

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Desarrollo de una versión mejorada del sistema AGATA de la ESPOL

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero/a en Ciencias de la Computación

Presentado por:

Luis Enrique Anchundia Solórzano

Scarlet Angelina Espinoza Moreno

GUAYAQUIL - ECUADOR

Año: 2022

DEDICATORIA

El presente proyecto se lo dedico a mis padres, que a pesar de que ya no estén conmigo, me dieron la educación y los valores necesarios para culminar mi carrera universitaria y convertirme en una profesional de la cual estarían orgullosos. Así mismo a mis hermanos, mi prima, madrina y amigas quienes han sido mi pilar durante todo este tiempo.

Scarlet Angelina Espinoza Moreno

El presente proyecto se lo dedico a mis padres, por siempre preocuparse por mi bienestar y por brindarme su apoyo incondicional durante mi carrera universitaria, por brindarme sus consejos y valores que hoy en día me convierten en un profesional. También se lo dedico a mi hermana y mi sobrina, a mi hermana por apoyarme siempre, a mi sobrina por siempre alegrarme el día con sus ocurrencias.

Luis Enrique Anchundia Solórzano

AGRADECIMIENTOS

Mi más sincero agradecimiento a Dios por cuidarme a lo largo de este camino, al Ing. David Jurado y al Ing. Erick Lavid Cedeño por el tiempo, los consejos y la ayuda brindado durante el desarrollo del proyecto. Finalmente, a la ESPOL por darme una excelente educación, de la cual me llevo sabiduría y conocimientos para mi vida profesional.

Scarlet Angelina Espinoza Moreno

Agradezco a Dios por permitirme culminar mi carrera de manera satisfactoria. También agradezco la excelente educación que me brindó ESPOLE y a cada uno de los profesores que me compartieron sus conocimientos desde el inicio hasta el fin de mi carrera que me permiten hoy ser un profesional.

Luis Enrique Anchundia Solórzano

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Luis Enrique Anchundia Solorzano* y *Scarlet Angelina Espinoza Moreno* damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

Luis Anchundia

Luis Enrique
Anchundia Solórzano

Scarlet Espinoza

Scarlet Angelina
Espinoza Moreno

EVALUADORES



Erick Vicente Lavid Cedeño, MSc.

PROFESOR DE LA MATERIA

David Alonso Jurado Mosquera, MSc.

PROFESOR TUTOR

RESUMEN

En la actualidad el proceso de pruebas de software es una de las actividades que tienen mayor importancia y relevancia durante el desarrollo de un proyecto debido a que permiten garantizar la calidad dentro de todo desarrollo. Dentro del contexto de la ESPOL, la facultad FIEC cuenta con un sistema llamado AGATA que posee varios módulos, entre estos se encuentran los módulos tutor, profesor, secretaria y subdecanato que deben ser sometidos a un proceso de pruebas de software para verificar su correcto funcionamiento. Se propone desarrollar pruebas de software unitarias y de integración que permitan corroborar el correcto funcionamiento de los módulos mencionados, también se propone utilizar una herramienta que cumpla con la función de automatizar las pruebas y por último analizar y mejorar la calidad del código. Las reuniones con el cliente permitieron definir los requerimientos de usuario y mediante el uso de un cronograma de tareas dividido por fechas y responsables se completó el desarrollo del proyecto. Esto requirió la selección, instalación y configuración de herramientas de software, reuniones semanales con presentación de avances y el desarrollo y automatización de pruebas. Como resultado los módulos mencionados anteriormente fueron probados a través de pruebas unitarias y de integración, las pruebas desarrolladas se ejecutan de manera automática y el análisis del código fue realizado con éxito. Se concluye que las pruebas desarrolladas, la automatización y el análisis de código efectuado permiten verificar el correcto funcionamiento de los módulos tutor, profesor, secretaria y subdecanato dentro del sistema AGATA.

Palabras Clave: Pruebas, Automatización, Análisis, Calidad, Funcionamiento

ABSTRACT

Currently, the software testing process is one of the most important and relevant activities during the development of a project because it allows to guarantee the quality of all development. Within the context of the ESPOL, the FIEC faculty has a system called AGATA that has several modules, among these are the tutor, professor, secretary, and subdeanery modules that must be subjected to a software testing process to verify its proper functioning. It is proposed to develop unit and integration software tests that allow to corroborate the correct operation of the mentioned modules, it is also proposed to use a tool that fulfills the function of automating the tests and finally to analyze and improve the quality of the code. The meetings with the client allowed us to define the user requirements and using a task schedule divided by dates and responsible parties, the development of the project was completed. This required the selection, installation and configuration of software tools, weekly meetings with presentation of progress and the development and automation of tests. As a result, the modules were tested through unit and integration tests, the developed tests are executed automatically, and the code analysis was successfully performed. It is concluded that the tests developed, the automation and the code analysis performed allow verifying the correct functioning of the tutor, teacher, secretary, and subdeanery modules within the AGATA system.

Keywords: *Testing, Automation, Analysis, Quality, Performance*

ÍNDICE GENERAL

RESUMEN	I
<i>ABSTRACT</i>	II
ÍNDICE GENERAL	III
ABREVIATURAS	VI
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	VIII
CAPÍTULO 1	1
1. Introducción	1
1.1 Descripción del problema	2
1.2 Justificación del problema	3
1.3 Objetivos	4
1.3.1 Objetivo General	4
1.3.2 Objetivos Específicos	4
1.4 Marco teórico	4
1.4.1 Django	4
1.4.2 React	4
1.4.3 Herramientas para Django	5
UnitTest Framework	5
1.4.4 Herramientas para React	6
1.4.5 Herramientas para análisis de código	6
1.4.6 Herramientas para automatización de pruebas de software	7
CAPÍTULO 2	8
2. Metodología	8

2.1 Análisis	8
2.1.1 Requerimientos funcionales	8
2.1.2 Requerimientos no funcionales	8
2.1.3 Alcance y limitaciones de la solución	8
2.1.4 Riesgos y beneficios de la solución	9
2.1.5 Usuarios de la solución	10
2.2 Prototipado	10
2.2.1 Herramientas seleccionadas para la solución	10
2.2.2 Flujo de eventos para las pruebas	10
2.3 Evaluación	11
2.4 Diseño de la solución	12
2.4.1 Diseño de pruebas	12
2.4.2 Construcción y ejecución de pruebas	13
2.4.3 Seguimientos y resultados	22
2.4.4 Corrección de errores	22
2.5 Consideraciones éticas y legales	22
CAPÍTULO 3	23
3. Resultados Y ANÁLISIS	23
3.1 Configuración para la automatización de pruebas	23
3.2 Determinación de funciones para pruebas unitarias y de integración.	24
3.3 Ejecución de pruebas	26
3.3.1 Resumen de las pruebas realizadas.	27
3.4 Análisis de la calidad del código y refactoring.	28
3.4.1 Proceso de <i>refactoring</i>	29
3.5 Análisis de costos.	35
3.6 Cierre de proyecto	36

CAPÍTULO 4	37
4. Conclusiones Y Recomendaciones	37
4.1 Conclusiones	37
4.2 Recomendaciones	37
BIBLIOGRAFÍA	38
5. Bibliografía	38
APÉNDICES.....	40

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
FIEC	Facultad de Ingeniería en Electricidad y Computación
AGATA	Aplicación web para la Gestión de Archivos de Titulación Agata
GTSI	Gerencia de Tecnologías y Sistemas de Información
API	Application Programming Interfaces
SAAC	Sistema de Administración Académica
STA	Secretaría Técnica Académica

ÍNDICE DE FIGURAS

Figura 2.1 Esquema gráfico de pruebas [autoría propia]	11
Figura 2.2 Flujo de la metodología de pruebas [autoría propia]	12
Figura 3.1 Proceso de automatización de pruebas [autoría propia]	23
Figura 3.2 Ejemplo de prueba para el FrontEnd [autoría propia].....	26
Figura 3.3 Código correspondiente a la prueba test_send_mail()[autoría propia].	27
Figura 3.4 Resumen de resultados del análisis de código con la herramienta SonarQube en el BackEnd [autoría propia]	28
Figura 3.5 Resumen de resultados del análisis de código con la herramienta SonarQube en el FrontEnd [autoría propia]	29
Figura 3.6 Bugs encontrados en el código que fueron corregidos [autoría propia]....	29
Figura 3.7 Resumen del grado de gravedad de code smells BackEnd [autoría propia]	30
Figura 3.8 Ejemplos de code smells que fueron resueltos [autoría propia]	31
Figura 3.9 Resumen del grado de gravedad de code smells FrontEnd [autoría propia]	32
Figura 3.10 Code Smells que fueron resueltos [autoría propia]	33
Figura 3.11 Resumen de resultados después del refactoring BackEnd [autoría propia]	33
Figura 3.12 Resumen del grado de gravedad después del refactoring [autoría propia]	34
Figura 3.13 Resumen de resultados después del refactoring FrontEnd [autoría propia]	34
Figura 3.14 Resumen del grado de gravedad después del refactoring [autoría propia]	35
Figura 6.10 Observaciones refactorizadas 42/73 [autoría propia].	107

ÍNDICE DE TABLAS

Tabla 2.1 Plantilla de pruebas	13
Tabla 2.2 Plantilla de control de versión	13
Tabla 3.1 APIs con sus respectivas funciones unitarias probadas [autoría propia] ...	24
Tabla 3.2 APIs y los respectivos métodos de integración probados [autoría propia].	25
Tabla 3.3 Ejemplo de documentación de pruebas [autoría propia]	27
Tabla 3.4 Resumen del total de pruebas ejecutadas [autoría propia]	28
Tabla 3.5 Funciones y clases refactorizadas [autoría propia]	31
Tabla 3.6 Tabla de costos [autoría propia].....	36

CAPÍTULO 1

1. INTRODUCCIÓN

Desde la antigüedad y hoy en día el proceso de pruebas de software es una de las actividades que tienen mayor importancia y relevancia durante el desarrollo de un proyecto, debido a que da lugar a procesos, normas de trabajo y herramientas necesarias que permiten garantizar la calidad dentro de cualquier desarrollo [1]. Las pruebas de software son un proceso o un conjunto de procesos diseñados con la finalidad de asegurar que un código computacional realiza aquello para lo cual fue elaborado y que no realice nada de manera involuntaria [2].

Para determinar un correcto funcionamiento en los programas o aplicaciones las pruebas de software tienen su esencia en determinar un conjunto de casos de prueba para los elementos que se van a probar. Un caso de prueba es el producto de un proceso conocido [3].

Dentro del contexto de la Escuela Superior Politécnica del litoral (ESPOL) específicamente en su facultad FIEC se ha desarrollado un sistema de gestión de documentos de titulación que tiene por nombre AGATA (Aplicación web para la gestión de Archivos de titulación Agata), dentro del mismo se encuentran los módulos tutor, profesor, secretaría y subdecanato, los cuales deben ser sometidos a pruebas de software, de tal manera que se pueda identificar posibles fallos en el sistema y proceder a la corrección de los mismos, de esta manera asegurar que el sistema cumple con el comportamiento esperado.

Con base en lo expuesto anteriormente el presente documento tiene por objetivo someter a pruebas de software al sistema AGATA perteneciente a la FIEC para asegurar la calidad y un correcto funcionamiento los módulos antes mencionados, de modo que pueda ser útil para optimizar la realización de actividades de cada actor involucrado.

Este documento se encuentra dividido en cuatro capítulos, los cuales se detallan de manera breve cada uno de ellos a continuación:

- Capítulo uno, se presenta la problemática encontrada, marco teórico, objetivos, justificación, alcance y limitaciones del proyecto.
- Capítulo dos, se describe la metodología utilizada para la resolución del problema.
- Capítulo tres, se presenta la implementación de la solución, resultados y análisis.
- Capítulo cuatro, se presentan conclusiones y recomendaciones.

1.1 Descripción del problema

Actualmente, la Facultad de Ingeniería en Electricidad y Computación (FIEC) de Espol está desarrollando un sistema llamado AGATA (Aplicación web para la gestión de Archivos de titulación Agata), que consta de 2 etapas, la primera etapa se encarga de guiar y dar acompañamiento al estudiante durante el proceso de titulación, con la intervención de otros usuarios como: tutor, secretaria y docente, quienes pueden acceder al sistema, visualizar los documentos entregados y proveerle una retroalimentación al estudiante; de este modo todos los actores involucrados en el proceso son alertados sobre el estado en el que se encuentran a través de notificaciones.

La segunda etapa se centra en permitirle a la secretaria llevar un control y acceso a los servicios que manejan la información no tan solo del estudiante, sino que también del tutor y profesor por medio de una API. El proceso inicia después de que el documento del proyecto haya sido verificado y aceptado durante la primera etapa, los estudiantes proceden a registrar sus documentos personales, la secretaria revisa que estos estén en orden y toma el acta de grado por medio del SAAC, por consiguiente, lo envía a firmar con subdecanato y finalmente lo entrega al STA para que este realice el trámite correspondiente para la emisión del título de grado del estudiante.

El sistema AGATA ya se encuentra en ambiente de producción empleando un servidor independiente para el FrontEnd y BackEnd respectivamente; por otro lado, la herramienta de desarrollo de lado del cliente que está empleando es

React mientras que para el BackEnd es Django en conjunto con PostgreSQL como gestor de base de datos. El sistema a más de ser desarrollado y corregido por estudiantes en semestres anteriores no ha sido sujeto a pruebas de software adecuadas, por lo que no se puede asegurar con exactitud, si el flujo de cada proceso en los módulos tutor, profesor, secretaria y subdecanato está funcionando como corresponde.

1.2 Justificación del problema

La ejecución de pruebas en software debe considerarse como una etapa fundamental durante el desarrollo de aplicaciones y más aún en procesos donde no se está seguro de como un componente se desempeña dentro de un sistema. Es complejo determinar a simple vista si un sistema es 100% perfecto, inclusive de haber sido desarrollado por buenos programadores puede presentar fallos, por ello todo software debe ser probado con el fin de garantizar un producto de calidad.

Existen dos tipos de pruebas de software funcionales y no funcionales. Las pruebas funcionales se encargan de verificar cada función de un software y se deben realizar con un conjunto de requisitos específicos para probar cómo se comporta el producto [1]. Por otro lado, las pruebas no funcionales permiten identificar que riesgos corre el producto tanto en desempeño como en rendimiento, se realizan con el fin de obtener información [4].

Determinar el tipo de estrategia de prueba que se aplicará conlleva inicialmente comprender los requerimientos y funcionalidades de un software. En el caso del sistema AGATA se maneja un flujo sucesivo en el que las acciones que realiza el usuario afectan al resto. Por lo tanto, para la FIEC es de suma importancia que el sistema opere sin fallos y que el proceso de titulación se realice correctamente de principio a fin. Por ello, se propone como solución realizar un plan de pruebas de software en el sistema AGATA que ayude a identificar posibles fallos en el mismo y corregirlos.

1.3 Objetivos

1.3.1 Objetivo General

Demostrar el correcto funcionamiento de los módulos tutor, profesor, secretaria y subdecanato del sistema AGATA aplicando un plan de pruebas de software.

1.3.2 Objetivos Específicos

- Definir las pruebas de software que se ejecutarán de acuerdo con los escenarios definidos.
- Identificar los errores presentes en el flujo de los módulos mencionados del sistema AGATA.
- Desarrollar una solución para cada fallo encontrado durante las pruebas realizadas.

1.4 Marco teórico

El sistema AGATA está desarrollado con dos herramientas, Django para el BackEnd y React para el FrontEnd.

1.4.1 Django

Django es un marco de trabajo web de alto nivel en Python que promueve un desarrollo de software rápido y limpio. Es gratuito y de código abierto. Es muy rápido en cuanto al desarrollo de aplicaciones, estas pueden ir del concepto hasta su completitud tan rápido como sea posible. Cuenta con herramientas necesarias para desarrollar tareas comunes en un ambiente web, también es muy seguro debido a que ayuda a los desarrolladores a evitar problemas de seguridad como SLQ injection, cross-site scripting, cross-site request forgery y clickjacking [5].

1.4.2 React

React sirve para la creación de interfaces de usuario interactivas de manera sencilla, está basado en componentes que poseen y controlan su propio

estado, estos se pueden convertir en interfaces de usuario complejas. React también actualiza y renderiza los componentes de las vistas en sus aplicaciones cuando los datos son modificados [6].

A continuación, se presentan herramientas que son utilizadas para la ejecución de pruebas de software para estas herramientas.

1.4.3 Herramientas para Django

UnitTest Framework

También conocido como PyUnit, el módulo de Python UnitTest se encuentra basado en el framework XUnit que fue diseñado por Kent Beck and Erich Gamma [7].

UnitTest da soporte a los siguientes conceptos [8]:

- Configuración de prueba: un test fixture hace mención de los pasos previos que se deben realizar antes de ejecutar una o más pruebas junto con las acciones de limpieza relacionadas.
- Caso de prueba: un test case es utilizado para verificar la respuesta de un conjunto bien definido de entradas.
- Conjunto de pruebas: todo conjunto de pruebas que deban ejecutarse juntas estará agrupado en un test suite.
- Ejecutor de pruebas: un test runner conduce la ejecución de las pruebas y entrega un resultado.

Pytest Framework

Pytest cuenta con una amplia lista de funcionalidades, por ejemplo, puede ejecutar desde pruebas pequeñas hasta pruebas de gran escala tales como pruebas funcionales de aplicaciones y librerías. Pytest a diferencia de UnitTest no necesita clases para desarrollar una prueba y es sencillo para comenzar. También posee comandos capaces de incrementar la productividad, por ejemplo, ejecuta solo aquellas pruebas que presenten fallos o ejecuta un conjunto conocido de pruebas identificadas por el nombre [9]. El principal uso que recibe Pytest es en la escritura de pruebas para APIs [10].

1.4.4 Herramientas para React

Jest

Es una herramienta empleada como marco de prueba, la cual se encarga de acceder al DOM (Document Object Model), buscar los archivos que contienen la extensión “.test“ y ejecutarlos. Cada programador con Jest puede implementar un sin número de pruebas para los diversos componentes que presente un sistema y así determinar si están trabajando correctamente [11].

React-testing-library

Es un conjunto de herramientas que ayuda a renderizar un componente mediante el DOM, posteriormente se podrá interactuar con el componente y verificar si en este funciona como debe sin la necesidad de emplear un navegador web [12].

1.4.5 Herramientas para análisis de código

Refactoring

El principal objetivo de esta técnica es mejorar la calidad del código de un sistema de tal manera que el software sea comprensible, extensible y mantenible sin afectar su comportamiento o funcionalidad [13].

Existen varias herramientas para analizar la calidad del código, a continuación, se muestran tres herramientas junto a sus características:

SonarQube

Es la herramienta más reconocida entre los desarrolladores cuando se trata de analizar la seguridad y calidad en el código. SonarQube es capaz de generar resultados para más de 25 lenguajes de programación, siendo este número más alto en comparación con otras herramientas disponibles [14]. Cuenta con los siguientes beneficios:

- Solo necesita una línea de código para integrarse en las canalizaciones de integración continua (CI) y entrega continua (CD).
- Analiza calidad del código, formato, declaraciones de variables, manejo de excepciones y más.
- Capaz de integrarse al ciclo de compilación de Maven y Gradle.

DeepScan

Excelente para mejorar la calidad de código en JavaScript brindando un análisis estático sin ruido. Es capaz de encontrar errores en tiempo de ejecución [15]. Entre sus beneficios se encuentran los siguientes:

- Cuenta con una vista gráfica para apreciar el escaneo de datos a través del tiempo.
- Auto-escanea repositorios.
- Fácil integración a un repositorio de GitHub con sincronización automática.

CodeSonar

Es una herramienta de análisis de código estadístico debido a que es capaz de generar modelos a partir del código, busca posibles amenazas de ejecución tales como interbloqueos, desbordamiento de memoria, punteros nulos, fugas de datos entre otros errores de programación que por lo general pueden ser difíciles de detectar [14]. Entre sus beneficios se encuentran los siguientes:

- En comparación con otras herramientas similares, el análisis del código es más profundo.
- Puede detectar de 3-5 veces más errores que otras herramientas.

1.4.6 Herramientas para automatización de pruebas de software

Jenkins

Se trata de un servidor de automatización, es de código abierto y autosuficiente. Puede ser utilizado en todo tipo de acciones que involucren la creación, despliegue, pruebas o la entrega de software [16].

CAPÍTULO 2

2. METODOLOGÍA

2.1 Análisis

En base a la entrevista realizada al cliente, se determinaron los requerimientos que resolverán las necesidades puntuales que el cliente tiene sobre el sistema AGATA.

2.1.1 Requerimientos funcionales

En este proyecto no se contemplan este tipo de requerimiento debido a que no se está agregando nueva funcionalidad, sino que se comprobará que las funciones ya existentes se estén ejecutando correctamente.

2.1.2 Requerimientos no funcionales

- El sistema deberá ser sometido a un plan de pruebas de software que incluya pruebas funcionales (unitarias e integración).
- El sistema debe ser analizado por una herramienta de análisis de código para asegurar su calidad.
- Las pruebas de software deberán ejecutarse de manera automática cuando existan cambios en el sistema.
- Mejorar la calidad del código mediante la técnica de refactoring para los respectivos módulos establecidos.
- Se deberá corregir errores que surjan de las pruebas aplicadas al sistema.

2.1.3 Alcance y limitaciones de la solución

Alcance

El proyecto permitirá desarrollar y ejecutar un plan de pruebas sobre los módulos tutor, profesor, secretaria y subdecanato del sistema AGATA, las pruebas que se realizarán únicamente serán de tipo unitarias y de integración. En caso de que una prueba falle debido a un error presente se procederá a

corregirlo siempre y cuando forme parte de las funcionalidades existentes de los módulos establecidos. Así mismo se evaluará la calidad del código se mediante la herramienta SonarQube para finalmente mejorarla mediante la técnica de refactoring.

Limitaciones

- La solución no incorpora pruebas para otros módulos que se encuentren implementados dentro del sistema.
- No se ejecutarán otros tipos de pruebas que no formen parte del alcance establecido.
- Los errores que surjan por servicios externos utilizados no se solucionarán.
- Únicamente se ejecutarán las pruebas de acuerdo con los escenarios escogidos por el cliente.

2.1.4 Riesgos y beneficios de la solución

Beneficios

- El sistema será probado en partes unitarias para verificar el correcto funcionamiento de cada una de ellas.
- Las pruebas integración permitirán comprobar si todas las partes unitarias del sistema funcionan en conjunto según lo diseñado.
- La automatización de las pruebas permitirá ahorrar tiempo en la búsqueda de errores durante el desarrollo de nuevas funcionalidades.

Riesgos

- Que las pruebas de software a realizarse no logren detectar errores importantes en el código y esto afecte la funcionalidad del sistema.
- Que las pruebas de software a realizarse no logren detectar errores importantes en el código y esto afecte la funcionalidad del sistema.

2.1.5 Usuarios de la solución

- **Personal administrativo:** podrán dar uso al sistema en los módulos que incluyan documentación personal y firma de documentos ya sean de parte de la secretaria o de subdecanato.
- **Docentes:** participarán en el sistema de acuerdo con el rol asignado, este rol puede ser como tutor o profesor, se encargarán de revisar y aprobar la documentación de índole académico que puede ser trabajo final o trabajo de investigación.
- **FIEC:** hará uso del sistema para mejorar y automatizar su proceso de titulación dentro de la facultad.

2.2 Prototipado

2.2.1 Herramientas seleccionadas para la solución

Para realizar las pruebas de software en el FrontEnd que está implementado con React se empleará Jest en conjunto con react-testing library, mientras que por el lado del BackEnd que está implementado en Django se utilizará Pytest, ambas herramientas fueron escogidas debido a su facilidad de uso y la cantidad de plugins que estas poseen.

Para ejecutar el análisis del código se optó por utilizar la herramienta SonarQube debido la gran cantidad de lenguajes que soporta a la hora de realizar análisis.

Por último, para la automatización de las pruebas de software se utilizará la herramienta Jenkins debido a que esta herramienta cumple con el requerimiento solicitado.

