



**Universidad de
Concepción**
Facultad de Ingeniería
Ingeniería Civil Informática

Nueva versión de la plataforma PA3P con evaluación por pruebas unitarias

*Memoria presentada para la obtención del título de
Ingeniera Civil Informática*

POR

IVONNE ELENA FLORES ROA

Profesor Patrocinante: Geoffrey Hecht
26 de marzo de 2024, Concepción

© Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

Dedico este trabajo a la memoria
de mi abuelita Marina Arias
Machuca, quien siempre será
recordada con cariño y
admiración. Su legado perdurará
en nuestros corazones.

Agradecimientos

Quiero expresar mi profundo agradecimiento a las personas que me han brindado su apoyo incondicional a lo largo de este camino:

Agradezco a mi familia, Jessica Roa Arias y Constanza Flores Roa, por su constante apoyo durante mis años de estudio. Su respaldo y sacrificio han sido fundamentales para mí.

A mi profesor patrocinante, Geoffrey Hecht, quiero agradecerle por su inquebrantable apoyo, paciencia y sabios consejos. Su guía ha sido fundamental en mi crecimiento académico y personal.

Además, quiero extender mi gratitud a Mauricio Echavarría por su constante ayuda y gestiones en el servidor y el desarrollo de esta Memoria de Título. Su colaboración ha sido invaluable para llevar a cabo este proyecto.

Agradezco a mis amigos más cercanos de la Universidad de Concepción, Leonardo, Patricio, Felipe, Vicente y Matías, por su constante apoyo incondicional durante estos años de estudio.

A todas estas personas, gracias por creer en mí, por estar a mi lado y por hacer posible este logro. Sin ustedes, este camino habría sido mucho más difícil.

Índice

1. Introducción	7
2. Antecedentes y Problemática	9
2.1. Plataforma de Apoyo al Aprendizaje Activo de Programación Orientada a Objetos (PA3P)	9
2.2. Identificación de Problemas y Limitaciones en la Plataforma	10
3. Propuesta de mejora	12
3.1. Descripción de la propuesta y beneficios esperados	12
3.2. Objetivos	17
3.2.1. Objetivos Generales	17
3.2.2. Objetivos Específicos	17
4. Metodología	19
4.1. Enfoque de Trabajo	19
4.2. Análisis de Requisitos Funcionales	19
4.3. Requisitos no Funcionales	20
4.4. Enfoque de diseño	20
4.4.1. Diseño del Backend:	22
4.4.2. Diseño del Frontend:	23
4.4.3. Comunicación entre el Frontend y el Backend:	23
4.4.4. Arquitectura del proyecto	24
4.5. Desarrollo y Configuración de la Base de Datos	24
4.5.1. Modelo Entidad Relación	26
4.5.2. Selección de Tecnologías de Bases de Datos	26
4.6. Entregable	28

5. Implementación	29
5.1. Base de datos	29
5.2. Estructura del Proyecto	29
5.3. Ejecución de Test Unitarios	31
5.3.1. Organización del Proyecto Maven	31
5.3.2. Integración de Test Unitarios	32
5.3.3. Configuración de Archivos del Estudiante en el Backend	35
5.3.4. Proceso de Retroalimentación	35
5.4. Desarrollo de la página web	37
5.5. Interfaz de Usuario	37
5.5.1. Login	37
5.5.2. Principales Vistas	38
6. Validación y Resultados	44
6.1. Validación	44
6.2. Resultados	48
7. Conclusiones	52
7.1. Impacto en la Experiencia de Enseñanza y Aprendizaje	52
7.2. Limitaciones	53
8. Recomendaciones	54
8.1. Trabajo a futuro	54
9. Anexo	57
9.1. Documentación Casos de Uso	57
9.2. Comentarios de la encuesta	61
9.3. Imágenes	63

Resumen

La Plataforma de Apoyo al Aprendizaje Activo de Programación Orientada a Objetos (PA3P) del Departamento de Informática de la Universidad de Concepción se ha actualizado para mejorar la evaluación de los estudiantes en la asignatura Programación II (Desarrollo Orientado al Objetivo) en Java. Esta actualización incluye la implementación de test unitarios para una evaluación más precisa del código de los alumnos.

Los test unitarios representan un cambio significativo en la plataforma, permitiendo una evaluación más rigurosa y detallada del rendimiento estudiantil. Se realizaron cambios en la interfaz y el diseño para mejorar la experiencia de docentes y alumnos, junto con la incorporación de un cuestionario para recopilar las percepciones de los estudiantes.

Los resultados muestran mejoras sustanciales en la plataforma, con una experiencia de enseñanza y aprendizaje más efectiva y enriquecedora para todos los usuarios. La retroalimentación de los usuarios respalda la efectividad de las mejoras implementadas, con una mayor satisfacción y preferencia por la nueva versión de PA3P para futuros cursos.

1. Introducción

El Departamento de Informática de la Universidad de Concepción dispone de una plataforma denominada "Plataforma de Apoyo al Aprendizaje Activo de Programación Orientada a Objetos"(PA3P) [1], desarrollada por el profesor Jorge López, la cual se utiliza para la formación y evaluación de los estudiantes de la asignatura Programación II en el lenguaje Java[2].

El docente, a través de esta herramienta, examina automáticamente el código de los alumnos y monitorea el progreso y el avance de todos los estudiantes en los ejercicios asignados en el lenguaje de programación Java[2]. Los alumnos cuentan con la oportunidad de analizar tanto el enunciado del ejercicio como su correspondiente diagrama UML, al ingresar a la plataforma a través de su número de matrícula y posteriormente cargar un archivo .java con la solución pertinente. La plataforma automáticamente evalúa el archivo y envía un mensaje indicando si el usuario ha cumplido las metas correctamente.

Para la memoria de título, se propone una nueva versión de la Plataforma de Apoyo al Aprendizaje Activo de Programación Orientada a Objetos. Esta versión incorpora cambios significativos, especialmente en la forma en que se analiza la corrección del código proporcionado por el alumno.

Se lleva a cabo una reestructuración en la interfaz, así como un cambio en el diseño y las funcionalidades tanto para el docente como para el alumno. La incorporación de test unitarios en PA3P permite una validación más precisa del código desarrollado por los estudiantes, ya que se realizan pruebas específicas en cada uno de los métodos implementados en su código. Esto permite una retroalimentación más rápida y eficiente, ayudando a los estudiantes a mejorar su comprensión de la programación orientada a objetos en Java y a desarrollar habilidades en la escritura de código de calidad.

Además, en el proceso de validación, se introducirá un cuestionario diseñado es-

pecíficamente para recopilar las percepciones y sugerencias de los estudiantes. Este enfoque tiene como propósito enriquecer el diseño y mejorar la eficacia de la plataforma, al integrar la valiosa retroalimentación. Como complemento a estas medidas, se llevarán a cabo pruebas detalladas en cada uno de los métodos implementados.

2. Antecedentes y Problemática

2.1. Plataforma de Apoyo al Aprendizaje Activo de Programación Orientada a Objetos (PA3P)

La "Plataforma de Apoyo al Aprendizaje Activo de Programación Orientada a Objetos"(PA3P) [1] representa una herramienta fundamental en el ámbito educativo del Departamento de Informática de la Universidad de Concepción. Desarrollada por el profesor Jorge López, la plataforma PA3P está diseñada para la formación y evaluación de los estudiantes en la asignatura de Programación II (Desarrollo Orientado al Objeto), específicamente centrada en el lenguaje Java.

Esta plataforma proporciona a los docentes la capacidad de examinar automáticamente el código generado por los estudiantes, monitorear el progreso y avance en los ejercicios asignados, y ofrecer retroalimentación instantánea. Por otro lado, los estudiantes tienen la oportunidad de acceder al enunciado del ejercicio y a su correspondiente diagrama UML, cargar sus soluciones en archivos .java, y recibir evaluaciones automáticas.



Figura 1: Plataforma PA3P

2.2. Identificación de Problemas y Limitaciones en la Plataforma

- Diversidad de Lenguajes de Programación:** La inclusión de varios lenguajes, como PHP, C y Bash, ha generado complejidades en el mantenimiento y actualización de la plataforma, afectando su estabilidad y cohesión.
- Ausencia de Base de Datos Dedicada:** La carencia de una base de datos específica para almacenar ejercicios y soluciones implica cargar directamente en el sistema de archivos, generando ineficiencias en la gestión de contenidos y la integridad de datos.
- Secuencialidad y Limitaciones en Ejercicios:** La presentación no secuencial de ejercicios y la restricción en la cantidad de preguntas activas por sesión limitan la adaptabilidad y flexibilidad del proceso educativo.

Selecciona una Actividad a habilitar

id	descripcion	problemas															
99	actividad vacia																
100	ejercicios Java no OO	1	2	3	4	5	6	9001	9002	9003							
101	ejercicios creacion primeras clases	7	8	9	10	11	12	13	14								
102	ejercicios reutilizacion, composicion y encapsulamiento	19	20	21	22	23	24	25	26	27	28						
103	ejercicios reutilizacion, composicion, herencia	38	39	40	41	42	43										
150	test basico de contener versus conocer y pabellon de pintura de autos	51	52	53	54	55											
1000	revisión tarea 1: sistema comprador-expendedor	101	102	103	104	105	106	107	108	109	110						
1200	test PA3P composicion polimorfica y delegacion	200	201	202	203	204	205	206	207								
1300	test PA3P conexion V invertida entre clases, polimorfismo formas geometricas	300	301	302	303												
5000	polimorfismo de bebidas 1 problema	5001															
6000	clases pelota, caja y pelota, tubo y pelotas	6001	6002	6003													
7000	clases de objetos que entregan secuencias circulares de digitos	7001	7002	7003	7004												
8000	caminos de ceros sobre clases con arreglos de 12x24	8001	8002	8003	8004	8005	8006	8007	8008								
9000	actividad para prueba de problemas nuevos	9001	9002	9003	9004	9005	9006	9007	9008	9009	9010						
9010	actividad con operaciones multiples arreglos tipo struct desde main	9004	9005	9006	9007	9008	9009	9010	9011	9012	9013	9014	9015	9016	9017		
9020	con clases que imprimen y metodos que trabaja a domicilio	9021	9022														
9030	problemas tipo GUI de matriz de char	9031	9032	9033	9034	9035	9036										
10000	camino sobre ceros de 12x24	19	20	10001	10002	10003	10004	10005	10006	10007	10008						

102 ✓ | Habilitar

Volver a menu principal ✓

Figura 2: Listado de ejercicios PA3P

- Proceso de Inicio de Sesión Inseguro:** El sistema de inicio de sesión basado en el número de matrícula plantea riesgos de seguridad, exigiendo mejoras para salvaguardar la integridad de los datos y la privacidad del usuario.
- Confusión en Retroalimentación del Código:** La retroalimentación del código presenta mensajes con colores contradictorios, lo que confunde a los estudiantes y dificulta la interpretación de los resultados. La elección de colores durante la revisión ha sido identificada como una fuente de confusión significativa. La

combinación de colores utilizada crea ambigüedades, lo que dificulta la identificación eficiente de errores en las soluciones propuestas y complica el proceso de corrección.

- **Dificultad en el Análisis de los Resultados:** La retroalimentación actual en la plataforma [1] es confusa para profesores y estudiantes. Los mensajes de error son a veces poco claros y conducen a correcciones incorrectas. Por ejemplo, se menciona un error en el último carácter de la salida, pero solo falta un salto de línea al final. En la figura 3, también podemos observar que se menciona una clase *res* aunque sea interna al sistema. La falta de explicaciones detalladas limita la comprensión del código, afectando el proceso de aprendizaje y corrección. Es crucial mejorar la claridad en la retroalimentación para optimizar el proceso educativo.
- **Problemas de Sintaxis y Filosofía de Programación:** La plataforma presenta limitaciones al abordar problemas de sintaxis y filosofía de programación, detectadas mediante análisis estático del código a través de comandos Bash. Estas limitaciones, no presentes en la programación Java convencional, incluyen restricciones en la colocación de llaves y paréntesis. Desde una perspectiva filosófica, la plataforma a menudo impone el uso específico de variables o métodos, excluyendo otras opciones, lo que puede entrar en conflicto con el principio de encapsulamiento y limitar la libertad del estudiante para estructurar su código.
- **Complejidad para añadir nuevos ejercicios:** La creación de nuevos ejercicios representa un desafío significativo para los profesores, ya que implica implementar análisis estáticos específicos para cada ejercicio. Esto agrega una capa adicional de complejidad al proceso de diseño de ejercicios, ya que anticipar los posibles errores de los estudiantes y configurar reglas de análisis consume tiempo adicional en la preparación del material didáctico.

```

Analisis dinamico:
Error: Could not find or load main class res

Resultados dinamicos:

|   |      |      |
|---|------|------|
| n | Tuyo | Debe |
|   |      |      |

Analisis estatico:
main ✓
<=Nok3=> en metodo main: para probar debes ejecutar var.llenar( .. )
<=Nok4=> en metodo main: para probar debes ejecutar var.verificar( .. )
<=metodo Balde=> no encontrado
<=Nok2=> en clase Balde: no esta bien el prototipo o falta el metodo llenar
<=Nok2=> en clase Balde: no esta bien el prototipo o falta el metodo verificar
<-Av2-> en metodo constructor Balde: debes revisar declaracion del metodo constructor, no debe tener parametro ni codigo
<=Nok2=> en clase Balde: no esta la propiedad privada entera que representa los litros almacenados en le balde

Compilacion:
Balde.java:6: error: ';' expected
    int balde += cantidad;
                ^
Balde.java:6: error: not a statement
    int balde += cantidad;
                ^
2 errors
Error de compilacion en Balde.java : NOK!
NOK!

```

Figura 3: Ejemplo: Retroalimentación plataforma PA3P

3. Propuesta de mejora

3.1. Descripción de la propuesta y beneficios esperados

La propuesta de mejora consiste en el desarrollo de una aplicación llamada CodeBending, diseñada para abordar diversas limitaciones y problemas identificados en la plataforma actual. CodeBending busca mejorar la experiencia de enseñanza y aprendizaje en la asignatura de Programación II (Desarrollo orientado al objeto) mediante la implementación de un entorno más eficiente y centrado en el aprendizaje activo. Cada solución se presenta en relación con el problema correspondiente, destacando los cambios específicos que se planean implementar.

- Diversidad de Lenguajes de Programación:** Con el objetivo de abordar la diversidad de lenguajes de programación, se propone llevar a cabo una reestructuración completa de la interfaz, desarrollando un nuevo software que utilice un enfoque más moderno y eficiente. Se empleará un único lenguaje de programación, Python, para el backend, respaldado por un framework de desarrollo robusto. Además, se implementará un framework específico para el frontend, optimizando así la cohesión del sistema.

Esta iniciativa no solo busca superar las complejidades asociadas con la diversidad de lenguajes, sino que también apunta a mejorar la estabilidad y facilitar futuras actualizaciones, a diferencia de la plataforma actual que carece de documentación.

