



**ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ
MANUEL FÉLIX LÓPEZ**

CARRERA DE COMPUTACIÓN

**INFORME DE TRABAJO DE INTEGRACIÓN CURRICULAR
PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
CIENCIAS DE LA COMPUTACIÓN**

**MECANISMO: SISTEMATIZACIÓN DE EXPERIENCIAS PRÁCTICAS
DE INVESTIGACIÓN Y/O INTERVENCIÓN**

TEMA:

**BACKEND DE APLICACIÓN MULTIPLATAFORMA DE
APRENDIZAJE PERSONALIZADO PARA LOS ESTUDIANTES
DE LA ESPAM MFL**

AUTORES:

**DANY ROLANDO ÁLAVA SANTANA
CARLOS ALBERTO SAAVEDRA ESCALANTE**

TUTOR:

MGTR. JAVIER HERNÁN LÓPEZ ZAMBRANO

CALCETA, FEBRERO DE 2023

DECLARACIÓN DE AUTORÍA

Nosotros, Dany Rolando Álava Santana, con cédula de ciudadanía 1317422291 y Carlos Alberto Saavedra Escalante, con cédula de ciudadanía 1316330685, declaramos bajo juramento que el Trabajo de Integración Curricular titulado: **BACKEND DE APLICACIÓN MULTIPLATAFORMA DE APRENDIZAJE PERSONALIZADO PARA LOS ESTUDIANTES DE LA ESPAM MFL** es de nuestra autoría, que no ha sido previamente presentado para ningún grado o calificación profesional, y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, concedemos a favor de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos, conservando a nuestro favor todos los derechos patrimoniales de autor sobre la obra, en conformidad con el Artículo 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación.

DANY ROLANDO ÁLAVA SANTANA
CC: 1317422291

CARLOS ALBERTO SAAVEDRA ESCALANTE
CC: 1316330685

AUTORIZACIÓN DE PUBLICACIÓN

Dany Rolando Álava Santana, con cédula de ciudadanía 1317422291 y Carlos Alberto Saavedra Escalante, con cédula de ciudadanía 1316330685, autorizamos a la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, la publicación en la biblioteca de la institución del Trabajo de Integración Curricular titulado: BACKEND DE APLICACIÓN MULTIPLATAFORMA DE APRENDIZAJE PERSONALIZADO PARA LOS ESTUDIANTES DE LA ESPAM MFL, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y total autoría.

DANY ROLANDO ÁLAVA SANTANA
CC: 1317422291

CARLOS ALBERTO SAAVEDRA ESCALANTE
CC: 1316330685

CERTIFICACIÓN DEL TUTOR

Javier Hernán López Zambrano, certifica haber tutelado el Trabajo de Integración Curricular titulado: **MULTIPLATAFORMA DE APRENDIZAJE PERSONALIZADO PARA LOS ESTUDIANTES DE LA ESPAM MFL**, que ha sido desarrollado por Dany Rolando Álava Santana y Carlos Alberto Saavedra Escalante, previo a la obtención del título de Ingeniero en ciencias de la computación, de acuerdo al **REGLAMENTO DE LA UNIDAD DE INTEGRACIÓN CURRICULAR DE CARRERAS DE GRADO** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

MGTR. JAVIER HERNÁN LÓPEZ ZAMBRANO
CC: 1308755238
TUTOR

APROBACIÓN DEL TRIBUNAL

Los suscritos integrantes del Tribunal correspondiente, declaramos que hemos **APROBADO** el Trabajo de Integración Curricular titulado: **BACKEND DE APLICACIÓN MULTIPLATAFORMA DE APRENDIZAJE PERSONALIZADO PARA LOS ESTUDIANTES DE LA ESPAM MFL**, que ha sido desarrollado por Dany Rolando Álava Santana y Carlos Alberto Saavedra Escalante, previo a la obtención del título de Ingeniero en ciencias de la computación, de acuerdo al **REGLAMENTO DE LA UNIDAD DE INTEGRACIÓN CURRICULAR DE CARRERAS DE GRADO** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

MGTR. DANIEL A. MERA MARTINEZ
CC: 1301932156
PRESIDENTE DEL TRIBUNAL

MGTR. FERNANDO MOREIRA MOREIRA
CC: 1311726689
MIEMBRO DEL TRIBUNAL

MGTR. RICARDO VELEZ VALAREZO
CC: 1316330685
MIEMBRO DEL TRIBUNAL

AGRADECIMIENTO

En primer lugar, agradecemos a los docentes que formaron parte de esta experiencia como estudiante universitario, los cuales plasmaron una huella tanto de conocimiento como de valores personales. También, es necesario hacer una mención especial a todos los actores involucrados en el desarrollo de este proyecto: al ingeniero Joffre Moreira y al ingeniero Fernando Moreira, quienes propusieron la idea base del mismo, y estuvieron presentes en todo el proceso desde el primer hasta el último día de nuestra instancia dentro de este proyecto. De la misma forma, al ingeniero Javier López, nuestro tutor, quien estuvo pendiente para que el desarrollo del presente trabajo fuera realizado correctamente. Por último, pero no menos importante, a nuestros compañeros del Trabajo de Integración Curricular complementario, Ariel Tuárez y Geovanny Zambrano, quienes literalmente, hicieron posible que este trabajo fuera un proyecto completo y con sentido.

LOS AUTORES

DEDICATORIA

A mis abuelos/as, a ustedes dedico esto y mucho más, son mi ejemplo a seguir, gracias por siempre estar a mi lado apoyándome en las buenas y en las malas y no dejarme solo, por ustedes nunca dejaré de luchar.

A mi mamá Carla, por creer en mí desde el día uno y ser ese apoyo incondicional en mi vida, por ser una madre y mujer excepcional y un gran ejemplo de superación.

A mi papá Rolando, gracias por corregirme siempre, que a pesar de no vivir juntos desde que era un niño nunca has dejado de amarme y de creer en mí.

A mis hermanos, gracias por su cariño y apoyo incondicional, en especial a mi hermana Karla, gracias por siempre aconsejarme y estar pendiente de mí como si de una madre se tratara.

A mi pareja y compañera de vida Trinidad, por estar conmigo en cada etapa de mi formación profesional y compartir conmigo cada experiencia vivida, por su apoyo incondicional y alentarme siempre a seguir adelante.

A mi mascota Lila que, pese a que no te encuentres a mi lado, empecé esto cuando sí lo estabas y te mereces formar parte de ello.

A mi compañero de Trabajo de Integración Curricular Carlos, por cada consejo que me ayudó en mi formación académica, gracias por creer en mí y por el apoyo brindado.

DANY ROLANDO ÁLAVA SANTANA

DEDICATORIA

El presente Trabajo de Integración Curricular, se la dedico a mi familia, en especial a mis padres, hermanos/as y sobrinos/as, quienes desde el comienzo fueron el apoyo primordial y la razón de ser de todo este proceso, y de la vida en general. Asimismo, a mis amigos y compañeros, los cuales sirvieron como pilar fundamental para la correcta finalización de este proyecto, tanto directa como indirectamente. A mi compañero de Trabajo de Integración Curricular, y gran amigo desde la infancia, Dany Álava, quien a través de su apoyo tanto moral como en el plano de desarrollador, contribuyó a que este proyecto concluyera satisfactoriamente. Por último, dedico este trabajo a mis mascotas (Coby y Mía), que, aunque no sepan leer, quiero que quede plasmado lo que representan para mí.

CARLOS ALBERTO SAAVEDRA ESCALANTE

CONTENIDO GENERAL

CARÁTULA.....	i
DECLARACIÓN DE AUTORÍA.....	ii
AUTORIZACIÓN DE PUBLICACIÓN.....	iii
CERTIFICACIÓN DEL TUTOR.....	iv
APROBACIÓN DEL TRIBUNAL.....	v
AGRADECIMIENTO.....	vi
DEDICATORIA.....	vii
DEDICATORIA.....	viii
CONTENIDO GENERAL.....	ix
CONTENIDO DE FIGURAS.....	x
CONTENIDO DE TABLAS.....	xiii
RESUMEN.....	xiv
PALABRAS CLAVE.....	xiv
ABSTRACT.....	xv
KEY WORDS.....	xv
CAPÍTULO I. ANTECEDENTES.....	1
1.1. DESCRIPCIÓN DE LA INSTITUCIÓN.....	1
1.2. DESCRIPCIÓN DE LA INTERVENCIÓN.....	3
1.3. OBJETIVOS.....	5
1.3.1. OBJETIVO GENERAL.....	5
1.3.2. OBJETIVOS ESPECÍFICOS.....	5
CAPÍTULO II. DESARROLLO METODOLÓGICO DE LA INTERVENCIÓN.....	6
2.1. TÉCNICAS.....	6
2.1.1. ENTREVISTA.....	6

2.2.	MÉTODOS	6
2.2.1.	PROGRAMACIÓN EXTREMA (XP)	6
CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA		9
3.1.	PLANIFICACIÓN.....	9
3.2.	DISEÑO.....	12
3.3.	DESARROLLO.....	15
3.3.1.	MÉTODO POST	16
3.3.2.	MÉTODO GET.....	16
3.3.3.	MÉTODO DELETE	17
3.3.4.	VISTAS DE LA GESTIÓN DOCENTE	17
3.3.5.	MÓDULO DE APRENDIZAJE PERSONALIZADO.....	19
3.4.	PRUEBAS	21
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES		23
4.1.	CONCLUSIONES.....	23
4.2.	RECOMENDACIONES	25
BIBLIOGRAFÍA		26
ANEXOS		29

CONTENIDO DE FIGURAS

Figura 1.	Diagrama de casos de uso del Docente.....	13
Figura 2.	Diagrama de casos de uso del Estudiante	13
Figura 3.	Diagrama de clases.....	14
Figura 4.	Diagrama de modelo de base de datos.....	15
Figura 5.	Petición: Crear nueva clase	16
Figura 6.	Petición: Obtener clases.....	16
Figura 7.	Petición: Expulsar a un estudiante de una clase	17
Figura 8.	Vista: Crear clase	17
Figura 9.	Vista: Clases	18

Figura 10. Vista: Detalles de una clase	18
Figura 11. Diagrama de flujo del proceso de agregación de relación entre preguntas	20
Figura 12. Diagrama de flujo del controlador de Aprendizaje personalizado...	21
Figura 13. Muestra de la entidad Student (Estudiante)	85
Figura 14. Agregación y definición del servicio DbContext.....	86
Figura 15. Fragmento de la clase derivada de DbContext	86
Figura 16. Consulta de lectura.....	87
Figura 17. Consulta de escritura.....	87
Figura 18. Fragmento de una migración.....	87
Figura 19. Diagrama de base de datos	88
Figura 20. Endpoint para añadir una clase	93
Figura 21. Endpoint para unirse a una clase	93
Figura 22. Endpoint para agregar una pregunta.....	94
Figura 23. Endpoint para agregar un test	95
Figura 24. Endpoint para asignar la calificación de un test	95
Figura 25. Endpoint para iniciar sesión	96
Figura 26. Endpoint para obtener las clases	96
Figura 27. Endpoint para obtener estudiantes de una clase	97
Figura 28. Endpoint para obtener las preguntas de una asignatura	97
Figura 29. Endpoint para obtener las calificaciones de un test.....	98
Figura 30. Endpoint para expulsar estudiante de una clase	98
Figura 31. Endpoint para eliminar un test.....	99
Figura 32. Endpoint para editar una clase	99
Figura 33. Endpoint para editar un perfil	100
Figura 34. Endpoint para editar una pregunta	100
Figura 35. Vista del login	101
Figura 36. Vista del home.....	101
Figura 37. Vista del módulo de crear clase	102
Figura 38. Vista del módulo de Clases	102
Figura 39. Vista para administrar una clase	103
Figura 40. Vista del módulo de editar clase.....	103
Figura 41. Vista del módulo de crear test	104

Figura 42. Agregar una nueva pregunta	104
Figura 43. Agregar pregunta existente	105
Figura 44. Eliminar pregunta agregada	105
Figura 45. Vista del módulo de Mis Tests.....	106
Figura 46. Vista del módulo de administración de un test	106
Figura 47. Vista del módulo de perfil de usuario.....	107
Figura 48. Vista del módulo de Mis preguntas.....	107
Figura 49. Instalación/importación de librerías y cuerpos de preguntas	117
Figura 50. Carga de los modelos.....	117
Figura 51. Cálculo de la similitud coseno y tiempo de ejecución.....	118
Figura 52. Cálculo del puntaje final entre modelos.....	118
Figura 53. Diagrama de flujo del proceso de agregación entre preguntas	125
Figura 54. Agregar nueva pregunta al test	126
Figura 55. Agregar nuevo test a la clase	126
Figura 56. Calcular relación entre preguntas.....	127
Figura 57. Relaciones encontradas, guardadas en la tabla correspondiente	127
Figura 58. Diagrama de flujo del controlador de Aprendizaje personalizado.	128
Figura 59. Endpoint automático de agregación de notas a un test.....	129
Figura 60. Preguntas respondidas erróneamente por un estudiante.....	129
Figura 61. Respuesta de la ejecución del endpoint de obtención de preguntas relaciones a los fallos del estudiante	131
Figura 62. Estructura del proyecto "API"	148
Figura 63. Contenido del directorio "API/Controllers"	149
Figura 64. Contenido del directorio "API/Data"	149
Figura 65. Contenido del directorio "API/Data/SeedData"	150
Figura 66. Contenido del directorio "API/DTOs"	150
Figura 67. Contenido del directorio "API/Entities"	151
Figura 68. Contenido del directorio "API/Errors"	151
Figura 69. Contenido del directorio "API/Extensions"	151
Figura 70. Contenido del directorio "API/Helpers"	152
Figura 71. Contenido del directorio "API/Interfaces"	152
Figura 72. Contenido del directorio "API/Middleware"	153
Figura 73. Contenido del directorio "API/Migrations"	153

Figura 74. Contenido del directorio "API/Properties"	153
Figura 75. Contenido del directorio "API/Services"	153
Figura 76. Diagrama de clases generado por Visual Studio	154
Figura 77. Contenido del directorio "TeacherClient/src"	155
Figura 78. Contenido del directorio "TeacherClient/ssl"	155
Figura 79. Contenido del directorio "TeacherClient/components"	156
Figura 80. Contenido del archivo "auth.guard.ts"	156
Figura 81. Contenido del archivo "jwt.interceptor.ts"	157
Figura 82. Estructura del modelo "user"	157
Figura 83. Fragmento del servicio "account"	158
Figura 84. Contenido del archivo "run.py"	159
Figura 85. Contenido del directorio "TransformersMicroService/app"	159
Figura 86. Contenido del directorio "TransformersMicroService/env"	159
Figura 87. Contenido del archivo "__init__.py"	159
Figura 88. Fragmento del archivo "routes.py"	160
Figura 89. Fragmento del archivo "routes.py"	160

CONTENIDO DE TABLAS

Tabla 1 Análisis de la entrevista	9
Tabla 2 Requisitos funcionales	11
Tabla 3 Requerimientos no funcionales	12
Tabla 4 Evaluación de los modelos puestos a prueba	19
Tabla 5 Componente unitario "Acceso a la plataforma"	21
Tabla 6 Muestra del cuerpo de preguntas utilizadas para las pruebas ...	116
Tabla 7 Evaluación final de los modelos	119

RESUMEN

El presente trabajo, tuvo como propósito el desarrollo del núcleo/backend de la aplicación multiplataforma “Gamivity”, la cual permita la gestión docente de actividades autónomas gamificadas, enfocadas hacia los estudiantes de la ESPAM MFL. Esta plataforma, tiene como eje fundamental la integración de dos componentes generales: la gamificación y el aprendizaje personalizado impulsado por inteligencia artificial. En lo que concierne a la experiencia personalizada, esta se implementó a través de una librería de Python de un modelo de tipo “sentence transformers”, la cual ofreció un rendimiento adecuado a las necesidades del sistema. Para la construcción del backend, se hizo uso de la metodología ágil “Programación Extrema”, la cual, a través de sus fases iterativas, permitió que el flujo de desarrollo se pudiera ejecutar satisfactoriamente. La primera fase llevada a cabo fue la planificación, en la cual se empleó la técnica de la entrevista para la toma de requerimientos; posteriormente, gracias a los diagramas UML empleados, se logró modelar el sistema correspondiente a la fase de diseño. Consecuentemente, se procedió con el desarrollo del núcleo del mismo, haciendo uso del framework “Microsoft Dot Net 6”. En la última fase, se procedió a realizar las respectivas pruebas para garantizar la integridad de los datos y un funcionamiento óptimo del sistema. El producto final permitió tanto gestionar las actividades evaluativas de los estudiantes por parte de los docentes de la ESPAM MFL, así como ofrecer un núcleo capaz de satisfacer las peticiones de la aplicación móvil gamificada.

PALABRAS CLAVE

Gamivity, inteligencia artificial, gamificación, sentence transformers, Dot Net 6.

ABSTRACT

The purpose of this work was to develop the core/backend of the multiplatform application "Gamivity", which allows the teaching management of autonomous gamified activities, focused on the students at ESPAM MFL university. This platform has as its fundamental axis the integration of two general components: gamification and personalized learning driven by artificial intelligence. As regards the personalized experience, it was implemented through a Python library of a "sentence transformers" type model, which offered adequate performance to the needs of the system. For the construction of the backend, the agile methodology "Extreme Programming" was used, which, through its iterative phases, allowed the development flow to be executed satisfactorily. The first phase carried out was planning, in which the interview technique was used to gather requirements; later, thanks to the UML diagrams used, it was possible to model the system corresponding to the design phase. Consequently, it proceeded with the development of its core, making use of the "Microsoft Dot Net 6" framework. In the last phase, the respective tests were carried out to guarantee the integrity of the data and optimal operation of the system. The final product made it possible to manage the evaluation activities of the students by the ESPAM MFL teachers, as well as to offer a nucleus capable of satisfying the requests of the gamified mobile application.

KEY WORDS

Gamivity, artificial intelligence, gamification, sentence transformers, Dot Net 6.

CAPÍTULO I. ANTECEDENTES

1.1. DESCRIPCIÓN DE LA INSTITUCIÓN

La Escuela Superior Politécnica Agropecuaria de Manabí "Manuel Félix López", (ESPAM MFL, de ahora en adelante), es una de las principales unidades de educación superior pública de la zona norte de la provincia de Manabí, y se encuentra ubicada en la ciudad de Calceta del Cantón Bolívar (Universidades de Ecuador, 2022), la cual se constituyó como tal un 29 de abril de 1999, con la intención de participar junto a la compañía de otras instituciones para ayudar al progreso y desarrollo del país a través de la enseñanza universitaria y la investigación científica (ESPAM MFL, 2016).

La ESPAM MFL cuenta con la siguiente misión: “formar profesionales pertinentes con compromisos ético y social garantizando aquello desde la calidad de funciones sustantivas. Y su visión: “ser un centro de referencia en la capacitación de profesionales que contribuyan al desarrollo agropecuario regional” (ESPAM MFL, 2021). Llevar a cabo la misión y visión de la institución es una responsabilidad de todos los miembros de la comunidad politécnica, y lleva consigo una corresponsabilidad en la consecución de los objetivos institucionales.

Por otro lado, esta persigue el objetivo de “dotar a la provincia y al país de profesionales capacitados para transformar la realidad productiva del entorno, gente que maneje éticamente los recursos y facilite la innovación en función del desarrollo de la Matriz productiva” (ESPAM MFL, 2019).

Desde una perspectiva más local, la universidad cuenta con el área de vinculación con la comunidad, la cual se basa en la contribución a la formación profesional del estudiante en un acercamiento a situaciones reales acordes a su profesión (ESPAM MFL, 2020). Es por ello que cuenta con la misión de “desarrollar programas y proyectos con pertinencia, comprometidos con el mejoramiento de la calidad de vida de la colectividad, para contribuir al desarrollo sostenido e integral de la región, buscando alternativas de solución a los problemas o necesidades, y enmarcados en los lineamientos establecidos por la

ESPAM MFL, en concordancia con los objetivos del Buen Vivir”, asimismo la visión de “convertirse en un centro de referencia de transferencia de conocimientos y tecnologías, resultado de investigaciones de la ESPAM MFL, orientados a impactar el desarrollo en su zona de influencia promoviendo la equidad, el progreso cultural, social, humanista y tecnológico, basado en los planes del buen vivir” (Coordinación de Vinculación con la Comunidad, 2022).

Por su relación con las distintas carreras pertenecientes a la institución, el departamento de vinculación con la sociedad coopera continuamente con los estudiantes presentando y receptando proyectos afines a sus destrezas profesionales para su posterior desarrollo e implementación (Honorable Consejo Politécnico, 2015), en este caso, el proyecto involucra a estudiantes y docentes de la carrera de Computación, con el fin de formar un entorno multiplataforma de aprendizaje complementario que conecte a ambas partes directamente a través de la gamificación.

1.2. DESCRIPCIÓN DE LA INTERVENCIÓN

En un mundo altamente modernizado, la transición de métodos tradicionales a otros más acordes a las tecnologías que rigen en la presente era digital, lentamente se va convirtiendo en una necesidad para el incremento de la efectividad de los mismos. Dichos métodos pueden ser evidenciados en el ámbito de la educación, de donde nace un fuerte interés por investigar nuevos modelos y técnicas de aprendizaje que empleen formas de integrar a las tecnologías en el ámbito de la enseñanza (Hernandez, 2017) (Martinez, 2017).

Sin embargo, esta necesidad de integrar a la tecnología en el plano académico, no se produce de manera automática o descontextualizada, ya que como comenta Facer y Selwyn (2021), las tecnologías digitales solas, no transforman a la educación, ni mucho menos mejoran al proceso de aprendizaje y, por supuesto, tampoco libran al docente de su rol dentro de este sistema. Es por esta razón, que se deben de considerar varios aspectos específicos, como elementos afectivos, de comportamiento, características tecnológicas, etc. Ya que dichos elementos, parecen influir en el proceso de aprendizaje si se pretenden alcanzar los resultados positivos esperados en dicha integración (Lai & Bower, 2019).

Teniendo en cuenta lo antes descrito, aparece el término gamificación, el cual es un paradigma construido bajo la premisa de innovar, a través de la aplicación de mecánicas de juego a contextos ajenos a estos mismos, obteniendo así los beneficios que un sistema gamificado pueda ofrecer, los cuales se pueden resumir en el aumento de motivación y compromiso al realizar una actividad que normalmente no genera dichos efectos (Andrade, Lacerda, Calado, Lima, & Canedo, 2017) (AlSaad & Durugbo, 2021) (Saggah, Atkins, & Campion, 2020).

A partir de los conceptos previamente establecidos, surge la aparición de la gamificación en el plano académico. Esta estrategia tiene como idea base, la introducción de elementos simples de juegos, tales como sistema de puntuación, insignias y tablas de clasificación, los cuales pueden transformar una tarea que genera poco interés o que sea repetitiva, en una actividad atractiva que promueva a la realización de las mismas de una manera más orgánica por parte

de los estudiantes (Araya, Arias, Bottan, & Cristia, 2019) (Dichev & Dicheva, 2017).

Complementando a dicho fin antes expuesto, cabe destacar que el aprendizaje se construye a través de interacciones individuales, experiencias, así como de la transferencia de conocimiento por parte de terceros. En pocas palabras, se podría decir que el aprendizaje se trata de una experiencia personalizada, la cual permite al individuo expandir sus conocimientos y que, además, dicho enfoque adaptativo/personalizado se ha convertido en un paradigma de aprendizaje fundamental en la comunidad investigadora de tecnologías educativas (Shemshack & Spector, 2020) (Peng, Ma, & Spector, 2019).

Bajo dicho contexto de aprendizaje personalizado apoyado por la tecnología, más concretamente, en su relación con países de bajos y medianos ingresos, aparece una limitación muy pronunciada, la cual es que tanto conceptualmente como en la práctica, esta sigue siendo una temática poco estudiada, en la que queda mucho por explorar acerca de los posibles beneficios o contras que pueda tener dicho enfoque. Sin embargo, en términos generales, se reportan resultados positivos y, sobre todo, alentadores en cuanto a la implementación de este enfoque adaptativo en el ámbito académico en dichos países, entre los cuales se encuentra incluido Ecuador (Major & Francis, 2020).

Consecuentemente, la ESPAM MFL intenta adaptarse a los nuevos cambios tecnológicos haciendo uso de plataformas como el aula virtual “Google Classroom” para realizar la gestión de las actividades curriculares asignadas a los estudiantes de dicha universidad. Esto último, a partir de los beneficios que ofrece su uso, tales como la oportunidad que brinda a los estudiantes de escoger las temáticas de interés que los docentes hayan propuesto, así como de poder acceder a dichos contenidos en cualquier lugar y cualquier momento, entre muchos otros (Soodtoetong & Rattanasiriwongwut, 2022).

A partir de esta premisa y de los métodos de interacción docentes-alumnos que un aula virtual tradicional ofrece, surge la necesidad por parte del grupo de investigación de la institución de complementar su uso con el desarrollo de una plataforma web, la cual cumpla con algunas de las prestaciones de gestión de

un aula virtual con la integración de gamificación en las actividades de la misma y la cual brinde una experiencia de aprendizaje personalizada mediante el uso de inteligencia artificial. Permitiendo así, aprovechar los beneficios que otorga el impulsar el proceso de aprendizaje de la mano de dichos mecanismos.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar el Backend de una aplicación multiplataforma de aprendizaje personalizado que permita la gestión docente de actividades autónomas gamificadas enfocadas hacia los estudiantes de la ESPAM MFL.

1.3.2. OBJETIVOS ESPECÍFICOS

- Establecer los requerimientos del sistema con base en las necesidades de los estudiantes y docentes.
- Elaborar el Backend de la aplicación multiplataforma de acuerdo a los requerimientos establecidos.
- Integrar métodos de inteligencia artificial al sistema acorde al diseño del aprendizaje personalizado.
- Implementar el núcleo del sistema en los servidores de la ESPAM MFL.

CAPÍTULO II. DESARROLLO METODOLÓGICO DE LA INTERVENCIÓN

El presente trabajo de titulación tuvo como objetivo desarrollar el Backend de una aplicación multiplataforma de aprendizaje personalizado para los estudiantes de la ESPAM MFL. Para cumplir con el objetivo previamente mencionado, se hizo uso de técnicas como la entrevista, además del uso de metodologías de desarrollo ágil como la Programación Extrema (XP), con el fin de organizar el trabajo de la forma más eficiente posible.

2.1. TÉCNICAS

2.1.1. ENTREVISTA

Se empleó la técnica de entrevista con el fin de recopilar información acerca de los requisitos del sistema. Esto se realizó mediante la interacción e intercambio de ideas entre los docentes encargados del proyecto de vinculación del cual surgió inicialmente esta propuesta y los autores del presente documento.

Cabe recalcar, que la entrevista es una de las técnicas para la recolección de datos más utilizadas en la investigación cualitativa ya que permite la obtención de datos o información del sujeto de estudio mediante la interacción verbal directa con el investigador (Troncoso & Amaya, 2017). Es por esta razón, que se optó por el uso de dicha técnica como medio de recopilación de información.

2.2. MÉTODOS

2.2.1. PROGRAMACIÓN EXTREMA (XP)

Esta metodología fue empleada debido a la flexibilidad que otorga a lo largo del proceso de construcción de un software, así como la relación estrecha que se mantiene con el cliente durante el ciclo de vida del mismo. Además, debido a la programación en parejas que contempla esta metodología, la cantidad de errores que aparecen en la etapa de desarrollo es considerablemente menor.

Otro aspecto por el que se consideró hacer uso de la misma, fue por su filosofía de alentar a los desarrolladores a responder a los requerimientos cambiantes de

los clientes, aún en fases tardías del ciclo de vida del desarrollo. Esto último, debido a la filosofía ágil que maneja esta metodología, la cual provee de muchos medios para poder adaptarse a cualquier entorno de desarrollo cambiante.

- **FASE DE PLANIFICACIÓN**

La primera fase contemplada en esta metodología fue la de planificación, en la cual el equipo de desarrollo conformado por los autores del presente documento Dany Álava y Carlos Saavedra, junto a Ariel Tuárez y Geovanny Zambrano (autores del Trabajo de Integración Curricular complementario), convirtió los requerimientos descritos en iteraciones que cubrieron una pequeña parte de las características o funcionalidades requeridas. Estas fueron producto de las entrevistas realizadas a los respectivos docentes encargados del proyecto de vinculación, las cuales forman parte del documento de especificación de requerimientos IEEE 830, documento en el cual se puntualiza cada una de las funcionalidades del sistema. Cabe recalcar, que en esta etapa se definió la visión del sistema, en la que los desarrolladores estimaron y priorizaron las actividades asociadas al desarrollo del mismo.

- **FASE DE DISEÑO**

Esta fase se empleó con el objetivo de crear un diseño simple del sistema, a partir de la definición de las características principales del código futuro, sin tener que agregar funcionalidades extras no requeridas. La misma, se centró en la modelación del sistema, es decir, en la creación de diagramas de estructuración UML (caso de uso, de clase y de paquetes), que fueron modelados gracias a la herramienta online Lucidchart; y de base de datos, generado por SQL Server. Asimismo, se procedió con la respectiva definición de las propiedades de gestión docente y otros procesos de abstracción necesarios, previos a la codificación del sistema.

- **FASE DE DESARROLLO**

En esta fase fue donde se realizó la respectiva codificación de todo el sistema modelado, en cuanto a las herramientas y tecnologías utilizadas, se tuvo como entorno de desarrollo integrado (IDE) a Microsoft Visual Studio Code; por otro

lado, en lo que concierne a los lenguajes de programación utilizados, estos fueron C# para el desarrollo del backend del sistema, Typescript en el desarrollo de la plataforma web de gestión docente y Python para los módulos de inteligencia artificial. Respecto a los frameworks empleados, se tuvo a Microsoft Dot Net 6 para la construcción de la API del sistema y Angular para el desarrollo del frontend asociado a la gestión docente.

- **FASE DE PRUEBAS**

Una vez finalizado el proceso de desarrollo, se procedió a realizar pruebas al sistema, más específicamente, pruebas unitarias a los métodos asociados a los miembros del sistema (docentes y estudiantes). Dichas pruebas permitieron comprobar que el funcionamiento del sistema fuera el correcto y no existiera alguna incongruencia entre módulos, ya que, de ser el caso, se debían tomar las medidas respectivas, para así poder solventar los errores encontrados. Por último, se migró el sistema para establecer conexión con los servidores de la ESPAM MFL y realizar la respectiva entrega del manual de programador.

CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA

A partir de lo descrito en el desarrollo metodológico en el capítulo anterior, se procede a detallar cómo se llevó a cabo la intervención del proyecto bajo la metodología planteada.

3.1. PLANIFICACIÓN

Esta primera fase, tuvo lugar a partir de las entrevistas realizadas a los docentes encargados del proyecto de vinculación, la cual tuvo como finalidad, establecer los aspectos básicos del aplicativo, y, sobre todo, el alcance del mismo. En la Tabla 1, se muestra un análisis realizado con base en las preguntas más esenciales respecto a la definición de mínimo producto viable. Dichas preguntas y sus respectivas respuestas, así como otros detalles asociados a las entrevistas en cuestión, se pueden apreciar en el Anexo 1.

Tabla 1

Análisis de la entrevista

N	Pregunta	Análisis
1	¿Qué se va a desarrollar?	Una solución software multiplataforma de tipo web, la cual presente un entorno gamificado en la que tanto alumnos como docentes de la institución, puedan desarrollar ciertas actividades académicas de una manera más ágil y gratificante.
2	¿A qué problemática se pretende aportar una solución y cómo pretende hacerlo?	Debido a que el sistema educativo reside en mecanismos que no han variado significativamente a lo largo del tiempo. La solución software propuesta, pretende implementar formas innovadoras de involucrar tanto al docente como a su alumnado en un entorno virtual interactivo, tratando así de minimizar procesos monótonos que puedan aparecer en el camino.
3	¿Qué metodologías/paradigmas se deberían incluir en la solución?	Se pretende realizar la utilización de técnicas de “gamificación” como una particularidad del aplicativo, la cual ocasione sobre los estudiantes una

N	Pregunta	Análisis
		motivación el realizar tanto las actividades propuestas por los docentes, como fomentar el autoaprendizaje en los mismos.
4	¿Cuál va a ser la división de la carga de trabajo para el proyecto?	Carlos Saavedra y Dany Álava: desarrollo del Backend y de los módulos de gestión docente. Ariel Tuárez y Geovanny Zambrano: desarrollo del frontend del aplicativo (versión móvil y web).
5	¿Qué tecnologías conviene utilizar para el desarrollo de la solución?	Para el desarrollo del backend se pretende hacer uso de la tecnología “dot net” de Microsoft, mientras que para el desarrollo de la plataforma web de gestión, se utilizará “Angular” y “Flutter” para la aplicación gamificada.

Fuente: Los autores

Luego de la definición del alcance del proyecto gracias a las entrevistas realizadas, se procedió a desarrollar un análisis por medio de un informe, acerca de las funcionalidades de experiencia personalizada en aplicaciones del mercado que implementen gamificación en la misma, el cual se encuentra en los Anexos 2-A y 2-B. En dicho análisis se seleccionaron un total de 40 aplicaciones basándose en criterios como el número de descargas y la valoración de las mismas, así como en elementos de entornos gamificados, los cuales se aprecian en la Ilustración 1, junto al respectivo análisis cuantitativo basado en dichas características.

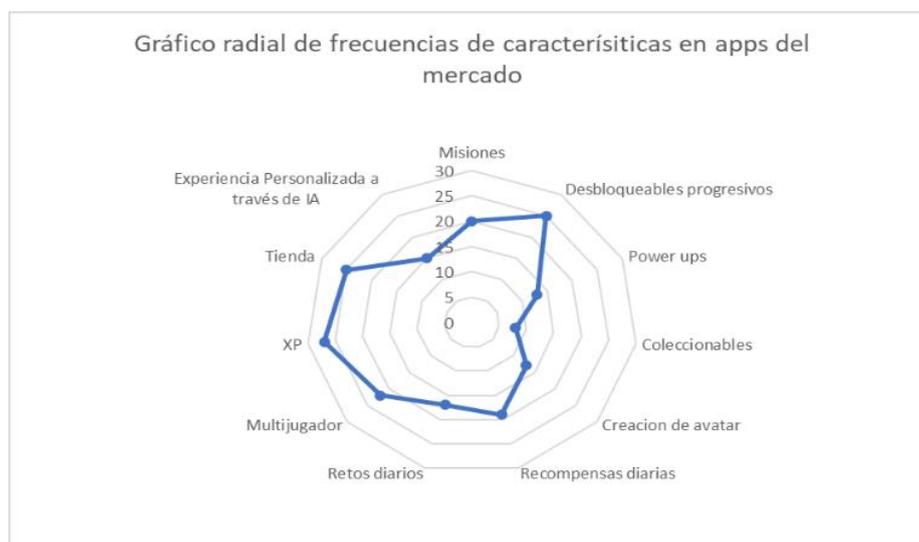


Ilustración 1. Características de las aplicaciones analizadas
Fuente: Los autores

Una vez realizado el análisis de las aplicaciones, se logró determinar la información necesaria para así, poder especificar los requerimientos del sistema y a su vez poder desarrollar el documento ERS, mismo que se puede encontrar en el Anexo 3. En este, se encuentran detallados los requisitos funcionales que se pueden evidenciar en la Tabla 2, así como los requisitos no funcionales evidenciados en la Tabla 3 y entre otras características y restricciones del sistema.

Tabla 2

Requisitos funcionales

Identificador	Nombre
RF001	Elementos de juego
RF001.1	Sistema de Puntuación
RF001.2	Tabla de posiciones (Leaderboards)
RF001.3	Insignias (Badgets)
RF001.4	Potenciadores (PowerUps)
RF001.5	Tienda
RF002	Estudiante
RF002.1	Editar perfil de usuario
RF002.2	Unirse a una clase

RF002.3	Realizar test
RF003	Repositorio general de preguntas
RF004	Docente
RF004.1	Creación de una clase
RF004.2	Creación de tests
RF004.3	Asignación de un test
RF004.4	Resultados de los test asignados
RF004.5	Gestionar clases
RF005	Experiencia Personalizada

Fuente: Los autores

Tabla 3

Requerimientos no funcionales

Nombre
Requerimientos de rendimiento
Restricciones de diseño
Estándares asociados
Atributos del sistema
Requerimientos de desarrollo

Fuente: Los autores

3.2. DISEÑO

Continuando con lo planificado, se procedió con el respectivo diseño de los diagramas UML utilizando la herramienta Lucidchart. En el Anexo 4 se exponen de manera más detallada cada uno de los diagramas realizados para la modelación del sistema.

Los primeros diagramas realizados fueron los de casos de uso, mismos que permitieron tener una visión general del funcionamiento del sistema y de cómo interactúan los dos diferentes tipos de usuarios que se identificaron, estos se muestran en las Figuras 1 y 2.