2.2.2 Flujo de eventos para las pruebas

En la Figura 2.1 se puede apreciar una idea de cómo se ejecutaría el proceso, en primera instancia se crean las pruebas unitarias y de integración de las cuales se obtendrá un resultado de éxito o falla, en caso de fallar se procederá a descubrir el error y corregirlo para posteriormente volver a ejecutar el

proceso y sea aprobada. Estas pruebas serán implementadas en un servidor perteneciente a la Espol y a su vez este servidor estará conectado con la herramienta de automatización Jenkins para posteriormente ejecutar las pruebas de manera automática.

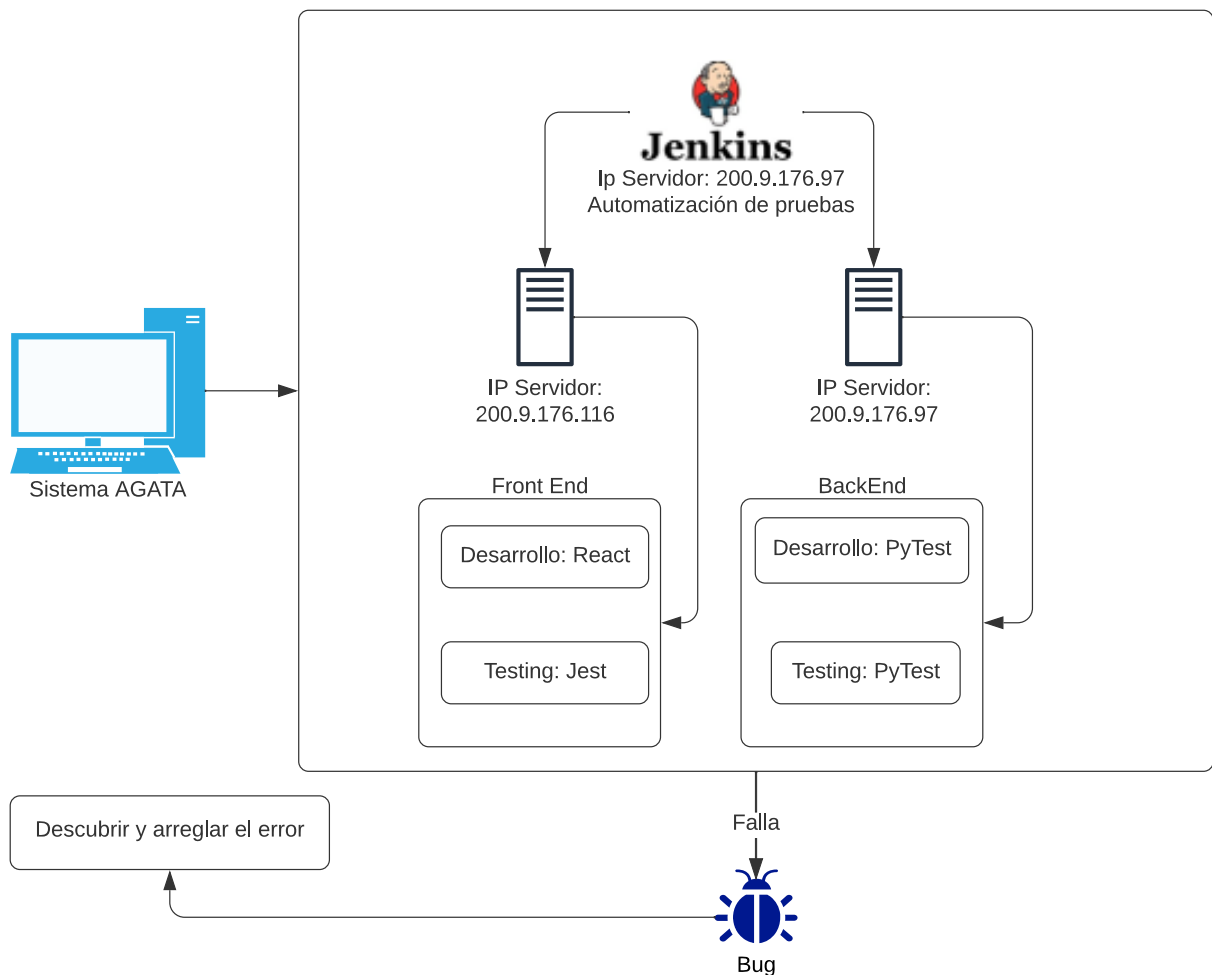


Figura 2.1 Esquema gráfico de pruebas [autoría propia]

2.3 Evaluación

Luego de las reuniones con el cliente se ha logrado obtener un total de 3 revisiones de los escenarios de prueba que se emplearán para la implementación de las pruebas de los módulos definidos en el alcance. El cliente enfatizó tomar en cuenta el proceso de envío y recibo de las notificaciones para

cada usuario dentro del flujo así mismo corroborar que se creen correctamente los semestres en el sistema correctamente para ambos casos los escenarios poseen sus respectivos prerrequisitos y postrequisitos que permitirá solventar dicha solicitud.

2.4 Diseño de la solución

El presente trabajo se centra generación y ejecución de pruebas como medio de verificación y validación del correcto funcionamiento de los módulos establecidos. En base a lo mencionado anteriormente, se propone una metodología basada en cuatro etapas:

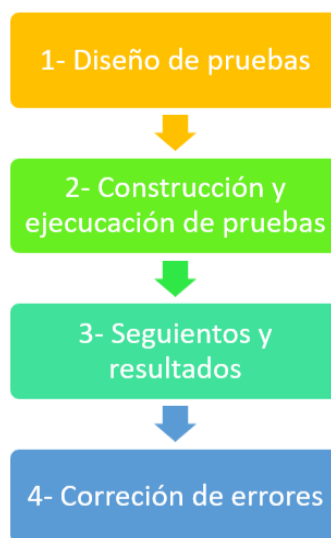


Figura 2.2 Flujo de la metodología de pruebas [autoría propia]

2.4.1 Diseño de pruebas

Gestión de plantillas de pruebas

Conocer las herramientas y estructuras que se emplearán para iniciar un proceso requiere de un conocimiento previo que se puede lograr mediante una correcta gestión de recursos. En el desarrollo de software, es indispensable gestionar y llevar un control de lo que se esté diseñando o implementando a través de la documentación, ya que con esta se podrá

respaldar no tan solo los requerimientos básicos, sino que hasta los archivos que en cada etapa se van generando [17].

Para este proyecto las plantillas serán los documentos que se usarán como guía y control de las pruebas que se ejecutarán en los respectivos módulos que a futuro proporcionará conocimiento a nuevos desarrolladores o personal del sistema sobre las características que fueron probadas y el responsable de aquello. A continuación, se presenta las plantillas que se empelarán en el proceso de pruebas:

Tabla 2.1 Plantilla de pruebas

Sistema	SISTEMA AGATA		
Entregable	Título de la prueba a realizar		
Autor	<Nombre del responsable>		
Versión/Edición	0100	Fecha Versión	DD/MM/AAAA
Aprobado por	<Nombre de la persona que aprueba el documento>	Fecha Aprobación	DD/MM/AAAA

Tabla 2.2 Plantilla de control de versión

Versión	Descripción	Autor	Fecha del Cambio
0100	Versión inicial	<Nombre Apellido1 Apellido2>	DD/MM/AAAA

2.4.2 Construcción y ejecución de pruebas

Escenarios de prueba

Supuesto: El estudiante carga sin problema alguno los documentos (titulación/investigación según sea el caso y agregar adjuntos, cuando se devuelve la documentación se puede modificar los docs adjuntos si se suben de nuevo se reemplazan) y el estado cambia a pendiente.

Solo el profesor y tutor mantienen un ciclo de revisión y cuando finalmente lo aprueben, pasa revisión final por parte de secretaría y si este no es aceptado en este paso se reinicia el proceso.

Consideración: tener aprobada la materia integradora.

Escenarios para el tutor

Escenario 1:	El tutor recibe la documentación para que esta sea revisada.
Prerrequisito:	<ul style="list-style-type: none"> • El estudiante carga los documentos requeridos. • El estado del sistema debe ser Pendiente. • El tutor recibe una notificación por correo de que los documentos han sido subidos.
Acción:	Revisar la documentación.
Resultado 1:	El contenido es el adecuado.
Resultado 2:	El contenido no es el adecuado.
Postrequisito 1 del resultado 2:	Envía un comentario de lo que no está correcto en el documento y lo devuelve al estudiante.
Postrequisito 2 del resultado 2:	Se devuelve un documento corregido al estudiante.
Estado del proceso del resultado 1:	Aprobado
Estado del proceso del resultado2:	Devuelto

Escenarios para el profesor

Escenario 2:	El profesor recibe la documentación previamente aprobada por el tutor para que esta sea revisada.
Prerrequisitos:	<ul style="list-style-type: none"> • La documentación debe estar aprobada por el tutor. • El estado del sistema debe ser Aprobado.

	<ul style="list-style-type: none"> El profesor recibe una notificación por correo.
Acción:	Revisar la documentación.
Resultado 1:	El contenido es el adecuado.
Resultado 2:	El contenido no es el adecuado.
Postrequisito 1 del resultado 2:	Envía un comentario de lo que no está correcto en el documento y lo devuelve al estudiante.
Postrequisito 2 del resultado 2:	Se devuelve un documento corregido al estudiante.
Estado del proceso del resultado 1:	Aprobado
Estado del proceso del resultado 2:	Devuelto.

Escenarios para la secretaria

Escenario 3:	La secretaria recibe la documentación previamente aprobada por el profesor para que esta sea revisada.
Prerrequisito:	<ul style="list-style-type: none"> La documentación debe estar aprobada por el tutor y el profesor. El estado del sistema debe ser Aprobado. La secretaria recibe una notificación por correo.
Acción:	Revisar la documentación.
Resultado 1:	La documentación está correcta.
Resultado 2:	La documentación no está correcta.
Postrequisito 1 del resultado 2:	<ul style="list-style-type: none"> Envía un comentario de lo que no está correcto en el documento y lo devuelve al estudiante. Se reinicia el proceso de revisión.
Postrequisito 2 del resultado 2:	<ul style="list-style-type: none"> La documentación debe estar aprobada por el tutor y el profesor. El estado del sistema debe ser Aprobado. La secretaria recibe una notificación por correo.
Estado del proceso del resultado 1:	Aprobado

Estado del proceso del resultado 2:	Devuelto
--	----------

Escenarios para la secretaria de manera individual.

Escenario 4:	La secretaria inicia el proceso de titulación y establece las fechas de inicio y fin.
Prerrequisito:	El estudiante debe haber culminado la malla curricular.
Acción:	Iniciar el proceso de recepción de documentos.
Resultado 1:	El sistema permite que los estudiantes puedan ingresar documentación del trabajo final.
Resultado 2:	El sistema no permite que los estudiantes puedan ingresar documentación del trabajo final.
Postrequisito	Ninguno.
Estado del proceso resultado 1:	Habilitado
Estado del proceso resultado 2:	Detenido.

Supuesto: El estudiante carga su hoja de vida, escribe su nombre como quiere que aparezca en el título y selecciona el tipo de mención (solo para el caso de investigación) y el estado cambia a pendiente. Se debe indicar que tiene que actualizar los datos en el académico en línea, debe haber subido el título de bachiller.

Escenario 5:	La secretaria revisa la hoja de vida y nombre ingresado por el estudiante.
Prerrequisito:	El estudiante debe tener actualizados los datos en el académico en línea.
Acción:	Revisar la documentación
Resultado 1:	La documentación está correcta.
Resultado 2:	La documentación no está correcta.

Posrequisito del resultado 1:	<ul style="list-style-type: none"> • La secretaria aprueba los documentos. • Se da paso al siguiente proceso que corresponde a extracción del acta.
Postrequisito del resultado 2:	<ul style="list-style-type: none"> • Envía un comentario de lo que no está correcto en el documento y lo devuelve al estudiante.
Estado del proceso del resultado 1:	Aprobado
Estado del proceso del resultado 2:	Devuelto

Escenario 6:	La secretaria aprueba los documentos personales en el sistema.
Prerrequisito:	Ninguno
Acción:	Unir el acta de grado con la hoja que contienen el nombre como aparecerá en el título.
Resultado:	Se habilita para subir al DSPACE.
Postrequisito:	Se extrae el acta y se desbloquea la deuda de no valor de manera automática.
Estado del proceso:	Habilitado

Escenario 7:	La secretaria sube los documentos al DSPACE.
Prerrequisito:	Los documentos deben haber sido previamente aprobados.
Acción:	Subir al DSPACE
Resultado 1:	Los documentos son cargados correctamente.
Resultado 2:	Los resultados no son cargados correctamente.
Postrequisito del resultado 1:	Se sube en el repositorio para consultarlos posteriormente y se habilita para enviar el acta a Subdecanato para su firma.
Postrequisito del resultado 2:	No se logra subir en el repositorio del DSPACE (queda pendiente), se habilita para enviar el acta a Subdecanato para su firma.

Estado del proceso del resultado 1:	Habilitado
Estado del proceso del resultado 2:	Habilitado

Escenario 8:	La secretaria envía el acta al subdecanato.
Prerrequisito:	Ninguno.
Acción:	Enviar acta a subdecanato
Resultado 1:	Subdecanato puede visualizar el acta para firmarla.
Postrequisito:	Ninguno
Estado del proceso:	Habilitado

Supuesto: El acta firmada por el subdecanato fue devuelta a la secretaria correctamente.

Escenario 9:	La secretaria sube el acta al SAAC.
Prerrequisito:	El acta debe estar firmada por subdecanato.
Acción:	Subir el acta al saac
Resultado 1:	El acta que fue recibida de subdecanato es subida al SAAC a través de AGATA.
Postrequisito:	Ninguno
Estado del proceso:	Finalizado

Escenarios para el Subdecanato.

Escenario 10:	El subdecanato recibe el acta que fue enviada por la secretaria.
Prerrequisito:	El acta debe ser enviada por la secretaria lista para ser firmada.

Acción:	Recibir el acta.
Resultado 1:	El acta llega correctamente y se puede descargar.
Resultado 2:	El acta no llega correctamente por lo que no se puede descargar.
Postrequisito del resultado 1:	Firmar el acta.
Postrequisito del resultado 2:	Notificar a la secretaria.
Estado del proceso del resultado 1:	Finalizado
Estado del proceso del resultado 2:	Pendiente

Escenario 11:	Cargar al sistema el acta firmada por parte del subdecanato y enviarla a la secretaria
Prerrequisito:	Haber firmado el acta.
Acción:	Enviar el acta a la secretaria.
Resultado 1:	El acta llega correctamente a la secretaria.
Resultado 2:	El acta no llega correctamente a la secretaria.
Postrequisito del resultado 1:	Descargar el acta.
Postrequisito del resultado 2:	Notificar a subdecanato.
Estado del proceso del resultado 1:	Finalizado
Estado del proceso del resultado 2:	Pendiente

Cronograma de actividades

ACTIVIDADES	Encargado	Fecha Inicio	Fecha Fin
Configurar el ambiente de pruebas y el servidor	Luis Anchundia Scarlet Espinoza	4/7/2022	8/7/2022
Realizar la configuración para la automatización de pruebas de software	Luis Anchundia Scarlet Espinoza	11/7/2022	15/7/2022
Pruebas unitarias			
Escenario 1: El tutor recibe la documentación para que esta sea revisada.	Luis Anchundia	18/7/2022	19/7/2022
Escenario 2: El profesor recibe la documentación previamente aprobada por el tutor para que esta sea revisada.	Luis Anchundia	20/7/2022	21/7/2022
Escenario 3: La secretaria recibe la documentación previamente aprobada por el profesor para que esta sea revisada.	Luis Anchundia	22/7/2022	25/7/2022
Escenario 4: La secretaria inicia el proceso de titulación y establece las fechas de inicio y fin.	Luis Anchundia	26/7/2022	27/7/2022
Escenario 5: La secretaria revisa la hoja de vida y nombre ingresado por el estudiante.	Luis Anchundia	28/7/2022	29/7/2022
Escenario 6: La secretaria aprueba los documentos personales en el sistema.	Luis Anchundia	1/8/2022	2/8/2022
Escenario 7: La secretaria sube los documentos al DSPACE.	Scarlet Espinoza	18/7/2022	19/7/2022
Escenario 8: La secretaria envía el acta al subdecanato.	Scarlet Espinoza	20/7/2022	21/7/2022
Escenario 9: La secretaria sube el acta al SAAC.	Scarlet Espinoza	22/7/2022	25/7/2022
Escenario 10: El subdecanato recibe el acta que fue enviada por la secretaria.	Scarlet Espinoza	26/7/2022	27/7/2022
Escenario 11: Cargar al sistema el acta firmada por parte del subdecanato y enviarla a la secretaria.	Scarlet Espinoza	28/7/2022	29/7/2022

Pruebas de integración			
Comprobar que el módulo tutor completa su flujo de actividades de manera exitosa.	Luis Anchundia	3/8/2022	4/8/2022
Comprobar que el módulo profesor completa su flujo de actividades de manera exitosa.	Luis Anchundia	5/8/2022	8/8/2022
Comprobar que el módulo secretaria completa su flujo de actividades de manera exitosa.	Scarlet Espinoza	3/8/2022	4/8/2022
Comprobar que el módulo subdecanato completa su flujo de actividades de manera exitosa.	Scarlet Espinoza	5/8/2022	8/8/2022
Refactoring			
Analizar la calidad del código en los módulos tutor, profesor, secretaria y subdecanato	Luis Anchundia Scarlet Espinoza	9/8/2022	9/8/2022
Refactorizar el módulo tutor	Luis Anchundia	10/8/2022	12/8/2022
Refactorizar el módulo profesor	Luis Anchundia	15/8/2022	17/8/2022
Refactorizar el módulo secretaria	Scarlet Espinoza	10/8/2022	12/8/2022
Refactorizar el módulo subdecanato	Scarlet Espinoza	15/8/2022	17/8/2022

Configuración del ambiente de producción

El desarrollo de las pruebas de software se realizará en el servidor local de las máquinas de los desarrolladores (localhost), para ello se deberá instalar todas las dependencias y requerimientos indicados en versiones anteriores de este proyecto. Una vez las pruebas hayan sido desarrolladas de manera local, serán trasladadas a los servidores de prueba en donde se encuentran ubicados los archivos que ejecutan el sistema AGATA, se cuenta con dos servidores: un servidor dedicado al FrontEnd alojado en la dirección IP: 200.9.176.166 y el otro servidor dedicado al BackEnd alojado en la dirección IP: 200.9.176.97.

2.4.3 Seguimientos y resultados

Una vez que las pruebas de software hayan sido desarrolladas y ejecutadas se procederá a realizar un análisis para determinar que pruebas se realizaron con éxito y cuales presentaron errores en su ejecución, las pruebas serán clasificadas en dos grupos: aprobadas y fallidas.

2.4.4 Corrección de errores

De acuerdo con los resultados obtenidos en el paso anterior, aquellas pruebas que hayan sido identificadas con errores serán puestas a revisión para proceder a realizar las correcciones necesarias en el código del sistema AGATA. Una vez que la corrección de errores haya finalizado, se volverán a ejecutar dichas pruebas que presentaron errores de tal manera que todas las pruebas pasen al estado de aprobadas.

2.5 Consideraciones éticas y legales

En el presente proyecto se apegará de acuerdo Código de Ética y Práctica Profesional de Ingeniería del Software aprobado por ACM (Association for Computing Machinery) y por la IEEECS (IEEE Computer Society). Puntualizando que los desarrolladores de este proyecto no podrán hacer uso del sistema AGATA de manera ilegal y sin autorización del cliente, así mismo mantener confidencial la información proporcionada como claves y datos de usuario o servicios proporcionados.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

3.1 Configuración para la automatización de pruebas.

Para la automatización de pruebas se empleó la herramienta Jenkins, la cual se implementó en el servidor 200.9.179.97, en donde también se encuentra alojado el BackEnd del sistema.

La automatización consistió en enlazar los repositorios de GitHub que contienen los archivos del sistema Agata tanto del BackEnd y el FrontEnd con el servidor de Jenkins, de tal manera que cada vez que se realice un nuevo cambio en los archivos con su respectivo commit en GitHub, Jenkins de manera automática hará una compilación del sistema y también ejecutará las pruebas que han sido creadas, de esta manera es posible evitar que el desarrollo de nuevas funcionalidades interfieran con la funcionalidad del sistema ya probado.

El proceso de instalación, configuración e inicialización véase en el apéndice B.

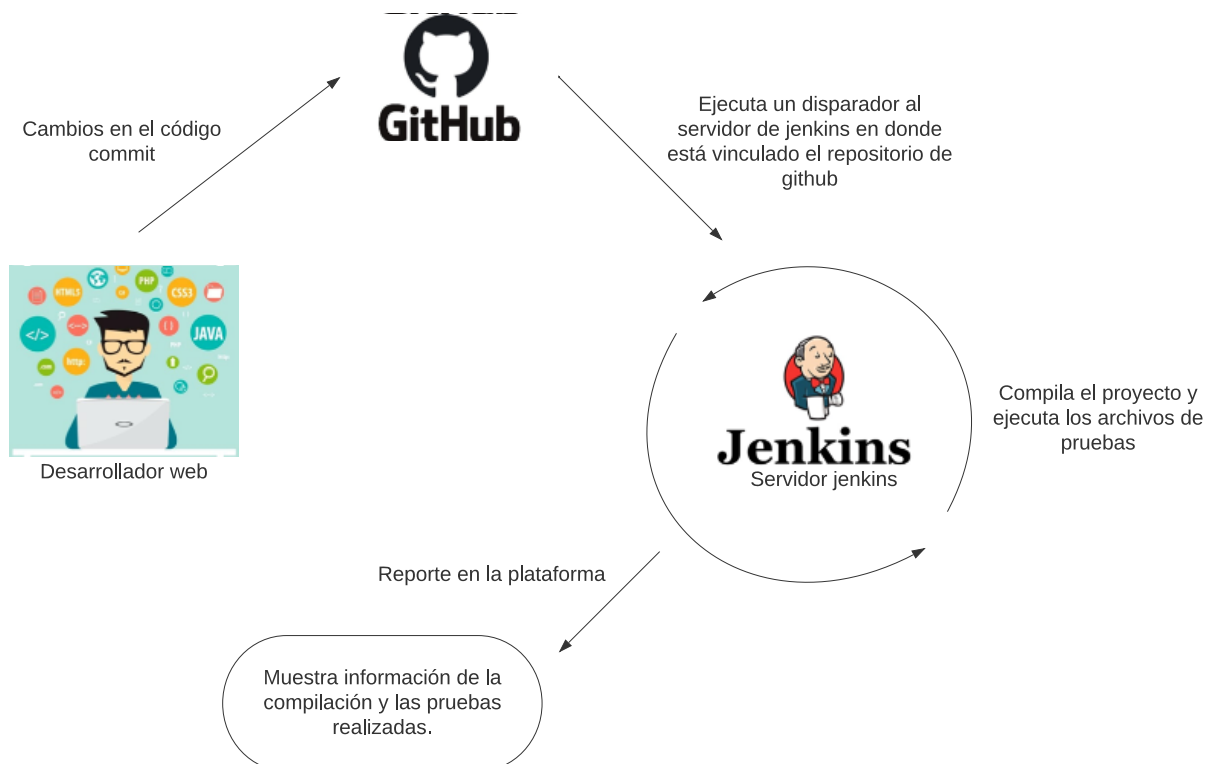


Figura 3.1 Proceso de automatización de pruebas [autoría propia]

3.2 Determinación de funciones para pruebas unitarias y de integración.

De acuerdo con los escenarios de prueba descritos en el capítulo 2 de este documento, se determinaron las APIs involucradas en los respectivos escenarios y se definieron como pruebas unitarias a las funciones que internamente emplean los métodos get y post para cada API, en la Tabla 3.1 se detalla lo mencionado.

