

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

VIRTUAL-POLI, PLATAFORMA WEB DISEÑADA PARA
COMPLEMENTAR LA EDUCACIÓN ACADÉMICA, GENERADO
POR ESTUDIANTES PARA ESTUDIANTES

DESARROLLO DE BACKEND

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
CIENCIAS DE LA COMPUTACIÓN

JHOSEL ALEXANDER GUILLIN FIERRO

jhosel.guillin@epn.edu.ec

DIRECTOR: MSc. HERNÁN DAVID ORDOÑEZ CALERO

hernan.ordonez@epn.edu.ec

DMQ, febrero 2024

CERTIFICACIONES

Yo, JHOSEL ALEXANDER GUILLIN FIERRO declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



JHOSEL ALEXANDER GUILLIN FIERRO

Certifico que el presente trabajo de integración curricular fue desarrollado por JHOSEL ALEXANDER GUILLIN FIERRO, bajo mi supervisión.

HERNAN DAVID ORDOÑEZ CALERO

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

JHOSEL ALEXANDER GUILLIN FIERRO

MSC. HERNAN DAVID ORDOÑEZ CALERO

LEONARDO MIJAIL ANDRADE CANDO

RICARDO ANDRES ERAZO TORRES

DEDICATORIA

Dedico este trabajo a mi tía Judith Fierro, por su apoyo incondicional durante mis últimos años de estudio, que sin ella no lograría estar aquí, en la etapa final de la culminación de mis estudios universitarios, la considero como una madre porque a pesar de los obstáculos que me presentó la vida, ella jamás renunció a mí, incluso sin ser su deber. Su amor incondicional con todo el mundo, su fuerza a pesar de los percances y su sabiduría en momentos de agonía, son las enseñanzas que me ha transmitido para seguir en este camino incierto denominado vida.

AGRADECIMIENTO

Agradezco a Dios por guiarme en cada paso que me ha llevado hasta este momento, por darme la fortaleza y sabiduría necesaria para seguir mi camino y enfrentar cada obstáculo.

A mi madre, por la vida y las largas noches de estudio junto a ella, lo cual me demostró que con un poco de esfuerzo y un par de lágrimas, el "no puedo" no existe.

A mi abuelita, mi segunda madre, que, mediante su bendición, día a día he ido escalando y venciendo las adversidades que me ha presentado la vida.

A mi tía Judith, mi tercera madre, que me acogió entre sus brazos en los momentos más difíciles de mi vida.

A mi hermana, por siempre buscar la forma de hacerme reír con sus ocurrencias y considerarme como un ejemplo a seguir.

A mi papá José, que me enseñó que un vínculo de sangre no establece una relación familiar, un nexo para ser padre e hijo.

A mis amigos, Leonardo Andrade, Ricardo Erazo, Jonathan Jiménez y Carlos Páez, que han sabido tolerar mis bromas fuera de lugar, pero que siempre han estado ahí, ofreciendo un consejo sabio, junto a un chiste, que pocos entenderán.

A mi tutor de trabajo, el Msc. Hernán Ordoñez, por sus enseñanzas durante todo este proceso.

A Boris Astudillo y Patricio Paccha por sus guías, que han permitido la culminación exitosa de este proyecto.

Y finalmente, a la persona que llegó en el momento menos esperado a mi vida, la niña de mis ojitos, Anahis, quien, con un par de abrazos, tan cálidos como el sol, me desprende de mis miedos y ansiedades; quien, con un consejo, me contiene de tomar decisiones impulsivas; quien me demuestra, que no necesito llegar a la luna, porque al mirar sus ojos veo el universo mismo.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco teórico	3
E-Learning.....	3
Principios del E-Learning	4
Desarrollo Backend	5
API REST	6
¿Backend y E-learning?.....	6
Código Limpio (Clean Code).....	7
Base de Datos Relacional vs. Base de Datos No Relacional	8
Python	9
Django.....	9
MongoDB.....	9
Scrum	10
Roles de Scrum	11
Artefactos de Scrum	11
Eventos de Scrum	12
Kanban	13
Estimación de esfuerzo.....	14
2 METODOLOGÍA.....	14
2.1 Inicio	15
Definición del objetivo del proyecto.....	15
Requerimientos del sistema.....	15
Stack de programación y diagrama de componentes	16
Equipo Scrum	16

Historias de usuario	17
Product Backlog.....	18
Planificación de lanzamiento.....	19
2.2 Planificación del Sprint.....	19
Sprint 1	19
Sprint 2	20
Sprint 3.....	21
2.3 Desarrollo del Sprint	22
Sprint 1	22
Reuniones Diarias de Scrum	28
Pruebas unitarias.....	28
2.4 Revisión del Sprint.....	32
Sprint 1	32
Sprint 2.....	33
Sprint 3.....	34
2.5 Retrospectiva del Sprint.....	34
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	36
3.1 Resultados	36
3.2 Conclusiones	39
3.3 Recomendaciones	40
4 REFERENCIAS BIBLIOGRÁFICAS	41
5 ANEXOS.....	44
ANEXO I. Historias de Usuario	44
ANEXO II. Tableros Kanban	51
ANEXO III. Modelo de base de datos	53
ANEXO IV. Código respecto al componente desarrollado	54
ANEXO V. Pruebas unitarias	63
ANEXO VI. Tableros Kanban – Flujo de tareas.....	85

RESUMEN

Durante los acontecimientos ocurridos por la pandemia de COVID-19, muchas instituciones educativas alrededor del mundo tuvieron que transferir sus actividades presenciales a un tipo de educación diferente como es la “educación virtual”. La educación virtual o e-learning surgió en los años noventa como un mecanismo de ayuda para la educación que se realizaba a distancia y se limitaba al uso del computador personal [1], pero hoy en día el concepto de e-learning tiene un mayor enfoque ocasionado por la evolución tecnológica de los últimos años.

El presente proyecto tiene como objetivo el desarrollo de la programación backend de la plataforma web “Virtual-POLI” para complementar la educación académica generada por estudiantes para estudiantes de la Facultad de Ingeniería en Sistemas. Para su desarrollo se combinó el marco de trabajo de Scrum que se utilizó para la planificación del proyecto y la metodología ágil de Kanban para la gestión del flujo de trabajo y el seguimiento de las tareas. Finalmente, en la fase de pruebas y evaluación junto con la fase de análisis, se entenderá la funcionalidad de cada punto de acceso o endpoint desarrollado y como este contribuye en desempeño para el usuario final.

PALABRAS CLAVE: e-learning, evolución tecnológica, programación backend, metodología ágil, endpoint, usuario final.

ABSTRACT

During the events triggered by the COVID-19 pandemic, many educational institutions worldwide had to shift their in-person activities to a different form of education known as "virtual education." Virtual education or e-learning emerged in the 1990s as a mechanism to support distance education, initially limited to the use of personal computers [1]. However, nowadays, the concept of e-learning has gained increased significance due to technological advancements in recent years.

The objective of the current project is to develop the backend programming of the web platform "Virtual-POLI" to complement academic education generated by students for students of the Faculty of Systems Engineering. For its development, the Scrum framework that was used for project planning and the agile Kanban methodology for workflow management and task tracking were combined. Finally, in the testing and evaluation phase together with the analysis phase, the functionality of each access point or endpoint developed will be understood and how it contributes to performance for the end user.

KEYWORDS: e-learning, technological evolution, backend programming, agile methodology, endpoint, end user.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El desarrollo del lado del servidor (backend) hace referencia al funcionamiento interno de cualquier plataforma, es decir, consta de todos los componentes esenciales con los que cada usuario respectivamente no interactúa [2].

El funcionamiento del backend empieza por el procesamiento de una solicitud en formato del protocolo de transferencia de hipertexto (HTTP) enviada desde una interfaz de usuario, vista o interfaz gráfica (frontend), con la cual los usuarios finales pueden relacionarse. Los elementos con los cuales interactúa el backend son [3]:

- Bases de datos para el almacenamiento y recuperación de información.
- Microservicios orientados a solventar las necesidades de los usuarios.
- Interfaces de programación de aplicaciones (APIs) externas para no crear todo desde cero.

Una arquitectura web enfocada al backend tiene como objetivo el desarrollo de programas que obtengan experiencias funcionales para cada usuario, es decir que permita una navegación fácil a través de cada página y use cada función sin tener que emplear dispositivos externos o comprender algún código complicado.

Por ejemplo, si se desea desarrollar un sitio web enfocado a establecer alarmas determinadas para cada usuario, el usuario no se debe preocupar por la infraestructura, dado que la plataforma está ejecutándose de forma externa en algún servidor, por tanto, se garantiza el funcionamiento de la plataforma independiente al computador local de cada usuario.

Considerando el enfoque de un producto mínimo viable (MVP) [4], dentro de una plataforma de e-learning es esencial una correcta gestión de usuarios, por lo tanto, un desarrollo correcto de la parte del backend permite administrar de manera eficiente el registro de usuarios, gestión de roles, autenticación y autorización, garantizando la seguridad y privacidad de la respectiva información de cada cliente.

Las plataformas de e-learning hoy en día integran herramientas externas como sistemas de gestión de aprendizaje (LMS), que facilitan la administración, entrega y seguimiento de cursos y contenido de aprendizaje en línea [5].

El propósito de este componente es enfocarse en el desarrollo de la programación backend de la plataforma web "Virtual-POLI" para complementar la educación académica generada por estudiantes para estudiantes de la Facultad de Ingeniería en Sistemas. Considerando

el uso de un marco de trabajo y una metodología ágil, se llevará a cabo una planificación del proyecto donde se definirán las tareas a realizarse y como se gestionarán. Se analizará la plataforma web para implementar el modelo de base de datos que mejor se adapte al flujo de escritura y solicitud de datos.

El proyecto se manejará bajo el concepto de desarrollo iterativo, es decir, que cada tarea se dividirá en ciclos, donde cada ciclo posee un levantamiento de requerimientos, planificación, implementación y pruebas.

1.1 Objetivo general

Desarrollar el backend para una plataforma web orientada a la educación virtual caso de aplicación "Virtual-POLI".

1.2 Objetivos específicos

1. Diseñar y desarrollar una arquitectura de software escalable, segura y eficiente de backend.
2. Diseñar y desarrollar una base de datos sólida y capaz de satisfacer las necesidades del sistema de manera efectiva.
3. Diseñar y desarrollar puntos de acceso para comunicación con el frontend.
4. Integrar servicios externos de ser necesario.

1.3 Alcance

El alcance de este componente tiene como objetivo el desarrollo de la programación backend de la plataforma web "Virtual-POLI" para complementar la educación académica generada por estudiantes para estudiantes de la Facultad de Ingeniería en Sistemas. Para cumplir con este proyecto se utilizará Scrum como marco de trabajo y Kanban para manejar el flujo de trabajo y el seguimiento de tareas.

Las fases consideradas para el desarrollo del proyecto en función del marco de trabajo de Scrum son:

- Inicio, para el análisis del objetivo y los requerimientos del sistema.
- Planificación del Sprint, para definir los objetivos a cumplirse durante el periodo del Sprint.
- Desarrollo del Sprint, para el desarrollo y las pruebas respectivas del producto final para cumplir con un incremento.

- Revisión del Sprint, para verificar el cumplimiento de cada objetivo considerado en la planificación del Sprint.
- Retrospectiva del Sprint, se considera los puntos fuertes y débiles llevados a cabo durante el desarrollo.

1.4 Marco teórico

En los últimos años, el aprendizaje en línea ha revolucionado la educación tradicional. El enfoque de aprender desde el hogar no solo ha simplificado el acceso a la enseñanza, si no también ha dado paso a tener horarios más flexibles y una adaptación del aprendizaje en función de los gustos de cada persona.

E-Learning

El E-Learning tiene como concepto la utilización de tecnología de Internet para la educación y permite a los estudiantes llevar un aprendizaje a un ritmo adecuado y en un entorno familiar y conocido como sus hogares. Entre los elementos esenciales del E-Learning se tiene la presentación de diversos contenidos en una gran variedad de formatos, la administración de la experiencia educativa, una red comunitaria de aprendices, así también creadores de contenido y especialistas en cada área educativa [6].

El E-Learning se clasificaría como un tipo de educación a distancia, por tanto, se ofrece a los estudiantes flexibilidad, accesibilidad, ahorro de tiempo y dinero, además de oportunidades para un aprendizaje personalizado y colaborativo, especialmente beneficioso para aquellos con restricciones geográficas, familiares o laborales [7].

La expansión del concepto de e-learning a nivel mundial va de la mano con el progreso de la era digital. El adquirir nuevos dispositivos electrónicos que facilitan la comunicación en diversos entornos, como hogares, instituciones educativas y empresariales, demuestra el cambio que la educación ha experimentado en los últimos años. La Figura 1.1 muestra el impacto positivo que este tipo de modalidad de educación ha tenido en la personalización del aprendizaje.

¿Qué aporta el e-learning a la mejora e innovación de la enseñanza?
Extender y facilitar el acceso a la formación a colectivos e individuos que no pueden acceder a la modalidad presencial
Incrementar la autonomía y responsabilidad del estudiante en su propio proceso de aprendizaje
Superar las limitaciones provocadas por la separación en espacio y/o tiempo del profesor-alumnos
Gran potencial interactivo entre profesor-alumno.
Flexibilidad en los tiempos y espacios educativos
Acceder a multiplicidad de fuentes y datos diferentes de los ofrecidos por el profesor en cualquier momento y desde cualquier lugar
Aprendizaje colaborativo entre comunidades virtuales de docentes y estudiantes

Figura 1.1 Aportación del e-learning a la mejora e innovación de la enseñanza [8]

Principios del E-Learning

En [9] se menciona que se tiene 10 principios orientados al e-learning, que son:

- 1) **Alineación con el plan de estudios:** se asegura que la pedagogía se ajuste con el plan de estudios, con objetivos claros, contenido adecuado y evaluaciones coherentes.
- 2) **Inclusión:** se fomenta la práctica inclusiva, tomando en cuenta diferentes niveles de logro, discapacidades físicas, variedad en grupos sociales y étnicos.
- 3) **Compromiso del alumno:** se debe crear un ambiente educativo donde se involucre de forma activa a los estudiantes.
- 4) **Enfoques innovadores:** se debe justificar el uso de tecnologías orientadas al aprendizaje que tengan como finalidad cumplir con un objetivo establecido.
- 5) **Aprendizaje efectivo:** se debe tener una variedad en los recursos de estudio para que el estudiante elija el que mejor se adapte a sus capacidades.
- 6) **Evaluación formativa:** se proporciona evaluaciones que retroalimenten el proceso de aprendizaje de los estudiantes.
- 7) **Evaluación sumativa:** se realizan evaluaciones comprensibles que no tengan como efecto un impacto emocional adverso.
- 8) **Coherencia, consistencia y transparencia:** se debe mantener la coherencia y consistencia entre los objetivos del plan de estudio, el contenido de enseñanza y las evaluaciones correspondientes.
- 9) **Facilidad de uso:** se garantiza que el E-Learning como nueva modalidad de educación es accesible y utilizable.

10) Rentabilidad: se debe asegurar que las herramientas tecnológicas sean asequibles para cada estudiante.

Desarrollo Backend

El desarrollo backend tiene como objetivo principal la programación del lado del servidor, por tanto, su enfoque es diferente al que se tiene con respecto al desarrollo frontend, dado que la programación backend se orienta en todo lo que no puede ser visto por el usuario, por ejemplo, en el caso particular de una aplicación web [10].

Hay diferentes aspectos que son abordados dentro del desarrollo backend, como la administración de las bases de datos, autenticación y autorización de sesiones de usuarios, manejo de interfaces de programación de aplicaciones (APIs), proveer una respuesta en función de las vistas desarrolladas por el frontend, entre otros [11].

El desarrollo de código backend puede ser considerado como la columna vertebral de cualquier sitio web o aplicación, por otro lado, el desarrollo de código frontend hace referencia a todo lo que es posible de ser visualizado de lado del usuario final. El backend solo es visible para el usuario cuando este no tiene funcionamiento [12].

Algunas herramientas utilizadas para el desarrollo backend son:

- **Lenguajes de programación:** es considerado como el idioma que se utiliza para escribir el código que permite el funcionamiento de una aplicación.
 - Python
 - PHP
 - JavaScript
- **Frameworks:** permite el uso de herramientas predeterminadas que facilitan el desarrollo dentro de un proyecto.
 - Django
 - Spring
 - Node.js
- **Bases de Datos:** permite la creación, recuperación, actualización y eliminación de los datos que se pretende manejar en una aplicación.
 - MongoDB

- MySQL
- Oracle

API REST

Una interfaz de programación de aplicaciones (API) es un grupo de normas que establecen cómo diferentes dispositivos pueden comunicarse entre sí. La transferencia de estado representacional (REST) no es considerado como un estándar, si no como un estilo de arquitectura que posee principios de diseño [13].

Una API REST sería considerada como un mecanismo de comunicación basado en un estilo arquitectónico mediante el uso del protocolo HTTP tomando como métodos principales: GET (obtener), POST (crear), PUT (actualizar) y DELETE (eliminar) para realizar operaciones sobre endpoints específicos.

Los 6 principios del estilo de diseño de REST [14] son:

- 1) Cada recurso debe poseer identificadores únicos (URIs) y almacenar la información necesaria para cada cliente.
- 2) El servidor no influye en la lógica del cliente, considerando el envío de información específica en función de la URI del recurso solicitado.
- 3) Cada solicitud emitida por el cliente no debe permitir el almacenamiento de información de sesión.
- 4) Se debe permitir el almacenamiento en cache por parte del cliente y el servidor, siempre y cuando sea posible.
- 5) Dada una solicitud emitida por el cliente, para llegar al servidor debe pasar por varias capas como de seguridad, balanceadores de carga, entre otras, dejando de lado el pensamiento de comunicación directa entre cliente y servidor.
- 6) El servidor puede enviar código ejecutable como respuesta al cliente siempre y cuando se realice de forma segura y bajo petición.

¿Backend y E-learning?

Una plataforma web de e-learning es un mecanismo de aprendizaje muy novedoso y efectivo que la mayoría de las personas optan en lugar de la educación tradicional, por esto es esencial resolver aspectos enfocados a la gestión de usuarios y cursos, mediante un buen desarrollo del lado del servidor.

La gestión de usuarios dentro de este tipo de plataformas juega un papel esencial, debido a los diferentes roles que se pueden manejar, como estudiantes, instructores y administradores. El desarrollo del lado del servidor tiene como objetivo buscar técnicas que permitan una asignación correcta de la información en función del rol de cada usuario, de no controlar esto, un estudiante podría crear cursos sin tener los permisos necesarios o incluso un administrador podría manipular contraseñas si estas no poseen algún método de encriptación.

Dentro de la gestión de cursos, se tiene un mayor volumen de información debido a los datos que se pretenden guardar y leer. Un backend sólido permite distribuir la información de manera eficaz logrando eliminar el pensamiento común de “se cayó la página”. Además, el desarrollo del lado del servidor junto con la interfaz de usuario busca ofrecer una información más limpia y material relevante para ser utilizado considerando lo que desea cada usuario.

La lógica dentro del desarrollo backend no es esencial para conocimiento del usuario, mientras se cumpla con sus necesidades y se logre una experiencia agradable de la mano con la interfaz de usuario.

Código Limpio (Clean Code)

El desarrollo de software a menudo se enfrenta al aumento de costos y complejidad cuando se requiere una nueva funcionalidad en un proyecto ya existente. Pensar en reescritura completa del código no es práctico en función del tamaño de un proyecto, lo cual da como consecuencia, una falta de flexibilidad en el programa.

Clean Code es considerado una buena práctica dentro de la programación, basado en la comprensión de lectura del código desarrollado, de tal manera que sea fácil entender, modificar y mantener el software creado, por otros programadores [15].

Los principios que se manejan dentro de Clean Code [16] son:

- 1) Manténlo simple (KISS):** permite obtener un código de mayor calidad, resolver problemas más rápidamente, trabajar mejor en equipo y tener una base de código más flexible.
- 2) Comentarios:** facilitan la actualización, depuración y análisis del código, y son especialmente útiles en el trabajo en equipo.
- 3) No te repitas (DRY):** filosofía de diseño minimalista que se base en establecer que cada pieza de conocimiento (código) debe ser única.

4) **No lo vas a necesitar (YAGNI):** es parte de la metodología de Extreme Programming (XP) que explica que un desarrollador no debe agregar una funcionalidad a menos que sea necesaria.

5) **Sangría al código:** al aplicar una estructura de organización coherente con sangrías, se mejora la comprensión del código y se facilita la identificación de bloques de código específicos.

Base de Datos Relacional vs. Base de Datos No Relacional

Una base de datos es considerada como una colección ordenada en la cual se permite insertar, leer, actualizar y eliminar registros en función de lo que desea un usuario.

Las bases de datos relacionales organizan la información en tablas con filas y columnas que mantienen cierta relación entre sí. Este tipo de bases de datos establecen relaciones mediante el uso de una claves primarias y claves foráneas. Una clave primaria es un campo único que identifica cada fila de forma respectiva, por otro lado, una clave foránea es un campo que identifica la relación que se tiene con una clave primaria, para conocer la relación que se tiene entre tablas [17] [18].

Las bases de datos no relacionales o también conocidas como NoSQL se diferencian del modelo tradicional por la forma en como almacena y organiza la información. Este tipo de bases de datos utilizan diferentes modelos de datos como colecciones de documentos o gráficos orientados al uso de nodos [19].

Para la selección de un modelo de bases de datos se debe considerar:

- **Tipo de datos:** si se manejan datos estructurados es mejor optar por el uso de una base de datos relacional, caso contrario, si se manejan datos no estructurados la mejor opción es usar un modelo no relacional.
- **Volumen de datos:** para grandes volúmenes de datos o que tienden a tener un crecimiento vertical, es recomendable emplear un modelo no relacional, mientras que para un conjunto de datos pequeño es mejor la utilización de un modelo relacional.
- **Gestión de datos y propósito:** el uso de bases de datos relacionales tiene menor complejidad al uso de bases no relacionales, pero si se desea administrar datos en tiempo real, una base de datos no relacional es la opción óptima.

Python

Python es un lenguaje de programación conocido por ser de alto nivel, dado que posee una sintaxis clara y legible pareciéndose al lenguaje natural, además es portátil, lo que permite su ejecución en varios sistemas operativos, como Windows, MacOS y Linux [20]. Este lenguaje permite a los desarrolladores ser más eficientes y tener menos errores, por su amplia documentación y gran comunidad, dando como consecuencia la resolución de problemas dentro de la programación de una forma dinámica y eficaz.

Lo que diferencia a Python de la mayoría de los lenguajes de programación es el enfoque amplio que posee. Se utiliza en el desarrollo web, ciencia de datos, desarrollo de inteligencia artificial, entre otros [20].

Python deja de lado la especificación del tipo de variables, lo cual para la mayoría de los programadores es de gran ayuda para evitar errores de declaración y asignación.

Udemy es una plataforma en línea de aprendizaje con miles de cursos impartidos por instructores destacados. El lenguaje de programación que es usado del lado del servidor es Python, debido a su popularidad [21].

Django

Django es uno de los marcos de trabajo (frameworks) más populares utilizados junto a Python por su enfoque de desarrollo rápido y limpio, este framework es útil en la construcción de aplicaciones con una curva de crecimiento constante.

En comparación a otros frameworks, Django ofrece los siguientes beneficios [22]:

- Posee un modelo de programación para mapear estructuras de bases de datos relacionales (ORM), permitiendo interactuar de forma directa con una base de datos como si fueran instrucciones SQL.
- Permite el desarrollo de aplicaciones web con un alto nivel de seguridad y fiabilidad, debido a su sistema de autenticación de usuarios.
- Amplia documentación que facilita el aprendizaje y la resolución de problemas.

MongoDB

MongoDB es un sistema de base de datos no relacional, basado en el almacenamiento en forma de documentos, donde cada dato es parecido a un elemento tipo JSON. Entre las características que lo convierten en un motor NoSQL superior en comparación a otros son [23]:

- Escala en función de la demanda que posea, permitiendo escalamiento vertical y horizontal.
- Permite el almacenamiento de documentos con diferentes tipos de características.
- Velocidad de lectura por su estructura enfocada en el uso de documentos.
- Optimización de código en la escritura de consultas.

Scrum

Su definición se da por primera vez en 1986 por Hirotaka Takeuchi e Ikujiro Nonaka, pero se presenta por primera vez al mundo en 1995 de la mano de Jeff Sutherland y Ken Schwaber. Scrum es un marco de desarrollo ágil de software que se caracteriza por no seguir los enfoques tradicionales con respecto al desarrollo de proyectos. En este marco de desarrollo se fomenta la autoorganización de equipos y la adaptación al cambio en función de los nuevos requerimientos establecidos por el cliente [24] [25].

Scrum basa sus principios en el manifiesto ágil, el cual es un documento detallado sobre los valores (Figura 1.2) y fundamentos (Figura 1.3) que se deberían seguir para llevar un correcto desarrollo de software.



Figura 1.2 Valores del Manifiesto Ágil [26]



Figura 1.3 Fundamentos del Manifiesto Ágil [27]

Roles de Scrum

- **Product Owner:** decide la funcionalidad del producto y la priorización de cada característica que se desea implementar. Representa los intereses que tienen los clientes y los usuarios finales [28].
- **Scrum Master:** es el responsable de verificar que se cumplan con los principios de Scrum, proteger al equipo y eliminar obstáculos [28].
- **Scrum Team:** este equipo está conformado por personas que realizan pruebas al código, diseñadores, expertos en experiencia de usuario y desarrolladores, que tienen como objetivo evitar cuellos de botella en el desarrollo del proyecto [28].

Artefactos de Scrum

- **Product Backlog:** es una lista dinámica de tareas, mejoras y requisitos esenciales para el éxito del proyecto. Además, es mantenido por el Product Owner y se adapta continuamente a cambios en prioridades y necesidades del mercado [28].
- **Sprint Backlog:** lista de tareas específicas seleccionadas del Product Backlog para completar durante el Sprint [28].

- **Incremento:** se refiere a un resultado tangible presentado al finalizar el Sprint, es decir, es un producto funcional [28].

Eventos de Scrum

- **Planificación del Sprint:** en este evento se definen los objetivos específicos a cumplir durante el Sprint y se realiza el Sprint Backlog [28].
- **Sprint:** es considerado como un periodo de tiempo donde el equipo Scrum trabaja para cumplir con un incremento. Cada sprint puede durar 2 semanas, pero esto varía en función de las tareas y el número de miembros del equipo [28].
- **Reunión diaria de Scrum:** es una sesión de 15 minutos como máximo, donde se reporta los avances del día y lo que se planea hacer hasta la próxima reunión, además, en caso de tener dificultades o problemas con alguna tarea también se reporta y posteriormente se toman medidas de solución [28].
- **Revisión del Sprint:** es considerada como una reunión informal para revisar si se cumplió con los objetivos del sprint y presentar el trabajo final al Product Owner [28].
- **Retrospectiva del Sprint:** este evento se lleva a cabo para tener una retroalimentación sobre lo que estuvo bien y mal dentro del Sprint [28].

Los eventos de Scrum siguen el ciclo establecido en la Figura 1.4, considerando una fase de **Inicio** [29], donde se analiza el objetivo y los requerimientos del proyecto, se forma el equipo Scrum, se crean las historias de usuario y el Product Backlog y se define la planificación de lanzamiento.

Traditional Scrum

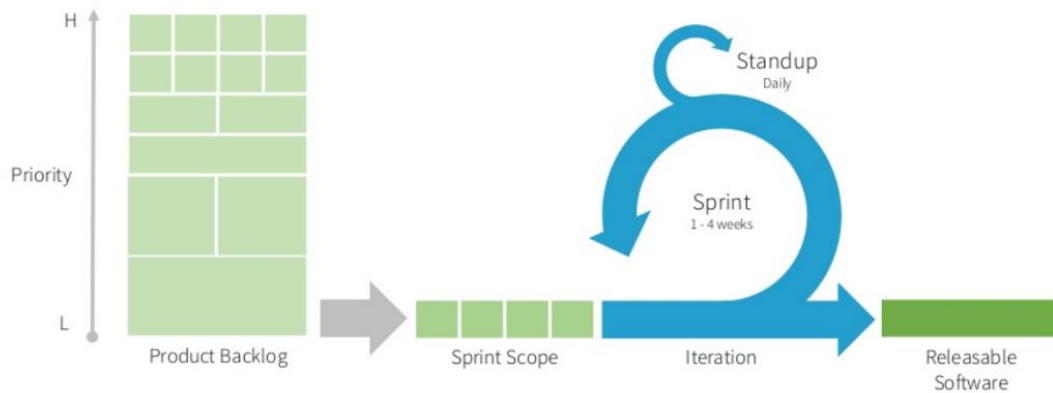


Figura 1.4 Sprint en Scrum [30]

Kanban

Kanban es una metodología ágil que tiene como objetivo ofrecer una ayuda a los equipos que desarrollan proyectos o algún producto, se enfoca en la mejora continua, buscando un punto de equilibrio entre las tareas de cada miembro del equipo y su tiempo para realizarlas. Esta metodología fue dada a conocer en 1940 de la mano de Taiichi Ohno todo con el fin de mejorar el sistema de producción de Toyota [31].

La herramienta principal que usa Kanban para manejar el flujo del trabajo es conocido como Tablero Kanban, que está conformado por 3 columnas inicialmente:

- **Por hacer (to do):** se colocan las tareas extraídas del Product Backlog.
- **Haciendo (doing):** se colocan las tareas que están siendo desarrolladas por cada miembro del equipo.
- **Terminado (done):** se colocan las tareas que ya fueron terminadas posteriormente de realizar las pruebas de funcionamiento correspondientes.

El tablero Kanban final donde se considera también la fase de pruebas (test) sería el que se puede observar en la Figura 1.5.

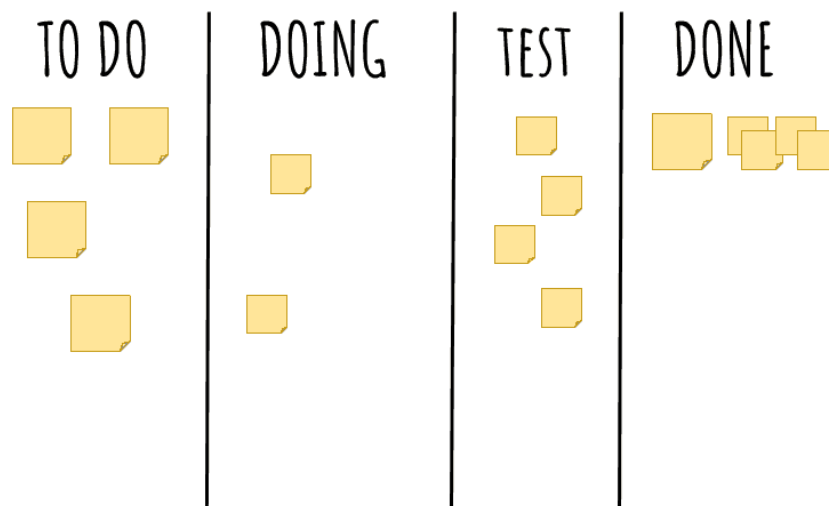


Figura 1.5 Tablero Kanban [32]

Estimación de esfuerzo

Dentro de Scrum la estimación de esfuerzo es una técnica utilizada para conocer la complejidad que puede llegar a tener una tarea o historia de usuario, permitiendo llevar una planificación más estructurada dentro del desarrollo del proyecto [33].

- **Tamaño relativo (Story Points)**

Este tipo de estimación de esfuerzo se basa en dar un puntaje en función del riesgo, cantidad de esfuerzo necesario para completar la tarea y el nivel de complejidad. Es una medida abstracta y relativa que no representa tiempo, como horas o días [33].

Para la asignación de puntos se considera la secuencia de Fibonacci (0, 1, 2, 3, 5, ...), por ejemplo, si una tarea es asignada con una puntuación de 5 y otra con una puntuación de 8, se puede inferir que la tarea con una puntuación de 8 posee una mayor complejidad para ser desarrollada.

Cada miembro del equipo Scrum establece un puntaje a cada tarea, si los puntajes concuerdan se establece el valor respectivo, caso contrario se discute sobre la valoración de cada tarea considerando los argumentos de cada miembro y se llega a un acuerdo entre todos.

2 METODOLOGÍA

En función de Scrum para la planificación y desarrollo del proyecto [29] y Kanban para la gestión del flujo de trabajo, en este capítulo se detalla las fases elaboradas para el

desarrollo de la programación backend de la plataforma web “Virtual-POLI” para complementar la educación académica generada por estudiantes para estudiantes.

2.1 Inicio

La fase de Inicio se implementó en los siguientes pasos:

- 1) Corresponde a la definición del objetivo del proyecto.
- 2) Se enfoca en detallar los requerimientos del sistema a ser desarrollado.
- 3) Se especifica el stack de programación y el diagrama de clases.
- 4) Se establece el equipo Scrum.
- 5) Se crean las historias de usuario.
- 6) En función de las historias de usuario, se crea el Product Backlog.
- 7) Se realiza la planificación de lanzamiento, que corresponde a la fecha de inicio y finalización del proyecto.

Definición del objetivo del proyecto

Desarrollar el backend para una plataforma web orientada a la educación virtual caso de aplicación “Virtual-POLI”.

Requerimientos del sistema

- Proporcionar puntos de acceso que permitan la gestión de cursos para los estudiantes de la Facultad de Ingeniería en Sistemas.
- Permitir almacenamiento en la base de datos de la inscripción de los estudiantes a los cursos que tengan interés.
- Crear puntos de acceso que permitan al frontend enviar la información necesaria con respecto a la gestión de categorías para tener un almacenamiento en la base de datos.
- Permitir el almacenamiento de comentarios por parte del estudiante con respecto a un curso en la base de datos.
- Definir puntos de acceso que permitan obtener los profesores y cursos destacados basados en una calificación promedio.
- Implementar un sistema de roles y permisos para garantizar la seguridad y el control del sistema.

- Implementar una infraestructura de software que permita el manejo de usuarios a medida que la demanda aumente.

Stack de programación y diagrama de componentes

Las tecnologías de programación consideradas para el desarrollo del proyecto se visualizan en la Tabla 2.1.

Tabla 2.1 Stack de programación

Tecnologías	Versión
Python	3.10.6
Django	4.1.13
MongoDB	7.0.3

El diagrama de componentes se puede observar en la Figura 2.1, el cual se basa en un patrón arquitectónico Modelo-Vista-Controlador (MVC) para visualizar la lógica de comunicación entre la vista (Frontend) con los diferentes modelos y controladores definidos en el backend.

