID	Requisito
RNF01	El sistema deberá de ser usable, es decir, deberá tener una puntuación mayormente positiva en las pruebas de usuario al alumnado.
RNF02	El sistema deberá ser completamente funcional, pudiéndose desplegar en las máquinas virtuales de la Escuela de Ingeniería Informática de la Universidad de Valladolid (Inf-UVa).
RNF03	El sistema deberá poder desplegarse con un solo comando sin tener que cambiar la configuración de las máquinas virtuales de Inf-UVa.
RNF04	Todas las funcionalidades de creación, edición y publicación de encuestas deben estar reservadas a los profesores.

Tabla 2.5: Requisitos no funcionales.

ID	Requisito
RI09	Los resultados deberán constar de las métricas: número de aciertos, número de preguntas, número de encuestas, número de encuestas correctas, número de Top1, mejor posición de ranking, posición media de ranking, tiempo de respuesta medio, tiempo de respuesta medio desde apertura, tasa de aciertos de preguntas, tasa de aciertos de encuestas.
RF20	El sistema deberá permitir realizar filtros de asignatura, encuesta y usuario en los resultados.
RF21	El sistema deberá permitir establecer dos tipos de criterios en lo relativo al tiempo usado en casos de empate a número de aciertos a la hora de realizar el ranking para cada encuesta: tiempo de respuesta desde que se empieza a responder la encuesta y tiempo de respuesta desde que el profesor crea la encuesta.
RNF05	El sistema deberá funcionar en una máquina virtual teniendo solamente público el puerto 80.

Tabla 2.6: Requisitos añadidos con posterioridad al inicio del desarrollo del proyecto.

2.3. Planificación y metodología

El proyecto ha sido planificado mediante el Proceso Unificado de Desarrollo Software, también denominado Proceso Unificado (UP). Es una metodología de desarrollo de software que se caracteriza por estar centrada en la arquitectura, dirigida por casos de uso, y por ser iterativa e incremental [4].

El UP consta de cinco flujos de trabajo fundamentales:

1. Requisitos: consiste en la elicitación de requisitos del cliente.
2. Análisis: en él se crea una solución conceptual que cumpla con los requisitos.
3. Diseño: se construye la solución concreta en base al análisis realizado.
4. Implementación: se crean los artefactos de software que forman el sistema final.
5. Pruebas: se verifica si cada artefacto resultante de cada flujo de trabajo cumple con los requisitos. Normalmente las pruebas se centran en comprobar los artefactos de la implementación.

Además de cuatro fases:

1. Inicio: donde se definen el alcance, los objetivos y los riesgos del proyecto.
2. Elaboración: se obtiene la arquitectura y una implementación de la estructura donde se desarrollaran las funcionalidades del sistema.
3. Construcción: fase de desarrollo de cada funcionalidad.
4. Transición: despliegue en producción del sistema.

El alcance del proyecto será desarrollar un sistema de software plenamente funcional, que haya pasado por pruebas de usuario de los usuarios finales, para verificar el correcto funcionamiento del sistema, identificar los problemas y proponer las soluciones con las que se continuará el desarrollo. En la Figura 2.1 se muestra la planificación concreta del proyecto. La elaboración de la memoria de este proyecto se ha realizado en paralelo al desarrollo de

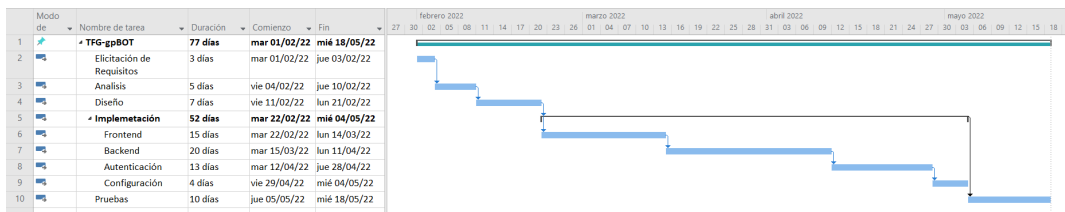


Figura 2.1: Diagrama de Gantt inicial.

la aplicación, por lo tanto se ha integrado el coste temporal de la escritura de esta en cada tarea correspondiente en el diagrama de la Figura 2.1. Esta decisión ha sido tomada con el motivo de tratar de evitar mezclar la planificación propia del desarrollo de software, que es la que se pretende mostrar, con la del desarrollo de la memoria de este TFG.

2.3.1. Software y hardware

En Tabla 2.7 se muestran las tecnologías software usadas, mientras que la Tabla 2.8 muestra las tecnologías hardware.

Nombre	Descripción	Uso
MS Project	Software de gestión de proyectos	Planificación del proyecto y generación de los diagramas de Gantt
Overleaf	Editor online de ficheros en LaTeX	Construcción de la memoria.
Astah Professional	Editor UML	Generación de los diagramas UML del análisis y el diseño
Advanced REST client	Herramienta para probar APIs REST	Realizar test del backend
Visual Studio Code	Editor código fuente	Desarrollo del código del software
Tomcat	Servidor de aplicaciones	Ejecutar el backend
Angular	Plataforma de desarrollo web	Desarrollo del frontend
Sprintboot	Framework desarrollo de aplicaciones auto-contenidas	Desarrollo del cliente backend
Docker	Plataforma de empaquetado de software	Empaquetado de los microservicios y despliegue
MySQL	Sistema de gestión de bases de datos	Almacenar los datos necesarios del sistema en persistencia
Boostrap	Framework de desarrollo de interfaces web	Desarrollo de la interfaz gráfica del frontend

Tabla 2.7: Detalle del software utilizado en el proyecto.

Nombre	Descripción	Uso
Máquina virtual	Sistema de computación simulado	Alojar y desplegar el sistema
PC	Sistema de computación	Desarrollo del proyecto

Tabla 2.8: Detalle del hardware utilizado en el proyecto.

2.3.2. Presupuesto Inicial

En la Tabla 2.9 se muestra el presupuesto inicial del proyecto. El presupuesto se ha realizado teniendo en cuenta el coste de las horas de trabajo del desarrollador del proyecto, más las licencias que han sido necesarias del software utilizado. Se ha considera que el coste de la hora de trabajo sería de 15€y que el total de horas trabajadas serán 336, según la planificación inicial.

Concepto	Cantidad	Precio unitario	Total
Mano de obra en horas	336	15,00€	5040,00€
Alojamiento	0	-	0€
Dominio	0	-	0€
Licencia Astah Professional	1	585,00€	585,00€
Licencia MS Project	1	303,60€	303,60€
Licencia Office	1	149,00€	
			6077,60€

Tabla 2.9: Presupuesto inicial del proyecto.

2.3.3. Análisis riesgos

En la Tabla 2.10 se muestra el criterio de evaluación los riesgos, según [18]. Los riesgos considerados son: l desarrollador suspende alguna asignatura (Tabla 2.11); el desarrollador enferma (Tabla 2.12); existen problemas de comunicación (Tabla 2.13); el sistema no puede ser desplegado en la máquina virtual (Tabla 2.14); Discord no se puede integrar (Tabla 2.15); y los posibles cambios de requisitos (Tabla 2.16). Cada tabla muestra, además, el correspondiente plan de mitigación y contingencia de cada riesgo.

	Despreciable	Marginal	Moderado	Crítico	Catastrófico
Cierto	Medio	Alto	Alto	Alto	Alto
Probable	Medio	Medio	Alto	Alto	Alto
Posible	Bajo	Medio	Medio	Alto	Alto
Improbable	Bajo	Bajo	Medio	Medio	Alto
Excepcional	Bajo	Bajo	Bajo	Medio	Alto

Tabla 2.10: Matriz de riesgos probabilidad e impacto.

2.3. PLANIFICACIÓN Y METODOLOGÍA

Identificador	R01 - El desarrollador suspende alguna asignatura
Descripción	Debido a que el desarrollador todavía no ha aprobado todas las asignaturas, si suspende alguna, no se podrá entregar el proyecto en la fecha esperada
Impacto	Crítico
Probabilidad	Improbable
Plan de mitigación	Dedicar más tiempo a las asignaturas que faltan por aprobar
Plan de contingencia	Replanificar el proyecto para entregarlo en una fecha posterior

Tabla 2.11: Riesgo R01: el desarrollador suspende alguna asignatura.

Identificador	R02 - El desarrollador enferma
Descripción	Debido a la situación de pandemia, si el desarrollador enferma, no se podrá retomar el desarrollo del proyecto hasta que este se recupere
Impacto	Moderado
Probabilidad	Posible
Plan de mitigación	Planificar el proyecto de forma pesimista para tener un margen de tiempo
Plan de contingencia	Retrasar la fecha de entrega y reducir funcionalidades

Tabla 2.12: Riesgo R02: el desarrollador enferma.

Identificador	R03 - Problemas de comunicación con los tutores
Descripción	Debido a que la realización del proyecto necesita de la comunicación alumno-tutores, si esta no es efectiva, los avances del proyecto no serán monitorizados, pudiéndose realizar tareas innecesarias
Impacto	Crítico
Probabilidad	Improbable
Plan de mitigación	Realizar reuniones semanales o bisemanales para validar los incrementos del proyecto, además del uso de un servidor Discord para realizar la comunicación alumno-tutores
Plan de contingencia	Aumentar el número de reuniones y hacer reuniones virtuales

Tabla 2.13: Riesgo R03: problemas de comunicación con los tutores.

Identificador	R04 - Las características de la máquina virtual impiden el despliegue
Descripción	Debido a que la configuración de puertos de las máquinas virtuales de Inf-UVa no se puede editar, si se necesitan más puertos públicos, el sistema no se podrá desplegar
Impacto	Crítico
Probabilidad	Cierto
Plan de mitigación	Realizar un diseño de software que permita el despliegue usando un solo puerto
Plan de contingencia	Usar otro tipo de máquina virtual o realizar el proyecto en local

Tabla 2.14: Riesgo R04: las características de la máquina virtual impiden el despliegue.

Identificador	R05 - El servicio externo Discord no se puede integrar
Descripción	Debido a que el desarrollador no tiene experiencia desarrollando herramientas que usen Discord, si este servicio no se puede integrar, la autenticación y gestión de usuarios del sistema no se podrá realizar
Impacto	Crítico
Probabilidad	Improbable
Plan de mitigación	Investigar sobre OAuth 2.0 de Discord antes de desarrollar
Plan de contingencia	Implementar un micro-servicio de gestión de usuarios independiente a Discord que cumpla con los requisitos del proyecto

Tabla 2.15: Riesgo R05: el servicio externo Discord no se puede integrar.

Identificador	R06 - Modificación de los requisitos
Descripción	Debido a la naturaleza del proyecto, si los requisitos se modifican, el proyecto no se entregará a tiempo
Impacto	Marginal
Probabilidad	Probable
Plan de mitigación	Realizar el desarrollo software usando patrones de diseño software y recomendaciones para la fácil modificación de funcionalidades
Plan de contingencia	Eliminar alguna de las funcionalidades planificadas

Tabla 2.16: Riesgo R06: modificación de los requisitos.

2.3.4. Seguimiento y desviaciones del plan inicial

En esta sección se realiza un resumen del seguimiento del proyecto y las desviaciones respecto al plan inicial. Estas desviaciones son normalmente manifestaciones de los riesgos detectados en la sección anterior o de otro tipo de riesgos. La valoración de los avances del proyecto se realizó mediante reuniones semanales o bisemanales, en función de los avances

2.3. PLANIFICACIÓN Y METODOLOGÍA

conseguidos. La fase de elicitación de requisitos transcurrió según lo previsto. En cuanto a las de Análisis y Diseño, su realización supuso un día menos cada una, por lo que la fase de implementación pudo comenzar dos días antes de lo previsto. Sin embargo, la implementación tuvo retrasos en todas sus fases, y los 52 días previstos inicialmente acabaron siendo finalmente 67 días. La fase de pruebas transcurrió según lo previsto. A continuación se especifican los motivos de los retrasos durante la fase de implementación:

- La interfaz no era responsive por lo que fue necesario editarla. Además, se modificaron algunos requisitos durante el proyecto que supusieron cambios en la interfaz.
- La modificación de algunos de los requisitos y algunas de las funcionalidades, sumado a la inexperiencia del desarrollador, hizo que se produjeran retrasos en el desarrollo del backend.
- La autenticación con Discord no se pudo realizar de la forma recomendada por la documentación API Discod [10] mediante tokens. En su lugar, la recuperación del “access_token” se realiza a través de un fragmento de la URL de respuesta de API Discord. La investigación de esta alternativa supuso un tiempo adicional no planificado.
- Se tuvo que refactorizar el sistema para que este use un solo puerto público, pues aunque el diseño ya consideraba esta solución, su implementación supuso dificultades para que el trafico se redirigiera y filtrase de la forma adecuada.

2.3.5. Diagrama de Gantt final

En la Figura 2.2 se muestra la planificación final del proyecto.

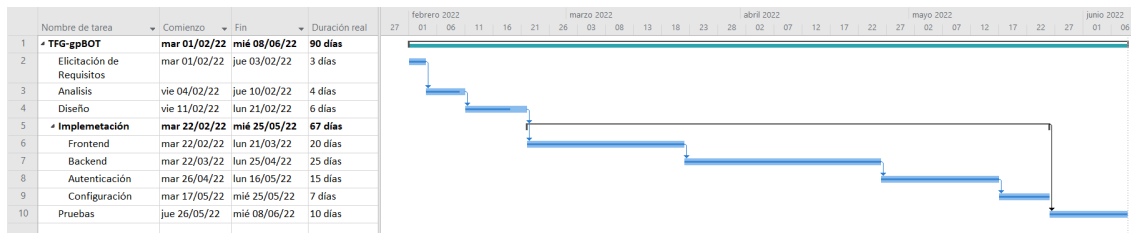


Figura 2.2: Diagrama de Gantt final.

2.3.6. Presupuesto Final

En la Tabla 2.17 se muestra el presupuesto final del proyecto considerando las desviaciones que se han producido durante la implementación. De nuevo, el presupuesto se ha realizado teniendo en cuenta el coste de las horas de trabajo del desarrollador del proyecto más las licencias que han sido necesarias del software utilizado. Se ha considerado que el coste de la hora de trabajo sería de 15€la hora y, según la planificación final, el número de horas finales que se han trabajado han sido 360 horas.

Concepto	Cantidad	Precio unitario	Total
Mano de obra en horas	360	15,00€	5400,00€
Alojamiento	0	-	0€
Dominio	0	-	0€
Licencia Astah Professional	1	585,00€	585,00€
Licencia MS Project	1	303,60€	303,60€
Licencia Office	1	149,00€	
			6437,60€

Tabla 2.17: Tabla de presupuesto final.

Capítulo 3

Análisis

3.1. Introducción

En este capítulo se muestra la fase de análisis del proyecto. El análisis es la etapa de un proyecto, en este caso de software, en el que se transforman los requisitos en una solución conceptual. En esta fase no se pretende obtener una solución concreta basada en una tecnología específica, sino que se busca conseguir una solución general que pueda ser implantada en la fase de diseño por la tecnología que se considere más adecuada o como consecuencia de algún requisito de diseño solicitado. Durante el análisis, se realizan una serie de diagramas que sirven, tanto para entender el problema a nivel conceptual, como para tratar de elaborar una solución conceptual que cumpla con los requisitos del proyecto.

A continuación, se mostrará el diagrama de clases del modelo conceptual de la solución, el diagrama de casos de uso, los diagramas de secuencia de los casos de uso del sistema y por último, la máquina de estados de una encuesta. Con ello se pretende explicar detenidamente cuál es la forma en la que se va a abordar la realización del sistema, a partir de los requisitos solicitados.

3.2. Características

El sistema constará de encuestas que deberán poder ser gestionadas tanto en su creación como en su borrado, edición, publicación y contestación. También debe existir una forma de registrar a los usuarios que crean y contestan las encuestas. Por último, una contestación de una encuesta deberá contener la respuesta a cada pregunta, además del momento en el que se ha realizado la contestación y el tiempo que se ha tardado en contestar cada pregunta. Además, el usuario debe poder contestar de forma anónima o no.

El sistema deberá permitir solo a los profesores la gestión de encuestas. El creador de

una encuesta podrá publicarla y los profesores del servidor en que se ha publicado tendrán los mismos permisos que el creador.

Una encuesta solo podrá ser contestada por el creador o por los usuarios que están en un servidor en el que la encuesta ha sido publicada; solo podrá ser contestada una vez y solo si está en estado abierta y el momento de la contestación esta entre los momentos de inicio y fin de la encuesta. Si el estado de la encuesta es finalizada, la encuesta no podrá ser contestada y no se podrá cambiar de estado.

3.3. Modelo conceptual

El modelo conceptual es un diagrama de clases en el que se pretende representar la estructura general de la solución. Los atributos de cada clase representa la información que debe de tener cada entidad y las relaciones de cada clase significan como está información de cada entidad esta estructurada. La Figura 3.1 muestra el diagrama de clases de la aplicación. Dentro de este diagrama, cabe destacar:

- Los usuarios pueden contestar encuestas, esta contestación siempre es a una encuesta y una encuesta está creada y es gestionada por un usuario profesor.
- Una contestación de un usuario es representada por la clase solución. Esta solución está compuesta de respuestas, que a su vez están compuestas de una opción.
- Una encuesta está compuesta de preguntas y las preguntas, por opciones.
- Una encuesta puede ser publicada en varios servidores o no ser publicada.
- Todo servidor debe incluir al bot, aunque el bot puede estar en varios servidores.
- Por último, un usuario tiene que estar al menos en un servidor.

3.4. Tipos de usuario

Fundamentalmente existen dos tipos de usuario en el sistema: los usuarios y los profesores. Un usuario puede visualizar sus resultados y contestar a las encuestas que tenga disponibles. Un profesor es un usuario que además puede gestionar encuestas y visualizar todos los resultados de las encuestas que ha creado o de las que es profesor. Por último, existe un usuario administrador cuya única función es la de añadir el bot a los servidores deseados.

