



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

Curso Académico 2023/2024

Trabajo Fin de Grado

**AUDITORÍA DEL FRONT-END DE LA PÁGIA WEB DE LA UNIVERSIDAD
REY JUAN CARLOS**

Autora: Marina Fernández Suárez

Director: Julio Guillén García

Licencia

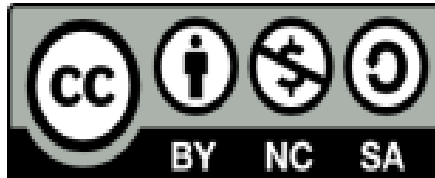


Ilustración 1. Licencia Creative Commons

Usted es libre de:

- **Compartir:**

Copiar y redistribuir el material en cualquier medio o formato.

- **Adaptar:**

Remezclar, transformar y construir a partir del material. **Bajo los siguientes términos:**

- **Atribución:**

Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciante.

- **No Comercial:**

Usted no puede hacer uso del material con propósitos comerciales.

- **Compartir Igual:**

Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.



Agradecimientos

Quiero manifestar mi profundo agradecimiento a todas las personas que han sido clave en la culminación de este proyecto y a lo largo de mi trayectoria académica. A mis profesores, por su dedicación y conocimientos, que han sido cruciales para mi formación. Y especialmente a mis tutores, por su apoyo constante, paciencia y confianza en mis capacidades, cuya guía ha sido indispensable para alcanzar este logro.

A mis padres, por su amor incondicional, por su constante apoyo y por motivarme en los momentos más difíciles, siendo siempre una fuente de inspiración y fortaleza. Y a mis hermanas, por su comprensión, motivación constante y su respaldo en cada paso del camino. Vosotros habéis sido mi mayor fortaleza y principal inspiración para seguir adelante.

A mis abuelos, por su amor y por estar siempre dispuestos a escucharme. Saber que siempre estaban ahí, aunque no siempre los tenga físicamente conmigo, ha sido un gran consuelo y un impulso para levantarme cada vez que tropezaba. A mis tíos, mis primos y al resto de mi familia, por estar presentes en los momentos difíciles, y por sus ánimos y palabras de aliento. Su cariño ha sido una fuerza vital en mi camino.

A una persona muy especial, por su aliento y por dedicar su tiempo a ayudarme y animarme siempre que lo he necesitado. Tu apoyo ha sido invaluable. A mis amigos, por su comprensión y disposición para escucharme, así como por ser una vía de desconexión y por sus consejos llenos de buenos deseos.

A mis compañeros de ambas carreras, por los recuerdos compartidos, tanto buenos como desafiantes, por las horas de estudio y trabajo conjunto, y por su compañerismo a lo largo de estos años. A mis compañeros de todos los cursos que he realizado durante estos años, por su colaboración y amistad, haciendo del aprendizaje una experiencia más enriquecedora.

Y a mis vecinos, por sus ánimos constantes y por estar siempre dispuestos a ayudar. Su apoyo ha sido un gran respaldo en mi camino.

Resumen

Este trabajo de fin de carrera se centra en la auditoría de la página oficial de la Universidad Rey Juan Carlos mediante la automatización de pruebas con Python y Gherkin, y el uso de herramientas externas para los casos no contemplados para desarrollar. El objetivo principal es evaluar la funcionalidad, rendimiento y seguridad del sitio web utilizando una variedad de herramientas y librerías especializadas. Se implementarán diversas pruebas, incluyendo pruebas de navegación, búsqueda, formularios y enlaces, tanto en la sección principal como en el dominio urjc.es. La metodología incluye la planificación detallada y la aplicación de prácticas de integración continua, asegurando así la calidad y consistencia de las pruebas realizadas.

Proceso de Desarrollo

- **Análisis de Requisitos:** Se realizó un análisis exhaustivo para identificar los requisitos funcionales y no funcionales del proyecto. Esto incluyó la identificación de las áreas clave del sitio web a auditar, así como la especificación de los criterios de éxito para cada tipo de prueba. El análisis se basó en revisiones de documentación existente, asegurando una comprensión clara de los objetivos y expectativas del proyecto.
- **Planificación:** La planificación involucró la creación de un cronograma que abarcaba todas las fases del proyecto, desde el desarrollo inicial de las pruebas, hasta su implementación y validación. Se definieron hitos específicos y se asignaron los recursos necesarios, incluyendo las herramientas de pruebas y desarrollos. Esta etapa también incluyó la identificación de errores potenciales y la formulación de estrategias para mitigarlos.
- **Desarrollo de Pruebas:** Durante esta fase, se diseñaron e implementaron los diferentes tipos de pruebas. Se desarrollaron *scripts* automatizados utilizando Python, Selenium y Gherkin para desarrollarlas.



- **Pruebas con Herramientas Externas:** Se utilizaron diversas herramientas de pruebas para complementar las pruebas desarrolladas. JMeter se empleó para pruebas de rendimiento, OWASP ZAP para pruebas de seguridad, Equal Web para las pruebas de usabilidad, y Hosted Scan para evaluar la accesibilidad. Estas pruebas permitieron una evaluación integral del sitio web desde múltiples perspectivas.
- **Validación:** En la fase de validación, se revisaron y analizaron los resultados de las pruebas para identificar áreas de mejora. Se llevaron a cabo iteraciones de prueba y corrección hasta cumplir con los requisitos establecidos. Se generaron informes que interpretaron los resultados y las acciones tomadas para resolver cualquier problema surgido.

Resultados y Beneficios

El proyecto resultó en la identificación de diversas áreas problemáticas en la página web de la URJC, gracias a lo cual se le brinda a la universidad la oportunidad de corregirlas y mejorar su rendimiento, seguridad y accesibilidad. Las pruebas automatizadas aseguraron una evaluación continua y consistente del sitio, lo que permite a la universidad la implementación de mejoras que optimicen la experiencia del usuario. El uso de herramientas especializadas garantizó la calidad y fiabilidad de los resultados obtenidos.

Abstract

This final degree project focuses on the audit of the official website of the Universidad Rey Juan Carlos through test automation with Python and Gherkin, and the use of external tools for cases not covered by development. The main objective is to evaluate the functionality, performance, and security of the website using a variety of specialized tools and libraries. Various tests will be implemented, including navigation, search, forms, and links tests, both in the main section and throughout the domain urjc.es. The methodology includes detailed planning and the application of continuous integration practices, thus ensuring the quality and consistency of the tests performed.

Development Process

- **Requirements Analysis:** A thorough analysis was conducted to identify the functional and non-functional requirements of the project. This included identifying key areas of the website to audit, as well as specifying success criteria for each type of test. The analysis was based on reviews of existing documentation, ensuring a clear understanding of the project's objectives and expectations.
- **Planning:** Planning involved creating a schedule that covered all phases of the project, from the initial development of tests to their implementation and validation. Specific milestones were defined, and necessary resources were allocated, including testing tools and development resources. This stage also included identifying potential errors and formulating strategies to mitigate them.
- **Test Development:** During this phase, different types of tests were designed and implemented. Automated scripts were developed using Python, Selenium, and Gherkin for their execution.
- **Testing with External Tools:** Various testing tools were used to complement the developed tests. JMeter was used for performance testing, OWASP ZAP for security testing, Equal Web

for usability testing, and Hosted Scan for accessibility testing. These tests allowed for a comprehensive evaluation of the website from multiple perspectives.

- **Validation:** In the validation phase, the test results were reviewed and analysed to identify areas for improvement. Iterations of testing and correction were carried out until the established requirements were met. Reports were generated interpreting the results and actions taken to resolve any identified issues.

Results and Benefits

The project resulted in the identification and resolution of various problem areas on the URJC website, thus providing the university with the opportunity to correct and improve its performance, security, and accessibility. Automated tests ensured a continuous and consistent evaluation of the site, allowing the university to implement improvements that optimize user experience. The use of specialized tools guaranteed the quality and reliability of the results obtained.



Índice de contenidos

Capítulo 1 Introducción	14
1.1 Descripción del problema	14
1.2 Normativas	15
1.3 Motivación	16
1.4 Objetivos	16
1.5 Planificación	17
1.6 Estructura de la memoria	20
Capítulo 2 Marco teórico	22
2.1 Análisis de conceptos principales	22
2.1.1 Testing	22
2.1.2 Auditorías	22
2.1.3 Automatización en testing	23
2.2 Herramientas para definir escenarios de pruebas	24
2.3 Otras herramientas	25
2.4 Librerías	29
2.5 Metodologías	31
2.5.1 Automatización	31
2.5.2 Behavior Driven Development	31



Capítulo 3 Planificación y desarrollo	32
3.1 Introducción	32
3.2 Planteamiento inicial	32
3.3 Diagramas de Actividades:	32
3.4 Uso de Gherkin para las pruebas	35
3.5 Desarrollo	35
Capítulo 4 Diseño de pruebas automáticas	38
4.1 Requisitos	38
4.1.1 Funcionales.....	38
4.1.2 No Funcionales	39
4.2 Casos de prueba	39
4.3 Diagramas UML de Casos de Uso	40
4.4 Implementación de las pruebas	46
4.4.1 Pruebas con Selenium	46
4.4.2 Pruebas de Rendimiento	55
4.4.3 Pruebas de Seguridad.....	55
4.4.4 Pruebas de Accesibilidad.....	56
4.4.5 Pruebas de Usabilidad	56
4.4.6 Resumen.....	57
Capítulo 5 Validación	58



5.1 Introducción	58
5.2 Corrección de Errores	58
5.3 Validación de requisitos	59
5.3.1 Requisitos Funcionales (RF).....	59
5.3.2 Requisitos no Funcionales (RNF)	61
5.4 Revisión de pruebas	62
5.4.1 Pruebas de rendimiento. JMeter	62
5.4.2 Pruebas de Seguridad. HostedScan.....	69
5.4.3 Pruebas de Seguridad con OWASP ZAP	72
5.4.4 Pruebas de Accesibilidad. WAVE.....	75
5.4.5 Pruebas de Usabilidad. Equal Web	78
5.5 Pruebas funcionales	81
5.5.1 Pruebas en la Sección Principal.....	81
5.5.2 Pruebas Simples	83
5.5.3 Pruebas en Profundidad del Dominio URJC	84
5.6 Resultados finales	87
Capítulo 6 Conclusiones	88
6.1 Logros alcanzados	88
6.2 Validación de Objetivos	88
6.2.1 Objetivo Principal (OP)	88



6.2.2 Objetivos Específicos (OE).....	88
6.3 Logros y Desafíos	90
6.4 Lecciones Aprendidas	91
6.5 Líneas futuras	92
Anexo	93
Glosario	94
Bibliografía	97



Índice de ilustraciones

Ilustración 1. Licencia Creative Commons	1
Ilustración 2. Diagrama de Gantt	19
Ilustración 3. Gherkin	25
Ilustración 4. Google Chrome.....	26
Ilustración 5. Apache JMeter	26
Ilustración 6. OWASP ZAP	27
Ilustración 7. HostedScan.....	27
Ilustración 8. WAVE.....	28
Ilustración 9. Equal Web	28
Ilustración 10. Diagrama de Actividades.....	34
Ilustración 11. CU 1 Configurar Entorno de Pruebas	40
Ilustración 12. CU 2 Definir Pruebas	41
Ilustración 13. CU 3 Ejecutar Pruebas.....	41
Ilustración 14. CU 4 Analizar Resultados.....	41
Ilustración 15. CU 5 Generar Reportes	42
Ilustración 16. CU 6 Corregir Errores	42
Ilustración 17. CU 7 Validar Requisitos	43
Ilustración 18. CU 8 Monitorear Rendimiento.....	43
Ilustración 19. CU 9 Proporcionar Retroalimentación	43
Ilustración 20. CU 10 Actualizar Configuraciones	44
Ilustración 21. CU 11 Gestionar Incidencias.....	44
Ilustración 22. CU 12 Realizar Pruebas de Seguridad	44
Ilustración 23. CU 13 Realizar Pruebas de Usabilidad.....	45



Ilustración 24. CU 14 Revisar Cumplimiento de Normativas	45
Ilustración 25. CU 15 Automatizar Pruebas	45
Ilustración 26. CU 16 Integrar Herramientas de Prueba.....	46
Ilustración 27. urjc_test_information_help_check.feature 1	49
Ilustración 28. urjc_test_information_help_check.feature 2	49
Ilustración 29. Steps urjc_test_information_help_check.feature 2	51
Ilustración 30. Steps urjc_test_information_help_check.feature 2	51
Ilustración 31. deep_urjc_test_check_links.feature	53
Ilustración 32. Steps deep_urjc_test_check_links.feature 1	54
Ilustración 33. Steps deep_urjc_test_check_links.feature 2	54
Ilustración 34. Gráfico de Tiempo de respuesta JMeter 1 usuario	62
Ilustración 35. Gráfico de Tiempo de respuesta JMeter 1000 usuarios	69
Ilustración 36. Hosted Scan - Exploraciones	70
Ilustración 37. Hosted Scan - Resultados	71
Ilustración 38. Análisis Owasp Zap	72
Ilustración 39. Prueba Wave	75
Ilustración 43. Wave Contraste	75
Ilustración 43. Wave Alertas	75
Ilustración 43. Wave Características	75
Ilustración 43. Wave Elementos estructurales	75
Ilustración 44. Extensión de Chrome Equal Web	78
Ilustración 45. Resultados Equal Web.....	78



Índice de tablas

Tabla 1. Cronograma del proyecto.....	18
Tabla 2. JMeter resultados 10 usuarios	64
Tabla 3. JMeter resultados 100 usuarios	66
Tabla 4. JMeter resultados 1000 usuarios	68
Tabla 5. HostedScan - Vulnerabilidades en Ap. Web Pasivas	71
Tabla 6. Alertas de OWASP ZAP	72
Tabla 7. Equal Web Errores	79

Introducción

1.1 Descripción del problema

En el ámbito educativo, la presencia digital de las instituciones universitarias es esencial para garantizar el acceso a la información y servicios académicos. La Universidad Rey Juan Carlos (en adelante URJC), consciente de esta necesidad, ha desarrollado su sitio web oficial como un punto de referencia para estudiantes, profesores y la comunidad en general. Sin embargo, la calidad, funcionalidad y seguridad de este portal deben ser constantemente evaluadas y mejoradas para asegurar una experiencia de usuario óptima.

Se plantea el desarrollo de una auditoría exhaustiva del sitio web de la URJC utilizando técnicas de pruebas automatizadas. Esta auditoría tiene como objetivo identificar posibles fallos, evaluar el rendimiento y asegurar el cumplimiento de los estándares de seguridad y accesibilidad. Mediante el uso de herramientas avanzadas como PyCharm [1], Gherkin [2] y Python [3], se llevarán a cabo una serie de pruebas diseñadas para cubrir diversos aspectos del sitio web.

El objetivo es realizar una auditoría del sitio web de la URJC, utilizando herramientas diseñadas para este ámbito, y pruebas automatizadas para garantizar su funcionalidad, rendimiento y seguridad. Para ello, se implementarán pruebas de navegación, búsqueda, formularios, enlaces y rendimiento, tanto en la página principal como en todo el dominio urjc.es. Los resultados obtenidos se analizarán para identificar áreas de mejora y garantizar que el sitio web cumpla con las expectativas de los usuarios y los estándares actuales [4].

1.2 Normativas

A nivel global y en España, existen diversas normativas relevantes que aseguran la calidad, accesibilidad y seguridad de un sitio web, y que deben considerarse.

Normativas Globales:

ISO/IEC 25010:2011: Esta norma internacional define un modelo de calidad para sistemas y software, incluyendo características como la funcionalidad, rendimiento, compatibilidad, usabilidad, fiabilidad, seguridad y mantenibilidad [6].

WCAG 2.1 (Web Content Accessibility Guidelines o Pautas de accesibilidad al contenido web):

Estas directrices proporcionan un estándar para la accesibilidad web, asegurando que el contenido sea accesible para personas con discapacidades. Incluyen principios y criterios de conformidad que deben cumplirse para garantizar una experiencia inclusiva [7].

GDPR (General Data Protection Regulation o Regulación General de Protección de Datos):

Aunque es una normativa de la Unión Europea, su impacto es global. Establece requisitos para la protección de datos personales y la privacidad, lo cual es esencial para cualquier sitio web que maneje información de usuarios [8].

Normativas en España:

Real Decreto 1112/2018 sobre accesibilidad de los sitios web y aplicaciones móviles del sector público:

Este decreto transpone la Directiva Europea 2016/2102, estableciendo requisitos específicos para la accesibilidad de los sitios web y aplicaciones móviles del sector público, incluyendo universidades [9].

Estas normativas proporcionan un marco legal y de calidad que guía el desarrollo y la auditoría de sitios web. Cumplir con ellas no solo asegura el cumplimiento legal, sino que también mejora la experiencia de usuario.

1.3 Motivación

La URJC es una institución educativa reconocida por su compromiso con la calidad y la innovación. Su sitio web oficial [5] es una herramienta fundamental para la comunicación y el acceso a recursos académicos y administrativos. La relevancia de este sitio web radica en su papel central como medio de interacción entre la universidad y sus diferentes públicos. Una auditoría eficaz contribuye a mejorar la usabilidad, accesibilidad y seguridad del sitio, asegurando que cumpla con las necesidades y expectativas de todos sus usuarios.

La motivación principal de este proyecto radica en la necesidad de garantizar que el sitio web oficial de la URJC funcione correctamente y ofrezca una experiencia de usuario óptima. Sus usuarios y el público en general dependen de la web para acceder a información académica, recursos y servicios. Una auditoría permite identificar y corregir fallos, mejorar el rendimiento y asegurar que el sitio cumple con los estándares de accesibilidad y seguridad.

1.4 Objetivos

Los objetivos se centran en la realización de una auditoría del sitio web de la URJC mediante diversas pruebas, con el fin de garantizar su funcionalidad, rendimiento y seguridad.

Objetivo Principal (OP)

OP: Realizar una auditoría del sitio web oficial de la URJC mediante pruebas automatizadas para garantizar su funcionalidad, rendimiento y seguridad.

Objetivos Específicos (OE)

OE1: Realizar un estudio sobre la auditoría de sitios web, incluyendo un análisis crítico de las herramientas y metodologías disponibles en la literatura y en la práctica profesional.

OE2: Implementar y ejecutar pruebas de rendimiento, seguridad, accesibilidad y usabilidad en el contexto de la auditoría de sitios web.

OE3: Desarrollar y llevar a cabo un conjunto de pruebas detalladas sobre el sitio web de la URJC, analizando los resultados obtenidos para identificar áreas de mejora.

1.5 Planificación

A continuación, se presenta una descripción de las actividades principales, su secuencia y las duraciones estimadas.

Fases y Actividades del Proyecto

1- Definición de los Objetivos de la Auditoría (1 semana)

- Reuniones iniciales y establecimiento de objetivos.
- Entrega de documentación de requisitos.

2- Selección de Herramientas y Configuración del Entorno (1 semana)

- Selección de herramientas (Gherkin, Behave, PyCharm, Selenium, JMeter, WAVE, etc.).
- Configuración de PyCharm y otros entornos de desarrollo.

3- Desarrollo de Pruebas (4 semanas)

- Desarrollo de Pruebas Simples (1 semana)
 - Creación de pruebas para navegación principal, enlaces a redes sociales, y funcionalidad de búsqueda.
- Desarrollo de Pruebas en la Sección Principal (1 semana)
 - Pruebas de rendimiento, enlaces a recursos PDF, navegación principal y secundaria.
- Desarrollo de Pruebas en Todo el Dominio URJC (2 semanas)
 - Pruebas de conexión con el servidor, rendimiento en todo el dominio, y verificación de enlaces externos y PDFs.