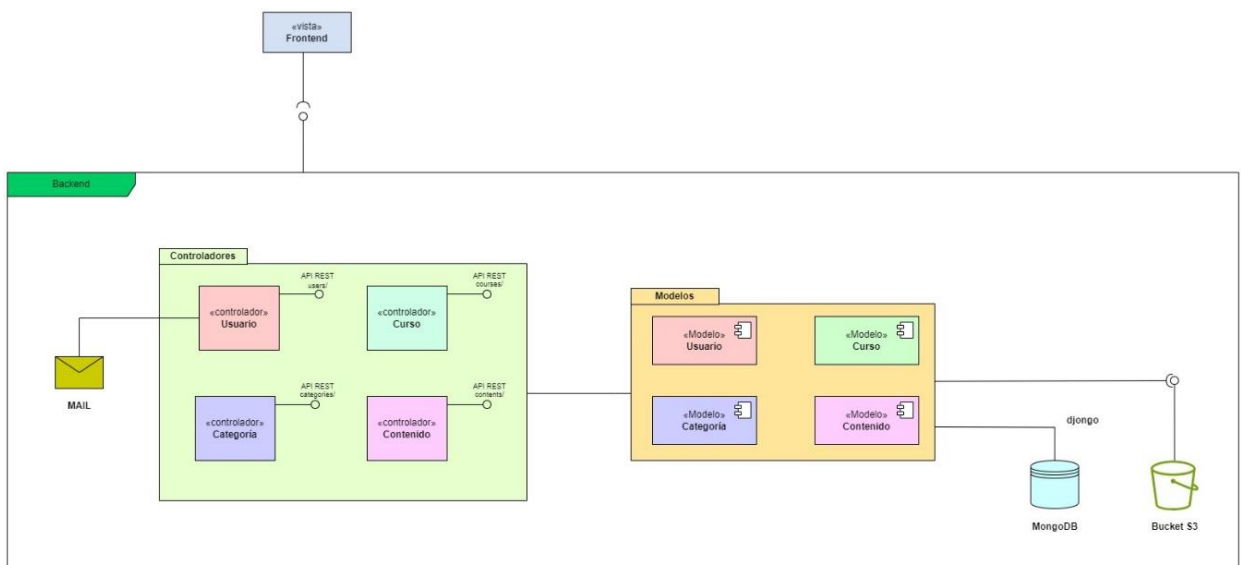


Figura 2.1 Diagrama de componentes

Equipo Scrum

En función de los roles que se menciona en Scrum, se tiene la Tabla 2.2.

Tabla 2.2 Equipo Scrum

Roles	Responsable
Product Owner	Presidente de la Asociación de Estudiantes de la Facultad de Ingeniería en Sistemas
Scrum Master	Msc. Hernán Ordoñez
Equipo de Desarrollo	Alexander Guillin - Programador backend

Historias de usuario

Las historias de usuario consideradas para el desarrollo del proyecto se basan en el modelo de la Tabla 2.3. El resto de las historias de usuario se adjunta en el Anexo I.

Tabla 2.3 Historia de usuario US-001

Número: US-001	Usuario: Estudiante
Nombre de la Historia: Registro de usuarios Como estudiante. Quiero poder registrarme en la página web. Para acceder a todos los cursos que se ofrecen.	
Esfuerzo: 8	
Conversación: El estudiante puede registrarse en la página web, para ello se le solicitan datos como: correo, nombre, apellido, contraseña y semestre. El estudiante debe verificar su correo electrónico mediante un enlace que le llegara a este y posteriormente puede ingresar a la página web.	
Observaciones: <ul style="list-style-type: none">- La contraseña es validada desde el frontend.- Se verificará si es un correo electrónico válido.- Se verificará la unicidad del usuario.- Se aplicará un hash + un dato aleatorio (salt) a la contraseña para protección de datos.- Si el estudiante no verifica el correo electrónico, este no puede ingresar a la página web.	

<ul style="list-style-type: none"> - Para el manejo de la sesión, cuando el usuario inicie sesión se le asignará un token de sesión.
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Se registran los datos de usuario con la respectiva validación de datos. - El usuario recibe un correo electrónico de verificación. - Existe unicidad de un usuario registrado. - Aplicación de un hash + salt a la contraseña previo al almacenamiento en la base de datos. - Verificación de email y contraseña en el inicio de sesión. - Generación de un token de sesión.

Product Backlog

En función de las historias de usuario desarrolladas, el Product Backlog a ser considerado dentro del proyecto se visualiza en la Tabla 2.4.

Tabla 2.4 Product Backlog

US	Elementos	Estimación de esfuerzo
Alto		
001	Registro de usuarios	8
003	Permitir ser un instructor	8
Medio		
002	Mostrar información del usuario y permitir actualización	5
004	Cambiar y restaurar contraseña	5
007	Creación de cursos	5
009	Permitir contacto con equipo de 'Virtual-POLI'	5
010	Permitir crear, obtener, actualizar y eliminar categorías de cursos	5
Bajo		
005	Permitir cierre de sesión	1
006	Conocer instructores, cursos destacados y agregados recientemente	1
008	Permitir ingresar comentarios y una calificación en los cursos	1
011	Obtener los cursos por instructor o categoría	1

Planificación de lanzamiento

En función del objetivo y los requerimientos del proyecto, la planificación del lanzamiento se observa en la Tabla 2.5.

Tabla 2.5 Planificación de lanzamiento

Planificación del lanzamiento	
Fecha de inicio del proyecto	20 de Octubre del 2023
Fecha de entrega del proyecto	22 de Diciembre del 2023

Considerando las historias de usuario definidas en el Product Backlog y su estimación de esfuerzo, se llevará a cabo 3 Sprints, como se observa en la Tabla 2.6.

Tabla 2.6 Planificación de Sprints

Sprints	Fecha de inicio	Fecha de finalización	Número de semanas
1	20 de Octubre del 2023	10 de Noviembre del 2023	3
2	11 de Noviembre del 2023	01 de Diciembre del 2023	3
3	02 de Diciembre del 2023	22 de Diciembre del 2023	3

Las siguientes fases, se basan en la definición de Sprint, es decir, cada una de estas se van a repetir en función del número de Sprints que se consideren para cumplir con el Product Backlog.

2.2 Planificación del Sprint

La fase de planificación del Sprint se realizó en 2 pasos:

- 1) Detallar historias de usuario a realizarse en el sprint.
- 2) Identificar tareas y crear Sprint Backlog.

Sprint 1

1) Historias de usuario del Sprint

Las historias de usuario consideradas para el Sprint 1, son las siguientes:

- Registro de usuarios.
- Mostrar información del usuario y permitir actualización.
- Permitir cierre de sesión.

2) Identificación de tareas y Sprint Backlog

En función de cada historia de usuario las tareas correspondientes a cada una de estas se aprecian en la Tabla 2.7 correspondiente al Sprint Backlog.

Tabla 2.7 Sprint 1 - Sprint Backlog

Tarea	Elementos	Estimación de esfuerzo
Alto		
1	Definir modelo de base datos	8
2	Crear proyecto con Python y Django	5
3	Conectar la base de datos	5
4	Definir función para la creación y verificación de un token de sesión	8
5	Definir función para enviar correos electrónicos	8
Medio		
6	Definir modelo User dentro del proyecto	5
7	Definir endpoint para el registro de usuarios	5
8	Definir función para el inicio de sesión	5
9	Definir endpoint para actualizar información del usuario	5
Bajo		
10	Definir función para verificar un correo válido	2
11	Pruebas unitarias	2

Sprint 2

1) Historias de usuario del Sprint

Las historias de usuario consideradas para el Sprint 2, son las siguientes:

- Permitir ser un instructor.
- Cambiar y restaurar contraseña.
- Creación de cursos.
- Permitir contacto con equipo de 'Virtual-POLI'.
- Permitir crear, obtener, actualizar y eliminar categorías de cursos.

2) Identificación de tareas y Sprint Backlog

En función de cada historia de usuario las tareas correspondientes a cada una de estas se aprecian en la Tabla 2.8 correspondiente al Sprint Backlog.

Tabla 2.8 Sprint 2 - Sprint Backlog

Tarea	Elementos	Estimación de esfuerzo
Alto		
1	Definir endpoint para enviar un correo de verificación para ser instructor	8
2	Definir endpoint para cambiar el rol del estudiante a instructor	5
3	Definir endpoint para enviar un correo de restauración de contraseña	5
Medio		
4	Definir endpoint para restaurar contraseña	5
5	Definir endpoint para cambiar contraseña	5
6	Definir modelo Courses en el proyecto	3
7	Definir endpoints para crear y obtener cursos	5
8	Definir modelo Content en el proyecto	3
9	Definir endpoint para almacenar el contenido del módulo de un curso	5
10	Definir endpoint para enviar un correo de contacto con equipo de 'Virtual-POLI'	5
Bajo		
11	Definir modelo Categories	3
12	Definir endpoints para crear, obtener, actualizar y eliminar una categoría	5
13	Pruebas unitarias	2

Sprint 3

1) Historias de usuario del Sprint

Las historias de usuario consideradas para el Sprint 3, son las siguientes:

- Conocer instructores, cursos destacados y agregados recientemente.
- Permitir ingresar comentarios y una calificación en los cursos.
- Obtener los cursos por instructor o categoría.

2) Identificación de tareas y Sprint Backlog

En función de cada historia de usuario las tareas correspondientes a cada una de estas se aprecian en la Tabla 2.9 correspondiente al Sprint Backlog.

Tabla 2.9 Sprint 3 - Sprint Backlog

Tarea	Elementos	Estimación de esfuerzo
Alto		
1	Definir endpoint que permita actualizar comentarios y calificación con respecto a un curso	2
Medio		
2	Definir endpoint para obtener instructores destacados	2
3	Definir endpoint para obtener los cursos destacados	2
4	Definir endpoint para obtener los cursos por categoría	1
5	Definir endpoint para obtener los cursos por instructor	1
6	Definir endpoint para obtener los cursos agregados recientemente	2
Bajo		
7	Definir endpoint para obtener el id de una categoría	1
8	Pruebas unitarias	2

2.3 Desarrollo del Sprint

La fase de desarrollo del Sprint se realizará en función de las tareas definidas en el Sprint Backlog.

Sprint 1

Para manejar el flujo de trabajo con respecto al Sprint 1 se utilizó un Tablero Kanban, como se puede apreciar en la Figura 2.2.

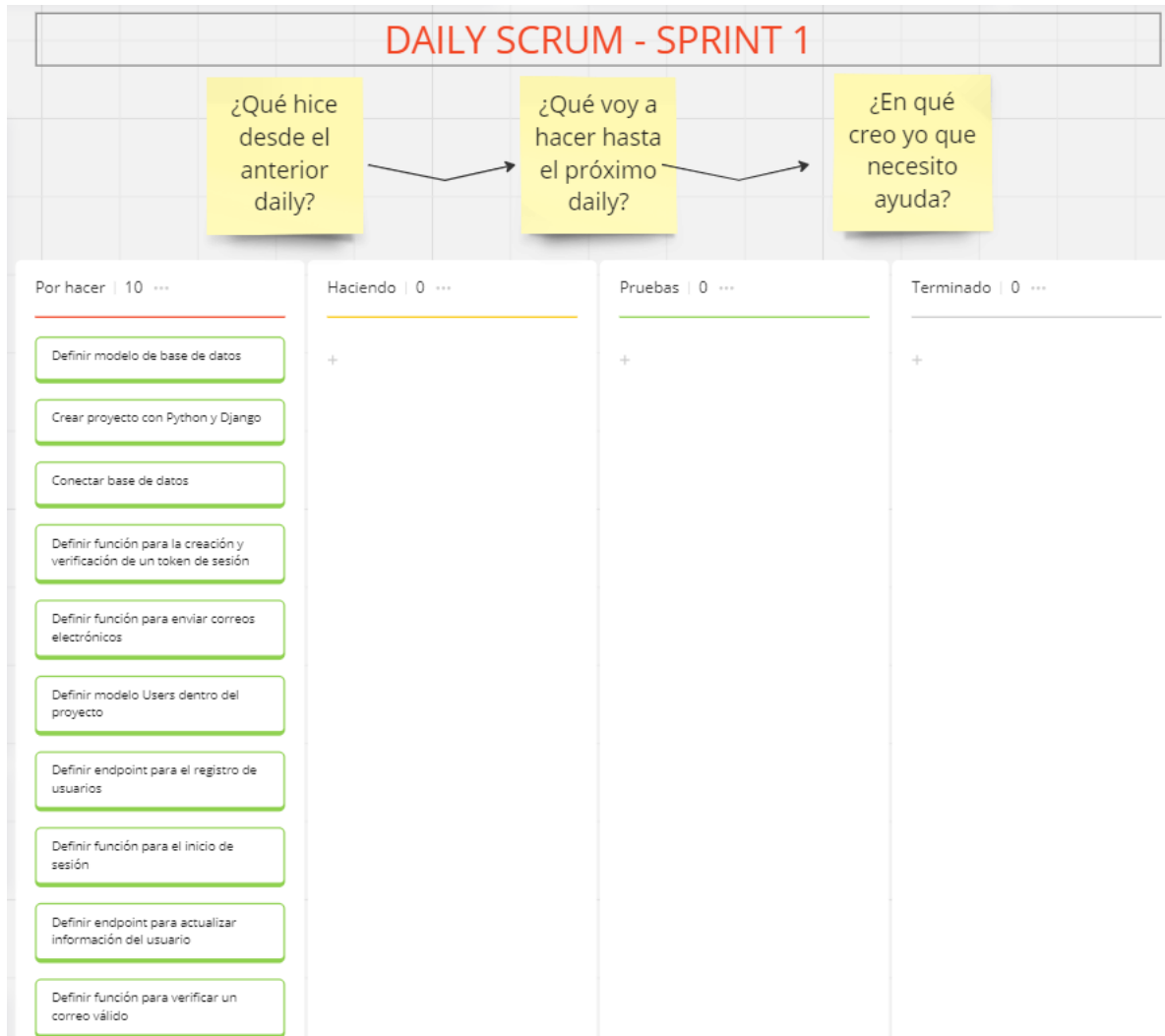


Figura 2.2 Tablero Kanban - Sprint 1

El Tablero Kanban correspondiente al Sprint 2 y 3, se observa en el Anexo II.

- **Modelo de base de datos**

Considerando una base de datos NoSQL por la escalabilidad y funcionalidad que esta ofrece, el modelo de base de datos establecido para el proyecto bajo el concepto de colecciones y documentos se observa en el Anexo III.

- **Configuraciones establecidas para el proyecto**

Dentro del proyecto se llevaron configuraciones enfocadas al funcionamiento del sistema, permitiendo la conexión entre Django y MongoDB, la configuración para el envío de correos electrónicos y para el almacenamiento en un Bucket S3.

El proyecto hizo uso de variables de entorno por temas de seguridad, dado que se tiene claves secretas, URLs a servicios externos, el nombre de la base de datos, entre otros. Las variables de entorno consideradas para el proyecto se visualizan en la Figura 2.3.

```
.env
1 DJANGO_SECRET_KEY=''
2 DEBUG=
3 DATABASE_NAME=''
4
5 SECRET_KEY_TOKEN=''
6
7 URL_FRONTEND=''
8
9 URL_IMAGE_COURSE_STORAGE = ''
10 URL_VIDEO_COURSE_STORAGE = ''
11 URL_VIDEO_COURSE_CONTENT_STORAGE = ''
12 URL_IMAGE_INSTRUCTOR_STORAGE = ''
13
14 EMAIL_HOST=''
15 EMAIL_PORT=
16 EMAIL_USE_TLS=
17 EMAIL_HOST_USER = ''
18 EMAIL_HOST_PASSWORD = ''
19
20 AWS_ACCESS_KEY_ID=''
21 AWS_SECRET_ACCESS_KEY=''
22 AWS_STORAGE_BUCKET_NAME = ''
```

Figura 2.3 Variables de entorno usadas en el proyecto

1) Conexión entre Django y MongoDB

Para establecer una conexión entre Django y MongoDB se utilizó **django** que es un controlador que deja de lado el uso de consultas largas y se centra en lo que se desea obtener como resultado. La versión del controlador y las configuraciones con respecto a la variable **DATABASES** dentro de DJANGO se puede observar en la Tabla 2.10.

Tabla 2.10 Configuración para la conexión entre Django y MongoDB

		Versión
Conector Django – MongoDB	Djongo	1.3.6

Conexión con base de datos	<pre> DATABASES = { 'default': { 'ENGINE': 'django', 'CLIENT':{ 'host': 'localhost', 'port': 27017, }, 'NAME': config('DATABASE_NAME'), } } </pre>
-----------------------------------	--

2) Configuración para el envío de correos electrónicos

Para configurar la funcionalidad de correo electrónico de la aplicación, primero se creó una cuenta de correo electrónico en GMAIL denominada polivirtual2023b@gmail.com.

Además, se utilizó la configuración de la Tabla 2.11. Donde:

- **EMAIL_BACKEND:** especifica el backend de correo electrónico que DJANGO usará para enviar correos electrónicos, para esta configuración se usó el backend SMTP.
- **EMAIL_HOST:** especifica el host del servidor de correo saliente (SMTP).
- **EMAIL_PORT:** especifica el puerto del servidor de correo.
- **EMAIL_USE_TLS:** especifica si se debe utilizar seguridad en la capa de transporte (TLS) para permitir un cifrado en la conexión con el servidor de correo.
- **EMAIL_HOST_USER:** especifica el nombre del usuario utilizado para autenticarse en el servidor de correo.
- **EMAIL_HOST_PASSWORD:** se especifica la contraseña del usuario host.

Tabla 2.11 Configuración para el envío de correos electrónicos

	Versión
Configuración para enviar correos electrónicos	<pre># Email settings EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend' EMAIL_HOST = config('EMAIL_HOST') EMAIL_PORT = config('EMAIL_PORT') EMAIL_USE_TLS = config('EMAIL_USE_TLS') EMAIL_HOST_USER = config('EMAIL_HOST_USER') EMAIL_HOST_PASSWORD = config('EMAIL_HOST_PASSWORD')</pre>

3) Configuración de almacenamiento en Bucket S3

Para configurar el almacenamiento en el Bucket S3 implementado por el componente de infraestructura y seguridad se considera la configuración expuesta en la Tabla 2.12. Donde:

- **AWS_ACCESS_KEY_ID:** especifica el id de clave de acceso de AWS que se utilizará para autenticar la conexión con los servicios de AWS.
- **AWS_SECRET_ACCESS_KEY:** especifica la clave de acceso secreta de AWS.
- **AWS_STORAGE_BUCKET_NAME:** especifica el nombre del bucket de almacenamiento de AWS S3.
- **AWS_QUERYSTRING_AUTH:** especifica si se establece una configuración del uso de cadenas de consulta para acceder a los archivos almacenados en AWS S3.
- **DEFAULT_FILE_STORAGE:** especifica el backend de almacenamiento predeterminado que se utilizará para almacenar los archivos multimedia del proyecto.

Tabla 2.12 Configuración de almacenamiento en Bucket S3

	Versión
Configuración para el almacenamiento en Bucket S3	<pre># AWS Configuration AWS_ACCESS_KEY_ID = config('AWS_ACCESS_KEY_ID') AWS_SECRET_ACCESS_KEY = config('AWS_SECRET_ACCESS_KEY') AWS_STORAGE_BUCKET_NAME = config('AWS_STORAGE_BUCKET_NAME') AWS_QUERYSTRING_AUTH = False # Configuración de almacenamiento para archivos estáticos y multimedia DEFAULT_FILE_STORAGE = 'storage.backends.s3boto3.S3Boto3Storage'</pre>

- **Librerías adicionales utilizadas en el proyecto**

Los paquetes o librerías adicionales que se utilizaron para el proyecto se pueden apreciar en la Tabla 2.13.

Tabla 2.13 Paquetes adicionales utilizados en el proyecto

Paquete	Versión	Funcionalidad
asgiref	3.7.2	Framework de Python que permite la comunicación asíncrona entre servidores y aplicaciones web.
django-cors-headers	4.3.1	Permite comunicación entre diferentes dominios.
django-rest-framework	3.14.0	Framework que facilita la creación de APIs web.
dnspython	2.4.2	Posee herramientas que permitan trabajar con Sistemas de Nombre de Dominio (DNS).
pymongo	3.12.0	Permite realizar operaciones CRUD con MongoDB.
setuptools	63.2.0	Posee herramientas para empaquetar, distribuir e instalar paquetes de Python.
python-decouple	^3.8	Permite cargar configuraciones de manera externa relacionadas a las variables de entorno.
pyjwt	^2.8.0	Herramienta para trabajar con tokens, Tokens de Internet tipo JSON (JWT).

Las tareas dentro del Sprint 1, 2 y 3, que tienen como fin la implementación de código, se pueden visualizar en el Anexo IV.

Reuniones Diarias de Scrum

Para las reuniones diarias de Scrum se consideró un tiempo de 15 minutos, donde se respondió 3 preguntas fundamentales:

- 4) ¿Qué hice ayer?
- 5) ¿Qué voy hacer hoy?
- 6) ¿Qué impedimentos tengo para realizar mi trabajo?

Dentro de la reunión se utilizó los Tableros Kanban que se pueden visualizar en el Anexo VI.

Pruebas unitarias

Para las pruebas unitarias se utilizó Thunder Client y MongoDB para verificar la funcionalidad de cada endpoint.

- **Registro de usuarios**

Para el registro de usuarios, se utiliza la siguiente configuración:

- **Método:** POST
- **Endpoint:** *user/sign-up/*
- **Entrada:** correo electrónico, nombres, apellidos, contraseña y semestre del estudiante a registrarse.
- **Salida:**
 - o **Estado:** 200
 - Usuario registrado (Figura 2.4).
 - Envío de correo de verificación de cuenta (Figura 2.5).
 - Almacenamiento en base de datos del usuario registrado (Figura 2.6).
 - o **Estado:** 400 -> Correo electrónico inválido (Revisar Anexo V. Figura 5.5).
 - o **Estado:** 409 -> Usuario ya registrado (Revisar Anexo V. Figura 5.4).

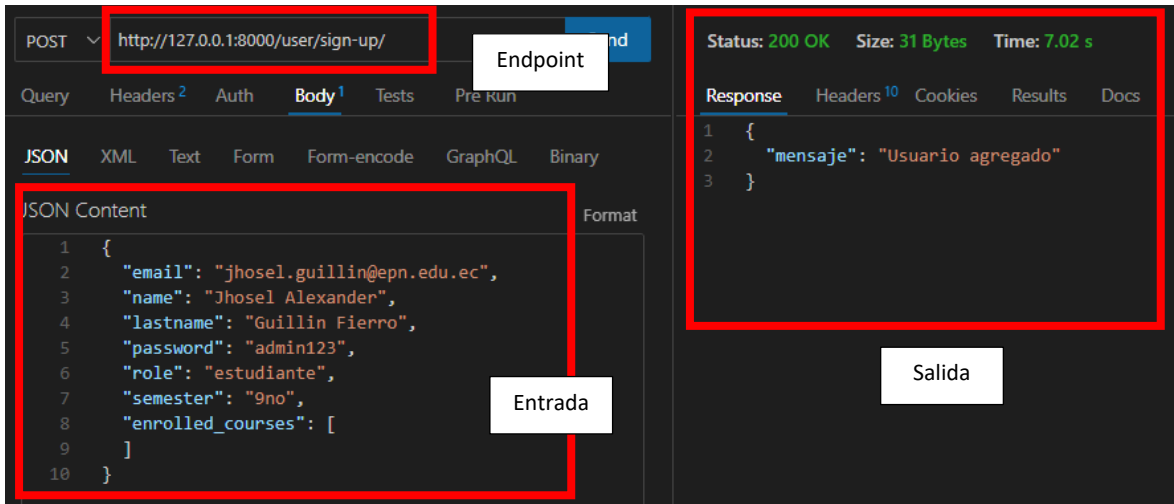


Figura 2.4 Endpoint para el registro de usuarios

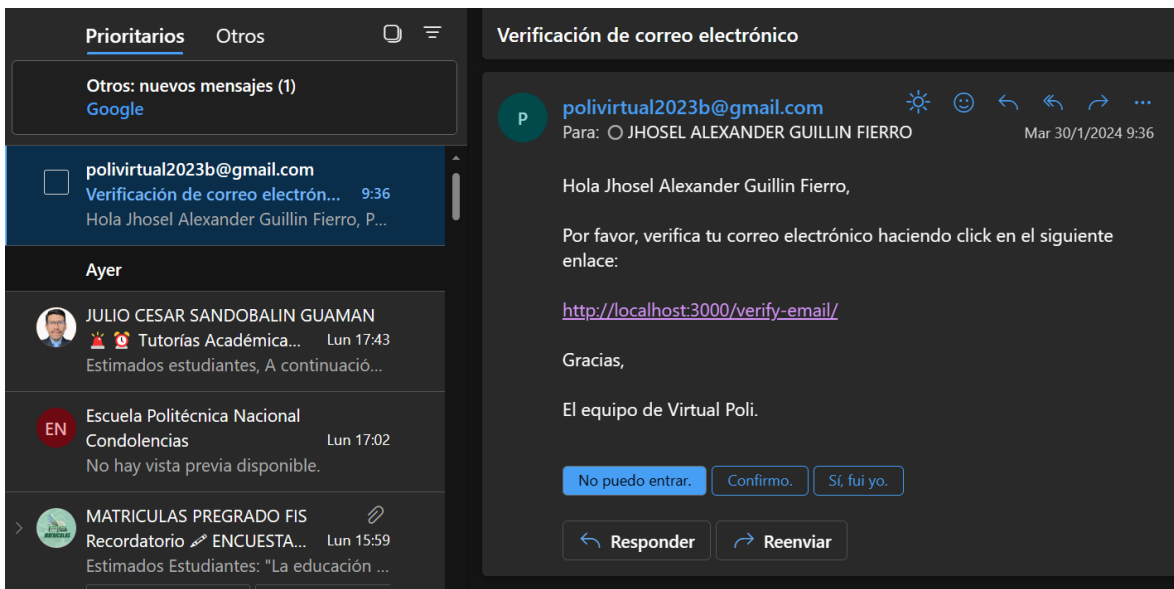


Figura 2.5 Correo de verificación de un usuario registrado

```
_id: ObjectId('65b9096429d1f0a5da3a350c')
email: "jhosel.guillin@epn.edu.ec"
name: "Jhosel Alexander"
lastname: "Guillin Fierro"
password: "pbkdf2_sha256$390000$kbJo0xEBmwWPiBaIUIieq$6tZiFv0Vhz3h5MPNG+YcSphfVr..."
role: "estudiante"
semester: "9no"
approve_teacher: ""
approve_teacher_email: ""
user_description: ""
score_teacher: 0
profile_image_url: ""
enrolled_courses: "[]"
email_verification: false
session_token: ""
```

Figura 2.6 Almacenamiento del usuario registrado en la base de datos

- **Inicio de sesión**

Para el inicio de sesión, se utiliza la siguiente configuración:

- **Método:** POST
- **Endpoint:** *user/sign-in/*
- **Entrada:** correo electrónico y contraseña del usuario a iniciar sesión.
- **Salida:**
 - o **Estado:** 200 -> Inicio de sesión exitoso y retorna token de sesión (Figura 2.8).
 - o **Estado:** 400 -> Correo electrónico inválido o correo electrónico y contraseña no ingresados.
 - o **Estado:** 401 -> Contraseña incorrecta (Revisar Anexo V. Figura 5.7).
 - o **Estado:** 403 -> Correo electrónico no verificado (Figura 2.7).
 - o **Estado:** 404 -> Usuario no encontrado (Revisar Anexo V. Figura 5.6).

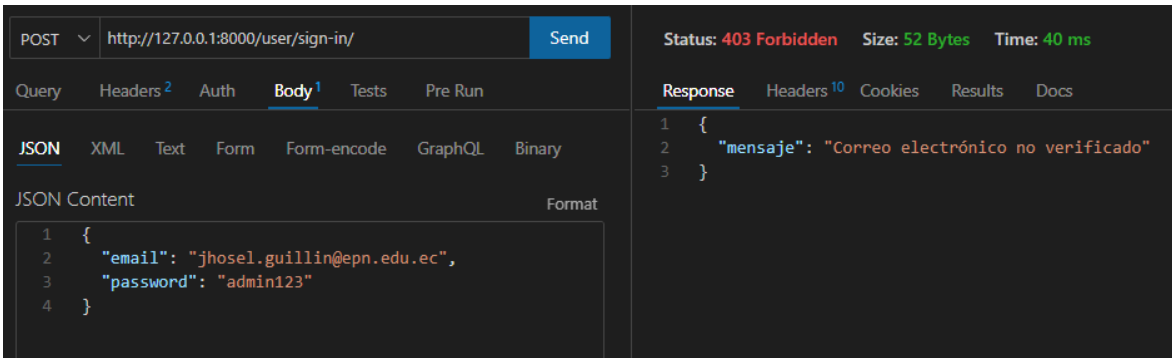


Figura 2.7 Fallo en inicio de sesión por falta de verificación del correo electrónico

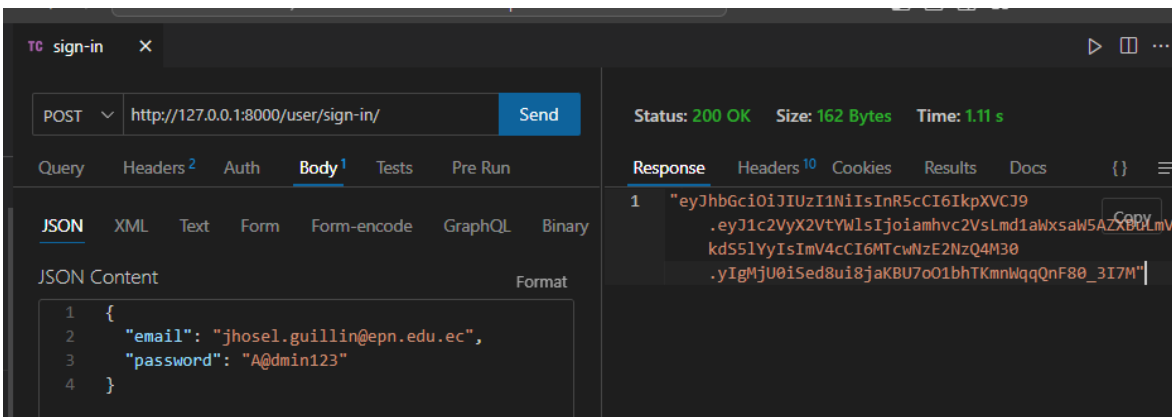


Figura 2.8 Inicio de sesión exitoso

Para la verificación del correo electrónico, se utiliza la siguiente configuración:

- **Método:** POST
- **Endpoint:** *user/send-email-verification/*
- **Entrada:** correo electrónico del usuario que se desea verificar su cuenta.
- **Salida:**
 - o **Estado:** 200 -> Correo electrónico verificado (Figura 2.9) y se cambia el estado de verificación a verdadero en la base de datos (Figura 2.10).
 - o **Estado:** 400 -> Correo electrónico inválido.
 - o **Estado:** 404 -> Usuario no encontrado.

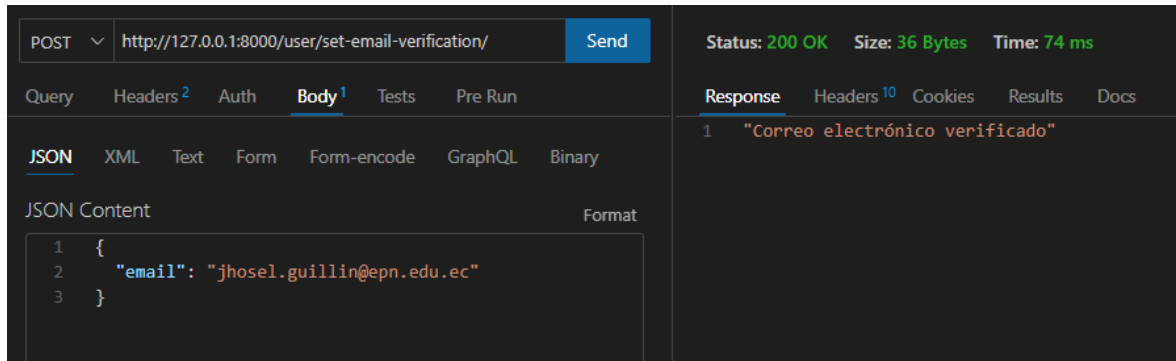


Figura 2.9 Verificación del correo electrónico

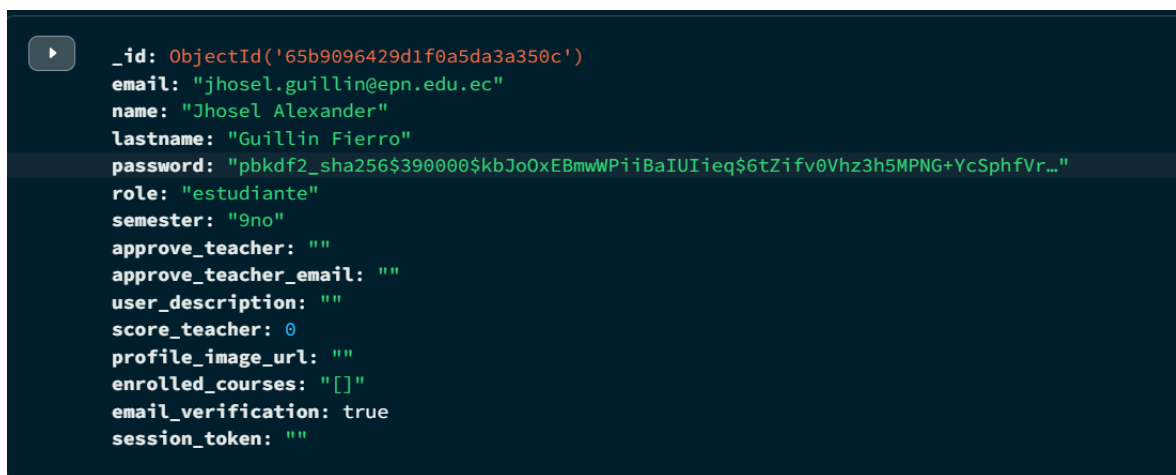


Figura 2.10 Visualización del estado de la verificación del correo electrónico

Las demás pruebas unitarias correspondientes a cada Sprint se visualizan en el Anexo V.

2.4 Revisión del Sprint

En esta fase se evaluó cada historia de usuario correspondiente a los Sprints, en función de sus criterios de aceptación y las pruebas unitarias respectivamente, dentro de un tiempo de 2 horas.