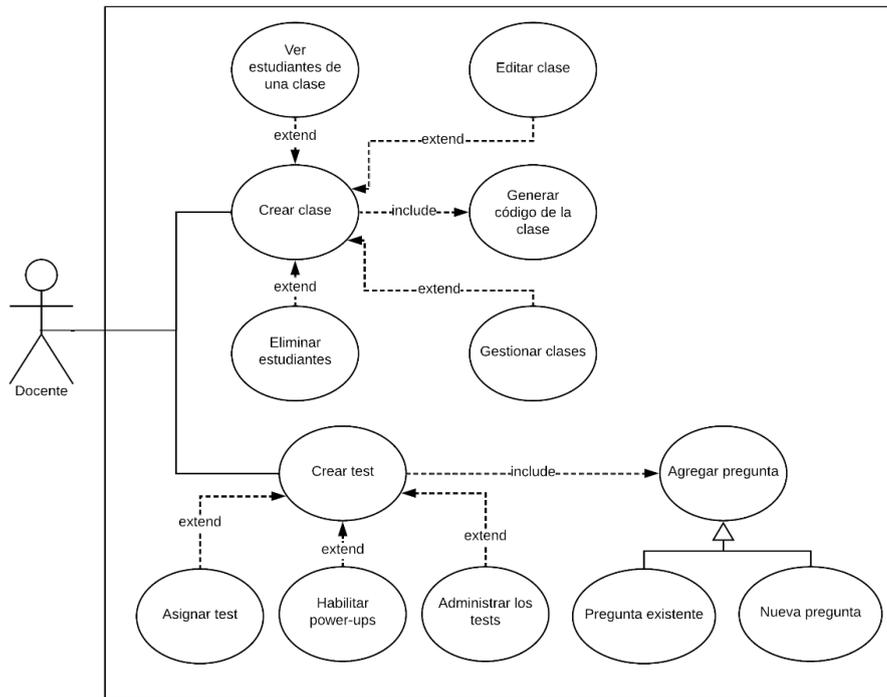


Figura 1. Diagrama de casos de uso del Docente
 Fuente: Los autores

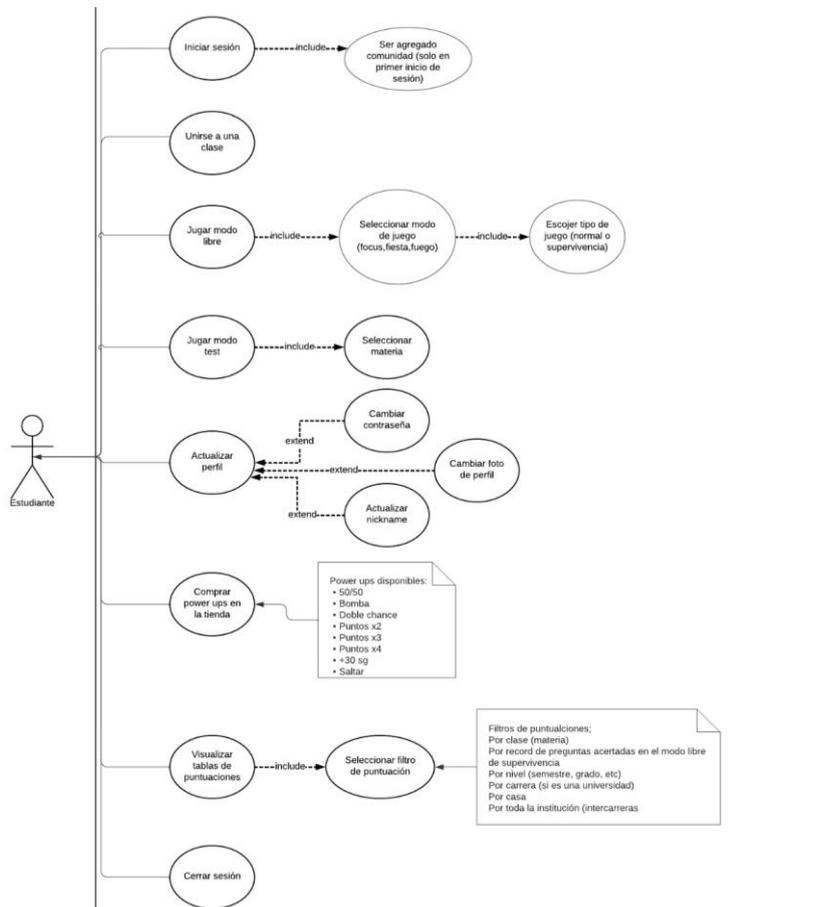


Figura 2. Diagrama de casos de uso del Estudiante
 Fuente: Los autores

De igual forma, se realizó el diagrama de clases, el cual sirve para tener claro cada una de las clases con sus respectivos atributos, además de las operaciones que cada uno de los actores desempeñan dentro del sistema. Este diagrama se puede apreciar en la Figura 3 y de igual forma se encuentra más detallado en el Anexo 4.

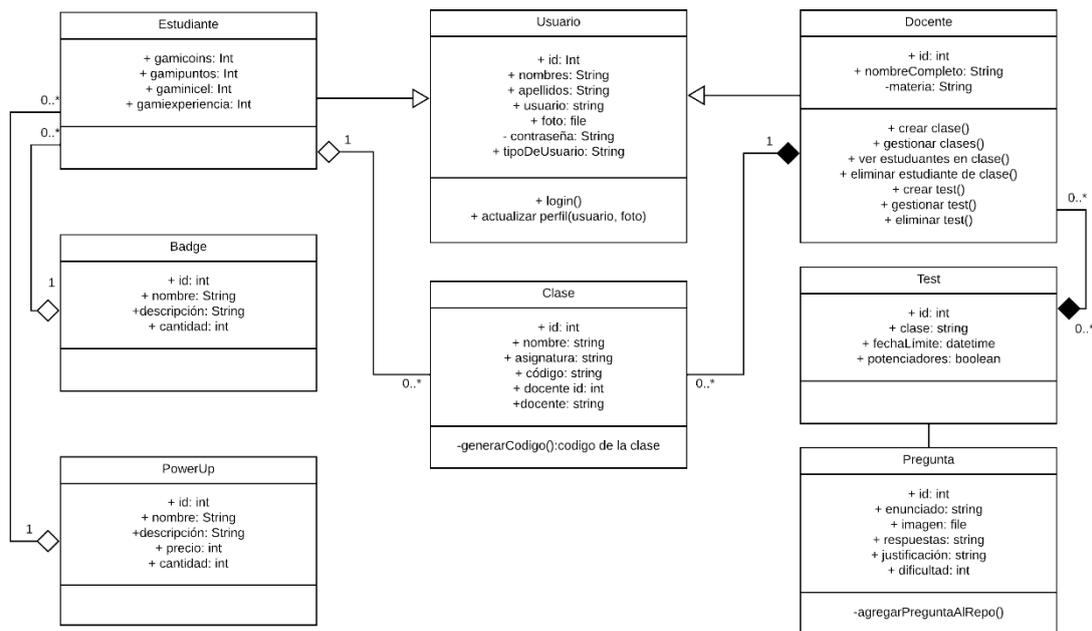


Figura 3. Diagrama de clases
Fuente: Los autores

3.3. DESARROLLO

Para la creación de la base de datos se optó por el uso de la metodología “Code First”, del cual se creó su respectivo informe que se encuentra en el Anexo 5. Una vez creada la base de datos se generó el modelo de base de datos, el cual se muestra en la Figura 4, y se obtuvieron un total de veintinueve (29) tablas relacionadas entre sí para su correcto funcionamiento.

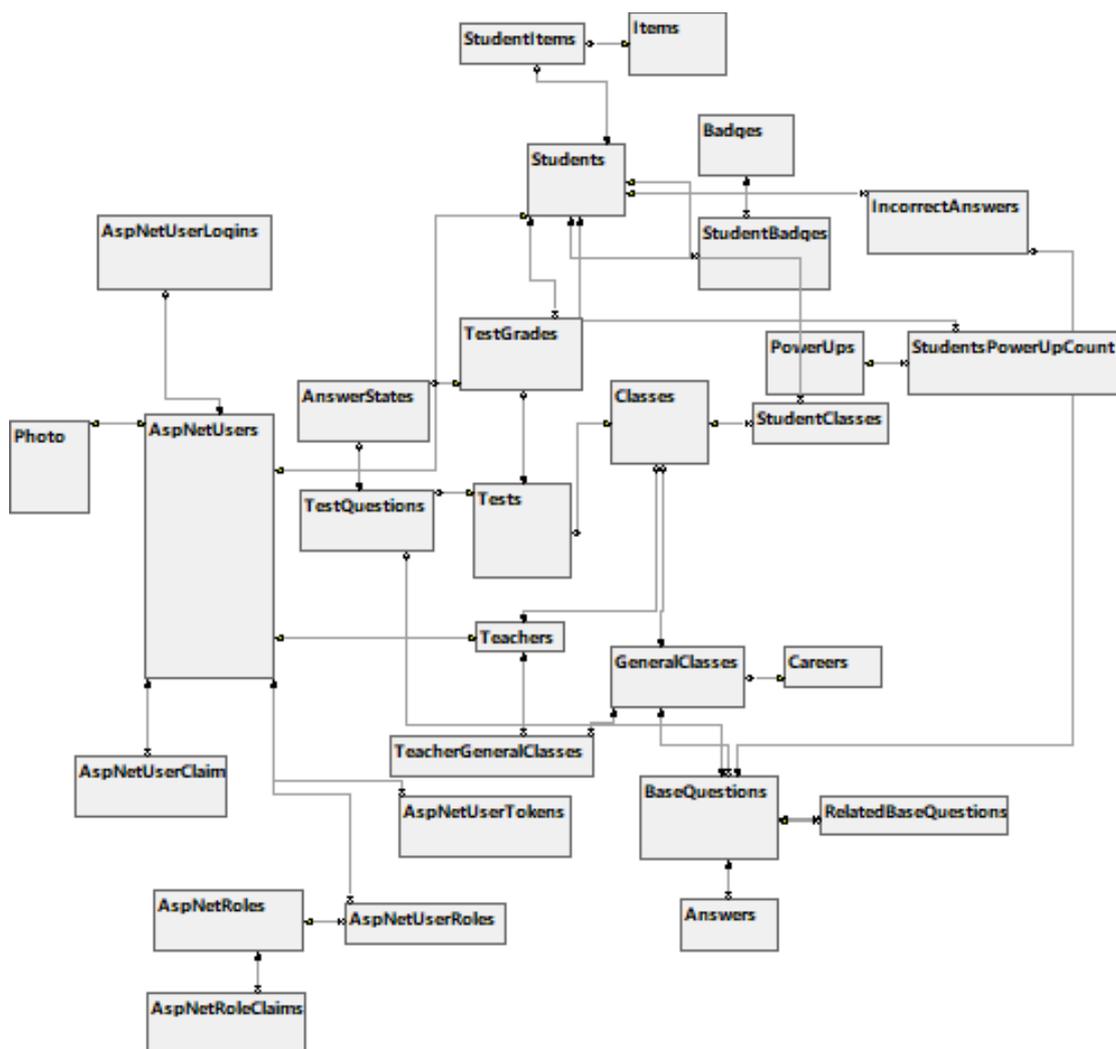


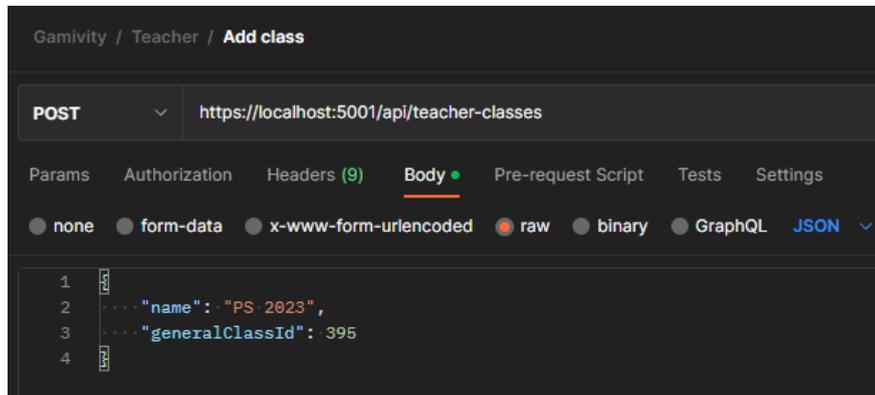
Figura 4. Diagrama de modelo de base de datos
Fuente: Los autores

Luego, para la creación de los módulos de la gestión docente, se desarrollaron una serie de endpoints asociados a esta misma, los cuales se detallan más a profundidad en el Anexo 6, sin embargo, se mostrarán superficialmente algunos de ellos a continuación.

3.3.1. MÉTODO POST

Crear nueva clase

Permite generar una nueva clase especificando el nombre y la asignatura.

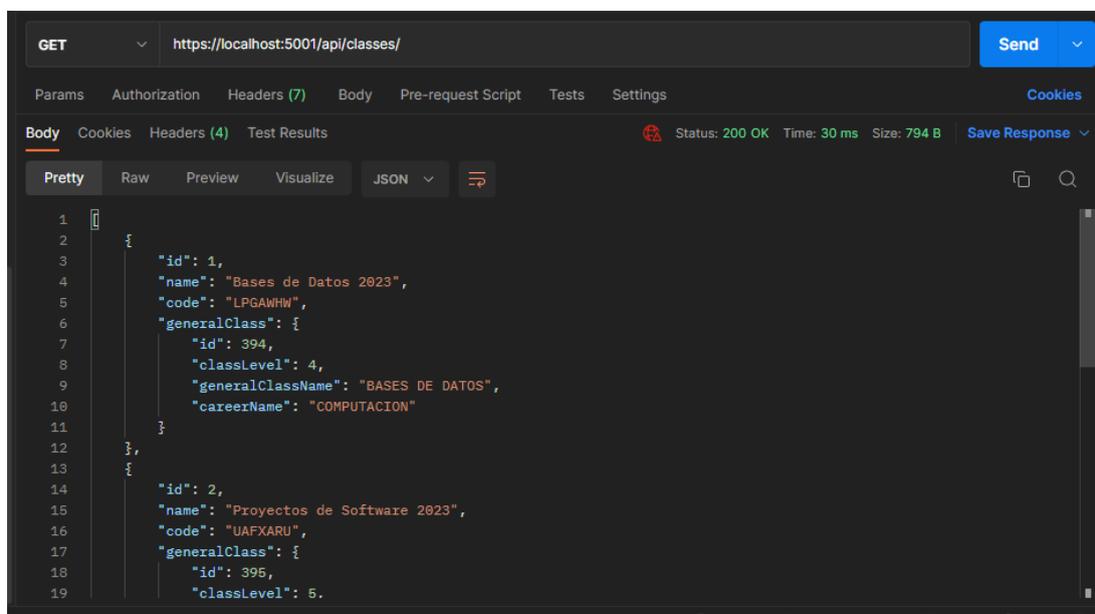


*Figura 5. Petición: Crear nueva clase
Fuente: Los autores*

3.3.2. MÉTODO GET

Obtener clases

Se recuperan todas las clases creadas por el docente junto con información de la asignatura.



*Figura 6. Petición: Obtener clases
Fuente: Los autores*

3.3.3. MÉTODO DELETE

Expulsar a un estudiante de una clase

Para ello, se deberá de especificar el id del estudiante, así como el id de la clase a la que se le quiere aplicar dicha acción (*classId*).

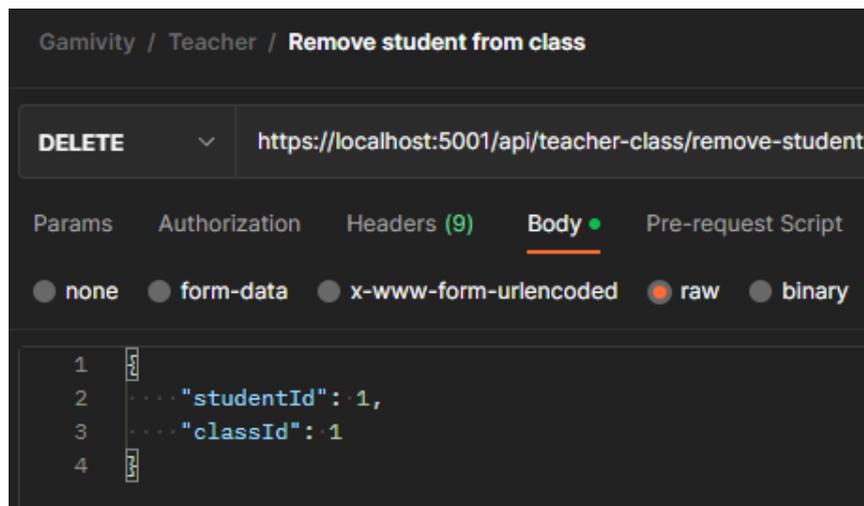


Figura 7. Petición: Expulsar a un estudiante de una clase
Fuente: Los autores

3.3.4. VISTAS DE LA GESTIÓN DOCENTE

Crear clase

Vista para crear las clases a partir de las asignaturas que el docente imparte.

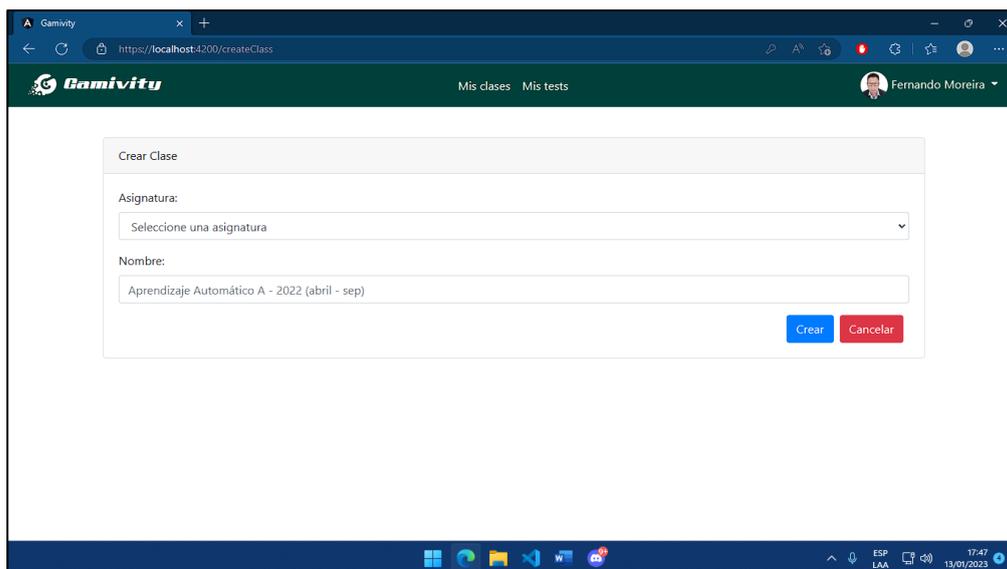


Figura 8. Vista: Crear clase
Fuente: Los autores

Mis clases

Se enlistarán las clases que el docente haya creado hasta el momento y desde aquí se administrarán cada una de ellas.

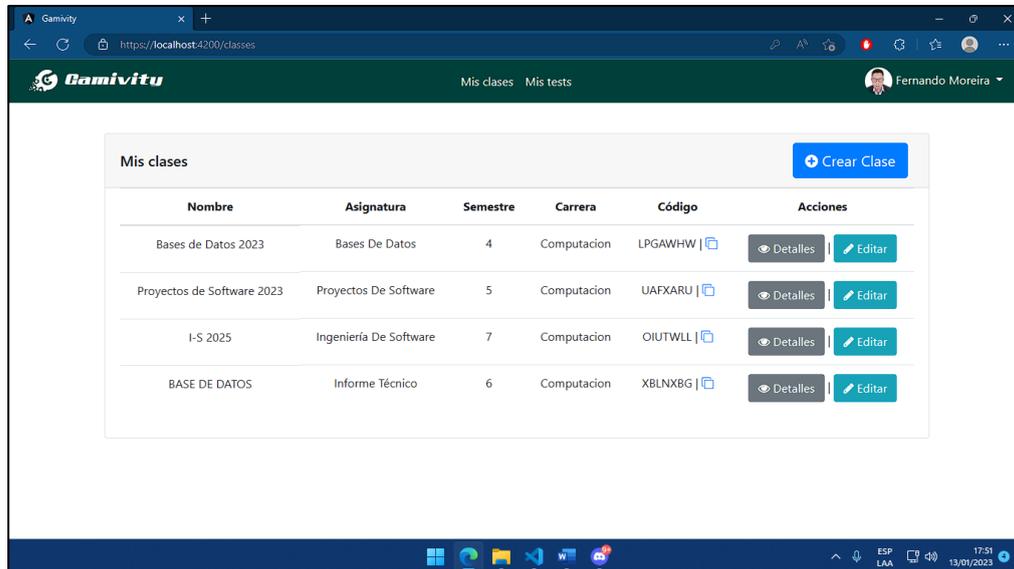


Figura 9. Vista: Clases
Fuente: Los autores

Detalles de una clase

En esta vista el docente puede controlar y administrar sus clases, teniendo la opción de poder expulsar a un estudiante.

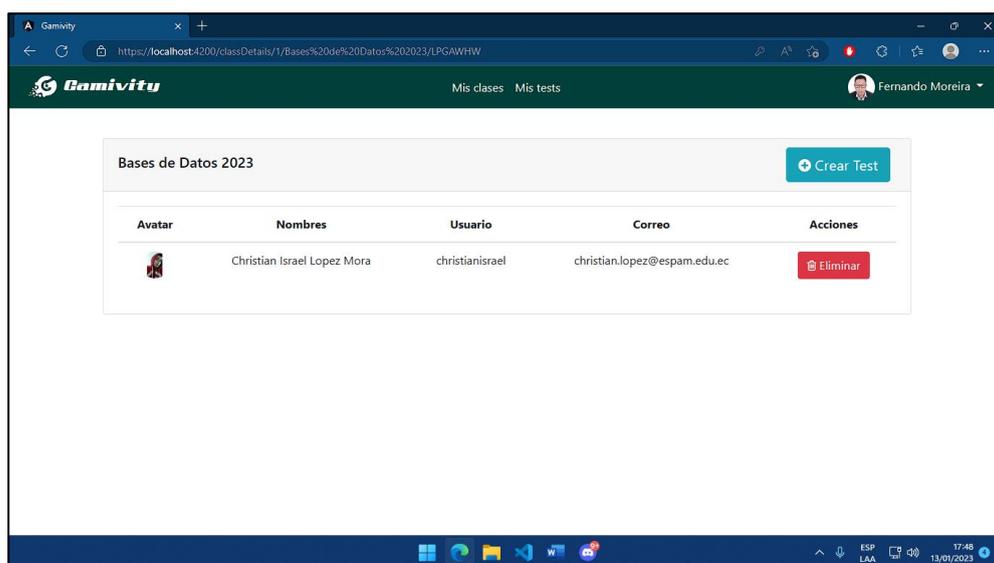


Figura 10. Vista: Detalles de una clase
Fuente: Los autores

3.3.5. MÓDULO DE APRENDIZAJE PERSONALIZADO

Referente a la integración del método de inteligencia artificial para implementar el módulo de aprendizaje personalizado al sistema, se realizó una revisión bibliográfica para poder determinar y seleccionar el modelo más adecuado que será utilizado. Dicho informe se encuentra en el Anexo 7, y del cual el modelo que salió vencedor fue “paraphrase-MiniLM-L3-v2”, ya que fue el que mejor promedio de tiempo de ejecución tuvo en comparación con los otros modelos, además de que la diferencia del promedio de puntaje respecto a sus semejantes no era tan significativa. Dichos resultados se muestran en la Tabla 4 que se presenta a continuación.

Tabla 4.

Evaluación de los modelos puestos a prueba

Modelo	Puntaje T1	Puntaje T2	Puntaje T3	Promedio Puntaje (máx. 30)	Promedio tiempo de ejecución
paraphrase-multilingual-MiniLM-L12-v2	20	22	24	22	1.219
distiluse-base-multilingual-cased-v1	22	22	16	20	1.7824
paraphrase-multilingual-mpnet-base-v2	20	18	22	20	3.2316
paraphrase-MiniLM-L3-v2	18	18	22	19.33	0.489
clip-ViT-B-32-multilingual-v1	20	20	24	21.33	2.273

Fuente: Los autores

Teniendo ya un modelo seleccionado a partir de la revisión mencionada previamente, se procedió a realizar su respectiva implementación al sistema. En el Anexo 8 se puede apreciar más a detalle cada uno de los procesos involucrados a la hora de relacionar preguntas junto a sus endpoints correspondientes. A continuación, se presentan los diagramas de flujo que dan a conocer los procesos a la hora de relacionar las preguntas y cómo estas se agregan a los controladores de aprendizaje personalizado.

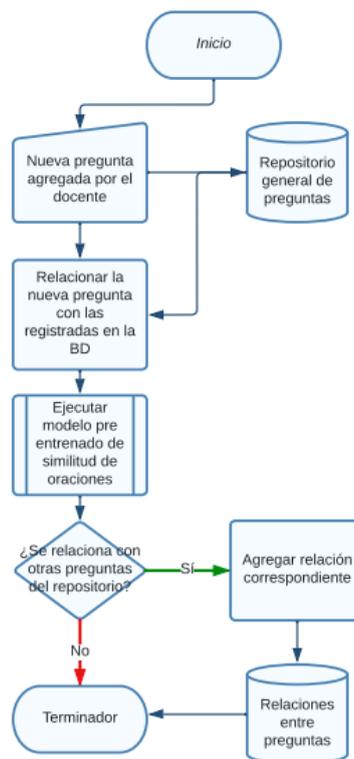


Figura 11. Diagrama de flujo del proceso de agregación de relación entre preguntas
Fuente: Los autores

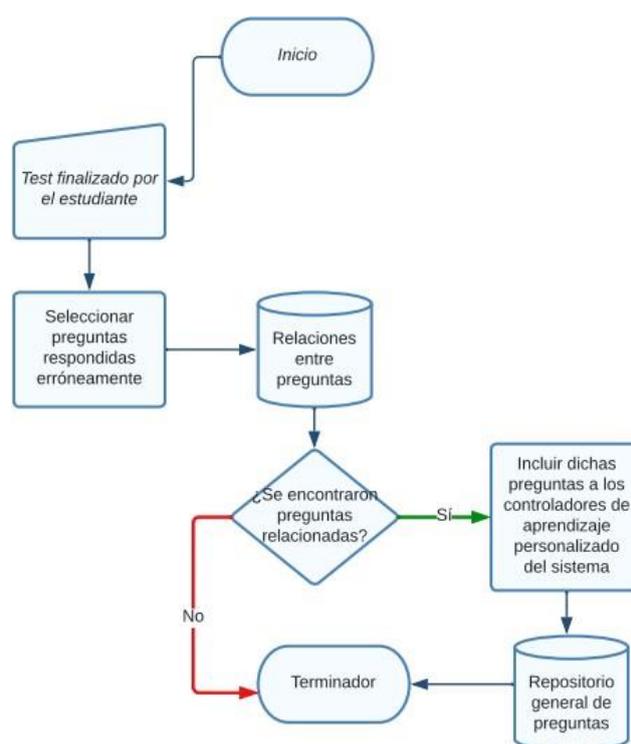


Figura 12. Diagrama de flujo del controlador de Aprendizaje personalizado
Fuente: Los autores

3.4. PRUEBAS

Siguiendo con la metodología utilizada finalmente se llegó a la fase de pruebas, en la cual se realizó el respectivo informe de pruebas unitarias con cada uno de los componentes necesarios para garantizar las funcionalidades del sistema, los cuales se encuentran establecidos en el Anexo 9. A continuación se muestra un ejemplo de pruebas unitarias, el cual representa el acceso a la plataforma (docentes y estudiantes), así como la edición del perfil de usuario.

Tabla 5
Componente unitario “Acceso a la plataforma”

Descripción	Resultado esperado	Resultado obtenido
¿Qué sucede si ingresa un usuario con rol estudiante o rol docente por primera vez con credenciales válidas?	Mensaje satisfactorio	Alerta de registro e inicio de sesión correcto
¿Qué sucede si ingresa un usuario con rol docente a través de credenciales válidas a la plataforma web?	Mensaje satisfactorio	Alerta de inicio de sesión correcto y posterior redirección al “Home”

¿Qué sucede si ingresa un usuario con rol estudiante a través de credenciales válidas a la plataforma web?	Mensaje de alerta (redirección plataforma gamificada)	Alerta de rol no permitido y posterior redirección a URL correcta
¿Qué sucede si ingresa un usuario con rol docente a través de credenciales válidas a la aplicación móvil gamificada?	Mensaje de alerta (redirección plataforma web de gestión)	Alerta de rol no permitido y posterior redirección a URL correcta
¿Qué sucede si ingresa un usuario con rol estudiante por primera vez con credenciales válidas, pero no se encuentra matriculado al periodo académico actual?	Mensaje de error	Alerta de error al registrar al usuario
¿Qué sucede si un usuario de cualquier rol accede al apartado de editar usuario?	Se muestran campos editables	Campos editables mostrados: nombre de usuario y foto de perfil

Una vez realizadas las respectivas pruebas unitarias, se procedió con la redacción del informe de migración del sistema, mismo que se encuentra en el Anexo 10. En este podemos encontrar la información necesaria para poder migrar e inicializar el sistema (API, teacherClient y microservicio Flask), así como el acta correspondiente del acceso a los servidores de la Universidad para el posterior despliegue del núcleo del sistema y la plataforma web de gestión.

Finalmente, se desarrolló el respectivo manual de programador, para así poder proveer documentación acerca del código fuente, y de esa forma facilitar la introducción al sistema a nuevos desarrolladores para poder escalar el sistema en futuras versiones. De la misma forma se documentaron los endpoints para así ofrecer mayor claridad a la hora de ser consumidos por los respectivos clientes.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

Una vez concluido el presente documento de trabajo titulación, los autores pudieron llegar a las siguientes conclusiones:

- Las entrevistas realizadas a los docentes involucrados dentro del proyecto de vinculación, permitieron establecer las directrices iniciales propias del sistema, para posteriormente hacer énfasis en las características de experiencia personalizada que se contemplaron a partir del análisis de funcionalidades de aplicaciones con entornos gamificados en el mercado, para así, poder establecer un marco de referencia acerca de la implementación de este tipo de características dentro del aplicativo. Consecuentemente, a partir de la información recopilada, se dio paso al desarrollo del respectivo documento ERS.
- El diseño de los diagramas de casos de uso, de clase y de paquete, permitió definir la estructura, relaciones y acciones del sistema al ser manipulado por alguno de los actores, en conjunto con la identificación de las tecnologías a utilizar, pudieron dar paso a la creación de la base de datos, la respectiva elaboración del backend, así como la codificación de la plataforma web de gestión docente.
- La implementación de un modelo NLP en el sistema multiplataforma “Gamivity” fue de suma importancia, y si bien existen un sinnúmero de modelos pre-entrenados de este tipo, los que mejores resultados ofrecieron fueron los del tipo “Transformers”, al ser estos muy eficientes con el entendimiento del lenguaje natural, sobre todo, a la hora de estimar la similitud entre oraciones. Después del respectivo análisis realizado, se definió que el modelo candidato para las necesidades planteadas del sistema fue “paraphrase-MiniLM-L3-v2”, ofreciendo este un rendimiento superior a los demás modelos evaluados.
- La implementación del núcleo del sistema junto a la plataforma web de gestión en los servidores de la ESPAM MFL, tuvo lugar después de las respectivas pruebas unitarias al sistema, las cuales otorgaron una capa extra

de fiabilidad a la hora de ejecutar cada uno de los endpoints. Consecuentemente la documentación del sistema tanto a nivel de estructura de proyecto como de endpoints, permitió que la información técnica asociada a este proyecto fuera de fácil acceso y comprensión, reduciendo así la curva de aprendizaje para todas las partes interesadas.

4.2. RECOMENDACIONES

A partir de las conclusiones mencionadas se realizan las siguientes recomendaciones:

- Hacer uso de la entrevista como una técnica para recolectar información, ya que esta permite determinar parámetros como el alcance del proyecto y ciertos requerimientos del sistema propuestos por el personal entrevistado mediante el intercambio de ideas.
- Realizar el documento ERS dispuesto por el estándar IEEE 830, ya que, de esta manera se podrá definir el sistema a desarrollar de forma completa, facilitando así el proceso de desarrollo del mismo.
- Modelar la información detallada en los requerimientos funcionales y no funcionales mediante diagramas de casos de usos y diagramas de clases, debido a que todos estos modelos definen la estructura de la aplicación y son el punto de partida para el desarrollo.
- Realizar un análisis previo a la selección de un modelo de inteligencia artificial, debido a que existe una gran variedad de estos y cada uno puede tener un mejor desempeño según qué necesidades se tengan.
- Utilizar herramientas especializadas en el diseño y testeo de APIs, tales como Postman o SwaggerUI para documentar los endpoints, ya que estas ofrecen una mayor facilidad a la hora de realizar cualquier operación del ciclo de vida del desarrollo de una API.
- Hacer uso de la metodología Extreme Programming (XP) para el desarrollo de un producto software, debido a que el proceso metodológico que proporciona cada una de sus fases, permite una mejor adaptabilidad entre el cliente y el equipo de desarrollo, obteniendo así los resultados esperados.

BIBLIOGRAFÍA

- AlSaad, F., & Durugbo, C. (2021). Gamification-as-Innovation: A Review. *International Journal of Innovation and Technology Management*.
- Andrade, S., Lacerda, A., Calado, P., Lima, T., & Canedo, E. (2017). Gamification in education: A methodology to identify student's profile. *2017 IEEE Frontiers in Education Conference (FIE)*, 1-8. doi:10.1109/FIE.2017.8190499
- Araya, R., Arias, E., Bottan, N., & Cristia, J. (2019). *Does Gamification in Education Work? Experimental Evidence from Chile*. Obtenido de <https://www.econstor.eu/handle/10419/208169>
- Coordinación de Vinculación con la Comunidad. (2022). Obtenido de <http://www.espam.edu.ec/web/unidades/vinculacion.aspx>
- Dichev, C., & Dicheva, D. (2017). Gamifying education: what is known, what is believed and what remains uncertain: a critical review. *International Journal of Educational Technology in Higher Education* 14. doi:<https://doi.org/10.1186/s41239-017-0042-5>
- ESPAM MFL. (2016). *Modelo Educativo Escuela Superior Politécnica Agropecuaria de Manabí "Manuel Félix López."*. Obtenido de <http://www.espam.edu.ec/recursos/sitio/espam/ModeloEducativo2016.pdf>
- ESPAM MFL. (2019). *Estatuto*. Obtenido de <http://www.espam.edu.ec/recursos/sitio/espam/EstatutoESPAMMFL.pdf>
- ESPAM MFL. (2020). *Instructivo para actividades de vinculación con la sociedad en la formación de estudiantes de grado y posgrado de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López*. Obtenido de <http://www.espam.edu.ec/recursos/sitio/informativo/archivos/reglamento/InstructivoActividadesVinculacionEstudiantesGradoPosgrado20200923.pdf>

- ESPAM MFL. (2021). *ESPAM MFL*. Obtenido de <http://www.espam.edu.ec/web/universidad/filosofia.aspx>
- Facer, K., & Selwyn, N. (2021). Digital technology and the futures of education – towards ‘non-stupid’ optimism. *Futures of Education UNESCO*.
- Hernandez, R. (2017). Impacto de las TIC en la educación: Retos y Perspectivas. *Propósitos Y Representaciones*, 325–347.
- Honorable Consejo Politécnico. (2015). *ESPAM MFL*. Obtenido de <http://www.espam.edu.ec/recursos/sitio/informativo/archivos/reglamento/ReglamentoVinculacionColectividad.pdf>
- Lai, J., & Bower, M. (2019). Evaluation of technology use in education: Findings from a critical analysis of systematic literature reviews. *Journal of Computer Assisted Learning Wiley*, 1-19. doi:<https://doi.org/10.1111/jcal.12412>
- Major, L., & Francis, G. (2020). Technology-Supported Personalised Learning: A Rapid Evidence Review. *EdTech Hub Rapid Evidence*. doi:10.5281/zenodo.4556925
- Martinez, G. (2017). Tecnologías y nuevas tendencias en educación: aprender jugando. *El caso de Kahoot*, 252–277.
- Peng, H., Ma, S., & Spector, J. (2019). Personalized adaptive learning: an emerging pedagogical approach enabled by a smart learning environment. *Smart Learning Environments* 6. doi:<https://doi.org/10.1186/s40561-019-0089-y>
- Saggah, A., Atkins, A., & Champion, R. (2020). A Review of Gamification Design Frameworks in Education. *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*.
- Shemshack, A., & Spector, J. (2020). A systematic literature review of personalized learning terms. *Smart Learning Environments*, 7-33. doi:<https://doi.org/10.1186/s40561-020-00140-9>

Soodtoetong, N., & Rattanasiriwongwut, M. (2022). Educational Transformation with Virtual Classroom: Integrated between TTF and Social Motivation. *TEM Journal*, 439-445. doi:10.18421/TEM111-56

Troncoso, C., & Amaya, A. (2017). Entrevista: guía práctica para la recolección de datos cualitativos en investigación de salud. *Revista de la Facultad de Medicina de la UNAL*, 329-332.

Universidades de Ecuador. (2022). Obtenido de <https://www.universidades.com.ec/escuela-superior-politecnica-agropecuaria-de-manabi>

ANEXOS

ANEXO 1. RESUMEN DE ENTREVISTA

Resumen de entrevista

Proyecto: Gamivity

Autores: Dany Álava – Carlos Saavedra

Introducción

El presente documento, recopila los aspectos más importantes discutidos en las sesiones iniciales de la fase de definición de requerimientos del proyecto “Gamivity”. Dichas entrevistas tuvieron lugar en los meses de octubre y noviembre del 2021. Los actores involucrados en las mismas, se resumen en la siguiente lista:

Entrevistadores

- Dany Álava, Carlos Saavedra
- Ariel Tuárez, Geovanny Zambrano: encargados de la tesis complementaria

Entrevistados:

- Ingeniero Fernando Moreira: encargado del proyecto de vinculación
- Ingeniero Joffre Moreira: director de la carrera de Computación

El eje central de las entrevistas realizadas, residió en la definición de aspectos básicos y, sobre todo, de alcance. Estas incluían tópicos, los cuales iban desde qué producto software se pretendía desarrollar, así como qué paradigmas y herramientas convenía utilizar para poder alcanzar el objetivo final satisfactoriamente.

Desarrollo

Si bien la finalidad de la entrevista ya fue mencionada, no se debe dejar de lado las ideas principales de la misma. Y es que, tratándose de un proyecto de vinculación perteneciente a la Escuela Superior Politécnica Agropecuaria de Manabí, en el cual no solo se encuentran involucrados los integrantes de este grupo de trabajo, sino que también otro grupo complementario, así como los docentes encargados del mismo proyecto.

Entonces, enfocándose en las ideas principales, se tendrían:

- Definición del alcance del proyecto
- Metodologías/paradigmas a implementar
- División de la carga de trabajo
- Tecnologías a utilizar
- Propuestas de funcionalidades

Una vez mencionadas las ideas principales tratadas en la entrevista de forma general, surge a raíz de esto una serie de preguntas respecto a lo correspondiente con la temática del presente documento, las cuales servirán de guía para poder dar inicio a la ejecución del proyecto. Una muestra de las preguntas más relevantes se detalla a continuación.

- **¿Qué se va a desarrollar?**

Una solución software multiplataforma de tipo web, la cual presente un entorno gamificado en la que tanto alumnos como docentes de la institución, puedan desarrollar ciertas actividades académicas de una manera más lúdica y gratificante.