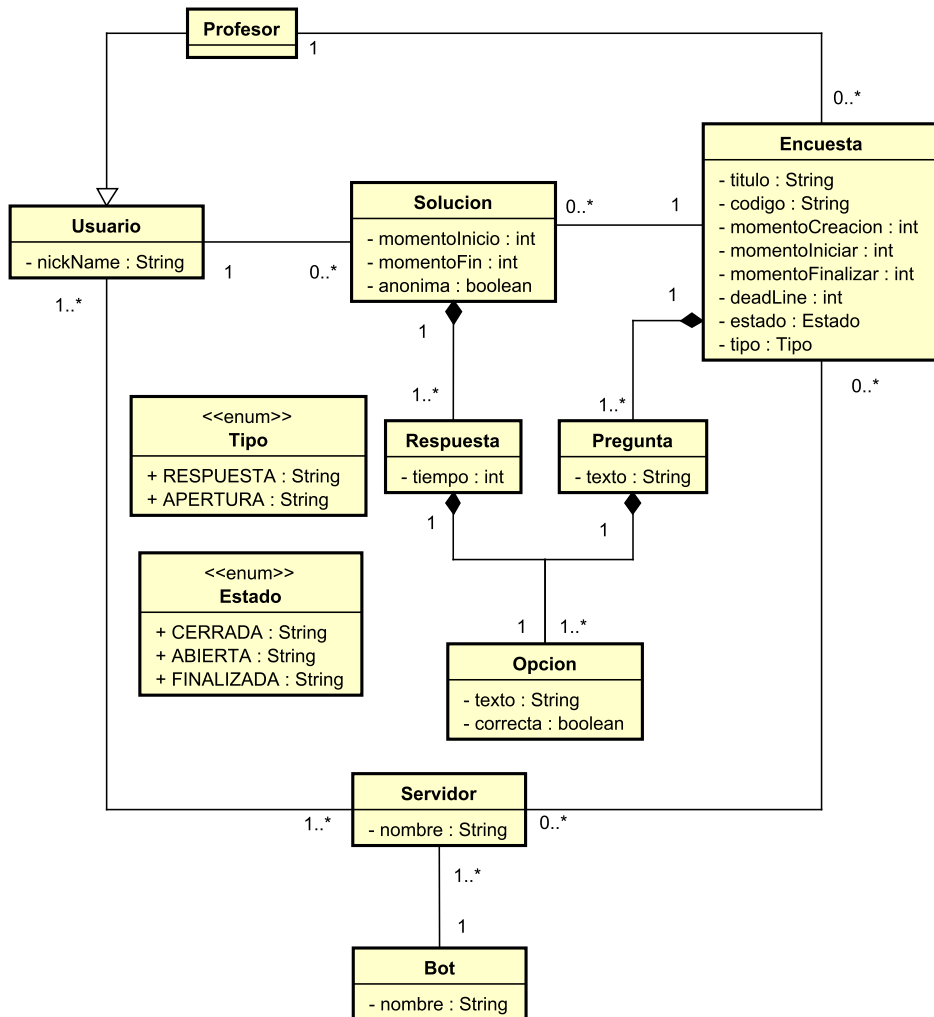


Figura 3.1: Diagrama de clases: modelo conceptual.

3.5. Casos de uso

A continuación, la Figura 3.2 presenta el diagrama de casos de uso. En este diagrama se muestran las tareas o “casos de uso” que deberán poder realizar los distintos actores. Los usuarios deberán poder identificarse en el sistema. También deben poder obtener sus encuestas disponibles, contestar las encuestas y obtener los resultados de sus encuestas. Los profesores son usuarios y podrán realizar todos los casos de uso que estos realizan, y además podrán crear encuestas, publicarlas, borrarlas y editarlas. Por último, los administradores tendrán una forma especial de identificarse y podrán añadir el bot a un servidor.

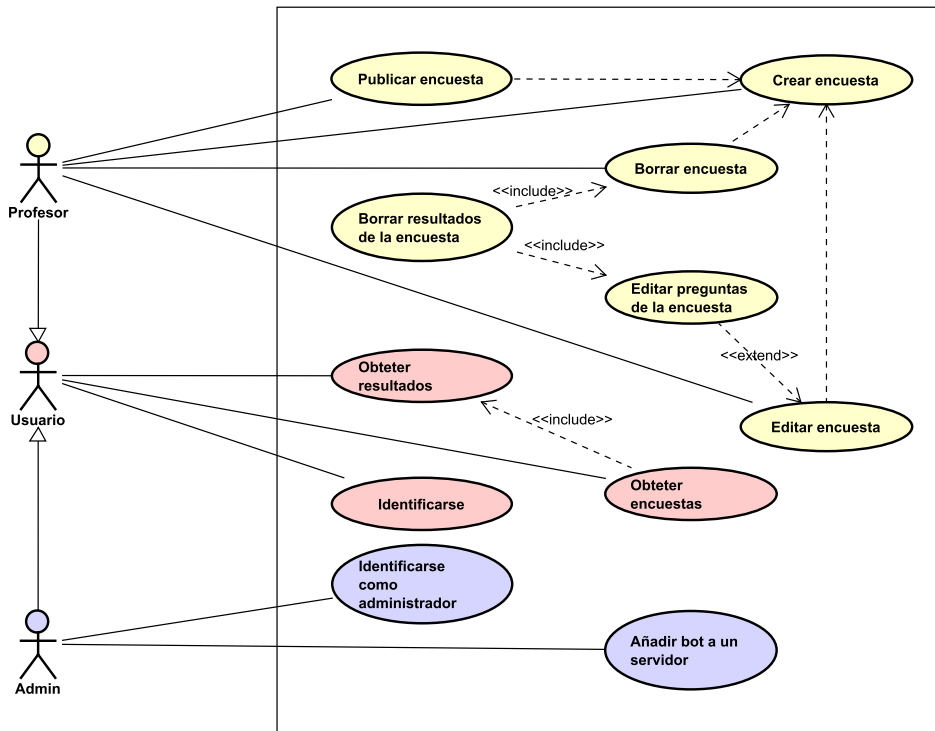


Figura 3.2: Diagrama de casos de uso

3.6. Análisis de los casos de uso

En esta sección se realiza el análisis de los casos de uso. Consiste en la representación del modelo dinámico de la solución propuesta. Para ilustrar este modelo, se muestran los diagramas de secuencia para los casos de uso del sistema. La Figura 3.3 muestra el diagrama de secuencia del caso de uso *Identificarse*, mientras que la Figura 3.4 muestra el del caso de uso *Identificarse como Administrador*. Quien se haya identificado como administrador podrá realizar el caso de uso *Añadir Bot*, cuyo diagrama de secuencia se muestra en la Figura 3.5.

En la Figura 3.6 se muestra como se crean las encuestas, en la Figura 3.7 como se editan, la Figura 3.8 muestra la forma de de publicarlas, la Figura 3.9 muestra como se borran las encuestas y por último, la Figura 3.10 los resultados. Finalmente, las Figuras 3.11 y 3.12 muestran el proceso de borrado de resultados y encuestas, respectivamente.

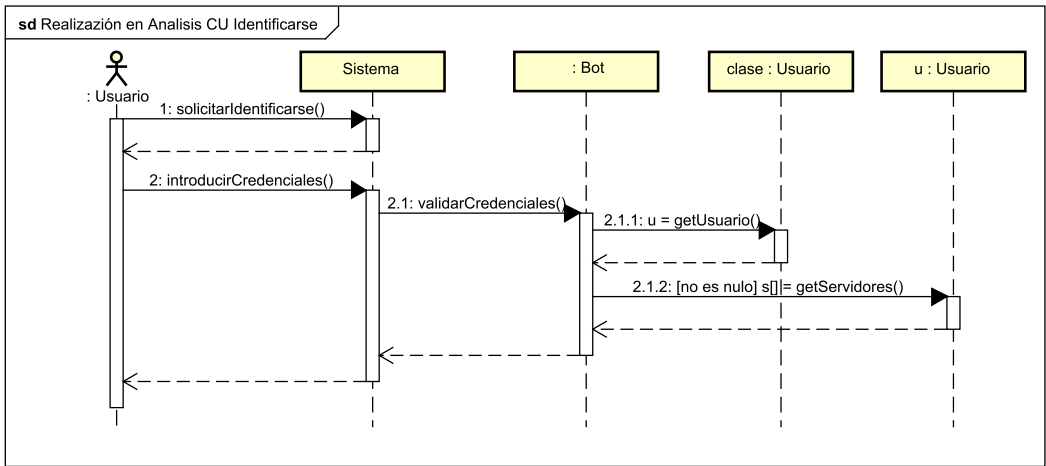


Figura 3.3: Diagrama de secuencia: caso de uso *Identificarse*.

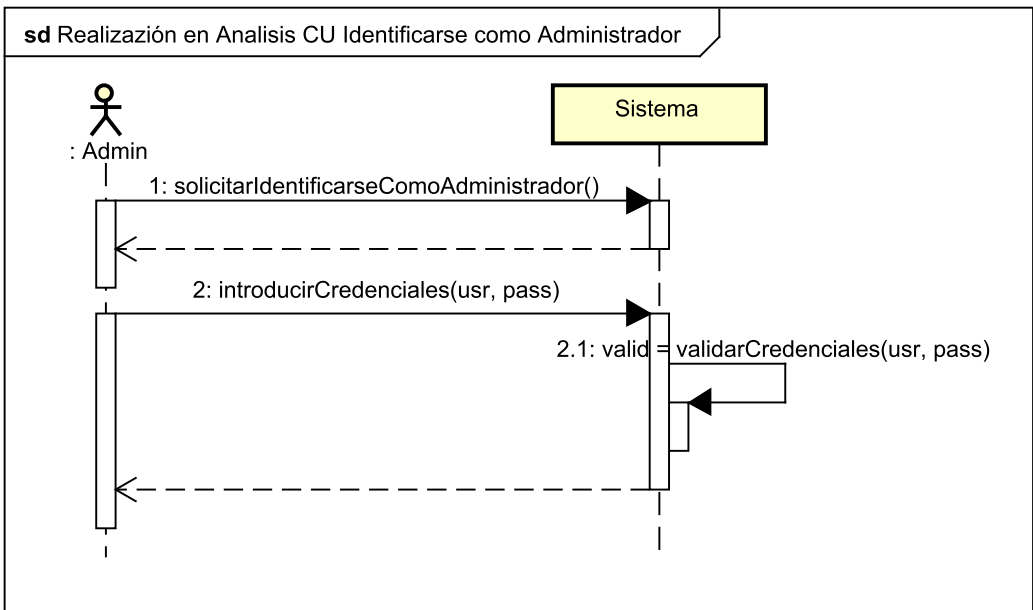


Figura 3.4: Diagrama de secuencia: caso de uso *Identificarse como Administrador*.

3.6. ANÁLISIS DE LOS CASOS DE USO

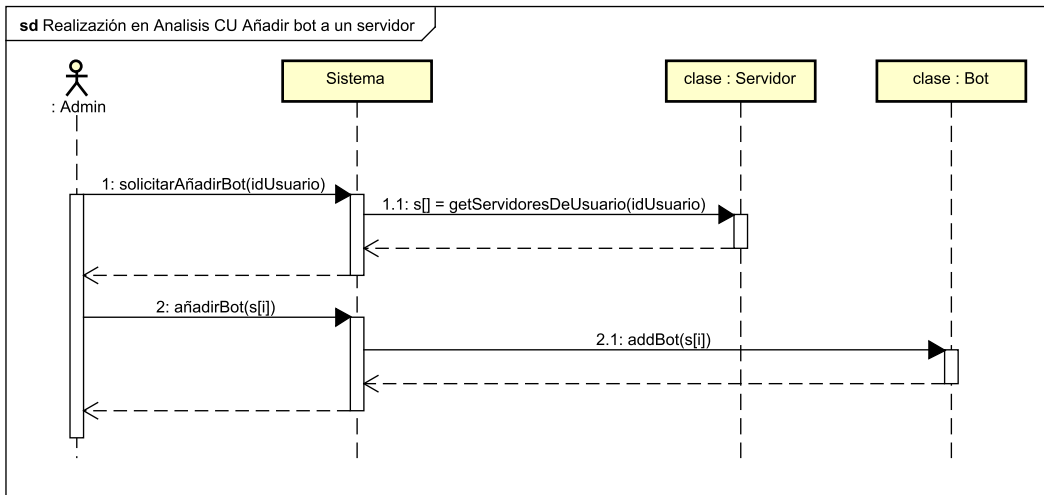


Figura 3.5: Diagrama de secuencia: caso de uso *Añadir Bot*.

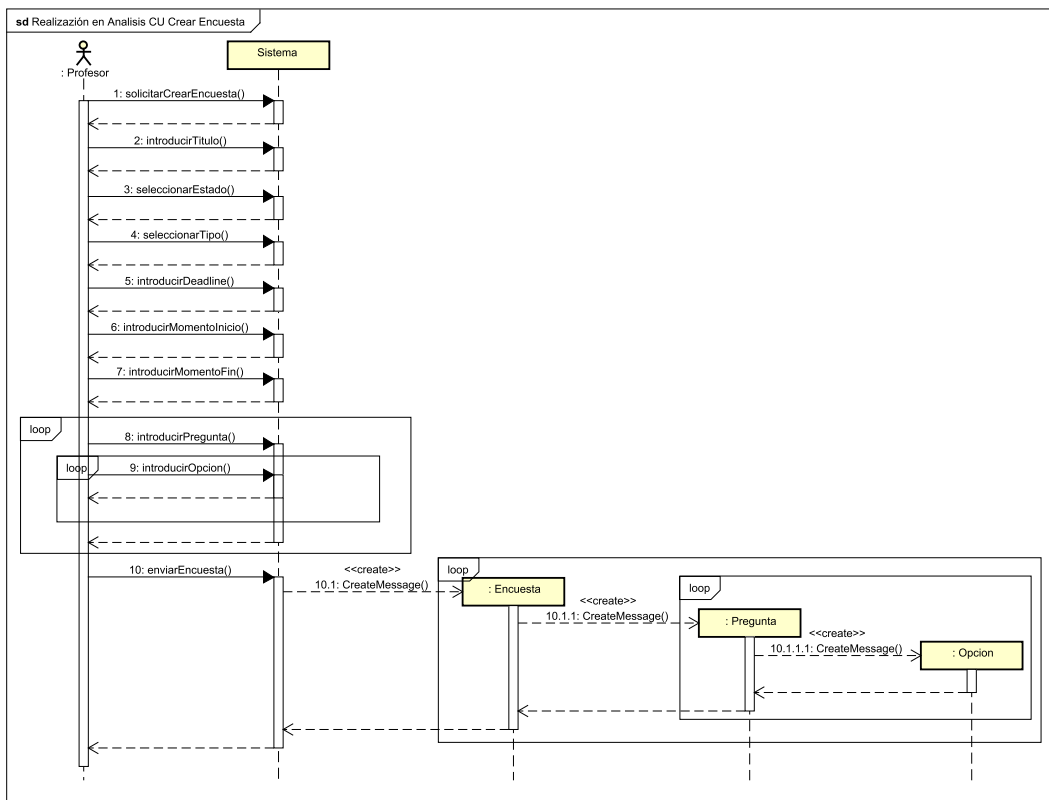


Figura 3.6: Diagrama de secuencia: caso de uso *Crear encuesta*.

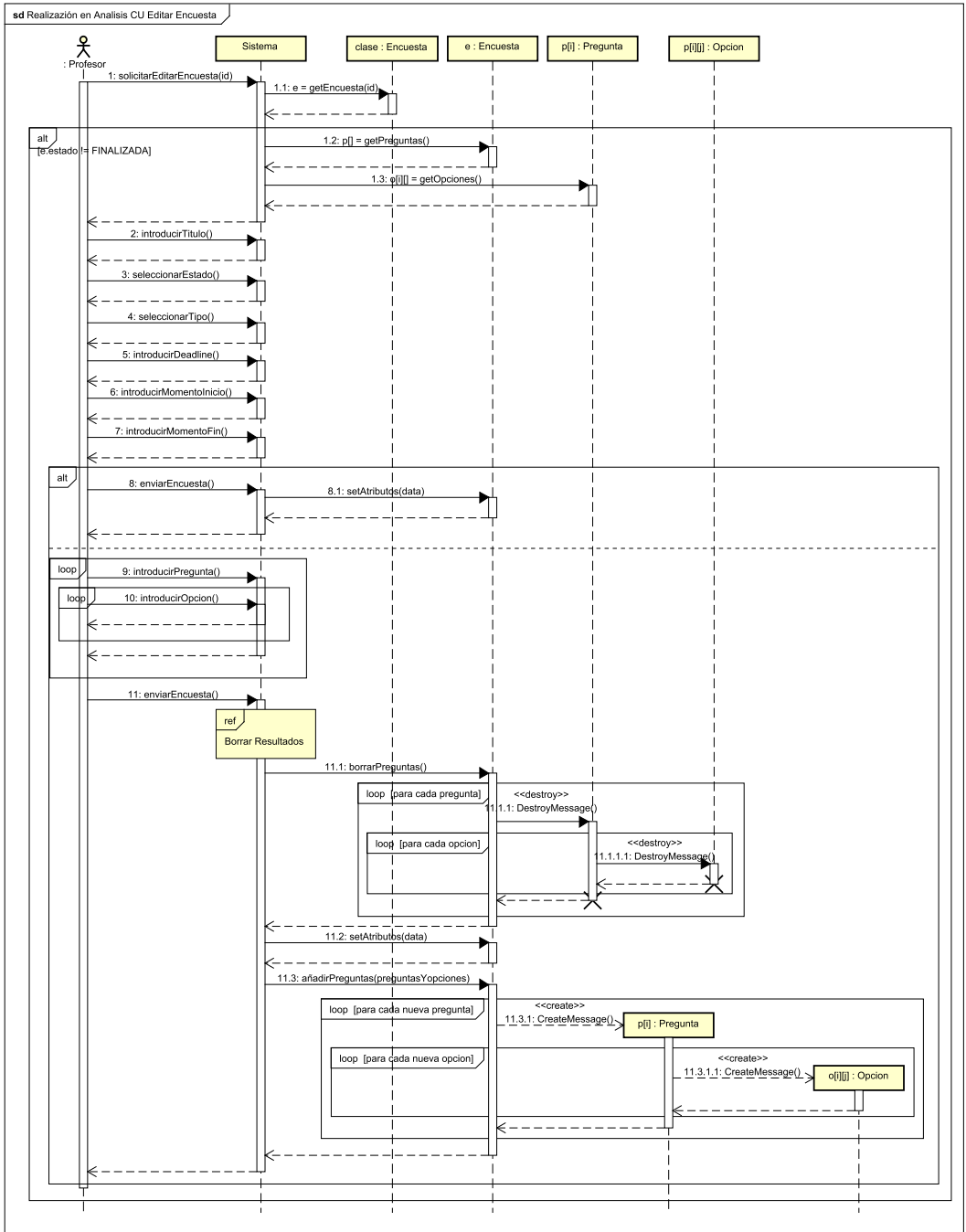


Figura 3.7: Diagrama de secuencia: caso de uso *Editar encuesta*.

