

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**REFACTORIZACIÓN DEL SISTEMA WEB DE GESTIÓN DE
RECURSOS DE LOS SERVICIOS DE PROTECCIÓN INTEGRAL
(SPI) EN LA SECRETARÍA DE DERECHOS HUMANOS (SDH).**

REFACTORIZACIÓN DE UN SISTEMA WEB

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

VICENTE DAVID DELGADO GÜIZ

vicente.delgado@epn.edu.ec

DIRECTOR: ING BYRON LOARTE

byron.loarteb@epn.edu.ec

DMQ, agosto de 2023

CERTIFICACIONES

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

VICENTE DAVID DELGADO GUIZ

vicente.delgado@epn.edu.ec

Certifico que el presente trabajo de integración curricular fue desarrollado por Vicente Delgado, bajo mi supervisión.

Ing. BYRON LOARTE, MSc.

DIRECTOR

byron.loarteb@epn.edu.ec

DECLARACIÓN DE AUTORÍA

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

VICENTE DAVID DELGADO GÜIZ

DEDICATORIA

Dedico este proyecto a todas las personas que hicieron posible que se cumpliera, en especial a mi familia que significa todo para mí, a mis padres que son y serán los pilares de principios y valores que me convirtieron en la persona que soy ahora, a mis hermanos por estar siempre allí y poder contar con su apoyo, los amos con todo el corazón.

VICENTE DAVID DELGADO GÜIZ

AGRADECIMIENTO

Agradezco a Dios, por mostrarme el camino que no fue el más fácil, pero si el correcto, por darme la fortaleza para enfrentar los desafíos que se cruzaron en el trayecto, a mis padres, por estar en cada momento que los necesité, por su apoyo, por los consejos, y sobre todo por el amor siempre me dan, a mis hermanos, por saber que siempre contaré con ellos en cada situación que se presente en mi vida.

Quiero mencionar a mis profesores de la prestigiosa Escuela de Formación de Tecnólogos, quienes me supieron guiarme y formarme como profesional, los recordaré como parte de mi vida.

Por último, gracias a mis compañeros de universidad por los momentos que pasamos juntos en esta etapa de la vida, siempre formaran parte de mi corazón.

VICENTE DAVID DELGADO GÜIZ

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	II
DECLARACIÓN DE AUTORÍA.....	III
DEDICATORIA.....	IV
AGRADECIMIENTO.....	V
ÍNDICE DE CONTENIDO.....	VI
RESUMEN	VIII
ABSTRACT	IX
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	4
2 METODOLOGÍA	7
2.1 Metodología de Desarrollo	7
Roles.....	8
Artefactos.....	9
2.2 Diseño de interfaces	12
Herramienta utilizada para el diseño.....	12
Diseño sistema <i>web</i>	12
2.3 Diseño de la arquitectura	13
Patrón arquitectónico	13
2.4 Herramientas de desarrollo.....	14
Librerías.....	15
3 RESULTADOS.....	16
3.1 <i>Sprint</i> 0. Configuración del ambiente de desarrollo.	16
Recopilación de requerimientos para el sistema <i>web</i>	16
Configuración de PostgreSQL para la Base de datos SQL.	18
Creación del proyecto Maven <i>Spring Boot</i>	18
Asignación de roles a los usuarios del sistema <i>web</i>	19
3.2 <i>Sprint</i> 1. Módulos refactorizados para el administrador.	19
Registro de usuarios.....	19
Inicio y cierre de sesión	20

Modificar usuarios.....	22
Modificar unidad, instituciones y roles	23
3.3 <i>Sprint 2. Módulos refactorizados para el encargado.</i>	24
Categorías.....	24
Gestionar bienes de los SPI	25
Gestionar recursos humanos de los SPI	27
Gestionar inventario por SPI	28
Generar reportes de los SPI	29
Subir verificables	29
3.4 <i>Sprint 3. Módulos refactorizados para visitante.</i>	31
Visualizar y generar reportes de inventario de los SPI.....	31
3.5 <i>Sprint 4. Pruebas del sistema web refactorizado.</i>	32
Pruebas unitarias.....	32
Pruebas de compatibilidad.....	33
Pruebas de aceptación	34
3.6 <i>Sprint 5. Despliegue del sistema web</i>	35
4 Conclusiones	36
5 Recomendaciones	37
6 REFERENCIAS BIBLIOGRÁFICAS.....	38

RESUMEN

La Secretaría de Derechos Humanos (SDH), ahora conocida como Ministerio de la Mujer y Derechos Humanos (MMD), disponía de un sistema web para la gestión de recursos en los Servicios de Protección Integral (SPI) con el objetivo de mejorar la administración de los pedidos solicitados por los SPI hacia a SDH. Sin embargo, dicho sistema presentaba varias falencias que dificultaba el manejo de la información y por tal motivo se requería una pronta actualización por temas de seguridad, herramientas, librerías y la incorporación de nuevos módulos.

Por tal razón, en este proyecto de Integración Curricular se ha refactorizado todo el sistema que actualmente dispone la SDH, proporcionando una serie de beneficios en términos de gestión de recursos para los Servicios de Protección Integral mediante el uso de tecnologías modernas, *Open Source*, escalables y en conjunto con buenas prácticas de programación. Aportando una serie de ventajas y sobre todo la prevención de problemas a futuro.

Por último, el presente escrito se encuentra dividido en cinco secciones, cada una detallando secciones clave. En la primera sección, se da a conocer sobre la problemática a resolver en la Secretaría de Derechos Humanos, objetivos específicos, alcance y el marco teórico respectivo. Posterior a ello, en la segunda sección se presenta la metodología ágil que ha servido para determinar roles y artefactos, también, se detalla cada una de las interfaces, herramientas, librerías, patrón de arquitectura. En la tercera sección, se detalla cada una de las tareas que se han establecido, así como el resultado que se han alcanzado conjuntamente con su prueba unitaria. Finalmente, se presenta las respectivas conclusiones y recomendaciones derivadas de todo el trabajo de Integración Curricular.

PALABRAS CLAVE: *Spring Boot, Maven, Java, Thymeleaf, Backend, Frontend.*

ABSTRACT

The Secretariat of Human Rights (SDH) now known as the Ministry of Women and Human Rights (MMD) had a web system for the management of resources in the Integral Protection Services (SPI) with the aim of improving the management of each of the requests made by the SPI. However, this system had several shortcomings, which made it difficult to manage the information and, therefore, it was necessary to update it as soon as possible in terms of security, tools, libraries and the incorporation of new modules.

For this reason, this Curriculum Integration project has refactored the entire system currently available to the SDH, providing a number of benefits in terms of resource management for the Integral Protection Services through the use of modern technologies, open source, scalable and in conjunction with good programming practices, among others. Providing a series of advantages and above all the prevention of future problems such as those mentioned above.

Finally, this paper is divided into five sections, each one detailing key sections. In the first section, the problem to be solved in the Human Rights Secretariat, specific objectives, scope and the respective theoretical framework are presented. Subsequently, the second section presents the agile methodology that has served to determine roles and artifacts, also, each of the interfaces, tools, libraries, architecture pattern is detailed. In the third section, each of the tasks that have been established are detailed, as well as the results that have been achieved together with their unit test. Finally, the respective conclusions and recommendations derived from all the work of Curricular Integration are presented.

KEYWORDS: *Spring Boot, Maven, Java, Thymeleaf, Backend, Frontend.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

La Secretaría de Derechos Humanos (SDH) Ecuador fue creada el 15 de noviembre de 2007 como un ministerio, ante la necesidad de coordinar el trabajo de todos los operadores de justicia y el de optimizar proyectos que tengan funciones judiciales [1]. A la fecha la afinidad de la SDH es la dirección de rehabilitación social e instituciones relacionadas con el sistema de justicia, también está encargada de la administración de los Servicios de Protección Integral (SPI) que ofrecen de manera gratuita servicios de apoyo psicológico, asesoramiento legal y asistencia social a individuos afectados por situaciones de violencia, trata de personas y casos referidos por entidades como la fiscalía y unidades especializadas, así como juntas de protección y otras organizaciones. Al momento existen 45 oficinas SPI, distribuidas en diferentes ciudades del país [2].

Una gestión eficaz de los recursos es crucial para una administración eficiente tanto en las empresas privadas como públicas [1]. En ese sentido, en el departamento administrativo de la SDH la gestión de los recursos en los SPI no era lo suficientemente eficiente debido al desorden en el manejo de estos con respecto a los pedidos de los SPI y sus necesidades, lo que ocasionaba duplicidad de información, falta de seguridad, confusión en la asignación de los recursos y una serie de molestias en los administradores de cada SPI a nivel nacional. Por otra parte, el sistema que se encuentra en producción requiere una pronta actualización, permitiendo solucionar parte de los problemas antes mencionados y así asegurar una gestión adecuada de los recursos de los SPI del país que son administrados por la SDH ubicada en la ciudad de Quito. Además, la relevancia de disponer de un sistema en línea en el marco de la SDH se fundamenta en dos motivos principales, la primera es trabajar con una herramienta informática moderna que sirva para gestionar los recursos destinados a los SPI de forma apropiada y pueda ser utilizada por los funcionarios a través de la intranet del Ministerio y la segunda tener la factibilidad de actualizarla constantemente según nuevos requerimientos y herramientas de desarrollo modernas [2].

Por otra parte, los directivos de la SDH señalan que el contar con un sistema *web*, puede mejorar la gestión de los recursos para trabajar de forma ordenada, eficiente y así dar atención ordenada a cada SPI con la inmediatez que corresponda. Además, este sistema, será de gran utilidad para la productividad de los servidores públicos que trabajan dentro de la SDH en el ámbito de gestionar de forma organizada y en tiempo real los distintos recursos para los diferentes SPI que se encuentran a nivel nacional repartidos en 9 zonas centrales [2].

Por lo antes expuesto, este Trabajo de Integración Curricular está orientado a la refactorización de todo el sistema que actualmente dispone la SDH, proporcionando una serie de beneficios en términos de gestión de recursos para los Servicios de Protección Integral mediante el uso de tecnologías modernas, *Open Source*, escalables y en conjunto con buenas prácticas de programación, entre otros. Aportando una serie de ventajas y sobre todo la prevención de problemas a futuro como los mencionados anteriormente.

1.1 Objetivo general

Refactorizar el sistema *web* de gestión de recursos de los Servicios de Protección Integral (SPI) para la Secretaría de Derechos Humanos (SDH).

1.2 Objetivos específicos

1. Identificar los requisitos funcionales y no funcionales para el sistema *web*.
2. Refactorizar la estructura de la base de datos de acuerdo a los requisitos presentados.
3. Refactorizar el *backend* en concordancia con los requerimientos que se han solicitado.
4. Refactorizar el *frontend* en concordancia con los requerimientos que se han solicitado.
5. Probar el correcto funcionamiento del sistema *web* refactorizado.
6. Desplegar a producción el sistema *web* refactorizado para su uso interno como plan piloto.

1.3 Alcance

En la actualidad un sistema de control de inventarios radica principalmente en la gestión adecuada de suministros, compras, pedidos, almacenamiento y *stock* de todos los bienes o activos, con el objetivo de satisfacer la demanda de todos los empleados, clientes y sobre todo sin causar disconformidad con el fin de mantener un balance estable a lo largo de su existencia [3]. Adicional a ello, si se habla de mantener un balance estable, es necesario destacar la significancia de disponer con un sistema de *software* actualizado ya que se agrega valor en cuanto a seguridad de los datos, experiencia de usuario y monitoreo constante de la entrada y salida de datos [4]. Hay que recalcar que se define como refactorización a los cambios menores que se realizan a un sistema *software* con el propósito de corregir vulnerabilidades, proteger de ataques de *malware*, corregir errores y

mejorar la productividad con la actualización de bibliotecas, entre otras acciones. Además, pueden estar ligadas también con el cambio de herramientas, nuevas tecnologías o simplemente a paquetes [5].

El sistema *web* se encuentra desarrollado y estructurado en un solo componente principal un *backend* y *frontend* el cual utiliza un motor de plantilla, este componente es el encargado del procesamiento, seguridad y presentación de la información ante los usuarios finales [6]. Empleando para este propósito, una serie de herramientas que permiten su escalabilidad, entre las cuales se destacan un *Framework Open Source* denominado *Spring Boot*, que es un módulo que facilita el desarrollo de aplicaciones de lado del servidor y la simplificación de configuraciones para el acceso a los datos usando JDBC y la preparación para entornos a producción. Un motor de plantillas Thymeleaf, el cual es especializado para la presentación de la información en una interfaz *web* ofreciendo una agradable experiencia de usuario gracias a sus notables características. También se debe resaltar la incorporación de un enfoque de desarrollo de *software* basado en una metodología ágil (*Scrum*), que ha permitido una correcta planificación con todo el equipo de trabajo al momento de refactorizar el sistema *web* y una serie de pruebas que garantizan la calidad del producto final antes de ser puesto en un ambiente de producción.