- **¿A qué problemática se pretende aportar una solución y cómo pretende hacerlo?**

El sistema educativo reside en mecanismos que no han variado significativamente a lo largo del tiempo. Esta afirmación, es la causa de que, por lo general, se categoriza al proceso de enseñanza-aprendizaje como uno que incluye metodologías obsoletas de cierta manera. La solución software

propuesta, pretende implementar formas innovadoras de involucrar tanto al docente como a su alumnado en un entorno virtual interactivo, tratando así de minimizar procesos monótonos que puedan aparecer en el camino.

- **¿Qué metodologías/paradigmas se deberían incluir en la solución?**

Dentro de estas metodologías/paradigmas aplicables en ámbitos educativos, se pretende realizar la utilización de técnicas de “gamificación” como una particularidad del aplicativo, la cual genere sobre los estudiantes una motivación el realizar tanto las actividades propuestas por los docentes, como fomentar el autoaprendizaje en los mismos. Adicional a esto, se busca que la solución conste con un sistema de aprendizaje personalizado a través de inteligencia artificial, el cual mediante recomendaciones de contenido sirva como apoyo para los estudiantes y estos puedan fortalecer sus conocimientos en las temáticas o asignaturas que les ocasione algún tipo de dificultad.

- **¿Cuál va a ser la división de la carga de trabajo para el proyecto?**

Existen dos grupos de trabajo en el proyecto, el grupo A está conformado por Dany Álava y Carlos Saavedra (autores del presente documento); por otro lado, el grupo B está conformado por Ariel Tuárez y Geovanny Zambrano.

La carga de trabajo para el desarrollo de la solución software, se resume de la siguiente forma:

Grupo A: desarrollo del backend y de los módulos de gestión docente.

Grupo B: desarrollo del frontend del aplicativo (versión móvil y web).

Cabe recalcar que los módulos de gestión docente, solo estarán disponibles en la versión web del aplicativo.

- **¿Qué tecnologías conviene utilizar para el desarrollo de la solución?**

Considerando que inicialmente el proyecto va a ser implementado en los servidores de la universidad, en la cual existen soluciones del lado del servidor construidas bajo la tecnología “Dot Net” de Microsoft, por conveniencia el backend del aplicativo se tendría que desarrollar bajo

dicho framework. Cabe recalcar que, para el desarrollo del frontend, se utilizará “Angular” para la plataforma web de gestión docente y “Flutter” para la aplicación móvil gamificada.

Conclusión

Las entrevistas realizadas en la fase inicial del proyecto, fueron de gran ayuda para poder sentar las bases de lo que se pretende buscar con la creación del producto- software en cuestión. En las mismas, se logró despejar las dudas más cruciales incluidas el qué, por qué y cómo se pretendía llegar a la aplicación en la forma de mínimo producto viable. Del mismo modo, se establecieron lineamientos base sobre qué tipo de funcionalidades o mecánicas podrían ser incluidas en la misma. Es por ello, que un aspecto muy destacable de esta serie de entrevistas, es el de la flexibilidad que otorgan los lineamientos expuestos, debido a que los entrevistados proponían ideas generalizadas que los desarrolladores mismos deberían de madurar a lo largo de la ejecución del proyecto, permitiendo así, un entorno de desarrollo en el que el proceso creativo no está restringido.

ANEXO 2-A. INFORME DE FUNCIONALIDADES

Informe de funcionalidades

FUNCIONALIDADES DE EXPERIENCIA
PERSONALIZADA EN APLICACIONES
DEL MERCADO

Proyecto: Gamivity

Autores: Dany Álava – Carlos Saavedra

Índice

Introducción.....	39
Desarrollo.....	40
Componentes de la gamificación.....	40
Conclusiones.....	43
Bibliografía.....	44

Introducción

Como parte del trabajo inicial de la fase del desarrollo de la aplicación “Gamivity”, más concretamente, en el ámbito de exploración de aplicaciones similares en el mercado, se realiza la presente investigación con la finalidad de extraer las características en común que dichas apps ofrecen. El eje central para la selección de las mismas, reside en criterios como: ¿es la gamificación el paradigma implementado en esta aplicación?, ¿tiene componentes de enseñanza-aprendizaje?, entre otros. De la misma manera, uno de los apartados de mayor interés dentro de esta fase explorativa, es el de determinar si dichas aplicaciones con los componentes previamente mencionados, también son impulsados por Inteligencia Artificial, ya que esta característica, es una de las principales funcionalidades que se busca ofrecer dentro de este proyecto. Cabe recalcar, que el objetivo final de este documento, no es el de seleccionar las características más frecuentes entre este tipo de aplicaciones para luego ser implementadas forzosamente en un producto que carezca de consistencia, sino más bien, el de formar un marco de contexto en el cual sirva como un bloque de referencia y punto de partida para la construcción de la misma.

Desarrollo

Componentes de la gamificación

Si bien los componentes de las aplicaciones que hacen uso de la gamificación son las dinámicas, mecánicas y componentes del juego, estos a su vez se dividen en diversos tipos. Sin embargo, al momento de realizar el debido estudio de las aplicaciones analizadas, se tomaron en cuenta las siguientes características a partir del “Modelo Dinámico para la Gamificación del Aprendizaje (DMGL, por su abreviación en inglés)” (Kim & Lee, 2015).

- **Misiones:** Son tareas las cuales tienen que ser resueltas por un usuario, o un grupo de estos, para conseguir una determinada recompensa.
- **Desbloqueables progresivos:** Se trata de aquellos objetos que inicialmente no se encuentran disponibles y se irán desbloqueando a lo largo del progreso del usuario dentro de la aplicación.
- **Power ups:** Los Power-ups o Potenciadores son objetos que añaden capacidades especiales al usuario de la aplicación y suelen obtenerse mediante la recopilación de varios accesorios relacionados.
- **Coleccionables:** Son elementos que se obtienen como recompensa por completar algún objetivo y pueden ser almacenados.
- **Creación de avatar:** Como su nombre lo indica, es una característica que permite crear o personalizar el personaje con el que el usuario se identifica dentro de una aplicación.
- **Recompensas diarias:** Estas son recompensas que se le otorgan a los usuarios por el simple hecho de abrir la aplicación a diario, esto con la intención de motivar a los usuarios a hacer uso de la misma con una mayor frecuencia.

- **Retos diarios:** Son retos sencillos que se plantean diariamente y tienen el mismo objetivo que las recompensas diarias.
- **Multijugador:** El multijugador es una característica en la que las aplicaciones pueden ser usadas por varios usuarios a la vez y estos pueden ser oponentes individuales, estar agrupados en equipos o formar un solo equipo contra un objetivo en común.
- **XP:** El XP o puntos de experiencia son aquellos que indican los triunfos en las aplicaciones, ya que cuanto más difícil sea conseguir el logro, más XP obtendrá el usuario.
- **Tienda:** Este es un apartado en el que los usuarios pueden hacer uso de sus bienes virtuales para poder canjear algún artículo o mejora que sea de su agrado.
- **Experiencia Personalizada a través de IA:** Esta característica hace referencia a cualquier metodología impulsada por inteligencia artificial, usada para ofrecer al usuario una serie de funcionalidades seleccionadas exclusivamente con base en sus necesidades y en cómo éste se desenvuelve en la aplicación.

Cabe recalcar, que además de todas las características previamente detalladas, la selección de las 40 aplicaciones, se realizó mediante criterios como el número de descargas y la valoración media de las mismas. Teniendo en cuenta lo antes descrito, se procede a realizar el respectivo análisis cuantitativo basado en dichas características, las cuales se pueden apreciar en la ilustración 1.

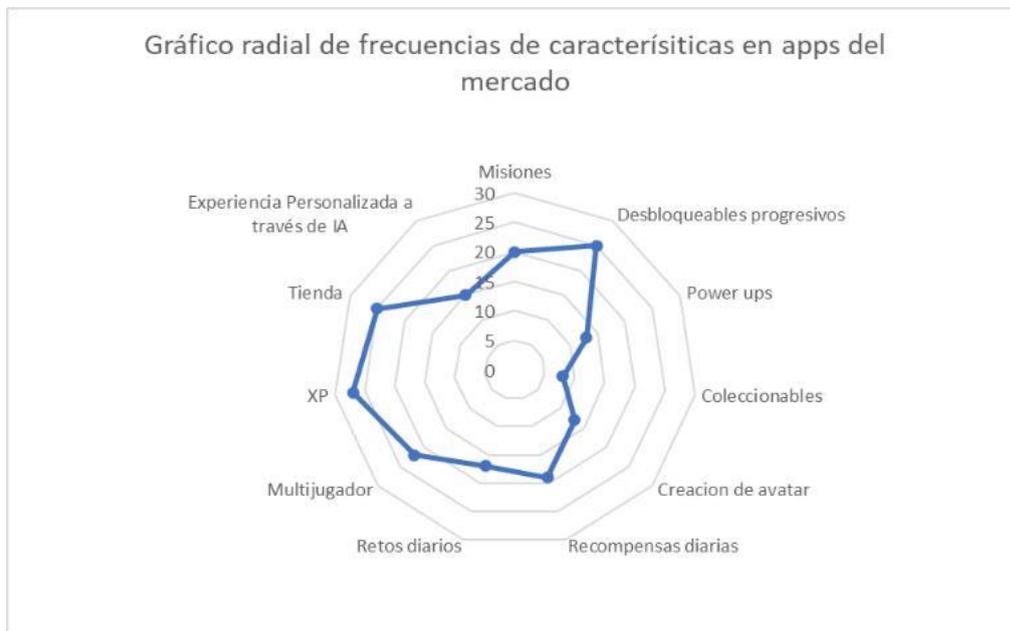


Ilustración 1: Características de las apps analizadas

A partir de lo mostrado en el gráfico radial de frecuencias, se logra apreciar una tendencia hacia las características “XP”, “tienda” y “desbloqueables progresivos” en dichas aplicaciones estudiadas. En su contraparte, las características “coleccionables”, “power ups” y “creación de avatar” son las menos frecuentes. En cuanto a la característica de “experiencia personalizada a través de IA”, la frecuencia de aplicaciones en las que está implementada, es un poco menos de la mitad, más específicamente en el 37.5% de las mismas.

Conclusiones

Teniendo en cuenta lo descrito en este documento, se llega a las siguientes conclusiones:

- Desde cierta perspectiva, era predecible que la característica “XP”, la cual involucra puntos, fuera una de las más frecuentes dentro de estas aplicaciones con entornos de gamificación, ya que este, por lo general, es un aspecto fundamental en la mayoría de modelos de gamificación enfocados al aprendizaje, incluido el modelo DMGL previamente mencionado.
- Al considerar que la característica de experiencia personalizada impulsada a través de inteligencia artificial, no es tan frecuente en los entornos gamificados analizados enfocados a la educación, se evidencia una oportunidad en este sector para poder experimentar con estas dos características de una manera híbrida y así poder crear una plataforma que pueda aprovechar a las mismas.
- Por último, pero no menos importante tenemos los aspectos relacionados con la experiencia personalizada a través de Inteligencia artificial, ya que como se mencionó anteriormente el 37.5% de las aplicaciones analizadas hacían uso de esta característica, y es que si bien, el uso de esta propiedad tiende a ser bastante flexible y se podría aplicar en varios escenarios diferentes, en la mayoría de las aplicaciones que hacían uso de esta, se encontraba enfocada en la recomendación de contenido de acuerdo a los gustos, tendencias y nivel de conocimiento de los usuarios.

Bibliografía

Kim, J. T., & Lee, W.-H. (octubre de 2015). *Modelo dinámico para la gamificación del aprendizaje (DMGL)*. Obtenido de <https://link.springer.com/article/10.1007/s11042-013-1612-8>

**ANEXO 2-B. TABLA DE CARACTERÍSTICAS DE
APLICACIONES**

Nombre de la app	Calificación	Misiones	Desbloqueables progresivos	Power ups	Coleccionables	Creación de avatar	Recompensas diarias	Retos diarios	Multijugador	Xp	Tienda	Experiencia personalizada a través de ia	Diseño	Categoría
Minecraft: education edition	3.9		X		X	X			X	X	X		Pixelart 3d	Adolescente/formal
Academons	3.7	X	X	X	X	X			X	X	X		Cartoon	Infantil
Brain scape	4.4		X			X			X	X			Minimalista	Adolescente/formal
Classcraft	3.9	X	X	X		X	X	X	X	X		X	Cartoon	Fantasía
Coroebus	3.3	X	X				X		X	X	X		Minimalista	Adolescente/formal
Epic to-do list — RPG planner with reminders	4.7	X	X	X		X	X	X			X		Cartoon	Fantasía
Lifeup: Gamificación To-Do & Tasks List habitrpg	4.6	X	X	X			X	X		X	X		Minimalista	Adolescente/formal
Arcademics	3.8		X	X					X	X			Cartoon	Infantil
Toovari		X	X			X	X	X	X	X	X		Cartoon	Infantil
Family team		X											Otro	Formal
Preguntados	4.4	X	X	X			X	X	X	X	X	X	Cartoon	Infantil/adolescente
Clash royale	4.2	X	X	X	X		X		X	X	X	X	Cartoon	Fantasía
Memrise	4.6		X				X	X		X			Cartoon	Formal
Duolingo	4.7			X	X		X	X		X	X	X	Cartoon/minimalista	Adolescente/formal
Songpop 2	4		X	X					X	X	X	X	Cartoon	Adolescente/formal
Enki: Learn data science, coding, tech skills	4.6		X				X	X		X		X	Minimalista	Formal
Boggle with friends	4.1			X			X	X	X	X	X		Cartoon	Infantil
Word logic	4.6		X	X						X	X		Cartoon	Infantil
Trivia star	4.8			X						X	X		Cartoon	Adolescente
Baikoh	4.1		X	X	X		X	X	X	X	X		Cartoon	Adolescente
Smartick	4.2	X	X			X				X	X	X	Otro	Infantil
Knowre	Null	X	X							X		X	Cartoon	Adolescente/formal
Kahoot!	4.7								X			X	Minimalista	Adolescente/formal
Lingokids	4.4	X	X									X	Minimalista /Cartoon	Infantil
Forest: Mantente enfocado	4.8	X	X		X		X	X	X	X	X		Minimalista	Adolescente/formal
Habitica	4.3	X	X		X	X	X		X	X	X		Pixelart	Fantasía
Classdojo	4.7	X				X			X			X	Cartoon	Infantil
Edmodo	3.9					X			X				Minimalista	Adolescente/formal
Level up life	4.1	X	X					X		X	X		Pixelart	Fantasía
Quizlet	4.7	X										X	Minimalista	Adolescente/formal
Mimo	4.8	X	X			X	X	X	X	X	X		Minimalista	Formal
Code combat	Null	X	X		X	X			X	X	X	X	Pixelart	Fantasía
Cuemath	Null		X					X			X		Otro	Formal
Word peace							X	X			X		Otro	Fantasía
Quizizz	4.8								X				Minimalista	Formal
Elever	Null	X				X	X	X	X		X	X	Minimalista	Fantasía
Science Experiments and Projects	Null												Minimalista	Formal
Clash of Clans	4.7		X	X			X	X	X	X	X	X	Cartoon	Fantasía
Pou	4.3				X	X	X			X	X		Cartoon	Infantil
Physics lab	Null	X							X	X	X		Minimalista	Formal
Total		21	26	14	9	14	19	17	23	28	26	15		

Tabla completa aquí: [APPS-PLAYSTORE.xlsx - Hojas de cálculo de Google](#)

ANEXO 3. ERS

Especificación de requisitos de software

Proyecto: Gamivity

Autores: Dany Álava – Carlos Saavedra

Índice

1.	Introducción	51
1.1.	Propósito	51
1.2.	Ámbito del sistema	51
1.3.	Definiciones, Acrónimos y Abreviaturas	52
1.3.1.	Definiciones	52
1.3.2.	Acrónimos.....	52
1.4.	Visión General del Documento	52
2.	Descripción General	52
2.1.	Perspectiva del producto	53
2.2.	Funciones del producto	53
2.3.	Características de los usuarios	53
2.3.1.	Docente	53
2.3.2.	Estudiante.....	54
2.4.	Restricciones.....	54
2.5.	Suposiciones y Dependencias	55
2.6.	Modelo de caso de uso	55
3.	Requisitos Específicos.....	56
3.1.	Requerimientos Funcionales	56
3.1.1.	Elementos de juego	56
3.1.1.1.	Sistema de Puntuación	57
3.1.1.2.	Tabla de posiciones (Leaderboards)	57
3.1.1.3.	Insignias (Badgets)	58
3.1.1.4.	Potenciadores (PowerUps)	59
3.1.1.5.	Tienda	59
3.1.2.	Estudiante.....	60

3.1.2.1. Editar perfil de usuario	60
3.1.2.2. Unirse a una clase	61
3.1.2.3. Realizar test	62
3.1.3. Repositorio general de preguntas.....	62
3.1.4. Docente	63
3.1.4.1. Creación de una clase	64
3.1.4.2. Creación de tests	65
3.1.4.3. Asignación de un test.....	66
3.1.4.4. Resultados de los tests asignados.....	67
3.1.4.5. Gestionar clases	67
3.1.5. Experiencia Personalizada	68
3.1.5.1. Tests personalizados en el apartado de “autoaprendizaje” ¡Error! Marcador no definido.	
3.1.5.2. Preguntas personalizadas en los modos de juego libre .. ¡Error! Marcador no definido.	
3.2. Requerimientos no funcionales	69
3.2.1. Requerimientos de rendimiento (Eficiencia)	69
3.2.2. Restricciones de Diseño	69
3.2.3. Estándares Asociados	70
3.2.4. Atributos del sistema	70
3.2.5. Requerimientos de Desarrollo	70

1. Introducción

1.1. Propósito

El presente documento, tiene como finalidad el proporcionar la información necesaria para la construcción del backend de un sistema multiplataforma de experiencia personalizada en un entorno de gamificación para los estudiantes de la ESPAM MFL, así como de los módulos de gestión docente respectivos. Este documento de especificación de requisitos software va dirigido hacia las partes interesadas en cuanto a los detalles del sistema, especialmente, para el responsable de proyecto y los desarrolladores o diseñadores que necesiten una documentación de los requerimientos asociados al mismo.

1.2. Ámbito del sistema

El sistema completo tiene el nombre de “Gamivity”, sin embargo, como se describió previamente, el presente documento especifica los requerimientos con base en la construcción del lado del servidor del sistema, es decir, los “endpoints” requeridos para que el frontend de la plataforma, pueda realizar sus operaciones correctamente (API). Además, el sistema web de escritorio, se encargará de proveer todos los elementos de gestión docente, para que este pueda realizar todas las operaciones relacionadas con la administración de sus clases y alumnos.

El sistema “Gamivity”, tiene como propósito, el de brindar un entorno gamificado en que a través de tests/trivias asignadas por el docente o presentes en el aplicativo de manera automática, el estudiante se sienta más motivado a realizar actividades de autoaprendizaje. Así como ofrecer un sistema de aprendizaje personalizado, en el que se le presente contenido recomendado en forma de tests a partir de las falencias del estudiante en ciertas temáticas particulares. Cabe recalcar, que, en el presente documento, se describe el futuro sistema como una primera iteración de lo que se espera del mismo, como parte inicial de un proyecto que involucrará muchos más aspectos para un sistema más robusto, es decir, escalable tanto en características propias, como en la aplicación en otras instituciones educativas.

1.3. Definiciones, Acrónimos y Abreviaturas

1.3.1. Definiciones

Backend: área lógica de toda página web, es decir, la arquitectura interna del sitio que asegura que todos los elementos desarrollen la función correcta.

Gamificación: es el empleo de mecánicas de juego en entornos y aplicaciones no lúdicas con el fin de potenciar la motivación, la concentración, el esfuerzo, la fidelización y otros valores positivos comunes a todos los juegos.

1.3.2. Acrónimos

API: Application Programming Interface

NLP: Natural Language Processing

1.4. Visión General del Documento

A partir de este punto en el documento, se presenta principalmente la descripción general, los requisitos específicos, así como los apéndices del mismo. En primer lugar, la descripción general expone los factores que afectan al backend del sistema y a los requisitos asociados al mismo. En otras palabras, en dicha sección, se establece el contexto de los requisitos que se especificarán en la tercera sección del documento. Esta sección de requisitos específicos, tal y como su nombre lo indica, permitirá a los diseñadores construir/diseñar el backend del sistema, así como los módulos de gestión docente, con base a lo descrito en los mismos; de la misma forma, hará posible que se diseñen pruebas para corroborar que el sistema funcione acorde a los criterios establecidos.

2. Descripción General

En esta sección del documento es en donde se describe el contexto de los requisitos del sistema. Los cuales, se encuentran enfocados principalmente a la mejora del proceso de enseñanza-aprendizaje tradicional, implementando

formas innovadoras de involucrar tanto al docente como a su alumnado en un entorno virtual interactivo, pudiendo así maximizar la atención del usuario final.

2.1. Perspectiva del producto

Gamivity es una solución software multiplataforma de tipo web, la cual presente un entorno gamificado en la que tanto alumnos como docentes de la institución, puedan desarrollar ciertas actividades académicas de una manera más ágil y gratificante. Todo esto bajo ciertas necesidades solicitadas por los miembros involucrados dentro del proyecto de vinculación.

2.2. Funciones del producto

A continuación, se describen las funciones principales del sistema, considerando que se trata de una solución software interactiva que principalmente se centra en la asignación y resolución de actividades tipo test por parte de los docentes y estudiantes de la institución respectivamente. Además, constará con diversas mecánicas, entre ellas, un sistema de puntuación que permitirá al usuario sentirse inmerso en la aplicación, ya que con estos podrá realizar acciones como canjear una serie de recompensas disponibles en un apartado que servirá como una tienda virtual, también tendrá un apartado en el cual estarán todos los aspectos visuales disponibles que el usuario haya podido canjear a partir de los puntos recolectados y además existirá una sección en la que se podrán ver las tablas de posiciones de los estudiantes dependiendo de la puntuación individual, representadas en distintas categorías.

2.3. Características de los usuarios

2.3.1. Docente

TIPO DE USUARIO	Docente
DESCRIPCIÓN	Este tipo de usuario será el encargado de crear y asignar tests referentes a la materia que imparte.
FORMACIÓN	Conocimiento avanzado en la materia impartida en clases.
HABILIDADES	Manejo decente de una computadora.

ACTIVIDADES	<ul style="list-style-type: none"> • Crear/gestionar las clases que está encargado de impartir. • Crear o importar preguntas almacenadas en el repositoriogeneral de preguntas. • Realizar las asignaciones de los test.
--------------------	---

2.3.2. Estudiante

TIPO DE USUARIO	Estudiante
DESCRIPCIÓN	Este usuario tendrá acceso a las diferentes actividades asignadas por el docente.
FORMACIÓN	Conocimientos básicos en las diferentes asignaturas impartidas en el semestre que esté cursando.
HABILIDADES	Manejo básico de una computadora o un smartphone.
ACTIVIDADES	<ul style="list-style-type: none"> • Realizar las diferentes actividades asignadas por el docente. • Realizar los test de las diferentes modalidades de juego del modo libre. • Editar su perfil de usuario. • Unirse a una clase.

2.4. Restricciones

Las restricciones que presenta el sistema se listan a continuación:

- El servidor donde se va a alojar el backend deberá ofrecer un rendimiento adecuado para evitar latencia y proporcionar tiempos de respuesta eficientes.
- El producto final tiene que ser estable e incorporable al sistema institucional.
- Las respuestas del backend de la aplicación deberán ser entregadas en formatojson.

- El sistema del lado del servidor se desarrollará en C# bajo el framework .NET 6.0
- Todas las respuestas de la API deben estar cifradas por el protocolo “https”

2.5. Suposiciones y Dependencias

Esta solución será desarrollada para un entorno multiplataforma (Android, iOS, navegador). El ambiente de la solución tendrá un aspecto general que busca mezclarse entre los límites de lo formal y lo llamativo, acompañado del objetivo primordial, complementar y fortalecer los conocimientos de los alumnos.

2.6. Modelo de caso de uso

El siguiente diagrama de caso de uso describe las funcionalidades del sistema desde un alto nivel, involucrando a los respectivos actores y sus interacciones con el mismo.

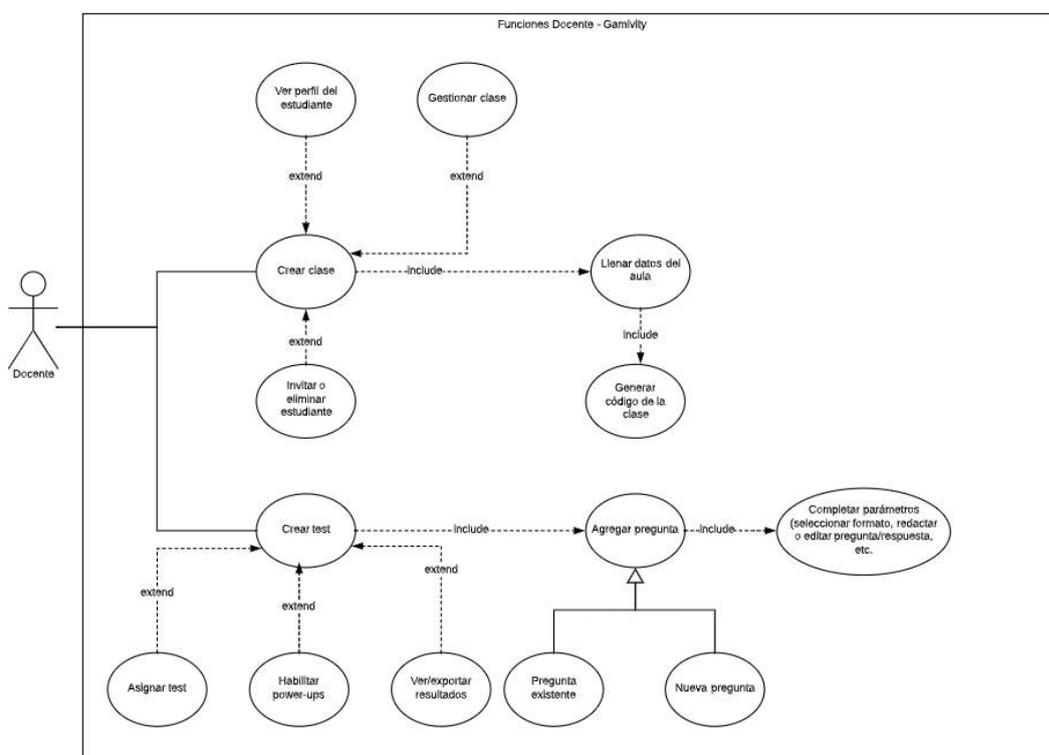


Ilustración 1. Diagrama de caso de uso: Docente

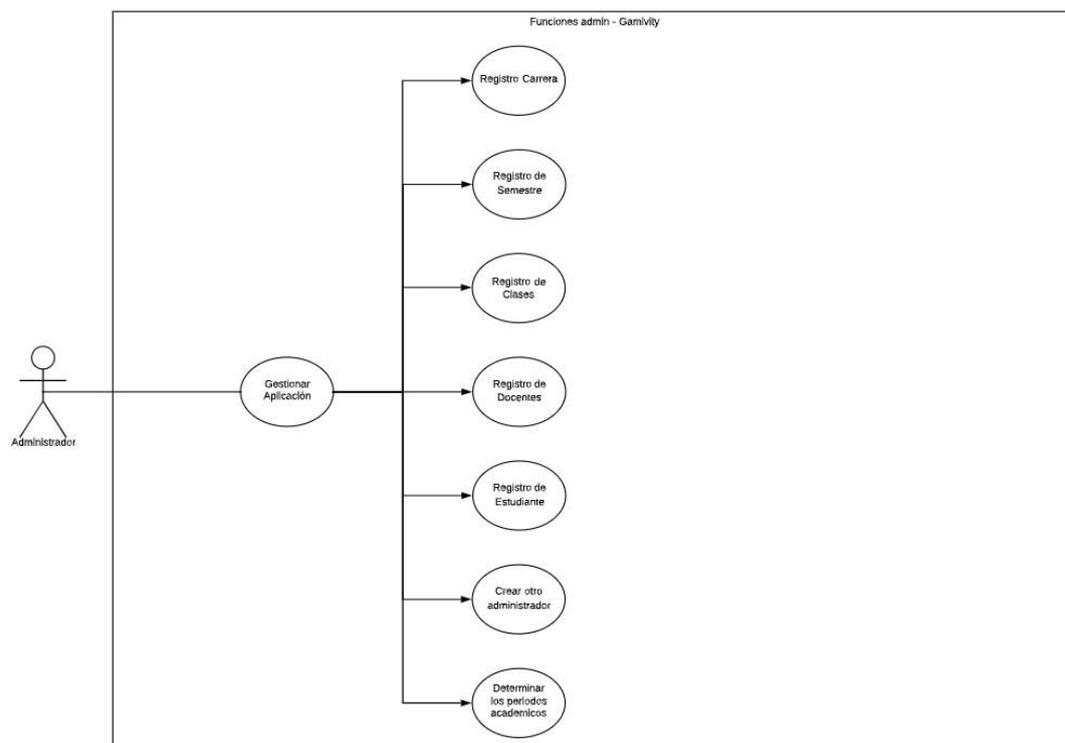


Ilustración 2. Diagrama de caso de uso: Administrador del sistema

3. Requisitos Específicos

3.1. Requerimientos Funcionales

3.1.1. Elementos de juego

Código de requisito	RF001
Nombre de requisito	Elementos de juego
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	El sistema constará con una serie de elementos de juego, los cuales son indispensables tanto para la jugabilidad como para las relaciones existentes entre el usuario y su interacción dentro del aplicativo.
PROCESO	Entre todos estos elementos de juegos podemos encontrar al sistema de puntuación, las tablas de posiciones, las insignias, los potenciadores y la tienda. Los cuales se encontrarán relacionados directamente a los atributos de los

	usuarios.
ENTRADAS	<ul style="list-style-type: none"> • Peticiones de las acciones del usuario
SALIDAS	<ul style="list-style-type: none"> • Atributo relacionado a dicho usuario
RESTRICCIONES	Ninguna

3.1.1.1. Sistema de Puntuación

Código de requisito	RF001.1
Nombre de requisito	Sistema de puntuación
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	El aplicativo constará con un sistema de puntuación entre los cuales se encuentran los gamipuntos, las gamimonedas, la gamiexperiencia y el gaminivel. Estos servirán para representar los puntos, monedas, experiencia y el nivel de usuario respectivamente.
PROCESO	Tomar la petición de obtención de cualquiera de estas recompensas y devolver el valor de dicho atributo asociado a un usuario en específico.
ENTRADAS	<ul style="list-style-type: none"> • Petición de obtención de gamipuntos • Petición de obtención de gamimonedas • Petición de obtención de gamiexperiencia • Petición de obtención de gaminivel
SALIDAS	<ul style="list-style-type: none"> • Valor del atributo de gamipuntos de dicho usuario • Valor del atributo de gamimonedas de dicho usuario • Valor del atributo de gamiexperiencia de dicho usuario • Valor del atributo de gaminivel de dicho usuario
RESTRICCIONES	Ninguna

3.1.1.2. Tabla de posiciones (Leaderboards)

Código de requisito	RF001.2
---------------------	---------

Nombre de requisito	Tabla de posiciones (Leaderboards)
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	El aplicativo constará con un sistema de posicionamiento, el cual se verá reflejado en una tabla de posiciones donde se clasificarán a los estudiantes de acuerdo a sus gamipuntos.
PROCESO	Tomar la petición de obtención de puntuación (gamipuntos) y devolver los valores de dichos atributos asociados a una serie de usuarios en específico.
ENTRADAS	<ul style="list-style-type: none"> Petición de obtención de gamipuntos
SALIDAS	<ul style="list-style-type: none"> Valores de los atributos de gamipuntos de dichos usuarios
RESTRICCIONES	Ninguna

3.1.1.3. Insignias (Badgets)

Código de requisito	RF001.3
Nombre de requisito	Insignias (Badgets)
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	El aplicativo constará con un sistema de insignias, las cuales serán las recompensas por completar ciertos hitos específicos dentro de la aplicación.
PROCESO	Tomar la petición de obtención de insignias (badgets) y devolver las insignias asociadas a un usuario en específico.
ENTRADAS	<ul style="list-style-type: none"> Petición de obtención de insignias
SALIDAS	<ul style="list-style-type: none"> Contenido del atributo de insignias de dicho usuario
RESTRICCIONES	Ninguna

3.1.1.4. Potenciadores (PowerUps)

Código de requisito	RF001.4
Nombre de requisito	Potenciadores (PowerUps)
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	El aplicativo constará con un sistema de potenciadores, los cuales podrán ser usados para obtener algún tipo de ayuda dentro de alguno de los test del modo libre o en los test de asignación solo si el docente lo configura así.
PROCESO	Tomar la petición de obtención de potenciadores (powerups) y devolver los potenciadores asociados a un usuario en específico.
ENTRADAS	<ul style="list-style-type: none"> Petición de obtención de potenciadores
SALIDAS	<ul style="list-style-type: none"> Contenido del atributo de potenciadores de dicho usuario
RESTRICCIONES	Ninguna

3.1.1.5. Tienda

Código de requisito	RF001.5
Nombre de requisito	Tienda
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	El aplicativo constará con un apartado de tienda, en el cual se mostrarán una serie de potenciadores que los usuarios podrán adquirir con sus gamimonedas obtenidas hasta el momento.
PROCESO	Tomar la petición de obtención de todos los potenciadores (powerups) y devolverlos respectivamente.
ENTRADAS	<ul style="list-style-type: none"> Petición de obtención de potenciadores

SALIDAS	<ul style="list-style-type: none"> Potenciadores disponibles hasta el momento
RESTRICCIONES	Ninguna

3.1.2. Estudiante

Código de requisito	RF002
Nombre de requisito	Estudiante
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	El estudiante es un tipo de usuario el cual consta con una serie importante de atributos, además este también puede desempeñar una serie de actividades.
PROCESO	Entre los atributos de un estudiante podemos encontrar sus nombres completos, nombre de usuario, semestre, insignias, gamipuntos, gamiexperiencia, gaminivel y las clases a las que se encuentra matriculado. Además, en cuanto a las actividades que este puede realizar tenemos el editar su perfil de usuario, unirse a las clases dentro del aplicativo y realizar test en cualquiera de sus tipos.
ENTRADAS	<ul style="list-style-type: none"> Peticiones de los atributos y acciones del usuario
SALIDAS	<ul style="list-style-type: none"> Atributo relacionado a dicho usuario
RESTRICCIONES	Ninguna

3.1.2.1. Editar perfil de usuario

Código de requisito	RF002.1
Nombre de requisito	Editar perfil de usuario
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>

DESCRIPCIÓN	El estudiante tendrá la opción de poder editar su perfil de usuario (nombre de usuario y foto de perfil) ya sea por algún error dentro de estos parámetros o simplemente por actualizar su información.
PROCESO	Tomar la petición de obtención de los parámetros que el usuario desea actualizar y devolverlos respectivamente.
ENTRADAS	<ul style="list-style-type: none"> • Petición de obtención de nombre de usuario o foto de perfil
SALIDAS	<ul style="list-style-type: none"> • Contenido del o los parámetros a actualizar
RESTRICCIONES	Ninguna

3.1.2.2. Unirse a una clase

Código de requisito	RF002.2
Nombre de requisito	Unirse a una clase
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	El estudiante tendrá la opción de poder unirse a una clase, para ello, primero deberá tener el código de acceso de la misma y la autorización del docente encargado de la clase para permanecer dentro de esta, ya que de no ser el caso el docente podrá expulsarlo de su clase en cualquier momento.
PROCESO	Tomar la petición de obtención de las clases en las que se encuentra matriculado un estudiante en específico y devolverlas respectivamente.
ENTRADAS	<ul style="list-style-type: none"> • Petición de obtención de las clases matriculadas
SALIDAS	<ul style="list-style-type: none"> • Contenido de las clases matriculadas
RESTRICCIONES	Ninguna

3.1.2.3. Realizar test

Código de requisito	RF002.3
Nombre de requisito	Realizar tests
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	El estudiante tendrá la opción de poder realizar tests, tanto los disponibles en el modo libre como los que los docentes estarán encargados de asignar. Todo esto se encuentra relacionado con atributos como la gamiexperiencia, el gaminivel, los gamipuntos, las gamimonedas y las insignias de cada usuario.
PROCESO	Tomar la petición de obtención de todos los parámetros mencionados en la descripción de este requisito y devolverlas respectivamente.
ENTRADAS	<ul style="list-style-type: none"> • Petición de obtención de gamipuntos • Petición de obtención de gamimonedas • Petición de obtención de gamiexperiencia • Petición de obtención de gaminivel • Petición de obtención de insignias
SALIDAS	<ul style="list-style-type: none"> • Valor del atributo de gamipuntos de dicho usuario • Valor del atributo de gamimonedas de dicho usuario • Valor del atributo de gamiexperiencia de dicho usuario • Valor del atributo de gaminivel de dicho usuario • Contenido del atributo de insignias de dicho usuario
RESTRICCIONES	Ninguna

3.1.3. Repositorio general de preguntas

Código de requisito	RF003
Nombre de requisito	Repositorio general de preguntas
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>

DESCRIPCIÓN	Las preguntas agregadas por los docentes se almacenarán en la base de datos, para su posterior uso en diferentes módulos del aplicativo. Esta tabla contenedora de preguntas podrá ser accedida por cualquier docente que imparta cualquier asignatura. Esto último, con el fin de importar alguna pregunta de interés e incluirla en algún test que dicho docente esté creando (Revisar el requisito RF004.2).
PROCESO	Cada vez que un docente cree una nueva pregunta, esta se añadirá al repositorio general de preguntas.
ENTRADAS	Pregunta creada por el docente.
SALIDAS	Pregunta agregada al repositorio general de preguntas.
RESTRICCIONES	Ninguna

3.1.4. Docente

Código de requisito	RF004
Nombre de requisito	Funcionalidades asociadas al docente
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	La plataforma web del aplicativo, será el portal en la que los docentes podrán realizar todas las acciones de gestión, estas se resumen a continuación: crear una clase, invitar o eliminar estudiantes a una clase, crear tests, ver resultados de dichos tests/exportar los mismos, administrar las clases que crearon y ver los perfiles de los estudiantes de su clase.
PROCESO	En principio, el sitio web de escritorio es el lugar idóneo para realizar todas las operaciones correspondientes a la gestión docente. A partir del inicio de sesión de los docentes, estos podrán realizar las acciones descritas previamente.
ENTRADAS	Inicio de sesión del docente y operaciones asociadas al mismo

SALIDAS	Resultado de la operación ejecutada por el docente
RESTRICCIONES	Ninguna

3.1.4.1. Creación de una clase

Código de requisito	RF004.1
Nombre de requisito	Creación de una clase
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	Un docente podrá crear clases en las que los estudiantes se podrán unir a través de un código único, dichas clases en el aplicativo, se definen como una instancia de las clases registradas en la universidad. Este espacio sirve para poder agrupar a los estudiantes que estén matriculados a dicha clase y así poder trasladarlos a la aplicación, en la cual los mismos podrán resolver los tests asignados por el docente.
PROCESO	<p>Para crear una clase, el docente deberá de darle al botón de “Crear clase”, en la cual se le presentará una lista de opciones, de las que deberá de seleccionar en el siguiente orden:</p> <ol style="list-style-type: none"> 1. Escoger carrera 2. Escoger semestre 3. Escoger asignatura <p>Luego de determinar la asignatura de la clase a crear, el docente deberá de darle un nombre descriptivo a dicha instancia de la clase. Por ejemplo: “Sistemas Expertos (abril – sep., 2022). Por último, se generará un código único de invitación a la clase, el cual el docente deberá de compartir hacia sus estudiantes, para que estos puedan unirse a la misma. Cabe recalcar, que cada vez que un estudiante se una a una clase, el docente deberá de confirmar que dicho estudiante corresponde a la clase en cuestión, y permitir o denegarle el acceso dependiendo del caso.</p>
ENTRADAS	Acción de crear clase

SALIDAS	Clase creada y código de invitación único creado
RESTRICCIONES	Ninguna

3.1.4.2. Creación de tests

Código de requisito	RF004.2
Nombre de requisito	Creación de tests
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	Dentro de una clase, el docente podrá crear tests para ser asignados a sus estudiantes, y que estos puedan resolverlos bajo los parámetros definidos por dicho docente. Habrá dos opciones disponibles para agregar las preguntas a un test a crear: crear nueva pregunta desde cero, o importar una pregunta del repositorio general de preguntas.
PROCESO	<p>Para poder crear un test, el docente deberá de acceder a la clase correspondiente, y darle al botón de “Crear nuevo test”, una vez en esta instancia, deberá de darle al botón de “Crear nueva pregunta”, en la que se le presentará una serie de campos por llenar. Dichos campos se resumen en la siguiente lista:</p> <ol style="list-style-type: none"> 1. Seleccionar alguno de los formatos de preguntas disponibles (trivia tradicional o trivia con imagen adjunta). 2. Redactar el enunciado de la pregunta 3. Agregar y redactar al menos dos respuestas posibles. 4. Seleccionar la respuesta correcta, o más de una en caso de respuestas múltiples. 5. Establecer la dificultad estimada para dicha pregunta (fácil, media, difícil o muy difícil) 6. Establecer el tiempo máximo para responder la pregunta. El tiempo por defecto/recomendado, dependerá de la dificultad seleccionada. 7. Redactar un pequeño feedback acerca de la respuesta correcta [opcional]. 8. Permitir o denegar el uso de “powerups”, esta opción estará desactivada por defecto.