4- Ejecución de Pruebas y Recolección de Datos (2 semanas)

- Ejecución de todas las pruebas definidas.
- Recolección y almacenamiento de los resultados.

5- Análisis de Resultados y Generación de Informes (2 semanas)

- Análisis de los resultados obtenidos de las pruebas.
- Redacción de informes detallados sobre el rendimiento, funcionalidad, y seguridad del sitio web.

6- Corrección de Errores y Validación de Requisitos (2 semanas)

- Identificación y corrección de errores encontrados.

Cronograma:

Duración Total del Proyecto: 12 semanas.

Fase / Actividad		Duración estimada	Inicio	Fin
Definición de los Objetivos de la Auditoría		1 semana	Semana 1	Semana 1
Selección de Herramientas y Configuración		1 semana	Semana 2	Semana 2
		4 semanas	Semana 3	Semana 6
Desarrollo de Pruebas	Pruebas Simples	1 semana	Semana 3	Semana 3
	Pruebas en la Sección Principal	1 semana	Semana 4	Semana 4
	Pruebas en Todo el Dominio URJC	2 semanas	Semana 5	Semana 6
Ejecución de Pruebas y Recolección de Datos		2 semanas	Semana 7	Semana 8
Análisis de Resultados y Generación de Informes		2 semanas	Semana 9	Semana 10
Corrección de Errores y Validación de Requisitos		2 semanas	Semana 11	Semana 12

Tabla 1. Cronograma del proyecto

Dependencias Principales:

- Definición de Objetivos debe completarse antes de la Selección de Herramientas y Configuración del Entorno.
- Desarrollo de Pruebas depende de la Selección de Herramientas y Configuración del Entorno.
- Ejecución de Pruebas no puede comenzar hasta que el Desarrollo de Pruebas esté completo.
- Análisis de Resultados solo puede comenzar después de que todas las pruebas se hayan ejecutado.
- Corrección de Errores y Validación de Requisitos depende del Análisis de Resultados.

Diagrama de Gantt

En la Ilustración 2, se muestra el diagrama de Gantt asociado a este proyecto.

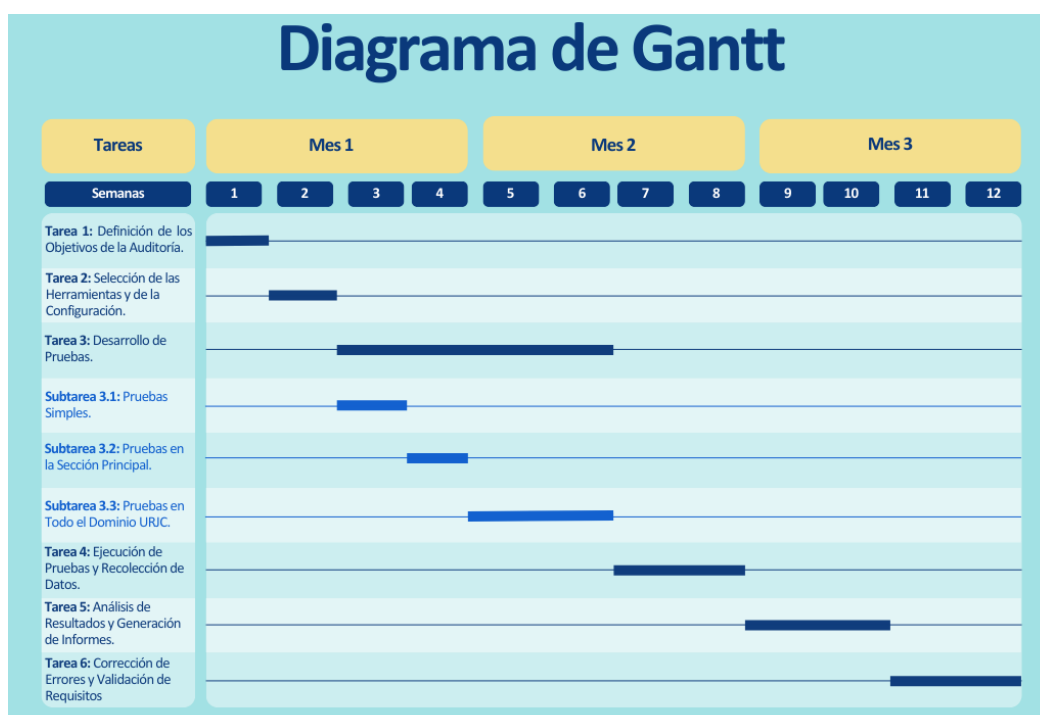


Ilustración 2. Diagrama de Gantt

1.6 Estructura de la memoria

La memoria consta de siete capítulos, los cuales se agrupan en tres secciones principales, además de contar con un anexo, un glosario y una bibliografía final.

Introducción y Marco Teórico (Capítulos I y II)

- **Introducción al problema y planteamiento inicial:** Se presenta una descripción general del problema que aborda el proyecto y se establecen los objetivos principales.
- **Investigaciones pertinentes:** Se revisan estudios y trabajos previos relacionados con la auditoría de sitios web, y se plantean los conceptos principales, los fundamentos teóricos y se plantea el análisis de herramientas, librerías y las metodologías de las que se va a hacer uso durante el proyecto.
- **Metodología:** Se detallan las metodologías utilizadas durante el desarrollo del proyecto.

Planificación y pruebas (Capítulos III, IV)

- **Planificación y desarrollo:** Se describe el proceso de planificación y las etapas de desarrollo de las pruebas automatizadas.
- **Diseño software:** Se presentan el diseño del software de las pruebas automatizadas, y los diagramas.

Validación y Conclusiones. (Capítulos VI, VII)

- **Pruebas con herramientas externas:** Se detallan las pruebas realizadas con herramientas externas como JMeter, WAVE, Hosted Scan, Equal Web y OWASP ZAP.
- **Pruebas funcionales:** Se detallan las pruebas realizadas con Selenium y Gherkin.
- **Validación de objetivos y requisitos, y de resultados:** Se analizan los resultados de las pruebas y se valida el cumplimiento de los objetivos y requisitos establecidos.
- **Conclusiones y propuestas futuras:** Se presentan las conclusiones del proyecto y se proponen futuras mejoras y desarrollos.



Secciones Adicionales

- **Anexo:** Incluye información adicional relevante:
 - El enlace de descarga de los documentos con los resultados de las pruebas automatizadas.
 - El enlace de descarga de los informes y resultados obtenidos mediante las herramientas externas.
 - El diagrama de Gantt en formato ampliado.
 - Los diagramas UML en formato ampliado.
 - El enlace de descarga del código de las pruebas automatizadas.
- **Bibliografía:** Lista de las referencias y fuentes consultadas durante el proyecto.

Marco teórico

2.1 Análisis de conceptos principales

Con el objetivo de desarrollar un marco teórico, se analizan a continuación los conceptos principales, que nos permitirán comprender la evolución de las prácticas y tecnologías utilizadas en el *testing*, las auditorías y la automatización. Este contexto resulta para valorar los avances y la importancia de las herramientas modernas en la auditoría de sitios web [10].

2.1.1 Testing

El *testing* de software (o pruebas de software), ha evolucionado significativamente desde sus inicios. En las primeras etapas del desarrollo de software, las pruebas eran realizadas manualmente, lo que era un proceso laborioso y propenso a errores. Con el tiempo, se introdujeron herramientas de *testing* automatizado que permitieron realizar pruebas de manera más eficiente y precisa [11].

Definición

El *testing* o prueba de software es el proceso de evaluar un sistema o sus componentes con la intención de encontrar si satisface los requisitos especificados o identificar diferencias entre las condiciones actuales y las requeridas. Este proceso puede involucrar la ejecución de un sistema o aplicación con el fin de encontrar errores, validar que el sistema funciona según lo esperado y verificar su calidad [11].

2.1.2 Auditorías

Las auditorías, tradicionalmente asociadas a la contabilidad y la gestión financiera, han ampliado su alcance al ámbito tecnológico con la evolución de los sistemas informáticos. Las auditorías de sistemas y auditorías web han surgido como disciplinas especializadas, enfocándose en evaluar la seguridad, el rendimiento y la funcionalidad de sistemas y aplicaciones web. La auditoría web, en particular, ha ganado relevancia con el crecimiento

exponencial de internet, requiriendo métodos y herramientas avanzadas para asegurar la calidad y la seguridad de los sitios web [12].

Definición

Una auditoría es un proceso sistemático y documentado para obtener evidencias y evaluarlas con el fin de determinar en qué medida se cumplen los criterios establecidos. En el ámbito informático, una auditoría puede centrarse en diferentes aspectos como la seguridad, el rendimiento, la accesibilidad y la funcionalidad de sistemas y aplicaciones [12].

Por su parte, la auditoría web es un tipo específico de auditoría que se enfoca en evaluar sitios web. Este proceso implica la revisión de la estructura del sitio, la validación de enlaces, la verificación de formularios y la evaluación del rendimiento y la seguridad.

2.1.3 Automatización en testing

La automatización ha transformado radicalmente el panorama de las pruebas y las auditorías. Desde simples *scripts* para ejecutar pruebas repetitivas hasta complejos sistemas de automatización de procesos, la tecnología ha permitido una eficiencia sin precedentes. La integración de herramientas de automatización en el ciclo de desarrollo y entrega de software ha permitido realizar pruebas continuas y en tiempo real, reduciendo el tiempo de desarrollo y mejorando la calidad del producto final [13].

Definición

La automatización en el *testing* se refiere a la práctica de utilizar *scripts* y herramientas para ejecutar pruebas de software de manera automática. Esto no solo reduce el tiempo necesario para realizar las pruebas, sino que también aumenta la precisión y consistencia de los resultados. En este proyecto, la automatización permite realizar una auditoría exhaustiva del sitio web de la URJC, cubriendo una amplia gama de pruebas funcionales, de rendimiento y de seguridad sin la intervención manual constante [16].

2.2 Herramientas para definir escenarios de pruebas

Hoy en día, para el proceso de auditoría de sitios webs, existen multitud de herramientas y metodologías clave para evaluar la funcionalidad, rendimiento y seguridad de un sitio. Una de las herramientas fundamentales es Gherkin, un lenguaje específico de dominio utilizado para escribir pruebas en un formato legible por humanos. Gherkin permite definir el comportamiento del software mediante escenarios escritos en un lenguaje natural estructurado, facilitando la colaboración entre equipos técnicos y no técnicos. Este enfoque de especificación basado en texto ayuda a clarificar los requisitos del sistema y permite una automatización eficiente a través de herramientas de pruebas como Behave, que interpreta y ejecuta estas especificaciones para verificar el comportamiento del software de manera consistente y repetible [2].

PyCharm, por otro lado, es un entorno de desarrollo integrado (IDE) que proporciona herramientas avanzadas para la escritura, depuración y ejecución de código Python [1]. En el contexto de la auditoría, PyCharm es utilizado para el desarrollo y mantenimiento de los *scripts* automatizados que interactúan con el sitio web a través de Selenium.

Por su parte, Selenium es una herramienta de automatización de pruebas que permite simular las interacciones de un usuario con el navegador web, como hacer clic en botones, llenar formularios y navegar por diferentes páginas [14]. Esta integración de PyCharm y Selenium asegura que las pruebas automatizadas sean robustas y eficaces, proporcionando resultados precisos sobre la funcionalidad del sitio y detectando problemas de manera eficiente.

Por último, Behave, como *framework* de pruebas de comportamiento en Python, complementa estas herramientas al permitir la ejecución de pruebas basadas en las especificaciones escritas en Gherkin. Esto facilita la implementación de pruebas de aceptación que validan el comportamiento del sistema desde la perspectiva del usuario final. Además de estas herramientas principales, se utilizan otras herramientas especializadas según las necesidades específicas de la auditoría, como herramientas de análisis de rendimiento y seguridad, que contribuyen a una evaluación integral del sitio web [15].

2.3 Otras herramientas

Aparte de las herramientas para definir escenarios de pruebas, en este apartado se describen otras herramientas no mencionadas anteriormente y que también se han usado para llevar a cabo la auditoría del sitio web de la URJC.

Estas herramientas junto con las mencionadas en el apartado anterior permiten la automatización de pruebas y la validación de diferentes aspectos del sitio, asegurando una evaluación completa y precisa.

Gherkin

Gherkin es un lenguaje específico de dominio utilizado para describir el comportamiento de un software de manera estructurada y legible por humanos. Permite la escritura de pruebas en un formato natural, utilizando palabras clave como *Given*, *When* y *Then* para definir el comportamiento esperado del sistema.

Estas especificaciones pueden ser interpretadas automáticamente por herramientas de pruebas como Behave, facilitando la automatización de pruebas y la colaboración entre equipos técnicos y no técnicos. Esto simplifica el proceso de pruebas y asegura que éstas sean comprensibles y mantenibles a lo largo del ciclo de desarrollo del software [16].



Ilustración 3. Gherkin

Google Chrome

Google Chrome es un navegador web desarrollado por Google. Es conocido por su velocidad, simplicidad y seguridad. Chrome incluye herramientas de desarrollo robustas, que permiten a los desarrolladores inspeccionar y depurar código, analizar el rendimiento y diagnosticar problemas en las aplicaciones web. Además, su compatibilidad con diversas extensiones y su

capacidad para ejecutar *scripts* de prueba lo convierten en una herramienta esencial para las pruebas automatizadas [17].



Ilustración 4. Google Chrome

JMeter

JMeter es una herramienta de código abierto diseñada para pruebas de carga y rendimiento. Es ampliamente utilizada para simular múltiples usuarios concurrentes que acceden a una aplicación web, permitiendo medir la capacidad de respuesta y estabilidad bajo diferentes condiciones de carga. JMeter facilita la creación de planes de prueba complejos mediante su interfaz gráfica de usuario, donde se pueden configurar distintos escenarios de prueba con múltiples hilos, controladores y escuchadores para recopilar y analizar datos de rendimiento [18].



Ilustración 5. Apache JMeter

OWASP ZAP

OWASP ZAP (Zed Attack Proxy) es una herramienta de seguridad de aplicaciones web que permite detectar y explotar vulnerabilidades en aplicaciones web. Es parte del proyecto OWASP (Open Web Application Security Project) y se utiliza ampliamente para realizar pruebas de penetración y auditorías de seguridad. ZAP ofrece una interfaz amigable y una serie de herramientas automáticas y manuales que facilitan la identificación de fallos comunes como inyecciones SQL, cross-site scripting (XSS), y otros tipos de vulnerabilidades [19].



Ilustración 6. OWASP ZAP

HostedScan Security

HostedScan Security es una herramienta avanzada en línea especializada en la seguridad de aplicaciones web. Esta plataforma ofrece una gama completa de servicios diseñados para identificar, evaluar y explotar vulnerabilidades presentes en aplicaciones web, ayudando a mejorar su seguridad y robustez.

Para este proyecto, se ha utilizado la funcionalidad de prueba de HostedScan Security con el objetivo de realizar una auditoría de seguridad exhaustiva en el sitio web de la Universidad Rey Juan Carlos (URJC). La herramienta ejecuta un análisis profundo para detectar diversas categorías de vulnerabilidades, incluyendo inyecciones SQL, cross-site scripting (XSS), configuraciones erróneas de seguridad y otros riesgos comunes en el entorno web [20].



Ilustración 7. HostedScan

WAVE:

WAVE es una herramienta de evaluación de accesibilidad web desarrollada por WebAIM (Web Accessibility In Mind). Su objetivo es ayudar a los desarrolladores a hacer que sus sitios web sean más accesibles para personas con discapacidades. WAVE analiza las páginas web y proporciona retroalimentación visual sobre posibles problemas de accesibilidad, destacando elementos que podrían presentar dificultades para los usuarios.

En el contexto de este proyecto, WAVE se utilizó para evaluar la accesibilidad del sitio web de la URJC. La herramienta identificó problemas de accesibilidad según las pautas WCAG 2.1 (Web Content Accessibility Guidelines). Los resultados permitieron realizar ajustes y mejoras en el diseño y la estructura del sitio para garantizar que sea accesible para todos los usuarios, incluyendo aquellos con discapacidades [21].



Ilustración 8. WAVE

Equal Web

Equal Web es una herramienta avanzada de prueba de accesibilidad automatizada diseñada para integrarse en los flujos de trabajo de desarrollo y pruebas automatizadas de aplicaciones web. Desarrollada por expertos en accesibilidad web, Equal Web proporciona una solución integral para garantizar que los sitios web cumplan con los estándares de accesibilidad, tales como WCAG 2.1 y las normativas locales e internacionales [22].

Para este proyecto se ha usado su extensión de Chrome, la cual facilita la evaluación de accesibilidad en cualquier página web. Esta extensión permite ejecutar auditorías de accesibilidad directamente desde el navegador.



Ilustración 9. Equal Web

2.4 Librerías

A continuación, se describen las principales librerías utilizadas en este proyecto, cada una de las cuales aporta funcionalidades específicas que contribuyen a un proceso de prueba robusto y completo.

Behave

Behave es una librería de pruebas basada en el enfoque BDD (Behavior Driven Development), que permite escribir pruebas en un lenguaje natural utilizando el formato Gherkin. Gherkin es un lenguaje legible por humanos que facilita la especificación de requisitos y escenarios de prueba.

Behave ejecuta estas pruebas traduciendo los escenarios escritos en Gherkin a código Python, permitiendo una automatización eficiente de las pruebas de comportamiento. Esta metodología fomenta la colaboración entre desarrolladores, *testers* y otros *stakeholders*, asegurando una comprensión común de los requisitos y pruebas. Behave soporta la integración con *frameworks* de pruebas y herramientas de automatización, lo que permite un flujo de trabajo cohesionado y efectivo en el desarrollo de software.

Selenium

Selenium es una suite (o agrupación) de herramientas para la automatización de navegadores web. Proporciona una Interfaz de Programación de Aplicaciones (API) robusta para simular la interacción de un usuario con un sitio web, permitiendo realizar acciones como clics, relleno de formularios y navegación entre páginas.

Selenium es ideal para pruebas funcionales y de regresión en aplicaciones web, asegurando que las interfaces de usuario funcionen como se espera. Soporta múltiples lenguajes de programación, incluyendo Python, Java y C#, y puede integrarse con *frameworks* de pruebas como Behave para una automatización de pruebas más completa. Selenium también permite la ejecución de pruebas en múltiples navegadores y plataformas, garantizando una cobertura amplia y exhaustiva.

URLjoin

URLjoin es una utilidad proporcionada por la biblioteca estándar de Python en el módulo `urllib.parse`. Esta herramienta se utiliza para combinar de manera segura y correcta diferentes partes de una URL, asegurando que las URLs se construyan correctamente y evitando errores comunes relacionados con la concatenación de cadenas.

URLjoin es esencial en la construcción de pruebas automatizadas que requieren navegar a diferentes rutas dentro de un sitio web. Al manejar adecuadamente la unión de rutas relativas y absolutas, URLjoin garantiza la generación de URLs válidas, lo que es crucial para la correcta navegación y prueba de sitios web complejos.

BeautifulSoup (bs4)

BeautifulSoup es una librería de Python utilizada para el scraping de datos de archivos HTML y XML. Facilita la extracción de información de las páginas web y permite analizar su estructura de manera sencilla y eficiente.