Por último, se ha utilizado *Spring Security* para la protección de datos, autenticación y autorización de los diferentes perfiles mencionados a continuación:

El perfil administrador en el sistema *web* puede:

- Gestionar el sistema *web* en general.
- Crear usuarios con perfil administrador, encargado y visitante.

El perfil encargado de los recursos en el sistema *web* puede:

- Gestionar los datos de los SPI.
- Gestionar los recursos de los SPI.
- Gestionar el recurso humano de los SPI.
- Gestionar inventarios.
- Generar reportes en dos tipos de formatos.

El perfil invitado en el sistema *web* puede:

- Iniciar sesión.
- Visualizar los datos de los recursos de los SPI.
- Generar reportes en varios formatos.

1.4 Marco Teórico

El proceso “desarrollo de *software*” es la agrupación de tareas informáticas relacionadas en diseñar, crear, probar y desplegar un *software*, las cuales deben ser realizadas a lo largo del ciclo de vida de un proyecto de *software* [7]. Por otra parte, un sistema *web*, es un medio tecnológico que los usuarios utilizan para acceder a cualquier tipo de información en particular. Además, se puede utilizar un navegador *web* como herramienta de las Tecnologías de la Información y las Comunicaciones (TIC), para acceder a esta información [8].

El *backend* es la parte lógica de un sistema *software* el cual trabaja con una tecnología en particular. Además, debe asegurar la correcta funcionalidad de todos los elementos en la arquitectura interna y el procesamiento de los datos para que sean gestionados a través de peticiones, así como la interacción con la Base de Datos, el procesamiento de datos multimedia, los procesos simples como consultas y la seguridad a la información [6].

Interfaz de programación de Aplicaciones (API), no es más que la unión de conceptos y protocolos que se establecen para unir y comunicar dos aplicaciones de forma fluida siguiendo una serie de pautas claras [9]. Por otra parte, Transferencia de Estado Representacional (REST) no es un protocolo, más bien es un estilo de arquitectura que los desarrolladores deben implementar al momento de crear APIs [10].

Java es un lenguaje de programación y en la actualidad es considerado una plataforma informática orientada a objetos, que contiene una máquina virtual Java la misma que favorece a la distribución porque habilita la ejecución de aplicaciones en un considerable número de ambientes y sistemas operativos [11].

Thymeleaf es un motor de plantilla Java por lado del servidor enfocado en ambientes *web* e independientes, además, se centra en trabajar con plantillas HTML5 para que sean compatibles en varios navegadores. A la fecha, debido a su gran popularidad tiene una

gran integración con *Spring* y otros *Frameworks* como Bootstrap, siendo una opción ideal para el desarrollo de sistemas *web* [12].

Spring es un *Framework* de licencia libre usado en el ámbito empresarial por facilitar la creación de aplicaciones de clase Kotlin, Java y Groovy. Otra característica notable es su inyección de dependencias que incrementa la productividad y reduce el estrés en el desarrollo, debido a los varios soportes que ofrece y lo sencillo que es implementarlo en un proyecto java, algunas dependencias usadas en este proyecto son [13]:

- *Spring Web Services*: Usado para el servicio en la *web* en conjunto con el protocolo *Simple Object Access Protocol (SOAP)*.
- *Spring Data*: Usado en el manejo y gestión de Base de datos relacionales, no relacionales y servicio en la nube.
- *Spring MVC*: Dependencia necesaria para el desarrollo de aplicaciones y sistemas *web*.
- *Spring Security*: Componente usado para proporcionar seguridad al sistema *web* ya que ofrece la implementación de módulos de autenticación y autorización con un acceso potente y sumamente personalizable [13].

Spring Boot es una tecnología y herramienta de desarrollo que facilita la creación de aplicaciones *web* basadas en dependencias, estas facultan reducir la longitud del código con configuraciones mínimas gracias a su IDE *Sprint Tool Suit* y que en producción simplemente se ejecute con todas las bibliotecas necesarias contenidas en un paquete denominado Maven [14].

PostgreSQL es un administrador de Base de datos potente de código *Open Source* con gran solidez en el mercado actual debido a su flexibilidad, fiabilidad y soporte de altos estándares técnicos libres [15].

Las pruebas en un sistema *software* son una parte fundamental en el desarrollo de un proyecto, es decir en el transcurso del ciclo se debe ejecutar acciones para comprobar el correcto funcionamiento según los requerimientos que va a cubrir el sistema *software*, antes de subirlo a producción. Una de estas acciones son las pruebas unitarias que se centran en probar elementos o unidades individuales. Otro tipo de pruebas son las de aceptación, las cuales consisten en la validación y aceptación de las funcionalidades del sistema *software*, por parte del dueño de producto. Por último, las pruebas de rendimiento y compatibilidad consisten en el correcto funcionamiento de todo el sistema *software*

independiente del navegador que se esté usando, es decir el sistema no debe presentar irregularidades de interfaz ni de respuestas del servidor en ningún navegador [16].

2 METODOLOGÍA

La descripción de un estudio de casos implica llevar a cabo una indagación minuciosa sobre un tema específico, que puede ser enfocado en estudios de índole social, tecnológico, clínico, educativo, empresarial, etc. Además, involucran métodos cualitativos sin descartar los cuantitativos dependiendo del caso y tiene la característica de ser descriptivo, comparativo y evaluativo con el problema que contiene la investigación [17].

En consecuencia, el presente proyecto utiliza el estudio de casos porque después de una exhaustiva investigación acerca de los beneficios de realizar una actualización de código y las ventajas que trae consigo, se ha realizado una refactorización al sistema *web* de la SDH, con la intención de que pueda ser utilizado únicamente dentro de la organización pública y no con fines comerciales, permitiendo que la gestión de recursos para los Servicios de Protección Integral sea de forma ordenada mediante el uso de tecnologías modernas.

2.1 Metodología de Desarrollo

Algunos enfoques que ayudan en el ámbito de la programación de un proyecto de *software* pueden encontrarse en el campo de la ingeniería. La combinación de procedimientos y métodos organizados que se utilizan para crear soluciones de *software* informático, se conoce como *metodología de software* [8].

La metodología ágil es un enfoque de desarrollo de *software* el cual otorga flexibilidad ya que posibilita volver a un punto del proceso para solventar errores o problemas sin afectar los resultados finales deseados por el cliente. En concreto las metodologías ágiles buscan la colaboración de trabajo en conjunto, dividiendo las tareas en actividades pequeñas con el propósito de alcanzar resultados óptimos y funcionales en un tiempo determinado [8].

Se ha optado por la metodología ágil *Scrum* para la refactorización de este proyecto *web*, porque permite dividirlo en varias tareas que son alcanzables en un período de tiempo corto con resultados aceptables e incrementales conocido como *Sprints*. En ese sentido, un *Sprint* no es más que un ciclo de interacción que se dispone dentro de un proyecto *software*, el cual tiene un ritmo de trabajo ligado a un tiempo no mayor a cuatro semanas, al final de cada *Sprint* se obtiene como resultado un entregable dependiendo del número de actividades que abarca cada iteración, dicho entregable aporta valor al cliente [18], en este caso a los directivos de la Secretaría de Derechos Humanos.

El valor que se aporta a los directivos de la SDH viene ligado con los requerimientos que ellos necesitan para el sistema *web*, escritos en una pila conocida como *Product Backlog*,

artefacto de *Scrum* que contribuye a determinar, seleccionar y ordenar las tareas dentro de un *Sprint*, estas tareas a su vez contribuyen a la construcción de las Historias de usuario que contienen una actividad en específico, la cual es asignada a un programador quien se encarga de desarrollarla dentro de los plazos estipulados. Por último, en *Scrum* también es fundamental conocer sobre el equipo de trabajo como parte de la metodología, este se encarga del desarrollo del proyecto y tiene acciones específicas dependiendo el rol que ocupa. En las siguientes líneas, se detallan los tres roles de *Scrum* que se han establecido para este proyecto.

Roles

Product Owner

Su trabajo es estar pendiente del desarrollo del sistema *software* y asegurarse de que se está construyendo a las especificaciones del cliente al tiempo que garantiza que cada Historia de usuario tiene la funcionalidad que se ha especificado [19]. Para este cargo se encuentra el Ingeniero William Cueva, ya que es el encargado del departamento de TIC y desarrollo de la SDH, además de ser embajador de la idea de que el sistema necesita una refactorización.

Scrum Master

Es responsable de supervisar el progreso del equipo en el transcurso del proyecto y de garantizar que las teorías y normas de *Scrum* se sigan adecuadamente [20]. El Ingeniero Byron Loarte, es asignado a este proyecto como tutor y el encargado de ocupar este cargo, por sus conocimientos en el desarrollo de *software* y ser profesor de la Escuela Politécnica Nacional.

Development team

El equipo de desarrollo está integrado solo por el Sr. Vicente Delgado estudiante de la Escuela Politécnica Nacional, quien es el encargado del desarrollo de las tareas que se han asignado en cada *Sprint* y el cual se compromete a entregar avances de la parte práctica como documentación dentro de un plazo establecido [21]. Por último, en la **Tabla 1** se muestran a cada uno de los roles como parte de *Scrum*.

Tabla 1 Roles de *Scrum*.

ROLES	NOMBRES
<i>Product Owner</i>	Ing. William Cueva.
<i>Scrum Master</i>	Ing. Byron Loarte, MSc.
<i>Development team</i>	Vicente Delgado.

Artefactos

Los artefactos son los componentes y herramientas que proporcionan la apertura y la documentación de cierta información durante el proceso *Scrum* [22]. Estos Artefactos sirven como apoyo fiable durante todo el proceso de refactorización del sistema *web* de gestión de bienes y servicios SPI, por tal motivo se presenta aquellos que han sido empleados en las siguientes líneas.

Recopilación de Requerimientos

Los requerimientos en la metodología *Scrum* son sumamente importantes porque marcan el entendimiento del cómo es el sistema *software* en diseño y funcionalidad una vez que se haya terminado [7]. Para el levantamiento de los requerimientos se han realizado varias reuniones con los directivos de la SDH, de las cuales una fue presencial y varias fueron virtuales y el resultado de las mismas se detalla en la **Tabla 2** en donde se establece el diseño que se ha empleado para recopilar los requerimientos necesarios, por otra parte, la tabla en su totalidad se encuentra expuesta en el **ANEXO II**.

Tabla 2 Formato para la recopilación de requerimientos.

RECOPIACION DE REQUERIMIENTOS		
TIPO DE SISTEMA	ID - RR	ENUNCIADO DEL ITEM
Sistema web	RR-001	Como usuario administrador, encargado y visitante necesitan: <ul style="list-style-type: none"> • Ser registrados por el administrador.
	RR-002	Como usuario administrador, encargado y visitante necesitan: <ul style="list-style-type: none"> • Iniciar sesión. • Cerrar sesión.

Historias de Usuario

Es una herramienta muy útil en diferentes metodologías ágiles, porque contiene un número de identificación, el usuario del sistema al que va dirigido la funcional una vez que sea implementado, nombre de la historia que por lo general es la tarea a ser desarrollada, prioridad del negocio baja, media o alta dependiendo la dificultad de la tarea, responsable, descripción y observaciones [23]. En ese sentido, la **Tabla 3** muestra la Historia Nro. 1 y el formato que se ha empleado en la creación de cada una de ellas, mientras que las tablas restantes pueden ser consultadas en el **ANEXO II**.

Tabla 3 Historia de usuario Nro. 1 - Registro de usuarios.

HISTORIA DE USUARIO	
Identificador: HU001	Usuario: Administrador, encargado y visitante
Nombre historia: Registro de usuarios.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable: Vicente Delgado	
Descripción: El usuario administrador, encargado y visitante necesitan ser registrados por un usuario con perfil administrador en el sistema <i>web</i> .	
Observación: Los usuarios con perfil administrador tienen la potestad de registrar al usuario encargado y visitante en el sistema a través de la creación de un nombre (<i>nickname</i>) y contraseña que les permita tener acceso al sistema.	

Product Backlog

El artefacto de uso indispensable es el *Product Backlog* debido a que es un inventario de los requerimientos del sistema *software* que fueron trasladados a Historias de usuario, con el fin de llevar un registro constante de los cambios que pueden ocurrir en las tareas o en los nuevos requerimientos [24]. El propietario de este artefacto es el *Product Owner*, el Ingeniero William Cueva que es el encargado de gestionar directamente los requerimientos con los directivos y funcionarios de SDH de la cual surge la **Tabla 4** en la cual se establece el diseño que se ha empleado para presentar el *Product Backlog*, por otra parte, la tabla en su totalidad se presenta en el **ANEXO II**.