	En caso de que el docente quiera ver las preguntas existentes en el repositorio general de preguntas, en lugar de darle al botón de “Crear nueva pregunta”, deberá de darle clic al botón de “Importar pregunta”, en la que él mismo, podrá filtrar la búsqueda de preguntas existentes en el repositorio general a través de la selección de la carrera, el semestre y por último la asignatura de la que desea importar dicha pregunta. Si el docente altera algún campo de dicha pregunta importada, se le mostrará la opción de si desea agregar dicha pregunta alterada, como una nueva pregunta al repositorio general de preguntas.
ENTRADAS	Acción de crear nuevo test
SALIDAS	Test creado con los parámetros establecidos.
RESTRICCIONES	Ninguna

3.1.4.3. Asignación de un test

Código de requisito	RF004.3
Nombre de requisito	Asignación de un test
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	Los docentes podrán asignar el test creado a los estudiantes de la clase correspondiente, este deberá de determinar la fecha máxima en la que puede ser accedido por el estudiante.
PROCESO	Una vez llenado todos los campos de preguntas, y que el test contenga más de 4 preguntas, el docente tendrá que establecer la fecha y hora máxima en la que el estudiante podrá acceder a dicho test y resolverlo.
ENTRADAS	Test creado.
SALIDAS	Test asignado a los estudiantes de la clase correspondiente.

RESTRICCIONES	Ninguna
---------------	---------

3.1.4.4. Resultados de los tests asignados

Código de requisito	RF004.4
Nombre de requisito	Resultados de los tests asignados
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	Los resultados de los tests que el docente asigne a sus estudiantes, se podrán exportar a diferentes formatos, como PDF o PNG. Asimismo, se mapearán los puntajes obtenidos por los estudiantes al sistema de puntaje de la universidad en cuestión (puntaje sobre 10).
PROCESO	Cada vez que un estudiante finalice un test, la tabla de resultados obtenidos se actualizará con el puntaje obtenido por dicho estudiante. El docente podrá exportar los resultados a diferentes formatos, una vez finalice el tiempo máximo asignado para dicho test. En la ventana en la que se visualiza el registro de puntuaciones alcanzadas, también se mostrará la opción de mapear dichas puntuaciones al sistema de puntuaciones de la universidad, en la que el docente será capaz de asignar dichos puntajes al sistema de notas de los alumnos si así lo desea.
ENTRADAS	Test finalizado por los estudiantes.
SALIDAS	Reporte final de puntuaciones.
RESTRICCIONES	Ninguna

3.1.4.5. Gestionar clases

Código de requisito	RF004.5
Nombre de requisito	Gestionar clases

Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>
DESCRIPCIÓN	El docente podrá administrar las clases que haya creado, estas opciones de gestión se resumen a continuación: ver clases que imparte, editar el nombre de la misma o eliminar dicha clase, eliminar a algún estudiante de la clase, eliminar o editar pregunta agregada al repositorio general de preguntas.
PROCESO	En el panel principal de gestión docente, se le presentará la lista de clases que imparte, para editar el nombre de la misma o eliminar dicha clase, tendrá que darle al botón de editar. Para ver más detalles de una clase en concreto, bastará con darle clic a la clase en cuestión, en la que se le presentará la lista de estudiantes unidos a dicha clase, y también se mostrará la opción para expulsar a un estudiante en concreto. Por último, para editar o eliminar una pregunta que dicho docente haya agregado al repositorio general de preguntas, este podrá hacerlo a través de la ventana de "Mis Preguntas", en la que se le mostrará todas las preguntas que haya creado para dicha clase.
ENTRADAS	Opción de gestión docente.
SALIDAS	Resultado de ejecución de la opción de gestión docente realizada.
RESTRICCIONES	Ninguna

3.1.5. Experiencia Personalizada

Código de requisito	RF005
Nombre de requisito	Experiencia personalizada
Tipo	Requisitos de producto <input checked="" type="radio"/> Requisitos de proyecto <input type="radio"/>
Prioridad del requisito	Alta/Esencial <input checked="" type="radio"/> Media/Deseado <input type="radio"/> Baja/Opcional <input type="radio"/>

DESCRIPCIÓN	La aplicación constará de un módulo basado en Inteligencia Artificial, el cual estará enfocado en ofrecer contenido personalizado a los estudiantes. Este se enfocará principalmente en la inclusión de preguntas en los próximos tests de los modos de juego libre de que el estudiante realice, estas estarán relacionadas a otras preguntas que el estudiante haya respondido de forma errónea en tests anteriores.
PROCESO	A través de la implementación de un algoritmo NLP (Natural Language Processing) para la asociación de oraciones similares, más específicamente, del tipo “sentence transformers”, el sistema tendrá que ser capaz de relacionar las nuevas preguntas que se ingresen con las existentes en el “repositorio general de preguntas” de acuerdo a las particularidades establecidas.
ENTRADAS	Modelo NLP entrenado
SALIDAS	Relaciones encontradas entre preguntas.
RESTRICCIONES	Ninguna

3.2. Requerimientos no funcionales

3.2.1. Requerimientos de rendimiento (Eficiencia)

Los equipos en los que se vaya a ejecutar el sistema deben de tener como mínimo:

- RAM: clientes 1 GB y 2 GB mínimo en el servidor.
- Resolución de pantalla: 800 x 480 píxeles
- Conexión a internet

3.2.2. Restricciones de Diseño

Al ser un producto multiplataforma, se diseñará tanto para pantallas de móviles y tabletas como para monitores de computadoras, por lo que la apariencia del sistema deberá ajustarse a estas resoluciones.

3.2.3. Estándares Asociados

El sistema aplicará el estilo de escritura CamelCase, de tipo lowerCamelCase, en la que los nombres de cada de parte del sistema empezará la primera letra con minúscula.

Ejemplo: obtenerGamipuntos()

3.2.4. Atributos del sistema

- Seguridad: El ingreso al sistema se contralará por medio de las credenciales institucionales.
- Portabilidad: Al iniciar sesión con las credenciales institucionales se podrá acceder desde cualquier otro dispositivo y obtener los avances previos.

3.2.5. Requerimientos de Desarrollo

Ya que el proyecto va a ser implementado en los servidores de la universidad, en la cual existen soluciones del lado del servidor construidas bajo la tecnología “Dot Net” de Microsoft, por conveniencia el Backend del aplicativo se tendría que desarrollar bajo dicho framework. Cabe recalcar, que para el desarrollo del frontend y el entorno multiplataforma, se pretende hacer uso de los frameworks “Angular” y “Flutter”.

ANEXO 4. INFORME DE DIAGRAMAS ESTRUCTURALES UML

INFORME DE DIAGRAMAS ESTRUCTURALES UML

Proyecto: Gamivity

Autores: Dany Álava – Carlos Saavedra

Índice

Índice 73

Introducción.....	74
Desarrollo.....	75
Diagrama de caso de uso.....	75
Diagrama de clases.....	77
Diagrama de paquete.....	78
Conclusiones.....	79
Bibliografía.....	80

Introducción

El concepto de UML ha tomado diversas definiciones a lo largo del tiempo en lo que a desarrollo de software se refiere. Autores como Grau et al., (2018) lo describen como un lenguaje de modelado de sistemas que permite a los desarrolladores de software visualizar, especificar y documentar aspectos del comportamiento de un sistema. Ahora bien, en el contexto de una tesis, UML puede ser utilizado para modelar el comportamiento de un sistema que está siendo estudiado en la fase del planteamiento del problema. Esto puede ser útil para comprender mejor el sistema y también puede proporcionar una base para el desarrollo de una implementación del mismo. La importancia de los diagramas UML en el desarrollo de proyectos de tesis radica en su capacidad para representar de forma gráfica y comprender de mejor manera cómo funcionan sus componentes y de qué forma interactúan entre ellos. A continuación, se presentan los diagramas UML considerados para explicar el backend de “Gamivity”.

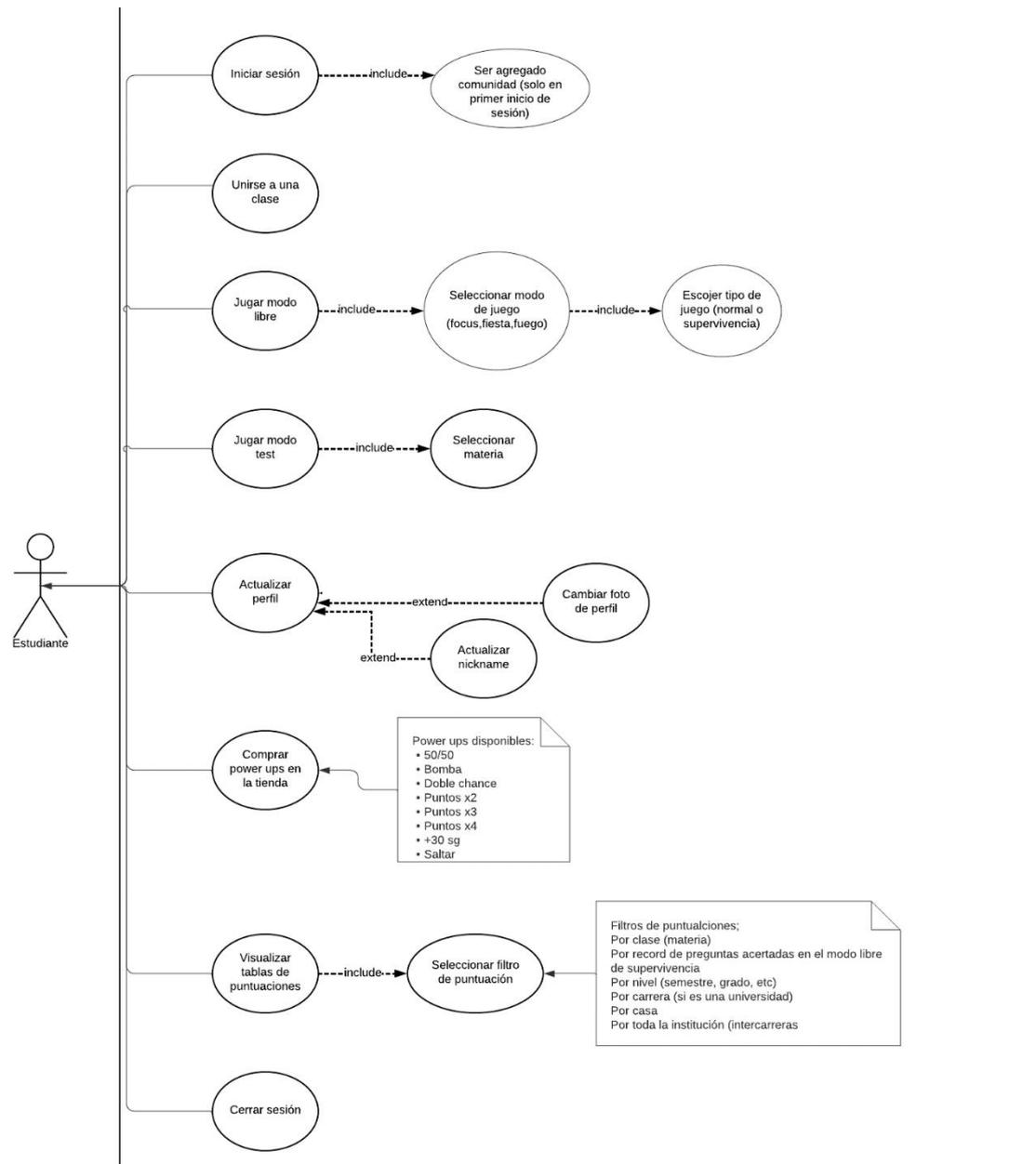


Diagrama 2. Funciones del Estudiante "Gamivity"

Diagrama de clases

El diagrama de clase fue implementado para tener claro la estructura del desarrollo del proyecto, ya que este describe los tipos de objetos de un sistema, así como los distintos tipos de relaciones que puedan existir entre ellos. ya que de esta manera se refleja la características y funcionalidades de los diferentes usuarios; Administrados, Estudiante, Profesor, dentro de la aplicación “Gamivity”. En la siguiente figura se muestra la representación de un diagrama de clase desde la perspectiva de los diferentes tipos de usuarios que existen en el aplicativo.

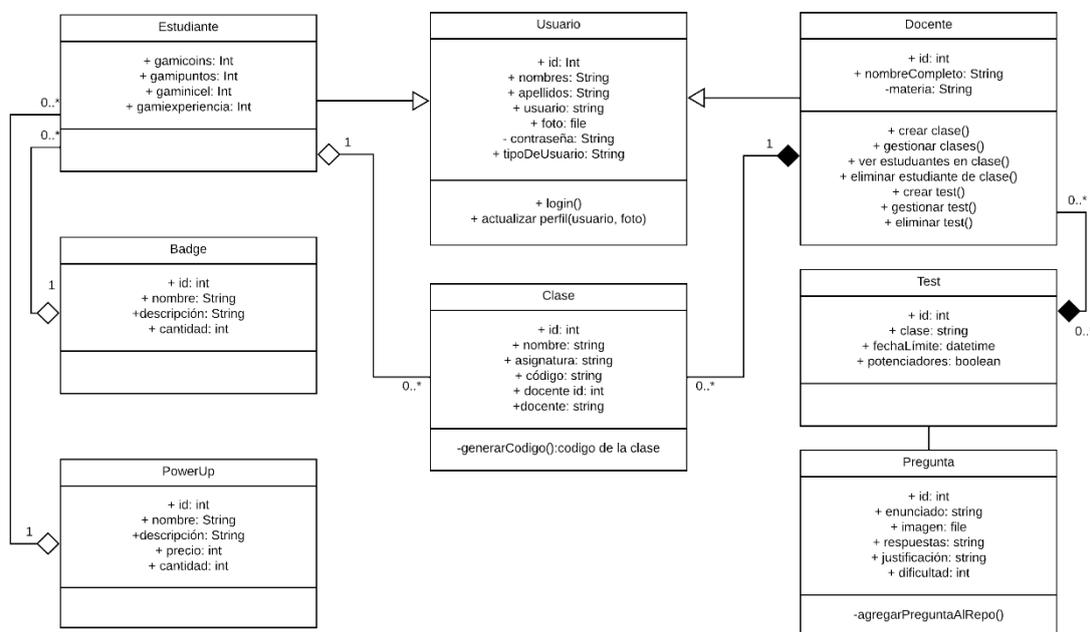


Diagrama 3. Distribución de clases "Gamivity"

Diagrama de paquete

Este diagrama se presenta como una especificación del diagrama de clases. El backend de “Gamivity” se divide en clases como ya se explicó anteriormente, pero a su vez, se debe detallar las interacciones de los paquetes. Además de mostrar gráficamente cómo dentro de cada paquete, una o más clases se pueden relacionar de acuerdo a las conexiones propuestas previamente, y de acuerdo a esto, identificar su relación de dependencia dentro de un nivel de abstracción más específico, y sin la necesidad de presenciar atributos y métodos.

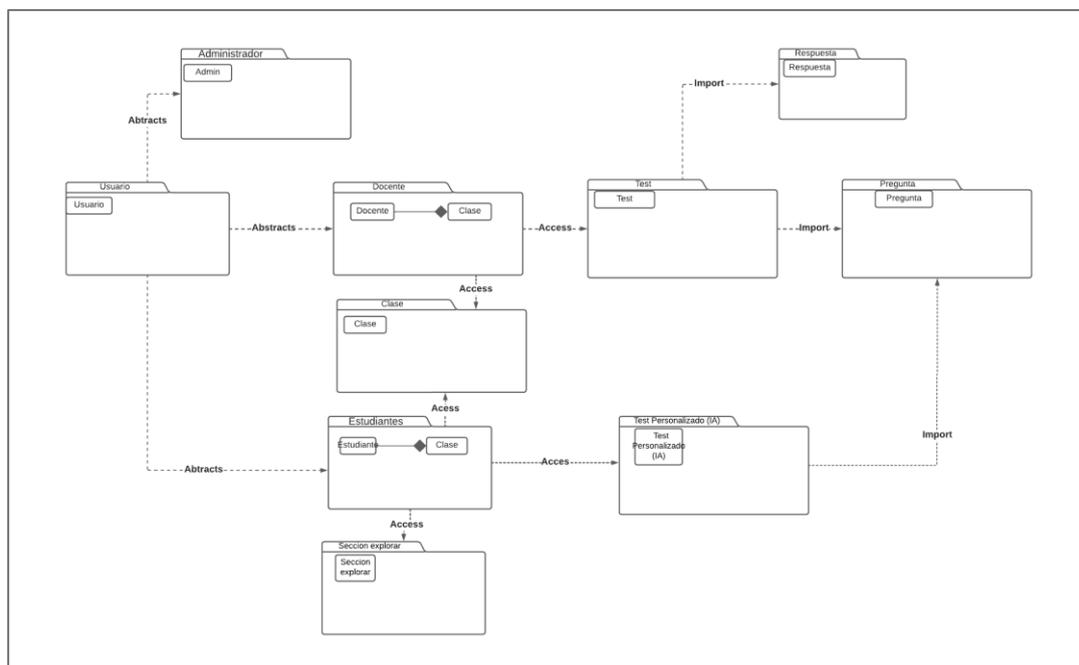


Diagrama 4. Diagrama de paquetes "Gamivity"

Conclusiones

Teniendo en cuenta lo descrito en este documento, se llega a las siguientes conclusiones:

- Los diagramas UML son una herramienta útil para la planificación y el diseño de proyectos de tesis. Estos pueden ayudar a los estudiantes a comprender mejor el proceso de investigación y el alcance del trabajo requerido. Además, estos diagramas pueden ser útiles para el seguimiento del progreso del proyecto y la identificación de posibles problemas.
- Entendiendo que la gamificación es el centro de “Gamivity”, los diagramas UML resultan un gran método para la creación de aplicaciones de minijuegos gamificados. Esto colabora a que los involucrados comprendan de mejor forma el código y hacer que el desarrollo de la aplicación y de los minijuegos sea más eficiente y guiada hacia las necesidades del cliente y de los usuarios.
- Para terminar, los diagramas UML son importantes porque permiten al usuario representar de forma gráfica y comprensible la estructura de un sistema, así como su comportamiento. Esto es útil tanto para el desarrollador que crea el sistema, como para el usuario que lo utilizará.

Bibliografía

Grau, X. F., Isabel, M., & Segura, S. (2018). *Desarrollo Orientado a Objetos con UML*.

Silva, A. E., Pérez, E. G. L., Abraham, J., Peña, C., & Alicia, G. (2019). *COMPARACIÓN ESTADÍSTICA DE LA RELACIÓN DE DEPENDENCIA DE USO DEL UML DENTRO DEL SECTOR EMPRESARIAL Y EDUCATIVO*.

ANEXO 5. INFORME DE CREACIÓN DE BASE DE DATOS

INFORME DE CREACIÓN DE BASE DE DATOS

Proyecto: Gamivity

Autores: Dany Álava – Carlos Saavedra

Índice

Índice	2
Introducción	3
Desarrollo	4
Creación del modelo.....	4
Crear un contexto	5
Lectura y escritura de datos	6
Migración.....	6
Diagrama de base de datos.....	7
Conclusión	8
Bibliografía	9

Introducción

Cada día que transcurre en todo el mundo, las personas generan millones de datos, ya sea realizando tareas o actividades como al usar alguna aplicación de compras, ver nuestras series favoritas en alguna plataforma de streaming, al momento de redactar alguna publicación o al pedir un medio de transporte mediante nuestro teléfono celular. Pues en todos estos casos existen programas que registran dicha información, dado que estos datos son muy útiles y pueden llegar a ser analizados para detectar algún tipo de patrón, ya sea de compra, interacción social y de movilidad. Es por ello que lo más probable es que esta información se almacene en una base de datos (Villafán & Pérez, 2021).

Las bases de datos son el método preferido para el almacenamiento estructurado de datos, ya que se utilizan para asegurar la integridad de los datos, además estas permiten tener un buen control y manejo de información. Esto conlleva a la realización de un sistema de información, mismo que ayuda a interactuar de manera sincrónica diversos datos, donde éstos se relacionan entre sí, arrojando resultados deseados y apoyando de esta manera las actividades desarrolladas en una empresa (Salazar, 2020).

Es por ello que el objetivo de este trabajo consiste en evidenciar el proceso de creación de la base de datos que se tuvo que hacer para que pueda ser utilizada por el sistema Gamivity.

Desarrollo

Para la creación de la base de datos se optó por el uso de la metodología “Code First”, la cual consiste en el desarrollo de aplicaciones donde no se tiene una base de datos creada en primera instancia, sino que esta misma se crea a partir de un modelo que es una clase, cuyo nombre sería el nombre de la tabla y dentro de esta clase estarían las propiedades que vendrían siendo los campos en la base de datos, una vez se tiene la clase se crea la base de datos (Microsoft, 2010).

A continuación, se detallan cada una de las fases que se siguieron para el desarrollo de la base de datos siguiendo la metodología mencionada anteriormente.

Creación del modelo

Se definió un modelo usando clases, para ello se debió de crear cada una de las entidades del sistema, incluyendo sus respectivas propiedades y relaciones con las demás entidades. A continuación, se muestra un ejemplo de una de las entidades del sistema, en este caso la entidad “Student”, la cual hace referencia al actor “Estudiante” la cual deriva de la clase base “AppUser”.

```
namespace API.Entities
{
    21 references
    public class Student : AppUser
    {
        0 references
        public int CurrentLevel { get; set; }
        0 references
        public int Gamipoints { get; set; }
        0 references
        public int Gamixp { get; set; }
        0 references
        public int Gamilevel { get; set; }
        0 references
        public int Gamicoins { get; set; }

        5 references
        public ICollection<StudentClass> StudentClasses { get; set; }
        1 reference
        public ICollection<StudentPowerUpCount> StudentPowerUpsCount { get; set; }
        1 reference
        public ICollection<StudentBadge> StudentBadges { get; set; }

        1 reference
        public ICollection<StudentItem> StudentItems { get; set; }

        0 references
        public ICollection<PersonalizedTest> PersonalizedTests { get; set; }
    }
}
```

Figura 13. Muestra de la entidad Student (Estudiante)

Crear un contexto

Ahora es momento de definir un contexto derivado, el cual represente una sesión con la base de datos y que esta nos permita consultar y guardar datos en la misma. Se define un contexto que deriva de `System.Data.Entity.DbContext` y expone un `DbSet<"Nombre de la entidad">` escrito para cada clase del modelo. Cabe recalcar para este punto se debe tener agregado el paquete `EntityFramework`.

```
services.AddDbContext<DataContext>(options =>
{
    options.UseSqlServer(config.GetConnectionString("DefaultConnection"));
});
```

Figura 14. Agregación y definición del servicio `DbContext`

```
24 references
public class DataContext : IdentityDbContext<AppUser, AppRole, int, IdentityUserClaim<int>, AppUserRole,
IdentityUserLogin<int>, IdentityRoleClaim<int>, IdentityUserToken<int>>
{
    0 references
    public DataContext(DbContextOptions options) : base(options)
    {
    }

    7 references
    public DbSet<Student> Students { get; set; }
    4 references
    public DbSet<Teacher> Teachers { get; set; }
    6 references
    public DbSet<GeneralClass> GeneralClasses { get; set; }
    3 references
    public DbSet<Class> Classes { get; set; }
    2 references
    public DbSet<StudentClass> StudentClasses { get; set; }
    1 reference
    public DbSet<Test> Tests { get; set; }
    4 references
    public DbSet<BaseQuestion> BaseQuestions { get; set; }

    1 reference
    protected override void OnModelCreating(ModelBuilder builder)
    {
        base.OnModelCreating(builder);

        builder.Entity<Student>().ToTable("Students");
        builder.Entity<Teacher>().ToTable("Teachers");

        builder.Entity<AppUser>()
            .HasMany(ur => ur.UserRoles)
            .WithOne(u => u.User)
            .HasForeignKey(ur => ur.UserId)
            .IsRequired();
    }
}
```

Figura 15. Fragmento de la clase derivada de `DbContext`

Lectura y escritura de datos

En esta fase se deben generar todas las consultas que se realizarán a la base de datos, a través del uso de LINQ (Consulta Integrada en el Lenguaje). En las Figuras 4 y 5 se puede apreciar dos ejemplos de consulta a la base de datos, una de lectura y otra de escritura.

```

public async Task<Teacher> GetTeacherWithClassesAsync(int id)
{
    return await _context.Teachers.Include(c => c.Classes).SingleOrDefaultAsync(t => t.Id == id);
}

```

Figura 16. Consulta de lectura

```

0 references
public async Task<bool> SaveAllAsync()
{
    return await _context.SaveChangesAsync() > 0;
}

```

Figura 17. Consulta de escritura

Migración

Hay que tener presente que el modelo de datos cambia con frecuencia y cada vez que el modelo cambia de sincronización con la base de datos. Se puede configurar Entity Framework para quitar y volver a crear automáticamente la base de datos cada vez que cambie el modelo de datos. La característica Migraciones permite a Code First actualizar la base de datos sin quitarla ni volver a crearla. A continuación se presenta el fragmento de una migración para la creación de la tabla "GeneralClasses".

```

migrationBuilder.CreateTable(
    name: "GeneralClasses",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        ClassLevel = table.Column<int>(type: "int", nullable: false),
        GeneralClassName = table.Column<string>(type: "nvarchar(max)", nullable: true),
        Course = table.Column<string>(type: "nvarchar(max)", nullable: true)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_GeneralClasses", x => x.Id);
    });

```

Figura 18. Fragmento de una migración

Diagrama de base de datos

Una vez teniendo nuestra base de datos migrada, podemos trabajar directamente desde SQL Server Management Studio y desde aquí poder generar automáticamente el modelo de base de datos para poder apreciar de mejor forma las relaciones de todas las entidades de nuestro sistema.

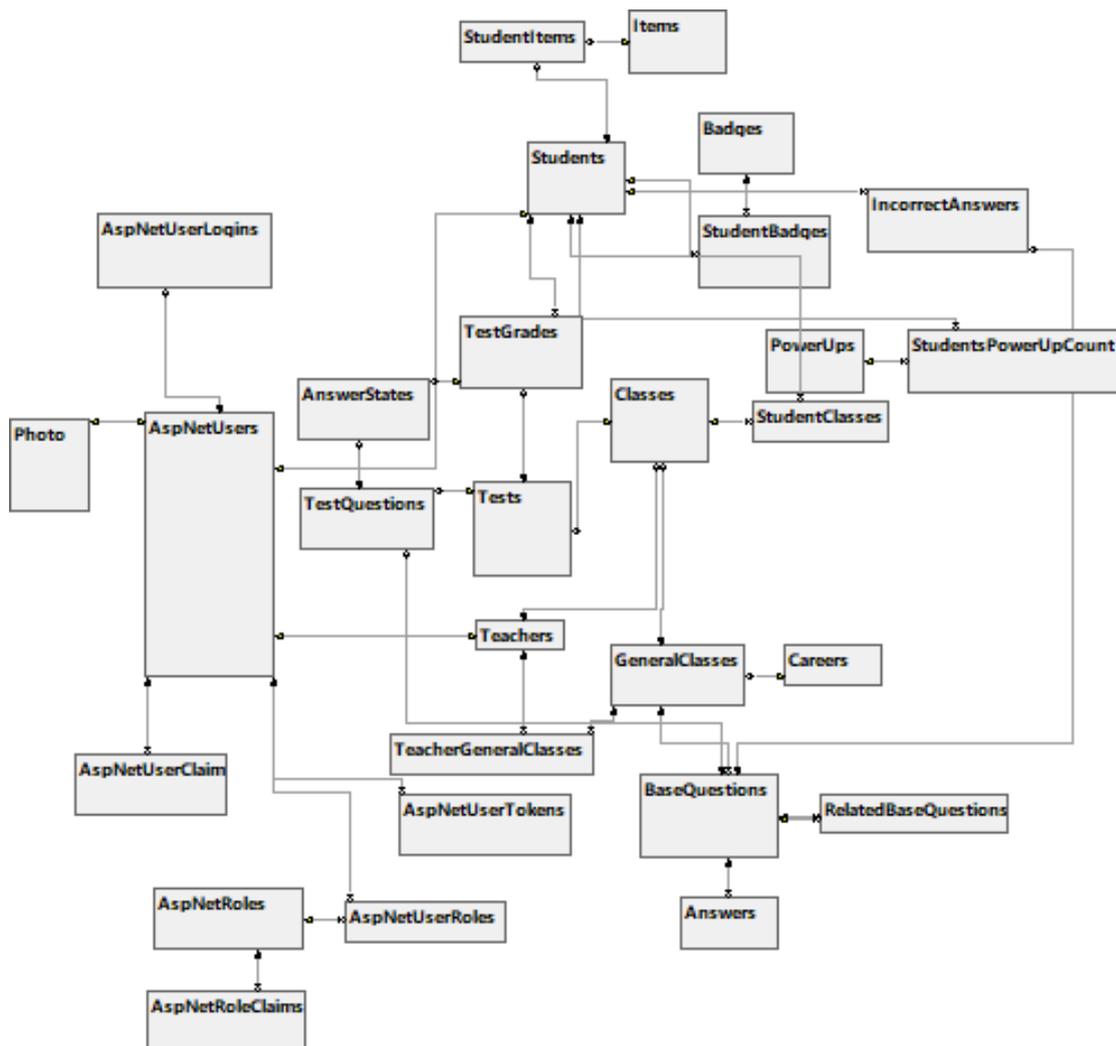


Figura 19. Diagrama de base de datos

Conclusión

Si bien el almacenamiento de la información debe ser ágil y confiable, el sistema no solo debe enfocarse en cómo almacenar y utilizar la información, sino también garantizar su confiabilidad y eficiencia, de la misma manera que verificar la fuente de la información, por lo que el sistema debe ser confiable en la búsqueda. y ágil en la velocidad con que se haga la misma.

El enfoque Code First de Entity Framework puede ser muy cómodo y ágil, ya que permite crear un modelo a partir de clases y delegarle a Entity Framework la creación automática de una base de datos. Además, permite tener un control total al manejarse todo a nivel de código.

Bibliografía

Microsoft. (14 de Octubre de 2010). *Code First to a New Database*. Obtenido de <https://docs.microsoft.com/en-us/ef/ef6/modeling/code-first/workflows/new-database>

Salazar, E. (25 de Marzo de 2020). *Investigación sobre las bases de datos*. Obtenido de <https://www.coursehero.com/file/58597738/INVESTIGACION-SOBRE-BASES-DE-DATOSdocx/>

Villafán, E., & Pérez, A. (6 de Julio de 2021). *La importancia de las bases de datos*. Obtenido de <https://www.inecol.mx/inecol/index.php/es/2013-06-05-10-34-10/17-ciencia-hoy/1426-la-importancia-de-las-bases-de-datos>

**ANEXO 6. DESARROLLO DE LOS ENDPOINTS Y LAS VISTAS
DE LOS MÓDULOS DE GESTIÓN DOCENTE**

**ENDPOINTS Y VISTAS DE LOS MÓDULOS
DE GESTIÓN DOCENTE**

Proyecto: Gamivity

Autores: Dany Álava – Carlos Saavedra

Peticiones HTTP asociadas a la gestión docente del sistema Gamivity

Métodos Post

Crear nueva clase

Para realizar dicha acción, el docente tendrá que especificar el nombre (name) de la misma, así como seleccionar la asignatura para poder determinar su respectivo ID (generalClassId). Cabe recalcar que el semestre (course) al que pertenece dicha asignatura no será solicitado al momento de crear una clase.

A partir de la ejecución de dicho endpoint, este generará un código de clase único por el que el estudiante deberá de unirse a la clase en cuestión.

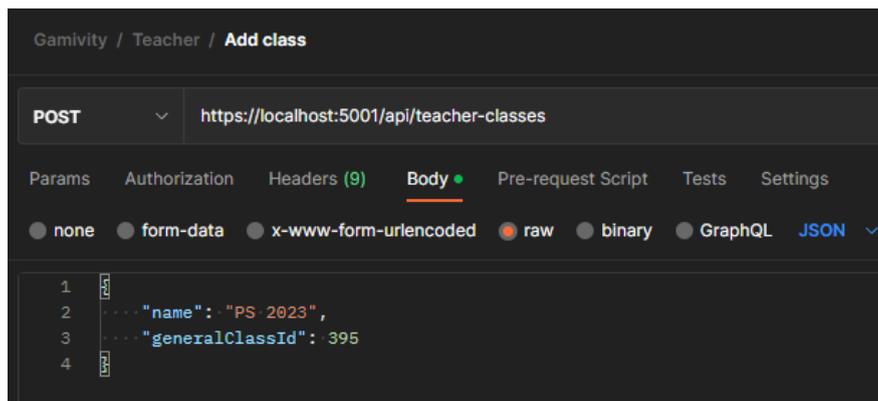


Figura 20. Endpoint para añadir una clase

Unirse a clase (estudiante)

Un estudiante autenticado, podrá unirse a una clase a través de la especificación del parámetro de consulta del código único de la clase.

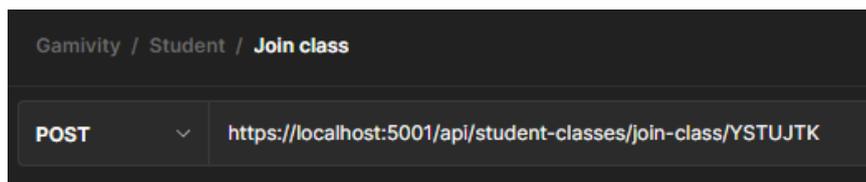
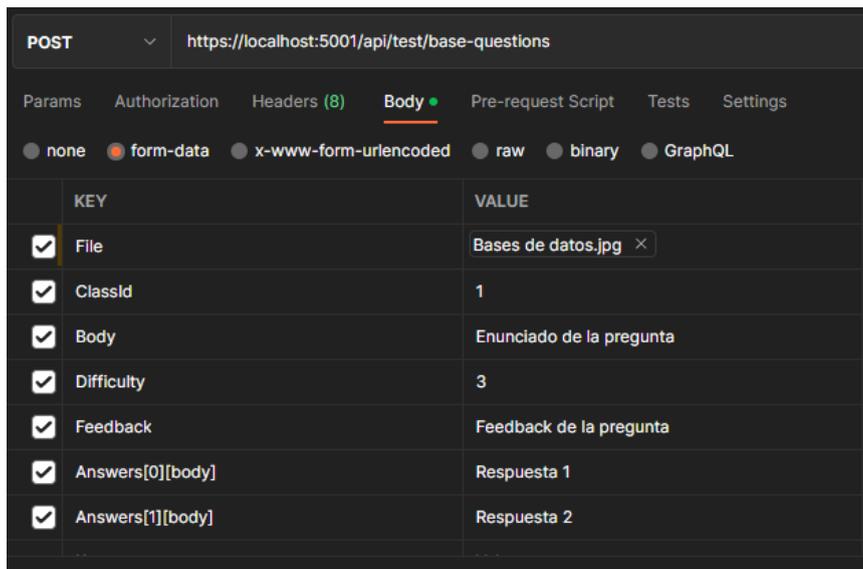


Figura 21. Endpoint para unirse a una clase

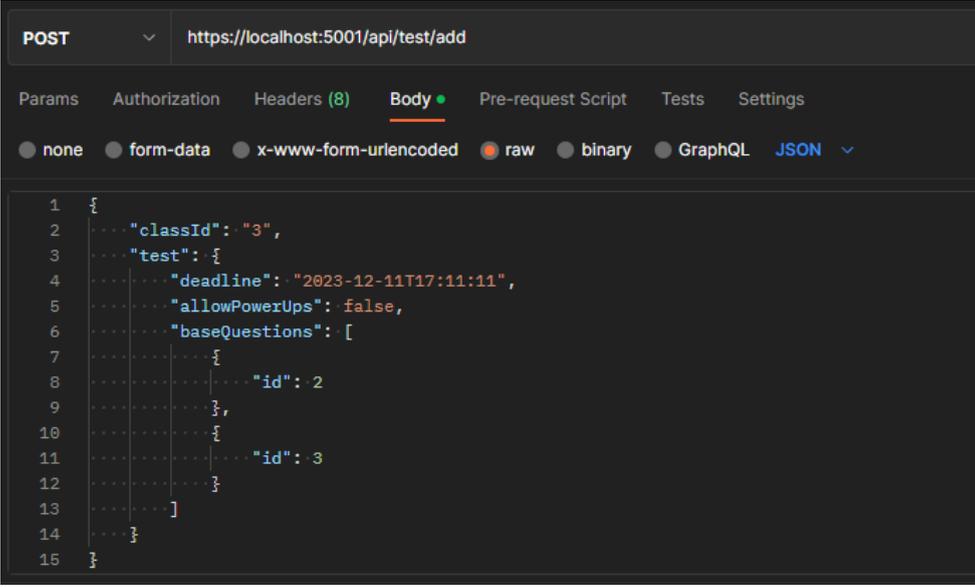
Crear un test

Para esto, el docente deberá de seleccionar a qué clase corresponderá el test, así el sistema le asignará el id correspondiente a dicha clase (*classId*), también deberá establecer la fecha de finalización (*deadline*) y definir si el estudiante puede utilizar power ups al resolver dicho test (*allowPowerUps*). Asimismo, escoger entre dos opciones, crear una pregunta desde cero, o seleccionar una ya existente en el repositorio general de preguntas. Si escoge la primera opción, se deberá ejecutar el endpoint dependiente (agregar pregunta), el cual permite definir las propiedades para cada pregunta: tiempo para responder la misma (*difficulty*), el cuerpo de la pregunta (*body*), en caso de que la pregunta amerite de adjuntar una imagen pues se cargará un archivo (*file*) y las respuestas (*answers*) y, por último, definir opcionalmente una pequeña descripción o explicación asociada a la respuesta correcta (*feedback*).