3.6. ANÁLISIS DE LOS CASOS DE USO

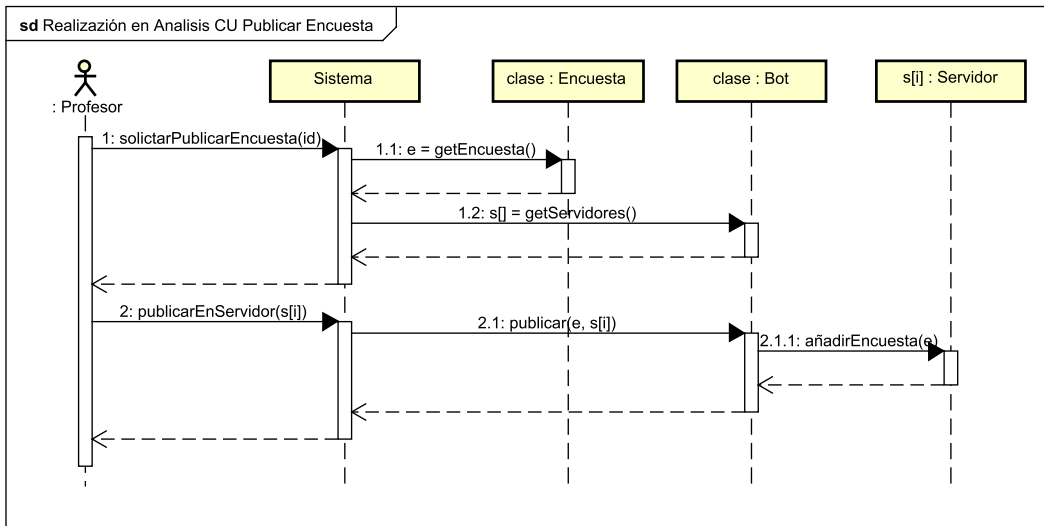


Figura 3.8: Diagrama de secuencia: caso de uso *Publicar encuesta*.

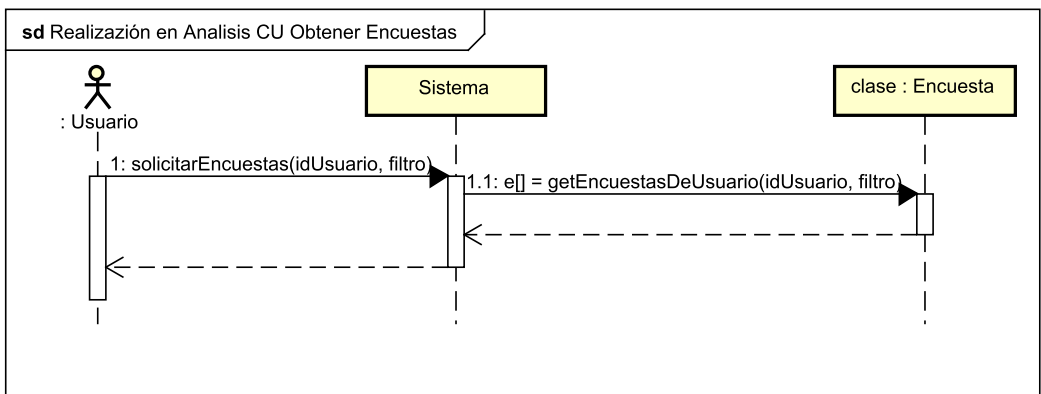


Figura 3.9: Diagrama de secuencia: caso de uso *Obtener encuesta*.

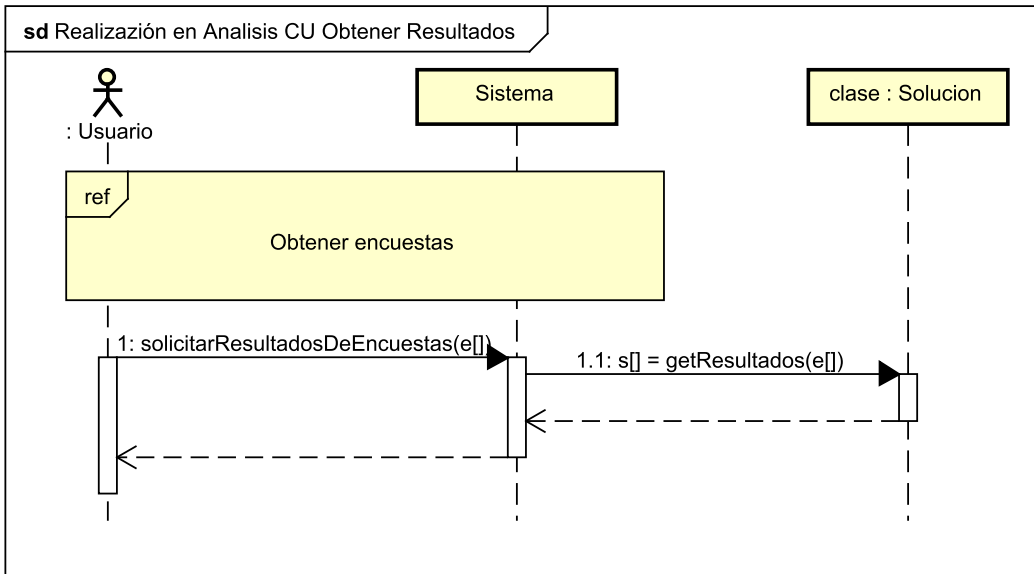


Figura 3.10: Diagrama de secuencia: caso de uso *Obtener Resultados*.

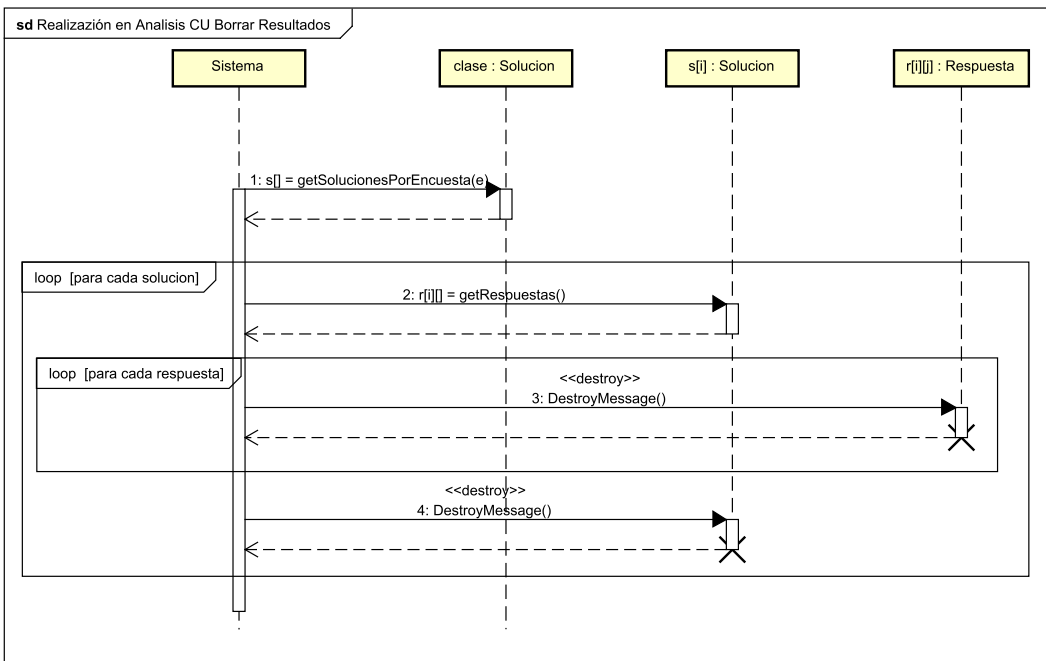


Figura 3.11: Diagrama de secuencia: caso de uso *Borrar resultados*.

3.7. MÁQUINA DE ESTADOS DE ENCUESTAS

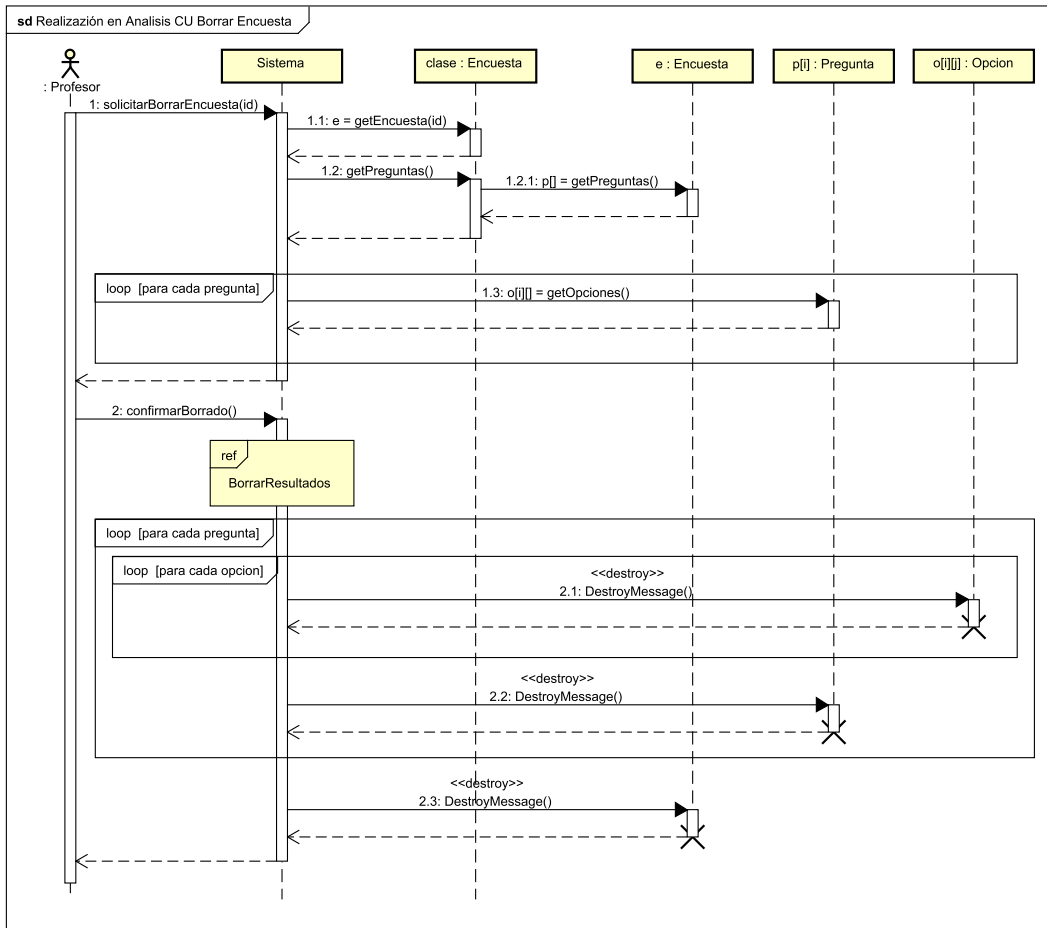


Figura 3.12: Diagrama de secuencia: caso de uso *Borrar encuesta*.

3.7. Máquina de estados de encuestas

En la Figura 3.13 se muestra el diagrama de máquina de estados de las encuestas. Los estados transicionan entre sí de forma manual y sin restricciones condicionales. Una encuesta cerrada puede pasar abierta y de abierta a cerrada. Desde cualquiera de estos dos estados puede pasar a finalizada, último estado de la encuesta del cual no se puede salir.

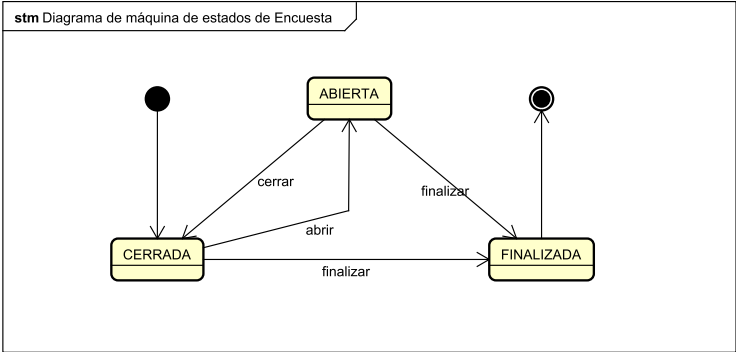


Figura 3.13: Máquina de estados de una encuesta.

Capítulo 4

Tecnologías utilizadas

4.1. Introducción

En este capítulo se presentan las tecnologías utilizadas durante el proyecto. Esto incluye toda herramienta que constituye el sistema, así como aquellas que han sido requisito para la gestión y monitorización del proyecto. También se definen las especificaciones de la maquina virtual empleada para el despliegue del sistema, incluyendo las tecnologías que alberga, desde los frameworks que se han utilizado, las especificaciones de la base de datos, contenedores que permiten el despliegue, etc. Todo ello justificándose el motivo de la selección de cada tecnología y el uso que se le da en el sistema.

En primer lugar, se definirán las tecnologías que han sido usadas para la gestión del proyecto, seguido de las tecnologías utilizadas en el backend, las utilizadas en del frontend, y por último, las tecnologías empleadas para la interacción con los canales de Discord.

4.2. Gestión del proyecto

En cuanto a la gestión del proyecto se ha tomado la decisión de optar por UP con una programación en cascada, además se busca realizar entregas de incrementos de forma semanal y permitir la leve modificación de los requisitos iniciales durante el desarrollo del proyecto en función a la evolución del mismo.

Los diagramas de Gantt para generar para cada flujo de trabajo del proyecto se han realizado usando MS Project. También se han usado hojas de cálculo Microsoft Excel para llevar un registro de las tareas realizadas y la prioridad de cada tarea, y Microsoft Word para la realización de informes.

La comunicación entre los tutores del proyecto se ha realizado mediante un servidor de

Discord y mediante reuniones semanales con los tutores. Discord también se ha usado para llevar un registro de lo acontecido en las reuniones y los requisitos que deben ser incorporados.

4.3. Backend

El backend es toda parte del sistema con la que los usuarios del mismo no van a interactuar directamente, sino a través del frontend. Consta de la base de datos, la lógica del negociación y una API de acceso que usa el cliente frontend para comunicarse con el mismo.

El software que conforma el backend está desarrollado íntegramente sobre SpringBoot. SpringBoot es un framework de Java que se ha convertido en el framework mas usado por encima de los estándares J2EE. Es por esto y por el conocimiento previo adquirido gracias a la asignatura *Desarrollo Basado en Componentes y Servicios*, por lo que se ha elegido esta tecnología, reduciendo la curva de aprendizaje, cosa que no habría ocurrido con otras alternativas como Django en Python.

SpringBoot incorpora un servidor web Tomcat [2], usando Java Persistence API (JPA) Hibernate [25], para realizar la comunicación con una base de datos MySQL. Tomcat es un servidor de aplicaciones desarrollado por Apache Software Foundation y su código es de libre acceso, mientras que Java Persistence API (JPA) es una API de persistencia de Java de datos relacionales, siendo Hibernate es una implementación específica de esta API. El objetivo de JPA es permitir el manejo de bases de datos manteniendo el paradigma de programación orientada a objetos. En cuanto a MySQL [24], es un sistema de gestión de base de datos relacional y es considerada una de las bases de datos de código abierto más popular.

Por último, el desarrollo de las funciones basadas un bot de Discord también han sido desarrolladas sobre Springboot con un servidor Tomcat, mediante programación reactiva y el uso del paquete Discord4j [11], un contenedor REST/WS basada en Java para la API oficial de Discord Bot.

Todo el sistema se ha alojado en una máquina virtual Linux 5.4.0-107-generic #121-Ubuntu SMP x86_64 x86_64 x86_64 GNU/Linux.

4.4. Frontend

El frontend es toda parte del sistema con la que los usuarios del mismo van a interactuar directamente para realizar los casos de uso del sistema. Para el desarrollo del frontend, se ha utilizado un cliente Angular desarrollado bajo Node.js y usando el framework Bootstrap para facilitar las tareas de realizar un diseño *responsive* y facilitar el diseño gráfico del mismo.

Angular [16] es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mientras que Node.js [23] es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript.

Node.js es asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Por último, se ha utilizado Bootstrap [29], una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web.

4.5. Enlace del sistema con Discord

La instalación del sistema en un host requiere que el host tenga Docker instalado y el puerto http 80 público y sin ser usado por ninguna otra aplicación, además de los puertos 8080, 8081, 8082 y 3307.

Una vez la web este funcionando, el administrador del sistema deberá de registrarse usando las credenciales de administración y añadir el bot al los servidores de Discord en los que de se desee permitir la publicación de encuestas. Una vez añadido el bot, este creará el rol Profesor de forma automática. El administrador de cada servidor deberá de añadir el rol Profesor a los profesores del servidor para permitirles la gestión de encuestas. Los profesores podrán crear las encuestas deseadas y publicarlas en el servidor para que los usuarios del mismo puedan contestarlas.

Capítulo 5

Diseño

5.1. Introducción

En este capítulo se describe cuál es el diseño que se ha elegido. El diseño es la fase del desarrollo de software en la que se decide cómo se va a transformar la solución conceptual obtenida en la fase de análisis en una solución que pueda ser implantada. En esta fase se decide la arquitectura que debe seguir el software, las tecnologías que se adoptan, los patrones de diseño a seguir, la estructura en la que se organizan los recursos del sistema y cómo debe de ser el despliegue del mismo.

En las sucesivas secciones, se mostrará el diagrama de la arquitectura del sistema, así como distintos diagramas que describen el estilo de módulos y el de usos de cada paquete que forma la estructura principal. Tras esto, se mostrará el diagrama de recursos REST y, finalmente, el de despliegue del sistema.