Beneficios esperados:

- Mejora en la estabilidad y eficiencia en las actualizaciones.
 - Facilitación de la incorporación fluida de nuevas funcionalidades.
 - Mayor consistencia y cohesión en la plataforma.
- **Ausencia de Base de Datos Dedicada:** Proponemos la implementación de una base de datos especializada para almacenar ejercicios, soluciones y datos relacionados con estudiantes y docentes. Emplear tecnologías de bases de datos modernas para optimizar la gestión de contenidos y asegurar la integridad de los datos. Esta iniciativa facilitará la organización, búsqueda y actualización de ejercicios, contribuyendo a mejorar la eficiencia del sistema.

Beneficios esperados:

- Mejora en la eficiencia del sistema.
 - Facilitación de la organización y búsqueda de ejercicios.
 - Mayor rapidez en la gestión de contenidos por parte de los docentes.
- **Secuencialidad y Limitaciones en Ejercicios:** En respuesta a las limitaciones presentes en la presentación secuencial de ejercicios y las restricciones en la cantidad de preguntas activas por sesión, se propone una solución integral. Esta solución implica la creación de una serie de ejercicios que, junto con la base de datos, mostrará los ejercicios ordenados y agrupados, brindando a los docentes la capacidad de presentarlos a los estudiantes de manera organizada y adaptable. Se maximizará la flexibilidad al permitir que los docentes activen o desacti-

ven la serie de ejercicios según sus necesidades específicas, proporcionando así una experiencia educativa más dinámica y adaptable a los ritmos individuales de aprendizaje.

Beneficios esperados:

- Mayor flexibilidad en la presentación de ejercicios.
 - Adaptabilidad a los ritmos individuales de aprendizaje.
 - Experiencia educativa más dinámica.
- **Proceso de Inicio de Sesión Inseguro:** Con el objetivo de mejorar significativamente la seguridad del acceso, se realizará la implementación de un sistema de inicio de sesión más robusto, que incluirá autenticación y el almacenamiento seguro de usuarios en la base de datos. Estas medidas están diseñadas para fortalecer la integridad de los datos y garantizar la privacidad de los usuarios.

Beneficios esperados:

- Mejora significativa en la seguridad del acceso.
 - Garantía de la integridad de los datos.
 - Protección de la privacidad de los usuarios.
- **Confusión en Retroalimentación del Código:** Con el objetivo de perfeccionar la experiencia de aprendizaje, se busca mejorar la claridad y especificidad de los mensajes de error generados durante la ejecución del código. Se implementará una estrategia que proporciona descripciones detalladas, sugerencias y enlaces directos a recursos educativos pertinentes. Esta medida tiene como propósito ayudar a los estudiantes a comprender y corregir los errores de manera eficiente, elevando así la calidad y utilidad de la retroalimentación.

Además, se planea implementar una paleta de colores coherente y fácil de interpretar, especialmente adaptada para indicar el estado de los ejercicios. Estos

colores estarán diseñados para representar visualmente si un ejercicio ha sido aprobado o contiene errores y ha sido rechazado. Esta codificación visual intuitiva busca simplificar la identificación y comprensión de los resultados por parte de los estudiantes, mejorando así la experiencia de aprendizaje en el entorno de programación.

Beneficios esperados:

- Mejora en la calidad y utilidad de la retroalimentación.
 - Simplificación en la identificación de resultados.
 - Mejora en la experiencia de aprendizaje.
- **Dificultad en el Análisis de los Resultados:** Para superar esta limitación, se propone una transformación integral en el proceso de revisión de ejercicios. Se sugiere evolucionar de la revisión actual, estática y dinámica, hacia un enfoque de prueba con pruebas unitarias utilizando la herramienta Maven. Con esta modificación, los errores se visualizarán de manera diferente, ya que el docente seleccionará la salida que se mostrará al estudiante. Este cambio busca proporcionar una retroalimentación más detallada y personalizada, mejorando significativamente la comprensión de los errores y permitiendo a los estudiantes abordar de manera más efectiva sus áreas de mejora.

También se destaca que estas pruebas permiten ser mucho más dinámicas en comparación con la metodología anterior, que se limitaba a comprobar la salida y análisis estático.

Ambas soluciones, la implementación de pruebas unitarias y la redefinición de la paleta de colores en la retroalimentación del código, trabajarán de manera conjunta para mejorar significativamente la visualización de errores. La combinación de estas estrategias proporcionará un enfoque más claro y comprensible

en la identificación y corrección de errores, optimizando así la experiencia de aprendizaje en el entorno de programación.

Beneficios esperados:

- Retroalimentación más detallada y personalizada.
- Mejora en la comprensión de errores.
- Facilitación para abordar áreas de mejora.

- **Problemas de Sintaxis y Filosofía de Programación:** Como continuación a la solución propuesta para la dificultad en el análisis de los resultados, la problemática de los problemas de sintaxis y filosofía de programación se abordará mediante la implementación de pruebas unitarias en conjunto con la estrategia anterior.

En lugar de depender únicamente del análisis estático y dinámico mediante comandos Bash, se propone incorporar pruebas unitarias en el proceso de revisión de ejercicios. Esto permitirá una evaluación más rigurosa y detallada del código de los estudiantes, superando las limitaciones impuestas por el enfoque estático actual. La introducción de pruebas unitarias ofrecerá una mayor flexibilidad, ya que el docente podrá definir conjuntos de pruebas específicos para cada ejercicio. Esto abordará la limitación filosófica al permitir una gama más amplia de estructuras de código, siempre que se cumplan con los requisitos de las pruebas unitarias establecidas. Las pruebas unitarias proporcionarán una retroalimentación más dinámica y específica, permitiendo al docente seleccionar qué salidas mostrar al estudiante. Esto mejorará la comprensión de los errores, ya que la retroalimentación será más directa y personalizada.

Beneficios esperados:

- Mejora en la detección de problemas de sintaxis y filosofía de programación.

- Mayor flexibilidad en la estructuración del código por parte de los estudiantes.

3.2. Objetivos

3.2.1. Objetivos Generales

Optimizar la experiencia de enseñanza y aprendizaje en la asignatura de Programación II (Desarrollo orientado al objeto) mediante la implementación de un entorno más eficiente y centrado en el aprendizaje activo. El objetivo principal es proporcionar a docentes y estudiantes herramientas que permitan una evaluación continua y detallada del rendimiento estudiantil. Se busca mejorar significativamente la retroalimentación, agilizar la corrección de ejercicios y reforzar la comprensión de la programación orientada a objetos en Java. Este enfoque tiene como finalidad contribuir al desarrollo de habilidades sólidas en la escritura de código de calidad, promoviendo así un ambiente propicio para el crecimiento académico y profesional de los estudiantes.

3.2.2. Objetivos Específicos

1. Desarrollar una nueva interfaz utilizando Python como único lenguaje de programación para el backend, respaldado por un framework robusto.
2. Implementar un framework específico para el frontend, mejorando la cohesión y estabilidad del sistema.
3. Optimizar la interfaz para facilitar futuras actualizaciones y la incorporación de nuevas funcionalidades.
4. Implementar una base de datos especializada para almacenar ejercicios, soluciones y datos relacionados con estudiantes y docentes.
5. Utilizar tecnologías modernas de bases de datos para mejorar la gestión de contenidos y garantizar la integridad de los datos.

6. Crear una serie de ejercicios estructurados, permitiendo a los docentes presentarlos de manera ordenada y adaptable.
7. Integrar la serie de ejercicios con la base de datos, maximizando la flexibilidad para activar o desactivar según las necesidades específicas del docente.
8. Establecer un sistema de evaluación basado en calificaciones para los problemas planteados.
9. Creación de un sistema de gestión de grupos que permita la realización de varias series de problemas simultáneamente.
10. Implementar un sistema de inicio de sesión más robusto con autenticación avanzada.
11. Almacenar de forma segura los usuarios en la base de datos, fortaleciendo la integridad de los datos y garantizando la privacidad de los usuarios.
12. Mejorar la claridad y especificidad de los mensajes de error durante la ejecución del código.
13. Implementar una paleta de colores coherente y fácil de interpretar para indicar el estado de los ejercicios.
14. Transformar el proceso de revisión de ejercicios hacia un enfoque de prueba con pruebas unitarias utilizando la herramienta Maven.
15. Redefinir la paleta de colores en la retroalimentación del código para proporcionar una representación visual clara del estado de los ejercicios.

4. Metodología

4.1. Enfoque de Trabajo

Durante la elaboración de esta Memoria de Título, se optó por la implementación de la metodología ágil, basada en un enfoque de desarrollo iterativo y progresivo de la aplicación. Para asegurar una dirección efectiva, se llevaron a cabo reuniones semanales con el profesor patrocinante, con el propósito de recibir retroalimentación valiosa y evaluar el avance continuo del desarrollo. Este método permitió una adaptación ágil a los cambios y una mejora constante en la calidad del trabajo desarrollado.

4.2. Análisis de Requisitos Funcionales

El diagrama de casos de uso de la Figura 4 representa las principales funcionalidades de la plataforma identificados en el primer sprint. A esto se añaden los siguiente requisitos funcionales :

- La interfaz debe permitir a los docentes presentar una serie de ejercicios de manera ordenada y adaptable.
- El sistema debe incluir una sección para que los estudiantes ingresen al sistema, accedan a los ejercicios y reciban retroalimentación.
- El sistema debe emplear un framework específico para el frontend, asegurando cohesión y estabilidad en el sistema.
- La plataforma deberá implementar una base de datos especializada para almacenar de manera eficiente ejercicios, soluciones y datos relacionados con estudiantes y docentes.
- Se requiere la creación de un sistema de gestión de grupos que permita la realización simultánea de varias series de problemas, brindando flexibilidad a los docentes en la presentación de contenidos.

- Es necesario almacenar de forma segura la información de los usuarios en la base de datos, fortaleciendo la integridad de los datos y asegurando la privacidad de los usuarios.
- El sistema debe permitir a los docentes crear series de ejercicios y presentarlos de manera organizada y adaptable.
- El sistema debe establecer un sistema de evaluación basado en calificaciones para los problemas planteados.

4.3. Requisitos no Funcionales

- La interfaz de usuario debe ser intuitiva y fácil de usar para docentes y estudiantes.
- La plataforma debe ser accesible desde diferentes dispositivos y tamaños de pantalla.
- El tiempo de carga de la plataforma y la respuesta a las acciones del usuario deben ser rápidos.
- Deben establecerse políticas de acceso y autenticación robustas.
- El sistema debe ser capaz de manejar un número creciente de usuarios y datos.
- Debe ser posible escalar la infraestructura fácilmente según sea necesario.
- La plataforma debe ser compatible con múltiples navegadores web y sistemas operativos.

4.4. Enfoque de diseño

En esta fase, se detallará la estrategia y los principios de diseño que se aplicaron en el desarrollo de la plataforma. Se abordarán aspectos como la arquitectura propuesta y cómo se organizaron los diferentes componentes.

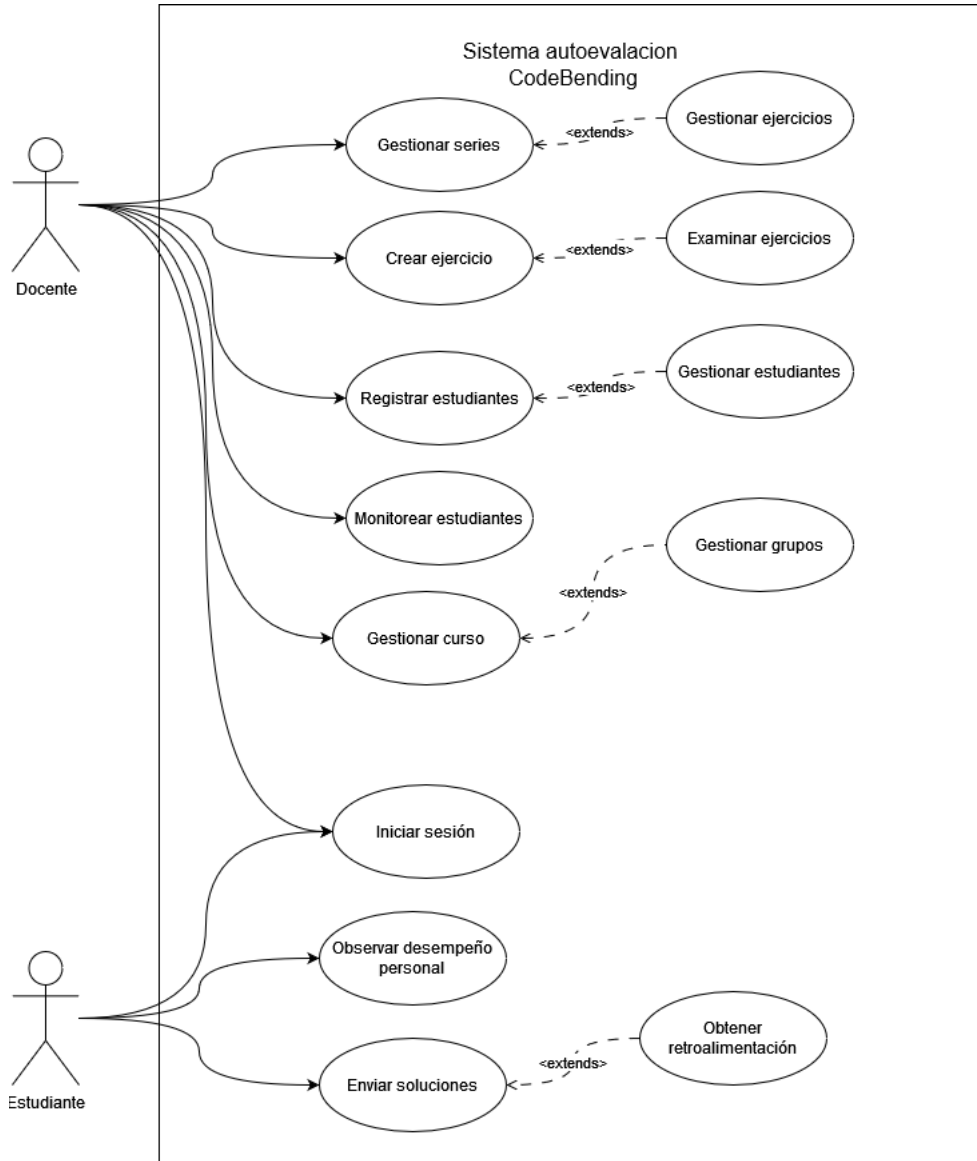


Figura 4: Esquema Casos de Uso

4.4.1. Diseño del Backend:

Al concebir el diseño del backend para la plataforma, se tomaron decisiones clave en cuanto a la tecnología y la arquitectura a utilizar. Entre ellas, se optó por Flask como el framework principal debido a su versatilidad y eficiencia en el desarrollo de aplicaciones web en Python. Esta elección se basó en el hecho de que ya se tenía cierto conocimiento previo en Flask. Sin embargo, más allá de la elección del framework, se introdujeron prácticas y herramientas adicionales para asegurar la calidad del código y facilitar su gestión.