BeautifulSoup transforma los documentos HTML o XML en una estructura de árbol que puede ser navegada y modificada mediante una API simple. En el contexto de las pruebas automatizadas, BeautifulSoup puede utilizarse para validar la estructura del DOM (Document Object Model) de las páginas web, asegurando que los elementos necesarios estén presentes y correctamente formateados. Además, es útil para extraer datos específicos de las páginas y verificar su contenido, contribuyendo a la precisión y exhaustividad de las pruebas de software.

2.5 Metodologías

En este proyecto, se han empleado diversas metodologías para planificar y ejecutar la auditoría del sitio web de la URJC de manera eficiente. Estas metodologías incluyen la automatización de pruebas, la integración continua y el uso de diagramas de Gantt para la gestión del proyecto, cada una de las cuales desempeña un papel crucial en la estructura y éxito del proceso de auditoría.

2.5.1 Automatización

La automatización se ha llevado a cabo mediante la implementación de *scripts* de prueba utilizando Selenium y Behave. Estos *scripts* simulan la interacción del usuario con el sitio web, realizando acciones como la navegación entre páginas, la validación de formularios y la comprobación de enlaces. La automatización permite realizar pruebas repetitivas de manera eficiente y consistente, reduciendo el tiempo y esfuerzo necesarios para la auditoría. Además, la automatización ayuda a identificar errores de forma temprana, minimizando el riesgo de defectos en el producto final. Con Selenium, se pueden ejecutar pruebas en múltiples navegadores y plataformas, asegurando una cobertura exhaustiva. Behave, por su parte, facilita la creación de escenarios de prueba claros y comprensibles mediante el uso del lenguaje Gherkin, lo que mejora la colaboración entre los equipos de desarrollo y pruebas.

2.5.2 Behavior Driven Development

Behavior Driven Development (BDD) es una metodología de desarrollo ágil que mejora la comunicación entre equipos técnicos y no técnicos. Mediante la creación de especificaciones de comportamiento claras y comprensibles, BDD se centra en el comportamiento deseado de una aplicación desde la perspectiva del usuario final, promoviendo un enfoque centrado en el cliente. Gherkin es un lenguaje común utilizado en BDD para describir casos de prueba siguiendo el patrón "Given-When-Then", lo que permite a todos los *stakeholders* comprender y participar en el proceso de definición de requisitos y pruebas. En este proyecto, BDD ha facilitado la creación de escenarios de prueba que reflejan fielmente los requisitos del usuario, asegurando que las pruebas automatizadas validen el comportamiento esperado del sistema de manera integral.

Planificación y desarrollo

3.1 Introducción

Tal como se ha especificado en el capítulo 1, se ha decidido realizar una auditoría exhaustiva del sitio web de la URJC utilizando herramientas de pruebas automatizadas y metodologías ágiles. El enfoque se centra en la evaluación de la funcionalidad, rendimiento y seguridad del sitio, asegurando que cumpla con los estándares actuales y ofrezca una experiencia de usuario óptima.

3.2 Planteamiento inicial

El planteamiento inicial del proyecto consistió en definir los objetivos específicos de la auditoría, seleccionar las herramientas adecuadas y planificar las fases del proyecto. Se realizaron reuniones iniciales para identificar las áreas críticas del sitio web que requerían evaluación y se establecieron criterios para la selección de herramientas y la implementación de pruebas.

3.3 Diagramas de Actividades:

El diagrama de actividades representa el flujo de trabajo para este proyecto de auditoría del sitio web de la URJC, destacando las actividades principales y las decisiones tomadas en el proceso.

Actividades Principales:

1- Definición de los objetivos de la auditoría.

Identificar los objetivos primarios y secundarios de la auditoría.

Alinear los objetivos con los RF y los RNF relacionados con el desarrollo de las pruebas automáticas:

- **RF 2:** Verificar el funcionamiento de los enlaces a redes sociales.
- **RF 3:** Evaluar la efectividad de la funcionalidad de búsqueda.
- **RF 4:** Comprobar la validez y funcionalidad del formulario de acceso de alumnos.
- **RF 5:** Realizar pruebas de carga para la página principal.
- **RF6:** Verificar la estabilidad de la conexión con el servidor.
- **RF 7:** Comprobar la funcionalidad de los enlaces a recursos PDF.
- **RF 8:** Validar los enlaces externos del sitio.
- **RF12:** Verificar que las páginas devuelvan el estado 200.
- **RNF 1:** Los resultados de las pruebas deben almacenarse y ser accesibles para su revisión.
- **RNF 2:** Las pruebas deben ser repetibles y consistentes.
- **RNF 3:** El tiempo de ejecución de las pruebas debe ser optimizado.
- **RNF 4:** Las pruebas deben ser escalables para cubrir futuras expansiones del sitio.

2- Selección de herramientas y configuración del entorno.

Herramientas seleccionadas: Gherkin, Behave, PyCharm, Selenium, Google Chrome, JMeter, OWASP ZAP, HostedScan Security, WAVE, Equal Web, librería URLjoin, librería BeautifulSoup.

Configuración del entorno de desarrollo en PyCharm con las librerías necesarias.

3- Desarrollo e implementación de las pruebas.

Desarrollar *scripts* de prueba utilizando la metodología Gherkin para los diferentes OS específicos: del OS1 al OS12.

4- Ejecución de pruebas y recolección de datos.

Ejecutar las pruebas desarrolladas y recolectar los resultados de cada prueba, asegurando el almacenamiento y accesibilidad de los datos (RNF 1).

5- Análisis de resultados y generación de informes.

Analizar los datos recolectados para identificar problemas de funcionalidad, rendimiento y seguridad, y generar informes detallados sobre el estado del sitio web, destacando los problemas encontrados y las áreas que cumplen con los requisitos.

6- Corrección de errores y validación de requisitos.

Identificar y documentar los errores encontrados durante la auditoría.

Validar que las correcciones implementadas cumplen con los requisitos funcionales (RF) y no funcionales (RNF).

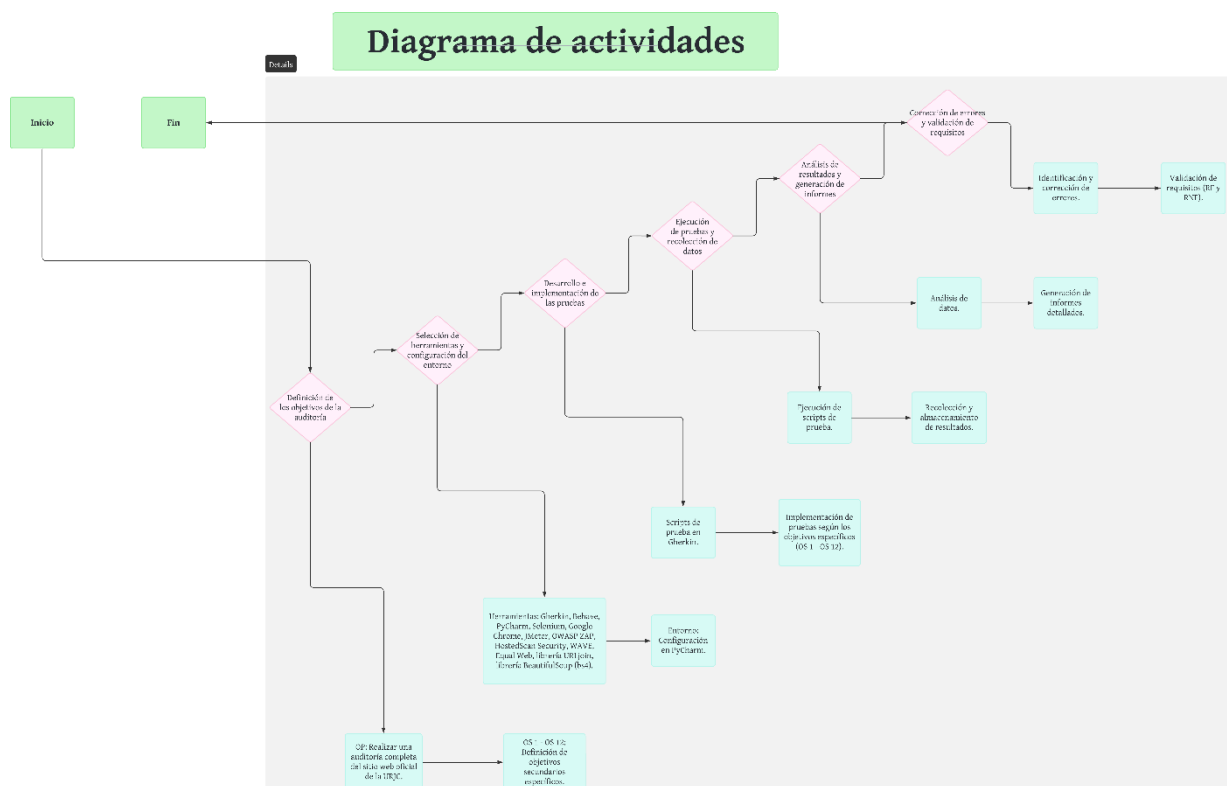


Ilustración 10. Diagrama de Actividades

El diagrama de actividades ilustra un proceso sistemático y estructurado para llevar a cabo la auditoría del sitio web, asegurando que todos los objetivos y requisitos sean abordados de manera efectiva.

3.4 Uso de Gherkin para las pruebas

Gherkin se ha utilizado como la herramienta principal para la automatización de pruebas en este proyecto. Esta herramienta permite escribir pruebas en un formato legible por humanos utilizando el lenguaje Gherkin, y luego ejecutarlas automáticamente integrando con Jenkins para la integración continua. Las pruebas implementadas con Gherkin cubren diversos aspectos del sitio web, incluyendo la navegación, validación de formularios y enlaces, y evaluación del rendimiento.

3.5 Desarrollo

Esta sección describe detalladamente el proceso de desarrollo de las pruebas automatizadas, desde la codificación inicial de los *steps* hasta su integración y verificación. Se explican las metodologías y herramientas utilizadas, destacando las mejoras aplicadas para optimizar el rendimiento y la efectividad de las pruebas.

Codificación de las *Features*

La codificación de las *features* se realizó utilizando el lenguaje Gherkin para definir escenarios de prueba legibles y comprensibles.

Para el desarrollo de las *features*, estas se clasificaron en tres grupos:

- Pruebas simples.
- Pruebas de la sección principal.
- Pruebas en todo el dominio de la URJC

Se comenzó con las pruebas simples, donde la verificación inicial de los pasos de cada *feature* se realizó manualmente debido a su menor complejidad.

Posteriormente, se procedió con las pruebas de la sección principal, también iniciando con una verificación manual.

Finalmente, para las pruebas en todo el dominio de la URJC, no fue necesaria una verificación manual previa, ya que en las secciones anteriores se habían realizado pruebas similares a las definidas para este grupo.

Codificación de los steps

La codificación de los *steps* se llevó a cabo utilizando Python, siguiendo las directrices de Gherkin para definir escenarios de prueba legibles. Cada *step* corresponde a una acción específica que el usuario realiza en el sitio web, como hacer clic en un enlace, llenar un formulario o buscar información. Se implementaron funciones Python que interactúan con Selenium WebDriver para simular estas acciones en el navegador, asegurando que las pruebas reflejen con precisión el comportamiento del usuario.

Integración de los steps con las features

Los *steps* codificados se integraron con las *features* definidas en Gherkin, que describen los escenarios de prueba de manera estructurada. Cada *feature* contiene uno o más escenarios, y cada escenario consiste en una serie de *steps*. Esta integración se gestionó mediante Behave, que enlaza los *steps* en Python con las definiciones en Gherkin. Esto permitió una ejecución fluida y coherente de las pruebas, asegurando que cada escenario se probara de manera exhaustiva.

Verificación de los test

La verificación de los test implicó la ejecución repetida de los escenarios de prueba para asegurar su precisión y fiabilidad. Se implementó un *sistema de reporting* para capturar los resultados de cada ejecución, identificando fallos y áreas de mejora. Los logs de Selenium y las capturas de pantalla automáticas ayudaron a diagnosticar problemas y ajustar los scripts de prueba según fuera necesario. Esta fase garantizó que los test fueran robustos y capaces de detectar inconsistencias en el comportamiento del sitio web.



Mejoras aplicadas

A lo largo del proceso de desarrollo, se aplicaron diversas mejoras para optimizar los scripts de prueba y el entorno de ejecución. Esto incluyó la refactorización del código para mejorar la legibilidad y mantenibilidad. Además, se añadieron validaciones adicionales para verificar la accesibilidad y rendimiento del sitio, asegurando una cobertura completa de los aspectos críticos de la auditoría.

Diseño de pruebas automáticas

4.1 Requisitos

Los requisitos del proyecto se dividen en funcionales y no funcionales, asegurando que se cubran todas las necesidades y expectativas.

4.1.1 Funcionales

- **RF 1:** Validar la accesibilidad de las secciones principales del sitio web.
- **RF 2:** Verificar el funcionamiento de los enlaces a redes sociales.
- **RF 3:** Evaluar la efectividad de la funcionalidad de búsqueda.
- **RF 4:** Comprobar la validez y funcionalidad del formulario de acceso de alumnos.
- **RF 5:** Realizar pruebas de carga para la página principal.
- **RF6:** Verificar la estabilidad de la conexión con el servidor.
- **RF 7:** Comprobar la funcionalidad de los enlaces a recursos PDF.
- **RF 8:** Validar los enlaces externos del sitio.
- **RF 9:** Realizar pruebas de carga en todo el dominio urjc.es.
- **RF 10:** Realizar pruebas de rendimiento en todo el dominio urjc.es.
- **RF 11:** Validar la accesibilidad de las secciones principales del sitio web.
- **RF12:** Verificar que las páginas devuelvan el estado 200.

- **RF13:** Validar la usabilidad de las secciones principales del sitio web.

4.1.2 No Funcionales

- **RNF 1:** Los resultados de las pruebas deben almacenarse y ser accesibles para su revisión.
- **RNF 2:** Las pruebas deben ser repetibles y consistentes.
- **RNF 3:** El tiempo de ejecución de las pruebas debe ser optimizado.
- **RNF 4:** Las pruebas deben ser escalables para cubrir futuras expansiones del sitio.

4.2 Casos de prueba

En la fase de desarrollo y auditoría de software, los casos de prueba y los diagramas UML (Unified Modeling Language ó Lenguaje de Modelado Unificado) son herramientas fundamentales que permiten planificar, ejecutar y documentar de manera eficiente el proceso de pruebas. Los casos de prueba aseguran que todas las funcionalidades del sistema se verifiquen exhaustivamente, mientras que los diagramas UML proporcionan una representación visual clara de la estructura y comportamiento del sistema.

Importancia de los Casos de Prueba

Los casos de prueba son esenciales para:

- **Verificar la funcionalidad del sistema:** Garantizan que cada componente del sitio web funcione correctamente según los requisitos especificados.
- **Identificar defectos:** Ayudan a descubrir errores y problemas antes de que el sistema se despliegue en producción.
- **Mejorar la calidad del software:** Aseguran que el producto final cumpla con los estándares de calidad y las expectativas del usuario.
- **Facilitar la repetibilidad y consistencia:** Permiten ejecutar las mismas pruebas de manera repetitiva para verificar que los cambios no introduzcan nuevos defectos.

4.3 Diagramas UML de Casos de Uso

Los diagramas UML son herramientas visuales que facilitan la comprensión y comunicación del diseño del sistema.

El diagrama de casos de uso muestra los diferentes casos de uso del sistema y los actores involucrados, proporcionando una visión clara de cómo interactúan con el sistema.

Casos de Uso:

Este conjunto de casos de uso cubre todas las actividades y responsabilidades esenciales para llevar a cabo una auditoría completa y exhaustiva del sitio web de la URJC.

Actores:

Administrador del Sistema: Configura el entorno y supervisa la ejecución de las pruebas.

Auditor: Define y ejecuta las pruebas, analiza los resultados.

Usuario Final: Interactúa con el sitio web y proporciona retroalimentación.

Configurar Entorno de Pruebas:

Descripción: Establecer las herramientas y configuraciones necesarias para la auditoría.

Actor: Administrador del Sistema.

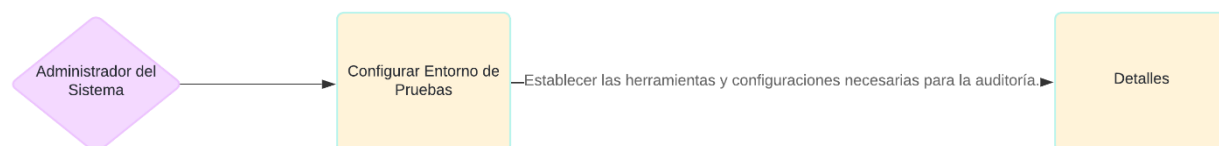


Ilustración 11. CU 1 Configurar Entorno de Pruebas

Definir Pruebas:

Descripción: Crear pruebas específicas para la auditoría utilizando el lenguaje Gherkin.

Actor: Auditor.

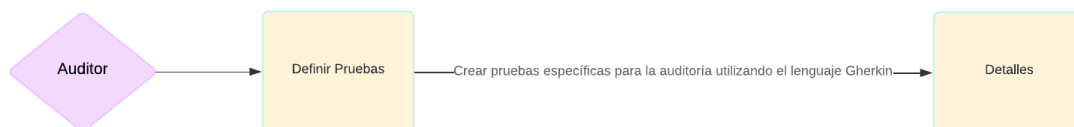


Ilustración 12. CU 2 Definir Pruebas

Ejecutar Pruebas:

Descripción: Ejecutar las pruebas definidas utilizando herramientas automatizadas.

Actor: Auditor.

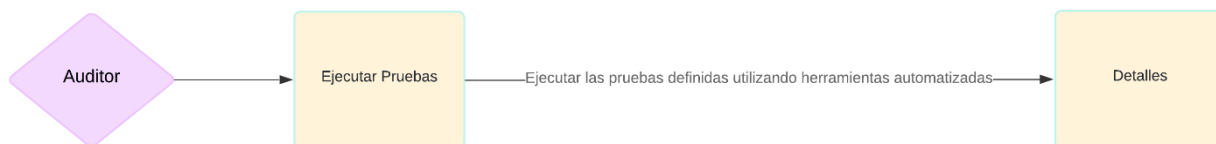


Ilustración 13. CU 3 Ejecutar Pruebas

Analizar Resultados:

Descripción: Revisar y analizar los resultados de las pruebas para identificar problemas.

Actor: Auditor.

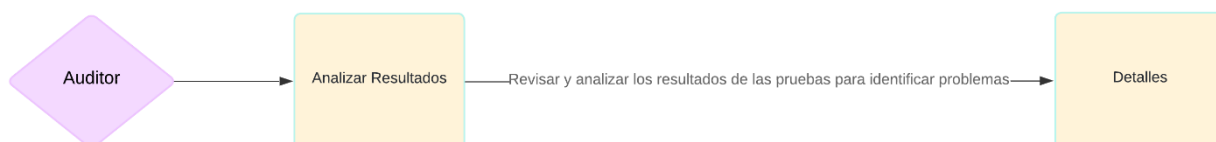


Ilustración 14. CU 4 Analizar Resultados

Generar Reportes:

Descripción: Crear informes detallados basados en los resultados de las pruebas.

Actor: Auditor.

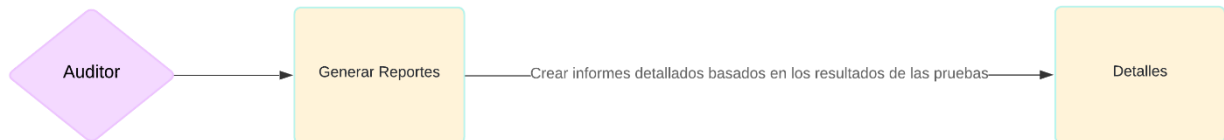


Ilustración 15. CU 5 Generar Reportes

Corregir Errores:

Descripción: Identificar y corregir los errores encontrados durante las pruebas.