Tabla 4 Formato del *Product Backlog*.

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU001	Registro de usuarios.	1	Finalizada	Alta
HU002	Iniciar y cerrar sesión.	1	Finalizada	Alta

Sprint Backlog

Es el artefacto y herramienta por excelencia en las metodologías ágiles porque se describe todas las pequeñas tareas de una Historia de usuario, donde se visualiza también todo el trabajado que se ha realizado por los programadores dentro de un determinado *Sprint*. Además, permite entender cuál es la evolución del proyecto, así como hacer un análisis de riesgos [24]. Por tal motivo, la **Tabla 5** presenta el diseño que se ha utilizado para crear el *Sprint Backlog*, además, la tabla completa se encuentra detallada en el **ANEXO II**.

Tabla 5 Formato del *Sprint Backlog*.

ELABORACIÓN DE <i>SPRINT BACKLOG</i>						
ID – SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREA	TIEMPO ESTIMADO
SB000	Configuración del entorno de desarrollo	N/A	N/A	N/A	<ul style="list-style-type: none"> • Instalación de herramientas de desarrollo. • Configuración de PostgreSQL. • Creación del proyecto Maven Sprint Boot. • Creación del proyecto Angular. 	10H

2.2 Diseño de interfaces

Es esencial que en una correcta estructura de un sistema de *software* se disponga de un prototipo o diseño preliminar que ayude a entender de mejor manera como se encuentra constituido visualmente [25]. Para este proyecto los dirigentes y funcionarios de la SDH decidieron por un diseño no muy vistoso, ya que principalmente se centraron en que sea funcional, entendible y que cumpla con los requisitos al momento de usarlo en producción. En ese sentido, se presenta la herramienta para la elaboración del diseño.

Herramienta utilizada para el diseño

Balsamiq Mockups

La herramienta elegida para realizar el prototipo o *mockups* es *Balsamiq Mockups*, porque es una aplicación de escritorio que contribuye al trabajo continuo sin contar con conexión a Internet, también ofrece un ambiente amigable para el usuario con elementos minimalistas de rápido acceso y un ambiente 100% colaborativo [26]. Esta herramienta facilita una vista a las páginas diseñadas con iconos sencillos y prácticos de entender no solo para el diseñador sino también para los directivos de la SDH.

Diseño sistema web

Cuando se habla de los diseño para un sistema *software* es necesario saber que estos deben cumplir con un parámetro muy importante la dimensión a la que se deben adaptar según la resolución y pixeles de cada pantalla, porque los componentes de la página deben verse siempre uniformes y ordenados sin tomar en cuenta la dimensión de la pantalla, en la **Figura 2.1** se visualiza el diseño de la página inicial, mientras que los demás prototipos están incluidos en el **ANEXO II**.

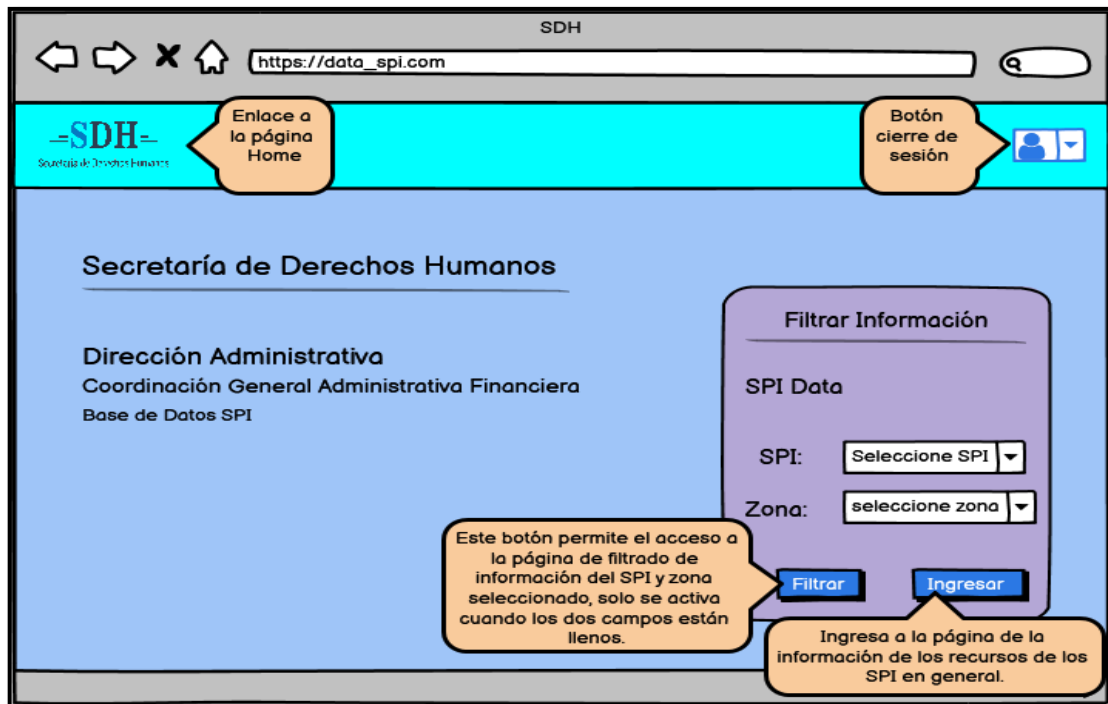


Figura 2.1 Prototipo de Interfaces – Pagina Inicial del sistema *web* SDH.

2.3 Diseño de la arquitectura

En la época presente un sistema de *software* que disponga de una arquitectura de datos sólida es de suma importancia, porque maneja información específica que debe ser tratada con cuidado. Para ello se decide trabajar con el Modelo Vista Controlador (MVC), como pilar fundamental para la refactorización del sistema *web* de la SDH.

Patrón arquitectónico

El Modelo Vista Controlador (MVC) del sistema *web* de la SDH se divide en tres componentes, para mejorar la eficacia de la administración de datos, la lógica de control y la interfaz de usuario son implementadas en las respuestas de los usuarios y las peticiones respectivamente [27]. Para la integración del patrón arquitectónico se utiliza *Spring MVC* con cada uno de los componentes antes mencionado como se puede apreciar de una forma mucho más clara y compacta en la **Figura 2.2**.

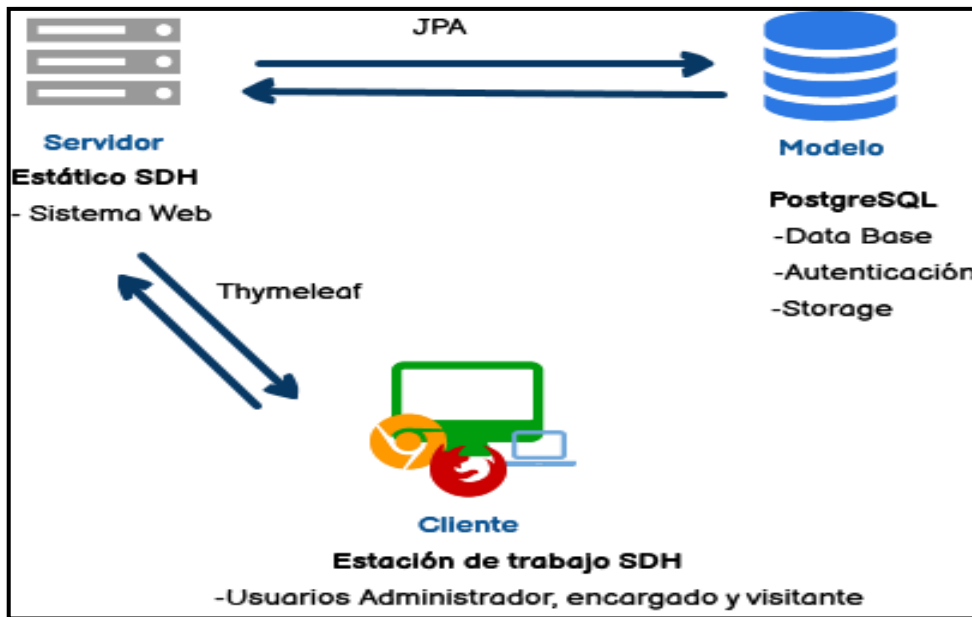


Figura 2.2 Modelo MVC del sistema web de la SDH.

2.4 Herramientas de desarrollo

Para la configuración del entorno de trabajo y sobre todo que sea *Open Source*, es necesario ocupar herramientas que brinde compatibilidad y garanticen escalabilidad, por tal motivo en la **Tabla 6** se muestra el nombre de la herramienta y la justificación del porqué su uso e integración.

Tabla 6 Herramientas para la refactorización del sistema web.

HERRAMIENTA	JUSTIFICACIÓN
Spring Tools	Agiliza el trabajo de codificación por ser un IDE de desarrollo especializado en el ambiente de <i>Spring Boot</i> [28].
Spring Boot	Permite trabajar con dependencias, clases y controladores, todo contenido en un paquete Maven vista-controlador [14].
PostgreSQL	Permite escalabilidad en la base de datos a largo plazo. Además, es una herramienta de licencia libre para trabajar en ambientes empresariales [15].
PowerDesigner	Simplifica la creación de un modelo de base de datos y resulta altamente provechoso en la modificación de una Base de datos relacional [29].

Thymleaf	Concede la compatibilidad con la mayoría de los navegadores, por ser un motor de plantilla en aplicaciones java <i>web</i> [12].
Bootstrap	Ofrece elementos de Html5 muy fáciles de implementar y sobre todo enfocados a ambientes empresariales [30].

Librerías

En la **Tabla 7** se listan las librerías que han sido utilizadas para la refactorización del sistema *web*.

Tabla 7 Librerías empleadas para la refactorización del sistema *web*.

LIBRERÍA	JUSTIFICACIÓN
<i>Spring Boot Dev Tools</i>	Brinda la ayuda necesaria en la configuración y ejecución de nuevos servicios eficazmente [31].
<i>Spring Web</i>	Facilita el soporte y desarrollo de páginas <i>web</i> basadas en Java [32].
<i>Spring Data JPA</i>	Agiliza las consultas con un gestor de almacenamiento de datos [33].
<i>Spring Security</i>	Minimiza la implementación de autenticación, seguridad y <i>middlewares</i> en el sistema <i>web</i> [34].
PostgreSQL Driver	Conecta el gestor de almacenamiento de datos con el sistema <i>web</i> [35].

3 RESULTADOS

En esta sección se exponen los resultados de cada *Sprint* en la refactorización de cada componente del sistema *web* de la SDH, junto con las funcionalidades requeridas. Tomando en cuenta que el *Sprint Backlog* está dividido en seis *Sprints*, cinco principales y uno de configuración del ambiente de desarrollo.

3.1 *Sprint* 0. Configuración del ambiente de desarrollo.

Este *Sprint* a pesar de no ser tomado como principal es necesario porque aquí se establece las condiciones del ambiente de desarrollo, es decir el área de trabajo del programador, con respecto a las herramientas de *software* que se han empleado en la refactorización del sistema *web*, así como otros aspectos de relevancia que se describen a continuación:

- Recopilación de requerimientos para el sistema *web*.
- Creación del proyecto Maven *Spring Boot*.
- Configuración de PostgreSQL para la Base de datos SQL.
- Asignación de roles a los usuarios del sistema *web*.

Recopilación de requerimientos para el sistema *web*

Verificación de usuarios

Los usuarios del sistema con perfil administrador, encargado y visitante pueden acceder al sistema *web* mediante un inicio de sesión creado a partir de la dependencia *Spring Security* que se encarga de validar en el sistema el usuario y contraseña una vez que son ingresados en el formulario correspondiente permitiendo el ingreso exitoso o fallido.

Gestión de registro de usuarios

Solamente el usuario que ostenta el rol de administrador tiene la capacidad de administrar los registros y roles de los usuarios, ya sean administradores, encargados o visitantes en el sistema *web*. Esto implica que únicamente el usuario con el rol de administrador está autorizado para registrar un nuevo usuario y asignarle tanto un rol como una contraseña.

Gestión de unidades e instituciones

Solamente el usuario que posee el rol de administrador está autorizado para manejar la información relacionada con las tablas de unidades e instituciones.

Gestión de bienes SPI

Tanto los usuarios con el rol de administrador como los encargados cuentan con la capacidad de manejar la información referente a bienes (activos materiales de los SPI), esta información solo puede ser creada y editada, mientras que el borrado es lógico y no permanente así se tiene un histórico de la información.

Gestión de recursos humanos

Tanto los roles administradores como encargados son capaces de administrar la información de los recursos humanos de los SPI (personal y funcionarios que laboran en las diferentes oficinas SPI del país). Tomando en cuenta que solo pueden crear y editar dicha información.

