

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE UNA PLATAFORMA WEB DE APRENDIZAJE EN CIBERSEGURIDAD “LEARN2HACK”

BACKEND

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

EDISON PAUL RIVERA MICHILENA

edison.rivera@epn.edu.ec

DIRECTOR: VANESSA KATHERINE GUEVARA BALAREZO

vanessa.guevarav@epn.edu.ec

DMQ, marzo 2024

CERTIFICACIONES

Yo, **EDISON PAUL RIVERA MICHILENA** 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.

EDISON PAUL RIVERA MICHILENA

edison.rivera@epn.edu.ec

Certifico que el presente trabajo de integración curricular fue desarrollado por **EDISON PAUL RIVERA MICHILENA**, bajo mi supervisión.

Ing. VANESSA GUEVARA

DIRECTORA

vanessa.guevarav@epn.edu.ec

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.

EDISON PAUL RIVERA MICHILENA

DEDICATORIA

Este proyecto se lo dedico a mi familia, los cuales fueron una enorme ayuda y motivación para conseguir mis sueños y metas. Ellos estuvieron desde el segundo uno en el que elegí formarme como desarrollador y siempre creyeron en mí.

A mi pareja Estela, por todo su apoyo, paciencia y motivación para siempre continuar adelante a pesar de lo difícil de la situación.

EDISON PAUL RIVERA MICHILENA

AGRADECIMIENTO

Expreso mi más sincero agradecimiento a todas las personas que ayudaron en mi formación académica. En primer lugar, a mis profesores, quienes me supieron encaminar durante toda la etapa académica y varios mentores, los cuales siempre me apoyaron y ayudaron ante cualquier duda o problema ocurrido.

EDISON PAUL RIVERA MICHILENA

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.....	4
2 METODOLOGÍA.....	6
2.1 Metodología de Desarrollo.....	6
Roles.....	7
Artefactos.....	8
2.2 Diseño de la arquitectura.....	11
Arquitectura de Datos.....	11
Patrón Arquitectónico.....	12
2.3 Herramientas de desarrollo.....	12
Librerías.....	13
3 RESULTADOS.....	15
3.1 <i>Sprint 0</i> . Configuración del ambiente de desarrollo.....	15
Recopilación de requerimientos y alcance del proyecto.....	15
Definición de la estructura de directorios para el <i>backend</i>	17
Diseño y creación de la base de datos.....	18
Definición y alcance de los roles de usuario.....	18
3.2 <i>Sprint 1</i> . Resultados del desarrollo de <i>endpoints</i> para el registro, inicio de sesión, verificación de cuenta y autenticación.....	18
Generar <i>endpoints</i> que permiten registrar, eliminar y verificar cuenta.....	19
Generar <i>endpoints</i> que permiten iniciar sesión, cerrar sesión y reestablecer contraseña.....	20
Generar un <i>endpoint</i> que permita modificar perfil.....	21
Generar un <i>endpoint</i> que permita enviar el correo para el registro del profesor.....	22

3.3	<i>Sprint 2. Resultados del desarrollo de endpoints para la gestión de los cursos</i>	23
	Generar un <i>endpoint</i> para crear un curso.....	23
	Generar <i>endpoints</i> para empezar y visualizar un curso.....	24
	Generar un <i>endpoint</i> para editar un curso.....	25
	Generar un <i>endpoint</i> para editar la imagen de un curso.....	26
3.4	<i>Sprint 3. Resultados del desarrollo de endpoints para visualizar y filtrar cursos</i>	26
	Generar <i>endpoints</i> para visualizar y filtrar todos los cursos	27
3.5	<i>Sprint 4. Resultados del desarrollo de endpoints para la gestión de reseñas</i>	27
	Generar <i>endpoints</i> para visualizar, crear, eliminar y editar reseñas	28
	Generar <i>endpoints</i> para visualizar y eliminar reseñas de los cursos creados.....	28
	Generar <i>endpoints</i> para visualizar, eliminar reseñas de todos los cursos.....	29
3.6	<i>Sprint 5. Resultados de las pruebas para del backend</i>	30
	Ejecución de pruebas unitarias.....	30
	Ejecución de pruebas de estrés.....	31
	Ejecución de pruebas de seguridad	32
3.7	<i>Sprint 6. Despliegue del backend</i>	33
4	CONCLUSIONES	34
5	RECOMENDACIONES.....	35
6	REFERENCIAS BIBLIOGRÁFICAS	36
7	ANEXOS	39
	ANEXO I.....	40
	ANEXO II.....	41
	Recopilación de Requerimientos.....	41
	Historias de Usuario	42
	<i>Product Backlog</i>	48
	<i>Sprint Backlog</i>	50
	Ejecución de Pruebas.....	55
	ANEXO III.....	59
	ANEXO IV	60

RESUMEN

Iniciar en el campo de la ciberseguridad suele presentar desafíos y complejidades debido a que la mayoría de la documentación y guías disponibles están en inglés y utilizan varios tecnicismos., lo que lleva a la necesidad de recurrir a cursos, los cuales a menudo son teóricos y carecen de prácticas con laboratorios controlados.

Con el propósito de asistir a las personas que están interesadas en este campo, se ha desarrollado la plataforma de aprendizaje de ciberseguridad “Learn2Hack”. Esta plataforma facilitará a las personas a dar sus primeros pasos. Los estudiantes tienen la opción de inscribirse en diversos cursos con temáticas variadas. Cada curso ofrece material adicional, como imágenes y documentos en formato PDF. Además, los estudiantes podrán contestar una serie de preguntas de prueba y contarán con acceso a un laboratorio controlado para poner en práctica los conocimientos adquiridos durante el curso.

La ejecución del proyecto implica la interacción entre el *frontend* y el *backend*. Este último emplea un API desarrollada con FastAPI, PostgreSQL como base de datos, Gmail para el envío de correos y Cloudinary para alojar los archivos. Este documento detalla el uso de todas las herramientas, librerías y servicios de terceros, así como la arquitectura del proyecto. Se adopta la metodología SCRUM para el desarrollo, ya que facilita un enfoque flexible y entregas incrementales, permitiendo así alcanzar los objetivos establecidos. Cada *sprint* se centrará en detallar las funcionalidades y cursos presentados.

Finalmente, se incluirán las conclusiones derivadas al concluir el proyecto, así como las recomendaciones necesarias para el despliegue exitoso del mismo en producción.

PALABRAS CLAVE: *backend*, *frontend*, FastAPI, PostgreSQL, Gmail, Cloudinary, SCRUM.

ABSTRACT

Starting a journey in the field of cybersecurity is often complex and challenging. Much of the available documentation and guides are in English and filled with technical terminology, necessitating reliance on courses that are typically heavy on theory but lack practical experience in controlled environments.

To aid individuals interested in this domain, the cybersecurity learning platform "Learn2Hack" has been developed to guide beginners through their initial steps. Students can enroll in various courses covering different topics. Each course provides supplementary materials such as images and PDFs, along with a series of test questions. Additionally, students will have access to a controlled laboratory environment to practice the concepts learned in each module.

The project is structured to ensure seamless interaction between the frontend and backend. The backend utilizes an API developed with FastAPI, while PostgreSQL serves as the database. Email communication will be facilitated through Gmail, and files will be hosted on Cloudinary. This document will meticulously outline the utilization of all tools, libraries, third-party services, and the project's architecture.

Adopting SCRUM as our development methodology enables flexible work practices and incremental deliveries. This approach allows us to meet our objectives effectively. Each sprint will focus on presenting specific functionalities or modules, with detailed descriptions provided.

Concluding the project, we will present the findings and recommendations necessary for deploying it into production, ensuring its effectiveness and sustainability in the long run.

KEYWORDS: *backend, frontend, FastAPI, PostgreSQL, Gmail, Cloudinary, SCRUM*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El proceso de aprendizaje de ciberseguridad puede resultar desafiante para aquellas personas sin conocimientos previos en el campo. La búsqueda de información sobre cómo empezar en este campo suele conducir a contenidos en inglés, con términos técnicos y una extensa ruta para llegar a ser un profesional, generando una sensación agobiante. Por otro lado, la participación en cursos en vivo, ya sea presencial o virtual, puede resultar complicada debido a los horarios.

La práctica es fundamental en la formación de profesionales en ciberseguridad. Muchos cursos existentes se centran en aspectos teóricos, son desactualizados y tienen una mínima parte práctica. A menudo se asume erróneamente que para tener un dominio en seguridad informática se debe invertir mucho dinero, cuando, en realidad, es posible iniciar en este campo con pocas herramientas.

La creación de la plataforma de aprendizaje "Learn2Hack" pretende dar a los estudiantes los primeros pasos en ciberseguridad, adaptándose a sus horarios con material disponible siempre. La plataforma aborda las principales vulnerabilidades del OWASP TOP 10 [1] y brinda material adicional como una prueba y un laboratorio para reforzar los conocimientos adquiridos. El administrador puede gestionar fácilmente los cursos, estudiantes, profesores y reseñas mediante su panel de administración. Los profesores pueden crear cursos fácilmente, incluyendo cuestionarios para evaluar a los estudiantes, y recibir retroalimentación a través de las reseñas al finalizar cada curso. Los usuarios, a su vez, tienen acceso a un filtro de búsqueda que les permite clasificar los cursos según su dificultad, palabras clave y visualizar los cursos completados.

Este Trabajo de Integración Curricular aborda el desarrollo, documentación, *testing* y despliegue en producción del *backend* de la plataforma "Learn2Hack". Se han establecido tres roles distintos para mitigar posibles riesgos de seguridad, y el desarrollo se realiza exclusivamente a través de APIs, separando la lógica de *frontend* y *backend*. La autenticación y autorización de recursos se lleva a cabo mediante JWT [2], proporcionando un nivel adicional de seguridad.

1.1 Objetivo general

Desarrollar el *backend* para la gestión de la plataforma web de aprendizaje en ciberseguridad “Learn2Hack”

1.2 Objetivos específicos

1. Levantar los requerimientos para el desarrollo del *backend*.
2. Diseñar la arquitectura del *backend* según la recopilación de requerimientos.
3. Diseñar el modelo de la base de datos relacional.
4. Codificar los *endpoints* y módulos para el *backend* según los roles.
5. Garantizar la correcta funcionalidad de cada *endpoint* a través de pruebas.
6. Desplegar el *backend* a un ambiente en la nube.

1.3 Alcance

Actualmente, el avance acelerado de la tecnología ha dado lugar a la creación constante de proyectos de *software*, aplicaciones y sistemas. Aunque estos avances proporcionan beneficios significativos para los usuarios, también conllevan riesgos considerables. La presencia de fallos de seguridad en una aplicación puede exponer datos sensibles al público, ponerlos en riesgo de venta ilícita o utilizarlos de manera perjudicial. Las principales amenazas en cuanto a ciberseguridad se centran en el robo de información y la suplantación de identidad, donde los ciberdelincuentes usan diferentes técnicas como el *phishing*, *malware* y *ransomware* para llevar a cabo sus ataques. Las empresas son los principales objetivos de estos ataques, porque manejan mucha información de sus usuarios [3].

El presente proyecto propone el desarrollo, *testing* y despliegue a producción del *backend* para la plataforma web de aprendizaje de ciberseguridad “Learn2Hack”. Esta plataforma permite a los estudiantes acceder a diferentes cursos sobre las vulnerabilidades del OWASP TOP 10, material adicional en formato PDF e imagen, pruebas para evaluar su conocimiento y laboratorios controlados para poner en práctica lo aprendido, además de permitirles dejar reseñas al finalizar cada curso.