Actor: Auditor, Administrador del Sistema.

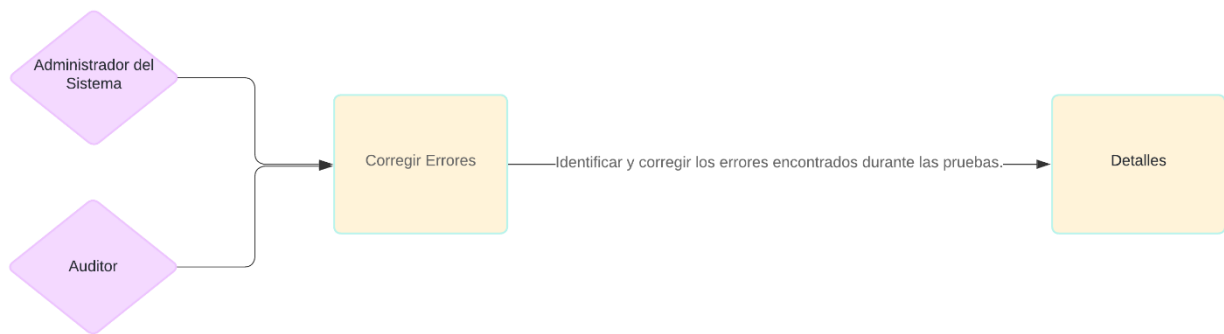


Ilustración 16. CU 6 Corregir Errores

Validar Requisitos:

Descripción: Asegurar que todas las pruebas cumplen con los requisitos establecidos.

Actor: Auditor.

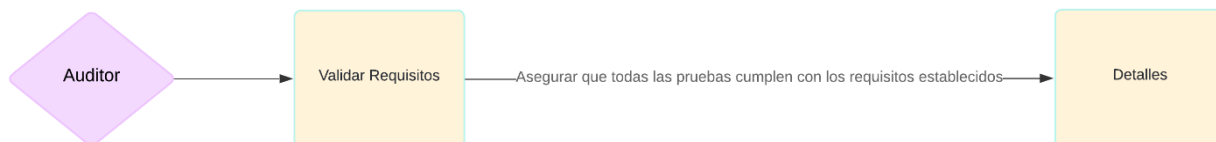


Ilustración 17. CU 7 Validar Requisitos

Monitorear Rendimiento:

Descripción: Supervisar el rendimiento del sitio web durante las pruebas.

Actor: Administrador del Sistema.



Ilustración 18. CU 8 Monitorear Rendimiento

Proporcionar Retroalimentación:

Descripción: Interactuar con el sitio web y proporcionar comentarios sobre su funcionamiento.

Actor: Usuario Final.



Ilustración 19. CU 9 Proporcionar Retroalimentación

Actualizar Configuraciones:

Descripción: Modificar configuraciones del entorno de pruebas según sea necesario.

Actor: Administrador del Sistema.



Ilustración 20. CU 10 Actualizar Configuraciones

Gestionar Incidencias:

Descripción: Registrar y gestionar las incidencias detectadas durante las pruebas.

Actor: Auditor.

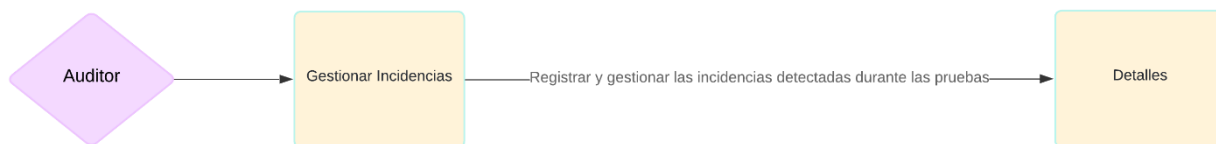


Ilustración 21. CU 11 Gestionar Incidencias

Realizar Pruebas de Seguridad:

Descripción: Ejecutar pruebas específicas para evaluar la seguridad del sitio web.

Actor: Auditor.

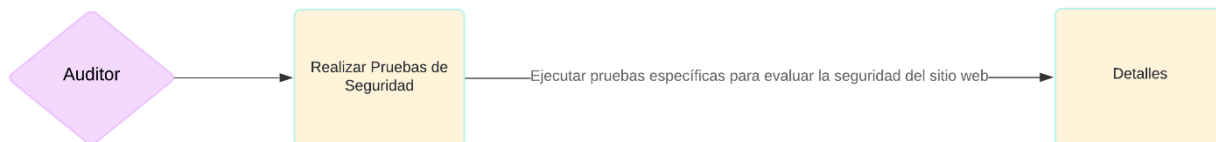


Ilustración 22. CU 12 Realizar Pruebas de Seguridad

Realizar Pruebas de Usabilidad:

Descripción: Ejecutar pruebas para evaluar la accesibilidad y usabilidad del sitio web.

Actor: Auditor.

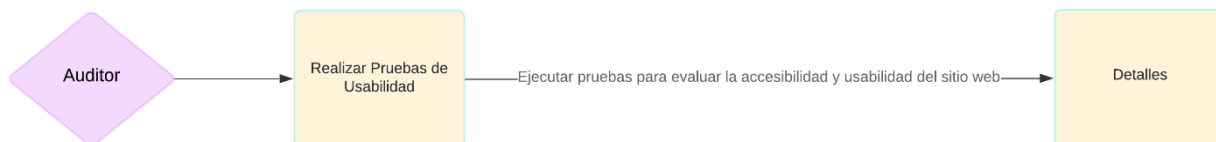


Ilustración 23. CU 13 Realizar Pruebas de Usabilidad

Revisar Cumplimiento de Normativas:

Descripción: Asegurar que el sitio web cumple con las normativas vigentes de accesibilidad y seguridad.

Actor: Auditor.



Ilustración 24. CU 14 Revisar Cumplimiento de Normativas

Automatizar Pruebas:

Descripción: Implementar *scripts* para la automatización de pruebas recurrentes.

Actor: Auditor.

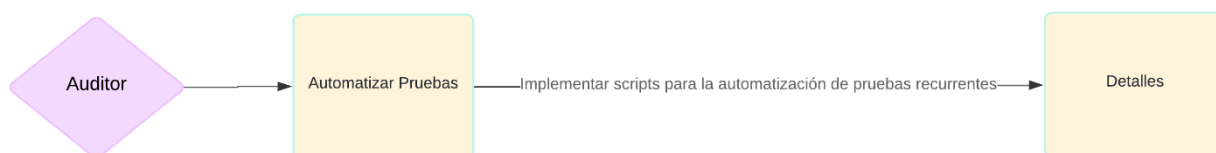


Ilustración 25. CU 15 Automatizar Pruebas

Integrar Herramientas de Prueba:

Descripción: Integrar diferentes herramientas para una ejecución de pruebas más eficiente.

Actor: Administrador del Sistema.

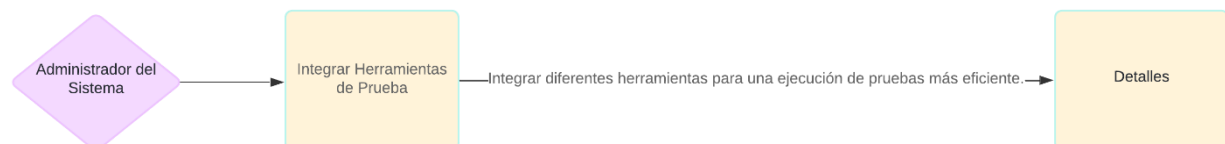


Ilustración 26. CU 16 Integrar Herramientas de Prueba

4.4 Implementación de las pruebas

En esta sección, se detalla la naturaleza y alcance de las pruebas desarrolladas, describiendo los diferentes tipos de pruebas implementadas y su propósito específico. El objetivo es proporcionar una visión integral de cómo estas pruebas contribuyen a la evaluación de la funcionalidad, rendimiento y seguridad del sitio web de la Universidad Rey Juan Carlos (URJC).

4.4.1 Pruebas con Selenium

Descripción

Cada prueba cuya extensión es *.feature* se compone de una secuencia de steps o pasos que el sistema va a seguir, definidos mediante la metodología Gherkin (Given – When – Then). Antes de cada *step* se indica el tipo de paso que es con una de las siguientes etiquetas:

- *Given* (Dado): Define los antecedentes de la prueba. Se utiliza para establecer el contexto inicial del test. En código Python, se utiliza la etiqueta `@given('texto del step')`.
- *When* (Cuando): Define los pasos del test. Se utiliza para describir las acciones que se realizan durante la prueba. En código Python, se utiliza la etiqueta `@when('texto del step')`.

- **Then** (Entonces): Define los pasos que finalizan la prueba. Se utiliza para describir los resultados esperados después de realizar las acciones. En código Python, se utiliza la etiqueta `@then('texto del step')`.
- **And** (Y): Define los pasos cuya etiqueta coincide con la indicada anteriormente (*Given*, *When* o *Then*). En código Python, se utiliza la etiqueta que corresponde al tipo de paso `@given('texto del step')`, `@when('texto del step')` o `@then('texto del step')`.

Al inicio de cada test se indica que feature es con la etiqueta **Feature**: seguida del título de la feature. Y antes de los primeros pasos del test se antepone una línea con la etiqueta **Background**: indicando el inicio del test.

En los test donde se dan varios escenarios se dividen los diferentes escenarios anteponiendo a los *steps* una línea con la etiqueta **Scenario**: para indicarlo, lo que hará que se ejecute un test por cada escenario.

Pruebas funcionales con Selenium

Las pruebas funcionales se centraron en verificar que los componentes del sitio web de la URJC cumpla con los requisitos establecidos y funcionen correctamente. Estas pruebas aseguran que las funcionalidades específicas del sitio web de la URJC operen sin errores, proporcionando una experiencia de usuario fluida y confiable.

- **Herramienta Utilizada**: Selenium
- **Metodología**: Se desarrollaron scripts automatizados para simular interacciones de usuario, como clics, entradas de datos en formularios, navegación entre páginas y verificación de resultados esperados.
- **Casos de Prueba**: Basados en los requisitos funcionales del sitio web, documentando las acciones a realizar y los resultados esperados para cada caso.

Features

- [urjc_test_information_help_check.feature](#)

Esta prueba automatizada navega a la sección de ayuda y comprueba el formulario de inicio de sesión de los alumnos, asegurando que el formulario funcione correctamente y que la navegación a esta sección sea fluida.
- [urjc_test_inquiries_section.feature](#)

Esta prueba verifica la navegación a la sección de consultas y la existencia de los elementos esperados en esa página, garantizando que los usuarios puedan acceder y visualizar correctamente la información.
- [urjc_test_load_homepage.feature](#)

Esta prueba verifica que la página principal del sitio web se cargue sin errores, asegurando que todos los elementos se carguen correctamente y que la página esté completamente funcional al acceder a ella.
- [urjc_test_main_menu_items.feature](#)

Esta prueba verifica la existencia y funcionalidad de los elementos del menú principal, asegurando que los enlaces y secciones del menú estén correctamente configurados y sean accesibles.
- [urjc_test_menu_links_on_homepage.feature](#)

Esta prueba verifica los enlaces del menú en la página principal, asegurando que cada enlace redirija correctamente a la sección correspondiente sin errores.
- [urjc_test_search_different_texts.feature](#)

Esta prueba automatizada navega por el sitio y utiliza el motor de búsqueda con diferentes textos, verificando que los resultados de búsqueda sean relevantes y correctos.
- [urjc_test_texts_header_and_cookies.feature](#)

Esta prueba verifica los textos en el encabezado y del popUp de las cookies, asegurando que la información mostrada sea correcta.
- [urjc_test_title_homepage.feature](#)

Esta prueba verifica el título de la página principal, asegurando que el título del sitio web sea el esperado.

Ejemplo de funcionamiento: urjc_test_information_help_check.feature

Esta prueba se centra en verificar la funcionalidad de la sección de ayuda en el sitio web de la URJC. A través de varios escenarios, se asegura que diferentes secciones y funcionalidades dentro del sitio estén operativas y respondan correctamente a las acciones del usuario.

```

urjc_test_information_help_check.feature x
1  Feature: Test URJC Help Section
2
3  Background:
4    Given I initialize the test results file and the browser
5    Given I open the URJC homepage
6    When I click on the menu item "CONTACTO"
7    And I check the text "SOLICITUD DE INFORMACIÓN"
8    And I click on the item "//*[@id='k2Container']/div[2]/div[1]/div[1]/div[1]/div/div[2]/a"
9    And I go to the new opened tab on the browser
10   And The url in the browser should be "https://soporte.urjc.es/servicedesk/customer/user/login?destination=portals"
11
12  Scenario: Test help section for future students
13    When I click on the menu item "Futuros estudiantes"
14    And I enter "contraseña" in the smart search field
15    And I should see the number of search results containing "contraseña"
16    And I click on the item "//*[@id='header']/div/nav/div[2]/ul/li[2]/a/div/div/img"
17    And I click on the menu item "Log out"
18    Then I close the browser
19    And I save the test results
20
  
```

Ilustración 27. urjc_test_information_help_check.feature 1

```

urjc_test_information_help_check.feature x
20
21  Scenario: Test help section for hosts
22    When I click on the menu item "Estudiante URJC"
23    When I click on the link button "Acceso para invitados"
24    And I enter "contraseña" in the smart search field
25    And I should see the number of search results containing "contraseña"
26    And I click on the item "//*[@id='header']/div/nav/div[2]/ul/li[2]/a/div/div/img"
27    And I click on the menu item "Log out"
28    Then I close the browser
29    And I save the test results
30
31  Scenario: Test help section for students
32    When I click on the menu item "Estudiante URJC"
33    #Aquí se deberá poner un nombre de usuario válido
34    When I enter "user" in the user name field
35    #Aquí se deberá poner la contraseña asociada al nombre de usuario
36    And I enter "psssword" in the user password field
37    And I enter "contraseña" in the smart search field
38    And I should see the number of search results containing "contraseña"
39    And I click on the item "//*[@id='header']/div/nav/div[2]/ul/li[2]/a/div/div/img"
40    And I click on the menu item "Cerrar sesión"
41    Then I close the browser
42    And I save the test results
  
```

Ilustración 28. urjc_test_information_help_check.feature 2

Background: Parte común a todos los escenarios.

- Inicializa un archivo para registrar los resultados de las pruebas y abre el navegador.
- Abre la página principal del sitio web de la URJC.

- Hace clic en el ítem del menú etiquetado como "CONTACTO".
- Verifica que el texto "SOLICITUD DE INFORMACIÓN" esté presente en la página.
- Hace clic en un elemento específico de la página utilizando su XPath.
- Cambia a la nueva pestaña abierta en el navegador.
- Verifica que la URL actual en el navegador sea la especificada.

Scenario: Test help section for future students Escenario de la sección de futuros estudiantes.

- Hace clic en el ítem del menú "Futuros estudiantes".
- Introduce la palabra "contraseña" en el campo de búsqueda inteligente.
- Verifica la cantidad de resultados de búsqueda que contienen la palabra "contraseña".
- Hace clic en un elemento específico utilizando su XPath.
- Hace clic en el ítem del menú "Log out".
- Cierra el navegador.
- Guarda los resultados de la prueba.

Scenario: Test help section for hosts Escenario de la sección de invitados

- Hace clic en el ítem del menú "Estudiante URJC".
- Hace clic en el botón de enlace "Acceso para invitados".
- Introduce la palabra "contraseña" en el campo de búsqueda inteligente.
- Verifica la cantidad de resultados de búsqueda que contienen la palabra "contraseña".
- Hace clic en un elemento específico utilizando su XPath.
- Hace clic en el ítem del menú "Log out".
- Cierra el navegador.
- Guarda los resultados de la prueba.

Scenario: Test help section for students Escenario de la sección de ayuda a estudiantes

- Hace clic en el ítem del menú "Estudiante URJC".
- Introduce un nombre de usuario válido en el campo correspondiente.
- Introduce la contraseña asociada al nombre de usuario en el campo correspondiente.
- Introduce la palabra "contraseña" en el campo de búsqueda inteligente.

- Verifica la cantidad de resultados de búsqueda que contienen la palabra "contraseña".
- Hace clic en un elemento específico utilizando su XPath.
- Hace clic en el ítem del menú "Cerrar sesión".
- Cierra el navegador.
- Guarda los resultados de la prueba.

En las imágenes 29 y 30 se puede ver la parte de la lógica de alguno de los pasos:

```
@When('I check the text "{text}"')
def step_impl(context, text):
    try:
        xpath_text_expression = f'//*[contains(text(), "{text}")]'
        name_elements = WebDriverWait(context.driver, timeout=10).until(
            EC.presence_of_all_elements_located((By.XPATH, xpath_text_expression))
        )
        assert name_elements, save_test_results(context, result=f'No se encontraron elementos '
            f'con el texto \"{text}\".\n')
        save_test_results(context, result=f'Se encontró al menos un elemento con el texto \"{text}\".\n')
    except TimeoutException:
        assert False, save_test_results(context,
            result=f'No se encontraron elementos con el texto \"{text}\" '
            f'dentro del tiempo de espera.\n')

@When('I click on the item "{xpath}"')
def click_item_with_xpath(context, xpath):
    menu_item = WebDriverWait(context.driver, timeout=10).until(
        EC.element_to_be_clickable((By.XPATH, xpath)))
    menu_item.click()
    save_test_results(context, result=f'Se ha hecho clic en ítem indicado correctamente.\n')

@When('I go to the new opened tab on the browser')
def go_to_new_tab(context):
    context.driver.switch_to.window(context.driver.window_handles[1])
    save_test_results(context, result=f'Se ha cambiado de pestaña.\n')
```

Ilustración 29. Steps urjc_test_information_help_check.feature 2

```
@When('I should see the number of search results containing "{search_text}"')
def verify_search_results(context, search_text):
    try:
        xpath_text_expression = f'//*[contains(text(), "{search_text}")]'
        name_elements = WebDriverWait(context.driver, timeout=10).until(
            EC.presence_of_all_elements_located((By.XPATH, xpath_text_expression)))
        n = len(name_elements)
        if n == 1:
            save_test_results(context, result=f'El texto \"{search_text}\" ha aparecido {n} vez en '
                f'la página.\n')
        else:
            save_test_results(context, result=f'El texto \"{search_text}\" ha aparecido {n} veces en '
                f'la página.\n')
    except TimeoutException:
        assert False, save_test_results(context, result=f'No se encontraron resultados para la '
            f'búsqueda \"{search_text}\".\n')

@When('I enter "{search_text}" in the user name field')
def enter_search_query(context, search_text):
    selector_expression = "#os_username"
    search_input = WebDriverWait(context.driver, timeout=30).until(
        EC.presence_of_element_located((By.CSS_SELECTOR, selector_expression)))
    search_input.clear()
    search_input.send_keys(search_text)
    search_input.submit()
    save_test_results(context, result=f'Nombre de usuario enviado correctamente.\n')
```

Ilustración 30. Steps urjc_test_information_help_check.feature 2

Pruebas de rendimiento con Selenium

Las pruebas de rendimiento se centraron en evaluar en el sitio web de la URJC que las distintas páginas carguen sin errores, y el tiempo de respuesta de los enlaces. Estas pruebas son esenciales para garantizar que el sitio web funcione eficientemente.

- **Herramienta Utilizada:** Selenium
- **Metodología:** Se desarrollaron scripts automatizados para medir el rendimiento del sitio web en términos de tiempos de carga y disponibilidad de recursos, simulando la interacción del usuario con el sitio web.