KEY	VALUE
<input checked="" type="checkbox"/> File	Bases de datos.jpg
<input checked="" type="checkbox"/> ClassId	1
<input checked="" type="checkbox"/> Body	Enunciado de la pregunta
<input checked="" type="checkbox"/> Difficulty	3
<input checked="" type="checkbox"/> Feedback	Feedback de la pregunta
<input checked="" type="checkbox"/> Answers[0][body]	Respuesta 1
<input checked="" type="checkbox"/> Answers[1][body]	Respuesta 2

Figura 22. Endpoint para agregar una pregunta



```

POST https://localhost:5001/api/test/add

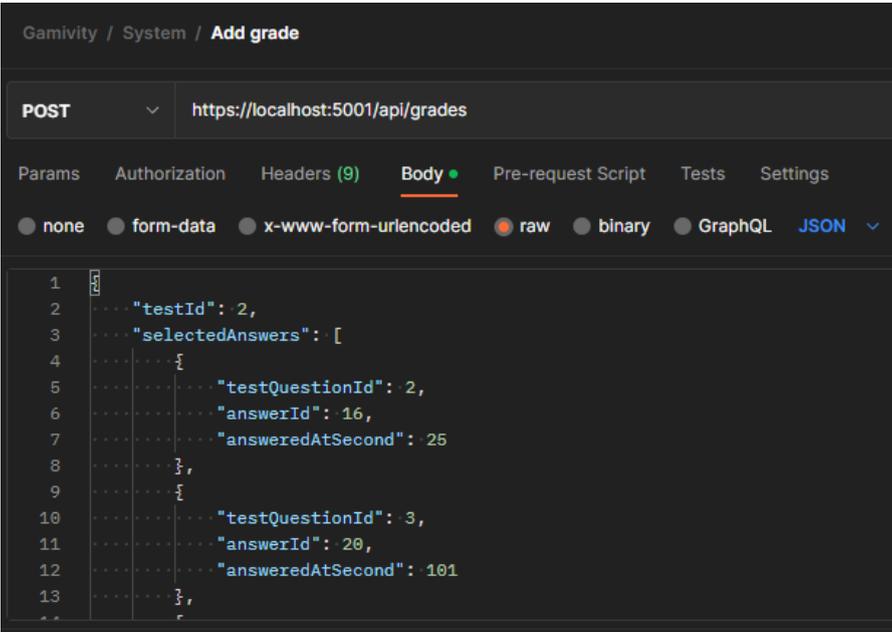
Body
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1 {
2   "classId": "3",
3   "test": {
4     "deadline": "2023-12-11T17:11:11",
5     "allowPowerUps": false,
6     "baseQuestions": [
7       {
8         "id": 2
9       },
10      {
11        "id": 3
12      }
13    ]
14  }
15 }

```

Figura 23. Endpoint para agregar un test

Asignar calificación a un test

Una vez un estudiante haya finalizado un test, se deberá registrar su calificación en el test en cuestión, es por ello que para que se realice esta acción se deberá especificar el id del test (testid), así como su respectiva nota obtenida (grade).



```

Gamivity / System / Add grade

POST https://localhost:5001/api/grades

Body
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1 {
2   "testId": 2,
3   "selectedAnswers": [
4     {
5       "testQuestionId": 2,
6       "answerId": 16,
7       "answeredAtSecond": 25
8     },
9     {
10      "testQuestionId": 3,
11      "answerId": 20,
12      "answeredAtSecond": 101
13    }
14  ]
15 }

```

Figura 24. Endpoint para asignar la calificación de un test

Iniciar sesión

Esto básicamente permite el ingreso de un usuario al sistema, deberá ingresar sus credenciales institucionales, número de cédula (*identification*) y su respectiva contraseña (*password*).

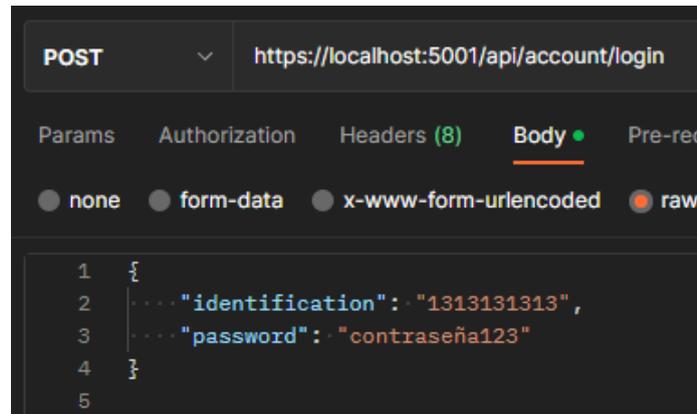


Figura 25. Endpoint para iniciar sesión

Métodos Get

Obtener clases

se recuperan todas las clases creadas por el docente junto con información de la asignatura, es decir, el nombre (*name*), el código generado de la misma (*code*), el semestre (*classLevel*), el nombre de la asignatura (*generalClassName*), así como la carrera (*course*) de la misma.

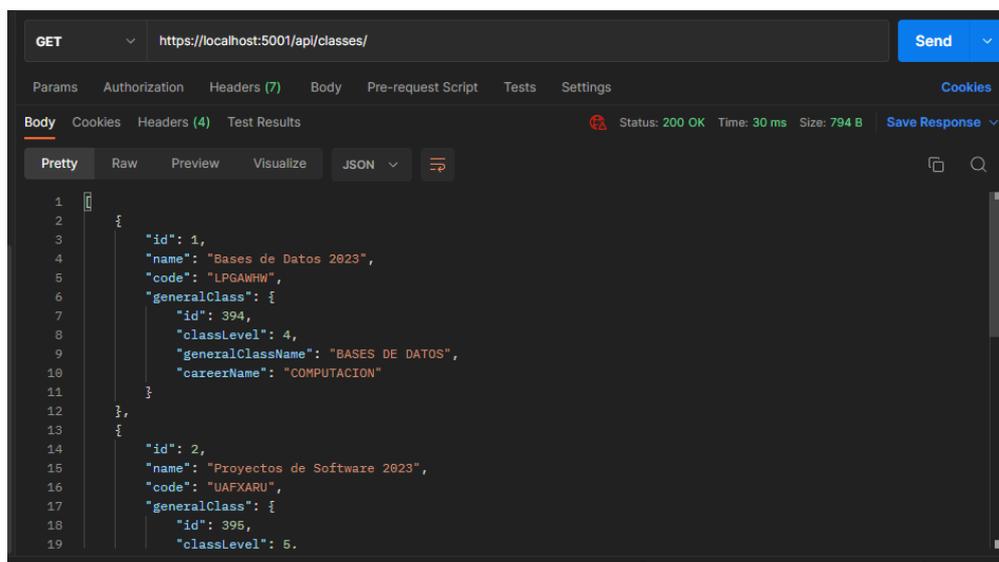


Figura 26. Endpoint para obtener las clases

Obtener estudiantes de una clase

simplemente se deberá de especificar el id de la clase como parámetro de la consulta, para así poder recuperar las propiedades públicas de los estudiantes de la clase en concreto.

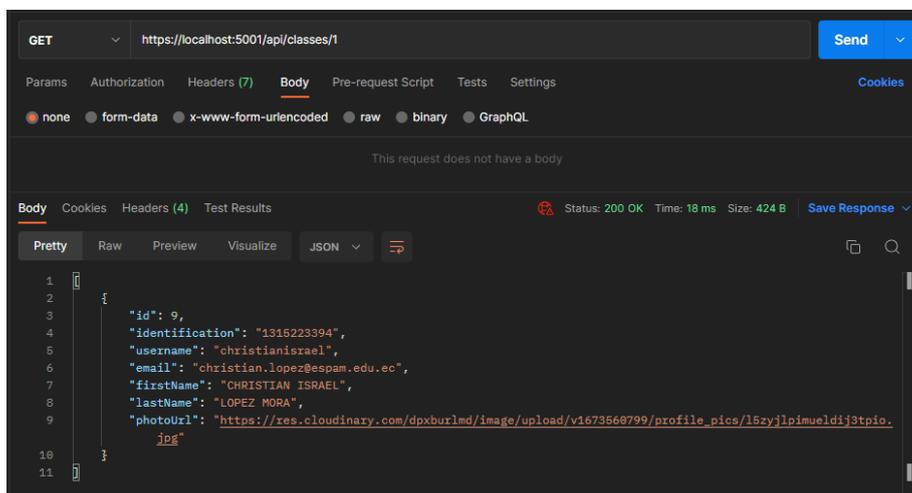


Figura 27. Endpoint para obtener estudiantes de una clase

Obtener preguntas del repositorio

Como se indicó previamente, el docente puede importar preguntas que ya existan para así poder agregarlas a un test en específico. Aquí se recuperan todas las preguntas asociadas a una asignatura; es por ello que se debe de definir los parámetros de consulta tanto del id de dicha asignatura, el número y tamaño de página, así como indicar si se quieren consultar las preguntas que el docente autenticado haya creado o simplemente todas las de dicha asignatura.

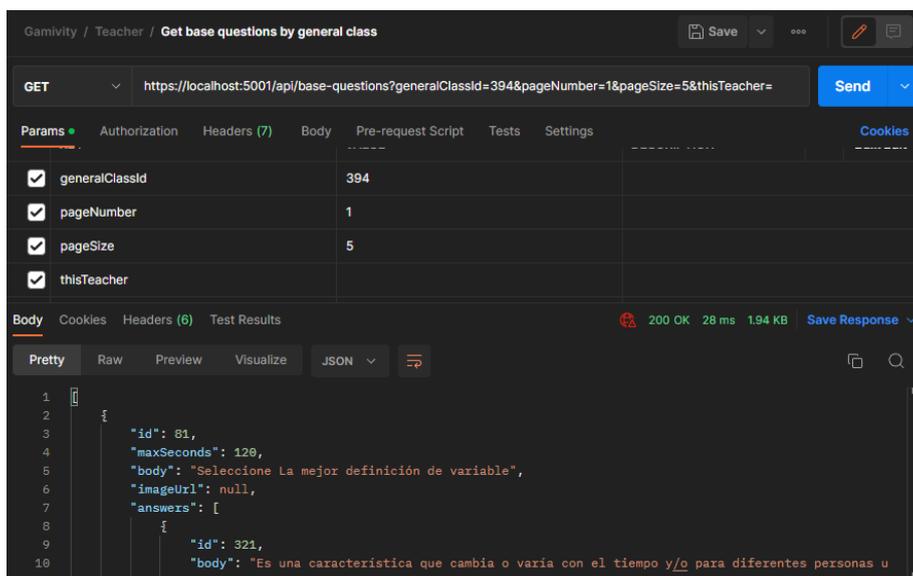


Figura 28. Endpoint para obtener las preguntas de una asignatura

Obtener calificaciones de los estudiantes

Para que el docente pueda recuperar las notas de los estudiantes de un test asignado y finalizado, se deberá de especificar como parámetro de consulta el id del test en cuestión. La respuesta de este endpoint está formada por las propiedades públicas de los estudiantes, nombre (`studentFirstName`) y apellido (`studentLastName`), junto a su respectiva calificación (`grade`).

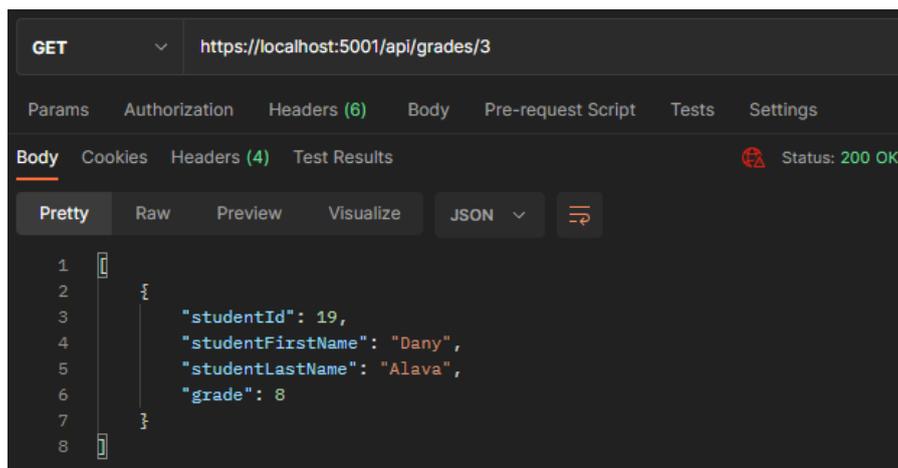


Figura 29. Endpoint para obtener las calificaciones de un test

Métodos Delete

Expulsar a un estudiante de una clase

Para ello, se deberá de especificar el id del estudiante (`studentId`), así como el id de la clase a la que se le quiere aplicar dicha acción (`className`).

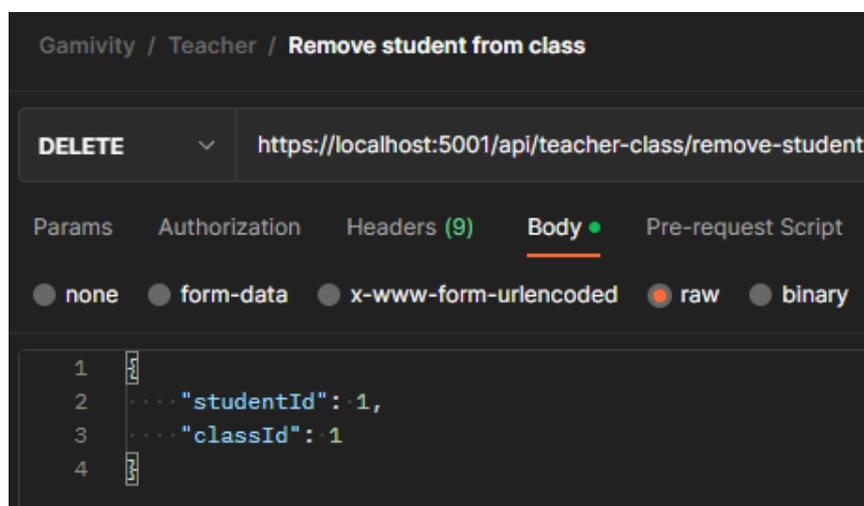


Figura 30. Endpoint para expulsar estudiante de una clase

Eliminar un test de una clase

Para poder realizar esta acción, únicamente hace falta especificar como parámetro de consulta el ID del test que se desea eliminar.

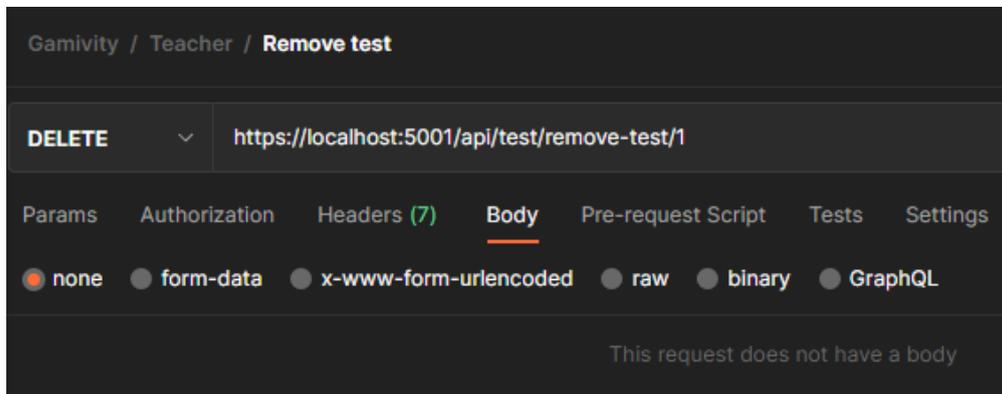


Figura 31. Endpoint para eliminar un test

Métodos Put

Editar clase

Este método es el que permite editar una clase, recibiendo el id de la clase que se desea editar junto con el nuevo nombre.

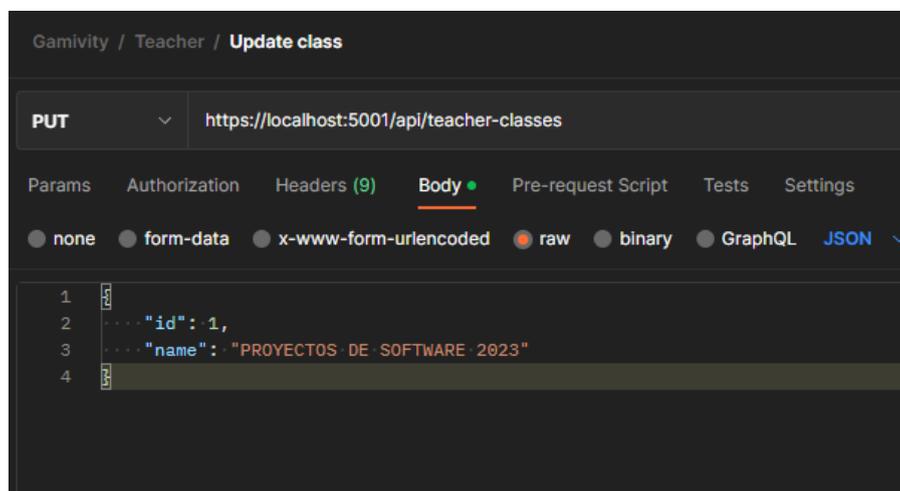


Figura 32. Endpoint para editar una clase

Editar perfil

Este método es el que permite editar los campos modificables de un usuario del sistema como lo son el nombre de usuario y la foto de perfil (avatar).

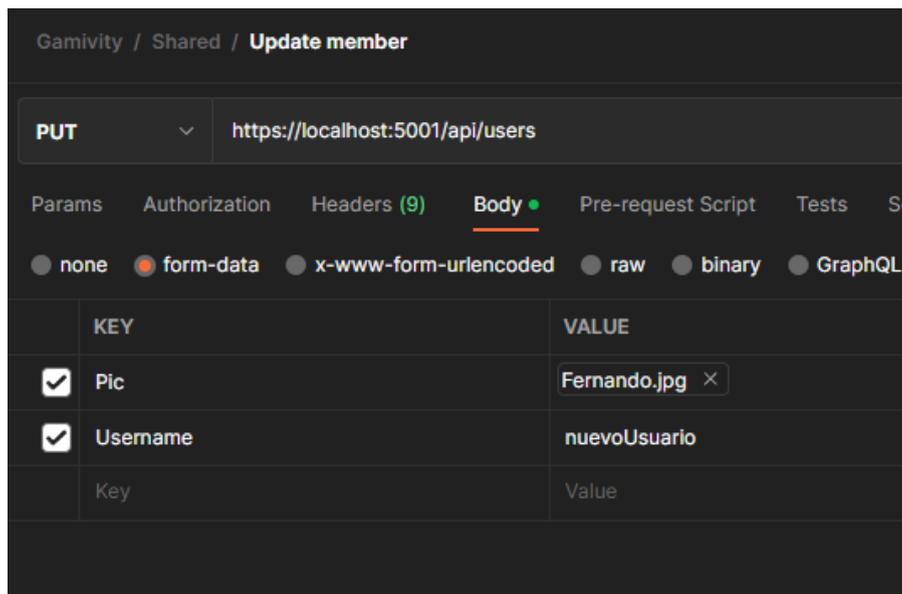


Figura 33. Endpoint para editar un perfil

Editar una pregunta

Este método es el que permite editar los campos modificables de una pregunta base del sistema como lo son el enunciado (body), la imagen de referencia (imageUrl), las respuestas (answers/body), la justificación (feedback), así como la dificultad (maxSeconds).

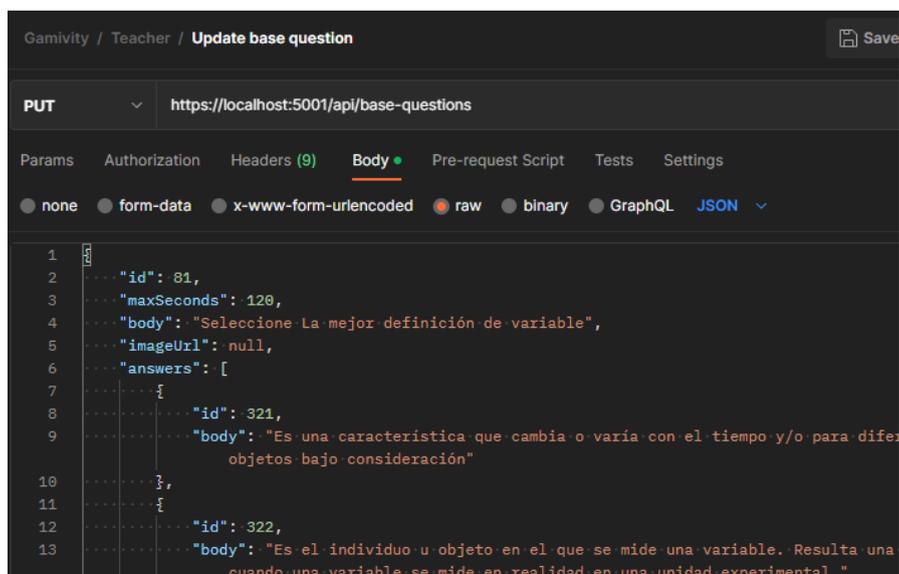


Figura 34. Endpoint para editar una pregunta

Vistas asociadas a la gestión docente

Inicio de sesión

Esta es la vista inicial que se le presentaría a un usuario que ingrese al sistema por primera vez o que no haya iniciado sesión.

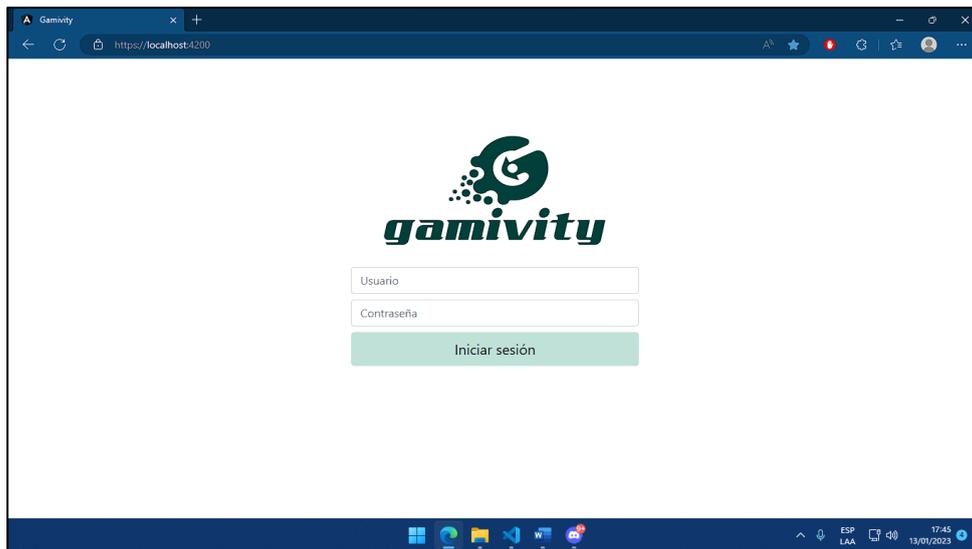


Figura 35. Vista del login

Inicio

Una vez el usuario haya iniciado sesión se lo redireccionará al inicio (home) de la administración docente del sistema Gamivity, en este se mostrarán en forma de tarjetas las clases que el docente haya creado.

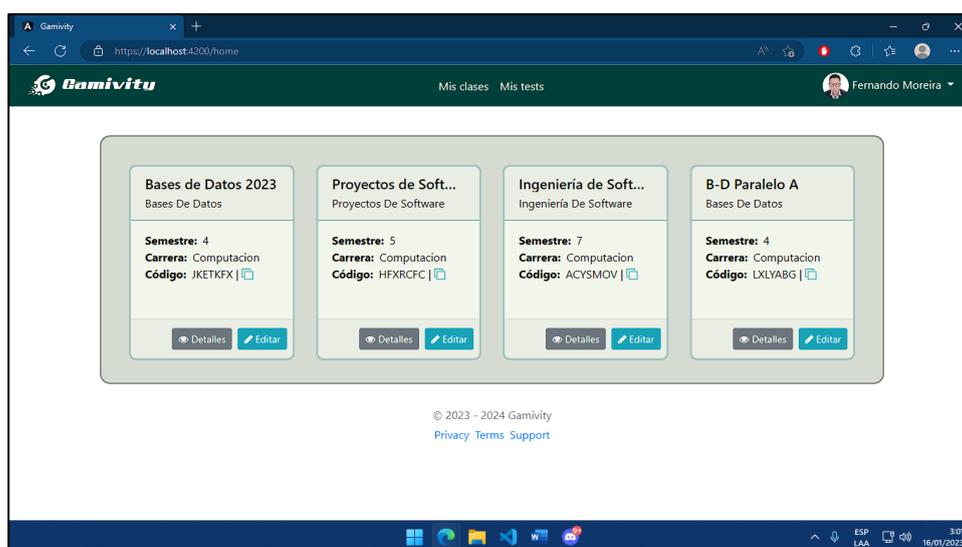


Figura 36. Vista del home

Crear clase

En este módulo se le permitirá al docente crear las clases de las asignaturas correspondientes que este mismo imparta. Deberá seleccionar entre una de sus asignaturas, así como darle el nombre que desee a la clase, para finalmente darle al botón de crear y se realice la acción correspondiente.

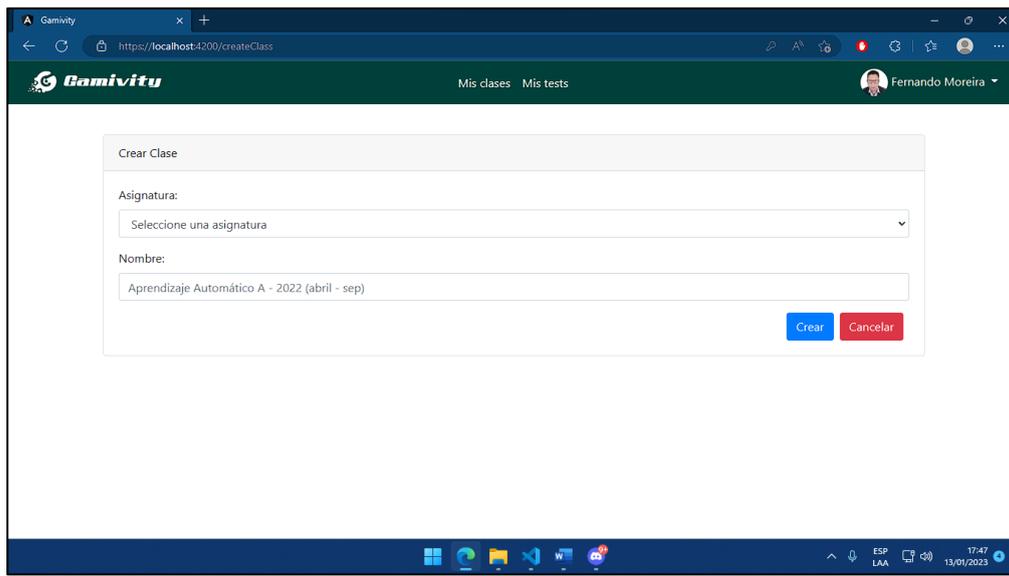


Figura 37. Vista del módulo de crear clase

Mis clases

Una vez el docente haya creado una clase, automáticamente se verá reflejada en este módulo, ya que es aquí donde se administrarán las mismas.

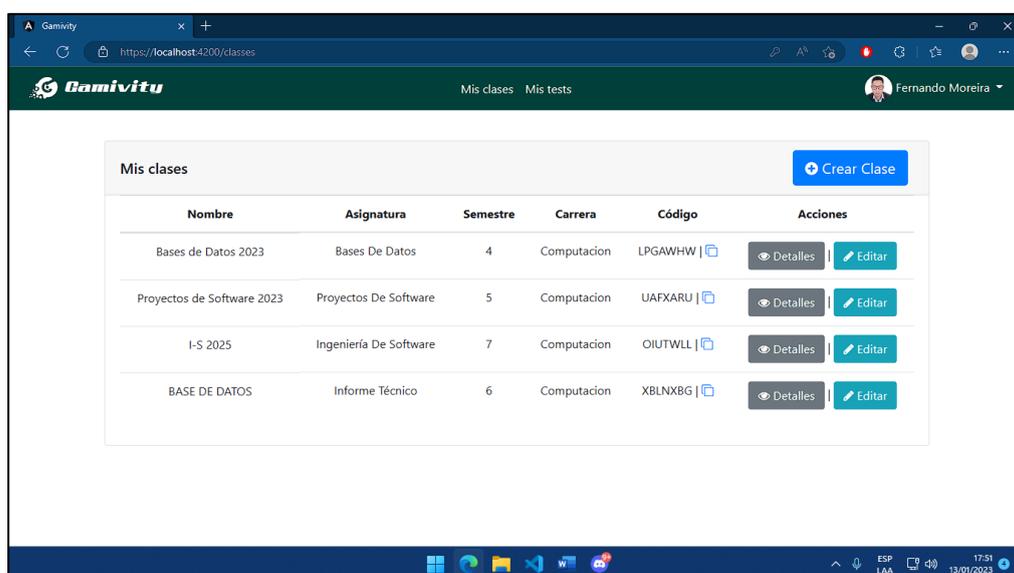


Figura 38. Vista del módulo de Clases

Detalles de una clase

Para que el docente pueda controlar y administrar sus clases, hace falta este módulo, en el cual se le presentarán todos los estudiantes que se hayan unido a una clase es particular, teniendo la opción de poder expulsar a un estudiante.

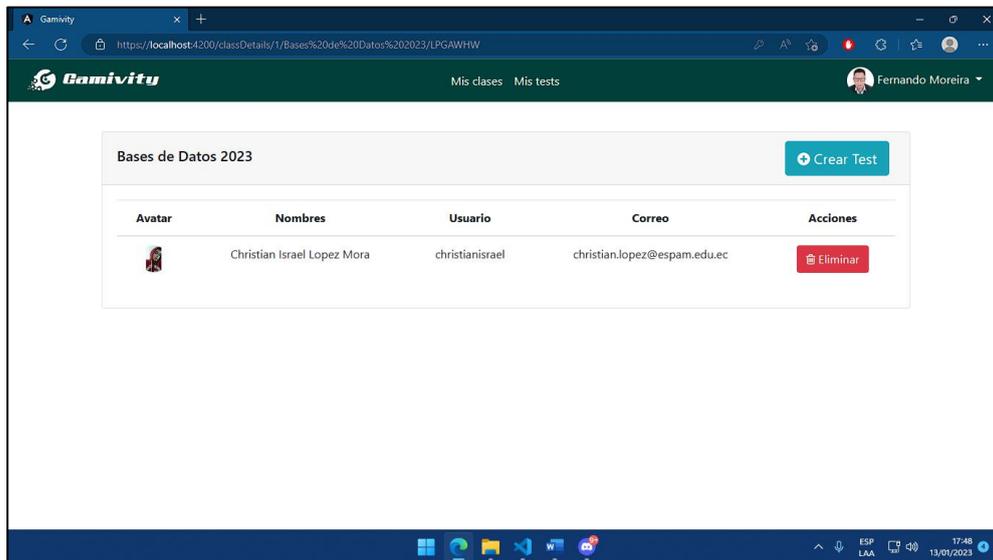


Figura 39. Vista para administrar una clase

Editar una clase

Debido a que existe la posibilidad de que, al momento de crear una clase, se escriba incorrectamente el nombre de la misma, o que simplemente el docente decida cambiarlo, está la opción de editar una clase.

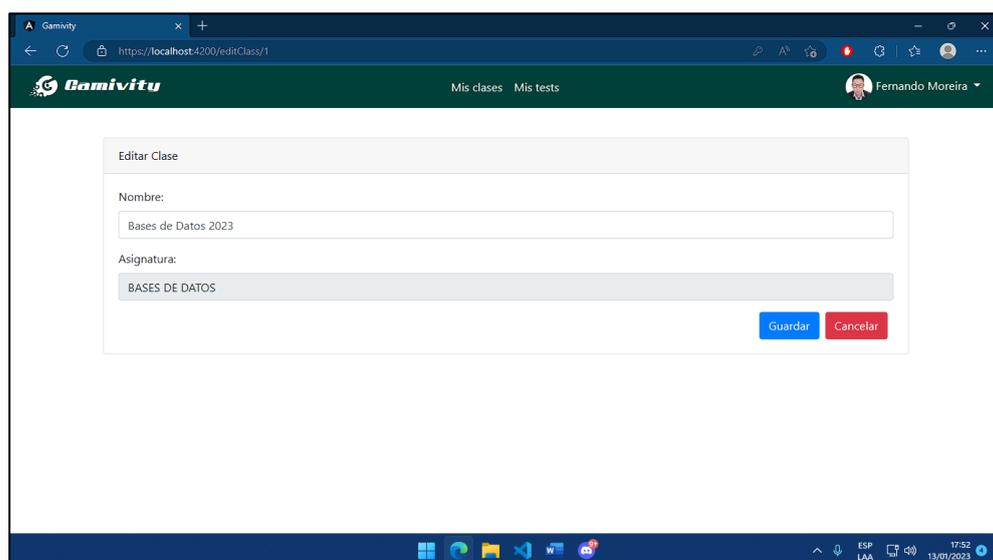


Figura 40. Vista del módulo de editar clase

Crear test

Para poder llevar a cabo la resolución de tests dentro de Gamivity, es indispensable que exista un módulo donde se creen dichos test, en este, se deberán llenar los campos necesarios y complementar con las preguntas que formarán el test en cuestión, teniendo las opciones de crear una nueva pregunta desde cero, utilizar preguntas que él mismo haya creado anteriormente o pudiendo elegir alguna ya registrada en el banco de preguntas.