Por último, tomando en consideración lo antes mencionado y la recopilación de requerimientos, la **Figura 3.1** muestra la distribución de las tareas que cada rol tiene la capacidad de llevar a cabo al acceder al sistema *web*.

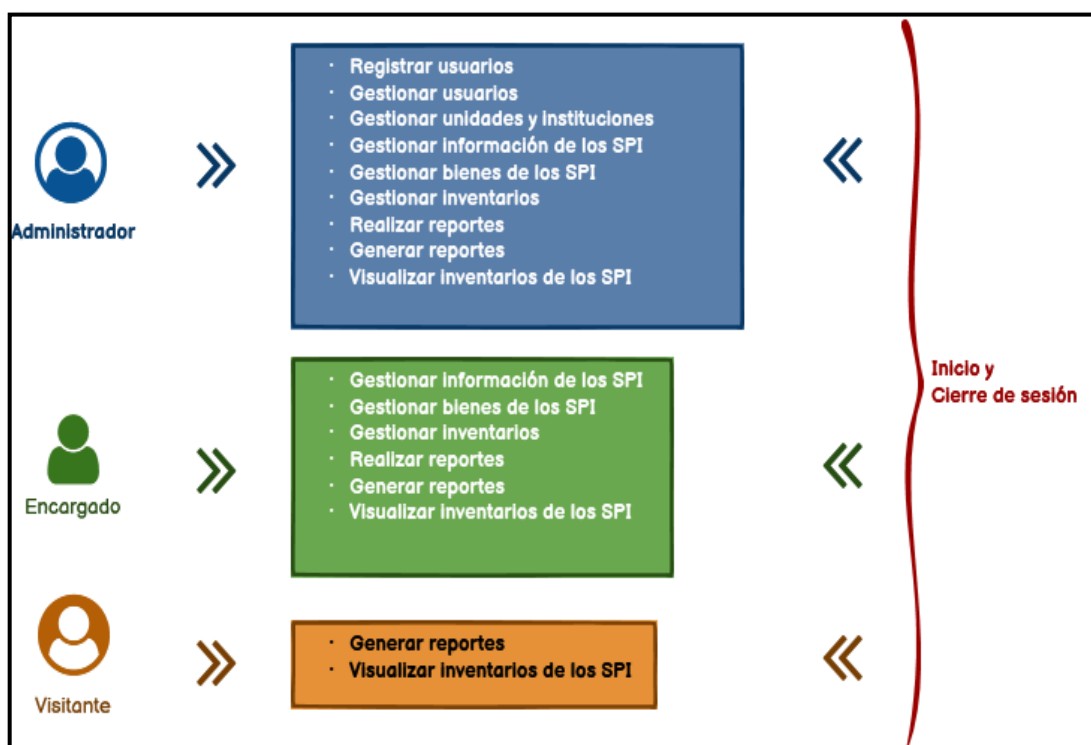


Figura 3.1 Tareas asignadas al administrador, encargado y visitante.

Configuración de PostgreSQL para la Base de datos SQL.

En la configuración y refactorización la Base de Datos de la SDH se agregando dos tablas User y Roles, necesarias en el manejo de usuarios del sistema, debido a que el sistema anterior no contaba con un sólido inicio de sesión. La **Figura 3.2** se plasma las dos entidades, mientras que el modelo completo se remite al **ANEXO II**.

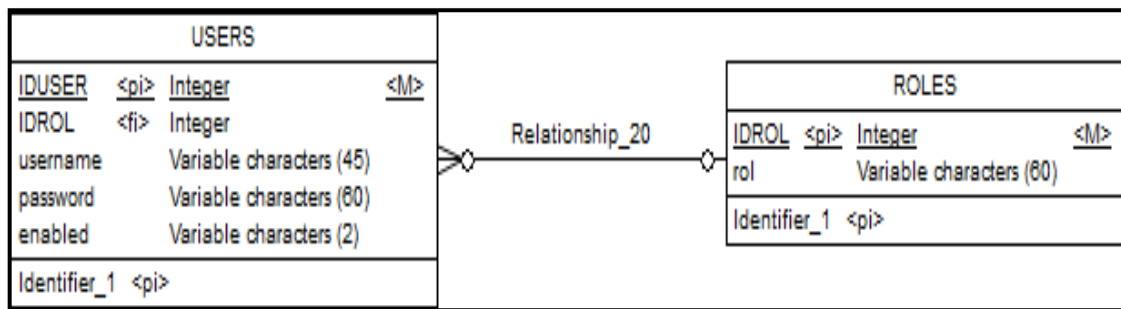


Figura 3.2 Refactorizado de la Base de Datos

Creación del proyecto Maven *Spring Boot*

Un proyecto Maven *Spring Boot* es un paquete .jar que contiene los componentes necesarios para crear un sistema *web*. Al referirse como componentes se menciona las dependencias para utilizar PostgreSQL, *Spring Web*, *Spring Security*, *Spring Boot Dev Tools* y *Thymeleaf* desde la ejecución de un solo comando. Muestra de ello en la **Figura 3.3** se presenta las carpetas de contiene el paquete Maven con todas las demencias requeridas.

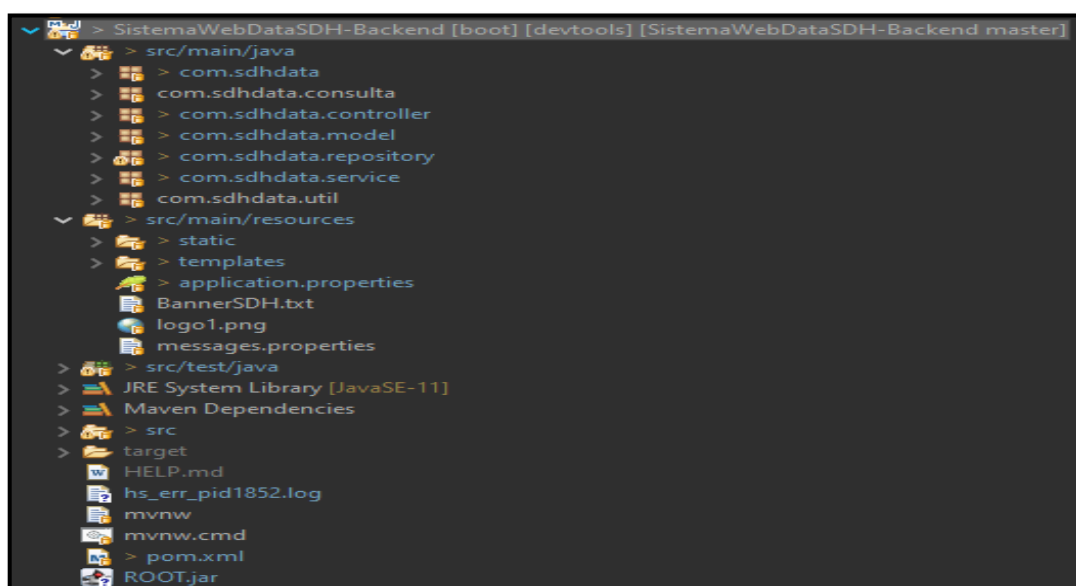


Figura 3.3 Estructura del paquete Maven.

Asignación de roles a los usuarios del sistema web

Dentro del sistema web de la gestión de bienes de la SDH se han designado tres roles administrador, encargado y visitante con el fin de optimizar la gestión de la información y cumplir con lo establecido por el dueño del producto, en la **Figura 3.4** se visualiza las funciones que pueden realizar cada uno de ellos.

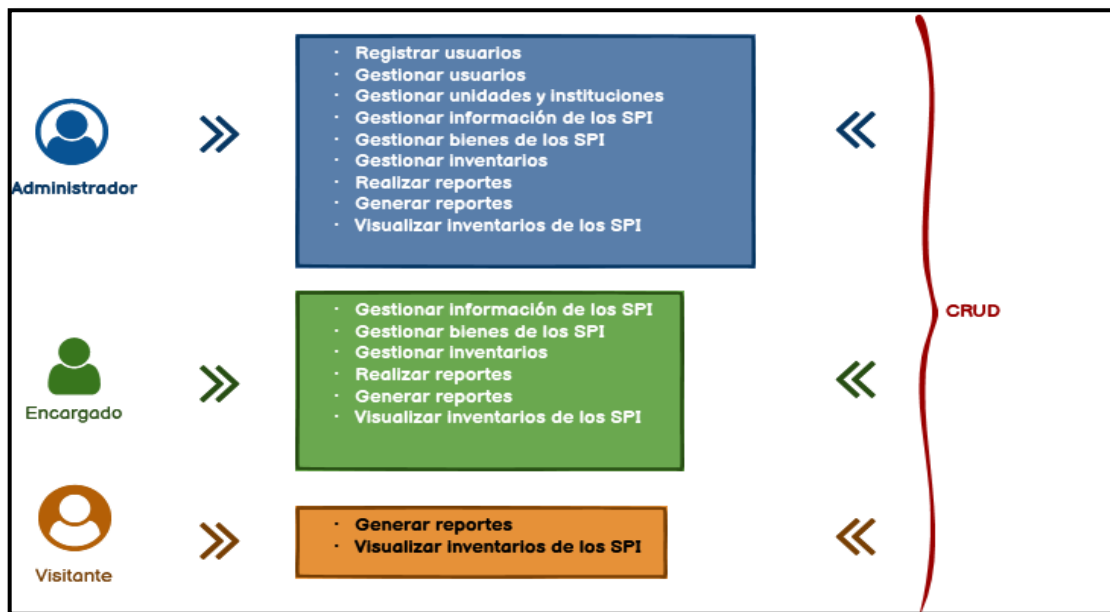


Figura 3.4 Asignación de roles y funciones de los usuarios.

3.2 *Sprint* 1. Módulos refactorizados para el administrador.

Tal como se indica en el *Spring Backlog*, el *Sprint 1* engloba las siguientes actividades:

- Registro de usuarios.
- Inicio y cierre de sesión.
- Modificar usuarios.
- Modificar unidad, instituciones y roles.

Registro de usuarios

La página de registro consta de un formulario simple de tres campos usuario, contraseña y rol tal como se presenta en la **Figura 3.5** necesarios para la creación de un nuevo usuario, mientras que el resultado de la prueba unitaria se muestra en la **Figura 3.6** y para las validaciones de los campos, así como la funcionalidad a detalle se presenta en el **ANEXO III**.

Figura 3.5 Formulario de registro de usuarios.

Figura 3.6 Pruebas unitarias registro de usuarios.

Inicio y cierre de sesión

El inicio de sesión es una página, donde se encuentra un formulario sencillo de dos campos usuario y contraseña, credenciales que el administrador entrega al usuario dependiendo del rol encargado o visitante. En la **Figura 3.7** se puede presenciar el formulario *login*, junto con el nombre de usuario que ha accedido al sistema y un botón para el cierre de sesión como se puede presenciar en la **Figura 3.8**, adicionalmente en la **Figura 3.9** se exhibe el resultado de la prueba unitaria y para las validaciones de los campos. La descripción detallada de la funcionalidad se encuentra en el **ANEXO III**.

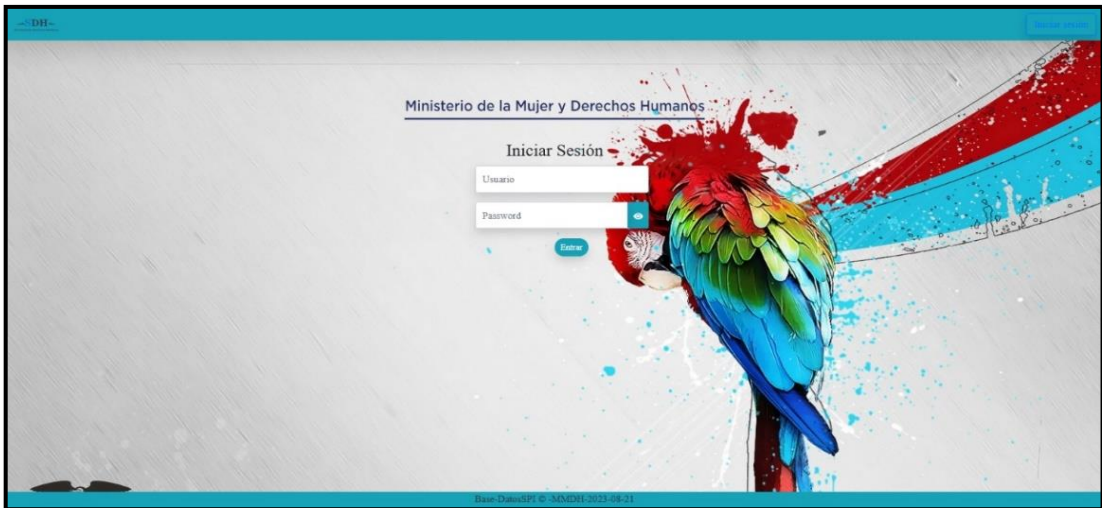


Figura 3.7 Login de usuarios.