Features

- [deep_urjc_test_check_links.feature](#)

Esta prueba verifica que todos los enlaces en el sitio web se carguen sin errores, asegurando que los enlaces funcionen correctamente y que no haya enlaces rotos.

- [deep_urjc_test_http_status_codes.feature](#)

Esta prueba verifica que los enlaces devuelvan un código HTTP 200, asegurando que las páginas y recursos vinculados estén disponibles y accesibles.

- [deep_urjc_test_loads_time.feature](#)

Esta prueba mide el tiempo de carga de los enlaces en el sitio web, asegurando que los tiempos de carga sean aceptables y no afecten negativamente la experiencia del usuario.

- [urjc_test_check_pdfs.feature](#)

Esta prueba verifica que los enlaces a PDFs devuelvan un código HTTP 200, asegurando que los documentos PDF estén disponibles y se pueda acceder a ellos correctamente.

- [urjc_test_links_check.feature](#)

Esta prueba verifica que los enlaces de la página principal del sitio web devuelvan un código HTTP 200, asegurando la disponibilidad y accesibilidad de los recursos vinculados.

- [urjc_test_verify_rrss.feature](#)

Esta prueba verifica la existencia y funcionalidad de los enlaces a redes sociales, asegurando que los enlaces redirijan correctamente a las respectivas plataformas de redes sociales y que devuelvan un código HTTP 200.

Ejemplo de funcionamiento: `deep_urjc_test_check_links.feature`

Esta prueba se enfoca en verificar que todos los enlaces, tanto internos como externos, así como los enlaces a archivos PDF en el dominio de la URJC, se carguen sin errores.

```
deep_urjc_test_check_links.feature x
1 >> Feature: Deep Test for all links should load without errors
2
3   Background:
4     Given I initialize the test results file and the browser
5
6 >> Scenario: Deep test all internal links load without errors
7     When I check all internal links in URJC domain should load without errors
8     Then I save the test results
9
10 >> Scenario: Deep test all external links load without errors
11     When I check all external links in URJC domain should load without errors
12     Then I save the test results
13
14 >> Scenario: Deep test all PDF links load without errors
15     When I check all PDFs in URJC domain should load without errors
16     Then I save the test results
```

Ilustración 31. `deep_urjc_test_check_links.feature`

Background: Parte común a todos los escenarios.

- Inicializa un archivo para registrar los resultados de las pruebas y abre el navegador.

Scenario: Deep test all internal links load without errors Escenario de los links que pertenecen a la URJC.

- Verifica que todos los enlaces internos dentro del dominio de la URJC se carguen sin errores.
- Guarda los resultados de la prueba.

Scenario: Deep test all external links load without errors Escenario de las links externos.

- Verifica que todos los enlaces externos en el dominio de la URJC se carguen sin errores.
- Guarda los resultados de la prueba.

Scenario: Deep test all PDF links load without errors Escenario de las links que finalizan por “.pdf”

- Verifica que todos los enlaces a archivos PDF dentro del dominio de la URJC se carguen sin errores.
- Guarda los resultados de la prueba.

En las imágenes 32 y 33 se puede ver la parte de la lógica de cada uno de los pasos:

```

def initialize_test_results_file(context):
    scenario_name = context.scenario.name
    scenario_valid_filename = re.sub(r'[\<>:"/\\|?]*', '_', scenario_name)
    file_path = f'Results {scenario_valid_filename}.txt'
    if os.path.exists(file_path):
        os.remove(file_path)
    open(file_path, 'a').close()

@given('I initialize the test results file and the browser')
def init_text_results(context):
    context.test_results = []
    initialize_test_results_file(context)

@when("I check all internal links in URJC domain should load without errors")
def step_verify_all_internal_links_load(context):
    links = obtain_links_deep("https://www.urjc.es")
    for link in links:
        if 'urjc.es' in link and '.pdf' not in link:
            #Corrijo un link mal puesto suprimiendo la indicación de formato
            if link.endswith("format=feed"):
                link = link[:-len("format=feed")]
            print(f"    {link}")
            verify_url_load_without_errors(context, link)
            step_save_test_results(context)
            close_browser(context)
  
```

Ilustración 32. Steps deep_urjc_test_check_links.feature 1

```

@when("I check all external links in URJC domain should load without errors")
def step_verify_all_external_links_load(context):
    links = obtain_links_deep("https://www.urjc.es")
    for link in links:
        if 'urjc.es' not in link and '.pdf' not in link:
            print(f"    {link}\n")
            verify_url_load_without_errors(context, link)
            step_save_test_results(context)
            close_browser(context)

@when("I check all PDFs in URJC domain should load without errors")
def step_verify_all_internal_links_load(context):
    links = obtain_links_deep("https://www.urjc.es")
    for link in links:
        if '.pdf' in link:
            print(f"    {link}")
            verify_url_load_without_errors(context, link)
            step_save_test_results(context)
            close_browser(context)

3 usages
@then('I save the test results')
def step_save_test_results(context):
    write_test_results_to_file(context)
  
```

Ilustración 33. Steps deep_urjc_test_check_links.feature 2

Resumen

Se han realizado pruebas funcionales y de rendimiento utilizando Selenium para asegurar que el sitio web funcione correctamente bajo diferentes condiciones. Las pruebas funcionales se enfocaron en verificar que los componentes del sitio cumplan con los requisitos establecidos y proporcionen una experiencia de usuario fluida.

Las pruebas de rendimiento evaluaron la capacidad del sitio para manejar enlaces y recursos, asegurando que los tiempos de carga sean óptimos y que todos los enlaces funcionen correctamente.

4.4.2 Pruebas de Rendimiento

Las pruebas de rendimiento evaluaron cómo el sitio web responde bajo diferentes condiciones de carga. Estas pruebas son cruciales para identificar cuellos de botella y garantizar que el sitio pueda manejar múltiples usuarios concurrentes sin degradar su desempeño.

- **Herramienta Utilizada:** JMeter
- **Metodología:** Simulación de usuarios concurrentes (10, 100, y 1000 usuarios) para medir tiempos de respuesta, estabilidad y capacidad de manejo de carga.
- **Parámetros Medidos:** Tiempo de respuesta promedio, tasa de éxito, rendimiento bajo carga y estabilidad del sitio.

Las pruebas de rendimiento con JMeter han permitido evaluar la capacidad del sitio web para manejar diferentes niveles de carga, asegurando que se mantenga funcional y eficiente incluso bajo condiciones de alta demanda. Los resultados han sido fundamentales para identificar áreas de mejora en la infraestructura y configuración del sitio.

4.4.3 Pruebas de Seguridad

Las pruebas de seguridad se enfocaron en identificar y mitigar vulnerabilidades que podrían ser explotadas por atacantes. Estas pruebas aseguran que el sitio esté protegido contra amenazas comunes y cumpla con los estándares de seguridad.

- **Herramientas Utilizadas:** HostedScan y OWASP ZAP
- **Metodología:** Escaneos de vulnerabilidades en servidores, redes, sitios web y aplicaciones seleccionadas, buscando fallos como inyecciones SQL, Cross-Site Scripting (XSS), y Cross-Site Request Forgery (CSRF).

Las pruebas de seguridad han identificado diversas vulnerabilidades en el sitio web, permitiendo priorizar su corrección según el nivel de riesgo. Las recomendaciones proporcionadas mejorarán significativamente la seguridad del sitio, protegiendo la información y la integridad del sistema contra posibles ataques.

4.4.4 Pruebas de Accesibilidad

Las pruebas de accesibilidad garantizaron que el sitio web sea usable para todos los usuarios, incluyendo aquellos con discapacidades, cumpliendo con las pautas WCAG 2.0 AA.

- **Herramientas Utilizadas:** WAVE y Equal Web
- **Metodología:** Evaluación de la conformidad del sitio con los estándares de accesibilidad, identificando y analizando problemas de accesibilidad presentes.

Las pruebas de accesibilidad han permitido identificar áreas donde el sitio web puede mejorar su usabilidad para usuarios con discapacidades. Las recomendaciones resultantes ayudarán a cumplir con las pautas WCAG 2.0 AA, asegurando una experiencia inclusiva para todos los usuarios.

4.4.5 Pruebas de Usabilidad

Las pruebas de usabilidad se realizaron para asegurar que el sitio web sea fácil de usar y proporcione una experiencia de usuario positiva.

- **Herramienta Utilizada:** Equal Web
- **Metodología:** Evaluación de la facilidad de uso del sitio web, incluyendo la capacidad de redimensionar texto, saltar bloques de contenido repetido, y otros aspectos relacionados con la interacción del usuario.



Las pruebas de usabilidad han destacado aspectos clave donde la experiencia del usuario puede ser mejorada. Las sugerencias derivadas de estas pruebas contribuirán a un sitio web más intuitivo y fácil de navegar, mejorando la satisfacción general del usuario.

4.4.6 Resumen

La implementación de estas pruebas ha permitido una evaluación de la funcionalidad, rendimiento y seguridad del sitio web de la URJC. Cada tipo de prueba ha aportado información valiosa sobre diferentes aspectos del sitio, asegurando que cumpla con los estándares y proporcionando una base sólida para futuras mejoras y optimizaciones.

Validación

5.1 Introducción

El objetivo de este capítulo se centra en validar las pruebas desarrolladas, evaluando su efectividad en cumplir con los objetivos del proyecto. Se analizan los resultados obtenidos y se discuten las mejoras implementadas a partir de los hallazgos de las pruebas, proporcionando una visión detallada del proceso de corrección de errores y optimización del sitio web de la URJC.

A parte de las pruebas automáticas definidas con Gherkin, se han llevado a cabo exhaustivas pruebas con el objetivo de identificar posibles errores y asegurar la estabilidad y funcionamiento óptimo de la página web de la URJC. Se han abordado diversas áreas, destacando los aspectos más comunes que suelen ser propensos a problemas en las páginas web. Se han utilizado métricas específicas para validar los resultados de las pruebas, tales como el tiempo de respuesta, tasa de éxito, rendimiento bajo carga, y cumplimiento de los estándares de accesibilidad

5.2 Corrección de Errores

Durante la fase de pruebas, se identificaron y corrigieron varios errores comunes que afectaban la funcionalidad y seguridad del sitio web. Las correcciones se centraron en los siguientes aspectos clave:

Captura de Excepciones

- **Implementación de Mecanismos de Manejo de Excepciones:** Se integraron técnicas robustas para capturar y manejar excepciones de manera adecuada. Esto incluye la implementación de bloques try-catch en el código para asegurar que los errores inesperados sean gestionados sin interrumpir la experiencia del usuario. Además, se registran detalladamente las excepciones en archivos de log para facilitar el diagnóstico y resolución de problemas.

- **Manejo de Excepciones Específicas:** Se mejoró el manejo de excepciones específicas, como errores de red, errores de base de datos, y fallos de API, proporcionando respuestas de error claras y amigables para el usuario final.

Optimización de Código

- **Refactorización de Código:** Se llevó a cabo una revisión y refactorización del código para mejorar la eficiencia y la mantenibilidad. Esto incluyó la eliminación de redundancias, la optimización de algoritmos y la mejora de la estructura del código.

5.3 Validación de requisitos

Todos los requisitos que se han planteado para el proyecto han sido cumplidos:

5.3.1 Requisitos Funcionales (RF)

- **RF 1:** Validar la accesibilidad de las secciones principales del sitio web.

Validación: Se realizaron pruebas de accesibilidad con WAVE, confirmando el cumplimiento de las WCAG 2.1.

- **RF 2:** Verificar el funcionamiento de los enlaces a redes sociales.

Validación: Se comprobó que todos los enlaces a redes sociales funcionan correctamente mediante pruebas automatizadas con Selenium.

- **RF 3:** Evaluar la efectividad de la funcionalidad de búsqueda.

Validación: Las pruebas confirmaron que la búsqueda produce resultados relevantes.

- **RF 4:** Comprobar la validez y funcionalidad del formulario de acceso de alumnos.

Validación: El formulario de acceso de alumnos fue validado y su funcionalidad comprobada.

- **RF 5:** Realizar pruebas de carga para la página principal.

Validación: JMeter fue utilizado para simular tráfico elevado.

- **RF6:** Verificar la estabilidad de la conexión con el servidor.

Validación: Las pruebas de estrés confirmaron la estabilidad de la conexión bajo condiciones de alta carga.

- **RF 7:** Comprobar la funcionalidad de los enlaces a recursos PDF.

Validación: Se realizaron pruebas automatizadas para verificar los enlaces a PDFs.

- **RF 8:** Validar los enlaces externos del sitio.

Validación: Se realizaron pruebas automatizadas para verificar la validez de todos los enlaces externos.

- **RF 9:** Realizar pruebas de carga en todo el dominio urjc.es.

Validación: Las pruebas de carga con JMeter en el dominio completo confirmaron su mala capacidad para manejar múltiples usuarios concurrentes.

- **RF 10:** Realizar pruebas de rendimiento en todo el dominio urjc.es.

Validación: Se utilizó JMeter para las pruebas de rendimiento, comprobando que el dominio urjc.es tiene un rendimiento aceptable para una carga de usuarios moderada.

- **RF 11:** Validar la accesibilidad de las secciones principales del sitio web.

Validación: Se realizaron pruebas de accesibilidad con WAVE y confirmando el cumplimiento de las WCAG 2.1.

- **RF12:** Verificar que las páginas devuelvan el estado 200.

Validación: Las pruebas automatizadas desarrolladas para la página principal de la URJC, además de como en todos los enlaces del dominio urjc.es, confirmaron los códigos de respuesta de cada enlace encontrado y evaluado.

- **RF13:** Validar la usabilidad de las secciones principales del sitio web.

Validación: Se realizaron pruebas de usabilidad con Equal Web y confirmando el cumplimiento de las WCAG 2.0 AA.

5.3.2 Requisitos no Funcionales (RNF)

- **RNF 1:** Los resultados de las pruebas deben almacenarse y ser accesibles para su revisión.

Validación: En las pruebas automatizadas, cada step genera un archivo con los resultados para su posterior consulta. Estos archivos, junto con los informes obtenidos tras las pruebas con herramientas externas, pueden ser consultados en el anexo.

- **RNF 2:** Las pruebas deben ser repetibles y consistentes.

Validación: Las pruebas automatizadas con Selenium son repetibles y produjeron resultados consistentes. Además, las pruebas realizadas con las herramientas externas también son repetibles, devolviendo resultados consistentes.

- **RNF 3:** El tiempo de ejecución de las pruebas debe ser optimizado.

Validación: Se optimizó el tiempo de ejecución de las pruebas automatizadas.

- **RNF 4:** Las pruebas deben ser escalables para cubrir futuras expansiones del sitio.

Validación: La arquitectura de pruebas es escalable y puede adaptarse a futuras expansiones del sitio.

5.4 Revisión de pruebas

A parte de las pruebas automáticas definidas con Gherkin, se han llevado a cabo exhaustivas pruebas con el objetivo de identificar posibles errores y asegurar la estabilidad y funcionamiento óptimo de la página web de la URJC. Se han abordado diversas áreas, destacando los aspectos más comunes que suelen ser propensos a problemas en las páginas web. Se han utilizado métricas específicas para validar los resultados de las pruebas, tales como el tiempo de respuesta, tasa de éxito, rendimiento bajo carga, y cumplimiento de los estándares de accesibilidad

5.4.1 Pruebas de rendimiento. JMeter

Las pruebas de rendimiento se realizaron para evaluar cómo el sitio web responde bajo diferentes condiciones de carga. Para estas pruebas, se utilizó JMeter, una herramienta de pruebas de rendimiento que permite simular múltiples usuarios concurrentes y medir diversos parámetros de rendimiento.

Métricas Evaluadas:

- Tiempo de respuesta
- Tasa de transferencia de datos
- Número de transacciones por segundo
- Tasa de error bajo carga

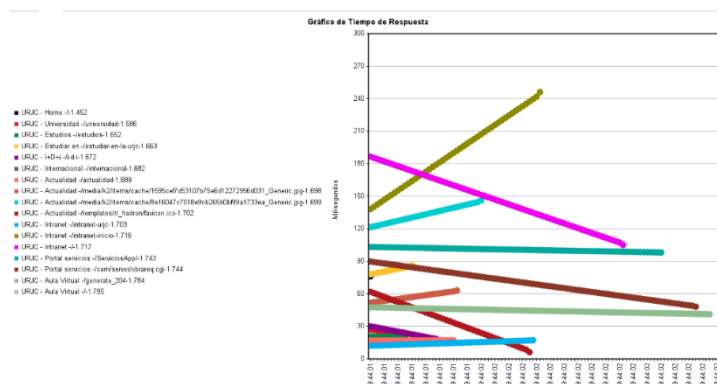


Ilustración 34. Gráfico de Tiempo de respuesta JMeter 1 usuario

Análisis

Se llevaron a cabo tres series de pruebas simulando 10, 100 y 1000 usuarios simultáneos para medir el tiempo de respuesta, el rendimiento y la estabilidad del sitio. A continuación, se presentan los resultados y el análisis de estas pruebas, proporcionando una visión clara de cómo el sitio maneja distintas cargas de usuarios.

Resultados con 10 Usuarios

Etiqueta	Mues	Med	Me	90%Line	95%Line	99%Line	Min	Max	%Error	Rend	Kb/s	Sent KB/sec
Home: -/-1.452	10	95	93	131	131	154	56	154	0,00%	9,8/sec	1577,05	6,71
Universidad: -/universidad-1.585	10	34	23	44	44	94	18	94	0,00%	10,2/sec	1147,91	7,04
Estudios: -/estudios-1.652	10	35	31	45	45	62	21	62	0,00%	10,0/sec	1271,20	7,00
Estudiar en: -/estudiar-en-la-urjc-1.663	10	116	111	158	158	190	78	190	0,00%	9,3/sec	5540,56	6,57
I+D+i: -/i-d-i-1.672	10	27	22	45	45	48	16	48	0,00%	9,9/sec	1093,29	6,96
Internacional: -/Internacional-1.682	10	28	26	34	34	44	17	44	0,00%	10,1/sec	1246,86	7,06
Actualidad: -/actualidad-1.689	10	26	24	39	39	44	15	44	0,00%	10,1/sec	1049,33	7,08
Intranet: Intranet-urjc-1.703	10	32	26	53	53	60	17	60	0,00%	10,0/sec	1049,52	7,05

Intranet: -/intranet-inicio- 1.716	10	243	242	277	277	323	211	323	0,00%	8,4/sec	95,03	24,03
Intranet: -/1.717	10	115	108	132	132	139	100	139	0,00%	9,4/sec	97,69	20,15
Portal de servicios: -/ServiciosApp/- 1.743	10	120	102	158	158	175	90	175	0,00%	9,7/sec	200,58	26,95
Portal de servicios: -/oam/server /obrareq.cgi-1.744	10	62	49	84	84	96	46	96	0,00%	10,7/sec	200,71	22,66
Aula virtual: -/generate_204- 1.784	10	16	16	21	21	31	12	31	0,00%	11,8/sec	1,46	3,57
Aula virtual: -/1.785	10	193	185	213	216	217	174	217	0,00%	9,9/sec	530,00	13,64
Total	140	82	48	190	217	277	12	323	0,00%	71,0/sec	7855,02	87,10

Tabla 2. JMeter resultados 10 usuarios

Leyenda: Muestra: Mues | Media: Med. | Mediana: Me. | Rendimiento: Rend.