Tabla 3.1 APIs con sus respectivas funciones unitarias probadas [autoría propia]

Escenario	Ambiente	API	Funciones
1, 2,3,5,6,7	Backend	api/documento/aprobar api/documento/tamano/<int:tipo>	estadoRevision () get_info_proyecto_individual () turno_aprobar () transformar () send_mail () transformar ()
1	Backend	api/documento/devolver	notificar_tutor ()
2	Backend	api/documento/devolver	notificar_profesor ()
3	Backend	api/documento/devolver	notificar_secretaria ()
4	BackEnd	api/secretaria/fechas/historico/<int:id_peri odo>	ultimo_termino () obtener_rol_termino ()
		api/secretaria/fechas/periodos_disponibl es/<int:id_peri odo>	
7	Backend	api/secretaria/<str:matricula>/proyecto/< str:id>/<str:usuario>	ultimo_termino () obtener_rol_termino () obtenerNombreEtapa ()
8	BackEnd	api/documento/acta/<str:tipo>/<str:matri cula>	buscarActaSAAC()
9	BackEnd	api/subdecanato	buscarActaGTSI()

10	BackEnd	api/subdecanato/estudiante/<str:matricula>	obtenerMatriculaSiguiente () enviaActaSubdecanato ()
----	---------	--	---

Por otro lado, las pruebas de integración realizadas se basaron en comprobar si la funcionalidad de los métodos get y post de las APIs, estaban realizando correctamente la obtención y posteo de los datos en la base de datos.

Tabla 3.2 APIs y los respectivos métodos de integración probados [autoría propia]

Ambiente	API	Método probado
BackEnd	api/rol	get
	api/idHoja	
	api/estados	
	api/gtsi/status	
	api/subdecanato	
	api/aniosregistro	
	api/subdecanato/estudiante/<str:matricula>	
	api/documento/<str:tipo>/estudiante/<str:matricula>	get y post
	api/ultimotermينو	
	api/secretaria/notificar	
	api/secretaria/fechas/registrar	
	api/secretaria/fechas/periodos_disponibles/<int:id_periodo>	
	api/secretaria/notificar/lista_estudiantes/<int:id_periodo>	
	api/secretaria/<str:matricula>/proyecto/<str:id>/<str:usuario>	
api/secretaria/fechas/historico/<int:id_periodo>		

Por último, para las pruebas del FrontEnd se presentó un problema de autenticación para acceder a las llamadas de las APIs y como alternativa se propuso simular la respuesta de las APIs pasando datos a las clases que forman los módulos del sistema, y de este modo poder analizar si la información dada está llegando correctamente a cada componente.


```
const lista_estudiantesprueba= [{key: 2, 'matricula': '201705100',
'estudiante': 'LUIS ENRIQUE ANCHUNDIA SOLORZANO',
'correo': 'estud-fiec1@espol.edu.ec',
'año': '2022', 'termino': '1S',
'etapa': 'NO INICIADA'},

{'key': 3, 'matricula': '201706300',
'estudiante': 'SCARLET ANGELINA ESPINOZA MORENO',
'correo': 'estud-fiec2@espol.edu.ec', 'año': '2022',
'termino': '1S', 'etapa': 'SUBIR AL DSPACE'}]

Debug
test(' Verificar si funciona bien la tabla estudiante' , ()=> {

  render(<Router><NotificarEstudiantes lista_estudiantes= {lista_estudiantesprueba}/></Router>)
  try {
    waitFor(() => {
      expect(screen.getByTestId('tablaEst')).toMatch('201705100')
    });
  } catch (err) {
    // noop
  }
})
```

Figura 3.2 Ejemplo de prueba para el FrontEnd [autoría propia]

3.3 Ejecución de pruebas.

En el apéndice C, se muestran las sentencias sql que fueron ejecutadas para crear la base de datos de pruebas sobre la cual están basados los resultados de las pruebas desarrolladas. Por otro lado, en el BackEnd se utilizó la función assert() proporcionada por Pytest y en el FrontEnd la función test() para darle a conocer al sistema que es una prueba que debía ejecutarse y devolver un resultado de éxito o falla. Finalmente se documentó cada una de las pruebas de software usando la plantilla que se puede visualizar en la Tabla 2.1.

Tabla 3.3 Ejemplo de documentación de pruebas [autoría propia]

Sistema	SISTEMA AGATA		
Entregable	test_send_mail()		
Autor	Luis Anchundia		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_send_mail():
    from titulacion import mailutils
    payload = dict(
        subject = "Prueba agata pytest backend",
        message = "El documento perteneciente al proyecto test fue aprobado",
        from_email = "noresponder@espol.edu.ec",
        to_emails = ['estud-fiec1@espol.edu.ec' , 'estud-fiec2@espol.edu.ec'],
        response_message='Se envió correo de notificacion'
    )

    response = mailutils.send_mail(payload["subject"], payload["message"],
    payload["from_email"], payload["to_emails"])
    assert response.status_code == 200
```

Figura 3.3 Código correspondiente a la prueba test_send_mail()[autoría propia].

3.3.1 Resumen de las pruebas realizadas.

De acuerdo con el plan de pruebas ejecutado, se obtuvo que por el lado del BackEnd no se encontraron errores durante la ejecución las pruebas unitarias y de integración, es decir, las funciones específicas están realizando correctamente sus procesos y los métodos get y post que invocan las APIs están devolviendo la información sin problema alguno y como debe ser. Así mismo, por el lado del FrontEnd, todas pruebas resultaron exitosas, lo que permitió verificar que los componentes se renderizan y muestran la información donde corresponde.

Tabla 3.4 Resumen del total de pruebas ejecutadas [autoría propia]

Pruebas unitarias				
BackEnd	FrontEnd	Aprobadas	Fallidas	Total
34	40	74	0	74
Pruebas de integración				
BackEnd	FrontEnd	Aprobadas	Fallidas	Total
24	0	24	0	24

3.4 Análisis de la calidad del código y refactoring.

Se realizó el análisis del código en los repositorios que contienen los archivos del proyecto que correspondan a los módulos tutor, profesor, secretaria y subdecanato tanto del BackEnd como del FrontEnd y se utilizó la herramienta SonarQube, a través de la cual en primera instancia se obtuvieron los siguientes resultados sin el proceso de refactoring:

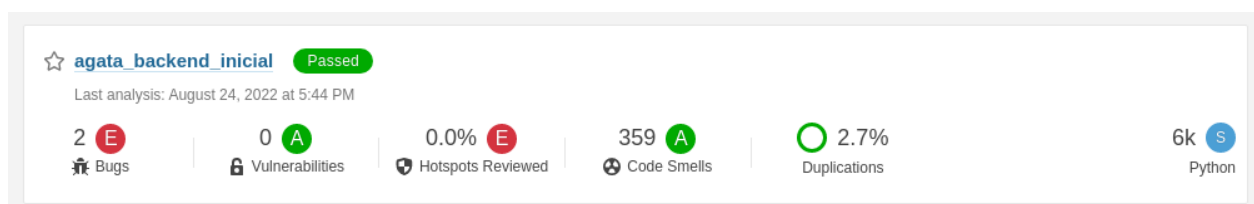


Figura 3.4 Resumen de resultados del análisis de código con la herramienta SonarQube en el BackEnd [autoría propia]

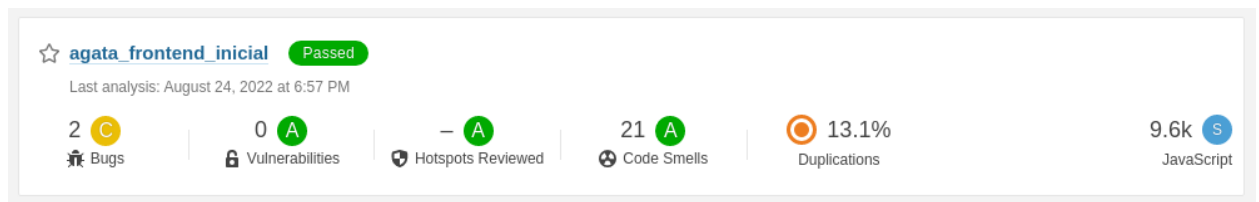


Figura 3.5 Resumen de resultados del análisis de código con la herramienta SonarQube en el FrontEnd [autoría propia]

3.4.1 Proceso de refactoring.

La herramienta SonarQube maneja un grado de severidad para las observaciones encontradas en el código, posee 5 grados de severidad, a continuación, se muestran ordenados de mayor a menor impacto en el código:

- Blocker
- Critical
- Major
- Minor
- Info

Para el proceso de *refactoring* se trabajó con los grados *Blocker* y *Critical* debido a que tienen mayor relevancia en la calidad del código, y las secciones a refactorizar fueron bugs y code smells.

BackEnd

Como se muestra en la Figura 3.4 Resumen de resultados del análisis de código con la herramienta SonarQube en el BackEnd [autoría propia] se encontraron 2 bugs en el código, entre ellos estaba reescribir una excepción

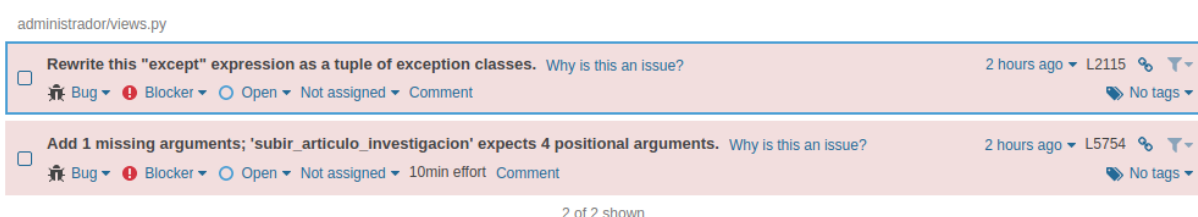
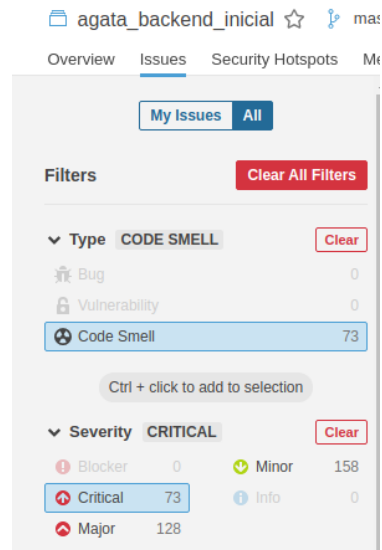


Figura 3.6 Bugs encontrados en el código que fueron corregidos [autoría propia] y añadir un argumento en una función que estaba faltando.

En la Figura 3.4 también se puede observar que se tuvo un total de 359 *code smells* de los cuales 73 pertenecían al grado de gravedad *critical* y por lo tanto sobre eso se trabajó, no se tomó en consideración las otras secciones debido a que las observaciones eran sobre renombrar variables o funciones.



**Figura 3.7 Resumen del grado de gravedad de code smells BackEnd
[autoría propia]**

Dentro de la sección *critical* se encontraron las siguientes observaciones que debían ser resueltas:

- Definir constantes para eliminar la repetición de secciones de texto.
- Especificar excepciones.
- Refactorizar funciones para reducir la complejidad cognitiva. Esto quiere decir que hay funciones demasiado grandes lo cual dificulta su entendimiento.

Véase el apéndice D donde se muestran las 73 observaciones que fueron refactorizadas.

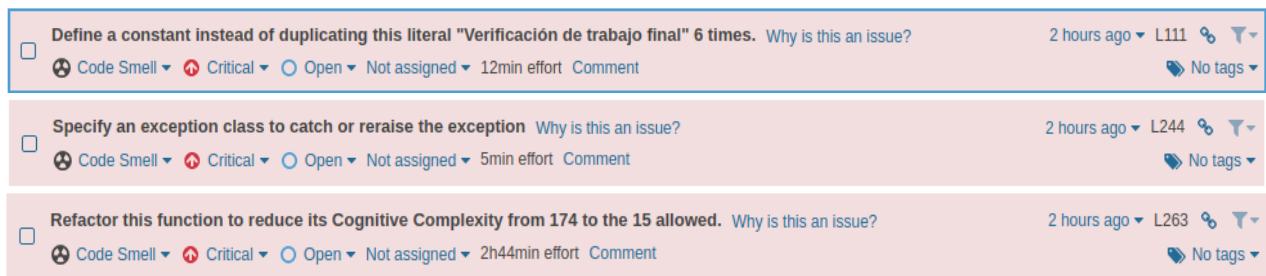


Figura 3.8 Ejemplos de code smells que fueron resueltos [autoría propia]

Adicionalmente, se eliminaron porciones de código que se encontraban comentadas y no aportaban nada a la funcionalidad del proyecto.

El proceso de *refactoring* se aplicó en el archivo `views.py` que se encuentra en la ruta `administrador/views.py` en los archivos del BackEnd.

A continuación, se presenta una tabla con el nombre de clases y funciones que fueron refactorizadas:

Tabla 3.5 Funciones y clases refactorizadas [autoría propia]

Funciones	Clases	
	Nombre	Método
subir_trabajo_graduacion(datos, proyecto, usuario, estado)	class AutenticacionCAS	get
subir_articulo_investigacion(datos, proyecto, usuario, estado)	class SecretariaDspace	post
subir_hoja_vida(datos, proyecto, usuario, estado)	class AvanceEtapa	get
estado_revision(id_rol, id_documento, id_proyecto)	class DocumentosProyecto	get
estado_proyecto(id_proyecto, tipo_proy)	class DocumentosProyecto	post
obtener_rol_proyecto(usuario, id_proyecto)	class InformacionProyecto	get
turno_realizar_revision(documento, usuario, rol, id_investigacion)	class InfoProyectoTutor	get
turno_aprobar(documento, usuario, rol)	class InfoProyectoSecretaria	get
	class ProyectosUltimoTermino	get
	class TutorUltimoTermino	get
	class RealizarRevision	post
	class AprobarTrabajo	post
	class DevolverTrabajo	post

	class Reporte	get
--	---------------	-----

FrontEnd

En la Figura 3.5 se muestra que en el lado del FrontEnd se encontraron 2 bugs, los cuales no fueron trabajados debido a que no presentaban un problema para el proyecto una vez fueron analizados y además estos no se encontraban dentro de los grados de severidad *blocker* o *critical*.

También en la Figura 3.5 se puede observar que se tenía 21 *code smells*, de los cuales 5 pertenecían a la sección *critical* y sobre estos se trabajó.

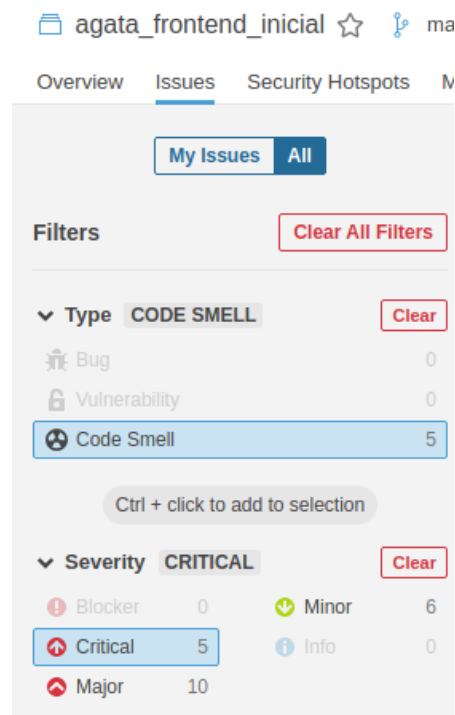


Figura 3.9 Resumen del grado de gravedad de code smells FrontEnd [autoría propia]

src/ModuloProfesor/AprobarProfesor.js

Use curly braces or indentation to denote the code conditionally executed by this "else". Why is this an issue? 5 hours ago L153
 Code Smell Critical Open Not assigned 10min effort [Comment](#) No tags

src/ModuloSecretaria/Fase1/AprobarGrupal.js

Use curly braces or indentation to denote the code conditionally executed by this "else". Why is this an issue? 5 hours ago L209
 Code Smell Critical Open Not assigned 10min effort [Comment](#) No tags

src/ModuloSecretaria/Fase1/PrincipalGrupal.js

Use curly braces or indentation to denote the code conditionally executed by this "else". Why is this an issue? 5 hours ago L135
 Code Smell Critical Open Not assigned 10min effort [Comment](#) No tags

src/ModuloTutor/AprobarTutor.js

Use curly braces or indentation to denote the code conditionally executed by this "else". Why is this an issue? 5 hours ago L80
 Code Smell Critical Open Not assigned 10min effort [Comment](#) No tags

src/ModuloTutor/RevisionTutor.js

Use curly braces or indentation to denote the code conditionally executed by this "else". Why is this an issue? 5 hours ago L202
 Code Smell Critical Open Not assigned 10min effort [Comment](#) No tags

5 of 5 shown

Figura 3.10 Code Smells que fueron resueltos [autoría propia]

Una vez terminado el proceso de *refactoring* se volvió a analizar el código con la herramienta SonarQube y se obtuvieron los siguientes resultados:

BackEnd

☆ **agata_backend_refactoring** Passed

Last analysis: August 24, 2022 at 5:47 PM

0 Bugs	0 Vulnerabilities	100% Hotspots Reviewed	327 Code Smells	1.9% Duplications	6k Python
-----------	----------------------	---------------------------	--------------------	----------------------	--------------

Figura 3.11 Resumen de resultados después del refactoring BackEnd [autoría propia]

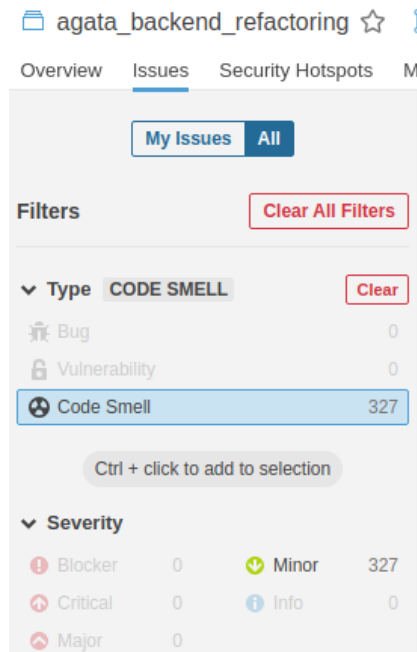


Figura 3.12 Resumen del grado de gravedad después del refactoring [autoría propia]

Como se puede observar en la Figura 3.11 se corrigieron todos los bugs y los *code smells* se redujeron de 359 a 327; el número de duplicaciones también se puede observar que pasó de 2.7% a 1.9%.

En la Figura 3.12 se observa que en la sección *code smells* los grados de severidad *blocker* y *critical* quedaron en 0.

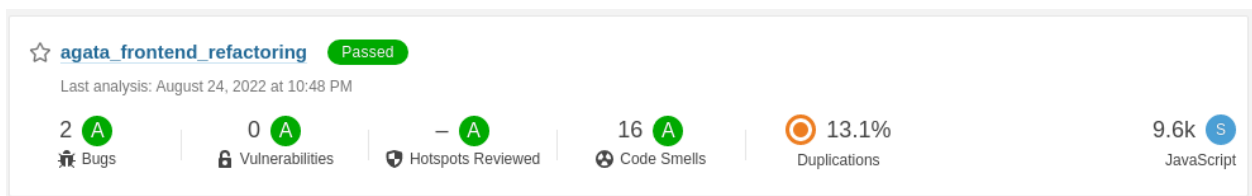


Figura 3.13 Resumen de resultados despues del refactoring FrontEnd [autoría propia]

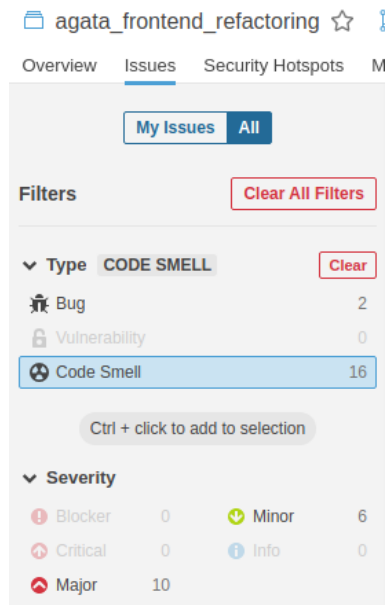


Figura 3.14 Resumen del grado de gravedad después del refactoring [autoría propia]

En la Figura 3.13 se puede observar que la cantidad de bugs se mantiene, pero luego del análisis se los cambió a la sección *minor*, también se puede observar que los *code smells* pasaron de 21 a 16.

En la Figura 3.14 se puede observar que las secciones *blocker* y *critical* quedaron en 0.

3.5 Análisis de costos.

Las herramientas utilizadas en el desarrollo del proyecto como Pytest, React, Jest y Jenkins no poseen un costo por uso, en ese caso el coste por uso de herramientas para el desarrollo el \$0.

A continuación, se muestra una tabla en donde se desglosa un total de 177 horas invertidas en el desarrollo del proyecto que se encuentran divididas por secciones que se presentaron a través de la fase de desarrollo:

Tabla 3.6 Tabla de costos [autoría propia]

Etapas	Horas Aproximadas	Valor por hora aproximado	Subtotal
Análisis de requerimientos	25	\$5.00	\$125.00
Automatización de pruebas	23	\$5.00	\$115.00
Pruebas unitarias	52	\$5.00	\$260.00
Pruebas de integración	52	\$5.00	\$260.00
Análisis del código y refactoring	25	\$5.00	\$125.00
Totales:	177	\$5.00	\$885.00

En la Tabla 3.6 se puede observar que el costo total aproximado del proyecto es de \$885.

Cabe recalcar que como el proyecto corresponde a un proyecto de materia integradora de Espol, el costo real es de \$0.

3.6 Cierre de proyecto

Una vez culminado del proyecto se hizo entrega al tutor y cliente, lo desarrollado durante la duración que tuvo el proyecto que fue lo siguiente: Una vez culminado del proyecto se hará entrega a los encargados del proyecto de lo siguiente a los encargados del proyectos:

- El programa Jenkins levantado en el servidor 200.9.176.97, configurado para la automatización de pruebas.
- Un total de 98 pruebas realizadas al sistema.
- Análisis del código del BackEnd y FrontEnd con la herramienta SonarQube.
- Código del sistema refactorizado.

En el apéndice F se puede ver a más detalle el acta de cierre, la cual se encuentra firmada por el cliente y tutor del proyecto.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El desarrollo de pruebas de software tanto unitarias como de integración en el BackEnd y el FrontEnd permitió probar la funcionalidad de los módulos tutor, profesor, secretaría y subdecanato y se comprobó su correcto funcionamiento dentro del sistema Agata.
- La herramienta Jenkins permitió que las pruebas de software desarrolladas se ejecuten de manera automática dentro de la plataforma, esta automatización permite descubrir errores que se puedan presentar en el desarrollo de nuevas funcionalidades justo después de un commit.
- El análisis del código utilizando la herramienta SonarQube aplicado al BackEnd y el FrontEnd permitió identificar porciones del código que necesitaban ser refactorizadas y una vez concluido el proceso de refactoring, la calidad del código mejoró en ambos repositorios.

4.2 Recomendaciones

- Se recomienda investigar métodos de prueba diferentes a los empleados en el proyecto con la finalidad de probar otros aspectos del sistema, como es el caso de las APIs externas que no son propiedad de este. Por ejemplo, las APIs pertenecientes al GTSI.
- También se recomienda realizar pruebas de software similares al resto de componentes que posee el sistema, debido a que en este proyecto solo se contemplaron 4 módulos.
- Por otra parte, si se utiliza un servidor remoto, se recomienda revisar las especificaciones de las herramientas que vayan a ser instaladas para evitar problemas con el firewall u otro tipo de inconvenientes, se menciona esto debido a que la herramienta Jenkins utiliza el puerto 8080, y en el servidor remoto que se estaba utilizando no tenía habilitado este puerto y por lo tanto significó una pérdida considerable de tiempo.

BIBLIOGRAFÍA

5. BIBLIOGRAFÍA

- [1] C. L. Goicochea, «atsistemas,» [En línea]. Available: <https://www.atsistemas.com/es/blog/la-importancia-del-testing-de-software-y-de-la-automatizacin-de-pruebas>. [Último acceso: 31 Mayo 2022].
- [2] S. Singh y A. Singh, Software Testing, Lucknow: Vandana Publications, 2019.
- [3] P. Jorgensen, Software testing a craftsman's approach, Boca Raton: CRC Press, 2014.
- [4] V. Soto, «Academia pragma,» 20 Mayo 2021. [En línea]. Available: <https://www.pragma.com.co/blog/conoce-que-son-las-pruebas-no-funcionales-de-software>. [Último acceso: 15 Junio 2022].
- [5] Django Software Foundation, «Django,» 2005. [En línea]. Available: <https://www.djangoproject.com/start/overview/>. [Último acceso: 8 Julio 2022].
- [6] Facebook open source, «React,» [En línea]. Available: <https://es.reactjs.org>. [Último acceso: 9 Julio 2022].
- [7] D. Hellman, «PyMOTW,» 11 Julio 2020. [En línea]. Available: <http://pymotw.com/2/unittest/>. [Último acceso: 6 Junio 2022].
- [8] P. S. Foundation, «Python,» [En línea]. Available: <https://docs.python.org/es/3.9/library/unittest.html>. [Último acceso: 7 Junio 2022].
- [9] G. Briceño y I. Oscco, «Club de tecnologia,» [En línea]. Available: <https://www.clubdetecnologia.net/cursos/pruebas-con-python/introduccion-a-pytest/#:~:text=Pytest%20es%20un%20framework%20maduro,de%20clases%2C%20como%20con%20unittest..> [Último acceso: 7 Junio 2022].
- [10] S. Campbell, «Guru99,» 3 Mayo 2022. [En línea]. Available: <https://www.guru99.com/pytest-tutorial.html>. [Último acceso: 7 Junio 2022].

- [11] J. Padilla, «Tests automatizados en React,» 01 agosto 2022. [En línea]. Available: <https://jorgeepc.dev/tests-automatizados-en-react>. [Último acceso: 8 junio 2022].
- [12] L. Bernalte, «Por qué usar Testing Library en lugar de Enzyme,» 10 Febrero 2021. [En línea]. Available: <https://lucasbernalte.com/blog/por-que-usar-testing-library-en-lugar-de-enzyme>. [Último acceso: 15 Junio 2022].
- [13] Z. Xing y E. Stroulia, «Refactoring Practice: How it is and How it Should be Supported - An Eclipse Case Study,» *2006 22nd IEEE International Conference on Software Maintenance*, pp. 458-468, 2006.
- [14] A. Kothari, «Geekflare,» 15 Febrero 2022. [En línea]. Available: <https://geekflare.com/es/code-review-tools/>. [Último acceso: 15 Junio 2022].
- [15] Saasworthy, «SaasWorthy,» 15 Junio 2022. [En línea]. Available: <https://www.saasworthy.com/product/deepscan-io>. [Último acceso: 18 Junio 2022].
- [16] Jenkins, «Jenkins,» [En línea]. Available: <https://www.jenkins.io/doc/>. [Último acceso: 10 07 2022].
- [17] L. S. P. FLORES, «MODELO PARA LA GENERACIÓN Y EJECUCIÓN DE PRUEBAS COMO MEDIO DE VERIFICACIÓN Y VALIDACIÓN DE PRODUCTOS DE SOFTWARE,» 02 Mayo 2016. [En línea]. Available: <https://rinacional.tecnm.mx/bitstream/TecNM/746/1/33224-2016.pdf>.

APÉNDICES

Apéndice A

Pruebas De Integración Y Unitarias Del FrontEnd Y BackEnd

Tabla 6.1 Datos de la prueba integración #1 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_Roles		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	25/06/2022
Aprobado por	David Jurado	Fecha de aprobación	25/06/2022