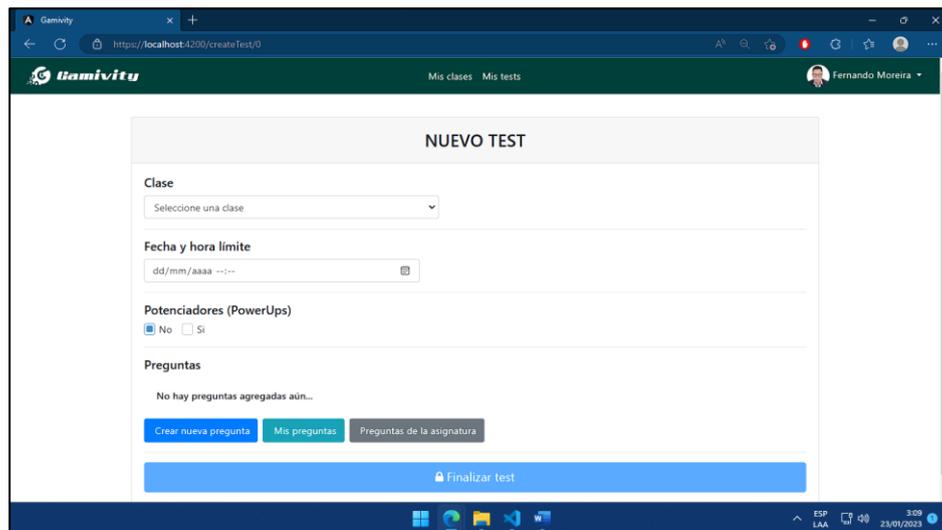


Figura 41. Vista del módulo de crear test

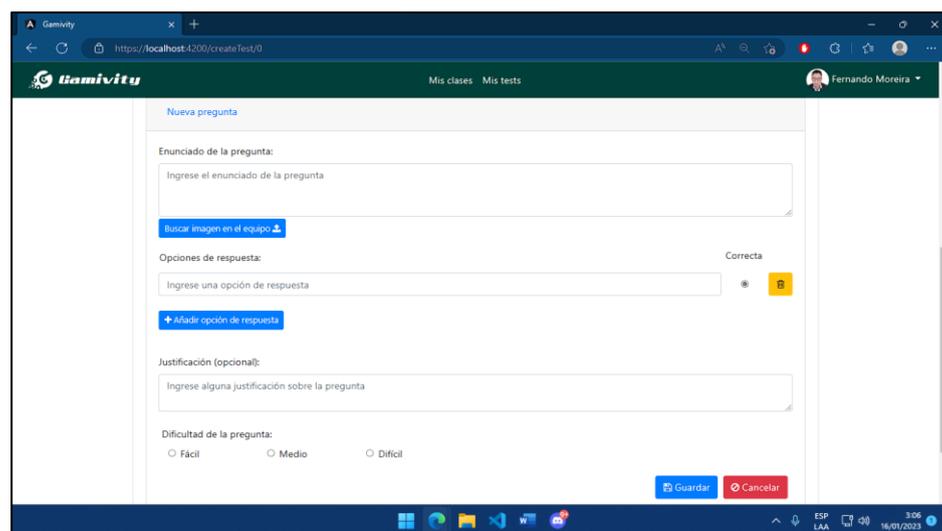


Figura 42. Agregar una nueva pregunta

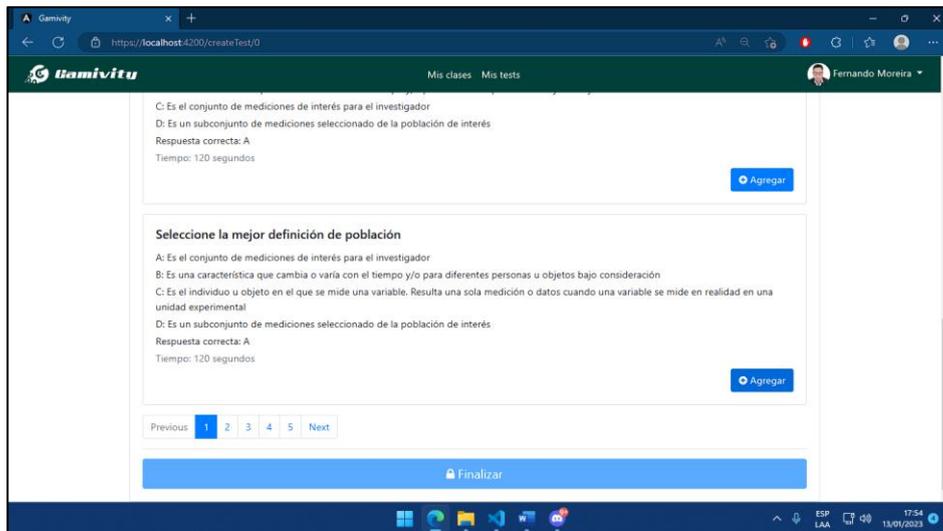


Figura 43. Agregar pregunta existente

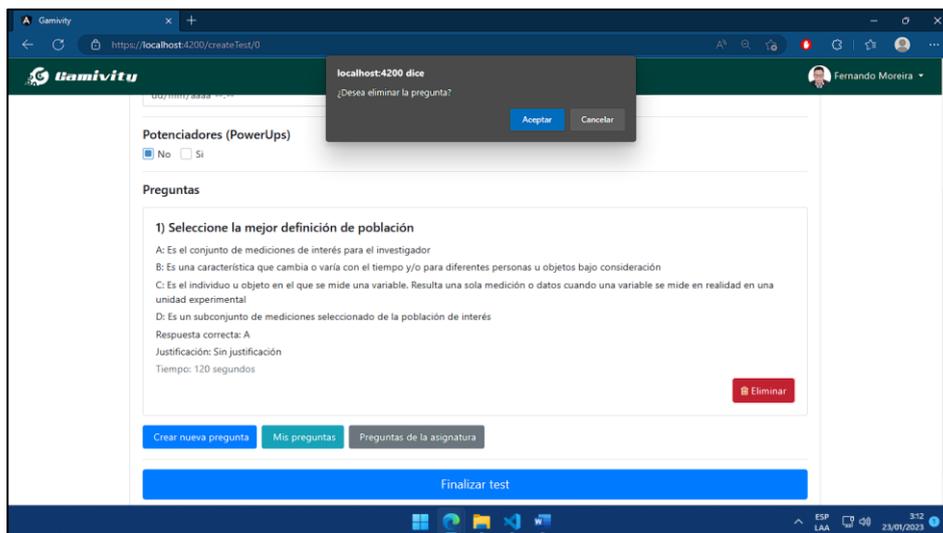


Figura 44. Eliminar pregunta agregada

Mis tests

Una vez que se haya creado algún test, es necesario un módulo donde se puedan gestionar estos mismos. Entre los campos que se muestran en la tabla tenemos: la clase a la que pertenece dicho test, el número de test asignado a esa clase en concreto, si están permitidos los powerups, la fecha en que se creó el test, así como cuándo finaliza.

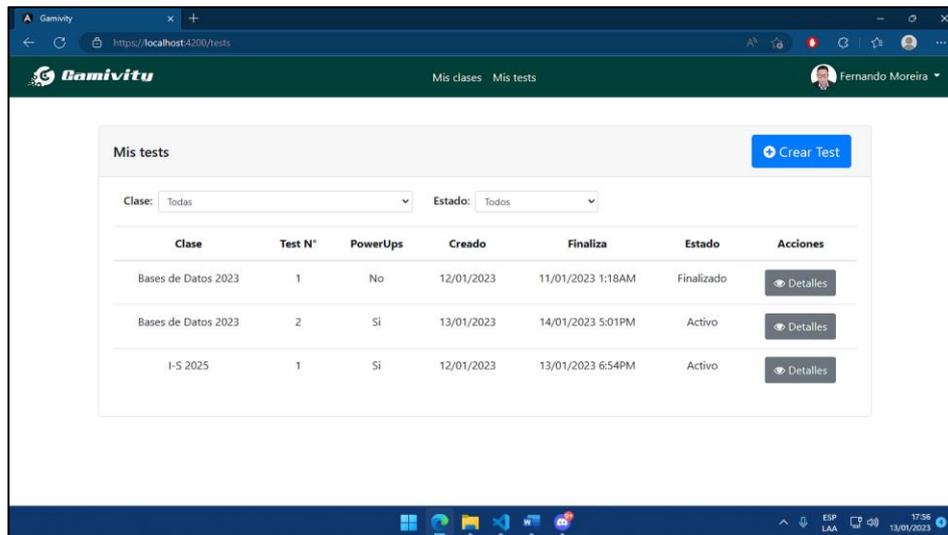


Figura 45. Vista del módulo de Mis Tests

Detalles de un test

Al igual que pasa con las clases creadas por el docente, hace falta un módulo donde se puedan ver los detalles de un test en concreto, aquí se registrarán las calificaciones que los estudiantes obtengan al resolver un test. También se presenta la opción de eliminar el test por si el docente lo ve necesario.

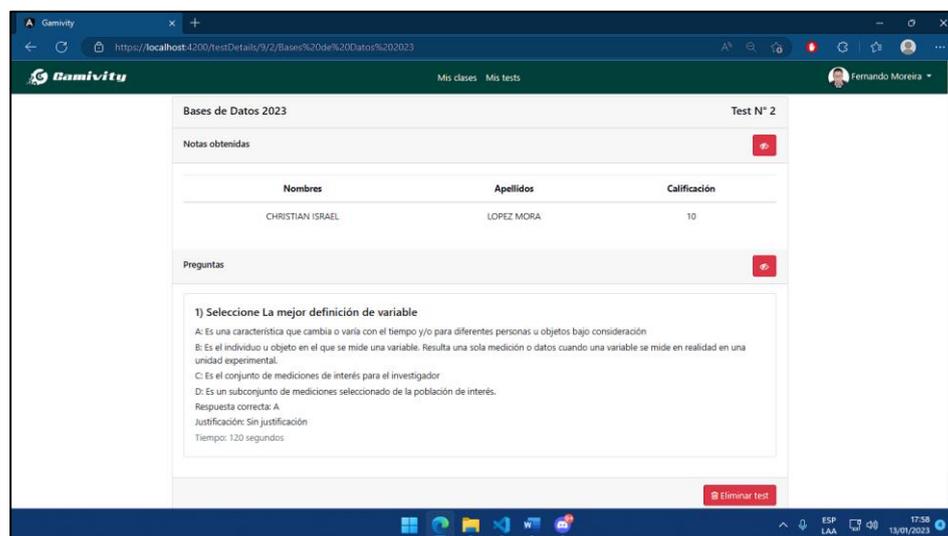


Figura 46. Vista del módulo de administración de un test

Perfil

Aquí es donde se muestra la información del usuario logueado al momento, junto con los respectivos controles que permiten modificar el nombre de usuario como el avatar.

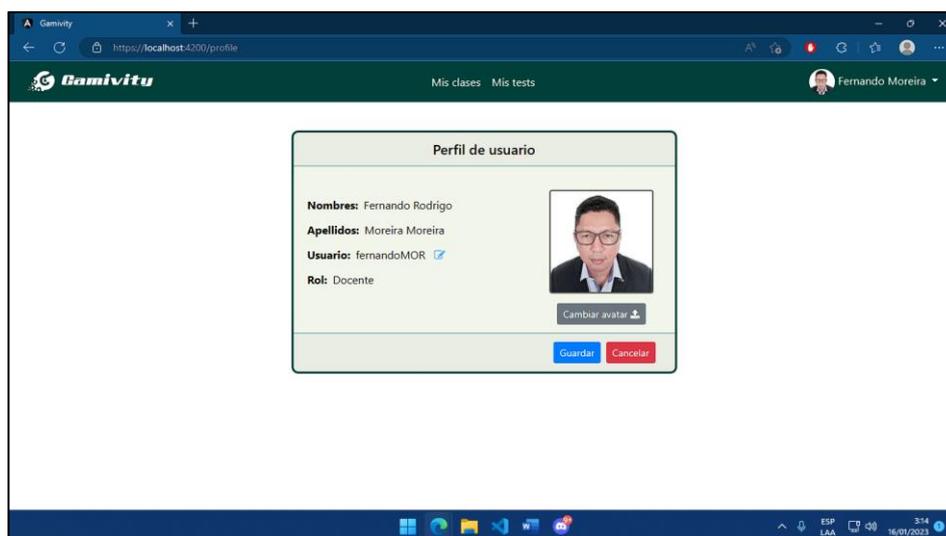


Figura 47. Vista del módulo de perfil de usuario

Mis preguntas

En este apartado se administran las preguntas que un docente haya creado, teniendo la opción de editar alguna de estas mismas, así como poder filtrar las preguntas por asignaturas..

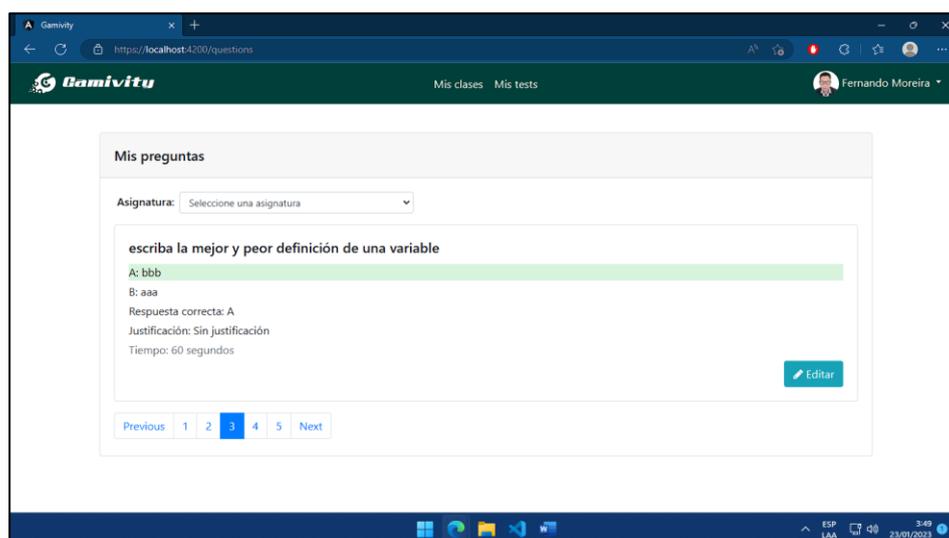


Figura 48. Vista del módulo de Mis preguntas

**ANEXO 7. SELECCIÓN DEL MODELO DE PROCESAMIENTO
DEL LENGUAJE NATURAL ACORDE A LAS NECESIDADES DE
GAMIVITY**

**SELECCIÓN DEL MODELO DE
PROCESAMIENTO DEL LENGUAJE NATURAL
ACORDE A LAS NECESIDADES DE GAMIVITY**

Proyecto: Gamivity

Autores: Dany Álava – Carlos Saavedra

Índice

Índice.....	¡Error! Marcador no definido.
Introducción.....	¡Error! Marcador no definido.
Desarrollo	¡Error! Marcador no definido.
Modelos NLP	¡Error! Marcador no definido.
Modelos del lenguaje.....	¡Error! Marcador no definido.
Embeddings de texto	¡Error! Marcador no definido.
Embeddings contextuales	¡Error! Marcador no definido.
Modelos de atención	¡Error! Marcador no definido.
El modelo Transformer.....	¡Error! Marcador no definido.
Selección del modelo.....	¡Error! Marcador no definido.
Entorno de pruebas de los modelos seleccionados	¡Error! Marcador no definido.
Evaluación final de los modelos.....	¡Error! Marcador no definido.
Conclusiones.....	¡Error! Marcador no definido.
Bibliografía	¡Error! Marcador no definido.

INTRODUCCIÓN

Gracias al incremento de información disponible en internet, han surgido una serie de investigaciones, las cuales permiten desarrollar técnicas capaces de poder identificar y organizar este tipo de información, algunas de estas son las denominadas técnicas de procesamiento del lenguaje natural o también conocidas como sistemas NLP por sus siglas en inglés, y es que, estas han ido ganando gran relevancia en aplicaciones orientadas a la interacción con los usuarios, como es el caso de los navegadores web o aplicaciones de asistencia técnica (Sánchez, 2021).

El área de estudio el cual se enfoca en la comprensión y la manipulación informática del lenguaje humano se denomina procesamiento del lenguaje natural. Esta se encuentra en la intersección entre la informática, la inteligencia artificial y la lingüística computacional (Lloret, 2019). En los últimos años se ha oído hablar mucho del procesamiento del lenguaje natural debido a la era digital en la que nos encontramos, sin embargo, el NLP es una disciplina que cuenta con más de 50 años de investigación y desarrollo (DataRobot, 2016).

Teniendo en cuenta los puntos antes descritos, el objetivo principal de este trabajo consiste en seleccionar un modelo de procesamiento de lenguaje natural que pueda ser utilizado en el sistema multiplataforma "Gamivity" basándose en las necesidades de este, el mismo debe permitir incorporar un módulo de experiencia personalizada que pueda ser aplicado en la relación de preguntas del sistema multiplataforma para la creación de tests personalizados basados en preguntas respondidas de forma errónea.

DESARROLLO

MODELOS NLP

Antes de hablar de los modelos de Procesamiento de lenguaje natural, debemos tener claro de qué maneras de puede expresar el lenguaje natural o lenguas humanas. El lenguaje natural puede expresarse de diversas formas, ya sea por escrito (texto), oralmente (voz) y también mediante signos. Generalmente, el NLP está más enfocado en el tratamiento de textos, ya que es donde hay muchos más datos y son más fáciles de conseguir en formato electrónico. Los audios, aunque encuentren en este formato, primero deben ser procesados para transcribirlos en letras o caracteres y, a partir de ahí, entender la pregunta (Moreno, 2018).

Los algoritmos de NLP generalmente se basan en algoritmos de aprendizaje automático (machine learning). En lugar de codificar a mano grandes conjuntos de reglas, el NLP confía en el aprendizaje automático para aprender automáticamente estas reglas mediante el análisis de un conjunto de ejemplos (es decir, un gran corpus) y haciendo una inferencia estadística. En general, cuantos más datos se analicen, más preciso será el modelo. Usando NLP podemos hacer tareas como resumen automático de textos, traducción de idiomas, extracción de relaciones, Análisis de sentimiento, reconocimiento del habla y clasificación de artículos por temáticas (Talamé, Cardoso, & Amor, 2019).

El término NLP es bastante genérico, sin embargo, este cubre un campo de aplicación bastante amplio. Entre sus principales aplicaciones DataScientest en (2021) destaca que podemos encontrar las siguientes:

- Traducción automática
- Análisis de sentimientos
- Marketing
- Chatbots
- Clasificación de texto
- Reconocimiento de caracteres
- Corrección automática
- Resumen automático

MODELOS DEL LENGUAJE

Existen diversos tipos de modelos NLP, cada uno con sus propias características y funcionalidades, de los cuales se detallan a continuación.

Embeddings de texto

Los embeddings son un tipo de representación vectorial de un texto donde las palabras con un significado similar, tienen vectores similares y cada palabra es representada por un único vector. Estos suelen aprender solo la semántica de las palabras, es decir que no toma en cuenta la sintaxis ni la gramática. Los embeddings de palabras también pueden llamarse espacio vectorial semántico o modelo semántico. Los modelos más relevantes que usan esta técnica son GLoVe, Word2Vec y FastText (Briceño, 2021).

Embeddings contextuales

Los embeddings contextuales se generan de forma dinámica, esto significa que una palabra puede tener diferentes representaciones dependiendo de su contexto. Esto se consigue añadiendo alguna información de los embeddings del resto de palabras en el texto a la palabra en cuestión. El modelo más notable de embeddings contextuales es ELMo (Chen, 2020).

Modelos de atención

Los modelos de atención toman una secuencia de objetos y dan como salida otra secuencia de objetos. Esto se utilizaba mucho en traducción automática, pero se ha expandido a otro tipo de modelos. La estructura de estos modelos está compuesta por un codificador que genera el contexto y un decodificador que genera la salida. En base a la estructura codificador-decodificador, los modelos de atención permiten al decodificador centrarse en las partes importantes de los datos. Esto se consigue pasándole todos los estados ocultos al codificador, no solo el último (García, 2021).

El modelo Transformer

El modelo transformer intenta juntar las mejores ideas del estado del arte y añade algunas mejoras nuevas. Los cambios más notables son el uso de las denominadas capas de auto-atención, lo que significa que el modelo consigue un entendimiento más profundo de la gramática del lenguaje y no solo de la semántica de palabras concretas. Cabe destacar que estos modelos son mucho más rápidos que sus alternativas, esto se debe al extenso uso de la multiplicación de matrices en las capas de autoatención, lo que le permite tener un mejor desempeño en las GPUs (Vaca, 2021).

SELECCIÓN DEL MODELO

Una vez expuestos los diferentes tipos de modelos de procesamiento del lenguaje natural, habría que elegir el más adecuado o que mejor se adapte a las necesidades del sistema "Gamivity". El modelo seleccionado fue el de tipo transformer, esto debido a que su estructura se encuentra basada en un Encoder y un Decoder, en el cual se utiliza el encoder para analizar el contexto de una secuencia de entrada y el decoder es el encargado de generar la secuencia de salida a partir de este contexto, permitiendo así poder analizar el contexto de una pregunta encontrada en el repositorio general de preguntas de Gamivity, para que pueda ser relacionada con otras preguntas del mismo repositorio.

Por otro lado, la segunda razón por la que se optó por este tipo de arquitectura, recae en un aspecto técnico en la implementación específica a este sistema. Y es que, los modelos pre-entrenados de tipo "Transformers" son bastante eficientes a la hora de realizar tareas como calcular la similitud entre oraciones.

En primera instancia, se seleccionaron varios modelos pre-entrenados bajo la arquitectura "Transformers" para su posterior evaluación. Dichos modelos fueron recuperados de la plataforma de código abierto centrada en inteligencia artificial "Hugging Face" (Huggin Face, 2022). Estos forman parte del framework de Python "Sentence Transformers", el cual está diseñado para crear incrustaciones de oraciones, texto e imágenes de última generación. Para dicha selección, se establecieron los siguientes criterios que la implementación de los modelos debía de cumplir, estos se resumen en: modelo base creado bajo la arquitectura

“Transformers”, entrenado con un corpus que incluya al menos el idioma español, así como que los mismos debían de trabajar bajo la librería de Python “sentence_transformers”. A continuación, se listan las diferentes implementaciones de modelos base seleccionadas.

- paraphrase-multilingual-MiniLM-L12-v2
- distiluse-base-multilingual-cased-v1
- paraphrase-multilingual-mpnet-base-v2
- paraphrase-MiniLM-L3-v2
- clip-ViT-B-32-multilingual-v1

ENTORNO DE PRUEBAS DE LOS MODELOS SELECCIONADOS

Consecuentemente, para la evaluación de dichos modelos seleccionados, se construyó un entorno de trabajo en la plataforma de desarrollo “Google Colab”. Este, consiste en un entorno análogo al que estaría presente en el núcleo del sistema “Gamivity”, bajo los mismos parámetros y librerías que condicionan a las funcionalidades asociadas al módulo de aprendizaje personalizado. La mayor diferencia entre este entorno análogo y el real a implementar, es la forma en la que se suministran los datos, ya que en el primero, se utilizan datos de semilla, más específicamente un cuerpo de preguntas de asignaturas, los cuales simulan los que llegarían a través de las llamadas a la base de datos correspondiente.

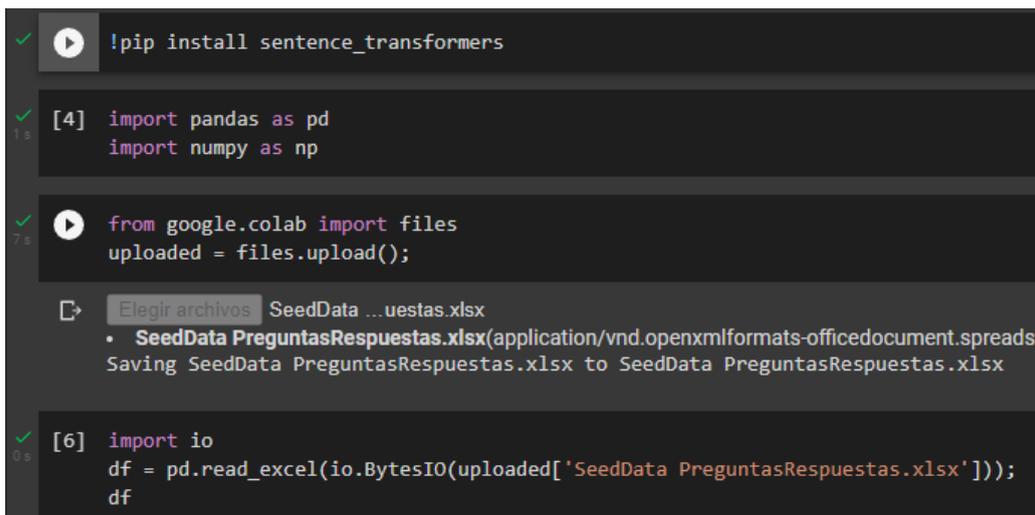
Se creó un cuerpo de preguntas referentes a tres asignaturas relacionadas a la carrera de ciencias computacionales con subtemas asociadas a las mismas, más específicamente, de programación en Python, compuertas lógicas de electrónica digital y sistemas de ficheros de sistemas operativos, tal como se puede apreciar en la siguiente Tabla, esto con el fin de ver qué modelo realiza mejores relaciones correctas y menores relaciones incorrectas. Cabe recalcar que este cuerpo de preguntas contiene diez registros por cada subtema, es decir, treinta preguntas en total junto con su respuesta correcta.

Tabla 6. Muestra del cuerpo de preguntas utilizadas para las pruebas

Programación en Python	Electrónica Digital (Compuertas lógicas, circuitos básicos)	Sistemas Operativos (Temáticas básicas, Sistemas de ficheros)
¿Qué son los generadores en Python? - Funciones que devuelven un conjunto iterable de elementos	¿Qué puerta lógica en sí es un circuito combinacional? - XOR	Un sistema de ficheros está formado por: - Ficheros(archivos) y Directorios(carpetas)
¿Cómo puedes generar números aleatorios en Python? - import random random.random	¿Cuál es el número de salidas de un circuito sumador completo? - Tres	¿Qué indica la tabla de particiones? - El principio, tamaño y tipo de cada partición
¿Es Python sensible a las mayúsculas y minúsculas? - Sí es sensible	¿Cuál es la expresión para la (s) suma (s) de un medio sumador si las entradas son A y B? - A XOR B	¿Qué contienen los ficheros texto plano? - Sólo caracteres imprimibles

Para poder establecer qué modelo pre-entrenado rinde mejor de cara al fin planteado, se optó por un criterio simple en el cual el mayor peso de la selección recae en la correcta asociación entre preguntas y luego, el coste computacional del mismo. La primera parte de este criterio de elección, consiste en que, al escoger tres preguntas de la tabla, una pregunta por temática, esta se asocie mayormente a las otras preguntas de la misma columna a través del cálculo de similitud de coseno, es decir, el modelo que genere la tabla más parecida a la original a partir de dicho cálculo, será el electo en cuanto a este criterio. Por último, en cuanto al coste computacional de los modelos refiere, este se calcula con base en el tiempo de ejecución promedio entre llamada de los procesos que intervienen en el mismo.

En cuanto a la codificación de dicho entorno, en primer lugar, tal y como se aprecia en la Figura a continuación, se realiza la respectiva instalación de la librería "sentence_transformers", así como la carga de la tabla con las preguntas que van a ser suministradas para las pruebas.



```

!pip install sentence_transformers

[4] import pandas as pd
import numpy as np

from google.colab import files
uploaded = files.upload();

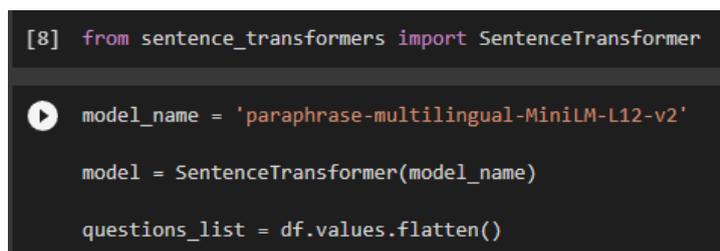
Elegir archivos SeedData ...uestas.xlsx
• SeedData PreguntasRespuestas.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet)
Saving SeedData PreguntasRespuestas.xlsx to SeedData PreguntasRespuestas.xlsx

[6] import io
df = pd.read_excel(io.BytesIO(uploaded['SeedData PreguntasRespuestas.xlsx']));
df

```

Figura 49. Instalación/importación de librerías y cuerpos de preguntas

Consecuentemente, se cargan los respectivos modelos para así poder mapear las preguntas a incrustaciones de las mismas, a través de lo señalado en la siguiente Figura.



```

[8] from sentence_transformers import SentenceTransformer

model_name = 'paraphrase-multilingual-MiniLM-L12-v2'

model = SentenceTransformer(model_name)

questions_list = df.values.flatten()

```

Figura 50. Carga de los modelos

Una vez cargado el modelo, se procede a calcular la similitud coseno entre la pregunta pivote con respecto a las demás preguntas, esta es la parte crucial en la que se calcula el tiempo de ejecución del modelo. En esta parte, también se define el cómo se recuperan las preguntas relacionadas que el modelo estima, esto bajo el criterio de que los diez valores más altos de similitud coseno calculados, teóricamente, tendrían que corresponder a las diez preguntas de la temática perteneciente a la pregunta pivote, esto, siempre y cuando el valor de umbral de similitud coseno sea mayor que 0,25.

```
[21] from sklearn.metrics.pairwise import cosine_similarity

import time
start = time.process_time()

sentence_vecs = model.encode(questions_list)
pivot_index = 2

similarity = cosine_similarity(
[sentence_vecs[pivot_index]],
sentence_vecs[:])[0]

similarity_array = list(similarity)
similarity_array.sort(reverse=True)
similarity_array = similarity_array[:10]

related_questions = []

for i in range(len(similarity_array)):
    for j in range(len(similarity)):
        if(similarity[j] == similarity_array[i] and similarity[j] > 0.25):
            related_questions.append(questions_list[j])

print(time.process_time() - start)
```

Figura 51. Cálculo de la similitud coseno y tiempo de ejecución

Finalmente, se realiza el cálculo correspondiente al puntaje asociado a la eficacia del modelo a la hora de relacionar las preguntas. Este valor está definido bajo la expresión: “10 + asociaciones correctas – asociaciones incorrectas”. El número diez corresponde al número de asociaciones que superan los criterios previamente establecidos; el valor de la variable “asociaciones correctas” hace referencia al número de preguntas que efectivamente el modelo logró de agrupar correctamente, es decir, las que pertenecen a la temática real; en su contraparte, la variable “asociaciones incorrectas” las que no pertenecen a ella. Desde, luego, dicho valor de puntaje final, es máximo (20), cuando el valor de “asociaciones correctas” es igual a diez.

```
correct_associations = 0
for item in related_questions:
    if item in df.iloc[:, 2].values:
        correct_associations += 1

total_associations = len(related_questions)
incorrent_associations = total_associations - correct_associations
final_score = 10 + correct_associations - incorrent_associations

print("Número de asociaciones correctas: ", correct_associations, " de 10")
print("Número de asociaciones incorrectas: ", incorrent_associations, " de 10")
print("Puntaje final: ", final_score, " de 20")
```

Figura 52. Cálculo del puntaje final entre modelos

EVALUACIÓN FINAL DE LOS MODELOS

Como se aprecia en la siguiente Tabla, se listan los modelos seleccionados, junto con el puntaje de eficacia a la hora de relacionar preguntas para las tres preguntas pivotes suministradas, así como del tiempo promedio de ejecución para dicho cálculo. Cabe recalcar que este tiempo de ejecución promedio, se calculó a partir de cinco ejecuciones del código de manera continua. Bajo dichos cálculos, se tiene que el modelo que mejor rendimiento ofreció fue el primero listado, tanto por el mayor puntaje promedio de eficacia, como por el menor tiempo promedio de tiempo de ejecución.

Tabla 7. Evaluación final de los modelos

Modelo	Puntaje T1	Puntaje T2	Puntaje T3	Promedio Puntaje (máx. 30)	Promedio tiempo de ejecución
paraphrase-multilingual-MiniLM-L12-v2	20	22	24	22	1.219
distiluse-base-multilingual-cased-v1	22	22	16	20	1.7824
paraphrase-multilingual-mpnet-base-v2	20	18	22	20	3.2316
paraphrase-MiniLM-L3-v2	18	18	22	19.33	0.489
clip-ViT-B-32-multilingual-v1	20	20	24	21.33	2.273

CONCLUSIONES

- La implementación de un modelo NLP en el sistema multiplataforma “Gamivity” es de suma importancia, para que este pueda realizar las respectivas relaciones entre las preguntas pertenecientes al repositorio general de preguntas, pudiendo así cumplir con el objetivo de que el sistema sea capaz de generar tests personalizados basados en preguntas que los estudiantes hayan respondido erróneamente.
- Si bien existen un sinnúmero de modelos pre-entrenados de tipo NLP, los que mejores se adaptaron a las necesidades planteadas fueron los del tipo “Transformers”, al ser estos muy eficientes con el entendimiento del lenguaje natural, sobre todo, a la hora de estimar la similitud entre oraciones.
- Es necesario comprobar la eficiencia de los algoritmos seleccionados, por este motivo se tuvieron que poner a prueba cada uno de ellos, relacionando las preguntas de un cuerpo de preguntas creado a partir de tres temáticas en concreto (10 preguntas por cada temática), con el fin de poder decantarse por el modelo que genere mejores resultados y menor tiempo computacional. Esto, para así poder simular desde un nivel alto, lo que estaría presente en el módulo de aprendizaje personalizado en un entorno real de producción.
- Una vez realizada la respectiva evaluación de los modelos, se pudo determinar que el modelo “paraphrase-multilingual-MiniLM-L12-v2” obtuvo el mejor promedio de eficacia a la hora de relacionar las preguntas. Sin embargo, considerando que dicha tarea en el aplicativo debe de hacerse cada vez que un docente agrega una nueva pregunta, el tiempo de ejecución para completar la misma, debe de ser el mínimo posible. Es por esta razón, que el modelo “paraphrase-MiniLM-L3-v2”, el cual mostró un tiempo promedio de ejecución hasta 2.5 veces menor con respecto al previamente mencionado, se estableció como el modelo candidato. Y es que, si bien es cierto que este modelo más rápido, tuvo menor puntaje final a la hora de asociar las preguntas, se estableció que la diferencia entre ellos no era significativa, sobre todo, considerando que el cuerpo de preguntas utilizadas para dichas pruebas no era demasiado grande en comparación al número de preguntas que estarían presentes en un entorno de producción del aplicativo.

BIBLIOGRAFÍA

Briceño, B. (2 de Julio de 2021). ¿Qué son los word embeddings y para qué sirven? Obtenido de <https://blogs.iadb.org/conocimiento-abierto/es/que-son-los-word-embeddings/>

Chen, Q. (10 de mayo de 2020). Contextual word embeddings — Part1. Obtenido de <https://medium.com/analytics-vidhya/contextual-word-embeddings-part1-20d84787c65>

DataRobot. (11 de Agosto de 2016). What is Natural Language Processing? Introduction to NLP. Obtenido de <https://www.datarobot.com/blog/what-is-natural-language-processing-introduction-to-nlp/>

DataScientest. (2 de diciembre de 2021). NLP Natural Language Processing: Introducción. Obtenido de <https://datascientest.com/es/nlp-natural-language-processing-introduccion#:~:text=Por%20lo%20general%2C%20podemos%20distinguir,y%20modelos%20de%20Deep%20Learning.&text=Los%20m%C3%A9todos%20basados%20%E2%80%8B%E2%80%8Ben%20reglas%20consisten%20en%20su,ejem>

García, G. (2021). Modelos de Transformers para la Clasificación de Texto. Madrid: Universidad Politécnica de Madrid.

Huggin Face. (2022). Huggin Face. Obtenido de <https://huggingface.co/>

Lloret, J. (2019). Procesamiento del Lenguaje Natural (PLN). Obtenido de <https://ebook.formaempleo.info/books/los-estados-de-la-inteligencia-artificial-%28ia%29-the-states-of-artificial-intelligence-%28ai%29/page/cap%C3%ADtulo-7-2%C2%BA-%7C-chapter-7-2%C2%BA>

Moreno, A. (2018). Procesamiento del lenguaje natural ¿qué es? Obtenido de Instituto de Ingeniería del Conocimiento: <https://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural/>

Sánchez, M. (2021). Clasificación de textos basado en los modelos pre-entrenados BERT. Valencia: Universitat Politècnica de València.

Talamé, L., Cardoso, A., & Amor, M. (2019). Comparación de herramientas de procesamiento de textos en español extraídos de una red social para Python. Sociedad Argentina de Informática e Investigación Operativa. Obtenido de <http://sedici.unlp.edu.ar/handle/10915/87854>

Vaca, A. (17 de noviembre de 2021). Transformers en Procesamiento del Lenguaje Natural. Obtenido de <https://www.iic.uam.es/innovacion/transformers-en-procesamiento-del-lenguaje-natural/>

**ANEXO 8. DESCRIPCIÓN DE LOS MÉTODOS INVOLUCRADOS
EN EL MÓDULO DE APRENDIZAJE PERSONALIZADO EN LA
PLATAFORMA “GAMIVITY”.**

**MÉTODOS INVOLUCRADOS EN EL MÓDULO
DE APRENDIZAJE PERSONALIZADO**

Proyecto: Gamivity

Autores: Dany Álava – Carlos Saavedra

Agregación de relación entre preguntas

A partir de lo mostrado en la siguiente figura, se procede a desglosar la implementación de los componentes involucrados en dicho diagrama.