Los profesores pueden crear y editar sus propios cursos, subir el material necesario, crear pruebas y eliminar y ver las reseñas asociadas a sus cursos. Por último, el administrador puede gestionar toda la plataforma fácilmente desde su panel. La metodología de desarrollo empleada es la de *software* ágil, respaldada por una base de datos relacional, un *framework*, librerías y módulos específicos para desarrollo y *testing*, garantizando así el correcto funcionamiento antes del despliegue a producción [4].

Roles que se establecen: Administrador, Profesor y Estudiante.

Funcionalidades del rol administrador:

- *Endpoints* para iniciar y cerrar sesión, así como para reestablecer la contraseña.
- *Endpoints* para visualizar y modificar su propio perfil.
- *Endpoint* para enviar enlace de registro al profesor.
- *Endpoints* para visualizar cursos y modificar su imagen.
- *Endpoints* para visualizar y eliminar las reseñas de todos los cursos.

Funcionalidades del rol profesor:

- *Endpoints* para iniciar y cerrar sesión, así como para reestablecer la contraseña.
- *Endpoints* para visualizar y modificar su propio perfil.
- *Endpoints* para crear, visualizar y editar sus propios cursos.
- *Endpoints* para visualizar y eliminar reseñas de sus propios cursos.

Funcionalidades del rol estudiante:

- *Endpoints* para iniciar, cerrar sesión y para reestablecer la contraseña.
- *Endpoints* para registrar, verificar y eliminar cuenta
- *Endpoints* para visualizar y modificar su propio perfil.
- *Endpoints* para visualizar todos los cursos

- *Endpoints* para filtrar los cursos en base a la dificultad y palabras clave.
- *Endpoints* para visualizar los cursos completados
- *Endpoints* para empezar en uno o más cursos.
- *Endpoints* para visualizar el contenido del curso
- *Endpoints* para comprobar respuestas de cuestionario
- *Endpoint* para gestionar reseñas.

1.4 Marco Teórico

Backend

Dentro del área de desarrollo de *software*, se desempeñan diversos roles, siendo el rol de *backend* fundamental en la gestión de la lógica de los procesos o funciones de un *software*. Este rol implica la creación y diseño de bases de datos, el mantenimiento o escalado de APIs, así como la optimización de procesos y aplicaciones, elementos esenciales para el desarrollo exitoso de un proyecto de *software* [5].

Lenguaje de Programación Python

Python es uno de los lenguajes de programación más demandados a nivel global. Lo que lo distingue es su facilidad de aprendizaje, gracias a su sintaxis amigable, rapidez y potencia. Al ser un lenguaje multipropósito, permite utilizarlo en diferentes campos de la tecnología, facilitando el desarrollo de proyectos complejos, haciéndolos más manejables y escalables. [6].

FastAPI

FastAPI es un moderno *framework*, rápido, robusto, sencillo e intuitivo. Al utilizar Python como base, posibilita la creación de APIs con unas pocas líneas de código. Además, ofrece la funcionalidad de Swagger para documentar y probar cada *endpoint* de manera sencilla [7].

JSON (JavaScript Object Notation)

JSON es una de las formas más comunes de intercambio de información entre APIs o aplicaciones web. Su simplicidad facilita su interpretación tanto por lenguajes de programación como por humanos. Su estructura se basa en pares de datos (clave y valor) [8].

JWT (JSON Web Tokens)

Los JWT son una forma de autenticación y autorización asignada a un usuario dentro de un sistema web. Se componen de tres partes, el encabezado, donde se define el algoritmo y tipo de *token* a usar; el cuerpo, donde se indican principalmente los datos de fecha de expedición, fecha de expiración y el nombre del usuario del *token*, y la tercera parte, la *secret key* que es la firma que válida al *token* [9].

API Restful (Interfaz de Programación de Aplicaciones basada en Transferencia de Estado Representacional)

API Restful es una interfaz de comunicación segura para intercambiar información. Cuando se busca la comunicación con un servidor tercero, esta interfaz permite esto ya que sigue los estándares de comunicación de *software*. Su principal característica es poder realizar operaciones HTTP (GET, POST, PUT, DELETE, OPTIONS, HEAD, PATCH) e indicar siempre el estado de un recurso [10].

Cloudinary

Cloudinary es un SaaS (*Software-as-a-service*) que permite a los desarrolladores la carga de archivos a la nube de forma rápida y sencilla. Proporciona un API Rest para administrar archivos y es compatible con la mayoría de los lenguajes de programación [11].

2 METODOLOGÍA

Los estudios de casos representan un enfoque detallado para investigar un tema específico y se aplican comúnmente en campos como la investigación social, la educación, el ámbito empresarial y la clínica. Su objetivo principal es describir, comprender y evaluar el problema presentado desde diferentes perspectivas, ofreciendo una visión integral. La elección de un estudio de caso resulta útil cuando se busca mantener un proyecto bien estructurado y flexible [12].

El presente Proyecto de Integración Curricular se basa en un estudio de casos. Esta decisión se fundamenta en la necesidad de llevar a cabo una investigación detallada acerca de los desafíos más significativos que enfrentan las personas al iniciar en el campo de la ciberseguridad. Como resultado de este enfoque, se lleva a cabo el desarrollo de un *backend* para una plataforma web de aprendizaje en ciberseguridad. En esta plataforma, todos los estudiantes tienen acceso a cursos y material relevante, facilitando así su introducción y desarrollo en el ámbito de la ciberseguridad.

2.1 Metodología de Desarrollo

La metodología de desarrollo constituye un conjunto de métodos, técnicas y herramientas empleadas en el proceso de desarrollo de un proyecto de *software*. La adopción de estas metodologías proporciona un control efectivo sobre el proyecto, anticipándose a posibles errores, optimizando el tiempo y gestionando de manera eficiente los recursos disponibles. El propósito principal radica en agilizar los procesos, adherirse a un marco de trabajo y garantizar la calidad del producto de *software* resultante [13].

Las metodologías de desarrollo ágil buscan la distribución del desarrollo de un proyecto de *software* en iteraciones rápidas. En este enfoque no se sigue de forma estricta una serie fija de pasos, sino que proporciona un marco de trabajo y una forma de pensar colaborativa y flexible. El objetivo es lograr mejoras constantes y aumentar la satisfacción del cliente final, adaptándose a los cambios que pueden surgir en las diferentes etapas del ciclo del *software* [14].

Considerando el enfoque de este proyecto de *software*, se ha optado por la metodología de desarrollo ágil SCRUM. La elección de SCRUM se fundamenta en la capacidad de evidenciar el progreso del proyecto a través de cada iteración presentada. Esta metodología permite adaptarse a los cambios que puedan surgir en el desarrollo del proyecto, garantizando una comunicación constante con el cliente. Esto permite que el cliente retroalimente y se puedan realizar los ajustes necesarios según los comentarios. En la implementación de SCRUM, se involucran diversos roles y artefactos, los cuales se detallan a continuación:

Roles

En la metodología SCRUM existen tres roles y cada uno tiene un papel fundamental en el desarrollo del proyecto de *software*:

Product Owner

El *Product Owner* mediante un análisis de los principales aspectos del producto, establece una lista de prioridades, las cuales guiarán el inicio del siguiente *sprint*. Este rol implica la continua modificación del *Product Backlog* para ajustar las prioridades según los análisis realizados [15]. La asignación específica para este rol se detalla en la **Tabla 2. 1**.

Development Team

El equipo de desarrollo es multifuncional, ya que cada miembro del equipo tiene una especialidad en determinada área. Cada integrante tiene la oportunidad de aprender de otras áreas a lo largo del proyecto, lo cual promueve el aprendizaje mutuo. El equipo tiene la responsabilidad de autoorganizarse para cumplir con cada *sprint* y tomar decisiones colaborativas sobre cómo lograrlo [15]. La persona asignada para desempeñar este rol se encuentra detallada en **Tabla 2. 1**

Scrum Master

El *Scrum Master* asiste al *Development Team* en la aplicación y comprensión de todas las características y ventajas de SCRUM. Su función principal es evitar conflictos o impedimentos durante el desarrollo del proyecto de *software*, a la vez garantiza el cumplimiento del marco de trabajo propuesto por SCRUM [15]. La persona asignada para este rol se encuentra detallada en la **Tabla 2. 1**.

Tabla 2. 1: Designación de roles para el proyecto

ROL	ENCARGADO
<i>Producto Owner</i>	Ing. Vanessa Guevara, MSc.
<i>Scrum Master</i>	Ing. Vanessa Guevara, MSc.
<i>Development Team</i>	Edison Paul Rivera Michilena

Artefactos

Los artefactos en SCRUM representan los recursos utilizados por el *Development Team* y las partes interesadas para detallar el producto de *software*, así como los requerimientos y tareas realizadas durante el desarrollo [16].

Recopilación de Requerimientos

Esta etapa del proyecto es crucial para llevar a cabo el desarrollo del proyecto de *software*, ya que implica la recopilación de requisitos, funcionalidades y necesidades a ser cumplidos. Todos estos elementos se detallan y consolidan con el propósito de facilitar la etapa de codificación, minimizando ambigüedades o confusiones. En la **Tabla 2. 2** se presenta el formato utilizado para la recopilación de requerimientos, los cuales se listan por completo en el **ANEXO II**.

Tabla 2. 2: Formato Recopilación de Requerimientos

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DE SISTEMA	ID - RR	ENUNCIADO DEL ITEM
<i>backend</i>	RR - 001	<p>Como usuario estudiante necesita varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Registrar Cuenta. • Verificar Cuenta. • Eliminar Cuenta.

Historias de Usuario

Las historias de usuario consisten en explicaciones proporcionadas por los usuarios finales, las cuales describen de manera no técnica el propósito de una funcionalidad. Estas representan un componente esencial en SCRUM, ya que permiten que el *Development Team* comprenda claramente qué se va a desarrollar [17]. En la **Tabla 2. 3:** se presenta el formato utilizado para las Historias de Usuario, las cuales se listan por completo en el **ANEXO II.**

Tabla 2. 3: Formato de Historia de Usuario

HISTORIA DE USUARIO	
Identificador: HU-001	Usuario: Estudiante
Nombre Historia: Registrar, eliminar y verificar cuenta.	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Media
Iteración asignada: 1	
Responsable: Edison Rivera	
Descripción: Se necesitan generar varios <i>endpoints</i> para el perfil estudiante, las acciones a realizar son: <ul style="list-style-type: none"> • Registrar Cuenta. • Verificar Cuenta. • Eliminar Cuenta. 	
Observación: Para verificar una cuenta se envía un correo electrónico al usuario estudiante al momento de registrarse.	

Product Backlog

El *Product Backlog* consiste en una lista de funcionalidades organizadas por orden de prioridad, las cuales resultan esenciales para alcanzar los objetivos del proyecto de *software*. El *Development Team* se enfoca en los primeros elementos definidos en el *Product Backlog*, según sus recursos y capacidades, con el fin de entregarlos en cada *sprint* [18]. A continuación, en la **Tabla 2. 4** se presenta el formato utilizado para el *Product Backlog*, las cuales se listan por completo en el **ANEXO II.**

Tabla 2. 4: Formato para el *Product Backlog*

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU-001	Registrar, eliminar y verificar cuenta.	1	Finalizado	Alta

Sprint Backlog

El *Sprint Backlog* es una lista detallada de las tareas que deben completarse durante cada *sprint*. Esto es útil ya que proporciona una visión detallada de las tareas para todo el equipo. Permite identificar los problemas en las tareas, evaluar el progreso y tomar decisiones informadas al respecto [19]. A continuación, en la **Tabla 2. 5** se presenta el formato utilizado en el *Sprint Backlog*, los cuales se detallan por completo en el **ANEXO II**.