Una de estas prácticas importantes es la implementación de pruebas unitarias, una técnica esencial en el desarrollo de software que permite evaluar individualmente unidades de código para garantizar su correcto funcionamiento. En este contexto, se adoptó JUnit como la herramienta principal para escribir estas pruebas. JUnit, siendo una biblioteca de pruebas unitarias para el lenguaje de programación Java, brinda una estructura robusta y organizada para la creación y ejecución de pruebas, lo que contribuye significativamente a la fiabilidad y estabilidad del sistema.

Además, para gestionar la ejecución de pruebas unitarias, se integró Maven[3] como una herramienta de gestión de proyectos. Maven es conocido por su capacidad para gestionar dependencias, compilación y generación de informes de proyectos Java, y se convirtió en una parte integral del flujo de trabajo del backend. Utilizando Maven, se facilitó la organización del proyecto, se automatizó el proceso de compilación a través de subprocessos y la ejecución de pruebas, lo que resultó en una mayor eficiencia y consistencia en el desarrollo. En las configuraciones de un proyecto Maven se realizan también las configuraciones de las dependencias de JUnit para las pruebas unitarias.

La combinación de estas tecnologías y prácticas ofrece una base sólida para el desarrollo y mantenimiento de la plataforma, asegurando su calidad y escalabilidad a

largo plazo.

La estructura planificada para el proyecto se organiza de la siguiente manera: los ejercicios propuestos se almacenan en una carpeta específica, mientras que en el mismo nivel se encuentran los ejercicios presentados por los estudiantes y una carpeta designada como plantilla de proyecto Java.

4.4.2. Diseño del Frontend:

En el momento de iniciar el desarrollo, la elección del framework fue un paso crítico, se optó por Bootstrap 4.6 debido a su estabilidad comprobada. La decisión se basó en varios factores, como la robustez del framework, su popularidad en la comunidad de desarrollo web, y la disponibilidad de recursos y documentación detallada. Esta elección aseguró un desarrollo eficiente y confiable del frontend, proporcionando una base sólida para la interfaz de usuario. El frontend se estructuró de manera clara y organizada para facilitar el desarrollo y mantenimiento. Se crearon dos carpetas principales que albergan la lógica y recursos visuales del frontend. Dentro de estas carpetas, se encuentran los tres templates principales: uno dedicado al proceso de inicio de sesión, otro diseñado para la interfaz del estudiante y el tercero destinado a la interfaz del docente. Estos templates actúan como moldes fundamentales desde los cuales se extienden todas las demás vistas. Esta estructura modular facilita la expansión del sistema y permite un desarrollo ágil y escalable.

Esta organización modular garantiza coherencia en el diseño y la funcionalidad a lo largo de la plataforma, al tiempo que proporciona flexibilidad para introducir nuevas características o realizar actualizaciones en secciones específicas.

4.4.3. Comunicación entre el Frontend y el Backend:

En la implementación de la plataforma, la comunicación entre el frontend y el backend se lleva a cabo principalmente mediante el uso de plantillas Jinja2 en Flask. El

backend utiliza el **motor de plantillas Jinja2** para renderizar las páginas HTML dinámicamente. Las variables de Jinja2, como `{{ variable }}`, se utilizan para incrustar datos dinámicos generados por el backend en las páginas web.

Se implementa el **manejo de formularios HTML** en el frontend, y el backend procesa las solicitudes POST correspondientes. Esto permite la interacción bidireccional entre el usuario y la aplicación.

Es importante destacar que para transmitir datos entre el frontend y el backend. La comunicación se realiza principalmente mediante la generación dinámica de contenido HTML en el lado del servidor y la manipulación de formularios.

La plataforma utiliza un enfoque tradicional, donde las acciones del usuario desencadenan solicitudes HTTP al servidor y este responde con la generación de nuevas páginas HTML. Este enfoque fue seleccionado debido a su simplicidad y adecuación a los requisitos y características específicos del proyecto en esta etapa de desarrollo.

4.4.4. Arquitectura del proyecto

En el siguiente apartado se exhibirá el modelo C4 de la plataforma, abarcando sus tres niveles principales: diagrama de contexto del sistema (figura 5), diagrama de (figura 6) y diagrama de componentes (figura 7).

También se incluye un diagrama que detalla la estructura de directorios y archivos en la figura 9, con el propósito de brindar una comprensión más clara sobre la organización de este proyecto.

4.5. Desarrollo y Configuración de la Base de Datos

Esta sección proporciona una visión detallada del diseño y la configuración de la base de datos subyacente que respalda la infraestructura de la plataforma propuesta. La base de datos desempeña un papel crucial en el almacenamiento y gestión eficiente de información crítica, incluyendo ejercicios, soluciones, datos de estudiantes y docentes.

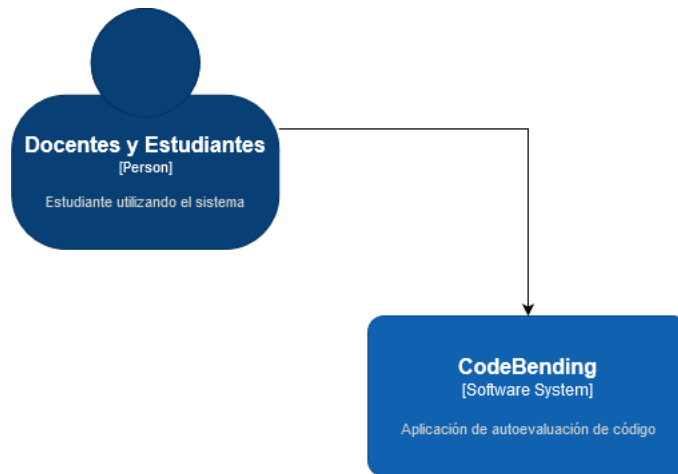


Figura 5: Diagrama de contexto

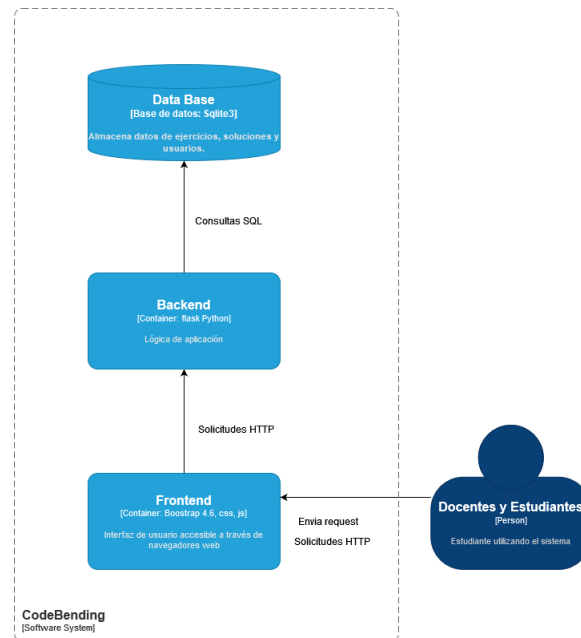


Figura 6: Diagrama de Contenedores

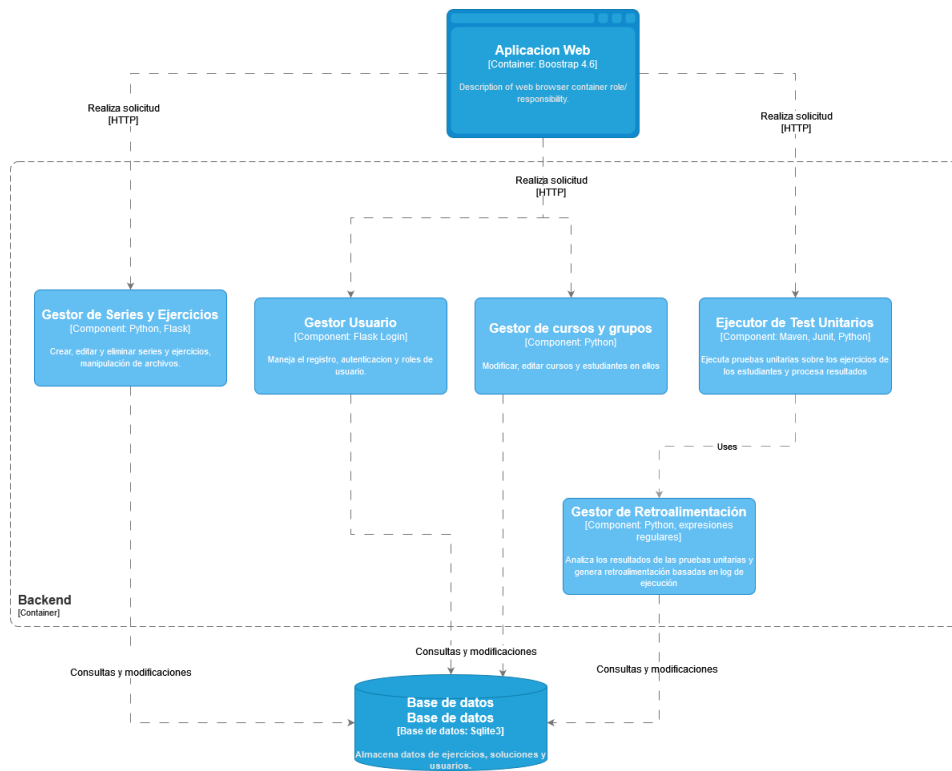


Figura 7: Diagrama de Componentes Backend

4.5.1. Modelo Entidad Relación

Como respuesta a la ausencia inicial de una base de datos, fundamental para facilitar el almacenamiento y organización de ejercicios y soluciones, se presenta a continuación el Modelo Entidad-Relación (MER) que servirá como base para la creación de la base de datos.

4.5.2. Selección de Tecnologías de Bases de Datos

Considerando la adopción de Python como el lenguaje de programación principal, se decidió utilizar SQLAlchemy[4] para desarrollar los modelos de la base de datos. SQLAlchemy es reconocido como un mapeador relacional de objetos, proporciona una notable flexibilidad y solidez en la gestión de datos SQL. Esta decisión se reforzó mediante el uso de SQLite 3, una alternativa liviana que facilita almacenar la base de datos

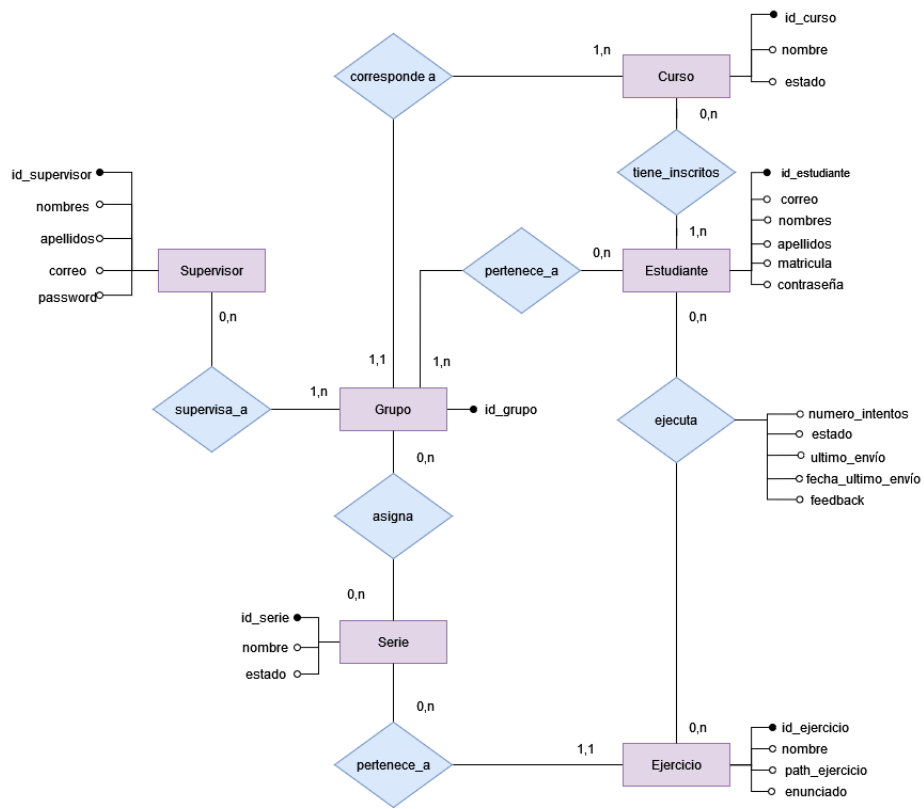


Figura 8: Modelo Entidad Relación
 Imagen en tamaño completo en Anexo

completa como un único archivo estándar en la máquina host.

4.6. Entregable

Como resultado final, el proyecto estará alojado en un servidor proporcionado por el departamento y se exportará a través de GitHub. La plataforma, junto con el desarrollo correspondiente, estará disponible en esta máquina, alojada por el Departamento de Informática. El proyecto podrá ser instalado en cualquier computadora utilizando el repositorio de GitHub.

Actualmente, la plataforma reside en una máquina virtual con las siguientes especificaciones: 2 GB de RAM, un procesador de 2 núcleos y ejecutando el sistema operativo Ubuntu 20.04. Dado que los recursos son limitados, se utiliza Gunicorn, que es un servidor web HTTP WSGI para Python, para poder ejecutar el proyecto con 2 workers (procesos de trabajo).

Gunicorn fue elegido debido a su capacidad para manejar aplicaciones web Python de manera eficiente y escalable. Este servidor procesa 2 workers para las solicitudes HTTP entrantes, lo que significa que puede manejar simultáneamente hasta 2 solicitudes HTTP. Sin embargo, las solicitudes adicionales se colocan en una cola (queue) y se procesan en orden una vez que los workers están disponibles.

La elección de 2 workers se basa en la optimización de los recursos disponibles en la máquina virtual. Esto permite distribuir la carga de trabajo de manera más efectiva y garantizar un rendimiento aceptable dentro de las limitaciones de hardware actuales. Además, se eligió este número de workers en función de la cantidad de núcleos que tiene el servidor, lo que ayuda a aprovechar al máximo el potencial de procesamiento.

Es importante destacar que, aunque la instancia inicial cuenta con recursos limitados, se prevé migrar a un servidor con capacidades superiores para que se haga el uso definitivo por el docente.

5. Implementación

5.1. Base de datos

En la sección anterior, tal como se indicó, se diseñaron los modelos de la base de datos utilizando SQLAlchemy y siguiendo el esquema MER. Posteriormente, se detallan a continuación las clases más relevantes de estos modelos. Con la creación exitosa de los modelos, el siguiente paso consiste en la instancia de la base de datos para establecer y poner en funcionamiento la estructura diseñada. Este proceso integral de desarrollo y configuración de la base de datos constituye un componente fundamental en la implementación del sistema.

5.2. Estructura del Proyecto

En esta sección se dará mas detalle de la estructura de carpetas y jerarquía del proyecto, utilizando como referencia la figura 9. La estructura del proyecto se organiza en diversas carpetas, cada una desempeñando un papel crucial en el funcionamiento general.