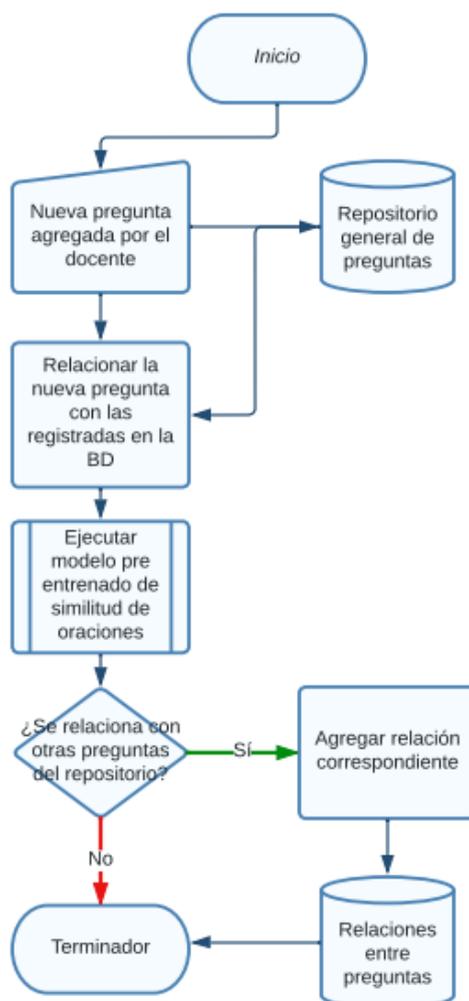
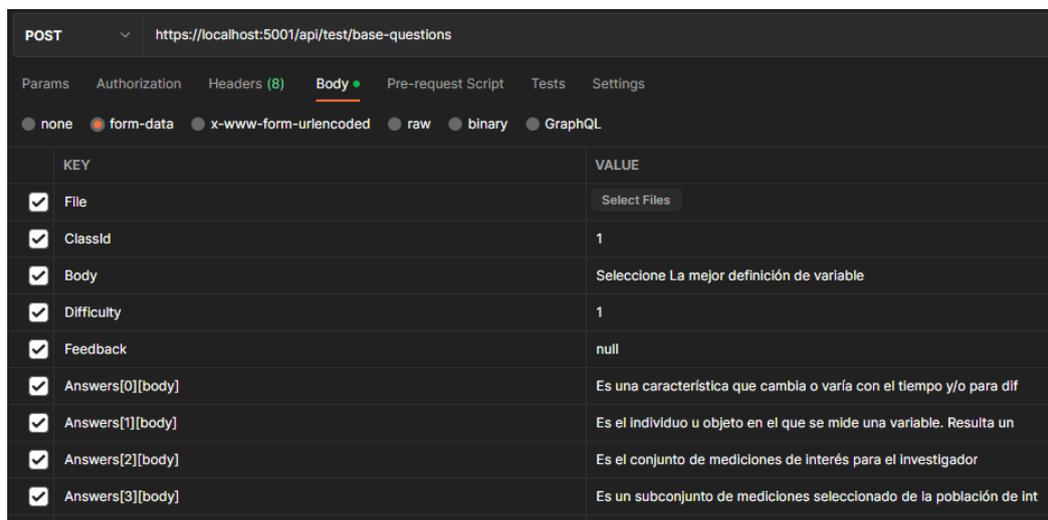


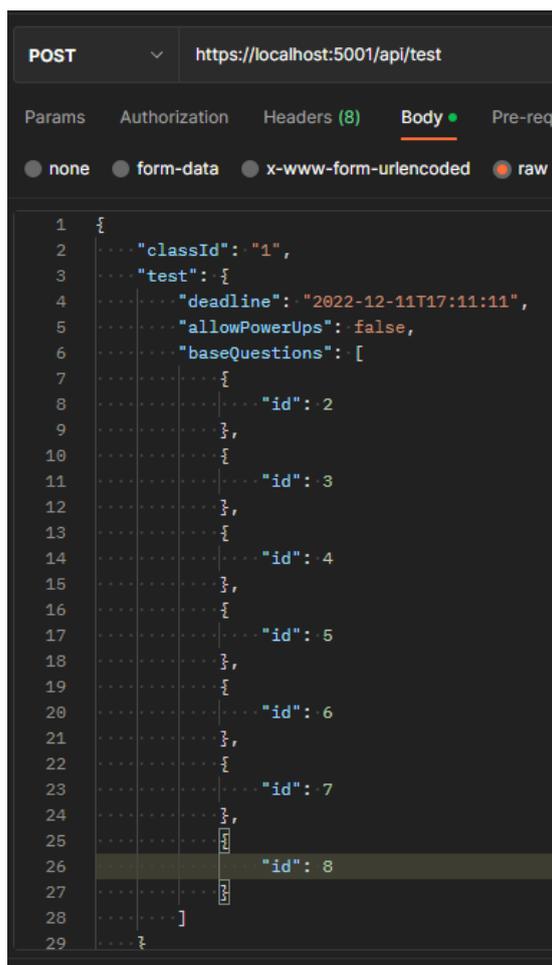
Figura 53. Diagrama de flujo del proceso de agregación entre preguntas

1. Docente agrega un nuevo test a la clase



KEY	VALUE
<input checked="" type="checkbox"/> File	Select Files
<input checked="" type="checkbox"/> ClassId	1
<input checked="" type="checkbox"/> Body	Seleccione La mejor definición de variable
<input checked="" type="checkbox"/> Difficulty	1
<input checked="" type="checkbox"/> Feedback	null
<input checked="" type="checkbox"/> Answers[0][body]	Es una característica que cambia o varía con el tiempo y/o para dif
<input checked="" type="checkbox"/> Answers[1][body]	Es el individuo u objeto en el que se mide una variable. Resulta un
<input checked="" type="checkbox"/> Answers[2][body]	Es el conjunto de mediciones de interés para el investigador
<input checked="" type="checkbox"/> Answers[3][body]	Es un subconjunto de mediciones seleccionado de la población de int

Figura 54. Agregar nueva pregunta al test



```

1  {
2  ... "classId": "1",
3  ... "test": {
4  ...   "deadline": "2022-12-11T17:11:11",
5  ...   "allowPowerUps": false,
6  ...   "baseQuestions": [
7  ...     {
8  ...       "id": 2
9  ...     },
10 ...     {
11 ...       "id": 3
12 ...     },
13 ...     {
14 ...       "id": 4
15 ...     },
16 ...     {
17 ...       "id": 5
18 ...     },
19 ...     {
20 ...       "id": 6
21 ...     },
22 ...     {
23 ...       "id": 7
24 ...     },
25 ...     {
26 ...       "id": 8
27 ...     }
28 ...   ]
29 ... }

```

Figura 55. Agregar nuevo test a la clase

2. Asumiendo que todas las preguntas del test agregado son nuevas, se procede a ejecutar el endpoint encargado de utilizar el servicio creado en el framework de Python “Flask” para el cálculo de similitud entre preguntas. Cabe recalcar que las relaciones encontradas, mostradas en la figura a continuación, son bidireccionales, es decir la columna izquierda está relacionada con la derecha y viceversa.

Por ejemplo: si la pregunta (baseQuestion) con id 2, está relacionada con otra pregunta con id 1, se asume que esta última, también está relacionada con la pregunta con id 2; (2 entonces 1 = 1 entonces 2).

```

1  {
2    "generalClassId": 395,
3    "baseQuestionIds": [
4      2,
5      3,
6      4,
7      5,
8      6,
9      7,
10     8
11   ]
12 }

```

Figura 56. Calcular relación entre preguntas

	BaseQuestion1Id	BaseQuestion2Id
1	2	1
2	3	1
3	4	1
4	2	3
5	2	4
6	3	4
7	6	12

Figura 57. Relaciones encontradas, guardadas en la tabla correspondiente

Controlador de aprendizaje personalizado

Siguiendo lo descrito en el diagrama de la siguiente figura, se detalla los procesos involucrados a la hora de construir los tests personalizados para el estudiante.

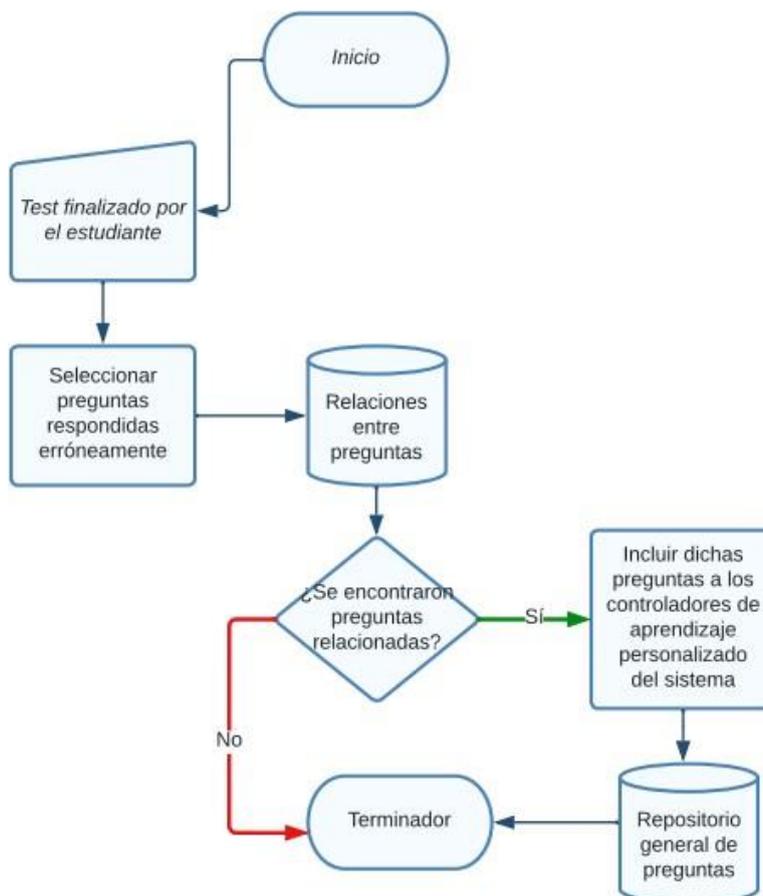


Figura 58. Diagrama de flujo del controlador de Aprendizaje personalizado

1. El endpoint que se muestra a continuación, se ejecuta cuando un estudiante finaliza el test asignado por el docente a una clase en particular, de aquí, se seleccionan las preguntas que el estudiante haya respondido de manera errónea, las cuales servirán para crear los tests personalizados consecuentes.

```

POST https://localhost:5001/api/grades
Params Authorization Headers (9) Body
● none ● form-data ● x-www-form-urlencoded
1 {
2   .... "testId": 1,
3   .... "selectedAnswers": [
4     ..... {
5       ..... "testQuestionId": 5,
6       ..... "answerId": 16,
7       ..... "answeredAtSecond": 25
8     ..... },
9     ..... {
10      ..... "testQuestionId": 6,
11      ..... "answerId": 20,
12      ..... "answeredAtSecond": 101
13     ..... },
14     ..... {
15      ..... "testQuestionId": 7,
16      ..... "answerId": 25,
17      ..... "answeredAtSecond": 80
18     ..... },
19     ..... {
20      ..... "testQuestionId": 8,
21      ..... "answerId": null,
22      ..... "answeredAtSecond": 0
23     ..... },
24     ..... {
25      ..... "testQuestionId": 9,
26      ..... "answerId": 38,
27      ..... "answeredAtSecond": 90
28     ..... }
  ]
}

```

Figura 59. Endpoint automático de agregación de notas a un test

	Id	StudentId	BaseQuestionId	AppearanceProbability
1	1	11	2	1
2	2	11	4	1
3	3	11	5	1
4	5	11	7	1
5	6	11	8	1

Figura 60. Preguntas respondidas erróneamente por un estudiante

Al ejecutar el siguiente endpoint, se recupera el contenido del test personalizado, en este caso, a través de la especificación del id de la materia "generalClassId=395". Este endpoint hace referencia a cuando un estudiante accede a la modalidad de juego libre por asignatura, en dicho test, el contenido del mismo, está conformado por un máximo del 40% proveniente de las preguntas almacenadas en la tabla de relación entre preguntas (figura 52), mientras que el porcentaje restante son preguntas seleccionadas aleatoriamente correspondientes a la asignatura especificada. Cabe recalcar, que cada vez que el usuario responda correctamente a una de estas preguntas relacionadas, el valor de la propiedad "AppearanceProbability", de la tabla mostrada en la figura 8, decrementará en 0.2, hasta eventualmente llegar a 0 que es cuando dicho registro es eliminado automáticamente. Esto último, con el fin de que estas preguntas con temáticas que se asume que el estudiante va mejorando, dejen de aparecer tan frecuentemente en los tests personalizados.

The screenshot displays a REST client interface with the following details:

- Method:** GET
- URL:** https://localhost:5001/api/gameModes/related-questions/395
- Status:** 200 OK
- Time:** 86 ms

The response body is a JSON array of 4 objects. The first object is shown in detail below:

```

{
  "id": 3,
  "maxSeconds": 120,
  "body": "Seleccione la mejor definición de población",
  "imageUrl": null,
  "answers": [
    {
      "body": "Es el conjunto de mediciones de interés para el investigador"
    },
    {
      "body": "Es una característica que cambia o varía con el tiempo y/o para diferentes personas u objetos bajo consideración"
    },
    {
      "body": "Es el individuo u objeto en el que se mide una variable. Resulta una sola medición o datos cuando una variable se mide en realidad en una unidad experimental"
    },
    {
      "body": "Es un subconjunto de mediciones seleccionadas de la población de interés"
    }
  ],
  "feedback": null
}

```

The interface also shows a table for the response body with the following columns: KEY, VALUE, and DESCRIPTION. The table is currently empty.

KEY	VALUE	DESCRIPTION
Key	Value	Description

Figura 61. Respuesta de la ejecución del endpoint de obtención de preguntas relaciones a los fallos del estudiante

ANEXO 9. INFORME DE PRUEBAS UNITARIAS

INFORME DE PRUEBAS UNITARIAS

Proyecto: Gamivity

Autores: Dany Álava – Carlos Saavedra

Componente unitario: Acceso a la plataforma (docentes y estudiantes) y edición del usuario.

Precondición: Tener credenciales activas del Sistema de Gestión Académica de la universidad.

Descripción	Resultado esperado	Resultado obtenido
¿Qué sucede si ingresa un usuario con rol estudiante o rol docente por primera vez con credenciales válidas?	Mensaje satisfactorio	Alerta de registro e inicio de sesión correcto
¿Qué sucede si ingresa un usuario con rol docente a través de credenciales válidas a la plataforma web?	Mensaje satisfactorio	Alerta de inicio de sesión correcto y posterior redirección al "Home"
¿Qué sucede si ingresa un usuario con rol estudiante a través de credenciales válidas a la plataforma web?	Mensaje de alerta (redirección plataforma gamificada)	Alerta de rol no permitido y posterior redirección a URL correcta
¿Qué sucede si ingresa un usuario con rol docente a través de credenciales válidas a la aplicación móvil gamificada?	Mensaje de alerta (redirección plataforma web de gestión)	Alerta de rol no permitido y posterior redirección a URL correcta
¿Qué sucede si ingresa un usuario con rol estudiante por primera vez con credenciales válidas, pero no se encuentra matriculado al periodo académico actual?	Mensaje de error	Alerta de error al registrar al usuario
¿Qué sucede si un usuario de cualquier rol accede al apartado de editar usuario?	Se muestran campos editables	Campos editables mostrados: nombre de usuario y foto de perfil

Componente unitario: Plataforma web de gestión docente.

Precondición: Usuario autenticado con rol docente.

Descripción	Resultado esperado	Resultado obtenido
¿Qué sucede si un docente accede al apartado de crear una nueva clase?	Se muestran los campos necesarios para la creación de una clase	Campos por definir de parte del docente: nombre de la clase y asignatura
Mientras el docente crea una clase, ¿qué asignaturas puede seleccionar para dicha clase?	Lista de asignaturas que el docente imparte en la universidad	Se muestran las asignaturas del docente agrupadas por carrera
¿Qué sucede si el docente crea una nueva clase de una misma asignatura con un nombre igual a otra clase existente?	Mensaje de error	Alerta de error al crear dicha clase
¿Qué sucede si el docente ingresa a la sección de “Mis clases”?	Se listan las clases que este usuario ha creado	Clases activas que ha creado dicho usuario junto a su respectiva información
¿Qué sucede si el docente accede al apartado de editar una clase en particular?	Se muestra el campo del nombre de la clase a editar	Campo del nombre de la clase a editar
¿Qué sucede si el docente guarda los cambios del nombre de una clase en particular?	Mensaje satisfactorio en caso de que sea un nombre único	Alerta de éxito en caso de que el nuevo nombre no sea igual al de las demás clases de dicha asignatura
¿Qué sucede cuando el docente accede al apartado de detalles de una clase?	Se muestran los estudiantes unidos a la clase	Lista de estudiantes de dicha clase con sus respectivos atributos públicos
¿Qué sucede si el docente accede al apartado de crear un test sin antes haber creado al menos una clase?	Se muestran los campos necesarios para la creación de un test	Alerta de error al intentar crear un test ya que no tiene clases por seleccionar

¿Qué sucede si el docente accede al apartado de crear un test una vez haya creado al menos una clase?	Se muestran los campos necesarios para la creación de un test	Campos por definir de parte del docente: clase, fecha límite, permitir usar “power ups”, añadir preguntas
Mientras el docente crea un test, ¿Qué sucede si accede al apartado de crear una nueva pregunta?	Se muestran los campos necesarios para la creación de una nueva pregunta	Campos por definir de parte del docente: Enunciado, imagen (opcional), respuestas, justificación (opcional) y la dificultad
Mientras el docente crea un test, ¿Qué sucede si en lugar de crear una pregunta, accede al apartado de selección de pregunta existente?	Lista de preguntas existentes para dicha asignatura	Se muestran las preguntas que este usuario u otros que imparten una clase de dicha asignatura hayan añadido previamente
¿Qué sucede si el docente ingresa a la sección de “Mis tests”?	Se listan los tests que el docente haya creado	Lista de tests creadas por el docente, en el que, además, puede especificar diferentes filtros: tests por clase específica y tests activos o inactivos
¿Qué sucede cuando se cumple la fecha límite de un test?	Test cambia de estado	El test en cuestión pasa de activo a inactivo
¿Qué sucede cuando un docente elimina un test de una clase?	Test eliminado de la clase	El test se elimina y no estará disponible para los estudiantes
¿Qué sucede cuando el docente accede al apartado de detalles de un test?	Se muestran las calificaciones de los estudiantes de dicho test junto a las preguntas que conforman el mismo	Lista de estudiantes a los que se les asignó el test con su respectiva calificación, así como las preguntas del test en cuestión
¿Qué sucede cuando un docente elimina a un estudiante de una clase?	Estudiante eliminado de la clase	Alerta de confirmación y posteriormente el estudiante deja de formar parte de la clase

Componente unitario: Endpoints asociados a las funcionalidades del estudiante.

Precondición: Usuario autenticado con rol estudiante.

Descripción	Resultado esperado	Resultado obtenido
¿Qué sucede si un estudiante ejecuta el método de unirse a una clase a través de un código de clase válido?	Estudiante agregado a la clase	Agregación de dicho estudiante a la clase en cuestión
¿Qué sucede cuando un estudiante ejecuta el método de obtener tests?	Se listan los tests que los docentes hayan asignado al estudiante	Lista de tests creados por los docentes de distintas clases, además, puede especificar diferentes filtros: tests activos, inactivos, o por clase específica
¿Qué sucede cuando un estudiante ejecuta el método de acceder a un test activo?	Se muestra el contenido del test	Contenido del test, con todos los respectivos campos
¿Qué sucede si un estudiante ejecuta el método de finalizar test?	Verificación de las respuestas seleccionadas –cálculos de puntos	Se comprueban si las respuestas son las correctas – cálculo de calificación, gamipoints, gamiXp y gamicoins
¿Qué sucede si un estudiante ejecuta el método de obtener detalles de un test resuelto?	Resumen del test seleccionado	Lista de preguntas y respuestas escogidas por el estudiante
¿Qué sucede si un estudiante ejecuta el método de obtener un usuario?	Se muestra la información pública de dicho usuario	Obtención de los atributos públicos del usuario: identificación, nombre de usuario, nombres completos, rol y foto de perfil

Componente unitario: Endpoints asociados a elementos de juego.

Precondición: Usuario autenticado con rol estudiante y formar parte de al menos una clase.

Descripción	Resultado esperado	Resultado obtenido
¿Qué sucede si un estudiante ejecuta el método de obtener preguntas por asignatura en el modo de juego libre?	Se obtiene una colección aleatoria de preguntas de dicha asignatura seleccionada	Lista de preguntas de la asignatura seleccionada en caso de que la asignatura corresponda a una de las clases que se haya unido
¿Qué sucede si un estudiante ejecuta el método de obtener preguntas por nivel en el modo de juego libre?	Se obtiene una colección aleatoria de preguntas de dicho nivel seleccionado	Lista de preguntas del nivel seleccionada en caso de que el nivel corresponda a uno de las clases que se haya unido
¿Qué sucede si un estudiante ejecuta el método de obtener preguntas generales en el modo de juego libre?	Se obtiene una colección aleatoria de preguntas de todas las asignaturas y niveles de clases que el estudiante se haya unido	Lista de preguntas correspondientes a toda la trayectoria de dicho estudiante en la aplicación
¿Qué sucede si un estudiante ejecuta el método de adquirir un power up?	Verificar que tenga las gamicoins necesarias para la adquisición	Se añade el power up al inventario del usuario siempre que el costo sea menor que las gamicoins del estudiante
¿Qué sucede si un estudiante ejecuta el método de recuperar power ups disponibles?	Se listan los power ups que el estudiante pueda adquirir de cada tipo, máximo 5.	Lista de power ups cuyo valor de conteo en el inventario del estudiante sea máximo 5
¿Qué sucede si un estudiante ejecuta el método de obtener insignias adquiridas?	Se muestran las insignias adquiridas por el usuario, junto el nombre, descripción y fecha de adquisición	Lista de insignias adquiridas con los respectivos atributos

¿Qué sucede si un estudiante ejecuta el método de obtener las tablas de clasificaciones?	Se listan hasta los 100 primeros estudiantes de acuerdo al número de gamipoints obtenidos	Lista de dichos estudiantes, ordenados descendientemente de acuerdo a los diferentes filtros: por asignatura, por clase o global.
--	---	---

Componente unitario: Endpoints asociados al módulo de aprendizaje personalizado.

Precondición: Microservicio en Flusk operativo.

Descripción	Resultado esperado	Resultado obtenido
¿Qué sucede si un docente crea nuevas preguntas al momento de agregar un nuevo test?	Petición al micro servicio en Flusk para calcular la relación entre preguntas	Colección de preguntas relacionadas respecto a las que se suministró a la llamada al microservicio
¿Qué sucede si un estudiante responde erróneamente una pregunta de un test?	Se almacenan dichos errores en la base de datos	Al terminar un test, se ejecuta el método para guardar las respuestas erróneas de un estudiante
¿Qué sucede si un estudiante ejecuta el método de obtener preguntas en cualquier modo de juego libre?	Se obtiene una colección aleatoria de preguntas relacionadas a las que respondió erróneamente en partidas anteriores	Lista de preguntas relacionadas que conforma hasta máximo un 40% del test en cuestión, el porcentaje restante son preguntas aleatorias de la asignatura correspondiente
¿Qué sucede si un estudiante responde correctamente una pregunta que forma parte de la tabla "respuestas incorrectas"?	La probabilidad de aparición de dicha pregunta decrementa en un 20%	El valor de la propiedad "AppearanceProbability" disminuye en un 20%. Si esta llega a cero, se elimina de dicha tabla

ANEXO 10. INFORME DE MIGRACIÓN DEL SISTEMA

INFORME DE MIGRACIÓN DEL SISTEMA

Proyecto: Gamivity

Autores: Dany Álava – Carlos Saavedra

Inicialización de la API (.NET 6)

1. Descargar e instalar .NET SDK 6.0.x: <https://dotnet.microsoft.com/es-es/download/dotnet/6.0>
2. Comprobar la correcta instalación y versión a través del comando “dotnet --version”.
3. Por defecto la API utiliza el motor de base de datos Microsoft Sql Server, por ello, se debe especificar el valor de las keys dentro de la propiedad “ConnectionStrings” dentro del archivo de configuración ubicado en la ruta “Gamivity/Api/appsettings.json”.
4. Cambiar el valor de la propiedad “TokenKey” dentro del archivo de configuración para establecer una cadena de al menos 128 bits de longitud y así garantizar la seguridad a la hora de realizar la firma de los JWT por parte del servidor.
5. Establecer las respectivas credenciales del servicio de host de contenido multimedia “Cloudinary”, dentro del archivo de configuración en la key con nombre “CloudinarySettings”.
6. Ejecutar el comando “dotnet run” para inicializar el servidor.

Inicialización del proyecto angular correspondiente a la administración docente del sistema

1. Descargar e instalar el gestor de dependencias “nodejs” desde el sitio oficial <https://nodejs.org/es/> La instalación de Node.js también instalará npm automáticamente.
2. A continuación, instalar angular con el comando “npm install @angular/cli”
3. Comprobar que se ha instalado correctamente con el siguiente comando “ng version”.
4. Abrir la ruta del proyecto en el terminal e instalar las respectivas dependencias con el comando “npm install”.
5. Una vez instaladas las dependencias ejecutar “ng serve” para inicializar el servidor.

Inicialización del microservicio de relaciones entre preguntas a través de Transformers (Python Flask en Windows).

1. Instalar Python 2.7 - Python 3.5 o uno más reciente.
2. Instalar las variables de entorno, a través del comando "py -3 -m pip install virtualenv".
3. Crear entorno a través del comando "py -3 -m venv < nombre del entorno >".
4. Activar el entorno a través del comando "<nombre del entorno >\Scripts\activate".
5. Instalar Flask dentro del entorno activado "pip install Flask".
6. Establecer la variable de entorno FLASK_APP a través de "set FLASK_APP run.py".
7. Ejecutar el comando "flask run" para poner en marcha el microservicio.

Solicitud de despliegue del backend y la plataforma web de gestión docente del sistema Gamivity en los servidores de la ESPAM MFL.



Caiceta, 21 de octubre de 2022

PARA: Magíster
Joffre Ramón Moreira Pico
DIRECTOR DE LA CARRERA DE COMPUTACIÓN

ASUNTO: Solicitud de despliegue del backend y plataforma web en la infraestructura tecnológica de la carrera.

Saludos cordiales, deseándole éxitos en sus labores diarias. Teniendo como antecedente la solicitud de acceso a la infraestructura tecnológica de la carrera, emitida el 4 de julio de 2022 por el director del proyecto de vinculación "LA GAMIFICACIÓN EN LA EDUCACIÓN SUPERIOR". Como autores del trabajo de integración curricular titulado "BACKEND DE APLICACIÓN MULTIPLATAFORMA DE APRENDIZAJE PERSONALIZADO PARA LOS ESTUDIANTES DE LA ESPAM MFL", el cual forma parte del primer objetivo del proyecto de vinculación antes mencionado. Solicitamos a usted de la manera más comedida nos de su autorización para realizar el respectivo despliegue del backend y la plataforma web de gestión docente del sistema **Gamivity**. Adjunto a esta solicitud el cronograma de actividades para realizar la operación antes mencionada. Para realizar correctamente lo antes descrito y no alterar la planificación del proyecto, esta operación requerirá de una (1) semana del mes de noviembre del presente año.

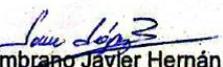
Agradeciendo su gentil atención. Se suscriben

Atentamente,




 Dany Rolando Alava Santana
 AUTORES DEL TRABAJO DE INTEGRACIÓN CURRICULAR


 Carlos Alberto Saavedra Escalante
 AUTORES DEL TRABAJO DE INTEGRACIÓN CURRICULAR


 López Zambrano Javier Hernán MSC.
 TUTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Cronograma de actividades

Actividades	Día				
	1	2	3	4	5
Compilar y testear la API desarrollada en .NET 6 en un servidor de desarrollo.	X				
Corregir los errores que se puedan presentar al poner a la API en ejecución	X				
Crear una "release build" o compilación de la versión de la API y probarlo en un servidor de ensayo.		X			
Desplegar la "release build" en el servidor de producción.		X			
Configurar el servidor de producción para alojar la API y plataforma web de gestión.		X			
Compilar y testear el microservicio desarrollado en Flask en un servidor de desarrollo.			X		
Testear la API y la plataforma web en el servidor de producción para asegurar su correcto funcionamiento.			X	X	
Configurar el servidor de producción para alojar el microservicio, incluida la configuración de las variables de entorno y las cadenas de conexión necesarias.			X	X	
Testear el microservicio en el servidor de producción para asegurar su correcto funcionamiento.				X	X

ANEXO 11. MANUAL DE PROGRAMADOR

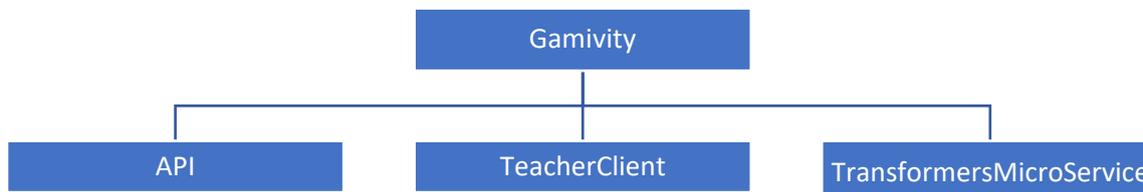
MANUAL DE PROGRAMADOR

Proyecto: Gamivity

Autores: Dany Álava – Carlos Saavedra

MANUAL DE PROGRAMADOR

ESTRUCTURA DEL PROYECTO (NIVEL 1)



API: Proyecto de .NET 6 que contiene todos los métodos y componentes necesarios para que tanto la plataforma web de gestión como la aplicación móvil gamificada puedan operar con los respectivos datos necesarios.

TeacherClient: Proyecto en Angular 14 que contiene los métodos y componentes necesarios para que el usuario (en este caso el docente, ya que este sería la administración docente) pueda interactuar con la API.

TransformersMicroService: Microservicio creado a partir del Framework de Python "Flask, para poder crear los métodos necesarios a la hora de realizar el cálculo de similitud entre preguntas a través de la librería "sentence_transformers".

Estructura del proyecto (Nivel 2 - API)

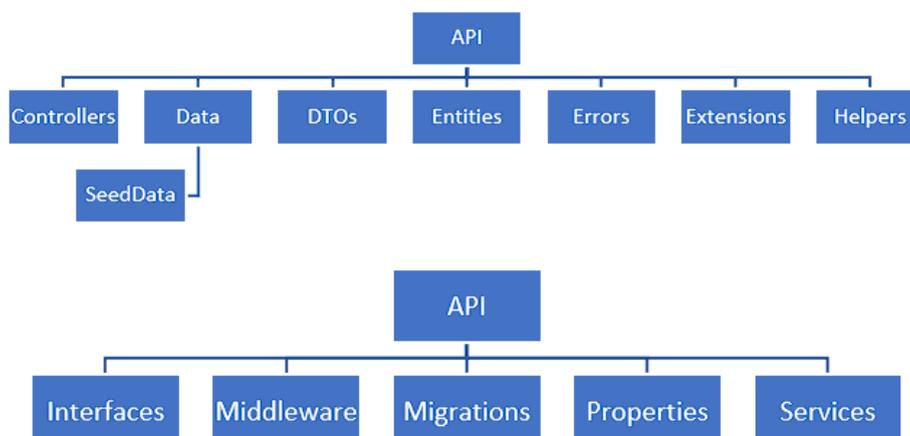


Figura 62. Estructura del proyecto "API"

Fuente: Los autores

Controllers: Objetos que se ejecutan al recibir una petición, actuando como intermediario entre el modelo y la vista.

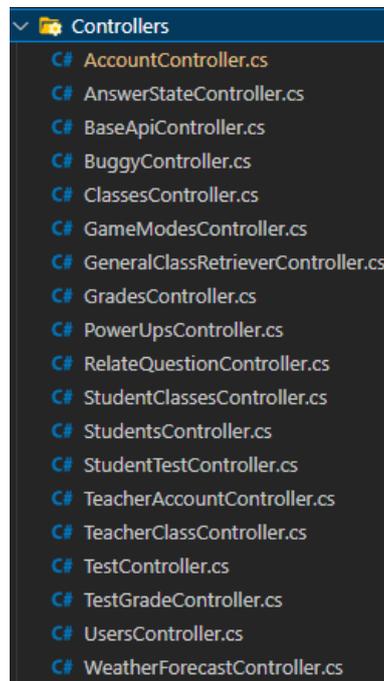


Figura 63. Contenido del directorio "API/Controllers"

Data: Es donde se define el contexto de los datos (DataContext), además de los diferentes repositorios de los modelos, cabe destacar que se usó el patrón de diseño conocido como "patrón de repositorio".

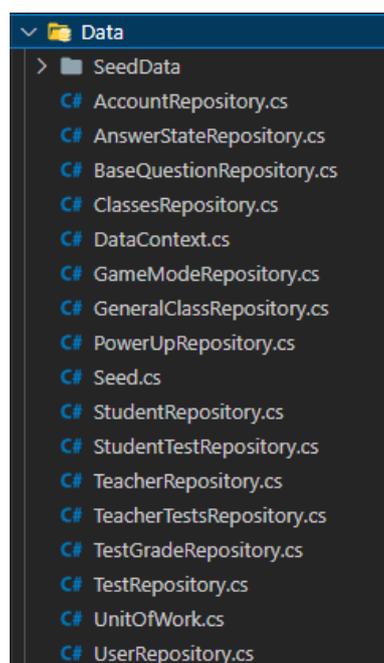


Figura 64. Contenido del directorio "API/Data"

Data/Seed Data: Archivos con extensión “. json” en los que se encuentran datos semillas para la inserción de datos iniciales en ciertas tablas, por ejemplo, las asignaturas de la universidad (GeneralClasses.json)

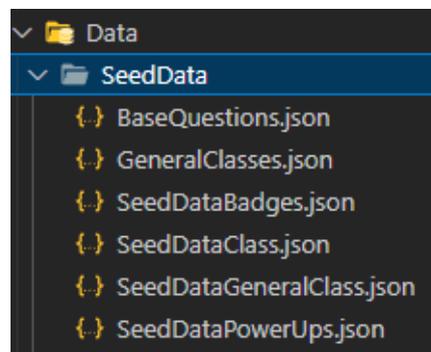


Figura 65. Contenido del directorio "API/Data/SeedData"

DTOs: Aquí se definen las clases que actúan como objetos de transferencia de datos, es decir, los que se encargan de llevar o traer datos entre procesos.

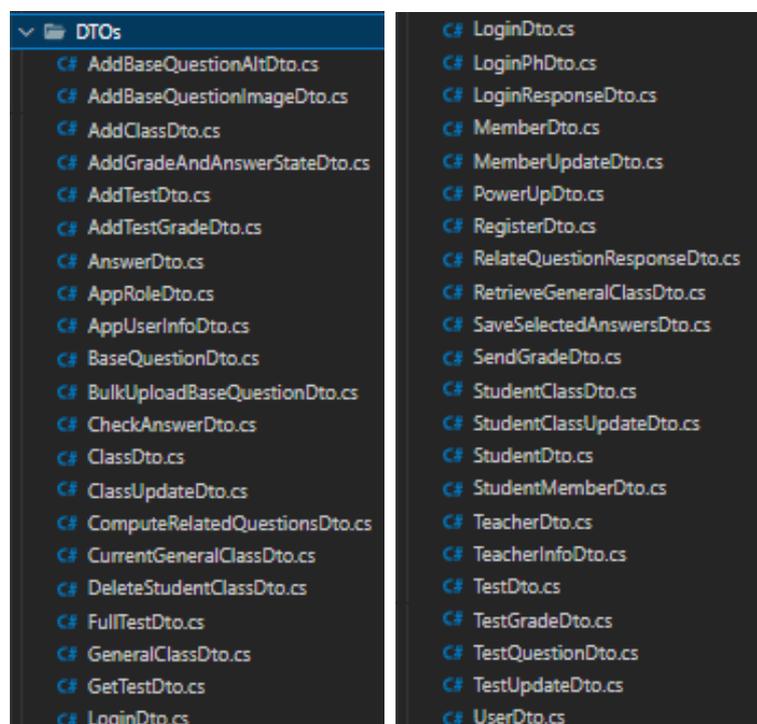


Figura 66. Contenido del directorio "API/DTOs"

Entities: Donde se definen las entidades o modelos del sistema, se utiliza para representar los valores de una sola fila de una tabla de base de datos.

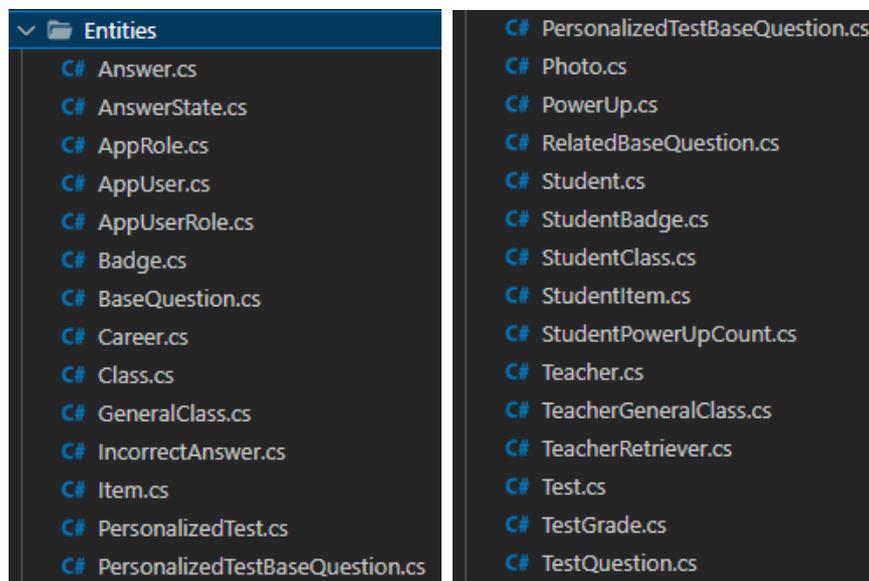


Figura 67. Contenido del directorio "API/Entities"

Errors: Clases que manejan los diferentes errores que puedan ocurrir mientras el sistema está operativo, mostrando los respectivos códigos de errores, mensajes y detalles.

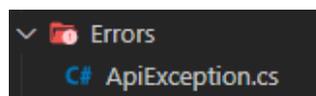


Figura 68. Contenido del directorio "API/Errors"

Extensions: En esta carpeta se definen las clases usadas para agregar o extender métodos a tipos existentes sin crear un nuevo tipo derivado, recompilar o modificar el tipo original.

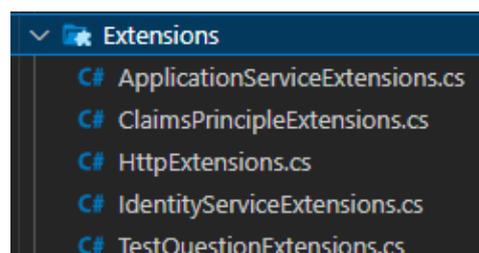


Figura 69. Contenido del directorio "API/Extensions"

Helpers: Carpeta en donde se encuentran métodos utilitarios que pueden ser utilizados entre distintas clases, por ejemplo, en la definición de los mapeos a través del uso de AutoMapper.

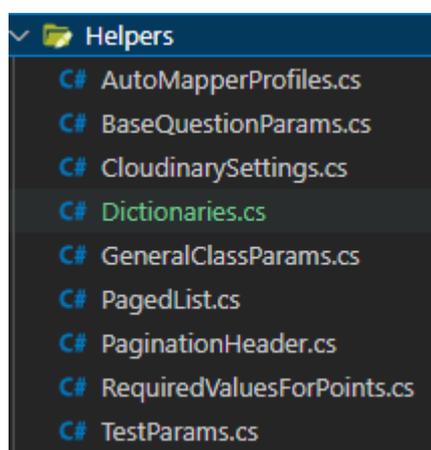


Figura 70. Contenido del directorio "API/Helpers"

Interfaces: Aquí se definen todas las interfaces del proyecto, la mayoría corresponden a los respectivos repositorios cuyas implementaciones se encuentran en la carpeta previamente descrita "Data".

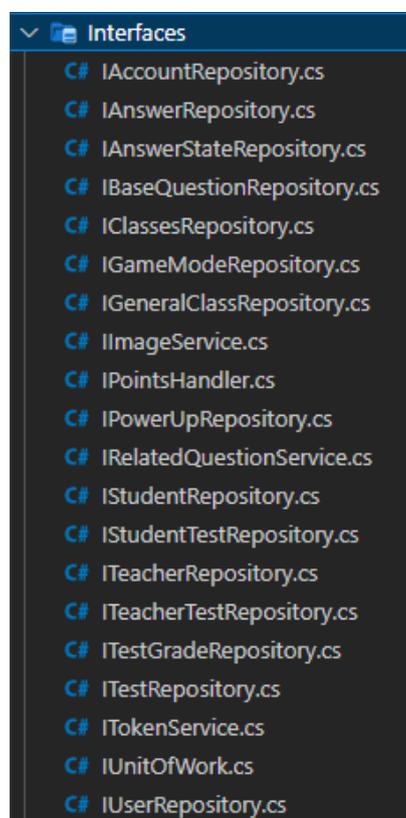


Figura 71. Contenido del directorio "API/Interfaces"

Middleware: En esta carpeta que encuentran los componentes que son ejecutados en cada petición realizada al sistema.