```
@pytest.mark.django_db
def test_Roles():
    from administrador import views
    request= HttpRequest()
    request.method='GET'
    request.user=Username("secre-fiecl")
    response=views.Roles().get(request)
    print(json.loads(response.content))
    assert response.status_code == 200
```

Figura 6.1 Prueba integración #1 para el BackEnd [autoría propia]

Tabla 6.2 Datos de la prueba integración #2 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_EstudianteMatricula		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	26/06/2022
Aprobado por	David Jurado	Fecha de aprobación	26/06/2022

```
@pytest.mark.django_db
def test_EstudianteMatricula():
    request= HttpRequest()
    request.method='GET'
    request.user=Username("secre-fiecl")
    from administrador import views
    response = views.EstudianteMatricula().get(request, '201706300')
    print(json.loads(response.content))
    assert response.status_code== 200
```

Figura 6.2 Prueba integración #2 para el BackEnd [autoría propia]

Tabla 6.3 Datos de la prueba integración #3 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_downloadDocuments		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/06/2022
Aprobado por	David Jurado	Fecha Aprobación	27/06/2022

```
@pytest.mark.django_db
def test_downloadDocuments():
    from administrador import views
    request= HttpRequest()
    request.method='GET'
    response= views.DocumentoApi().get(request,'hojavida','201706300')
    print(json.loads(response.content))
    assert response.status_code ==200
```

Figura 6.3 Prueba integración #3 para el BackEnd [autoría propia]

Tabla 6.4 Datos de la prueba unitaria #4 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_ultimoTermino		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	28/06/2022
Aprobado por	David Jurado	Fecha Aprobación	28/06/2022

```
@pytest.mark.django_db
def test_ultimoTermino():
    periodoacademico = Periodoacademico.objects.order_by('-fechacreacion')
    from administrador import views
    resultado = views.ultimo_termino()
    print(periodoacademico[1].nombre)
    print(periodoacademico[2].nombre)
    assert resultado.nombre == '15 - 2022'
```

Figura 6.4 Prueba unitaria #4 para el BackEnd [autoría propia]

Tabla 6.5 Datos de la prueba integración #5 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_idHoja		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	29/06/2022
Aprobado por	David Jurado	Fecha Aprobación	29/06/2022

```
@pytest.mark.django_db
def test_IdHoja():
    from administrador import views
    request= HttpRequest()
    response= views.IdHoja().get(request)
    print(json.loads(response.content))
    """assert (json.loads(response.content)['id']== 3 ) '''
    assert (json.loads(response.content)['value']== 'HOJA DE VIDA' )"""
    assert response.status_code ==200
```

Figura 6.5 Prueba integración #5 para el BackEnd [autoría propia]

Tabla 6.6 Datos de la prueba integración #6 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_cargarEstados		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	30/06/2022
Aprobado por	David Jurado	Fecha Aprobación	30/06/2022

```
@pytest.mark.django_db
def test_cargarEstados():
    from administrador import views
    request= HttpRequest()
    request.method='GET'
    response= views.EstadosRevision().get(request)
    print(json.loads(response.content))
    """listprueba=[{'text': 'APROBADO', 'value': 'APROBADO'},
    {'text': 'DEVUELTO', 'value': 'DEVUELTO'},
    {'text': 'NO INICIADO', 'value': 'NO INICIADO'},
    {'text': 'PENDIENTE', 'value': 'PENDIENTE'}]
    assert json.loads(response.content)['estados']== listprueba"""
    assert response.status_code ==200
```

Figura 6.6 Prueba integración #6 para el BackEnd [autoría propia]

Tabla 6.7 Datos de la prueba unitaria #2 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_obtenerRolTermino		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	31/06/2022
Aprobado por	David Jurado	Fecha Aprobación	31/06/2022

```
@pytest.mark.django_db
def test_obtenerRolTermino():
    from administrador import views
    secretaria = rolesUsuarios[4]
    ultimoTermino = views.ultimo_termino()
    rolTermino=views.obtener_rol_termino(secretaria,ultimoTermino)
    rolsecretaria=0
    for rol in rolTermino:
        print(rol)
        rolsecretaria=rol
    assert rolsecretaria == 4
```

Figura 6.7 Prueba unitaria #2 para el BackEnd [autoría propia]

Tabla 6.8 Datos de la prueba unitaria #3 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_obtenerNombreEtapa		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	01/07/2022
Aprobado por	David Jurado	Fecha Aprobación	01/07/2022

```
@pytest.mark.django_db
def test_ObtenerNombreEtapa():
    from administrador import views
    idnombreetapa = views.obtenerNombreEtapa("Subir al DSPACE")
    assert idnombreetapa == 7
```

Figura 6.8 Prueba unitaria #3 para el BackEnd [autoría propia]

Tabla 6.9 Datos de la prueba unitaria #4 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_enviaActaSubdecanato		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	02/07/2022
Aprobado por	David Jurado	Fecha Aprobación	02/07/2022

```
@pytest.mark.django_db
def test_envioActaSubdecanato():
    from administrador import views
    envio=views.envioActaSubdecanato('201706300')
    assert envio.status_code == 200
```

Figura 6.9 Prueba unitaria #4 para el BackEnd [autoría propia]

Tabla 6.10 Datos de la prueba unitaria #5 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_notiSubdecanato		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	03/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/07/2022

```
@pytest.mark.django_db
def test_notiSubdecanato():
    from administrador import views
    notificar_subdecanato=views.notiSubdecanato ('201706300')
    assert notificar_subdecanato == True
```

Figura 6.10 Prueba unitaria #5 para el BackEnd [autoría propia]

Tabla 6.11 Datos de la prueba integración #6 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_nombresEtapas		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	04/07/2022
Aprobado por	David Jurado	Fecha Aprobación	04/07/2022

```
@pytest.mark.django_db
def test_nombresEtapas():

    request= HttpRequest()
    request.method='GET'
    from administrador import views
    resultado = views.NombresEtapas().get(request)
    assert resultado.status_code == 200
```

Figura 6.11 Prueba integración #6 para el BackEnd [autoría propia]

Tabla 6.12 Datos de la prueba integración #7 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_AniosRegistro		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	05/07/2022
Aprobado por	David Jurado	Fecha Aprobación	05/07/2022

```
@pytest.mark.django_db
def test_AniosRegistro():

    request= HttpRequest()
    request.method='GET'
    from administrador import views
    resultado = views.AniosRegistro().get(request)
    print(json.loads(resultado.content))
    assert resultado.status_code == 200
```

Figura 6.12 Prueba integración #7 para el BackEnd [autoría propia]

Tabla 6.13 Datos de la prueba integración #8 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_Termino		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_Termino():
    from administrador import views
    request= HttpRequest()
    request.method='GET'
    response= views.Termino().get(request)
    print(json.loads(response.content)['periodos'])
    assert response.status_code== 200
```

Figura 6.13 Prueba integración #8 para el BackEnd [autoría propia]

Tabla 6.14 Datos de la prueba integración #9 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_obtenerHistoricos		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_obtenerHistoricos(): #EstenesOBETNERHistorioco de la secretaria

    request= HttpRequest()
    request.method='GET'
    request.user=Username("secre-fiecl")
    from administrador import views
    resultado = views.FechasRecepcion().get(request,3)
    print(json.loads(resultado.content))
    assert resultado.status_code == 200
```

Figura 6.14 Prueba integración #9 para el BackEnd [autoría propia]

Tabla 6.15 Datos de la prueba integración #10 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_registrarPeriodoPost		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_registrarPeriodoPost():
    request= HttpRequest()
    request.method='POST'
    request.user=Username("secre-fiecl")
    request.__setattr__ ('data',dict())
    request.data['periodo']=3
    request.data['inicioProceso']="2022-08-16T14:12:06.000-0500"
    request.data['inicioRecepcion']="2022-08-17T14:12:06.000-0500"
    request.data["finRecepcion"]="2022-09-17T14:12:06.000-0500"
    from administrador import views
    resultado = views.FechasRecepcion().post(request)
    assert resultado.status_code == 200
```

Figura 6.15 Prueba integración #10 para el BackEnd [autoría propia]

Tabla 6.16 Datos de la prueba integración #11 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_postInfoProyecto1		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_postInfoProyecto1():
    request= HttpRequest()
    from administrador import views
    response = views.SecretariaDspace().post(request,'201706300', 1 , 'false', 'true')
    assert response.status_code== 200
```

Figura 6.16 Prueba integración #11 para el BackEnd [autoría propia]

Tabla 6.17 Datos de la prueba integración #12 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_postInfoProyecto4		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_postInfoProyecto4():
    request= HttpRequest()
    from administrador import views
    response = views.SecretariaDspace().post(request,'201706300', 1 , 'false', 'false')
    assert response.status_code== 200
```

Figura 6.17 Prueba integración #12 para el BackEnd [autoría propia]

Tabla 6.18 Datos de la prueba unitaria #6 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_subirActaSAAC		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_subirActaSAAC():
    from administrador import views
    actaSAAC=views.subirActaSAAC('201706300', "A", "ICC", "NA", "acta")
    assert actaSAAC == True
```

Figura 6.18 Prueba unitaria #6 para el BackEnd [autoría propia]

Tabla 6.19 Datos de la prueba integración #13

Sistema	Sistema AGATA		
Entregable	test_periodosDisponibles		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_periodosDisponibles():
    from administrador import views
    request= HttpRequest()
    request.method='GET'
    request.user=Username("secre-fiecl")
    response= views.PeriodosDisponibles().get(request,3)
    print(json.loads(response.content))
    assert response.status_code== 200
```

Figura 6.19 Prueba integración #13 para el BackEnd

Tabla 6.20 Datos de la prueba integración #14

Sistema	Sistema AGATA		
Entregable	test_DocumentoGET		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_DocumentosGET():
    from administrador import views
    request= HttpRequest()
    request.method='GET'
    request.user=Username("secre-fiecl")
    response=views.Documentos().get(request,3)
    print(json.loads(response.content))
    assert response.status_code == 200
```

Figura 6.20 Prueba integración #14 para el BackEnd [autoría propia]

Tabla 6.21 Datos de la prueba integración #15

Sistema	Sistema AGATA		
Entregable	test_AutenticacionCAS		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_AutenticacionCAS():
    from administrador import views
    request= HttpRequest()
    request.method='GET'
    request.user=Username("secre-fiecl")
    response=views.AutenticacionCAS().get(request)
    print(json.loads(response.content))
    assert response.status_code == 200
```

Figura 6.21 Prueba integración #15 para el BackEnd [autoría propia]

Tabla 6.22 Datos de la prueba integración #16 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_SubdecanatoGet		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_SubdecanatoGET():
    from administrador import views
    request= HttpRequest()
    request.method='GET'
    request.user=Username("subde-fiecl")
    response=views.Subdecanato().get(request)
    print(json.loads(response.content))
    assert response.status_code == 200
```

Figura 6.22 Prueba de integración #16 para el BackEnd [autoría propia]

Tabla 6.23 Datos de la prueba integración #17 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_SubdecanatoEstudianteGET		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_SubdecanatoEstudianteGET():
    from administrador import views
    request= HttpRequest()
    request.method='GET'
    request.user=Username("subde-fiecl")
    response=views.SubdecanatoEstudiante().get(request,'201706300')
    print(json.loads(response.content))
    assert response.status_code == 200
```

Figura 6.23 Prueba de integración #17 para el BackEnd [autoría propia]

Tabla 6.24 Datos de la prueba de integración #18 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_InfoProyectoTutor		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

Figura 6.24 Prueba de integración #18 para el BackEnd [autoría propia]

Tabla 6.25
prueba
[autoría

```
def test_infoProyectoTutor():
    from administrador import views
    request= HttpRequest()
    request.method='GET'
    request.user=Username("tutor-fiecl")
    response= views.InfoProyectoTutor().get(request,1)
    print(response)
    assert response.status_code== 200
```

Datos de la
unitaria #7
propia]

Sistema	Sistema AGATA		
Entregable	test_obtenerMatriculaSiguiente		
Autor	Scarlet Angelina Espinoza Moreno		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_obtenerMatriculaSiguiente():
    from administrador import views
    response=views.obtenerMatriculaSiguiente('201706300')
    print(response)
    assert response == '201705100'
```

Figura 6.25 Prueba unitaria #7 para el BackEnd [autoría propia]

Tabla 6.26 Datos de la prueba unitaria #8 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_estadoRevision		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

@pytest.mark.django_db
def test_estadoRevision():
    from administrador import views
    response = views.estado_revision(1,1,1)
    assert "estado" in response and "fecha" in response

```

Figura 6.26 Prueba unitaria #8 para el BackEnd [autoría propia]

Tabla 6.27 Datos de la prueba unitaria #9 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_estado_Revision_pendiente		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

@pytest.mark.django_db
def test_estado_Revision_pendiente():
    from administrador import views
    response = views.estado_revision(1,1,1)
    assert response['estado'] == 'P'

```

Figuras 6.27 Prueba unitaria #9 para el BackEnd [autoría propia]

Tabla 6.28 Datos de la prueba unitaria #10 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_ultima_revision_usuario_aprobado		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022

Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022
---------------------	--------------	-------------------------	-------------

```
@pytest.mark.django_db
def test_ultima_revision_usuario_aprobado():
    from administrador import views
    response = views.ultima_revision_usuario(1,'tutor-fiecl',2)
    assert response['estado'] == 'A'
```

Figura 6.28 Prueba unitaria #10 para el BackEnd [autoría propia]

Tabla 6.29 Datos de la prueba unitaria #11 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_obtener_rol_proyecto_estudiante		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_obtener_rol_proyecto_estudiante():
    #rol estudiante
    from administrador import views
    estudiante = rolesUsuarios[1][1]
    response = views.obtener_rol_proyecto(estudiante,1)
    assert response == 1
```

Figura 6.29 Prueba unitaria #11 para el BackEnd [autoría propia]

Tabla 6.30 Datos de la prueba unitaria #12 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_obtener_rol_proyecto_tutor		
Autor	Luis Enrique Anchundia Solorzano		

Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_obtener_rol_proyecto_tutor():
    #rol tutor
    from administrador import views
    tutor = rolesUsuarios[2]
    response = views.obtener_rol_proyecto(tutor,1)
    assert response == 2
```

Figura 6.30 Prueba unitaria #12 para el BackEnd [autoría propia]

Tabla 6.31 Datos de la prueba unitaria #13 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_obtener_rol_proyecto_profesor		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_obtener_rol_proyecto_profesor():
    #rol profesor
    from administrador import views
    profesor = rolesUsuarios[3]
    response = views.obtener_rol_proyecto(profesor,1)
    assert response == 3
```

Figura 6.31 Prueba unitaria #13 para el BackEnd [autoría propia]

Tabla 6.32 Datos de la prueba unitaria #14 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_obtener_rol_proyecto_secretaria		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_obtener_rol_proyecto_secretaria():
    #rol secretaria
    from administrador import views
    secretaria = rolesUsuarios[4]
    response = views.obtener_rol_proyecto(secretaria,1)
    assert response == 4
```

Figura 6.32 Prueba unitaria #14 para el BackEnd [autoría propia]

Tabla 6.33 Datos de la prueba unitaria #15 [autoría propia]

Sistema	Sistema AGATA		
Entregable	test_obtener_rol_proyecto_subdecanato		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_obtener_rol_proyecto_subdecanato():
    #rol subdecanato
    from administrador import views
    subdecanato = rolesUsuarios[5]
    response = views.obtener_rol_proyecto(subdecanato,1)
    assert response == 5
```

Figura 6.33 Prueba unitaria #15 para el BackEnd [autoría propia]

Tabla 6.34 Datos de la prueba unitaria #16 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_get_info_proyecto_individual		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_get_info_proyecto_individual():
    from titulacion.utils.reportes.proyectos import funciones
    response = funciones.get_info_proyecto_individual(1)
    assert response.proyecto.nombreproyecto == "Proyecto integrador de prueba"
```

Figura 6.34 Prueba unitaria #16 para el BackEnd [autoría propia]

Tabla 6.35 Datos de la prueba unitaria #17 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_turno_aprobar		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_turno_aprobar():
    documento = Documento.objects.get(id=1)
    from administrador import views
    response = views.turno_aprobar(documento,"tutor-fiecl",1)
    assert "mensaje" in response and "puede_aprobar" in response
```

Figura 6.35 Prueba unitaria #17 para el BackEnd [autoría propia]

Tabla 6.36 Datos de la prueba unitaria #18 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_transformarKB		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_transformarKB():
    from administrador import views
    unidad = "KB"
    valor_unidad = 1024
    tamaño = 540
    valor = tamaño * valor_unidad
    resultado = views.transformar(tamaño, unidad)
    assert resultado == valor
```

Figura 6.36 Prueba unitaria #18 para el BackEnd [autoría propia]

Tabla 6.37 Datos de la prueba unitaria #19

Sistema	Sistema AGATA		
Entregable	Test_transformarGB		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_transformarGB():
    from administrador import views
    unidad = "GB"
    valor_unidad = 1073741824
    tamaño = 540
    valor = tamaño * valor_unidad
    resultado = views.transformar(tamaño, unidad)
    assert resultado == valor
```

Figura 6.37 Prueba unitaria #19 para el BackEnd [autoría propia]

Tabla 6.38 Datos de la prueba unitaria #20 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_transformarTB		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_transformarTB():
    from administrador import views
    unidad = "TB"
    valor_unidad = 1099511627776
    tamaño = 540
    valor = tamaño * valor_unidad
    resultado = views.transformar(tamaño, unidad)
    assert resultado == valor
```

Figura 6.38 Prueba unitaria #20 para el BackEnd [autoría propia]

Tabla 6.39 Datos de la prueba unitaria #21 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_TransformarMB		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_transformarMB():
    from administrador import views
    unidad = "MB"
    valor_unidad = 1048576
    tamaño = 540
    valor = tamaño * valor_unidad
    resultado = views.transformar(tamaño, unidad)
    assert resultado == valor
```

Figura 6.39 Prueba unitaria #21 para el BackEnd [autoría propia]

Tabla 6.40 Datos de la prueba unitaria #22 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_convertir_estado		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

pytest.mark.parametrize(
    ['estado', 'esperado'], [
        ('Aprobado', 'A'),
        ('Pendiente', 'P'),
        ('Devuelto', 'D'),
        ('Corregido', 'C'),
    ]
)

@pytest.mark.django_db
def test_convertir_estado(estado, esperado):
    from administrador import views
    resultado = views.convertir_estado(estado)
    assert resultado == esperado
    
```

Figura 6.40 Prueba unitaria #22 para el BackEnd [autoría propia]

Tabla 6.41 Datos de la prueba unitaria #23 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_send_mail		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

6.42
de la
#24

```
#SE NECESITA LA VPN ESPOL
@pytest.mark.django_db
def test_send_mail():
    from titulacion import mailutils
    payload = dict(
        subject = "Prueba agata pytest backend",
        message = "El documento perteneciente al proyecto test fue aprobado",
        from_email = "noresponder@espol.edu.ec",
        to_emails = ['estud-fiec1@espol.edu.ec', 'estud-fiec2@espol.edu.ec'],
        response_message='Se envió correo de notificacion'
    )

    response = mailutils.send_mail(payload["subject"], payload["message"],
    payload["from_email"], payload["to_emails"])
    assert response.status_code == 200
```

Tabla
Datos
prueba
unitaria
[autoría

propia]

Sistema	Sistema AGATA		
Entregable	Test_send_mail		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_obtener_rol_termino_tutor():
    from administrador import views
    response = views.obtener_rol_termino('tutor-fiec1', 3)
    assert response[0] == 2
```

Figura 6.42 Prueba unitaria #24 para el BackEnd [autoría propia]

Tabla 6.43 Datos de la prueba unitaria #25 [autoría propia]

Sistema	Sistema AGATA
----------------	---------------

Entregable	Test_obtener_rol_termino_profesor		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_obtener_rol_termino_profesor():
    from administrador import views
    response = views.obtener_rol_termino('prof-fiec1', 3)
```

Figura 6.43 Prueba unitaria #25 para el BackEnd [autoría propia]

Tabla 6.44 Datos de la prueba unitaria #26 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_obtener_rol_termino_secretaria		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_obtener_rol_termino_secretaria():
    from administrador import views
    response = views.obtener_rol_termino('secre-fiec1', 3)
    assert response[0] == 4
```

Figura 6.44 Prueba unitaria #26 para el BackEnd [autoría propia]

Tabla 6.45 Datos de la prueba unitaria #27 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_obtener_rol_subdecanato		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_obtener_rol_termino_subdecanato():
    from administrador import views
    response = views.obtener_rol_termino('subde-fiec1', 3)
    assert response[0] == 5
```

Figura 6.45 Prueba unitaria #27 para el BackEnd [autoría propia]

Tabla 6.46 Datos de la prueba integración #19 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_get_no_authorized		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_get_no_authorized():
    from administrador import views
    request = HttpRequest()
    request.user = Username("tutor-fiec1")
    request.method = "GET"
    response = views.FechasRecepcion().get(request, 3)
    assert json.loads(response.content)['msg'] == 'Usuario no autorizado'
```