Sprint 1

Tabla 2.14 Revisión Sprint 1

US	Criterios de aceptación	Aprobación
Registro de usuarios	Registro de datos del usuario con su respectiva validación	<input checked="" type="checkbox"/>

	El usuario recibe un correo electrónico de verificación	<input checked="" type="checkbox"/>
	Existe unicidad de un usuario registrado	<input checked="" type="checkbox"/>
	Aplicación de un hash + salt a la contraseña	<input checked="" type="checkbox"/>
	Verificación de email y contraseña para inicio de sesión	<input checked="" type="checkbox"/>
	Generación de un token de sesión	<input checked="" type="checkbox"/>
Mostrar información del usuario y permitir actualización	Desplegar la información del usuario en función de su rol	<input checked="" type="checkbox"/>
	Permitir la actualización de información del usuario	<input checked="" type="checkbox"/>
	Permitir subir o actualizar foto de perfil si el rol es de instructor	<input checked="" type="checkbox"/>
Permitir cierre de sesión	Permitir el cierre de sesión	<input checked="" type="checkbox"/>
	Eliminación del token de sesión	<input checked="" type="checkbox"/>

Sprint 2

Tabla 2.15 Revisión Sprint 2

US	Criterios de aceptación	Aprobación
Permitir ser un instructor	Permitir actualización de los datos del usuario en función de su rol	<input checked="" type="checkbox"/>
	Envío de correos electrónicos al profesor para aprobación	<input checked="" type="checkbox"/>
	Envío de correos electrónicos al estudiante en caso de confirmación para ser instructor	<input checked="" type="checkbox"/>
Cambiar y restaurar contraseña	Permitir cambiar contraseña	<input checked="" type="checkbox"/>
	Envío de correo electrónico para restauración de contraseña	<input checked="" type="checkbox"/>
	Validación de existencia del usuario en la base de datos	<input checked="" type="checkbox"/>
	Permitir restauración de contraseña	<input checked="" type="checkbox"/>
Creación de cursos	Permitir creación de cursos	<input checked="" type="checkbox"/>

Permitir contacto con equipo de 'Virtual-POLI'	Permitir el envío de correos al equipo de soporte de 'Virtual-POLI'	<input checked="" type="checkbox"/>
Permitir crear, obtener, actualizar y eliminar categorías de cursos	Permitir crear una categoría (nombre y descripción)	<input checked="" type="checkbox"/>
	Permitir obtener una categoría por el id	<input checked="" type="checkbox"/>
	Permitir actualizar una categoría dado su id	<input checked="" type="checkbox"/>
	Permitir eliminar una categoría dado su id	<input checked="" type="checkbox"/>

Sprint 3

Tabla 2.16 Revisión Sprint 3

US	Criterios de aceptación	Aprobación
Conocer instructores, cursos destacados y agregados recientemente	Mostrar los cursos destacados en función de la calificación que tiene cada uno	<input checked="" type="checkbox"/>
	Mostrar los instructores destacados en función de la calificación que tiene cada uno	<input checked="" type="checkbox"/>
	Mostrar los 5 cursos agregados recientemente	<input checked="" type="checkbox"/>
Permitir ingresar comentarios y una calificación en los cursos	Permitir ingresar comentarios y una calificación dentro de cada curso siempre y cuando el usuario este registrado en este	<input checked="" type="checkbox"/>
Obtener los cursos por instructor o categoría	Obtener cursos dado un instructor	<input checked="" type="checkbox"/>
	Obtener cursos dado una categoría	<input checked="" type="checkbox"/>

2.5 Retrospectiva del Sprint

Para la retrospectiva de cada Sprint se llevó a cabo una reunión de 2 horas con el equipo Scrum respondiendo a las siguientes preguntas:

- ¿Qué salió bien en el último Sprint?
- ¿Qué salió mal?
- ¿Qué aprendimos?

- ¿Qué deberíamos hacer diferente la próxima vez?

En función a esto se tiene la Tabla 2.17, 2.18 y 2.19, que resumen la retrospectiva de cada Sprint.

Tabla 2.17 Retrospectiva del Sprint 1

Preguntas	Retrospectiva
¿Qué salió bien en el último Sprint?	Se logró completar todas las historias de usuario planificadas dentro del tiempo asignado.
¿Qué salió mal?	<ul style="list-style-type: none"> - Un avance no muy rápido en el desarrollo por la falta de conocimiento del framework. - Creación de un token de sesión sin tiempo de expiración. - Manejo de operaciones del usuario sin un token de sesión. - Uso inadecuado de los serializadores dentro del framework.
¿Qué aprendimos?	<ul style="list-style-type: none"> - Manejar diferentes conectores entre framework y base de datos. - Aumento de seguridad con respecto a la respuesta de una solicitud mediante el uso de tokens de sesión.
¿Qué deberíamos hacer diferente la próxima vez?	<ul style="list-style-type: none"> - Resolver todas las dudas en la reunión diaria respectiva. - Manejar variables de entorno para mayor seguridad.

Tabla 2.18 Retrospectiva del Sprint 2

Preguntas	Retrospectiva
¿Qué salió bien en el último Sprint?	Se logró completar todas las historias de usuario planificadas dentro del tiempo asignado.
¿Qué salió mal?	Tiempos de respuestas grandes en el envío de correos para contacto con Equipo de Virtual-POLI.
¿Qué aprendimos?	Hacer uso del framework para simplificar la reutilización de código.

¿Qué deberíamos hacer diferente la próxima vez?	Segmentar de mejor manera la información ha ser almacenada en la base de datos, con respecto a la colección de cursos.
---	--

Tabla 2.19 Retrospectiva del Sprint 3

Preguntas	Retrospectiva
¿Qué salió bien en el último Sprint?	Se logró completar todas las historias de usuario planificadas dentro del tiempo asignado.
¿Qué salió mal?	Mal manejo en el envío innecesario de información como respuesta.
¿Qué aprendimos?	Consolidar el tipo de información que debe ser enviada como respuesta a una petición.
¿Qué deberíamos hacer diferente la próxima vez?	Establecer mejores criterios de aceptación que permitan el desarrollo de pruebas de una forma más ágil.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

Considerando como objetivo principal el desarrollo de la programación backend de la plataforma web “Virtual-POLI” para complementar la educación académica generada por estudiantes para estudiantes de la Facultad de Ingeniería en Sistemas. Se realizó una simulación de tráfico de usuarios en Postman, para medir el rendimiento de cada endpoint en función de las peticiones realizadas desde el cliente (usuario virtual), con las características establecidas en la Tabla 3.1.

Tabla 3.1 Configuraciones para pruebas de rendimiento

Tipo de configuración	Valor
Perfil de carga	Pico
Usuarios virtuales (VU)	60
Duración de la prueba	5 minutos
Carga base (número de usuarios inicial)	20

Establecer un perfil de carga como “Pico” significa que cuando se llega a un tiempo específico el número de usuarios baja o sube, simulando variabilidad en el número de peticiones que puede llegar a tener un sitio web.

En función a esto, se obtuvo la Figura 3.1, 3.2 y 3.3, respecto a la gestión de usuarios, cursos y categorías, respectivamente, donde se representa el número de solicitudes enviadas en función del crecimiento o decrecimiento del número de usuarios a lo largo del tiempo (5 minutos), teniendo como valor inicial 20 usuarios y máximo 60.

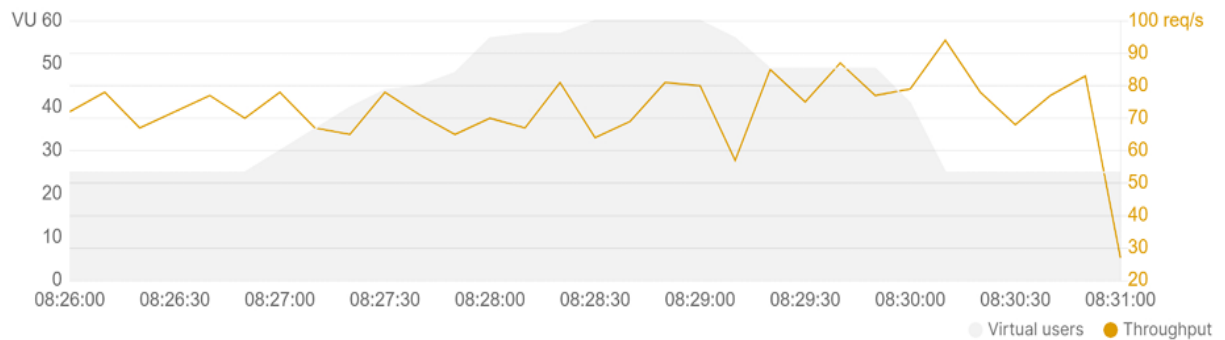


Figura 3.1 Tasa de solicitudes por segundo en la gestión de usuarios

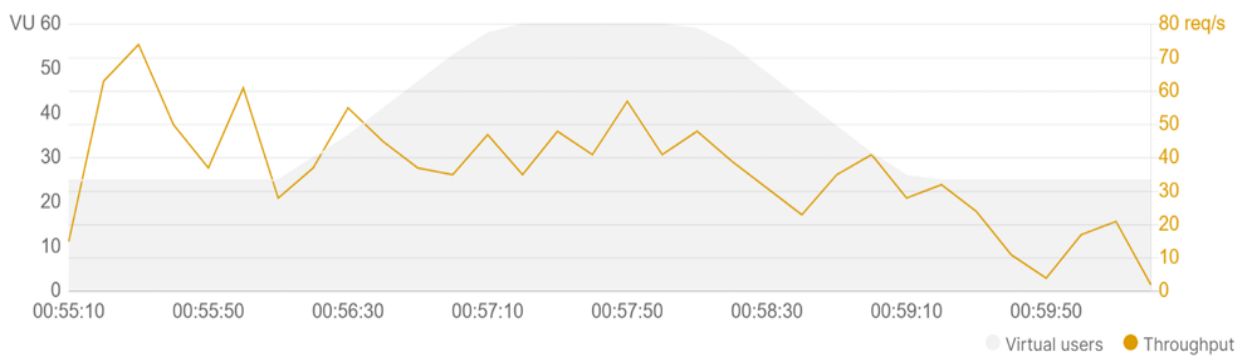


Figura 3.2 Tasa de solicitudes por segundo en la gestión de cursos

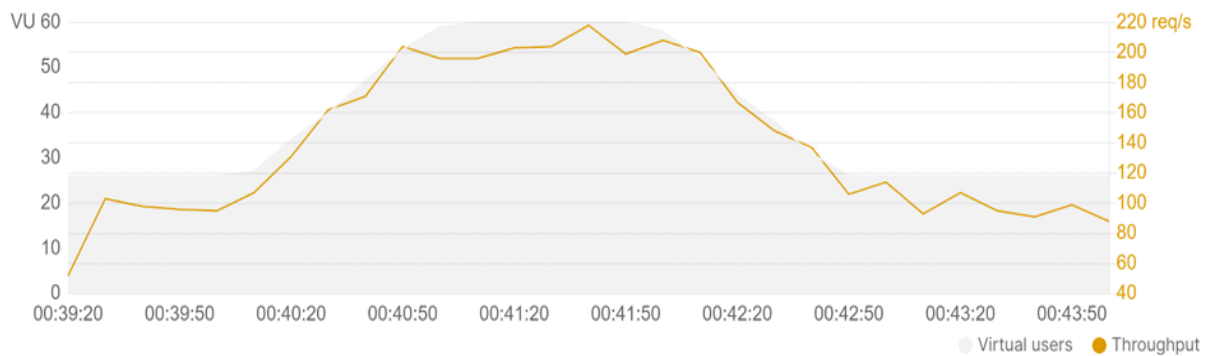


Figura 3.3 Tasa de solicitudes por segundo para la gestión de categorías

La Tabla 3.2, resume las diferentes tendencias obtenidas en función del número total de solicitudes, solicitudes por segundo, tiempo de respuesta promedio, el porcentaje de error (respuesta 404, no se encontró el recurso solicitado) y el número de solicitudes fallidas.

Tabla 3.2 Resumen pruebas de rendimiento

	Total de solicitudes	Solicitudes por segundo	Tiempo promedio (ms)	Error %	Número de solicitudes fallidas
Gestión de usuarios	2,259	7	2,394	5,75	130
Gestión de cursos	1,162	4	4,434	9,98	116
Gestión de categorías	4,008	13	1,635	2,50	100

Luego, mediante la Ecuación 3.1 y la Figura 3.4, se calcula el índice de rendimiento correspondiente a la gestión de usuarios, cursos y categorías, para comprobar la satisfacción del usuario en función del tiempo de respuesta.

$$\frac{\#respuestasSatisfactorias + \frac{\#respuestasTolerantesFallidas}{2}}{\#totalDeSolicitudes}$$

Ecuación 3.1 Índice de rendimiento (Apdex)

1.00 0.94	EXCELLENT
0.93 0.85	GOOD
0.84 0.70	FAIR
0.69 0.50	POOR
0.49 0.00	UNACCEPTABLE

Figura 3.4 Intervalos definidos dentro del cálculo del índice de rendimiento [34]

En base, a esta ecuación se tiene los siguientes índices de rendimiento.

Tabla 3.3 Índices de rendimiento dentro de la gestión de usuarios, cursos y categorías

	Apdex
Gestión de usuarios	0,97
Gestión de cursos	0,95
Gestión de categorías	0,99

Considerando la Figura 3.4 y la Tabla 3.3. Se puede establecer la satisfacción del usuario final con el rendimiento del proyecto como **Excelente**, teniendo como conclusión que el desarrollo backend llevado a cabo en el presente proyecto es funcional, donde su principio es una arquitectura de software escalable, segura y eficiente, desarrollada en Python y Django, junto a una base de datos (MongoDB) sólida y capaz de satisfacer las necesidades del sistema.

3.2 Conclusiones

- El desarrollo de una arquitectura de software implementada en Django como framework, que se basa en una estructura modular permitiendo la división de la aplicación en diferentes componentes (módulos), se obtiene como consecuencia

alta escalabilidad y eficiencia, además, se tiene medidas enfocadas en la autenticación y autorización de usuarios mediante el uso de tokens de sesión, dando como resultado la protección contra inyecciones SQL, ataques de Cross-Site Scripting (XSS), entre otros.

- El porcentaje de error obtenido en las pruebas de rendimiento del proyecto puede estar relacionado a los parámetros de configuración de la prueba, considerando que es una simulación de peticiones, si el número de peticiones por segundo disminuye, el porcentaje de error puede disminuir.
- La elección e implementación de una base de datos no relacional facilitó la comunicación con el controlador, permitiendo lectura, almacenamiento, actualización y eliminación de datos en función de las necesidades del sistema, evitando el manejo de relaciones o uniones entre tablas, otorgando una respuesta más rápida al usuario final.
- Los puntos de acceso creados en la API permitieron al frontend dejar de lado la lógica del sistema y enfocarse en lo que ve el usuario, de esta manera cada endpoint otorga una funcionalidad específica con respecto a la gestión de usuarios, cursos y categorías.
- El protocolo simple de transferencia de correo (SMTP) de Gmail es un servicio externo que agilizó el proceso para la comunicación mediante correos electrónicos entre estudiantes, instructores y el equipo de soporte de Virtual-POLI, limitando la creación de un servidor SMTP en la infraestructura.
- La implementación del desarrollo backend fue realizado por el componente de Infraestructura y Seguridad, teniendo como resultado el siguiente subdominio <https://dev.polivirtual.me/>, donde se puede consumir cada endpoint desarrollado en este componente.

3.3 Recomendaciones

- El presente proyecto se basa en una arquitectura monolítica por el alcance definido, es decir, los diferentes componentes que forman la aplicación estructuran una sola entidad, pero si se pretende seguir con su desarrollo a un alcance mayor, es recomendable utilizar microservicios, dividiendo cada componente en un servicio pequeño e independiente y de esta manera evitar problemas de disponibilidad.

- Para el envío de correos electrónicos se utilizó el protocolo simple de transferencia de correo de Gmail, lo cual influye en cierto grado en el rendimiento de la respuesta de cada endpoint donde es utilizado, motivo por el cual, se recomienda implementar un servidor SMTP para el envío de correos, dando como resultado un mejor tiempo de respuesta para el usuario final.
- En el caso que se desee probar cada endpoint mediante alguna herramienta como Postman o Thunder Client, es recomendable leer la documentación para conocer las entradas que posee cada punto de acceso y su respectiva salida con su estado.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] B. G. Salvat, «Redalyc.org,» 18 Diciembre 2017. [En línea]. Available: <https://www.redalyc.org/jatsRepo/3314/331455826005/331455826005.pdf>. [Último acceso: 01 Diciembre 2023].
- [2] J. Juviler, «HubSpot,» 23 Octubre 2023. [En línea]. Available: <https://blog.hubspot.com/website/back-end-developer>. [Último acceso: 04 Diciembre 2023].
- [3] AWS, «AWS,» 2023. [En línea]. Available: <https://aws.amazon.com/es/compare/the-difference-between-frontend-and-backend/#:~:text=El%20desarrollo%20de%20back%20end,sistemas%20se%20comunican%20entre%20s%C3%AD..> [Último acceso: 01 Diciembre 2023].
- [4] R. Touza, Manual 51 Producto Mínimo Viable, Valencia: Centro Europeo de Empresas Innovadoras de Valencia, 2017.
- [5] CEDIA, «CEDIA,» [En línea]. Available: <https://cedia.edu.ec/docs/LMS.pdf>. [Último acceso: 19 Diciembre 2023].
- [6] D. W. Titthasiri, 06 Noviembre 2013. [En línea]. Available: https://d1wqtxts1xzle7.cloudfront.net/36768469/Wanwipa-libre.pdf?1424872951=&response-content-disposition=inline%3B+filename%3DA_Comparison_of_E_Learning_and_Tradition.pdf&Expires=1705460867&Signature=Fn2rAcSs3C9towOEml4l6RgylzYHjGreApAvsFk024~ucbMT7PGvub. [Último acceso: 20 Diciembre 2023].
- [7] A. Hassenburg, «Escholarship,» 2009. [En línea]. Available: <https://escholarship.org/content/qt3859m52h/qt3859m52h.pdf>. [Último acceso: 21 Diciembre 2023].
- [8] M. Area y J. Adell, «ResearchGate,» 18 Marzo 2016. [En línea]. Available: <https://blogs.fcecon.unr.edu.ar/asesoriapedagogica/wp-content/uploads/sites/3/2020/03/e-learning.pdf>. [Último acceso: 22 Diciembre 2023].

- [9] D. N. Kaushal. [En línea]. Available: https://www.lkouniv.ac.in/site/writereaddata/siteContent/202004120815046665Nandita_Kaushal_E_learning.pdf. [Último acceso: 22 Diciembre 2023].
- [10] C. Staff, «Coursera,» 29 Noviembre 2023. [En línea]. Available: <https://www.coursera.org/articles/back-end-developer>. [Último acceso: 25 Diciembre 2023].
- [11] A. F. Mestres. [En línea]. Available: https://openaccess.uoc.edu/bitstream/10609/141486/1/Tecnologias%20y%20herramientas%20para%20el%20desarrollo%20web_Modulo1_Introduccion%20al%20frontend%20y%20backend.pdf. [Último acceso: 25 Diciembre 2023].
- [12] Autentia, «www.autentia.com,» [En línea]. Available: https://www.autentia.com/wp-content/uploads/libros/Back_IntroduccionAlBack-Autentia.pdf. [Último acceso: 25 Diciembre 2023].
- [13] IBM, «IBM,» [En línea]. Available: <https://www.ibm.com/es-es/topics/rest-apis>. [Último acceso: 23 Diciembre 2023].
- [14] R. Hat, «Red Hat,» 31 Julio 2023. [En línea]. Available: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>. [Último acceso: 23 Diciembre 2023].
- [15] K. Ljung, 18 Mayo 2021. [En línea]. Available: <https://www.diva-portal.org/smash/get/diva2:1559372/FULLTEXT01.pdf>. [Último acceso: 25 Diciembre 2023].
- [16] D. Ogut, «Medium,» 04 Julio 2021. [En línea]. Available: <https://ogutdgnn.medium.com/what-does-clean-code-mean-2190e4aed818>. [Último acceso: 25 Diciembre 2023].
- [17] Oracle, «Oracle,» [En línea]. Available: <https://www.oracle.com/database/what-is-a-relational-database/>. [Último acceso: 27 Diciembre 2023].
- [18] MongoDB, «MongoDB,» [En línea]. Available: <https://www.mongodb.com/compare/relational-vs-non-relational-databases>. [Último acceso: 27 Diciembre 2023].
- [19] Z. Tejada, «Microsoft,» [En línea]. Available: <https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data>. [Último acceso: 27 Diciembre 2023].
- [20] D. V., «Hostinger,» 11 Febrero 2023. [En línea]. Available: <https://www.hostinger.es/tutoriales/java-vs-python>. [Último acceso: 28 Diciembre 2023].
- [21] K. Gupta, «FreelancingGig,» 23 Octubre 2018. [En línea]. Available: <https://www.freelancinggig.com/blog/2018/10/23/what-programming-language-udemy-is-written-in/>. [Último acceso: 28 Diciembre 2023].

- [22] B. V. Ridge, «MEDIUM MULTIMEDIA,» 22 Diciembre 2023. [En línea]. Available: <https://www.mediummultimedia.com/apps/cual-es-el-mejor-framework-de-python/>. [Último acceso: 28 Diciembre 2023].
- [23] A. Ramirez, «Platzi,» 2019. [En línea]. Available: <https://platzi.com/blog/7-razones-mongodb/>. [Último acceso: 28 Diciembre 2023].
- [24] J. Roche, «Deloitte,» [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-scrum.html>. [Último acceso: 28 Diciembre 2023].
- [25] S. Sachdeva, 06 Junio 2016. [En línea]. Available: https://d1wqtxts1xzle7.cloudfront.net/46357609/11_ijecs-libre.pdf?1465482218=&response-content-disposition=inline%3B+filename%3DScrum_Methodology.pdf&Expires=1705642556&Signature=fYbgisZmvgg6lgtH4Keza1jdMWApyj0JPz5wjRAbbktk2elFle1sKJn9mm2h0ucv3Q4r1OfBi4lr. [Último acceso: 28 Diciembre 2023].
- [26] J. Ruiz, «JorgeRuiz.Agile,» 08 Agosto 2018. [En línea]. Available: <https://jorgeruizagile.com/2018/08/04/los-4-valores-de-la-agilidad-el-manifiesto-agil/>. [Último acceso: 28 Diciembre 2023].
- [27] R. Chito, 05 Febrero 2022. [En línea]. Available: <https://blog.buhoos.com/21-anos-del-manifiesto-agil/>. [Último acceso: 28 Diciembre 2023].
- [28] AWS, «AWS,» [En línea]. Available: <https://aws.amazon.com/es/what-is/scrum/#:~:text=Scrum%20is%20a%20management%20framework,experience%2C%20and%20adapt%20to%20change..> [Último acceso: 28 Diciembre 2023].
- [29] A. Arrarte. [En línea]. Available: https://alvaroarrarte.com/fases-de-la-metodologia-scrum/#Desarrollar_las_Epicas. [Último acceso: 28 Diciembre 2023].
- [30] A. R. Mesa, «OpenWebinars,» 19 Diciembre 2018. [En línea]. Available: <https://openwebinars.net/blog/que-es-un-sprint-scrum/>. [Último acceso: 28 Diciembre 2023].
- [31] J. Martins, «Asana,» 10 Octubre 2022. [En línea]. Available: <https://asana.com/es/resources/what-is-kanban>. [Último acceso: 28 Diciembre 2023].
- [32] J. Palacios. [En línea]. Available: <https://jeronimopalacios.com/agile/el-primer-tablero-kanban/>. [Último acceso: 29 Diciembre 2023].
- [33] G. Mancuzo, «ComparaSoftware,» 21 Abril 2021. [En línea]. Available: <https://blog.comparasoftware.com/tecnicas-de-estimacion-en-scrum/#:~:text=la%20metodolog%C3%ADa%20Scrum.-,%2BFQu%C3%A9%20es%20la%20estimaci%C3%B3n%20en%20Scrum%3F,el%20trabajo%20y%20el%20tiempo..> [Último acceso: 03 Enero 2024].

- [34] R. Bourgault, «DZone,» 16 Julio 2021. [En línea]. Available: <https://dzone.com/articles/apdex-rating-reporting-in-jmeter>. [Último acceso: 04 Enero 2023].

5 ANEXOS

ANEXO I. Historias de Usuario

Tabla 5.1 Historia de usuario US-002

Número: US-002	Usuario: Estudiante/Instructor
<p>Nombre de la Historia: Mostrar información del usuario y permitir actualización</p> <p>Como estudiante o instructor. Quiero poder ver mi información de usuario actual en la página web. Para actualizarla según sea necesario.</p>	
Esfuerzo: 5	
<p>Conversación: Esta funcionalidad permitirá a los estudiantes y profesores acceder y revisar su información de usuario actual, incluyendo detalles como nombre, apellido, correo electrónico, semestre, entre otros, según el rol que tenga (estudiante-instructor).</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> - En función del rol (estudiante-instructor) se mostrará la información respectiva a cada usuario. - Un instructor puede actualizar su foto de perfil. - El correo electrónico no puede ser actualizado. 	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Desplegar la información del usuario en función de su rol. - Permitir la actualización de información del usuario. - Permitir subir o actualizar foto de perfil si el rol es de instructor. 	

Tabla 5.2 Historia de usuario US-003

Número: US-003	Usuario: Estudiante
-----------------------	----------------------------

<p>Nombre de la Historia: Permitir ser un instructor</p> <p>Como estudiante. Quiero tener la opción de poder convertirme en instructor. Para crear nuevos cursos en la plataforma web 'Virtual-POLI'.</p>
<p>Esfuerzo: 8</p>
<p>Conversación: Esta funcionalidad permitirá a los estudiantes solicitar ser instructores en la plataforma, lo que les dará la capacidad de crear y gestionar sus propios cursos. Los estudiantes interesados en convertirse en instructores podrán enviar una solicitud a un profesor de la Facultad de Ingeniería en Sistemas y en el caso de recibir la aprobación del profesor, el estudiante se convertirá en instructor.</p>
<p>Observaciones:</p> <ul style="list-style-type: none"> - El estudiante que desea convertirse en instructor deberá completar cierta información como: nombres, apellidos y correo electrónico del profesor que lo aprueba y una descripción, donde se detalle información importante al nuevo rol que va a desarrollar. - Al llenar esta información, la plataforma web envía una solicitud al correo electrónico del profesor ingresado por el estudiante. El profesor ingresa al enlace correspondiente enviado al correo electrónico y en caso de aceptar se le asigna el nuevo rol de instructor al estudiante permitiendo la gestión de cursos en la plataforma web. - En el caso que el profesor apruebe al estudiante, el estudiante recibirá un mensaje de confirmación, caso contrario seguirá con un rol de estudiante.
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Permitir actualización de los datos del usuario en función de su rol. - Envío de correos electrónicos al profesor para aprobación. - Envío de correos electrónicos al estudiante en caso de confirmación para ser instructor.

Tabla 5.3 Historia de usuario US-004

Número: US-004	Usuario: Estudiante/Instructor
-----------------------	---------------------------------------

<p>Nombre de la Historia: Cambiar y restaurar contraseña</p> <p>Como usuario registrado en la plataforma. Quiero poder cambiar o restaurar mi contraseña en caso de olvido. Para mantener la seguridad de mi cuenta y acceder a la plataforma de forma segura.</p>
<p>Esfuerzo: 5</p>
<p>Conversación: Se les permitirá a los usuarios registrados en la plataforma cambiar su contraseña dentro de su perfil, donde se pedirá que ingresen una nueva contraseña. Además, en el caso que deseen restaurar su contraseña, el usuario deberá ingresar un correo electrónico válido y existente en la plataforma para que de esta manera le llegue un enlace al correo electrónico donde se podrá realizar este proceso.</p>
<p>Observaciones:</p> <ul style="list-style-type: none"> - Tanto para cambiar o restaurar la contraseña, se aplicará un hash + salt para la seguridad de los datos. - Se debe verificar la existencia del correo electrónico para la restauración de la contraseña.
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Si un usuario que ha iniciado sesión desea cambiar de contraseña, si cumple con las validaciones establecidas por el frontend, se envía una petición al backend y se cambia la contraseña. - Para la restauración de contraseña, se envía un correo electrónico desde el frontend, si en el backend se verifica que este correo no es válido o no existe en la base de datos, se retorna una respuesta de usuario no encontrado. En el caso que, si exista, se envía un correo electrónico al usuario con un link para poder restaurar la contraseña.

Tabla 5.4 Historia de usuario US-005

Número: US-005	Usuario: Estudiante/Instructor
Nombre de la Historia: Permitir cierre de sesión	

<p>Como usuario.</p> <p>Quiero poder cerrar sesión en la plataforma.</p> <p>Para proteger la seguridad y privacidad de mi información.</p>
<p>Esfuerzo: 1</p>
<p>Conversación: Todo usuario registrado e iniciado sesión en la plataforma, puede cerrar sesión de forma segura cuando haya completado sus actividades.</p>
<p>Observaciones:</p> <ul style="list-style-type: none"> - Cuando el usuario cierre sesión el token de sesión se elimina.
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Permitir el cierre de sesión. - Eliminación del token de sesión.

Tabla 5.5 Historia de usuario US-006

<p>Número: US-006</p>	<p>Usuario: Estudiante</p>
<p>Nombre de la Historia: Conocer instructores, cursos destacados y agregados recientemente</p> <p>Como usuario.</p> <p>Quiero ver la información de instructores, cursos destacados y agregados recientemente.</p> <p>Para saber que cursos tienen mayor factibilidad de tomarse.</p>	
<p>Esfuerzo: 1</p>	
<p>Conversación: Esta funcionalidad permitirá a los usuarios conocer cuáles son los instructores, cursos destacados y agregados recientemente, en función de la calificación que posee cada uno respectivamente.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> - Esta funcionalidad no debe influir con el registro, es decir, cualquier persona que ingrese a la página web, podrá visualizar esta información sin necesidad de registrarse. 	
<p>Criterios de aceptación:</p>	

- Mostrar los cursos destacados en función de la calificación que tiene cada uno.
- Mostrar los instructores destacados en función de la calificación que tiene cada uno.
- Mostrar los 5 cursos agregados recientemente.

Tabla 5.6 Historia de usuario US-007

Número: US-007	Usuario: Instructor
Nombre de la Historia: Creación de cursos Como instructor registrado en la plataforma web. Quiero tener la capacidad de crear cursos. Para gestionar mi contenido educativo.	
Esfuerzo: 5	
Conversación: Se permitirá a los instructores registrar nuevos cursos en la plataforma con el contenido que consideren necesario.	
Observaciones: <ul style="list-style-type: none"> - Para la creación de cursos, el frontend enviará la información respectiva como: nombre, descripción, categoría, instructor, módulos, contenido del curso (título y video), video de introducción al curso e imagen del curso. 	
Criterios de aceptación: <ul style="list-style-type: none"> - Cuando se cree un curso, se enviará una respuesta "Curso agregado" desde el servidor. 	

Tabla 5.7 Historia de usuario US-008

Número: US-008	Usuario: Estudiante
Nombre de la Historia: Permitir ingresar comentarios y una calificación en los cursos Como estudiante. Quiero ingresar comentarios y una calificación en los cursos que he completado. Para compartir mi experiencia y proporcionar una retroalimentación positiva.	

Esfuerzo: 1
Conversación: Los estudiantes interactuarán con los cursos de manera más dinámica y participativa, lo que fomentará la colaboración y el intercambio de conocimientos entre los usuarios de la plataforma.
Observaciones: <ul style="list-style-type: none"> - Los comentarios y la calificación estarán disponibles para todos los usuarios que ingresen a la plataforma, no es necesario estar registrado para visualizar estos comentarios.
Criterios de aceptación: <ul style="list-style-type: none"> - Permitir ingresar comentarios y una calificación dentro de cada curso siempre y cuando el usuario este registrado en este.

Tabla 5.8 Historia de usuario US-009

Número: US-009	Usuario: Estudiante/Instructor
Nombre de la Historia: Permitir contacto con equipo de Virtual-POLI	
<p>Como usuario.</p> <p>Quiero contactarme con el equipo de soporte de la plataforma web.</p> <p>Para emitir comentarios del funcionamiento de la plataforma o solucionar problemas relacionados al perfil, gestión de cursos, entre otros.</p>	
Esfuerzo: 5	
Conversación: Esta funcionalidad permitirá que los usuarios tengan un canal directo de comunicación con el equipo de soporte de la plataforma web, para la resolución de cualquier problema.	
Observaciones: Ninguna	
Criterios de aceptación: <ul style="list-style-type: none"> - Permitir el envío de correos al equipo de soporte de 'Virtual-POLI'. 	

Tabla 5.9 Historia de usuario US-010

Número: US-010	Usuario: Administrador
<p>Nombre de la Historia: Permitir crear, obtener, actualizar y eliminar categorías de cursos.</p> <p>Como administrador.</p> <p>Quiero crear, obtener, actualizar y eliminar las categorías de los cursos.</p> <p>Para tener una mejor estructura con respecto a cada curso.</p>	
Esfuerzo: 5	
<p>Conversación: Esta funcionalidad permitirá al administrador crear, obtener, actualizar y eliminar las categorías respecto a cada curso, para que cada curso creado sea dividido dentro de una categoría para una mejor organización de la información.</p>	
<p>Observaciones:</p> <p>Ninguna</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Permitir crear una categoría (nombre y descripción). - Permitir obtener una categoría por el id. - Permitir actualizar una categoría dado su id. - Permitir eliminar una categoría dado su id. 	

Tabla 5.10 Historia de usuario US-011

Número: US-011	Usuario: Estudiante
<p>Nombre de la Historia: Obtener los cursos por instructor o categoría</p> <p>Como estudiante.</p> <p>Quiero ver los cursos que le pertenecen a un instructor o pertenecen a una categoría.</p> <p>Para encontrar cursos rápidamente.</p>	
Esfuerzo: 1	

Conversación: Esta funcionalidad permitirá que los usuarios puedan conocer que cursos le pertenece a cada instructor y en función de esto seleccionar algún curso. Además, se podrá hacer una búsqueda de los cursos en función de la categoría.
Observaciones: Ninguna
Criterios de aceptación: <ul style="list-style-type: none">- Obtener cursos dado un instructor.- Obtener cursos dado una categoría.