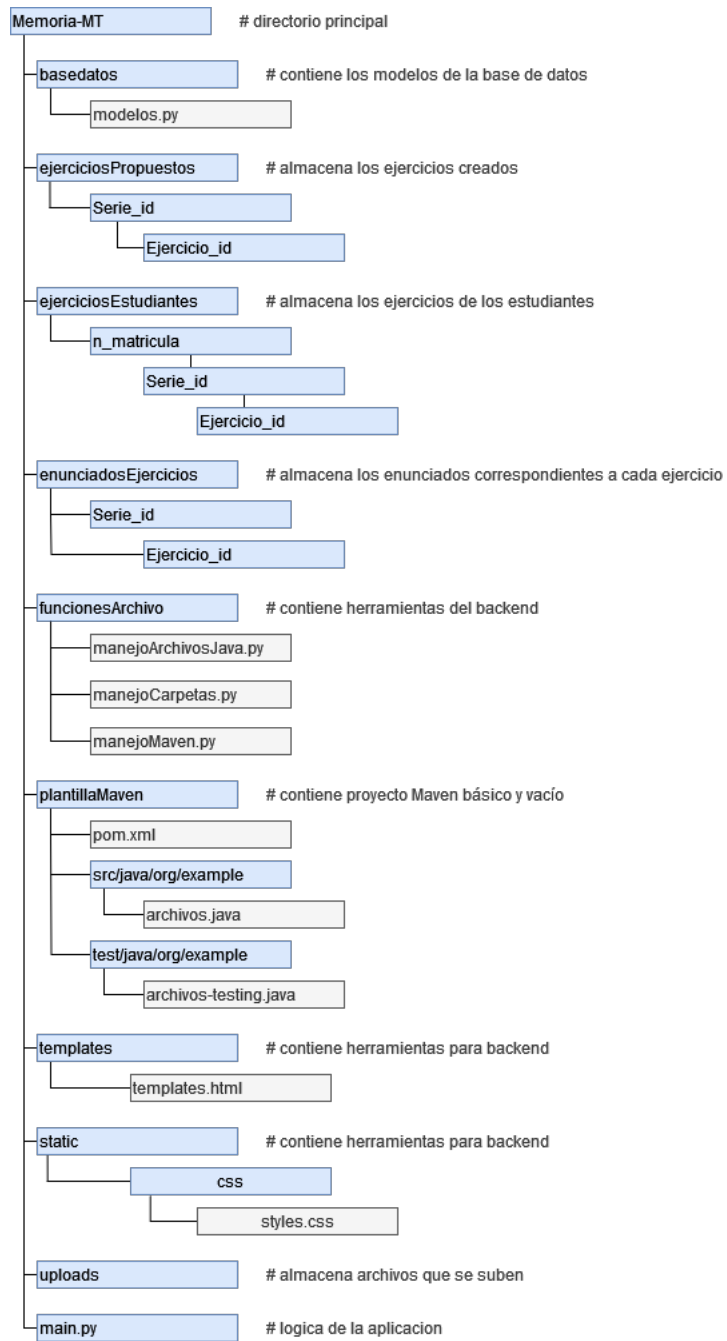


Figura 9: Estructura directorios y archivos

5.3. Ejecución de Test Unitarios

La transición hacia la ejecución de test unitarios marca una transformación significativa en la plataforma, representando un cambio sustancial respecto a las anteriores pruebas estáticas y dinámicas en diversos lenguajes de programación. Ahora, la evaluación se realiza ejecutando proyectos Java que incorporan pruebas unitarias escritas en el mismo lenguaje.

El proceso de implementación comenzó con el desarrollo de la ejecución mediante subprocesos de un proyecto con características idénticas a las utilizadas en clases. Este proyecto se creó en IntelliJ[5] utilizando la distribución Maven. Posteriormente, se diseñaron funciones en Python para la ejecución de los test unitarios en el proyecto, aprovechando las capacidades de Maven.

5.3.1. Organización del Proyecto Maven

El proyecto Maven se organiza de la siguiente manera: se tiene una carpeta específica con el nombre del proyecto. En el interior de esta carpeta, se encuentra otra denominada *'src'*, que contiene los componentes esenciales del proyecto. A su vez, al mismo nivel que la carpeta *'src'*, se encuentra el archivo *pom.xml*. Este archivo centraliza las características y configuraciones necesarias para la gestión efectiva del proyecto mediante Maven.

Dentro de la carpeta *'src'*, se establecen dos rutas clave:

- La ruta *'test/java/org/example'* alberga los archivos *.java* que contienen los test unitarios destinados a evaluar el código generado por el estudiante.
- Simultáneamente, en la ruta *'main/java/org/example'*, se encuentran los archivos *.java* que componen los ejercicios creados por el estudiante como parte de su práctica en la plataforma.

El archivo *pom.xml*, que significa "Project Object Model"(Modelo de Objeto de Pro-

yecto), es una representación XML de un proyecto Maven. Va más allá de ser simplemente una colección de archivos de código; abarca la configuración del proyecto, la organización y las licencias, la URL donde reside el proyecto, las dependencias del proyecto y todos los demás elementos que intervienen para dar vida al código. Es un lugar centralizado para todos los aspectos relacionados con el proyecto.

5.3.2. Integración de Test Unitarios

Para integrar la ejecución de test unitarios al proyecto de memoria de título se hace mediante subprocesos, utilizando el comando `'mvn clean test'` a la carpeta que contiene el proyecto Maven. Esto genera un registro de consola que detalla posibles errores en la ejecución de los test unitarios.

En caso de encontrar que el código no pasa los test unitarios, estos se registran en un log de consola. En caso contrario, si es exitoso y aprueba todos los test unitarios, también se registra un log indicando que todos los test unitarios han sido aprobados con éxito.

Para extraer la información relevante de este registro de consola, se emplean funciones en Python que utilizan expresiones regulares (Python RegeX)[6]. Estas funciones permiten recopilar la información más importante para la retroalimentación, facilitando así la identificación y corrección de posibles errores en el código de los estudiantes.

En la figura 10 se presenta el registro del log generado cuando se ejecuta comandos en la terminal, fuera del contexto de la aplicación. Este log, que captura la salida de la consola, contiene una cantidad extensa de texto que puede resultar abrumador para los estudiantes. Por tanto, se opta por seleccionar únicamente una parte relevante del registro para mostrarla. En la siguiente figura, se muestra lo que se ha extraído del log, presentado de manera destacada y con colores correspondientes, con el propósito de hacer más comprensible y accesible la información para los estudiantes.

```

Ivonne@pa3p2: ~/MemoriaTituloIvonne/ejerciciosEstudiantes/2017425857/Serie_1/Ejerc...
pom.xml src target
Ivonne@pa3p2:~/MemoriaTituloIvonne/ejerciciosEstudiantes/2017425857/Serie_1/Ejercicio_3$ mvn clean
test
[INFO] Scanning for projects...
[INFO] -----< org.example:UnitTest >-----
[INFO] Building UnitTest 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO] --- clean:3.2.0:clean (default-clean) @ UnitTest ---
[INFO] Deleting /home/Ivonne/MemoriaTituloIvonne/ejerciciosEstudiantes/2017425857/Serie_1/Ejercicio_3
/target
[INFO] --- resources:3.3.1:resources (default-resources) @ UnitTest ---
[INFO] skip non existing resourceDirectory /home/Ivonne/MemoriaTituloIvonne/ejerciciosEstudiantes/201
7425857/Serie_1/Ejercicio_3/src/main/resources
[INFO] --- compiler:3.11.0:compile (default-compile) @ UnitTest ---
[INFO] Changes detected - recompiling the module! :source
[INFO] Compiling 1 source file with javac [debug target 21] to target/classes
[INFO] --- resources:3.3.1:testResources (default-testResources) @ UnitTest ---
[INFO] skip non existing resourceDirectory /home/Ivonne/MemoriaTituloIvonne/ejerciciosEstudiantes/201
7425857/Serie_1/Ejercicio_3/src/test/resources
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ UnitTest ---
[INFO] Changes detected - recompiling the module! :dependency
[INFO] Compiling 1 source file with javac [debug target 21] to target/test-classes
[INFO] --- surefire:3.1.0:test (default-test) @ UnitTest ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running Parte3Test
[ERROR] Tests run: 3, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.349 s <<< FAILURE! - in Par
te3Test
[ERROR] Parte3Test.testMultiplosDe5 Time elapsed: 0.017 s <<< FAILURE!
org.opentest4j.AssertionFailedError: El método debe devolver Buzz si el argumento es múltiplo de 5 ==
> expected: <Buzz> but was: <5>
    at org.junit.jupiter.api.AssertionFailureBuilder.build(AssertionFailureBuilder.java:151)
    at org.junit.jupiter.api.AssertionFailureBuilder.buildAndThrow(AssertionFailureBuilder.java:1
32)
    at org.junit.jupiter.api.AssertEquals.failNotEqual(AssertEquals.java:197)
    at org.junit.jupiter.api.AssertEquals.assertEquals(AssertEquals.java:182)
    at org.junit.jupiter.api.Assertions.assertEquals(Assertions.java:1153)
    at Parte3Test.testMultiplosDe5(Parte3Test.java:28)
    at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccesso
r.java:193)
    at java.base/java.lang.reflect.Method.invoke(Method.java:580)
    at org.junit.platform.commons.util.ReflectionUtils.invokeMethod(ReflectionUtils.java:725)
    at org.junit.jupiter.engine.execution.MethodInvocation.proceed(MethodInvocation.java:60)
    at org.junit.jupiter.engine.execution.InvocationInterceptorChain$ValidatingInvocation.proceed
    at org.junit.platform.launcher.core.EngineExecutionOrchestrator.lambda$execute$0(EngineExecu
tionOrchestrator.java:55)
    at org.junit.platform.launcher.core.EngineExecutionOrchestrator.withInterceptedStreams(Engine
ExecutionOrchestrator.java:102)
    at org.junit.platform.launcher.core.EngineExecutionOrchestrator.execute(EngineExecutionOrches
trator.java:54)
    at org.junit.platform.launcher.core.DefaultLauncher.execute(DefaultLauncher.java:114)
    at org.junit.platform.launcher.core.DefaultLauncher.execute(DefaultLauncher.java:86)
    at org.junit.platform.launcher.core.DefaultLauncherSession$DelegatingLauncher.execute(Default
LauncherSession.java:86)
    at org.apache.maven.surefire.junitplatform.LazyLauncher.execute(LazyLauncher.java:50)
    at org.apache.maven.surefire.junitplatform.JUnitPlatformProvider.execute(JUnitPlatformProvide
r.java:184)
    at org.apache.maven.surefire.junitplatform.JUnitPlatformProvider.invokeAllTests(JUnitPlatform
Provider.java:148)
    at org.apache.maven.surefire.junitplatform.JUnitPlatformProvider.invoke(JUnitPlatformProvide
r.java:122)
    at org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:385)
    at org.apache.maven.surefire.booter.ForkedBooter.execute(ForkedBooter.java:162)
    at org.apache.maven.surefire.booter.ForkedBooter.run(ForkedBooter.java:507)
    at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:495)
[INFO] Results:
[INFO] Failures:
[ERROR] Parte3Test.testMultiplosDe5:28 El método debe devolver Buzz si el argumento es múltiplo de
5 ==> expected: <Buzz> but was: <5>
[INFO] Tests run: 3, Failures: 1, Errors: 0, Skipped: 0
[INFO] -----
[INFO] BUILD FAILURE
[INFO] -----
[INFO] Total time: 9.293 s
[INFO] Finished at: 2024-03-21T21:20:32-03:00
[INFO] -----
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-surefire-plugin:3.1.0:test (default-tes
t) on project UnitTest: There are test failures.
[ERROR] Please refer to /home/Ivonne/MemoriaTituloIvonne/ejerciciosEstudiantes/2017425857/Serie_1/Eje
rcicio_3/target/surefire-reports for the individual test results.
[ERROR] Please refer to dump files (if any exist) [date].dump, [date]-jvmRun[N].dump and [date].dump
srean.
[ERROR] -> [Help 1]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR] For more information about the errors and possible solutions, please read the following artic
les:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException

```

Figura 10: Ejecución por consola de test unitarios

Como se mencionó, en la ruta *'test/java/org/example'* se almacenan los test unitarios únicos para cada ejercicio, los cuales deben ser creados por el docente junto con el enunciado correspondiente. Estos test unitarios deben ser preparados previamente por el docente. Cuando un docente crea una nueva serie dentro de la página web, se crea una carpeta que contiene "Serie_id", luego cuando se crea un nuevo ejercicio este sigue el mismo formato, de la forma *Ejercicio_id* que contiene el proyecto Maven el cual copia de una plantilla vacía que contiene solo las carpetas y el archivo pom.xml, luego se le agregan a la ruta *'test/java/org/example'* los archivos *.java* con los test unitarios creados anteriormente por el docente y así queda listo el archivo de proyecto Maven para ser utilizado.

5.3.3. Configuración de Archivos del Estudiante en el Backend

Cuando un estudiante carga sus archivos desde la interfaz de la plataforma, se inicia un proceso en el backend para gestionar estos archivos. Internamente, se llama a una función que crea una carpeta específica que contiene la matrícula del estudiante. Esta carpeta contendrá todas las series y ejercicios asignados al estudiante, siguiendo el mismo formato de la creación de la serie y ejercicio realizada por el docente.

Luego de esta carga de archivos por parte del estudiante, se hace una copia del ejercicio y se le añaden en la ruta correspondiente los archivos .java que subió el estudiante. Esta acción garantiza que los archivos del estudiante estén correctamente organizados y listos para la evaluación mediante los test unitarios.

Una vez que los archivos están en el lugar correspondiente, se procede a ejecutar los test unitarios utilizando subprocesos y Maven en la ruta creada recientemente, donde quedaron finalmente los archivos del estudiante. Luego de la ejecución de los test unitarios, se extrae la información relevante a través de expresiones regulares para proporcionar la retroalimentación al estudiante.

5.3.4. Proceso de Retroalimentación

Una vez obtenida la información relevante de la ejecución de los test unitarios, se procede a proporcionar retroalimentación al estudiante. Esta retroalimentación se presenta de manera clara y estructurada, destacando los errores encontrados, si los hubiera. Además, se resalta el éxito en la aprobación de los test unitarios para reforzar los logros del estudiante.