Tabla 2. 5: Formato de *Sprint Backlog*

ELABORACIÓN DE <i>SPRINT BACKLOG</i>						
ID–SB	NOMBRE	MÓDULO	ID–HU	HISTORIA DE USUARIO	TAREA	TIEMPO ESTIMADO
SB-001	Diseño e implementación de <i>endpoints</i> para el registro, inicio de sesión, verificación de cuenta y autenticación	Módulo de inicio registro, verificación y eliminación de cuenta.	HU-001	Registrar, eliminar y verificar cuenta.	<ul style="list-style-type: none"> • Diseño de <i>endpoints</i> para registrar, eliminar y verificar cuenta. • Validación de datos. • Registro y consulta en la base de datos. 	10H

2.2 Diseño de la arquitectura

En el desarrollo de *software*, el diseño de la arquitectura implica la construcción y consolidación de una base robusta sobre la cual se construirá un proyecto de *software*. Este proceso requiere la toma de decisiones importantes que abordan aspectos clave como la capacidad de gestión, flexibilidad, rendimiento y seguridad. Las decisiones tomadas durante esta etapa tendrán un impacto significativo a largo plazo, especialmente al implementar nuevas funcionalidades, escalar el proyecto y brindar mantenimiento continuo [20]. A continuación, se detalla el diseño de la arquitectura contemplada en el proyecto.

Arquitectura de Datos

La estructura de datos se refiere a la descripción de cómo se manejarán los datos dentro de un sistema. Su importancia radica en comprender claramente el plan y la ruta que siguen los datos a través del sistema de almacenamiento. La **Figura 2. 1** muestra el modelado de la base de datos del sistema utilizada para el *backend* del proyecto.

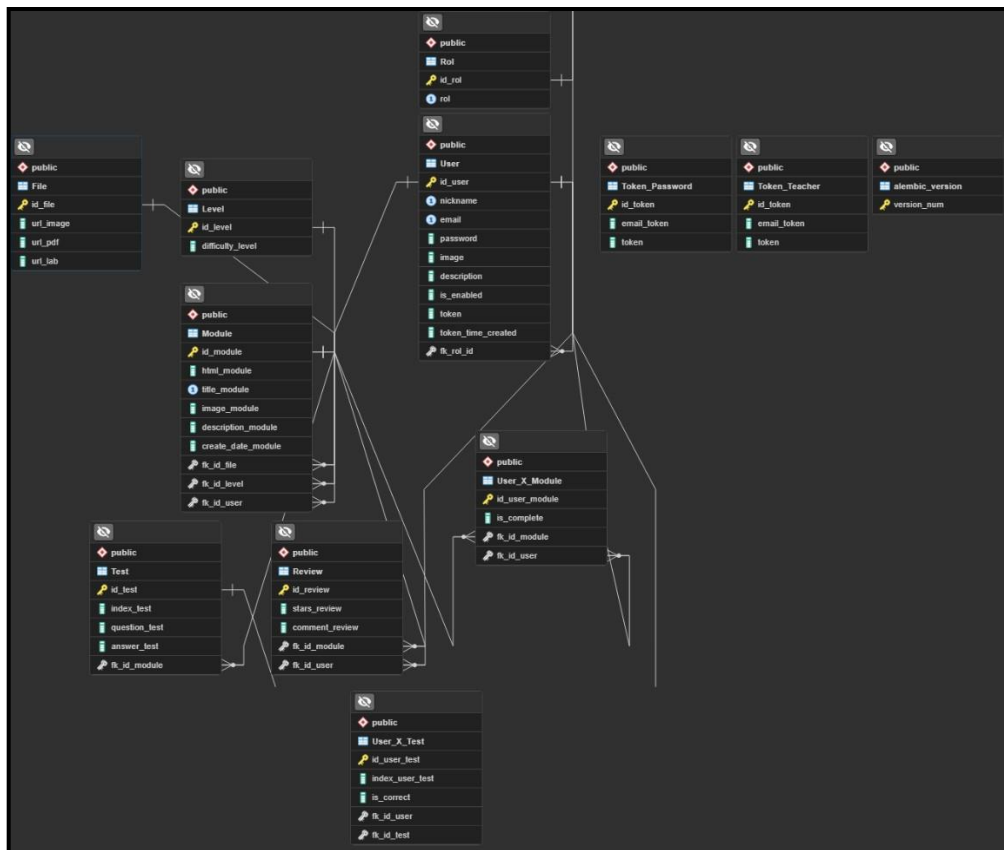


Figura 2. 1: Estructura de la base de datos

Patrón Arquitectónico

El patrón arquitectónico Modelo-Vista-Controlador (*MVC*) es un patrón de diseño de *software* que tiene como objetivo separar las interfaces de usuario (*frontend*), los datos y la lógica de control. Este enfoque proporciona al equipo de trabajo una mejor visibilidad de todo el proyecto, divide las responsabilidades en roles distintos y, a largo plazo, contribuye a una mayor eficacia en el mantenimiento del proyecto [21].

El principal objetivo del patrón de diseño MVC es distribuir las responsabilidades de manera eficiente dentro del equipo de trabajo. Cada persona o grupo de personas puede trabajar de manera individual en un componente específico, lo que facilita la implementación rápida de los cambios cuando surgen. A continuación, en la **Figura 2. 2** se detalla en patrón de arquitectura utilizado en el *backend*, además de detallar el proceso de comunicación con servicios de terceros.

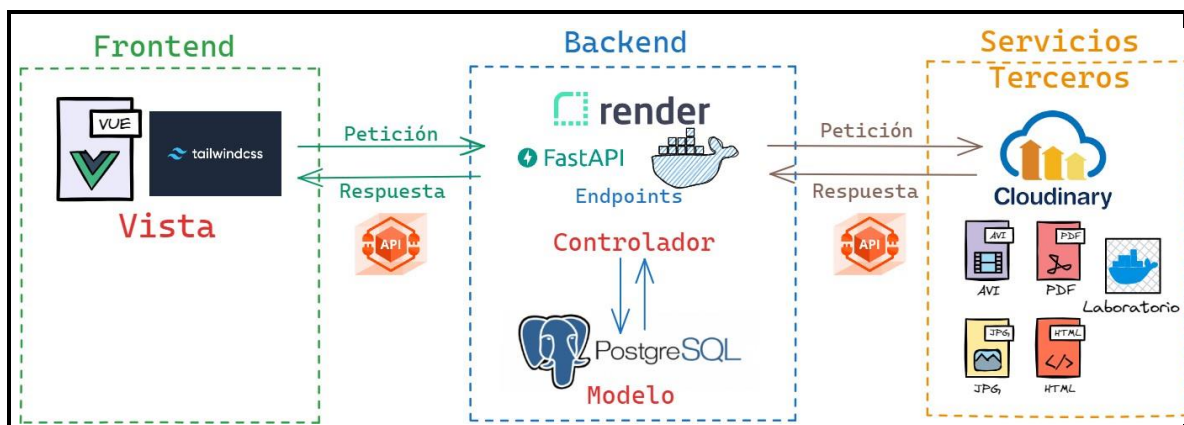


Figura 2. 2: Patrón Arquitectónico backend

2.3 Herramientas de desarrollo

Las herramientas de desarrollo desempeñan un papel fundamental al facilitar la integración ágil y sencilla de funcionalidades o tareas complejas en el proyecto. Esto impulsa la productividad y acelera el tiempo de desarrollo al evitar tareas repetitivas y la creación desde cero de herramientas ya diseñadas. Además, la elección de utilizar servicios de terceros se presenta como una buena opción para aumentar la productividad y simplificar el desarrollo de un proyecto. En la **Tabla 2. 6** se detalla el conjunto de herramientas usadas para el *backend*.

Tabla 2. 6: Herramientas del *backend*

Herramienta	Justificación
Visual Studio Code	Editor de texto utilizado para el desarrollo del <i>backend</i> .
GitHub	Plataforma que facilita el versionamiento y alojamiento del código, manteniendo la organización de este.
Render	Servicio que ofrece alojamiento para proyectos de Python y bases de datos.
PostgreSQL	Base de datos relacional destinada a almacenar los datos de la aplicación.
Docker	Herramienta que posibilita empaquetar el código en un contenedor, facilitando su portabilidad y despliegue.

Librerías

Las librerías son conjuntos de archivos de código empleados durante el desarrollo de *software*. Su función principal es optimizar el tiempo en el proceso de desarrollo, ya que están diseñadas para abordar y solventar necesidades específicas [22]. En la **Tabla 2. 7** se listan todas las librerías usadas para el desarrollo del *backend* de este proyecto.

Tabla 2. 7: Librerías de *Backend*

Librería	Descripción
FastAPI	<i>Framework</i> que facilita el desarrollo de una API robusta y documentada.
Uvicorn	Herramienta que permite crear un servidor sencillo capaz de ejecutar un proyecto de FastAPI.
psycopg2-binary	Facilita la interacción fácil y sencilla con PostgreSQL.
SQLAlchemy	ORM que permite la creación y gestión de conexiones con una base de datos.
python-dotenv	Permite cargar información sensible, como credenciales, desde variables de entorno, garantizando un desarrollo seguro.

alembic	Herramienta que posibilita realizar migraciones automáticas al iniciar el servicio de FastAPI.
passlib[bcrypt]	Permite la creación de contraseñas encriptadas, asegurando la seguridad de los datos.
python-jose[ryptography]	Facilita la creación, decodificación y manipulación de JSON Web Tokens, esenciales para la autenticación y autorización.
pydantic	Permite la creación de modelos que representarán tablas en la base de datos.
regex	Facilita la creación de expresiones regulares para validar datos y trabajar con cadenas de texto a un nivel más complejo.
cloudinary	Librería creada por Cloudinary para interactuar con su API.
Fernet	Posibilita la encriptación asíncrona y síncrona de datos.
smtplib	Permite la interacción con un servicio SMTP, necesario para el envío de correos con Gmail.
secrets	Genera números pseudoaleatorios criptográficamente fuertes, comúnmente utilizados para <i>tokens</i> o secretos.
dataclasses	Decorador de clases que genera automáticamente métodos, brindando mayor facilidad en su creación.

3 RESULTADOS

En esta sección, se presentan de forma detallada los avances progresivos logrados a lo largo del desarrollo del proyecto. Además, se exponen las pruebas realizadas y se indican las funciones de cada *endpoint* implementado.

3.1 *Sprint 0*. Configuración del ambiente de desarrollo

En este *sprint*, se abordan las siguientes tareas:

- Recopilación de requerimientos del proyecto.
- Definición de la estructura de directorios para el *backend*.
- Diseño y creación de la base de datos.
- Definición y alcance de los roles de usuario.