ANEXO II. Tableros Kanban

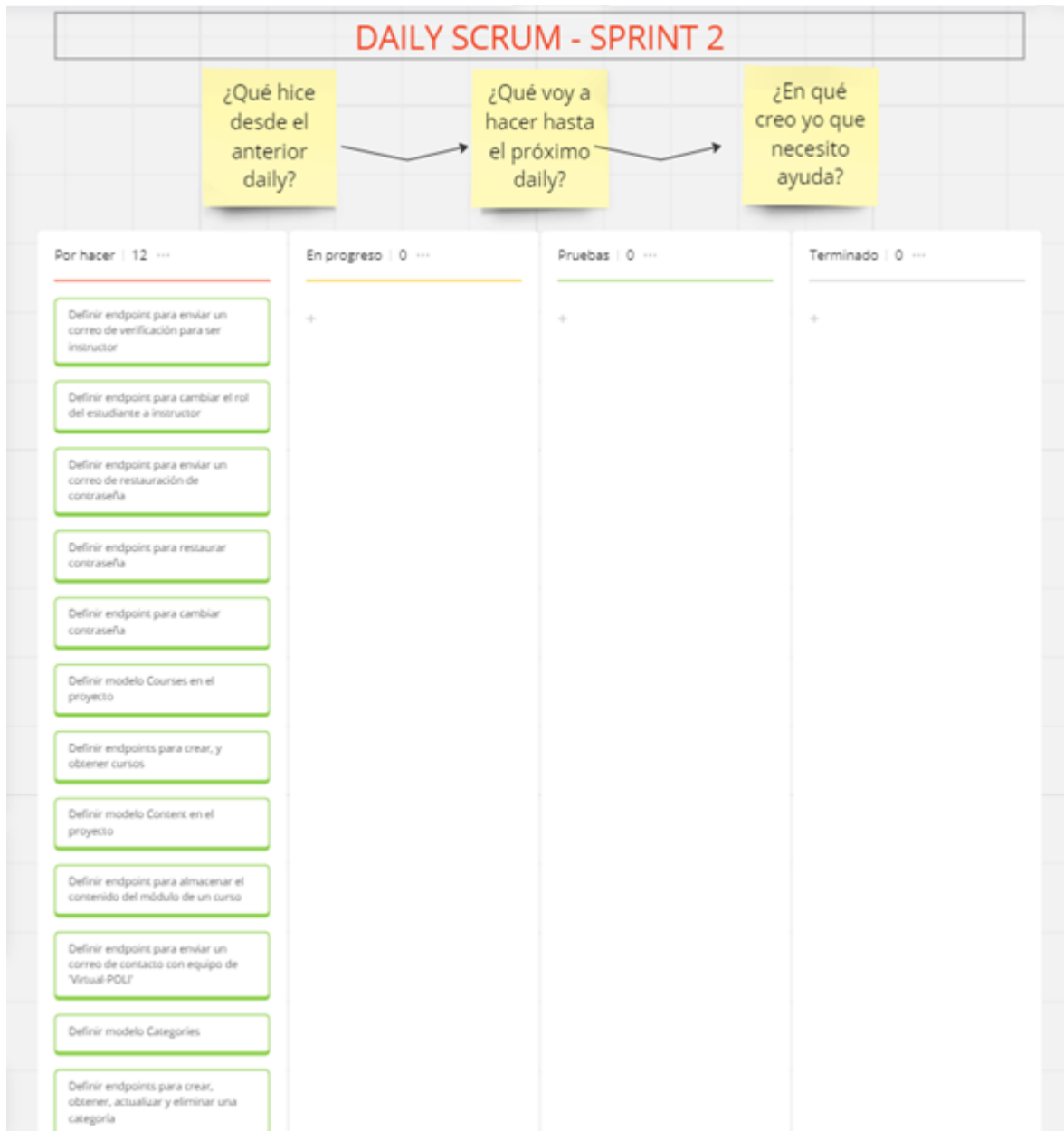


Figura 5.1 Tablero Kanban - Sprint 2

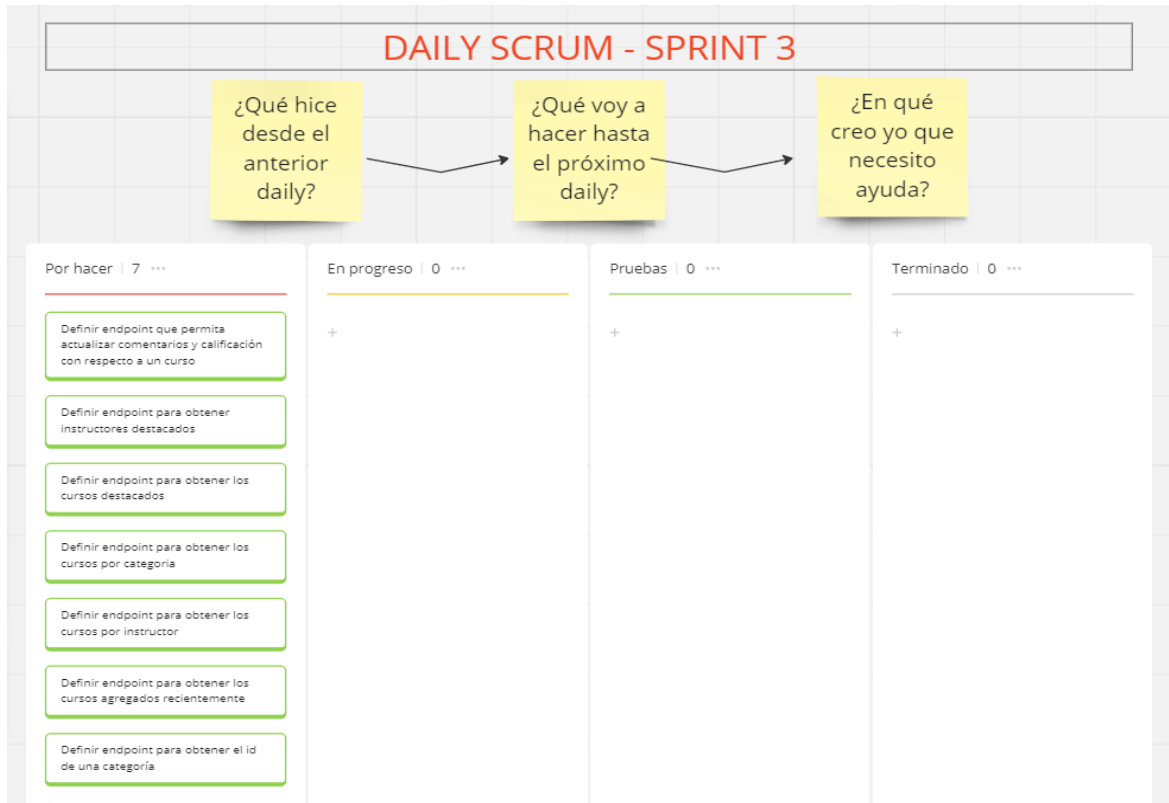


Figura 5.2 Tablero Kanban - Sprint 3

ANEXO III. Modelo de base de datos

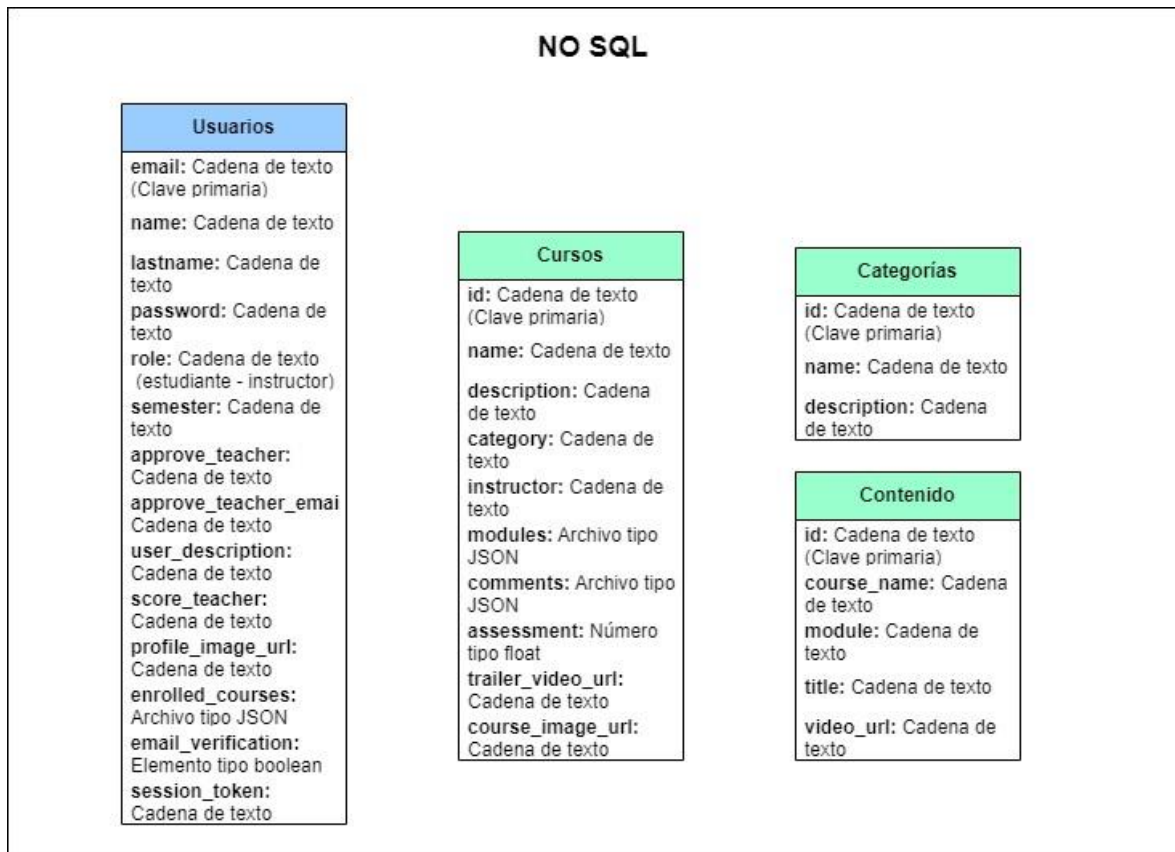


Figura 5.3 Modelo de base de datos considerado para el proyecto

ANEXO IV. Código respecto al componente desarrollado

El fragmento de código de la Tabla 5.11 se utiliza para definir el modelo “Usuario” que se utilizó para hacer referencia a un estudiante o a un instructor. Este modelo posee diferentes campos para el almacenamiento en la base de datos como: correo electrónico, nombres, apellidos, contraseña, rol (estudiante o instructor), semestre, nombre y correo electrónico del profesor que aprueba solicitud para que el estudiante sea instructor, descripción del usuario, calificación como profesor, imagen de perfil si es un instructor, cursos inscritos del usuario, verificación del correo y el token de sesión.

Tabla 5.11 Modelo User

```

from django.db import models
from decouple import config

class User(models.Model):
    email = models.EmailField(max_length=100, primary_key=True,
unique=True)
    name = models.CharField(max_length=100)
    lastname = models.CharField(max_length=100)
    password = models.CharField(max_length=100)
    role = models.CharField(max_length=50)

```

```

semester = models.CharField(max_length=10)
approve_teacher = models.CharField(max_length=100, blank=True)
approve_teacher_email = models.EmailField(max_length=100,
blank=True)
user_description = models.TextField(blank=True)
score_teacher = models.FloatField(default=0.0)
profile_image_url =
models.ImageField(upload_to=config('URL_IMAGE_INSTRUCTOR_STORAGE'),
blank=True)
enrolled_courses = models.JSONField(default=dict, blank=True)
email_verification = models.BooleanField(default=False)
session_token = models.CharField(max_length=300, default="")

def __str__(self):
    RETURN SELF.EMAIL

```

El fragmento de código de la Tabla 5.12, se utiliza para enviar correos electrónicos, teniendo como parámetros el asunto y mensaje del correo, correo de donde se envía y correo destinatario.

Tabla 5.12 Enviar correos electrónicos

```

# Send an email
def send_email(email, subject, message):
    email = EmailMessage(
        subject,
        message,
        settings.EMAIL_HOST_USER,
        [email],
    )

    email.fail_silently = False # If the email fails, it will not be
silent
    email.send()

```

El fragmento de código de la Tabla 5.13, primero genera un token de sesión para un correo electrónico específico y una fecha de expiración que ayudará a que, si el token es utilizado después de cierta cantidad de días, este será inválido. Luego, se verifica si el token no ha expirado y si es así se valida que el token pertenezca algún usuario registrado en la base de datos.

Tabla 5.13 Generar y verificar token de sesión

```

# Generate JWT token
def generate_token(user_email):
    token_payload = {
        'user_email': user_email,
        'exp': datetime.utcnow() + timedelta(days=1),
    }

```



```

token = encode(token_payload, config('SECRET_KEY_TOKEN'),
algorithm='HS256')
return token

# Verify Token
def verify_token(request):
    # Verify the expiration and decode the token
    token = request.headers.get('Authorization', '').split(' ')[1]
    try:
        decoded_payload = jwt.decode(token, config('SECRET_KEY_TOKEN'),
algorithms=['HS256'])

        user_email = decoded_payload['user_email']

        # Verify the current session token
        user = User.objects.get(email=user_email)
        user_session_token = user.session_token

        if user_session_token == token:
            return user_email
        else:
            return False
    except jwt.ExpiredSignatureError:
        return False
    except jwt.InvalidTokenError:
        return False

```

El fragmento de código de la Tabla 5.14, se utiliza para verificar si un correo electrónico es válido, con la finalidad de evitar que se envíe códigos maliciosos desde la interfaz de usuario cuando un usuario de registre o en el envío de correos electrónicos destinados al equipo de soporte de la plataforma.

Tabla 5.14 Verificar un correo válido

```

# Verify if it is a valid email
def is_valid_email(email):
    try:
        validate_email(email)
        return True
    except ValidationError:
        return False

```

El fragmento de código de la Tabla 5.15, se utiliza para actualizar, obtener y eliminar un usuario en función del token de sesión. Además, permite la actualización de los cursos a los cuales el estudiante está registrado.

Tabla 5.15 Endpoint para actualizar – eliminar un usuario y obtener - actualizar cursos inscritos del usuario

```

# API views (Update and delete user - Get/update enrolled courses)
@csrf_exempt

```

```

@api_view(['GET', 'PUT', 'DELETE'])
def user_api(request):
    user_token = verify_token(request) # return the email of the user
    if the token is valid

        if user_token is False:
            return JsonResponse({'mensaje': f'Acceso no autorizado'},
safe=False, status=401)

        else:
            # Get the enrolled courses of the user
            if request.method == 'GET':
                try:
                    user = User.objects.get(email=user_token)
                    enrolled_courses = []

                    for course in user.enrolled_courses:

                        try:
                            # Get the course
                            enrolled_course =
Course.objects.get(name=course["name"])
                            enrolled_course_serializer =
CourseSerializer(enrolled_course)

                            # Transform the course to a dictionary
                            enrolled_course_dict =
enrolled_course_serializer.data

                            # Add the state of the course
                            enrolled_course_dict["state"] = course["state"]

                            # Add the converted course to the list
                            enrolled_courses.append(enrolled_course_dict)

                            # Remove some fields in enrolled_courses
                            for index in range(len(enrolled_courses)):
                                del enrolled_courses[index]['description']
                                del enrolled_courses[index]['modules']
                                del enrolled_courses[index]['comments']
                                del
enrolled_courses[index]['trailer_video_url']

                                except Course.DoesNotExist:
                                    return JsonResponse({'mensaje': f'Error al
retornar los cursos del usuario'}, safe=False, status=404)

                                    return JsonResponse(enrolled_courses, safe=False,
status=200)

                                except User.DoesNotExist:
                                    return JsonResponse({'mensaje': f'Usuario no
encontrado'}, safe=False, status=404)

                                # Update user
                                # Now it can update the fields trailer_video_url and
course_image_url
                                elif request.method == 'PUT':
                                    user = User.objects.get(email=user_token)

```

```

        # Update only enrolled courses
        if "enrolled_courses" in request.data:
            user_serializer = UserSerializer(user,
data={'enrolled_courses': user.enrolled_courses +
request.data["enrolled_courses"]}, partial=True)

            if user_serializer.is_valid():
                user_serializer.save()
                return JsonResponse({'mensaje': f'Cursos del
usuario actualizados'}, safe=False)

        # Update the rest of the user information
        else:
            user_serializer = UserSerializer(user,
data=request.data, partial=True)

            if user_serializer.is_valid():

                if(request.FILES.get('profile_image_url')):
                    profile_image =
request.FILES.get('profile_image_url')
                    current_time =
datetime.utcnow().strftime('%Y%m%d%H%M%S')
                    profile_image.name =
f'profile_image_{user.email}_{current_time}'
                    user_serializer.profile_image_url =
profile_image

                    user_serializer.save()

                return JsonResponse({'mensaje': f'Usuario
actualizado'}, safe=False, status=200)

            return JsonResponse({'mensaje': f'Error al actualizar
usuario'}, safe=False, status=400)

        # Delete user
        elif request.method == 'DELETE':
            try:
                user = User.objects.get(email=user_token)
                user.delete()
                return JsonResponse({'mensaje': f'Usuario eliminado'},
safe=False, status=200)

            except User.DoesNotExist:
                return JsonResponse({'mensaje': f'Usuario no
encontrado'}, safe=False, status=404)

```

El fragmento de código de la Tabla 5.16, se utiliza para registrar un nuevo usuario, donde se permite la verificación del correo electrónico, la unicidad del usuario y el envío de un correo electrónico para verificación de la cuenta.

Tabla 5.16 Endpoint para el registro de usuarios

```
# Sign up
@csrf_exempt
@api_view(['POST'])
def sign_up(request):
    if request.method == 'POST':
        data = JSONParser().parse(request)

        # Verify if the email is valid
        if is_valid_email(data.get("email")) is False:
            return JsonResponse("Correo electrónico inválido",
safe=False, status=400)
        else:
            # Check if the user is registered
            existing_user =
User.objects.filter(email=data.get("email")).first()

            if existing_user is not None:
                # Existing user
                response_data = {'mensaje': f'Este usuario ya existe'}
                status = 402
            else:
                # Hash the password with a random salt
                data["password"] = make_password(data["password"])

                # If you are not registered, add user
                user_serializer = UserSerializer(data=data)
                if user_serializer.is_valid():
                    user_serializer.save()
                    user = user_serializer.data

                    response_data = {'mensaje': f'Usuario agregado'}
                    status = 200

                    # Send email to verify
                    subject = "Verificación de correo electrónico"
                    message = f"Hola {user.get('name')}
{user.get('lastname')},\n\nPor favor, verifica tu correo electrónico
haciendo click en el siguiente
enlace:\n\n{config('URL_FRONTEND')}/verify-email/\n\nGracias,\n\nEl
equipo de Virtual Poli."
                    send_email(user.get("email"), subject, message)
                else:
                    response_data = {'mensaje': f'Error al guardar el
usuario'}
                    status = 400

            return JsonResponse(response_data, safe=False,
status=status)
```

El fragmento de código de la Tabla 5.17, se utiliza para permitir el inicio de sesión de un usuario, siempre y cuando este haya verificado su correo electrónico.

Tabla 5.17 Endpoint para el inicio de sesión

```
# Sign in
@csrf_exempt
```

```

@api_view(['POST'])
def sign_in(request):
    if request.method == 'POST':
        data = JSONParser().parse(request)

        # Verify if the email is valid
        if is_valid_email(data.get("email")) is False:
            return JsonResponse({'mensaje': f'Correo electrónico
inválido'}, safe=False, status=400)
        else:
            # Get the email and password from the request
            email = data.get("email")
            password = data.get("password")

            # Verify if email is not empty
            if email != "" and password != "":
                try:
                    user = User.objects.get(email=email) # Get the user
of the BD
                    email_verification = user.email_verification # Get
the email verification of the user

                    if email_verification:
                        password_user = user.password

                        # Compare hashed password with the password
entered by the user
                        if check_password(password, password_user):
                            user_serializer = UserSerializer(user)

                            # Generate JWT token
                            user_session_token =
generate_token(user_serializer.data.get("email"))

                            # Update the session token of the user
                            user_serializer = UserSerializer(user,
data={'session_token': user_session_token}, partial=True)
                            if user_serializer.is_valid():
                                user_serializer.save()

                                return JsonResponse({'token':
user_serializer.data['session_token']}, safe=False)
                            else:
                                return JsonResponse({"mensaje": "Contraseña
incorrecta"}, status=401)

                        else:
                            return JsonResponse({"mensaje": "Correo
electrónico no verificado"}, status=403)

                    except User.DoesNotExist:
                        return JsonResponse({"mensaje": "Usuario no
encontrado"}, status=404)

                else:
                    return JsonResponse({"mensaje": "Correo electrónico y
contraseña no ingresados"}, status=400)

```

El fragmento de código de la Tabla 5.18, se basa en cerrar la sesión de un usuario tomando como mecanismo la eliminación del token de sesión.

Tabla 5.18 Endpoint para cerrar sesión

```
# Sign out
@csrf_exempt
@api_view(['PUT'])
def sign_out(request):
    if request.method == 'PUT':
        user_token = verify_token(request) # return the email of the
        user if the token is valid

        if user_token is False:
            return JsonResponse({'mensaje': f'Acceso no autorizado'},
            safe=False, status=401)

        else:
            try:
                user = User.objects.get(email=user_token)
                user_serializer = UserSerializer(user,
                data={'session_token': 'None'}, partial=True)

                if user_serializer.is_valid():
                    user_serializer.save()
                    return JsonResponse({'mensaje': f'Sesión cerrada'},
                    safe=False, status=200)
                else:
                    return JsonResponse({'mensaje': f'Error al cerrar
                    sesión'}, safe=False, status=400)

            except User.DoesNotExist:
                return JsonResponse({'mensaje': f'Usuario no
                encontrado'}, safe=False, status=404)
```

El fragmento de código de la Tabla 5.19 describe los endpoints definidos dentro del modelo Usuario.

Tabla 5.19 Endpoints dentro del modelo Usuario

```
from django.urls import path
from Users import views

urlpatterns = [
    path('users/', views.user_api, name='user-api'),
    path('users/sign-up/', views.sign_up, name='user-sign-up'), # User
    sign up
    path('users/sign-in/', views.sign_in, name='user-sign-in'), # User
    sign in
    path('users/sign-out/', views.sign_out, name='user-sign-out'), #
    User sign out
    path('users/set-email-verification/', views.set_email_verification,
    name='user-set-email-verification'), # Set email verification
    path('users/change-password/', views.change_password, name='user-
    change-password'), # Change password
```

```

path('users/send-email-to-restore-password/',
views.send_email_to_restore_password, name='user-restore-password'), #
Send email to restore password
path('users/restore-password/', views.restore_password, name='user-
restore-password'), # Restore password
path('users/contact-with-us/', views.contact_with_us, name='user-
contact-with-us'), # Contact with us
path('users/be-an-instructor/', views.be_an_instructor, name='user-
be-an-instructor'), # Be an instructor
path('users/featured-teachers/', views.featured_teachers,
name='user-featured-teachers'), # Get featured teachers
path('users/send-email-to-approve-teacher/',
views.send_email_to_approve_teacher, name='user-sned-email-to-approve-
teacher'), # Send email to approve teacher
path('users/profile/', views.get_user_profile, name='user-get-user-
profile'), # Get user profile
path('users/instructor-profile/<str:name_lastname>',
views.get_instructor_profile, name='user-get-inst-profile'), # Get
instructor profile
path('users/last-watched-course/', views.add_last_watched_course,
name='user-add-last-watched-course'), # Add last watched course
path('users/last-watched-course/<str:course_name>',
views.get_last_watched_course, name='user-get-last-watched-course'), #
Get last watched course
path('users/is-enrolled-in-course/<str:course_name>',
views.is_enrolled_in_course, name='user-is-enrolled-in-course'), # Is
enrolled in course
path('users/instructors-by-key-word/<str:key_word>',
views.get_instructors, name='user-get-instructors'), # Get instructors
or courses
]

```

El resto del código del componente se puede encontrar en el siguiente [repositorio de Github](#).

ANEXO V. Pruebas unitarias

The screenshot shows a REST client interface. The request is a POST to `http://127.0.0.1:8000/user/sign-up/`. The body is a JSON object with the following content:

```
1 {
2   "email": "alexanderguillin1999@gmail.com",
3   "name": "Alexander",
4   "lastname": "Guillin Fierro",
5   "password": "admin123",
6   "role": "estudiante",
7   "semester": "9no",
8   "enrolled_courses": []
9 }
```

The response is a 409 Conflict with a size of 37 Bytes and a time of 34 ms. The response body is:

```
1 {
2   "mensaje": "Este usuario ya existe"
3 }
```

Figura 5.4 Verificación de unicidad en el registro de usuarios

The screenshot shows a REST client interface. The request is a POST to `http://127.0.0.1:8000/user/sign-up/`. The body is a JSON object with the following content:

```
1 {
2   "email": "<script>@gmail.com",
3   "name": "Jhosel Alexander",
4   "lastname": "Guillin Fierro",
5   "password": "admin123",
6   "role": "estudiante",
7   "semester": "9no",
8   "enrolled_courses": [
9   ]
10 }
```

The response is a 400 Bad Request with a size of 39 Bytes and a time of 4 ms. The response body is:

```
1 "Correo electrónico inválido"
```

Figura 5.5 Verificación de un correo electrónico válido

The screenshot shows a REST client interface. The request is a POST to `http://127.0.0.1:8000/user/sign-in/`. The body is a JSON object with the following content:

```
1 {
2   "email": "jhosel.guillin1@epn.edu.ec",
3   "password": "admin1234"
4 }
```

The response is a 404 Not Found with a size of 36 Bytes and a time of 50 ms. The response body is:

```
1 {
2   "mensaje": "Usuario no encontrado"
3 }
```

Figura 5.6 Verificación de existencia del usuario para inicio de sesión

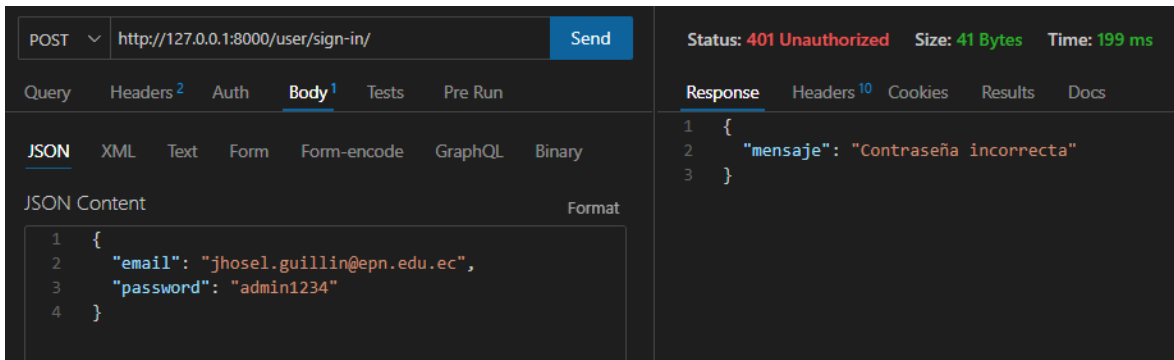


Figura 5.7 Verificación de contraseña para inicio de sesión

Para cerrar sesión de un usuario, se utiliza la siguiente configuración:

- **Método:** PUT
- **Endpoint:** *user/sign-out/*
- **Entrada:** token de sesión
- **Salida:**
 - o **Estado:** 200 -> Sesión cerrada (Figura 5.9).
 - o **Estado:** 400 -> Error al cerrar sesión.
 - o **Estado:** 401 -> Acceso no autorizado (Figura 5.8).
 - o **Estado:** 404 -> Usuario no encontrado.

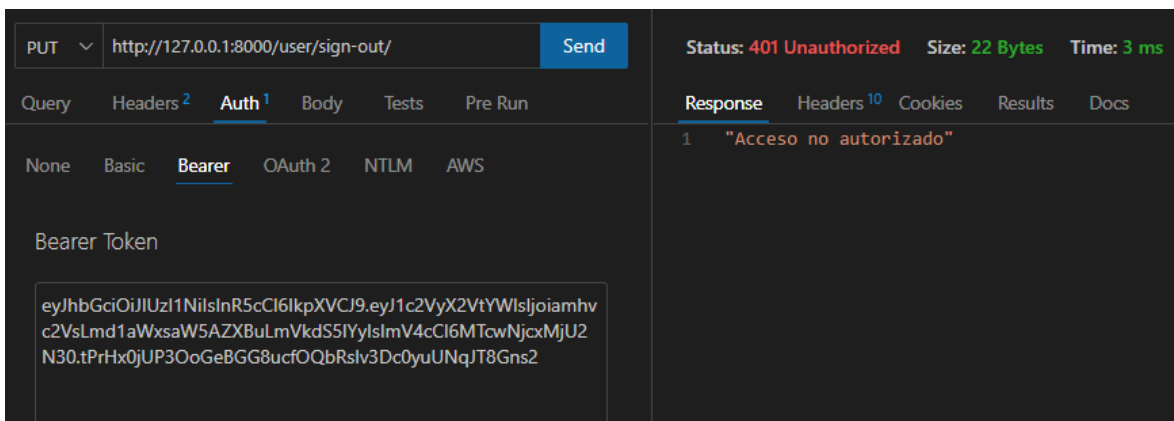


Figura 5.8 Cerrar sesión con token no válido

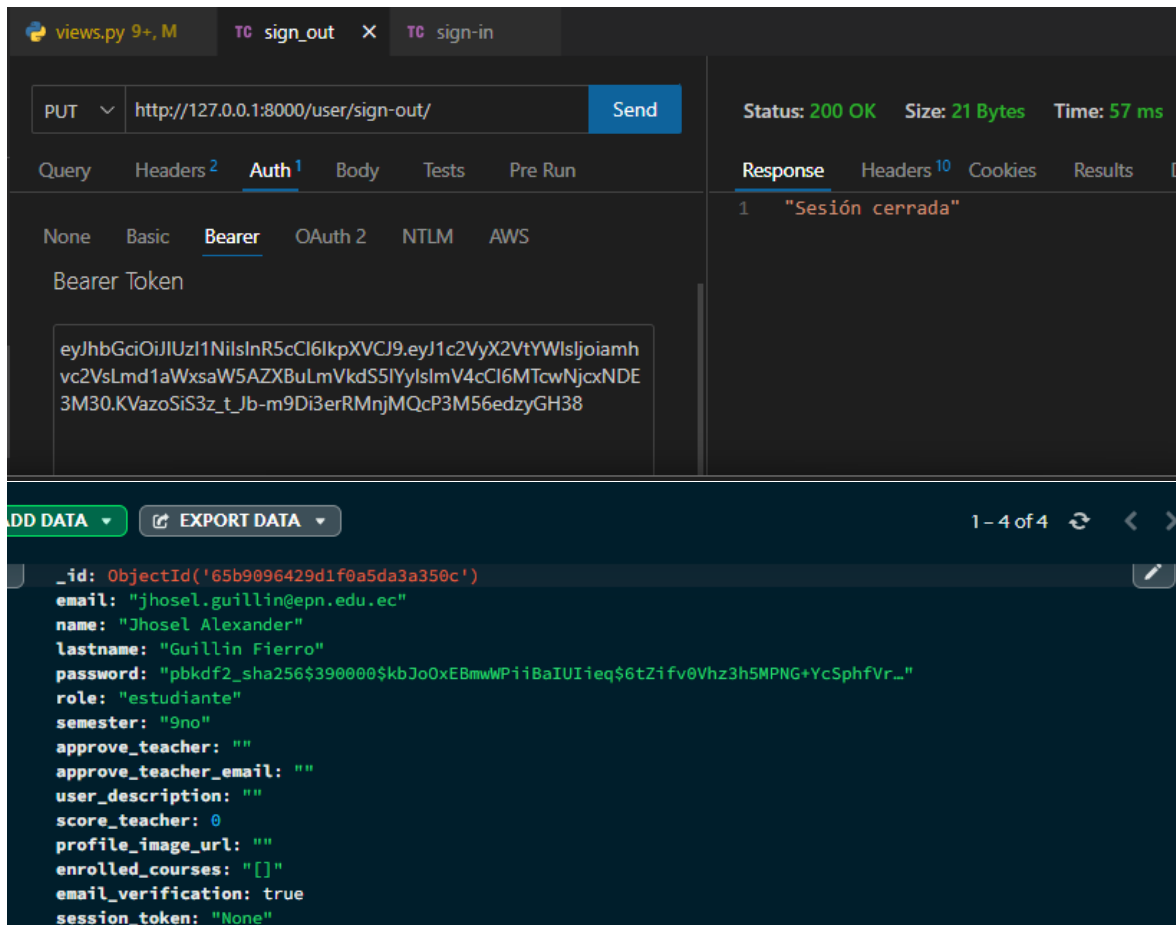


Figura 5.9 Cerrar sesión con token válido

Para actualizar la contraseña de un usuario, se utiliza la siguiente configuración:

- **Método:** PUT
- **Endpoint:** *user/change-password/*
- **Entrada:** token de sesión y nueva contraseña
- **Salida:**
 - o **Estado:** 200 -> Contraseña actualizada (Figura 5.10).
 - o **Estado:** 400 -> Error al actualizar contraseña.
 - o **Estado:** 401 -> Acceso no autorizado.
 - o **Estado:** 404 -> Usuario no encontrado.

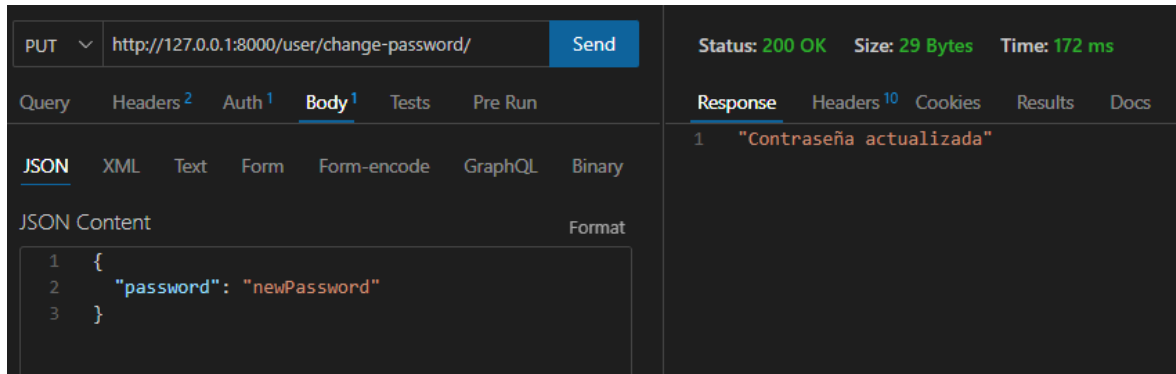


Figura 5.10 Actualización de contraseña

Para enviar un correo de restauración de contraseña, se utiliza la siguiente configuración:

- **Método:** PUT
- **Endpoint:** *user/send-email-to-restore-password/*
- **Entrada:** correo electrónico del usuario que desea restaurar la contraseña
- **Salida:**
 - o **Estado:** 200 -> Correo electrónico enviado (Figura 5.12).
 - o **Estado:** 400 -> Correo electrónico inválido.
 - o **Estado:** 404 -> Usuario no encontrado (Figura 5.11).