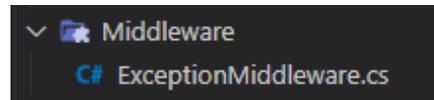


Figura 72. Contenido del directorio "API/Middleware"

Migrations: Al utilizar el paradigma "Code first", aquí se encuentra el historial de migraciones auto generado a partir de las definiciones realizadas en la carpeta "Entities".

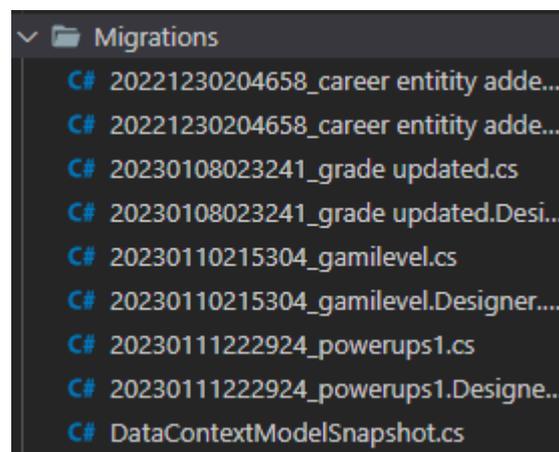


Figura 73. Contenido del directorio "API/Migrations"

Properties: Propiedades asociadas a la ejecución de la API, en las que se encuentra el archivo "launchSettings.json" para configurar variables de entorno o la URL de la aplicación entre otras opciones.

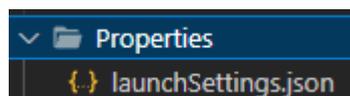


Figura 74. Contenido del directorio "API/Properties"

Services: Clases que representan servicios necesarios para que otros componentes del proyecto puedan operar correctamente, por ejemplo, el servicio de creación de Json Web Tokens para la autenticación del usuario.

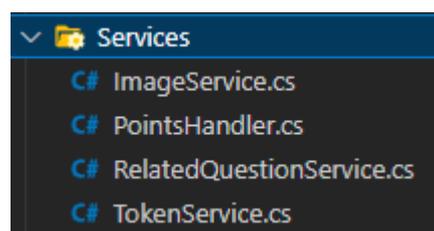


Figura 75. Contenido del directorio "API/Services"

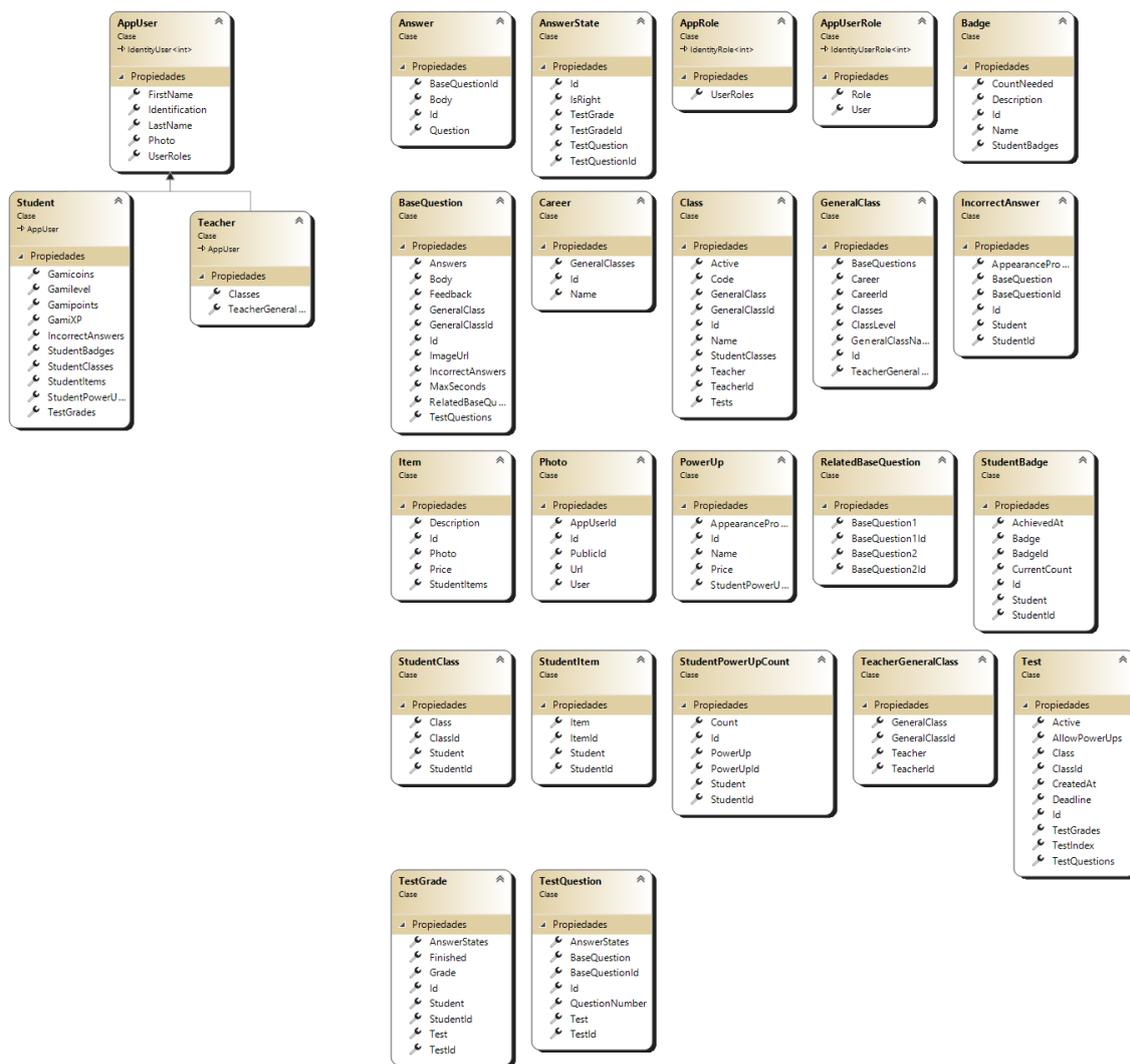
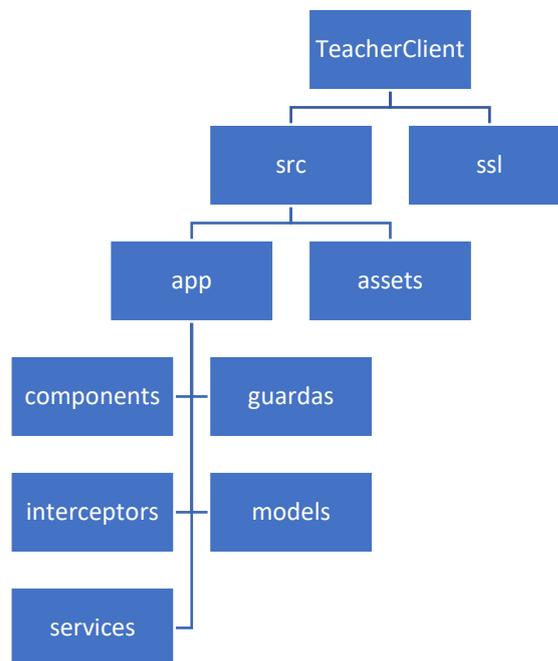


Figura 76. Diagrama de clases generado por Visual Studio

Estructura del proyecto (Nivel 2 - TeacherClient)



src: Este directorio es básicamente la fuente de nuestro proyecto, ya que es aquí donde se encuentra todo lo necesario para poder lanzar un proyecto angular.

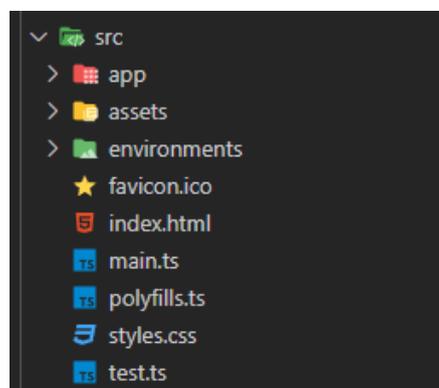


Figura 77. Contenido del directorio "TeacherClient/src"

SSL: Aquí únicamente se encuentra el certificado Secure Sockets Layer (capa de sockets seguros) junto a su respectiva clave para mantener segura una conexión a Internet, así como para proteger cualquier información confidencial.

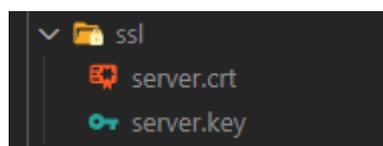


Figura 78. Contenido del directorio "TeacherClient/ssl"

components: En esta carpeta podemos encontrar todos los componentes que se manejan dentro de la administración docente.

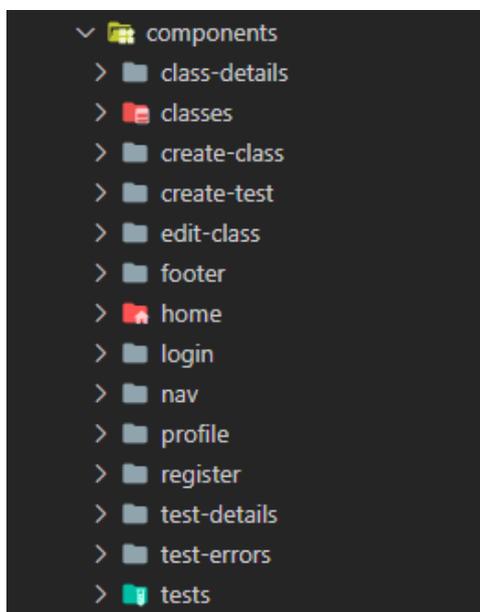


Figura 79. Contenido del directorio "TeacherClient/components"

guards: Aquí es donde se maneja la autorización de los usuarios, para así no cargar rutas a las que los usuarios no tengan acceso.

```
auth.guard.ts X
1 import { Injectable } from '@angular/core';
2 import { CanActivate } from '@angular/router';
3 import { Observable } from 'rxjs';
4 import { AccountService } from '../services/account.service';
5 import { ToastrService } from 'ngx-toastr';
6 import { map } from "rxjs/operators";
7
8 @Injectable({
9   providedIn: 'root'
10 })
11 export class AuthGuard implements CanActivate {
12
13   constructor(private accountService: AccountService, private toastr: ToastrService){}
14
15   canActivate(): Observable<boolean> {
16     return this.accountService.currentUser$.pipe(
17       map(user => {
18         if (user) return true;
19         this.toastr.error('No autorizado!');
20         return false;
21       })
22     )
23   }
24
25 }
26
```

Figura 80. Contenido del archivo "auth.guard.ts"

interceptors: En este se presentan los interceptores, los cuales permiten interceptar las solicitudes HTTP antes de que se envíen al servidor, para agregar información a las request, manipular datos, entre otras utilidades.

```

jwt.interceptor.ts M X
1  import { Injectable } from '@angular/core';
2  import { HttpRequest, HttpResponse, HttpEvent, HttpInterceptor } from '@angular/common/http';
3  import { Observable, take } from 'rxjs';
4  import { AccountService } from '../services/account.service';
5  import { User } from '../models/user';
6
7  @Injectable()
8  export class JwtInterceptor implements HttpInterceptor {
9
10     constructor(private accountService: AccountService) {}
11
12     intercept(request: HttpRequest<unknown>, next: HttpHandler): Observable<HttpEvent<unknown>> {
13
14         let currentUser: User
15
16         this.accountService.currentUser$.pipe(take(1)).subscribe(user => currentUser = user);
17         if(currentUser){
18             request = request.clone({
19                 setHeaders: {
20                     Authorization: `Bearer ${currentUser.token}`
21                 }
22             });
23         }
24
25         return next.handle(request);
26     }
27 }
28

```

Figura 81. Contenido del archivo "jwt.interceptor.ts"

models: En esta carpeta encontramos los modelos de los diferentes objetos, los cuales permiten la utilización de los mismos desde clases externas.

```

user.ts X
1  export interface User {
2      username: string;
3      token: string;
4      firstname: string;
5      lastname: string;
6      picUrl: string;
7  }

```

Figura 82. Estructura del modelo "user"

services: Aquí podemos encontrar los microservicios realizados los cuales tienen la función de realizar todas las solicitudes directamente a la API.

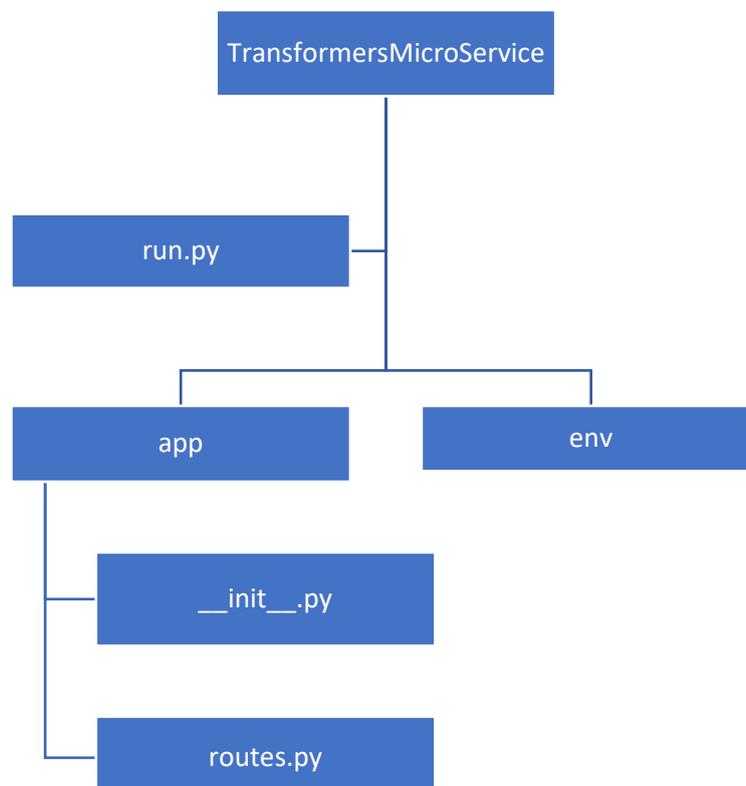
```

1 export class AccountService {
2
3   baseUrl = 'https://localhost:5001/api/';
4   private currentUserSource = new ReplaySubject<User | null>(1);
5   currentUser$ = this.currentUserSource.asObservable();
6
7   constructor(private http: HttpClient, private router: Router) { }
8
9   login(model: any) {
10    return this.http.post<User>(this.baseUrl + 'account/login', model).pipe(
11      map((response: User) => {
12        const user = response;
13        if (user) {
14          localStorage.setItem('user', JSON.stringify(user));
15          this.currentUserSource.next(user);
16        }
17      })
18    )
19  }
20 }

```

Figura 83. Fragmento del servicio "account"

Estructura del proyecto (Nivel 2 - TransformersMicroService)



run.py: Ejecución de la aplicación en modo development.

```
TransformersMicroService > run.py
1  from app import app
2
3  if __name__ == "__main__":
4  app.run(debug=False, port=4000)
```

Figura 84. Contenido del archivo "run.py"

app: Carpeta con los scripts asociados a las funcionalidades del micro servicio.

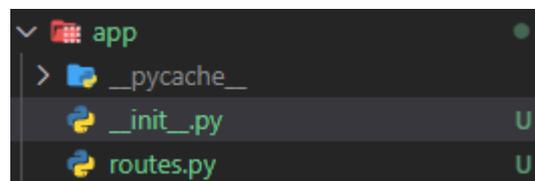


Figura 85. Contenido del directorio "TransformersMicroService/app"

env: Un entorno virtual de Python donde se instalan Flask y otras dependencias.

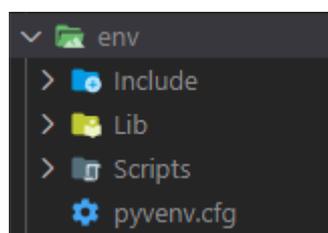


Figura 86. Contenido del directorio "TransformersMicroService/env"

app/__init__.py: Importación de la librería de Flask y las rutas construidas para el servicio de relaciones entre preguntas.

```
TransformersMicroService > app > __init__.py >
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  from app import routes
```

Figura 87. Contenido del archivo "__init__.py"

app/routes.py: Encargado de mapear las URL a una función específica que manejará la lógica asociada a esta, en este caso, la operación de calcular la similitud entre preguntas y retornar dichas preguntas relacionadas respecto a las suministradas en el cuerpo de la petición.

```
@app.route('/relate-question', methods=['POST'])
def relate_question():
    req_data = request.get_json()

    general_class_id = req_data['generalClassId']
    input_questions_ids = req_data['baseQuestionIds']

    input_questions_unpack = []

    input_questions = []
    for input_question in input_questions_unpack:
        input_questions.append(input_question[0][1])
```

Figura 88. Fragmento del archivo "routes.py"

```
def compute_similarity(input_question_vec, base_questions_vecs, base_questions_list_tuple):

    similarity = cosine_similarity(
        [input_question_vec],
        base_questions_vecs[:])

    similarity_list = list(similarity)
    similarity_list = similarity_list[0]

    related_questions = []

    for index, cosine_distance in enumerate(similarity_list):
        if cosine_distance > 0.65:
            related_questions.append(base_questions_list_tuple[index][0])

    return related_questions
```

Figura 89. Fragmento del archivo "routes.py"

Documentación de los endpoints

Endpoints asociadas al docente:

POST - Add class: /api/teacher-classes

Crea una nueva clase a partir de la especificación del id de la asignatura (generalClassId) y el nombre de la clase a crear.

Headers - Authorization: Bearer {{token}}

Body: raw

```
{
  "name": "PS 2023",
  "generalClassId": 395
}
```

POST - Add test: /api/tests

Crea un nuevo test para la clase especificada. Se debe hacer referencia a los id de las preguntas (baseQuestions).

Headers - Authorization: Bearer {{token}}

Body: raw

```
{
  "classId": "1",
  "test": {
    "deadline": "2023-01-12T01:18:11",
    "allowPowerUps": true,
    "baseQuestions": [
      {
        "id": 2
      },
      {
        "id": 3
      },
      {
        "id": 4
      },
      {
        "id": 5
      }
    ]
  }
}
```

```
    },  
    {  
      "id": 6  
    },  
    {  
      "id": 7  
    },  
    {  
      "id": 8  
    }  
  ]  
}
```

POST - Upload base question: /api/teacher-tests/base-questions

Añade una nueva pregunta para poder ser referencia en el test a crear.

Body: formdata

File: null

ClassId: 1

Body: Enunciado de ejemplo

Difficulty: 1

Feedback: Feedback de la pregunta

Answers[0][body]: Respuesta 1

Answers[1][body]: Respuesta 2

GET - Get base questions by general class: /api/teacher-tests/base-questions?generalClassId=395

Obtiene las preguntas de una asignatura especificada a través del parámetro "generalClassId", el parámetro "pageSize" hace referencia al número de preguntas mostrado por cada llamada.

Respuesta

```
[
  {
    "id": 1,
    "maxSeconds": 120,
    "body": "Seleccione La mejor definici3n de variable",
    "imageUrl": null,
    "answers": [
      {
        "id": 1,
        "body": "Es una caracterstica que cambia o varía con el
tiempo y/o para diferentes personas u objetos bajo consideraci3n"
      },
      {
        "id": 2,
        "body": "Es el individuo u objeto en el que se mide una
variable. Resulta una sola medici3n o datos cuando una variable se mide en
realidad en una unidad experimental."
      },
      {
        "id": 3,
        "body": "Es el conjunto de mediciones de inter3s para el
investigador"
      },
      {
        "id": 4,
        "body": "Es un subconjunto de mediciones seleccionado de la
poblaci3n de inter3s."
      }
    ],
    "feedback": null
  },
  {
    "id": 2,
    "maxSeconds": 120,
    "body": "Seleccione La mejor definici3n de unidad experimental",
    "imageUrl": null,
    "answers": [
      {
        "id": 5,
        "body": "Es el individuo u objeto en el que se mide una
variable. Resulta una sola medici3n o datos cuando una variable se mide en
realidad en una unidad experimental"
      },
      {
        "id": 6,
        "body": "Es una caracterstica que cambia o varía con el
tiempo y/o para diferentes personas u objetos bajo consideraci3n."
      },
      {
        "id": 7,
        "body": "Es el conjunto de mediciones de inter3s para el
investigador"
      },
      {
        "id": 8,
```

```

        "body": "Es un subconjunto de mediciones seleccionado de la
población de interés"
    },
    "feedback": null
},

```

Headers – Authorization: Bearer {{token}}

Parámetros

generalClassId: 395

pageNumber: 2

pageSize: 3

GET - Get grades: /api/grades/{testId}

Obtiene las notas de los estudiantes en el test especificado a través del testId.

Parámetros

- testId: id del test

Respuesta

```

[
  {
    "studentId": 10,
    "studentFirstName": "CHRISTIAN ISRAEL",
    "studentLastName": "LOPEZ MORA",
    "grade": 4
  }
]

```

Headers – Authorization: Bearer {{token}}

GET - Get general classes: /api/teacher-class/general-classes

Obtiene las asignaturas (generalClass) que el docente imparte en la universidad.

Respuesta

```

[ {
  "id": 394,
  "classLevel": 4,
  "generalClassName": "BASES DE DATOS",
  "careerName": "COMPUTACION"
},
{

```

```

    "id": 395,
    "classLevel": 5,
    "generalClassName": "PROYECTOS DE SOFTWARE",
    "careerName": "COMPUTACION"
  },
  {
    "id": 396,
    "classLevel": 7,
    "generalClassName": "INGENIERÍA DE SOFTWARE",
    "careerName": "COMPUTACION"
  }
]

```

GET - Get general classes: /api/teacher-class/general-classes/all

Obtiene todas las asignaturas de la universidad

Respuesta

```

[
  {
    "id": 6,
    "classLevel": 1,
    "generalClassName": "FÍSICA VECTORIAL",
    "careerName": "INGENIERIA AGRÍCOLA"
  },
  {
    "id": 7,
    "classLevel": 1,
    "generalClassName": "METEOROLOGÍA",
    "careerName": "INGENIERIA AGRÍCOLA"
  },
  {
    "id": 10,
    "classLevel": 1,
    "generalClassName": "QUÍMICA GENERAL",
    "careerName": "INGENIERIA AGRÍCOLA"
  },
  {
    "id": 147,
    "classLevel": 1,
    "generalClassName": "GEOMETRÍA ANALÍTICA Y TRIGONOMETRÍA",
    "careerName": "INGENIERIA AGRÍCOLA"
  },
  {
    "id": 154,
    "classLevel": 1,
    "generalClassName": "BOTÁNICA (ANATOMÍA Y TAXONOMÍA)",
    "careerName": "INGENIERIA AGRÍCOLA"
  }
]

```

Headers – Authorization: Bearer {{token}}

DEL - Remove student from class: /api/teacher-class/remove-student

Elimina a un estudiante de la clase especificada a través del classId.

Headers – Authorization: Bearer {{token}}

Body: raw

```
{
  "studentId": 1,
  "classId": 1
}
```

DEL - Remove test: /api/test/remove-test/{testId}

Elimina al test de una clase a través de la especificación del id del test (testId).

Headers – Authorization: Bearer {{token}}

POST - Relate question: /api/relate-question

Relaciona las preguntas especificadas en el body, con las que ya se encuentran en la base de datos, el cálculo se hace respecto a las preguntas de la misma asignatura, definida a través de la propiedad "generalClassId".

Headers – Authorization: Bearer {{token}}

Body: raw

```
{
  "generalClassId": 395,
  "baseQuestionIds": [
    15,
    17,
    19,
    23
  ]
}
```

PUT - Update class: /api/teacher-classes

Actualiza el nombre de una clase a través de la especificación del id.

Headers – Authorization: Bearer {{token}}

Body: raw

```
{
  "id": 1,
  "name": "PROYECTOS DE SOFTWARE 2023"
}
```

GET - Get careers: /api/teacher-classes/careers

Obtiene las carreras de la universidad.

Respuesta

```
[
  {
    "id": 1,
    "name": "INGENIERIA AGRÍCOLA"
  },
  {
    "id": 2,
    "name": "INGENIERÍA AGROFORESTAL"
  },
  {
    "id": 3,
    "name": "INGENIERIA AMBIENTAL"
  },
  {
    "id": 4,
    "name": "INGENIERÍA EN RIESGOS DE DESASTRES"
  },
  {
    "id": 5,
    "name": "AGROINDUSTRIA"
  },
  {
    "id": 6,
    "name": "ELECTRÓNICA Y AUTOMATIZACIÓN"
  },
  {
    "id": 7,
    "name": "COMPUTACION"
  },
  {
    "id": 8,
    "name": "MEDICINA VETERINARIA"
  },
  {
    "id": 9,
    "name": "TURISMO"
  },
  {
    "id": 10,
    "name": "ADMINISTRACIÓN DE EMPRESAS"
  },
  {
    "id": 11,
```

```
[
  {
    "name": "ADMINISTRACIÓN PÚBLICA"
  }
]
```

Headers – Authorization: Bearer {{token}}

Endpoints en común entre los docentes y estudiantes.

GET - Get tests: /api/tests

Obtiene los tests asignados a un estudiante de las clases que se haya unido.

Parámetros

- **“active”:** true=muestra solo los tests activos
- **“classId”:** especifica el id de la clase del test

Respuesta

```
[
  {
    "id": 2,
    "testIndex": 1,
    "active": false,
    "allowPowerUps": true,
    "createdAt": "2023-01-15T15:57:50.0283638",
    "deadline": "2023-01-12T01:18:11",
    "className": "PROYECTOS DE SOFTWARE 2023"
  }
]
```

GET - Get tests: /api/tests/{testId}

Obtiene los detalles del test asignado a un estudiante {testId}, estos incluyen el contenido de las preguntas.

Params

active: false

classId: 2

Respuesta

```
{
```

```

"id": 2,
"testIndex": 1,
"allowPowerUps": true,
"createdAt": "2023-01-15T15:57:50.0283638",
"deadline": "2023-01-12T01:18:11",
"className": "PROYECTOS DE SOFTWARE 2023",
"testQuestions": [
  {
    "id": 2,
    "questionNumber": 1,
    "baseQuestion": {
      "id": 2,
      "maxSeconds": 120,
      "body": "Seleccione La mejor definici3n de unidad
experimental",
      "imageUrl": null,
      "answers": [
        {
          "id": 5,
          "body": "Es el individuo u objeto en el que se mide
una variable. Resulta una sola medici3n o datos cuando una variable se mide
en realidad en una unidad experimental"
        },
        {
          "id": 6,
          "body": "Es una caracter3stica que cambia o var3a con
el tiempo y/o para diferentes personas u objetos bajo consideraci3n."
        },
        {
          "id": 7,
          "body": "Es el conjunto de mediciones de inter3s para
el investigador"
        },
        {
          "id": 8,
          "body": "Es un subconjunto de mediciones seleccionado
de la poblaci3n de inter3s"
        }
      ],
      "feedback": null
    }
  },
  {
    "id": 3,
    "questionNumber": 2,
    "baseQuestion": {
      "id": 3,
      "maxSeconds": 120,
      "body": "Seleccione la mejor definici3n de poblaci3n",
      "imageUrl": null,
      "answers": [
        {
          "id": 9,
          "body": "Es el conjunto de mediciones de inter3s para
el investigador"
        },
        {

```


PUT - Update member: /api/users

Actualiza las propiedades de un usuario (foto de perfil y nombre de usuario).

Headers – Authorization: Bearer {{token}}

Body: formdata

Pic: “foto.png”

Username: fernan

GET - Get students in class: /api/classes/{classId}

Obtiene a los estudiantes de la clase especificada a través de su id {classId}.

Respuesta

```
[
  {
    "id": 8,
    "identification": "1311118440",
    "username": "sandroantonio",
    "email": "sandro.palau@espam.edu.ec",
    "firstName": "SANDRO ANTONIO",
    "lastName": "PALAU DELGADO",
    "photoUrl":
    "https://res.cloudinary.com/dpxbur1md/image/upload/v1672265557/profile_pics/default-user-icon_qzkyhb.jpg"
  },
  {
    "id": 10,
    "identification": "1315223394",
    "username": "christianisrael",
    "email": "christian.lopez@espam.edu.ec",
    "firstName": "CHRISTIAN ISRAEL",
    "lastName": "LOPEZ MORA",
    "photoUrl":
    "https://res.cloudinary.com/dpxbur1md/image/upload/v1672265557/profile_pics/default-user-icon_qzkyhb.jpg"
  }
]
```

Headers – Authorization: Bearer {{token}}

GET - Get classes: /api/classes

Obtiene las clases a las que se ha unido (en caso de ser estudiante) o creado (en caso de ser docente).

Respuesta

```
[
  {
    "id": 1,
    "name": "PROYECTOS DE SOFTWARE 2023",
    "code": "YSTUJTK",
    "generalClass": {
      "id": 395,
      "classLevel": 5,
      "generalClassName": "PROYECTOS DE SOFTWARE",
      "careerName": "COMPUTACION"
    }
  }
]
```

Headers – Authorization: Bearer {{token}}

GET - Get member: /api/users

Obtiene información del usuario con la sesión abierta junto al rol asociado al mismo.

Respuesta

```
{
  "id": 8,
  "identification": "1311118440",
  "username": "sandroantonio",
  "firstName": "SANDRO ANTONIO",
  "lastName": "PALAU DELGADO",
  "photoUrl":
  "https://res.cloudinary.com/dpxburldm/image/upload/v1672265557/profile_pics/default-user-icon_qzkyhb.jpg",
  "userRoles": [
    {
      "userId": 8,
      "roleName": "Student"
    }
  ]
}
```

Headers – Authorization: Bearer {{token}}

Endpoints asociados al estudiante

POST - Join class: /api/student-classes/join-class/{code]

Utilizado para que un estudiante se una a la clase especificada a través del código de dicha clase "{code}".

Headers – Authorization: Bearer {{token}}

GET - Get students: /api/students/{studentId}

Obtiene las propiedades públicas de un estudiante a través de su id {studentId}.

Respuesta

```
{
  "id": 8,
  "identification": "1311118440",
  "gamilevel": 3,
  "gamipoints": 203,
  "gamiXP": 123,
  "gamicoins": 40,
  "username": "sandroantonio",
  "email": "sandro.palau@espam.edu.ec",
  "firstName": "SANDRO ANTONIO",
  "lastName": "PALAU DELGADO",
  "photoUrl":
  "https://res.cloudinary.com/dpxbur1md/image/upload/v1672265557/profile_pics/default-user-icon_qzkyhb.jpg"
}
```

Headers – Authorization: Bearer {{token}}

GET - Get available power ups: /api/power-ups/availables

Obtiene los power ups que el estudiante puede comprar, es decir, los que no han alcanzado el límite de 5 unidades en el inventario.

Respuesta

```
[
  {
    "id": 1,
    "name": "FiftyFifty",
    "price": 60
  },
  {
    "id": 2,
    "name": "Bomb",
    "price": 40
  },
  {
    "id": 3,
    "name": "DoubleChances",
    "price": 60
  },
  {
    "id": 4,
    "name": "PointsX2",

```

```

    "price": 60
  },
  {
    "id": 5,
    "name": "PointsX3",
    "price": 60
  },
  {
    "id": 6,
    "name": "PointsX4",
    "price": 60
  },
  {
    "id": 7,
    "name": "Plus30Seconds",
    "price": 60
  },
  {
    "id": 8,
    "name": "Skip",
    "price": 30
  }
]

```

Headers – Authorization: Bearer {{token}}

GET - Get power ups: /api/power-ups

Obtiene los power ups en el inventario del estudiante.

Respuesta

```

[
  {
    "id": 2,
    "name": "Bomb",
    "price": 40
  },
  {
    "id": 3,
    "name": "DoubleChances",
    "price": 60
  }
]

```

Headers – Authorization: Bearer {{token}}

PUT - Buy power up: /api/power-ups/{powerUpId}

Actualiza el número de power ups en el inventario del power up especificado por el id {powerUpId}.

Headers – Authorization: Bearer {{token}}

PUT - Use power up: /api/power-ups/in-game/{powerUpId}

Usado para activar un power up {powerUpId} en la partida, por lo tanto, decremente el número de este tipo de power ups en el inventario del estudiante en una unidad.

Headers – Authorization: Bearer {{token}}

GET - Get obtained badges: /api/badges

Obtiene las insignias que el usuario ha adquirido.

Respuesta

```
[
  {
    "studentId": 10,
    "achievedAt": "2023-01-12T01:18:11",
    "badge": {
      "id": 1,
      "name": "FlawlessMatch",
      "description": "¡Primera partida sin errores!"
    }
  }
]
```

Headers – Authorization: Bearer {{token}}

GET - Get leaderboards: /api/leaderboards?context=class

Obtiene la tabla de clasificaciones de los estudiantes (50 primeros) por el contexto especificado usado como filtro, es decir, puede ser filtrado por clase, semestre, carrera, o toda la universidad.

Parámetros

- context: [class, level, career, global]

Respuesta

```
[
  {
    "id": 8,
    "identification": "1311118440",
    "gamilevel": 3,
    "gamipoints": 203,
    "gamiXP": 123,
    "gamicoins": 40,
    "username": "sandroantonio",
    "email": "sandro.palau@espam.edu.ec",
  }
]
```

```

        "firstName": "SANDRO ANTONIO",
        "lastName": "PALAU DELGADO",
        "photoUrl":
"https://res.cloudinary.com/dpxburldm/image/upload/v1672265557/profile_pics/default-user-icon_qzkyhb.jpg"
    },
    {
        "id": 10,
        "identification": "1315223394",
        "gamilevel": 2,
        "gamipoints": 193,
        "gamiXP": 69,
        "gamicoins": 23,
        "username": "christianisrael",
        "email": "christian.lopez@espam.edu.ec",
        "firstName": "CHRISTIAN ISRAEL",
        "lastName": "LOPEZ MORA",
        "photoUrl":
"https://res.cloudinary.com/dpxburldm/image/upload/v1672265557/profile_pics/default-user-icon_qzkyhb.jpg"
    }
]

```

Headers – Authorization: Bearer {{token}}

Params

context: class

POST - Add grade: /api/grades

Envía el test a revisión, para hacer los respectivos cálculos de los puntos y la nota con base en el rendimiento del estudiante en el test especificado por la propiedad "testId". Además, se tiene que especificar la propiedad "answeredAtSecond" por cada una de las respuestas a las preguntas del test, esta propiedad hace referencia al segundo exacto en el que el estudiante escogió su respuesta.

Headers – Authorization: Bearer {{token}}

Body: raw

```

{
  "testId": 2,
  "selectedAnswers": [
    {
      "testQuestionId": 2,

```

```

        "answerId": 16,
        "answeredAtSecond": 25
    },
    {
        "testQuestionId": 3,
        "answerId": 20,
        "answeredAtSecond": 101
    },
    {
        "testQuestionId": 4,
        "answerId": 25,
        "answeredAtSecond": 80
    },
    {
        "testQuestionId": 5,
        "answerId": null,
        "answeredAtSecond": 0
    },
    {
        "testQuestionId": 6,
        "answerId": 38,
        "answeredAtSecond": 90
    }
    ]
}

```

GET - Get questions by class: `/api/game-modes/general-classes/{generalClassId}`

Obtiene 10 preguntas para una partida del modo de juego libre por asignatura `{generalClassId}`. Hasta 4 preguntas de este test pueden provenir de preguntas recomendadas (IA), el resto son aleatorias.

Respuesta

```

[
  {
    "id": 22,
    "maxSeconds": 120,
    "body": " Determine la varianza de la siguiente distribución de variable aleatoria  $x = 0,1,2,3,4$ ;  $p(4) = 1/2$ ;  $p(3) = 1/4$ ;  $p(2) = 1/8$ ;  $p(1) = 1/16$ ;  $p(0) = 1/16$ ",
    "imageUrl": null,
    "answers": [
      {

```

```

        "id": 85,
        "body": "1,4336"
      },
      {
        "id": 86,
        "body": "1,1973"
      },
      {
        "id": 87,
        "body": "1,4685"
      },
      {
        "id": 88,
        "body": "1,2118"
      }
    ],
    "feedback": null
  },

```

Headers – Authorization: Bearer {{token}}

GET - Get questions by class level: /api/game-modes/class-levels/{classLevel}

Obtiene 10 preguntas para una partida del modo de juego libre por semestre {classLevel}. Dicho semestre debe de ser de al menos una de las clases que el estudiante se haya unido. Hasta 4 preguntas de este test pueden provenir de preguntas recomendadas (IA), el resto son aleatorias.

```

[
  {
    "id": 23,
    "maxSeconds": 120,
    "body": "Determine la desviación estándar de la siguiente distribución de variable aleatoria  $x = 0,1,2,3,4$ ;  $p(4) = 1/2$ ;  $p(3) = 1/4$ ;  $p(2) = 1/8$ ;  $p(1) = 1/16$ ;  $p(0) = 1/16$ ",
    "imageUrl": null,
    "answers": [
      {
        "id": 89,
        "body": "1,1973"
      },
      {
        "id": 90,
        "body": "1,4336"
      },
      {
        "id": 91,
        "body": "1,4685"
      },
      {
        "id": 92,
        "body": "1,2118"
      }
    ]
  }
]

```

```

    }
  ],
  "feedback": null
},

```

Headers – Authorization: Bearer {{token}}

GET - Get questions all: /api/game-modes/general-classes/shuffle

Obtiene 10 preguntas para una partida del modo de juego libre combinado, es decir de cualquier asignatura o nivel perteneciente a clases que se haya unido. Hasta 4 preguntas de este test pueden provenir de preguntas recomendadas (IA), el resto son aleatorias.

```

[
  {
    "id": 23,
    "maxSeconds": 120,
    "body": "Determine la desviación estándar de la siguiente
distribución de variable aleatoria  $x = 0,1,2,3,4$ ;  $p(4) = 1/2$ ;  $p(3) = 1/4$ ;  $p(2) = 1/8$ ;  $p(1) = 1/16$ ;  $p(0) = 1/16$ ",
    "imageUrl": null,
    "answers": [
      {
        "id": 89,
        "body": "1,1973"
      },
      {
        "id": 90,
        "body": "1,4336"
      },
      {
        "id": 91,
        "body": "1,4685"
      }
    ]
  },
  "feedback": null
},
{

```

Headers – Authorization: Bearer {{token}}