Recopilación de requerimientos y alcance del proyecto

Implementación de *endpoints* para el registro y verificación de usuarios

Se desarrollan dos *endpoints* para el registro de estudiantes y profesores respectivamente. Se solicitan 3 campos: correo electrónico, nombre de usuario y contraseña. En el registro de profesores, el administrador envía un enlace único a su correo, solicitando los campos de nombre de usuario y contraseña. La verificación de cuenta se realiza únicamente para los estudiantes a través de un correo electrónico.

Implementación de *endpoints* para el inicio de sesión, cierre de sesión, modificación de perfil y eliminar cuenta

Se desarrollan tres *endpoints* que cumplen la misma función para los tres roles, iniciar sesión, cerrar sesión y modificar perfil. En la modificación de perfil, no se permite cambiar el correo electrónico con el que se registraron inicialmente. El *endpoint* de eliminar cuenta es para el rol de estudiante.

Implementación de *endpoints* para crear un curso

Se desarrollan tres *endpoints*: el primero para subir el material adicional del curso (imagen, *PDF* y laboratorio); el segundo guarda la información del curso (título,

descripción, imagen del curso, nivel de dificultad, fecha de creación y contenido del curso); y, por último, el tercero se encarga de guardar las preguntas del curso.

Implementación de *endpoints* para reestablecer contraseña

Se desarrolla un *endpoint* para cada uno de los tres roles, donde los usuarios deben indicar un correo electrónico existente al cual llegará el correo de restablecimiento de contraseña.

Implementación de *endpoints* para editar un curso

Se implementan dos *endpoints*: el primero sirve para editar el material adicional del curso y el segundo para editar la información básica. En este caso, no se pueden modificar las preguntas ya planteadas. En el caso del administrador, solo puede modificar la imagen del curso.

Implementación de *endpoints* para visualizar y filtrar un curso y comprobar respuestas

Se desarrollan tres *endpoints*: el primero sirve para ver la información básica de todos los cursos (imagen, título, descripción y nivel de dificultad); el segundo indica la información relacionada con el curso (contenido, material adicional y preguntas). El último *endpoint* permite a los estudiantes comprobar sus respuestas.

Implementación de *endpoints* para visualizar, crear, editar y eliminar reseñas

Se desarrollan varios *endpoints* para que el estudiante pueda dejar una única reseña al finalizar un curso, la cual puede editar o eliminar. Debe asignar una puntuación del 1 al 5 y dejar un comentario. El profesor puede ver y eliminar todas las reseñas de sus cursos, mientras que el administrador puede eliminar y visualizar las reseñas de todos los cursos.

En la **Figura 3. 1** se presenta un diagrama UML que detalla los roles y las funciones a las que tendrán acceso cada uno de ellos. El objetivo de esta representación es simplificar y mejorar la comprensión de las funcionalidades asignadas a cada rol en el contexto de este proyecto.

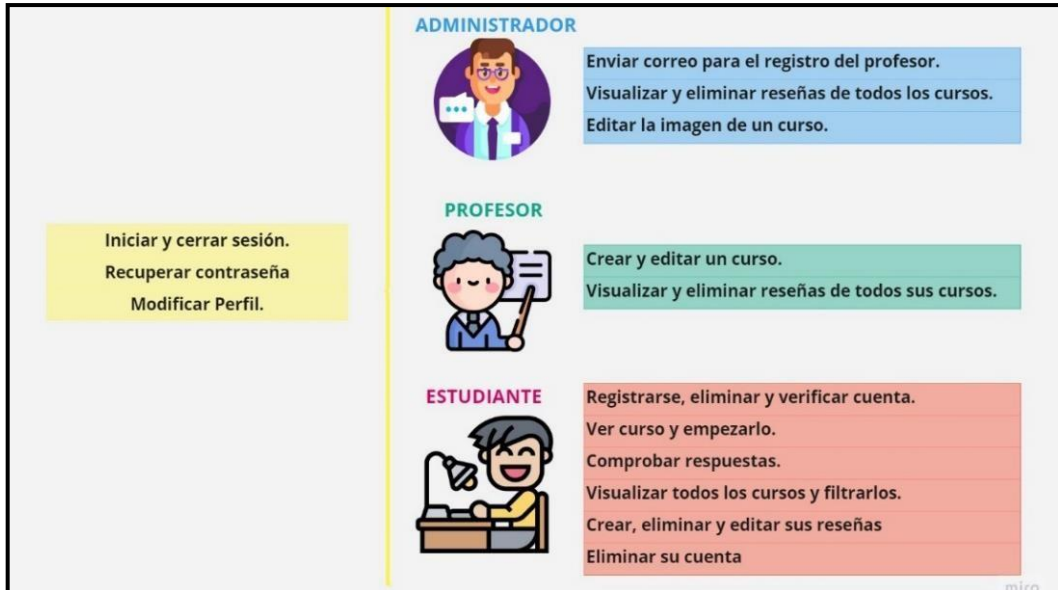


Figura 3. 1: Perfiles de usuario y funcionalidades del *backend*

Definición de la estructura de directorios para el *backend*

Se utilizó el editor de texto Visual Studio Code, respaldado por diversos *plugins* que agilizaron el desarrollo del proyecto. En este contexto, se definió una estructura de directorios con el objetivo de mantener una organización efectiva del código, tal como se detalla en la **Figura 3. 2**.

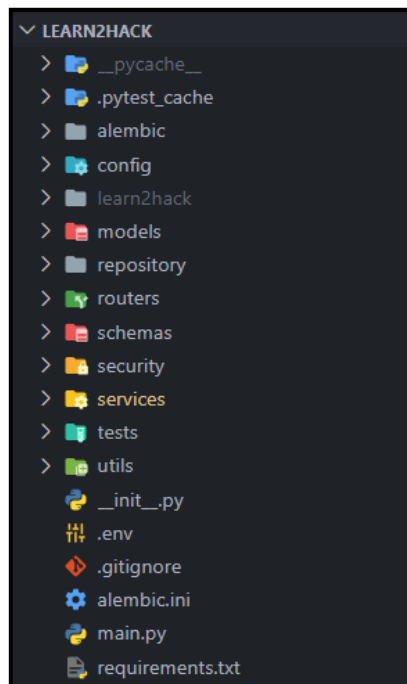


Figura 3. 2: Estructura del proyecto

Diseño y creación de la base de datos.

Para el desarrollo de la plataforma de aprendizaje de ciberseguridad “Learn2Hack” se optó por emplear PostgreSQL como sistema de gestión de bases de datos. En la **Figura 3. 3** se presenta de manera detallada todas las tablas correspondientes a esta estructura.

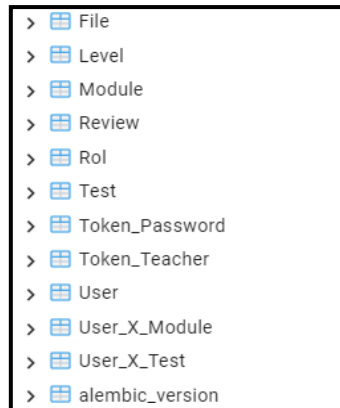


Figura 3. 3: Diseño de la base de datos

Definición y alcance de los roles de usuario.

En la definición y alcance de los roles de usuario para este proyecto, se establecieron tres roles, cada uno con permisos específicos para asegurar un control de acceso adecuado. En la **Figura 3. 4**, se indican los roles de usuario definidos.



Figura 3. 4: Roles de usuario

3.2 *Sprint 1. Resultados del desarrollo de endpoints para el registro, inicio de sesión, verificación de cuenta y autenticación*

En este *sprint* se incluyen las siguientes tareas:

- Generar *endpoints* para registrar, eliminar y verificar cuenta.

- Generar *endpoints* para iniciar sesión, cerrar sesión y reestablecer contraseña.
- Generar un *endpoint* para modificar el perfil.
- Generar un *endpoint* para enviar un correo para el registro del profesor.

Generar *endpoints* que permiten registrar, eliminar y verificar cuenta

Se han desarrollado *endpoints* destinados al registro, verificación y eliminación de cuenta para el rol de estudiante. En el caso de la acción de registro, el estudiante realiza una solicitud POST, lo que genera la creación de una cuenta. Inmediatamente, se envía un correo de verificación con un *token* único al correo electrónico asociado a la cuenta recién creada. Para verificar la cuenta, se utiliza este *token* mediante una petición GET. Por último, mediante una petición POST el estudiante podrá eliminar su cuenta.

En la **Figura 3. 5** se presenta el correo de verificación de cuenta enviado al estudiante y en la **Figura 3. 6** se muestra el resultado de la prueba unitaria correspondiente. Es importante destacar que los procesos de consumo, almacenamiento de registros y validaciones se detallan en el **ANEXO II**.

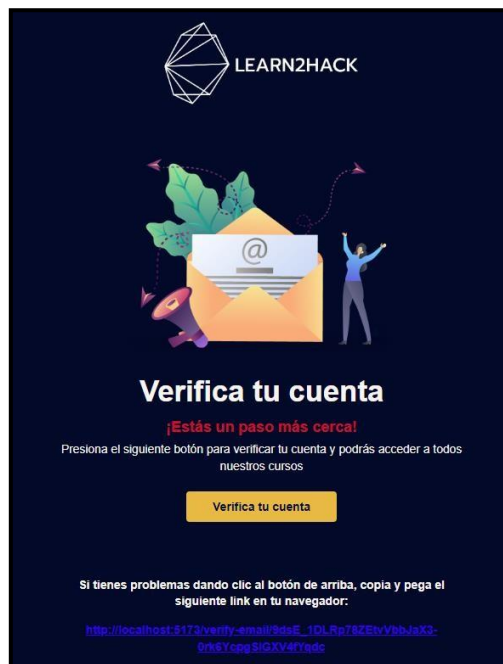


Figura 3. 5: Correo de verificación de cuenta

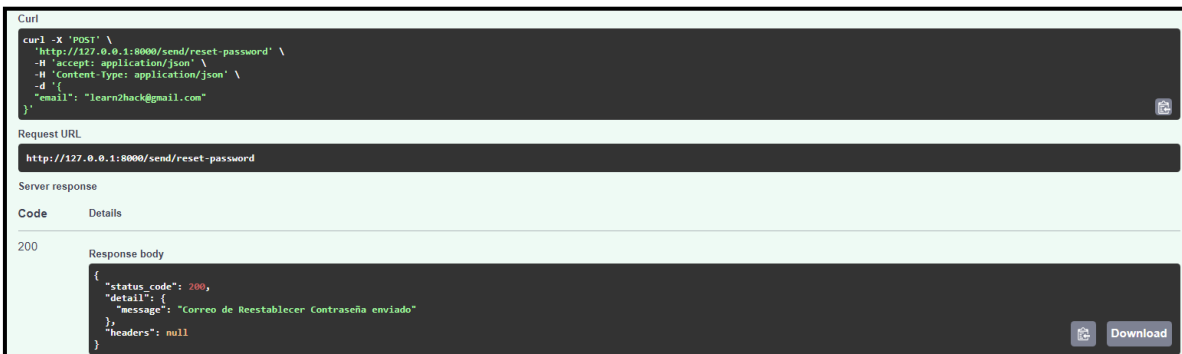
```
tests\register_test.py . [100%]
===== 1 passed in 2.88s =====
```

Figura 3. 6: Resultado de la prueba unitaria de registro

Generar *endpoints* que permiten iniciar sesión, cerrar sesión y reestablecer contraseña

Se han diseñado *endpoints* destinados a los tres roles establecidos en el proyecto. En el proceso de inicio de sesión, se realiza una solicitud mediante el método POST, incluyendo los datos de correo electrónico y contraseña. Para cerrar sesión, se ejecuta una petición GET. En el caso de restablecer la contraseña, se efectúa una solicitud mediante POST, proporcionando el correo electrónico al cual se enviará un *token* único para la modificación de la contraseña.