Figura 6.46 Prueba integración #19 para el BackEnd [autoría propia]

Tabla 6.47 Datos de la prueba integración #20 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_get_fechaRecepcion		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_get_FechasRecepcion():
    from administrador import views
    jsondeseado = [{"id": "8", "termino": "1S",
    "anio": "2022", "fechaRegistro": "2022-08-04", "fechaInicio": "2023-10-27",
    "fechaFin": "2023-11-03", "etapa": "FIN DEL PROCESO", "realizadoPor": "MARIA ROCIO PEREZ SUAREZ"},
    "1": {"id": "6", "termino": "1S", "anio": "2022", "fechaRegistro": "2022-08-04", "fechaInicio": "2023-10-13",
    "fechaFin": "2023-10-20", "etapa": "SUBIR AL DSPACE", "realizadoPor": "MARIA ROCIO PEREZ SUAREZ"},
    "2": {"id": "7", "termino": "1S", "anio": "2022", "fechaRegistro": "2022-08-04", "fechaInicio": "2023-10-20",
    "fechaFin": "2023-10-27", "etapa": "SUBIR ACTA DE GRADO", "realizadoPor": "MARIA ROCIO PEREZ SUAREZ"}, "3":
    {"id": "1", "termino": "1S", "anio": "2022", "fechaRegistro": "2022-08-04", "fechaInicio": "2022-08-05", "fechaFin": "2022-08-20",
    "etapa": "RECEPCIÓN DE TRABAJO FINAL", "realizadoPor": "MARIA ROCIO PEREZ SUAREZ"},
    "4": {"id": "2", "termino": "1S", "anio": "2022", "fechaRegistro": "2022-08-04", "fechaInicio": "2022-08-20",
    "fechaFin": "2022-08-26", "etapa": "VERIFICACIÓN DE TRABAJO FINAL", "realizadoPor": "MARIA ROCIO PEREZ SUAREZ"},
    "5": {"id": "3", "termino": "1S", "anio": "2022", "fechaRegistro": "2022-08-04", "fechaInicio": "2022-08-26",
    "fechaFin": "2023-09-15", "etapa": "RECEPCIÓN DE DOCUMENTOS PERSONALES", "realizadoPor": "MARIA ROCIO PEREZ SUAREZ"},
    "6": {"id": "4", "termino": "1S", "anio": "2022", "fechaRegistro": "2022-08-04", "fechaInicio": "2023-09-15",
    "fechaFin": "2023-10-06", "etapa": "VERIFICACIÓN DE DOCUMENTOS PERSONALES", "realizadoPor": "MARIA ROCIO PEREZ SUAREZ"},
    "7": {"id": "5", "termino": "1S", "anio": "2022", "fechaRegistro": "2022-08-04", "fechaInicio": "2023-10-06",
    "fechaFin": "2023-10-13", "etapa": "EXTRAER ACTA DE GRADO", "realizadoPor": "MARIA ROCIO PEREZ SUAREZ"},
    "8": {"id": "11", "termino": "1S", "anio": "2022", "fechaRegistro": "2020-09-30", "fechaInicio": "2022-08-03",
    "fechaFin": "2022-08-08", "etapa": "SUBIR AL DSPACE", "realizadoPor": "MARIA ROCIO PEREZ SUAREZ"},
    "9": {"id": "10", "termino": "1S", "anio": "2022", "fechaRegistro": "2020-09-30", "fechaInicio": "2022-08-03",
    "fechaFin": "2022-08-08", "etapa": "SUBIR AL DSPACE", "realizadoPor": "MARIA ROCIO PEREZ SUAREZ"}]
    request = HttpRequest()
    request.user=Username("secre-fiecl")
    response = views.FechasRecepcion().get(request,3)
    jsondata = json.loads(response.content)
    print(jsondata)
    assert jsondata == jsondeseado
```

Figura 6.47 Prueba integración #20 para el BackEnd [autoría propia]

Tabla 6.48 Datos de la prueba integración #21 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_post_FechasRecepcion		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_post_FechasRecepcion():
    from administrador import views
    request = HttpRequest()
    diccionario = dict(
        periodo = 2,
        inicioProceso = '2022-08-01T00:00:00.000-0500',
        inicioRecepcion = '2022-09-01T00:00:00.000-0500',
        finRecepcion = '2022-10-01T00:00:00.000-0500'
    )
    request.__setattr__('data',diccionario)
    request.user = Username('secre-fiecl')
    response = views.FechasRecepcion().post(request)
    jsonresponse = json.loads(response.content)
    assert jsonresponse['mensaje'] == 'Periodo académico registrado con éxito.'
```

Figura 6.48 Prueba integración #21 para el BackEnd [autoría propia]

Tabla 6.49 Datos de la prueba integración #22 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_post_AprobarTrabajo		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

@pytest.mark.django_db
def test_post_AprobarTrabajo():
    from administrador import views
    request = HttpRequest()
    request.POST['id_documento']=1
    request._setattr_('data',dict())
    request.data['usuario']='tutor-fiec1'
    request.data["id_proyecto"]=1
    instancia = views.AprobarTrabajo()
    instancia._setattr_('request',HttpRequest())
    response = views.AprobarTrabajo.post(instancia,request)
    jsonresponse = json.loads(response.content)
    assert jsonresponse['mensaje'] == 'Se envió correo de notificacion'

```

Figura 6.49 Prueba integración #22 para el BackEnd [autoría propia]

Tabla 6.50 Datos de la prueba integración #23 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_info_proyecto_tutor		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

@pytest.mark.django_db
def test_info_proyecto_tutor():
    from administrador import views
    request = HttpRequest()
    request.user = Username('tutor-fiec1')
    response = views.InfoProyectoTutor().get(request,1)
    jsonresponse = json.loads(response.content)
    esperado = {
        'id': 1, 'titulo': 'PROYECTO INTEGRADOR DE PRUEBA',
        'tipo': 'INTEGRADOR', 'codigo': 'PG-001', 'fechaInicio': '2022-07-27',
        'termino': '1S', 'año': 2022, 'facultad': 'FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN',
        'carrera': 'INGENIERÍA EN COMPUTACIÓN', 'profesor': 'EDUARDO SEGUNDO CRUZ RAMIREZ',
        'estudiantes': [{ 'usuario': 'estud-fiec2', 'matricula': '201706300', 'nombres': 'SCARLET ANGELINA ESPINOZA MORENO' }],
        'tutores': [ 'RAFAEL IGNACIO BONILLA ARMIJOS' ], 'etapa': 'SUBIR AL DSPACE' }
    #print(jsonresponse['proyecto'])
    assert jsonresponse['proyecto'] == esperado

```

Figura 6.50 Prueba integración #23 para el BackEnd [autoría propia]

Tabla 6.51 Datos de la prueba integración #24 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_devolverTrabajo_sinComentario		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_devolverTrabajo_sinComentario():
    from administrador import views
    request = HttpRequest()
    request.method="POST"
    request._setattr__ ('data',dict())
    request.data['comentario'] = ""
    response = views.DevolverTrabajo().post(request)
    jsonresponse = json.loads(response.content)
    assert jsonresponse['error']=='Debe al menos escribir un comentario'
```

Figura 6.51 Prueba integración #24 para el BackEnd [autoría propia]

Tabla 6.52 Datos de la prueba integración #25 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_obtenerUrkund		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_obtenerUrkund():
    from administrador import views
    request = HttpRequest()
    id_proyecto = 1
    response = views.ObtenerUrkund().get(request,id_proyecto)
    jsonresponse = json.loads(response.content)

    assert jsonresponse['urkund']['nombre'] == 'PG-001_antiplagio.pdf'
```

Figura 6.52 Prueba integración #25 para el BackEnd [autoría propia]

Tabla 6.53 Datos de la prueba integración #26 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_get_ultimotermينو		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_get_ultimoTermino():
    from administrador import views
    request = HttpRequest()
    response = views.UltimoTerminoAcademico().get(request)
    jsonresponse = json.loads(response.content)
    assert jsonresponse['anio']==2022
    assert jsonresponse['termino']=='1S'
```

Figura 6.53 Prueba integración #26 para el BackEnd [autoría propia]

Tabla 6.54 Datos de la prueba integración #27 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_get_EstadoPorfesor		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_get_EstadoProfesor():
    from administrador import views
    request = HttpRequest()
    response = views.EstadoProfesor().get(request,'prof-fiec1',1)
    assert response.status_code == 200
```

Figura 6.54 Prueba integración #27 para el BackEnd [autoría propia]

Tabla 6.55 Datos de la prueba integración #28 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_notificarEstudianteGET		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_notificarEstudianteGET():
    from administrador import views
    request= HttpRequest()
    request.method='GET'
    request.user=Username("secre-fiec1")
    response= views.NotificarEstudiantes().get(request,3)
    print(json.loads(response.content)['secretaria'])
    assert response.status_code== 400
```

Figura 6.55 Prueba integración #28 para el BackEnd [autoría propia]

Tabla 6.56 Datos de la prueba integración #29 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_notificarEstudiantePOST		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_notificarEstudiantePOST():
    from administrador import views
    request= HttpRequest()
    request.method='GET'
    request.__setattr__ ('data',dict())
    request.data['asunto']="NOTIFICAR"
    request.data['mensaje']="Hola"
    request.data['remitente']="noresponder@espol.edu.ec"
    request.data["destinatarios"]=["stud-fiec2@espol.edu.ec"]
    response= views.NotificarEstudiantes().post(request)
    print(json.loads(response.content))
    assert response.status_code== 200
```

Figura 6.56 Prueba integración #29 para el BackEnd [autoría propia]

Tabla 6.57 Datos de la prueba integración #30 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test_registrarPeriodoGet		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_registrarPeriodoGet():
    request= HttpRequest()
    request.method='POST'
    request.user=Username("secre-fiecl")
    from administrador import views
    resultado = views.FechasRecepcion().get(request,3)
    assert resultado.status_code == 200
```

Figura 6.57 Prueba integración #30 para el BackEnd [autoría propia]

Tabla 6.58 Datos de la prueba unitaria #28 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar combobox de periodos		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
afterEach(() => {
    cleanup();
});

test(' Verificar si el combo de Periodos disponibles funcion' , async())=> {
    let actualErrorMsg;
    render(<Router><FechaRecepcion periodosDisponibles= {PeriodosDisponibles}></Router>)
    try {
        waitFor(() => {
            const dropdown = screen.queryByTestId('Option1')
            expect(dropdown).toHaveTextContent(/25 - 2021/)
        });
    } catch (err) {
        actualErrorMsg = err.message;
    }
})
```

Figura 6.58 Prueba unitaria #28 para el FrontEnd [autoría propia]

Tabla 6.59 Datos de la prueba unitaria #29 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar Datapicker de fechas		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si el data picker de fecha inicio y fin funciona para seleccionar etapa 1 ', async () => {
  let actualErrorMsg;
  render(<Router><FechaRecepcion periodosDisponibles= {PeriodosDisponibles}/></Router>)
  try {
    waitFor(() => {
      userEvent.type(screen.getByPlaceholderText('Seleccione fecha de inicio de etapa 1'), '1970-01-01')
      expect(screen.getByPlaceholderText('Seleccione fecha de inicio de etapa 1') ).toHaveValue('1970-01-01')
    });
  } catch (err) {
    actualErrorMsg = err.message;
  }
})
```

Figura 6.59 Prueba unitaria #29 para el FrontEnd [autoría propia]

Tabla 6.60 Datos de la prueba unitaria #30 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test si la información se almacene en la tabla		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si la informacion de la planificacion se está mostrando en la tabla', async () => {
  let actualErrorMsg;
  render(<Router><FechaRecepcion periodosDisponibles= {PeriodosDisponibles}/></Router>)
  try {
    waitFor(() => {
      const tabla = screen.queryByTestId('tablaHistorico');
      expect(tabla).toBeInTheDocument()
    });
  } catch (err) {
    actualErrorMsg = err.message;
  }
})
```

Figura 6.60 Prueba unitaria #30 para el FrontEnd [autoría propia]

Tabla 6.61 Datos de la prueba unitaria #31 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si el botón guardar es de clase primary		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

test(' Verificar si se crear el boton guardar con las propiedad de clase primary' , async ()=> {
  let actualErrorMsg;
  render(<Router><FechaRecepcion periodosDisponibles= {PeriodosDisponibles}/></Router>)
  try {
    waitFor(() => {
      const dropdown = screen.queryByTestId('botonGuardar');
      expect(dropdown).toHaveClass('ant-btn-primary')
    });
  } catch (err) {
    actualErrorMsg = err.message;
  }
})

```

Figura 6.61 Prueba unitaria #31 para el FrontEnd [autoría propia]

Tabla 6.62 Datos de la prueba unitaria #32 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test si el botón cancelar redirecciona		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

Debug
test(' Verificar si se crear el boton cancelar redirecciona a la seccion e secretaria' , async ()=> {

  render(<Router><FechaRecepcion periodosDisponibles= {PeriodosDisponibles}/></Router>)
  try {
    waitFor(() => {
      const dropdown = screen.queryByTestId('botoncancelar');
      expect(dropdown).toHaveAttribute('href', '/Secretaria')
    });
  } catch (err) {
    // noop
  }
})

```

Figura 6.62 Prueba unitaria #32 para el FrontEnd [autoría propia]

Tabla 6.63 Datos de la prueba unitaria #33 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si aparece la notificación de error		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si aparece la notificacion de error de la secretaria ', async ()=> {
  let actualErrorMsg;
  render(<Router><FechaRecepcion periodosDisponibles= {PeriodosDisponibles}/></Router>)
  try {
    waitFor(() => {
      const dropdown = screen.queryByTestId('titulo3');
      expect(dropdown).toBeDefined()
    });
  } catch (err) {
    actualErrorMsg = err.message;
  }
})
```

Figura 6.63 Prueba unitaria #33 para el FrontEnd [autoría propia]

Tabla 6.64 Datos de la prueba unitaria #34 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si se redirecciona a la pestaña de fecha recepción		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si redirecciona a la pestaña de fecha recepcion' , ()=> {
  render(<Router><Inicio /></Router>)
  const elemento = screen.queryByTestId('a1');
  expect(elemento).toHaveAttribute('href', '/fecha-recepcion')
})
```

Figura 6.64 Prueba unitaria #34 para el FrontEnd [autoría propia]

Tabla 6.65 Datos de la prueba unitaria #35 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si redirecciona a la pestaña de proyecto		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si redirecciona a la pestaña de secretaria proyecto' , ()=> {
  render(<Router><Inicio /></Router>)
  const dropdown = screen.queryById('a2');
  expect(dropdown).toHaveAttribute('href', '/secretaria/proyectos')
})
```

Figura 6.65 Prueba unitaria #35 para el FrontEnd [autoría propia]

Tabla 6.66 Datos de la prueba unitaria #36 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si redirecciona a la pestaña estudiante		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si redirecciona a la pestaña de secretaria estudiante' , ()=> {
  render(<Router><Inicio /></Router>)
  const dropdown = screen.queryById('a3');
  expect(dropdown).toHaveAttribute('href', '/Secretaria/Estudiantes')
})
```

Figura 6.66 Prueba unitaria #36 para el FrontEnd [autoría propia]

Tabla 6.67 Datos de la prueba unitaria #37 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si redirecciona la pestaña de notificar		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si redirecciona a la pestaña de notificar al estudiante' , ()=> {
  render(<Router><Inicio /></Router>)
  const dropdown = screen.queryById('a4');
  expect(dropdown).toHaveAttribute('href', '/notificar-estudiantes')
})
```

Figura 6.67 Prueba unitaria #37 para el FrontEnd [autoría propia]

Tabla 6.68 Datos de la prueba unitaria #38 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si redirecciona a la pestaña de rezagados		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si redirecciona a la pestaña de rezagados' , ()=> {
  render(<Router><Inicio /></Router>)
  const dropdown = screen.queryById('a5');
  expect(dropdown).toHaveAttribute('href', '/rezagados')
})
```

Figura 6.68 Prueba unitaria #38 para el FrontEnd [autoría propia]

Tabla 6.69 Datos de la prueba unitaria #39 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si renderia el card de proceso		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si se renderiza card de iniciar proceso' , ()=> {
  render(<Router><Inicio /></Router>)
  const dropdown = screen.queryById('card1');
  expect(dropdown).toHaveTextContent("INICIAR PROCESO")
})
```

Figura 6.69 Prueba unitaria #39 para el FrontEnd [autoría propia]

Tabla 6.70 Datos de la prueba unitaria #40 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test si renderiza card de proyecto		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si se renderiza card de proyecto' , ()=> {
  render(<Router><Inicio /></Router>)
  const dropdown = screen.queryById('card2');
  expect(dropdown).toHaveTextContent("PROYECTOS")
})
```

Figura 6.70 Prueba unitaria #40 para el FrontEnd [autoría propia]

Tabla 6.71 Datos de la prueba unitaria #41 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si renderiz card de estudiante		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si se renderiza card de estudiantes' , ()=> {
  render(<Router><Inicio /></Router>)
  const dropdown = screen.queryById('card3');
  expect(dropdown).toHaveTextContent("ESTUDIANTES")
})
```

Figura 6.71 Prueba unitaria #41 para el FrontEnd [autoría propia]

Tabla 6.72 Datos de la prueba unitaria #42 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test si renderiza card de estudiante		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si se renderiza card de estudiantes' , ()=> {
  render(<Router><Inicio /></Router>)
  const dropdown = screen.queryById('card3');
  expect(dropdown).toHaveTextContent("ESTUDIANTES")
})
```

Figura 6.72 Prueba unitaria #42 para el FrontEnd [autoría propia]

Tabla 6.73 Datos de la prueba unitaria #43 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si se renderiza card de notificar		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si se renderiza card de notificar' , ()=> {
  render(<Router><Inicio /></Router>)
  const dropdown = screen.queryByTestId('card4');
  expect(dropdown).toHaveTextContent("NOTIFICAR")
})
```

Figura 6.73 Prueba unitaria #43 para el FrontEnd [autoría propia]

Tabla 6.74 Datos de la prueba unitaria #44 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test si se renderiza card de rezagados		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test(' Verificar si se renderiza card de rezagados' , ()=> {
  render(<Router><Inicio /></Router>)
  const dropdown = screen.queryByTestId('card5');
  expect(dropdown).toHaveTextContent("REZAGADOS")
})
```

Figura 6.74 Prueba unitaria #44 para el FrontEnd [autoría propia]

Tabla 6.75 Datos de la prueba unitaria #45 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si funciona si muestra la matrícula del estudiante		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
const lista_estudiantesprueba= [{key: 2, 'matricula': '201705100',
'estudiante': 'LUIS ENRIQUE ANCHUNDIA SOLORZANO',
'correo': 'estud-fiec1@espol.edu.ec',
'año': '2022', 'termino': '1S',
'etapa': 'NO INICIADA'},

{'key': 3, 'matricula': '201706300',
'estudiante': 'SCARLET ANGELINA ESPINOZA MORENO',
'correo': 'estud-fiec2@espol.edu.ec', 'año': '2022',
'termino': '1S', 'etapa': 'SUBIR AL DSPACE'}]

test(' Verificar si funciona bien la tabla estudiante mostrando el campo matricula' , ()=> {
  let actualErrorMsg;
  render(<Router><NotificarEstudiantes lista_estudiantes= {lista_estudiantesprueba}/></Router>)
  try {
    waitFor(() => {
      expect(screen.getByTestId('tablaEst')).toMatch('201705100')
    });
  } catch (err) {
    actualErrorMsg= err.message
  }
})
```

Figura 6.75 Prueba unitaria #46 para el FrontEnd [autoría propia]

Tabla 6.76 Datos de la prueba unitaria #47 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test si la tabla se renderizs bien		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

const datos = {
  loading: true,
  searchText: "",
  searchedColumn: "",
  registros: [ {key:1,
    año: "2022",
    termino: '1S',
    tipo: FuncionesAuxiliares.asignarTipo('Integrador'),
    titulo: ['Proyecto integrador de prueba', '201706300'],
    carrera: 'Ingeniería en Computación',
    tutor:'Rafael Ignacio Bonilla Armijos',
    profesor: 'Eduardo Segundo Cruz Ramirez',
    matricula: '201706300',
    estudiante:'SCARLET ANGELINA ESPINOZA MORENO',
    estado: 'No iniciado' }],
  json: "",
  error: false
}

afterEach(cleanup)

test(' Verificar la tabla se renderiza bien' , ()=> {

  render(<Router><PrincipalEstudiante registros= {datos.registros}/></Router>)
  const dropdown = screen.queryById('TablaPrincipalEstudiante');
  expect(dropdown).toBeInTheDocument()

})

```

Figura 6.76 Prueba unitaria #47 para el FrontEnd [autoría propia]

Tabla 6.77 Datos de la prueba unitaria #48 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si se muestra la información del proyecto.		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022


```

const estudiante2={
  matricula:'201706300',
  apintegradora :true,
  culminomalla:true,
  tienedeuda:false,
  codigocarrera:'CCPG',
  id_usuario:3,
  gestiontitulo:'',
  nombretitulo:'Scarlet Angelina Espinoza Moreno'
}

const proyecto = {
  id: Number,
  titulo : "PROYECTO INTEGRADOR DE PRUEBA",
  termino : "1S",
  tipo : "INTEGRADOR",
  profesor : "EDUARDO SEGUNDO CRUZ RAMIREZ",
  tutores : ["RAFAEL IGNACIO BONILLA ARMIJOS"],
  estudiantes : [estudiante2, estudiantel],
  año : "2022",
  fechaInicio : "2022-07-27",
  programaAcademico : "",
  codigo : "PG-001"
}

afterEach(cleanup)

test (['Probar si se renderiza bien la información del titulo del proyecto',() =>{
  render(<Router> <InformacionProyecto proyecto = {proyecto} /></Router>)
  expect(screen.getByText(/PROYECTO INTEGRADOR DE PRUEBA/)).toBeInTheDocument();
}])

```

Figura 6.77 Prueba unitaria #48 para el FrontEnd [autoría propia]

Tabla 6.78 Datos de la prueba unitaria #49 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si muestra el año del proyecto y si se carga		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

test ('Probar si se renderiza bien el año del proyecto y si este se carga', () =>{
  render(<Router> <InformacionProyecto proyecto = {proyecto} /></Router>)
  expect(screen.queryAllByText(/2022/)).toBeTruthy();
})

```

Figura 6.78 Prueba unitaria #49 para el FrontEnd [autoría propia]

Tabla 6.79 Datos de la prueba unitaria #50 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar si aparece el titulo Integrador		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

test ('Probar si es verdad que no aparece la data como debe ser cargada en el tipo de tesis', () =>{
  render(<Router> <InformacionProyecto proyecto = {proyecto} /></Router>)
  expect(screen.queryAllByText(/INTEGRADOR/)).toBeUndefined()
})

```

Figura 6.79 Prueba unitaria #50 para el FrontEnd [autoría propia]

Tabla 6.80 Datos de la prueba unitaria #51 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test aprobar profesor caso 1 de paper		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

test('mostrarPaper caso1',()=>{
  const tipo = "investigacion"
  const estadoEncargado = {estado : 'P', fecha : '2022-07-22'}

  const estadoEstudiante = {estado : 'P', fecha : '2022-07-10'}

  const estadoSecretaria = {estado : '', fecha : ''}

  const resultado = aprobarProfesor.mostrarPaper(tipo, estadoEncargado,estadoEstudiante,estadoSecretaria)

  expect(resultado).toBe(true)
})

```

Figura 6.80 Prueba unitaria #51 para el FrontEnd [autoría propia]

Tabla 6.81 Datos de la prueba unitaria #52 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test aprobar profesor caso 2 de paper		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

test('mostrarPaper caso2',()=>{
  const tipo = "investigacion"
  const estadoEncargado = {estado : 'A', fecha : '2022-07-22'}

  const estadoEstudiante = {estado : 'C', fecha : '2022-07-25'}

  const estadoSecretaria = {estado : '', fecha : ''}

  const resultado = aprobarProfesor.mostrarPaper(tipo, estadoEncargado,estadoEstudiante,estadoSecretaria)

  expect(resultado).toBe(true)
})