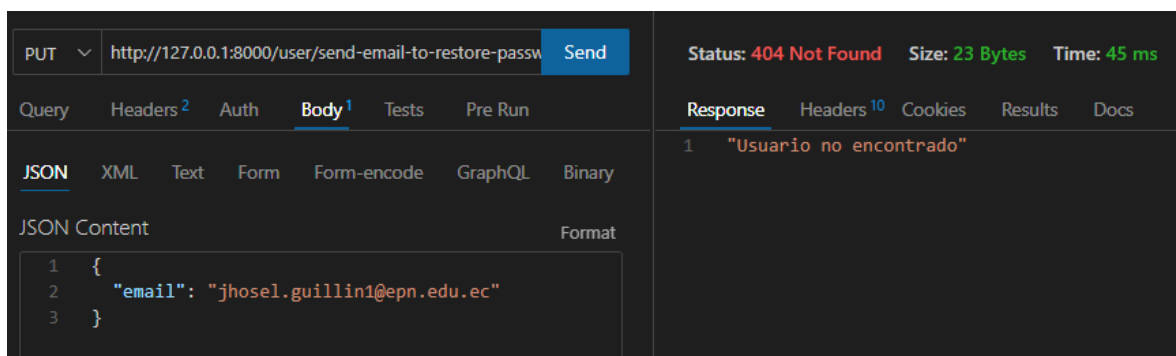


Figura 5.11 Enviar correo para restaurar contraseña - usuario no válido

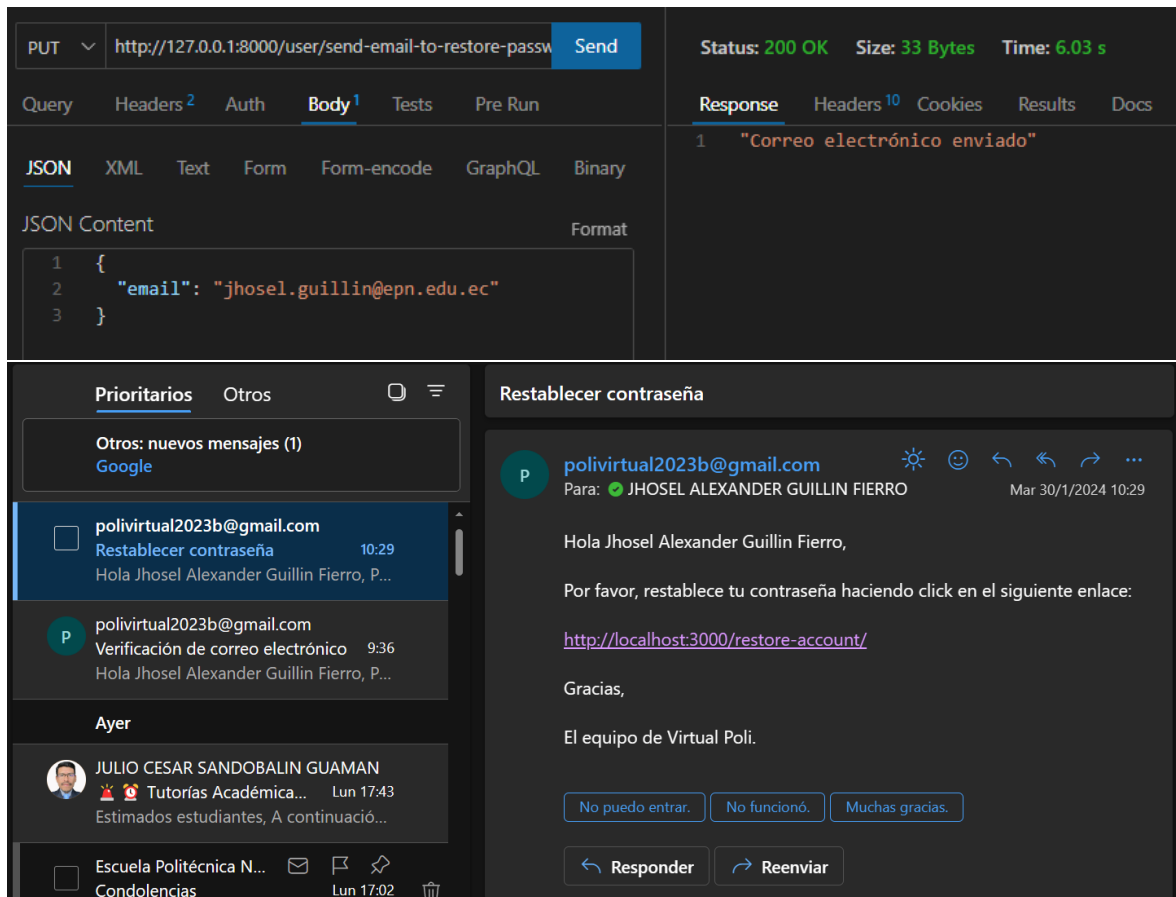


Figura 5.12 Enviar correo para restaurar contraseña - usuario válido

Luego de enviar el correo de restauración de contraseña, para restaurar contraseña y cambiar en la base de datos, se utiliza la siguiente configuración:

- **Método:** PUT
- **Endpoint:** `user/restore-password/`
- **Entrada:** correo electrónico y nueva contraseña para restauración.
- **Salida:**
 - o **Estado:** 200 -> Contraseña actualizada (Figura 5.13).
 - o **Estado:** 400 -> Correo electrónico inválido o error al actualizar contraseña.
 - o **Estado:** 404 -> Usuario no encontrado.

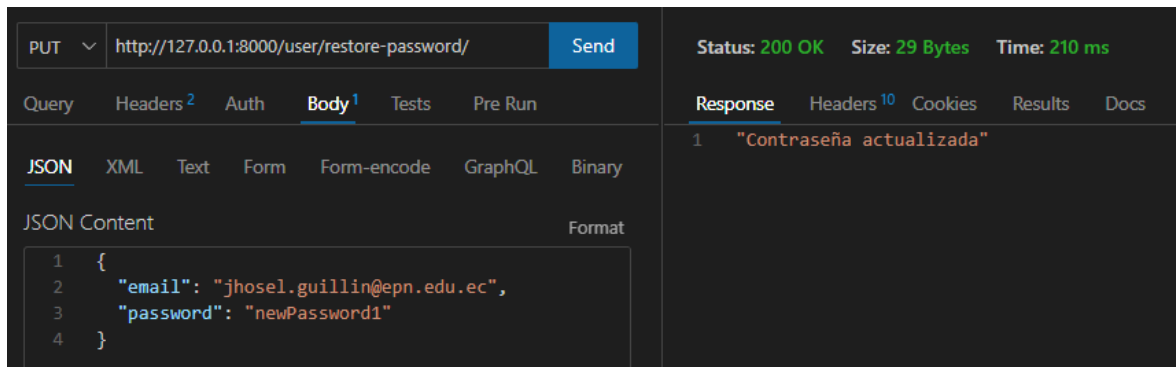


Figura 5.13 Restaurar contraseña

Para enviar un mensaje de correo electrónico al equipo de soporte de Virtual-POLI, se utiliza la siguiente configuración:

- **Método:** POST
- **Endpoint:** *user/contact-with-us/*
- **Entrada:** nombre, correo y mensaje de la persona que necesita contactarse con equipo de Virtual-POLI.
- **Salida:**
 - o **Estado:** 200 -> Correo electrónico enviado (Figura 5.14).
 - o **Estado:** 400 -> Correo electrónico inválido.

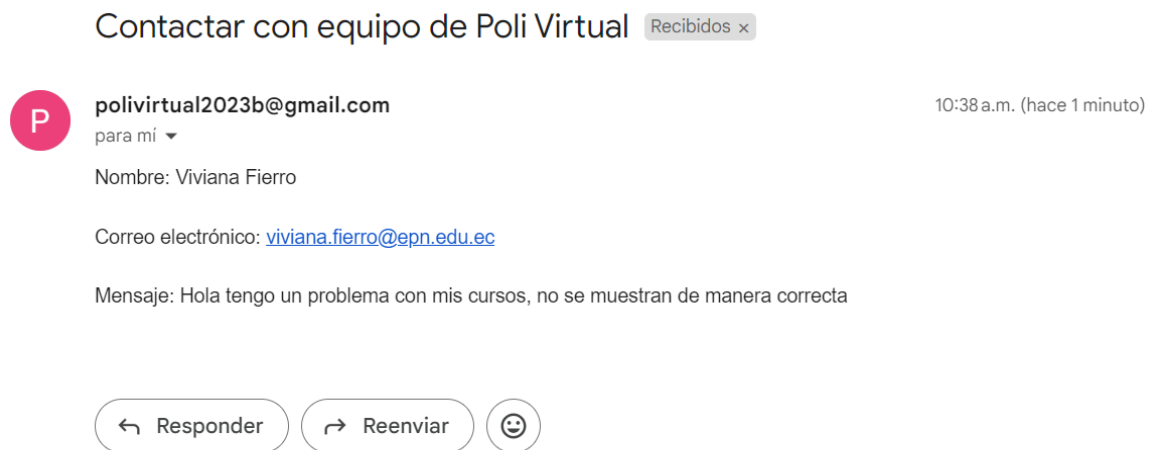
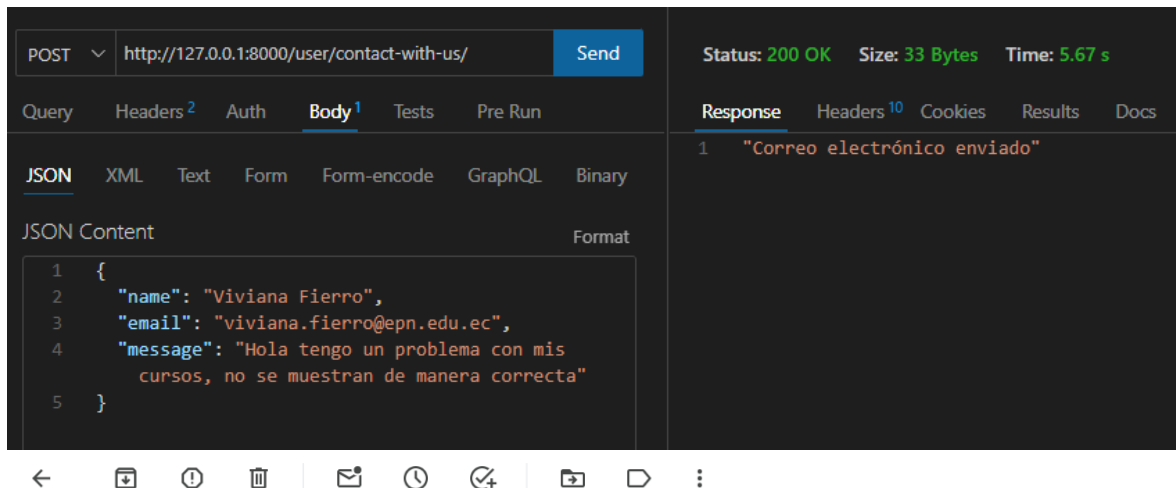


Figura 5.14 Envío de correo para contactar con equipo de Virtual-POLI

Para que un estudiante envíe una solicitud para ser instructor, se utiliza la siguiente configuración:

- **Método:** PUT
- **Endpoint:** *user/send-email-to-approve-teacher/*
- **Entrada:** token de sesión, correo electrónico y nombre completo del profesor quien aprueba la solicitud.
- **Salida:**
 - o **Estado:** 200 -> Correo electrónico enviado (Figura 5.15).
 - o **Estado:** 400 -> Correo electrónico inválido.

- **Estado:** 401 -> Acceso no autorizado.
- **Estado:** 404 -> Usuario no encontrado.

The image shows a REST client interface with a PUT request to `http://127.0.0.1:8000/user/send-email-to-approve-teach`. The request body is a JSON object: `{ "approve_teacher": "Jhosel Fierro", "approve_teacher_email": "alexanderguillin1999@gmail.com" }`. The response is `"Correo electrónico enviado"`. Below the REST client, an email interface shows an incoming message from `polivirtual2023b@gmail.com` with the subject "Solicitud para ser Instructor en la plataforma Poli Virtual". The email content includes a request for authorization to become an instructor, the student's name (Jhosel Alexander Guillin Fierro), institutional email (`jhosel.guillin@epn.edu.ec`), and a link to `http://localhost:3000/approve-to-be-teacher/`.

Figura 5.15 Envío de correo para aprobación de ser instructor

Para actualizar el rol del estudiante en la base de datos, se utiliza la siguiente configuración:

- **Método:** PUT
- **Endpoint:** `user/be-an-instructor/`
- **Entrada:** correo electrónico del estudiante que actualiza su rol a instructor.
- **Salida:**
 - **Estado:** 200 -> Rol del estudiante actualizado (Figura 5.16).
 - **Estado:** 400 -> Correo electrónico inválido.

- **Estado:** 404 -> Usuario no encontrado.

The image shows a REST client interface. The top part displays a PUT request to `http://127.0.0.1:8000/user/be-an-instructor/` with a JSON body: `{ "email": "jhosel.guillin@epn.edu.ec" }`. The response is a 200 OK status with a message: `"Rol del estudiante actualizado"`. Below this, there is a screenshot of an email inbox. The selected email is from `polivirtual2023b@gmail.com` with the subject `Solicitud para ser Instructor e...` and the body text: `Saludos cordiales Jhosel Alexander Guillin Fierro, Nos comunicamos con usted para notificarle que su solicitud para ser Instructor en la plataforma Poli Virtual ha sido aprobada. Gracias, El equipo de Poli Virtual.` There are buttons for `Muchas gracias.`, `Muchas gracias por la confirmación.`, `¡Qué buena noticia!`, `Responder`, and `Reenviar`.

Figura 5.16 Confirmación para ser instructor

Para obtener la información de un usuario que ha iniciado sesión, se utiliza la siguiente configuración:

- **Método:** GET
- **Endpoint:** `user/get-user-profile/`
- **Entrada:** token de sesión
- **Salida:**
 - **Estado:** 200 -> Retorna información del usuario (Figura 5.17).
 - **Estado:** 401 -> Acceso no autorizado.
 - **Estado:** 404 -> Usuario no encontrado.

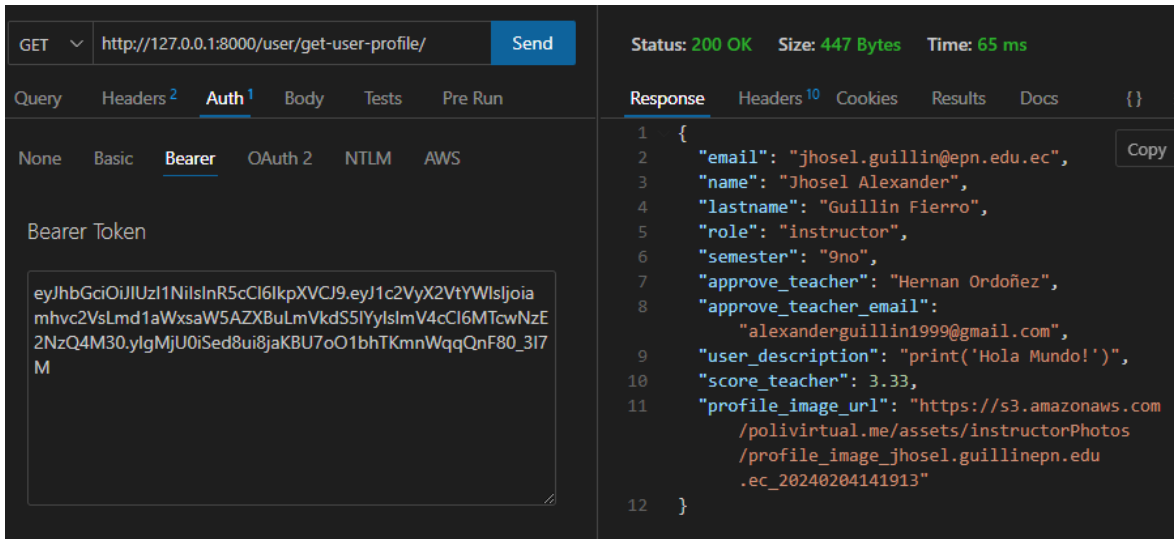


Figura 5.17 Obtener información del usuario

Para obtener la información de un instructor, se utiliza la siguiente configuración:

- **Método:** GET
- **Endpoint:** `user/get-instructor-profile/<str: name_lastname>/`
- **Entrada:** nombres y apellidos del instructor (Nombres-Apellidos)
- **Salida:**
 - o **Estado:** 200 -> Retorna información del instructor (Figura 5.18).
 - o **Estado:** 404 -> Usuario no encontrado.

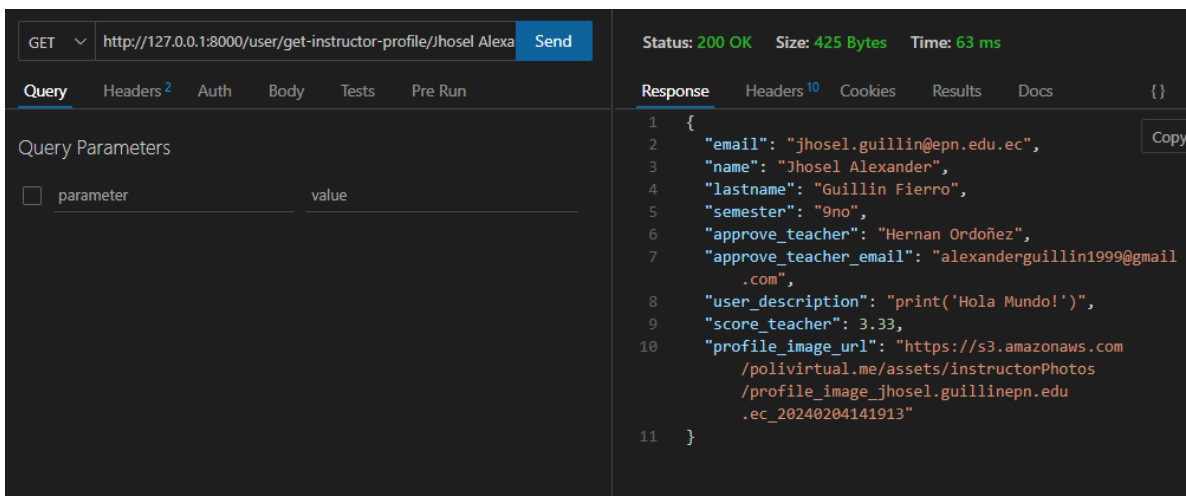


Figura 5.18 Obtener información de un instructor

Para obtener los instructores destacados, se utiliza la siguiente configuración:

- **Método:** GET
- **Endpoint:** *user/featured-teachers/*
- **Entrada:** Ninguna
- **Salida:**
 - o **Estado:** 200 -> Retorna la información de los instructores destacados (Figura 5.19); se considera que un instructor es destacado si su calificación es mayor o igual a 4.
 - o **Estado:** 404 -> No hay instructores destacados.

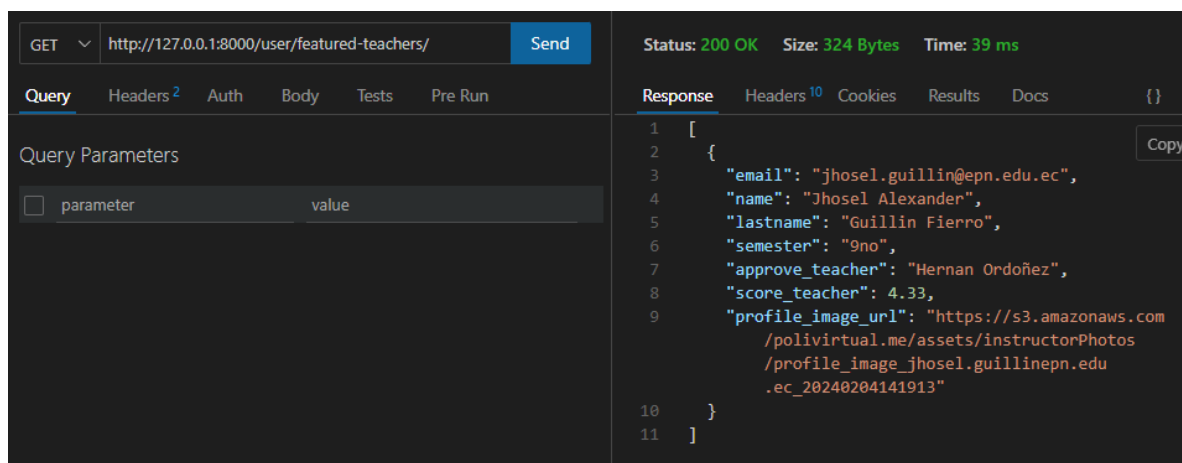


Figura 5.19 Obtener instructores destacados

Para realizar la inscripción de un nuevo curso con respecto a un estudiante, se utiliza la siguiente configuración:

- **Método:** PUT
- **Endpoint:** *user/*
- **Entrada:** token de sesión y el nombre y el estado del curso inscrito.
- **Salida:**
 - o **Estado:** 200 -> Cursos del usuario actualizados (Figura 5.20).
 - o **Estado:** 401 -> Acceso no autorizado.

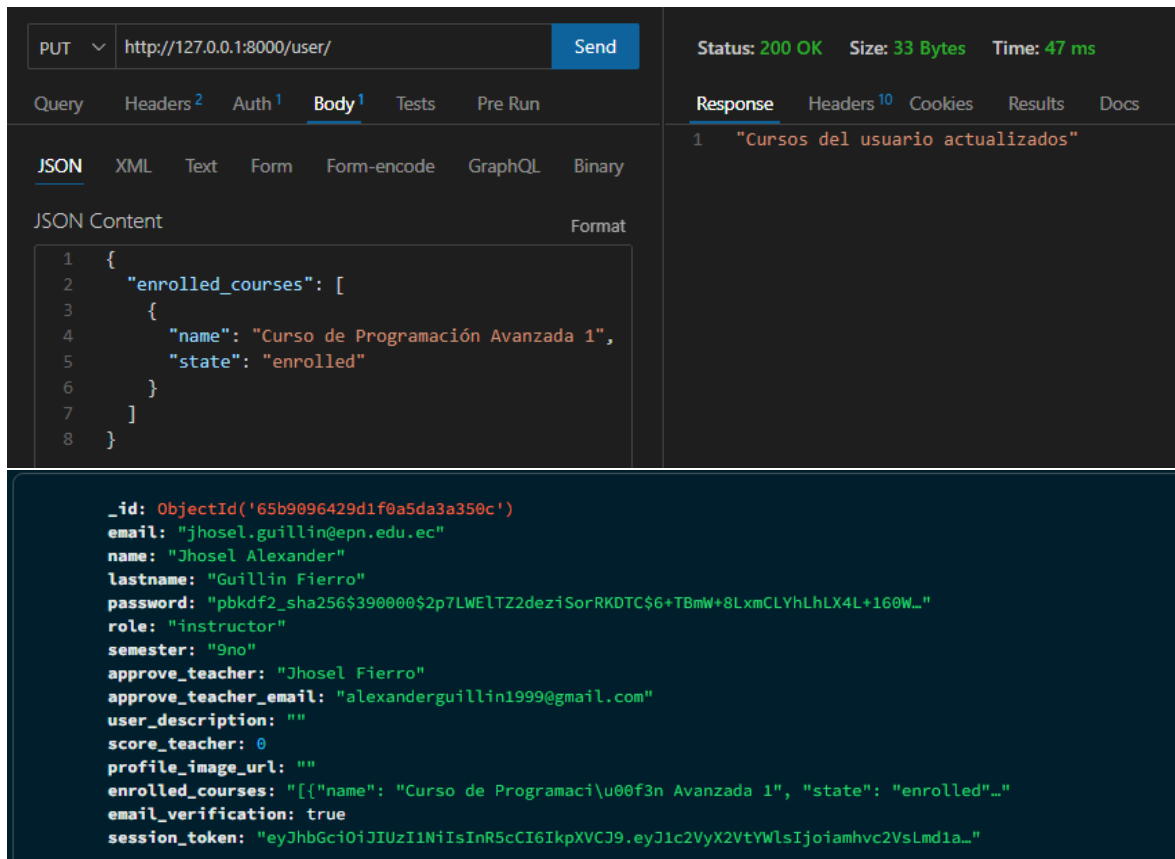


Figura 5.20 Inscripción a un nuevo curso

Para actualizar la información de un usuario, se utiliza la siguiente configuración:

- **Método:** PUT
- **Endpoint:** *user/*
- **Entrada:** token de sesión, nombres, apellidos, semestre, descripción e imagen del usuario a actualizar.
- **Salida:**
 - o **Estado:** 200 -> Usuario actualizado (Figura 5.21).
 - o **Estado:** 400 -> Error al actualizar usuario.
 - o **Estado:** 401 -> Acceso no autorizado.

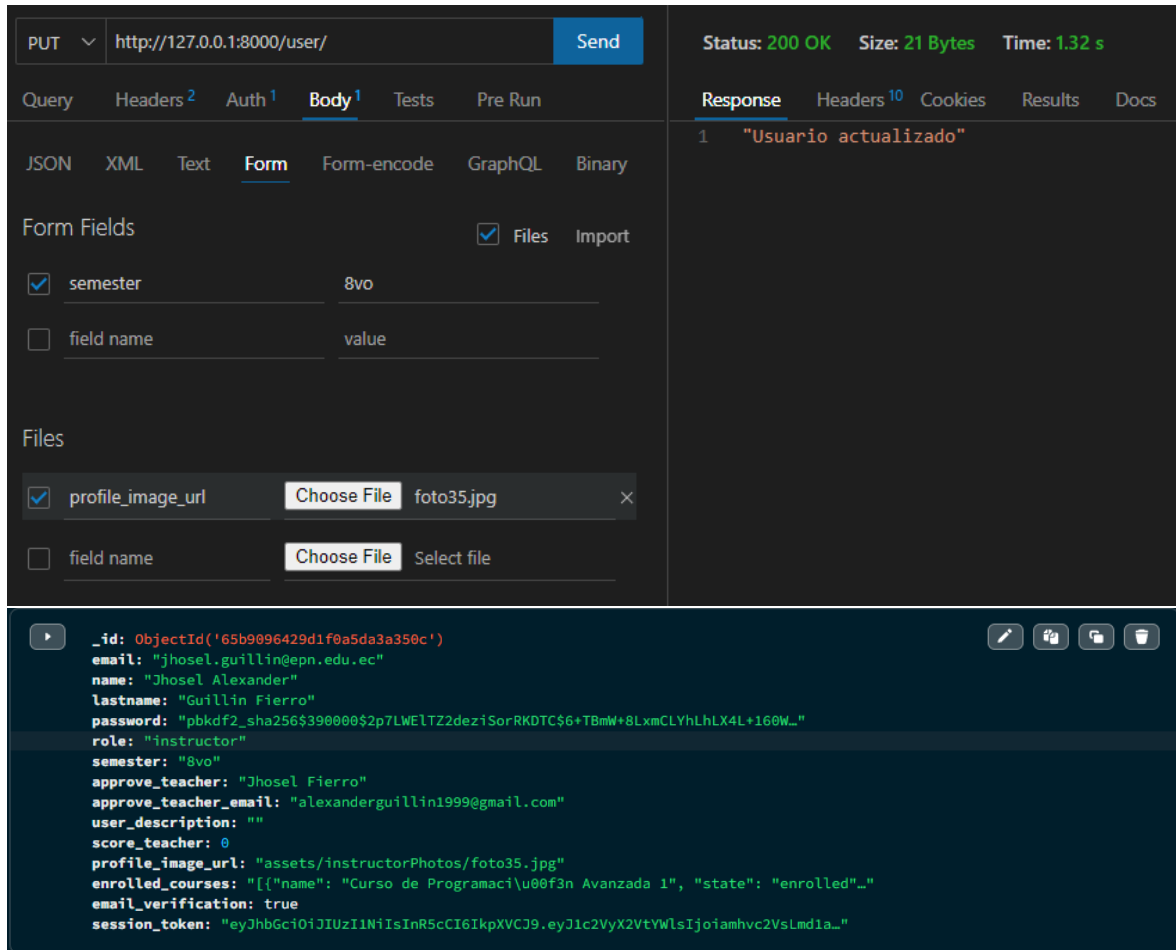


Figura 5.21 Actualización de semestre y foto de perfil del usuario

Para obtener un curso conociendo su id, se utiliza la siguiente configuración:

- **Método:** GET
- **Endpoint:** `course/<str:id>/`
- **Entrada:** id del curso
- **Salida:**
 - o **Estado:** 200 -> Obtiene la información de un curso en función del id (Figura 5.22), si el id = 0, se obtiene todos los cursos (Figura 5.23).
 - o **Estado:** 404 -> Curso no encontrado.

GET ⌵ http://127.0.0.1:8000/course/044bea7f-b5ed-4138-a1ea- Send

Status: 200 OK Size: 1.58 KB Time: 68 ms

Query Headers 2 Auth Body Tests Pre Run

Response Headers 10 Cookies Results Docs {}

Query Parameters

parameter value

```

1  {
2  "id": "044bea7f-b5ed-4138-a1ea-fac6f266819e",
3  "name": "Curso de C++",
4  "description": "Este es un curso enfocado al
5  aprendizaje de la programación en C++.",
6  "category": "Fundamentos de programación",
7  "instructor": "Jhosel Alexander Guillin Fierro",
8  "modules": [
9  {
10     "title": "Módulo 1",
11     "description": "Fundamentos",
12     "content": [
13       {
14         "module": "Módulo 1",
15         "title": "Historia y características
16         básicas",
17         "video_url": "https://s3.amazonaws.com
18         /polivirtual.me/assets/coursesVideos
19         /content/content_video_080fcd67-476c
20         -474b-9aa2-0d2ece12ff3c"
21       },
22       {
23         "module": "Módulo 1",
24         "title": "Instalación y configuración del
25         entorno de desarrollo",
26         "video_url": "https://s3.amazonaws.com
27         /polivirtual.me/assets/coursesVideos
28         /content/content_video_bad986a3-bf4e
29         -4937-9c99-1e8f35d33b1f"
30       }
31     ]
32   },
33   {
34     "title": "Módulo 2",
35     "description": "Fundamentos de la programación
36     en C++",
37     "content": [
38       {
39         "module": "Módulo 2",

```

Figura 5.22 Obtener un curso dado su id

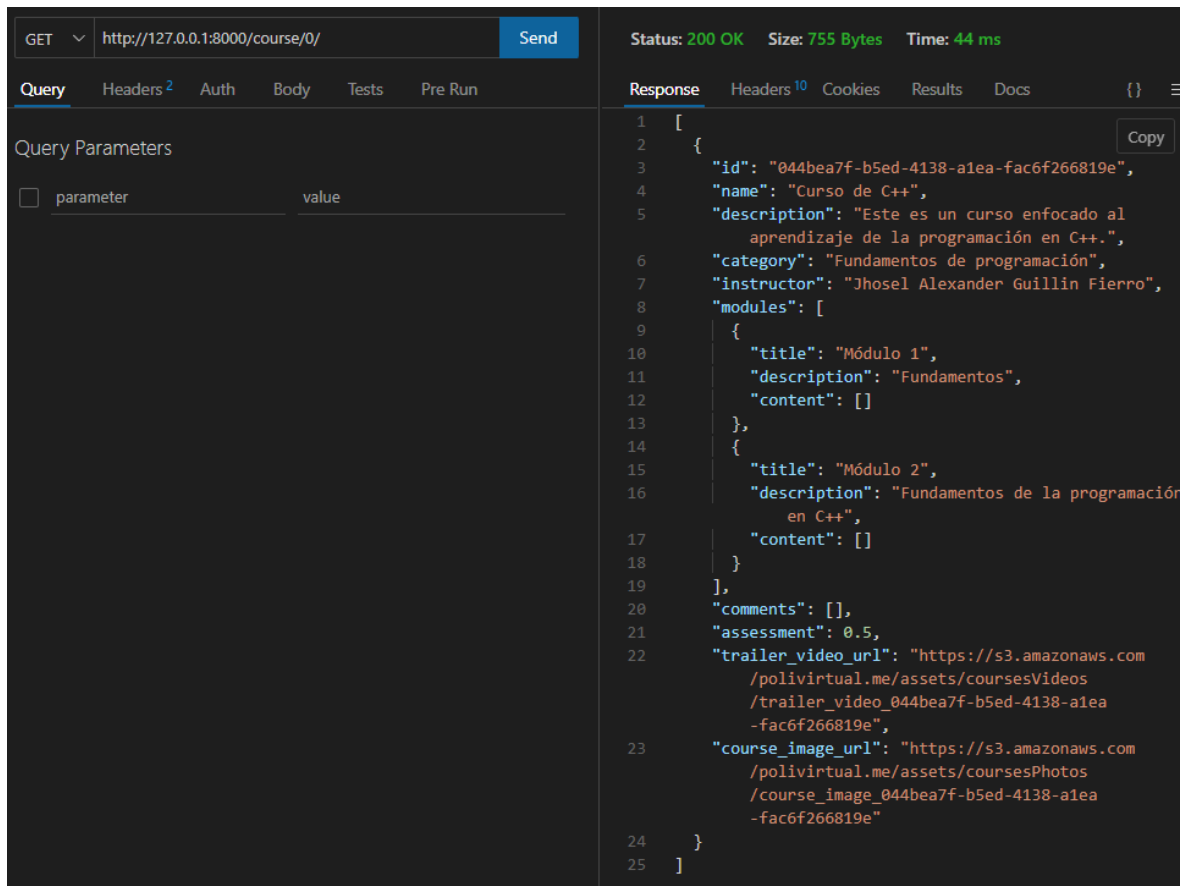


Figura 5.23 Obtener todos los cursos

Para eliminar un curso en función de su id, se utiliza la siguiente configuración:

- **Método:** DELETE
- **Endpoint:** *course/<str:id>/*
- **Entrada:** id del curso
- **Salida:**
 - o **Estado:** 200 -> Curso eliminado (Figura 5.24).
 - o **Estado:** 404 -> Curso no encontrado.