En la **Figura 3. 7** se evidencia la implementación del *endpoint* para reestablecer contraseña, en la **Figura 3. 9** se muestra el correo electrónico enviado para reestablecer la contraseña y en la **Figura 3. 8** se presenta el resultado de la respectiva prueba unitaria. Es relevante notar que el consumo, almacenamiento de registros y validaciones están detalladas en el **ANEXO II**.



```
curl -X 'POST' \
'http://127.0.0.1:8000/send/reset-password' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "email": "learn2hack@gmail.com"
}'

Request URL
http://127.0.0.1:8000/send/reset-password

Server response
Code    Details
200     Response body
{
  "status_code": 200,
  "detail": {
    "message": "Correo de Reestablecer Contraseña enviado"
  }
},
  "headers": null
}
```

Figura 3. 7: *Endpoint* para reestablecer contraseña

```
tests\send_password_reset.py . [100%]
===== 1 passed in 2.37s =====
```

Figura 3. 8: Resultado de la prueba unitaria de reestablecer contraseña


```
tests\send_email_teacher.py . [100%]
===== 1 passed in 2.34s =====
```

Figura 3. 14: Resultado de la prueba unitaria de enviar correo de registro de profesor

3.3 *Sprint 2. Resultados del desarrollo de endpoints para la gestión de los cursos*

En este *sprint* se incluyen las siguientes tareas:

- Generar un *endpoint* para crear un curso.
- Generar *endpoints* para empezar y visualizar un curso.
- Generar un *endpoint* para editar un curso.
- Generar un *endpoint* para editar la imagen de un curso.

Generar un *endpoint* para crear un curso

Se ha desarrollado un *endpoint* dedicado a la creación de cursos, y su implementación involucra varios *endpoints* adicionales. Tres *endpoints* están diseñados para verificar si la información suministrada es correcta. El primero se enfoca en el material adicional del curso, el segundo en la información básica, y el tercero en la verificación del *test* asociado a cada curso. Asimismo, se han creado tres *endpoints* adicionales para almacenar la información después de ser validada. La **Figura 3. 15** detalla la implementación del *endpoint* encargado de verificar el material extra del curso, la **Figura 3. 16** muestra la implementación del *endpoint* para almacenar el material extra del curso, y en la **Figura 3. 17** se presenta el resultado de la correspondiente prueba unitaria del *endpoint* de verificación del material extra. Es relevante destacar que el consumo, almacenamiento de registros y validaciones están detalladas en el **ANEXO II**.

mediante una solicitud POST. El segundo *endpoint* posibilita la eliminación de reseñas de sus cursos mediante una solicitud POST.

En la **Figura 3. 28** se muestra la implementación del *endpoint* para visualizar las reseñas y en la **Figura 3. 29** se presenta el resultado de la respectiva prueba unitaria. Es relevante notar que el consumo, almacenamiento de registros y validaciones están detalladas en el **ANEXO II**.



Figura 3. 28: Endpoint para visualizar reseñas

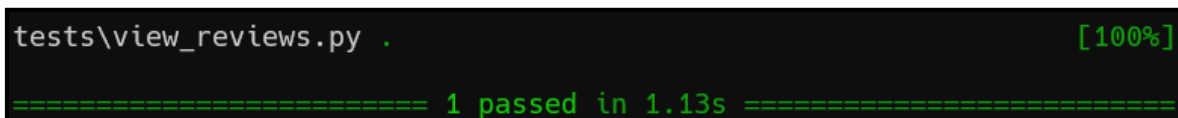


Figura 3. 29: Resultado de la prueba unitaria para visualizar reseñas

Generar *endpoints* para visualizar, eliminar reseñas de todos los cursos

Se han desarrollado *endpoints* destinados al rol de administrador, que facilitan la visualización y eliminación de reseñas asociadas a todos los cursos publicados. El primer *endpoint* permite al administrador listar todas las reseñas de todos los cursos mediante una solicitud POST. Para la eliminación de una reseña, se utiliza una solicitud POST.

En la **Figura 3. 30** se muestra la implementación del *endpoint* para visualizar reseñas y en la **Figura 3. 31** se presenta el resultado de la respectiva prueba unitaria. Es relevante notar que el consumo, almacenamiento de registros y validaciones están detalladas en el **ANEXO II**.



Figura 3. 30: *Endpoint* para visualizar todas las reseñas

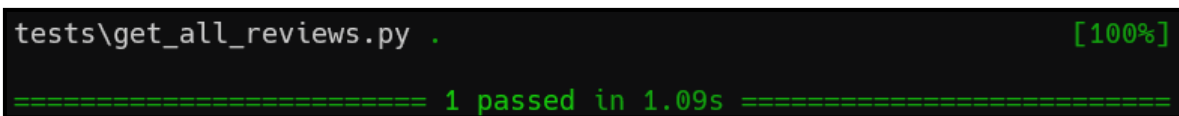


Figura 3. 31: Resultado de la prueba unitaria para visualizar todas las reseñas

3.6 *Sprint 5. Resultados de las pruebas para del backend*

En este *sprint* se incluyen las siguientes pruebas:

- Ejecución de pruebas unitarias.
- Ejecución de pruebas de estrés.
- Ejecución de pruebas de seguridad.

Ejecución de pruebas unitarias

La ejecución de pruebas unitarias es esencial para verificar el correcto funcionamiento de las distintas unidades de código en un proyecto. El propósito fundamental de estas pruebas es asegurar la calidad y seguridad de cada funcionalidad, permitiendo la detección y corrección de errores de manera eficiente, estas pruebas permiten estructurar el proyecto de forma modular [23].

En el proceso de ejecución de las pruebas unitarias, se empleó "PyTest", una herramienta que facilita la automatización de este tipo de pruebas. La **Figura 3. 32** proporciona un fragmento de código para el inicio de sesión, mientras que en la **Figura 3. 33** se muestra el resultado de la prueba unitaria correspondiente. Se destaca que todas las demás pruebas unitarias se detallan en el **ANEXO II**.

```

1  from fastapi.testclient import TestClient
2  from main import app
3
4  client = TestClient(app)
5
6  EMAIL = "teacher@gmail.com"
7  PASSWORD = "Teacher123!$"
8
9  data = {
10     "username": EMAIL,
11     "password": PASSWORD
12 }
13
14 def test_login_user():
15     response = client.post("/login", data=data).json()
16     assert response.get("access_token") ≠ None

```

Figura 3. 32: Código para el test de inicio de sesión

```

tests\login.py . [100%]
===== 1 passed in 1.31s =====

```

Figura 3. 33: Resultado de la prueba unitaria para inicio de sesión

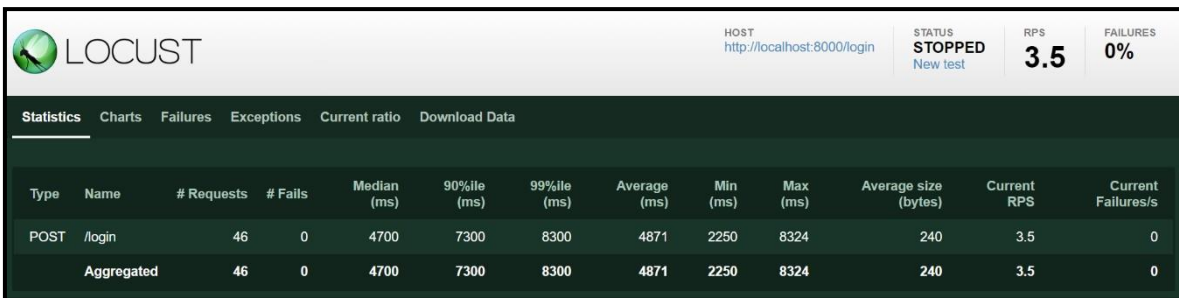
Los resultados obtenidos de las pruebas unitarias proporcionan una sólida validación de la funcionalidad eficiente de los diferentes módulos, lo que demuestra que todas las validaciones necesarias han sido implementadas con éxito, asegurando así la integridad y la fiabilidad del sistema en su conjunto.

Ejecución de pruebas de estrés

La ejecución de pruebas de estrés tiene como objetivo evaluar el desempeño y la funcionalidad del *software* en diferentes escenarios, especialmente al aumentar el número de peticiones, transacciones y usuarios. Este enfoque va más allá de las condiciones de funcionamiento normales, permitiendo a los desarrolladores examinar detenidamente el comportamiento del *software* en situaciones de carga intensa. La realización de pruebas de estrés proporciona información que facilita la detección de errores, la optimización de procesos y la toma de decisiones sobre la escalabilidad del proyecto [24].

Para realizar estas pruebas se usó “Locust”, ya que facilita la creación y automatización de este tipo de pruebas. En la **Figura 3. 34** se presenta el resultado

de una prueba de estrés realizada al *endpoint* de inicio de sesión. En este escenario, se simuló la interacción de 100 usuarios concurrentes.



Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	/login	46	0	4700	7300	8300	4871	2250	8324	240	3.5	0
Aggregated		46	0	4700	7300	8300	4871	2250	8324	240	3.5	0

Figura 3. 34: Resultado de la prueba de estrés de inicio de sesión

Los resultados revelan que no hubo fallos durante la prueba, dado que el porcentaje de errores fue nulo. Esto sugiere que el sistema pudo gestionar la carga simulada sin dificultades, garantizando que todos los usuarios pudieran acceder al *endpoint* de manera satisfactoria.

Ejecución de pruebas de seguridad

La ejecución de pruebas de seguridad es fundamental para garantizar la integridad del *software*, asegurando que la aplicación esté libre de vulnerabilidades, amenazas y riesgos. Estas pruebas se dividen en dos tipos principales: caja blanca y caja negra. Las pruebas de caja blanca se llevan a cabo cuando se posee un conocimiento profundo del funcionamiento del *software*, incluso teniendo acceso al código fuente o a credenciales específicas. Por otro lado, las pruebas de caja negra se realizan sin tener conocimiento previo sobre el *software* [25].