```

Figura 6.81 Prueba unitaria #52 para el FrontEnd [autoría propia]

Tabla 6.82 Datos de la prueba unitaria #53 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test aprobar profesor caso 3 de paper		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test('mostrarPaper caso3',()=>{
  const tipo = "investigacion"
  const estadoEncargado = {estado : 'D', fecha : '2022-07-22'}

  const estadoEstudiante = {estado : 'C', fecha : '2022-07-25'}

  const estadoSecretaria = {estado : '', fecha : ''}

  const resultado = aprobarProfesor.mostrarPaper(tipo, estadoEncargado,estadoEstudiante,estadoSecretaria)

  expect(resultado).toBe(true)
})
```

Figura 6.82 Prueba unitaria #53 para el FrontEnd [autoría propia]

Tabla 6.83 Datos de la prueba unitaria #54 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test aprobar profesor caso 4 paper		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test('mostrarPaper caso4',()=>{
  const tipo = "investigacion"
  const estadoEncargado = {estado : '', fecha : ''}

  const estadoEstudiante = {estado : 'C', fecha : '2022-07-25'}

  const estadoSecretaria = {estado : 'D', fecha : '2022-07-15'}

  const resultado = aprobarProfesor.mostrarPaper(tipo, estadoEncargado,estadoEstudiante,estadoSecretaria)

  expect(resultado).toBe(true)
})
```

Figura 6.83 Prueba unitaria #54 para el FrontEnd [autoría propia]

Tabla 6.84 Datos de la prueba unitaria #55 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test mostrar mensaje caso 1		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test('mostrarMensajeIntegrador caso1',()=>{
  const estadoEstudiante = {estado : '', fecha : ''}
  const resultado = aprobarProfesor.mostrarMensajeIntegrador(estadoEstudiante)
  expect(resultado).toBe("El estudiante aún no ha subido el trabajo integrador.")
})
```

Figura 6.84 Prueba unitaria #55 para el FrontEnd [autoría propia]

Tabla 6.85 Datos de la prueba unitaria #56 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test mostrar mensaje caso 2		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test('mostrarMensajeIntegrador caso2',()=>{
  const estadoEstudiante = {estado : 'P', fecha : ''}
  const resultado = aprobarProfesor.mostrarMensajeIntegrador(estadoEstudiante)
  expect(resultado).toBe("El trabajo final ha sido devuelto debe esperar a que el tutor lo apruebe.")
})
```

Figura 6.85 Prueba unitaria #56 para el FrontEnd [autoría propia]

Tabla 6.86 Datos de la prueba unitaria #57 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test mostrar mensaje paper caso 1		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

test('mostrarMensajePaper caso1', ()=>{
  const profesor= new AprobarProfesor()
  profesor.props = {
    estadoEstudianteInv: {
      estado : ""
    },
    tipo : 'investigacion'
  }
  profesor.props.tipo = "investigacion"

  const resultado = profesor.mostrarMensajePaper()
  expect(resultado).toBe('El estudiante aún no ha subido el trabajo de investigación para su aprobación.')
})

```

Figura 6.86 Prueba unitaria #57 para el FrontEnd [autoría propia]

Tabla 6.87 Datos prueba unitaria #58

Sistema	Sistema AGATA		
Entregable	Test mostrar mensaje 2 caso paper		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

test('mostrarMensajePaper caso2', ()=>{
  const profesor= new AprobarProfesor()
  profesor.props = {
    estadoEstudianteInv: {
      estado : "A"
    },
    tipo : 'investigacion'
  }
  profesor.props.tipo = "investigacion"
  const resultado = profesor.mostrarMensajePaper()
})

```

Figura 6.87 Prueba unitaria #58 para el FrontEnd [autoría propia]

Tabla 6.88 Datos de la prueba unitaria #59 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test mostrar mensaje caso 4 paper		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

test('mostrarPaper caso4', ()=>{
  const tipo = "investigacion"
  const estadoEncargado = {estado : '', fecha : ''}

  const estadoEstudiante = {estado : 'C', fecha : '2022-07-25'}

  const estadoSecretaria = {estado : 'D', fecha : '2022-07-15'}

  const resultado = aprobarProfesor.mostrarPaper(tipo, estadoEncargado, estadoEstudiante, estadoSecretaria)

  expect(resultado).toBe(true)
})

```

Figura 6.88 Prueba unitaria #59 para el FrontEnd [autoría propia]

Tabla 6.89 Datos de la prueba unitaria #60 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test aprobar tutor caso 1		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```

test('mostrarIntegrador true caso1', ()=>{
  const estadoEncargado = {estado : 'P', fecha : '2022-08-01'}

  const estadoEstudiante = {estado : 'P', fecha : '2022-08-01'}

  const estadoSecretaria = {estado : 'C', fecha : '2022-08-01'}

  const estadoProfesor = {estado : 'D', fecha : '2022-08-01'}

  const resultado = aprobarTutor.mostrarIntegrador(estadoEncargado, estadoEstudiante, estadoProfesor, estadoSecretaria)

  expect(resultado).toBe(true)
})

```

Figura 6.89 Prueba unitaria #60 para el FrontEnd [autoría propia]

Tabla 6.90 Datos de la prueba unitaria #61 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test aprobar tutor caso 2		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test('mostrarIntegrador true caso2',()=>{
  const estadoEncargado = {estado : 'P', fecha : '2022-08-01'}
  const estadoEstudiante = {estado : 'P', fecha : '2022-08-01'}
  const estadoSecretaria = {estado : 'P', fecha : '2022-08-01'}
  const estadoProfesor = {estado : 'P', fecha : '2022-08-01'}
  const resultado = aprobarTutor.mostrarIntegrador(estadoEncargado,estadoEstudiante,estadoProfesor,estadoSecretaria)
  expect(resultado).toBe(true)
})
```

Figura 6.90 Prueba unitaria #61 para el FrontEnd [autoría propia]

Tabla 6.91 Datos de la prueba unitaria #62 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test aprobar tutor caso 3		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test('mostrarIntegrador true caso3',()=>{
  const estadoEncargado = {estado : 'D', fecha : '2022-07-01'}
  const estadoEstudiante = {estado : 'C', fecha : '2022-08-01'}
  const estadoSecretaria = {estado : 'p', fecha : '2022-08-01'}
  const estadoProfesor = {estado : 'p', fecha : '2022-08-01'}
  const resultado = aprobarTutor.mostrarIntegrador(estadoEncargado,estadoEstudiante,estadoProfesor,estadoSecretaria)
  expect(resultado).toBe(true)
})
```

Figura 6.91 Prueba unitaria #61 para el FrontEnd [autoría propia]

Tabla 6.92 Datos de la prueba unitaria #62 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test aprobar tutor caso 4		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test('mostrarIntegrador true caso4',()=>{
  const estadoEncargado = {estado : 'P', fecha : '2022-08-01'}

  const estadoEstudiante = {estado : 'C', fecha : '2022-08-01'}

  const estadoSecretaria = {estado : 'p', fecha : '2022-08-01'}

  const estadoProfesor = {estado : 'D', fecha : '2022-07-01'}

  const resultado = aprobarTutor.mostrarIntegrador(estadoEncargado,estadoEstudiante,estadoProfesor,estadoSecretaria)

  expect(resultado).toBe(true)
})
```

Figura 6.92 Prueba unitaria #62 para el FrontEnd [autoría propia]

Tabla 6.93 Datos de la prueba unitaria #63 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test aprobar tutor caso 5		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test('mostrarIntegrador true caso5',()=>{
  const estadoEncargado = {estado : 'A', fecha : '2022-08-01'}

  const estadoEstudiante = {estado : 'C', fecha : '2022-08-01'}

  const estadoSecretaria = {estado : 'D', fecha : '2022-07-01'}

  const estadoProfesor = {estado : 'p', fecha : '2022-08-01'}

  const resultado = aprobarTutor.mostrarIntegrador(estadoEncargado,estadoEstudiante,estadoProfesor,estadoSecretaria)

  expect(resultado).toBe(true)
})
```

Figura 6.93 Prueba unitaria #63 para el FrontEnd [autoría propia]

Tabla 6.94 Datos de la prueba unitaria #64 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test aprobar tutor caso 6		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test('mostrarIntegrador true caso6',()=>{
  const estadoEncargado = {estado : '', fecha : '2022-08-01'}
  const estadoEstudiante = {estado : 'P', fecha : '2022-08-01'}
  const estadoSecretaria = {estado : '', fecha : '2022-08-01'}
  const estadoProfesor = {estado : 'A', fecha : '2022-08-01'}
  const resultado = aprobarTutor.mostrarIntegrador(estadoEncargado,estadoEstudiante,estadoProfesor,estadoSecretaria)
  expect(resultado).toBe(true)
})
```

Figura 6.94 Prueba unitaria #64 para el FrontEnd [autoría propia]

Tabla 6.95 Datos de la prueba unitaria #65 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test mostrar integrador caso 7		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test('mostratIntegrador false',()=>{
  const estadoEncargado = {estado : 'p', fecha : '2022-08-01'}
  const estadoEstudiante = {estado : 'p', fecha : '2022-08-01'}
  const estadoSecretaria = {estado : 'p', fecha : '2022-08-01'}
  const estadoProfesor = {estado : 'p', fecha : '2022-08-01'}
  const resultado = aprobarTutor.mostrarIntegrador(estadoEncargado,estadoEstudiante,estadoProfesor,estadoSecretaria)
  expect(resultado).toBe(false)
})
```

Figura 6.95 Prueba unitaria #65 para el FrontEnd [autoría propia]

Tabla 6.96 Datos de la prueba unitaria #66 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test mostrar integrador mensaje caso 1		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test('mostrarMensajeIntegrador caso1', ()=>{
  const estadoEstudiante = {estado: ''}
  const resultado = aprobarTutor.mostrarMensajeIntegrador(estadoEstudiante)
  expect(resultado).toBe("El estudiante aún no ha subido el trabajo integrador.")
})
```

Figura 6.96 Prueba unitaria #66 para el FrontEnd [autoría propia]

Tabla 6.97 Datos de la prueba unitaria #67 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test mostrar integrador mensaje caso 2		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
test('mostrarMensajeIntegrador caso2', ()=>{
  const estadoEstudiante = {estado: 'P'}
  const resultado = aprobarTutor.mostrarMensajeIntegrador(estadoEstudiante)
  expect(resultado).toBe("El trabajo final ha sido devuelto. Debe esperar a que el estudiante suba el documento nuevamente.")
})
```

Figura 6.97 Prueba unitaria #67 para el FrontEnd [autoría propia]

Tabla 6.98 Datos de la prueba unitaria #68 [autoría propia]

Sistema	Sistema AGATA		
Entregable	Test verificar status GTSI		
Autor	Luis Enrique Anchundia Solorzano		
Versión/Edición	1.0	Fecha Versión	27/07/2022
Aprobado por	David Jurado	Fecha Aprobación	03/08 /2022

```
@pytest.mark.django_db
def test_GtsiStatus():
    request= HttpRequest()
    from administrador import views
    response = views.GtsiStatus().get(request)
    print(response)
    assert response.status_code== 200
```

Figura 6.98 Prueba unitaria #68 para el FrontEnd [autoría propia]

APÉNDICE B

INSTALACIÓN, CONFIGURACIÓN E INICIALIZACIÓN DE JENKINS (LINUX). REQUERIMIENTOS DE HARDWARE QUE NECESITA EL SISTEMA

mínimos:

- 256 MB de RAM
- 1 GB de espacio em disco (10GB es un mínimo recomendado si se ejecuta Jenkins como contenedor Docker)

Recomendados:

- 4 GB+ de RAM
- 50 GB+ de espacio en disco.

INSTALACIÓN

1. Para que Jenkins pueda ejecutarse es necesario tener instalado java, si es que no se tiene se lo puede instalar con los siguientes comandos:

```
$ sudo apt update
```

```
$ sudo apt install openjdk-11-jre
```

```
$ java -version
```

Después del último comando se deberá ver lo siguiente:

```
openjdk version "11.0.12" 2021-07-20
```

```
OpenJDK Runtime Environment (build 11.0.12+7-post-Debian-2)
```

```
OpenJDK 64-Bit Server VM (build 11.0.12+7-post-Debian-2, mixed mode, sharing)
```

2. Una vez que se cuente con java instalado, en este punto ya se puede instalar Jenkins con los siguientes comandos:

```
$ curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \  
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

```
$ echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \  
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \  
  /etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install Jenkins
```

```
$ sudo apt-get install Jenkins
```

3. Una vez Jenkins Instalado se debe habilitar e iniciar el servicio con los siguientes comandos:

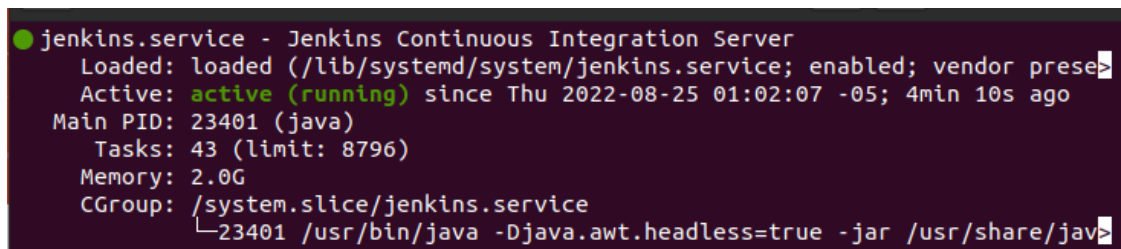
```
$ sudo systemctl enable Jenkins
```

```
$ sudo systemctl start jenkins
```

Con el comando a continuación se podrá ver el estado del servicio:

```
$ sudo systemctl status Jenkins
```

Si el proceso de instalación ha sido correcto el estado debe ser activo como se muestra a continuación:



```
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor prese
   Active: active (running) since Thu 2022-08-25 01:02:07 -05; 4min 10s ago
     Main PID: 23401 (java)
        Tasks: 43 (limit: 8796)
       Memory: 2.0G
      CGroup: /system.slice/jenkins.service
             └─23401 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/jav
```

Figura 5.1 Estado activo de la herramienta Jenkins [autoría propia].

CONFIGURACIÓN INICIAL

1. Por defecto Jenkins se instalará en el puerto 8080 de nuestro servidor local (localhost) o servidor remoto. Si tenemos un servidor remoto lo más seguro es que se debe permitir el tráfico de datos a través de ese puerto con un firewall, y esto lo realizamos con el siguiente comando:

```
$ sudo ufw allow 8080
```

2. Luego en el navegador escribimos la siguiente dirección:

<http://IP-SERVIDOR:8080>

Nos mostrará la siguiente ventana de inicio:

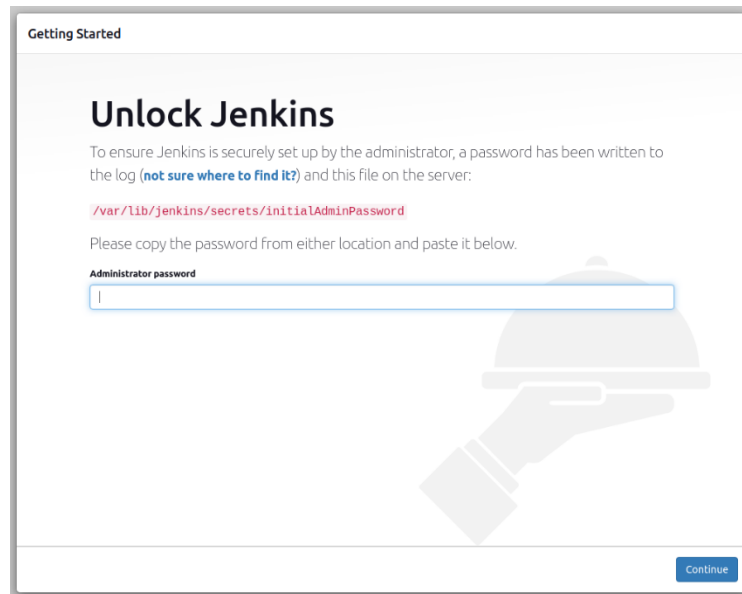


Figura 5.2 Primera vista de la herramienta Jenkins [autoría propia].

En la ventana de inicio se muestra dónde encontrar la contraseña inicial. La leemos desde la consola y la ingresamos.

3. Después de ingresar la contraseña se mostrará una ventana de personalización en donde se escogerá la opción:
Install suggested plugins



Figura 5.3 Instalación de plugin recomendados [autoría propia].

Luego de que termine el proceso de instalación Jenkins estará listo para usarse.

Inicialización en el proyecto

La instalación de Jenkins se realizó en el servidor con la dirección 200.9.176.97, y por defecto se usa el puerto 8080, por lo tanto, para ingresar a Jenkins debemos colocar en el navegador la siguiente dirección:

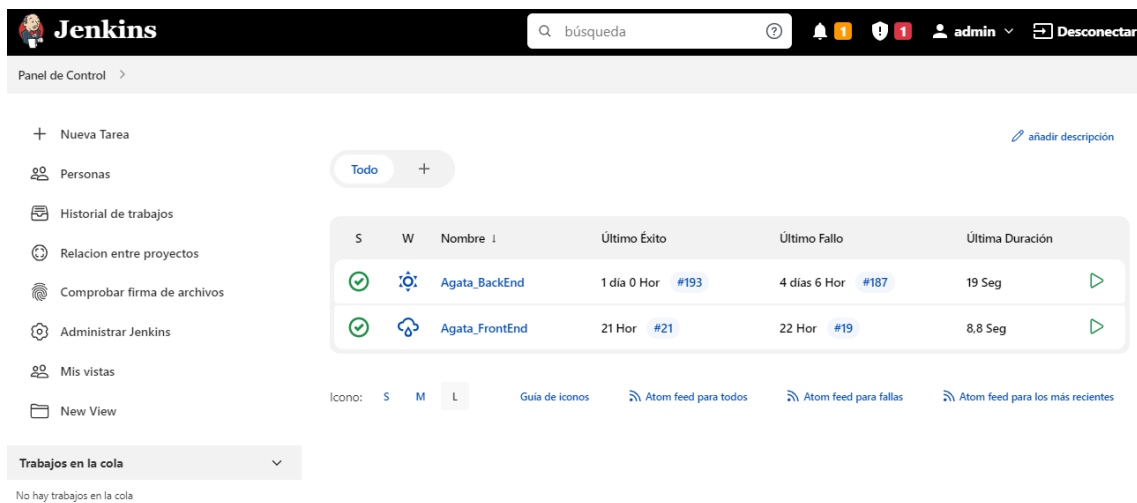
<http://200.9.176.97:8080>

y se usan las siguientes credenciales para ingresar como admin:

usuario: admin

contraseña: 94b48639e72340fcaecb417c7bfceaff

Una vez las credenciales hayan sido ingresadas se mostrará la siguiente pantalla de



The screenshot shows the Jenkins dashboard interface. At the top, there is a search bar and a user profile for 'admin'. The main content area displays a table of build jobs. The table has columns for 'S' (Success), 'W' (Warning), 'Nombre' (Name), 'Último Éxito' (Last Success), 'Último Fallo' (Last Failure), and 'Última Duración' (Last Duration). Two jobs are listed: 'Agata_BackEnd' and 'Agata_FrontEnd'. The 'Agata_BackEnd' job shows a success status, a last success time of 1 day 0 hours, and a last failure time of 4 days 6 hours. The 'Agata_FrontEnd' job shows a success status, a last success time of 21 hours, and a last failure time of 22 hours. Below the table, there are links for 'Guía de iconos', 'Atom feed para todos', 'Atom feed para fallas', and 'Atom feed para los más recientes'. On the left side, there is a sidebar with navigation options like 'Nueva Tarea', 'Personas', 'Historial de trabajos', etc. At the bottom, there is a section for 'Trabajos en la cola' (Jobs in queue) which currently shows 'No hay trabajos en la cola' (No jobs in queue).

S	W	Nombre	Último Éxito	Último Fallo	Última Duración
✓	⚠	Agata_BackEnd	1 día 0 Hor #193	4 días 6 Hor #187	19 Seg
✓	⚠	Agata_FrontEnd	21 Hor #21	22 Hor #19	8.8 Seg

Figura 5.4 Página principal herramienta Jenkins [autoría propia].

inicio:

En la pantalla de inicio se muestran los proyectos que se encuentran enlazados a través de un repositorio de GitHub, en este caso se tienen los dos repositorios correspondientes al proyecto Agata.

Observación:

Jenkins ya se encuentra instalado en el servidor 200.9.176.97 por lo tanto, no es necesario realizar el proceso de instalación ni de configuración.

APÉNDICE C

CÓDIGO SQL UTILIZADO PARA LA EJECUCIÓN DE LAS PRUEBAS DE SOFTWARE

```
INSERT INTO public.periodoacademico (id, nombre, "añoregistro", terminoacademico,
fechacreacion) VALUES (1, '1S - 2021', 2021, '1S', '2021-04-01 00:00:00');

INSERT INTO public.periodoacademico (id, nombre, "añoregistro", terminoacademico,
fechacreacion) VALUES (2, '2S - 2021', 2021, '2S', '2021-09-30 00:00:00');

INSERT INTO public.periodoacademico (id, nombre, "añoregistro", terminoacademico,
fechacreacion) VALUES (3, '1S - 2022', 2022, '1S', '2022-04-01 00:00:00');

INSERT INTO public.usuario (id, usuario, nombres, apellidos, cedula, correo,
recibecorreo, activo) VALUES (2, 'estud-fiec1', 'Luis Enrique', 'Anchundia
Solorzano', '0910257081', 'estud-fiec1@espol.edu.ec', true, true);

INSERT INTO public.usuario (id, usuario, nombres, apellidos, cedula, correo,
recibecorreo, activo) VALUES (3, 'estud-fiec2', 'Scarlet Angelina', 'Espinoza
Moreno', '0910857081', 'estud-fiec2@espol.edu.ec', true, true);

INSERT INTO public.usuario (id, usuario, nombres, apellidos, cedula, correo,
recibecorreo, activo) VALUES (4, 'prof-fiec1', 'Eduardo Segundo', 'Cruz Ramirez',
'0910256081', 'prof-fiec1@espol.edu.ec', true, true);

INSERT INTO public.usuario (id, usuario, nombres, apellidos, cedula, correo,
recibecorreo, activo) VALUES (5, 'secre-fiec1', 'Maria Rocio', 'Perez Suarez',
'0910256871', 'secre-fiec1@espol.edu.ec', true, true);

INSERT INTO public.usuario (id, usuario, nombres, apellidos, cedula, correo,
recibecorreo, activo) VALUES (6, 'tutor-fiec1', 'Rafael Ignacio', 'Bonilla
Armijos', '0910256099', 'tutor-fiec1@espol.edu.ec', true, true);

INSERT INTO public.usuario (id, usuario, nombres, apellidos, cedula, correo,
recibecorreo, activo) VALUES (7, 'subde-fiec1', 'Juan Carlos', 'Duarte Criollo',
'0910277081', 'subde-fiec1@espol.edu.ec', true, true);

INSERT INTO public.estudiante (matricula, apintegradora, culminomalla,
tienedeuda, codigocarrera, id_usuario, gestiontitulo, nombretitulo) VALUES
('201705100', true, true, false, 'CCPG', 2, '', 'Luis Enrique Anchundia
Solorzano');

INSERT INTO public.estudiante (matricula, apintegradora, culminomalla,
tienedeuda, codigocarrera, id_usuario, gestiontitulo, nombretitulo) VALUES
```

```

('201706300', true, true, true, 'CCPG', 3, '', 'Scarlet Angelina Espinoza
Moreno');

INSERT INTO public.rolusuario (idrol, idusuario, idperiodo, activo) VALUES (1, 2,
3, true);

INSERT INTO public.rolusuario (idrol, idusuario, idperiodo, activo) VALUES (1, 3,
3, true);

INSERT INTO public.rolusuario (idrol, idusuario, idperiodo, activo) VALUES (3, 4,
3, true);

INSERT INTO public.rolusuario (idrol, idusuario, idperiodo, activo) VALUES (2, 6,
3, true);