- **Media de Respuesta Total: 82 ms**
- **Máximo Tiempo de Respuesta: 323 ms**
- **Porcentaje de Errores: 0%**
- **Rendimiento Total: 70.96 transacciones por segundo**
- **Tasa de Transferencia: 7855.02 Kb/sec**

Con 10 usuarios simultáneos, el sitio web de la URJC muestra tiempos de respuesta rápidos y consistentes, con una media de 82 ms. No se registraron errores, indicando una estabilidad y capacidad de manejo de la carga en este nivel. La página "Intranet - Inicio" presentó el mayor tiempo de respuesta con 323 ms, pero sigue siendo aceptable.

Resultados con 100 Usuarios

Etiqueta	Mues	Med	Me	90%Line	95%Line	99%Line	Min	Max	%Error	Rend	Kb/s	Sent KB/sec
Home: -/-1.452	100	538	424	1012	1090	1950	58	2563	0,000%	29,55956 /sec	4736,28	20,15
Universidad: -/universidad- 1.585	100	307	287	535	585	1811	22	2064	0,000%	25,36140 /sec	2864,80	17,56
Estudios: -/estudios-1.652	100	355	390	554	590	1410	25	1503	0,000%	22,48707 /sec	2858,41	15,75
Estudiar en: -/estudiar-en-la- urjc-1.663	100	1346	1412	1584	2380	2999	228	5254	0,000%	17,95977 /sec	10717,32	12,72
I+D+i: -/i-d-i-1.672	100	243	244	347	403	438	9	443	1,000%	18,74063 /sec	1953,73	13,08
Internacional: -/internacional- 1.682	100	284	233	326	350	460	46	5225	0,000%	18,84304 /sec	2326,14	13,18
Actualidad: -/actualidad-1.689	100	215	223	301	322	424	0	435	1,000%	19,08761 /sec	1971,40	13,31
Intranet: Intranet-urjc- 1.703	100	204	204	307	325	443	18	1162	0,000%	19,50078 /sec	2044,90	13,73
Intranet: -/intranet-inicio- 1.716	100	1228	1188	1704	1846	2019	286	2045	0,000%	16,38538 /sec	185,52	46,88
Intranet: -/-1.717	100	245	221	388	405	423	123	425	0,000%	20,04008 /sec	209,22	43,17

Portal de servicios: -/ServiciosApp/-1.743	100	333	169	964	1144	1388	92	2122	0,000%	20,92050 /sec	433,01	58,12
Portal de servicios: -/oam/server /obrareq.cgi-1.744	100	113	85	153	315	439	49	458	0,000%	28,96871 /sec	541,72	61,17
Aula virtual: -/generate_204-1.784	100	36	28	54	59	157	14	215	0,000%	33,10162 /sec	4,11	10,02
Aula virtual: -/1.785	100	265	260	310	326	364	185	536	0,000%	32,90556 /sec	1764,76	45,41
Total	1400	408	244	1180	1429	1975	0	5254	0,143%	187,4916 /sec 3	20728,54	229,93

Tabla 3. JMeter resultados 100 usuarios

Leyenda: Muestra: Mues | Media: Med. | Mediana: Me. | Rendimiento: Rend.

- **Media de Respuesta Total:** 408 ms
- **Máximo Tiempo de Respuesta:** 5254 ms
- **Porcentaje de Errores:** 0.143%
- **Rendimiento Total:** 187.49 transacciones por segundo
- **Tasa de Transferencia:** 20728.54 Kb/sec

Con 100 usuarios simultáneos, el tiempo de respuesta medio aumentó a 408 ms. Aunque el rendimiento general sigue siendo bueno, hay un incremento notable en los tiempos de respuesta máximos, especialmente en las páginas "Estudiar en la URJC" y "Intranet - Inicio". Se observó un pequeño porcentaje de errores (0.143%), indicando que la capacidad del sitio empieza a ser desafiada a este nivel de carga.



Resultados con 1000 Usuarios

Etiqueta	Mues	Med	Me	90%Line	95%Line	99%Line	Min	Max	%Error	Rend	Kb/s	Sent KB/sec
Home: -/-1.452	1000	8761	8129	15198	17401	22705	151	29847	1,100%	32,37 /sec	5146,35	21,82
Universidad: -/universidad-1.585	1000	1408	4	5051	7423	11305	0	23798	60,600%	32,10 /sec	1468,92	8,76
Estudios: -/estudios-1.652	1000	5355	4109	10917	13702	18866	0	32041 1	10,400%	3,05 /sec	348,79	1,92
Estudiar en: -/estudiar-en-la-urjc-1.663	1000	9931	11187	20487	22861	25556	0	29058	32,300%	3,05 /sec	1237,44	1,47
I+D+i: -/i-d-i-1.672	1000	1889	3	6544	8760	17549	0	22382	64,900%	3,10 /sec	118,91	0,77
Internacional: -/internacional-1.682	1000	5113	3065	15100	21035	21049	0	21067	18,400%	3,11 /sec	315,70	1,78
Actualidad: -/actualidad-1.689	1000	2575	831	6220	15082	21041	0	21053	15,300%	3,12 /sec	277,50	1,87
Intranet: Intranet-urjc-1.703	1000	1821	562	4643	7063	21038	0	21052	7,300%	3,15 /sec	307,04	2,06
Intranet: -/intranet-inicio-1.716	1000	10797	6385	23905	24409	24800	0	29922	37,900%	3,19 /sec	28,13	7,17
Intranet: -/-1.717	1000	7292	764	21045	22171	28174	0	36169	25,100%	3,22 /sec	27,78	5,31

Portal de servicios: -/ServiciosApp/-1.743	1000	12015	11185	22069	24217	31222	87	39173	36,200%	3,23 /sec	48,21	6,16
Portal de servicios: -/oam/server/obrareq.cgi-1.744	1000	6449	1091	21036	21046	24099	43	36125	18,300%	3,24 /sec	52,30	5,71
Aula virtual: -/generate_204-1.784	1000	3897	61	15057	21033	21045	10	21063	8,800%	3,25 /sec	1,10	0,90
Aula virtual: -/1.785	1000	2836	260	15187	21029	21045	172	21053	6,200%	3,25 /sec	164,44	4,22
Total	14000	5724	2023	21026	21053	24487	0	320411	24,486%	42,62 /sec	3496,53	40,34

Tabla 4. JMeter resultados 1000 usuarios

Leyenda: Muestra: Mues | Media: Med. | Mediana: Me. | Rendimiento: Rend.

- **Media de Respuesta Total:** 5724 ms
- **Máximo Tiempo de Respuesta:** 320411 ms
- **Porcentaje de Errores:** 24.486%
- **Rendimiento Total:** 42.62 transacciones por segundo
- **Tasa de Transferencia:** 3496.53 Kb/sec

Con 1000 usuarios simultáneos, el sitio web muestra una significativa degradación en el rendimiento. La media de respuesta aumentó dramáticamente a 5724 ms y el máximo tiempo de respuesta alcanzó valores extremadamente altos (320411 ms). El porcentaje de errores también se incrementó notablemente a 24.486%, indicando que el sitio no puede manejar esta cantidad de usuarios simultáneos sin experimentar problemas serios de rendimiento y estabilidad.

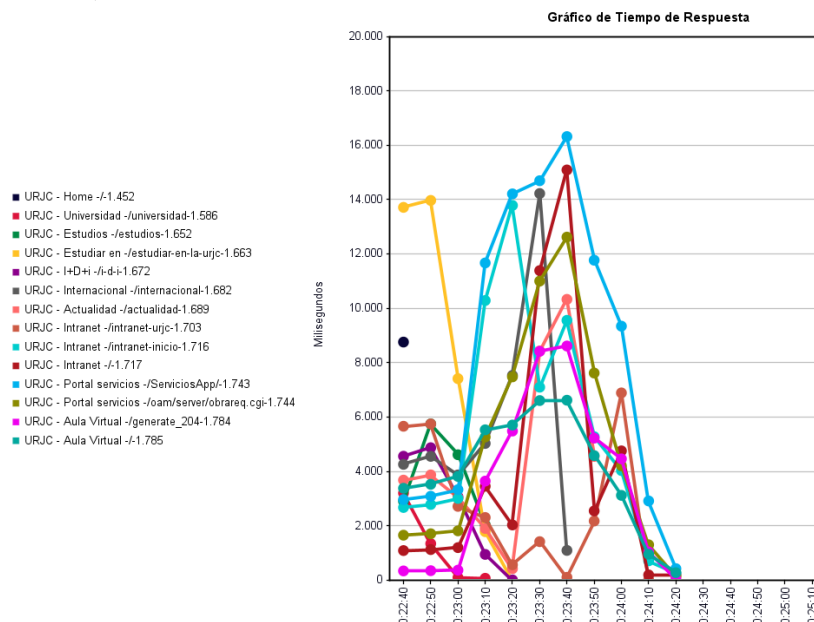


Ilustración 35. Gráfico de Tiempo de respuesta JMeter 1000 usuarios

Análisis General

Los resultados de las pruebas de carga muestran que el sitio web de la URJC maneja adecuadamente cargas bajas (10 usuarios) y moderadas (100 usuarios) con tiempos de respuesta rápidos y mínimos errores. Sin embargo, al aumentar la carga a 1000 usuarios, el rendimiento y la estabilidad del sitio se deterioran significativamente, con tiempos de respuesta excesivos y una alta tasa de errores. Estos resultados sugieren la necesidad de optimizar el rendimiento del sitio y considerar mejoras en la infraestructura para manejar cargas altas de manera más eficiente.

5.4.2 Pruebas de Seguridad. HostedScan

Las pruebas de seguridad se enfocaron en identificar vulnerabilidades que podrían ser explotadas por atacantes. Se utilizó HostedScan, una herramienta de análisis de seguridad que permite detectar los fallos más comunes en el ámbito de la ciberseguridad.

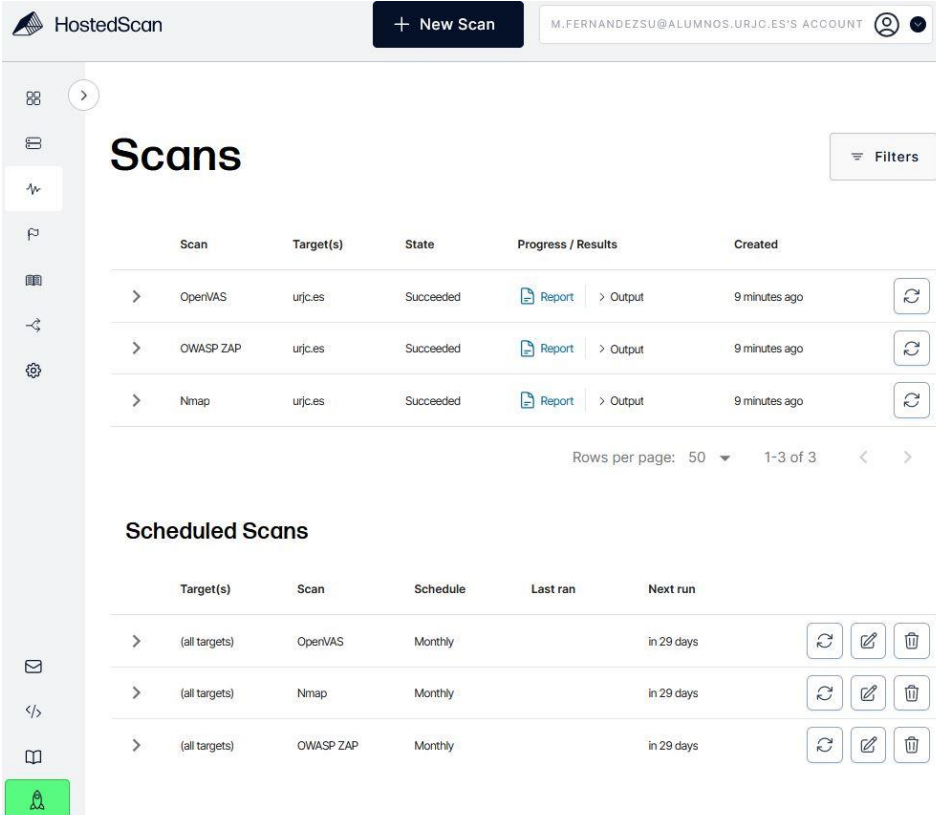
Métricas Evaluadas:

- Número de vulnerabilidades detectadas
- Gravedad de las vulnerabilidades

- Tiempo de respuesta a las amenazas identificadas

Análisis

Se realizaron escaneos de vulnerabilidades en servidores, redes, sitios web y aplicaciones seleccionadas. Este informe contiene las vulnerabilidades potenciales descubiertas en dichos escaneos, clasificadas por su gravedad. Una mayor gravedad indica un mayor riesgo de una violación de datos, pérdida de integridad o disponibilidad de los objetivos.



The screenshot shows the HostedScan interface. At the top, there is a navigation bar with the HostedScan logo, a '+ New Scan' button, and a user account dropdown for 'M.FERNANDEZSU@ALUMNOS.URJO.ES'S ACCOUNT'. Below the navigation bar is a sidebar with various icons for navigation. The main content area is titled 'Scans' and features a table of completed scans. Below this table is a 'Scheduled Scans' section with another table listing recurring scans.

Scan	Target(s)	State	Progress / Results	Created
>	OpenVAS	urjc.es	Succeeded	Report Output 9 minutes ago
>	OWASP ZAP	urjc.es	Succeeded	Report Output 9 minutes ago
>	Nmap	urjc.es	Succeeded	Report Output 9 minutes ago

Target(s)	Scan	Schedule	Last ran	Next run
>	(all targets)	OpenVAS	Monthly	in 29 days
>	(all targets)	Nmap	Monthly	in 29 days
>	(all targets)	OWASP ZAP	Monthly	in 29 days

Ilustración 36. Hosted Scan - Exploraciones

Vulnerabilidades Totales

A continuación, se presenta el número total de vulnerabilidades encontradas, clasificadas por gravedad. Las vulnerabilidades críticas son las más graves y deben evaluarse primero. Una vulnerabilidad aceptada es aquella que ha sido revisada manualmente y clasificada como aceptable para no ser corregida en este momento, como una detección de falso positivo o una parte intencional de la arquitectura del sistema.

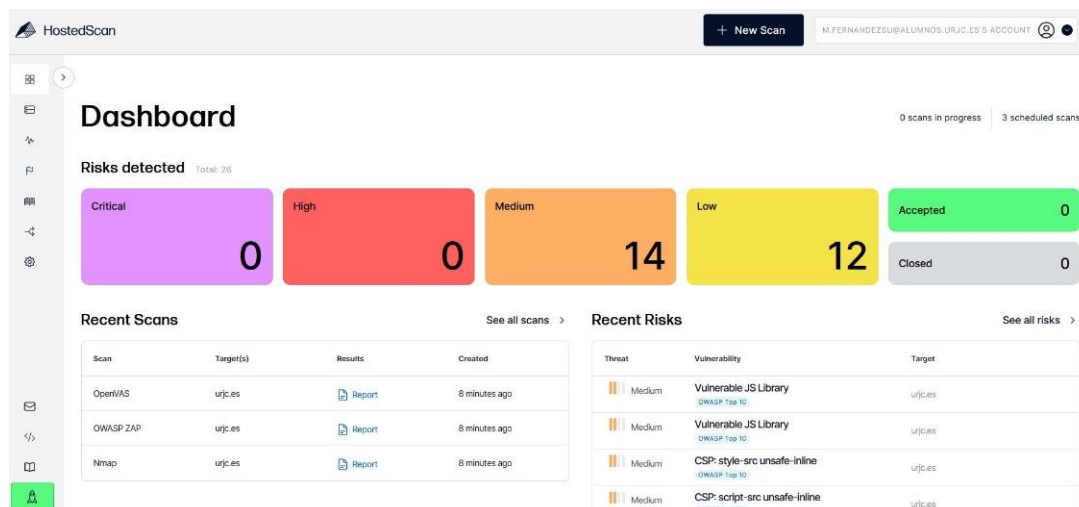


Ilustración 37. Hosted Scan - Resultados

Vulnerabilidades en Aplicaciones Web Pasivas

El escaneo pasivo de OWASP ZAP recorre las páginas de un sitio web o aplicación web, inspeccionando cada página, así como las solicitudes y respuestas enviadas entre el servidor. Este escaneo pasivo busca vulnerabilidades como configuraciones erróneas de dominios cruzados, *cookies* inseguras, dependencias de JavaScript vulnerables, entre otras.

Críticas	Altas	Medias	Bajas	Aceptadas
0	0	14	12	0

Tabla 5. HostedScan - Vulnerabilidades en Ap. Web Pasivas

Vulnerabilidades en Aplicaciones Web Activas: No se detectaron vulnerabilidades.

Vulnerabilidades de Red: No se detectaron vulnerabilidades.

Puertos TCP Abiertos: No se detectaron vulnerabilidades.

Puertos UDP Abiertos: No se detectaron vulnerabilidades.

En el anexo se incluye un enlace para la descarga del informe propio de la herramienta.

5.4.3 Pruebas de Seguridad con OWASP ZAP

Para garantizar la seguridad del sitio web de la Universidad Rey Juan Carlos (URJC), se realizaron pruebas de seguridad utilizando OWASP ZAP (Zed Attack Proxy), una herramienta líder en la detección de vulnerabilidades en aplicaciones web. OWASP ZAP permite identificar y analizar posibles fallos de seguridad, ayudando a proteger el sitio contra amenazas como inyecciones SQL, Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF). En este apartado, se presentan los resultados obtenidos, clasificando las alertas por nivel de riesgo y proporcionando recomendaciones para mitigar las vulnerabilidades detectadas.

Análisis

OWASP ZAP Scan Report

Target: <https://www.urjc.es/>

All scanned sites: <https://identifica.urjc.es> <https://consent.cookiefirst.com> <https://cied.urjc.es> <https://urjcx.urjc.es> <https://cide.urjc.es> <http://miportal.urjc.es> <http://www.cuesa.urjc.es> <http://www.cetinia.urjc.es> <http://www.urjc.es> <https://gestion.urjc.es> <https://miportal.urjc.es> <https://infotic.urjc.es> <https://tv.urjc.es> <https://eventos.urjc.es> <http://transparencia.urjc.es> <http://online.urjc.es> <https://ayuda.urjc.es> <https://sede.urjc.es> <http://ayuda.urjc.es> <https://gestion3.urjc.es> <https://online.urjc.es> <https://eulist.urjc.es> <https://gestion2.urjc.es> <https://en.urjc.es> <https://www.urjc.es>

Javascript included from: <https://identifica.urjc.es> <https://chat.1millionbot.com> <https://www.gstatic.com> <https://cdnjs.cloudflare.com> <https://www.google-analytics.com> <https://consent.cookiefirst.com> <https://cied.urjc.es> <https://urjcx.urjc.es> <https://cide.urjc.es> <http://miportal.urjc.es> <http://www.cuesa.urjc.es> <http://www.cetinia.urjc.es> <http://www.urjc.es> <https://gestion.urjc.es> <https://miportal.urjc.es> <https://infotic.urjc.es> <https://tv.urjc.es> <https://eventos.urjc.es> <http://transparencia.urjc.es> <http://online.urjc.es> <https://ayuda.urjc.es> <https://sede.urjc.es> <http://ayuda.urjc.es> <https://gestion3.urjc.es> <https://online.urjc.es> <https://eulist.urjc.es> <https://gestion2.urjc.es> <https://en.urjc.es> <https://www.urjc.es>

Generated on Thu, 27 Jun 2024 17:44:21

ZAP Version: 2.15.0

Ilustración 38. Análisis Owasp Zap

Resumen de Alertas

Nivel de riesgo	Alto	Medio	Bajo	Informativo
Nº de alertas	0	10	11	5

Tabla 6. Alertas de OWASP ZAP

Alertas Medias Principales

- **Ausencia de Tokens Anti-CSRF:** No se encontraron tokens Anti-CSRF en un formulario HTML, lo que expone al sitio a ataques de Cross-Site Request Forgery.