En este caso, las pruebas de seguridad que se ejecutaron son del tipo caja blanca, utilizando herramientas automatizadas como sqlmap. En la **Figura 3. 35** se muestra el resultado de la respectiva prueba de seguridad al *endpoint* de inicio de sesión. Es esencial destacar que todas las pruebas de seguridad están detalladas en el **ANEXO II**.

```

sqlmap-dev python .\sqlmap.py -u http://localhost:8000/login --method POST --data 'grant_type=USERNAME&password=FUZZ&scope=client_id=client_secret=' --batch --dbms=PostgreSQL --level 5 --risk 3 --url https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

```

Figura 3. 35: Resultado de la prueba de seguridad de inicio de sesión

3.7 *Sprint 6. Despliegue del backend*

En este *sprint* se incluyen la siguiente tarea:

- Despliegue del *backend* en Render

Una vez completada la fase de codificación de todos los módulos y con el éxito de todas las pruebas, se procedió al despliegue del *backend* en la plataforma Render. La URL de acceso pública a la documentación de la API es la siguiente:

<https://learn2hack-ws.onrender.com/docs>

4 CONCLUSIONES

En esta sección, se presentan las conclusiones derivadas del desarrollo del *backend* del Trabajo de Integración Curricular:

- Establecer objetivos, alcances y requerimientos de manera clara al iniciar un proyecto es esencial. Estos elementos proporcionan una visión precisa del proyecto a desarrollar, lo que facilita su ejecución y garantiza la alineación con las expectativas del cliente.
- El diseño adecuado de la base de datos es un factor crítico que influye en la eficiencia del proyecto. Una estructura bien definida permite almacenar, organizar y recuperar datos de manera óptima, lo que contribuye a la funcionalidad y rendimiento general del sistema.
- La creación de un API ofrece numerosas ventajas, destacando su capacidad para integrarse con diversas plataformas. Por ejemplo, facilita la expansión del proyecto hacia aplicaciones móviles sin necesidad de realizar cambios significativos en la estructura del *backend*, lo que agiliza el desarrollo y optimiza los recursos.
- La realización de pruebas al finalizar cada *sprint* es un componente importante del desarrollo ágil de *software*. Estas pruebas permiten identificar y resolver errores de manera oportuna, lo que mejora la calidad del producto final y permite la corrección de fallos.
- El despliegue del *backend* en la nube ofrece una serie de ventajas, incluyendo la capacidad de llegar a un gran número de usuarios a nivel mundial, la posibilidad de monitorizar y escalar la aplicación según la demanda, y la adaptabilidad a las necesidades cambiantes del mercado.

5 RECOMENDACIONES

En esta sección, se presentan las recomendaciones a las que se han llegado una vez terminado el desarrollo del *backend* del presente Trabajo de Integración Curricular:

- Para mantener un desarrollo ordenado y evitar prácticas de programación deficientes, es recomendable modularizar el código. Esta práctica facilita la reutilización de código previamente escrito y ayuda a evitar la creación de un código confuso y difícil de mantener.
- La documentación del código es fundamental para comprender la funcionalidad de cada módulo. Esto permite a los desarrolladores entender rápidamente el propósito de cada sección del código y reduce la necesidad de revisar código previamente escrito.
- En proyectos de gran extensión, seguir la estructura predefinida por el *framework* utilizado es esencial. Además, es importante adherirse a una guía de estandarización para escribir un código claro y comprensible, lo que facilita la colaboración entre desarrolladores y el mantenimiento del proyecto a largo plazo.
- Se recomienda utilizar herramientas diseñadas específicamente para la tarea que se necesita realizar. Esto garantiza la compatibilidad y evita errores que podrían resultar en pérdida de tiempo y recursos.
- La utilización de servicios de terceros para satisfacer necesidades complejas del proyecto puede ser beneficioso. Estos servicios proporcionan seguridad, adaptabilidad y una rápida implementación, lo que permite a los equipos de desarrollo concentrarse en aspectos críticos del proyecto sin tener que desarrollar soluciones desde cero.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] OWASP, «OWASP Top Ten,» 12 Septiembre 2019. [En línea]. Available: <https://owasp.org/www-project-top-ten/>. [Último acceso: 13 Noviembre 2023].
- [2] JWT.IO, [En línea]. Available: <https://jwt.io/>. [Último acceso: 13 Noviembre 2023].
- [3] S. Juan, «7 principales amenazas de ciberseguridad para empresas y cómo prevenirlas,» DELTA PROTECT, 20 Abril 2023. [En línea]. Available: <https://www.deltaprotect.com/blog/amenazas-de-ciberseguridad>. [Último acceso: 13 Noviembre 2023].
- [4] IBM, «¿Qué es el desarrollo de software? | IBM,» [En línea]. Available: <https://www.ibm.com/es-es/topics/software-development>. [Último acceso: 14 Noviembre 2023].
- [5] CoderHouse, «¿Qué hace un desarrollador backend?,» 27 Junio 2023. [En línea]. Available: <https://latam.coderhouse.com/blog/desarrollador-backend-perfil-sueldo-estudiar>. [Último acceso: 14 Noviembre 2023].
- [6] Python, «Welcome to Python.org,» [En línea]. Available: <https://www.python.org/>. [Último acceso: 14 Noviembre 2023].
- [7] tiangolo, «FastAPI,» [En línea]. Available: <https://fastapi.tiangolo.com/>. [Último acceso: 14 Noviembre 2023].
- [8] D. A., «¿Qué es JSON?,» 10 Enero 2023. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-json>. [Último acceso: 14 Noviembre 2023].
- [9] L. Miguel, «Qué es Json Web Token y cómo funciona,» 17 Enero 2020. [En línea]. Available: <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>. [Último acceso: 14 Noviembre 2023].

- [10] Amazon Web Services, Inc., «¿Qué es una API RESTful?,» [En línea]. Available: <https://aws.amazon.com/es/what-is/restful-api/>. [Último acceso: 14 Noviembre 2023].
- [11] Cloudinary, «Cloudinary Programmable Media Overview | Cloudinary,» [En línea]. Available: https://cloudinary.com/documentation/programmable_media_overview. [Último acceso: 14 Noviembre 2023].
- [12] QuestionPro, «¿Qué es un estudio de caso y cómo realizarlo?,» [En línea]. Available: <https://www.questionpro.com/blog/es/que-es-un-estudio-de-caso/>. [Último acceso: 18 Noviembre 2023].
- [13] Instituto Tecnológico Superior Quito Metropolitano, «Las 8 metodologías de desarrollo de software,» [En línea]. Available: <https://itsqmet.edu.ec/los-8-tipos-de-metodologia-de-desarrollo-de-software/>. [Último acceso: 22 Noviembre 2023].
- [14] Red Hat, «¿Qué es la metodología ágil?,» 19 Julio 2022. [En línea]. Available: <https://www.redhat.com/es/devops/what-is-agile-methodology>. [Último acceso: 22 Noviembre 2023].
- [15] Amazon Web Services, Inc., «¿En qué consiste Scrum? - Explicación sobre la metodología Scrum - AWS,» [En línea]. Available: <https://aws.amazon.com/es/what-is/scrum/>. [Último acceso: 14 Noviembre 2023].
- [16] H. Chandler, «Artefactos del scrum ágil,» Atlassian, [En línea]. Available: <https://www.atlassian.com/es/agile/scrum/artifacts>. [Último acceso: 22 Noviembre 2023].
- [17] M. Rehkopf, «Historias de usuario con ejemplos y plantilla,» Atlassian, [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/user-stories>. [Último acceso: 22 Noviembre 2023].

- [18] D. Radigan, «Backlog del producto: qué es y cómo crearlo,» Atlassian, [En línea]. Available: <https://www.atlassian.com/es/agile/scrum/backlogs>. [Último acceso: 22 Noviembre 2023].
- [19] integratit, «Sprint y Sprint Backlog: puntos esenciales de SCRUM,» [En línea]. Available: <https://integratit.com.mx/blog/sprint-y-sprint-backlog/>. [Último acceso: 27 Noviembre 2023].
- [20] Edraw, «Diagrama de Arquitectura de Software,» 16 Marzo 2023. [En línea]. Available: <https://www.edrawsoft.com/es/software-architecture.html>. [Último acceso: 27 Noviembre 2023].
- [21] MDN Web Docs, «MVC - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web | MDN,» [En línea]. Available: <https://developer.mozilla.org/es/docs/Glossary/MVC>. [Último acceso: 28 Noviembre 2023].
- [22] Gobierno de España, «11 librerías para crear visualizaciones de datos,» 03 Mayo 2022. [En línea]. Available: <https://datos.gob.es/es/blog/11-librerias-para-crear-visualizaciones-de-datos>. [Último acceso: 16 Diciembre 2023].
- [23] AWS Amazon, «¿Qué son las pruebas unitarias?,» [En línea]. Available: <https://aws.amazon.com/es/what-is/unit-testing/>. [Último acceso: 27 Diciembre 2023].
- [24] KeepCoding Team, «¿Qué son las pruebas de estrés de software?,» 18 Febrero 2023. [En línea]. Available: <https://keepcoding.io/blog/que-son-pruebas-de-estres-de-software/>. [Último acceso: 27 Diciembre 2023].
- [25] R. Herraiz, «¿Caja negra o caja blanca?,» 28 Enero 2021. [En línea]. Available: <https://www.irium.es/post/caja-negra-o-caja-blanca-descubre-qu%C3%A9-es-un-test-de-intrusi%C3%B3n-o-hacking-%C3%A9tico>. [Último acceso: 27 Diciembre 2023].

7 ANEXOS

A continuación, se presenta cada uno de los Anexos que se ha realizado para el desarrollo del *backend*:

ANEXO I. Certificado de Originalidad

ANEXO II. Manual de Usuario

ANEXO III. Manual de Instalación

ANEXO IV. Credenciales de acceso y despliegue.

ANEXO I



**ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"**

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 12 de febrero de 2024

De mi consideración:

Yo, VANESSA KATHERINE GUEVARA BALAREZO, en calidad de Director del Trabajo de Integración Curricular titulado **DESARROLLO DE UN BACKEND** asociado al **DESARROLLO DE UNA PLATAFORMA WEB DE APRENDIZAJE EN CIBERSEGURIDAD "LEARN2HACK"** elaborado por el estudiante **EDISON PAUL RIVERA MICHILENA** de la carrera en **TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE**, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito secciones: Descripción del componente desarrollado, Metodología, Resultados, Conclusiones y Recomendaciones (sin anexos), producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,



VANESSA KATHERINE
GUEVARA BALAREZO

Vanessa Guevara
Docente Ocasional a Tiempo Completo
ESFOT

ANEXO II

Recopilación de Requerimientos

A continuación, en la **Tabla 1** se listan los requerimientos con los que el proyecto de *software* cumple.

Tabla 1: Recopilación de requerimientos

RECOPIACIÓN DE REQUERIMIENTOS		
Tipo de sistema	ID-RR	Enunciado Del Ítem
<i>backend</i>	RR001	Como usuario estudiante necesita varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Registrar Cuenta. • Verificar Cuenta. • Eliminar Cuenta.
	RR002	Como usuario estudiante, profesor y administrador necesitan generar varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Iniciar Sesión. • Cerrar Sesión. • Reestablecer Contraseña.
	RR003	Como usuario estudiante, profesor y administrador necesitan generar un <i>endpoint</i> para: <ul style="list-style-type: none"> • Modificar su perfil.
	RR004	Como usuario administrador necesita generar un <i>endpoint</i> para: <ul style="list-style-type: none"> • Enviar correo para que el profesor se registre.
	RR005	Como usuario profesor necesita varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Crear curso. • Subir archivo HTML. • Crear preguntas. • Almacenar un laboratorio, imágenes y archivo en formato PDF.
	RR006	Como usuario profesor necesita un <i>endpoint</i> para: <ul style="list-style-type: none"> • Editar curso.

<i>backend</i>	RR007	Como usuario estudiante necesita varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Visualizar todos los cursos. • Filtrar cursos por dificultad y palabras clave. • Visualizar cursos completados.
	RR008	Como usuario estudiante necesita varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Empezar un curso. • Visualizar contenido del curso. • Comprobar respuesta.
	RR009	Como usuario estudiante necesita varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Visualizar reseñas del curso. • Crear reseña del curso. • Eliminar reseña del curso. • Editar reseña del curso.
	RR010	Como usuario profesor necesita varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Visualizar reseñas de sus cursos. • Eliminar reseñas de sus cursos.
	RR011	Como usuario administrador necesita varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Visualizar y eliminar reseñas de todos los cursos.
	RR012	Como usuario administrador necesita varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Editar la imagen de cada curso.