Es importante señalar que la información que se muestra en las ejecuciones de los test se basa en lo que haya designado el docente en los test unitarios (archivos .java) que se prepararon con anticipación. Esto permite un control detallado del problema y la identificación de los errores, así como la provisión de explicaciones detalladas sobre los conceptos involucrados y sugerencias para mejorar el código.

```

import org.example.FizzBuzz;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class Parte4Test {

    @Test
    public void testNumerosEstandar() {
        FizzBuzz fizzBuzz = new FizzBuzz();
        assertEquals("1", fizzBuzz.forValue(1), "El metodo debe
            devolver el numero utilizado como argumento, pero como
            String");
        assertEquals("4", fizzBuzz.forValue(4), "El metodo debe
            devolver el numero utilizado como argumento, pero como
            String");
        assertEquals("7", fizzBuzz.forValue(7), "El metodo debe
            devolver el numero utilizado como argumento, pero como
            String");
    }

    @Test
    public void testMultiplosDe3() {
        FizzBuzz fizzBuzz = new FizzBuzz();
        assertEquals("Fizz", fizzBuzz.forValue(3), "El metodo debe
            devolver Fizz si el argumento es multiplo de 3");
        assertEquals("Fizz", fizzBuzz.forValue(6), "El metodo debe
            devolver Fizz si el argumento es multiplo de 3");
        assertEquals("Fizz", fizzBuzz.forValue(99), "El metodo debe
            devolver Fizz si el argumento es multiplo de 3");
    }

    @Test
    public void testMultiplosDe5() {
        FizzBuzz fizzBuzz = new FizzBuzz();
        assertEquals("Buzz", fizzBuzz.forValue(5), "El metodo debe
            devolver Buzz si el argumento es multiplo de 5");
        assertEquals("Buzz", fizzBuzz.forValue(10), "El metodo debe
            devolver Buzz si el argumento es multiplo de 5");
        assertEquals("Buzz", fizzBuzz.forValue(100), "El metodo debe
            devolver Buzz si el argumento es multiplo de 5");
    }

    @Test
    public void testMultiplosDe3y5() {
        FizzBuzz fizzBuzz = new FizzBuzz();
        assertEquals("FizzBuzz", fizzBuzz.forValue(15), "El metodo
            debe devolver FizzBuzz si el argumento es multiplo de 3 y
            de 5");
        assertEquals("FizzBuzz", fizzBuzz.forValue(30), "El metodo
            debe devolver FizzBuzz si el argumento es multiplo de 3 y
            de 5");
        assertEquals("FizzBuzz", fizzBuzz.forValue(450), "El metodo
            debe devolver FizzBuzz si el argumento es multiplo de 3 y
            de 5");
    }
}

```

Listing 1: Ejemplo de código Java para test unitarios

```
package org.example;

public class FizzBuzz {

    public String forValue(int value) {
        if (value % 3 == 0 && value % 5 == 0) {
            return "FizzBuzz";
        } else if (value % 3 == 0) {
            return "Fizz";
        } else if (value % 5 == 0) {
            return "Buzz";
        } else {
            return String.valueOf(value);
        }
    }
}
```

Listing 2: Ejemplo de solución para resolver ejercicio

Este proceso de retroalimentación es fundamental para fomentar un aprendizaje efectivo y el desarrollo de habilidades de programación en los estudiantes.

5.4. Desarrollo de la página web

El proceso de desarrollo de la página web fue llevado a cabo de manera integral, abordando tanto el frontend como el backend de manera complementaria. La implementación se estructuró en dos aspectos fundamentales: la vista y los componentes destinados al docente, así como la perspectiva diseñada para el estudiante. El proyecto se organiza en rutas específicas, las cuales gestionan operaciones como solicitudes POST, GET y DELETE para manipular la página de manera eficiente.

5.5. Interfaz de Usuario

5.5.1. Login

Dentro de las diversas rutas establecidas en el proyecto, se destaca el proceso de inicio de sesión compartido tanto para estudiantes como para docentes. Se empleó Flask Login para gestionar las sesiones de usuario, encargándose de tareas comunes como

el inicio y cierre de sesión. Para validar el acceso a una ruta específica, se implementó el decorador *@login_required*, configurándolo para discernir entre un estudiante y un docente. Esta diferenciación se logró a través de la configuración de la base de datos, donde se añadieron propiedades particulares a los modelos de Supervisor y Estudiante. Adicionalmente, se introdujo un método que devuelve un identificador mediante la inclusión de “s” seguido del ID para el supervisor, y “e” seguido del ID para el estudiante. Flask Login, al reconocer la identificación del usuario, lo redirige automáticamente a rutas específicas, ajustándose al rol del usuario como estudiante o docente.

5.5.2. Principales Vistas

La interfaz de usuario se organiza en torno a dos vistas principales: la del docente y la del estudiante. Estas vistas sirven como puntos de acceso desde los cuales se derivan otras pantallas relevantes. Cada una de estas interfaces está sujeta a autenticación mediante Flask Login. A continuación, se detallan las características distintivas de cada vista y su funcionalidad asociada.

■ Vista Estudiante

En la vista del estudiante, se presentan series de ejercicios con sus respectivos componentes, permitiendo la selección de una. Al elegir una serie, se abre una nueva vista que muestra los ejercicios correspondientes y la calificación del estudiante hasta ese momento. El cálculo de la nota se realiza considerando que cada ejercicio aprobado otorga 1 punto, evaluando automáticamente la calificación según la cantidad total de ejercicios en la serie, con una exigencia del 60 %.

Dentro de la vista del estudiante, al hacer clic en un ejercicio, se visualiza su enunciado y se proporciona la opción de cargar archivos con las clases correspondientes. En la parte inferior, se presentan los nombres de los ejercicios de la serie, indicando cuáles han sido aprobados, fallidos y los que aún deben ser com-

pletados. La carga de archivos requiere que el estudiante suba todas sus clases en formato .java desde su proyecto IntelliJ, siguiendo las instrucciones proporcionadas por el docente.

Luego de cargar los archivos, el backend recibe la información y, en caso de que algún test no sea superado, se presenta un anuncio en color rojo detallando los errores encontrados según la configuración del docente en los test unitarios. En caso contrario, si todos los tests son aprobados, la plataforma informa sobre el éxito de la operación.

Cabe destacar un aspecto importante de la carga de archivos. Como se mostró en el esquema MER, existe una relación *asigna* entre el estudiante y el ejercicio. En el momento en que el estudiante carga sus soluciones a la plataforma, se crea una nueva relación. Esta es responsable de almacenar toda la información relacionada con la carga de archivos de cada estudiante en la plataforma. Por lo tanto, se puede conocer la cantidad de intentos que se llevaron a cabo, además de si el estudiante aprueba o no el ejercicio. También se registra el output que proporciona la consola durante la ejecución, la fecha del último intento y el código proporcionado por el estudiante.

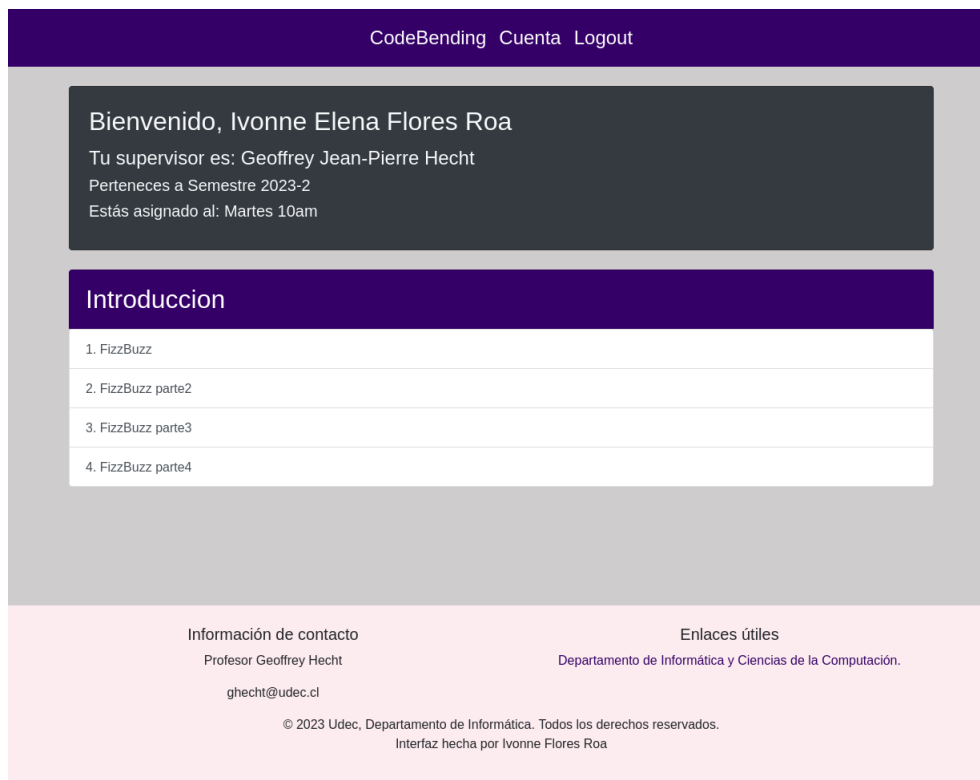


Figura 11: DashBoard del estudiante

■ Vista Docente

La vista docente es un componente más complejo que la del estudiante y representa una parte significativa del desarrollo debido a sus múltiples operaciones. Al ingresar a la plataforma con sus credenciales, el docente accede a su panel principal, el cual contiene una lista de los cursos creados, las series disponibles y un conjunto de opciones que le permiten gestionar el contenido educativo.

Una vez en el panel principal, el docente puede realizar diversas acciones, como crear un nuevo curso, registrar estudiantes, diseñar series de ejercicios y asignar ejercicios a cada serie. Al seleccionar un curso, se muestra una vista detallada que incluye la lista de cursos, los grupos y las series asociadas, junto con opciones para activar o desactivar el curso, crear nuevos grupos, supervisar el progreso del curso y eliminar el curso.

CodeBending Cuenta Logout

<

FizzBuzz parte2

Y ahora también, si `value` es múltiplo de de 3 entonces debe devolver `Fizz`.

Selecciona tus archivos .java (Proyecto/src/main/java/org/example)

Examinar... Ningún archivo seleccionado.

Subir Solución

Errores en la ejecución de pruebas unitarias

- ERROR Failures:
- ERROR Parte2Test.testMultiplosDe3:20 El método debe devolver Fizz si el argumento es múltiplo de 3 expected: <Fizz> but was: <3>

FizzBuzz FizzBuzz parte2 FizzBuzz parte3 FizzBuzz parte4

Calificación Total

2.25

Información de contacto
Profesor Geoffrey Hecht
ghecht@udec.cl

Enlaces útiles
Departamento de Informática y Ciencias de la Computación.

© 2023 Udec, Departamento de Informática. Todos los derechos reservados.
Interfaz hecha por Ivonne Flores Roa

Figura 12: Ejemplo ejercicio fallido y retroalimentación

Para analizar el progreso del curso, el docente puede seleccionar una serie específica y visualizar si los estudiantes han aprobado o no los ejercicios de la serie, así como la cantidad de intentos realizados. Además, el docente tiene la capacidad de seleccionar un grupo dentro del curso para revisar sus detalles, modificar su nombre y eliminarlo. También puede ver la lista de estudiantes que pertenecen a dicho grupo.

Asimismo, el docente tiene la opción de seleccionar a un estudiante en particular para revisar su información personal, así como los ejercicios que ha completado y aquellos que aún tiene asignados para desarrollar.

Para el docente, uno de los aspectos a considerar es el registro de los estudiantes en el curso. Inicialmente, el docente crea el curso asignándole un nombre representativo. Luego, tiene la opción de añadir estudiantes mediante la carga de un archivo en formato .csv. Es importante destacar que este archivo debe cumplir con ciertas especificaciones, como se detalla en la tabla adjunta, que incluyen campos como matrícula, apellidos, nombres, correo electrónico UDEC y carrera. Al registrar a un estudiante por primera vez, se utiliza su matrícula como contraseña, la cual luego los estudiantes podrán modificar desde su propia interfaz. Es crucial seguir este formato para garantizar un registro exitoso de todos los estudiantes; de lo contrario, el proceso de registro no se llevará a cabo.

Matricula	Apellidos	Nombres	Correo UDEC	Carrera
2017425857	Flores Roa	Ivonne Elena	ivflores2017@udec.cl	(3319)INGENIERIA CIVIL INFORMATICA

Cuadro 1: Ejemplo archivo de registro estudiantes (.csv)

Una parte destacada de la vista docente es el proceso de crear un ejercicio. Para comenzar, es necesario que exista al menos una serie previamente creada. Una vez que se ha asegurado este requisito, se permite al docente iniciar el proceso de creación del ejercicio.

Al crear un ejercicio, el docente debe seguir varios pasos. Primero, se le solicita

que proporcione un nombre para el ejercicio. Además, tiene la opción de cargar un archivo en formato markdown[7] que contenga el enunciado del ejercicio y adjuntar imágenes, si es necesario. Posteriormente, debe añadir los archivos .java que contienen los test unitarios que serán ejecutados en el ejercicio específico.

Además de los elementos mencionados anteriormente, el docente debe seleccionar la serie a la cual desea agregar el ejercicio. Esta selección se realiza a través de un menú desplegable que presenta todas las series disponibles en la plataforma. Una funcionalidad adicional que el docente tiene permitido realizar es asignar una serie a un grupo específico. Esto permite que los estudiantes asociados a ese grupo puedan ver y completar los ejercicios dentro de esa serie como parte de su práctica en la plataforma.

Una vez que se han completado todos los pasos y se ha verificado la información proporcionada, el docente puede proceder a agregar el ejercicio a la serie seleccionada. Esto se logra mediante la activación de un botón designado para esta acción.

Una vez que se ha confirmado la adición del ejercicio a la serie, este se crea en la base de datos y queda disponible para que los estudiantes accedan y completen como parte de su práctica en la plataforma.



Figura 13: DashBoard Docente

6. Validación y Resultados

6.1. Validación

Como se mencionó anteriormente, la validación de este trabajo de memoria de título implicó la realización de una prueba en la que se invitó a estudiantes y ex estudiantes del curso que hubieran utilizado la plataforma PA3P en ocasiones anteriores. La prueba consistió en probar la plataforma desarrollada en este trabajo de memoria de título y posteriormente responder a una encuesta.

La encuesta utilizó el sistema de escalas de usabilidad SUS (System Usability Scale) para evaluar la eficacia, eficiencia y satisfacción de la plataforma. Este método fue elegido debido a su facilidad para calcular el resultado final con enunciados predefinidos, y su brevedad para los usuarios al responder.

La encuesta consta de cuatro secciones. En la primera sección, se evalúa la plata-

CodeBending Cuenta Logout

Volver

Agregar nuevo ejercicio

El nombre no puede contener más de 50 caracteres.

Nombre del Ejercicio:

Cargar enunciado (.md):
 Ningún archivo seleccionado.

Cargar imágenes para el enunciado (Múltiples):
 Ningún archivo seleccionado.

Seleccionar Serie:

Cargar archivos test unitarios .java (Múltiples):
 Ningún archivo seleccionado.

Información de contacto
Profesor Geoffrey Hecht
ghecht@udec.cl

Enlaces útiles
Departamento de Informática y Ciencias de la Computación.