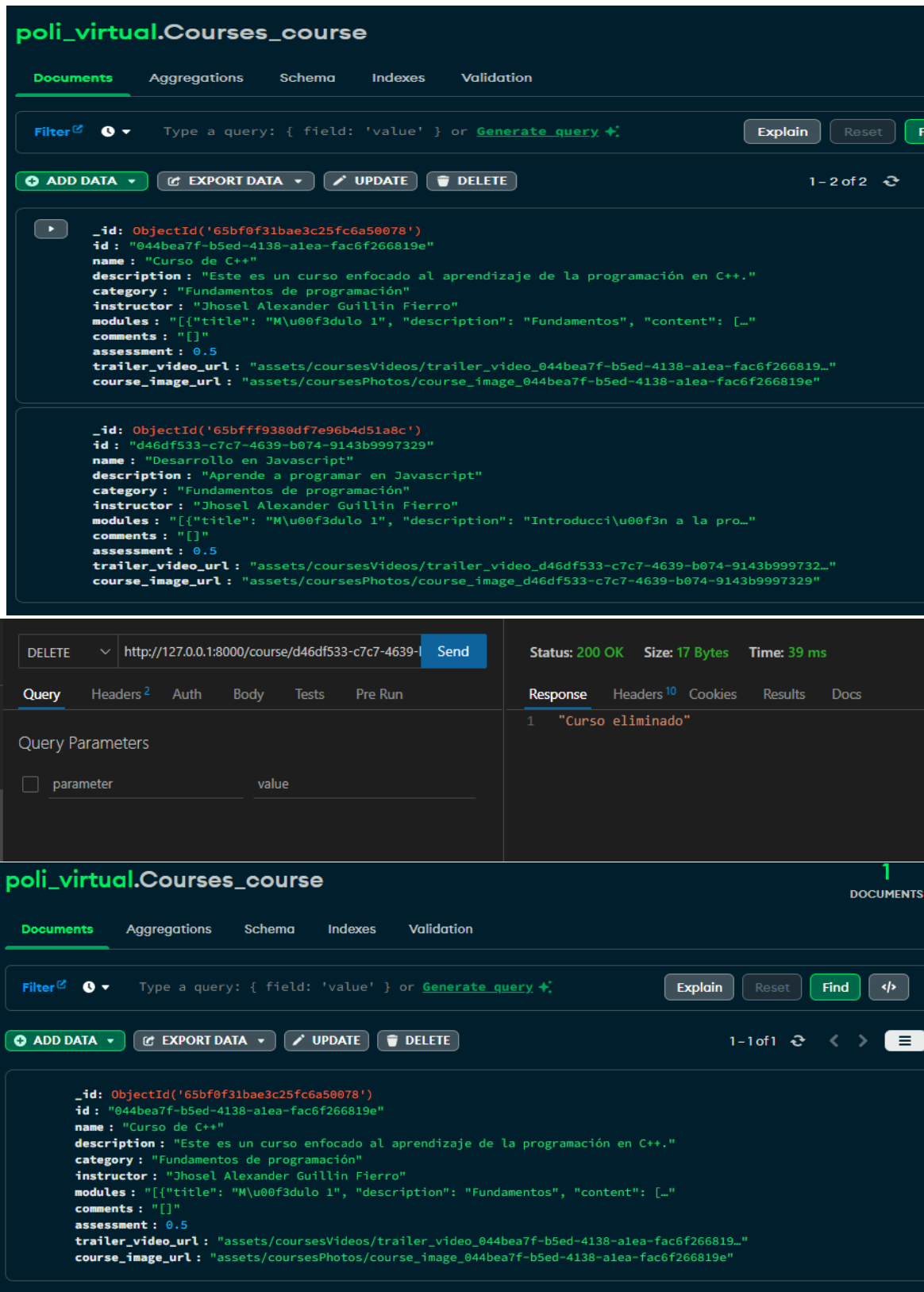


Figura 5.24 Eliminar curso dado su id

Para obtener los cursos con respecto a una categoría, se utiliza la siguiente configuración:

- **Método:** GET
- **Endpoint:** `course/courses-by-category/<str:category>/`
- **Entrada:** nombre de la categoría
- **Salida:**
 - o **Estado:** 200 -> Devuelve los cursos correspondientes a la categoría (Figura 5.25).
 - o **Estado:** 404 -> No hay cursos disponibles.

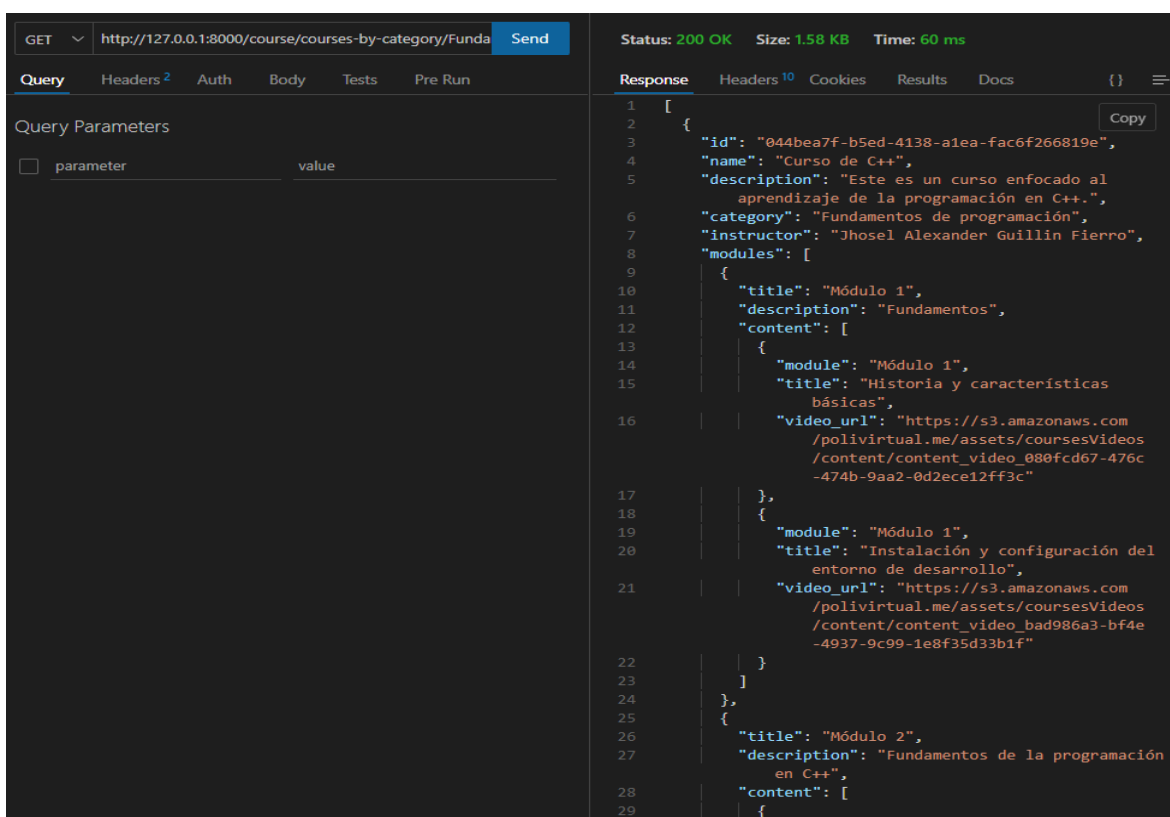


Figura 5.25 Obtener cursos en función de la categoría

Para obtener los cursos destacados, se utiliza la siguiente configuración:

- **Método:** GET
- **Endpoint:** `course/featured-courses/`
- **Entrada:** Ninguna
- **Salida:**

- **Estado:** 200 -> Retorna los cursos destacados (Figura 5.26).
- **Estado:** 404 -> No hay cursos destacados o disponibles.

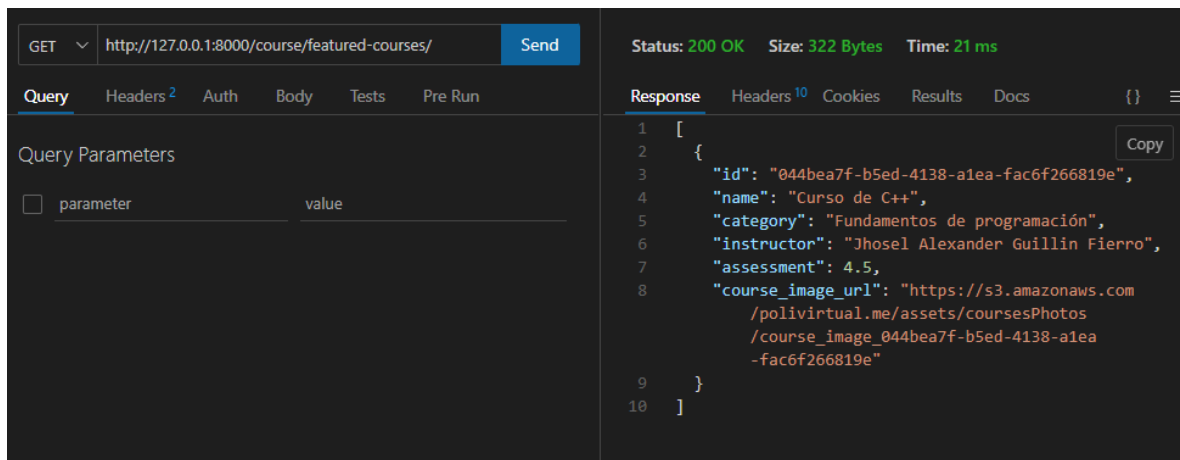


Figura 5.26 Obtener cursos destacados

Para obtener los cursos en función del instructor, se utiliza la siguiente configuración:

- **Método:** GET
- **Endpoint:** `course/courses-by-instructor/<str:instructor_name>/`
- **Entrada:** nombre del instructor
- **Salida:**
 - **Estado:** 200 -> Retorna los cursos del instructor (Figura 5.27).
 - **Estado:** 404 -> No hay cursos disponibles.

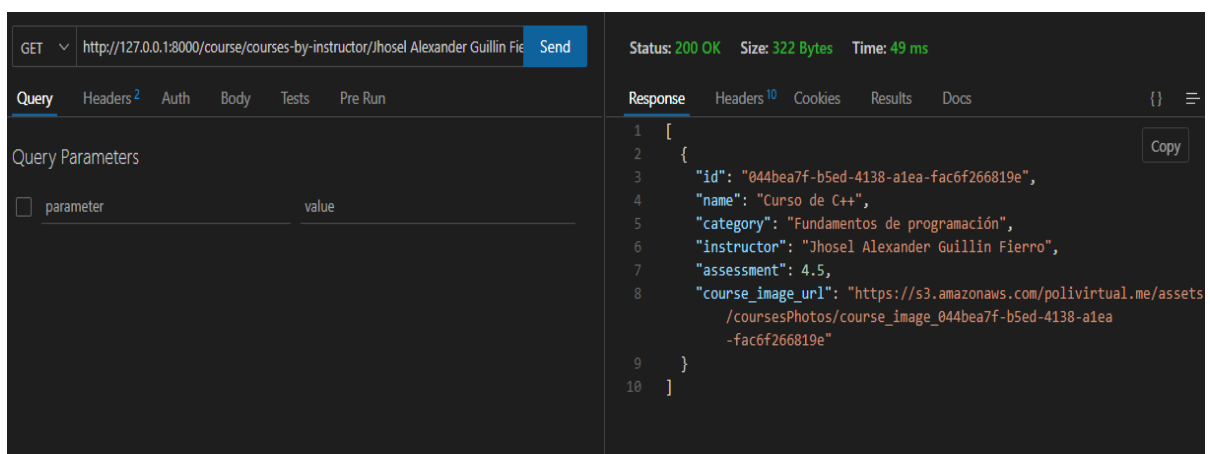


Figura 5.27 Obtener cursos en función del instructor

Para crear una nueva categoría, se utiliza la siguiente configuración:

- **Método:** POST
- **Endpoint:** `category/`
- **Entrada:** nombre y descripción de la categoría
- **Salida:**
 - o **Estado:** 200 -> Categoría agregada (Figura 5.28).
 - o **Estado:** 404 -> Error al guardar la categoría, revisar los campos que se envían.

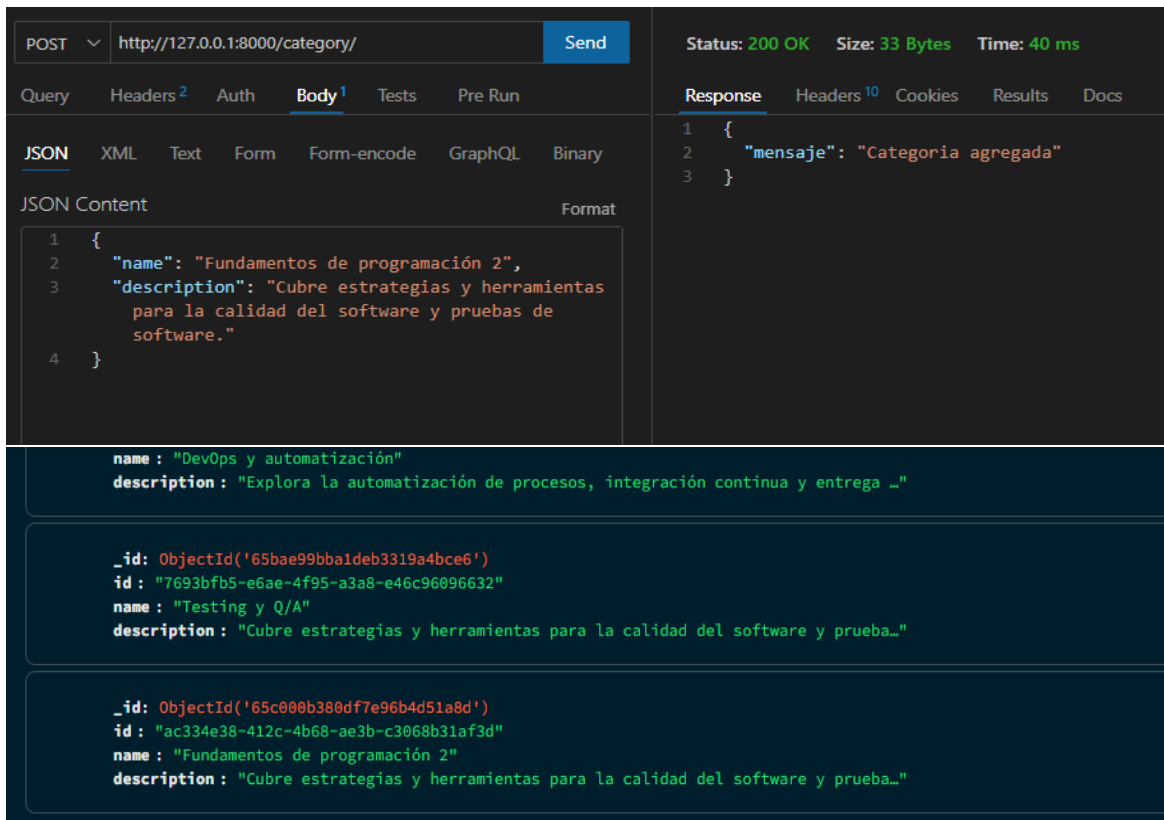


Figura 5.28 Agregar nueva categoría

Para obtener una categoría conociendo su id, se utiliza la siguiente configuración:

- **Método:** GET
- **Endpoint:** `category/<str:id>/`

- **Entrada:** id de la categoría
- **Salida:**
 - o **Estado:** 200 -> Si el id = 0, retorna todas las categorías (Figura 5.30), caso contrario retorna la categoría respectiva al id (Figura 5.29).
 - o **Estado:** 404 -> Categoría no encontrada.

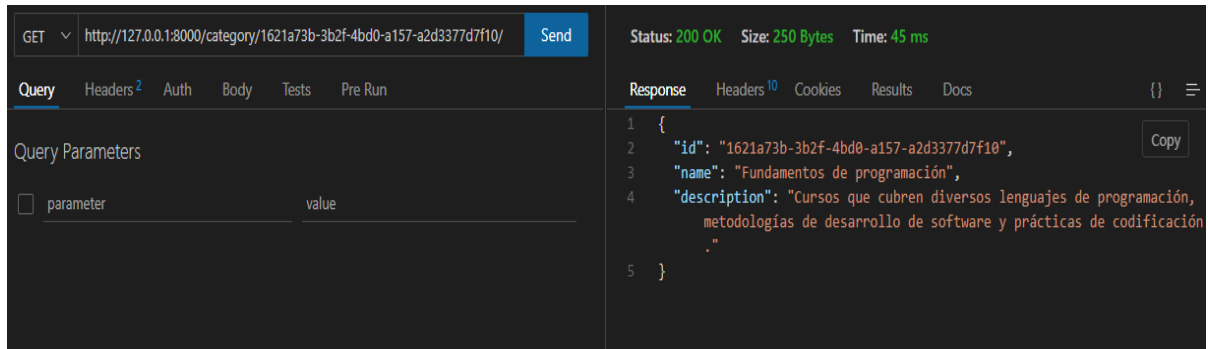


Figura 5.29 Obtener categoría por id

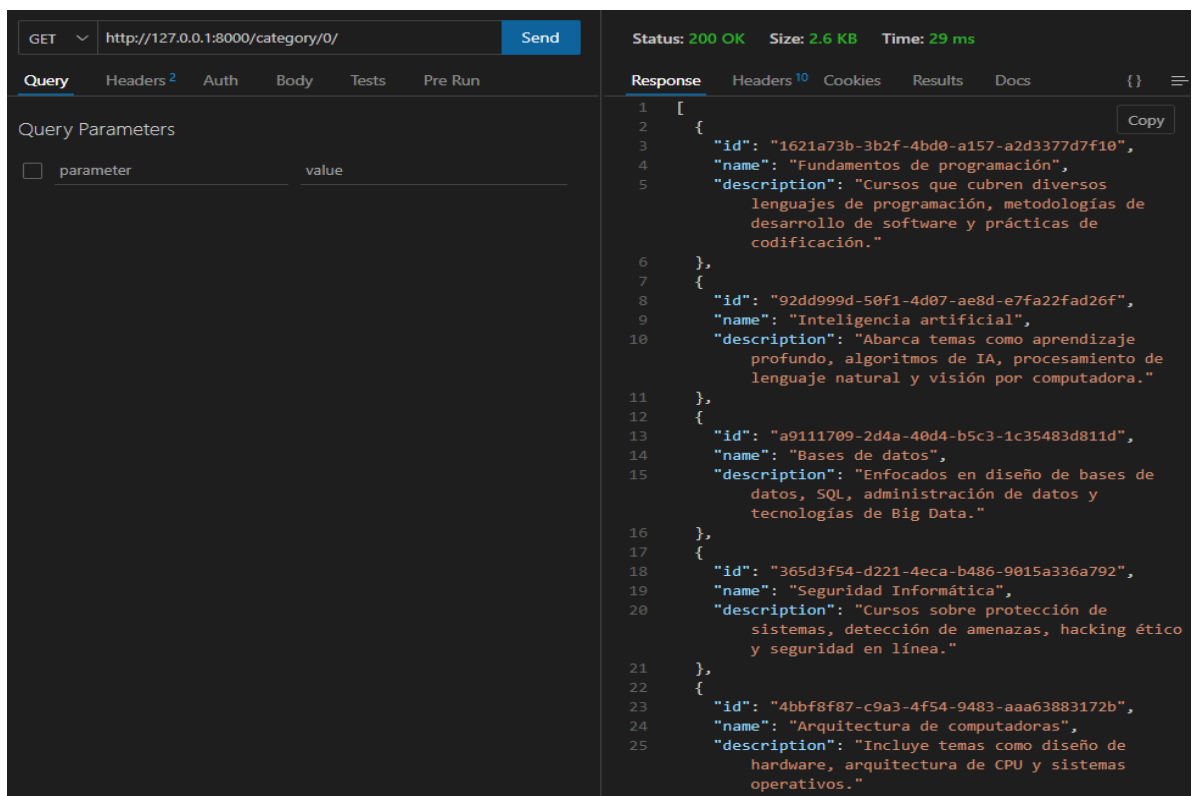


Figura 5.30 Obtener todas las categorías

Para actualizar una categoría, se utiliza la siguiente configuración:

- **Método:** PUT
- **Endpoint:** *category/*
- **Entrada:** id, nombre y descripción de la categoría.
- **Salida:**
 - o **Estado:** 200 -> Categoría actualizada (Figura 5.31).
 - o **Estado:** 404 -> Categoría no encontrada.

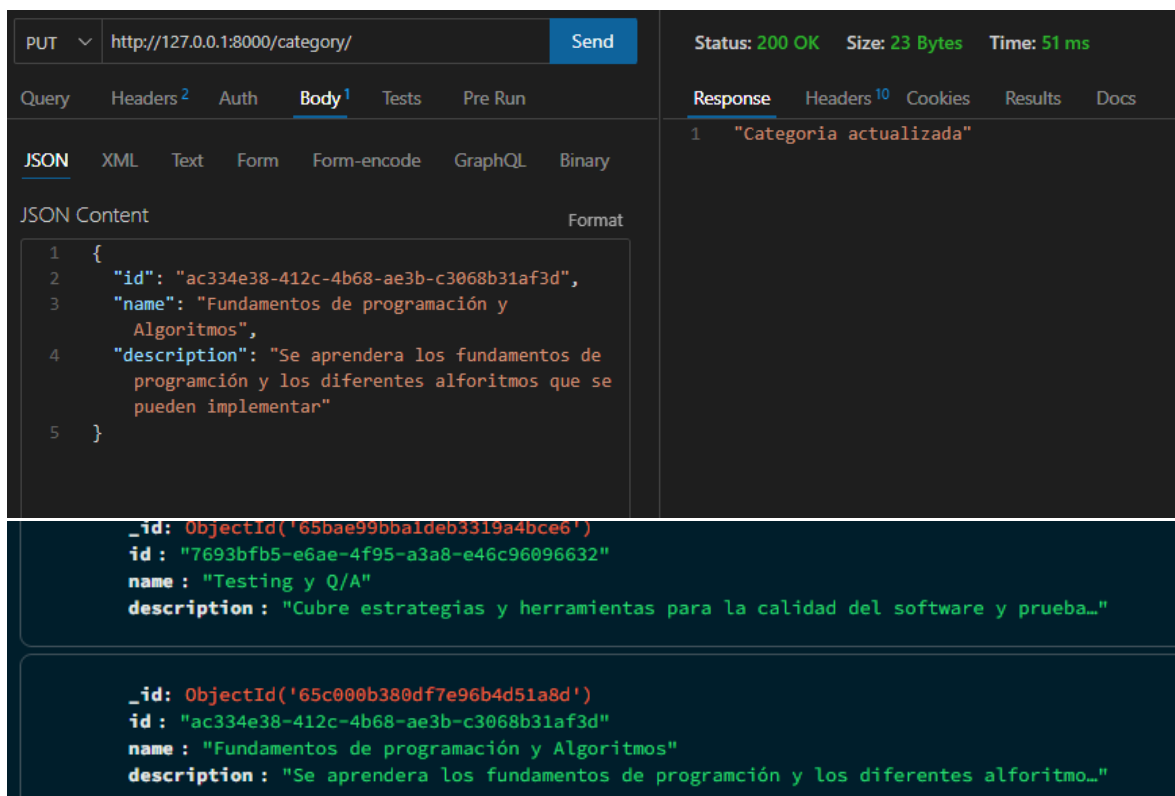


Figura 5.31 Actualizar una categoría

Para eliminar una categoría en función de su id, se utiliza la siguiente configuración:

- **Método:** DELETE
- **Endpoint:** *category/<str:id>/*
- **Entrada:** id de la categoría
- **Salida:**
 - o **Estado:** 200 -> Categoría eliminada (Figura 5.32).

- **Estado:** 404 -> Categoría no encontrada.

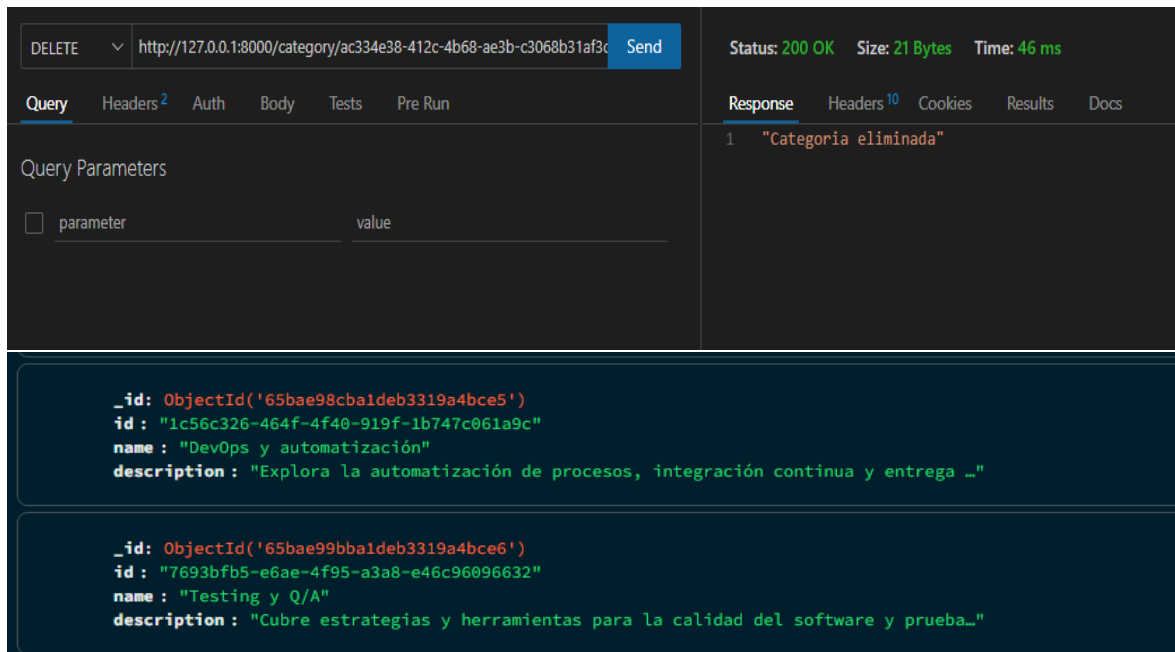


Figura 5.32 Eliminar categoría por id

Para obtener el id de una categoría en función de su nombre, se utiliza la siguiente configuración:

- **Método:** GET
- **Endpoint:** `category/get_category_id/<str:category_name>/`
- **Entrada:** nombre de la categoría
- **Salida:**
 - **Estado:** 200 -> Retorna el id de la categoría (Figura 5.33).
 - **Estado:** 404 -> Categoría no encontrada.

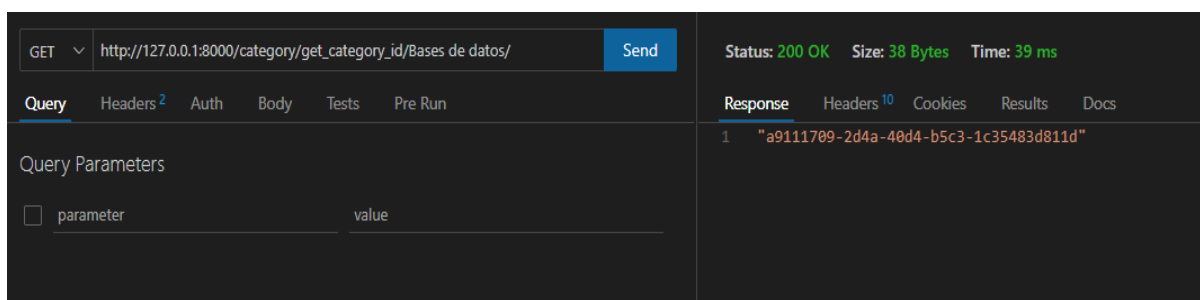


Figura 5.33 Obtener id de la categoría en función de su nombre

Para agregar el contenido de un curso, se utiliza la siguiente configuración:

- **Método:** POST
- **Endpoint:** content/
- **Entrada:** nombre del curso, nombre del módulo, título y video del tema correspondiente al módulo.
- **Salida:**
 - o **Estado:** 200 -> Contenido agregado (Figura 5.34).
 - o **Estado:** 404 -> Revisar campos y contenido de entrada.