REST son las siglas de *representational state transfer*, transferencia de estado representacional. Es una interfaz para conectar sistemas basada en el protocolo HTTP, transfiriendo los datos en formatos específicos, como XML y JSON. Es una alternativa a la interfaz SOAP que pretende reducir la complejidad de la misma [12].

5.2. Arquitectura

La arquitectura la constituyen tanto la forma en la que se han estructurado los paquetes como sus dependencias. En la Figura 5.1 se muestra el diagrama de paquetes del sistema.

El paquete “gateway” es la parte del sistema que recibe todo el tráfico externo al sistema y hace las funciones de punto de entrada en el patrón de diseño *fachada*. Este es necesario debido a la restricción de que solamente hay un puerto habilitado en la máquina virtual para

gestionar el tráfico externo. Aunque esta restricción sea el motivo de esta decisión de diseño, presenta la ventaja de que permite que el sistema sea mas fácilmente migrable, puesto que no siempre se puede disponer de más de un puerto público.

A nivel interno, existen otros 4 paquetes. En primer lugar, el paquete “Discord” es el sistema externo del que el sistema en desarrollo depende. En segundo lugar, el paquete “client” contiene el cliente Angular, encargado de las vistas del sistema y el controlador de interfaces, además de alguna funcionalidad adicional que permite que el controlador descargue carga de trabajo en el backend, como el cálculo de métricas o el tiempo de respuesta medio. En tercer lugar, el paquete “discordbot”, el cual depende también de “Discord”, ya que lo usa para la monitorización de servidores (se diría que “discordbot” es observador de “Discord”). Por último, en cuarto lugar, el paquete “encuestas” constituye el controlador de encuestas y la persistencia, es decir, “discordbot” depende de este para realizar las operaciones de almacenamiento de datos en persistencia. Cabe añadir que “client” depende de “encuestas” sobre un patrón Modelo Vista Controlador (MVC), ya que el cliente se comunica mediante peticiones REST con “encuestas” y mediante “gateway”, dado que esta comunicación requiere un puerto público. Por lo tanto, es “gateway” el que redirigirá a “encuestas”.

En la Figura 5.2 se muestra el diagrama de estilo de módulos y estilo de usos del paquete “client”, la Figura 5.3 del paquete “surveys” 5.3, la Figura 5.4 muestra el del paquete “discordbot” y finalmente, la Figura 5.5 muestra el del “gateway”.

5.3. Recursos

Debido a que la arquitectura del sistema es REST, en la Figura 5.6 se muestra el diagrama de recursos. En este diagrama se muestran los Path de cada recurso del sistema desde la perspectiva de la fachada, es decir, también se muestran los recursos del cliente en el que se encuentran las vistas a partir de “/client”.

5.4. Modelo lógico de datos

En la Figura 5.7 se muestra el modelo de datos del sistema, es decir, la estructura de la base de datos. Además en la parte inferior de la figura se muestra la existencia de un “HashMap”, en el que se almacena como clave el identificador de usuario de Discord “Snowflake” de todos los usuarios que estén en un servidor en el que esté el bot, y como valor otro “HashMap” en el que clave es el identificador de cada servidor “Snowflake” en el que se ha encontrado a este usuario y como valor si tiene o no el rol “Profesor”. Este “HashMap” que se encuentra en “discordbot”, no es como tal persistente, pero se construye mediante la observación de “discordbot” a “Discord”, de forma que este se actualiza de forma asíncrona.

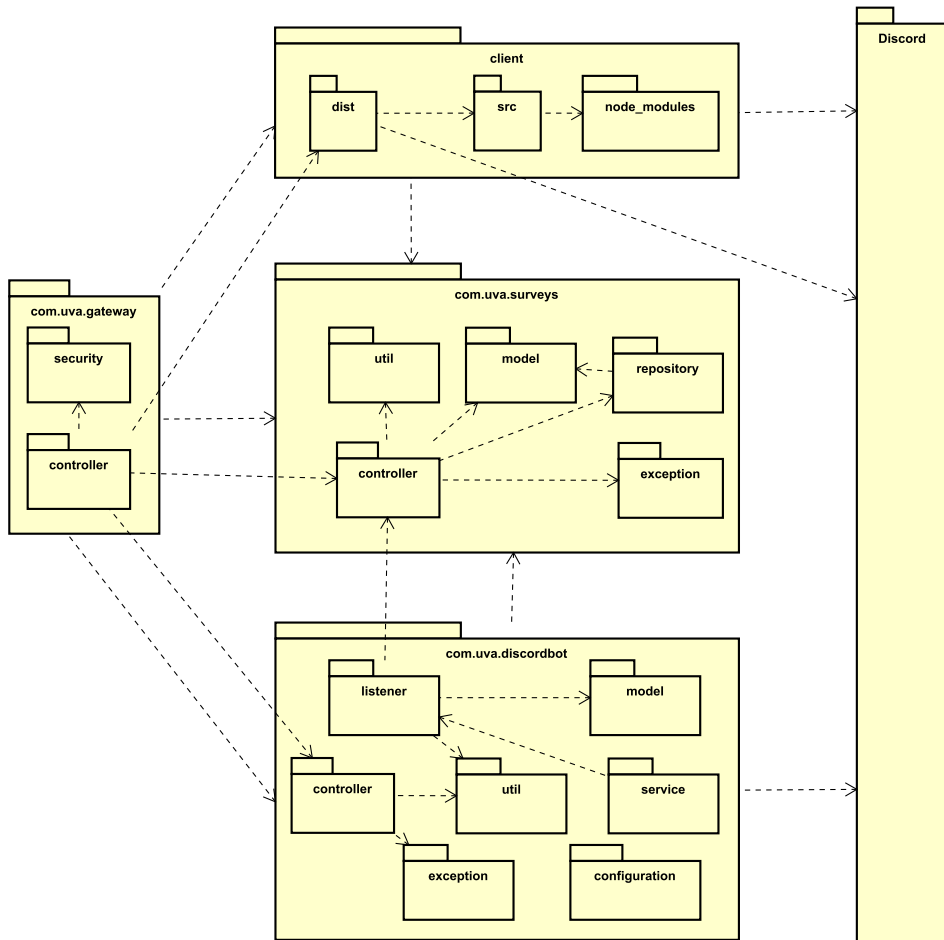


Figura 5.1: Diagrama de paquetes de la arquitectura.

5.5. Despliegue

El despliegue es la forma en la que el sistema se ejecuta. En la Figura 5.8 se muestra el diagrama de despliegue del sistema. La arquitectura del despliegue se basa en micro-servicios. Estos se ejecutan en contenedores Docker. El contenedor “Poll Microservice” depende del contenedor “Poll Database” para almacenar o consultar datos de persistencia.

Toda la comunicación entre las partes del sistema desarrollado es mediante http, a excepción de la comunicación con la base de datos, en la que se usa JPA para mantener las virtudes de la programación orientada a objetos y la comunicación con la API Discord en https.

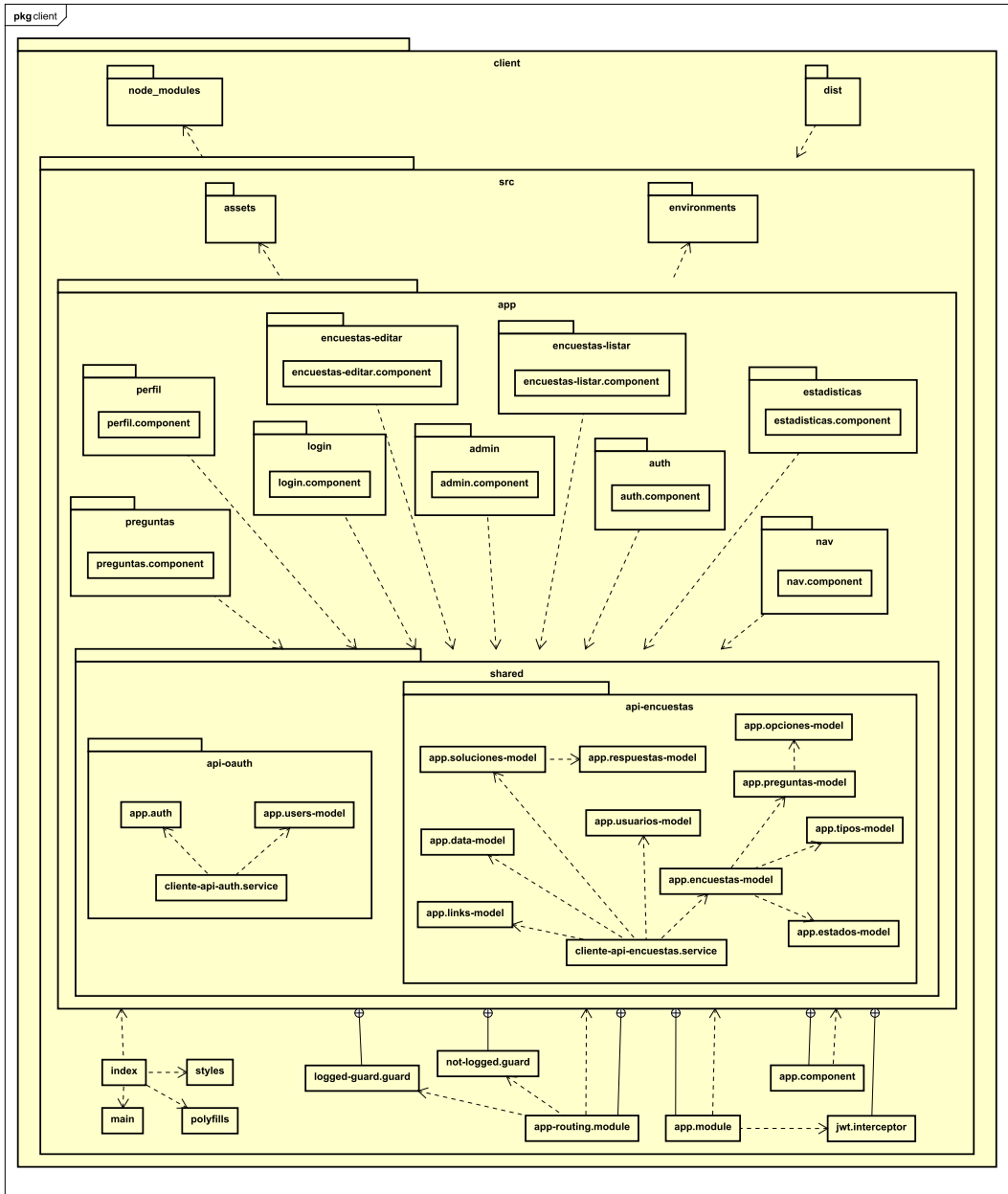


Figura 5.2: Diagrama de clases (ModulesAndUsesStyle) de "client".

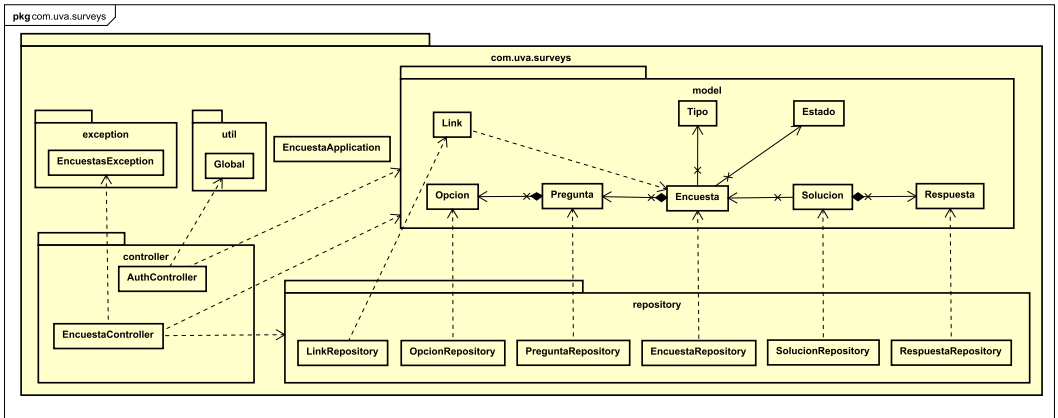


Figura 5.3: Diagrama de clases (ModulesAndUsesStyle) de “surveys”.

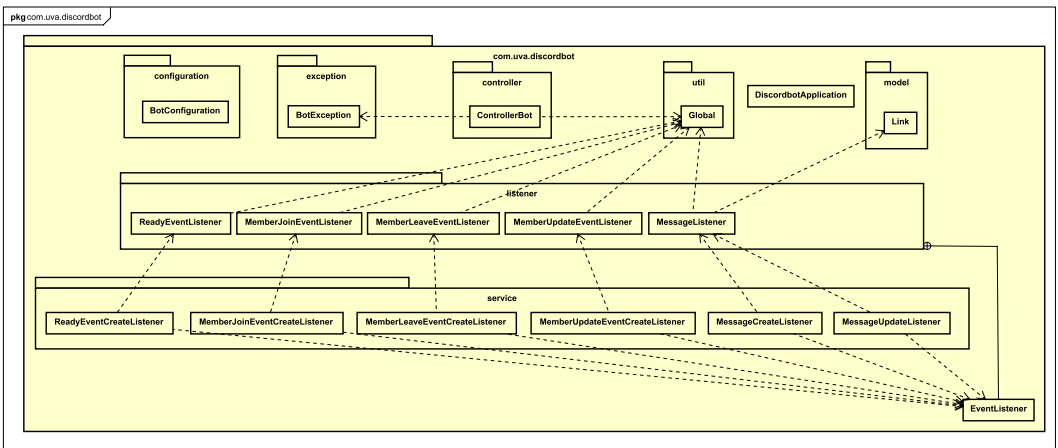


Figura 5.4: Diagrama de clases (ModulesAndUsesStyle) de “discordbot”.

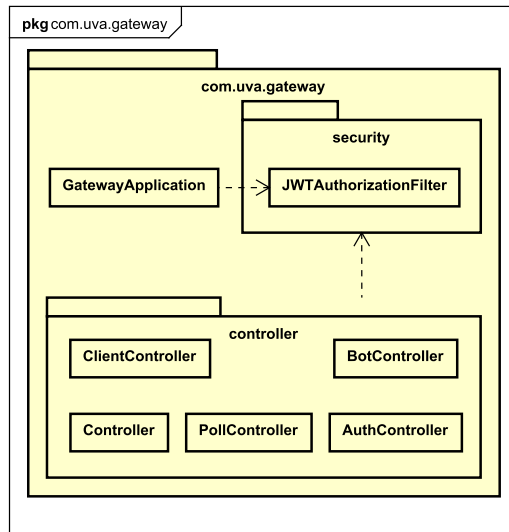


Figura 5.5: Diagrama de clases (ModulesAndUsesStyle) de “gateway”.

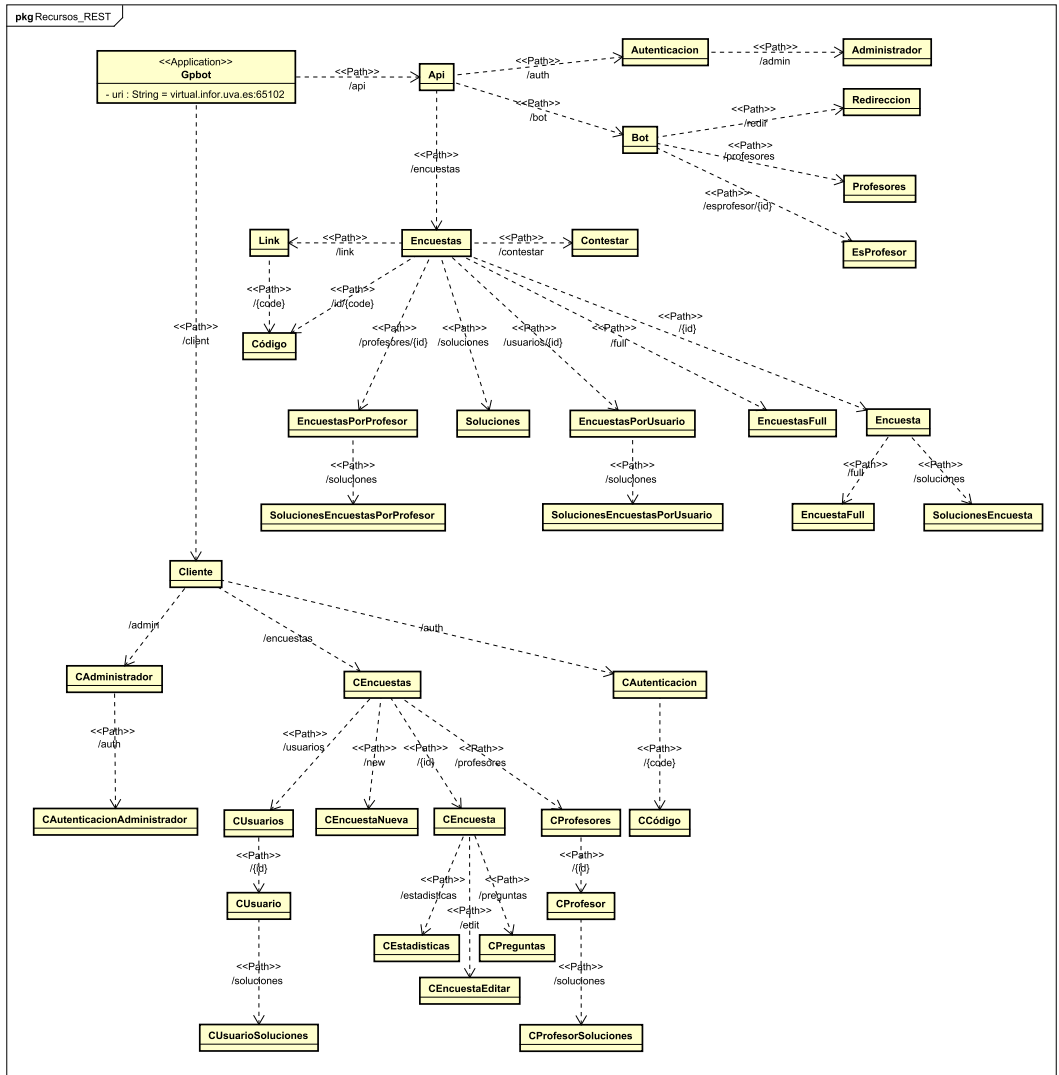


Figura 5.6: Diagrama de clases de los Recursos REST.