Figura 3.8 Cierre de sesión de usuarios.

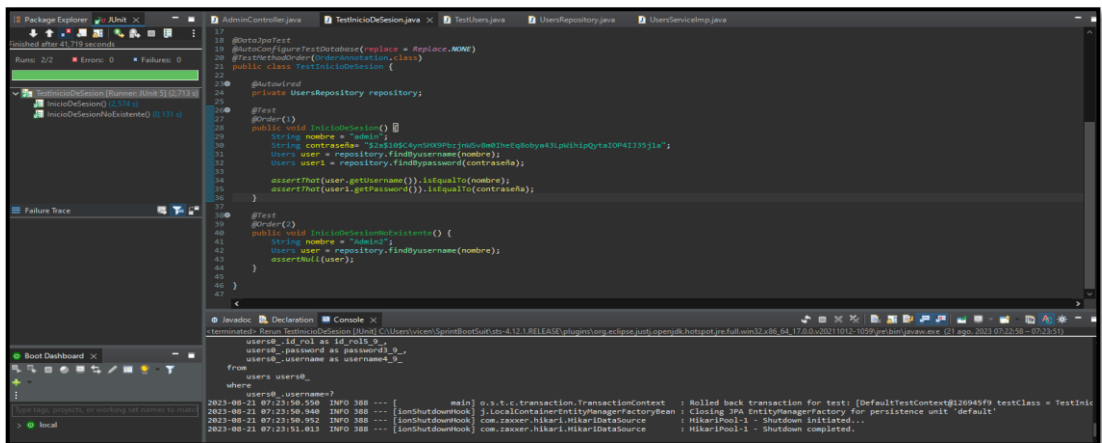


Figura 3.9 Resultado de la prueba unitaria.

Modificar usuarios

La página modificación de usuarios contiene una tabla listando los usuarios que son parte del sistema web, que solamente puede ser editada por el rol administrador, los resultados se visualizan en la **Figura 3.10**. El formulario para editar es desplegado en un modal tal como se observa en la **Figura 3.11**, mientras que la prueba unitaria con su resultado se muestra en la **Figura 3.12** y para obtener más información sobre las validaciones de campos y la funcionalidad detallada, se remite al **ANEXO III**.

Página: Registro de Usuarios
Lista de usuarios del sistema

Crear nuevo usuario

Mostrar 10 Registros

Buscar:

ID	EDITAR	ELIMINAR	Usuario	Rol
1	<input type="checkbox"/>	<input type="checkbox"/>	admin	ROLE_ADMIN
2	<input type="checkbox"/>	<input type="checkbox"/>	User	ROLE_ATTEND
3	<input type="checkbox"/>	<input type="checkbox"/>	Visitor	ROLE_VISITOR
6	<input type="checkbox"/>	<input type="checkbox"/>	David	ROLE_ADMIN
8	<input type="checkbox"/>	<input type="checkbox"/>	Sandra	ROLE_ATTEND

Mostrando registros del 1 al 5 de un total de 5 registros

Anterior 1 Siguiente

Figura 3.10 Página registro y modificación de usuarios.

Formulario: Editar Usuario

Usuario:

Contraseña:

Rol:

Cancelar Guardar cambios

Figura 3.11 Formulario editar perfil de usuario.

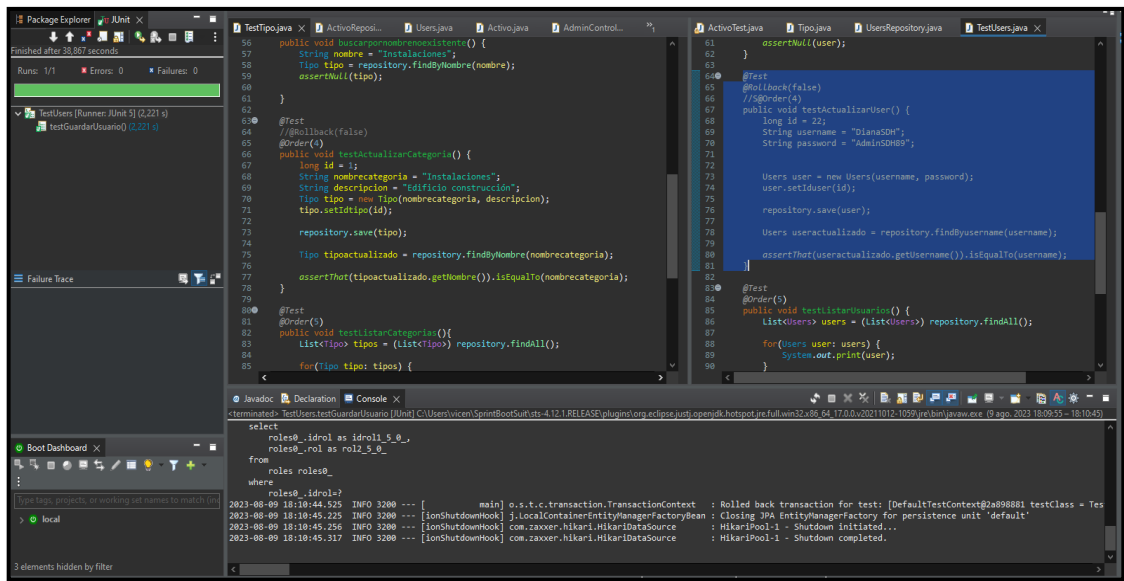


Figura 3.12 Resultado de prueba unitaria.

Modificar unidad, instituciones y roles

La información de unidad, instituciones se presentan únicamente al administrador, quien puede modificar la información a través de páginas que contienen una tabla para hacer el CRUD respectivo, los formularios son desplegados por un modal activado por un botón que se encuentra en dicha tabla, tal como se presenta en la **Figura 3.13** el cual presenta el resultado de página de instituciones, mientras que el resultado de la prueba unitaria se muestra en la **Figura 3.14** y para las validaciones de los campos, así como la funcionalidad a detalle, se remite al **ANEXO III**.



Figura 3.13 Página para listar las instituciones.

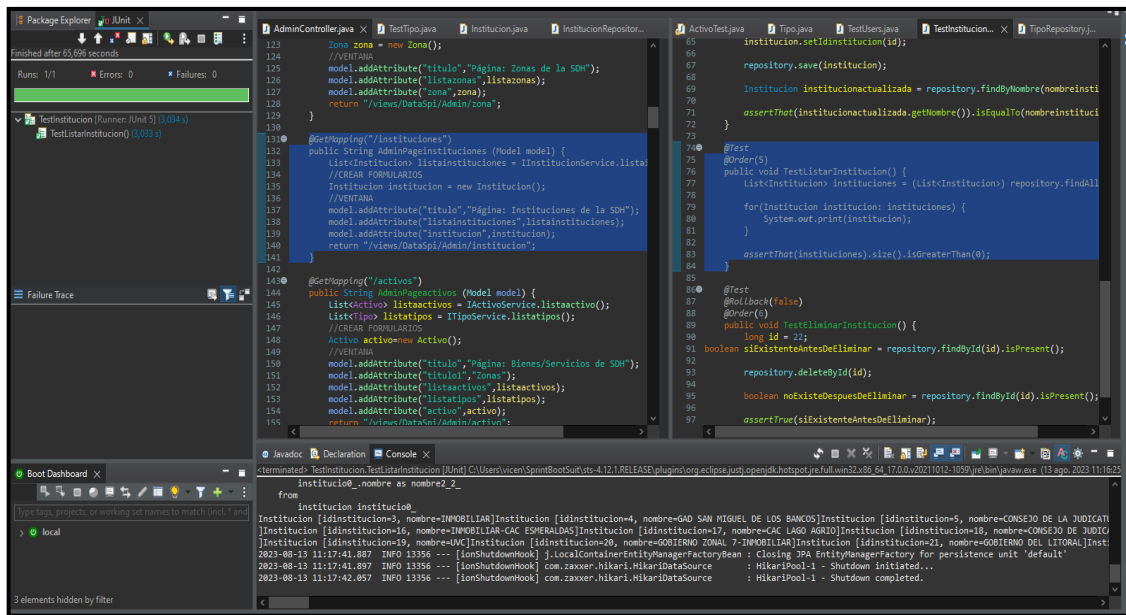


Figura 3.14 Resultados prueba unitaria de la página para listar instituciones.

3.3 Sprint 2. Módulos refactorizados para el encargado.

Tal como se indica en el *Spring Backlog*, el *Sprint 2* engloba las siguientes actividades:

- Categorías.
- Gestionar bienes de los SPI.
- Gestionar recursos humanos de los SPI.
- Gestionar inventario por SPI.
- Generar reportes de los SPI.
- Subir verificables.

Categorías

La clasificación de categorías es generada por el sistema gracias a una consulta *Query* la misma que se obtiene gracias a un método GET, presentado en la página inventario a través de 6 tablas, siendo muy fácil de usar para el usuario encargado o administrador, en la **Figura 3.15** se observa las categorías de instalaciones y bienes con sus respectivas tablas, mientras que el resultado de la prueba unitaria se visualiza en la **Figura 3.16**, la funcionalidad a detalle se presenta en el **ANEXO III**.

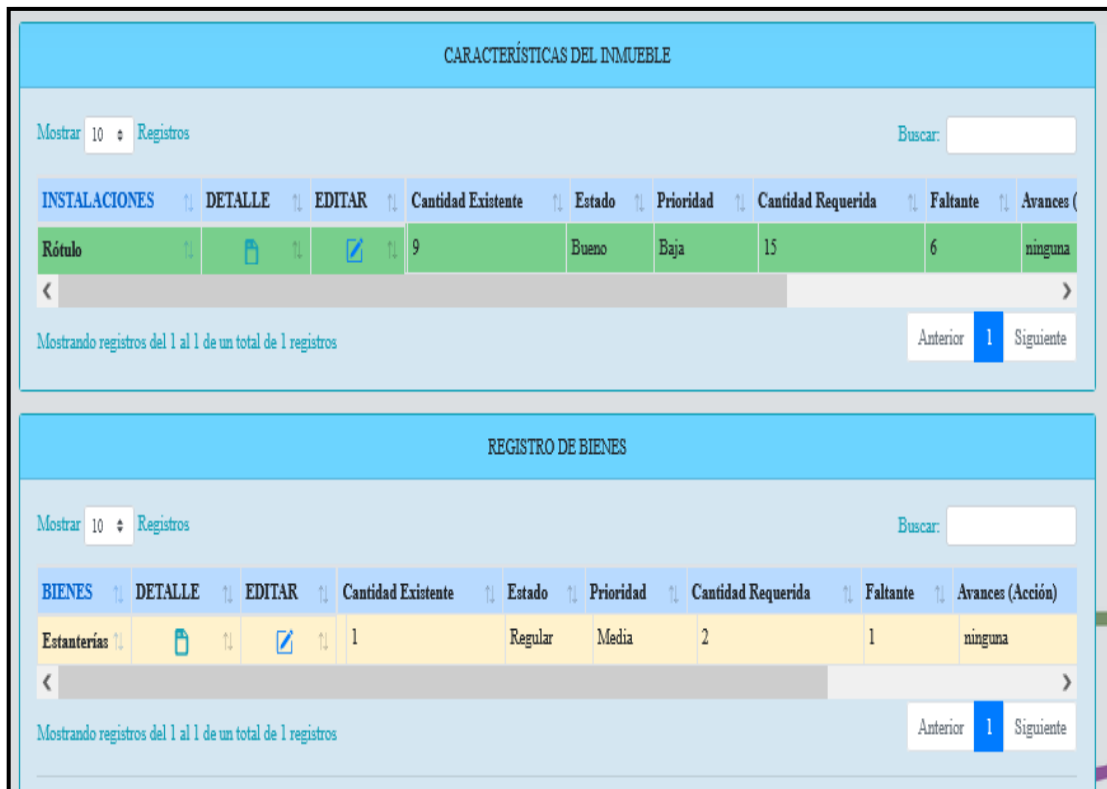


Figura 3.15 Categorías.