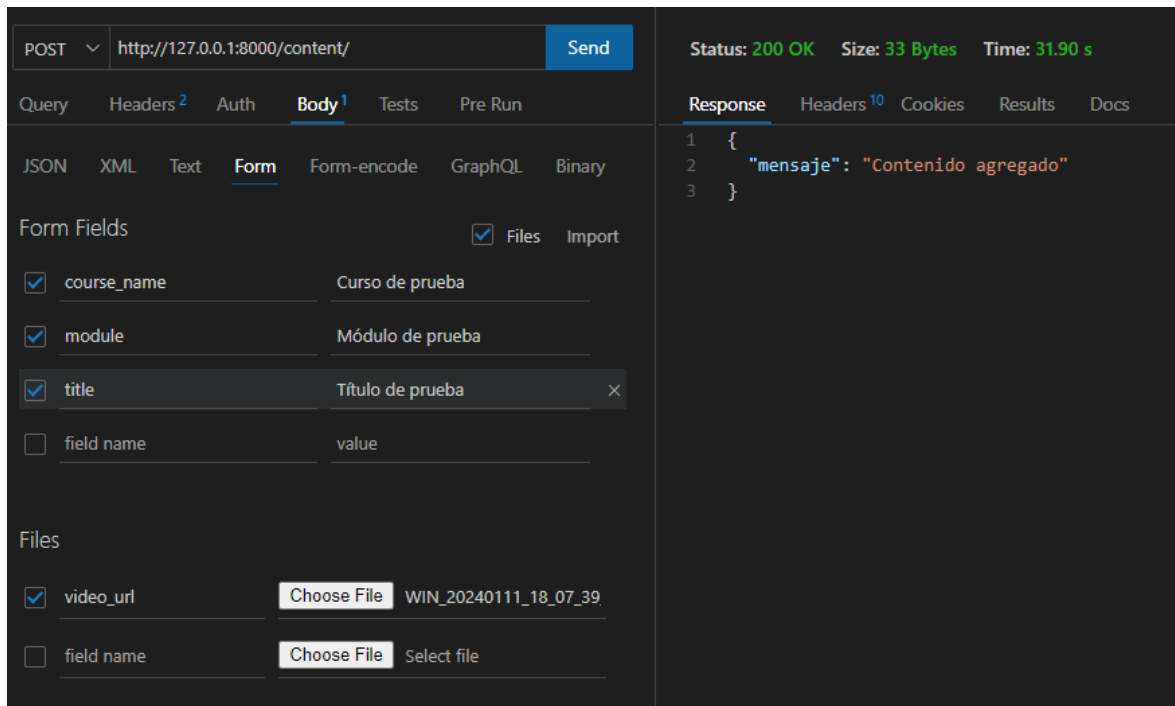


Figura 5.34 Agregar contenido de un curso

ANEXO VI. Tableros Kanban – Flujo de tareas

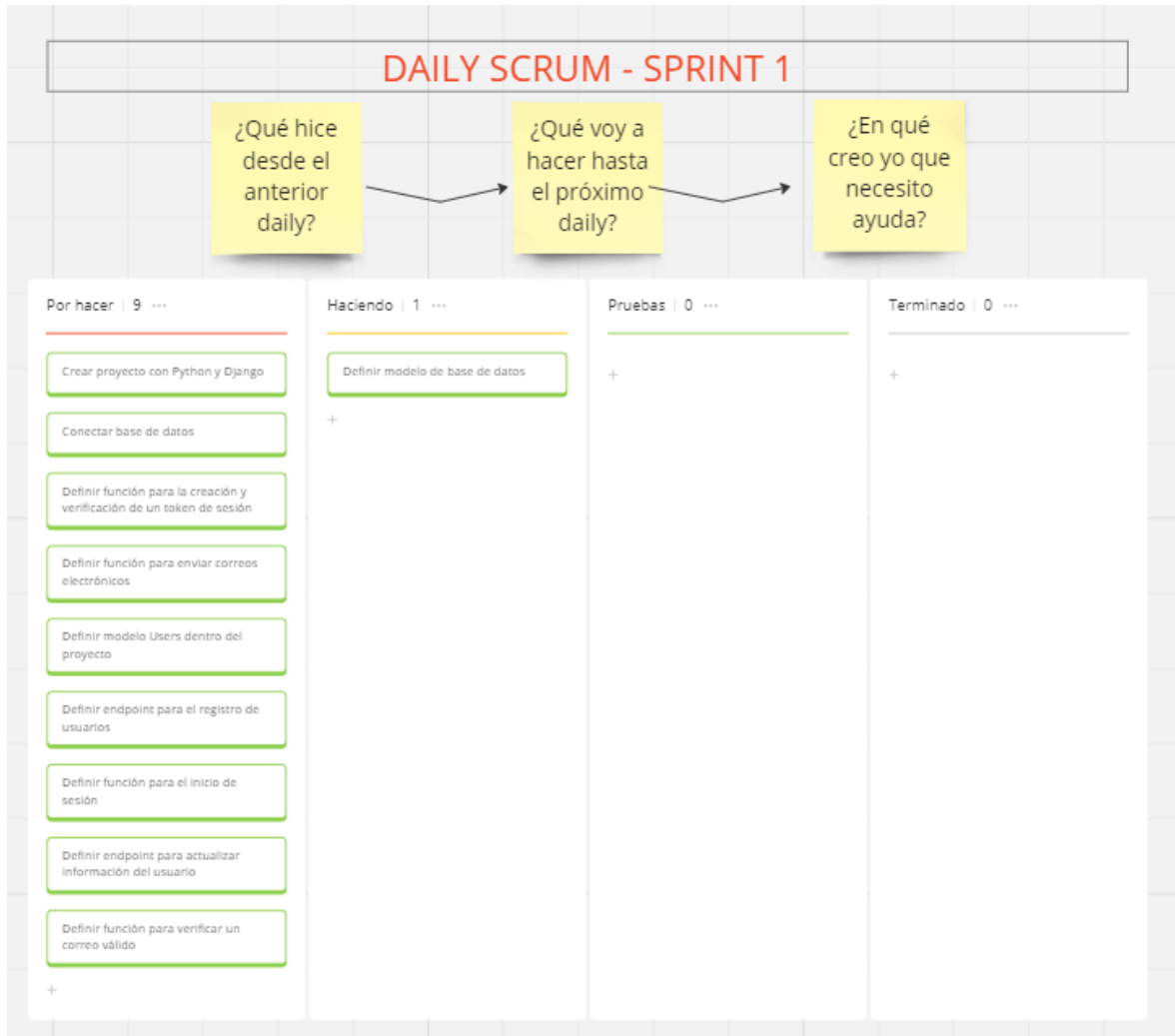


Figura 5.35 Tablero Kanban Sprint 1 - 20/10/2023 al 21/10/2023

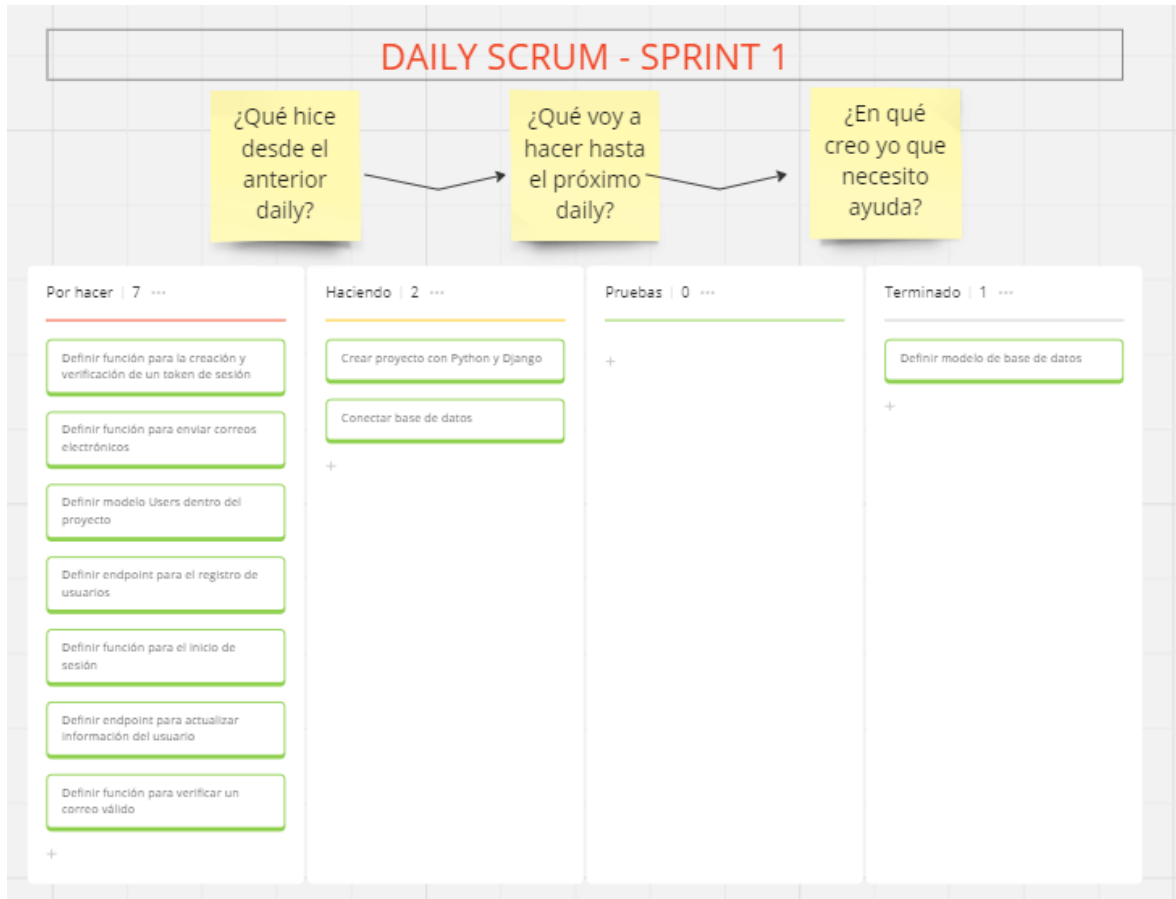


Figura 5.36 Tablero Kanban Sprint 1 - 23/10/2023 al 25/10/2023

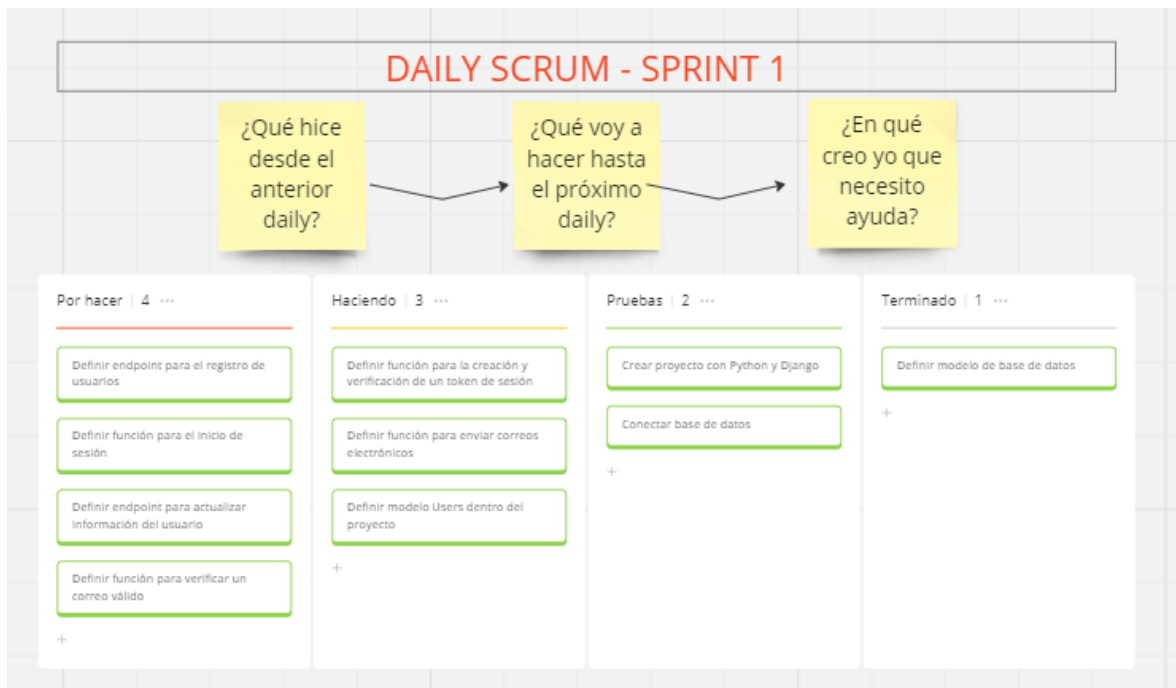


Figura 5.37 Tablero Kanban Sprint 1 - 26/10/2023 al 31/10/2023

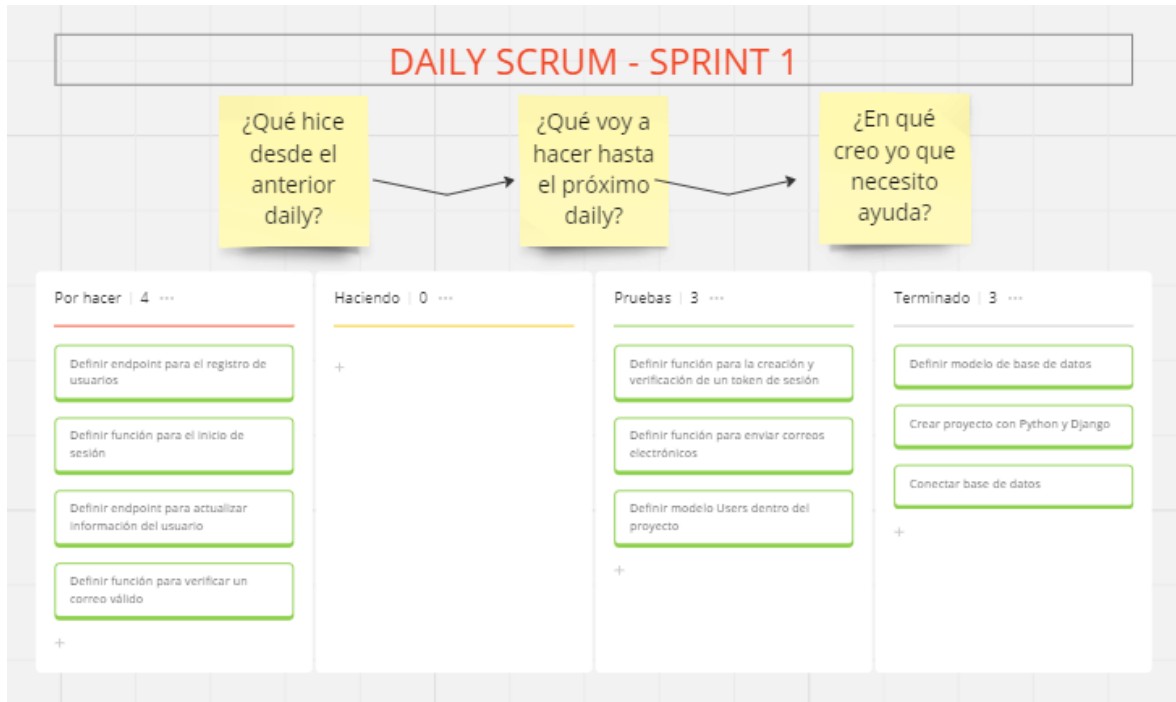


Figura 5.38 Tablero Kanban Sprint 1 - 01/11/2023 al 03/11/2023

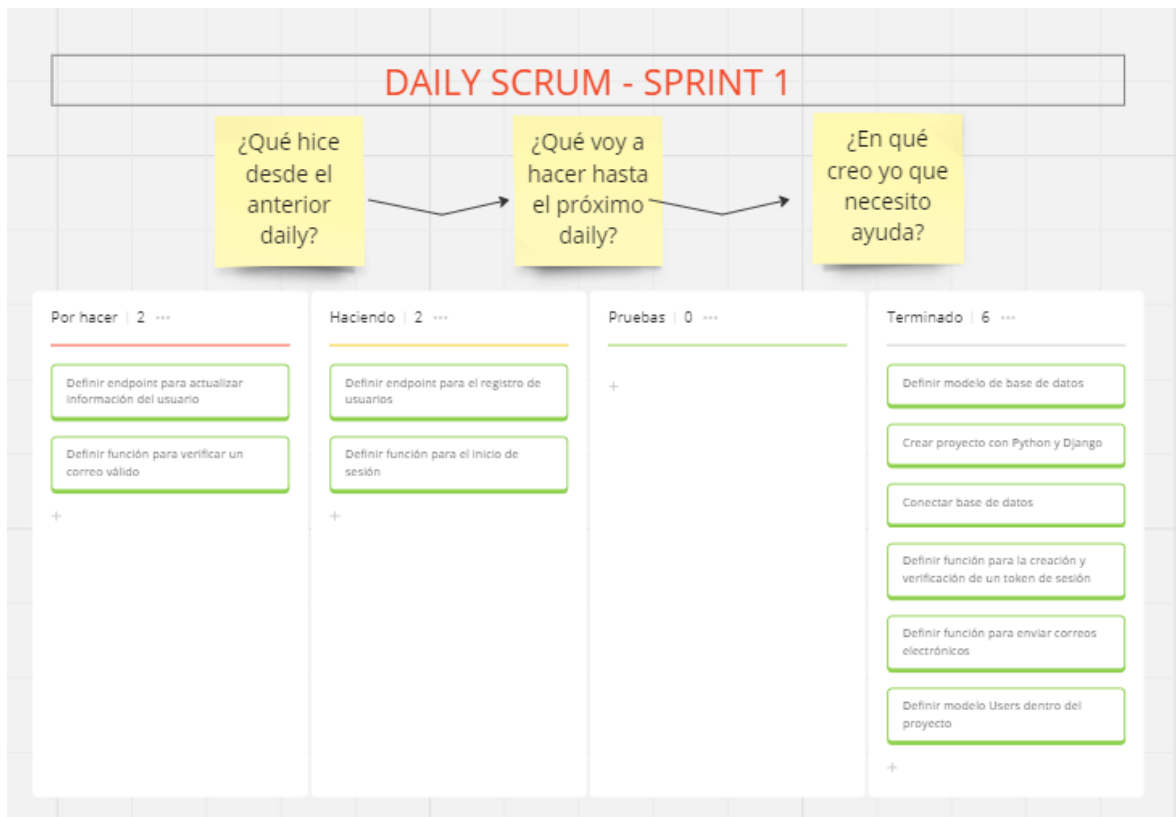


Figura 5.39 Tablero Kanban Sprint 1 - 04/11/2023 al 06/11/2023

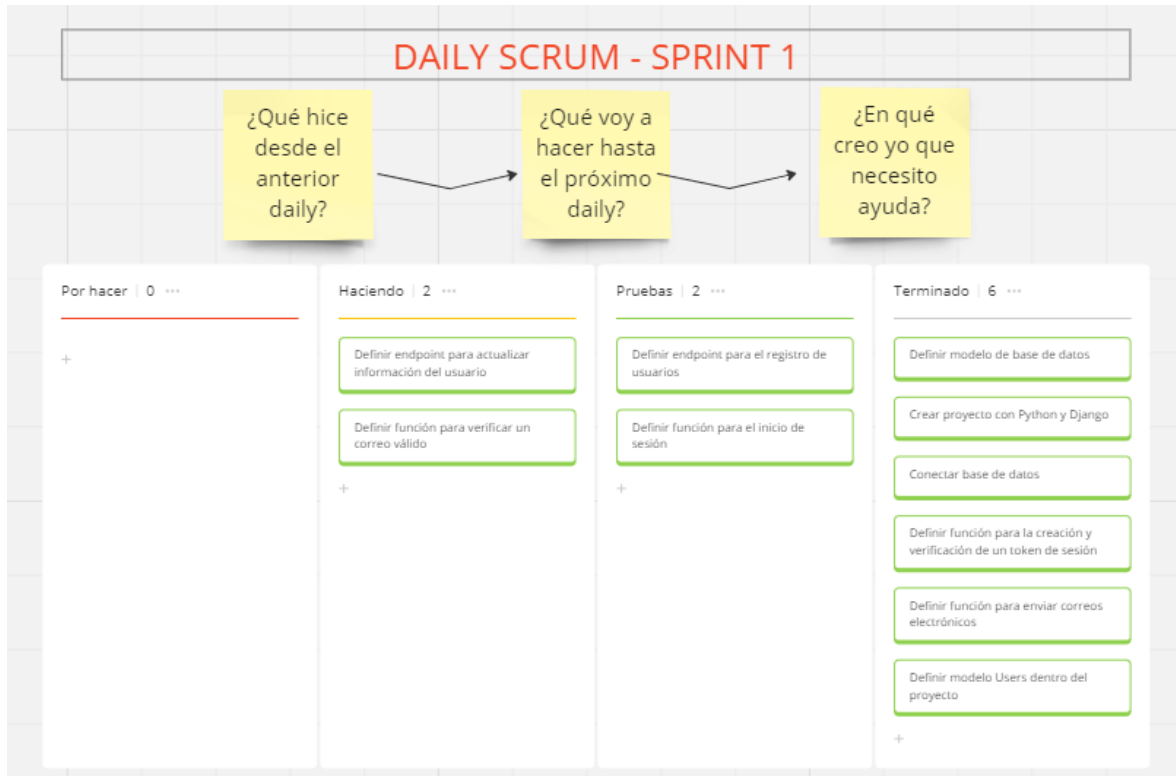


Figura 5.40 Tablero Kanban Sprint 1 - 07/11/2023 al 08/11/2023

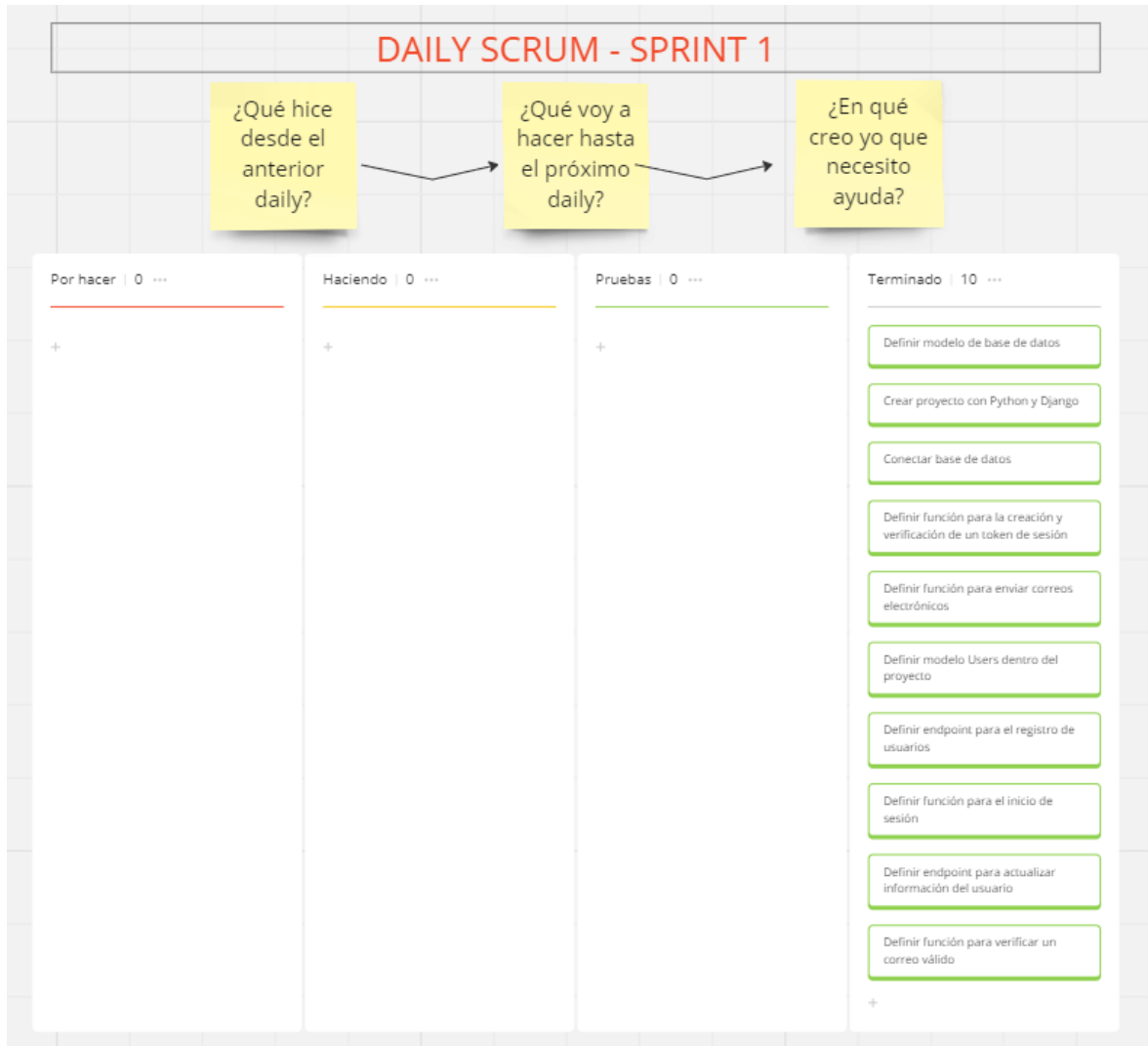


Figura 5.41 Tablero Kanban Sprint 1 - 09/11/2023 al 10/11/2023 – Finalización Sprint 1