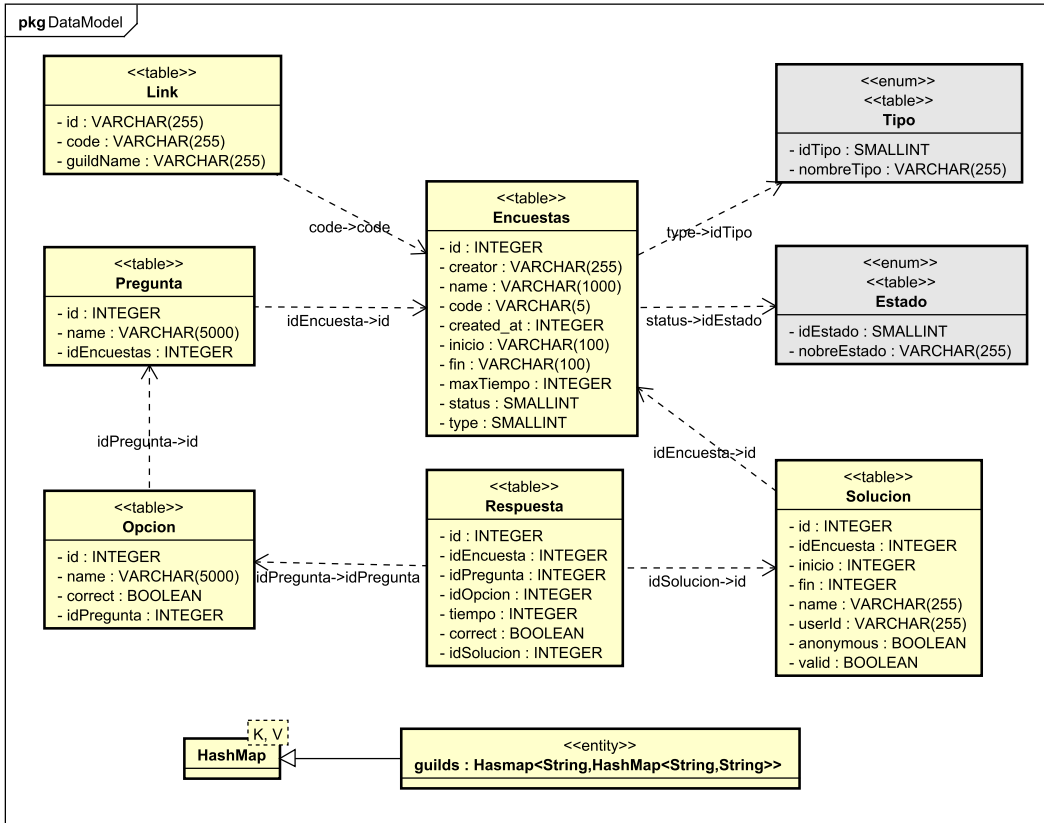


Figura 5.7: Diagrama de clases del Modelo Lógico de Datos.

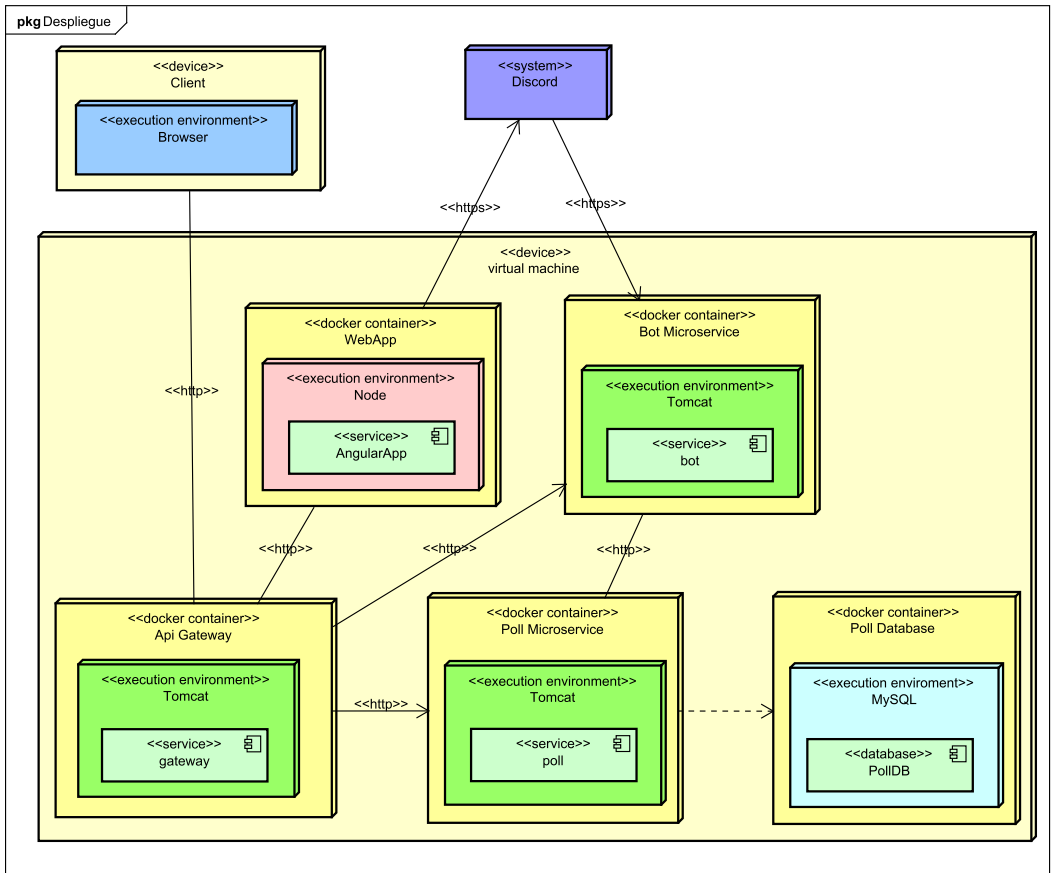


Figura 5.8: Diagrama de Despliegue.

Capítulo 6

Implementación y pruebas

6.1. Introducción

Este capítulo presenta tanto la implementación del sistema como las pruebas realizadas al mismo. La implementación es la forma específica en la que se ha organizado todo el software del sistema y, con la finalidad de ilustrarla, se mostrarán capturas de pantalla en la que se visualizará cómo funciona en su totalidad. Las pruebas constituyen la etapa de la ingeniería de software en la que se pretende validar si los requisitos han sido satisfechos. En este caso, se han realizado pruebas de caja negra, consistentes en validar si un determinado módulo del sistema solo permite las entradas que cumplen la pre-condición, solo genera valores que están dentro de la post-condición y mantiene lo definido como invariante. Además, también se han realizado pruebas de usuario tanto a profesores como a alumnos, con las que se busca comprobar cómo utilizan los usuarios finales el sistema. Para ello, a los usuarios se les solicita que realicen una determinada tarea con el sistema y se los monitoriza para verificar si son capaces de realizarla debidamente.

6.2. Implementación del sistema

6.2.1. Estructura de directorios

El software que compone el sistema está dividido en cuatro directorios, tal como se ilustra en la Figura 6.1. Cada uno de estos directorios está asociado a un micro-servicio:

- El directorio “client” contiene todo del software encargado de la visualización de las vistas y consiste en un cliente Angular que realiza las peticiones al backend.
- El directorio “encuestas” contiene el software que forma la API del backend de gestión de encuestas.

6.2. IMPLEMENTACIÓN DEL SISTEMA

- El directorio “discordbot” contiene el software de monitorización de servidores de Discord.
- El directorio “gateway” contiene el software encargado de redirigir el tráfico http para que solo sea necesario el uso de un puerto público.

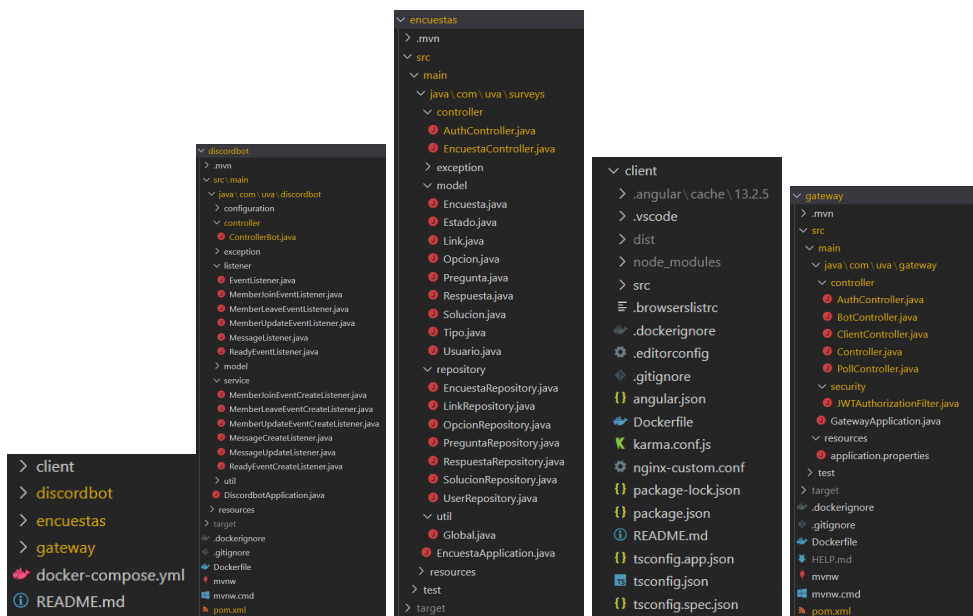


Figura 6.1: Estructura de directorios del sistema.

6.2.2. Vistas del sistema

El sistema se compone de 16 vistas: autenticación, inicial, responder una pregunta, responder una pregunta y terminar la encuesta, administración, login de administración, perfil de profesor, perfil de usuario, ranking, perfil de usuario con resultados, perfil de profesor con resultados, crear una encuesta, crear una encuesta y enviarla, añadir una pregunta a una encuesta, añadir una respuesta a una encuesta y comando de publicación. Cada una de ellas se explica detalladamente a continuación.

Al utilizar el sistema por primera vez, se accede a la vista en la que se realiza la autenticación en Discord (ver Figura 6.2). El sistema redirige al usuario a esta vista para realizar una autenticación de tipo OAuth. Cabe señalar que el sitio web desarrollado no tendrá acceso nunca a la credenciales de Discord, sino que será el token de acceso que devuelve esta API el que se usará para obtener los datos de usuario necesarios.

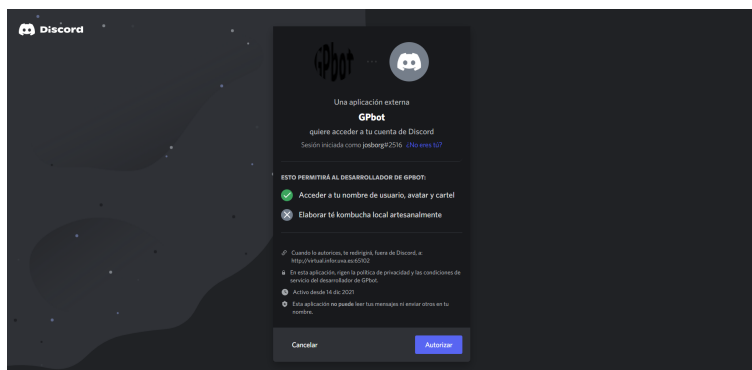


Figura 6.2: Pantalla (vista) de autenticación.

Una vez realizada la autenticación, se redirigirá al usuario a la pantalla que se muestra en la Figura 6.3. Esta es la pantalla inicial, donde se muestra la política de privacidad del sistema. Además, se muestra un botón con el texto “Ir a encuesta”, el cual permitirá acceder a la pantalla de contestación de encuestas si se ha accedido al sistema mediante un “link” generado en un servidor Discord.

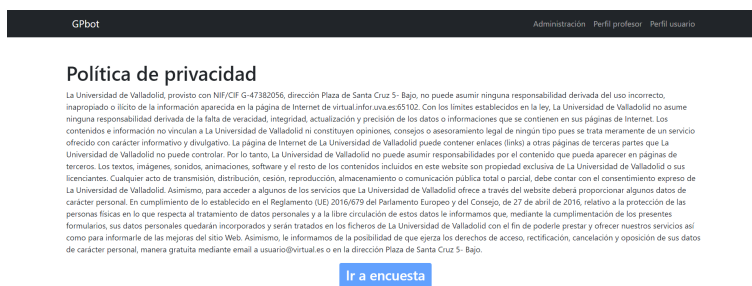


Figura 6.3: Pantalla (vista) inicial donde se muestra la política de privacidad.

La vista que se mostrará cuando se vaya a contestar una pregunta de una determinada encuesta será como la que puede verse en la Figura 6.4, en el caso de que meramente se vaya a responder a una pregunta, o bien como la ilustrada en la Figura 6.5 en el caso de que, además de responder a una pregunta, vaya a darse por concluida la encuesta.



Figura 6.4: Pantalla (vista) para responder una pregunta.

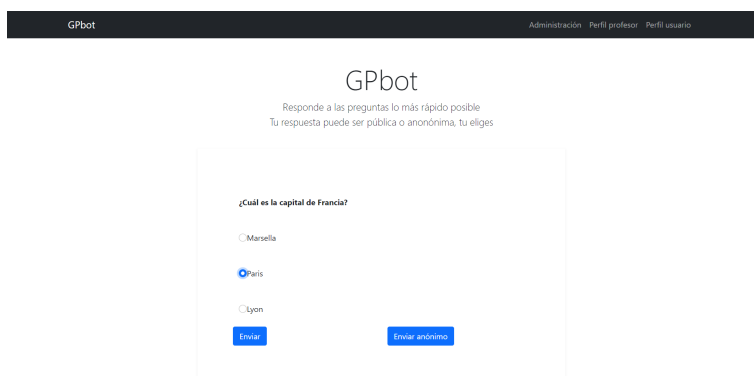


Figura 6.5: Pantalla (vista) para responder una pregunta y terminar la encuesta.

Además de estas vistas, asociadas al proceso de contestación de una determinada encuesta, hay otras disponibles únicamente para las gestiones asociadas al administrador del sistema. En la parte superior de las vistas del sistema, se muestra la barra de navegación que permite acceder a la pantalla de administración (ver Figura 6.6). La función de administración del sistema es la que permite añadir el bot a un servidor Discord. En el caso de no estar registrado como administrador, será necesario registrarse mediante las credenciales de administración (accediéndose para ello a la vista mostrada en la Figura 6.7).

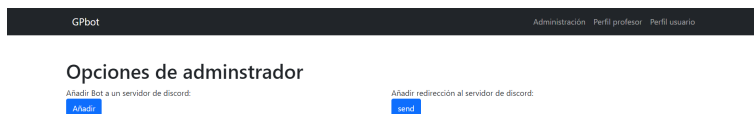


Figura 6.6: Pantalla (vista) de administración del sistema.

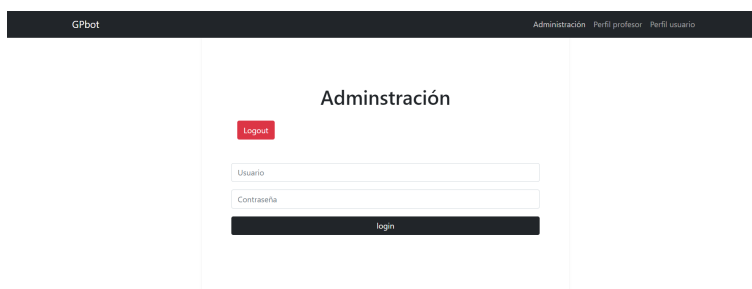


Figura 6.7: Pantalla (vista) de login para acceder a la administración del sistema.

Además de acceder a las opciones asociadas al rol de administrador, a través de la mencionada barra de navegación también se puede acceder al perfil de usuario (ver Figura 6.9), que es el correspondiente a los/as alumnos/as, y al perfil de profesor (ver Figura 6.8).

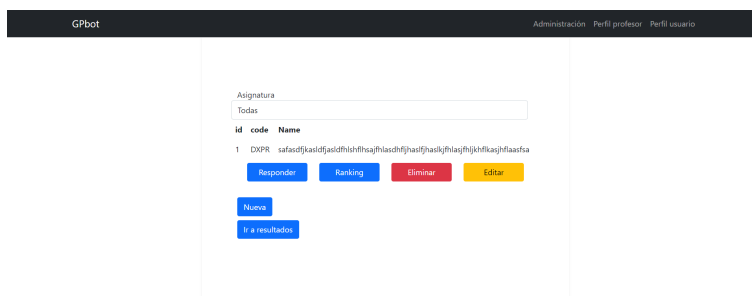


Figura 6.8: Pantalla (vista) del perfil de profesor.

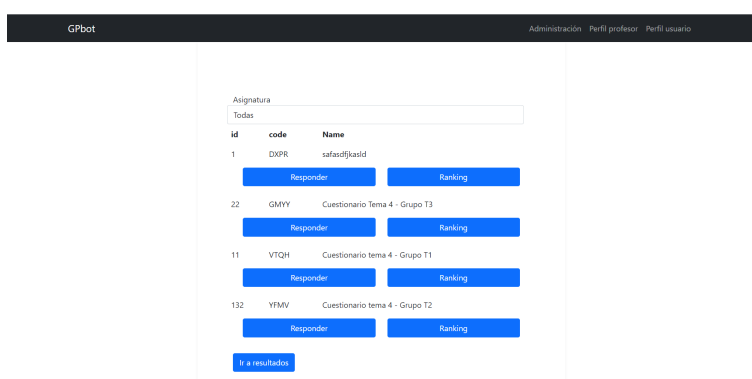


Figura 6.9: Pantalla (vista) del perfil de usuario.

El perfil de usuario muestra un listado de las encuestas disponibles, permite acceder al ranking correspondiente a una determinada encuesta mediante el botón “ranking” (ver

6.2. IMPLEMENTACIÓN DEL SISTEMA

Figura 6.10), empezar a contestar una de las encuestas no completadas todavía mediante el botón “responder” que conduce a vistas mostradas anteriormente (Figura 6.4 o 6.5) y, por último, también permite mostrar los resultados de usuario (ver Figura 6.11) mediante el botón “Ir a resultados”, donde se permite el filtrado de resultados por encuestas y asignaturas.