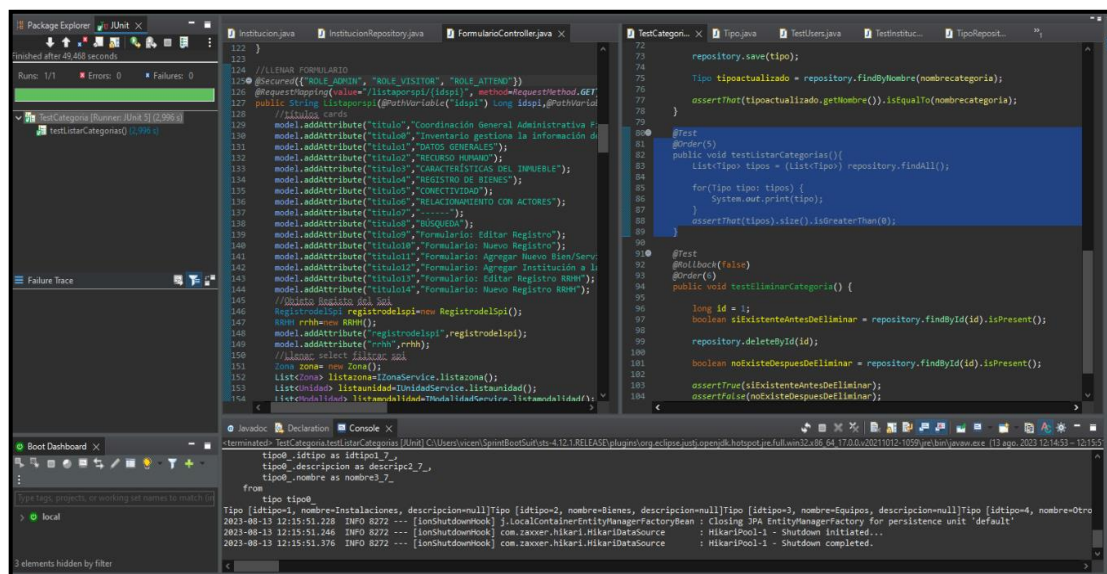


Figura 3.16 Prueba unitaria para listar categorías.

Gestionar bienes de los SPI

La información de los bienes de los SPI se maneja con la ayuda de una tabla para realizar el CRUD de forma universal, es decir de todos los bienes de los SPI como se visualiza en

la **Figura 3.17**, mientras que el resultado de la prueba unitaria se muestra en la **Figura 3.18** y para las validaciones de los campos, así como la funcionalidad detallada, se remite al **ANEXO III**.

DETALLE	EDITAR	Bien/Servicio	Zona	SPI	Pertenencia bien/servicio	Estado	Cantidad existente	Cantid
2	[icon]	Baño	Zona 1	Ibarra	MIES	Bueno	2	2
3	[icon]	Anaqueles	Zona 1	Ibarra	SDH	Malo	2	4
4	[icon]	Parquedero	Zona 1	Ibarra	SDH	Regular	1	1
5	[icon]	Rótulo	Zona 1	Ibarra	FISCALÍA	Malo	0	1
6	[icon]	Cuadro misión-visión	Zona 1	Tulcan	INMOBILIAR	Bueno	1	1
7	[icon]	Servicio de limpieza	Zona 1	Tulcan	SDH	Regular	1	1
8	[icon]	Anaqueles	Zona 1	Tulcan	CONSEJO DE LA JUDICATURA	Malo	1	1
9	[icon]	Papelera	Zona 1	Tulcan	SDH	Bueno	1	1
10	[icon]	Computadoras	Zona 1	Tulcan	GAD RUMIÑAHUI	Regular	3	5
11	[icon]	Impresoras	Zona 1	Tulcan	SDH	Regular	1	2

Figura 3.17 Página para la gestión de bienes de los SPI.

```

TestRegistroDeSpi (Runner: JUnit 5) (0.749 s)
  testGuardarRegistroDeSpi (4.361 s)
  testBuscarPorArchivo (0.763 s)
  testBuscarPorArchivoNoExistente (0.239 s)
  testActualizarRegistroDeSpi (1.331 s)
  testListaRegistrosDeSpi (1.003 s)
  Failure Trace
  Javadoc Declaration Console
  terminated: TestRegistroDeSpi [JUnit] C:\Users\vicen\SpringBoot\sts-4.12.1\RELEASE\plugins\org.eclipse.jdt.junit.hotspot.jre.full.win32.x86_64.17.0.0.v20211012-1059\re\bin\javaw.exe (14 ago. 2023 03:22:45 - 03:23:38)
  institution@_idinstitution=
  where
  testException = [null], mergedContextConfiguration = [MergedContextConfiguration@796caf7a testClass = TestRegistroDeSpi, locations = '{}', classes = '{class com.sdhwdata.Sistemae
  2023-08-14 03:23:57.715 INFO 12868 --- [IonShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
  2023-08-14 03:23:57.727 INFO 12868 --- [IonShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
  2023-08-14 03:23:57.862 INFO 12868 --- [IonShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
  
```

Figura 3.18 Resultados de la prueba unitaria gestión de bienes/servicios de los SPI.

Gestionar recursos humanos de los SPI

La información del recurso humano de igual forma se la maneja con la ayuda de una tabla para realizar el CRUD en una página clasificada por zonas, también se la puede filtrar por SPI para facilitar la búsqueda a los usuarios con perfil encargado y administrador, el resultado para gestionar los recursos humanos (RRHH) se observa en la **Figura 3.19**, mientras que el resultado de la prueba unitaria es visualizado en la **Figura 3.20** y para las validaciones de los campos, así como la funcionalidad a detalle son presentados en el **ANEXO III**.



Figura 3.19 Página de gestión del RRHH de los SPI.

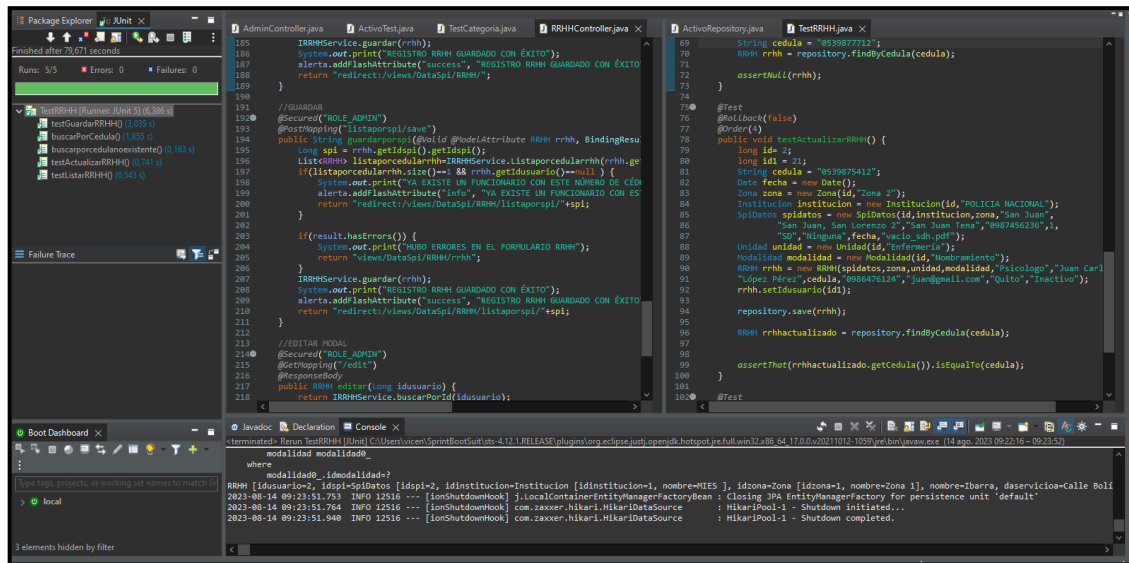


Figura 3.20 Resultado de las pruebas unitarias gestión del RRHH de los SPI.

Gestionar inventario por SPI

El inventario de bienes, servicios y recurso humano se maneja a través de una página, que segrega la información por SPI organizada en categorías, gracias a un filtrado de dos campos zona y SPI, también contiene complementos (botones) que habilitan los formularios correspondientes para realizar un CRUD en específico, agregar, editar y detalle, como se observa en la **Figura 3.21** se presenta la página y sus elementos, mientras que el resultado de la prueba unitaria se muestra en la **Figura 3.22** y para las validaciones de los campos, así como la funcionalidad detallada, se remite al **ANEXO III**.

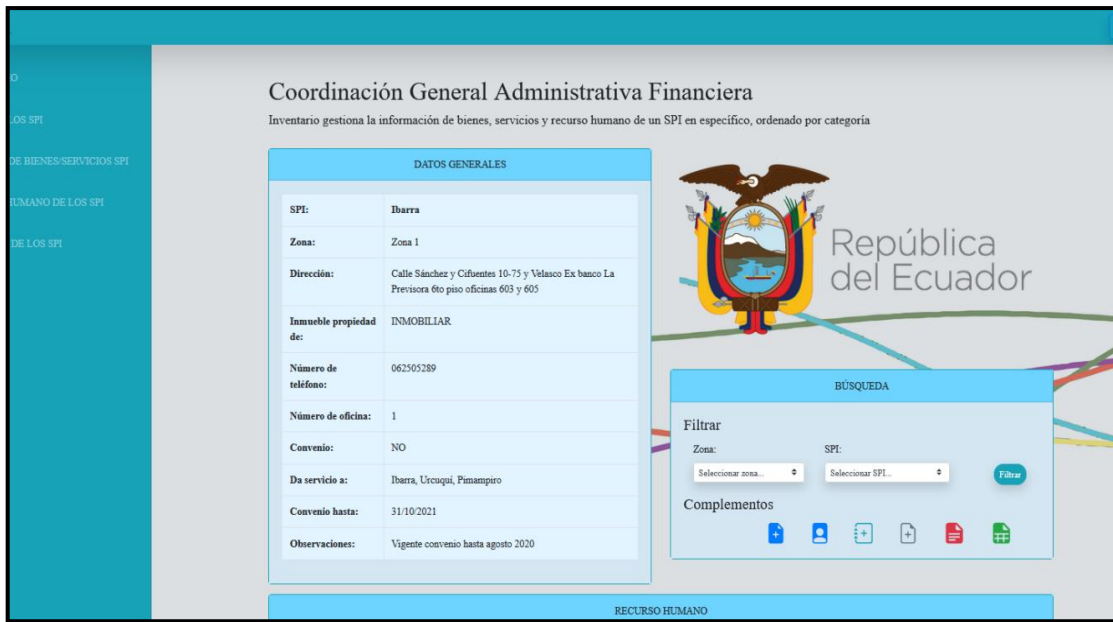


Figura 3.21 Página de inventario.

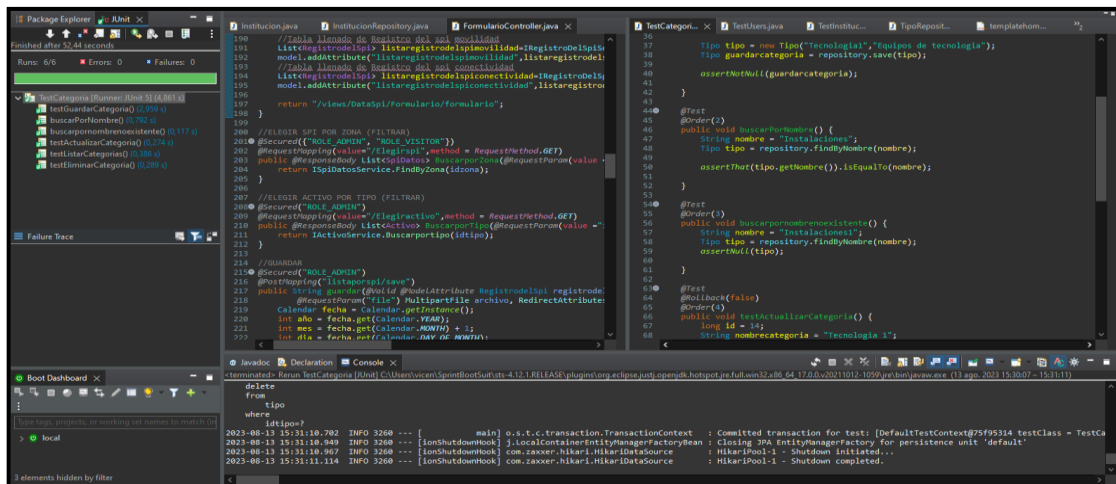


Figura 3.22 Pruebas unitarias de inventario.

Generar reportes de los SPI

Cada página contiene 2 botones un rojo y un verde como se muestra en la **Figura 3.23**, el botón de color rojo genera un reporte en formato PDF y el botón de color verde descarga la información de la página en formato Excel. Respecto al resultado de la prueba unitaria se plasma en la **Figura 3.24** y para las validaciones de los campos, así como la funcionalidad, están proporcionados en el **ANEXO III**.



Figura 3.23 Generación de reportes.