- **Directiva CSP con Comodines:** La Política de Seguridad de Contenidos (CSP) usa comodines, lo que debilita la protección contra ataques de inyección de datos y Cross-Site Scripting (XSS).
- **CSP: script-src unsafe-inline:** El uso de 'unsafe-inline' en la directiva script-src de CSP permite la ejecución de *scripts* potencialmente inseguros.
- **CSP: style-src unsafe-inline:** Similar al anterior, pero para estilos en línea.
- **Encabezado CSP no Configurado:** La falta de un encabezado CSP adecuado incrementa la vulnerabilidad a ataques de XSS e inyección de datos.
- **Configuración Incorrecta de CORS:** Una mala configuración de CORS puede permitir la carga de datos del navegador, exponiendo información potencialmente sensible.
- **Falta de Encabezado Anti-Clickjacking:** La ausencia de los encabezados necesarios para proteger contra ataques de *Clickjacking*.
- **Contenido Mixto en Páginas Seguras:** Inclusión de contenido accesible a través de HTTP en lugar de HTTPS, comprometiendo la seguridad.
- **Configuración Incorrecta de X-Frame-Options:** El encabezado X-Frame-Options está presente pero mal configurado, lo que puede facilitar ataques de *Clickjacking*.

Alertas Bajas Principales

- **Gran Redirección Detectada:** Respuestas del servidor con redirecciones grandes pueden indicar exposición de información sensible.
- **Cookies sin HttpOnly Flag:** *Cookies* accesibles mediante JavaScript, facilitando potenciales ataques de secuestro de sesiones.
- **Cookies sin Secure Flag:** *Cookies* accesibles a través de conexiones no cifradas.
- **Incorporación de Archivos JavaScript de Terceros:** Inclusión de *scripts* de dominios externos, lo que puede introducir vulnerabilidades.
- **Divulgación de IP Privadas:** Exposición de direcciones IP privadas en las respuestas HTTP.
- **Encabezado X-Powered-By Filtrando Información:** Divulgación de información sobre el servidor y sus versiones a través de encabezados HTTP.
- **Falta de Encabezado Strict-Transport-Security:** La ausencia de este encabezado compromete la política de seguridad de transporte estricta (HSTS).

Alertas Informativas Principales:

- **Cookies con Alcance Relajado:** *Cookies* configuradas con un alcance de dominio amplio, lo que aumenta el riesgo de acceso no autorizado.
- **Directivas de Control de Caché Reexaminadas:** Falta o configuración incorrecta del encabezado de control de caché, lo que puede permitir el almacenamiento en caché de contenido sensible.
- **Contenido Recuperado de Caché Compartida:** Riesgo de fuga de información sensible a través de cachés compartidas.
- **Elemento HTML Controlable por el Usuario (Potencial XSS):** Valores de atributos HTML controlados por el usuario, lo que puede facilitar ataques de XSS.

En resumen, el informe de OWASP ZAP identificó varias vulnerabilidades que deben ser abordadas para mejorar la seguridad del sitio web de la URJC. Las vulnerabilidades más críticas incluyen la falta de tokens Anti-CSRF, configuraciones CSP inseguras y la falta de encabezados de seguridad esenciales.

En el anexo se incluye un enlace para la descarga del informe completo proporcionado por la herramienta.

5.4.4 Pruebas de Accesibilidad. WAVE

Las pruebas de usabilidad se realizaron para asegurar que el sitio web sea fácil de usar y accesible para todos los usuarios, incluidos aquellos con discapacidades. Se empleó la herramienta de para evaluar la accesibilidad y usabilidad del sitio web.

Métricas Evaluadas:

- Cumplimiento de las WCAG 2.1 (Web Content Accessibility Guidelines)
- Número de problemas de usabilidad detectados



Ilustración 39. Prueba Wave

Análisis

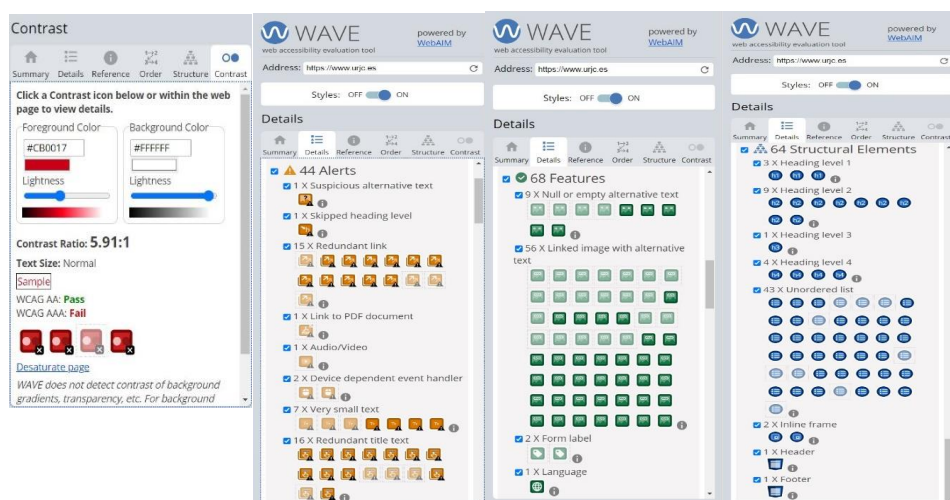


Ilustración 43. Wave Contraste

Ilustración 43. Wave Alertas

Ilustración 43. Wave Características

Ilustración 43. Wave Elementos

Contraste

- **Ratio de Contraste:** 5.91:1
- **WCAG AA:** Cumple
- **WCAG AAA:** No cumple

El ratio de contraste de **5.91:1** cumple con los requisitos de accesibilidad de WCAG AA, lo que garantiza que el texto tiene suficiente contraste con el fondo para ser legible para la mayoría de los usuarios. Sin embargo, este ratio no alcanza el nivel WCAG AAA, indicando que podría haber dificultades de legibilidad para usuarios con deficiencias visuales más severas.

Alertas

Total Alertas: 44

- **1x Suspicious Alternative Text:** Texto alternativo sospechoso que puede no describir adecuadamente la imagen.
- **1x Skipped Heading Level:** Nivel de encabezado omitido, lo que puede causar problemas de navegación.
- **15x Redundant Link:** Enlaces redundantes que conducen a la misma URL y pueden confundir a los usuarios.
- **1x Link to PDF Document:** Enlace a un documento PDF sin advertencia.
- **1x Audio/Video:** Elementos de audio o video sin controles accesibles.
- **2x Device Dependent Event Handler:** Manejadores de eventos dependientes del dispositivo, dificultando la accesibilidad.
- **7x Very Small Text:** Texto muy pequeño que puede ser difícil de leer.
- **16x Redundant Title Text:** Títulos redundantes que no aportan información adicional.

El análisis de alertas sugiere varias áreas donde la accesibilidad puede ser mejorada, incluyendo añadir descripciones alternativas adecuadas para imágenes, la corrección de encabezados, la eliminación de enlaces redundantes y la mejora de la accesibilidad de elementos multimedia.

Características

Total *Features*: 68

- **9x Null or Empty Alternative Text:** Imágenes con texto alternativo nulo o vacío.
- **56x Linked Image with Alternative Text:** Imágenes con texto alternativo adecuado.
- **2x Form Label:** Formularios con etiquetas adecuadas.
- **1x Language:** Idioma de la página correctamente especificado.

La mayoría de las características positivas incluyen imágenes vinculadas con texto alternativo adecuado, lo que es crucial para los usuarios de lectores de pantalla. Sin embargo, hay algunas imágenes con texto alternativo nulo o vacío que necesitan ser corregidas.

Elementos estructurales

Total de Elementos Estructurales: 64

- **3x Heading Level 1:** Encabezados de nivel 1.
- **9x Heading Level 2:** Encabezados de nivel 2.
- **1x Heading Level 3:** Encabezado de nivel 3.
- **4x Heading Level 4:** Encabezados de nivel 4.
- **43x Unordered List:** Listas no ordenadas.
- **2x Inline Frame:** Marcos en línea.
- **1x Header:** Encabezado de la página.
- **1x Footer:** Pie de página.

La estructura del sitio web incluye una buena jerarquía de encabezados y listas, lo que facilita la navegación. No obstante, la presencia de múltiples niveles de encabezados sugiere la necesidad de revisar la estructura de los contenidos para asegurar una jerarquía clara y lógica.

Tras la realización de una prueba de accesibilidad del sitio web de la Universidad Rey Juan Carlos, se descubrieron varias áreas de mejora. Aunque la puntuación global de accesibilidad es relativamente alta, la presencia de 19 errores de accesibilidad sugiere la necesidad de realizar ajustes para cumplir plenamente con las pautas de WCAG 2.0 AA.

En el anexo se incluye un enlace para la descarga del informe propio de la herramienta.

5.4.5 Pruebas de Usabilidad. Equal Web

En esta sección se presentan los resultados de la evaluación de accesibilidad realizada, utilizando la herramienta de extensión de Chrome de Equal Web, del sitio web de la URJC. Esta prueba tiene como objetivo identificar y analizar los problemas de accesibilidad presentes en el sitio web para asegurar su conformidad con el estándar WCAG 2.0 AA.

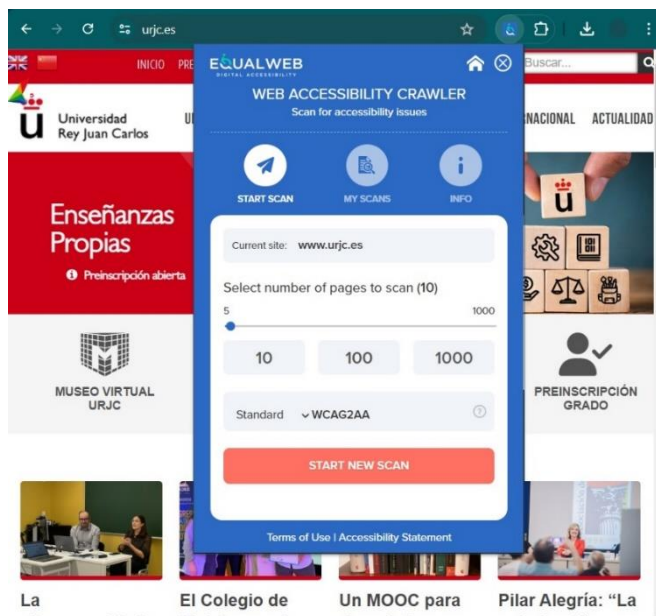


Ilustración 44. Extensión de Chrome Equal Web

Análisis

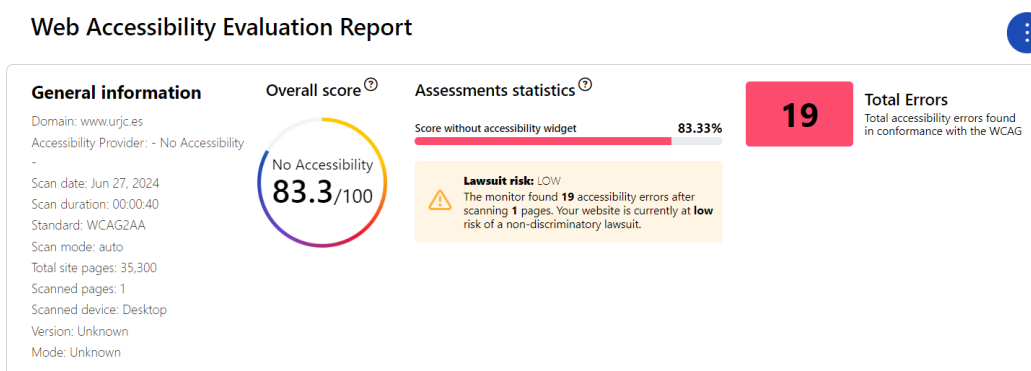


Ilustración 45. Resultados Equal Web

Puntuación General

Accesibilidad: 83.3/100

Estadísticas de Evaluación

- Puntuación sin Widget de Accesibilidad: 83.33%
- Riesgo de Demanda: Bajo

La monitorización encontró 19 errores de accesibilidad tras escanear 1 página. El sitio web actualmente presenta un riesgo bajo de recibir una demanda por discriminación.

Errores Totales

Total de Errores de Accesibilidad Encontrados según WCAG: 19

Atributos de Rol	Claves de Acceso	Encabezados
0	0	6
Atributos ARIA	Iframes	Imágenes
3	2	52
Imgs s/Texto Alternativo	Enlaces	Audio
9	232	0
Campos sin Etiqueta	PDFs	Video
4	1	0

Tabla 7. Equal Web Errores

Pautas para WCAG2AA: Problemas Restantes 19

- [SC 1.4.4: Redimensionar Texto \(Nivel AA\)](#)

Fallos: 12

- [G142](#): Uso de tecnología que tiene agentes de usuario comúnmente disponibles que soportan zoom.

- [SC 2.4.1: Saltar Bloques \(Nivel A\)](#)

Fallos: 3

- [H64](#): Uso del atributo `title` de los elementos `frame` e `iframe`.
- [H69](#): Provisión de elementos de encabezado al comienzo de cada sección de contenido.

- [SC 4.1.1: Análisis Sintáctico \(Nivel A\)](#)

Fallos: 4

- [F77](#): Fallo del Criterio de Éxito 4.1.1 debido a valores duplicados de tipo ID.

La evaluación de accesibilidad realizada con Equal Web en el sitio web de la Universidad Rey Juan Carlos ha identificado 19 errores de accesibilidad. A pesar de obtener una puntuación general de 83.3/100, lo cual indica un cumplimiento aceptable de las pautas de accesibilidad, los errores detectados sugieren áreas significativas de mejora. Entre los problemas más destacados se encuentran fallos en la redimensión del texto, la capacidad de saltar bloques de contenido repetido y errores de análisis sintáctico.

La corrección de estos problemas es crucial para mejorar la accesibilidad del sitio web y asegurar que cumple con los estándares WCAG 2.0 AA.

En el anexo se incluye un enlace para la descarga del informe completo generado por la herramienta Equal Web.

5.5 Pruebas funcionales

Las pruebas funcionales se centraron en verificar que las funcionalidades del sitio web de la URJC funcionen correctamente según los requisitos establecidos. Estas pruebas son esenciales para asegurar que cada componente del sitio web desempeñe su función prevista sin errores y que la experiencia del usuario final sea óptima. Se utilizaron herramientas como Selenium para la automatización de estas pruebas, lo que permitió realizar un análisis exhaustivo y repetitivo con mayor eficiencia y precisión.

Para la realización de las pruebas funcionales, se identificaron y documentaron los casos de prueba basados en los requisitos funcionales del sitio web. Cada caso de prueba especificaba las acciones a realizar, y los resultados esperados.

Utilizando Selenium, se desarrollaron *scripts* automatizados para ejecutar los casos de prueba definidos. Estos *scripts* simulaban interacciones de usuario como clics, entradas de datos en formularios, navegación entre páginas y verificación de resultados.

5.5.1 Pruebas en la Sección Principal

Pruebas de Rendimiento y Carga de la Página Principal

Las pruebas de rendimiento y carga demuestran que la página principal de la URJC carga eficientemente, proporcionando una experiencia de usuario rápida y sin errores, crucial para la satisfacción del usuario.

Prueba de Enlaces a Recursos PDF

El análisis de los enlaces a recursos PDF revela que la mayoría de los documentos están accesibles, salvo el enlace al "seguro escolar" que retorna un error 404, lo que indica que el documento no está disponible y necesita ser corregido.

Prueba de Navegación Principal

Prueba de Navegación en el Menú Principal: Las pruebas de navegación en el menú principal indican que todos los *items* del menú son accesibles y redirigen correctamente a sus

respectivas secciones. No se encontraron errores en la navegación, lo cual asegura que los usuarios pueden acceder fácilmente a las diferentes áreas del sitio.

Prueba de Enlaces en la Página Principal: La verificación de enlaces muestra que todos los textos importantes en la página principal están presentes y correctos, facilitando la navegación y el acceso a la información relevante para los usuarios.

Las pruebas de navegación principal muestran que la mayoría de los elementos de la página principal de la URJC funcionan correctamente y son accesibles.

Prueba de Navegación Secundaria

La verificación de enlaces HTTP revela problemas similares a los encontrados en pruebas anteriores:

- **Errores de Conexión:** Enlaces a "ayuda.urjc.es" presentan errores de conexión, indicando posibles problemas del servidor.
- **Restricciones de Acceso:** Enlace a LinkedIn con código 999, sugiriendo restricciones de acceso automatizado.
- **Acceso Prohibido:** Enlace a "campusenergiainteligente.es" con código 403, indicando acceso prohibido.

Las pruebas de navegación secundaria y verificación de enlaces indican que las secciones de ayuda y orientación de la URJC son accesibles y funcionales. Sin embargo, se identificaron problemas de conexión y acceso en algunos enlaces HTTP que requieren atención para garantizar la funcionalidad completa del sitio web. Estos problemas deben ser abordados para asegurar que todos los enlaces sean accesibles y funcionales, mejorando así la experiencia del usuario.

5.5.2 Pruebas Simples

Prueba de Búsqueda

Las pruebas de búsqueda confirman que el motor de búsqueda del sitio funciona correctamente y devuelve resultados relevantes, lo que facilita la navegación y la localización de información específica para los usuarios.

Prueba de Enlaces a Redes Sociales

La mayoría de los enlaces a redes sociales funcionan correctamente. Sin embargo, el enlace a LinkedIn presenta una restricción de acceso, que debe revisarse para garantizar que los usuarios puedan acceder al perfil de la universidad sin inconvenientes.

Prueba de Formulario

La prueba del formulario muestra que los procesos de inicio de sesión y navegación a través de la sección de ayuda para estudiantes están funcionando correctamente, permitiendo a los usuarios interactuar con el sistema sin problemas.

Pruebas de Verificación de la página

Las verificaciones de los elementos en la página principal de la URJC muestran que los textos de la política de *cookies*, el título de la página y los elementos del menú del encabezado están implementados correctamente, asegurando una buena experiencia de usuario.

Pruebas de verificación de enlaces

Prueba de Todos los Enlaces HTTP en la Página Principal: La prueba de todos los enlaces HTTP revela varios problemas de accesibilidad:

- **Errores de Conexión:** Los enlaces a "ayuda.urjc.es" presentan errores de conexión, lo que puede indicar problemas con el servidor o con la configuración de la página.
- **Restricciones de Acceso:** El enlace a LinkedIn devuelve un código 999, lo que sugiere que LinkedIn está restringiendo el acceso automatizado.

- **Acceso Prohibido:** El enlace a "campusenergiainteligente.es" devuelve un código 403, indicando que el acceso está prohibido.

Las pruebas de verificación de todos los enlaces HTTP identifica varios problemas de conexión y acceso que deben abordarse para mejorar la experiencia del usuario y la funcionalidad del sitio web. Estos problemas deben ser investigados y resueltos para garantizar que todos los enlaces en la página principal sean accesibles y funcionen correctamente.

5.5.3 Pruebas en Profundidad del Dominio URJC

Pruebas de Conexión con el servidor en todo el Dominio urjc.es

Códigos 200 (Éxito): 180 enlaces.

Códigos no 200: 40 enlaces.