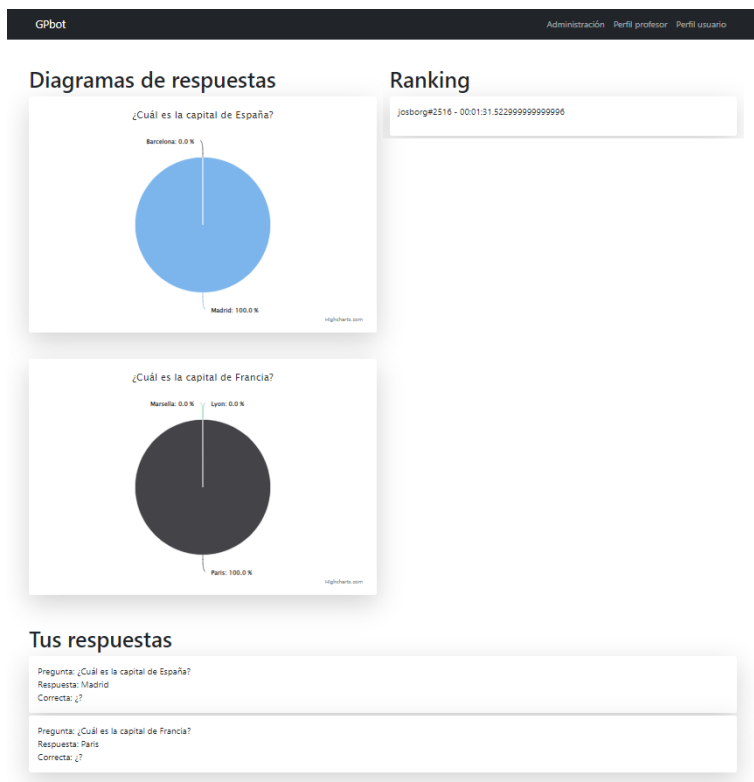


Figura 6.10: Pantalla (vista) del ranking correspondiente a una determinada encuesta.

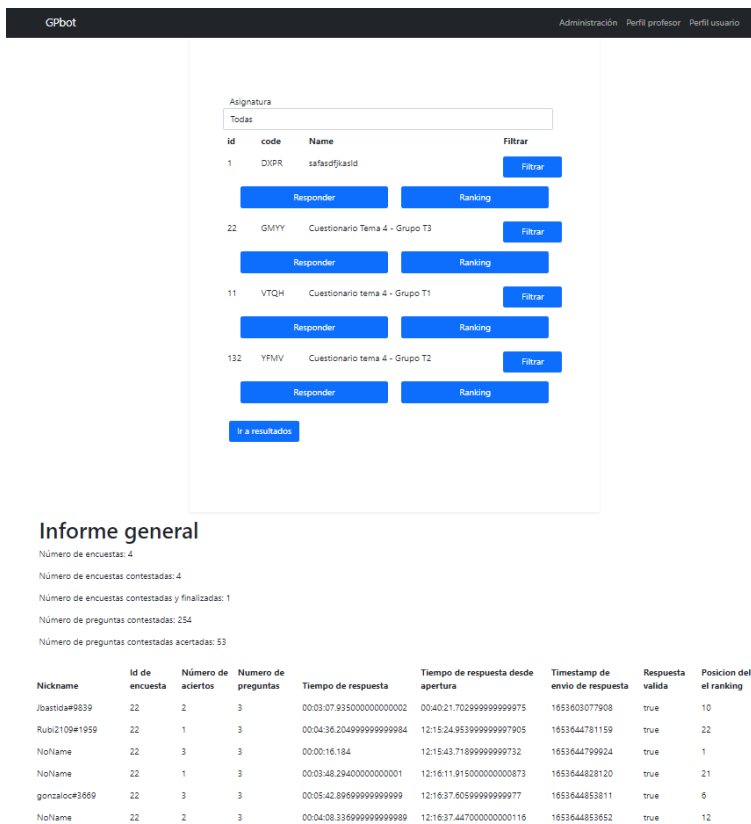


Figura 6.11: Pantalla (vista) del perfil de usuario con resultados.

El perfil de profesor permite acceder a las funcionalidades asociadas a los/as profesores/as. Estas son las mismas que las permitidas para un usuario (alumno/a) y, adicionalmente, la posibilidad de crear, editar y borrar encuestas. Las pantallas de edición y creación de encuestas son las mostradas en las Figuras 6.12, 6.13, 6.14 y 6.15. Para que los cambios realizados en una encuesta se almacenen, será necesario “enviar” la encuesta. Cabe notar que si se editan las preguntas u opciones de una determinada encuesta que ya haya recibido respuestas de usuarios, estas respuestas se perderán; además, si el estado de una encuesta es “finalizada”, no se podrá editar. En la Figura 6.16 se muestran los resultados a los que puede acceder un “profesor”. Estos son los correspondientes a las respuestas de todos los usuarios que han respondido a las encuestas de las que tiene acceso el/la profesor/a. En la Figura 6.17 se muestra el comando que se utiliza desde el perfil de profesor para la publicación de encuestas.

GPbot Administración Perfil profesor Perfil usuario

Título
Prueba dos preguntas

Tipo de ranking
RESPUESTA

Estado
CERRADA

DeadLine, Tiempo máximo de respuesta en segundos (0 si no hay Tiempo máximo)
3600

Momento desde el que se podrá contestar, si el estado es ABIERTA
06/06/2022 18:33

Momento desde el que ya no se podrá contestar
07/06/2022 18:33

Momento de creación (Si tipo de ranking APERTURA, el ranking tendrá encuesta el tiempo de respuesta desde aquí)
1654540381613

Figura 6.12: Pantalla (vista) para crear una encuesta.

Añadir pregunta

Pregunta
¿Cuál es la capital de España?

Cancelar Añadir pregunta

Figura 6.13: Pantalla (vista) para añadir una pregunta a una encuesta.

Añadir respuesta

Respuesta
Madrid

Marcar, si es correcta:

Cancelar Añadir respuesta

Figura 6.14: Pantalla (vista) para añadir una respuesta a una encuesta.

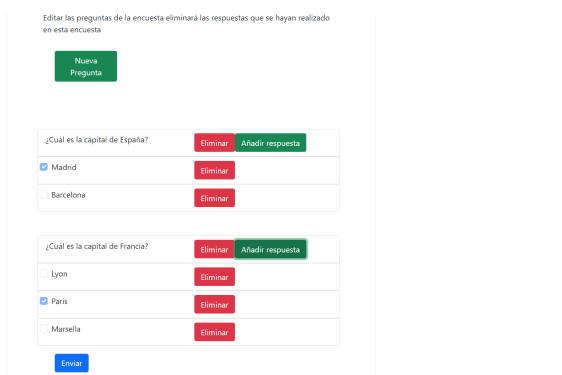
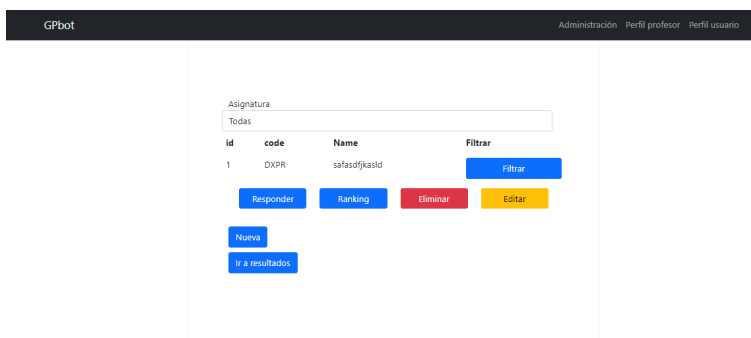


Figura 6.15: Pantalla (vista) para crear una encuesta y enviarla.



Informe general

Número de encuestas: 1
 Número de encuestas contestadas: 1
 Número de preguntas contestadas: 2
 Número de preguntas contestadas acertadas: 1

Nickname	Número de aciertos	Número de preguntas	Número de encuestas	Número de encuestas correctas	Número de nTop1	Mejor posición de ranking	Posición media de ranking	Tiempo de respuesta medio	Tiempo de respuesta medio desde apertura	Tasa de aciertos de preguntas	Tasa de aciertos de encuestas	Filtrar
josborg#2516	1	2	1	0	1	1	1	00:00:14.62	00:10:05.9900000000000000	0.5	0	Filtrar

Nickname	Id de encuesta	Número de aciertos	Número de preguntas	Tiempo de respuesta	Tiempo de respuesta desde apertura	Timestamp de envío de respuesta	Respuesta valida	Posicion del el ranking
josborg#2516	1	1	2	00:00:14.615	00:10:09.9930000000000052	1653395783681	true	1

Figura 6.16: Pantalla (vista) del perfil de profesor con resultados.

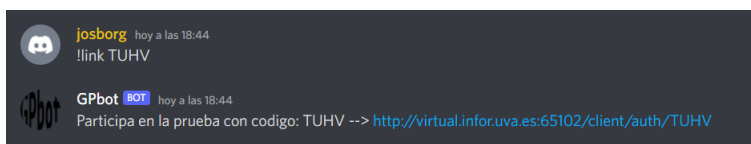


Figura 6.17: Pantalla (vista) del comando de publicación de una determinada encuesta.

6.3. Pruebas del sistema

6.3.1. Pruebas de integración

Para comprobar el funcionamiento del servicio, se ha realizado una batería de pruebas de caja negra de cada una de las operaciones de la API desarrollada, a través de la aplicación Advanced REST client.

En esta sección se exponen los casos de prueba más importantes recogidos en las Tablas 6.1-6.23. En cada una de dichas tablas se indica, en cada fila: el código asociado al caso de prueba concreto (CPX) y nombre; la descripción de dicho caso de prueba que define en que consiste dicha prueba; API REST define el tipos de operación REST de la prueba y la ubicación del recurso que se está probando; la entrada es el contenido del body de la petición http asociada a la prueba; el resultado esperado es el contenido de la respuesta que se debería de recibir para que la prueba se considere superada.

CP01	Crear encuesta
Descripción	Crear una nueva encuesta en el sistema
API REST	POST:/api/encuestas
Entrada	{id:0, name:Pruaba, creator:689553946569605159, code:, createdAt:1655983117032, preguntas: [{id:0, name:¿Pregunta 1?, opciones: [{id:0, name:Sí, correct:true}, {id:0, name:No, correct:false}] }, {id:0, name:La suma de 3 mas 4 es...”, opciones: [{id:0, name:6, correct:false}, {id:0, name:7, correct:true}, {id:0, name:8, correct:false}] }]], inicio:2022-06-23T11:18, fin:2022-06-24T11:18, maxTiempo:3600, status:0, type:0}
Resultado esperado	Nuevo registro creado

Tabla 6.1: Caso de prueba *Crear encuesta*.

CP02	Obtener encuesta completa
Descripción	Obtener la encuesta que se ha creado anteriormente
API REST	GET:/api/encuestas/{id}/full
Entrada	{}
Resultado esperado	{ id:502, creator:689553946569605159, name:Pruaba, code:ZCZT, preguntas: [{ id:503, name:¿Pregunta 1?, opciones: [{ id:504, name:Sí, correct:true }, { id:505, name:No, correct:false }] }, { id:506, name:La suma de 3 mas 4 es..., opciones: [{ id:507, name:6, correct:false }, { id:508, name:7, correct:true }, { id:509, name:8, correct:false }] }]], inicio:2022-06-23T11:18, fin:2022-06-24T11:18, maxTiempo:3600, status:CERRADA, type:RESPUESTA, createdAt:1655983117032 }

Tabla 6.2: Caso de prueba *Obtener encuesta completa*.

CP03	Editar encuesta
Descripción	Editar la encuesta especificada
API REST	PUT:/api/encuestas/{id}
Entrada	{ id:502, creator:689553946569605159, name:Pruaba, code:ZCZT, preguntas: [{ id:503, name:¿Pregunta 1?, opciones: [{ id:504, name:Sí, correct:true }, { id:505, name:No, correct:false }] }, { id:506, name:La suma de 3 mas 4 es..., opciones: [{ id:507, name:6, correct:false }, {id:508, name:7, correct:true }, { id:509, name:8, correct:false }] }]], inicio:2022-06-23T11:18, fin:2022-06-24T11:18, maxTiempo:3600, status:1, type:0, createdAt:1655983117032 }
Resultado esperado	Encuesta editada

Tabla 6.3: Caso de prueba *Editar encuesta*.

CP04	Obtener encuesta
Descripción	Obtener la encuesta especificada
API REST	GET:/api/encuestas/{id}
Entrada	{}
Resultado esperado	{ id:502, creator:689553946569605159, name:Pruaba, code:ZCZT, preguntas: [{ id:503, name:¿Pregunta 1?, opciones: [{ id:504, name:Sí, correct:false }, { id:505, name:No, correct:false }] }, { id:506, name:La suma de 3 mas 4 es..., opciones: [{ id:507, name:6, correct:false }, {id:508, name:7, correct:false }, { id:509, name:8, correct:false }] }]], inicio:2022-06-23T11:18, fin:2022-06-24T11:18, maxTiempo:3600, status:ABIERTA, type:RESPUESTA, createdAt:1655983117032 }

Tabla 6.4: Caso de prueba *Obtener encuesta*.

CP05	Borrar encuesta
Descripción	Borra la encuesta especificada
API REST	DELETE:/api/encuestas/{id}
Entrada	{}
Resultado esperado	Encuesta borrada

Tabla 6.5: Caso de prueba *Borrar encuesta*.

CP06	Responder encuesta especificada
Descripción	Responder a la encuesta especificada
API REST	POST:/api/encuestas/{id}
Entrada	{ id:0, idEncuesta:502, respuestas: [{ id:0, idEncuesta:502, idPregunta:503, idOpcion:504, tiempo:11454, correct:false }, {id:0, idEncuesta:502, idPregunta:506, idOpcion:508, tiempo:12213, correct:false }], inicio:1655986641867, fin:1655986669089, diff:0, name:josborg#2516, anonymous:false, valid":false, guild: , userId:689553946569605159 }
Resultado esperado	Conjunto de respuestas almacenado

Tabla 6.6: Caso de prueba *Responder encuesta especificada*.

CP07	Obtener resultados de la encuesta especificada
Descripción	Obtener los resultados asociados a la encuesta especificada
API REST	GET:/api/encuestas/{id}/soluciones
Entrada	{}
Resultado esperado	[{id:510, idEncuesta:502, respuestas: [{ id:511, idEncuesta:502, idPregunta:503, idOpcion:504, tiempo:11454, correct:false }, { id:512, idEncuesta:502, idPregunta:506, idOpcion:508, tiempo:12213, correct:false }]], inicio:1655986641867, fin:1655986669089, name:josborg#2516, userId:689553946569605159, anonymous:false, valid:true, diff:27222 }]

Tabla 6.7: Caso de prueba *Obtener resultados de la encuesta especificada*.

CP08	Obtener encuestas completas
Descripción	Obtener la información de las encuestas completas
API REST	GET:/api/encuestas/full
Entrada	{}
Resultado esperado	[{ id:502, creator:689553946569605159, name:Pruaba, code:ZCZT, preguntas: [{ id:503, name:¿Pregunta 1?, opciones: [{ id:504, name:Sí, correct:true }, { id:505, name:No, correct:false }] }, { id:506, name:La suma de 3 mas 4 es..., opciones: [{ id:507, name:6, correct:false }, {id:508, name:7, correct:true }, { id:509, name:8, correct:false }] }]], inicio:2022-06-23T11:18, fin:2022-06-24T11:18, maxTiempo:3600, status:CERRADA, type:RESPUESTA, createdAt:1655983117032 }]

Tabla 6.8: Caso de prueba *Obtener encuestas completas*.

CP09	Obtener encuestas de usuario con id especificado
Descripción	Obtener las encuestas realizadas por el usuario cuyo id coincide con el especificado
API REST	GET:/api/encuestas/usuarios/{id}
Entrada	{}
Resultado esperado	[{ id:502, creator:689553946569605159, name:Pruaba, code:ZCZT, preguntas: [{ id:503, name:¿Pregunta 1?, opciones: [{ id:504, name:Sí, correct:false }, { id:505, name:No, correct:false }] }, { id:506, name:La suma de 3 mas 4 es..., opciones: [{ id:507, name:6, correct:false }, {id:508, name:7, correct:false }, { id:509, name:8, correct:false }] }]], inicio:2022-06-23T11:18, fin:2022-06-24T11:18, maxTiempo:3600, status:CERRADA, type:RESPUESTA, createdAt:1655983117032 }]

Tabla 6.9: Caso de prueba *Obtener encuestas de usuario con id especificado*.

CP10	Obtener soluciones de usuario con id especificado
Descripción	Obtiene las soluciones de usuario con el id especificado
API REST	GET:/api/encuestas/usuarios/{id}/soluciones
Entrada	{}
Resultado esperado	[{id:510, idEncuesta:502, respuestas: [{ id:511, idEncuesta:502, idPregunta:503, idOpcion:504, tiempo:11454, correct:false }, { id:512, idEncuesta:502, idPregunta:506, idOpcion:508, tiempo:12213, correct:false }], inicio:1655986641867, fin:1655986669089, name:josborg#2516, userId:689553946569605159, anonymous:false, valid:true, diff:27222 }]

Tabla 6.10: Caso de prueba *Obtener resultados de usuario con id especificado*.

CP11	Obtener encuestas de profesor con id especificado
Descripción	Obtiene las encuestas del profesor cuyo id coincide con el especificado
API REST	GET:/api/encuestas/profesores/{id}
Entrada	{}
Resultado esperado	[{ id:502, creator:689553946569605159, name:Pruaba, code:ZCZT, preguntas: [{ id:503, name:¿Pregunta 1?, opciones: [{ id:504, name:Sí, correct:true }, { id:505, name:No, correct:false }] }, { id:506, name:La suma de 3 mas 4 es..., opciones: [{ id:507, name:6, correct:false }, {id:508, name:7, correct:true }, { id:509, name:8, correct:false }] }]], inicio:2022-06-23T11:18, fin:2022-06-24T11:18, maxTiempo:3600, status:ABIERTA, type:RESPUESTA, createdAt:1655983117032 }]

Tabla 6.11: Caso de prueba *Obtener encuestas de profesor con id especificado*.