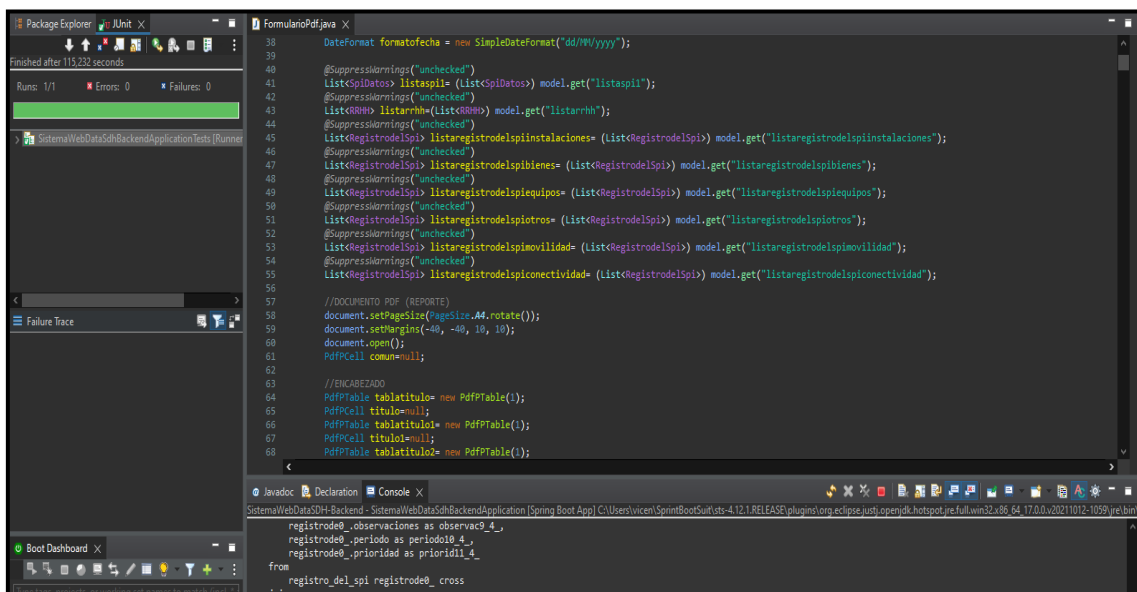


Figura 3.24 Prueba unitaria para la generación de reportes con el sistema.

Subir verificables

Los verificables se suben al sistema *web* mediante los formularios que contiene un campo de manejo de archivos (campo opcional si existe verificable), como se observa en la **Figura 3.25**, la previsualización y descarga de los mismos se presenta en la página detalle cómo se puede observar en la **Figura 3.26**. Respecto al resultado de la prueba unitaria se plasma en la **Figura 3.27** y para las validaciones de los campos, así como la funcionalidad detallada, se presenta en el **ANEXO III**.

Formulario: Editar Registro

Pertenencia bien/servicio: MIES Estado: Bueno

Cantidad existente: 2 Cantidad requerida: Ingrese la cantidad requerida del bien

Faltante (-)Sobrante: -2 Prioridad: Baja

Avances (acción): Entregados Fecha de acción: 18 / 08 / 2022

Verificable: Examinar... No se ha seleccionado ningún archivo.
Archivos en formato pdf, jpg y png

Observaciones: Ninguna

Guardar Cambios

Figura 3.25 Formulario de página para verificable.

Ver Detalle: Chompa del SPI Copotaxi

Bien/Servicio: Chompa

SPI: Copotaxi

Pertenencia del Bien/Servicio: INMOBILIAR

Cantidad existente: 5

Cantidad requerida: 10

Faltante: 5

Estado: Bueno

Prioridad: Media

Acción tomada: acción

Periodo: 2023

Fecha de acción: 20/06/2023

Verificable: vacio_sdh.pdf

Observaciones: Sin observaciones

=SDH=
Secretaría de Derechos Humanos

*Este Registro
no tiene
archivo adjunto*

Figura 3.26 Previsualización y descarga de verificables.

Carga de archivos

Organizar Nueva carpeta

OneDrive - Personal

Este equipo

Descargas

Documentos

Escritorio

Imágenes

Música

Objetos 3D

Videos

Disco local (C:)

Red

gallery

pc

prueba heroku

PRUEBAS

Acordes para guitarra (principales).jpg

DGIP_SAEw_InformacionPagos2023.pdf

DGIP_SAEw_Pagare02023.pdf

harv-eker--secretos-de-la-mente-millonaria.pdf

Nombre de archivo: Todos los tipos admitidos (*.gif, ...)

Abrir Cancelar

Seleccionar Institución...

Seleccionar SPI...

Ingrese la cantidad actual del bien

Ingrese el desfaz de la cantidad requerida

Acción tomada sobre el bien o servicio

dd / mm / aaaa

Verificable: Examinar... No se ha seleccionado ningún archivo.
Archivos en formato pdf, jpg y png

Observaciones: Ingrese observaciones sobre el SPI

Cancelar Guardar

Computadoras Zona 1 Tulcan GAD RUMISAHUI Regular 3

Figura 3.27 Resultado de la prueba unitaria para subir verificables.

3.5 *Sprint* 4. Pruebas del sistema web refactorizado.

Tal como se indica en el *Spring Backlog*, el *Sprint 4* engloba las siguientes actividades:

- Pruebas unitarias.
- Pruebas de compatibilidad.
- Pruebas de aceptación.

Pruebas unitarias

Al terminar la programación de un módulo, se realizan las pruebas unitarias que es un testeo de la funcionalidad del código en pequeños fragmentos, es decir permiten evaluar un método java para comprobar que los resultados sean los deseados después de su ejecución [36]. En la implementación de estas pruebas se utilizan la herramienta de JUnit que está diseñada para el testeo de proyectos Maven en java. En ese sentido, en la **Figura 3.30** se ilustra un fragmento del código para crear una nueva categoría con el perfil administrador, en la **Figura 3.31** se expone el resultado de la prueba y el en **ANEXO II** se encuentran el resto de las pruebas unitarias.

```
@Autowired
private SpiDatosRepository repository;

@Test
@Rollback(false)
@Order(1)
public void testGuardarSpiDato() {
    long id = 2;
    Date fecha = new Date();
    Zona zona = new Zona(id,"Zona 2");
    Institucion institucion = new Institucion(id,"POLICIA NACIONAL");
    SpiDatos spidatos = new SpiDatos(id,institucion,zona,"San Vicente",
        "San Vicente, San Jorge 3","San Vicente Morona","0977452631",1,
        "SD","Ninguna",fecha,"vacio_sdh.pdf");
    SpiDatos guardarspidatos = repository.save(spidatos);

    assertNotNull(guardarspidatos);
}
```

Figura 3.30 Código de prueba crear SPI.

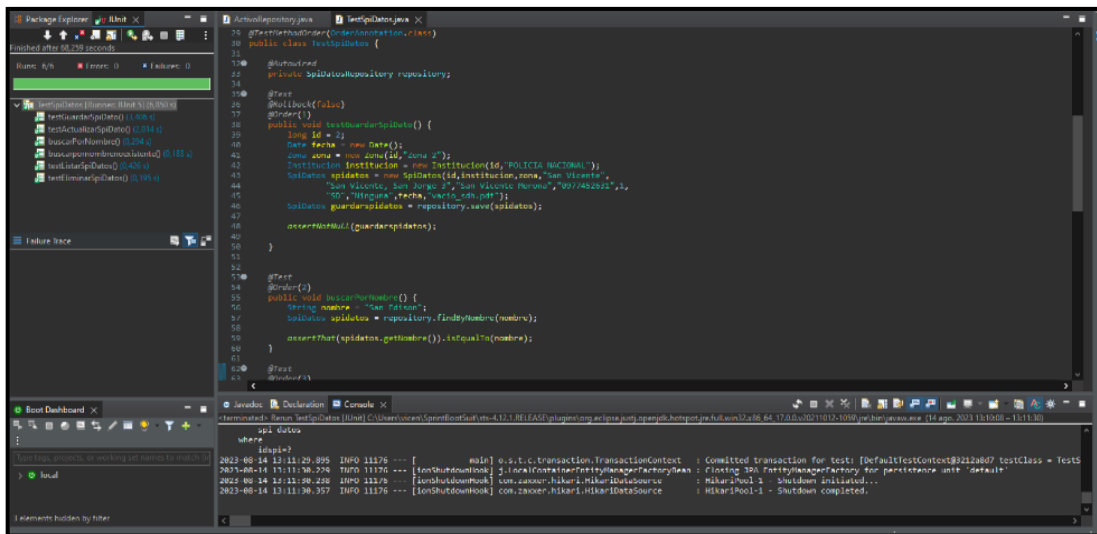


Figura 3.31 Resultados de la prueba unitaria.

Los resultados que se han obtenido en esta prueba unitaria revelan un desempeño aceptable por parte de los módulos que han sido refactorizados adecuadamente.

Pruebas de compatibilidad

Este tipo de prueba consiste en comprobar el correcto desempeño del sistema *web* en diferentes navegadores y versiones que generalmente son usados, priorizando la interfaz de usuario y la funcionalidad al mismo tiempo, en su mayoría estas pruebas son realizadas por el equipo de calidad de *software* y por el equipo de desarrollo quienes realizan estas pruebas en los navegadores de diferentes dispositivos con varias dimensiones de pantallas [37]. En ese sentido, para estas pruebas se han utilizado varias plataformas de buscadores *web*, en la **Tabla 8** se aprecia los distintos navegadores con sus respectivas versiones, en la **Figura 3.32** se observa una interfaz del sistema *web* en el navegador Opera y en el **ANEXO II** se encuentran todos los resultados de compatibilidad con otras plataformas *web*.

Tabla 8 Pruebas de compatibilidad – navegadores *web*.

NOMBRE	VERSIÓN	OBSERVACIÓN
Firefox	114.0.2	Funcional
Google Chrome	114.0.5735.135	Funcional
Microsoft Edge	114.0.1823.58	Funcional
Brave	1.52.130	Funcional
Opera	100.0.4815.30	Funcional



Figura 3.32 Resultados de compatibilidad con el navegador Opera.

Al concluir con las pruebas de compatibilidad se puede afirmar que el sistema *web* de la SDH es compatible en interfaz y contenido con varias plataformas *web*, así se garantiza la portabilidad y el correcto funcionamiento.

Pruebas de aceptación

La meta primordial de las pruebas de aceptación es asegurarse de que el sistema de la SDH funcione de acuerdo con las especificaciones y requisitos que se han establecido y que los usuarios finales puedan interactuar con él de manera eficaz. Las pruebas de aceptación abordan diferentes aspectos del sistema *web* como funcionalidad, usabilidad, seguridad y el rendimiento [38]. En la **Tabla 9** se puede apreciar la prueba de funcionalidad en uno de los módulos que se han establecido en los requerimientos del cliente. Mientras que el restante de las demás pruebas de aceptación se encuentra en el **ANEXO II**.

Tabla 9 Prueba de aceptación - Gestionar bienes de los SPI.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA-006	Identificador historia de Usuario: HU-006
Nombre: Gestionar bienes de los SPI.	
Descripción: Los usuarios con perfil administrador y encarado son capaces de gestionar la información de los bienes y servicios de los SPI.	

Pasos de ejecución:

- Digitar la URL en el navegador *web* de intranet.
- Ingresar el usuario y contraseña con perfil administrador o encargado.
- Filtre la información de un SPI en el formulario de entrada o seleccione la opción Inventario en el menú lateral.

Resultado deseado:

El sistema *web* de la SDH permite que los usuarios puedan asignar un nuevo bien o servicio a un SPI, editar la información.

Evaluación de la prueba

El sistema *web* cumple con la conformidad del cliente al 100%.

Al término de las pruebas de aceptación el sistema *web* se comporta y funciona de acuerdo a los requerimientos del cliente, cumpliendo con las exigencias de este tal y como se ha detallado al inicio de la Recopilación de requisitos.

3.6 *Sprint* 5. Despliegue del sistema *web*

En este *Sprint* se realiza el despliegue a producción del sistema *web* de la SDH en los servidores internos, específicamente en una máquina virtual donde corre el sistema operativo *Windows* y en esta se levanta un servidor Apache Tomcat. El cual puede ser accedido desde la siguiente URL.

<http://192.168.61.28:8080/>

Por último, el certificado expedido por el Ingeniero William Cueva, quien es Analista de Desarrollo de la Secretaría de Derechos Humanos, certifica la plena satisfacción de todos los requerimientos y características del sistema *web* que fueron inicialmente requeridos para el proyecto. En ese sentido, el certificado se encuentra en el **ANEXO II**.

4 CONCLUSIONES

Las conclusiones redactadas a continuación son tomadas en base de la experiencia de todo el tiempo de refactorización del sistema *web*.