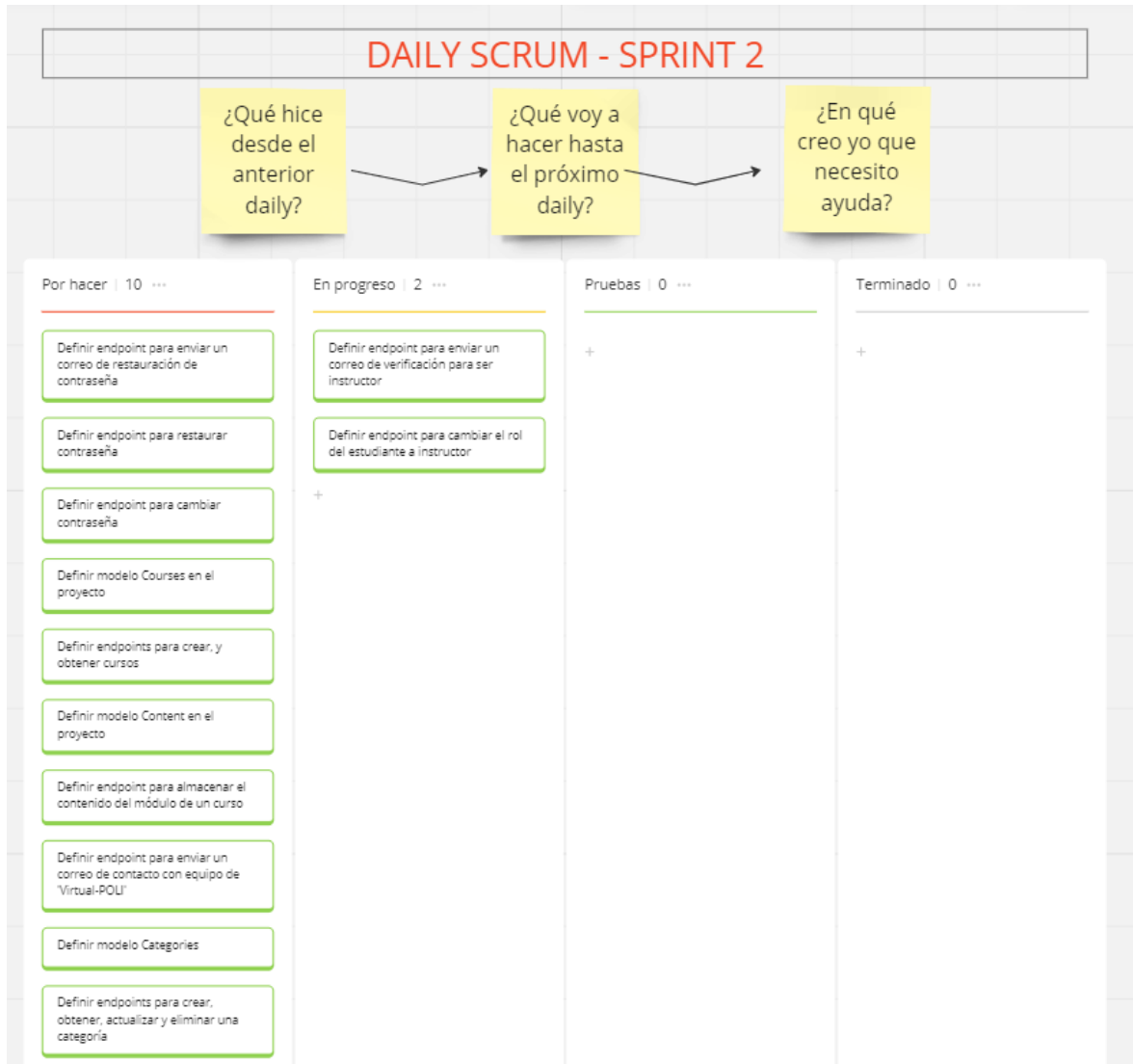


Figura 5.42 Tablero Kanban Sprint 2 - 11/11/2023 al 14/11/2023

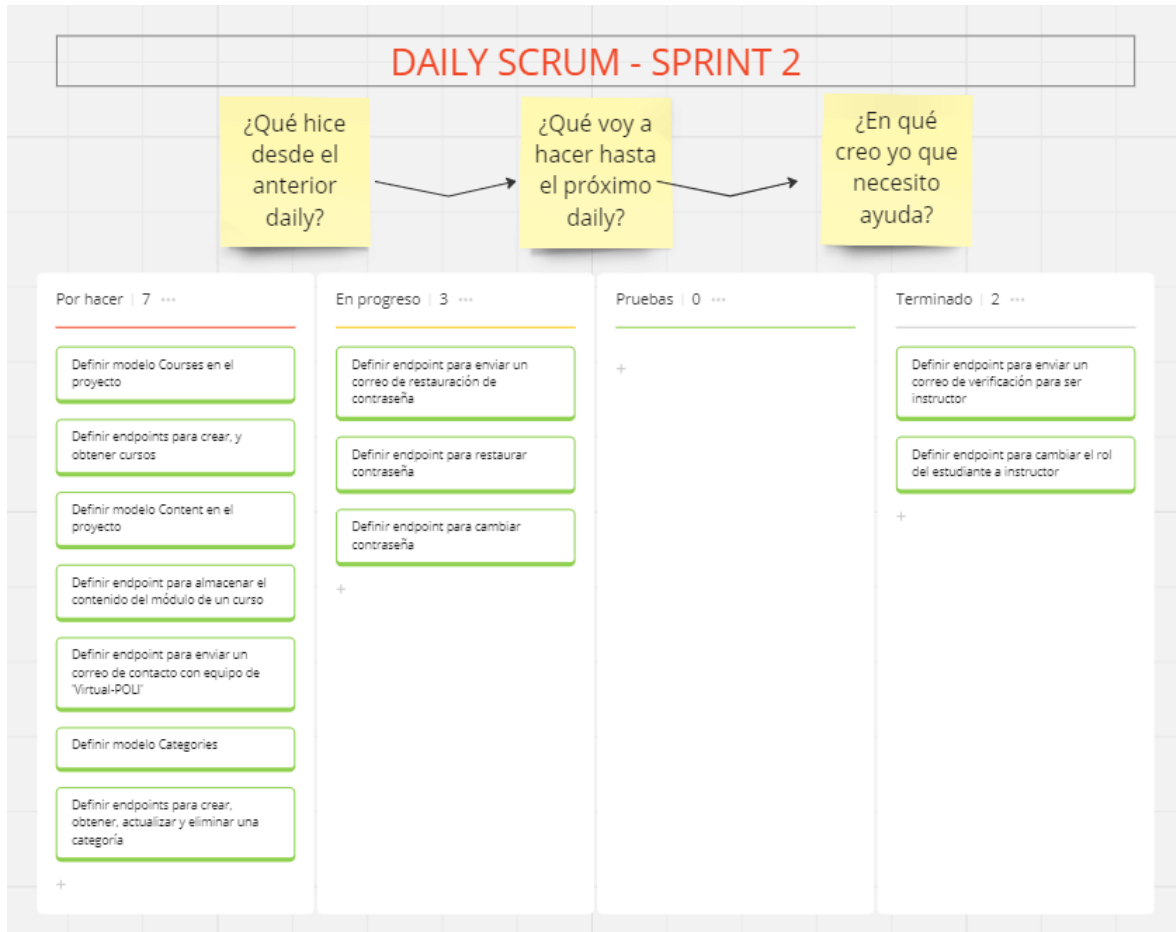


Figura 5.43 Tablero Kanban Sprint 2 - 15/11/2023 al 18/11/2023

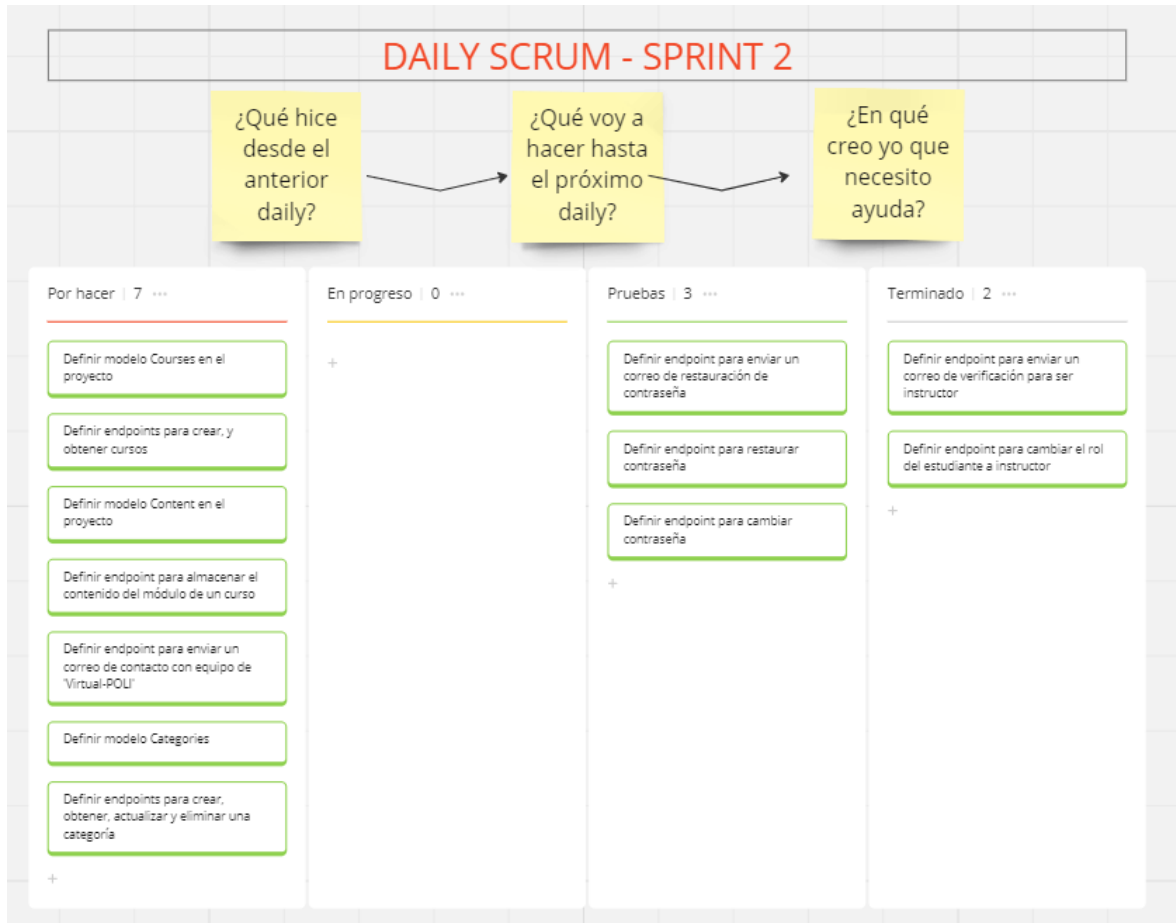


Figura 5.44 Tablero Kanban Sprint 2 - 19/11/2023

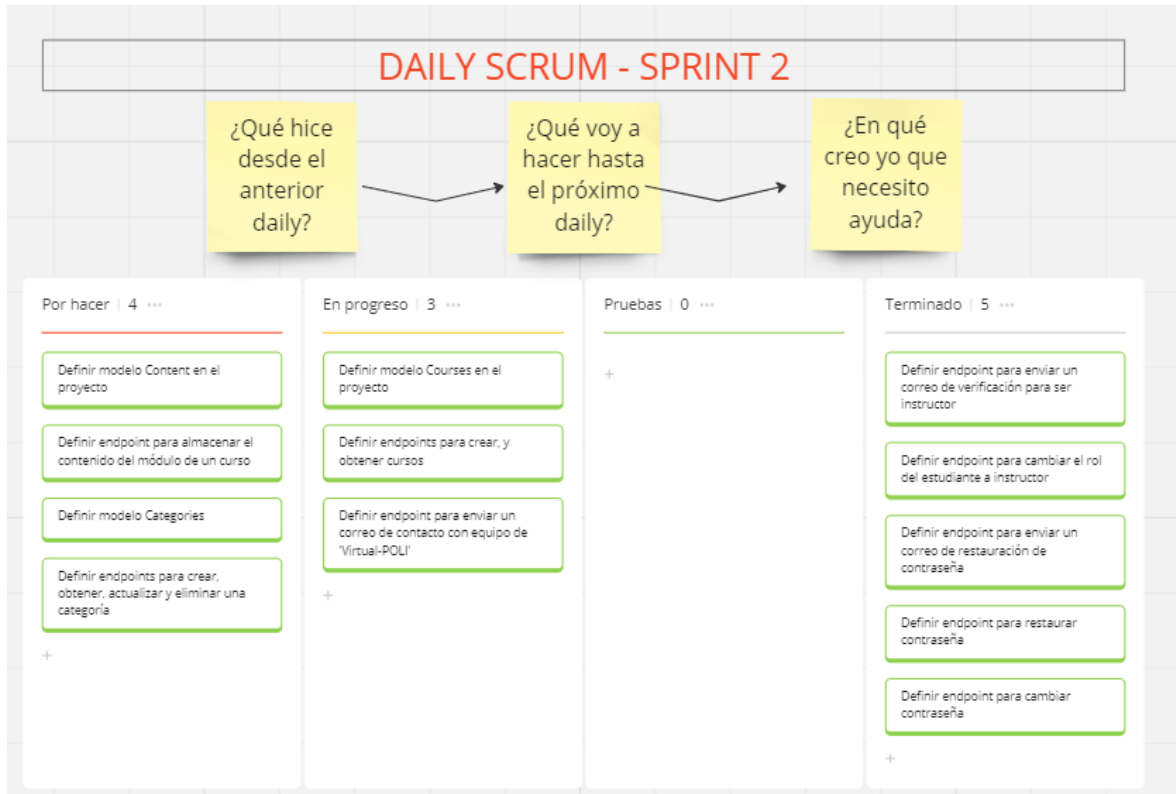


Figura 5.45 Tablero Kanban Sprint 2 - 20/11/2023 al 24/11/2023

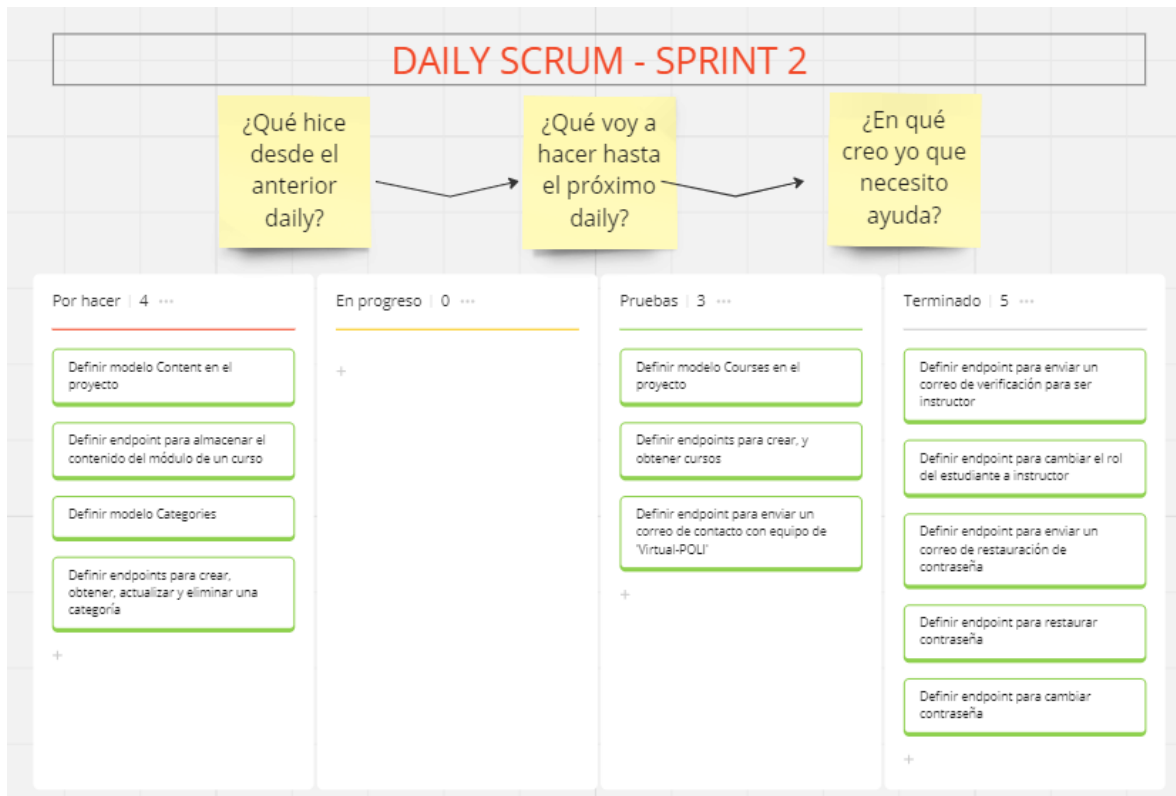


Figura 5.46 Tablero Kanban Sprint 2 - 25/11/2023

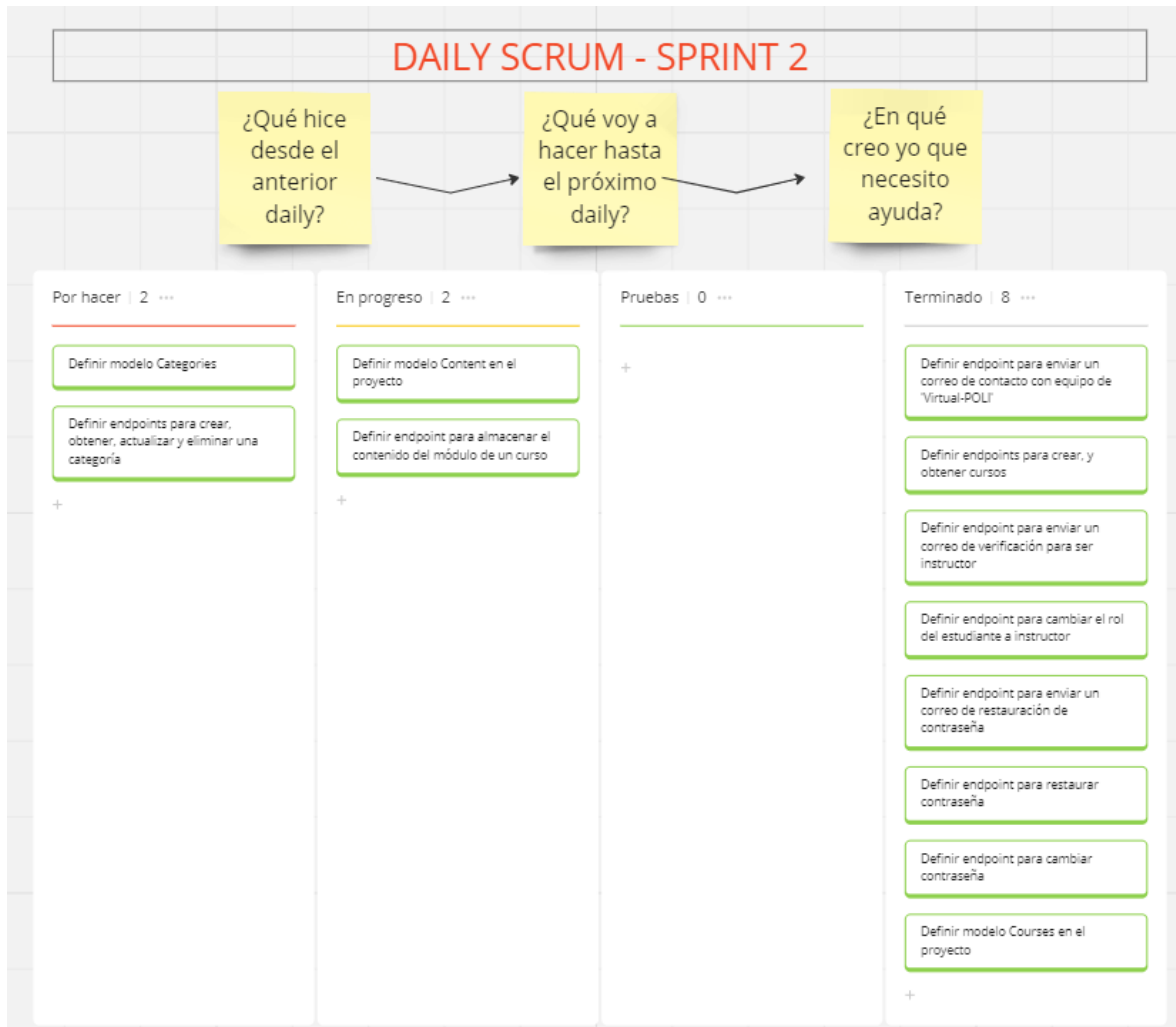


Figura 5.47 Tablero Kanban Sprint 2 - 26/11/2023 al 28/11/2023

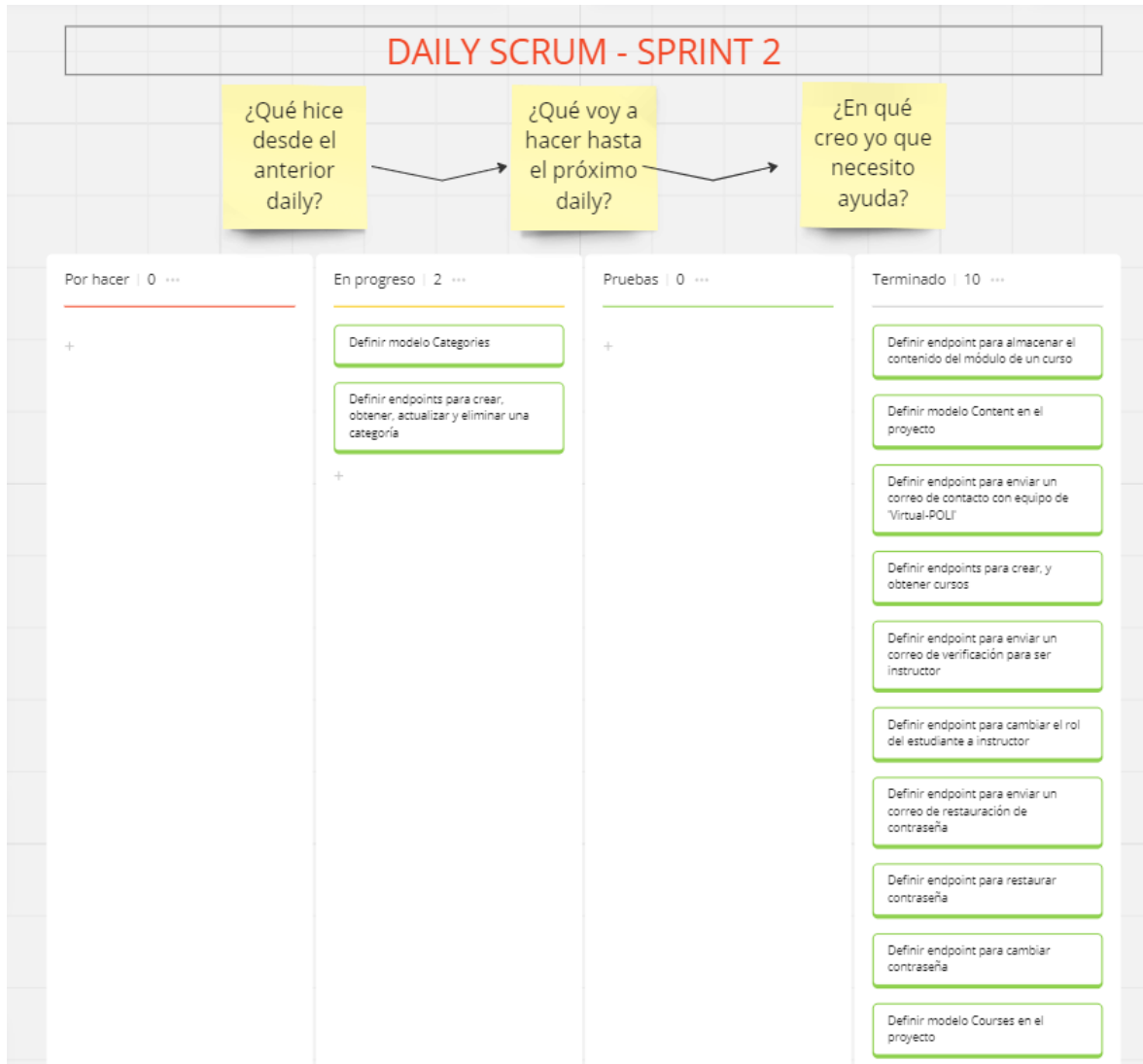


Figura 5.48 Tablero Kanban Sprint 2 - 29/11/2023 al 30/11/2023



Figura 5.49 Tablero Kanban Sprint 2 - 01/12/2023 – Finalización Sprint 2

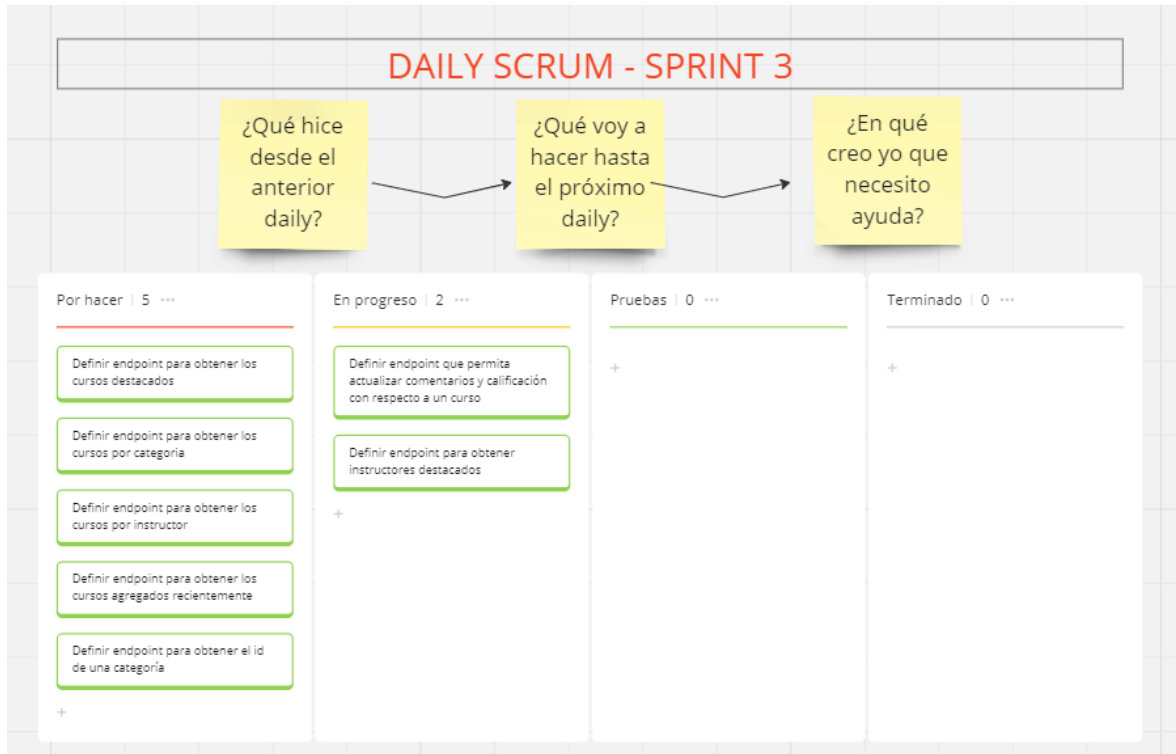


Figura 5.50 Tablero Kanban Sprint 3 - 02/12/2023 al 05/12/2023

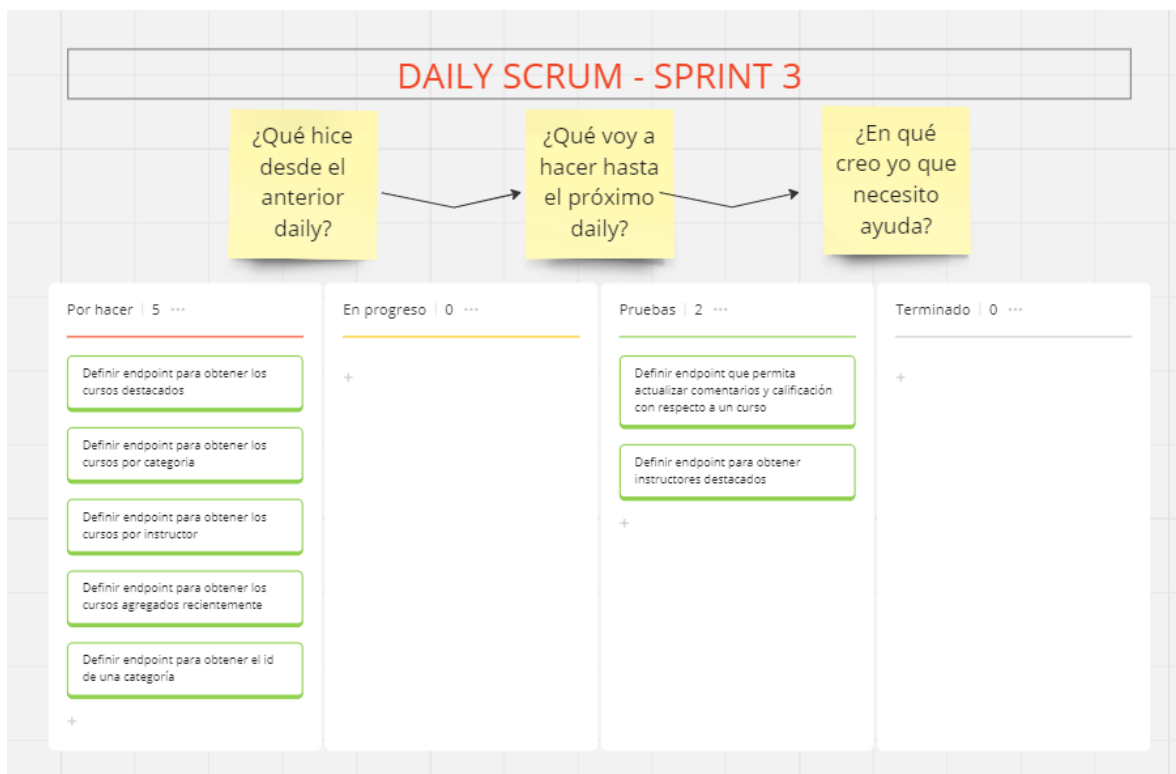


Figura 5.51 Tablero Kanban Sprint 3 - 06/12/2023 al 07/12/2023

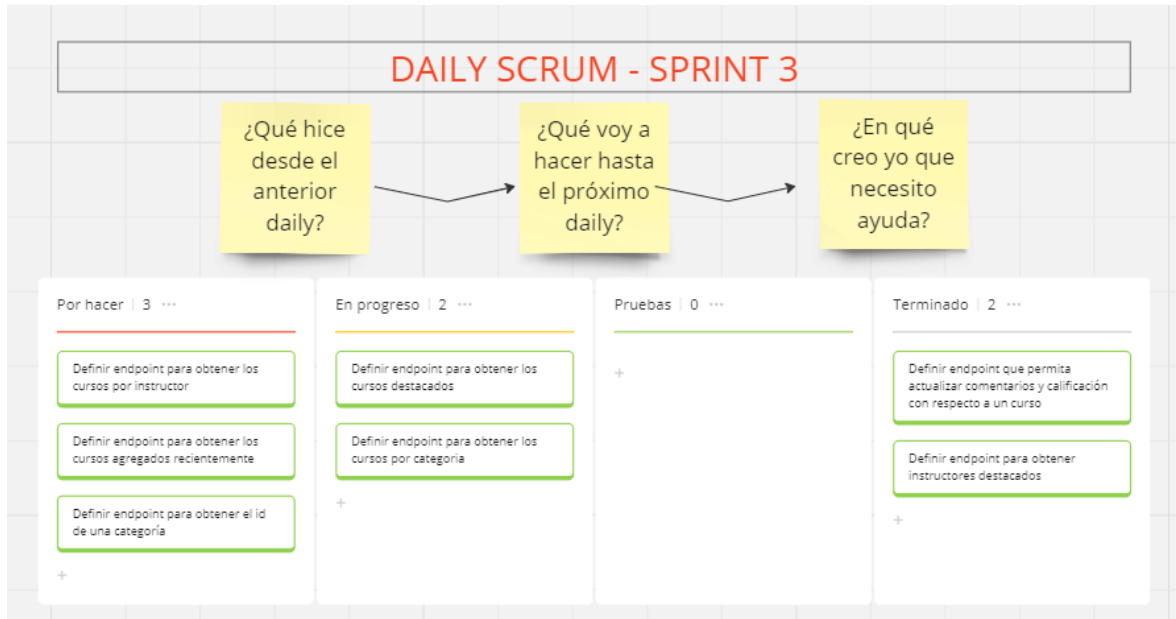


Figura 5.52 Tablero Kanban Sprint 3 - 08/12/2023 al 12/12/2023

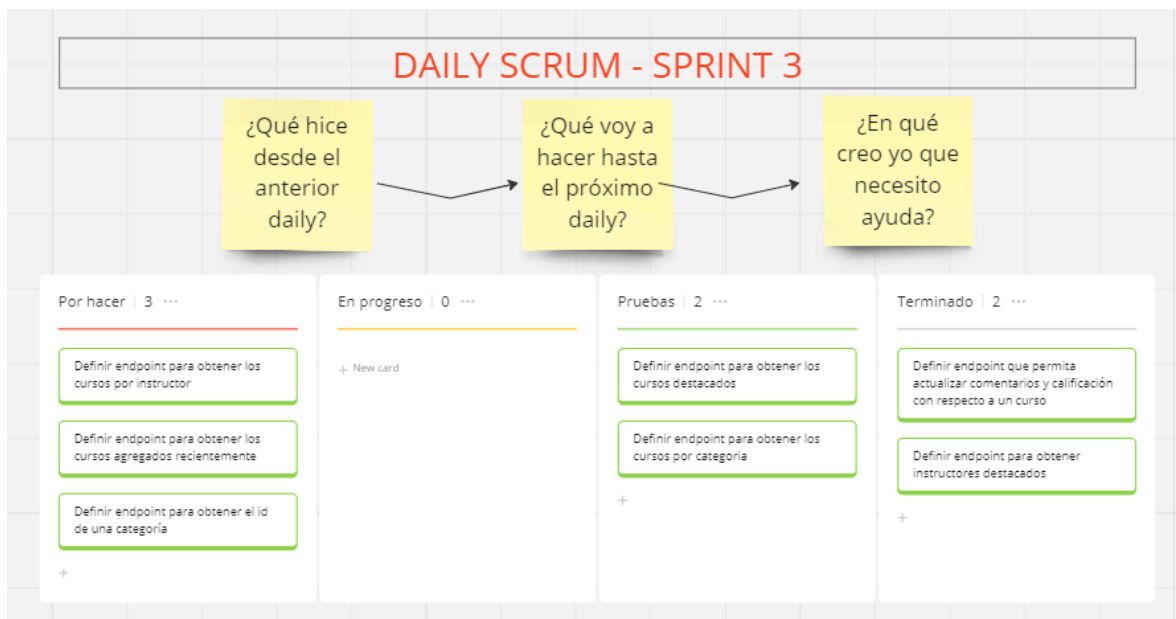


Figura 5.53 Tablero Kanban Sprint 3 - 13/12/2023 al 14/12/2023

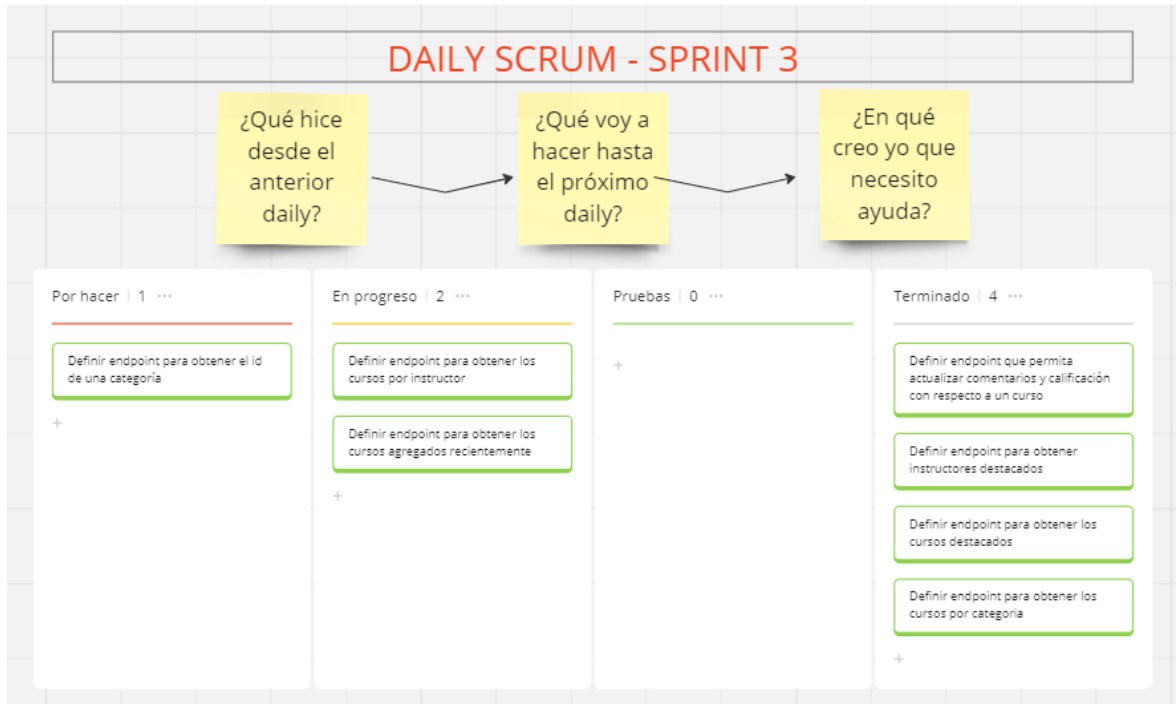


Figura 5.54 Tablero Kanban Sprint 3 - 15/12/2023 al 16/12/2023

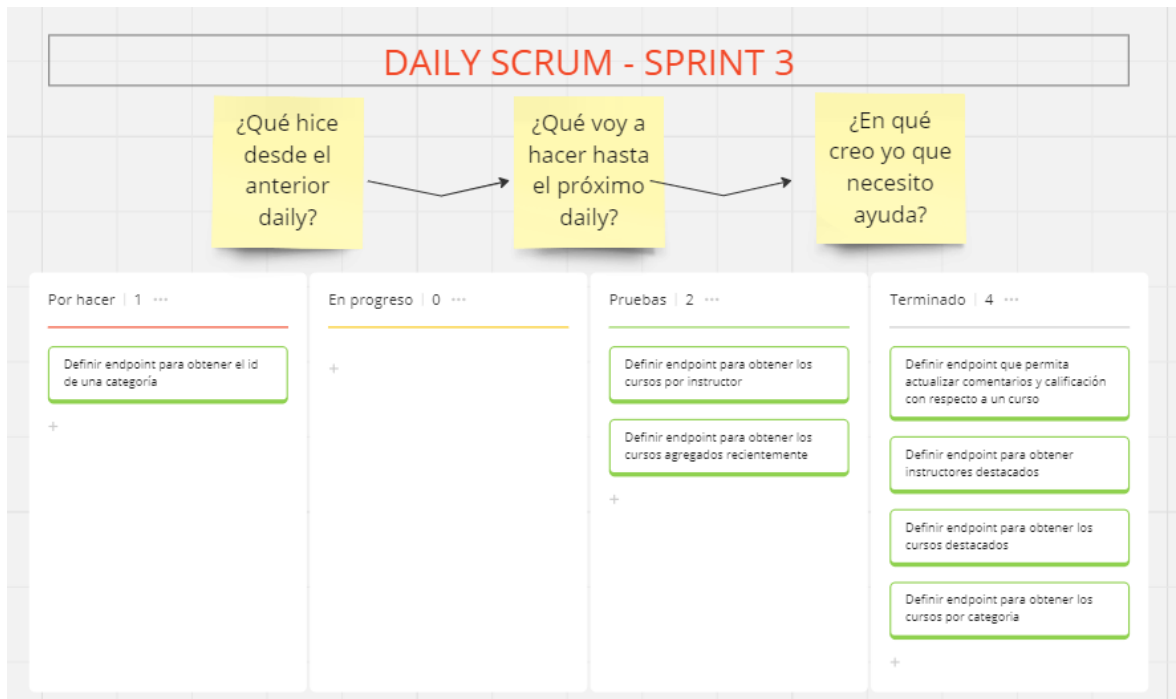


Figura 5.55 Tablero Kanban Sprint 3 - 17/12/2023

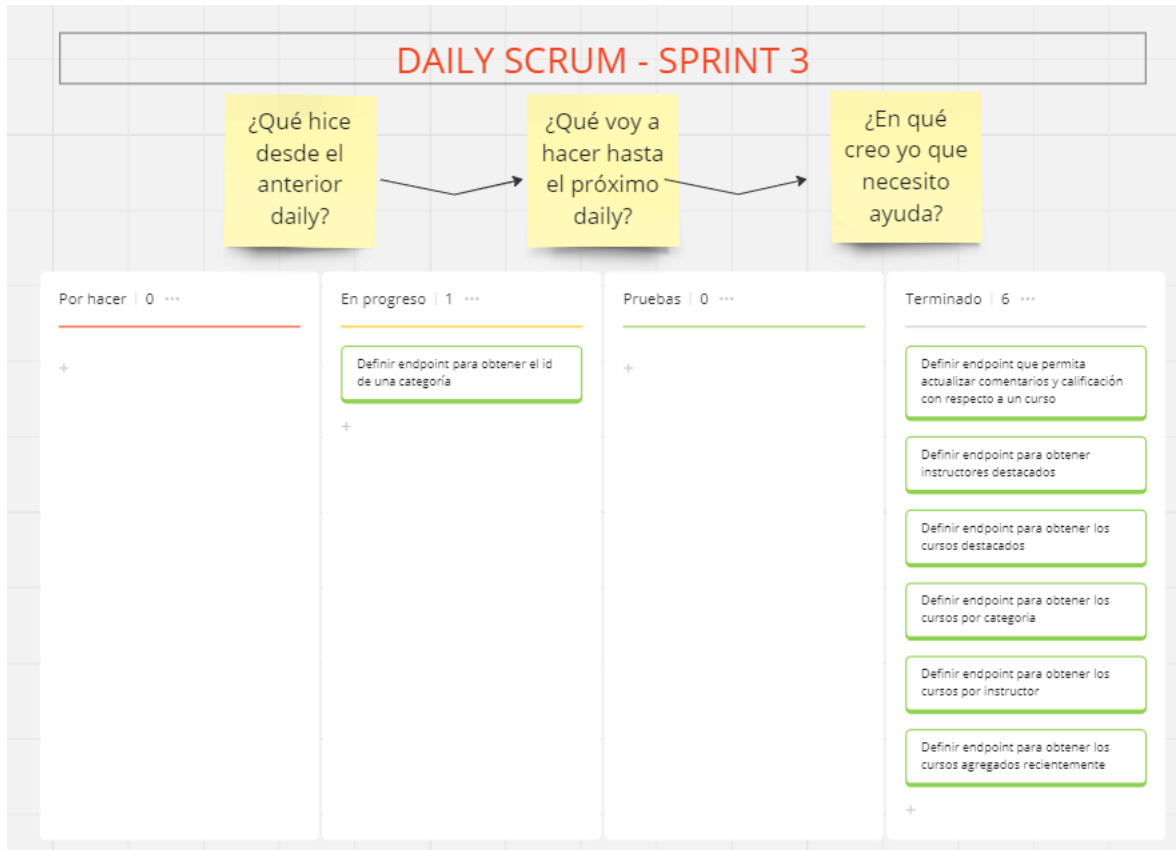


Figura 5.56 Tablero Kanban Sprint 3 - 18/12/2023

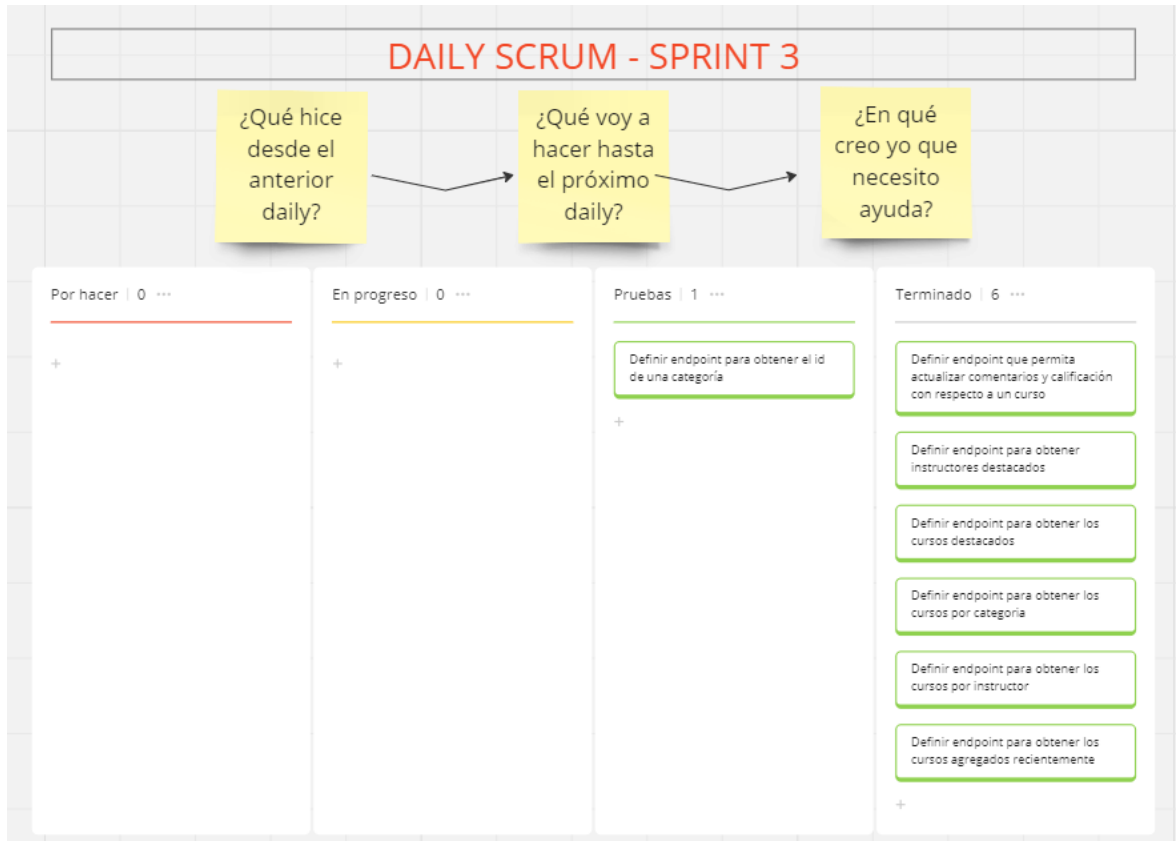


Figura 5.57 Tablero Kanban Sprint 3 - 19/12/2023

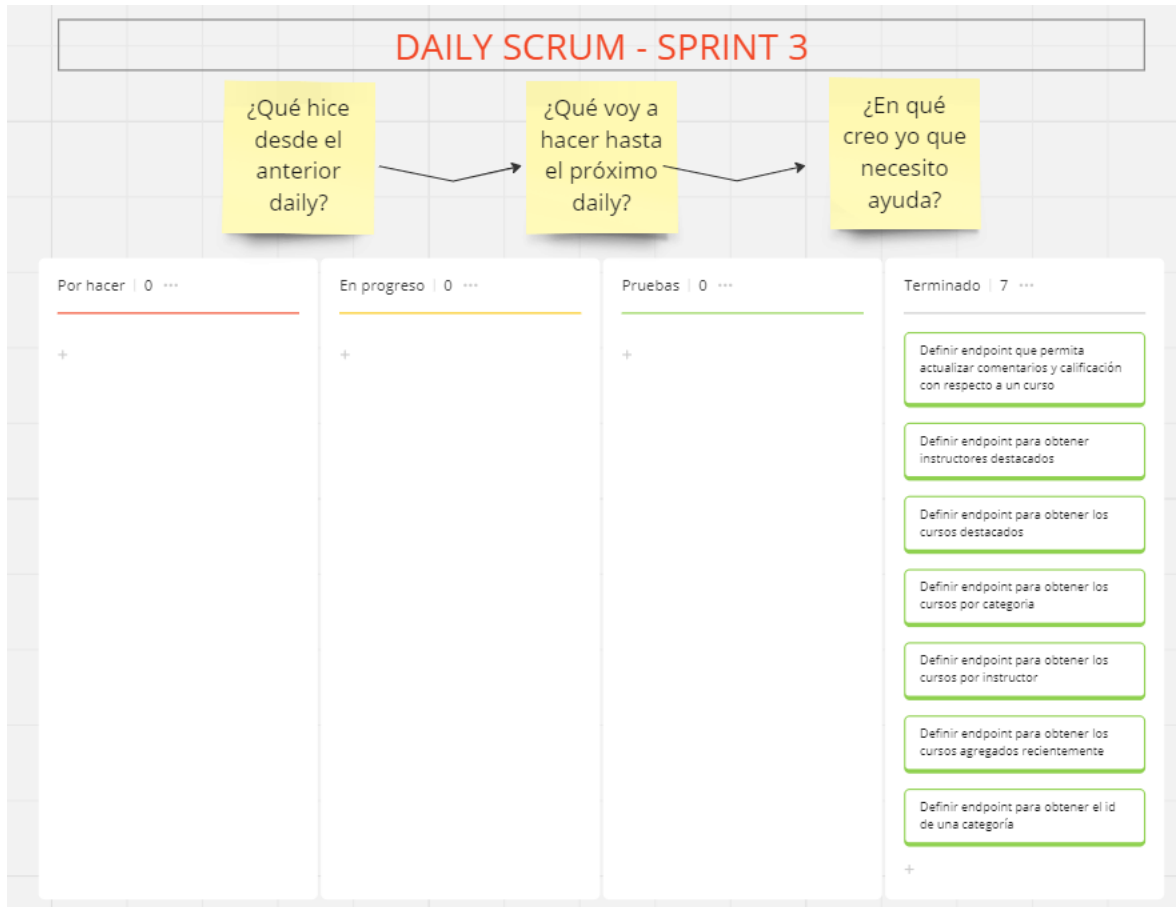


Figura 5.58 Tablero Kanban Sprint 3 - 20/12/2023 – Finalización Sprint 3