© 2023 Udec, Departamento de Informática. Todos los derechos reservados.
Interfaz hecha por Ivonne Flores Roa

Figura 14: Formulario para crear un ejercicio

forma actual PA3P, calificando la experiencia general y respondiendo al cuestionario SUS. La segunda sección replica la primera, pero para la nueva plataforma desarrollada en esta memoria de título. La tercera sección compara ambas plataformas con las siguientes cuatro preguntas específicas. A continuación se detalla el cuestionario en su totalidad:

1. Plataforma PA3P actual.

Primero, se inicia con la siguiente pregunta: según su conocimiento previo de la plataforma PA3P, califique su rendimiento general con un número del 1 al 5, donde 1 representa una experiencia poco satisfactoria y 5 una experiencia muy satisfactoria.

Luego, se aplica el cuestionario SUS (System Usability Scale) a la plataforma PA3P, para así tener un buen nivel de comparación con los resultados siguientes.

Las preguntas son las siguientes:

- Creo que me gustaría utilizar este sistema con frecuencia
- Encontré el sistema innecesariamente complejo
- Antes de entrar a la plataforma, pensé que el sistema era fácil de usar
- Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema
- Encontré que las diversas funciones de este sistema estaban bien integradas
- Pensé que había demasiada inconsistencia en este sistema
- Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente
- Encontré el sistema muy complicado de usar
- Me sentí muy seguro usando el sistema
- Necesitaba aprender muchas cosas antes de empezar con este sistema

Las respuestas a cada enunciado se piden siguiendo la Escala de Likert, de modo que hay cinco opciones:

- a) Totalmente en desacuerdo
- b) En desacuerdo
- c) Neutro
- d) De acuerdo
- e) Totalmente de acuerdo

2. **Nueva Plataforma.** Después de que los estudiantes hayan utilizado la nueva plataforma en la práctica, se les pide calificar su rendimiento general con un número del 1 al 5, donde 1 representa una experiencia poco satisfactoria y 5 una experiencia muy satisfactoria. Además, se aplica el mismo cuestionario para la sección 1 utilizando la Escala de Likert[8] para las respuestas.

3. **Comparación de ambas plataformas:**

En esta sección, se han formulado las siguientes preguntas, las cuales utilizan la escala de Likert para las respuestas. Estas cuatro interrogantes tienen como objetivo validar las problemáticas más significativas identificadas en la sección de 2. Es importante destacar que en esta sección, las respuestas se interpretaron como una calificación otorgada a la nueva plataforma en comparación con la antigua.

- ¿En comparación con la plataforma antigua, qué tan atractiva encuentras la interfaz y el diseño general de la nueva plataforma de autoevaluación de código? Esta pregunta se responde con un número del 1 al 5 donde 1 es poco atractiva y 5 es muy atractiva.
- Comparando con la forma de inicio de sesión en la plataforma antigua, ¿sientes que la nueva forma de inicio de sesión ha mejorado la seguridad y pri-

vacidad de tus datos? Esta pregunta se responde con un número del 1 al 5 donde 1 es menos segura y 5 es más segura.

- Comparando con la retroalimentación en la plataforma antigua, ¿cómo describirías la retroalimentación proporcionada sobre tus soluciones de código en la nueva plataforma? Esta pregunta se responde con un número del 1 al 5 donde 1 es muy mala y 5 es muy buena.
- ¿En comparación con la plataforma antigua, qué tan claro encuentras la identificación de errores en tus soluciones en la nueva plataforma? Esta pregunta se responde con un número del 1 al 5 donde 1 es poco claro y 5 es muy claro.

4. **Sugerencias y comentarios.** En esta sección se dejaron las siguientes casillas de comentarios:

- ¿Qué le ha parecido esta Nueva Versión de PA3P?
- Otros comentarios
- ¿Qué versión de PA3P preferiría utilizar para un futuro curso? Esta es una pregunta para que los encuestados puedan elegir una plataforma para usar en algún curso futuro.

6.2. Resultados

A continuación, se presentarán los resultados obtenidos tras aplicar la encuesta de validación a una muestra de 16 estudiantes y ex estudiantes

- **Plataforma PA3P actual** La calificación promedio para la experiencia general con la plataforma PA3P actual fue de 3 sobre 5. Este puntaje indica una percepción moderadamente satisfactoria de la plataforma entre los usuarios. Para los resultados del cuestionario sus se hará mas adelante en una sección aparte.

- Nueva Plataforma** La calificación promedio para la experiencia general con la nueva plataforma fue de 4.18 sobre 5. Este puntaje indica una mejora en la percepción de los usuarios con respecto a la plataforma anterior.

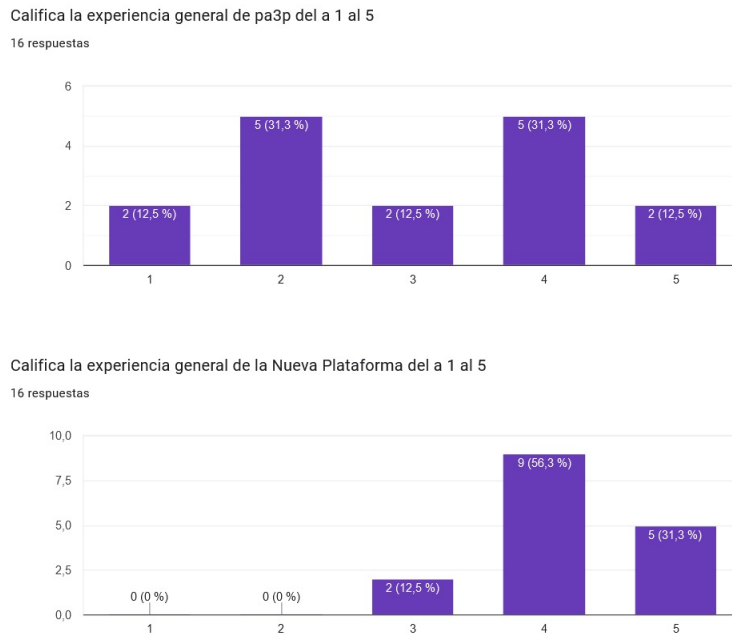


Figura 15: Experiencia General

- Comparación de ambas plataformas** Al comparar ambas plataformas, se observaron las siguientes tendencias: Es importante recordar que las respuestas se interpretaron como una calificación otorgada a la nueva plataforma en comparación con la antigua.
 - Atractivo de la Interfaz:** La nueva plataforma recibió una calificación promedio de 4.6 sobre 5 en cuanto a su atractivo visual y diseño general comparado con PA3P[1]. Esta puntuación indica una mejora considerable en la percepción del aspecto estético de la plataforma, lo que sugiere una interfaz más atractiva para los usuarios.
 - Seguridad y Privacidad:** La nueva forma de inicio de sesión fue evalua-

da con un promedio de 4.3 sobre 5 en términos de seguridad y privacidad comparado con PA3P[1]. Esta puntuación refleja una mejora significativa en la percepción de los usuarios sobre la protección de sus datos personales al utilizar la nueva plataforma.

- **Retroalimentación sobre Soluciones de Código:** La retroalimentación proporcionada sobre soluciones de código en la nueva plataforma obtuvo una calificación promedio de 3.8 sobre 5 comparado con PA3P[1]. Aunque esta puntuación indica una mejora respecto a la plataforma anterior, aún sugiere áreas donde se puede seguir trabajando para mejorar la calidad y utilidad de la retroalimentación ofrecida.
- **Claridad en la Identificación de Errores:** La claridad en la identificación de errores en la nueva plataforma fue calificada en promedio con 3.5 sobre 5 comparado con PA3P[1]. Aunque esta puntuación indica una mejora respecto a la plataforma anterior, sugiere que todavía hay margen para mejorar la facilidad con la que los usuarios identifican los errores en sus soluciones de código.

Finalmente, en la encuesta se encuentra la pregunta: "¿Qué versión de PA3P preferiría utilizar para un futuro curso?" Donde el 100 % de los encuestados optó por la nueva versión propuesta. Este resultado indica una clara preferencia por la nueva plataforma entre los usuarios, destacando su aceptación y probable satisfacción con las mejoras introducidas.

- **Resultados SUS** Los resultados del cuestionario SUS proporcionaron una evaluación cuantitativa de la usabilidad percibida de ambas plataformas, los resultados van del 0 al 100 según la siguiente tabla:

Utilizando la fórmula estándar del System Usability Scale[9], se calcularon las puntuaciones promedio y se presentan a continuación:

Puntuación SUS	Interpretación
0-49	Usabilidad percibida baja
50-68	Usabilidad percibida moderada
69-84	Usabilidad percibida buena
85-100	Usabilidad percibida excelente

Cuadro 2: Interpretación de las puntuaciones del SUS

- **Plataforma PA3P Actual:** La puntuación promedio del SUS para la plataforma PA3P fue de 49.53. Esta puntuación sugiere una usabilidad percibida relativamente baja entre los usuarios de la plataforma actual, ya que está por debajo del umbral típico de 68 que se considera aceptable. Indica que los usuarios encuentran la plataforma actual menos satisfactoria en términos de eficacia, eficiencia y satisfacción. Es posible que estén experimentando dificultades significativas al utilizar la plataforma, lo que podría afectar negativamente su experiencia y eficiencia al trabajar con ella.
- **Nueva Plataforma:** La puntuación promedio del SUS para la nueva plataforma fue de 78.59, reflejando una mejora significativa en la usabilidad percibida en comparación con la plataforma anterior. Los usuarios encuentran la nueva plataforma más fácil de usar y están más satisfechos con su experiencia general. La alta puntuación sugiere que la nueva plataforma puede ofrecer mejoras significativas en términos de facilidad de uso, eficiencia y satisfacción del usuario.

Estos resultados corroboran las tendencias observadas en la encuesta y proporcionan una comprensión cuantitativa más sólida de la experiencia del usuario con ambas plataformas. La nueva plataforma parece haber logrado mejoras significativas en aspectos como el atractivo de la interfaz y la seguridad. Sin embargo, también resaltan áreas de oportunidad, como la retroalimentación sobre soluciones de código y la claridad en la identificación de errores, que podrían ser objeto de futuras mejoras y refinamientos.

7. Conclusiones

El desarrollo de este proyecto ha permitido identificar diversas problemáticas en la plataforma existente PA3P y proponer soluciones con el objetivo de mejorar la experiencia de enseñanza y aprendizaje, el proyecto ha demostrado un cumplimiento efectivo de los objetivos propuestos, logrando mejoras sustanciales en la plataforma PA3P y brindando una experiencia de enseñanza y aprendizaje más efectiva y enriquecedora para todos los usuarios involucrados. Las soluciones implementadas han permitido superar las limitaciones identificadas, proporcionando herramientas más eficientes y centradas en el aprendizaje activo.

Los resultados obtenidos de la validación, incluyendo la encuesta de usuarios y el análisis de la usabilidad, respaldan la efectividad de las mejoras implementadas. Los usuarios han expresado una mayor satisfacción y han percibido una mejora significativa en la usabilidad y funcionalidad de la plataforma. Además, la preferencia unánime por la nueva versión de PA3P para futuros cursos indica un claro reconocimiento de los beneficios y mejoras introducidas.

7.1. Impacto en la Experiencia de Enseñanza y Aprendizaje

Las soluciones propuestas han impactado positivamente en la experiencia de enseñanza y aprendizaje. Se ha logrado proporcionar herramientas que permiten una evaluación continua y detallada del rendimiento estudiantil, mejorar la retroalimentación, agilizar la corrección de ejercicios y reforzar la comprensión de la programación orientada a objetos en Java. Esto contribuye al desarrollo de habilidades sólidas en la escritura de código de calidad, promoviendo un ambiente propicio para el crecimiento académico y profesional de los estudiantes.

Además, la introducción de pruebas unitarias en el proceso de revisión de ejercicios ha abordado efectivamente la problemática relacionada con los problemas de sintaxis

y filosofía de programación. Al permitir una evaluación más rigurosa y detallada del código de los estudiantes, estas pruebas han mejorado la detección de errores de sintaxis y han proporcionado una mayor flexibilidad en la estructuración del código. Esto no solo ayuda a los estudiantes a comprender mejor los conceptos de programación, sino que también fomenta un enfoque más sólido y reflexivo hacia la resolución de problemas de codificación.

7.2. Limitaciones

A pesar de los avances significativos evidenciados por las soluciones propuestas, según lo revelado por el análisis de la encuesta SUS, persisten áreas de oportunidad que podrían ser objeto de futuras mejoras y refinamientos. Por ejemplo, la retroalimentación sobre las soluciones de código y la claridad en la identificación de errores podrían ser mejoradas aún más para ofrecer una experiencia aún más enriquecedora para los usuarios. Estos aspectos representan puntos críticos donde la plataforma podría beneficiarse de ajustes adicionales con el fin de optimizar la experiencia del usuario y promover un entorno de aprendizaje más eficaz y gratificante.

8. Recomendaciones

8.1. Trabajo a futuro

Para garantizar la continua evolución y mejora del proyecto, se sugiere considerar las siguientes mejoras potenciales:

- **Refinamiento de la Retroalimentación:** Continuar mejorando la calidad y utilidad de la retroalimentación proporcionada sobre las soluciones de código. Esto puede incluir la incorporación de sugerencias más específicas y detalladas para ayudar a los estudiantes a comprender y corregir sus errores de manera más efectiva.
- **Panel de Monitoreo de Ayudantes:** Desarrollar una nueva vista y funcionalidades específicas para que los estudiantes ayudantes puedan revisar el progreso de los estudiantes en el curso. Esto permitirá una mayor colaboración entre estudiantes y facilitará la asistencia personalizada.
- **Sección de Asistencia al Curso:** Agregar una sección dedicada para tomar la asistencia al curso, lo que proporcionará una herramienta adicional para los docentes para gestionar la participación de los estudiantes.
- **Cambios Gráficos en la Aplicación:** Realizar cambios gráficos en la aplicación para mejorar la experiencia visual de los usuarios y hacerla más atractiva y fácil de usar.
- **Backup de la Base de Datos y Archivos:** Implementar un sistema automatizado para realizar copias de seguridad de la base de datos y los archivos del proyecto, garantizando la integridad y disponibilidad de los datos en caso de fallos o pérdidas.
- **Editor de Texto Online:** Integrar un editor de texto online similar a Exercism[10],

que permita a los estudiantes practicar la escritura de código en un entorno controlado.

- **Estadísticas y Datos en Tiempo Real:** Incorporar la capacidad de recopilar datos estadísticos y de desempeño en tiempo real de los estudiantes mientras trabajan en los ejercicios. Esto proporcionará información valiosa para los docentes sobre el progreso individual y colectivo de los estudiantes, permitiendo ajustes y mejoras continuas en el contenido y la metodología de enseñanza.