Historias de Usuario

Luego de recopilar toda la información necesaria de la etapa de Recopilación de Requerimientos, se inicia con la creación de las historias de usuario, numeradas desde la **Tabla 2** hasta la **Tabla 13**.

Tabla 2: Historia de Usuario 1 – Registrar, eliminar y verificar cuenta

HISTORIA DE USUARIO	
Identificador: HU-001	Usuario: Estudiante
Nombre Historia: Registrar, eliminar y verificar cuenta.	

Prioridad en el negocio: Alta	Riesgo en el desarrollo: Media
Iteración asignada: 1	
Responsable: Edison Rivera	
Descripción: Se necesitan generar varios <i>endpoints</i> para el perfil estudiante, las acciones a realizar son: <ul style="list-style-type: none"> • Registrar Cuenta. • Verificar Cuenta. • Eliminar Cuenta. 	
Observación: Para verificar una cuenta se envía un correo electrónico al usuario estudiante al momento de registrarse.	

Tabla 3: Historia de Usuario 2 – Iniciar sesión, cerrar sesión y reestablecer contraseña

HISTORIA DE USUARIO	
Identificador: HU-002	Usuario: Estudiante, Profesor, Administrador
Nombre Historia: Iniciar sesión, cerrar sesión y reestablecer contraseña.	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Media
Iteración asignada: 1	
Responsable: Edison Rivera	
Descripción: Se necesitan generar varios <i>endpoints</i> para los perfiles estudiante, profesor, la acciones a realizar son: <ul style="list-style-type: none"> • Iniciar Sesión. • Cerrar Sesión. • Reestablecer Contraseña. 	
Observación: Los usuarios deberán completar el proceso de registro antes de tener la capacidad de cerrar sesión. Además, para restablecer la contraseña, se envía un correo electrónico al usuario, asegurando así un proceso seguro y verificable de recuperación de contraseñas.	

Tabla 4: Historia de Usuario 3 – Modificar perfil

HISTORIA DE USUARIO	
Identificador: HU-003	Usuario: Estudiante, Profesor, Administrador
Nombre Historia: Modificar perfil.	
Prioridad en el negocio: Media.	Riesgo en el desarrollo: Bajo
Iteración asignada: 1	
Responsable: Edison Rivera	
Descripción: Se necesita generar un <i>endpoint</i> para los perfiles estudiante, profesor y administrador, la acción a realizar es: <ul style="list-style-type: none"> • Modificar su perfil. 	
Observación: Podrán editar todos los campos a excepción del correo electrónico.	

Tabla 5: Historia de Usuario 4 – Enviar correo para que el profesor se registre

HISTORIA DE USUARIO	
Identificador: HU-004	Usuario: Administrador
Nombre Historia: Enviar correo para que el profesor se registre.	
Prioridad en el negocio: Alta.	Riesgo en el desarrollo: Alta
Iteración asignada: 1	
Responsable: Edison Rivera	
Descripción: Se necesita generar un <i>endpoint</i> para el perfil administrador, la acción a realizar es: <ul style="list-style-type: none"> • Enviar correo para que el profesor se registre. 	
Observación: El usuario administrador deberá indicar el correo del profesor, al cual se enviará un <i>token</i> único que servirá para el registro.	

Tabla 6: Historia de Usuario 5 – Crear un curso

HISTORIA DE USUARIO	
Identificador: HU-005	Usuario: Profesor
Nombre Historia: Crear curso.	
Prioridad en el negocio: Alta.	Riesgo en el desarrollo: Alta
Iteración asignada: 2	

Responsable: Edison Rivera
<p>Descripción: Se necesitan generar varios <i>endpoints</i> para el perfil profesor, las acciones a realizar son:</p> <ul style="list-style-type: none"> • Crear curso. • Subir archivo HTML. • Crear preguntas. • Almacenar un laboratorio, imágenes y archivo en formato PDF.
<p>Observación: Para registrarse deberá recibir un correo y en la sección de crear un curso el profesor podrá crear un archivo en formato <i>Markdown</i>, el cual será transformado automáticamente a <i>HTML</i> en la plataforma. Además, el profesor tendrá la capacidad de cargar imágenes utilizadas en el curso a <i>Cloudinary</i> y todo el material necesario para el curso. Se habilitará un apartado para crear un máximo de 4 preguntas abiertas.</p>

Tabla 7: Historia de Usuario 6 – Editar un curso

HISTORIA DE USUARIO	
Identificador: HU-006	Usuario: Profesor
Nombre Historia: Editar un curso	
Prioridad en el negocio: Alta.	Riesgo en el desarrollo: Alta
Iteración asignada: 2	
Responsable: Edison Rivera	
<p>Descripción: Se necesitan generar varios <i>endpoints</i> para el perfil profesor, las acciones a realizar son:</p> <ul style="list-style-type: none"> • Editar curso. 	
<p>Observación: El profesor podrá editar todos los aspectos de su curso, a excepción, de las preguntas previamente establecidas.</p>	

Tabla 8: Historia de Usuario 7 - Visualizar y filtrar todos los cursos

HISTORIA DE USUARIO	
Identificador: HU-007	Usuario: Estudiante.
Nombre Historia: Visualizar y filtrar todos los cursos.	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Bajo

Iteración asignada: 3
Responsable: Edison Rivera
<p>Descripción: Se necesitan generar varios <i>endpoints</i> para el perfil estudiante, las acciones a realizar son:</p> <ul style="list-style-type: none"> • Visualizar todos los cursos. • Filtrar cursos por dificultad y palabras clave. • Visualizar cursos completados.
<p>Observación: El estudiante podrá ver sus cursos finalizados y completados, además de poder filtrar por palabras clave.</p>

Tabla 9: Historia de Usuario 8 – Empezar y visualizar curso

HISTORIA DE USUARIO	
Identificador: HU-008	Usuario: Estudiante.
Nombre Historia: Empezar y visualizar curso.	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Alta
Iteración asignada: 2	
Responsable: Edison Rivera	
<p>Descripción: Se necesitan generar varios <i>endpoints</i> para el perfil estudiante, las acciones a realizar son:</p> <ul style="list-style-type: none"> • Empezar un curso. • Visualizar contenido del curso. • Comprobar respuesta. 	
<p>Observación: El estudiante puede iniciar un curso para ver su contenido, además, podrá comprobar la respuesta de las preguntas de ese curso.</p>	

Tabla 10: Historia de Usuario 9 – Visualizar, crear, eliminar y editar reseñas

HISTORIA DE USUARIO	
Identificador: HU-009	Usuario: Estudiante.
Nombre Historia: Visualizar, crear, eliminar y editar reseñas.	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Baja
Iteración asignada: 4	
Responsable: Edison Rivera	

<p>Descripción: Se necesitan generar varios <i>endpoints</i> para el perfil estudiante, las acciones a realizar son:</p> <ul style="list-style-type: none"> • Visualizar reseñas del curso. • Crear reseña del curso. • Eliminar reseña del curso. • Editar reseña del curso.
<p>Observación: El estudiante puede ver todas las reseñas sobre un curso, y cuando finalice un curso podrá dejar una única reseña la cual podrá eliminar y editar.</p>

Tabla 11: Historia de Usuario 10 – Visualizar y eliminar reseñas

HISTORIA DE USUARIO	
Identificador: HU-010	Usuario: Profesor
Nombre Historia: Visualizar y eliminar reseñas de los cursos creados.	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Baja
Iteración asignada: 4	
Responsable: Edison Rivera	
<p>Descripción: Se necesitan generar varios <i>endpoints</i> para el perfil profesor, las acciones a realizar son:</p> <ul style="list-style-type: none"> • Visualizar reseñas de sus cursos. • Eliminar reseñas de sus cursos. 	
Observación: El profesor puede ver todas las reseñas de sus cursos y podrá eliminarlas.	

Tabla 12: Historia de Usuario 11 – Visualizar y eliminar reseñas de todos los cursos

HISTORIA DE USUARIO	
Identificador: HU-011	Usuario: Administrador
Nombre Historia: Visualizar y eliminar reseñas de todos los cursos	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Baja
Iteración asignada: 4	
Responsable: Edison Rivera	
<p>Descripción: Se necesitan generar varios <i>endpoints</i> para el perfil administrador, las acciones a realizar son:</p> <ul style="list-style-type: none"> • Visualizar y eliminar reseñas de todos los cursos. 	

Observación: El administrador puede ver las reseñas de todos los cursos y podrá eliminarlas.

Tabla 13: Historia de Usuario 12 – Editar imagen de un curso

HISTORIA DE USUARIO	
Identificador: HU-012	Usuario: Administrador
Nombre Historia: Editar la imagen de un curso.	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Baja
Iteración asignada: 2	
Responsable: Edison Rivera	
Descripción: Se necesitan generar varios <i>endpoints</i> para el perfil administrador, las acciones a realizar son: <ul style="list-style-type: none"> • Editar la imagen de cada curso. 	
Observación: El administrador puede editar la imagen de cada curso publicado.	

Product Backlog

A continuación, en la **Tabla 14**, se presenta por completo el *Product Backlog*

Tabla 14: Product Backlog

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU – 001	Registrar, eliminar y verificar cuenta.	1	Finalizado	Alta
HU – 002	Iniciar sesión, cerrar sesión y reestablecer contraseña	1	Finalizado	Alta
HU – 003	Modificar perfil	1	Finalizado	Media
HU – 004	Enviar correo para que el profesor se registre	1	Finalizado	Alta
HU – 005	Crear curso	2	Finalizado	Alta
HU – 006	Editar un curso	2	Finalizado	Alta
HU – 007	Visualizar y filtrar todos los cursos.	3	Finalizado	Media
HU – 008	Empezar y visualizar curso.	2	Finalizado	Alta

HU – 009	Visualizar, crear, eliminar y editar reseñas	4	Finalizado	Media
HU – 010	Visualizar y eliminar reseñas de los cursos creados	4	Finalizado	Media
HU – 011	Visualizar y eliminar reseñas de todos los cursos	4	Finalizado	Media
HU – 012	Editar la imagen de un curso.	2	Finalizado	Media

Sprint Backlog

En la **Tabla 15** se detallan los *sprints* que se han desarrollado por el *backend*, incluyendo todas las actividades realizadas e indicando el tiempo establecido para cada una de ellas.