CP12	Obtener soluciones de profesor con id especificado
Descripción	Obtiene las soluciones del profesor con el id especificado
API REST	GET:/api/encuestas/profesores/{id}/soluciones
Entrada	{ }
Resultado esperado	[{id:510, idEncuesta:502, respuestas: [{ id:511, idEncuesta:502, idPregunta:503, idOpcion:504, tiempo:11454, correct:true }, { id:512, idEncuesta:502, idPregunta:506, idOpcion:508, tiempo:12213, correct:true }], inicio:1655986641867, fin:1655986669089, name:josborg#2516, userId:689553946569605159, anonymous:false, valid:true, diff:27222 }]

Tabla 6.12: Caso de prueba *Obtener resultados de profesor con id especificado*.

CP13	Obtener soluciones
Descripción	Obtener las soluciones
API REST	GET:/api/encuestas/soluciones
Entrada	{ }
Resultado esperado	[{id:510, idEncuesta:502, respuestas: [{ id:511, idEncuesta:502, idPregunta:503, idOpcion:504, tiempo:11454, correct:false }, { id:512, idEncuesta:502, idPregunta:506, idOpcion:508, tiempo:12213, correct:false }]

Tabla 6.13: Caso de prueba *Obtener soluciones*.

CP14	Generar code
Descripción	Genera un código de encuesta válido
API REST	GET:/api/encuestas/link
Entrada	{ }
Resultado esperado	Code generado

Tabla 6.14: Caso de prueba *Generar code*.

CP15	Obtener una lista de los servidores en los que una encuesta ha sido publicada
Descripción	Obtiene la lista los servidores en los que la encuesta con code especificado ha sido publicada, probando una encuesta que todavía no ha sido publicada
API REST	GET:/api/encuestas/link/{code}
Entrada	{ }
Resultado esperado	[]

Tabla 6.15: Caso de prueba *Obtener una lista de los servidores en los que una encuesta ha sido publicada*.

CP16	Obtener una lista los servidores en los que una encuesta ha sido publicada
Descripción	Obtiene una lista los servidores en los que una encuesta ha sido publicada, probando una encuesta que ya ha sido publicada en el servidor con id 693623757809516585
API REST	GET:/api/encuestas/link/{code}
Entrada	{}
Resultado esperado	[{ guild:693623757809516585, code:ZCZT, guildName:LPPS 2 }]

Tabla 6.16: Caso de prueba *Obtener una lista los servidores en los que una encuesta ha sido publicada.*

CP17	Obtener el code a partir de un id
Descripción	Obtiene el código de una encuesta a partir de un id
API REST	GET:/api/encuestas/{id}/code
Entrada	GET:/api/encuestas/ZCZT/code
Resultado esperado	502

Tabla 6.17: Caso de prueba *Obtener el id a partir de un code.*

CP18	Iniciar sesión
Descripción	Inicia sesión
API REST	POST:/api/auth
Entrada	{ user:689553946569605159, code:null }
Resultado esperado	Token jwt de autenticación de usuario

Tabla 6.18: Caso de prueba *Iniciar sesión.*

CP19	Iniciar sesión como administrador
Descripción	Inicia sesión como administrador
API REST	POST:/api/auth/admin
Entrada	{ user:root, code:root }
Resultado esperado	Token jwt de autenticación de administrador

Tabla 6.19: Caso de prueba *Iniciar sesión como administrador.*

6.3. PRUEBAS DEL SISTEMA

CP20	Comprobar encuesta
Descripción	Comprueba si la encuesta se puede contestar
API REST	POST:/api/encuestas/contestar
Entrada	{ idEncuesta:502, idUser:689553946569605159, username:josborg#2516 }
Resultado esperado	true

Tabla 6.20: Caso de prueba *Comprobar encuesta*.

CP21	Enviar URL del cliente al backend
Descripción	Envía la URL del cliente al backend
API REST	GET:/api/bot/redir
Entrada	http://virtual.infor.uva.es:65102/client/auth
Resultado esperado	OK

Tabla 6.21: Caso de prueba *Enviar URL del cliente al backend*.

CP22	Listar profesores
Descripción	Lista los usuarios que son profesor
API REST	GET:/api/bot/profesores
Entrada	{ }
Resultado esperado	Lista de usuarios profesor

Tabla 6.22: Caso de prueba *Listar profesores*.

CP23	Comprobar profesor
Descripción	Comprueba si el usuario con el id especificado es profesor
API REST	GET:/api/bot/esprofesor/{id}
Entrada	GET:/api/bot/esprofesor/689553946569605159
Resultado esperado	true

Tabla 6.23: Caso de prueba *Comprobar profesor*.

Prueba	Resultado
CP01	OK
CP02	OK
CP03	OK
CP04	OK
CP05	Error
CP06	OK
CP07	OK
CP08	Error
CP09	OK
CP10	OK
CP11	OK
CP12	OK
CP13	Error
CP14	OK
CP15	OK
CP16	OK
CP17	OK
CP18	OK
CP19	OK
CP20	OK
CP21	OK
CP22	Error
CP23	OK

Tabla 6.24: Tabla de resultados de Casos de Prueba

Como resultado de esa prueba se detectaron los errores mostrados en la Tabla 6.24. En cuanto al error de ejecución detectado mediante el caso de prueba CP05 referente a probar el borrado de encuestas, no se ha conseguido encontrar una solución por lo tanto se ha decidido bloquear esta funcionalidad. En cuanto a la prueba CP08, el error consistió en que no se devolvía de forma correcta cuales eran las opciones correctas de las encuestas y en su lugar todas se mostraban como incorrectas. La solución fue añadir una operación adicional en el backend que se encargase de este recuso de forma específica y no reutilizar código para esa parte en concreto. CP13 detectó que el recurso “*GET:/api/encuestas/soluciones*” estaba bloqueado por la seguridad del backend. Para solucionarlo, simplemente fue necesario añadir un filtro en la configuración de la seguridad del micro-servicio “gateway”. Por último el error detectado mediante CP22 estaba causado porque el filtrado de peticiones del backend impedía el acceso al recurso, se añadió un nuevo filtro en la configuración de la seguridad que permitiese el acceso a este y el caso de prueba pudo terminar con éxito.

6.3.2. Pruebas de usuario

Durante el desarrollo de proyecto se validaron los requisitos y se comprobó la usabilidad del sistema mediante pruebas de usuario a los profesores y también a varios grupos de alumnos.

En ciertos puntos del desarrollo, con un prototipo del sistema ya listo, se mantuvieron dos reuniones con profesores que utilizan Discord en su asignatura y estas, detalladas a continuación, fueron las primeras pruebas de usuario realizadas. Tras ajustar el sistema según las carencias planteadas, se realizaron algunas pruebas con alumnos, tal como se detalla más adelante en esta misma sección.

Prueba de usuario con profesores

Tal como se ha mencionado, se mantuvieron dos reuniones con profesores del departamento que utilizan Discord en sus asignaturas. A continuación se presenta lo más destacado de dichas reuniones.

La primera reunión tuvo lugar el día 3 de mayo de 2022, siendo los profesores asistentes Javier Bastida Ibáñez, Arturo González Escribano, Rocío Carratalá Sáez y Franciso J. Andújar Muñoz. La prueba de usuario realizada tras presentar el sistema consistió en solicitar a los profesores que usaran el sistema para crear una encuesta, tras esto editarla fijando su estado a “abierta” (y permitiendo modificar los parámetros que considerasen), seguidamente publicar la encuesta en un servidor de Discord de prueba creado a propósito para la realización del proyecto, y, finalmente, revisar todas las encuestas publicadas, responderlas, acceder al ranking y a la pantalla resultados de profesor, editar la encuesta para fijar su estado a estado “finalizada” y, tras comprobar el cambio, borrar la encuesta.

Como resultado de esta prueba se descubrieron (y posteriormente se resolvieron) los siguientes problemas:

- El tamaño máximo del texto de una pregunta era demasiado pequeño.
- La interfaz del sistema no era responsive.
- El ranking no ordenaba a los participantes de una encuesta de la forma deseada por los profesores.
- No era posible conocer tus propias respuestas en el perfil de usuario.
- No se mostraba correctamente la respuesta si se respondía de forma anónima.

La segunda reunión tuvo lugar el día 12 de mayo de 2022, siendo los profesores asistentes Jesús Cámara Moreno, Yuri Torres de la Sierra, Javier Bastida Ibáñez, Rocío Carratalá Sáez y Franciso J. Andújar Muñoz. La prueba de usuario realizada coincidió con la planteada en la reunión anterior.

Como resultado de esta prueba de descubrieron (y posteriormente resolvieron) las siguientes carencias:

- Errores ortográficos en diferentes vistas del sistema.
- La distribución de botones del perfil de profesor resultaba poco intuitiva.
- Se sugirió aumentar la separación entre opciones al responder una pregunta.
- Se sugirió aumentar la separación entre preguntas a la hora de crear las encuestas.

Prueba de usuario a alumnos

Durante todo el proyecto se ha buscado tener un enfoque de desarrollo centrado en el usuario final. Para ello, los tutores han validado el funcionamiento durante el desarrollo del sistema, pero para validar si el sistema es realmente usable se consideró necesario realizar una prueba de usuario con los alumnos de la asignatura *Fundamentos de Computadoras*, ya habituados al uso de Discord. Esta prueba se realizó el día 30 de mayo de 2022.

Para llevar acabo la prueba, en primer lugar, se añadió el bot de Discord gpBOT al servidor de Discord “FC 2022” (el propio de la asignatura), y se publicaron tres encuestas: una para el grupo T1, otra para el T2 y la última para el T3. A los alumnos de cada grupo se les solicitó que contestaran la encuesta correspondiente y navegasen por la aplicación explorando sus funciones y tratando de buscar sus respuestas.

Esta prueba la realizaron un total de 66 alumnos/as, de los cuales 30 (el 45 % de la muestra) utilizaron un PC, 34 (el 48 %) un smartphone, y 2 (el 3 %) una tablet, tal como se ilustra en la Figura 6.18. Por lo tanto, se ha realizado de forma proporcional entre usuarios que han usado un PC y los que no y se considerar que las respuestas, en caso de existir tendencias en cuando a la satisfacción o insatisfacción, no se deben al dispositivo utilizado.

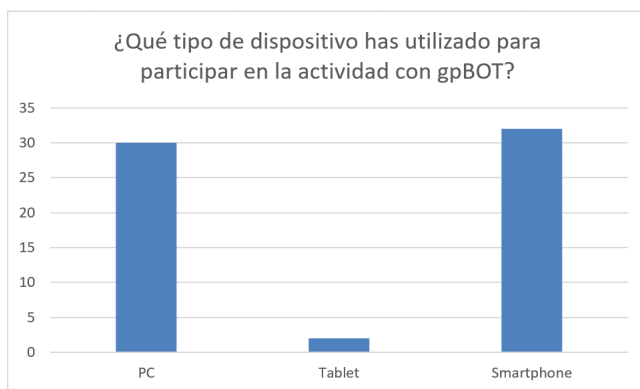


Figura 6.18: Clasificación de los participantes en la prueba de usuario según el dispositivo utilizado.

Tras la realización de la prueba de usuario, se solicitó al alumnado contestar a un cuestionario de satisfacción, mediante una herramienta independiente a este sistema (la propia del Campus Virtual de la UVa). Las ocho preguntas que componían dicha encuesta se detallan en la Tabla 6.25, siendo todas ellas de respuesta numérica, puntuando en una escala del 1 al 5 donde un 1 equivale a una opinión muy negativa, un 3 a una opinión neutral y un 5 a una muy positiva.

Identificador	Pregunta	Rango de respuestas posibles
P1	¿Estás satisfecho/a con la experiencia general de uso de gpBot?	1 (poco satisfecho/a) - 5 (muy satisfecho/a)
P2	¿En qué medida te parece intuitivo el uso de gpBot?	1 (poco intuitivo) - 5 (muy intuitivo)
P3	¿Cómo valorarías la interfaz de gpBot?	1 (muy mal) - 5 (muy bien)
P4	¿En qué medida te parece que la realización de este tipo de actividades en clase son adecuadas, la hacen más interesante y fomentan una participación más activa?	1 (muy poco) - 5 (mucho)
P5	¿En qué medida te parece sencillo el acceso a la encuesta de gpBOT una vez publicado el link correspondiente en Discord?	1 (muy difícil) - 5 (muy fácil)
P6	¿En qué medida te parece sencilla la participación en la encuesta de gpBOT una vez se accede a la misma?	1 (muy difícil) - 5 (muy fácil)
P7	¿En qué medida te parece importante poder contestar la encuesta de gpBOT de manera anónima?	1 (poco importante) - 5 (muy importante)
P7-2	Si has contestado de forma anónima a la encuesta, di el motivo.	Respuesta de texto libre
P8	¿En qué medida te parece adecuada la presentación de la evaluación de tus respuestas (correctas/incorrectas) una vez el profesor da por finalizada la encuesta?	1 (poco adecuada) - 5 (muy adecuada)

Tabla 6.25: Preguntas planteadas en la encuesta realizada a los/as alumnos/as tras completar la prueba de usuario.

Las Figuras 6.19-6.26 muestran las respuestas obtenidas en cada una de las preguntas de la encuesta.

En lo que respecta a la pregunta P1, en la Figura 6.19 se representan las respuestas obtenidas y se puede concluir, a partir de ellas, que el alumnado se muestra mayoritariamente satisfecho con la experiencia general de uso de gpBot. Además, según lo que puede observarse en la Figura 6.20, que muestra las respuestas referentes a la pregunta P2, puede afirmarse también que el uso de gpBot es altamente intuitivo, lo cual se alinea con una buena valoración

por parte de la mayor parte de los alumnos de su interfaz, como puede observarse en la Figura 6.21, donde se representan las respuestas obtenidas para la pregunta P3.

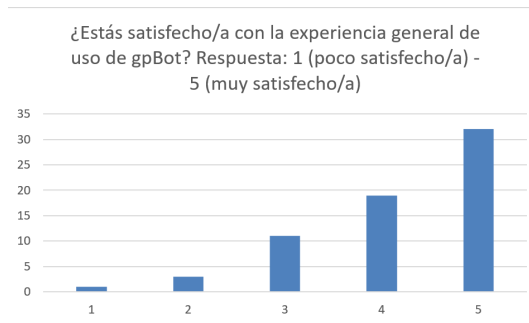


Figura 6.19: Respuestas obtenidas para la pregunta P1 tras la realización de la prueba de usuario del sistema.

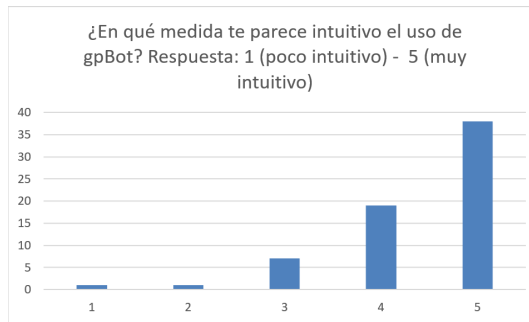


Figura 6.20: Respuestas obtenidas para la pregunta P2 tras la realización de la prueba de usuario del sistema.

El interés principal del uso de gpBot reside en conseguir motivar la participación del alumnado mediante técnicas de gamificación. En la Figura 6.22 se representan las respuestas obtenidas para la pregunta P4, las cuales muestran que una gran parte del alumnado considera que esta herramienta incentiva su participación. Por lo tanto, se puede concluir que se han logrado los objetivos que la gamificación persigue.

En cuanto a usabilidad, puede afirmarse que el objetivo de que el acceso a las encuestas sea sencillo ha sido satisfecho, puesto que en la Figura 6.23, que muestra los resultados para la pregunta P5, se observa que la mayoría del alumnado considera que este es sencillo. Asimismo, también se considera que la participación es fácil, ya que a la pregunta P6 la mayoría responde positivamente, tal como puede verse en la Figura 6.24.

Para conocer si la incorporación de la posibilidad de responder a las encuestas de forma anónima es importante para los participantes, se realizó la pregunta P7, cuyas respuestas están representadas en la Figura 6.25. En esta pregunta se observa una respuesta más dispar que en el resto, si bien es cierto que sí existe una porción del alumnado que valora positivamente la posibilidad de contestar anónimamente. Por pequeña que fuese la cantidad de

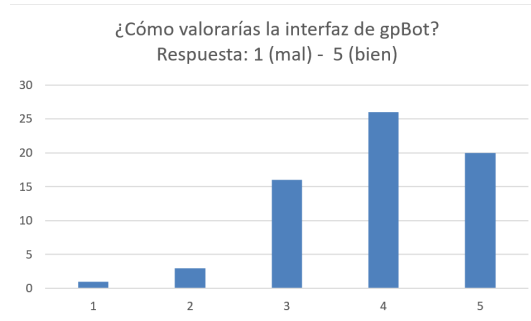


Figura 6.21: Respuestas obtenidas para la pregunta P3 tras la realización de la prueba de usuario del sistema.

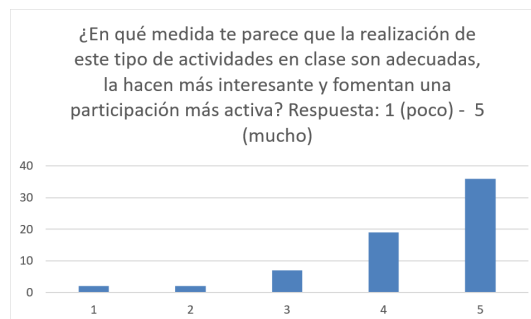


Figura 6.22: Respuestas obtenidas para la pregunta P4 tras la realización de la prueba de usuario del sistema.

usuarios que considera importante esta opción, la importancia de esta posibilidad reside en la intención de incentivar la participación de aquellos que por timidez o vergüenza, no existiendo esta opción, no se atreverían a contestar, con el fin de maximizar la participación en la encuesta. Por lo tanto, si a una sola persona le sirve disponer de esta funcionalidad, quedaría justificado su desarrollo y mantenimiento y, en este caso, las respuestas muestran que es útil para un número considerable de alumnos/as.