- **404 (No Encontrado):** 1. No se encuentra disponible y requiere revisión
- **302 (Redirección Temporal):** 27. Indican que los enlaces han sido redirigidos a nuevas URLs. Esto puede ser normal, pero debe ser revisado para asegurarse de que las redirecciones se comporten correctamente y no afecten la experiencia del usuario.
- **301 (Redirección Permanente):** 8. Indica que los enlaces han sido redirigidos a nuevas URLs. Esto puede ser normal, pero debe ser revisado para asegurarse de que las redirecciones se comporten correctamente y no afecten la experiencia del usuario.
- **999 (Error no especificado):** 1. Suele ser un bloqueo de acceso automatizado; no es necesariamente un problema de enlace, pero puede requerir manejo especial.
- **403 (Prohibido):** 2. Indica un problema de acceso que debe ser investigado para restaurar el acceso.
- **503 (Servicio No Disponible):** 1. Indica un problema de disponibilidad que debe ser investigado para restaurar el acceso.

La mayoría de los enlaces externos y de PDF cargan correctamente, asegurando un acceso continuo y sin problemas para los usuarios. No obstante, hay un número significativo de enlaces con redirecciones y errores de acceso que deben ser revisados y corregidos para mejorar la accesibilidad y la funcionalidad del sitio web.

Pruebas de Rendimiento en todo el Dominio urjc.es

La prueba de rendimiento realizada consistió en medir el tiempo de carga de las URLs del dominio de la URJC. Se obtuvieron tiempos de carga para 151 URLs diferentes.

A continuación, se presenta un análisis detallado de los resultados.

Estadísticas Descriptivas

- Media (promedio) de tiempo de carga: 7.05 segundos
- Mediana de tiempo de carga: 6.68 segundos
- Desviación estándar: 1.34 segundos
- Tiempo de carga mínimo: 1.62 segundos
- Tiempo de carga máximo: 12.05 segundos

Distribución de los Tiempos de Carga

La distribución de los tiempos de carga muestra que la mayoría de las páginas tienen un tiempo de carga cercano a la media. Sin embargo, hay algunas páginas con tiempos de carga significativamente mayores, lo cual incrementa la desviación estándar.

- Rango Intercuartil (IQR): 1.42 segundos
- Percentil 25: 6.51 segundos
- Percentil 75: 7.93 segundos
- URL más rápida: https://www.urjc.es/images/EstudiarURJC/oficina_estudiante/seguro_escolar.pdf con 1.62 segundos
- URL más lenta: <https://eulist.urjc.es/> con 12.05 segundos

Outliers positivos (tiempos de carga significativamente altos):

- <https://eulist.urjc.es/> (12.05s)
- <https://eventos.urjc.es/115480/detail/cv-21-playinn-co-diseno-de-prototipo-para-jovenes-en-fuenlabrada-co-create-new-european-bauhaus-.html> (10.70s)
- <https://itunes.apple.com/es/app/urjc-app-univ-rey-juan-carlos/id1176072357?mt=8> (9.73s)



- <https://eventos.urjc.es> (9.81s)
- <https://eventos.urjc.es/115327/detail/cv-15-estado-de-derecho-gobernabilidad-y-enfrentamiento-institucional-.html> (9.79s)

Estos *outliers* indican problemas de rendimiento en estas URLs específicas que deberían ser investigados y optimizados para mejorar la experiencia del usuario.

El tiempo de carga promedio de las URLs de la URJC está dentro de un rango aceptable para la mayoría de las páginas, aunque hay algunas URLs con tiempos de carga que podrían afectar negativamente la experiencia del usuario. Es recomendable realizar una optimización específica en las páginas identificadas como *outliers* para mejorar su rendimiento.

Pruebas de Enlaces a Recursos PDF en todo el Dominio urjc.es

Solo se ha encontrado un enlace a PDF correctamente identificado, y ha sido verificado y carga sin problemas, proporcionando acceso a información importante sin errores.

Pruebas de Carga de Todos los Enlaces Externos de todo el Dominio urjc.es

Los enlaces externos principales han sido verificados y cargan sin problemas. Esto asegura que los usuarios pueden acceder a recursos y plataformas asociadas sin interrupciones.

5.6 Resultados finales

Se implementaron diversas pruebas utilizando Gherkin y Python para evaluar aspectos críticos del sitio web de la URJC. Las pruebas de navegación aseguraron que los usuarios pudieran acceder a todas las secciones clave del sitio sin encontrar errores ni enlaces rotos. Las pruebas de búsqueda evaluaron la efectividad y relevancia de los resultados obtenidos al utilizar la funcionalidad de búsqueda del sitio. Además, se realizaron pruebas exhaustivas en el formulario de acceso de alumnos del sitio para la realización de consultas para verificar que todas las entradas y validaciones funcionaran correctamente.

Las pruebas de enlaces externos y recursos PDF garantizaron que todos los enlaces apuntaran a destinos válidos y que los archivos PDF se descargaran y visualizaran correctamente. Se llevaron a cabo pruebas de rendimiento utilizando JMeter para simular tráfico de usuarios y evaluar la capacidad del sitio para manejar múltiples conexiones simultáneas sin degradar su rendimiento. Las pruebas de seguridad con OWASP ZAP identificaron y ayudaron a mitigar vulnerabilidades potenciales, mejorando la protección del sitio contra amenazas comunes.

En conjunto, estas pruebas proporcionaron una evaluación integral del sitio web, destacando áreas de mejora y asegurando una experiencia de usuario óptima.

Conclusiones

6.1 Logros alcanzados

El desarrollo del proyecto de auditoría del sitio web de la URJC ha sido un proceso desafiante que ha abarcado diversos aspectos, desde la investigación de los temas relacionados, pasando por la planificación inicial hasta la implementación y las pruebas exhaustivas. Al combinar pruebas automatizadas con el uso de herramientas externas de pruebas, se ha asegurado que se cumplan los estándares de funcionalidad, rendimiento y seguridad. Las herramientas y técnicas implementadas proporcionaron una evaluación integral, identificando y resolviendo problemas críticos que pueden mejorar significativamente la calidad del sitio web.

6.2 Validación de Objetivos

6.2.1 Objetivo Principal (OP)

- **OP:** Realizar una auditoría del sitio web oficial de la URJC mediante pruebas automatizadas para garantizar su funcionalidad, rendimiento y seguridad.

Validación: Se llevaron a cabo sendas pruebas que confirmaron la funcionalidad, rendimiento y seguridad del sitio web, cumpliendo así con este objetivo principal. Los resultados detallados de estas pruebas se encuentran en el anexo.

6.2.2 Objetivos Específicos (OE)

- **OE 1:** Realizar un estudio sobre la auditoría de sitios web, incluyendo un análisis crítico de las herramientas y metodologías disponibles en la literatura y en la práctica profesional.

Validación: Se completó un estudio exhaustivo sobre la auditoría de sitios web, que incluyó una revisión detallada de las herramientas y metodologías disponibles tanto en la literatura académica como en la práctica profesional. Los hallazgos y el análisis crítico de este estudio se documentaron y se incluyen en el anexo.

- **OE 2:** Implementar y ejecutar pruebas de rendimiento, seguridad, accesibilidad y usabilidad en el contexto de la auditoría de sitios web.

Validación: Se implementaron y ejecutaron diversas pruebas centradas en el rendimiento, seguridad, accesibilidad y usabilidad del sitio web de la URJC. Estas pruebas proporcionaron información valiosa sobre el estado actual del sitio web y ayudaron a identificar áreas de mejora. Los resultados de estas pruebas están documentados en el anexo.

- **OE 3:** Desarrollar y llevar a cabo un conjunto de pruebas detalladas sobre el sitio web de la URJC, analizando los resultados obtenidos para identificar áreas de mejora.

Validación: Se desarrollaron y ejecutaron un conjunto de pruebas detalladas en el sitio web de la URJC. Los resultados obtenidos fueron minuciosamente analizados para identificar posibles áreas de mejora. Este análisis permitió realizar recomendaciones específicas para optimizar el sitio web. Los detalles de estas pruebas y sus resultados se encuentran en el anexo.

A través de la validación de los objetivos planteados, se ha confirmado el cumplimiento satisfactorio de cada uno de ellos. La auditoría realizada proporcionó una evaluación del sitio web de la URJC, asegurando su funcionalidad, rendimiento y seguridad. El estudio de las herramientas y metodologías de auditoría, junto con las pruebas implementadas y ejecutadas, y las realizadas mediante herramientas externas, permitió identificar y documentar áreas de mejora específicas.

Este proceso no solo garantiza que el sitio web cumpla con los estándares necesarios, sino que también proporciona una base sólida para futuras optimizaciones. Los resultados y análisis presentados en los anexos reflejan el éxito de este proyecto y su contribución a la mejora continua del sitio web oficial de la URJC.

6.3 Logros y Desafíos

Este proyecto ha traído consigo diversos logros y desafíos significativos que se han superado con éxito a lo largo del proceso.

Logros:

- Implementación exitosa de pruebas automatizadas, asegurando un proceso de auditoría efectivo y repetible.
- Realización de una evaluación detallada de la seguridad y el rendimiento del sitio web de la URJC, proporcionando información útil para su optimización.
- Validación de las normativas y los estándares de accesibilidad y usabilidad en el sitio web de la URJC, verificando que la experiencia de usuario sea inclusiva y efectiva.

Desafíos:

- Superación de la complejidad en la integración de diversas herramientas de pruebas automatizadas, lo que requirió un enfoque meticuloso y bien documentado.
- Adaptación de las pruebas para cubrir todos los posibles escenarios de uso, asegurando una evaluación completa y precisa del sitio web de la URJC.

6.4 Lecciones Aprendidas

Al desarrollar el proyecto, se han podido reconocer las siguientes lecciones:

- **La importancia de una planificación detallada y estructurada:** Una planificación meticulosa es crucial para el éxito de un proyecto, permitiendo identificar y mitigar riesgos anticipadamente.
- **El valor de las pruebas automatizadas para garantizar la calidad y fiabilidad del sitio web:** Las pruebas automatizadas no solo ahorran tiempo y recursos, sino que también aseguran una cobertura de prueba más amplia y consistente.
- **La necesidad de una constancia y versatilidad para superar los desafíos técnicos:** La constancia y la versatilidad a la hora de abordar un desafío técnico son muy importantes para resolver problemas complejos y asegurar el avance del proyecto.
- **La importancia de seleccionar las herramientas adecuadas y configurar correctamente el entorno de pruebas:** La elección correcta de herramientas y una configuración adecuada del entorno de pruebas son fundamentales para obtener resultados precisos y fiables. Esto incluye tanto herramientas de automatización como entornos de ejecución y análisis.

Estas lecciones no solo han contribuido al éxito de este proyecto, sino que también proporcionan una base sólida para futuros trabajos en el campo de la auditoría y automatización de pruebas de sitios web.

6.5 Líneas futuras

El desarrollo de este proyecto abre la puerta a futuras mejoras y expansiones, asegurando que el sitio web de la URJC se mantenga actualizado y cumpla con los estándares más exigentes.

Actualizaciones y Mejoras

- **Ampliación del alcance de las pruebas:** Incluir nuevas áreas del sitio web y funcionalidades adicionales para una cobertura más completa.
- **Optimización continua del rendimiento:** Implementar mejoras basadas en los resultados de las pruebas de rendimiento para asegurar tiempos de respuesta aún más rápidos.
- **Mejoras en la seguridad:** Actualizar y mejorar las medidas de seguridad para proteger contra nuevas amenazas y vulnerabilidades.
- **Incorporación de nuevas herramientas:** Evaluar e integrar nuevas herramientas de pruebas y auditoría para mejorar la *scripts* eficiencia y precisión de las pruebas.
- **Automatización avanzada:** Desarrollar de automatización más sofisticados que puedan adaptarse a cambios en el sitio web de manera más eficiente.



Anexo

En el anexo se incluye información adicional relevante:

- Los documentos con los resultados de las pruebas automatizadas.
- Los informes y resultados obtenidos mediante las herramientas externas.
- El diagrama de Gantt en formato ampliado.
- Los diagramas UML en formato ampliado.
- El código de las pruebas automatizadas.
- Instrucciones en el lanzamiento de los test

El enlace de descarga del anexo es el siguiente:

<https://mega.nz/folder/E74DFRbY#B63yQtRM1dNgUjkam6ct4w>

Glosario

Este glosario proporciona definiciones claras y concisas de los términos clave utilizados en el proyecto. Su propósito es facilitar la comprensión de conceptos técnicos y asegurar una comunicación efectiva con los lectores.

Términos

Clickjacking

Técnica maliciosa en la que se engaña al usuario para que haga clic en algo diferente a lo que percibe, comprometiendo así la seguridad del usuario y del sistema.

Cookies: Pequeños archivos de datos que un sitio web almacena en el dispositivo del usuario para recordar información sobre la visita del usuario, como preferencias y sesiones.

Feature

Característica o funcionalidad específica de un sistema o software que describe qué debe hacer el sistema.

Frame

Elemento HTML utilizado para mostrar otra página web dentro de una página web, comúnmente usado para incrustar documentos o contenidos de otras fuentes.

Framework

Conjunto de herramientas y bibliotecas que proporcionan una estructura reutilizable para desarrollar y probar software.



Given, When, Then

Palabras clave utilizadas en la metodología Gherkin para definir el comportamiento esperado en los escenarios de prueba: "Given" establece el contexto, "When" describe una acción, y "Then" define el resultado esperado.

Iframe

Elemento HTML similar a un frame, pero utilizado específicamente para incrustar una página web dentro de otra página web.

Item

Elemento o unidad dentro de un conjunto, lista o grupo que se está evaluando o utilizando en un proyecto.

Outliers

Valores atípicos que se encuentran fuera del rango normal de un conjunto de datos y pueden indicar anomalías o variaciones significativas.

Rango Intercuartil

Medida estadística de dispersión que describe la distancia entre el primer y el tercer cuartil de un conjunto de datos, útil para identificar la variabilidad central de los datos.

Scripts

Secuencias de comandos o instrucciones escritas en un lenguaje de programación o scripting que se ejecutan para automatizar tareas o realizar pruebas.

Sistema de reporting

Herramienta o sistema utilizado para generar informes detallados sobre el rendimiento, resultados de pruebas, y otros aspectos del sistema evaluado.



Stakeholder

Persona, grupo u organización con interés o preocupación en un proyecto, que puede afectar o ser afectado por el resultado del proyecto.

Step

Paso o acción individual que se debe realizar dentro de un proceso, especialmente en la ejecución de pruebas.

Tester

Persona responsable de diseñar, ejecutar y evaluar pruebas para asegurar que un sistema o software funcione correctamente y cumpla con los requisitos.

Testing

Proceso de evaluar un sistema o software mediante la ejecución de pruebas para identificar defectos, asegurar la calidad y verificar que cumple con los requisitos especificados.

Bibliografía

- [1] JetBrains s.r.o., «JetBrains - Pycharm,» JetBrains s.r.o., [En línea]. Available: <https://www.jetbrains.com/es-es/pycharm/>.
- [2] M. A. C. Gaité, «Paradigma Digital,» [En línea]. Available: <https://www.paradigmadigital.com/dev/bdd-gherkin-comunicar-mundos-diferentes/>.
- [3] A. P. España, «Python España,» [En línea]. Available: <https://es.python.org/aprende-python/>.
- [4] Y. R. Jennifer Realpe, *Auditoria al portal web del Hospital Universitario Departamental de Nariño ESE*, 2013.
- [5] «Universidad Rey Juan Carlos,» [En línea]. Available: <https://www.urjc.es>.
- [6] iso25000.com, «ISO25000.com,» 2022. [En línea]. Available: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>.
- [7] (. C. Muncharaz, «WCAG 2.1 de un vistazo,» W3C WAI (Web Accessibility Initiative), 29 11 2023. [En línea]. Available: <https://www.w3.org/WAI/standards-guidelines/wcag/glance/es>.
- [8] U. Europea, «Reglamento general de protección de datos (RGPD),» European Union law, 07 01 2022. [En línea]. Available: <https://eur-lex.europa.eu/ES/legal-content/summary/general-data-protection-regulation-gdpr.html>.

- [9] G. d. España, «Real Decreto 1112/2018, de 7 de septiembre, sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público.,» «BOE» núm. 227, de 19 de septiembre de 2018, páginas 90533 a 90549 (17 págs.), vol. Sección A, nº 227, p. 17 (90533 a 90549), 07 09 2018.
- [10] J. F. M. Vasquez, *QA Multiproyectos Testing*, 2023.
- [11] «PRUEBAS DE SOFTWARE: HISTORIA Y EVOLUCIÓN,» FYCGroup, 02 02 2023. [En línea]. Available: <https://www.fycorp.com/articulo-pruebas-de-software:-historia-y-evolucion>.
- [12] «El futuro de las auditorías y las nuevas prácticas tecnológicas,» CTA,AUDITORES, [En línea]. Available: https://es.linkedin.com/pulse/el-f%C3%BAturo-de-las-auditor%C3%ADas-y-nuevas-pr%C3%A1cticas-tecnol%C3%B3gicas-af5ee?trk=public_post.
- [13] ¿. e. e. p. d. l. t. y. l. a. e. l. auditoría?, Aud&Con Quality S.L., 19 06 2024. [En línea]. Available: <https://audycon.es/el-papel-de-la-tecnologia-y-la-automatizacion-en-la-auditoria/>.
- [14] R. C. Montenegro, *Testing y Calidad Software. Automatización de Pruebas con Selenium WebDriver*, ETSIIST, Ed., UPM, p. 156.
- [15] F. Capital, «Desarrollo impulsado por el comportamiento BDD Introduccion a BDD mejores practicas y ejemplos,» 17 04 2024. [En línea]. Available: <https://fastercapital.com/es/contenido/Desarrollo-impulsado-por-el-comportamiento--BDD--Introduccion-a-BDD--mejores-practicas-y-ejemplos.html>.

- [16] R. Hernández, *BDD, Cucumber y Gherkin. Desarrollo dirigido por comportamiento*, Genbeta, 2018.
- [17] Google, «Google,» Google, [En línea]. Available: <https://www.google.com/>.
- [18] S. Mazzocchi, «JMeter en Español,» Apache, [En línea]. Available: <https://jmeterenespanol.org/>.
- [19] D. G. SAFLA, «VULNERABILIDADES EN APLICACIONES WEB UTILIZANDO LA METODOLOGÍA DE “PROYECTO ABIERTO DE SEGURIDAD DE APLICACIONES WEB”,» 2021.
- [20] HostedScan, «OWASP ZAP Online Scan,» HostedScan, [En línea]. Available: <https://hostedscan.com/owasp-vulnerability-scan>.
- [21] W. WebAim, «WAVE Web Accessibility Evaluation Tools,» [En línea]. Available: <https://wave.webaim.org/>.
- [22] «The World's #1 Web Accessibility Solution,» EqualWeb, [En línea]. Available: https://www.equalweb.com/html5/?_id=8591&did=1116&trace=EU_WEBAccessibility&_gid=654321&utm_source=adwords&utm_medium=ppc&utm_term=website%20accessibility&utm_campaign=EQUALWEB+EU&hsa_cam=1649673773&hsa_grp=149864524493&hsa_ver=3&hsa_mt=b&hsa_kw=website%2.
- [23] «Real Academia Española,» [En línea]. Available: <https://www.rae.es/>.



- [24] G. N. G. Ugarte, «Automatización de Pruebas.,» *Journal Boliviano de Ciencias*, vol. 10, nº 30, pp. 54-57, 2014.
- [25] Lucidchart, «Lucidchart,» [En línea]. Available:
<https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-gantt>.