Tabla 15: *Sprint Backlog*

ELABORACIÓN DE <i>SPRINT BACKLOG</i>						
ID-SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREA	TIEMPO ESTIMADO
SB-000	Configuración del ambiente de desarrollo	N/A	N/A	N/A	<ul style="list-style-type: none"> Recopilación de requerimientos del proyecto. Definición de la estructura de directorios para el <i>backend</i>. Diseño y creación de la base de datos. Definición y alcance de los roles de usuario. 	10H
SB-001	Diseño e implementación de <i>endpoints</i> para el registro, inicio de sesión, verificación	Módulo de registro, verificación y eliminación de cuenta.	HU-001	Registrar, eliminar y verificar cuenta.	<ul style="list-style-type: none"> Diseño de <i>endpoints</i> para registrar, eliminar y verificar cuenta. Validación de datos. Registro y consulta en la base de datos. 	40H

	de cuenta y autenticación	Módulo de inicio y cierre de sesión, y reestablecer contraseña	HU-002	Iniciar sesión, cerrar sesión y reestablecer contraseña	<ul style="list-style-type: none"> • Diseño de <i>endpoints</i> para iniciar y cerrar sesión y reestablecer la contraseña. • Validación de datos. • Consulta en la base de datos. • Enviar correo para reestablecer la contraseña. 	
		Módulo de modificar perfil.	HU-003	Modificar perfil	<ul style="list-style-type: none"> • Diseño de un <i>endpoint</i> para modificar perfil. • Validación de datos. • Consulta en la base de datos. • Registro en la base de datos. 	
		Módulo correo para el registro del profesor	HU-004	Enviar correo para que el profesor se registre	<ul style="list-style-type: none"> • Diseño de un <i>endpoint</i> para enviar un correo para que el profesor se registre. • Validación de datos. • Registro en la base de datos. 	
SB-002	Diseño e implementación de <i>endpoints</i> para gestión de los cursos	Módulo para crear un curso	HU-005	Crear un curso	<ul style="list-style-type: none"> • Diseño e implementación de <i>endpoints</i> para crear un curso, subir archivo HTML, crear preguntas y almacenar laboratorio, imagen y archivo PDF. 	100H

					<ul style="list-style-type: none"> • Validación de datos. • Almacenar ficheros en un servicio de tercero. • Registro en la base de datos. 	
		Módulo para editar curso	HU-006	Editar curso	<ul style="list-style-type: none"> • Diseño de <i>endpoints</i> para editar un curso. • Validación de datos. • Registro en la base de datos. • Consulta en la base de datos. • Almacenar ficheros en un servicio de tercero. 	
		Módulo para empezar, visualizar un curso, comprobar respuestas y terminar un curso	HU-008	Empezar y visualizar curso.	<ul style="list-style-type: none"> • Diseño de <i>endpoints</i> para empezar y visualizar un curso y comprobar respuestas. • Consulta en la base de datos. • Registro en la base de datos. • Validación de datos. 	

		Módulo para editar la imagen de un curso	HU-012	Editar la imagen de un curso.	<ul style="list-style-type: none"> • Diseño de un <i>endpoint</i> para editar la imagen de un curso. • Validación de datos. • Registro en la base de datos. • Almacenar ficheros en un servicio tercero. 	
SB-003	Diseño e implementación de <i>endpoints</i> para filtrar y visualizar cursos	Módulo para visualizar y filtrar cursos	HU-007	Visualizar y filtrar todos los cursos.	<ul style="list-style-type: none"> • Diseño de <i>endpoints</i> para visualizar todos los cursos y filtrarlos. • Consulta en la base de datos. • Aplicar filtros por: dificultad, cursos completados y por palabras clave. 	10H
SB-004	Diseño e implementación de <i>endpoints</i> para gestión de reseñas	Módulo para visualizar, crear, eliminar y editar sus propias reseñas.	HU-009	Visualizar, crear, eliminar y editar reseñas	<ul style="list-style-type: none"> • Diseño de <i>endpoints</i> para visualizar, crear, eliminar y editar reseñas. • Validación de datos. • Registro en la base de datos. • Consulta en la base de datos. 	40H

		Módulo para visualizar reseñas de todos sus cursos creados y eliminarlas	HU-010	Visualizar y eliminar reseñas de los cursos creados	<ul style="list-style-type: none"> • Diseño de <i>endpoints</i> para visualizar y eliminar reseñas. • Validación de datos. • Registro en la base de datos. • Consulta en la base de datos. 	
		Módulo para visualizar reseñas de todos los cursos y eliminarlas	HU-011	Visualizar, y eliminar reseñas de todos los cursos	<ul style="list-style-type: none"> • Diseño de <i>endpoints</i> para visualizar reseñas de todos los cursos y eliminarlas. • Validación de datos. • Registro en la base de datos. • Consulta en la base de datos. 	
SB-005	Pruebas del <i>backend</i>	<ul style="list-style-type: none"> • Pruebas unitarias. • Pruebas de estrés. • Pruebas de seguridad. 				10H
SB-006	Despliegue del <i>backend</i>	<ul style="list-style-type: none"> • Despliegue del <i>backend</i> en Render 				10H
Documentación		<ul style="list-style-type: none"> • Trabajo de integración curricular. • Anexos. 				20H
TOTAL						240

Ejecución de Pruebas

Al concluir cada *sprint*, se realizaron pruebas para asegurar el óptimo funcionamiento de cada módulo.

Pruebas unitarias

Se llevaron a cabo pruebas unitarias en el *backend* con el objetivo de evaluar el código, identificar posibles errores y corregirlos. La primera prueba examina un fragmento de código responsable de editar el perfil de usuario, presentado en la **Figura 1**, el resultado de la prueba unitaria se presenta en la **Figura 2**. La segunda prueba se centra en un bloque de código diseñado para verificar la validez de la respuesta a una pregunta de un curso, como se muestra en la **Figura 3**, el resultado de la prueba unitaria se presenta en la **Figura 4**. La tercera prueba evalúa un fragmento de código encargado de editar una reseña, presentado en la **Figura 5**, el resultado de la prueba unitaria se presenta en la **Figura 6**.

```

data_login = { "username": "teacher@gmail.com", "password": "Teacher123!$" }

data_edit_profile = {
    "nickname": "Teacher EsTX",
    "description": "Soy el nuevo profesor de la plataforma"
}

def test_edit_profile_user():
    response_login = client.post("/login", data=data_login).json()
    jwt_token = response_login["access_token"]
    assert jwt_token != None

    headers = {"Authorization": f"Bearer {jwt_token}"}

    response_edit = client.put("/user", data=data_edit_profile, headers=headers).json()
    assert response_edit["status_code"] == 200

    response_user_profile = client.get("/user", headers=headers).json()
    assert response_user_profile["detail"]["nickname"] == data_edit_profile["nickname"]
    assert response_user_profile["detail"]["description"] == data_edit_profile["description"]

```

Figura 1: Código para el test de editar perfil

```

tests\test_update_profile.py . [100%]
===== 1 passed in 1.52s =====

```

Figura 2: Resultado de la prueba unitaria para editar perfil


```

data_login = { "username": "student@gmail.com", "password": "Student123!$" }

data_check_answer = {
    "id_module": "1",
    "index_test": "1",
    "answer": "Ataque"
}

def test_check_answer():
    response_login = client.post("/login", data=data_login).json()
    jwt_token = response_login["access_token"]
    assert jwt_token != None

    headers = {"Authorization": f"Bearer {jwt_token}"}

    response_check_answer = client.post("/check/answer", headers=headers, json=data_check_answer).json()
    assert response_check_answer.get("status_code") == 200

```

Figura 3: Código para el test de verificar repuesta

```

tests\test_check_answer.py . [100%]
===== 1 passed in 1.39s =====

```

Figura 4: Resultado de la prueba unitaria para verificar respuesta

```

data_edit_review = {
    "stars_review": "3",
    "comment_review": "string",
    "id_module": "1"
}

def test_edit_review():
    response_login = client.post("/login", data=data_login).json()
    jwt_token = response_login["access_token"]
    assert jwt_token != None

    headers = {"Authorization": f"Bearer {jwt_token}"}

    response_edit_review = client.put("/edit/review", headers=headers, json=data_edit_review).json()
    assert response_edit_review.get("status_code") == 200

```

Figura 5: Código para el test de editar reseña

```

tests\test_edit_review.py . [100%]
===== 1 passed in 1.33s =====

```

Figura 6: Resultado de la prueba unitaria para editar reseña

Pruebas de estrés

Se llevaron a cabo pruebas de estrés en el *backend* con el fin de evaluar el rendimiento del *software* frente a un alto volumen de solicitudes. En la **Figura 7** se muestra el resultado del *endpoint* para filtrar cursos, en la **Figura 8** se muestra el

resultado del *endpoint* para ver un curso y en la **Figura 9** se muestra el resultado del *endpoint* para conseguir todas las reseñas de todos los cursos.



The screenshot shows the Locust web interface with the following data:

LOCUST		HOST	STATUS	RPS	FAILURES							
		http://localhost:8000/	STOPPED New test	10.3	0%							
Statistics Charts Failures Exceptions Current ratio Download Data												
Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	/courses	232	0	2014	2100	2100	2046	2014	2076	277	10.3	0
	Aggregated	232	0	2014	2100	2100	2046	2014	2076	277	10.3	0

Figura 7: Resultado de la prueba de estrés para filtrar cursos



The screenshot shows the Locust web interface with the following data:

LOCUST		HOST	STATUS	RPS	FAILURES							
		http://localhost:8000/	STOPPED New test	10.8	0%							
Statistics Charts Failures Exceptions Current ratio Download Data												
Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/course/1	255	0	2013	2100	2100	2039	2013	2072	598	10.8	0
	Aggregated	255	0	2013	2100	2100	2039	2013	2072	598	10.8	0

Figura 8: Resultado de la prueba de estrés para ver un curso



The screenshot shows the Locust web interface with the following data:

LOCUST		HOST	STATUS	RPS	FAILURES							
		http://localhost:8000/	STOPPED New test	11.2	0%							
Statistics Charts Failures Exceptions Current ratio Download Data												
Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/get/all/reviews	254	0	2009	2100	2100	2042	2009	2072	78	11.2	0
	Aggregated	254	0	2009	2100	2100	2042	2009	2072	78	11.2	0

Figura 9: Resultado de la prueba de estrés para ver todas las reseñas

Pruebas de seguridad

Con el objetivo de asegurar que la aplicación esté libre de vulnerabilidades, se llevaron a cabo pruebas de seguridad en el *backend*. En la **Figura 10**, se presenta la petición utilizada con sqlmap en el *endpoint* de filtrar cursos, y el resultado de la prueba de inyección SQL se muestra en la **Figura 11**. Para la segunda prueba, se evaluó el *endpoint* de previsualización de un curso; en la **Figura 12** se muestra la petición usada con sqlmap y en la **Figura 13** se muestra el resultado. Finalmente,

ANEXO III

Para obtener información sobre la implementación de cada *endpoint* y una descripción completa de cada rol en este proyecto, se ha preparado un Manual de Usuario del *backend*. Para acceder al manual es necesario ingresar a la siguiente URL:

<https://www.youtube.com/watch?v=OEA1H7MiHv0>

Este recurso proporciona una guía completa para comprender y utilizar eficientemente todas las funcionalidades del *backend* en el proyecto

ANEXO IV

En esta sección del documento, se proporcionan las credenciales de acceso para el *backend*, la URL del *backend* desplegado, todo el código fuente y los pasos para la instalación en el README del repositorio.

Enlace del *backend* desplegado

Se puede acceder a través del siguiente URL:

<https://learn2hack-ws.onrender.com/docs>

Credenciales del usuario administrador

- Correo del administrador: eclearn2hack@gmail.com
- Contraseña: **ba9GnIEMzMbi4walmvqr!**

Credenciales del usuario profesor

- Correo del profesor: profesorlearnhack@gmail.com
- Contraseña: **Vrcyyg3qOpzYr62kMkzr!\$**

Credenciales del usuario estudiante

- Correo del estudiante: studentlearnhack@gmail.com
- Contraseña: **oL6xcSTKprDYxlchRbw!\$**

Repositorio del *backend*

El código del proyecto se aloja en GitHub, el cual puede ser accedido a través del siguiente enlace:

<https://github.com/edisonrivera/Learn2Hack>