INSERT INTO public.rolusuario (idrol, idusuario, idperiodo, activo) VALUES (4, 5,
3, true);

INSERT INTO public.rolusuario (idrol, idusuario, idperiodo, activo) VALUES (5, 7,
3, true);

INSERT INTO public.proyectotitulacion (id, codigo, tipoproyecto, nombreproyecto,
fechainicio, fechafin, estado, multidisciplinario, coincidetitulo, subidospace,
idperiodo, activo, rezagado) VALUES (1, 'PG-001', 'Integrador', 'Proyecto
integrador de prueba', '2022-07-27 14:26:41', NULL, 'No iniciado', false, true,
false, 3, true, false);

INSERT INTO public.proyectotitulacion (id, codigo, tipoproyecto, nombreproyecto,
fechainicio, fechafin, estado, multidisciplinario, coincidetitulo, subidospace,
idperiodo, activo, rezagado) VALUES (2, 'PG-001', 'Investigacion', 'Proyecto
investigacion de prueba', '2022-07-27 14:26:41', NULL, 'No iniciado', false,
true, true, 3, true, false);

INSERT INTO public.documento (id, nombre, uri, fechasubida, fechaactualizacion,
idproyecto, idtipodocumento, idusuario, itemuri, metadatauri, bitstreamuri)
VALUES (1, 'PG-001_trabajo_final.pdf', 'files/PG-001_trabajo_final.pdf', '2021-
10-28 14:26:41', NULL, 1, 1, 2, NULL, NULL, NULL);

INSERT INTO public.documento (id, nombre, uri, fechasubida, fechaactualizacion,
idproyecto, idtipodocumento, idusuario, itemuri, metadatauri, bitstreamuri)
VALUES (2, '201706300_hoja_vida.pdf', 'files/201706300_hoja_vida.pdf', '2022-08-
01 14:26:41', NULL, 1, 3, 3, NULL, NULL, NULL);

INSERT INTO public.documento (id, nombre, uri, fechasubida, fechaactualizacion,
idproyecto, idtipodocumento, idusuario, itemuri, metadatauri, bitstreamuri)
VALUES (3, 'PG-001_paper.pdf', 'files/PG-001_paper.pdf', '2022-08-02 14:26:41',
NULL, 2, 2, 2, NULL, NULL, NULL);

```

```
INSERT INTO public.documento (id, nombre, uri, fechasubida, fechaactualizacion, idproyecto, idtipodocumento, idusuario, itemuri, metadatauri, bitstreamuri) VALUES (4, 'PG-001_antiplagio.pdf', 'files/PG-001_antiplagio.pdf', '2022-08-04 17:26:41', NULL, 1, 6, 3, NULL, NULL, NULL);

INSERT INTO public.proyektousuario (id, idusuario, idproyecto, idrol, activo, estado) VALUES (1, 2, 2, 1, true, 'No iniciado');

INSERT INTO public.proyektousuario (id, idusuario, idproyecto, idrol, activo, estado) VALUES (2, 3, 1, 1, true, 'No iniciado');

INSERT INTO public.proyektousuario (id, idusuario, idproyecto, idrol, activo, estado) VALUES (3, 4, 1, 3, true, 'No iniciado');

INSERT INTO public.proyektousuario (id, idusuario, idproyecto, idrol, activo, estado) VALUES (4, 5, 1, 4, true, 'No iniciado');

INSERT INTO public.proyektousuario (id, idusuario, idproyecto, idrol, activo, estado) VALUES (5, 6, 1, 2, true, 'No iniciado');

INSERT INTO public.proyektousuario (id, idusuario, idproyecto, idrol, activo, estado) VALUES (6, 7, 1, 5, true, 'No iniciado');

INSERT INTO public.etapa (id, fechainicio, fechafin, realizadopor, fechacreacion, idperiodo, idnombreetapa) VALUES (10, '2022-08-03 00:00:00', '2022-08-08 00:00:00', 'secre-fiec1', '2020-09-30 00:00:00', 3, 7);

INSERT INTO public.etapa (id, fechainicio, fechafin, realizadopor, fechacreacion, idperiodo, idnombreetapa) VALUES (11, '2022-08-03 00:00:00', '2022-08-08 00:00:00', 'secre-fiec1', '2020-09-30 00:00:00', 3, 7);

INSERT INTO public.etapaproceso (id, fechainicio, fechafin, idetapa, iddocumento) VALUES (1, '2022-08-03 00:00:00', '2022-08-09 00:00:00', 10, 1);

INSERT INTO public.etapaproceso (id, fechainicio, fechafin, idetapa, iddocumento) VALUES (2, '2022-08-03 00:00:00', '2022-08-09 00:00:00', 10, 2);

INSERT INTO public.pasostitulacion (id, idproyecto, idusuario, num_step, max_step, estadorevision) VALUES (1, 1, 3, 3, 6, 'Pendiente');

INSERT INTO public.pasostitulacion (id, idproyecto, idusuario, num_step, max_step, estadorevision) VALUES (2, 2, 2, 3, 6, 'Pendiente');

INSERT INTO public.actasaac (numeroacta, matricula, nombre, uri, subidasaac) VALUES (1, '201706300', '201706300_actaOriginal.pdf', 'actas/201706300_actaOriginal.pdf', true);
```

```
INSERT INTO public.actasaac (numeroacta, matricula, nombre, uri, subidasaac)
VALUES (2, '201705100', '201705100_actaOriginal.pdf',
'actas/201705100_actaOriginal.pdf', true);

INSERT INTO public.actasubdecanato (id, idestado, idacta, fecharecibo,
fechaenvio, nombre, uri) VALUES (2, 4, 1, '2022-08-11 00:00:00', '2022-08-10
00:00:00', '201706300_actaFirmada.pdf', 'actas/201706300_actaFirmada.pdf');

INSERT INTO public.actasubdecanato (id, idestado, idacta, fecharecibo,
fechaenvio, nombre, uri) VALUES (3, 4, 2, '2022-08-11 00:00:00', '2022-08-10
00:00:00', '201705100_actaFirmada.pdf', 'actas/201705100_actaFirmada.pdf');

INSERT INTO public.etapa(id, fechainicio, fechafin, realizadopor, fechacreacion,
idperiodo, idnombreetapa) VALUES(1, '2022-08-05 00:00:00','2022-08-20 00:00:00',
'secre-fiec1', '2022-08-04 22:00:07',3,1 );

INSERT INTO public.etapa(id, fechainicio, fechafin, realizadopor, fechacreacion,
idperiodo, idnombreetapa) VALUES(2, '2022-08-20 00:00:00','2022-08-26 00:00:00',
'secre-fiec1', '2022-08-04 22:00:07',3,2 );

INSERT INTO public.etapa(id, fechainicio, fechafin, realizadopor, fechacreacion,
idperiodo, idnombreetapa) VALUES(3, '2022-08-26 00:00:00','2023-09-15 00:00:00',
'secre-fiec1', '2022-08-04 22:00:07',3,3 );

INSERT INTO public.etapa(id, fechainicio, fechafin, realizadopor, fechacreacion,
idperiodo, idnombreetapa) VALUES(4, '2023-09-15 00:00:00','2023-10-06 00:00:00',
'secre-fiec1', '2022-08-04 22:00:07',3,4 );

INSERT INTO public.etapa(id, fechainicio, fechafin, realizadopor, fechacreacion,
idperiodo, idnombreetapa) VALUES(5, '2023-10-06 00:00:00','2023-10-13 00:00:00',
'secre-fiec1', '2022-08-04 22:00:07',3,6 );

INSERT INTO public.etapa(id, fechainicio, fechafin, realizadopor, fechacreacion,
idperiodo, idnombreetapa) VALUES(6, '2023-10-13 00:00:00','2023-10-20 00:00:00',
'secre-fiec1', '2022-08-04 22:00:07',3,7 );

INSERT INTO public.etapa(id, fechainicio, fechafin, realizadopor, fechacreacion,
idperiodo, idnombreetapa) VALUES(7, '2023-10-20 00:00:00','2023-10-27 00:00:00',
'secre-fiec1', '2022-08-04 22:00:07',3,8 );

INSERT INTO public.etapa(id, fechainicio, fechafin, realizadopor, fechacreacion,
idperiodo, idnombreetapa) VALUES(8, '2023-10-27 00:00:00','2023-11-03 00:00:00',
'secre-fiec1', '2022-08-04 22:00:07',3,5 );

INSERT INTO public.estadodocumento (id, fecha, idestado, realizadopor,
iddocumento, idrol) VALUES (1, '2022-08-01 04:07:04', 3, 'estud-fiec2', 1, 1);
```

APÉNDICE D

OBSERVACIONES REALIZADAS POR LA HERRAMIENTA SONARQUBE QUE FUERON REFACTORIZADAS.

```
110 obj_verificacion_trabajo = NombreEtapa.objects.get(
111     nombre="Verificación de trabajo final")
```

Define a constant instead of duplicating this literal "Verificación de trabajo final" 6 times. 3 days ago ▾ L111 🔗
Why is this an issue?
🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 12min effort Comment 🏷️ No tags ▾

```
115     nombre="Verificación de documentos personales")
116 obj_extraer_acta = NombreEtapa.objects.get(nombre="Extraer Acta de Grado")
```

Define a constant instead of duplicating this literal "Extraer Acta de Grado" 3 times. Why is this an issue? 3 days ago ▾ L116 🔗
🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 6min effort Comment 🏷️ No tags ▾

```
144 }
145 headers = {
146     'Content-type': 'application/json',
```

Define a constant instead of duplicating this literal 'application/json' 6 times. Why is this an issue? 3 days ago ▾ L146 🔗
🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 12min effort Comment 🏷️ No tags ▾

```
242     if request.session["justLoggedIn"]:
243         request.session["justLoggedIn"] = False
244     except:
```

Specify an exception class to catch or reraise the exception Why is this an issue? 3 days ago ▾ L244 🔗
🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 5min effort Comment 🏷️ No tags ▾

```
254     return JsonResponse({"status": "running" if response.status_code == 200 else "error"}, status=resp
255     except:
```

Specify an exception class to catch or reraise the exception Why is this an issue? 3 days ago ▾ L255 🔗
🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 5min effort Comment 🏷️ No tags ▾

```
259
260 class AutenticacionCAS(generics.GenericAPIView):
261     permission_classes = [IsAuthenticated]
262
263     def get(self, request):
```

Refactor this function to reduce its Cognitive Complexity from 174 to the 15 allowed. 3 days ago ▾ L263 🔗
Why is this an issue?
🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 2h44min effort Comment 🏷️ No tags ▾

Figura 5.5 Observaciones refactorizadas 6/73 [autoría propia].

848	<code>info_hoja['nombre'] = hoja.nombre</code>	
849	<code>info_hoja['fechaSubida'] = hoja.fechasubida.strftime("%Y-%m-%d")</code>	<p>Define a constant instead of duplicating this literal "%Y-%m-%d" 18 times. Why is this an issue? 3 days ago ▾ L849 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 36min effort Comment 🏷️ No tags ▾</p>
867	<code>roles = obtener_rol_termino(request.user.username, id_perodo)</code>	
868	<code>if not id_rol_secretaria in roles:</code>	
869	<code>return JsonResponse({'msg': 'Usuario no autorizado'}, status=403)</code>	<p>Define a constant instead of duplicating this literal 'Usuario no autorizado' 15 times. Why is this an issue? 3 days ago ▾ L869 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 30min effort Comment 🏷️ No tags ▾</p>
394	<code># proyectoIntegradorExiste)</code>	
395	<code>if(proyectoIntegradorExiste is False):</code>	
396	<code>proyectoTitulacion = ProyectoTitulacion(tipoProyecto=tipoProyecto, codigo=codigoPro)</code>	<p>Define a constant instead of duplicating this literal "No iniciado" 6 times. Why is this an issue? 3 days ago ▾ L396 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 12min effort Comment 🏷️ No tags ▾</p>
713	<code>return JsonResponse(res, status=200)</code>	
714	<code>except:</code>	<p>Specify an exception class to catch or reraise the exception Why is this an issue? 3 days ago ▾ L714 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 5min effort Comment 🏷️ No tags ▾</p>
814	<code>else:</code>	
815	<code>return JsonResponse({'error': 'Tipo de usuario no autorizado'}, status=403)</code>	<p>Define a constant instead of duplicating this literal 'Tipo de usuario no autorizado' 5 times. Why is this an issue? 3 days ago ▾ L815 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 10min effort Comment 🏷️ No tags ▾</p>
816	<code>except (MateriaIntegradora.DoesNotExist, ProyectoUsuario.DoesNotExist, Estudiante.DoesNotExist):</code>	
817	<code>print(traceback.format_exc())</code>	
818	<code>return JsonResponse({"error": "No se ha encontrado ningún registro de proyecto."}, status=404)</code>	<p>Define a constant instead of duplicating this literal "No se ha encontrado ningún registro de proyecto." 4 times. Why is this an issue? 3 days ago ▾ L818 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 8min effort Comment 🏷️ No tags ▾</p>
956	<code>if fs.exists(file_path):</code>	
957	<code>with fs.open(file_path) as pdf:</code>	
958	<code>response = HttpResponse(pdf, content_type='application/pdf')</code>	<p>Define a constant instead of duplicating this literal 'application/pdf' 6 times. Why is this an issue? 3 days ago ▾ L958 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 12min effort Comment 🏷️ No tags ▾</p>
959	<code>response['Content-Disposition'] = 'attachment'</code>	
960	<code>return response</code>	
961	<code>return JsonResponse({'msg': 'no exists'}, status=400)</code>	<p>Define a constant instead of duplicating this literal 'no exists' 4 times. Why is this an issue? 3 days ago ▾ L961 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🟡 Open ▾ Not assigned ▾ 8min effort Comment 🏷️ No tags ▾</p>

Figura 5.6 Observaciones refactorizadas 14/73 [autoría propia].

969 970	<pre>file_name = matricula + '_' + tipo + '.pdf' file_path = 'files/' + file_name</pre>	<p>Define a constant instead of duplicating this literal 'files/' 9 times. Why is this an issue? 3 days ago ▾ L970 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🔓 Open ▾ Not assigned ▾ 18min effort Comment 🏷️ No tags ▾</p>
1022 1023	<pre>pdf, content_type='application/pdf') response['Content-Disposition'] = 'attachment; filename='+file_name+'''</pre>	<p>Define a constant instead of duplicating this literal 'attachment; filename=' 3 times. Why is this an issue? 4 days ago ▾ L1023 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🔓 Open ▾ Not assigned ▾ 6min effort Comment 🏷️ No tags ▾</p>
1034 1035	<pre>file_name = matricula + '_actaOriginal.pdf' file_path = 'actas/' + file_name</pre>	<p>Define a constant instead of duplicating this literal 'actas/' 3 times. Why is this an issue? 4 days ago ▾ L1035 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🔓 Open ▾ Not assigned ▾ 6min effort Comment 🏷️ No tags ▾</p>
1081 1082 1083	<pre>return response return JsonResponse({'msg': 'no exists'}, status=400) except:</pre>	<p>Specify an exception class to catch or reraise the exception Why is this an issue? 4 days ago ▾ L1083 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🔓 Open ▾ Not assigned ▾ 5min effort Comment 🏷️ No tags ▾</p>
1084 1085	<pre>print(traceback.format_exc()) return JsonResponse({'msg': 'Archivo no existe'}, status=400)</pre>	<p>Define a constant instead of duplicating this literal 'Archivo no existe' 3 times. Why is this an issue? 4 days ago ▾ L1085 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🔓 Open ▾ Not assigned ▾ 6min effort Comment 🏷️ No tags ▾</p>
1160 1161 1162	<pre># return JsonResponse({'msg': file_path}, status=200) return JsonResponse({'url': file_path}, status=200) except:</pre>	<p>Specify an exception class to catch or reraise the exception Why is this an issue? 4 days ago ▾ L1162 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🔓 Open ▾ Not assigned ▾ 5min effort Comment 🏷️ No tags ▾</p>
1270 1271 1272	<pre>def notiSecretaria(matricula): estudiante = Estudiante.objects.filter(matricula=matricula) print("tengo al estudiante")</pre>	<p>Define a constant instead of duplicating this literal "tengo al estudiante" 3 times. Why is this an issue? 4 days ago ▾ L1272 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🔓 Open ▾ Not assigned ▾ 6min effort Comment 🏷️ No tags ▾</p>
1275	<pre>mensaje = "El acta del estudiante " + \</pre>	<p>Define a constant instead of duplicating this literal "El acta del estudiante " 3 times. Why is this an issue? 4 days ago ▾ L1275 🔗</p> <p>🗑️ Code Smell ▾ 🚨 Critical ▾ 🔓 Open ▾ Not assigned ▾ 6min effort Comment 🏷️ No tags ▾</p>

Figura 5.7 Observaciones refactorizadas 22/73 [autoría propia].

1273
1274

```
subdecanato = Rolusuario.objects.filter(idrol=id_rol_subdecanato)
print("obtengo el rol del subdecanato", subdecanato)
```

Define a constant instead of duplicating this literal "obtengo el rol del subdecanato" 3 times. 4 days ago ▾ L1274 🔗
Why is this an issue?
🔧 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 6min effort Comment No tags ▾

1276
1277

```
estudiante[0].id_usuario.nombres+ " " + \
estudiante[0].id_usuario.apellidos+" acaba de ser enviado."
```

Define a constant instead of duplicating this literal "acaba de ser enviado." 3 times. 4 days ago ▾ L1277 🔗
Why is this an issue?
🔧 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 6min effort Comment No tags ▾

1278

```
print("escribo el mensaje a notificar")
```

Define a constant instead of duplicating this literal "escribo el mensaje a notificar" 3 times. 4 days ago ▾ L1278 🔗
Why is this an issue?
🔧 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 6min effort Comment No tags ▾

1430
1431

```
def post(self, request, matricula, id, dspace, deuda):
```

Refactor this function to reduce its Cognitive Complexity from 17 to the 15 allowed. 4 days ago ▾ L1431 🔗
Why is this an issue?
🔧 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 7min effort Comment No tags ▾

1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450

```
estudiante = Estudiante.objects.filter(matricula=matricula)
usuario = Usuario.objects.filter(id=estudiante[0].id_usuario.id)
documento = Documento.objects.filter(
    idproyecto=id).order_by('idtipodocumento')
proyecto1 = Proyectousuario.objects.filter(
    idusuario=estudiante[0].id_usuario.id)
proyecto = Proyectotitulacion.objects.filter(id=id)
ultimo = ultimo_termino()
etapa = Etapa.objects.filter(
    idperiodo=ultimo.id, idnombreetapa=obtenerNombreEtapa("Subir al DSPACE"))

1 if(deuda == "true"):
    respuesta = eliminarDeuda(matricula)
2 if(respuesta['codigo'] == 200):
    estudiante.update(tienedeuda=False)
    log = Logtitulacion(fechaevento=datetime.date.today(), descripcion="Se ha quitado la deuda de r
        idproyecto=proyecto1[0].idproyecto, idetapa=etapa[0], idproyuser=proyecto1
    log.save()
    raise
```

1651

```
raise Http404("Registro no encontrado")
```

Define a constant instead of duplicating this literal "Registro no encontrado" 3 times. 4 days ago ▾ L1651 🔗
Why is this an issue?
🔧 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 6min effort Comment No tags ▾

1926
1927

```
document = request.FILES['file']
if document is None or document.content_type != "application/pdf":
```

Define a constant instead of duplicating this literal "application/pdf" 7 times. Why is this an issue? 4 days ago ▾ L1927 🔗
🔧 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 14min effort Comment No tags ▾

Figura 5.8 Observaciones refactorizadas 28/73 [autoría propia].

1650

`except:`

Specify an exception class to catch or reraise the exception Why is this an issue? 4 days ago ▾ L1650 🔗

🚫 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment 🏷️ No tags ▾

2014

`class AvanceEtapa(generics.GenericAPIView):`

2015

`def get(self, request, estudiante):`

Refactor this function to reduce its Cognitive Complexity from 39 to the 15 allowed. 4 days ago ▾ L2015 🔗

Why is this an issue?

🚫 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 29min effort Comment 🏷️ No tags ▾

2016

`try:`

2017

`usuario = Usuario.objects.get(usuario=estudiante, activo=True)`

2018

`proyecto = ''`

2019

`try:`

2020

`proyecto = Proyecto.objects.filter(idusuario=usuario.id).order_by('-id').first()`

2021

`#print("proyecto id: "+str(proyecto.id))`

2022

`1 except MultipleObjectsReturned:`

2023

`print(traceback.format_exc())`

2024

`return JsonResponse({'error': 'El usuario no está autorizado para realizar la petición'}, stat`

2025

`documentos = Documento.objects.filter(`

2026

`idproyecto=proyecto.idproyecto)`

2027

`print(documentos)`

2028

`# Lista donde se guardan todos los archivos que ha subido el estudiante.`

2029

`lista_docs = list()`

2030

`2 if proyecto.idproyecto.tipoproyecto == 'Investigacion':`

2031

`articulo_inv = documentos.get(idtipodocumento=id_inv).last()`

2024

`return JsonResponse({'error': 'El usuario no está autorizado para realizar la petición'}, status=403)`

Define a constant instead of duplicating this literal 'El usuario no está autorizado para realizar la petición' 4 times. Why is this an issue? 4 days ago ▾ L2024 🔗

🚫 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 8min effort Comment 🏷️ No tags ▾

2055

`fecha = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")`

Define a constant instead of duplicating this literal "%Y-%m-%d %H:%M:%S" 12 times. 4 days ago ▾ L2055 🔗

Why is this an issue?

🚫 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 24min effort Comment 🏷️ No tags ▾

2145

`def get(self, request, tipo, id_documento):`

2146

Refactor this function to reduce its Cognitive Complexity from 18 to the 15 allowed. 4 days ago ▾ L2146 🔗

Why is this an issue?

🚫 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 8min effort Comment 🏷️ No tags ▾

2147

`r"""Realiza la descarga de archivos relacionados a la fase grupal.`

2148

`Parameters`

2149

`-----`

2150

`tipo : str`

2151

`Tipo del documento que puede ser final, paper, correccion o entregable.`

2152

2153

2154

`id_documento: int`

2328

`)`

2329

`descrip = "El usuario " + usuario.usuario + " subió el entregable " + nombre`

Define a constant instead of duplicating this literal "El usuario " 16 times. Why is this an issue? 4 days ago ▾ L2329 🔗

🚫 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 32min effort Comment 🏷️ No tags ▾

Figura 5.9 Observaciones refactorizadas 34/73 [autoría propia].

2366

```
def subir_trabajo_graduacion(datos, proyecto, usuario, estado):
```

Refactor this function to reduce its Cognitive Complexity from 30 to the 15 allowed. 4 days ago L2366

Why is this an issue?

Code Smell Critical Open Not assigned 20min effort Comment No tags

2422

```
    contenido = 'El estudiante ' + (usuario.nombres + " "+usuario.apellidos).upper(
```

Define a constant instead of duplicating this literal 'El estudiante ' 6 times. Why is this an issue? 4 days ago L2422

Code Smell Critical Open Not assigned 12min effort Comment No tags

2383

```
    if file_data is None or file_data.content_type != "application/pdf":
```

2384

```
        return {'Mensaje': 'Por favor, suba un archivo en formato PDF', "Estado": 400}
```

Define a constant instead of duplicating this literal 'Por favor, suba un archivo en formato PDF' 4 times. Why is this an issue? 4 days ago L2384

Code Smell Critical Open Not assigned 8min effort Comment No tags

2219

```
def post(self, request, tipo, estudiante):
```

Refactor this function to reduce its Cognitive Complexity from 16 to the 15 allowed. 4 days ago L2219

Why is this an issue?

Code Smell Critical Open Not assigned 6min effort Comment No tags

2220

```
    r"""Permite subir los archivos correspondientes a la fase grupal del proceso de titulación.
```

2221

```
    Parameters
```

2222

```
    -----
```

2223

```
    tipo : str
```

2224

```
        Tipo del documento que se subirá.
```

2225

```
    estudiante : str
```

2226

```
        Nombre de usuario del estudiante que sube el documento.
```

2227

2433

```
        descripcion = "El usuario " + usuario.usuario + \
```

2434

```
            " subió la corrección del documento " + file_name
```

Define a constant instead of duplicating this literal " subió la corrección del documento " 3 times. 4 days ago L2434

Why is this an issue?

Code Smell Critical Open Not assigned 6min effort Comment No tags

2436

```
        fechaevento=fecha, descripcion=descripcion, idusuario=usuario, iddocumento=doc_final)
```

2437

```
        log.save()
```

2438

```
        return {"Mensaje": 'El documento con id: '+str(doc_final.id)+' se actualizó con éxito', "Es
```

Define a constant instead of duplicating this literal 'El documento con id: ' 3 times. Why is this an issue? 4 days ago L2438

Code Smell Critical Open Not assigned 6min effort Comment No tags

2455

```
    contenido = 'El estudiante ' + (usuario.nombres + ' ' + usuario.apellidos).upper(
```

2456

```
    ) + ' ha subido el documento.' + file_name + ' como trabajo final.'
```

Define a constant instead of duplicating this literal ' ha subido el documento ' 3 times. 4 days ago L2456

Why is this an issue?

Code Smell Critical Open Not assigned 6min effort Comment No tags

2470

```
        datos_proyecto = {
```

2471

```
            'estado': 'En proceso', 'fechainicio': fecha}
```

Define a constant instead of duplicating this literal 'En proceso' 3 times. Why is this an issue? 4 days ago L2471

Code Smell Critical Open Not assigned 6min effort Comment No tags

Figura 5.1 Observaciones refactorizadas 42/73 [autoría propia].