Referencias

- [1] "Plataforma pa3p." [Online]. Available: http://pa3p.inf.udec.cl/JAVA/public_html
- [2] "Lenguaje java." [Online]. Available: <https://www.oracle.com/java/>
- [3] "Maven." [Online]. Available: <https://maven.apache.org/>
- [4] "Sqlalchemy." [Online]. Available: <https://www.sqlalchemy.org/>
- [5] "Intellij." [Online]. Available: <https://www.jetbrains.com/es-es/idea/>
- [6] "Python regex." [Online]. Available: https://www.w3schools.com/python/python_regex.asp
- [7] "Markdown." [Online]. Available: <https://www.markdownguide.org/>
- [8] "Escala de likert." [Online]. Available: <https://www.questionpro.com/blog/es/que-es-la-escala-de-likert-y-como-utilizarla/>
- [9] "Blog sobre la escala de usabilidad del sistema." [Online]. Available: <https://blog.uxtweak.com/system-usability-scale/>
- [10] "Exercism - java." [Online]. Available: <https://exercism.org/tracks/java/>
- [11] "Codewars." [Online]. Available: <https://www.codewars.com/>
- [12] "JUnit 5." [Online]. Available: <https://junit.org/junit5/>
- [13] "Cómo medir la usabilidad: ¿qué es el sus?" [Online]. Available: <https://www.uifrommars.com/como-medir-usabilidad-que-es-sus/>
- [14] "Project object model." [Online]. Available: https://maven.apache.org/pom.html#What_is_the_POM.3F
- [15] "Gunicorn." [Online]. Available: <https://gunicorn.org/>
- [16] "Flask." [Online]. Available: <https://flask.palletsprojects.com/en/3.0.x/>

9. Anexo

9.1. Documentación Casos de Uso

1. Iniciar Sesión:

- **Descripción:** El usuario puede iniciar sesión en la plataforma.
- **Actores:** Docente, Estudiante
- **Flujo Principal:**
 - a) El actor ingresa su nombre de usuario y contraseña.
 - b) El sistema autentica las credenciales del usuario.
 - c) El sistema redirige al usuario a su panel principal.

2. Gestionar Cursos

- **Descripción:** El docente puede crear, modificar y eliminar cursos.
- **Actores:** Docente
- **Flujo Principal:**
 - a) El docente accede al panel principal.
 - b) El docente selecciona la opción para crear un nuevo curso o modifica/elimina un curso existente.

3. Registrar Estudiantes en un Curso:

- **Descripción:** El docente puede registrar estudiantes en un curso.
- **Actores:** Docente
- **Flujo Principal:**
 - a) El docente selecciona un curso.
 - b) El docente carga un archivo CSV con la información de los estudiantes.
 - c) El sistema verifica y registra a los estudiantes en el curso.

4. Gestionar Series:

- **Descripción:** El docente puede crear, modificar y eliminar series de ejercicios.
- **Actores:** Docente
- **Flujo Principal:**
 - a) El docente accede al panel de gestión de series.
 - b) El docente selecciona la opción para crear una nueva serie o modifica/elimina una serie existente.

5. Gestionar Estudiantes:

- **Descripción:** El docente puede agregar, modificar y eliminar estudiantes.
- **Actores:** Docente
- **Flujo Principal:**
 - a) El docente accede al panel de gestión de estudiantes.
 - b) El docente selecciona la opción para agregar un nuevo estudiante o modifica/elimina un estudiante existente.

6. Gestionar Grupos:

- **Descripción:** El docente puede crear, modificar y eliminar grupos de estudiantes.
- **Actores:** Docente
- **Flujo Principal:**
 - a) El docente accede al panel de gestión de grupos.
 - b) El docente selecciona la opción para crear un nuevo grupo o modifica/elimina un grupo existente.

7. Crear Ejercicios:

- **Descripción:** El docente puede crear nuevos ejercicios y asignarlos a series específicas.
- **Actores:** Docente
- **Flujo Principal:**
 - a) El docente accede al panel de creación de ejercicios.
 - b) El docente proporciona un nombre, enunciado y archivos de test unitarios para el nuevo ejercicio.
 - c) El docente selecciona la serie a la que desea asignar el ejercicio.
 - d) El docente confirma la adición del ejercicio a la serie.

8. Gestionar Ejercicios:

- **Descripción:** El docente puede modificar y eliminar ejercicios existentes.
- **Actores:** Docente
- **Flujo Principal:**
 - a) El docente accede al panel de gestión de ejercicios.
 - b) El docente selecciona un ejercicio existente.
 - c) El docente modifica o elimina el ejercicio según sea necesario.

9. Examinar Ejercicios:

- **Descripción:** El docente puede revisar los detalles de un ejercicio.
- **Actores:** Docente
- **Flujo Principal:**
 - a) El docente accede al panel de visualización de ejercicios.
 - b) El docente selecciona un ejercicio para examinar sus detalles y archivos asociados.

10. Monitorear Estudiantes:

- **Descripción:** El docente puede supervisar el progreso y desempeño de los estudiantes en los cursos y series.
- **Actores:** Docente
- **Flujo Principal:**
 - a) El docente accede al panel de seguimiento de progreso de estudiantes.
 - b) El docente selecciona una serie.
 - c) El docente visualiza el progreso y desempeño de los estudiantes en la serie seleccionada.

11. Observar Desempeño Personal:

- **Descripción:** El estudiante puede observar su propio desempeño y progreso en la plataforma.
- **Actores:** Estudiante
- **Flujo Principal:**
 - a) El estudiante accede a su panel personal.
 - b) El estudiante accede a una serie asignada.
 - c) El estudiante visualiza su historial de ejercicios completados, calificaciones y progreso general en la plataforma.

12. Enviar Soluciones:

- **Descripción:** El estudiante puede enviar soluciones a los ejercicios cargando archivos de código.
- **Actores:** Estudiante
- **Flujo Principal:**
 - a) El estudiante selecciona un ejercicio para el cual desea enviar una solución.

- b) El estudiante carga archivos de código (.java) desde su proyecto IntelliJ.
- c) El sistema evalúa los archivos cargados ejecutando los test unitarios asociados al ejercicio.

13. **Obtener Retroalimentación:**

- **Descripción:** El estudiante puede obtener retroalimentación sobre las soluciones que ha enviado.
- **Actores:** Estudiante
- **Flujo Principal:**
 - a) El estudiante accede a la vista del ejercicio para el cual envió una solución.
 - b) El estudiante observa la retroalimentación proporcionada por el sistema sobre su solución, incluyendo errores identificados y áreas de mejora.

9.2. **Comentarios de la encuesta**

A continuación se presentan las respuestas de los encuestados a la pregunta "¿Qué le ha parecido esta Nueva Versión de PA3P?"

- Más llamativa que la anterior, además de ser más fácil de navegar en ella. Entrega mejor feedback sobre los problemas o razón del por qué el código no corre.
- Me ha parecido muy buena y fácil de entender y de usar.
- Mejor, aunque el error de nombre de paquete debe ser solucionado.
- Me parece más fácil, moderna y agradable de utilizar que la versión anterior.
- Muy buena y mucho más descriptiva con respecto a la anterior.
- Me ha parecido una interfaz muy bonita en comparación con la anterior, creo que es un poco complicado la interfaz a la hora de darte ciertos errores, pero yo creo que con un poco de orientación por parte del docente no debería dar problemas.

- Mejor que la antigua, sobre todo porque no es tan sensible a cambios menores en la forma de ordenar el código. Me parece que en eso ha mejorado mucho, porque antes se requería que incluso no hubieran espacios en blanco de más o que luego de cerrar una llave se deba agregar un salto de línea.
- Me parece una mejora comparada con la anterior. La usaría más, ya que explica mejor los errores.
- Creo que está muy genial, la encuentro más simple y más atractiva de usar, pero es más difícil comprender los errores con facilidad las primeras veces, luego uno cacha.
- Es una versión atractiva y funcional, sin embargo, a la hora de retroalimentar se me dificultó reconocer los errores en el código, a mi parecer no era claro donde se encontraban los errores. Más allá de ello lo considero una buena plataforma que puede ser de mucha utilidad para apoyar y corregir a la hora de aprender a programar en Java.
- Me pareció muy agradable visualmente y cómoda, el diseño era agradable y estaba muy bien explicado donde estaba cada función a realizar.
- Me pareció muy buena la interfaz gráfica, pero el problema estaba un poco difícil de entender por la redacción y por lo mismo me costó un poco identificar los errores que tenía.
- Buena.
- Mejor que la anterior, más cómoda y clara. No tuve que hacer un gran esfuerzo para aprender a usarla.
- Su usabilidad es mucho mayor en cuanto a la versión anterior, su accesibilidad y claridad respecto a su uso también. Además, el feedback respecto a ejercicios subidos es mucho mayor.

Además, se adjuntan las respuestas de algunos encuestados a la sección "Otros Comentarios".

- Sería bueno que siguiera evolucionando esta plataforma.
- Explicitaría, al menos en lenguaje natural, los test que se aplicarán, para entender mejor los mensajes de error de los tests. Podría ser como un checklist de los tests que se aplicarán y deberían estar separados los tests de los errores de compilación, de la misma forma que los test unitarios generan un informe de qué funciona y qué no.
- Lo único que no me quedaba del todo claro eran la retroalimentación de los errores, quizás algo más explícito me hubiese ayudado un poco más, pero en general es fácil familiarizarse con la plataforma y sus usos.
- A veces cuesta entender el por qué de cada error ya que no se muestra el código que se ejecuta para las pruebas, sumado a que la redacción del problema suele ser ambigua.
- Está muy buena, debería ser usada.

9.3. Imágenes

Selecciona una Actividad a habilitar

id	descripcion	problemas										
99	actividad vacia											
100	ejecicios Java no OO	1	2	3	4	5	6	9001	9002	9003		
101	ejecicios creacion primeras clases	7	8	9	10	11	12	13	14			
102	ejecicios reutilizacion, composicion y encapsulamiento	19	20	21	22	23	24	25	26	27	28	
103	ejecicios reutilizacion, composicion, herencia	38	39	40	41	42	43					
150	test basico de contener versus conocer y pabellon de pintura de autos	51	52	53	54	55						
1000	revison tarea 1: sistema comprador-expendedor	101	102	103	104	105	106	107	108	109	110	
1200	test PA3P composicion polimorfica y delegacion	200	201	202	203	204	205	206	207			
1300	test PA3P conexion V invertida entre clases, polimorfismo formas geometricas	300	301	302	303							
5000	polimorfismo de bebidas 1 problema	5001										
6000	clases pelota, caja y pelota, tubo y pelotas	6001	6002	6003								
7000	clases de objetos que entregan secuencias circulares de digitos	7001	7002	7003	7004							
8000	caminos de ceros sobre clases con arreglos de 12x24	8001	8002	8003	8004	8005	8006	8007	8008			
9000	actividad para prueba de problemas nuevos	9001	9002	9003	9004	9005	9006	9007	9008	9009	9010	
9010	actividad con operaciones multiples arreglos tipo struct desde main	9004	9005	9006	9007	9008	9009	9010	9011	9012	9013	9014
9020	con clases que imprimen y metodos que trabaja a domicilio	9021	9022									
9030	problemas tipo GUI de matriz de char	9031	9032	9033	9034	9035	9036					
10000	camino sobre ceros de 12x24	19	20	10001	10002	10003	10004	10005	10006	10007	10008	

102 ▾ Habilitar

Volver a menu principal ▾

Figura 16: Captura de interfaz docente PA3P

```

Analisis dinamico:
metal: X

Resultados dinamicos:
se cambian los datos del arreglo de tu programa a:
{-382,70,339,-370,235,11};


| n | Tuyo         | Debe                    |
|---|--------------|-------------------------|
| 1 | -382,70,-370 | 70,339,-370,235,11,-382 |


Imprimes 11 caracter(es) menos de lo que debes
En la linea 1, posicion 1, imprimes en pantalla: '-' en lugar de: '7'.

Analisis estatico:
main ✓

Compilacion:
Compilacion exitosa de Problema_8.java -> ComoStruct.class Problema_8.class OK!

```

Figura 17: Captura de ejemplo ejercicio PA3P

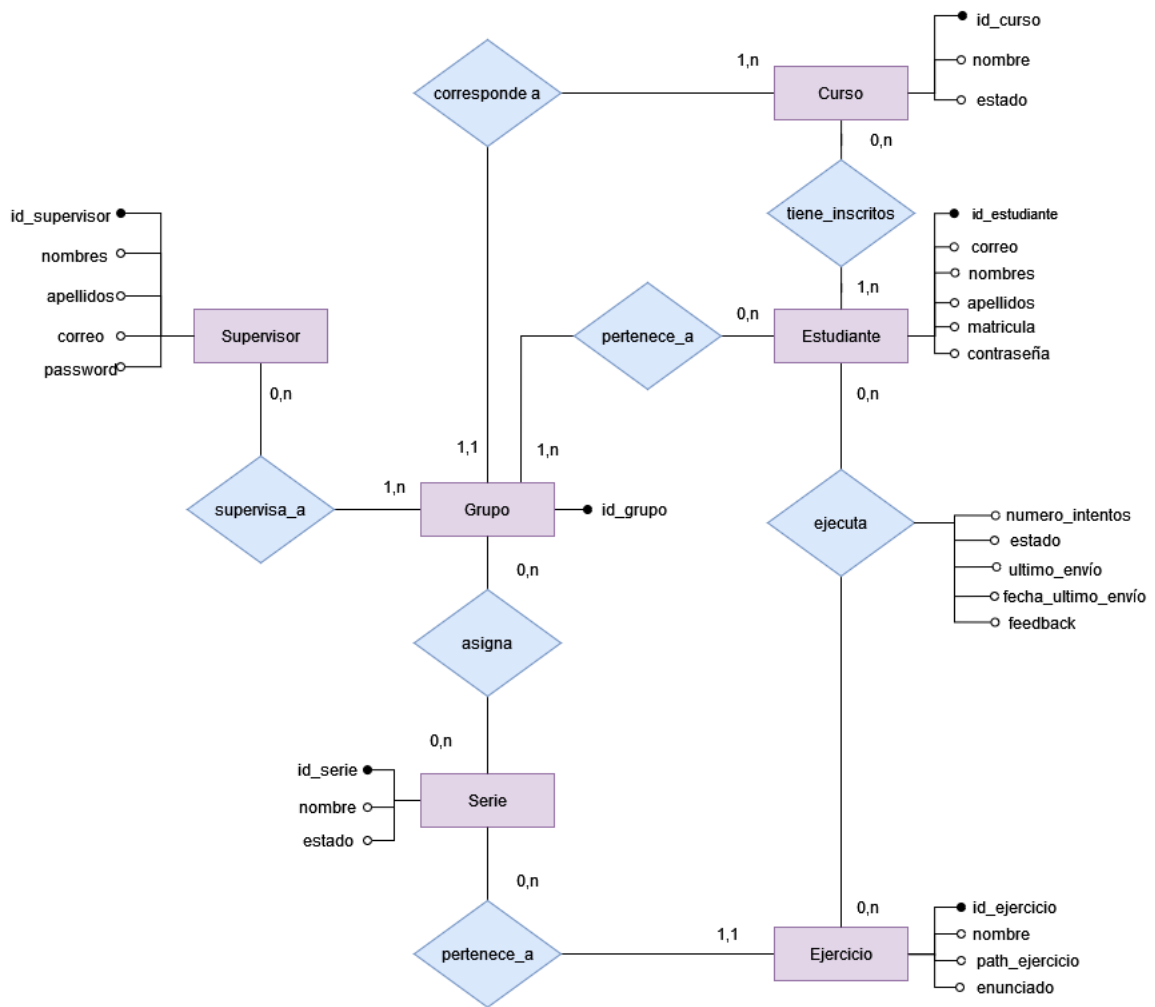


Figura 18: Modelo Entidad Relación (Completo)