- Los requerimientos recopilados contribuyeron positivamente en la clara comprensión de las necesidades del cliente con respecto al sistema *web*, permitiendo acelerar el proceso de refactorización de módulos en todo el sistema *web*.
- Con la reestructuración de la Base de Datos de la SDH, se consiguió la incorporación de tablas necesarias para el proceso de autenticación de usuarios contribuyendo al fortalecimiento de la seguridad y otorgar una capa extra de protección con el uso de *Sprint Security* logrando el resguardo de datos sensibles.
- La reestructuración del sistema *web* conformado por dos componentes (*backend* y *frontend*) en un solo paquete *Maven*, permitió optimizar el código, con la incorporación de nuevas funcionalidades y módulos como: inicio de sesión, generación de informes pdf y excel, modificar unidad, instituciones, usuarios y roles, categorías y modificar usuarios, también favoreció en la optimización de validaciones obligatorias en los diferentes formularios. Además, se ha eliminado errores de lógica de programación mejorando la eficiencia y experiencia de usuario en todo el sistema *web* alineándose en todo instante con los requerimientos que han sido solicitados.
- El uso de *Spring Boot* y *Thymeleaf* ha permitido que la codificación como parte de la reestructuración de todo el sistema sea más ordenado y tenga buenas prácticas de programación y que las pruebas se puedan ejecutar después de cada *Sprint*.
- Las diferentes pruebas sobre el sistema *web* han permitido validar todo el código de los métodos CRUD en los diferentes controladores, así también ha permitido la compatibilidad con varios navegadores *web*, garantizando en todo momento el correcto funcionamiento previamente a ser desplegado a un ambiente de producción.
- El despliegue a producción del sistema *web* como parte de un plan piloto y solamente de uso interno, ha permitido habilitar nuevamente la plataforma como herramienta de uso administrativo, así se ha comprobado el desempeño y buen funcionamiento, alineándose con lo solicitado y abriendo la posibilidad de ser usada en los diferentes SPI a futuro.

5 RECOMENDACIONES

Aquí se presenta las recomendaciones que se pueden dar al realizar un proyecto de refactorización como parte de este Trabajo de Integración Curricular.

- Se recomienda consultar siempre la documentación de *Spring Boot* y *Thymeleaf* ya que al ser fuentes oficiales se puede consultar en todo el transcurso del desarrollo.
- Periódicamente durante el transcurso de todo el proyecto es recomendable tener un respaldo de la Base de datos para ser utilizada al momento de las pruebas y posibles fallas que se puedan presentar.
- Se recomienda aplicar todas las validaciones necesarias para garantizar una excelente experiencia de usuario.
- Se recomienda siempre utilizar herramienta de versionamiento de código como GitHub para almacenar el proyecto en caso de posibles fallas.
- Se recomienda tener presente los permisos necesarios en las máquinas virtuales de un servidor físico para evitar errores de despliegue de cualquier sistema *software*.
- Se recomienda siempre trabajar con las últimas versiones de las herramientas y bibliotecas debido a los conflictos que puede presentar en el transcurso del desarrollo.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] Wikipedia, «Secretaría de Derechos Humanos,» Wikipedia, 29 Agosto 2022. [En línea]. Available: [https://es.wikipedia.org/wiki/Secretar%C3%ADa_de_Derechos_Humanos_\(Ecuador\)](https://es.wikipedia.org/wiki/Secretar%C3%ADa_de_Derechos_Humanos_(Ecuador)). [Último acceso: 25 10 2022].
- [2] SDH, «Servicio de Protección Integral,» SDH, 25 Octubre 2022. [En línea]. Available: <https://www.derechoshumanos.gob.ec/servicio-de-proteccion-integral/>. [Último acceso: 25 Octubre 2022].
- [3] C. americas, «La importancia del control de inventarios .,» Connect americas, 2015. [En línea]. Available: <https://connectamericas.com/es/content/la-importancia-del-control-de-inventarios>. [Último acceso: 22 Mayo 2022].
- [4] T. B. B. Lab, «La importancia de mantener tu página web actualizada.,» he Black Box Lab, 2022. [En línea]. Available: <https://theblackboxlab.com/2022/10/03/importancia-mantener-tu-web-actualizada/#:~:text=Evitamos%20que%20ciertos%20enlaces%20dirijan,p%C3%A1gina%20web%20se%20monitorizar%C3%A1%20constantemente..> [Último acceso: 22 Mayo 2022].
- [5] Aratecna, «QUÉ SON LAS ACTUALIZACIONES DE SOFTWARE Y SU IMPORTANCIA PARA LOS EQUIPOS,» Aratecna, 7 Septiembre 2022. [En línea]. Available: <https://aratecna.es/que-son-las-actualizaciones-de-software-y-su-importancia-para-los-equipos/#:~:text=En%20qu%C3%A9%20consisten%20las%20actualizaciones,malware%20que%20perjudiquen%20el%20sistema..> [Último acceso: 19 Octubre 2022].
- [6] I. J. B. García, « Backend y Fontend,» Servnet, 30 Mazo 2021. [En línea]. Available: <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web>. [Último acceso: 25 Octubre 2022].
- [7] IBM, «¿Qué es el desarrollo de software?,» IBM, 2022. [En línea]. Available: <https://www.ibm.com/es-es/topics/software-development>. [Último acceso: 22 Mayo 2022].
- [8] R. Hat, «¿Qué es la metodología Agil?,» Red Hat, 15 Enero 2020. [En línea]. Available: <https://www.redhat.com/es/devops/what-is-agile-methodology>. [Último acceso: 16 Mayo 2022].
- [9] Y. FERNÁNDEZ, «Xataka API: qué es y para qué sirve,» Xataka, 2019 Agosto 23. [En línea]. Available: <https://www.xataka.com/basics/api-que-sirve>. [Último acceso: 16 Mayo 2022].
- [10] RedHat, «¿Qué es una API de REST?,» RedHat, 8 Mayo 2020. [En línea]. Available: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>. [Último acceso: 23 Mayo 2022].

- [11] Java, «¿Qué es la tecnología Java y por qué la necesito?,» Java, 2022. [En línea]. Available: https://www.java.com/es/download/help/whatis_java.html. [Último acceso: 23 Mayo 2022].
- [12] Thymeleaf, «Thymeleaf,» Thymeleaf, 6 Diciembre 2022. [En línea]. Available: <https://www.thymeleaf.org/>. [Último acceso: 23 Mayo 2023].
- [13] Spring, «Spring,» Spring , 2023. [En línea]. Available: <https://spring.io/>. [Último acceso: 23 Mayo 2023].
- [14] RootStack, «¿Qué es Spring Boot,» RootStack, [En línea]. Available: <https://rootstack.com/es/learning/que-es-spring-boot-y-que-ofrece#:~:text=Se%20utiliza%20para%20crear%20aplicaciones%20web%20spring%20independientes.,es%20menor%2C%20la%20productividad%20aumenta..>. [Último acceso: 19 Octubre 2022].
- [15] IBM, «¿Qué es PostgreSQL?,» IBM, 2023. [En línea]. Available: <https://www.ibm.com/mx-es/topics/postgresql>. [Último acceso: 23 Mayo 2023].
- [16] TestiM, «Pruebas de Front-End: Una Descripción conceptual completa,» TestiM, 12 2020. [En línea]. Available: <https://www.testim.io/blog/front-end-testing-complete-overview/>.
- [17] S. McCombes, «¿Qué es un estudio de caso?,» Scribbr, 14 Septiembre 2022. [En línea]. Available: <https://www.scribbr.com/methodology/case-study/>. [Último acceso: 18 10 2022].
- [18] A. R. Mesa, «OpenWebinars,» 19 Diciembre 2018. [En línea]. Available: <https://openwebinars.net/blog/que-es-un-sprint-scrum/>. [Último acceso: 3 Junio 2022].
- [19] B. A. M. Friends, «Product Owner Que es y cuales son sus funciones,» BE AGILE My Friends, [En línea]. Available: https://beagilemyfriend.com/product-owner/#%C2%BFQue_es_un_Product_Owner. [Último acceso: 3 Junio 2022].
- [20] B. A. m. Friends, «Scrum Master: ¿Qué es y cual es su función?,» BE AGILE My Friends, [En línea]. Available: <https://beagilemyfriend.com/scrum-master/>. [Último acceso: 3 Junio 2022].
- [21] p. agiles.org, «Equipo de desarrollo (Development Team),» proyectos agiles.org, [En línea]. Available: <https://proyectosagiles.org/equipo-team/>. [Último acceso: 22 Junio 2022].
- [22] Viewnext, «Artefactos Scrum ¿Qué son y para que sirven?,» Viewnext, [En línea]. Available: <https://www.viewnext.com/artefactos-scrum/>. [Último acceso: 3 Junio 2022].
- [23] E. Ledesma, Proyectum, 03 Septiembre 2020. [En línea]. Available: <https://www.proyectum.com/sistema/blog/scrum-como-escribir-historias-de-usuarios-sin-morir-en-el-intento/>. [Último acceso: 27 Junio 2023].

- [24] J. Roche, «Artefactos Scrum: las 3 herramientas clave de gestión,» Deloitte, 2022. [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/artefactos-scrum.html>. [Último acceso: 06 Junio 2022].
- [25] Wikipedia, «Diseño de interfaz de usuario,» Wikipedia, 21 Marzo 2023. [En línea]. Available: https://es.wikipedia.org/wiki/Dise%C3%B1o_de_interfaz_de_usuario. [Último acceso: 27 Junio 2023].
- [26] Balsamiq, «Balsamiq Mockups,» Balsamiq, 2008-2022. [En línea]. Available: <https://balsamiq.com/wireframes/>. [Último acceso: 07 Junio 2022].
- [27] U. d. Alicante, «Modelo vista controlador (MVC),» Universidad de Alicante, 1996-2022. [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Último acceso: 06 Junio 2022].
- [28] Spring, «Spring Tools 4,» VMware Tanzu, 27 Junio 2023. [En línea]. Available: <https://spring.io/tools>. [Último acceso: 27 Junio 2023].
- [29] P. Bert, «Funcionalidades principales de PowerDesigner,» PowerDesigner, 27 Junio 2023. [En línea]. Available: <https://www.powerdesigner.biz/ES/powerdesigner/powerdesigner-features.html>. [Último acceso: 27 Junio 2023].
- [30] Bootstrap, «Bootstrap,» Bootstrap, 27 Junio 2023. [En línea]. Available: <https://getbootstrap.com/>. [Último acceso: 27 Junio 2023].
- [31] Baeldung, «Descripción general de las herramientas de desarrollo de Spring Boot,» baeldung, 3 Junio 2021. [En línea]. Available: <https://www.baeldung.com/spring-boot-devtools>. [Último acceso: 27 Junio 2023].
- [32] V. Tanzu, «Servicios Spring Web,» VMware Tanzu, 04 Junioo 2023. [En línea]. Available: <https://spring.io/projects/spring-ws>. [Último acceso: 28 Junio 2023].
- [33] Baeldung, «Todas las guías de datos de Spring,» Baeldung, 04 Abril 2023. [En línea]. Available: <https://www.baeldung.com/spring-data>. [Último acceso: 28 Junio 2023].
- [34] Baeldung, «Seguridad con Spring,» Baeldung, 29 Diciembre 2022. [En línea]. Available: <https://www.baeldung.com/security-spring>. [Último acceso: 28 Junio 2023].
- [35] V. TAnzu, «Configuración de base de datos,» VMware TAnzu, 02 Junio 2023. [En línea]. Available: <https://docs.spring.io/spring-cloud-dataflow/docs/1.1.2.RELEASE/reference/html/configuration-rdbms.html>. [Último acceso: 28 Junio 2023].
- [36] Eduesqui, «Aprende a crear pruebas unitarias con JUnit y Mockito en 15 minutos,» FM Funciona en mi máquina, 11 Marzo 2021. [En línea]. Available: <https://funcionaenmimaquina.com/aprende-a-crear-pruebas-unitarias-con-junit-y-mockito-en-15-minutos/>. [Último acceso: 28 Junio 2023].

- [37] Alexandria, «Qué es un test multi navegador: definición y ejecución,» Comparium, 2023. [En línea]. Available: <https://www.electronic.us/es/cross-browser-testing-tool/what-is-cross-browser-testing/#:~:text=Las%20pruebas%20de%20compatibilidad%20con%20los%20navegadores%20son%20el%20proceso,detectar%20y%20solucionar%20los%20problemas..> [Último acceso: 28 Junio 2023].
- [38] ChatGpt, «ChatGpt,» ChatGpt, 14 Agosto 2023. [En línea]. Available: <https://chat.openai.com>. [Último acceso: 14 Agosto 2023].