Dado que, además, se desea corroborar el motivo por el que (en su caso) se considera importante la posibilidad de responder de forma anónima, se añadió la pregunta P7-2. Tras la lectura de todas las justificaciones, se observan tres respuestas habituales: 1) “por error”; 2) “simplemente fue el primer modo de respuesta que se mostraba en la encuesta”; y 3) “no quería que el resto de participantes pudieran conocer mis resultados”. Si bien se estudiará la posibilidad de realizar algún retoque estético para evitar los casos 1) y 2), la cantidad de alumnos que mostraron respuestas en la línea de la tercera razón fue significativa, por lo que se concluye que la incorporación de esta funcionalidad permite que la competición gamificada que este modo de encuesta plantea no vaya en detrimento de la participación si no se desea formar parte de la competición.

Finalmente, en la Figura 6.26 se muestran las respuestas obtenidas a la pregunta P8. Pese

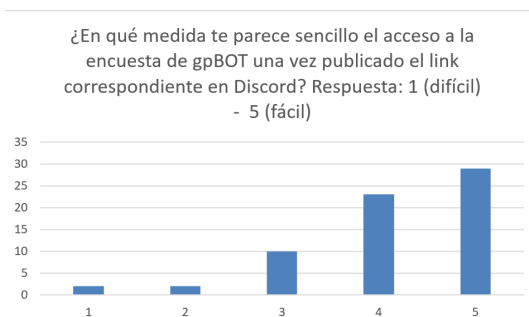


Figura 6.23: Respuestas obtenidas para la pregunta P5 tras la realización de la prueba de usuario del sistema.

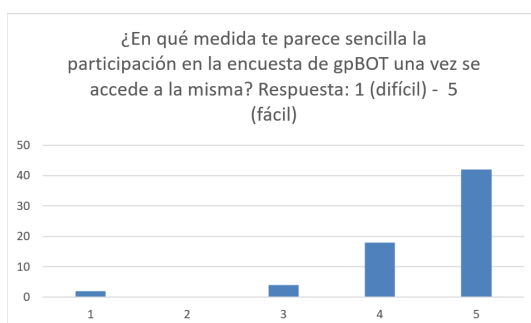


Figura 6.24: Respuestas obtenidas para la pregunta P6 tras la realización de la prueba de usuario del sistema.

a que la mayoría del alumnado considera la presentación de la evaluación de las respuestas a la encuesta adecuada, se observa que el máximo se desplaza en esta ocasión a la puntuación 4, en lugar de situarse en el 5. Gracias a estas opiniones se ha revisado el sistema y se han planteado líneas de trabajo futuras para mejorar la presentación de resultados.

Como valoración general de los resultados de las pruebas de usuarios realizadas, se concluye que el servicio cumple con los requisitos de usabilidad, funcionalidades y propósitos establecidos, así como que la experiencia de uso es satisfactoria tanto para profesores/as como para alumnos/as.

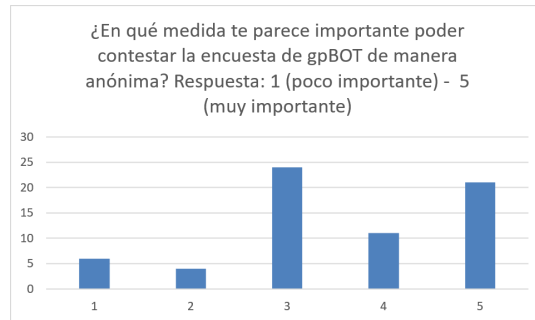


Figura 6.25: Respuestas obtenidas para la pregunta P7 tras la realización de la prueba de usuario del sistema.

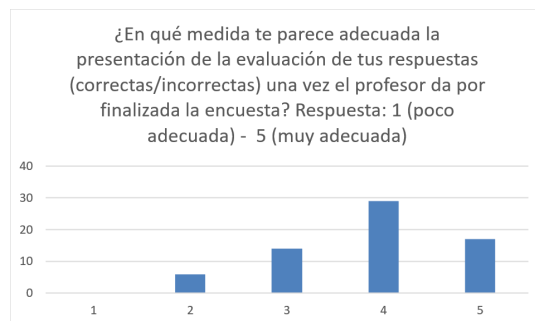


Figura 6.26: Respuestas obtenidas para la pregunta P8 tras la realización de la prueba de usuario del sistema.

Capítulo 7

Conclusiones

7.1. Conclusiones

En primer lugar, cabe destacar que se han alcanzado los objetivos fijados al inicio del proyecto. El diseño, desarrollo y prueba del sistema implementado en el contexto de este TFG ha resultado satisfactorio tanto a nivel de resultados obtenidos como en lo que a objetivos cumplidos respecta.

El resultado principal de este TFG es que se ha desarrollado el bot gpBot y el sistema que permite enlazarlo con un determinado canal de Discord y, a su vez, con la API para la gestión y respuesta de encuestas, alojada en una máquina virtual. Para ello, se ha desarrollado un sistema que consta de un cliente web Angular funcional (implementado usando Node.js) que se comunica con una API de gestión de encuestas desarrollada como micro-servicio del backend, donde se gestiona la persistencia del sistema. La persistencia se ha desarrollado sobre una base de datos MySQL a través de la herramienta de mapeo objeto-relacional (ORM) Hibernate, que implementa el estándar JPA. En cuanto a la monitorización de servidores de Discord, se ha desarrollado un micro-servicio que la lleva a cabo usando la biblioteca *Discord4J*. Todo el software se ha alojado en una máquina virtual de Inf-UVa y se ha desplegado de forma efectiva permitiendo usar el sistema de manera efectiva.

Se han realizado pruebas de uso en entornos reales; concretamente, en tres clases de teoría, con tres grupos diferentes de alumnos, de la asignatura de Fundamentos de Computadoras, de primer curso del Grado en Ingeniería Informática. Este escenario es exactamente el contexto para el que se pensó originalmente gpBot: sesiones de clase en las que se plantea la resolución de una encuesta.

Las pruebas de uso realizadas han demostrado (a través de una encuesta realizada a los alumnos tras responder una encuesta usando gpBot) que actualmente el sistema es intuitivo y fácil de usar, así como que resulta atractivo y motivador para el alumnado. Esto confirma la consecución del objetivo planteado referente a gamificación de encuestas en el aula. Además, han evidenciado que gpBot funciona correctamente con independencia del dispositivo (PC,

smarphone o tablet) utilizado para responder a la encuesta. Asimismo, una parte de los alumnos han valorado positivamente la posibilidad de responder de forma anónima, lo cual indica que la hipótesis de partida de que algunos/as alumnos/as prefieren no participar por timidez o vergüenza es cierta y que, posibilitando el anonimato, se puede lograr una mayor tasa de participación en la encuesta.

Si bien durante el desarrollo del servicio se manifestaron algunos de los riesgos que se previeron, estos se pudieron solventar gracias a la planificación planteada. Ha quedado patente que esta es una parte fundamental del desarrollo de software, especialmente importante en lo que a gestión de riesgos se refiere. Del mismo modo, gracias a las pruebas de uso realizadas con algunos profesores que utilizan un canal de Discord como vía habitual de comunicación en sus asignaturas, se pusieron de manifiesto algunas carencias o defectos del sistema que se corrigieron debidamente.

7.2. Valoraciones personales

Este proyecto ha sido una oportunidad para mejorar mis conocimientos sobre desarrollo web basado en componentes y micro-servicios. Además, me ha permitido conocer más sobre Discord, una herramienta de la que soy usuario habitual, pero para la que nunca había desarrollado ningún componente. Considero que el resultado de este proyecto es funcional, si bien podría continuarse su desarrollo para así incorporar más estrategias de gamificación. Conociendo el resultado que el uso de este proyecto ha dado en el aula, considero que tiene un gran potencial y que sí aumentará la participación del alumnado en la realización cuestionarios y encuestas, que es el objetivo principal de esta propuesta.

7.3. Líneas de trabajo futuras

Tanto la pruebas de usuario al alumnado como las del profesorado, así como las valoraciones de mis tutores y las mías propias han dado lugar al planteamiento de funcionalidades nuevas que podrían ser integradas en el sistema:

- Integrar las funcionalidades necesarias para que el administrador gestione las credenciales del bot para poder instalar el bot en otros servidores.
- Incluir la posibilidad de modificar las credenciales de login como administrador.
- Revisar ciertas partes del diseño del cliente web, basándonos en el feedback recibido de los alumnos, para hacer más intuitivo su uso. Por ejemplo, la forma de seleccionar si una contestación que realiza un usuario es anónima o no, ya que un número significativo de usuarios (alumnos/as) han indicado que se equivocaron en su decisión.
- Añadir el uso de la imagen (avatar) de Discord en los rankings para hacerlos más atractivos.

- Incorporar como estrategia de gamificación la recompensa mediante ventajas para la contestación de encuestas. Por ejemplo, podrían obtenerse comodines que, al ser activados, sirvan para descartar una de las opciones incorrectas.
- Permitir la gestión de mensajes de los canales de los servidores de Discord mediante comandos, haciendo que gpBot sea un “Profesor” más que pueda ser controlado por cualquier “Profesor”.
- Como se comentó en el capítulo 1, el uso de Discord pretende fomentar la colaboración entre los alumnos premiando la participación y colaboración entre el alumnado. Se quiere que el gpBot también monitorice la participación, interacciones, etc., de los alumnos, para que la aplicación lleve un recuento estadístico y los profesores puedan cuantificar fácilmente esta recompensa.

Bibliografía

- [1] Francisco J. Andújar, Arturo González-Escribano, Javier Bastida, and Yuri Torres de la Sierra. Aplicación de gamificación competitiva y colaborativa en asignaturas básicas de arquitectura de computadoras. In *Actas de las XX Jornadas sobre Enseñanza Universitaria de la Informática (JENUI 2020)*, volume 5, pages 85–92, Junio 2020.
- [2] Apache Software Foundation. Página web de Tomcat. Disponible en: <https://tomcat.apache.org/>. Accedido el 12 de Mayo de 2022.
- [3] S. Arnab, T. Lim, M. B. Carvalho, F. Bellotti, S. Freitas, S. Louchart, and et al. Mapping learning and game mechanics for serious games analysis. *British Journal of Educational Technology*, 46(2):391–411, 2014.
- [4] I. Jacobson G. Booch and J. Rumbaugh. *El proceso unificado de desarrollo de software*. Pearson Educación, 2000.
- [5] Jonasson C. *Gamification in Education: Breakthroughs in Research and Practice*, chapter 24, Vocational Learning Mediated by Constructive Competition, pages 453–470. IGI Global, 2018.
- [6] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. From game design elements to gamefulness: defining gamification. In *15th International Academic MindTrek Conference: Envisioning Future Media Environments*, pages 9–15. ACM, 2011.
- [7] D. Dicheva, C. Dichev, G. Agre, and G. Angelova. Gamification in education: A systematic mapping study. *Journal of Educational Technology & Society*, 18(3):75–88, 2015.
- [8] D. Dicheva, K. Irwin, C. Dichev, and S. Talasila. A course gamification platform supporting student motivation and engagement. In *International Conference on Web and Open Access to Learning (ICWOAL)*, pages 1–4. IEEE, 2014.
- [9] Discord Inc. Página web de Discord. Disponible en: <https://discord.com/developers/docs/reference>. Accedido el 12 de Mayo de 2022.
- [10] Discord Inc. Página web de Discord para desarrolladores. Disponible en: <https://discord.com/developers/docs/intro>. Accedido el 15 de Mayo de 2022.
- [11] Discord4J. Página web de Discord4J. Disponible en: <https://discord4j.com/>. Accedido el 12 de Mayo de 2022.

- [12] Roy Thomas Fielding and Richard N. Taylor. *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.
- [13] J. Fresno, A. Ortega-Arranz, H. Ortega-Arranz, A. Gonzalez-Escribano, and D.R. Llanos. *Gamification-Based E-Learning Strategies for Computer Programming Education*, chapter 6. Applying Gamification in a Parallel Programming Course. IGI Global, 2017.
- [14] A. Gonzalez-Escribano. Generalizando la integración de gamificación competitiva y colaborativa de forma ágil (iii). In *2016 PID-065/Generalizando la integración de gamificación competitiva y colaborativa de forma ágil (III)*. Universidad de Valladolid, 2016.
- [15] A. Gonzalez-Escribano, V. Lara-Mongil, E. Rodriguez-Gutierrez, and Y. Torres. Toward improving collaborative behaviour during competitive programming assignments. In *2019 IEEE/ACM Workshop on Education for High-Performance Computing (EduHPC)*, pages 1–6. IEEE, 2019.
- [16] Google. Página web de Angular. Disponible en: <https://angular.io/>. Accedido el 12 de Mayo de 2022.
- [17] L. Hakulinen, T. Auvinen, and A. Korhonen. Empirical study on the effect of achievement badges in trakla2 online learning environment. In *Learning and Teaching in Computing and Engineering (LaTiCE)*, pages 47–54. IEEE, 2013.
- [18] Project Management Institute. *A Guide to the Project Management Body of Knowledge*. Project Management Institute, 4 edition, 2008.
- [19] Kahoot! Página web de Kahoot! Disponible en: <https://kahoot.com/es/>. Accedido el 12 de Mayo de 2022.
- [20] Sailer M. and Homner L. The gamification of learning: a meta-analysis. *Educational Psychology Review, Springer*, (first online):1–36, 2019.
- [21] Moodle Partners. Página web de Moodle. Disponible en: <https://moodle.org/>. Accedido el 12 de Mayo de 2022.
- [22] C. I. Muntean. Raising engagement in e-learning through gamification. In *6th International Conference on Virtual Learning ICVL*, pages 323–329, 2011.
- [23] OpenJS Foundation. Página web de Node.js. Disponible en: <https://nodejs.org/>. Accedido el 12 de Mayo de 2022.
- [24] Oracle Corporation. Página web de MySQL. Disponible en: <https://www.mysql.com/>. Accedido el 12 de Mayo de 2022.
- [25] Oracle Corporation. Página web de Oracle. Disponible en: <https://www.oracle.com/java/technologies/>. Accedido el 12 de Mayo de 2022.
- [26] Combéfis S., Beresnevičius G., and Dagienė V. Learning programming through games and contests: Overview, characterisation and discussion. *Olympiads in Informatics*, 10:39–60, 2016.
- [27] Showbie Inc. Página web de Socrative. Disponible en: <https://www.socrative.com/plans/>. Accedido el 12 de Mayo de 2022.

- [28] Grupo Trasgo. Tablon. Página web, (última modificación mayo 2020). en <http://trasgo.infor.uva.es/tablon/>.
- [29] Twitter, Inc. Página web de Bootstrap. Disponible en: <https://getbootstrap.com/>. Accedido el 12 de Mayo de 2022.
- [30] Universidad de Valladolid. Proyecto docente del trabajo de fin de grado 2019-2020 (Mención Ingeniería de Software). https://alojamientos.uva.es/guia_docente/uploads/2019/545/46976/1/Documento.pdf. Accessed: 2019-7-6.
- [31] Ruth S. Contreras Espinosa y Jose Luis Eguia. Gamificación en aulas universitarias. page 130. Instituto de comunicación, Universidad Autónoma de Barcelona, 2016.

Apéndice A

Manuales

A.1. Manual de despliegue e instalación

Para la instalación y despliegue del sistema:

1. Clonar el repositorio <https://gitlab.inf.uva.es/josborg/tfg.git> en la máquina en la que se desea a desplegar el sistema.
2. Comprobar que el puerto 80 de la máquina no está en uso mediante el comando “*sudo lsof -i -P -n — grep “:80 (LISTEN)”*”. Si el comando no devuelve nada, significará que no está en uso. En la Figura A.1 se muestra la salida del comando en una máquina en la que el puerto 80 sí está en uso.

```
usuario@tfg-6510:~$ sudo lsof -i -P -n | grep ":80 (LISTEN)"
docker-pr  746633      root    4u    IPv4  57328037   0t0  TCP *:80 (LISTEN)
docker-pr  746643      root    4u    IPv6  57329458   0t0  TCP *:80 (LISTEN)
usuario@tfg-6510:~$
```

Figura A.1: Salida del comando en una máquina en la que el puerto 80 sí esta en uso.

3. Ir a al directorio raíz del repositorio clonado y ejecutar el comando “*docker-compose up*”. Esto iniciará el proceso de instalación y despliegue, es necesario que la máquina donde se instalará el sistema disponga de acceso a Internet, ya que este comando iniciará la descarga de los paquetes necesarios para el sistema. La primera vez que se ejecute este comando el proceso puede tardar bastante tiempo, alrededor de 15 o 30 minutos.
4. Comprobar que el despliegue ha sido efectuado a través del comando “*sudo docker stats*”. Si el sistema está desplegado correctamente, deberá aparecer en pantalla un resultado similar al mostrado en la Figura A.2.

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
e825346d720b	tfg_api-poll_1	0.28%	407.3MiB / 3.845GiB	10.34%	98.6MB / 71.1MB	11.3MB / 0B	37
bae116cf22e9	tfg_client_1	0.00%	5.012MiB / 3.845GiB	0.13%	3.0MB / 356MB	0.19kB / 9.72kB	5
2858fc548b33	tfg_poll-database_1	0.52%	377.4MiB / 3.845GiB	9.58%	65.8MB / 97.2MB	10.2MB / 166MB	50
3f66777c10e9	tfg_gateway_1	0.32%	298.7MiB / 3.845GiB	7.59%	374MB / 373MB	4.55MB / 0B	35
2314906665d1	tfg_api-discord_1	0.31%	441.2MiB / 3.845GiB	11.21%	26MB / 20.2MB	13.3MB / 0B	36

Figura A.2: Estadísticas de Docker.

5. Ir al sitio web desplegado.
6. Autenticarse con una cuenta Discord en el sistema.
7. Autenticarse como administrador en el sistema, lo cual será necesario siempre que se inicie un despliegue para activar al bot de Discord.
8. En la pantalla de administración del sitio web, añadir el bot a los servidores Discord deseados.

A modo de nota, en caso de que el sistema tenga un fallo en ejecución, ejecutar en la máquina donde se aloja el sistema el comando “*docker-compose down*”, después “*docker-compose up*” y finalmente, en el sitio web, autenticarse como administrador en el sistema para activar al bot de Discord. En caso de que siga fallando es posible que sea necesario reinstalar el sistema.

Apéndice B

Resumen de enlaces adicionales

Los enlaces útiles de interés en este Trabajo Fin de Grado son:

- Repositorio del código: <https://gitlab.inf.uva.es/josborg/tfg.git>.
- Url del sistema desplegado: <http://virtual.infor.uva.es:65102/client>
- Url del Discord: <https://discord.com/>