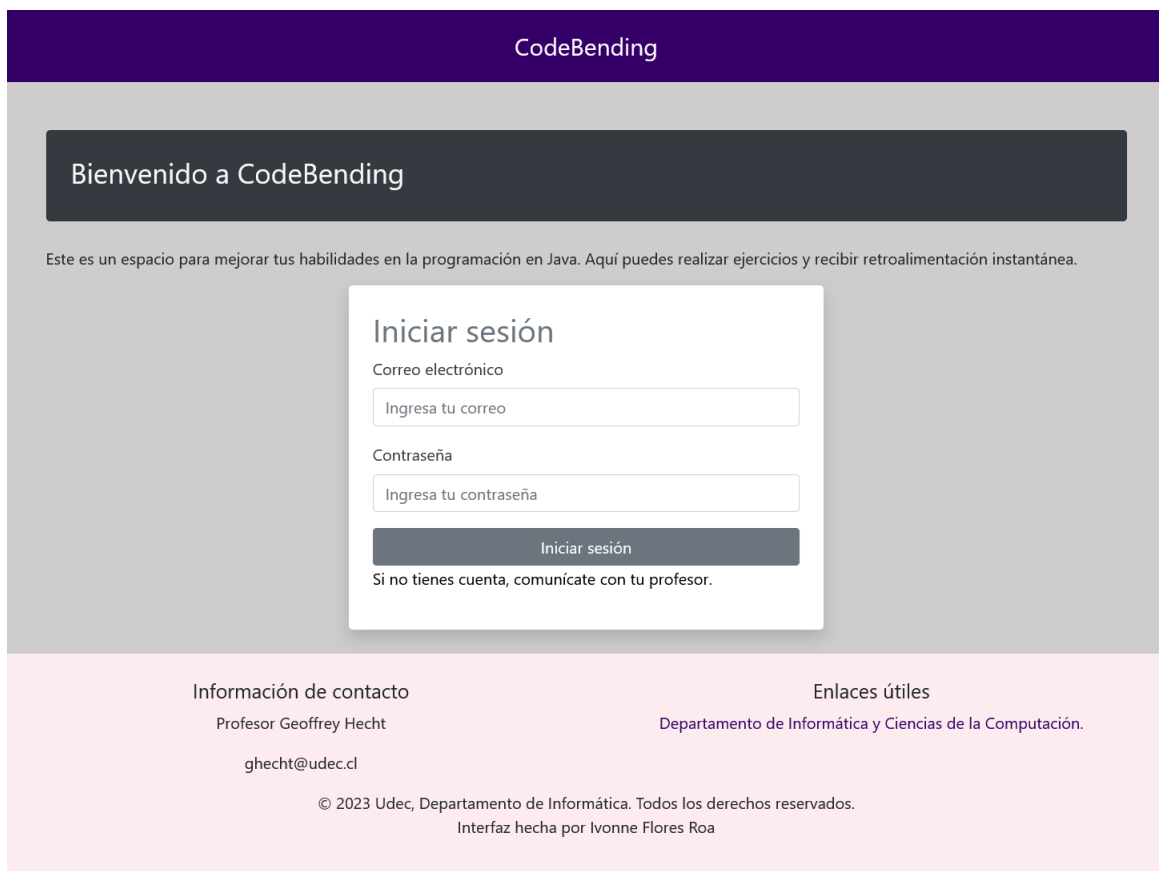


Figura 19: Captura del Login

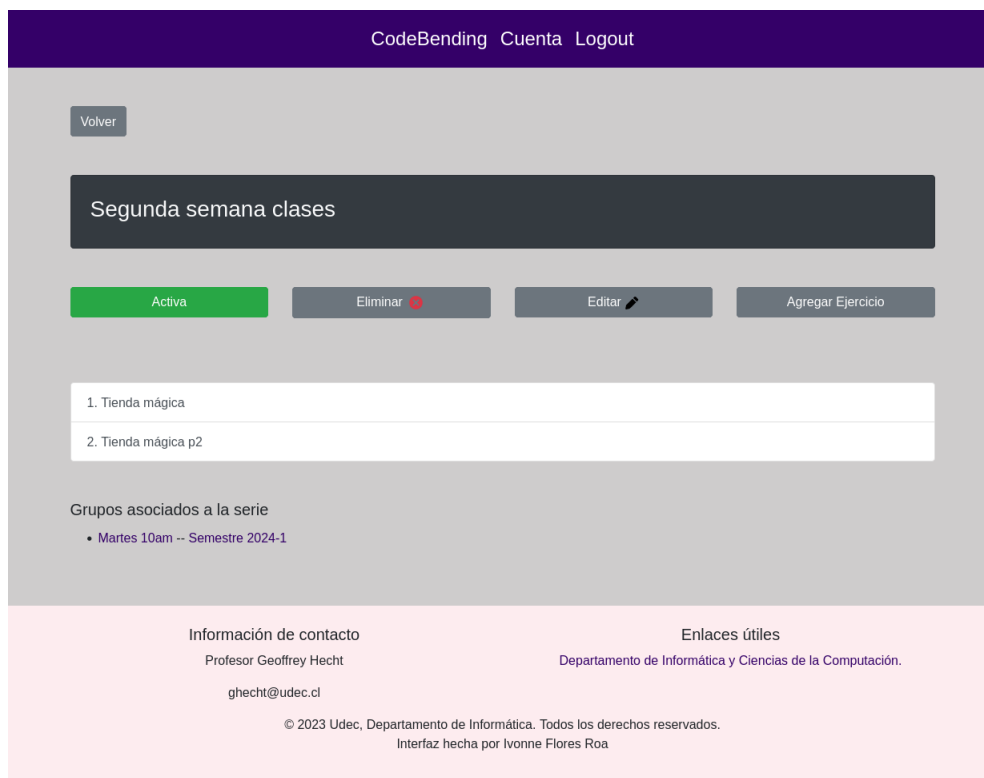


Figura 20: Captura de gestión docente de una serie de ejercicios

CodeBending Cuenta Logout

Volver

Semestre 2024-1

Activo Asignar nuevo grupo Ver progreso del curso Eliminar

Lista de clases
Daniel Ignacio Aburto Rivera
Rodrigo Alexis Bascuñan Leon
Antonio Jesus Benavides Puentes
Gustavo Alonso Benitez Cespedes
Bastian Alexander Bernal Villegas
Javier Alejandro Campos Contreras
Gabriel Sebastian Castillo Castillo
Andres Ignacio Chaparro Maldonado
Francisco Sebastian Cifuentes Ramirez
Ariel Eduardo Cisternas Bustos
Roberto Antonio Cruz Pinto
Santiago Alexander Diaz Barra
Matias Ignacio Figueroa Vasquez

Grupos del curso

Clase entera

Asignar serie a grupo

Series:

Introduccion

Grupos:

Clase entera

Asignar

Series en este curso

No hay series asignadas

Figura 21: Captura de vista de un curso

Usabilidad

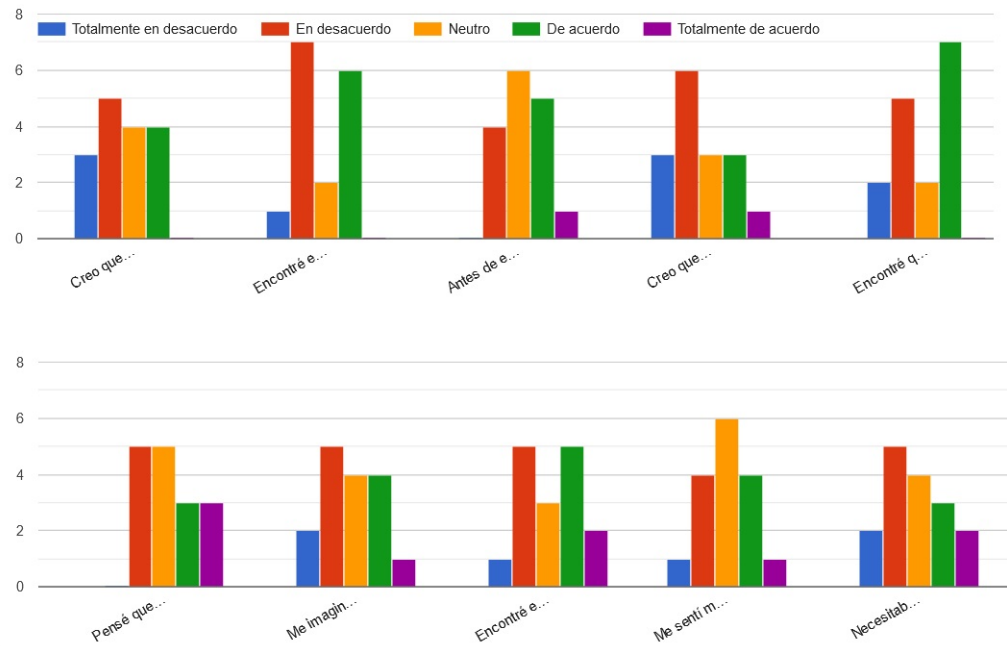


Figura 22: Resultados SUS PA3P

Usabilidad

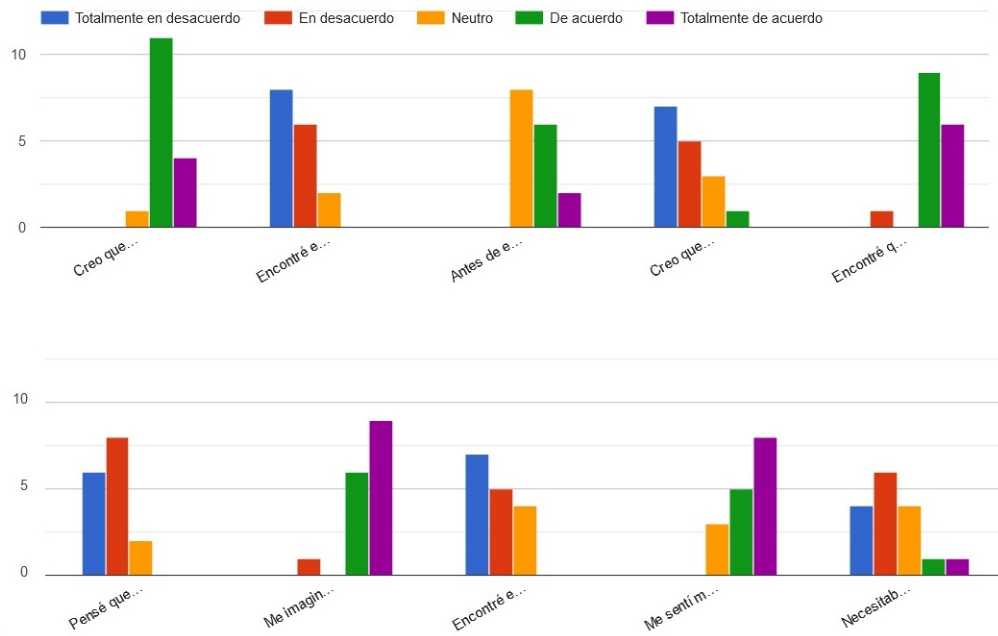
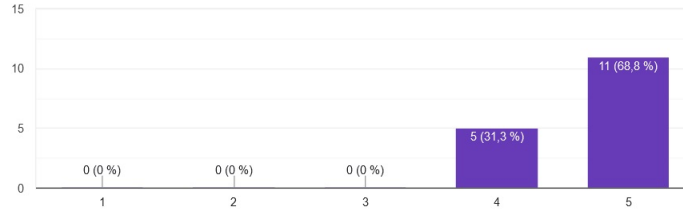


Figura 23: Resultados SUS CodeBending

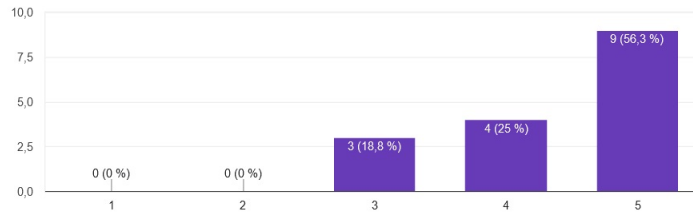
En comparación con la plataforma antigua, ¿qué tan atractiva encuentras la interfaz y el diseño general de la nueva plataforma de autoevaluación de código? [Copiar](#)

16 respuestas



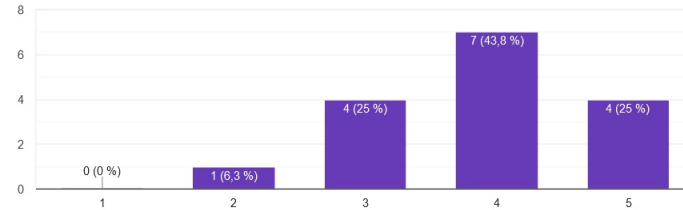
Comparando con la forma de inicio de sesión en la plataforma antigua, ¿sientes que la nueva forma de inicio de sesión ha mejorado la seguridad y privacidad de tus datos?

16 respuestas



Comparando con la retroalimentación en la plataforma antigua, ¿cómo describirías la retroalimentación proporcionada sobre tus soluciones de código en la nueva plataforma?

16 respuestas



En comparación con la plataforma antigua, ¿qué tan claro encuentras la identificación de errores en tus soluciones en la nueva plataforma?

16 respuestas

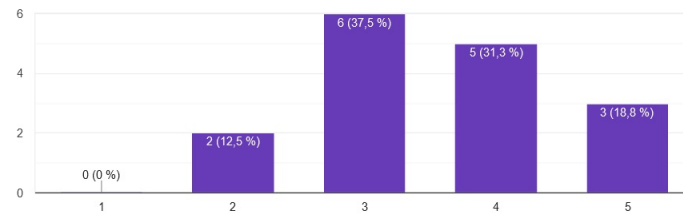


Figura 24: Resultados sección comparaciones

¿Qué versión de PA3P preferiría utilizar para un futuro curso?

16 respuestas

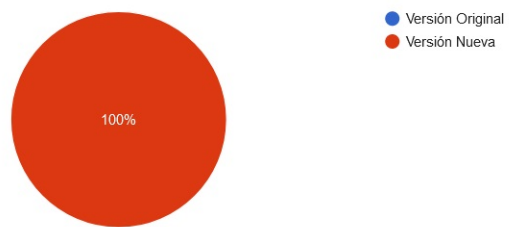


Figura 25: Versión que eligen los encuestados para futuros trabajos