```

2514
2515 def subir_articulo_investigacion(datos, proyecto, usuario, estado):
2516
2517     """Función que permite realizar la subida del artículo de investigación.
2518
2519     Parameters
2520     -----
2521     datos: request
2522         Datos provenientes del request.
2523     proyecto : Projectousuario
2524         Proyecto usuario que se desea subir o actualizar.
2525     usuario : Usuario
2526         Usuario que sube o actualiza el archivo.
2527     estado : Estadodocumento
2528         Estado del documento que se desea asignar.
2529     """
2530
2531     try:
2532         with transaction.atomic():
2533             file_data = datos.FILES['file']
2534             1 if file_data is None 2 or file_data.content_type != "application/pdf":
2535                 return {"Mensaje": 'Por favor, suba un archivo en formato PDF', "Estado": 400}
2536             3 else:
2537                 # Datos del artículo: titulo, resumen, palabras clave
2538
2539                 return JsonResponse({'msg': 'no exists'}, status=400)
2540             except:
2541
2542                 except Exception:
2543                     print("function subir_paper: ", traceback.format_exc())
2544                     return {'Mensaje': 'No se pudo completar la petición, intente más tarde', "Estado": 400}
2545
2546                 except IntegrityError:
2547                     print(traceback.format_exc())
2548                     return {"Mensaje": "Ocurrió un error al registrar los datos del trabajo final", "Estado": 500}
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000

```

Refactor this function to reduce its Cognitive Complexity from 23 to the 15 allowed. 4 days ago L2515

Why is this an issue?

Code Smell Critical Open Not assigned 13min effort Comment No tags

Specify an exception class to catch or reraise the exception Why is this an issue? 4 days ago L2675

Code Smell Critical Open Not assigned 5min effort Comment No tags

Define a constant instead of duplicating this literal 'No se pudo completar la petición, intente más tarde' 6 times. Why is this an issue? 4 days ago L2649

Code Smell Critical Open Not assigned 12min effort Comment No tags

Define a constant instead of duplicating this literal "Ocurrió un error al registrar los datos del trabajo final" 3 times. Why is this an issue? 4 days ago L2492

Code Smell Critical Open Not assigned 6min effort Comment No tags

Refactor this function to reduce its Cognitive Complexity from 16 to the 15 allowed. 4 days ago L2805

Why is this an issue?

Code Smell Critical Open Not assigned 6min effort Comment No tags

Figura 5.11 Observaciones refactorizadas 47/73 [autoría propia].

```

2679
2680 def subir_hoja_vida(datos, proyecto, usuario, estado):

```

Refactor this function to reduce its Cognitive Complexity from 26 to the 15 allowed. 4 days ago ▾ L2680 [🔗](#)

Why is this an issue?

Code Smell ▾ Critical ▾ Open ▾ Not assigned ▾ 16min effort [Comment](#) No tags ▾

```

2681     """Función que permite realizar la subida de la hoja de vida e información personal.
2682
2683     Parameters
2684     -----
2685     datos: request
2686         Datos provenientes del request.
2687     proyecto : Proyectousuario
2688         Proyecto usuario que se desea subir o actualizar.
2689     usuario : Usuario
2690         Usuario que sube o actualiza el archivo.
2691     estado : Estadodocumento
2692         Estado del documento que se desea asignar.
2693     """
2694     try:
2695         with transaction.atomic():
2696             file_data = datos.FILES['file']
2697             1 if file_data is None 2 or file_data.content_type != "application/pdf":
2698                 return {'Mensaje': 'Por favor, suba un archivo en formato PDF', 'Estado': 400}
2699             3 else:
2700                 estudiante = Estudiante.objects.get(id_usuario=usuario.id)
2701                 nombresCompletos = datos.data['nombresCompletos']

```

```

2483
2484         # Agrega registro en LogTitulacion.
2485         nuevo_descripcion = "El usuario " + usuario.usuario + \
                " subió el documento " + file_name

```

Define a constant instead of duplicating this literal " subió el documento " 5 times. Why is this an issue? 4 days ago ▾ L2485 [🔗](#)

Code Smell ▾ Critical ▾ Open ▾ Not assigned ▾ 10min effort [Comment](#) No tags ▾

```

2487         fechaevento=fecha, descripcion=nuevo_descripcion, idusuario=usuario, iddocumento=documento_nuevo.id)
2488         log_nuevo.save()
2489         return {"Mensaje": "El documento se ha creado con éxito. Id: " + str(documento_nuevo.id), "Estado": 200}

```

Define a constant instead of duplicating this literal "El documento se ha creado con éxito. Id: " 4 times. Why is this an issue? 4 days ago ▾ L2489 [🔗](#)

Code Smell ▾ Critical ▾ Open ▾ Not assigned ▾ 8min effort [Comment](#) No tags ▾

```

3100
3101 def get(self, request, id_proyecto):

```

Refactor this function to reduce its Cognitive Complexity from 20 to the 15 allowed. 4 days ago ▾ L3101 [🔗](#)

Why is this an issue?

Code Smell ▾ Critical ▾ Open ▾ Not assigned ▾ 10min effort [Comment](#) No tags ▾

```

3102     r"""Permite obtener información del proyecto de titulación de un estudiante, dado el ID del proyecto.
3103
3104     Parameters
3105     -----
3106     id_proyecto : int
3107         ID del proyecto de titulación
3108
3109     Returns
3110     -----
3111     JsonResponse
3112         Objeto con los datos de proyecto, trabajo final, paper y entregables.
3113     """
3114     try:
3115         secretaria = Rolusuario.objects.filter(
3116             idrol=id_rol_secretaria, idusuario__usuario=request.user.username).exists()
3117         1 if (secretaria):
3118             proyecto = Proyectotitulacion.objects.get(id=id_proyecto)
3119             respuesta = dict()
3120             respuesta['id'] = proyecto.id

```

Figura 5.12 Observaciones refactorizadas 51/73 [autoría propia].

```

3007     def get(self, request, id_proyecto):

```

Refactor this function to reduce its Cognitive Complexity from 22 to the 15 allowed. 4 days ago ▾ L3007 🔗

Why is this an issue?

🚩 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 12min effort Comment

🏷️ No tags ▾

```

3008         r"""Permite obtener información del proyecto de titulación de un estudiante, dado el ID del proyecto.
3009
3010         Parameters
3011         -----
3012         id_proyecto : int
3013             ID del proyecto de titulación
3014
3015         Returns
3016         -----
3017         JsonResponse
3018             Objeto con los datos de proyecto, trabajo final, paper y entregables.
3019         """
3020         try:
3021             proyecto = Proyectousuario.objects.filter(
3022                 idusuario__usuario=request.user.username, idproyecto=id_proyecto)
3023             lista_rols = [rol.idrol.id for rol in proyecto]
3024             1 if (len(lista_rols) == 1 2 or len(lista_rols) == 2) 3 and id_rol_tutor in lista_rols:
3025                 proyecto = Projectotitulacion.objects.get(id=id_proyecto)
3026                 respuesta = dict()
3027                 respuesta['id'] = proyecto.id
3028                 respuesta['titulo'] = proyecto.nombreproyecto.upper()
3029                 respuesta['tipo'] = proyecto.tipoproyecto.upper()
3030                 respuesta['codigo'] = proyecto.codigo

```

Define a constant instead of duplicating this literal 'No se pudo completar la solicitud, intente más tarde.' 4 times. 4 days ago ▾ L2892 🔗

Why is this an issue?

🚩 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 8min effort Comment

🏷️ No tags ▾

```

2892         return JsonResponse({'error': 'No se pudo completar la solicitud, intente más tarde.'}, status=500)

```

```

3671         inicio_proceso = datetime.datetime.strptime(
3672             request.data['inicioProceso'], '%Y-%m-%dT%H:%M:%S.%F%z')

```

Define a constant instead of duplicating this literal '%Y-%m-%dT%H:%M:%S.%F%z' 3 times. 4 days ago ▾ L3672 🔗

Why is this an issue?

🚩 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 6min effort Comment

🏷️ No tags ▾

```

3348     def estado_revisión(id_rol, id_documento, id_proyecto):

```

Refactor this function to reduce its Cognitive Complexity from 43 to the 15 allowed. 4 days ago ▾ L3348 🔗

Why is this an issue?

🚩 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 33min effort Comment

🏷️ No tags ▾

```

3349         r"""Devuelve un diccionario con el estado más reciente que ha sido asignado a un documento por determinado rol.
3350
3351         Parameters
3352         -----
3353         id_rol : int
3354             id del rol del que se desea obtener la revisión.
3355         id_documento : int
3356             id del documento del que se desea obtener el estado de revisión.
3357         id_proyecto : int
3358             id del proyecto del que se desea obtener el estado de revisión.
3359         Returns
3360         -----
3361         dict
3362             Diccionario con el estado y fecha de revisión.

```

Figura 5.13 Observaciones refactorizadas 55/73 [autoría propia].

```

3879     def get(self, request, usuario, id_periodo):

```

Refactor this function to reduce its Cognitive Complexity from 20 to the 15 allowed. 4 days ago ▾ L3879 🔗
 Why is this an issue?
🚫 Code Smell ▾ 🔴 Critical ▾ 🔵 Open ▾ 📌 Not assigned ▾ 🕒 10min effort 💬 Comment 🏷️ No tags ▾

```

3880         try:
3881             roles = obtener_rol_termino(usuario, id_periodo)
3882             # Se obtiene la etapa.
3883             etapa = Etapa.objects.get(
3884                 idperiodo_id=id_periodo, idnombreetapa_nombre="Verificación de trabajo final")
3885             # Si NO es un rol autorizado
3886             1 if not ((len(roles) > 1 2 and (id_rol_profesor in roles 3 or id_rol_tutor in roles)) 4 or (ler
3887                 return JsonResponse({'error': 'Usuario no autorizado'}, status=403)
3888
3889             listaProyectos = list()
3890             info_panel = {'noIniciado': 0, 'pendientes': 0,
3891                 'devueltos': 0, 'aprobados': 0}
3892             noIniciados = 0
3893             pendientes = 0
3894             devueltos = 0

```

```

3956     def get(self, request, usuario, id_periodo):
3957

```

Refactor this function to reduce its Cognitive Complexity from 16 to the 15 allowed. 4 days ago ▾ L3957 🔗
 Why is this an issue?
🚫 Code Smell ▾ 🔴 Critical ▾ 🔵 Open ▾ 📌 Not assigned ▾ 🕒 6min effort 💬 Comment 🏷️ No tags ▾

```

3958         try:
3959             roles = obtener_rol_termino(usuario, id_periodo)
3960             # Se obtiene la etapa.
3961             etapa = Etapa.objects.get(
3962                 idperiodo_id=id_periodo, idnombreetapa_nombre="Verificación de trabajo final")
3963             # Si no es de un rol autorizado
3964             1 if not (id_rol_tutor in roles):
3965                 return JsonResponse({'error': 'Usuario no autorizado'}, status=403)
3966
3967             listaProyectos = list()
3968             info_panel = {'noIniciado': 0, 'pendientes': 0,
3969                 'devueltos': 0, 'aprobados': 0}
3970             noIniciados = 0
3971             pendientes = 0

```

```

4026     def estado_proyecto(id_proyecto, tipo_proy):
4027

```

Refactor this function to reduce its Cognitive Complexity from 29 to the 15 allowed. 4 days ago ▾ L4027 🔗
 Why is this an issue?
🚫 Code Smell ▾ 🔴 Critical ▾ 🔵 Open ▾ 📌 Not assigned ▾ 🕒 19min effort 💬 Comment 🏷️ No tags ▾

```

4028         """
4029         Función que establece el estado global que posee un proyecto.
4030
4031         Parameters
4032         -----
4033             id_proy: int
4034                 id del proyecto a consultar.
4035
4036             tipo_proy: string

```

```

4916                 notificar_por_rol(
4917                     rol_usuario, fecha, "aprobación", user, documento)

```

Define a constant instead of duplicating this literal "aprobación" 6 times. Why is this an issue? 4 days ago ▾ L4917 🔗
🚫 Code Smell ▾ 🔴 Critical ▾ 🔵 Open ▾ 📌 Not assigned ▾ 🕒 12min effort 💬 Comment 🏷️ No tags ▾

Figura 5.14 Observaciones refactorizadas 59/73 [autoría propia].

```
4237 def obtener_rol_proyecto(usuario, id_proyecto):
4238
4239     Rol del usuario en el proyecto de interés.
4240
4241     Parameters
4242     -----
4243     usuario: string
4244         Nombre del usuario a consultar.
4245
4246     id_proyecto: int
4247         Clave primaria del proyecto que se desea consultar.
4248
4249     Returns
4250     -----
4251     int:
4252         La clave primaria del rol.
4253
4254     '''
4255     roles = Proyectousuario.objects.filter(
4256         idusuario__usuario=usuario, idproyecto=id_proyecto, activo=True)
4257     1 if roles.exists():
4258         # Convierte el resultado en un arreglo con el/los rol(es).
4259         lista_rols = [rol.idrol.id for rol in roles]
4260     2 else:
4261         lista_rols = []
4262     return lista_rols

4704 def turno_realizar_revision(documento, usuario, rol, id_investigacion):
4705
4706     Función que permite validar si al usuario le corresponde realizar la revisión.
4707
4708     Parameters
4709     -----
4710     documento: Documento
4711         Objeto de tipo Documento al que se le realizará la revisión.
4712     usuario: string

4872 def post(self, request):
4873
4874     try:
4875         with transaction.atomic():
4876             usuario = request.data["usuario"]
4877             id_doc = request.POST.get("id_documento")
4878             fecha = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
4879             documento = Documento.objects.get(id=id_doc)
4880             user = Usuario.objects.get(usuario=usuario)
4881             tipo_doc = Tipodocumento.objects.get(
4882                 id=documento.idtipodocumento.id)
4883             rol_usuario = obtener_rol_proyecto(
4884                 usuario, documento.idproyecto)
4885             # Acción que sólo pueden realizar los encargados de revisión
4886             info_proyecto = get_info_proyecto_individual(
4887                 request.data["id_proyecto"])
4888             proyecto = info_proyecto.proyecto
4889             1 if rol_usuario in [id_rol_profesor, id_rol_secretaria, id_rol_tutor]:
4890                 condicion = turno_aprobar(
4891                     documento, usuario, rol_usuario)
4892                 2 if (condicion['puede_aprobar']):
4893                     3 if 'file' in self.request.FILES:
4894                         4 and rol_usuario == id_rol_profesor:
4895                             file_data = self.request.FILES['file']
4896                             tamaño = transformar(
```

Figura 5.15 Observaciones refactorizadas 62/73 [autoría propia].

```

4970 def turno_aprobar(documento, usuario, rol):
4971
4972     '''
4973     Función que permite validar si al usuario le corresponde aprobar un documento.
4974
4975     Parameters
4976     -----
4977     documento: Documento
4978         Objeto de tipo Documento al que se le realizará la revisión.
4979     usuario: string
4980         Nombre de usuario del que está realizando la revisión.
4981
4982     '''
4983
4984     def post(self, request):
4985
4986         1 if request.method == 'POST':
4987             try:
4988                 2 if request.data["usuario"] is None 3 or request.data["id_documento"] is None:
4989                     return JsonResponse({"error": "No fue posible realizar esta acción"}, status=204)
4990                 4 else:
4991                     usuario = request.data["usuario"]
4992                     documento = Documento.objects.filter(
4993                         id=request.data["id_documento"]).first()
4994                     print(documento)
4995                     id_investigacion = None
4996                     5 if 'id_articulo' in request.data:
4997                         id_investigacion = request.data["id_articulo"]
4998                         user = Usuario.objects.get(usuario=usuario)
4999                         fecha = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
5000                         rol_usuario = obtener_rol_proyecto(
5001                             usuario, documento.idproyecto)
5002                         print(rol_usuario)
5003                         rol = Rol.objects.filter(id=rol_usuario).first()
5004                         print("Se obtiene de Rol filtro rol:")
5005
5006             else:
5007                 print(traceback.format_exc())
5008                 return JsonResponse({'error': 'Acción no permitida'}, status=403)
5009
5010         elif evento == "devolución":
5011             2 descrip = ("El documento " + documento.nombre + " del proyecto " + nombre_proyecto +
5012
5013             2 descrip = ("El documento " + documento.nombre + " del proyecto " + nombre_proyecto +
5014
5015             2 descrip = ("El documento " + documento.nombre + " del proyecto " + nombre_proyecto +

```

Refactor this function to reduce its Cognitive Complexity from 75 to the 15 allowed. 4 days ago L4970

Why is this an issue?

Code Smell Critical Open Not assigned 1h5min effort Comment No tags

Refactor this function to reduce its Cognitive Complexity from 54 to the 15 allowed. 4 days ago L4608

Why is this an issue?

Code Smell Critical Open Not assigned 44min effort Comment No tags

Define a constant instead of duplicating this literal 'Acción no permitida' 3 times. Why is this an issue? 4 days ago L4701

Code Smell Critical Open Not assigned 6min effort Comment No tags

Define a constant instead of duplicating this literal "El documento " 4 times. Why is this an issue? 4 days ago L5796

Code Smell Critical Open Not assigned 8min effort Comment No tags

Define a constant instead of duplicating this literal " del proyecto " 4 times. Why is this an issue? 4 days ago L5796

Code Smell Critical Open Not assigned 8min effort Comment No tags

Figura 5.16 Observaciones refactorizadas 67/73 [autoría propia].


```

5224     def post(self, request):

```

Refactor this function to reduce its Cognitive Complexity from 33 to the 15 allowed. 4 days ago ▾ L5224 🔗

Why is this an issue?

🔗 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 23min effort Comment

🏷️ No tags ▾

```

5225         1 if request.method == 'POST':
5226             comentario = request.data["comentario"]
5227         2 if comentario == "" 3 or comentario is None:
5228             return JsonResponse({'error': 'Debe al menos escribir un comentario'}, status=400)
5229         4 else:
5230             try:
5231                 with transaction.atomic():
5232                     id_documento = request.data["id_documento"]
5233                     fecha = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
5234                     documento = Documento.objects.get(id=id_documento)
5235                     info_proyecto = get_info_proyecto_individual(
5236                         documento.idproyecto.id)
5237                     realizado = Usuario.objects.get(
5238                         usuario=request.data["usuario"])
5239                     tipo_doc = Tipodocumento.objects.get(
5240                         id=documento.idtipodocumento.id)
5241                     tamaño = transformar(
5242                         tipo_doc.tamaño_max, tipo_doc.unidad)
5243                     rol_usuario = obtener_rol_proyecto(

```

```

5796     describ = ("El documento " + documento.nombre + " del proyecto " + nombre_proyecto +
5797             " ha sido devuelto por " + nombres_emisor+". Usted debe realizar la revisión nuevamente.")

```

Define a constant instead of duplicating this literal " ha sido devuelto por " 4 times. 4 days ago ▾ L5797 🔗

Why is this an issue?

🔗 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 8min effort Comment

🏷️ No tags ▾

```

6287     except:

```

Specify an exception class to catch or reraise the exception Why is this an issue? 4 days ago ▾ L6287 🔗

🔗 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment

🏷️ No tags ▾

```

5793     describ = ("El estudiante " + nombres_emisor + " ha subido el documento " +
5794             documento.nombre + " en el proyecto " + nombre_proyecto)

```

Define a constant instead of duplicating this literal " en el proyecto " 3 times. Why is this an issue? 4 days ago ▾ L5794 🔗

🔗 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 6min effort Comment

🏷️ No tags ▾

```

6040     class Reporte(generics.GenericAPIView):
6041         permission_classes = [IsAuthenticated]
6042
6043         def get(self, request, id_periodo):

```

Refactor this function to reduce its Cognitive Complexity from 18 to the 15 allowed. 4 days ago ▾ L6043 🔗

Why is this an issue?

🔗 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 8min effort Comment

🏷️ No tags ▾

```

6044         res = dict()
6045         total = dict()
6046         detalle = dict()
6047         documentos = Documento.objects.filter(
6048             idtipodocumento=1, idproyecto_activo=True, idproyecto_idperiodo=id_periodo)
6049         1 for doc in documentos:
6050             tmp = dict()
6051             proyecto = Proyectotitulacion.objects.filter(
6052                 activa=True, id=doc.idproyecto.id)
6053             # Se crea la notificación que se mostrará en la web
6054             notificar_por_rol(
6055                 rol_usuario, fecha, "devolución", realizado, documento)

```

Define a constant instead of duplicating this literal "devolución" 7 times. Why is this an issue? 4 days ago ▾ L5292 🔗

🔗 Code Smell ▾ 🚨 Critical ▾ 🔵 Open ▾ Not assigned ▾ 14min effort Comment

🏷️ No tags ▾

Figura 5.17 Observaciones refactorizadas 73/73 [autoría propia].

APÉNDICE E

ACTA DE ACEPTACIÓN DE PRUEBAS DE SOFTWARE

En la ciudad de Guayaquil, a los 30 días del mes de agosto del 2022 en cumplimiento del proyecto “Desarrollo de una versión mejorada del sistema AGATA de la ESPOL”, se hace entrega a los ingenieros, Ing. Margarita del Rocío Filián Gómez e Ing. David Alonso Jurado Mosquera un total de 98 pruebas de software que corresponden a pruebas unitarias y de integración para el BackEnd y el FrontEnd del sistema, dichas pruebas fueron aceptadas de manera satisfactoria por los Ingenieros mencionados anteriormente.



firmado electrónicamente por:
MARGARITA DEL
ROCIO FILIAN
GOMEZ

Ing. Margarita del Rocío Filián Gómez



Ing. David Alonso Jurado Mosquera

APÉNDICE F

ACTA DE CIERRE


En la ciudad de Guayaquil, a los 30 días del mes de agosto del 2022 en cumplimiento del proyecto “Desarrollo de una versión mejorada del sistema AGATA de la ESPOL”, se hace entrega a los ingenieros, Ing. Margarita del Rocío Filián Gómez e Ing. David Alonso Jurado Mosquera, las implementaciones realizadas durante el lapso del proyecto para el sistema AGATA, las cuales cumplen con los requerimientos propuestos y fueron aceptados de manera satisfactoria por los Ingenieros mencionados anteriormente. Los entregables se describen a continuación:

- Automatización de pruebas de software con la herramienta Jenkins alojada en el servidor 200.9.176.97 y un total de 98 pruebas desarrolladas y aprobadas.
- Análisis de la calidad del código del BackEnd y FrontEnd con la herramienta SonarQube.
- Código del sistema refactorizado, el mismo que ya se encuentra modificado en los repositorios proporcionados.



Firmado digitalmente por:
MARGARITA DEL
ROCIO FILIAN
GOMEZ

Ing. Margarita del Rocío Filián Gómez



Ing. David Alonso Jurado Mosquera